Studies in Computational Intelligence 1090

Ahmad Taher Azar
Ibraheem Kasim Ibraheem
Amjad Jaleel Humaidi *Editors*

Mobile Robot: Motion Control and Path Planning

Springer

Studies in Computational Intelligence

Volume 1090

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

Ahmad Taher Azar · Ibraheem Kasim Ibraheem ·
Amjad Jaleel Humaidi
Editors

Mobile Robot: Motion Control and Path Planning

Springer

Editors
Ahmad Taher Azar
College of Computer and Information Sciences
Prince Sultan University
Riyadh, Saudi Arabia

Automated Systems and Soft Computing Lab (ASSCL)
Prince Sultan University
Riyadh, Saudi Arabia

Faculty of Computers and Artificial Intelligence
Benha University
Benha, Egypt

Amjad Jaleel Humaidi
Department of Control and Systems Engineering,
University of Technology
Baghdad, Iraq

Ibraheem Kasim Ibraheem
Department of Electrical Engineering,
College of Engineering
University of Baghdad
Baghdad, Iraq

Department of Computer Engineering Techniques
Al-Rasheed University College
Baghdad, Iraq

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-031-26563-1 ISBN 978-3-031-26564-8 (eBook)
https://doi.org/10.1007/978-3-031-26564-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Mobile robots have been an intriguing study topic for the last 30 years. Ground vehicles, underwater vehicles, and airborne vehicles are the three types. They may move on land, on the surface of bodies of water, underwater, and in the air. Fixed-base robotic manipulators, on the other hand, are widespread in industrial factories and manufacturing facilities such as automotive and airplane assembly, welding, and spray painting. While mobile robots are more typically encountered in mine excavation, nuclear site monitoring, warehouses, military activities, material delivery, man-machine interactions for disabled individuals, and, more lately, hospitals and restaurants. It is evident from this vast variety of mobile robot applications that the technology at hand is exceedingly complicated and multidisciplinary. A motion control system that allows the mobile robot to move through locomotion and steering must be designed in order for it to perform correctly in a certain application. Because of this regulated motion, these complicated activities can be accomplished successfully. Sensory tasks, such as positioning and data capture using a camera, are possible. In an unobstructed environment, motion control of a mobile robot can be performed by directing motion from a starting point to an endpoint or by reference trajectory tracking. While the mobile robot performs its responsibilities, its supervisor must keep track of its location and orientation. Only then can sensed data be reported accurately and completely exploited. As a result, route planning is essential. When directing a mobile robot to a certain location, path planning is essential. Another reason for path planning is because robots often operate in surroundings with a variety of limits, impediments, and demands, all of which contribute to the definition of the desired path that brings the robots to the objective point. Previously, the bulk of books decided to present either motion control or path planning topics individually.

Objectives of the Book

This book is unique in that it combines both topics into a single coherent monograph. The purpose of this book is to provide in-depth coverage of the most modern motion control approaches, including linear and nonlinear control techniques, as well as the application of swarm optimization and its integration with control design for motion control. Furthermore, the volume includes information on numerous strategies utilized in mobile robot path planning. This book is an invaluable resource for academic and industrial researchers studying nonlinear systems in electrical engineering. It is suitable as a textbook for first-year graduate courses in computer science, electrical, electronic, systems, and mechanical engineering. Furthermore, in the last year of undergraduate study in control systems, electrical devices, scientific modeling, computational science, and many other subjects, it can accommodate a mobile robot navigation course.

Organization of the Book

This well-structured book consists of 18 full chapters.

Book Features

- The book chapters deal with the recent research problems in the areas of robotics, control systems, dynamical modeling, and optimization.
- The book chapters present advanced techniques of motion control and path planning of mobile robots.
- The book chapters contain a good literature survey with a long list of references.
- The book chapters are well written with a good exposition of the research problem, methodology, block diagrams, and mathematical techniques.
- The book chapters are lucidly illustrated with numerical examples and simulations.
- The book chapters discuss details of applications and future research areas.

Audience

The book is primarily meant for researchers from academia and industry, who are working on in the research areas such as robotics engineering, control engineering, mechatronic engineering, biomedical engineering, medical informatics, computer science, and data analytics. The book can also be used at the graduate or advanced undergraduate level and many others.

Acknowledgements

As the editors, we hope that the chapters in this well-structured book will stimulate further research in robotics, control systems, dynamical modeling, and optimization, and utilize them in real-world applications.

We hope sincerely that this book, covering so many different topics, will be very useful for all readers.

We would like to thank all the reviewers for their diligence in reviewing the chapters.

Special thanks go to Springer, especially the book Editorial team.

Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

Riyadh, Saudi Arabia/Benha, Egypt
Prof. Ahmad Taher Azar
Ph.D., IEEE senior Member, ISA Member
aazar@psu.edu.sa
ahmad.azar@fci.bu.edu.eg
ahmad_t_azar@ieee.org

Baghdad, Iraq
Prof. Ibraheem Kasim Ibraheem
ibraheemki@coeng.uobaghdad.edu.iq

Baghdad, Iraq
Prof. Amjad Jaleel Humaidi
amjad.j.humaidi@uotechnology.edu.iq

Contents

Leader-Follower Formation Control of Mobile Robots Based on Simplified Super-Twisting Algorithm . . . 1
Susana V. Gutiérrez-M., Jesús De León-Morales, and Oscar S. Salas-Peña

Deep Reinforcement Learning Applied to Multi-agent Informative Path Planning in Environmental Missions . . . 31
Samuel Yanes Luis, Manuel Perales Esteve, Daniel Gutiérrez Reina, and Sergio Toral Marín

Analysis of Computer Vision-Based Techniques for the Recognition of Landing Platforms for UAVs . . . 63
J. A. García-Pulido and G. Pajares

Monitoring Peak Pollution Points of Water Resources with Autonomous Surface Vehicles Using a PSO-Based Informative Path Planner . . . 93
Micaela Jara Ten Kathen, Princy Johnson, Isabel Jurado Flores, and Daniel Gutiérrez Reina

Robot-Based Medicine. Robots in Medicine: Mobile Robots Versus Mobile Decision, Necessity Versus Possibility and Future Challenges . . . 127
Zbigniew Nawrat and Dariusz Krawczyk

Event-Based Robust Control Techniques for Wheel-Based Robots Under Cyber-Attack and Dynamic Quantizer . . . 163
Mobin Saeedi, Jafar Zarei, Mehrdad Saif, and Allahyar Montazeri

Path Optimization and Multi-level Path Planning for the Steam Field Navigation Algorithm . . . 197
Hussein M. Fawzy, Hisham M. El-Sherif, and Gerd Baumann

Modeling and Simulation of Quadcopter Using Self-tuning Fuzzy-PI Controller 231
Arezki Fekik, Ahmad Taher Azar, Mohamed Lamine Hamida, Hakim Denoun, Sabrina Mohandsaidi, Amar Bousbaine, Nashwa Ahmad Kamal, Ibraheem Kasim Ibraheem, Amjad J. Humaidi, Ammar K. Al Mhdawi, and Alaa Khamis

Using an Interactive Theorem Prover for Formally Analyzing the Dynamics of the Unmanned Aerial Vehicles 253
Adnan Rashid, Osman Hasan, and Sa'ed Abed

Adaptive Fault-Tolerant Control Design for Multi-linked Two-Wheel Drive Mobile Robots 283
Ayad Al-Dujaili, Vincent Cocquempot, Maan El Badaoui El Najjar, Daniel Pereira, and Amjad Humaidi

Design and Implementation of a Robust 6-DOF Quadrotor Controller Based on Kalman Filter for Position Control 331
Amar Bousbaine, Abdelkader Fareha, Ajay K. Josaph, Arezki Fekik, Ahmad Taher Azar, Riad Moualek, Nabil Benyahia, Nacereddine Benamrouche, Nashwa Ahmad Kamal, Ammar K. Al Mhdawi, Amjad J. Humaidi, and Ibraheem Kasim Ibraheem

Wireless Sensor Network Based Mobile Robot Applications 365
Muayad Sadik Croock

Fault Diagnosis and Fault Tolerant Control for *n*-Linked Two Wheel Drive Mobile Robots 403
Ayad Al-Dujaili, Vincent Cocquempot, Maan E. EI Najjar, Daniel Pereira, and Amjad Humaidi

Fractional Order Extended State Observer Enhances the Performance of Controlled Tri-copter UAV Based on Active Disturbance Rejection Control 439
Alaq F. Hasan, Amjad J. Humaidi, Abdulkareem Sh. Mahdi Al-Obaidi, Ahmad Taher Azar, Ibraheem Kasim Ibraheem, Ayad Q. Al-Dujaili, Ammar K. Al-Mhdawi, and Farah Ayad Abdulmajeed

Robust Adaptive Sliding Mode Controllers Design for a Non-holonomic Mobile Robot 489
Shibly Ahmad Al-Samarrai, Yasir Khudhair Al-Nadawi, Taif Ghadban Hama, and Taha Adel Al-Gadery

Algorithmic Design of Block Backstepping Motion and Stabilization Control for Segway Mobile Robot 557
Amjad J. Humaidi, Mustafa R. Hameed, Alaq F. Hasan, Abdulkareem Sh. Mahdi Al-Obaidi, Ahmad Taher Azar, Ibraheem Kasim Ibraheem, Ayad Q. Al-Dujaili, Ammar K. Al Mhdawi, and Farah Ayad Abdulmajeed

Extremum Seeking Based PID Control of Quadrotor System 609
Safanah M. Raafat, Abdullah N. Muhsen, and Ahmed A. A. Alawsai

Wireless Sensor Network for Robot Navigation 643
Ekhlas Kadum Hamza, Kawther Dawood Salman, and Shahad Nafea Jaafar

Leader-Follower Formation Control of Mobile Robots Based on Simplified Super-Twisting Algorithm

Susana V. Gutiérrez-M., Jesús De León-Morales, and Oscar S. Salas-Peña

Abstract Formation control involves groups of robots moving together while holding a predefined geometric configuration. The formation control problem for a group of differential mobile robots by Leader-Follower (L-F) approach is addressed in this work. The objective of control is to drive follower to track the trajectory of the leader with a specific distance between the robots. With this aim, a simplified super-twisting controller is designed based on a Lyapunov approach, which simplifies the analysis of stability in closed loop and reduces gain tuning effort. Furthermore, proposed methodology increases robustness against disturbances and unmodeled dynamics, while exact knowledge of all parameters of the system is not required to be known, while chattering effect is attenuated. Finally, numerical and experimental results are provided to illustrate feasibility and performance of proposed methodology.

Keywords Mobile robots · Super twisting algorithm leader-follower

1 Introduction

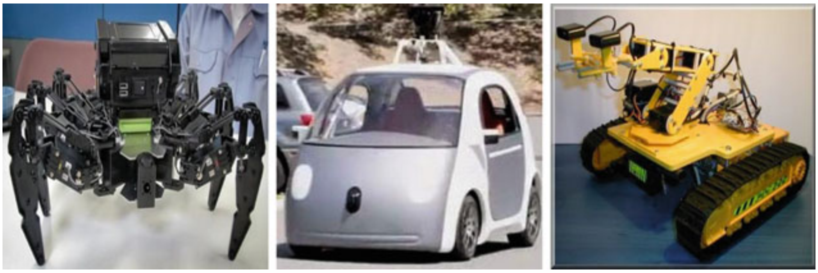
Mobile robotics has a very important role to play in several areas and applications such as industrial, military, space exploration, entertainment, medical care, security, among many other applications [1, 28]. According to the environment in which they perform, mobile robots can be classified into aerial, underwater and ground. Moreover, mobile robots are characterised by their capacity to move autonomously in an unknown or partially known environment. This work focuses on the trajectory tracking control of ground mobile robots. The basics of ground mobile robots is addressed in next section.

S. V. Gutiérrez-M. (✉) · J. De León-Morales · O. S. Salas-Peña
Faculty of Mechanical and Electrical Engineering, Department of Electrical Engineering, Autonomous University of Nuevo León, San Nicolás de Los Garza, Nuevo León, Mexico
e-mail: svgutierrezmr@gmail.com

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_1

(a) Space exploration. (b) First aid care. (c) Agriculture.

Fig. 1 Ground mobile robots applications in different domains



(a) Legged robots (b) wheeled robots (c) tracked robots

Fig. 2 Locomotion systems for ground mobile robots

1.1 Ground Mobile Robots

Ground mobile robots are notable for numerous applications fields such as medicine, homeland security, surveillance, demining, reconnaissance in dangerous situations and agriculture; as can be seen in the Fig. 1.

Ground mobile robots traction and steering systems election depend on ground characteristics and requirements of the task, such as speed and manoeuvrability. Then, the required precision and velocity by a mobile robot to reach its destination, implies a traction and steering systems that can be provide the robot with reliable manoeuvrability [3, 31]. In the case of ground mobile robots, different kinds of locomotion systems can be chosen; for instance legged robots [4], wheeled robots [32] and tracked robots [29] (see Fig. 2).

Despite extensive studies on the mobility of legged and tracked robots, the most significant development has been in wheeled locomotion. The main advantage of using wheels as a means of locomotion is that they are easy to construct, and furthermore it is only necessary to supply power to the drive wheel axle.

Moreover, wheeled ground mobile robots are vehicles capable of autonomous movement (without human intervention), as they are capable to be equipped and driven by a computer for controlling the actuators. On the other hand, depending

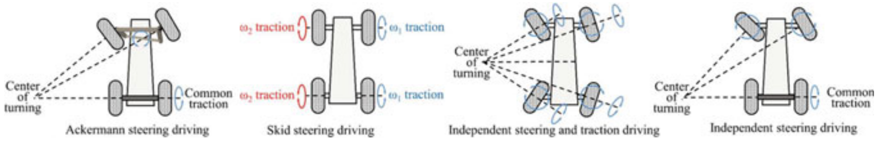


Fig. 3 Wheel arrangement configurations

on the arrangement of the wheels, several interesting configurations can be chosen [3, 31]; for instance, Ackerman configuration, tricycle, skid steer, synchronous, omnidirectional traction and differential traction (see Fig. 3).

According to objective, motion control of ground mobile robots can be classified into four fundamental tasks:

- Localisation
- Trajectory planning
- Trajectory tracking
- Obstacle avoidance

On the other hand, it is important to notice that groups of robots present benefits over single-robot systems [25]. The most important issues for developing solutions using multi-robot systems are the following:

- Tasks complexity excessive for a single robot to carry out.
- In general, it is better to build multiple simple robots than having a single forceful robot.
- Task can be addressed by a distributed approach.
- Problems can be solved faster through parallelism.
- Redundancy through robot groups increases robustness.

The use of multi-robots depends on the requirements of the task and, in addition, on the available capabilities of the robots' sensors and actuators. However, the coordination of robots implies a tracking control to perform the task efficiently. Several methodologies for solving formation problem have been addressed in the literature, which can be divided into two approaches as follows:

- Centralized control. An agent is defined to control and monitor the rest of the robots, using a central processor [7].
- Decentralized control. Individual control for each robot using its relative position with respect to its neighbors [18].

Regarding formation and tracking problems, several methodologies have been proposed to deal with; for instance, Behaviour-based approach [15, 35], Virtual structure [22], Leader-follower (L-F) [6] and Graph theory [2].

Regarding the behavior-based approach, it addresses the weighing of relative importance of the task assigned to each robot, such as obstacle avoidance, collision avoidance, moving-to-target and keeping-formation, to name a few. Furthermore, it is practical to drive a system of multiple mobile robots over an unknown dynamical environment. However, describing the dynamics of the group and also analyzing stability of the entire system is not a trivial task [23].

On the other hand, a single virtual rigid structure is considered for algorithms based on virtual structure technique [22], so that the robotic system formation behaves similarly to a physical object. Desired tracking references are not given to each robot, whole formation is treated as a single entity. Main advantage of this methodology is that the coordinated behaviour of formation becomes simpler to describe. Although, it is difficult to find some formation applications and it is necessary a large inter-robot communication band width.

Regarding L-F technique, a robot is assigned as a leader that drives through a given path, while the rest of the robots in the group perform as followers that must maintain a predefined distance from the leader [6]. In fact, once the leader receives the movement command, the desired distance between followers and leader is achieved by selecting an appropriate local control law based on relative position dynamics. As a result, formation stability is ensured, i.e. the desired formation is achieved and maintained throughout the group. Then, formation control issue becomes an extension of trajectory-following problem, and thus the L-F approach stands out for its simplicity and scalability. However, the main drawback of the L-F approach is a strong dependence on the leader to achieve the goal, and excessive dependency on a single robot may be undesirable, especially under adverse conditions [11].

Another point of view to address the formation control problem is given by graph theory, where the detection or communication information link between robots is considered as an edge and each robot is considered as a node. This approach facilitates the representation of the formation thanks to graph theory and its well-known theoretical results. The drawback is that it is difficult to take into account the limitation of the actual configuration of the robot [2].

1.2 Problem Statement

Now, the problem of tracking control with multiple mobile robots based on the L-F approach will be addressed.

Multi-robot system modeling contains plenty of uncertainties and disturbances, for this reason, several research tracking control schemes combined with nonlinear control methodologies have been proposed, for instance, feedback linearization [36], backstepping [21], and sliding-mode control [9, 37].

Sliding mode control represents a successful strategy for tracking and formation control problems, due to its robustness to external disturbances, parameter variation, and uncertainties [27, 33]. Nevertheless, main disadvantage of this approach is the "*chattering*" phenomenon, caused by the high frequency switching of the control law [26]. Commonly, efforts to avoid this phenomenon have focused on three methodologies: high-order sliding mode control [20], observer-based solution [34], and continuous approximation of the relay [19].

Super-twisting control represents a remarkable alternative among the high-order sliding mode algorithms, due to its properties such as robustness and finite-time convergence. Although it is a second-order sliding mode algorithm (SOSMC), it requires only the sliding variable (and not the time derivative of the variable sliding, in contrary of most of SOSMC). Furthermore, the super-twisting algorithm allows finite-time convergence of a sliding surface [5, 10]. For instance, the convergence time estimation is reported in [24], based on a strong Lyapunov function. Nevertheless, the structure of aforementioned function must satisfy several constraints that depend on control gains and bounds on uncertainties; besides, the choice of a symmetric, positive definite matrix.

Furthermore, gain selection for the super-twisting algorithm is a time-consuming task with a trade-off between chattering attenuation and dynamic system performance (see [26]). With this aim, a strict Lyapunov function that achieves sufficient stability conditions with stable parameter ranges has been reported [30], in order to guarantee the proper choice of gains and convergence of the system with chattering attenuation. Notice that an inadequate gains selection that satisfy these constraints would guarantee a faster convergence of the system, albeit without chattering reduction; or a slower convergence of the system, although reducing chattering.

1.3 Main Contribution

The main contributions of this study are following:

- The design of a super-twisting based controller for a group of differential robots, using a parametrization to simplify tuning effort.
- The design of a sliding mode controller is based on the construction of a strong Lyapunov function.
- Finite-time convergence analysis to the equilibrium point of the closed-loop system is addressed.
- With the aim of preserving the robustness properties while reducing control effort, proposed control design relies on a L-F strategy using a super twisting control, which drive the trajectory to track a desired reference separated to a specific distance between leader and follower.

1.4 Chapter Structure

The layout is organized as follows: Mathematical model of a differential mobile robot is described in Sect. 2. The L-F control design for solving tracking and formation problem using a super-twisting control is addressed in Sect. 3. Then, tracking error tends to zero under parametric uncertainties and external perturbations by means of obtaining sufficient conditions to ensure the convergence in finite time. In Sects. 4 and 5, numerical and experimental validations of the methodology are presented for the tracking of trajectories in L-F systems, respectively. Section 6 are presented conclusions and future work.

2 Differential Mobile Robot Model

A system with N differential mobile robots (see Fig. 4) described by the following mathematical model

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \end{pmatrix} = \begin{pmatrix} v_i \cos\theta_i - \omega_i \sin\theta_i \\ v_i \sin\theta_i + \omega_i \cos\theta_i \\ \omega_i \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0\,0 \\ 0\,0 \\ 0\,0 \\ 1\,0 \\ 0\,1 \end{pmatrix} \begin{pmatrix} u_{1i} \\ u_{2i} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \rho_{1i} \\ \rho_{2i} \end{pmatrix}; i = 1, \ldots, N. \quad (1)$$

the Cartesian space correspond to the position of the robot given by x_i and y_i, the orientation angle is represented θ_i, the linear velocity and the angular velocity described by v_i ω_i respectively, the control inputs of agent i are u_{1i} and u_{2i}, ρ_{1i} and ρ_{2i} are uncertain terms; for $i = 1, \ldots, N$.

Furthermore, we introduce the following assumption about the differential Mobile robot model

Assumption 1 Inertia moments of motor, gear and wheel are neglected. Furthermore, mass and inertia of the robots represent only uncertain parameters.

Let be the inputs control vector as $^\circ_i = (u_{1i}, u_{2i})^T$, the states vector as $\mathbf{X}_i = (x_i, y_i, \theta_i, v_i, w_i)^T$, the perturbations vector as $\Delta_i = (0, 0, 0, \rho_{1i}, \rho_{2i})^T$ and the output as $\mathbf{Y}_i = (x_i, y_i)^T$ [14]; for $i = 1, \ldots, N$. Then, system (1) can be written in this compact form

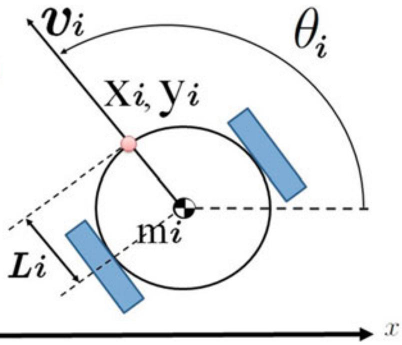


Fig. 4 Differential mobile robot diagram

$$\dot{\mathbf{X}}_i = f(\mathbf{X}_i) + g(\mathbf{X}_i)^{\circ}{}_i + {}_i \tag{2}$$

$$\mathbf{Y}_i = \mathbf{C}\mathbf{X}_i \tag{3}$$

where the vector fields $f(\mathbf{X}_i)$ and $g(\mathbf{X}_i)$ are given by

$$f(\mathbf{X}_i) = \begin{pmatrix} v_i \cos\theta_i - \omega_i \sin\theta_i \\ v_i \sin\theta_i + \omega_i \cos\theta_i \\ \omega_i \\ 0 \\ 0 \end{pmatrix}, \quad g(\mathbf{X}_i) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

with $f(\mathbf{X}_i)$ and $g(\mathbf{X}_i)$ are smooth.

Furthermore, the center of mass and the geometric center of each mobile robot are assumed to coincide.

Now, introducing the following map $\varphi(\xi_i)$

$$\varphi(\mathbf{X}_i) = \begin{pmatrix} h_{1i}(\mathbf{X}_i) \\ h_{2i}(\mathbf{X}_i) \\ L_f h_{1i}(\mathbf{X}_i) \\ L_f h_{2i}(\mathbf{X}_i) \\ \xi(X_i) \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ v_i \cos\theta_i - \omega_i \sin\theta_i \\ v_i \sin\theta_i + \omega_i \cos\theta_i \\ \theta_i \end{pmatrix} \tag{4}$$

It easy to see that the Jacobian of $\varphi(\mathbf{X}_i)$ is non-singular, then form the theorem of implicit function (see [16]) $\varphi(\mathbf{X}_i)$ is a diffeomorphism. Furthermore, since the system (1) has two inputs, it is necessary to determine the characteristic indexes in order to know if the system (1) can be feedback linearizable. In the case, the characteristic indexes are {2, 2}. Then, the system (3) is partially state feedback linearizable [14].

Now, introducing the following change of coordinates,

$$\begin{pmatrix} \varsigma_{1i} \\ \varsigma_{2i} \\ \varsigma_{3i} \\ \varsigma_{4i} \\ \xi(X_i) \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ v_i \cos\theta_i - \omega_i \sin\theta_i \\ v_i \sin\theta_i + \omega_i \cos\theta_i \\ \theta_i \end{pmatrix} \tag{5}$$

it follows that system (3) can be expressed as follows

$$\dot{\varsigma}_{1i} = \varsigma_{3i} \tag{6}$$

$$\dot{\varsigma}_{2i} = \varsigma_{4i} \tag{7}$$

$$\dot{\varsigma}_{3i} = (u_{1i} + \rho_{1i})\cos\xi_i - (u_{2i} + \rho_{2i})\sin\xi_i - \omega_i\varsigma_{4i} \tag{8}$$

$$\dot{\varsigma}_{4i} = (u_{1i} + \rho_{1i})\sin\xi_i + (u_{2i} + \rho_{2i})\cos\xi_i + \omega_i\varsigma_{3i} \tag{9}$$

$$\dot{\xi}_i = -u_{1i}\sin(\xi_i) + u_{2i}\cos(\xi_i) \tag{10}$$

where the internal dynamic is given by

$$\dot{\xi}_i = -u_{1i}\sin(\xi_i) + u_{2i}\cos(\xi_i) \tag{11}$$

Notice that system (11) is bounded input bounded state [8], and since its zero dynamic ($\dot{\xi}_i = \dot{\theta}_i = \omega_i$) is stable, then the system (10) is weakly minimum phase [16].

Applying the following partial feedback linearizable

$$\begin{pmatrix} u_{1i} \\ u_{2i} \end{pmatrix} = \begin{pmatrix} \cos\xi_i & \sin\xi_i \\ -\sin\xi_i & \cos\xi_i \end{pmatrix} \begin{pmatrix} \mathrm{v}_{1i} + \omega_i\varsigma_{4i} + \gamma_{1i} \\ \mathrm{v}_{2i} - \omega_i\varsigma_{3i} + \gamma_{2i} \end{pmatrix} \tag{12}$$

where $\gamma_{1i} = -\rho_{1i}\cos\xi_i + \rho_{2i}\sin\xi_i$, and $\gamma_{2i} = -\rho_{1i}\sin\xi_i - \rho_{2i}\cos\xi_i$ are the uncertain terms, and v_{1i} and v_{2i} are new input, which will be defined in leader-follower tracking control.

Then, system (10) in closed loop with the feedback linearized control (12) can be expressed as

$$\dot{\mathbf{\Sigma}}_i = \mathbf{A}\mathbf{\Sigma}_i + \mathbf{B}(\mathbf{v}_i + \gamma_i), \quad i = 1, 2, \ldots, N; \tag{13}$$

$$\mathbf{A} = \begin{pmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0_{2\times2} \\ I_{2\times2} \end{pmatrix}$$

where $\mathbf{\Sigma}_i = (\varsigma_{1i}, \varsigma_{2i}, \varsigma_{3i}, \varsigma_{4i})^T$, $\mathbf{v}_i = (\mathrm{v}_{1i}, \mathrm{v}_{2i})^T$ and $\boldsymbol{\gamma}_i = (\gamma_{1i},\ \gamma_{2i})^T$. Then, driving differential robot formations could be designed of robust controller in the presence of disturbances and unmodeled dynamics.

Assumption 2 Perturbation term γ_i, and $\dot{\gamma}_i$, i=1,2, …,N; are bounded and theirs bounds are known, i.e. there exist $D_1 > 0$ and $D_2 > 0$ known such that $||\gamma_i|| \leq D_1$ and $||\dot{\gamma}||_i \leq D_2$.

3 Leader-Follower (L-F) Tracking Control

In this section, a L-F based control is proposed to achieve the trajectory tracking of mobile robots.

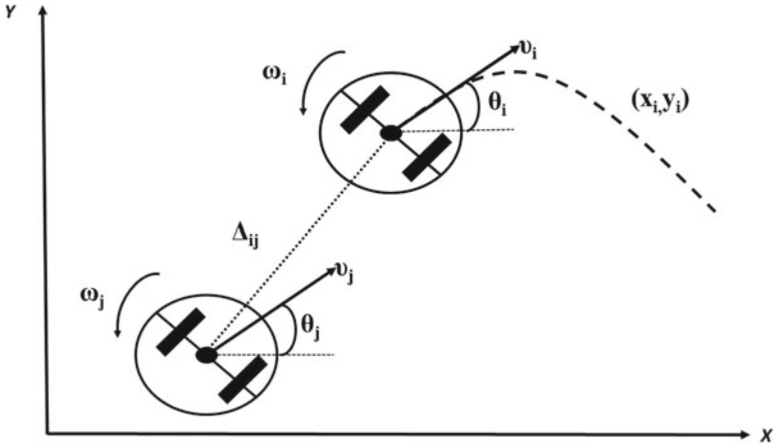


Fig. 5 L-F unicycle robot scheme

Consider system

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}_i &= \boldsymbol{A}\boldsymbol{\Sigma}_i + \mathbf{B}(\mathbf{v}_i + \gamma_i)\\ \mathbf{y}_i &= \mathcal{C}\boldsymbol{\Sigma}_i\end{aligned}$$

where $\mathcal{C} = \begin{pmatrix}1\ 0\ 0\ 0\\0\ 1\ 0\ 0\end{pmatrix}$, v_i is the control input. As shown in Fig. 5, $(\varDelta_{ij})$ represents the separation of the follower with the leader to achieve the formation, and the leader follows the desired trajectory (x_{refi}, y_{refi}).

Defining the tracking and formation error as

$$\mathbf{e}_j = \mathbf{y}_j - \varDelta_{ij} - \mathbf{y}_{refi}, \;\; i, j = 1, \ldots, N, \;\; j \neq i \tag{14}$$

where $\mathbf{y}_{refi}$ is a reference trajectory being tracked by the leader of the formation.

Defining the sliding surface as follows

$$\mathbf{S}_j = \varLambda_j(\mathbf{y}_j - \varDelta_{i,j} - \mathbf{y}_{refi}) + (\dot{\mathbf{y}}_j - \dot{\mathbf{y}}_{refi}) \tag{15}$$

where $\mathbf{S}_j = [S_{1j} \;\; S_{2j}]^T$, $\varLambda_j$ is a positive constant. Taking the time derivative of the sliding surface with respect to time, it follows that

$$\dot{\mathbf{S}}_j = \varLambda_j(\dot{\mathbf{y}}_j - \dot{\mathbf{y}}_{refi}) + \mathrm{v}_j + \gamma_j - \ddot{\mathbf{y}}_{refi} \tag{16}$$

Now, choosing the control input v_j in (13) as follows

$$\mathrm{v}_j = -\varLambda_j(\dot{\mathbf{y}}_j - \dot{\mathbf{y}}_{refi}) + \ddot{\mathbf{y}}_{refi} - K_{1j}\varPi_j - K_{2j}\varGamma_j \tag{17}$$

where

$$\varPi_j = \begin{pmatrix} |S_{1j}|^{1/2} sign(S_{1j}) \\ |S_{2j}|^{1/2} sign(S_{2j}) \end{pmatrix};\ \varGamma_j = \begin{pmatrix} \int_0^t sign(S_{1j}) d\tau \\ \int_0^t sign(S_{2j}) d\tau \end{pmatrix}$$

and $K_{1j},\ K_{2j}$ are the gains of the control v_j, which are parametrized, in terms of the parameter $\varUpsilon_j$, as $K_{1j} = 2\varUpsilon_j I_{2\times2},\ K_{2j} = \dfrac{\varUpsilon_j^2}{2} I_{2\times2}$, and $\varUpsilon_j > 0$.

Remark 1 Notice that only one parameter ($\varUpsilon_j$) is required as gain, which simplifies the analysis, implementation and tuning of the control.

Substituting the control (17) into (16), we obtain

$$\dot{\mathbf{S}}_j = -K_{1j}\varPi_j - K_{2j}\varGamma_j + \gamma_j \tag{18}$$

Introducing the following change of variable

$$\zeta_{1j} = \mathbf{S}_j \tag{19}$$

$$\zeta_{2j} = -\mathbf{K}_{2j}\varGamma_j + \gamma_j \tag{20}$$

whose dynamics is given by

$$\dot{\zeta}_{1j} = -\mathbf{K}_{1j}\varPi_j + \zeta_{2j} \tag{21}$$

$$\dot{\zeta}_{2j} = -\mathbf{K}_{2j} sign(\boldsymbol{\imath}_{1j}) + \dot{\gamma}_j \tag{22}$$

Next, consider the following change of coordinates

$$\begin{aligned} \xi_{1j} &= \varPi_j \\ \xi_{2j} &= \zeta_{2j} \end{aligned} \tag{23}$$

the dynamics of the system (22), in terms of the new change of coordinates, is represented in the following compact form as follows

$$\dot{\xi}_j = \frac{1}{2|\xi_{1j}|} \begin{pmatrix} -K_{1j} & I_{2\times2} \\ -K_{2j} & 0_{2\times2} \end{pmatrix} \xi_j + \begin{pmatrix} 0_{2\times2} \\ I_{2\times2} \end{pmatrix} \dot{\gamma}_j \tag{24}$$

where $\xi_j = \begin{bmatrix} \xi_{1j} & \xi_{2j} \end{bmatrix}^T$. System (24) can be represented, in a compact form, as follows

$$\dot{\xi}_j = \frac{1}{2|\xi_{1i}|} \{[\mathbf{A} - \mathbf{S}_{\varUpsilon_j}^{-1}\mathbf{C}^T C]\xi_j + \gamma_j\} \tag{25}$$

where

$$\mathbf{A} = \begin{pmatrix} 0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} \end{pmatrix}, \quad \mathbf{S}_{\Upsilon_j}^{-1}\mathbf{C}^T C = \begin{pmatrix} 2\Upsilon_j I_{2\times 2} & 0_{2\times 2} \\ \Upsilon_j^2 I_{2\times 2} & 0_{2\times 2} \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} I_{2\times 2} & 0_{2\times 2} \end{pmatrix}, \quad \Phi_j = \begin{pmatrix} 0_{2x2} \\ I_{2x2} \end{pmatrix} 2|\xi_{1i}|\dot{\gamma}_j$$

where Υ_j is an arbitrary positive constant, and $\mathbf{S}_{\Upsilon_j}$ is the solution of the following algebraic matrix equation

$$\Upsilon_j S_{\Upsilon_j} + A^T S_{\Upsilon_j} + S_{\Upsilon_j} A - B^T B = 0 \tag{26}$$

Assumption 3 The uncertain term Φ_j, from Eq. (25), satisfies the following inequality $||\Phi_j|| \leq \eta_j ||\xi_j||$, for $\eta_j > 0$.

In the following proposition is formulated the main result.

Proposition 1 *Consider system (13) in close loop under the action of the controller (17). Furthermore, the Assumptions 1 and 2 are satisfied. Then, the states of closed-loop system (22) converge to tracking and formation error which tends to zero in finite time in presence of disturbances.*

Proof Considering the Lyapunov function candidate

$$V_{(\xi_j)} = \xi_j^T S_{\Upsilon_j} \xi_j \tag{27}$$

whose the time derivative is given by

$$\dot{V}_{(\xi_j)} = \frac{1}{2|\xi_{1j}|} \left\{ -\Upsilon_j \xi_j^T S_{\Upsilon_j} \xi_j + 2\mathbf{X}_j^T S_{\Upsilon_j} \Phi_j \right\} \tag{28}$$

Taking the norm of the nonlinear term $2\xi_j^T S_{\Upsilon_j} \Phi_j$, from Assumption 2, i.e. $||\Phi_j|| \leq \eta_j ||\xi_j||$, for $\eta_j > 0$, and using the following inequality

$$\lambda_{min}(S_{\Upsilon_j}) \left\| \xi_j \right\|^2 \leq V_{(\xi_j)} \leq \lambda_{max}(S_{\Upsilon_j}) \left\| \xi_j \right\|^2$$

where $\lambda_{min}(S_{\Upsilon_j})$ and $\lambda_{max}(S_{\Upsilon_j})$ are the minimum and maximum eigenvalues of the matrix S_{Υ_j}, respectively. Then, it follows that

$$\dot{V}_{(\xi_j)} \leq \frac{-\Upsilon_j}{2|\xi_{1j}|} \xi_j^T S_{\Upsilon_j} \xi_j + \frac{2\,\eta_j ||S_{\Upsilon_j}||||\xi_j||^2}{2|\xi_{1j}|} \tag{29}$$

$$\leq \frac{-1}{2|\xi_{1j}} (\Upsilon_j - \tilde{\eta}_j) V_{(\xi_j)} \tag{30}$$

where $\tilde{\eta}_j = \dfrac{2\eta_j ||S_{\Upsilon_j}||}{\lambda_{max}(S_{\Upsilon_j})}$.

Considering the following inequality $|\xi_1| \leq ||\xi|| \leq \dfrac{V^{1/2}_{(\xi)}}{\lambda^{1/2}_{min}(S_{\varUpsilon_j})}$ is satisfied. Then, it follows that

$$\dot{V}_{\xi_j} \leq -\varkappa_j V^{1/2}_{\xi_j} \tag{31}$$

where $\varkappa_j = \dfrac{(\varUpsilon_j - \tilde{\eta}_j)\lambda^{1/2}_{min}(S_j)}{2}$.

Choosing $\varUpsilon_j$ sufficiently large such that the inequality $\varUpsilon_j > \tilde{\eta}_j$ holds, then $\dot{V}_{\xi_j}$ is negative definite. The convergence time could be estimated by solving the following differential equation

$$\dot{\vartheta} = -\varkappa_j \vartheta^{1/2}, \quad \vartheta(0) = \vartheta_0$$

is given by

$$\vartheta(t) = (\vartheta_0^{1/2} - \frac{\varkappa_j}{2}t)^{1/2}$$

From the comparison principle [17], where $V_{(\xi_j)}(t) \leq \vartheta(t)$ when $V_{(\xi_j)}(0) \leq \vartheta_0$. Therefore, ξ_j converges to zero in finite time and reaches that value at time given by

$$T = \frac{2\sqrt{V_{(\xi_j)}(0)}}{\varkappa_j}$$

Thus, the states ξ_{1j} and ξ_{2j} converge to zero in finite time. As a result, the states ζ_{1j} and ζ_{2j} will converge to zero. Therefore, $\mathbf{S}_j$ and $\dot{\mathbf{S}}_j$ tend to zero. Then, the tracking and formation error (14), from system (1)–(10) also converge to origin.

4 Simulation Results

In this section, numerical results carried out in a Matlab/Simulink environment are introduced, in order to illustrate robustness and performance of proposed scheme. For controller design, the distance between wheels of i-th robot is considered as $\varDelta_{FL} = 0.05$ meters with a leader and two followers. The leader moves along circular trajectory, while followers were given same tracking reference with a specific distance to the leader.

The control gains are $\varUpsilon_L = 10$, $\varUpsilon_F 1 = 8$ and $\varUpsilon_F 2 = 8$. The corresponding formation response among leader and followers are shown in Fig. 6, where $\varDelta_{LF} = \varDelta_{ij}$ converge to 0.05 m as the desired distance ($\varDelta_d$), as illustrated in the Fig. 7 and corresponding position error Fig. 8.

Figure 8 shows the formation error which converge to zero in finite time.

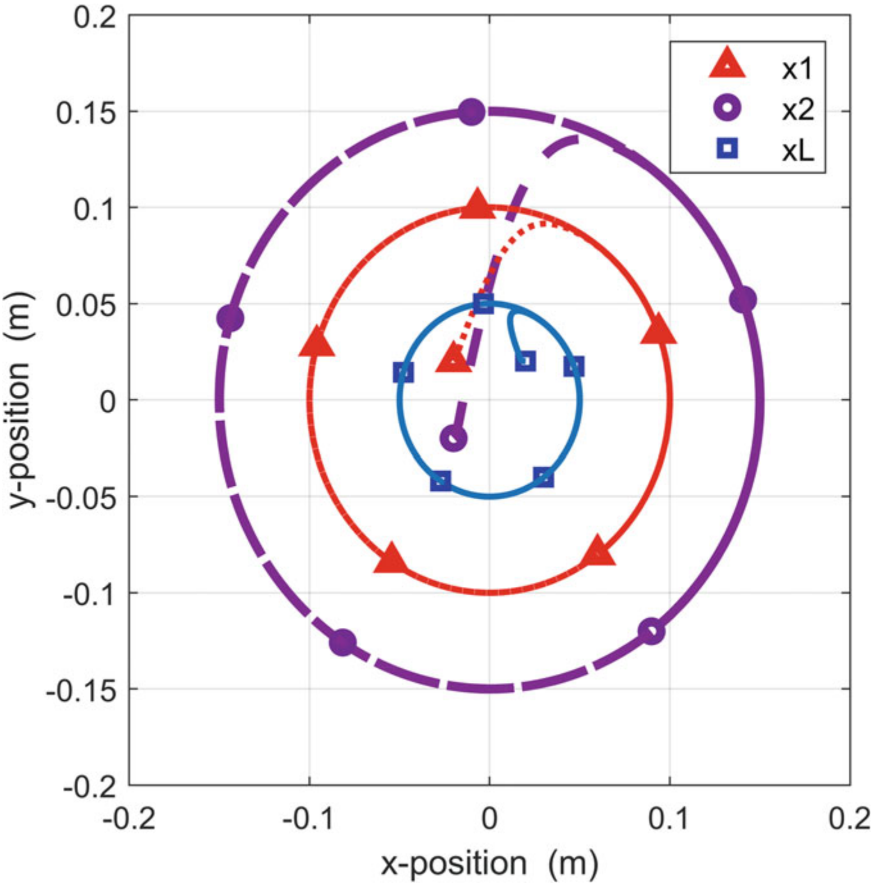


Fig. 6 Simulation position response for leader and followers

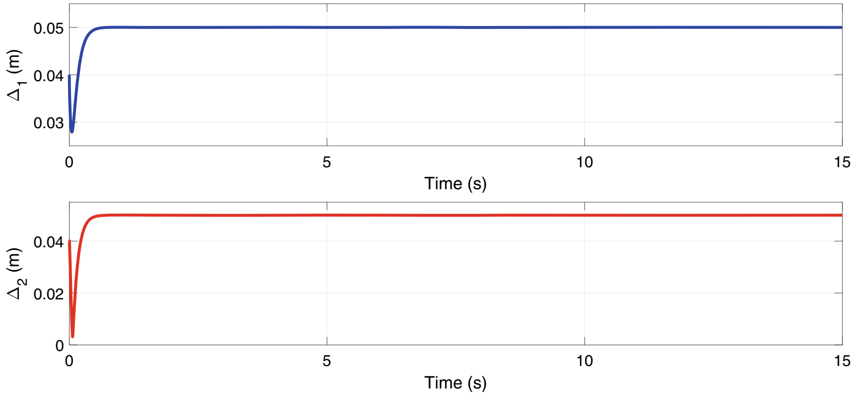


Fig. 7 Simulation desired distance (Δ_{LF})

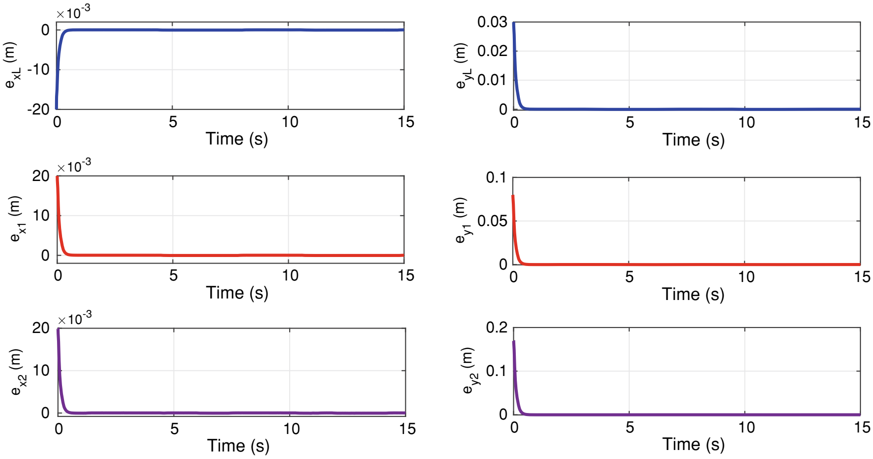


Fig. 8 Simulation position error

5 Experimental Results

Now, experimental results are shown for illustrating the performance of proposed methodology. The experiments were performed using three E-Puck mobile robots (see Fig. 9).

The E-Puck unicycle robot has two individually driven wheels using stepper motors [12]. Furthermore, odometry is used to get the actual position of the robots, while a PC generates robot trajectories and compute the control robot signals, where it is considered that each follower has access only to leader information. The control signals to each wheel are sent via Bluetooth from PC to robots. This form of implementation is chosen due to the limited bandwidth of Bluetooth communication and due to the limiting processing power of the robot's embedded processors(see [13] for further details).

Each robot get the velocity of each wheels by means of following equations

$$v_{l,i} = v_i + \frac{\omega_i}{2}; \quad v_{r,i} = v_i - \frac{\omega_i}{2}, \quad i = 1, \ldots, N \tag{32}$$

where $v_{l,i}$ and $v_{r,i}$ are left and right speed of each motor respectively. Figure 10 present the scheme of communication between computer and each e-puck for a L-F approach.

For controller design the following data are considered:

- Maximum linear speed $\nu_{i_{max}} = 0.13$ m/s.
- The maximum angular speed $\gamma_{i_{max}} = 4.96$ rad/s.
- The distance between wheels of i-th robot $b = 0.0525$ m.
- Diameter of wheels $r = 0.041$ m.

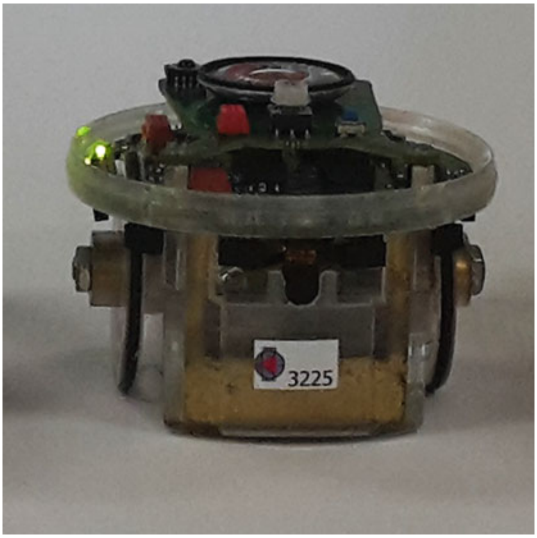


Fig. 9 E-puck unicycle experimental robot

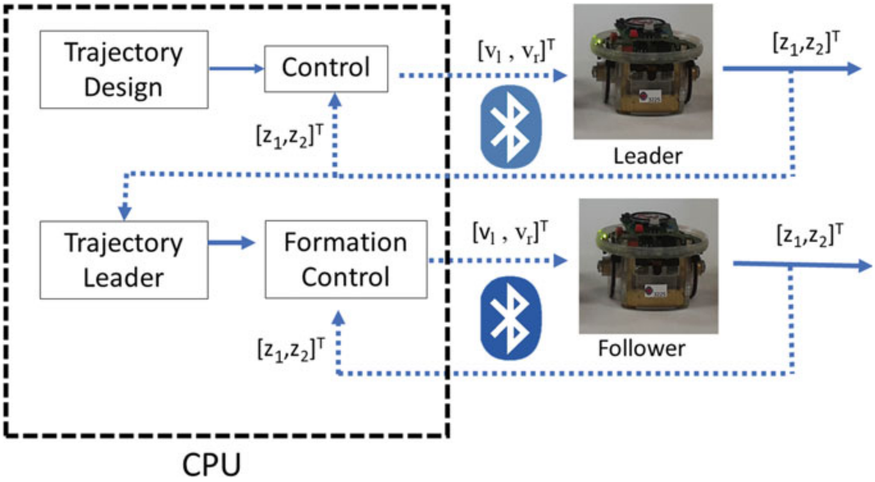


Fig. 10 Communication design based on a L-F approach

• Mass of robot	$m_i = 0.2\,\text{Kg}.$
• Distance from geometric center	$L_i = 0.09\,\text{m}.$

In order to illustrate the robustness to parametric uncertainties, a variation of ±50% in followers mass is considered.

5.1 Circular Trajectory

In this experimental case, initial conditions of the robots are chosen as $[x_L(0), y_L(0)]^T = [0, 0.300]$, $[x_F(0), y_F(0)]^T = [0, 0]$ where leader is denoted by L and follower by F subindexes respectively; while $\Delta_{FL} = 0.100$ denotes the distance between leader and follower (Δ_{FL}).

The sliding surface for the leader robot is give by

$$\mathbf{S}_L = {}_L(\mathbf{y}_L - \mathbf{y}_{refL}) + (\dot{\mathbf{y}}_L - \dot{\mathbf{y}}_{refL}) \tag{33}$$

The sliding surface for the follower robot is given by

$$\mathbf{S}_F = {}_F(\mathbf{y}_F - {}_{LF} - \mathbf{y}_{refL}) + (\dot{\mathbf{y}}_F - \dot{\mathbf{y}}_{refL}) \tag{34}$$

Then, the leader robot tracks the desired reference $\mathbf{y}_{refL}$, in this case the reference is a circular trajectory of radius R, while follower robot tracks a circular trayectory reference $\mathbf{y}_{refF}$ of radius $R + \Delta_d$, where Δ_{LF} converges to Δ_d, the desired distance with respect to the leader.

Figures 11, 12 and 13 display the robots transitioning between formation using control strategy given by

$$\begin{pmatrix} u_{1i} \\ u_{2i} \end{pmatrix} = \begin{pmatrix} \cos\xi_i & \sin\xi_i \\ -\sin\xi_i & \cos\xi_i \end{pmatrix} \begin{pmatrix} \mathrm{v}_{1i} + \omega_i z_{4i} + \gamma_{1i} \\ \mathrm{v}_{2i} - \omega_i z_{3i} + \gamma_{2i} \end{pmatrix} \tag{35}$$

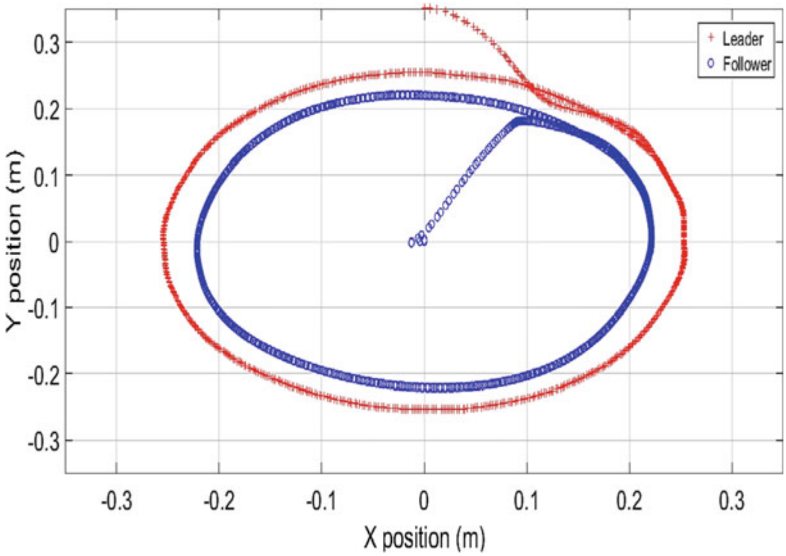


Fig. 11 Scatter of robots position

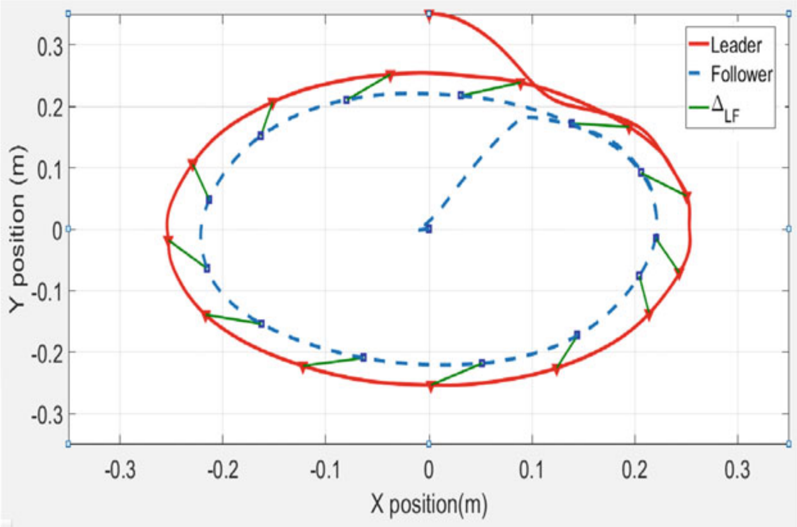


Fig. 12 Formation tracking response

for $i = L, F$; combined with the super twisting control

$$\mathrm{v}_j = -_j(\dot{\mathbf{y}}_j - \dot{\mathbf{y}}_{refi}) + \ddot{\mathbf{y}}_{refi} - K_{1j}\Pi_j - K_{2j}\Gamma_j \tag{36}$$

where

$$\Pi_j = \begin{pmatrix} |S_{1j}|^{1/2} sign(S_{1j}) \\ |S_{2j}|^{1/2} sign(S_{2j}) \end{pmatrix}; \quad \Gamma_j = \begin{pmatrix} \int_0^t sign(S_{1j}) d\tau \\ \int_0^t sign(S_{2j}) d\tau \end{pmatrix}$$

and $K_{1j},\ K_{2j}$ are the gains of the control v_j, which are parametrized, in terms of the parameter Υ_j, as $K_{1j} = 2\Upsilon_j I_{2\times 2}$, $K_{2j} = \dfrac{\Upsilon_j^2}{2} I_{2\times 2}$, and $\Upsilon_j > 0$, for $j = L, F$. The control gains used in the experiments were selected as $\Upsilon_L = 100$ and $\Upsilon_F = 80$.

The leader moves along circular trajectory and follower was making the same tracking with a specific distance to the leader. Figure 12 illustrate the trajectory of the leader (solid), the follower (dash), while holding desired distance.

Furthermore, the corresponding formation between leader and follower is shown in Fig. 14, where Δ_{LF} converges to the desired distance Δ_d, which is chosen equal to a constant, i.e. $\Delta_d = 0.100$ m.

From these results we can appreciate the good performance of the proposed control strategy, despite possible effects due to communication delays. These results show the good performance of the control strategy both in the presence of noise and under the effect of chattering, which did not affect its performance.

Figure 15 shows the formation error $e = \Delta_d - \Delta_{LF}$, which converges to zero in finite time and is robust in presence of uncertainties. Notice that the friction and

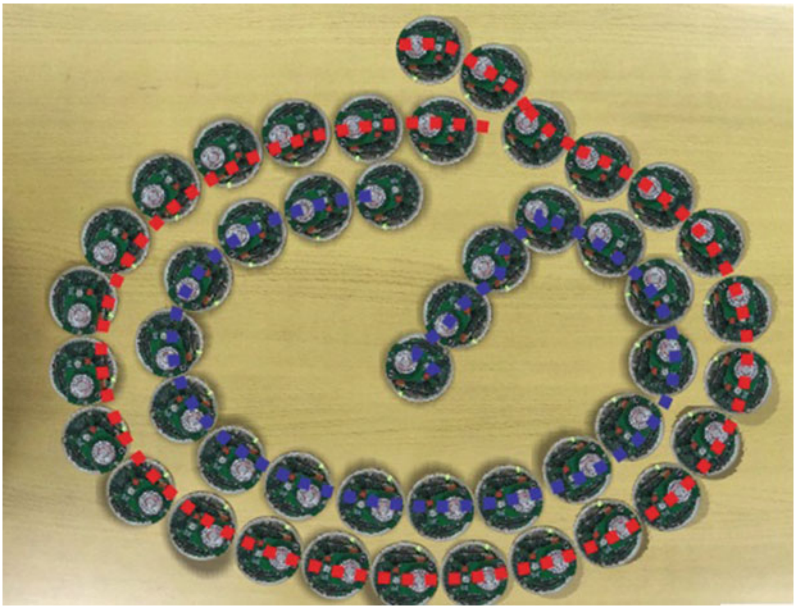


Fig. 13 Experimental formation trajectory tracking

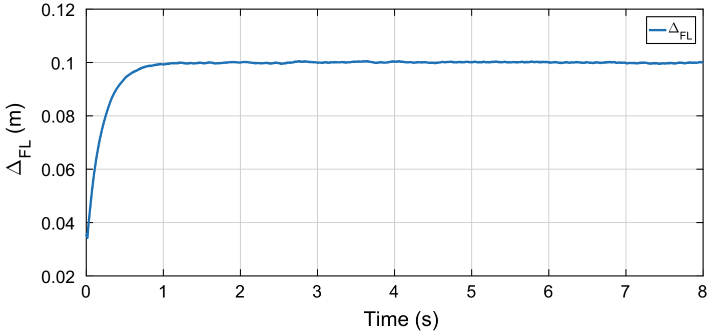


Fig. 14 Distance between leader and follower mobile robots

other external disturbances are attenuated by the control strategy, and a good control tracking is obtained.

Figure 16 shows left and right velocities of the follower, respectively. As can be seen, Super Twisting control design remarkable reduced chattering effect.

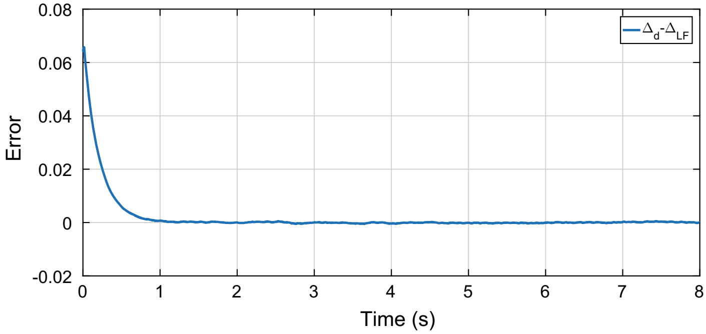


Fig. 15 Formation error for circular trajectory

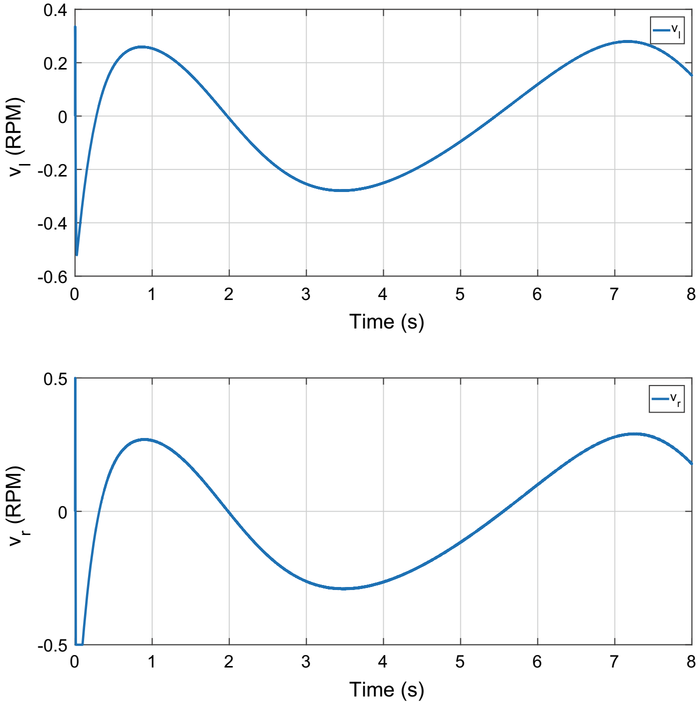


Fig. 16 Speed of follower. (Top) Left-wheel velocity. (Bottom) Right-wheel velocity

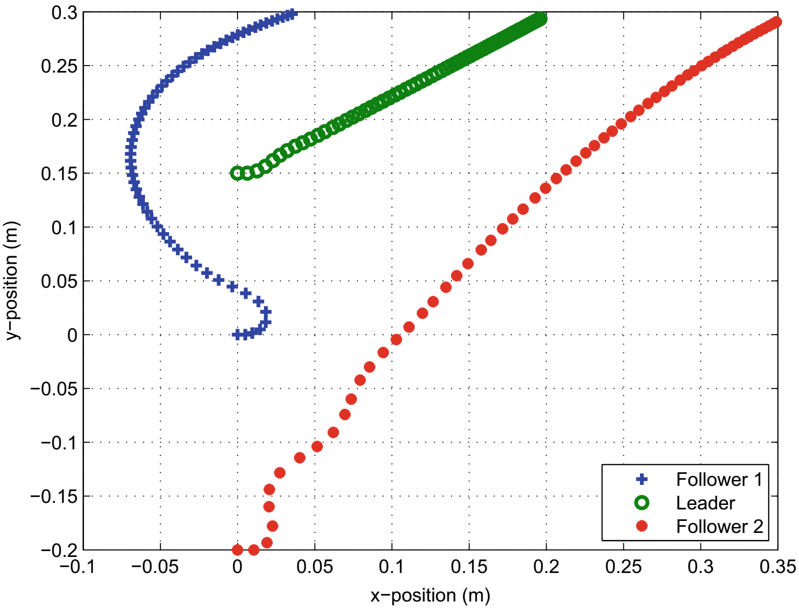
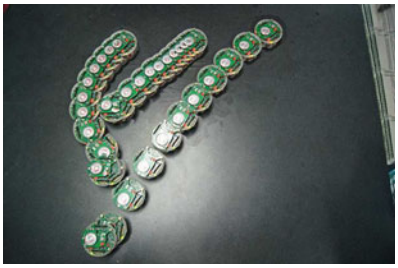


Fig. 17 Position scatter for line formation

Fig. 18 Experimental line formation



5.2 *Line Formation*

In this case, the leader and two follows must be kept a linear formation. i.e. the separation distance among robots is constant.

For this experiment, the mobile robots start from the following initial conditions (x_0, y_0): for the leader at (0, 0.1), and for the followers, follower 1 (−0.2, 0) and for the follower two at (−0.3, 0.2), (see Figs. 17 and 18).

In all the experiments, we use the same controller gains considered in the circular case.

As can be seen from Figs. 19 and 20, after 6 s the trajectories of the robots achieve the formation. Furthermore, the corresponding robot speeds are plotted in the Fig. 21. Finally, from these plots the proposed control strategy performs well.

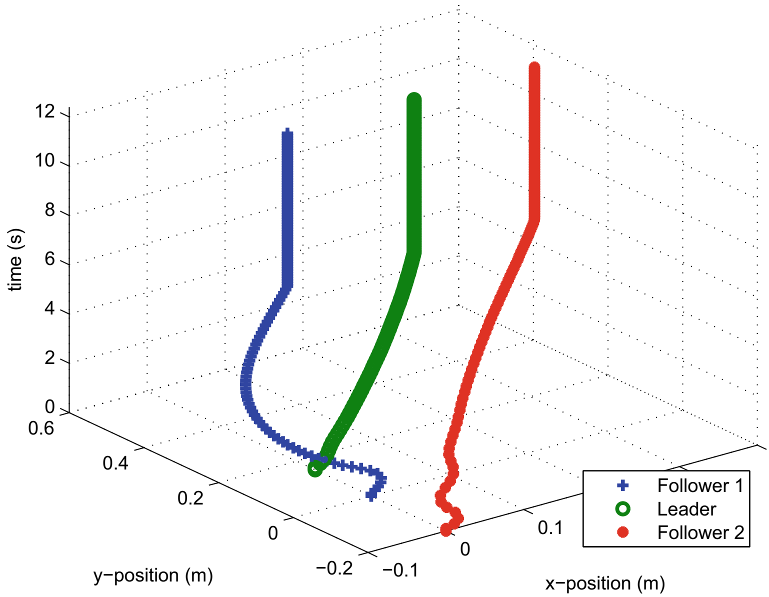


Fig. 19 Line formation along space

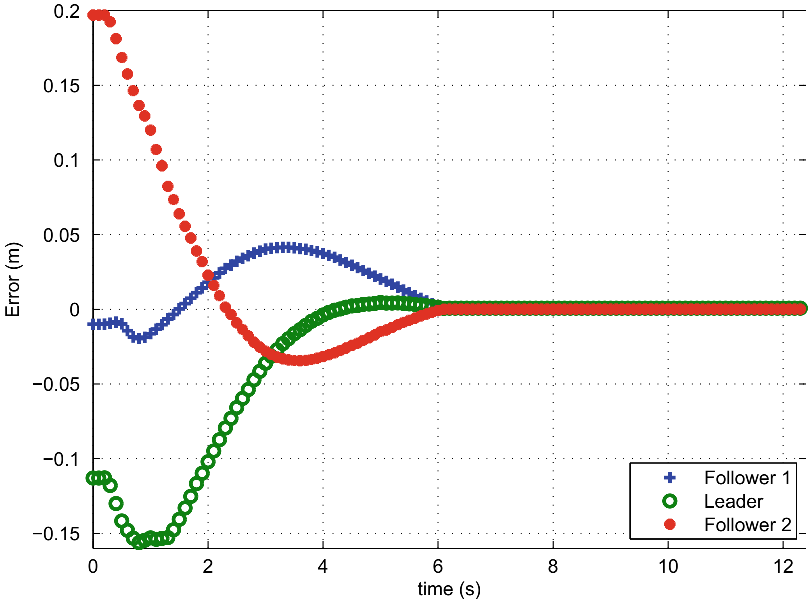


Fig. 20 Distance errors in line formation

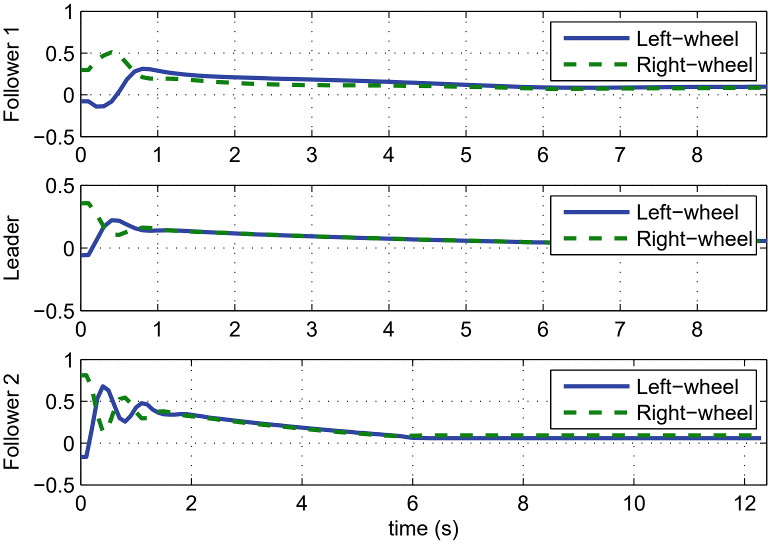


Fig. 21 Wheels velocities for line formation

5.3 *Column Formation*

For this experimental case, a column formation is considered, where the initial conditions of the robots (x_0, y_0) are selected as follows: for the leader at (0, 0.15), (0, 0) and for the follower 1 at (0, 0) and for second follower at (0, 0.2) (see Figs. 22 and 23). In the column formation the distance of separation between the robots is selected so that the component in y is kept constant.

From the Figs. 24 and 25, we can appreciate that after 7 s the formation is achieved and kept, while robots speeds are illustrated in Fig. 26.

5.4 *Switching Formation*

Now, we consider the case where the separation distance between robots varies so that the formation configuration changes from column to line, and then to delta configuration; while holding relative distances among robots.

The initial conditions of the mobile unicycles are selected as follows: for leader robot is (0, 0.1), and for the first and second followers are (0, 0) and (0, 0.2), respectively; (see Fig. 27).

As can be seen in the Figs. 28 and 29, before 5 s the formation is achieved and kept along transitioning. The corresponding robot speeds are shown in the Fig. 30. From these responses, the control strategy performs well and the formation problem

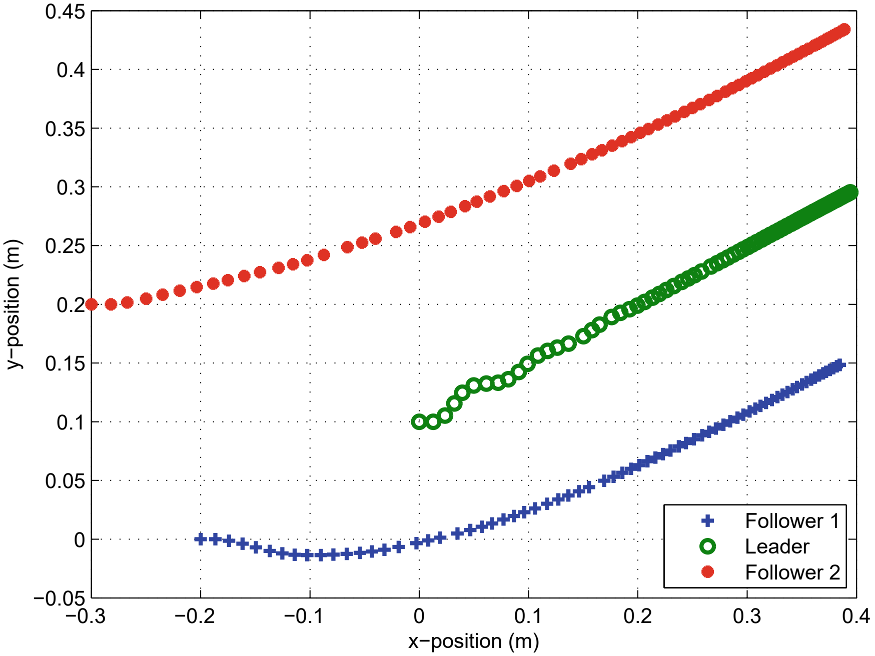
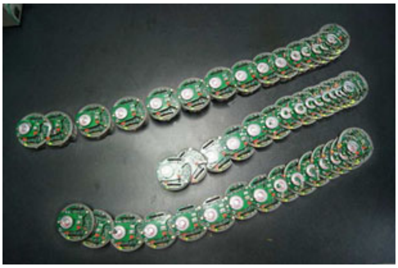


Fig. 22 Position scatter for column

Fig. 23 Experimental column formation



is solved. It is worth mentioning the robustness of the control strategy applied to the formation problem has been solved satisfactorily, in presence of external disturbances (time delays in communication, friction) and parametric uncertainties (mass, parameters of the motors of the robot).

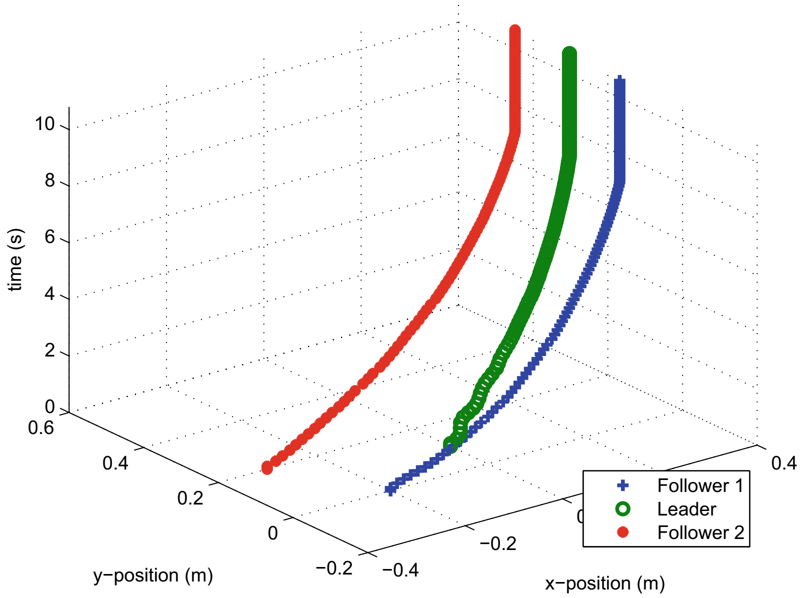


Fig. 24 Column formation along space

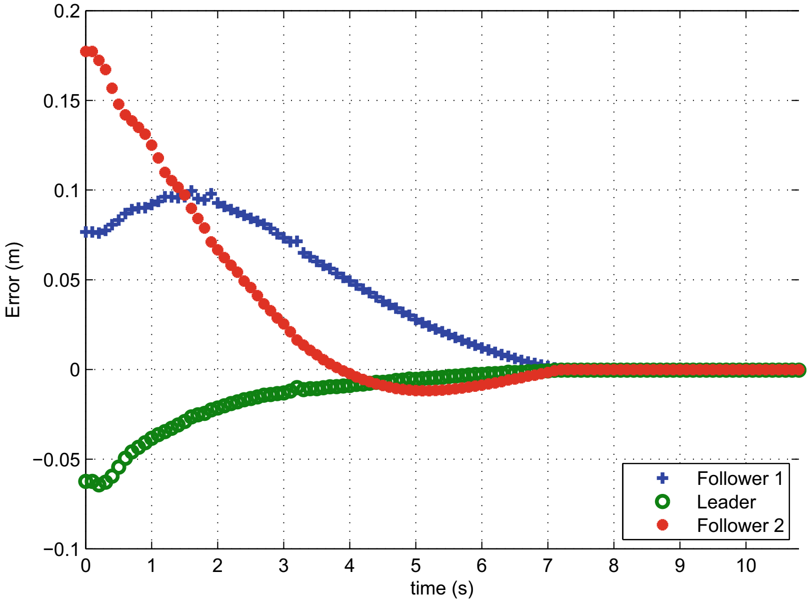


Fig. 25 Column error formation

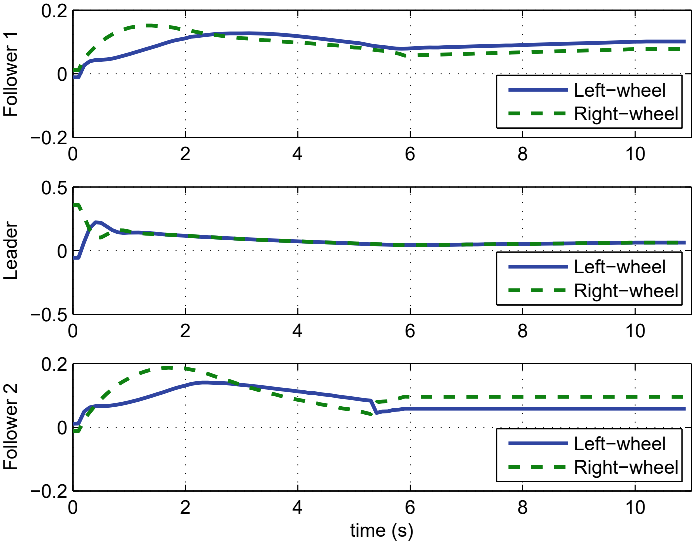


Fig. 26 Wheels velocities for column formation

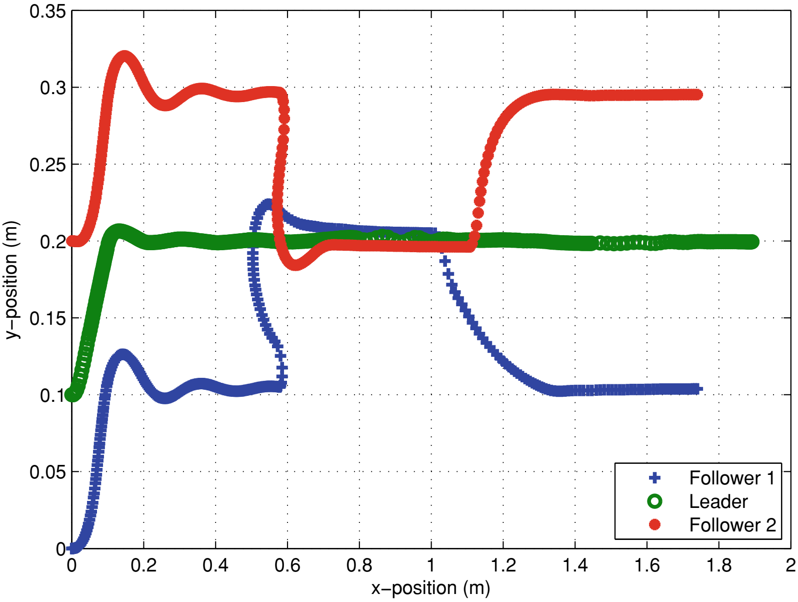


Fig. 27 Position scatter for switching formation

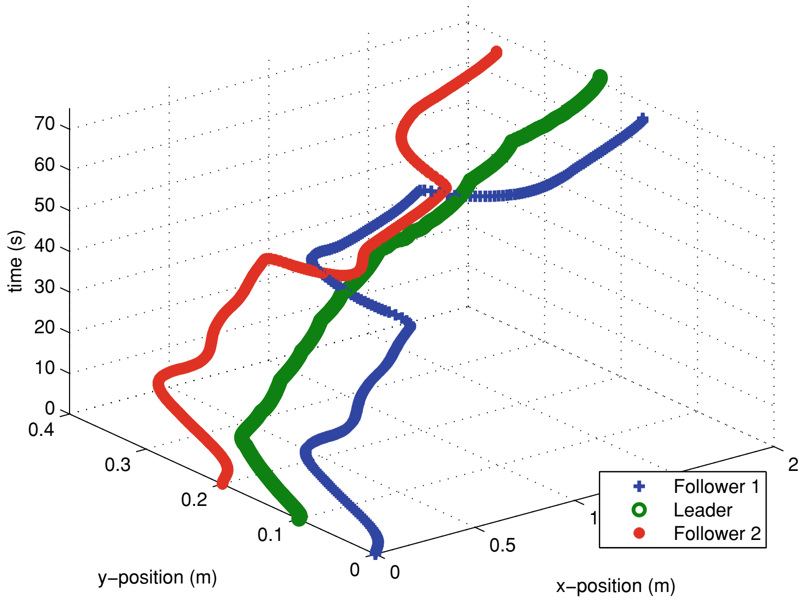


Fig. 28 Switching between formation along space

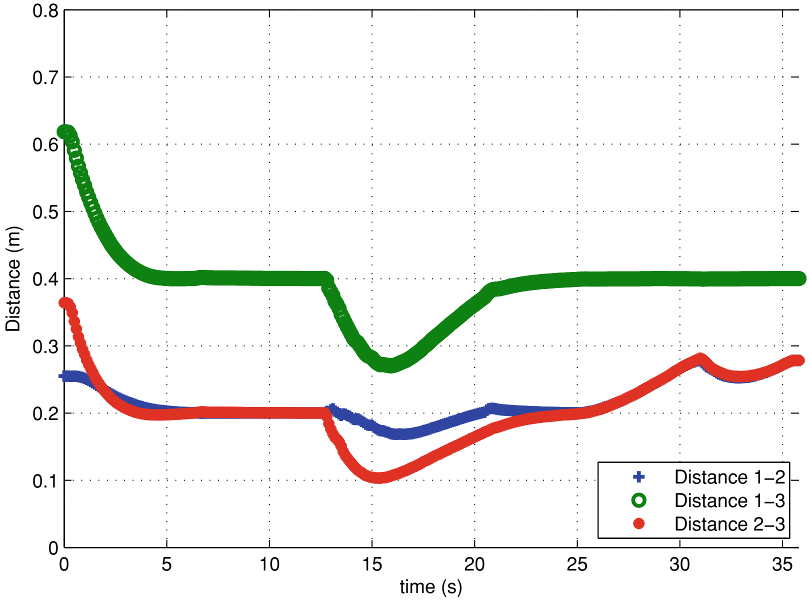


Fig. 29 Distance among robots for switching formation

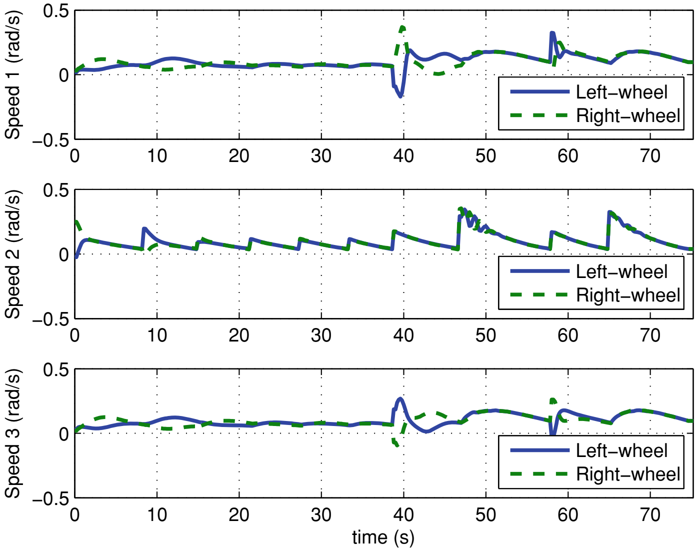


Fig. 30 Wheels velocities for switching formation

6 Conclusion

This chapter has focused on group synchronization of robots, where the goal of the group was to achieve a desired formation while keeping the distances among the agents. With this aim, a control strategy based on a simplified Super-Twisting algorithm has been designed. Proposed approach required one parameter to be tuned, and hence, implementation on multi-agent systems is simplified, due to the reduction of the conditions to be satisfied. Numerical and experimental validation demonstrated the effectiveness of the proposed methodology to maintain formations using the Leader-Follower strategy.

Furthermore, an interesting extension of this work is consider the adaptive version of the gains in order to improve the performance of this strategy and avoid the over-estimation of the gains, as well as introduce the use of observers to estimate non-measurable variables or reduce the number of sensors.

Finally, a study related to communications between robots and the effects of delays in communications will be necessary.

Acknowledgements The work in this document has been funded by the Mexican National Council on Science and Technology (CONACYT) and PAICYT-UANL, Project 479-IT-2022.

References

1. Alatise MB, Hancke GP (2020) A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access 8:39,830–39,846. https://doi.org/10.1109/ACCESS.2020.2975643
2. Anderson BD, Yu C, Fidan B, Hendrickx JM (2008) Rigid graph control architectures for autonomous formations. IEEE Control Syst 28(6):48–63
3. Bräunl T (2008) Embedded robotics: mobile robot design and applications with embedded systems. Springer Science & Business Media
4. Carpentier J, Tonneau S, Naveau M, Stasse O, Mansard N (2016) A versatile and efficient pattern generator for generalized legged locomotion. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3555–3561
5. Chalanga A, Kamal S, Fridman LM, Bandyopadhyay B, Moreno JA (2016) Implementation of super-twisting control: super-twisting and higher order sliding-mode observer-based approaches. IEEE Trans Industr Electron 63(6):3677–3685
6. Das AK, Fierro R, Kumar V, Ostrowski JP, Spletzer J, Taylor CJ (2002) A vision-based formation control framework. IEEE Trans Robot Autom 18(5):813–825
7. De La Cruz C, Carelli R (2006) Dynamic modeling and centralized formation control of mobile robots. In: IEEE 2006-32nd annual conference on industrial electronics (IECON). IEEE, pp 3880–3885
8. De La Cruz C, Carelli R (2008) Dynamic model based formation control and obstacle avoidance of multi-robot systems. Robotica 26(3):345–356
9. Defoort M, Floquet T, Kokosy A, Perruquetti W (2008) Sliding-mode formation control for cooperative autonomous mobile robots. IEEE Trans Industr Electron 55(11):3944–3953
10. Elyoussef ES, De Pieri ER, Moreno UF, Jungers M (2012) Super-twisting sliding modes tracking control of a nonholonomic wheeled mobile robot. IFAC Proc Vol 45(22):429–434
11. Fax JA, Murray RM (2004) Information flow and cooperative control of vehicle formations. IEEE Trans Autom Control 49(9):1465–1476
12. GCtronic (2018) e-puck education robot. http://www.e-puck.org/
13. GonçSalves P, Torres P, Alves C, Mondada F, Bonani M, Raemy X, Pugh J, Cianci C, Klaptocz A, Magnenat S, Zufferey JC, Floreano D, Martinoli A (2009) The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions, vol 1
14. Gutiérrez Martínez SV (2021) Control de robots móviles para el seguimiento de trayectorias. PhD thesis, Universidad Autónoma de Nuevo León
15. Hsu HCH, Liu A (2005) Multiagent-based multi-team formation control for mobile robots. J Intell Robot Syst 42(4):337–360
16. Isidori A (2013) Nonlinear control systems. Springer Science & Business Media
17. Khalil HK (1996) Noninear systems. Prentice-Hall, New Jersey 2(5):1–5
18. Lawton JR, Beard RW, Young BJ (2003) A decentralized approach to formation maneuvers. IEEE Trans Robot Autom 19(6):933–941
19. Lee H, Utkin VI (2007) Chattering suppression methods in sliding mode control systems. Annu Rev Control 31(2):179–188
20. Levant A (2003) Higher-order sliding modes, differentiation and output-feedback control. Int J Control 76(9–10):924–941
21. Li X, Xiao J, Cai Z (2005) Backstepping based multiple mobile robots formation control. In: 2005 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 887–892
22. Low CB (2014) A flexible virtual structure formation keeping control design for nonholonomic mobile robots with low-level control systems, with experiments. In: IEEE international symposium on intelligent control (ISIC). IEEE, pp 1576–1582
23. Monteiro S, Bicho E (2010) Attractor dynamics approach to formation control: theory and application. Auton Robot 29(3–4):331–355
24. Moreno JA, Osorio M (2008) A Lyapunov approach to second-order sliding mode controllers and observers. In: 2008 47th IEEE conference on decision and control. IEEE, pp 2856–2861

25. Parker LE, Rus D, Sukhatme GS (2016) Multiple mobile robot systems. In: Springer handbook of robotics. Springer, pp 1335–1384
26. Pilloni A, Pisano A, Usai E (2012) Parameter tuning and chattering adjustment of super-twisting sliding mode control system for linear plants. In: 2012 12th international workshop on variable structure systems (VSS). IEEE, pp 479–484
27. Polyakov A, Poznyak A (2009) Reaching time estimation for "super-twisting" second order sliding mode controller via Lyapunov function designing. IEEE Trans Autom Control 54(8):1951–1955
28. Rubio F, Valero F, Llopis-Albert C (2019) A review of mobile robots: concepts, methods, theoretical framework, and applications. Int J Adv Robot Syst 16(2):1729881419839596. https://doi.org/10.1177/1729881419839596
29. Saab W, Rone WS, Ben-Tzvi P (2018) Discrete modular serpentine robotic tail: design, analysis and experimentation. Robotica 36(7):994–1018
30. Seeber R, Horn M (2017) Stability proof for a well-established super-twisting parameter setting. Automatica
31. Siegwart R, Nourbakhsh IR, Scaramuzza D (2011) Introduction to autonomous mobile robots. MIT Press
32. Silva-Ortigoza R, Silva-Ortigoza G, Hernández-Guzmán VM, Barrientos-Sotelo VR, Albarrán-Jiménez JM, Silva-Garcia VM (2008) Trajectory tracking in a mobile robot without using velocity measurements for control of wheels. IEEE Lat Am Trans 6(7):598–607
33. Solea R, Cernega D (2015) Super twisting sliding mode controller applied to a nonholonomic mobile robot. In: 2015 19th international conference on system theory, control and computing (ICSTCC). IEEE, pp 87–92
34. Utkin VI, Poznyak AS (2013) Adaptive sliding mode control with application to super-twist algorithm: equivalent control method. Automatica 49(1):39–47
35. Xu D, Zhang X, Zhu Z, Chen C, Yang P (2014) Behavior-based formation control of swarm robots. Mathematical Problems in Engineering 2014
36. Yousuf BM, Qazi HA, Ali N, Dayo AA, Ali R, Hassan SA (2017) Formation control of nonholonomic mobile robots based on feedback linearization and observers. In: International multi-topic conference (INMIC). IEEE, pp 1–8
37. Zhao Y, Park D, Moon J, Lee J (2017) Leader-follower formation control for multiple mobile robots by a designed sliding mode controller based on kinematic control method. In: 2017 56th annual conference of the society of instrument and control engineers of Japan (SICE). IEEE, pp 186–189

Deep Reinforcement Learning Applied to Multi-agent Informative Path Planning in Environmental Missions

Samuel Yanes Luis, Manuel Perales Esteve, Daniel Gutiérrez Reina, and Sergio Toral Marín

Abstract Deep Reinforcement Learning algorithms have gained attention lately due to their ability to solve complex decision problems with a model-free and zero-derivative approach. In the case of multi-agent problems, these algorithms can help to easily find efficient cooperative policies in a feasible amount of time. In this chapter, we present the Informative Patrolling Problem, a commonplace task in the conservation of water resources. The approach is presented here as a convenient methodology for the synthesis of cooperative policies than can solve simultaneous objectives present in the unmanned monitoring of lakes and rivers: maximizing the collected information of water parameters and the collision-free routing with multiple surface vehicles. For this mixed objective, it is proposed a Deep Q-Learning scheme with a convolutional network as a shared fleet policy. In order to solve the credit assignment problem, it is proposed an effective multiagent decomposition of the informative reward with a discussion of other several state-of-the-art topics of Reinforcement Learning: noisy networks for enhanced exploration of the state-action domain, the use of a visual states, and the shaping of the reward function. This methodology, as it is quantitative demonstrated, allows a significant improvement in water resource monitoring compared to other heuristics.

Keywords Deep reinforcement learning · Information gathering · Multiagent · Environmental monitoring

S. Yanes Luis (✉) · M. Perales Esteve · D. Gutiérrez Reina · S. Toral Marín
Department of Electronics, University of Sevilla, Av. de Los Descubrimientos s/n, 41003 Sevilla, Spain
e-mail: syanes@us.es

M. Perales Esteve
e-mail: mperales@us.es

D. Gutiérrez Reina
e-mail: dgutierrezreina@us.es

S. Toral Marín
e-mail: storal@us.es

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_2

1 Introduction

Conservation of drinking water reserves is a strategic objective for societies. This resource has a direct impact on the health of people, agriculture, and industry in a country. However, due to human related causes, these large bodies of water are continuously being polluted: untreated discharges, oil spills from boats, toxic algae proliferation, etc., drastically impoverish the quality of the aquatic ecosystem and alter the biological balance of the local fauna and flora. In the case of particularly large water bodies, such as Lake Ypacaraí (Paraguay, 60 km^2) or the Mar Menor (Spain, 170 km^2), manual samplings for continuous monitoring of the variables of water quality (WQ) become strenuous and costly. Several human and material resources are needed, and there is always a risk to operators in contaminated scenarios. An efficient way to collect physicochemical water data is to use autonomous surface vehicles (ASVs) [22] equipped with highly sensitive WQ sensors. These sensors can measure physical parameters such as pH, temperature, oxygen saturation levels, and water turbidity, or chemical parameters such as nitrites, sulphates, and dissolved chlorophyll. All these parameters can be used, by means of a proper data analysis, to infer the biological state of a lake or a river. In such enormous water bodies it is mandatory to use a fleet of multiple vehicles, as the battery budget of one single agent do not provide enough coverage. With multiple vehicle it is easy to obtain a complete set of measures that truly represents the useful information to be considered for resource conservation. The challenge of the multi-vehicle paradigm lies in the need for an effective coordination policy that allows them to take samples autonomously in the lake following a low redundancy criterion. The vehicles must coordinate to share the search space to collect as much information as possible. Information, represented by the values of the water variables, will be obtained sequentially as the vehicles acquire the measurements one by one at different locations in the scenario. An effective fleet policy must consider taking samples in those places with the higher information available, which is to sample in places of high uncertainty. The definition of the information depends on the context but we can define a general corollary of such task, the Informative Path Planning (IPP) [5], where the objective is to obtain an optimal acquisition path Ψ^* that maximizes this informative value I on a fixed time budget T (Fig. 1):

$$\Psi^* = \max_{\Psi} \; I(\Psi) \tag{1}$$

Information I can be defined stochastically in terms of the reduction of uncertainty of each sampled point of the environment [24]. A reasonable initial hypothesis states that every navigable point p in the complete set X behaves like a Gaussian random variable $p \in X \sim \mathcal{N}(\mu, \sigma)$. Then, let being $\Sigma[X, X_{meas}]$ the spatial correlation matrix that describes the statistical relationship between the samples, that is, the uncertainty level for each point in the locations where X_{meas} was sampled. This matrix indicates the level of uncertainty of each point, considering the locations where X_{meas} has been sampled. This approach implements a Matérn Kernel Function (MKF) [21] to spatially correlate samples as a function of their adjacency, under the acceptable

Fig. 1 Autonomous vehicles carrying out a surveillance mission in a recreational lake in Seville (Spain)

assumption that physically close samples will be more closely related. Given that, the IPP will consist of sequentially deciding the next physical point at which to take a sample, following the information gain maximization criterion. Following the definition of information gain in information theory, $\Sigma[X, X_{meas}]$ defines how informative a point is according to the decrease in entropy of the model formed by the set in points of the lake X and the sample points X_{meas} [21]. Finally, the IPP problem to be solved here consists of minimizing the total entropy of the scenario $H\,[X|X_{meas}]$.

In addition to the informative criterion, the fleet policy must consider in the movement planification any obstacles and nonnavigable zones. In this multiagent paradigm, it is also needed to consider the agents to be moving obstacles, so the path planning becomes intrinsically dynamic and harder to solve. As the complexity of this problem explodes because of the many feasible paths and restrictions in every realistic scenario, it is needed a proper algorithm that can deal with the IPP problem with multiple agents and can scale to different fleet sizes. In this chapter, a Reinforcement Learning (RL) approach is proposed to deal with the high dimension of this problem, inspired by the success of [15] where a Q-Learning algorithm solves hard tasks of proficiency. The RL paradigm tries to solve sequential problems by trial and error. In the RL paradigm, an agent tries to maximize the reward obtained r_t in the instant t, by taking an action a_t in an environment [25]. The information about the environment, called the state s_t, is usually observable in some form, so it can be interpreted to maximize the reward. This whole process ends when the agent has synthesized a policy $\pi(s)$ that maps a state s into the best learned action a (see Fig. 2). To adapt the IPP problem to this paradigm, a formulation has been proposed in terms of a Markov Decision Process (MDP) where the aforementioned agent interacts sequentially in a finite control horizon T, by taking discrete actions, which physically corresponds to movements, that maximize the reward at the end of a mission:

$$\pi(s)^* = \max_{\pi} \sum_{t=0}^{T} [r(s_t, a = \pi(s_t))] \tag{2}$$

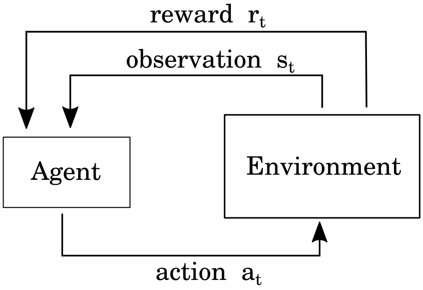


Fig. 2 Basic reinforcement learning scheme. In every RL case, there is at least an agent that interacts through actions with an environment. Returns information of two types: the next state and a reward that evaluates the action in the current state

For this purpose, the reward function $r(s, a)$ must be tailored to meet the objectives of the path planning, i.e., the most informative, non-redundant, obstacle-free routes for every ASV in the fleet. In the following sections, an appropriate reward function for the multi-agent IPP is presented.

In [15], the use of deep convolutional neural networks (CNN) was introduced as a novelty in the RL scene to deal with high-dimensional states. This was a methodological revolution for RL. CNNs were used to represent the state-action function Q of the MDP. This deep Q function represents the expected future reward given a current state s and an action a. The capacity of neural networks to fit non-linear and complex behavior allowed the algorithm to supersede other benchmarks. Through many iterations, the neural function is optimized with the given reward signal by randomly interacting with the environment. This algorithm, called Deep Q-Learning, constitutes a robust algorithm to solve sequential decision problems such as the one presented in this chapter. All things considered, this chapter adapts the single-agent Deep-Q Learning (DQL) method to solve the multi-agent IPP, taking advantage of the model-free ability of this algorithm. There is additional complexity in doing so, as this problem deals with multiple agents at the same time. Multi-agent RL (MARL) involves certain difficulties [9], such as learning scalability, noninjectivity of the reward function, and nonstationery of the environment. In this chapter, we present an observational method to include all the information in the visual input of the network for better efficiency. Additionally, to find a suitable policy, it is necessary to sufficiently explore the state-action set with an increasing size with each agent included in the fleet. Therefore, classic exploration strategies, such as $\varepsilon - greedy$ [34], fail to find the optimal actions for success in the IPP task. To overcome this situation, in this chapter, we propose the use of noisy neural networks (noisy-NN) [7]. Noisy NN are a nondeterministic version of the classical neural networks. Using a Gaussian distribution, each parameter introduces some level of noise that can enhances exploration.

This chapter addresses certain research gaps in the literature on multiagent optimization, information gathering, and deep reinforcement learning. With this approach, we propose an entropic criterion to synthesize the most effective routes that maximizes the information collected for a fleet of vehicles. While other algorithms designed for entropy minimization have been designed solely for single agent paradigm [20], our multiagent formulation by means of DRL results in a very convenient and flexible approach. Another highlight is the use of a global visual state, which allows a better scaling for higher number of vehicles. Previous works tend to use low resolutions, local observations, and a formulation that does not scale well with the number of agents [10, 29, 34]. Our proposal places value on the use of noisy neural networks for multi-agent policies and proposes a global visual observation method for homogeneous fleets. Finally, this chapter's contribution can be summarized as follows:

- The formulation of the IPP from the perspective of multi-agent uncertainty reduction.
- The application of DRL for the multi-agent IPP resolution.
- A policy formulation with noisy neurons and a state that allows for interchangeability between agents.

The chapter is organized as follows:

In Sect. 2, a brief review of the literature is developed to place the reader in the context of the problem to be solved. In Sect. 3, the IPP is formally described with all the constraints. In Sect. 4, the methods used to solve IPP are described, that is, the DRL algorithm, the state, the neural network, etc. In Sect. 5, the characteristics and results of the experiments are presented. In Sect. 6, these results are discussed and compared with other heuristics. Finally, in Sect. 7, the conclusions of this work and future research lines are presented.

2 Related Work

The use of ASVs for surveillance or monitoring has been gaining momentum recently due to advances in robotics and vehicle autonomy [4]. The use of small robotic vehicles is not only more efficient but is also cheaper compared to the human cost of periodic surveillance missions. In [6] there is a good example of an ASV application for bathymetric measurements of lakes and seas. In this work, a low-cost vehicle is designed to perform bathymetry (depth of lakes and river basins) in an unmanned way. Another example of a use of ASVs is in [16], where a surface vehicle is implemented to monitor disaster zones after hurricanes. The vehicle is equipped with cameras and an underwater sonar camera to search for obstacles in the presence of turbid waters. Another widespread utility of ASVs is the conservation of aquatic natural resources, such as lakes and rivers. Here we can distinguish between the different sub-tasks the conservation of such resources involves: (I) Patrolling [33–35], which consists of continuously monitor the resource as a method of early warning for contamination or sudden changes in the healthiness levels. (II) Efficient model regression [11, 17, 18], which is to efficiently obtain an accurate WQ model of the waters with one or more vehicles. (III) Peak detection [11], which consists in detecting the maxima of contamination in the shortest possible time. (IV) Informative coverage [1, 2], which consists of covering efficiently a certain area to acquire the most information possible. The IPP fits

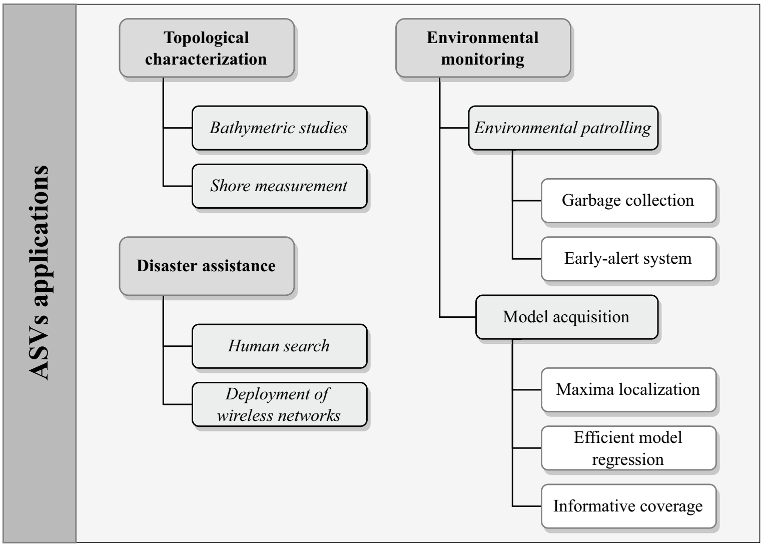


Fig. 3 Different applications of ASVs in the literature

into the set of problems within Informative coverage. In Fig. 3 it has been depicted typical uses of ASVs in the literature and organized in three main branches: disaster assistance, topological characterization, and environmental monitoring.

In [33] the non-homogeneous patrolling problem was formulated as a single agent directive graph problem. The ASV was trained using DRL, resulting in an efficient policy to minimize the average waiting time in lakes with a dissimilar importance criterion. Similarly, the patrol case was extended in [34], with a different number of vehicles for each simulation and a low state resolution. These works have demonstrated the ability of DRL to synthesize high-performance behaviors for patrolling tasks with ASVs but also the high dimension and complexity of these problems. In [35] it was compared the use of such DRL techniques with Genetic Algorithms (GA). The scalability was proven to be better with DRL, which motivates this chapter to choose the DQL for solving the IPP. ASVs were also used in [18] to obtain a contamination model of the Ypacaraí Lake. This work proposes a Bayesian approach within Gaussian Processes (GP) for obtaining an accurate WQ scalar map. In the same line, [17] generalized the algorithm for the multi-agent case. The sampling space was divided using a Voronoi-based tessellation method. The main difference between this approach and ours is that, when continuously monitoring, it is not considered any cost for sampling. In the proposed method, the IPP is solved by taking as many samples as possible to compel with higher accuracy. Another branch of algorithms for monitoring are based on particle swarm optimization (PSO). They were used for the same purpose in [11, 26, 27]. The PSO was applied in a fleet of four ASVs for detection of contamination peaks. This heuristic approach is a simple and model-free approach to localize global and local maxima that adapts naturally to the multi-agent paradigm. Nonetheless, its performance and behavior heavily depends on the hyperparameters. In the informative coverage, some solutions has been proposed by the means of GAs [1, 2, 36]. In [1], the informative problem is modeled as a Travelman Sales Problem (TSP), where the objective is to cover the Ypacaraí Lake in the search of green algae blooms. The algorithm resulted in Eulerian circuits that maximize the effective area in a single-agent scenario. In [2] it is proposed a similar approach but this time focused on an online re-optimization once algae blooms are detected. Those algorithms generates quasi-optimal solutions but too long to be applied in a real scenario. Our approach seeks an effective policy for coverage within a fixed time budget. Additionally, in [36], the problem is extended to multiple agents and multiple objectives for obtaining closed patrolling paths. This work uses a graph formulation and analyses the disperse Pareto-optimal solutions of using multiple agents in coverage problems.

When addressing the use of DRL with unmanned vehicles, there are several interesting examples in the literature [12, 32, 37]. In [12], it is proposed a DRL-based approach to fulfill the complete coverage in cleaning tasks with a multi-form robot. This single-agent application deals with high action dimensionality, uses an algorithm named Actor-Critic with Experience Replay (ACER). The DRL approaches with ASVs usually put the focus on the low-lever control or trajectory planning [37]. In this work, RL was applied to obtain a motor controller for path tracking with a underwater unmanned vehicle (UUV). The applied algorithm resulted in a suitable and realistic controller that quickly reduces the positioning error. In [32] there is an example of the use of DRL for collision avoidance. It is proposed a deep policy trained using DQL to learn how to modify the control signals in order to avoid moving obstacles in the trajectory. This way the agent must learn two simultaneous tasks: to avoid the obstacles and to track the reference path. This is similar to our approach, where the agent must perform the informative gathering task at the same time as it avoids non-navigable areas. Other single-agent approaches deal with informative tasks instead of pure path tracking [19, 28]. In [19], it is proposed the use of DQL for patrolling tasks using a camera for survey scenarios. This approach seeks a policy that optimally patrols the environment by focusing on high-importance zones, but does not deal with obstacle or boundaries avoidance. In [28], an aerial vehicle agent is trained to perform an informative coverage with take-off and landing restrictions. This hard task is achieved by also using DQL. A simple reward function is tailored to award the correct actions and the information collection (Table 1).

With regard to the multi-agent paradigm, DRL approaches must deal with communication problems between agents, the credit assignment problem, and others [9]. In [14] it is proposed a method for decentralized learning in hybrid competitive-cooperative scenarios. This approach adapts the single-agent Deep Deterministic Policy Gradient algorithm [13] to comply with decentralized deploy for continuous actions. Our approach differs from this in the fact that our problem is fully-cooperative and the learning, as the agents are homologous in actions and in their observational abilities, do not require a decentralized learning. In [10], an algorithm to train two agents to survey for wildfires is proposed. This work uses the DQL algorithm with a centralized policy for different agents and an observation-based formulation of

Table 1 Summary of related works using autonomous vehicles for environmental monitoring tasks

References	Application	Algorithm	Vehicle	Multiagent
[6]	Topological characterization of lakes	Manual path	ASV	No
[16]	Hurricane disaster-scenario monitoring	Manual path	ASV	No
[1]	Informative coverage of lakes	Genetic algorithm	ASV	No
[36]	Multi-objective Non-homogeneous patrolling	Evolutionary strategy	ASV	Yes
[2]	Adaptive informative coverage of lakes	Genetic algorithm	ASV	No
[18]	WQ model acquisition of lakes	Bayesian optimization	ASV	No
[17]	WQ model acquisition of lakes	Bayesian optimization	ASV	Yes
[11]	Peak detection and model acq.	Particle swarm optimization	ASV	Yes
[26]	Peak detection and model acq.	Particle swarm optimization	ASV	Yes
[27]	Peak detection and model acq.	Particle swarm optimization	ASV	Yes
[33]	Non-homogeneous patrolling	Deep Q-learning	ASV	No
[34]	Non-homogeneous patrolling	Multi-agent deep Q-learning	ASV	Yes
[34]	Non-homogeneous patrolling	Multi-agent deep Q-learning	ASV	Yes
[10]	Dynamic informative tracking	Multi-agent deep Q-learning	UAVs	Yes
[30]	Dynamic informative tracking	Multi-agent deep Q-learning	UAVs	Yes

the process. Different from our approach, this work does not directly act on uncertainty, and the agent's observations are only useful for the fire-front detection. Our approach, in contrast, directly tries to minimize the uncertainty which can be a valid approach independently to the process under survey. In [30], a similar approach is solved using different architectures for the DQL algorithm. This work addresses the study of different multi-agent algorithms to monitor wildfires. The conclusions of this work serve this chapter as a guideline for choosing a centralized approach for the IPP solving using DRL. The proposed network in this chapter differs from the one used in [30] because the observation of every agent is fully visual and can include all others agent information in an image to be processed. This alleviates the need of retraining when the fleet grows in size. Another relevant aspect is that the state in this chapter proposal reunites the global information of the scenario whereas in [30] the observation is local to every agent.

3 Statement of the Problem

The main objective of this monitoring application is to find a policy that coordinates a fleet of ASVs to efficiently collect information of the WQ measurements. This section will explain the stochastic assumptions under the information criterion and the statement of the Informative Path Planning.

3.1 Information Framework

The IPP starts defining a navigation space $X \in \mathbf{R}^2$, where every vehicle in the fleet can take a measurements of the water. It is also defined the gathered samples locations subset $X_{meas} \in \mathbf{R}^2$, and a navigation map $M|X = \forall p := [p_x, p_y] \mid; M(p) = 1$. It is a reasonable hypothesis to assume that every possible point that can be sampled behaves like a Gaussian random variable with mean μ and variance c such that $p \approx \mathscr{N}(\mu, c)$. Now, we can assert that the visitable space X behaves as a Multivariate Gaussian Distribution (MGD), where $X \sim \mathscr{N}_n(\bar{\mu}, \Sigma)$, with Σ being the correlation matrix of X.

Here, we can define a function that serves as a surrogate model of the correlation as we take samples. This function will serve as an indicator of the information we have over the navigable space X depending on the measurements taken X_{meas}. As we expect the WQ distribution to be smooth to a certain level, it can be used a Matérn Kernel Function (MKF) to define the correlation between two samples in two different locations, in a smooth and exponential manner following Eq. (3). The hypothesis has sense because the WQ parameters cannot change drastically from one point to a close one like it has been studied before.[1] Consequently, near samples from the same or various agents will be highly correlated and the reduction of the uncertainty in this locations, will be high but redundant. Two parameters will model the MKF: (i) the parameter ν models the smoothness of the uncertainty decay with the samples distance $\|p, p'\|$, and (ii) the parameter l will serve to scale how much correlated two measurements (p, p') are with each other. These values are usually chosen based on prior knowledge of the environment to be monitored [20, 29] or on how intensively the environment is meant to be covered. In Fig. 4 it is depicted the effect on the uncertainty when using a MKF with $(\nu, l) = (1.5, 1)$.

$$MKF(p, p') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\|p - p'\|_2}{l}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{\|p - p'\|_2}{l}\right) \tag{3}$$

When evaluating with the kernel function every sample in a instant t, it is possible to obtain the conditional correlation matrix $\Sigma\,[X|X_{meas}]$ according with the following expression [21]:

[1] https://marmenor.upct.es/maps/.

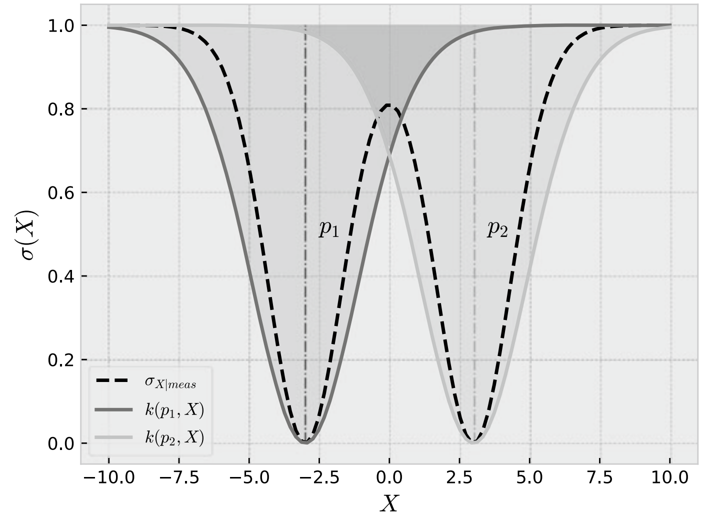


Fig. 4 Uncertainty conditioning process. When we take two samples p_1 and p_2, and incorporate to X_{meas}, it is updated $\Sigma\,[X|X_{meas}]$ according to Eq. (4). It is observed that the uncertainty $\sigma(X)$ associated with $\Sigma\,[X|X_{meas}]$ becomes zero at the sample points (assuming no sampling noise) and in their environments according to the Matérn function. The entropy $H\,[X|X_{meas}]$ of the process decreases accordingly. The intermediate uncertainty between samples supposes a redundancy in the acquisition. Thus, two very near samples makes no significant reduction to $\sigma(X)$

$$\begin{gathered}\Sigma\,[X|X_{meas}] = \\ \Sigma\,[X, X] - \Sigma\,[X, X_{meas}]\times\Sigma\,[X_{meas}, X_{meas}]^{-1} \times \Sigma\,[X_{meas}, X]^{T}\end{gathered} \tag{4}$$

Now, the monitoring objective involves decreasing the entropy associated with the conditional correlation. The information entropy $H\,[X|X_{meas}]$ gives a measure of the uncertainty about the monitoring domain and the randomness of a sample at an arbitrary point in that space. The lower the entropy, the more confidence one has about the scenario. Finally, the entropy can be calculated as [21]:

$$H\,[X|X_{meas}] = \frac{1}{2}\log(|\Sigma\,[X|X_{meas}]\,|) + \frac{dim(X)}{2}\log(2\pi e) \tag{5}$$

It can be seen that by decreasing the covariance matrix determinant, the entropy will be reduced. Reducing this determinant can be done by two ways according to [24]: by reducing directly the product of the eigenvalues of $\Sigma\,[X|X_{meas}]$[2] (called D-criterion) or by reducing the sum of the eigenvalues. The latter, called A-criterion for the uncertainty reduction, can be achieved by decreasing the trace of $\Sigma\,[X|X_{meas}]$, which is the sum of its diagonal. By reducing this values, we will be reducing the determinant of $\Sigma\,[X|X_{meas}]$ and consequently, the entropy of the gathered information:

[2] As it is demonstrated in [21] and because Σ is a positive semi-definite matrix, $|\Sigma| = \prod_{i=0}^{dim(X)} \lambda_i$.

$$Tr(\Sigma\,[X|X_{meas}]) \equiv Tr(\Sigma) = \sum_{i=0}^{dim(X)} \lambda_i = \sum_{i=0}^{dim(X)} \Sigma_{ii} \tag{6}$$

In Fig. 4 it is depicted how the uncertainty $\sigma(X)$ for one-dimensional space associated to $\Sigma\,[X|X_{meas}]$, its diagonal, decrease with two independent samples p_1 and p_2. In the vicinity of those sample points, σ is not 1.0 at all since a smooth correlation is assumed with the MKF function.

3.2 Informative Path Planning

Once the information framework is formulated, we can continue by stating the sequential decision problem under the IPP. The final objective of the IPP is has been reduced to minimize the total entropy of the information. For this work, we will take the uncertainty, the trace of $\Sigma\,[X|X_{meas}]$, $Tr(\Sigma)$, as an analogous measurement of the entropy (it has been aforementioned that they are directly related). The objective is to find a path $\Psi = \{\psi_1, \psi_2, \ldots, \psi_N\}$ for every agent in a fleet of N agents that minimize the uncertainty at the end of a mission:

$$\Psi^* = \min_{\Psi} Tr(\Sigma, \Psi) \tag{7}$$

Given that every agent can move sequentially from a point p_t^j to another p_{t+1}^j, every path ψ is composed by T possible movements. The distance d_{meas} from one position to the next one is fixed. This way, the speed of every vehicle is fixed to a constant value. Additionally, with every movement a new WQ sample is obtained for every vehicle in the fleet. With regards to obstacles, given the navigation map $\boldsymbol{M}$, which indicates if a point p is navigable or not, the solutions will be restricted to those paths that are bounded to navigable zones. This way, the fleet must coordinate to find the best solutions **starting from any arbitrary initial points** $\Psi_0 = \{\psi_1^{t=0}, ..., \psi_N^{t=0}\}$ that efficiently minimize the uncertainty without any collisions between them and the non-navigable zones. This means that taking samples on the very same place than other agent (or the same) has measured is, obviously, not desired. Finally, the IPP corollary can be enunciated as:

$$\begin{aligned} &\underset{\Psi = \{\psi_1, \ldots, \psi_N\}}{\text{minimize}} \quad Tr\,(\Sigma[X|X_{meas}(\Psi)]) \\ &\qquad\text{Subject to} \qquad \forall p \in X_{meas}(\Psi) \rightarrow M(p) = 1 \end{aligned} \tag{8}$$

In Fig. 5 it is depicted a typical situation were two vehicles intersects their paths taking redundant samples in-between.

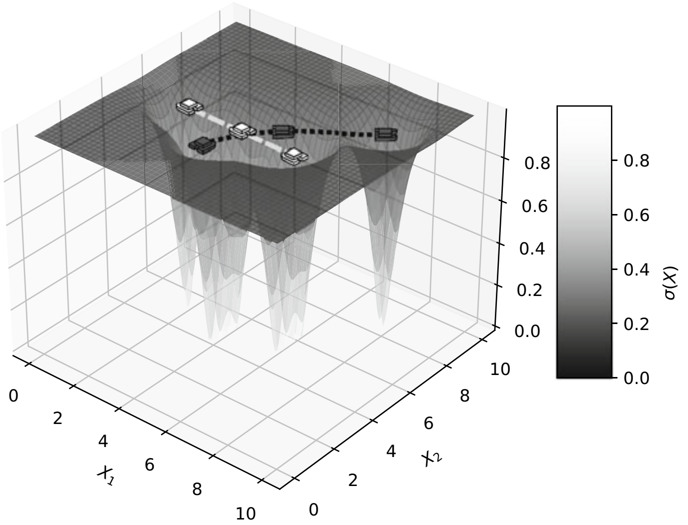


Fig. 5 When addressing the multi-agent IPP, it is necessary to consider that two very near samples are redundant, and the closer, the less interesting are both. This figure represent two intersecting path that incurs in redundancy (lower values of $\sigma(X)$). Taking samples in the very borders also is considered useless, as the uncertainty cannot be reduced outside the limits

4 Methodology

First, in this Section it is explained the Markov Decision Process that serves as the sequential framework to implement a Reinforcement Learning approach for the IPP: the state representation, the reward function and the actions. Secondly, the Deep Q-Learning approach is presented and the different sub-modules depicted: the Deep policy, the noisy implementation of the neural network and the Prioritized Buffer Replay mechanism.

4.1 Markov Decision Process

The IPP can be formulated as a Markov Decision Process (MDP) to fulfill the requirements of the Reinforcement Learning theory [25]. In any MDP there is, at least, one agent that can perform an action a_t from a set of valid actions A at instant t. This action interacts with the environment which produces a state s_{t+1} and a reward r_t. The state represents all the environment information available and, in the strict sense, it is all the information necessary to reconstruct it. The agent can observe the scenario trough an observation mechanism $o = O(s)$ that maps the environment state into an observation o. In general terms, the observation has incomplete information of the dynamics of the scenario. When the observation is perfect, $s_t \equiv o_t$, the MDP is defined to be a Fully-Observable MDP (FOMDP). In the contrary case, it is known as a Partially-Observable MDP (POMDP). Every agent also has a policy $a = \pi(o)$ that maps an observation into an action. The policy can be deterministic or stochastic, and defines the strategy the agent follows to obtain the maximum reward possible in a finite action time T:

$$\pi^*(o) = \max_{\pi} \left[\sum_{t=0}^{T} r(a_t = \pi(o_t), s_t) \right] \tag{9}$$

The optimal policy $\pi^*(o)$ is reached and the MDP is considered solved when there is no better strategy that obtains, on average, a better reward.

4.1.1 Agents Actions

As it was aforementioned in Sect. 3.2, every agent perform the same type of actions, which are movements in $|A|$ possible directions with a constant distance d_{meas}. Those actions will correspond to the cardinal directions $\{N, S, E, W, NE, NW, SE, SW\}$. With a constant speed, the fleet will move coordinately and every path in Ψ will have the same length d_{max}. Every action also involves taking a sample in the next resulting point and obtaining a new uncertainty matrix Σ.

4.1.2 The State and Observation

There is not a unique way to represent the state of this problem. For accomplish the problem, the observation made by agents must be feasible and must include all the information possible to gather the real problem state. Here we propose a full visual observation, similarly to related works such as [15, 33]. This observation is composed by the following elements of the state, as channels of an image of [58 × 38] pixels:

1. The navigation map $\mathbf{M}$: formed by an image with its values being $\{0, 1\}$. A value 1 means this position is considered non-navigable.
2. The agent j position: also formed by a binary image of the same size with 1 in the position of the agent that is making the observation.
3. The fleet state: another binary image with a 1 in those cells occupied by the observed agents positions from the perspective of the agent making the observation.
4. The uncertainty map $\sigma(X)$ which is the diagonal values of $\Sigma\,[X|X_{meas}]$ to represent the values of the trace according to Eq. 6.

In Fig. 6 it is depicted the different channels of the state.

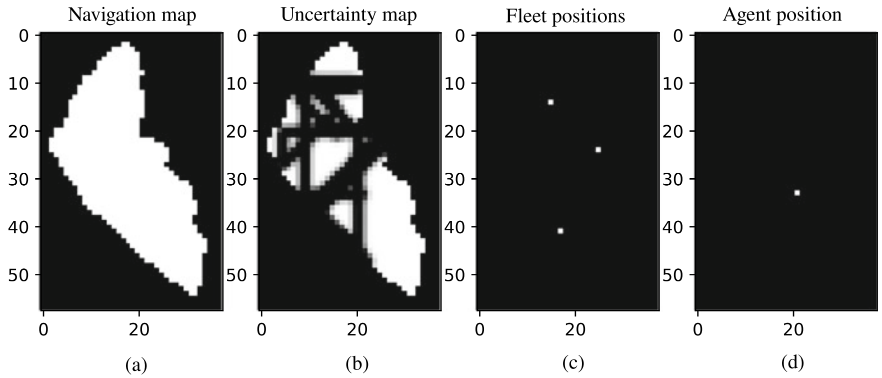


Fig. 6 An arbitrary observation o_t of an agent: **a** The navigation map **M** which is the same for every agent. **b** The uncertainty values mapped into an image. **c** The other agents apart from the observer discretized in an image. **d** The agent that observes the state within its position

4.1.3 The Reward Function

For the DRL to optimize an useful policy for the information gathering, it is vital to design an effective reward function. The reward function must evaluate how good an action is in a particular state. The total reward at the end of an episode, which is, the sum of the consecutive rewards for every agent, will represent how good the fleet performs in the given scenario. The reward must be positive for good actions and negative for bad actions. The *goodness* or *badness* of every action is something that is worth to discuss.

For this particular problem, the total minimization of the uncertainty could be a good measurement of the quality of each vehicle movement. Being a fleet action $\bar{a}_t$, with every agent i in the instant t, $\bar{a}_t = \{a_t^1, a_t^2, \ldots, a_t^N\}$ at instant t:

$$r(s_t, a_t^j) = \frac{Tr(\Sigma)^t - Tr(\Sigma)^{t-1}}{dim(Tr(\Sigma))} = \frac{\Delta U^t}{dim(Tr(\Sigma))} \tag{10}$$

This is equivalent to say that every agent will be rewarded with the total amount of uncertainty reduced *by all the fleet movements*. Here we can expect the agents to learn to minimize the uncertainty, but there is an intrinsic problem with this direct formulation. A bad action of an agent and a good action of another will assign the same credit for both agents, so the learning will be harder since agents are partially biased by others behavior. This problem, called the credit assignment problem, is typical of multi-agent games [9], and in this particular case it will be significant as the process explained in Eq. 4 is difficult to compute for different agents. Here we propose a solution for this problem using simple knowledge of the scenario dynamics: we can compute the individual contribution to the uncertainty reduction as if the agent was alone. This way, for every agent action $a_t^j \in \hat{a}_t$ it can be computed the individual contribution ΔU_j^t as:

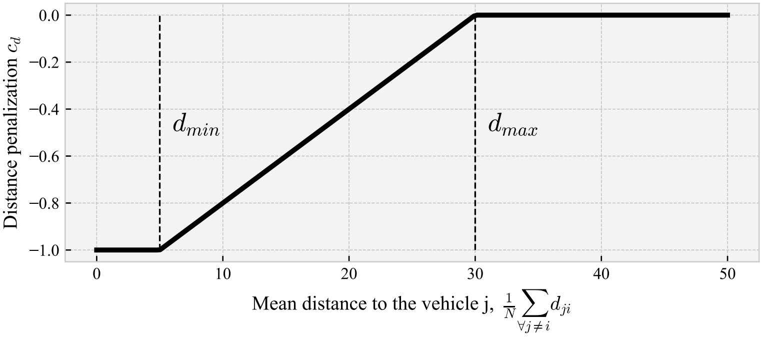


Fig. 7 The penalization for the distance will be proportional to the mean distance between the agent j and others. It is imposed that, from a maximum distance d_{max}, the penalization will be zero and vice versa for a distance close to d_{min}

$$\Delta U_j^t = Tr(\Sigma\,[X|X_{meas}]^{t-1}) - Tr(\Sigma\left[X|X_{meas}^{t-1} \cup p_j^t\right]) \tag{11}$$

This will be equivalent to considering an individualistic reward that will not take into account the redundancies of the sampling at instant t. Taking two samples in near places will be rewarded as if they were unique samples, so the reward signal must consequently be modulated in terms of the redundancy. It is not difficult to understand that the closer the agents taking samples the higher the redundancy. We propose a linear decrease of the individual contribution as a function of the distance of each agent to enhance the dispersion of the agents trough the search space. It can be imposed a clipped linear function to do so like in Fig. 7.

To address the avoidance of nonnavigable areas, it is also imposed that any movement to an invalid zone will be penalized with $c_p < 0$. The final reward function for an agent j will be:

$$r(s_j^t, a_j^t) = \begin{cases} c_p & \text{when } a \text{ involves a collision} \\ \Delta U_j^t + \lambda \times c_d(d_{ji}) & \text{otherwise} \end{cases} \tag{12}$$

4.2 Deep Q Learning

Deep Q-Learning was applied for the first time successfully in [15], were the Google Mind Team trained an agent to play several classic ATARI games to have superhuman capabilities. The success of this work resides in the use of CNNs to represent the *state-action function*, which is called the Q function. This function represents, in the sense of the Bellman equations [3], the expected discounted future reward given a state s and an action a is

$$Q(s^t, a^t) = \mathbb{E}\left[r(s^t, a^t) + \gamma \sum_{s' \in S} p(s'|s, a) \max_{a' \in A} Q(s', a') \right] \tag{13}$$

where γ is a discounted factor that will ponder the importance of earlier rewards, and $p(s'|s, a)$ is the probability of transition to s^{t+1} from the current state and given the current action.[3]

In DQL, Q will be represented by a neural network with $Q(s, a; \theta)$ (with θ being the parameters) and must be optimized by trial and error using a Stochastic Gradient Descent (SGD) algorithm to find the optimal policy according to Bellman's principle of optimality [3]:

$$\pi(s; \theta) = \arg\max_{a \in A} Q(s, a; \theta) \tag{14}$$

The deep policy $\pi(s; \theta)$ will be optimized by generating many tuples of experiences $<s^t, a^t, r^t, s^{t+1}>$ and minimizing the Bellman error E:

$$E(s^t, a^t, r^t, s^{t+1}) = Q(s^t, a^t) - \underbrace{\left[r^t + \gamma \times \max_a Q(s^{t+1}, a) \right]}_{\text{Target value } y(r^t, s^{t+1})} \tag{15}$$

This error value, usually called *time difference error*, represents the prediction error when predicting the expected future value with $Q(s, a)$. This error is minimized by taking SGD steps with respect to a loss function, so the parameters of $Q(s, a; \theta)$ move towards the direction of minimizing such value. This optimization method, as we can see in Eq. 15, uses the very same function under optimization to predict future values, also called target values. This method, named *bootstrapping*, involves certain instabilities [25] that can cause the learning to be a total failure. In [8, 15], the solution to this comes from the use of two effective methods:

1. The use of a twin target function Q to predict the values Q in the $t + 1$ instant. This function, $Q_{target}(s', a, \theta')$, is not optimized directly, but instead periodically copies its parameters θ' from $Q(s, a; \theta)$.
2. The use of an Experience Replay (ER). The ER technique consists of saving every experience in a memory and training the neural policy periodically using batches of samples randomly chosen from it. Random sampling ensures that there is no correlation between experiences to reduce network bias.

Finally, this algorithm uses the trial and error method to optimize $Q(s, a)$ so that it can be used following Eq. 14. When addressing the multi-agent problem, the same Q function can be extended to all agents as they are homologous in actions and in observations. Since the objective in this problem is intended to be fully cooperative and the interactions of every agent are included through the reward as was explained before, the neural policy can be trained using every individual state to process one action for every agent. Here, it is worth to mention that, as every agent situation is

[3] In a total deterministic environment, this probability is assumed to be 1.

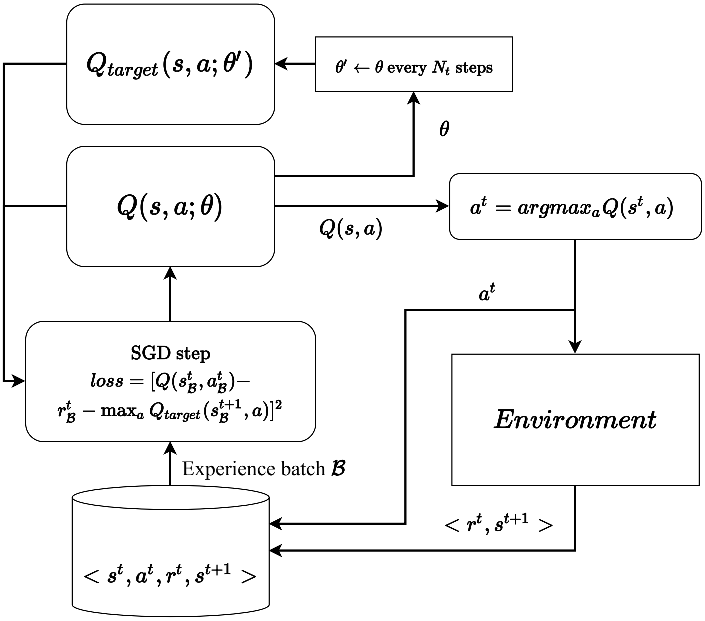

$Q_{target}(s, a; \theta')$
$\theta' \leftarrow \theta$ every N_t steps
θ
$Q(s, a; \theta)$
$Q(s, a)$
$a^t = argmax_a Q(s^t, a)$
a^t
SGD step
$loss = [Q(s^t_{\mathcal{B}}, a^t_{\mathcal{B}}) -$
$r^t_{\mathcal{B}} - \max_a Q_{target}(s^{t+1}_{\mathcal{B}}, a)]^2$
Environment
Experience batch $\mathcal{B}$
$< r^t, s^{t+1} >$
$< s^t, a^t, r^t, s^{t+1} >$

Fig. 8 Deep Q learning algorithm scheme

part of the state and the behavior of every agent evolves within learning, the dynamics of the environment becomes nonstationary. This involves a higher complexity than classical stationary problems, as in the ATARI cases [15], and, in the end, a higher need for exploration of the state-action domain (Fig. 8).

4.2.1 Prioritized Experience Replay

As it is mentioned before, the ER consists of sampling batches of previous experiences, saved in a memory buffer $\mathscr{M}$ as they occur, to fit the $Q(s, a; \theta)$ function. When the experiences are sampled with equal probability we are neglecting that some experiences could be more useful for training than others. This situation usually happens when a very unknown state is sampled and the prediction error is big. Therefore, it makes sense the possibility to sample the experiences in a no-uniform fashion. With a Prioritized Experience Replay (PER) [23], every experience in the memory $E_i \in \mathbf{M}$ has a probability p_i to be sampled for the loss computing. This probability will be proportional to the Bellman error according to Eq. 15. The higher the prediction error, the higher the probability to be sampled in the future for a better fit of the prediction. Nonetheless, it is important to let other *less erratic* experiences to be also sampled to avoid bias. This way, the probability of sampling is imposed to be dependent to a prioritizing exponent $\alpha \in [0, 1]$ that modulates the uniformity of the priorities:[4]

$$p_i = \frac{E(s_i^t, a_i^t, r_i^t, s_i^{t+1})^\alpha}{\left[\sum_{i=0}^{|\mathbf{M}|} E(s_i^t, a_i^t, r_i^t, s_i^{t+1})^\alpha\right]} \tag{16}$$

Nonetheless, this sampling is a potential source of bias because the loss will be much bigger for those experiences with high prediction errors, specially in the beginning of the learning. To modulate the importance of every error in the loss, it is computed an importance weight w_i for every experiences in the memory. This weight depends on a parameter β that will represent the level of compensation of the error in the loss. This way, the importance weight is expressed as

$$w_i = \left(\frac{1}{|\mathbf{M}|} \cdot \frac{1}{p_i}\right)^\beta \tag{17}$$

and the loss is computed with the following formula:

$$L(\mathscr{B}) = \left[w_i \cdot \frac{1}{|\mathscr{B}|} \sum_{i=0}^{|\mathscr{B}|} (Q(s_i, a_i) - y_i)^2\right] \tag{18}$$

4.2.2 Deep Policy

The convolutional policy designed to represent $Q(s, a)$ has three initial convolutional filters to extract the features of the graphical state. This first convolutional head is made up of 64, 32, and 16 filters of the same kernel size (3×3). After this, the features are flattened and processed by three linear layers of 256 neurons each. Every neuron uses a Leaky Rectifier Linear Unit (Leaky-ReLU) as the activation function. In the final stage of the policy, a dueling architecture is implemented [31]. With the dueling architecture, the value function $V(s)$ is segregated from the Advantage values $A(s, a)$ (see Eq. 19). The former represents the expected accumulated reward in the current state. The latter represents the expected action-state value with respect to the current state value.

$$A(s, a) = Q(s, a) - V(s)$$

This technique enables a better generalization of action learning in the presence of similar states, and, according to [31], the calculation of $Q(s, a)$ will be as follows (Fig. 9):

[4] $\alpha = 0$ means full uniform sampling and vice versa.

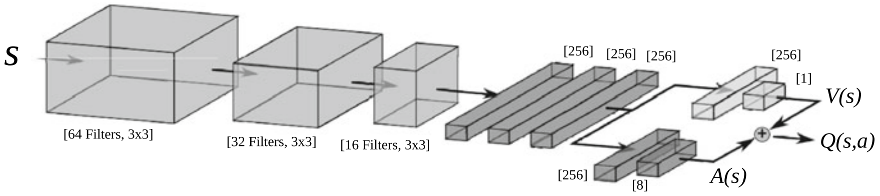


Fig. 9 The convolutional neural network implemented to process the state and to estimate the Q function. It is made up of three convolutional blocks for feature extraction and three dense noisy layers. In the end, the dense layer is decomposed into a value and advantage head

$$Q(s,a) = V(s,a) + A(s,a) - \frac{1}{|A|}\sum A(s,a) \tag{19}$$

In this work, we also propose a possible application of noisy neural networks as a technique to enhance the explorative behavior in multi-agent problems. Noisy neural networks (nNN) have been shown to be useful in complex scenarios, but as far as authors are concerned, their use has been limited to cases with a single agent [7]. With nNN, a stochastic term sampled from a Gaussian distribution is added to every weight and bias. Therefore, every neuron parameter (w_i, b_i) has two distributions $\hat{\xi} = (\xi_w, \xi_b)$. These distributions are modeled by two parameters each: (μ^w, σ^w) for the weights and (μ^b, σ^b) for the biases. With every evaluation of $Q(s,a)$, new values are sampled from $\hat{\xi}$ (see Fig. 10). This makes the policy intrinsically stochastic as different evaluations of the very same state shall return different values. This effect causes the network to have an inherent adaptive exploratory capacity. The balance between exploration of the state-action domain and the exploitation in the sense of Eq. 14 is incorporated parametrically. In classical DQL implementations, the most common exploration method is the ε-greedy policy, where there is a probability of ε of choosing a total random action. By annealing ε from 1 (at the beginning of optimization) and near 0, the agent explores. Nevertheless, in multi-agent problems like this, random exploration could be insufficient. Furthermore, the annealing values must be chosen to explore enough in the first training stages.

5 Simulation and Results

This section describes the simulation conditions for each experiment and the results obtained. First, we compare the importance of individual assignment in the reward function, as mentioned in Sect. 4.1.3. Then, the benefits of using noisy networks are explained in a comparison with an ε-greedy exploratory policy. All simulations were performed with a fleet size N of four agents. The navigation map corresponds to the Ypacaraí Lake (Paraguay) as a real case scenario (see the past Fig. 6a). All optimizations were carried out on a double Intel Xeon Silver 4210R 3.4GH, 187Gb RAM, and a Nvidia RTX 3090 with 24Gb VRAM. All hyperparameters of the algorithms are resumed in Table 2.

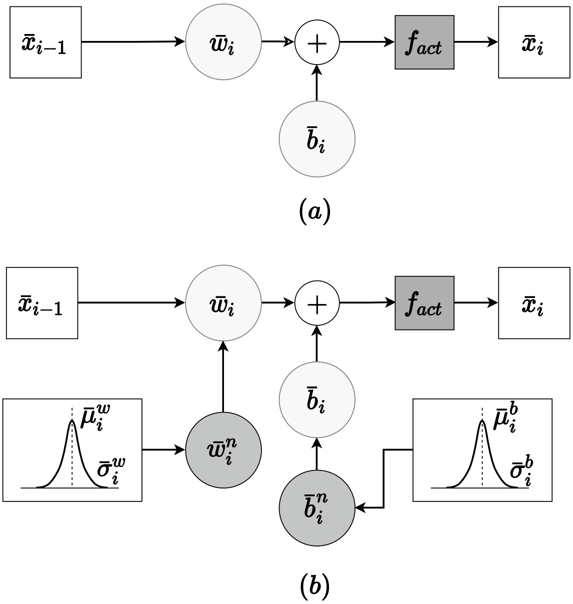


Fig. 10 In **a** a classic neural network layer with parameters $\bar{w}$ and biases $\bar{b}$. In **b** a noisy layer is presented. For each previous parameter, a stochastic value is added by sampling a Gaussian distribution of mean μ and deviation σ. Note that these parameters are trainable and will modulate the stochasticity of the policy

Table 2 Hyperparameters and scenario values used in the DRL training experiments

Hyperparameter	Value
Learning rate (lr)	1×10^{-4}
Batch size ($\lvert\mathscr{B}\rvert$)	64
Memory replay size	1×10^{6}
Update frequency for target	1000 epochs
Discount factor (γ)	0.99
Loss function	Mean squared error
SGD optimizer	Adam
ε value (if applied)	[1, 0.05] (annealed)
ε decay interval (% of learning progress)	[0, 33]
Prioritizing exponent (α)	0.4
Importance exponent (β)	[0.6, 1] (annealed)
Number of episodes	50.000
Scenario parameter	Value
Matérn Kernel lenghtscale (l)	0.75
Matérn Kernel differential parameter (ν)	1.5
Path maximum length (d_{max})	29 km
ASV speed	$2\,\mathrm{ms}^{-1}$
Fleet size N	4
Max. collisions per missions permitted	5

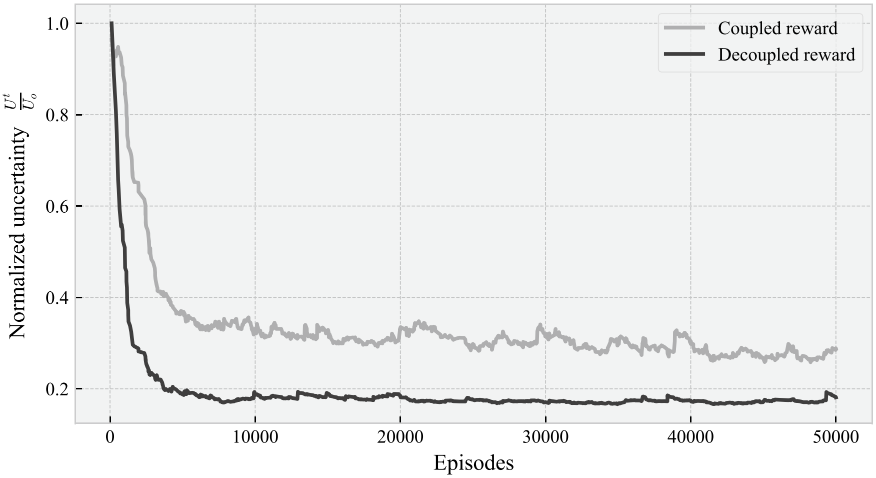


Fig. 11 Normalized uncertainty $\hat{U}$ at the end of the mission along the training process for two variants of the reward: coupled and decoupled

The hyperparametrization of this algorithm has been conducted following previous values in similar approaches [33, 35]. It has been observed that the selected batch size is adequate for this problem and decreasing the number of experiences per SGD step only augments the variance. Other parameters such as the Prioritized Replay or the discount factor, are taken from the original papers [15, 23].

5.1 Reward Function Comparison

In order to validate the reward function in terms of the credit assignment, two variants are tested: (i) the total uncertainty reduced from one instant to another is rewarded equally to each agent in the sense of Eq. 7 (coupled reward), and (ii) every agent receives only reward for its own contribution without any considerations (decoupled reward; see Eq. 12). Note that the distance penalization is applied exactly the same for both cases.

In Fig. 11, it can be seen the uncertainty over the training progress for both variations. With the decoupled reward, the algorithm is able to resolve better the credit assignment and obtain a better policy than in the coupled case. There is a 40% improvement on average from one case to another, which is significant to the exploration objective. The behavior of the coupled reward agents tend to be more redundant as can be seen in Fig. 12b. As the reward does not properly evaluates the individual contribution of every agent, the policy tend to overestimate its behavior by producing *lazy* agents. Since a good performance of one agent augments the valor of Q for the others, this leads to unequal contributions to the task. The proposed algorithm is able

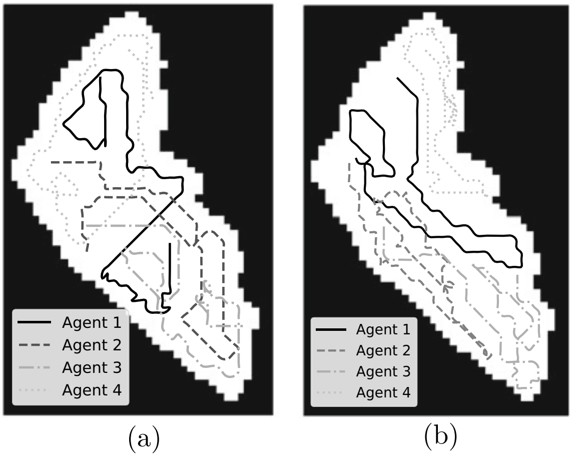


Fig. 12 Resulting paths of the same mission using **a** the decoupled reward and **b** the coupled one, both with a nNN policy

to synthesize good policies as seen in Fig. 12a, where the Lake scenario is almost completely covered and the agents tend to move more efficiently across the search space with much less self intersecting trajectories.

5.2 Exploration Efficiency

When addressing the exploration ability of nNN, it has been observed that the speed of convergence and performance of noisy neural network overcomes in this case the ε-greedy implementation.[5] In Fig. 13, it can be seen the accumulated reward the agents receive on average for both trainings. The very first steps of the ε-greedy training consists of an exploration phase where the actions are randomly taken. The performance of the training slowly increases as ε decreases over time. This exploration period must be selected and properly tuned. With less exploration (see ε-greedy in the same figure), we have observed a worse performance. On the contrary, the nNN policy needs many less episodes to converge to a high performance policy. Whereas in the ε-greedy case it is necessary to process 16,500 episodes to greedily exploit the policy, the nNN is able to find collision-free paths in 1000 episodes and higher rewards in 5000 episodes. In terms of the uncertainty, the noisy proposal is able to obtain the same performance if not slightly better (1% better on average) for higher explorative ε-greedy values. This means, for the same performance, the algorithm takes 5 times more episodes. Nonetheless, with less exploration, the agent convergence is faster but the performance becomes worse with higher variation). This indicates similar policies are encountered but the enhanced exploration policy of nNN allows a more sample efficient training without any $epsilon$ decay tuning.

[5] From this point, the decoupled reward is selected for better performance.

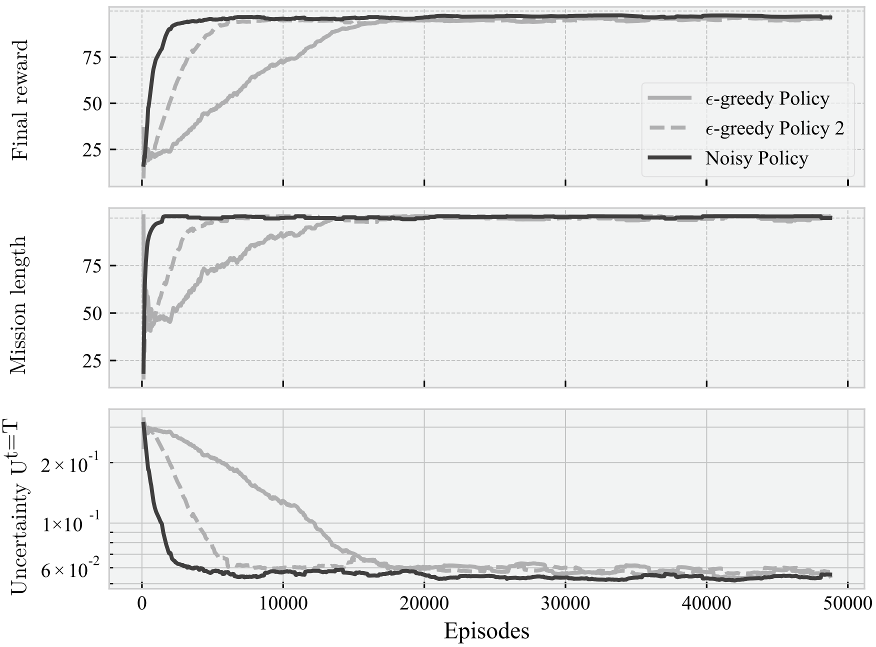


Fig. 13 Average accumulated reward (up), mission length (middle), and final uncertainty (bottom), over the training process for an ε-greedy policy and the proposed noisy policy

5.3 Comparison with Other Algorithms

In order to validate the performance of the proposed framework, we propose different algorithms to compare the policy efficiency. The metrics under study will be: (i) the average accumulated reward at the end of the episode among the agents AER, (ii) the remaining uncertainty at the end of the episode $U^{t=T}$, the mean distance between ASVs, and finally, the root mean squared error (RMSE) of the inferred WQ model respect to the ground truth. For the latter, we propose Gaussian Process (GPs), like in [18], to perform a regression with the sampled values as they are acquired. The proposed benchmark used to represent the WQ scalar field will be a sum of positive peaks and negatives peaks using two inverse Shekel functions.[6] This scalar field disposes a random number of peaks across the navigable surface to represent the maxima and minima of the water quality parameters with different intensities each (see Fig. 14).

With regards to the benchmark algorithms, we present different heuristics and optimization methods based in previous proposed approaches:

[6] See https://deap.readthedocs.io/en/master/api/benchmarks.html for the complete definition.

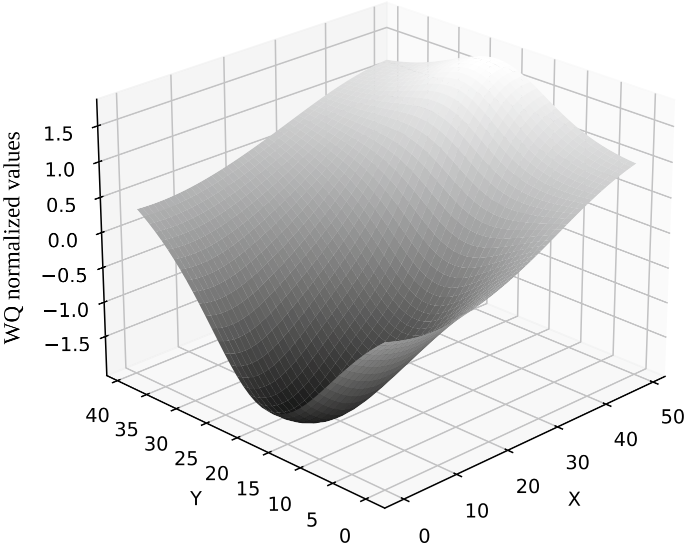


Fig. 14 Example of a WQ scalar field used for RMSE evaluation by means of a Shekel function

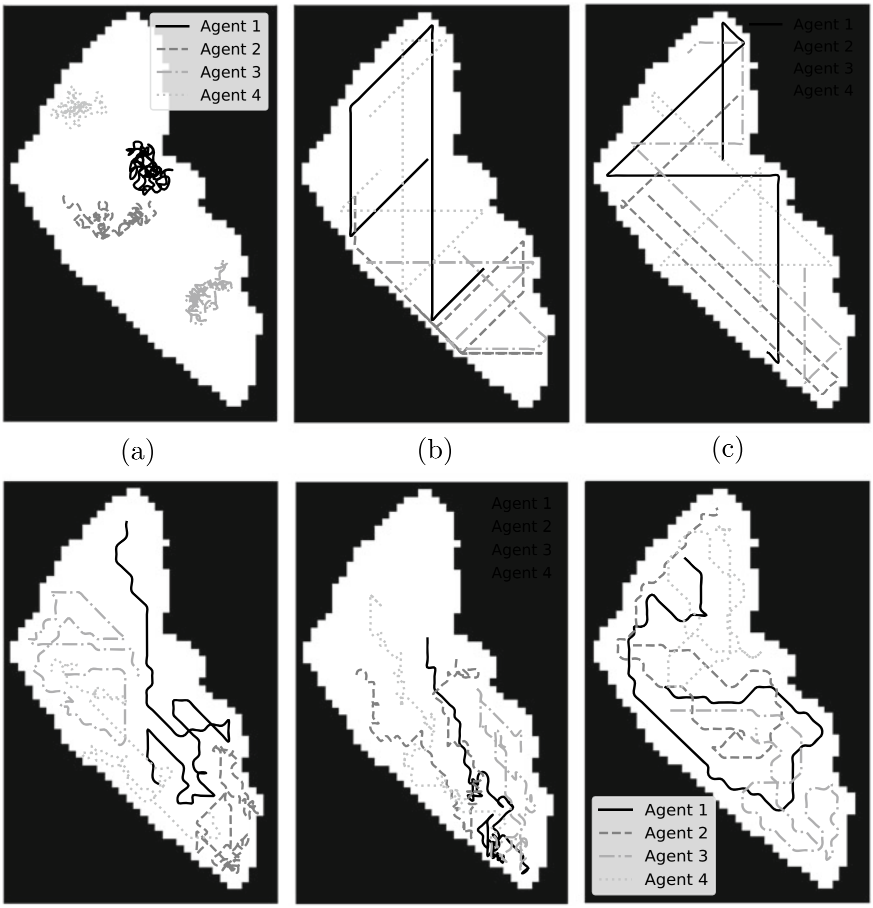
1. **Random Safe Search** (RSS): This heuristic will generate random paths without collisions of any kind. This is not a good path planning strategy, but serves as a lower bound to validate the problem complexity.
2. **Random Wandering** (RW): In this heuristic, agents will travel following straight trajectories until an obstacle is reached. Then, the agent chooses a new straight path different from the last direction to avoid going back in its footsteps.
3. **offline Genetic Algorithm Optimization** (Offline-GA): This approach will be similar to [1], were a GA is used to optimize the paths. For every evaluation scenario, paths are optimized using mutation, crossover, and selection operations to obtain the best collective path. The designed reward is used as the fitness function for each individual (here individual refers to the ASVs paths). To avoid colliding paths, the death penalty is applied for individuals that contains collisions. Note that, for every different starting conditions, Offline-GA must re-optimize, which makes this algorithm unfeasible for online deploy. This algorithm will serve as well as a performance indicator of how high the reward can be achieved when intensively optimizing every individual case.
4. **Receding Horizon Optimization** (RH): In the RH approach, the paths are optimized according to the reward function up to an horizon of H steps from the current point using a copy of the current online scenario. Once the optimization budget is reached, only the first action for every ASV is processed and a new H-step optimization begins. This approach is similar to an Model Predictive Control (MPC) algorithm used in [10] which can be useful in online optimization scenarios.

Table 3 Statistical results of evaluating the proposed algorithm and other benchmarks in 10 scenarios with different starting points and WQ scalar fields

Metric	AER		$U^{t=T}$		Distance		RMSE	
Algorithm	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
RSS	41.039	6.65	0.23	0.019	18.64	2.63	0.152	0.233
RW	70.970	8.90	0.13	0.028	18.13	5.09	0.027	0.035
Offline-GA	92.112	4.34	0.062	0.005	19.71	4.54	0.0057	0.005
RH $_{H=5}$	77.688	8.50	0.117	0.026	17.46	4.43	0.0452	0.062
RH $_{H=10}$	80.114	9.72	0.109	0.031	18.25	4.45	0.0285	0.034
DRL	97.013	1.45	0.053	0.003	18.94	4.46	0.0053	0.003

In Table 3, the results of the different approaches are presented. It has been evaluated every algorithm in 10 test scenarios with the same starting conditions and WQ ground truths. In terms of the AER metric, the DRL approach obtains better results than others algorithms with a lower deviation which indicates a more reliable and stable behavior. The final uncertainty after a mission is consequently reduced to 0.053 on average, a 14% less respect to the second best-performance algorithm Offline-GA. With respect to the estimation error, the DRL approach obtains a better accuracy than any other approach. Regardless the estimation method could affect this metric, in terms of performance, it can be seen that the most informative paths (those with lower uncertainty) obtains better models of the WQ parameters. In Fig. 16, are presented example paths generated using the aforementioned algorithms. The RW heuristic, despite being aleatory, tend to generate long paths across the lake surface but with unnecessary redundancy and self intersecting trajectories. This strategy is enhanced in the Offline-GA, where the redundancy is reduced with excellent performance. Nonetheless, it happens that as it is easier to optimize straight paths it is difficult to improve those paths, since big zones of the surface remain unvisited. The DRL approach, on the contrary, not only learns how to resolve almost every scenario, but also learns that those visits adjacent to the shores return lower rewards. This generates a shore-aware behavior and those areas are avoided. Nonetheless, the GA algorithm is able to obtain acceptable solutions but at the cost of a highly intensive computation *for every single case*. With respect to the receding horizon, we have observed that it is prone to stall in local minima. With higher prediction horizons, the algorithm have difficulties in finding good local sub-solutions and with lower ones, the performance is higher but still sub-optimal.

If we consider the progress of the uncertainty reduction over a mission progress, we can see in Fig. 15 how the RMSE score decreases along with the uncertainty. The DRL approach obtains lower results (14 and 7% improvement respect to the Offline-GA approach) in less time and with less deviation.



(a) (b) (c)

(d) (e) (g)

Fig. 15 Resulting paths of the proposed algorithms for comparison and the DRL approach: **a** RSS, **b** RW, **c** Offline-GA, **d** RH ($H = 5$), **e** RH ($H = 10$), and **f** DRL

6 Discussion of the Results

The use of the DRL involves always the design of a reward function as a way to describe the desired behavior of the policy. The results have shown that, in the particular IPP case, it is vital to design a proper function in order to achieve a good performance. This situation aggravated when the state-action space increases which is the case of every multi-agent paradigm. The multi-agent IPP has demonstrated to need a decoupling of the real information dynamic in order to cope with the credit assignment problem: when addressing this, it is better to consider individual contributions to the task and add an additional term for redundancy such as the

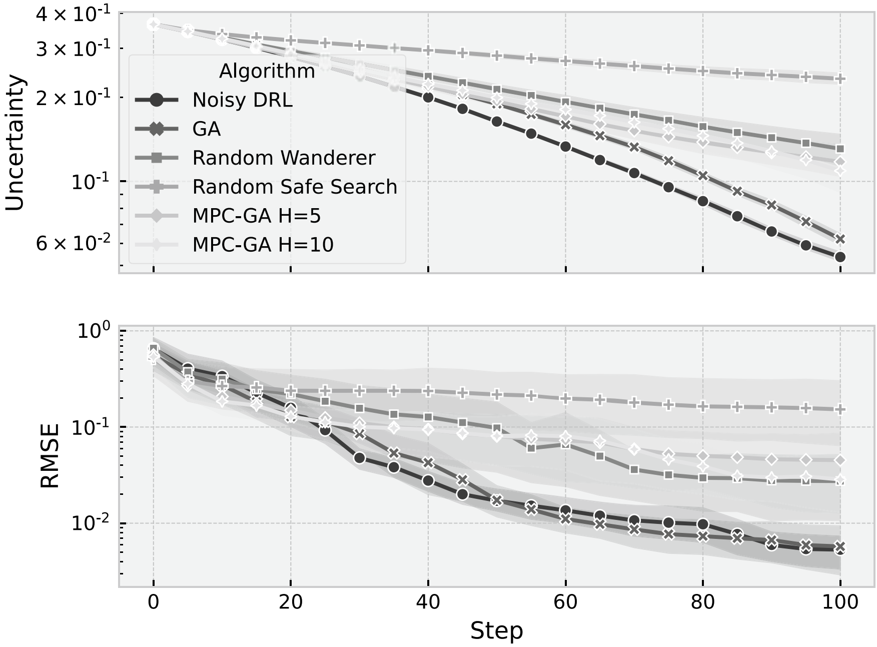


Fig. 16 Example of a WQ scalar field used for RMSE evaluation by means of a Shekel function

distance. According to our simulations the improvement could reach up to a 84% respect to the couple baseline. This modification, while it involves more calculations, it is absolutely necessary for the DRL to converge in a competitive policy.

Regarding the explorative behavior of the two variants under test, ε-greedy and noisy networks, it is clear that the noisy layers can boost up the algorithm efficiency to adjust the explorative behavior as part of the parameters of the network. This method, while it involves more parameters to tune, avoid the tedious task of fine-tuning the ε-greedy parameters. It has been observed a decrease in performance when the explorative behavior is not enough. For contrary, when the exploration phase is bigger in time, the policy performance is not as high as the obtained using nNN. This fact presents a discussion about the exploration needs of every complex or multi-agent problem: how much is it necessary to explore? Additionally, it could happen that the full random behavior of the agents are ineffective and some form of parametric noise is required (as it happens in much more complex games like ATARI Moctezuma's Revenge [7]). In this particular case, although the scope is limited, the nNN have resulted better in sample efficiency and in terms of the obtained score.

When addressing the comparison with other approaches, DRL have proven to be a good methodology to cope with the IPP objectives. The algorithm is able to deal with arbitrary initial points and obtain top performance among other approaches. When comparing the results of the DRL-trained policy with the Offline-GA, it can be observed that the scores are better in RMSE and uncertainty, which validates the proposal for online deploy. Although we obtain good performance with the Offline-GA, it is only useful to optimize scenarios one-by-one, evaluating several solutions for the same scenario. In the simulations, DRL evaluates the same number of times the simulator but with different missions and starting points. This way, the policy generalized the task independently to the fleet situation. The fact that the DRL policy overcomes the Offline-GA performance without any in-place optimization, validates its ability to generally solve the IPP.

Another important task that it is worth to discuss is the ability to reduce the model error for every algorithm. It can be seen that the lower the uncertainty, the lower the RMSE. Nonetheless, the lowest bound of the error will be related not only to the acquisition path Ψ, but also the regression method used and the ground truth itself. For this case, a plausible WQ parameter benchmark has been used, following the behavior of WQ parameters in the Mar Menor (Spain) and previous works [17, 18]. Nonetheless, when different benchmarks are used representing other variable distributions, the error could behaves differently from our case. In the end, the error reduction is dependent on a suitable regression method. Once the regression method comply with the dynamics of what we want to measure, the uncertainty criteria serves to reduce this error in a fast and powerful way.

7 Conclusions

In this chapter a Deep Reinforcement Learning approach have been proposed to deal with the Informative Path Planning problem using multiple surface vehicles. The proposed informative framework uses a Kernel function to correlate the samples in order to obtain the level of uncertainty remaining in the scenario under monitoring. Within this problem it is also considered the nonnavigable zones of the Ypacaraí Lake as a testing scenario. All this restrictions are considered in a tailored reward function that take multi-agent credit assignment in consideration. The proposed reward function deals better with the IPP problem when the individual contributions to the uncertainty are explicit when compared to using the total reduced uncertainty for every agent. It has been also proposed the use of several mechanisms of DRL that have been useful in previous works: Prioritized Replay, Advantage networks, and noisy neurons. The latter, have been observed to enhance the exploration and sample efficiency of the algorithm respect to the classic ε-greedy strategy. In the end, the algorithm had been compared to many other heuristics to validate its performance. The algorithm is able to perform better than any other with a 20% higher rewards respect to the receding horizon counterpart or 5% better to the intensive offline-GA optimization. In future lines, it is intended to study how this problem can be expanded to an arbitrary number of agents in every mission, instead of selecting a fixed size per training. This would result in a fleet-size concerned policy that considers the appearing and disappearing of agents in the course of a mission. Other interesting aspect to investigate is how the action and state formulation affects the learning. It is worth to study if noisy state inputs could help in generalization with different boundaries or with moving obstacles in the middle of the navigable zones.

Acknowledgements This work has been funded by the Spanish "Ministerio de Ciencia, Innovación y Universidades" under the PhD grant FPU-2020 (Formación del Profesorado Universitario) of Samuel Yanes Luis.

References

1. Arzamendia M, Gregor D, Gutierrez-Reina D, Toral S (2019) An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of ypacarai lake. Soft Comput 23(5):1723–1734
2. Arzamendia M, Gutierrez D, Toral S, Gregor D, Asimakopoulou E, Bessis N (2019) Intelligent online learning strategy for an autonomous surface vehicle in lake environments using evolutionary computation. IEEE Intell Transp Syst Mag 11(4):110–125
3. Bellman RE (2003) Dynamic Programming. Dover Publications Inc, USA
4. Coley K (2015) Unmanned surface vehicles: the future of data-collection. Ocean Chall 21:14–15
5. Cover TM, Thomas JA (2006) Elements of information theory. Wiley Series in telecommunications and signal processing. Wiley-Interscience, USA
6. Ferreira H, Almeida C, Martins A, Almeida J, Dias N, Dias A, Silva E (2009) Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle. In: OCEANS 2009-EUROPE, pp 1–6. https://doi.org/10.1109/OCEANSE.2009.5278235
7. Fortunato M, Azar MG, Piot B, Menick J, Osband I, Graves A, Mnih V, Munos R, Hassabis D, Pietquin O, Blundell C, Legg S (2017) Noisy networks for exploration. CoRR arXiv:1706.10295
8. van Hasselt H, Guez A, Silver D (2015) Deep reinforcement learning with double Q-learning. CoRR arXiv:1509.06461
9. Hoen PJ, Tuyls K, Panait L, Luke S, La Poutré JA (2006) An overview of cooperative and competitive multiagent learning. In: Tuyls K, Hoen PJ, Verbeeck K, Sen S (eds) Learning and adaption in multi-agent systems. Springer, Berlin, Heidelberg, pp 1–46
10. Julian KD, Kochenderfer MJ (2018) Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning. CoRR arXiv:1810.04244
11. Kathen MJT, Flores IJ, Reina DG (2021) An informative path planner for a swarm of ASVs based on an enhanced PSO with gaussian surrogate model components intended for water monitoring applications. Electronics 10(13):1605
12. Krishna Lakshmanan A, Elara Mohan R, Ramalingam B, Vu Le A, Veerajagadeshwar P, Tiwari K, Ilyas M (2020) Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot. Autom Constr 112(May 2019):103078. https://doi.org/10.1016/j.autcon.2020.103078
13. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. In: Bengio Y, LeCun Y (eds) ICLR, http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15
14. Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. NIPS'17, Curran Associates Inc., Red Hook, NY, USA
15. Mnih V, Kavukcuoglu K, Silver D et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236

16. Murphy RR, Steimle E, Griffin C, Cullins C, Hall M, Pratt K (2008) Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane Wilma. J Field Robot 25(3):164–180. https://doi.org/10.1002/rob.20235
17. Peralta F, Reina DG, Toral S, Arzamendia M, Gregor D (2021) A Bayesian optimization approach for multi-function estimation for environmental monitoring using an autonomous surface vehicle: Ypacarai lake case study. Electronics 10(8):963
18. Peralta Samaniego F, Reina DG, Toral Marín SL, Gregor DO, Arzamendia M (2021) A Bayesian optimization approach for water resources monitoring through an autonomous surface vehicle: the ypacarai lake case study. IEEE Access 9(1):9163–9179. https://doi.org/10.1109/ACCESS.2021.3050934
19. Piciarelli C, Foresti GL (2019) Drone patrolling with reinforcement learning. ACM Int Conf Proc Ser 1:1–6. https://doi.org/10.1145/3349801.3349805
20. Popović M, Vidal-Calleja T, Hitz G (2020) An informative path planning framework for UAV-based terrain monitoring. Auton Robot 44:889–911. https://doi.org/10.1007/s10514-020-09903-2
21. Rasmussen C, Williams C (2006) Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, Cambridge, MA, USA. https://doi.org/10.7551/mitpress/3206.003.0001
22. Sánchez-García J, García-Campos J, Arzamendia M, Reina D, Toral S, Gregor D (2018) A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications. Comput Commun 119:43–65. https://doi.org/10.1016/j.comcom.2018.02.002
23. Schaul T, Quan J, Antonoglou I, Silver D (2015) Prioritized experience replay. arXiv:1511.05952
24. Sim R, Roy N (2005) Global a-optimal robot exploration in slam. pp 661–666. https://doi.org/10.1109/ROBOT.2005.1570193
25. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. A Bradford Book, Cambridge, MA, USA
26. Ten Kathen MJ, Flores IJ, Reina DG (2021) A comparison of PSO-based informative path planners for autonomous surface vehicles for water resource monitoring. In: 7th international conference on machine learning technologies (ICMLT 2022). ACM
27. Ten Kathen MJ, Reina DG, Flores IJ (2021) A comparison of PSO-based informative path planners for detecting pollution peaks of the Ypacarai lake with autonomous surface vehicles. In: International conference on optimization and learning (OLA'2022)
28. Theile M, Bayerlein H, Nai R, Gesbert D, Caccamo M (2020) UAV coverage path planning under varying power constraints using deep reinforcement learning. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 1444–1449
29. Viseras A, Garcia R (2019) Deepig: multi-robot information gathering with deep reinforcement learning. IEEE Robot Autom Lett 4(3):3059–3066. https://doi.org/10.1109/LRA.2019.2924839
30. Viseras A, Meißner M, Marchal J (2021) Wildfire front monitoring with multiple UAVs using deep Q-learning. IEEE Access 1–1. https://doi.org/10.1109/ACCESS.2021.3055651
31. Wang Z, de Freitas N, Lanctot M (2015) Dueling network architectures for deep reinforcement learning. CoRR arXiv:1511.06581
32. Woo J, Kim N (2020) Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. Ocean Eng 199(107):001. https://doi.org/10.1016/j.oceaneng.2020.107001. www.sciencedirect.com/science/article/pii/S0029801820300792
33. Yanes Luis S, Reina DG, Toral Marín SL (2020) A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: the Ypacarai lake case. IEEE Access 6(1):1–1. https://doi.org/10.1109/ACCESS.2020.3036938
34. Yanes Luis S, Reina DG, Marín SLT (2021) A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: the Ypacaraí lake patrolling case. IEEE Access 9:17,084–17,099

35. Yanes Luis S, Gutiérrez-Reina D, Toral Marin S (2021) A dimensional comparison between evolutionary algorithm and deep reinforcement learning methodologies for autonomous surface vehicles with water quality sensors. Sensors 21(8). https://doi.org/10.3390/s21082862. https://www.mdpi.com/1424-8220/21/8/2862
36. Yanes Luis S, Peralta F, Tapia Córdoba A, Rodríguez Álvaro, del Nozal Toral, Marín S, Gutiérrez Reina D (2022) An evolutionary multi-objective path planning of a fleet of ASVs for patrolling water resources. Eng Appl Artif Intell 112(104):852 www.sciencedirect.com/science/article/pii/S0952197622001051
37. Zhang Q, Lin J, Sha Q, He B, Li G (2020) Deep interactive reinforcement learning for path following of autonomous underwater vehicle. CoRR arXiv:2001.03359

Analysis of Computer Vision-Based Techniques for the Recognition of Landing Platforms for UAVs

J. A. García-Pulido and G. Pajares

Abstract Although advanced UAVs use self-guided landing with the support of GPS devices, issues such as signal loss, support for indoor environments, accuracy issues, etc., must be considered, which contribute greatly to the overall success of autonomous flight. Therefore, this chapter proposes two different methods that allow for the recognition of a landing platform using Computer Vision techniques in order to assist autonomous landing. The first method is based on an Expert System that allows for the recognition of a patented black and white platform by performing a geometric analysis of the regions based on thresholds that allow for a degree of plane distortion. The second method, based on Cognitive Computing, can be used to solve the limitations to plane distortion inherent to the first approach, and further uses a specific landing platform with six different colors in order to combine color segmentation techniques with pattern recognition algorithms, together with an intelligent geometric analysis based on a decision-making technique. As a result, recognition can be achieved at different ranges and inclination angles between the vision system and the platform. It is not affected by distortions to the image figures due to perspective projection, even making it possible to perform the recognition with only a partial view of the platform, something that has received scant attention in the literature to date. The novelty is therefore the robustness and precision in the recognition from a wide variety of perspectives, different lighting conditions, and even problems that result in only a partial view of the platform, such as those resulting from partial focus or blind spots due to overexposure.

J. A. García-Pulido (✉)
Department of Computer Science and Automatic Control, Higher Technical School of Computer Science Engineering ETSII (Escuela Técnica Superior Ingeniería Informática), Universidad Nacional de Educación a Distancia, 28040 Madrid, Spain
e-mail: jgarcia5695@alumno.uned.es

G. Pajares
Instituto de Tecnología del Conocimiento (Institute of Knowledge Technology), Universidad Complutense, 28040 Madrid, Spain

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_3

Keywords UAV · Landing platform · Pattern recognition · Artificial intelligence · Recognition probability · Perception system · Thresholding · Binarization · Segmentation

1 Introduction

In a similar manner to robots, Unmanned Aerial Vehicles (UAVs) are excellent tools that are coming into rapid and growing demand because of their general utility for a variety of purposes. In order to assist and complement GPS in autonomous UAV flight, landing platform recognition is a common practice used by various researchers that is under extensive study. Therefore, this chapter proposes two different methods based on the recognition of certain regions (markers) that from part of a landing platform in color images using computer vision techniques.

On the one hand, a novel automatic Expert System [1] that recognizes the platform in an image was initially developed. The platform contains a black geometric figure drawn on a white background [2], as used in the SALACOM project [3] funded by the Ministry of Economy and Competitiveness in Spain. The system designed incorporates human expert knowledge using a sequence of high-level vision and image processing techniques; as a result, a smart application for the recognition of the landing platform has been developed. It is reliable enough to allow for its incorporation in a UAV equipped with a color camera, being especially robust for the rotary wing UAVs.

During the experimental phase, the problems inherent to this type of design have been addressed, as principally derived from two adverse circumstances: (1) the reflections originating from the natural lighting conditions that result in the disappearance of certain parts of the figure as a consequence of the saturation of intensity levels, causing black parts to appear white as a result of reflection; and (2) the necessity of interrelating regions, by extracting relative measures, causes the distortion of the figures, for example, circles are perceived as ellipses, or the same disappearance of regions is observed due to the above-mentioned saturation, with origin-relative measurements being insufficient in themselves for recognition.

In this sense, and to overcome this drawback, a new strategy based on a Cognitive Computation paradigm has been proposed [4]. Both the color image processing and pattern recognition techniques have been combined in a Euclidean Distance Geometric Analysis (EDGA). This has led to the design of a new specific platform for identification that allows the above problems to be solved. The design of this new platform is based on (1) color information in the visible spectral range, with unique colors for each figure that otherwise have little or no presence in natural and urban environments, and (2) geometric figures, which allow for partial individualized perception. This method calculates the degree of recognition of the platform from the overall figures detected in conjunction with the geometric relationships between them and, in this way, with these two groups, a decision can be made about recognition.

The experimental evidence validates the two recognition procedures developed, taking into account the fact that the problem identified during the first approach will have been fixed for the second.

1.1 General Considerations

Machine vision systems are highly useful devices that see common usage on board UAVs (Unmanned Aerial Vehicles) to perform various tasks, such as crop and weed identification in corn fields [5], road traffic management [6], or flood damage review [7], to name but a few. Therefore, their use is growing significantly for a wide range of purposes and missions [8]. Some of these missions are related to surveillance tasks, such as the identification and rescue of people, or the monitoring of civil infrastructure [9], while others address navigation and obstacle avoidance. In addition, some e-commerce multinationals, such as Amazon, are developing systems to deliver packages to their customers using a landing platform with a code that tells the UAV where to deliver the package, and, consequently, where to land, or at least approach spatially [10].

Autonomous landing of UAVs is a complex task with several associated problems that demand special attention, and which are, accordingly, areas of intense research. Once the mission is about to be completed, the UAV must land in a safe place, which can be determined via a landing platform that features a known pattern or structure. Most of these missions involve following a planned trajectory that requires a precise and reliable landing. Such a landing is a particularly complex maneuver, mainly because, as mentioned previously, the signal coming from the GPS system that guides the flight often becomes ineffective, inaccurate, or non-existent for indoor scenarios.

The use of an artificial vision system, together with a landing platform and a specific algorithm on board the UAV that is responsible for recognizing it, represents an excellent tool with which to overcome this problem. However, as a previous step to the incorporation of the recognition system on board a UAV with operational nature, it has been necessary to experiment in different scenarios on the ground in order to assess the efficiency of the methods developed. This particular aspect of automated landing constitutes the main objective of this proposal.

1.2 Proposed Methods

Although the navigation system used by UAVs is essentially based on a global positioning satellite system (GPS), the use of an approach platform is required to determine the orientation and relative position of the vehicle with respect to its base in order to reach the exact landing site at the end of the landing phase. Likewise, the

GPS receiver allows the vehicle to approach the surrounding zone where the platform is located. The platform itself then serves as a means of self-orientation and determines the exact place for landing.

To achieve this objective, the platform can be designed according to different configurations, including symbols and specific geometric figures, which facilitate its identification in the images captured by the UAV's on-board camera, where the camera operates in the visible spectrum. To this end, specific computer vision techniques are used, conveniently structured, which transfer human knowledge and reasoning to the artificial vision system. For this purpose, two different methods that can be used to address this problem are described, both under the design of two platforms that are specific to each.

Therefore, two different patterns have been used or designed, which have to be recognized according to each applied method. The first method, hereafter referred to as method 1, is based on an Expert System [1] and whose aim is to recognize the platform shown in Fig. 1.

The second proposed method, identified herein as method 2, which uses a combination of color and geometric relations, is based on the design of a new platform with the associated methodology also in the field of computer vision, and whose objective is to solve the problems that arose during the experiments carried out with the platform used in method 1.

The underlying idea for such a design is based on the fact that, instead of using complex geometric figures based on circles or ellipses, six simpler geometric figures are used that are each entirely different from the others, as can be seen in Fig. 2. In addition, another key aspect is that each of the six figures has a unique color. In this way, the platform can be identified by the differentiation between the geometric shapes and the spectral variations in the visible range.

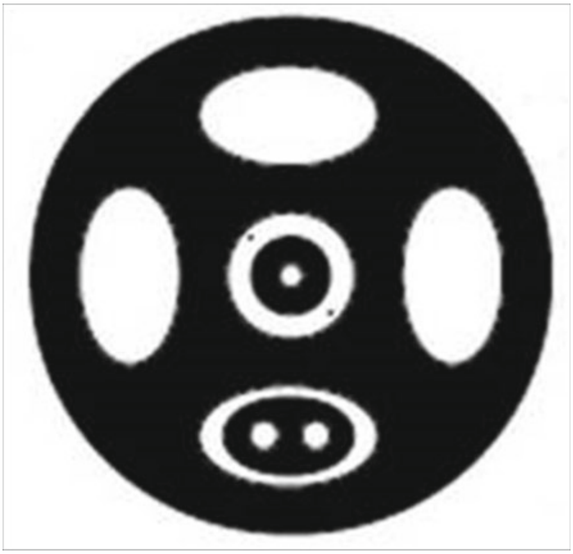


Fig. 1 Landing platform for Expert System method [2]

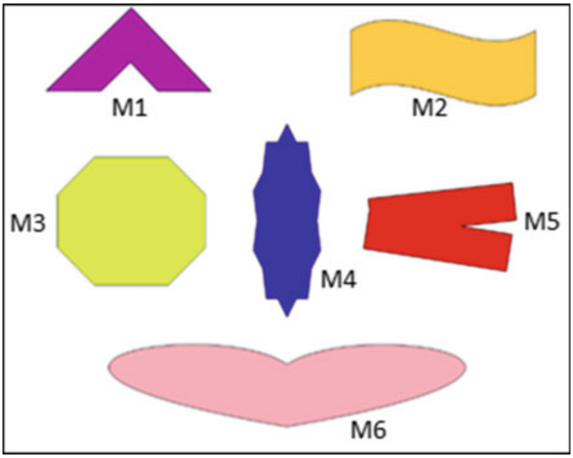


Fig. 2 Landing platform for Cognitive Computation method with marker names

1.3 Problem Description

From the second premise postulated above, it is feasible to intuit the inherent difficulty involved in an image acquisition process, particularly aerial, in an outdoor environment where it is difficult to control relevant parameters such as the level of luminosity, the degree of inclination of the capture device (camera) with respect to the platform, or the distance between the camera and the platform itself.

Taking the platform in Fig. 1 as a reference, this gives rise to several adverse effects, as shown in Fig. 3. In fact, in part (a) of this figure there is an intense reflection in the lower part due to the luminosity, which hinders the segmentation of the figure for subsequent identification. In part (b), the effects produced by the inclination of the camera and its distance from the platform are clearly perceptible, showing the geometric distortion of the figure, and the fact that the figure can barely be differentiated from the background.

Add to this the fact that the vehicle's autonomy is limited, there is, therefore, only a certain amount of time is available (in an aerial environment it is even more important since the UAV can fall uncontrollably) to take an image of the platform that is of sufficiently good quality to identify the figure via the use of computer vision techniques; indeed, this problem has no trivial solution.

1.4 Outlines

In order to describe the research, this study is structured in different chapters, the first of which is Chapter "Deep Reinforcement Learning Applied to Multi-agent Informative Path Planning in Environmental Missions", where a study of the state of the art is carried out and an analysis of the main patents in this field is performed.

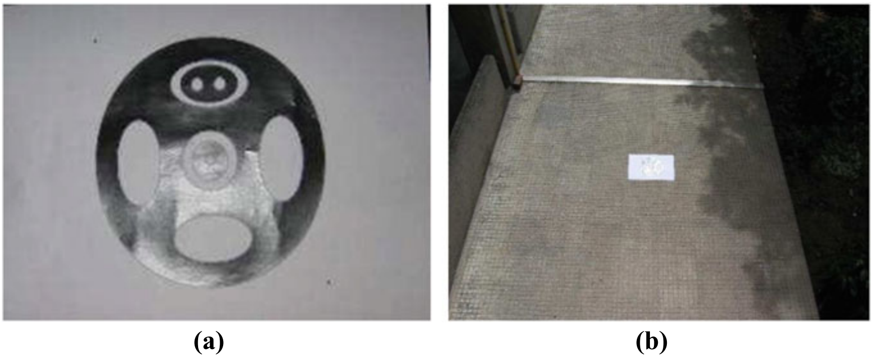


Fig. 3 Acquisition **a** acquisition problem 1, **b** acquisition problem 2

The main part of the study, Chapters "Analysis of Computer Vision-Based Techniques for the Recognition of Landing Platforms for UAVs", "Monitoring Peak Pollution Points of Water Resources with Autonomous Surface Vehicles Using a PSO-Based Informative Path Planner", and "Robot-Based Medicine. Robots in Medicine: Mobile Robots Versus Mobile Decision, Necessity Versus Possibility and Future Challenges", describe the two methods employed for the recognition of landing platforms and, in the fifth chapter, the formal theoretical aspects used in the previous methods. Finally, in Chapter "Event-Based Robust Control Techniques for Wheel-Based Robots Under Cyber-Attack and Dynamic Quantizer" an analysis and comparison of the results is given and, in Chapter "Path Optimization and Multi-level Path Planning for The Steam Field Navigation Algorithm", the main conclusions and future lines of action are established.

2 Related Works

A wide variety of active and passive methods based on computer vision strategies have been proposed for accurate landing of autonomous UAVs [11, 12]. Active methods require some additional intervention besides the vision system itself for guidance during landing, such as the use of lamps [13] or infrared (IR) emitters [14–16]. IR systems based on image capture by means of a camera installed on the ground detect an IR laser lamp located in the vicinity of the UAV nose [17] in the case of fixed-wing systems; stereo IR vision systems have also been proposed in some developments of interest [18–20], allowing the distances at which objects are located to be established with some precision. Alternatively, passive systems only require the use of the machine vision system, without the need for additional elements or actions, and therefore without intervention in the environment. Some approaches of this nature attempt to recognize the UAV in flight for subsequent guidance from a ground control station [21, 22], or otherwise through the use of optical flow sensors [23, 24] embedded in the UAV itself.

Other methods determine a safe landing area via terrain analysis [25, 26], as based on machine learning [27], by 3D terrain reconstruction [28–31], or based on 3D positioning with two cameras placed on the ground for fixed-wing UAVs [11]. Another common practice is the use of ground figures with more or less significant typologies, i.e., landing pads with specific layouts containing certain drawings. Various image processing techniques specially designed for this purpose allow for the recognition of these figures, which can be located even when in motion [32–37], and also when on ships [38, 39].

In this work, the passive approach has been used, where the final objective is to recognize a landing platform with black and white figures [2], or with color figures using an RGB monocular camera, which in normal operation mode is embedded in the UAV itself. In this sense, in the following, the state of the art is analyzed in order to compile methods based on the field of computer vision, where additional figures (black and white, or colored) are used on a given background on the platforms.

Chen et al. [40] proposed a landing platform based on a black and white figure containing a pentagon that acts as a key landing sign and which is placed in the center of a white circle. The recognition algorithm uses a technique called Faster R-CNN (Fast Region-based Convolutional Neural Network) where clusters and convolutional layers that provide specific properties of the image are included and encoded in the corresponding feature map in order to allow its recognition. The figure to be identified is easy to recognize with a relatively low computational complexity, thus providing good real-time performance. However, despite the good performance of this type of network, no additional mechanisms are provided to perform recognition in the case of missing parts of the figure due to intensity saturation as a consequence of excessive brightness caused by reflected light, which is quite common on sunny days with directly incident solar rays. Polvara et al. [41] also applied CNN to detect a black circle enclosing a black cross on a white background. This design entails the same intrinsic difficulties for recognition. Also, in both cases, this circumstance constitutes a significant limitation from the point of view of platform identification.

Wang et al. [42] proposed an H-shaped monochromatic figure [43] enclosed within a circle. Binarization techniques are applied in order to segment the image, after which the points of interest (corners) of the figure are obtained by establishing a geometric relationship between them. However, this relationship is affected by image rotation when the image is obtained from different viewpoints. In addition, the perspective projection–with which the image of the scene is projected–and the inclination angle of the camera with respect to the platform make the design of this figure highly sensitive to these types of distortions when manifest in the image. Also, the use of a single figure does not effectively mitigate problems arising in parts that are not present in the image, due, for example, to intensity saturation or partial capture of the platform, such that a full view of the platform is required to obtain all points of interest. For low altitude flights, moreover, this presents serious difficulties as, very commonly, only partial or biased views of the figure are available. Sharp et al. [44] also favored using black and white squares using point-of-interest descriptors such as the Harris operator [45] for corner detection. However, in this design the difficulties and limitations described previously regarding the absence of relevant parts of the image as a consequence of brightness and distortions still persist. Zhao and Pei [46] used a single green H-shaped figure applying the SURF (Speed Up Robust Features) descriptor [47, 48], which is invariant to small rotations and scale changes. However, the problem concerning the absence of certain parts of the figure still persists.

Saripalli et al. [49, 50] also used an H-shaped figure printed on the landing pad surface. The method is based on the following image processing techniques: filtering, thresholding, segmentation, and related component labeling. The ultimate goal is to determine the orientation angle of the figure with respect to the UAV using the aforementioned techniques. However, in this case, the same limitations as reported in [42] reappear. If a part or region is missing, either totally or partially, it is not possible to obtain a complete orientation in the range 0–360°.

Cocchioni et al. [51] designed a figure based on two concentric circles, one large and one much smaller, together with two small equilateral triangles of different sizes. Li et al. [52] also created a figure using a pattern of concentric circles, specifically using a figure that contained six circles combined together. The recognition algorithm extracts feature patterns based on the invariant affine moments [45] that are provided to a Support Vector Machines (SVMs)-type classifier [53]. In their design, the authors considered possible in-plane distortions arising from perspective projection. However, these same distortions cause the circles to be perceived as ellipses due to the changes that occur in the eccentricity; therefore, as a geometric figure it is unsuitable to handle changes in inclination angle. Furthermore, the chromatic design does not provide additional mechanisms to avoid, to the full extent possible, potential saturated areas due to brightness, which end up resulting in the total or partial absence of relevant parts of the figure in the image.

Guili et al. [54] designed a T-shaped figure that serves to determine the relative orientation angle between the platform and the UAV. This technique relies on infrared (IR) electromagnetic radiation to acquire a sufficiently high temperature to allow for a thermal contrast between the figure and the environment in which it has been placed. Two different image processing techniques are used, edge extraction and region segmentation, for recognition by extracting properties provided by Hu invariant moments [45], which allow subsequent recognition based on the similarity of these properties with characteristic patterns. However, the algorithm is not able to compensate for simplicity in pattern recognition, which causes problems in outdoor environments. Therefore, the method is susceptible to confusing the single element to be recognized with natural elements. In addition to the problems mentioned above, there is also the disadvantage that no mechanisms have been provided to handle possible missing parts of the image.

Lange et al. [55, 56] proposed a figure similar to the one used in [49], based on several concentric rings that were white in color. However, in this case, instead of using a pentagon for recognition, the use of circular crowns defined by circles whose width varied with distance from the center of the figure was employed. The key is, therefore, to achieve differentiation and identification of each ring. Although recognition of this platform is possible even when not all the circular crowns are within the field of view of the camera, which can occur when flying at low altitude, the method assumes that the UAV is always positioned parallel to the platform in the zenith position. Therefore, it ignores inclination angle and cannot be used to position the UAV until it is positioned over the platform. It is also impossible to use it with fixed-wing UAVs because their approach is not vertical. Thus, there is an important practical limitation to this approach, even when the reconnaissance is intended for rotorcraft of the type capable of vertical flight, but which are only at small separations from the platform.

AprilTag figures [57, 58] were used in [35, 59–61] by using cameras with low resolutions and robust Sobel-based edge detectors [45] to deal with occlusions, rotations, and variations in illumination. Some improvements in this regard were made in [61]. On the other hand, ArUco figures, which are similar to AprilTags, were used in [62, 63]. Chaves et al. [64] used this type of figure together with the Kalman filter [45] to guide the UAV during the final phase of landing in an approach designed to allow UAVs to perform landings on ships. However, it is well known that edges are noise-sensitive structures, and are therefore prone to typical broken edges, resulting in incorrect detections. Araar et al. [65] also used an AprilTags-based system to generate the figures that are printed on the landing platform. The objective lies in identifying the figures by developing a procedure based on computer vision techniques that includes two main components, namely the tag detector and the coding system. The detector groups the pixels according to their magnitudes and gradient directions, while the coding system reads the parts of each label field to identify the shapes. However, these figures are based on squares and are therefore very sensitive to the in-plane distortions caused by perspectives with high angles of inclination.

Nguyen et al. [12] used three concentric circles that each defined eight black and white areas. The platform can be recognized even if areas are missing. However, the figure contains a starting point for developing the recognition pattern, which requires an identification step prior to obtaining the features, which is a very serious disadvantage, particularly when the area containing such a point is intensely bright. The orientation of the UAV with respect to the platform is defined by an asymmetric black area. An adaptive template is used to establish the correspondences, which are based on image correlation, in order to find the platform. This process is guided by the previous figures obtained by applying the Kalman filter [66].

In summary, there are various approaches [40, 42, 43, 49, 54] that use a single figure that do not solve the problem of missing regions in the image due to brightness saturation from direct illumination, as caused by reflections. Other methods [1, 4, 35, 52, 55, 59–62] use several figures (interlaced or interleaved) to address the problem of missing regions or regions in the image, or complex figures based on circles [67] and ellipses. Some of the latter [1, 4] allow for a degree of distortion of the plane as a consequence of perspective projection and the UAV being positioned in an oblique manner to the platform. In others, as in [55], there is an assumption that the UAV is positioned in a zenith position with respect to the platform during the landing phase, and which thus ignore the inclination angle. Edges and points of interest have also been considered in other proposals, although one must take into consideration the high sensitivity to noise that this type of structure can entail [42, 44, 46, 54]. In summary, the methodology is highly varied as there are dissimilar approaches, e.g., some rely on size differences between figures [44, 52, 55, 56, 68], while others use single markers [32, 34, 39–43, 46, 49, 50, 54, 67, 69], not to mention methods using databases with landing sites [27, 70].

2.1 Patents

Within the scope of the proposed project for the recognition of UAV landing platforms, a number of patented platforms deserve special mention in addition to the one outlined in [2], and which are described below:

CN101109640 [71]: presented in July 2006, this is based on the use of a camera to extract features of the runway. From this information, and by performing an iterative search through the image, the landing maneuver can be carried out. The patent has two parts: (a) software that provides a computer vision analysis, the fusion of the information provided by the vision system, and an estimation of the aerodynamic parameters of the aircraft's flight state (height, orientation, speed, acceleration); and (b) a hardware arrangement comprising the runway feature, the on-board sensor subsystem, and an information fusion subsystem to collect the measurement data from the sensors. It considers two geometric shapes, a triangle and a rhombus, located longitudinally along the runway. It uses the information extracted from the figure analysis by the vision system, in conjunction with the information provided by other sensors (differential GPS and IMU), to obtain greater accuracy for the relative position of the aircraft with respect to the runway. The figures are separate and are not intended to operate with vertical take-off and landing vehicles.

US20080215204 [72]: presented in December 2007, this approach provides methods, apparatus, and systems for enhanced synthetic vision and multi-sensor data fusion, which serve in the attempt to improve the operational capability of UAVs. Basically, what is proposed is a communication system that enables the exchange of information between the UAV and a control center. This exchange is essentially based on the use of a computer vision system that is responsible for capturing and analyzing the scene of the environment over which the mission is carried out.

US20090306840 [73]: presented in April 2009, this approach proposes an automatic computer vision system. The central axis of the method is a cross-shaped figure containing a rectangle on its major axis. From the image captured by the vehicle in mid-flight, the orientation angle, or yaw, and the elevation of the UAV with respect to the central axis of the figure are determined. Therefore, the logical flow of the process is as follows:

- An image is captured in the direction of movement of the UAV.
- The image is analyzed to determine whether it is a potential target.

- The target is evaluated to determine the glide angle and elevation.
- The height, position, and orientation of the UAV are corrected accordingly.

US20070093945 [74]: as presented in October 2006, this approach provides a processing system for UAVs that combines the computer vision methodology with a motion compensation method. For this reason, the system uses two types of opposing sensors. On the one hand, it employs at least one vision sensor that serves to capture the visual information of the environment, whilst on the other, a motion sensor that is responsible for recording the UAV's main kinematic data is used. Internally, the system combines the inputs from both sensors in order to increase the estimation capabilities based on computer vision, both in the fields of navigation and stabilization. This patent provides a system that is capable of exploring previously unknown environments, giving it the ability to operate in cluttered environments or environments with a wide variety of obstacles. By way of summary, it should be noted that the patent describes an organizational structure of elements that give a UAV equipped with a vision system the ability to navigate based, at least in part, on image data corrected with information provided by other traditional sensors, making it able to adapt to dynamic environments.

KR1020070058957 [75]: this is a Korean patent whose publication date dates back to June 2008, and which serves to provide an automatic navigation method for a UAV from a given symbol located in an area containing a landing grid or net. The system recognizes the image, measures the distance using a laser pointer, and then transmits the information to an external control station. The system proceeds through the following stages, which are schematically described below:

- Real-time photography through the use of a front camera that is built into the UAV; the symbol to be recognized is located on a landing grid.
- After image acquisition, an image analysis process is carried out via a dedicated graphics processor.
- Using a laser pointer, the distance between the landing symbol and the UAV is measured in real time.
- All information regarding the symbol and the distance between the symbol and the UAV is transmitted in real time, as well as the control commands necessary to carry out the landing.

US5525883A [76]: this patent, presented in July 1994, describes a method for determining the orientation and position of an autonomous vehicle through the use of reference points. These reference points are similar to the cells of a chessboard, containing rectangular figures of contrasting colors (black and white) which are arranged in such a way as to create a pseudo-random pattern. The UAV incorporates a recognition system based on a camera and an onboard processor, which, through the use of computer vision techniques such as binarization in conjunction with DTL (Direct Linear Transformation) and a symbolic map stored in the recognition and measurement unit, serve to describe the position, orientation, and dimensions of each pseudo-random pattern on the chessboard. The recognition and measurement unit processes the image of the dashboard on the ground to identify the reference point where the pattern is displayed, and accordingly attempts to calculate the position and orientation of the vehicle.

3 Method 1: Expert System-Based Approach

The aim of this method 1, based on an Expert System [1], is to identify the platform shown in Fig. 4, specifically the ellipses and circles that serve as markers. To this end, once the image is obtained through the camera incorporated in the UAV, the five main processes shown in the header of the blocks shown below in Fig. 4 will be performed:

The process can be briefly summarized as follows:

1. Conversion: The image is acquired from the smartphone onboard the UAV in RGB format. Subsequently, a transformation from the RGB to HSV color spaces is applied to binarize a grayscale intensity image.
2. Binarization [77, 78]: First, a threshold is obtained, which can be established by a direct approach (and is global) or by Otsu's method [79]. The objective is to obtain a black and white image from the threshold that serves to establish the limit when deciding whether a pixel is represented as black or white.
3. Segmentation: From the binary image, a morphological closing process [80, 81] is performed to remove small (spurious) regions. Then, the image is segmented

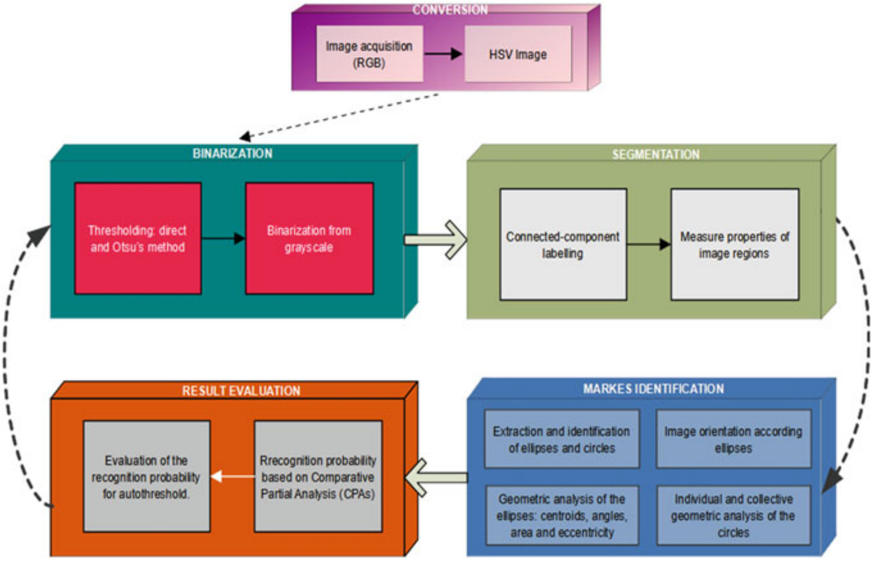

CONVERSION
Image acquisition (RGB)
HSV Image
BINARIZATION
Thresholding: direct and Otsu's method
Binarization from grayscale
SEGMENTATION
Connected-component labelling
Measure properties of image regions
RESULT EVALUATION
Evaluation of the recognition probability for autothreshold.
Rrecognition probability based on Comparative Partial Analysis (CPAs)
MARKES IDENTIFICATION
Extraction and identification of ellipses and circles
Image orientation according ellipses
Geometric analysis of the ellipses: centroids, angles, area and eccentricity
Individual and collective geometric analysis of the circles

Fig. 4 The coarse-grained block diagram of method 1, based on an expert system

using a connected-component labeling process [82], as based on [83], which allows the metric and topological properties of each region [84] to be obtained.

4. Marker Identification: From the metric and topological properties of each image region, relationships are established based on the known patterns (ellipses and circles) that form part of the platform.
5. Results Evaluation: Finally, the degree of probability of recognition is obtained from the Partial Comparative Analyses and their associated thresholds that serve to give flexibility to the distortions in the plane that can result from different possible perspectives.

Finally, to allow for a better understanding of the above, Fig. 5 shows a fine granularity block diagram that gives a higher level of detail to the implemented algorithm:

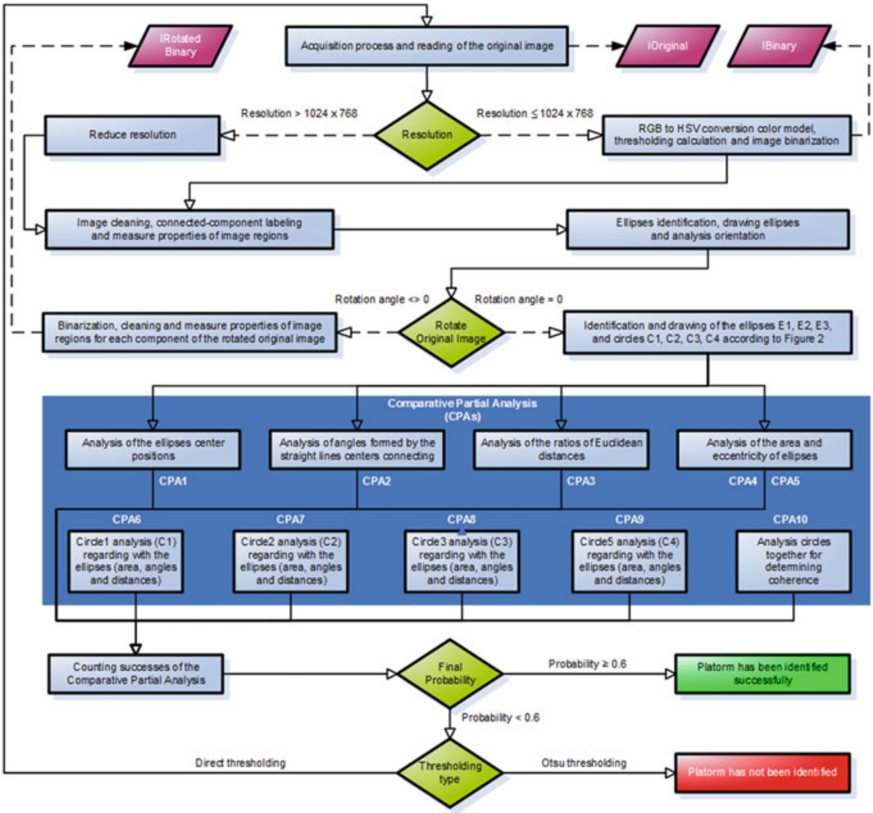

IRotated Binary
Acquisition process and reading of the original image
IOriginal
IBinary
Resolution > 1024 x 768
Resolution
Resolution ≤ 1024 x 768
Reduce resolution
RGB to HSV conversion color model, thresholding calculation and image binarization
Image cleaning, connected-component labeling and measure properties of image regions
Ellipses identification, drawing ellipses and analysis orientation
Rotation angle <> 0
Rotation angle = 0
Binarization, cleaning and measure properties of image regions for each component of the rotated original image
Rotate Original Image
Identification and drawing of the ellipses E1, E2, E3, and circles C1, C2, C3, C4 according to Figure 2
Comparative Partial Analysis (CPAs)
Analysis of the ellipses center positions
Analysis of angles formed by the straight lines centers connecting
Analysis of the ratios of Euclidean distances
Analysis of the area and eccentricity of ellipses
CPA1
CPA2
CPA3
CPA4
CPA5
CPA6
CPA7
CPA8
CPA9
CPA10
Circle1 analysis (C1) regarding with the ellipses (area, angles and distances)
Circle2 analysis (C2) regarding with the ellipses (area, angles and distances)
Circle3 analysis (C3) regarding with the ellipses (area, angles and distances)
Circle5 analysis (C4) regarding with the ellipses (area, angles and distances)
Analysis circles together for determining coherence
Counting successes of the Comparative Partial Analysis
Final Probability
Probability ≥ 0.6
Platorm has been identified successfully
Probability < 0.6
Direct thresholding
Thresholding type
Otsu thresholding
Platorm has not been identified

Fig. 5 The fine-grained block diagram of method 1, based on an expert system

4 Method 2: Cognitive Computation-Based Approach

Method 2, which is based on a Cognitive Computation approach [4], aims to allow for the recognition of the platform in Fig. 2. As mentioned previously, it can be adopted to overcome the limitations to the previous method when acquisition conditions are highly adverse, i.e., there is considerable distortion in the plane due to the inclination angle at which the image is taken, or where brightness or reflections are produced by overexposure of light, among others, resulting in blind markers, as shown in Fig. 3b.

As in the previous method, a block diagram has been developed to illustrate the main processes involved in the cognitive computation approach (Fig. 6).

Finally, a brief summary has been provided that highlights the main differences between this and the previous method:

1. Conversion: The image is again captured with a smartphone onboard the UAV in RGB format. However, since the binarization will be carried out from the colors, the image is transformed to the L*a*b* color space because this contains luminance as a separate, distinct component, which minimizes the impact of reflection caused by intense light sources.
2. Binarization: Unlike method 1, binarization is performed on the basis of the image colors instead of using a grayscale. Color is an excellent attribute in image processing [85]. Thus, this process is performed according to six different colors (one per marker) and the associated thresholds, as reported in Table 1. The threshold is direct and global, so the Otsu method is not used.

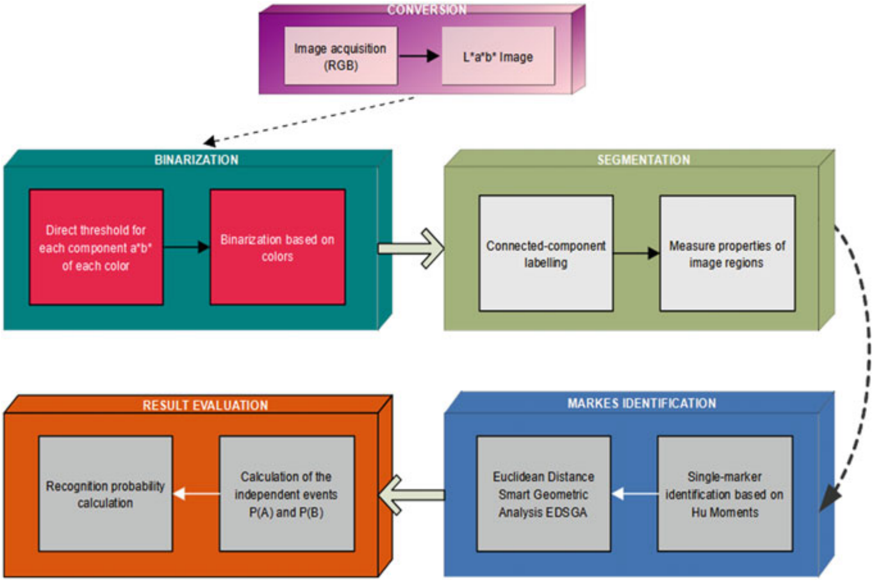

CONVERSION
Image acquisition (RGB)
L*a*b* Image
BINARIZATION
Direct threshold for each component a*b* of each color
Binarization based on colors
SEGMENTATION
Connected-component labelling
Measure properties of image regions
RESULT EVALUATION
Recognition probability calculation
Calculation of the independent events P(A) and P(B)
MARKES IDENTIFICATION
Euclidean Distance Smart Geometric Analysis EDSGA
Single-marker identification based on Hu Moments

Fig. 6 The coarse-grained block diagram of method 2, based on cognitive computation

Table 1 Color component values and thresholds for the markers used in method 2, as based on cognitive computing

Marker (r)	a* (TI_a)	b* (TI_b)	Threshold (Tl)
M1	44.2	−34.9	22.8
M2	−2.91	29.88	23.1
M3	−15.7	33.9	24.0
M4	15.21	−41.2	26.7
M5	48.23	6.2	27.8
M6	21.9	−6.2	20.2

3. Segmentation: In this phase, there is no difference between this and the previous method, since from the connectivity of the pixels of the binary image a process is established to obtain all the binary regions, along with their metric and topological properties.
4. Markers Identification: To make the system more robust, instead of relying on metric properties, Hu's seven invariant moments [86] are used together with other color-based descriptors through a Decision-Making Strategy [87]. The resulting descriptor is invariant to in-plane deformations, such as scale changes, rotations, etc.
5. Results Evaluation: The probability function is improved by taking into account two independent events, P(A) and P(B), based on the total number of markers found and the geometric relationships between them. The final calculation is modeled as the intersection under the assumption that they are two independent events, i.e., based on probability theory [88].

5 Theoretical Aspects

As we can see, both methods use very well-known and well-used techniques in computer vision. The following gives a brief description of the most relevant.

5.1 Color Models

Color models give digital representations of the spectral components reflected by objects in the scene. There are methods of transformation between them. The three models, which are used in each of the methods above, are described below.

5.1.1 RGB

In the RGB model, each color appears in its primary spectral components: red, green, and blue. Each pixel in the image contains a combination of such components. This is the original model provided by the camera of the smartphone onboard the UAV since it is the most widely used color space in consumer microelectronics such as monitors, camcorders, etc.

5.1.2 HSV

HSV (Hue, Saturation, Value) is a color model designed to be intuitive, as it comes closest to mimicking the human perception of color. H, expressed in sexagesimal degrees, measures the color class involved. At 0° (or 360°) we find red, and by varying H from 0 to 360°, we go through yellow, green, cyan, blue, magenta, and back to red. S ranges from 0 to 100%, and is sometimes referred to as the "purity". The lower the saturation of a color, the higher the level of gray it will have, and thus the more attenuated it will appear. V, also called brightness (B), represents the color gamut value, and ranges from 0 to 100%. This model is used in method 1 because it facilitates the conversion to grayscale, and the binarization will be performed from the pixel densities in such a grayscale.

5.1.3 CIE 1976 L*a*b*

The L*a*b* color space models color as three separate components: L* for perceptual lightness, and a* and b* for the four unique colors of human vision: red, green, blue, and yellow. It is the model used in method 2 to perform the segmentation [89] because it contains luminosity as a separate component, which minimizes the impact of light reflections because this component is not used during recognition. Its advantages in providing robustness under adverse acquisition conditions are as follows:

1. It belongs to the category of uniform color spaces, where changes in color coordinates correspond to similar or identical changes in terms of visible color shades or saturation [85].
2. It is device independent, and has been sufficiently tested to deal with similar spectral component values. This occurs under adverse lighting conditions, such as those found in outdoor environments where high illumination levels lead to saturation with low color contrast.

5.2 Binarization

Binarization is a basic process used to represent the image using only two color tones, black and white. Each intensity pixel is compared with a previously computed threshold (T) value. Intensities lower than T are transformed to 0 (black), and to 1 (white) otherwise. This results in a black and white image represented by a color depth of 2^1 bits, as shown in Fig. 7. Threshold determination is a complex task in outdoor scenarios, where lighting conditions are highly variable.

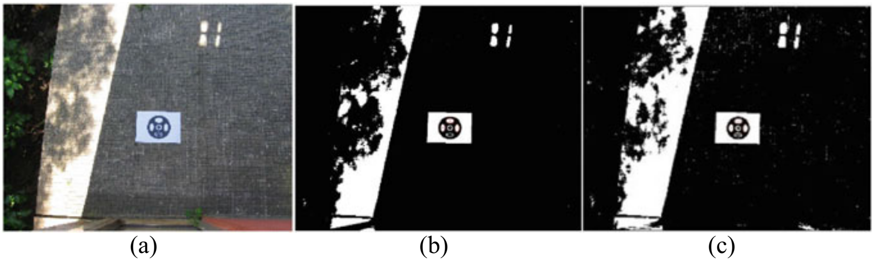


Fig. 7 Binarization and ellipse detection of a landing platform for method 1 based on an expert system: **a** original color image in RGB format, **b** binary image using a direct threshold, **c** binary image using a threshold obtained via Otsu's method

5.3 Thresholding

As we have just seen, thresholding is the most important and complex process when binarizing an image. From a formal point of view, we can consider thresholding as an operation that involves checking with respect to a function T of the form,

$$T = T[x, y, p(x, y), f(x, y)] \tag{1}$$

where $f(x,y)$ is the intensity at the point (x,y) and $p(x,y)$ is some local property of that same point; for example, the average intensity of a neighborhood environment centered at (x,y). A binary image $g(x,y)$ can be created by defining:

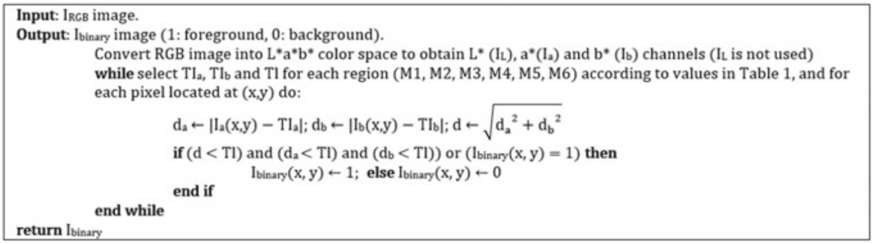
$$g(x, y) = \begin{cases} 0\, si\ f(x, y) > T \\ 1\, si\ f(x, y) \le T \end{cases} \tag{2}$$

Examination of $g(x,y)$ shows that pixels assigned a value of 0 correspond to objects, while those corresponding to the environment have a value of 1.

When T depends only on $f(x,y)$, the threshold is considered to be global. If T depends on both $f(x,y)$ and $p(x,y)$, then the threshold is considered local. If T depends on the spatial coordinates x and y, it is referred to as a dynamic threshold.

5.3.1 Direct Thresholding

Dynamic thresholding consists of establishing a constant global threshold for the entire image. The threshold value is established through a fine-tuning process based on a heuristic method of trial and error, as is carried out in the two methods proposed in this study. Figure 7c shows the result of applying this process to Fig. 7a to obtain a binary image of the platform used in method 1.



```
Input: $I_{RGB}$ image.
Output: $I_{binary}$ image (1: foreground, 0: background).
        Convert RGB image into L*a*b* color space to obtain L* ($I_L$), a*($I_a$) and b* ($I_b$) channels ($I_L$ is not used)
        while select $TI_a$, $TI_b$ and TI for each region (M1, M2, M3, M4, M5, M6) according to values in Table 1, and for
        each pixel located at (x,y) do:
                $d_a \leftarrow |I_a(x,y) - TI_a|$; $d_b \leftarrow |I_b(x,y) - TI_b|$; $d \leftarrow \sqrt{d_a^2 + d_b^2}$
                if $(d < TI)$ and $(d_a < TI)$ and $(d_b < TI))$ or $(I_{binary}(x, y) = 1)$ then
                        $I_{binary}(x, y) \leftarrow 1$;  else $I_{binary}(x, y) \leftarrow 0$
                end if
        end while
return $I_{binary}$
```

Fig. 8 An algorithm for obtaining a binary image with foreground extraction in method 2, as based on cognitive computation

For this process, a set of training images is used, which have to be as heterogeneous as possible (different perspectives, scales, etc.) and recorded under different environmental lighting conditions, namely cloudy, sunny, sun and shadow, and artificial light. As the threshold will be constant for all the images processed, it is therefore essential that the training set contains as many and diverse images as possible.

In both methods, a supervised training process, based on a Naïve Bayes classifier, was performed using 100 images obtained at different distances and camera inclination angles with respect to the platform, and under different lighting conditions. From this process, we obtain an averaged value that ensures adequate performance for any type of image. Table 1 reports the values of each marker (described in Fig. 2) used in method 2, as based on cognitive computation.

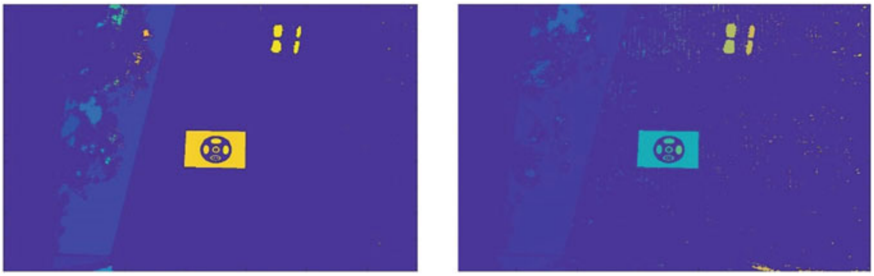
From the values in Table 1, if we apply the algorithm shown in Fig. 8 based on [89], a binary image is obtained with foreground segmentation [90].

5.3.2 Otsu's Method

Otsu's method was used in method 1 as an alternative to direct thresholding, as per Fig. 7c. It is a well-known approach where the threshold T is obtained for the intensity that achieves the minimum weighted variance among all intensity levels for each image.

5.4 Connected Components Labeling

It is assumed that the image is binarized (composed of zeros and ones). The objective is to label the connected components so that the final result defines a region. The concept of connected components is as follows: all pixels that have a binary value of "1" and are connected to each other by a path or set of pixels, all with a binary value of "1", are assigned the same identifying label, which must be unique to the region to which the pixels belong, and which constitutes their identifier. The component-labeling algorithm is based on connectivity-8. Two adjoining pixels are part of the



(a) (b)

Fig. 9 Result of segmentation of Fig. 7a, where each color represents a different region: **a** segmented image obtained from Fig. 7b, **b** segmented image obtained from Fig. 7c

same object if they are both on and are connected along the horizontal, vertical, or diagonal directions. Figure 9 shows the result of applying the connected component algorithm to the binary images shown in Fig. 7b and c.

5.5 Feature Extraction: Hu Moments

Once the regions of the image have been segmented, they must be described with robust features [79]. The seven Hu's moments [86] are excellent candidates for discrimination between regions as they are invariant (i.e., similar values for objects of the same type) to the geometric transformations (translation, rotation, and scaling) that objects can suffer.

Table 2 shows the calculation of the seven Hu invariant moments [91] for the six markers (displayed in Fig. 2) used in the method 2 platform, as based on cognitive computation, which is shown in Fig. 2. The values have been obtained from a Naïve Bayes classifier and a training set of 100 images used specifically for this work.

Table 2 Average value per marker of the seven invariant Hu moments for the landing platform shown in Fig. 2

Marker (r)	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7
M1	−1.2792	−3.7469	3.996	−6.7799	−12.7987	−8.8983	−13.5732
M2	−1.2497	−2.9803	−12.3341	−14.6017	−28.6875	−16.9007	−28.2904
M3	−1.8236	−7.0982	−14.7245	−18.2122	−35.1413	−22.4637	−35.7149
M4	−1.3987	−3.2981	−15.0776	−16.3802	−32.0565	−18.1099	−33.4995
M5	−1.5137	−4.6541	−6.9887	−10.0093	−18.4399	−12.0298	−20.6809
M6	−0.8699	−2.1896	−5.2098	−7.5789	−14.7979	−8.6935	−15.2345

6 Results

For the testing phase, 1000 images were processed and acquired following the same design scheme as the experiments performed in [1, 4]. Therefore, all tests were performed under similar acquisition conditions, such as distances, angles, etc., and with the same UAV flying in the same environment.

Specifically, then, this means that the images were acquired from a Samsung Galaxy S6 mobile phone camera onboard an md-400 quadrotor UAV with four brushless engines operating at 1000 rpm/V each, driving a 10-inch propeller and powered by its own battery, as shown in Fig. 10.

The tests were carried out under the following conditions:

1. The landing platforms shown in Figs. 1 and 2 were each printed on a blank piece of A3 paper (297 × 420 mm).
2. 1000 images were captured with the Samsung Galaxy S6 smartphone placed onboard the UAV at a resolution of 4 Megapixels with dimensions of 2560 × 1944 pixels without the use of zoom or flash.
3. The color depth used is 3 bytes (24 bits), which means 8 bits per RGB channel, and where the compression format for all images is based on the JPEG standard.
4. The test scenario was carried out using the UAV flying in the courtyard of a building within a residential area.

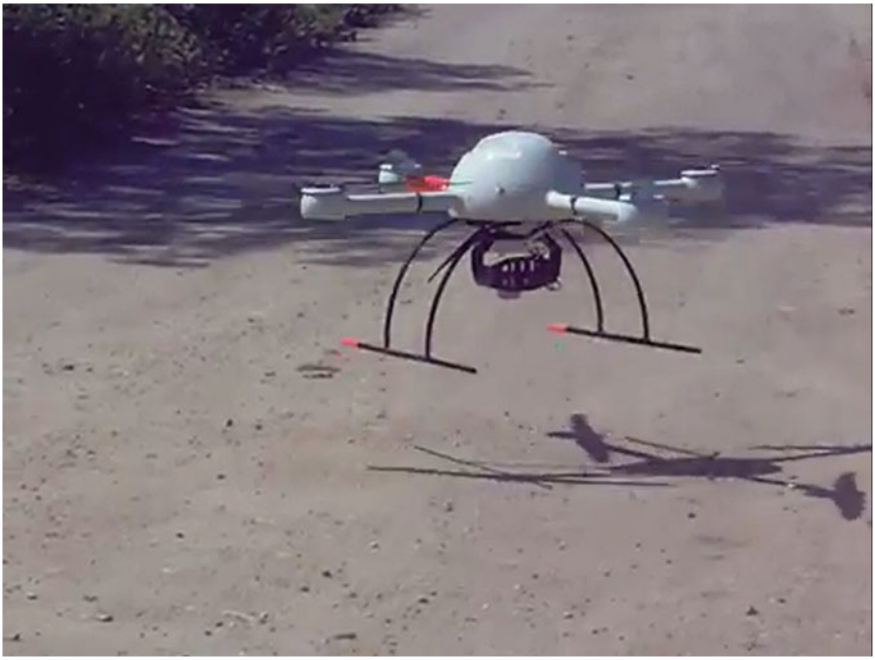
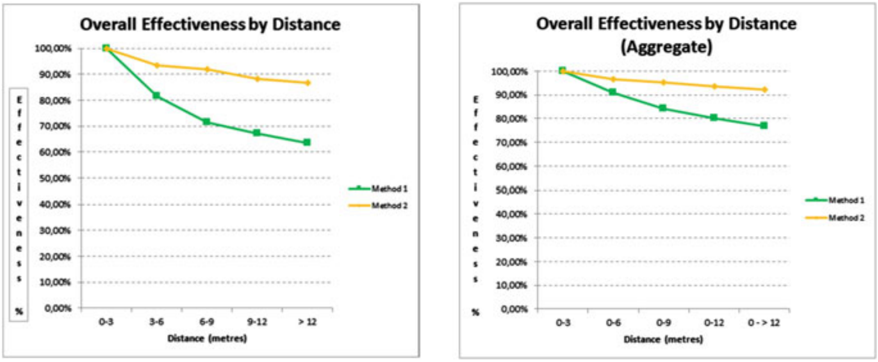


Fig. 10 The UAV used during the testing, which in this image has just taken off



(a)

(b)

Fig. 11 Comparison of effectiveness in terms of distance (meters) between method 1 and method 2: **a** exclusive intervals, **b** aggregate (cumulative) intervals

As far as the interpretation of the results is concerned, the same supervision process was applied as in [1, 4]. An outcome is considered successful when the operator, by visual inspection, determines that there are at least three red regions identified in the original image, and that they match the appropriate markers on the platform.

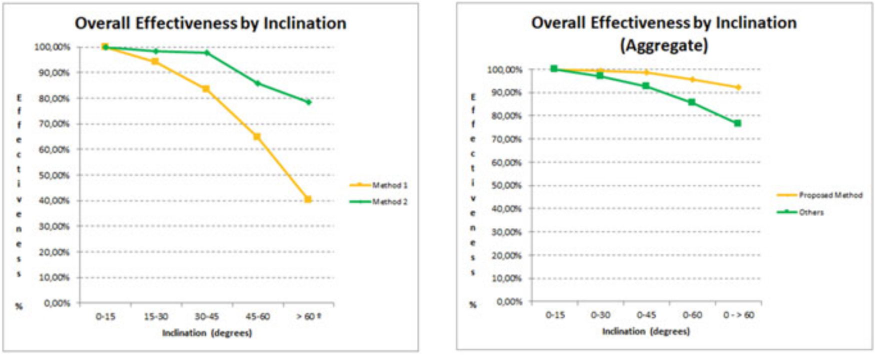
The purpose of the testing phase is to assess the robustness and accuracy of the proposed methods under the following three variable conditions: distance, inclination angle (under different perspectives), and lighting (outdoor: cloudy, sun and shadow, sunny; indoor: artificial light).

6.1 Distance Test

1000 images were used with distance (expressed in meters) defined by the range [0.4, 15] between the UAV and the platform. The results obtained are summarized in Fig. 11, where percentage effectiveness versus distance is plotted. The green lines represent the results of method 1, while the orange lines represent those of method 2.

6.2 Inclination Angle Test

1000 images were used that used a range of inclination angles from 1 to 72°, defined as the angle in degrees formed by the vertical axis in the image and the imaginary straight line connecting the center of the camera lens with the center of the landing platform. The results obtained are shown in Fig. 12, where percentage effectiveness versus inclination angle in degrees is graphically represented. As previously, the



(a)

(b)

Fig. 12 Comparison of percentage effectiveness in terms of inclination angle (degrees) between method 1 and method 2: **a** exclusive intervals, **b** aggregate (cumulative) intervals

green lines represent the results of method 1, while the orange lines represent those of method 2.

6.3 Lighting Condition Test

1000 images were acquired on different days and under different daytime illumination conditions, both in outdoor (cloudy, sun and shadow, sunny) and indoor (artificial illumination) environments. The assignment of the illumination condition was made at the discretion of the quadrotor operator during different flight sessions. Figure 13 shows a graphical representation of the averaged percentage effectiveness for each category over all the images used for each type of lighting condition (category). As in the previous graphs, the green lines represent the results of method 1, while the orange lines represent those of method 2.

7 Conclusion and Future Work

7.1 Conclusion

In the research developed and reported in this chapter, a general review of the state of the art regarding the problems inherent to the recognition of a UAV landing platform in images captured by an onboard vision system has been carried out. The existing problem has been analyzed, and a solution has been found by means of the analysis of two methods that deal with very different computer vision techniques. The first

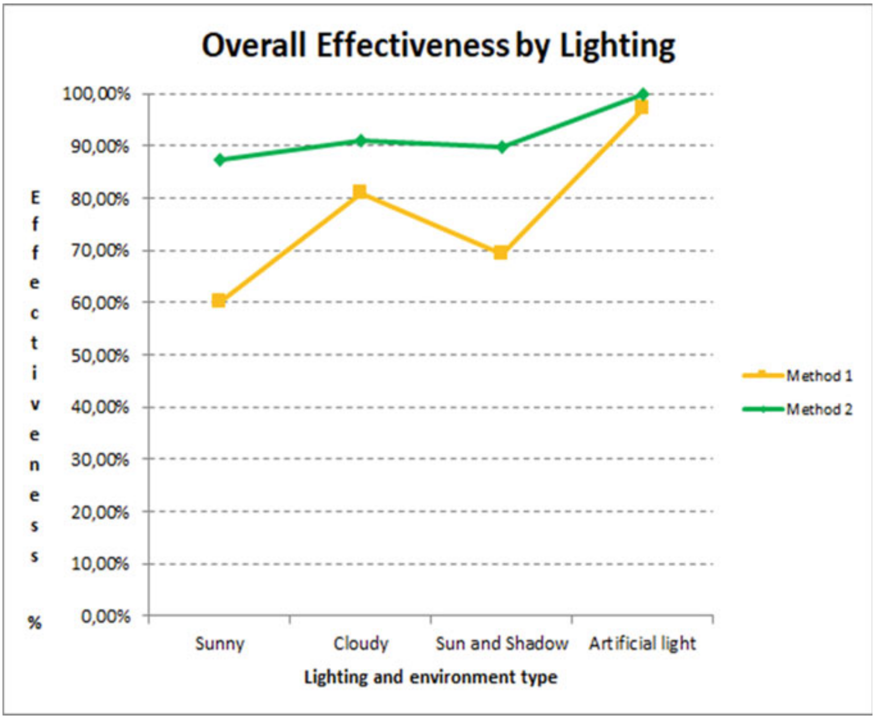


Fig. 13 Comparison of percentage effectiveness in terms of different lighting and environmental conditions between method 1 and method 2

approach is based on identifying a patented black and white platform [2] using an approach based exclusively on the similarity between regions and the study of the geometric relationships that can be established between them.

Method 1, based on an Expert System [1], behaves appropriately when images are obtained at different heights, or at a given height with different levels of image resolution. In a real, operational environment, the most important issue is that the inclination angle and the proportion that the area of the figure occupies with respect to the total size of the figure is constant. This facilitates the assignment of thresholds for a given resolution to keep the acquisition process within a controlled environment.

Due to the design of the patented platform [2], the environment where it is located can be determinant from the point of view of recognition, mainly due to factors such as leaves that can give rise to ellipses, trees that generate shaded areas, or natural fruits on the ground that can have elliptical or circular shapes. It is therefore considered vital that the platform is printed on a surface that is as uniform as possible in terms of colors and shapes, such as asphalt, for example.

In order to overcome the limitations detected in the first method, as well as those observed in the methods described in the literature, a new platform was developed based on simple figures with unique colors and shapes. The technique that allows their identification is different since it is based on the use of descriptors and techniques invariant to the deformations produced in the plane as a consequence of the different perspectives from which images can be obtained.

In this way, the recognition is performed in a more robust way that represents the various different scenarios that can occur in the real world, therefore allowing for operation in a wide variety of situations: multiple distances, inclination angles, different lighting conditions (sunny, cloudy, sun and shadow, etc.), scenarios (indoor, outdoor), and with non-existent markers in the scene. Robustness and accuracy are, therefore, the main characteristics that define method 2, which outperforms the previous method in these areas.

Method 2, based on Cognitive Computation [4], has proven to be significantly more effective than method 1 in terms of overall efficiency, achieving a 92% success rate in terms of recognition, while method 1 achieved 76.5% efficiency using the same parameters and similar conditions for image acquisition: identical platform size (A3 paper), identical test environment, distances, angles, and lighting conditions. All this was done to allow for test conditions that were essentially identical (insofar as was possible).

Although the relative sensitivity to the inclination angle that arises as a consequence of perspective could represent a limiting factor in the performance of method 2, since at angles greater than 45° the green line in Fig. 12a clearly starts to curve downwards, the improvements with respect to method 1 are evident. Indeed, from all the graphs shown in the previous section, it can be clearly observed that the inclination angle is the main limiting factor inherent to both methods. However, the loss of effectiveness in this respect is much less pronounced in method 2 than in method 1, as can be seen by comparing the curves in Fig. 12.

Finally, if we analyze the graph in Fig. 13, we can also observe how method 2 is more stable with respect to lighting conditions, since the green line is less curved than the yellow, the latter representing method 1 in this graph. This is mainly due to the color-based design of the platform used in method 2. This platform is based on highly spectrally differentiated colors, so that if there is light overexposure in the image, as in Fig. 3 (as a consequence of the sun's rays, for example), the probability that not all the markers will affected and that some will remain visible, even if only partially, is greater than if a single color is used for all markers.

7.2 Future Work

Future lines of action should improve the performance of method 2 [4] further. One option might be to provide further support by using additional recognition approaches, perhaps similar to those used in face recognition [92] but applied to the recognition of Fig. 2. In this sense, it is worth noting that the face detection function that is incorporated into the cell phone connected to the UAV framed the platform within the box used in human face detection. This means that the color platform used in the second method, whose design follows that of a face, can already be detected by the cell phone's current face detection system. However, in order to improve effectiveness in this regard, it may be necessary to carry out additional modifications to the design of the platform drawing.

On the other hand, since the objective of this work is to recognize a specific object (platform) in an image, it may be feasible to delve into the field of Convolutional Neural Networks (CNN) of the R-CNN type, taking into account the improvements already proposed in [40, 41]. Another possible line of action may consist of using Deep Learning through the application of deep neural networks [93]. In this regard, the approach adopted for the autonomous landing of a UAV proposed in [94] should be appropriately considered. However, for both detector-based and R-CNN and Deep Learning network-based enhancement, deep learning processes with huge numbers of images would be required.

Regarding the lines of improvement related to method 1, the following lines of action are proposed. On the one hand, to implement an algorithm that establishes a correlation between the thresholds and the proportion that the area of the figure represents in relation to the total number of pixels in the image. When the image provides an indication of closeness, i.e., when the ratio between the area of the figure and the total area of the image tends to 1, better results are obtained at lower resolutions, i.e., with a value close to the megapixel level (e.g., 1024 × 768). However, when a sense of remoteness is perceived in the image, i.e., when the ratio between the area of the figure and the total area of the image tends to 0, good results can be obtained when working at higher resolutions (greater than 3 megapixels).

On the other hand, the algorithm developed for the identification and recognition of the patented platform [2] yields good results when the inclination angle above the horizontal does not exceed 45°. When this angle is exceeded, the deformations in the plane become too pronounced, causing large changes in the eccentricities of circles and ellipses, also affecting their areas such that they can no longer be managed by the thresholds. Therefore, for inclination angles greater than 50°, the system does not behave properly since the distortions in the plane with respect to the zenith perspective cannot be properly managed. Thus, a future course of action to improve recognition at inclination angles greater than 50° would support one of the major limitations inherent to platform detection.

Acknowledgements The authors would especially thank Dr. Mark Watkins for his proofreading, proof-editing services and grammatical improvement. Without his selfless assistance and dedication this work would undoubtedly not have been possible.

Finally, we would also like to thank Mestrelab, especially Santi Dominguez and Carlos Cobas, for having created a company like this that provides the flexibility and conditions necessary to make research like this possible. Special mention to Agustín Barba from Mestrelab also and Rebeca Cuiñas, for their unconditional support in good and especially in bad moments.

References

1. García-Pulido JA, Pajares G, Dormido S, de la Cruz JM (2017) Recognition of a landing platform for unmanned aerial vehicles by using computer vision-based techniques. Expert Syst Appl 76:152–165
2. Cruz JM, Sánchez B, Pajares G (2012) System for guiding an unmanned vehicle towards a platform using visual analysis. Patent number 201001592, 2013
3. SALACOM (2013) Sistema Autónomo para la Localización y Actuación ante Contaminantes en el Mar, DPI2013–46665-C1
4. García-Pulido JA, Pajares G, Dormido S (2022) UAV landing platform recognition using cognitive computation combining geometric analysis and computer vision techniques. Cogn Comp
5. Peña JM, Torres-Sanchez J, de Castro AI, Kelly M, López-Granados F (2013) Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle (UAV) images. PLoS One 8(10):e77151
6. Jiang T, Geller J, Ni D, Collura J (2016) Unmanned aircraft system traffic management: concept of operation and system architecture. Int J Transp Sci Technol 5(3):123–135
7. Popescu D, Ichim L, Stoican F (2017) Unmanned aerial vehicle systems for remote estimation of flooded areas based on complex image processing. Sensors 17(3):446
8. Pajares G (2015) Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). Photogramm Eng Remote Sens 81(4):281–329
9. Menéndez O, Pérez M, Auat Cheein F (2019) Visual-based positioning of aerial maintenance platforms on overhead transmission lines. Appl Sci 9(1)
10. Amazon Prime Air Available online: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011 Accessed June 2022
11. Kong W, Hu T, Zhang D, Shen L, Zhang J (2017) Localization framework for real-time UAV autonomous landing: an on-ground deployed visual approach. Sensors 2017(17):1437
12. Nguyen PH, Kim KW, Lee YW, Park KR (2017) Remote marker-based tracking for UAV landing using visible-light camera sensor. Sensors 2017(17):1987
13. Anitha G, Kumar RNG (2012) Vision based autonomous landing of an unmanned aerial vehicle. Procedia Eng 2012(38):2250–2256
14. Xu G, Zhang Y, Ji S, Cheng Y, Tian Y (2009) Research on computer vision–based for UAV autonomous landing on a ship. Pattern Recognit Lett 2009(30):600–605
15. Xu G, Qi X, Zeng Q, Tian Y, Guo R, Wang B (2013) Use of land's cooperative object to estimate UAV's pose for autonomous landing. Chin J Aeronaut 2013(26):1498–1505
16. Gui Y, Guo P, Zhang H, Lei Z, Zhou X, Du J, Yu Q (2013) Airborne vision-based navigation method for UAV accuracy landing using infrared lamps. J Intell Robot Syst 72(2): 197
17. Yang T, Li G, Li J, Zhang Y, Zhang X, Zhang Z, Li Z (2016) A ground–based near infrared camera array system for UAV auto–landing in GPS–denied environment. Sensors 2016(16):1–20
18. Kong W, Zhang D, Wang X, Xian Z, Zhang J (2013) Autonomous landing of an UAV with a ground–based actuated infrared stereo vision system. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. Tokyo, Japan, pp 2963–2970
19. Zhou D, Zhong Z, Zhang D, Shen L, Yan C (2015) Autonomous landing of a helicopter UAV with a ground-based multisensory fusion system. In: Seventh international conference on machine vision (ICMV 2014). International Society for Optics and Photonics, 2015, 94451R
20. Tang D, Hu T, Shen L, Zhang D, Kong W, Low KH (2016) Ground stereo vision-based navigation for autonomous take-off and landing of UAVs: a Chan-Vese model approach. Int J Adv Robot Syst 13(2):67
21. Martínez C, Campoy P, Mondragón I, Olivares–Méndez MA (2009) Trinocular ground system to control UAVs. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. St. Louis, MO, USA, pp 3361–3367

22. Kong W, Zhou D, Zhang Y, Zhang D, Wang X, Zhao B, Yan C, Shen L, Zhang J (2014) A ground-based optical system for autonomous landing of a fixed wing UAV. In: Proceedingsof the IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2014) September 14–18. Chicago, IL, USA, pp 1–8
23. Al-Sharman MK, Emran BJM, Jaradat A, Najjaran H, Al-Husari R, Zweiri Y (2018) Precision landing using an adaptive fuzzy multi-sensor data fusion architecture. Appl Soft Comput 69:149–164
24. Al-Sharman M, Al-Jarrah MA, Abdel-Hafez M (2018) Auto takeoff and precision terminal-phase landing using an experimental optical flow model for GPS/INS enhancement. ASCE-ASME J Risk Uncertainty Eng Syst Part B Mech Eng
25. Asadzadeh M, Palaiahnakote S, Idris MYI, Anisi MH, Lu T, Blumenstein M, Noor NM (2019) An automatic zone detection system for safe landing of UAVs. Expert Syst Appl 122:319–333
26. Patterson T, McClean S, Morrow P, Parr G, Luo C (2014) Timely autonomous identification of UAV safe landing zones. Image Vis Comput 32(9):568–578
27. Li X (2013) A software scheme for UAV's safe landing area discovery. AASRI Procedia 2013(4):230–235
28. Forster C, Faessler M, Fontana F, Werlberger M, Scaramuzza D (2015) Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles. In: Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 111–118
29. Johnson A, Montgomery J, Matthies L (2005) Vision guided landing of an autonomous helicopter in hazardous terrain. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)
30. Bosch S, Lacroix S, Caballero F (2006) Autonomous detection of safe landing areas for an UAV from monocular images. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
31. Desaraju V, Michael N, Humenberger M, Brockers R, Weiss S, Matthies L (2015) Vision-based landing site evaluation and trajectory generation toward rooftop landing. Auton Robots 39(3):445–463
32. Davide F, Alessio Z, Alessandro S, Jeffrey D, Scaramuzza D (2017) Vision-based autonomous quadrotor landing on a moving platform. J Intell Robot Syst 85(2):369–384
33. Lee D, Ryan T, Kim HJ (2012) Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In: 2012 IEEE international conference on robotics and automation. pp 971–976
34. Lee H, Jung S, Shim DH (2016) Vision–based UAV landing on the moving vehicle. In: Proceedings of the International Conference on Unmanned Aircraft System. Arlington, MA, USA, pp 1–7
35. Feng Y, Zhang C, Baek S, Rawashdeh S, Mohammadi A (2018) Autonomous landing of a UAV on a moving platform using model predictive control. Drones 2(34)
36. Line V (2018) Autonomous landing of a multirotor UAVon a platform in motion. Master Thesis, Norwegian University of Science and Technology, Noruega. https://brage.bibsys.no/xmlui/handle/11250/2558185 Accessed June 2022
37. Rodríguez-Ramos A, Sampedro C, Bavle H, Milosevic Z, García-Vaquero A, Campoy P (2017) Towards fully autonomous landing on moving platforms for rotary unmanned aerial vehicle. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS). pp 170–178
38. Polvara R, Sharma S, Wan J, Manning A, Sutton R (2017) Towards autonomous landing on a moving vessel through fiducial markers. In: IEEE European conference on mobile robotics (ECMR). IEEE
39. Lin S, Garratt MA, Lambert AJ (2017) Monocular vision-based real-time target recognition and tracking for autonomously landing an uav in a cluttered shipboard environment. Auton Robots 41(4):881–901
40. Chen J, Miao X, Jiang H, Chen J, Liu X (2017) Identification of autonomous landing sign for unmanned aerial vehicle based on faster regions with convolutional neural network. In: IEEE international conference on chinese automation congress (CAC). pp 2019–2114

41. Polvara R, Patacchiola M, Sharma S, Wan J, Manning A, Sutton R, Cangelosi A (2017) Autonomous quadrotor landing using deep reinforcement learning. arXiv preprint arXiv:1709.03339
42. Wang L, Yang Q, Guo X (2016) Recognition algorithm of the apron for unmanned aerial vehicle based on image corner points. Laser J 08:71–74
43. Nyein EE, Tun HM, Naing ZM, Moe WK (2015) Implementation of vision-based landing target detection for VTOL UAV using raspberry Pi. Int J Sci Technol Res 4(8):184–188
44. Sharp CS, Shakernia O, Sastry SS (2001) A vision system for landing an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation. pp 1720–1727
45. Pajares G, Cruz JM (2007) Visión por computador: imágenes digitales y aplicaciones Fundamentos de Robótica. RA-MA., Segunda Edición, Madrid
46. Zhao YJ, Pei HL (2013) An improved vision–based algorithm for unmanned aerial vehicles autonomous landing. Appl Mech Mater 273:560–565
47. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) SURF: speeded up robust features. Comput Vis Image Underst 110(3):346–359
48. Saavedra-Ruiz M, Pinto-Vargas AM, Romero-Cano V (2018) Detection and tracking of a landing platform for aerial robotics applications. In: IEEE 2nd Colombian Conference on Robotics and Automation (CCRA). Barranquilla, pp 1–6
49. Saripalli S, Montgomery JF, Sukhatme GS (2002) Vision-based autonomous landing of an unmanned aerial vehicle. IEEE Int Conf Robot Autom 11–15:2799–2804
50. Saripalli S, Sukhatme G (2006) Landing on a moving target using an autonomous helicopter. Field and service robotics. Springer Tracts Adv Robot 2006:277–286
51. Cocchioni F, Mancini A, Longhi S (2014) Autonomous navigation, landing and recharge of a quadrotor using artificial vision. In: International conference on unmanned aircraft systems (ICUAS). pp 418–429
52. Li Y, Wang Y, Luo H, Chen Y, Jiang Y (2012) Landmark recognition for UAV autonomous landing based on vision. Appl Res Comp 07:2780–2783
53. Vega JA, Dormido-Canto S (2010) Máquinas de Vectores Soporte. Aprendizaje Automático (Pajares G. y de la Cruz J.M., Eds.). RA-MA, Madrid
54. Guili X, Yong Z, Shengyu J, Yuehua CH, Yupeng T (2009) Research on computer vision-based for UAV autonomous landing on a ship. Pattern Recogn Lett 30(6):600–605
55. Lange S, Sünderhauf N, Protzel P (2008) Autonomous landing for a multirotor uav using vision. In: Workshop Proceeding of SIMPAR 2008 International conference on simulation, modeling, and programming for autonomous robots. pp 482–491
56. S Lange N Sunderhauf P Protzel 2009 A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments. In: 2009 International conference on advanced robotics. IEEE, pp 1–6
57. Olson E (2011) AprilTag: a robust and flexible visual fiducial system. In: IEEE international conference on robotics and automation (ICRA)
58. AprilTag. Available online: https://april.eecs.umich.edu/software/apriltag.html Accessed June 2022
59. Ling K (2014) Precision landing of a quadrotor uav on a moving target using low–cost sensors. Master Thesis, University of Waterloo, Canada. UWSpace. https://uwspace.uwaterloo.ca/handle/10012/8803 Accessed June 2022
60. Kyristsis S, Antonopoulos A, Chanialakis T, Stefanakis E, Linardos C, Tripolitsiotis A, Partsinevelos P (2016) Towards autonomous modular UAV missions: the detection, geolocation and landing paradigm. Sensors 16(11):1844
61. Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Marín-Jiménez MJ (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn 47(6):2280–2292
62. Sani F, Karimian G (2017) Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. In: International Conference on Computer and Drone Applications (IConDA). pp 102–107

63. Detection of ArUco Markers. http://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html Accessed June 2022
64. Chaves SM, Wolcott RW, Eustice RM (2015) NEEC research: toward GPS–denied landing of unmanned aerial vehicles on ships at sea. Nav Eng J 127:23–35
65. Araar O, Aouf N, Vitanov I (2017) Vision based autonomous landing of multirotor uav on moving platform. J Intell Robot Syst 85:369–384
66. Kalman RE (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME. J Basic Eng 82:35–45
67. Rabah M, Rohan A, Talha M, Nam K, Kim S (2018) Autonomous vision-based target detection and safe landing for UAV. Int J Control Autom Syst 16:3013–3025
68. Lôbo-Medeiros FL, Faria-Gomes VC, Campos de Aquino MR, Geraldo D, Lopes-Honorato ME, Moreira-Dias LH (2015) A computer vision system for guidance of vtol uavs autonomous landing. In: Brazilian Conference on Intelligent Systems (BRACIS). pp 333–338
69. Nguyen PH, Arsalan M, Koo JH, Naqvi RA, Truong NQ, Park KR (2018) LightDenseYOLO: a fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone. Sensors 18(6):1703
70. Cesetti A, Frontoni E, Mancini A, Zingaretti P, Longhi S (2010) A vision-based guidance system for uav navigation. J Intell Robot Syst 57(1–4):233–257
71. Zongji C, Lei C, Rui Z, Weiqi L (2008) Unmanned aircraft landing navigation system based on vision. http://www.google.com/patents/CN101109640A Accessed June 2022. Patent CN 101109640
72. Roy P, Yu J, Linden DS (2008) Methods, apparatus and systems for enhanced synthetic vision and multi-sensor data fusion to improve operational capabilities of unmanned aerial vehicles. http://www.google.sc/patents/US20080215204 Accessed June 2022. Patent US20080215204
73. Blenkhorn KP, O'Hara SV (2009) Vision-based automated landing system for unmanned aerial vehicles. http://www.google.com/patents/US20090306840 Accessed June 2022. Patent US20090306840
74. Grzywna J, Da Frota B, Meuse S (2007) System and method for onboard vision processing. http://www.google.nl/patents/US20070093945 Accessed June 2022. Patent US20070093945
75. Shin-Je C, Jung-Ho M, Seung-Kie C, San K, Sung-Sik S (2008) Automatic recovery method of a UAV using image information to stably recover the UAV by obtaining real-time image information of a landing symbol on a recovery netting. http://patent.ipexl.com/KR/1020070058957.html Accessed June 2022. Patent 1020070058957
76. Avitzour D (1996) Mobile robot location determination employing error-correcting distributed landmarks. http://www.google.com/patents/US5525883 Accessed June 2022. Patent US5525883
77. Asaad A, Al-Salih M, Ahson SI (2009) Object detection and features extraction in video frames using direct thresholding. Multimedia, Signal Processing and Communication Technologies. IMPACT '09. International, pp 221–224
78. The MathWorks (2019) The Matlab. https://es.mathworks.com/products/matlab Accessed on June 2022
79. Otsu N (1979) A threshold selection method from gray-level histogram. IEEE Trans Syst Man Cybern 9:62–66
80. Kong TY, Rosenfeld A (1996) Topological algorithms for digital image processing. Elsevier Science, Inc.
81. Dougherty ER, Lotufo RA (2003) Hands-on morphological image processing. SPIE Tutorial Texts in Optical Engineering Vol. TT5. SPIE Publications
82. Dillencourt MB, Samet H, Tamminen M (1992) A general approach to connected-component labeling for arbitrary image representations. J Assoc Comp Mach 39(2):252–280
83. Sedgewick R (1998) Algorithms in C, 3rd Ed., Addison-Wesley, pp 11–20
84. Haralick RM, Shapiro GL (1992) Computer and robot vision, vol 1. Addison-Wesley, pp 28–48
85. Koschan A, Abido M (2008) Digital color image processing. John Wiley & Sons
86. Hu MK (1962) Visual Problem recognition by Moment Invariant. IRE Trans Inform Theory, IT-8. pp 179–187

87. Bohanec M (2009) Decision making: a computer-science and information-technology viewpoint. Interdiscip Descr Complex Syst 7(2):22–37
88. Duda RO, Hart PE, Stork DG (2006) Pattern classification, 2nd edn. New York; New Delhi: Wiley
89. The MathWorks (2016) Color-Based Segmentation Using the L*a*b* Color Space. https://es.mathworks.com/help/images/examples/color-based-segmentation-using-the-l-a-b-color-space.html Accessed June 2022
90. Campos Y, Sossa H, Pajares G (2016) Spatio-temporal analysis for obstacle detection in agricultural videos. Appl Soft Comput 45:86–97
91. Mercimek M, Gulez K, Mumcu TV (2005) Real object recognition using moment invariants. IEEE Trans Pattern Anal 30:765–775
92. Viola P, Michael J (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition, vol 1:511–518
93. Fan Z, Lu J, Gong M, Xie H, Goodman ED (2018) Automatic tobacco plant detection in UAV images via deep neural networks. IEEE J Select Top Appl Earth Observ Remote Sens 11(3):876–887
94. Bicer Y, Moghadam M, Sahin C, Eroglu B, Üre NK (2019) Vision-based UAV guidance for autonomous landing with deep neural networks. In: American Institute of Aeronautics and Astronautics. AIAA Scitech 2019 Forum, AIAA SciTech Forum, (AIAA 2019–0140)

Monitoring Peak Pollution Points of Water Resources with Autonomous Surface Vehicles Using a PSO-Based Informative Path Planner

Micaela Jara Ten Kathen, Princy Johnson, Isabel Jurado Flores, and Daniel Gutiérrez Reina

Abstract The preservation of water resources is an increasingly urgent issue. Therefore, monitoring the water quality of these resources is a very important task so that appropriate actions could be taken. This chapter focuses on water resource monitoring using a fleet of Autonomous Surface Vehicles equipped with sensors capable of measuring water quality parameters. The objective is to obtain the maximum points of contamination of the water resource through the exploration and exploitation of the water surface. The proposed algorithm is based on Particle Swarm Optimization (PSO) in combination with some machine learning techniques (Gaussian Process, Bayesian Optimization, among others) to address the limitations of PSO, such as premature convergence and difficulty in setting the initial values of the coefficients. To validate the performance of the algorithm, uni-modal and multi-modal benchmark functions are used in the simulation experiments. The results show that the proposed algorithm, the Enhanced GP-based PSO, based on the epsilon greedy method has the best performance for detecting water resource pollution peaks. It was also demonstrated that this algorithm is the one that generates the most accurate water quality model. However, when it comes to finding the highest pollution peak, the algorithm with the best response is the Enhanced GP-based PSO with a focus on exploitation.

Keywords Informative path planning · Particle swarm optimization · Autonomous surface vehicles · Machine learning · Water monitoring · Multi-modal problems

M. J. Ten Kathen (✉) · I. J. Flores
Loyola Andalucía University, Av. de las Universidades, s/n, 41704 Seville, Spain
e-mail: mcjaratenkathen@al.uloyola.es

I. J. Flores
e-mail: ijurado@uloyola.es

P. Johnson
Liverpool John Moores University, Byrom St, Liverpool, L3 3AF, UK
e-mail: p.johnson@ljmu.ac.uk

D. Gutiérrez Reina
University of Seville, C. Américo Vespucio, 41092 Seville, Spain
e-mail: dgutierrezreina@us.es

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_4

1 Introduction

Water has always been a valuable resource on the planet. However, there are major problems affecting this resource, two of which are: (1) water scarcity is increasing, and (2) human beings pollute water resources through industrial and agricultural activities, among others [7]. These problems not only affect the quality of life of human beings, but also affect the planet. One of the main consequences of water pollution by effluents can be algae blooms, which are caused by an excess of nutrients in the water (nitrates and phosphorus). This phenomenon is known as eutrophication [3]. Algae blooms are considered a major problem because they are toxic to human life and deplete oxygen from the water [2, 3]. This problem is increasingly common in the world; serious cases can be found in China, the United States, Kenya and Sri Lanka [11, 21, 39], among other places around the world.

In 2007, a massive cyanobacteria bloom in Taihu Lake, China, led to the interruption of the water supply for more than one million people [11]. It should be noted that in 1990 the lake had already suffered a similar situation, which led to the closure of 116 factories and left 3 million people without water supply [39]. A severe case of algae bloom in the United States occurred in Lake Erie in 2011. According to research [21], increased phosphorus loading in the western basin of the lake owing to agricultural operations, poor lake circulation, and meteorological conditions in the spring season that year were the main causes [21]. A case in point in Latin America is the contamination of the Ypacarai Lake, the largest lake in Paraguay (about 60 km^2). This lake has been polluted by domestic and industrial waste-water, agricultural activities, among others. As a result, the lake has an excess of nutrients, such as phosphorus and nitrates, thus promoting the appearance of cyanobacteria and affecting the population, the economy and the environment of the region [2].

Government agencies, research centers and other entities are constantly monitoring the water quality of water resources, as in the Ypacarai Lake. The traditional monitoring method consists of taking samples of the water body at specific points and analyzing these samples in laboratories. This method can be time consuming, human error can affect measurements, and costs can be high [43]. Several studies have proposed the use of intelligent vehicles for monitoring tasks [2, 24, 40]. These intelligent vehicles are surface vehicles that move autonomously, Autonomous Surface Vehicles (ASVs). The ASVs are guided by a guidance, navigation and control (GNC) system while taking water samples and measuring water quality through sensors for conductivity, nitrate, ammonium, among others. Figure 1 shows one of the Loyola University ASVs while performing control and navigation tests at Lago de la Vida, Dos Hermanas, Spain.[1]

The guidance system is in charge of determining the path that the ASVs must take. The path planners can be a global path planner [2], a local path planner [23], and an informative path planner [6]. Unlike the global and the local path planners, with the informative global path planner, the data collected by the sensors are used

[1] The following video shows the ASV operating at Lago de la Vida https://www.youtube.com/watch?v=3nYzSSGeyHw.

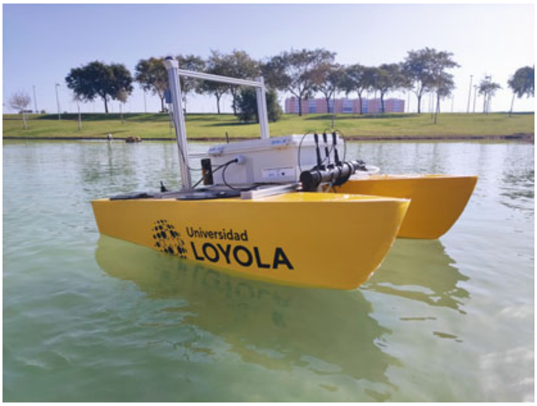


Fig. 1 An autonomous surface vehicle used for the water resource monitoring

to generate a path based on an underlying regression model that is used to make optimal decisions. This allows to create an estimated model of the water quality parameters of the aquatic environment through machine learning (ML) techniques [6]. In this work, a meta-heuristic approach like Particle Swarm Optimization is used as a framework for developing informative path planners for monitoring tasks. The Particle Swarm Optimization (PSO) technique is one algorithm that has received a lot of attention in recent years [20, 30]. This algorithm can be used as a planner. In this piece of research, PSO is used as an informative path planner for monitoring the maximum pollution points of a water body. In addition, Gaussian Process (GP) is proposed as a surrogate model for predicting water quality parameters. GP presents important advantages for water monitoring using AVSs such as generating models with a short dataset (low number of samples) and providing the uncertainty of the generated model.

In this work, a comparison of eight PSO-based path planners is presented: the Classic PSO; the Enhanced GP-based PSO [33], using two different combinations coefficient values, [34, 35]; 4 variants of the Enhanced GP-based PSO; and the Epsilon Greedy-based Enhanced GP-based PSO. The comparison metric used is the mean square error (MSE) between the estimated maximum pollution peaks and the actual maximum contamination points of the water resource. In addition, two more scenarios are presented: the performance of these algorithms to detect the highest pollution peak of the water resource, and to generate the most accurate model of water quality of the water resource. The main contribution of this work is the comparison of eight PSO-based path planners for monitoring maximum pollution peaks of a water resources using a fleet of autonomous surface vehicles.

The chapter layout is as follows: in Sect. 2, the state of the art of path planners is presented. In Sect. 3, the monitoring problem is explained, as well as the assumptions made to carry out the monitoring tasks. Section 4 explains the comparison of the path planners. Section 5 presents the simulation setup, experiments, and comparisons of the performances of the path planners. Section 6 provides a discussion of the results and future research directions. A conclusion of the work is presented in Sect. 7.

2 State of the Art

In recent years, the study of autonomous vehicles for the purpose of monitoring water resources has been increasing [2, 25, 40]. The monitoring is done through ASVs, equipped with sensors capable of measuring water quality parameters, which travel along the surface of the water resource taking water samples. Existing research on path planners to guide the vehicles can be found in the literature [1, 5, 24, 25]. Many path planners are based on artificial intelligence techniques that range from meta-heuristic algorithms like Genetic Algorithms (GA) [2] or Swarm-based intelligence [33], Deep Reinforcement Learning (DRL) [40], and Bayesian Optimization [25], among others.

As for meta-heuristic based approaches, the authors developed an offline path planner of an ASV in [2]. The monitoring problem was modeled as the Traveling Salesman Problem (TSP) and they used a meta-heuristic algorithm to solve it like a Genetic Algorithm (GA). In order to explore as much as possible, the authors set beacons on the lake shore. This work was extended in [1], where the monitoring problem was proposed as the Chinese Postman Problem (CPP). Unlike the TSP, with the CPP, vehicles can revisit beacons that they have previously visited, this improvement allows maximizing the coverage area to be explored. In [3], the authors proposed a two-phase approach for monitoring water resources. During the first phase, the algorithm covers the whole water body in order to find possible contaminated areas. After that, an exploitation phase is executed, where the vehicles focused on monitoring those contaminated areas found during the first stage. For both cases, the authors employed a genetic algorithm to determine the movement of an ASV. In [44], a multi-objective GA is employed for patrolling multiple water quality parameters simultaneously. The patrolling problem is different from the one proposed in this paper where the aim is to focus more frequently in those areas where higher values of contamination are found. The authors achieved Pareto-based solutions for the multi-objective problem using NSGA-II algorithm.

The path planner can also be designed using Deep Reinforcement Learning (DRL) as in [40] and Bayesian Optimization (BO) as in [25]. In [40], the authors have developed a global path planning for the monitoring of Ypacarai Lake, in which, the potential positions of the ASVs and the states are represented by RGB images of the lake. The main objective is patrolling the Ypacarai lake. A Markov Decision Problem is used to simulate the monitoring problem. The authors used a tailored reward function for training a Deep Q-learning approach based on convolutional neural networks. In [43], the authors extended the previous work, developing a multi-agent monitoring system and applied a centralized approach strategy. The authors proposed a centralized approach for training the whole fleet of ASVs with only one neural network. In addition, they demonstrated that the centralized approach is more suitable than the distributed Q-learning for the targeted patrolling problem. A comparison between Evolutionary Algorithms (EA) and DRL for monitoring systems was carried out in [41, 42]. The results showed that, in low resolution scenarios (patrolling problem), EA is more efficient. However, when the resolution of the scenario increases, DRL is more efficient. Despite requiring a greater number of hyper-parameters in the DRL than in the EA, it was shown that DRL is more suitable for problems with high complexity. In [25], the authors used the BO framework for developing an informative path planning for an ASV. The approach is focused on developing contamination maps for the Ypacarai Lake using GPs. Several tailored acquisition functions are proposed to determine the optimal locations for water quality parameter sampling. This work has been extended in [24] considering multiple water quality parameters simultaneously. The authors tested several multi-objective techniques for developing acquisition functions that fusion the information coming from several water quality parameter models to determine the optimal sample locations. A multi-agent version of this approach with region partitioning can be found in [26]. This variant allows the use of multiple ASVs to speed up the generation of contamination maps.

Regarding swarm-intelligence approaches for water monitoring, in [33], an IPP based on PSO and GP is proposed for monitoring water resources. The authors use Gaussian Processes (GP) to estimate water quality parameters. For the generation of the next points to be followed by the ASVs, the path planner uses the hyper-parameters of the PSO and the surrogate model data, uncertainty and mean. The results show that the monitoring system obtains a model very close to the real model of the water resource. The algorithm proposed by the authors, the Enhanced GP-based PSO, was compared with six other PSO-based path planners in [34, 35]. In [34], the main goal of the monitoring system was to obtain the most accurate water quality model of the Ypacarai Lake, so the ASVs should focus on exploring the surface to achieve the goal. The results show that the Enhanced GP-based PSO, tuned with the BO, had the best performance. On the contrary, in [35], the main objective was to detect and exploit the area where the highest peak of water resource contamination was located. The algorithm that performed the best was one of the variants of the Enhanced GP-based PSO, the Contamination algorithm, where the only active term in the algorithm, apart from the velocity term, was the contamination term. In [13], the authors propose a hybrid path planning algorithm, based on Grey Wolf Optimization (GWO) and PSO, composed of 3 stages. In the first stage, the algorithm based on GWO and PSO focuses on generating and optimizing the path, smoothing the path and minimizing the path distance. The second stage, converts the infeasible points of the generated path to feasible point solutions, integrating the first stage algorithm with a Local Search technique. And the last part consists of obstacle detection and collision avoidance, using a collision avoidance algorithm. The results showed that the algorithm proposed by the authors generates short feasible paths, being an efficient path planner.

This book chapter makes a step forward with respect to previous works in terms of bench marking PSO-based algorithms for water monitoring. The proposed approach consists of comparing some PSO-based route planners in solving a multi-modal problem in the area of water resources monitoring. The objective is to search for peaks of high pollution levels in a water resource. This task is carried out by eight algorithms and the performances of these algorithms are compared.

3 Statement of the Problem

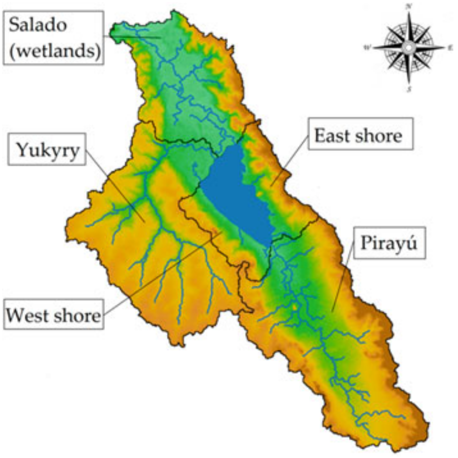
The monitoring challenge that the fleet of ASVs must solve, as well as the main assumptions that were used to create the simulated environment are explained in this section.

3.1 Monitoring Problem

Ypacarai Lake is the largest lake in Paraguay. This lake serves as a source of drinking water for neighboring cities, as a tourist attraction mainly in the summer season and also as a source of water for farms and crops in the surrounding area. Because of this, the contamination of the lake represents a major problem for society and for nature. The lake has two main inflows, the Pirayú stream and the Yukyry stream, and only one outflow, the Salado river, Fig. 2. The Pirayú stream is located towards the south of the lake, while the Yukury stream is located in the northwest of the lake. Other minor streams are found to the east and west of the lake shore. In addition to the lake, these streams are also often polluted [19]. Governmental and non-governmental entities conduct monitoring using the traditional monitoring method on a regular basis in order to control the pollution levels of the lake [9, 10]. Performing monitoring with ASVs will make the process more efficient and faster. As sampling with the traditional monitoring method is done at the shore of the lake, monitoring with ASVs will make it possible to detect areas of high contamination in any part of the lake, whether at the shore or in the middle of the lake.

In the monitoring system with ASVs, the water parameters are measured through water quality sensors S that are on board of the ASVs. With these measurements, the

Fig. 2 Inflows and outflow of the Ypacarai Lake [19]



water quality model of the water body can be obtained. The function $f(\mathbf{x})$ represents the real function of the water quality model, where $\mathbf{x}$ is the position (x, y) in the water resource. The water samples are taken in an environment in a sub-space of the $\mathbb{R}^n$ space. Each ASV takes n measurements. In the vector $\mathbf{s} = \{s_k \mid k = 1, 2, \ldots, n\}$ are stored the water samples taken by the sensors, the variable k is the number of measures taken. The positions where the samples were taken are stored in $\mathbf{q} = \{q_k \mid k = 1, 2, \ldots, n\}$. The vector $D = \{(q_k, s_k) \mid k = 1, 2, \ldots, N\}$ represents the data acquired by the fleet of ASVs. The input of the vector D represents the position q_k of the ASVs, and the output of the vector represents the sensor data s_k. Each D_k:

$$s_k = f(q_k) \tag{1}$$

The relationship between water quality parameters y and location $\mathbf{x}$ in the Ypacarai Lake domain can be estimated by applying the regression model given by Eq. 2. Given enough data D, a real function $f(\mathbf{x})$ can be calculated.

$$y \approx f(\mathbf{x}) \tag{2}$$

3.2 Assumptions

In order to implement the proposed monitoring system, some assumptions are taken into account:

- **Ypacarai Lake**: The monitoring space model is constructed of $m \times n$ squares of side d in a matrix $\mathcal{N}$. Each element $\mathcal{N}_{i,j}$ has a value that represents the current state of the grids, 0 (white) indicates forbidden zone, land, among others, and 1 (black) indicates the zone where the ASVs can travel. Figure 3 shows the occupation grid of Ypacarai Lake. For simulations, the distribution map was scaled, each element $\mathcal{N}_{ij}$ is 100 m × 100 m.
- **Coordinator**: A centralized system is used for control and communication between the ASVs. To this end, the ASV fleet is connected to a global coordinator via the cloud, using 4G or 5G technology. To avoid jams and bumps, a safety zone is considered on the lake shore, so ASVs are not allowed to travel too close to the lake shores.
- **Sensors**: The measurements made by the sensors are error-free. In addition, the position estimated by the GPS of the ASVs is free of noise.
- **Navigation**: The movements of the ASVs are flawless. Because of this, the trajectories of the ASVs are completely accurate. Obstacles and collisions are not considered. The speed limit for the ASVs is 2 m/s.
- **Vehicle autonomy**: Battery consumption is acceptable for the duration of the tests. The ASVs can reach a maximum distance of about 15,000 m.

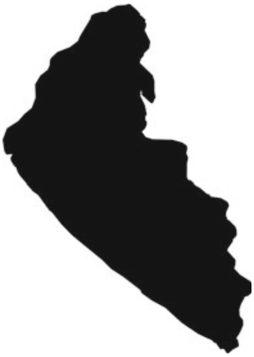


Fig. 3 Model of the occupancy grid of the Ypacarai Lake

4 PSO-Based Path Planning Algorithms

This section presents the informative path planners considered for the proposed comparison.

4.1 Classic Particle Swarm Optimization (PSO)

The PSO algorithm was developed by the authors in [15] using the social behavior of birds flocking as inspiration. PSO is a heuristic optimization algorithm in which each individual (particle) is a potential solution of the optimization problem. The particle has a position, which represents a possible solution, and a velocity, which determines the distance and direction each particle must travel. The velocity of the particle is updated according to the best position found by each particle (local best) and the best position found by all the particles in the swarm (global best) during the search process. The position of the particle is updated according the velocity. The velocity $\mathbf{v}_p^{t+1}$ and the position $\mathbf{x}_p^{t+1}$ of the pth particle at time step t, t represents discrete time steps, are described as follows:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_1 r_1^t \left[\mathbf{pbest}_p^t - \mathbf{x}_p^t\right] + c_2 r_2^t \left[\mathbf{gbest}^t - \mathbf{x}_p^t\right] \tag{3a}$$

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \mathbf{v}_p^{t+1} \tag{3b}$$

where w represents the inertia weighting, $\mathbf{pbest}_p^t$ is the local best of the particle and $\mathbf{gbest}^t$ is the global best of the swarm. These two terms refer to the exploitation and exploration of the search space respectively. The constants c_1 and c_2 are acceleration coefficients that describe the relative relevance of the local best and global best. r_1 and r_2 are random numbers in the interval [0, 1]. Notice that in the proposed monitoring system each particle will represent an ASV. Thus, the movements of the ASV will be influenced by the maximum level of contamination sensed by an ASV of the fleet $\mathbf{gbest}^t$ and its local maximum level, that is, the maximum contamination level sensed by an ASV so far $\mathbf{pbest}^t_p$.

4.2 Enhanced GP-Based PSO

This subsection is divided into four parts. The first part explains the surrogate model that is used to guide the vehicles to the positions of maximum uncertainty and zones of maximum contamination level of the water resource. The second part explains how the path planner itself works. In the third part, the BO, which is the technique selected for the optimization of the hyper-parameters, is described. Finally, the four variants of the Enhanced GP-based PSO that are also compared with the other path planners are described.

4.2.1 Gaussian Process Regression (GPR)

GP are Bayesian inference-based probabilistic machine learning models [27]. The input is constructed as a multivariate normal distribution. Each data of the input represents a random variable. The GP is completely defined by its mean function and covariance (or kernel) function [27]. However, the mean function is usually considered to be zero. Therefore, the component that defines the behavior of the GP is the kernel function. In the path planner to be analyzed, the kernel function plays the role of specifying the expected variability, smoothness and shape of the water quality parameters to be monitored. Taking into account the results of [25], the kernel function used for the experiments is the Radial Basis Function (RBF), because it is the most suitable for water resources.

The input data D is marginalized and conditioned to update the GPR [36]. For the computation of the unknown response $(\mu(\mathbf{x}_*), \sigma(\mathbf{x}_*))$ of the GPR $(\hat{f}(\mathbf{x})_*)$, the following set of Eq. 4 is used:

$$\mu_{\hat{f}(\mathbf{x}_i)_*|D} = K_*^T K^{-1} f(\mathbf{x}) \tag{4a}$$

$$\sigma_{\hat{f}(\mathbf{x}_i)_*|D} = K_{**} - K_*^T K^{-1} K_* \tag{4b}$$

where K, K_{**} and K_* are extracted from the fitted kernel, which includes covariances between known data $k(\mathbf{x}, \mathbf{x})$ and unknown data $k(\mathbf{x}*, \mathbf{x}*)$, as well as covariances between both the known and unknown data $k(\mathbf{x}, \mathbf{x}*)$.

$$K = \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} = \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, \mathbf{x}*) \\ k(\mathbf{x}*, \mathbf{x}) & k(\mathbf{x}*, \mathbf{x}*) \end{bmatrix} \tag{5}$$

4.2.2 Path Planner

Aiming to overcome the limitations of the Classic PSO, research in [33] developed an informative path planner based on PSO and GP (as a surrogate model). The two components of the GP are considered in the path planner, the covariance or kernel function (σ) and the mean function (μ). The kernel function is considered to guide the ASVs to areas of the water body that have not yet been explored, and the mean function is used for the purpose of exploiting areas where water resource contamination levels are high. The ASV velocity is updated according to Eq. 6 and the position is updated according to Eq. 3b.

$$\begin{aligned}\mathbf{v}_p^{t+1} &= w\mathbf{v}_p^t + c_1 r_1^t[\mathbf{pbest}_p^t - \mathbf{x}_p^t] + c_2 r_2^t[\mathbf{gbest}^t - \mathbf{x}_p^t] + c_3 r_3^t[\mathbf{max\_un}^t - \mathbf{x}_p^t] \\ &\quad + c_4 r_4^t[\mathbf{max\_con}^t - \mathbf{x}_p^t]\end{aligned} \tag{6}$$

where the coordinate of the maximum value of the uncertainty of the surrogate model uncertainty $max\sigma^t$ at tth iteration is represented by $\mathbf{max\_un}^t$ and the coordinate of the maximum value of the mean (or contamination) $max\mu^t$ of the surrogate model at tth iteration is represented by **max_con**. The constants c_3 and c_4, as well as c_1 and c_2, are acceleration coefficients that determine the importance of exploration (uncertainty) and exploitation (mean or contamination). These four acceleration coefficients must be tuned after starting the monitoring task.

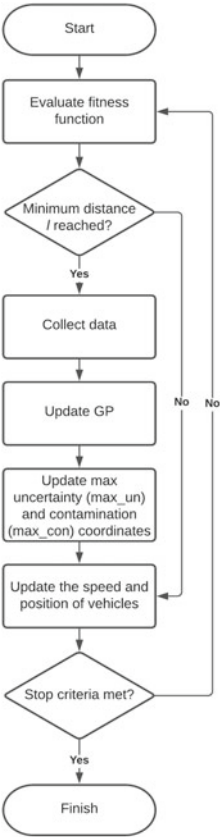
The water sample is taken after traveling a distance of l between the current ASV position ($\mathbf{x}^t$) and the last position where a sample was taken ($\mathbf{x}_{sample}$). The distance l is calculated by applying the Eq. 7:

$$l = \lambda \times \ell^t \tag{7}$$

where the term λ represents a ratio of the different length scales, and the ℓ^t refers to the posterior length scale of the surrogate model [24]. By applying this condition, the ASVs take samples at specific points, increasing the speed of execution of the monitoring system.

Figure 4 shows the flowchart of the path planner based on PSO and GP. At the beginning, the coordinates of the maximum uncertainty and maximum contamination are equal to 0. As a result, until the first sample is taken, the behavior of the path planner is the same as the Classic PSO. A water sample is taken after the ASV travels a minimum distance l. With these data, the GP is updated. Then, the coordinates of the maximum uncertainty and contamination values are found to update the ASV velocity and position.

Fig. 4 Flowchart of the enhanced GP-based PSO


Start
Evaluate fitness function
Minimum distance / reached?
Yes
No
Collect data
Update GP
Update max uncertainty (max_un) and contamination (max_con) coordinates
Update the speed and position of vehicles
Stop criteria met?
No
Yes
Finish

4.2.3 Hyper-Parameter Optimization

In order for the navigation system to behave correctly, the acceleration coefficients, c_1, c_2, c_3 and c_4, must be tuned. Hyper-parameter optimization techniques range from meta-heuristic algorithms to neural networks [4, 14, 18]. These techniques allow to obtain a better performance of the algorithms with reduced human effort [12]. The selected technique to tune the Enhanced GP-based PSO coefficients is the BO.

BO is a very successful optimization technique for solving tuning tasks, and it may be used in functions that are difficult to calculate, functions with a significant difficulty in analyzing their derivatives, and non-convex functions [37]. The procedure applies the Bayes theorem, Eq. 8, to integrate previous information about the unknown function with data information, G, to obtain the posterior of the function:

$$P(F|G) = P(G|F) \times P(F) \tag{8}$$

where $P(F|G)$ refers to the posterior probability of the model F; given model F, $P(G|F)$ represents the likelihood of overserving G; and $P(F)$ refers to the previous probability.

BO optimises a function using a surrogate model that is updated according to the information provided by the evaluated points [29]. The acquisition function, such as Probability of improvement (PI) [17], Expected improvement (EI) [22], GP upper confidence bound (GP-UCB) [32], is used to balance exploration and exploitation in the selection of the next location to explore. Readers are referred to [16, 29, 37] for a full examination of BO-based hyper-parameter optimization. The results obtained in [37] indicate that the PI function is slower than the EI and GP-UCB functions, and that GP-UCB is more complicated than EI because in the GP-UCB function hyper-parameters must be tuned. Because of this, EI has been selected as the acquisition function of the BO.

4.2.4 Variants of the Enhanced GP-Based PSO

These variants consist of keeping the velocity term and one of the exploration and exploitation term from Eq. 3a. The main objective is to evaluate the performance of the Enhanced GP-based PSO when only one of these terms is considered. The variants are as follows:

- **Local Best**: the active term in this algorithm is the local best $\mathbf{pbest}_p^t$ term. As a result, the trajectory generated by the algorithm is guided by the best position reached so far by the ASV. Since only the local best is considered, this algorithm focuses on the exploitation considering the zones where the ASV has already traveled. The velocity is updated according to Eq. 9:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_1 r_1^t[\mathbf{pbest}_p^t - \mathbf{x}_p^t] \tag{9}$$

- **Global Best**: in this algorithm the best position of the swarm found so far, the global best $\mathbf{gbest}^t$, is considered. The ASVs learn from the swarm experience and focus on surface exploration. The equation used for updating the velocity of the ASVs is shown below:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_2 r_2^t[\mathbf{gbest}^t - \mathbf{x}_p^t] \tag{10}$$

- **Uncertainty**: unlike the first two variants that consider the experience of the ASVs, this algorithm considers the data obtained from the surrogate model. This is because the term that remains active is the uncertainty of the model **max_un**. Because of this, the algorithm focuses on the exploration of unexplored areas. The velocity of the ASVs are updated according to Eq. 11:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_3 r_3^t[\mathbf{max\_un}^t - \mathbf{x}_p^t] \tag{11}$$

- **Contamination**: as the uncertainty algorithm, this algorithm considers the data obtained from the surrogate model, the mean of the model **max_con**. This allows the algorithm to focus on the exploitation of the areas with the highest levels of water resource contamination. The velocity of the ASVs is updated according to Eq. 12:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_4 r_4^t[\mathbf{max\_con}^t - \mathbf{x}_p^t] \tag{12}$$

The position of the ASVs in all variants of the algorithm is updated according to Eq. 3b.

4.3 Enhanced GP-Based PSO Based on Epsilon Greedy Method

The epsilon greedy method allows the PSO-based path planner to have dynamic coefficients. With dynamic coefficients, the enhanced GP-based PSO can balance exploration and exploitation by randomly choosing between the two. As a result, the path planner will be switching between exploration and exploitation phases.

Algorithm 1 shows the pseudocode of the Epsilon Greedy method. The value of the epsilon function has three parts. First, the epsilon function ε has a high value, 0.95. This means that the probability of exploring unknown areas is 95%. Once an ASV reaches a traveled distance $d\varepsilon_0$, the epsilon greedy moves to the second part. In this second part, the value of the epsilon function ε decreases $\Delta\varepsilon$ as the ASVs travel approximately 1,000 m. When an ASV reaches a traveled distance $d\varepsilon_f$, the value of the epsilon function ε is fixed at 0.05; this is the third part. In other words, the ASVs have a 5% of probability of exploring unknown areas and a 95% of probability of exploiting areas with high levels of contamination. After every 1,000 m traveled by the ASVs, a random number val is obtained and compared to the epsilon function ε. If the random number val is less than the epsilon function ε, the ASV's focus is exploring ($Explore$) the surface water resource, and if the random number val is greater than the epsilon function ε, the ASV's focus is exploiting ($Exploit$) areas with high levels of contamination. The behavior of the path planner is the same as that the Enhanced GP-based PSO (Sect. 4.2) once the coefficient values are defined.

In Algorithm 1, the average distance traveled by an ASV, p, between the current position and the initial position is represented by the term $dist_{\text{total}}$ and the average distance traveled by an ASV between the current position and the position where the last sample was taken is represented by $dist_{\text{sample}}$.

Algorithm 1: Enhanced GP-based PSO based on Epsilon Greedy method pseudo-code

```
$\mathbf{x}_p^0$ ← Initialize PSO; // where $\mathbf{x}^0$ represents the initial position of $p$
while not done do
    if $dist_{total} \leq d\varepsilon_0$ then
        $\varepsilon^t \leftarrow 0.95$
    else if $dist_{total} \geq d\varepsilon_f$ then
        $\varepsilon^t \leftarrow 0.05$
    else
        $\varepsilon^t \leftarrow \varepsilon^{t-1} - \Delta\varepsilon$
    $val \leftarrow random()$ if $\varepsilon^t$then then
        $c_1, c_2, c_3, c_4$ ← Explore
    else
        $c_1, c_2, c_3, c_4$ ← Exploit
    $\mathbf{pbest}_p$, $\mathbf{gbest}$ ← Evaluate fitness function;
    $dist \leftarrow dist_{\text{total}} - dist_{\text{sample}}$ ← Calculate distance;
    if $dist \geq l$ then
        $s$ ← Collect sensor data;
        $\sigma^t$, $\mu^t$ ← Update GP;
        $max\sigma^t$, $max\mu^t$ ← Calculate maximum values;
        $\mathbf{max\_un}^t$, $\mathbf{max\_con}^t$ ← Find coordinates of the maximum values;
    $\mathbf{v}_p^{t+1}$, $\mathbf{x}_p^{t+1}$ ← Update speed and position of the particles;
```

5 Results

This section presents the obtained simulations results. First, the ground truth function used as water quality parameter model is described. Second, the performance metric used to compare the algorithms is mentioned. Third, the simulation parameters used for the comparison are detailed. Next, the results of the hyper-parametrization carried out for the PSO-based algorithms are shown. Finally, the performance comparison of the algorithms is presented.

The path planners were developed in Python and the following libraries were used: Scikit-learn[2] (accessed on 5 May 2022), DEAP[3] (accessed on 5 May 2022) and Bayesian Optimization[4] (accessed on 5 May 2022).

[2] https://scikit-learn.org/stable/.

[3] https://deap.readthedocs.io/en/master/.

[4] https://github.com/fmfn/BayesianOptimization.

5.1 Ground Truth

As ground truth a scaled model of the Ypacarai Lake is used, each element has a dimension of 100 m x 100 m. A benchmark function, the Shekel function (Eq. 13), is used as a distribution map of water quality parameters. This function is used because it is a multidimensional, multimodal, deterministic, and continuous function [24].

$$f_{\text{Shekel}}(\mathbf{x}) = \sum_{i=1}^{M} \frac{1}{c_i + \sum_{j=1}^{N}(x_j - a_{ij})^2} \tag{13}$$

The function allows specifying multiple maximum points by means of two arguments, a_{ij} and c_{ij}. The element a_{ij} belongs to the matrix A. This matrix has a size of $M \times N$, M corresponds to the number of maximum points and N to the dimensions of the space. The matrix C, whose elements are c_{ij} and whose size is $M \times 1$, defines the inverse significance value of the maximum positions. Ten different ground truths are used to analyze the behavior of path planners. Since the objective is to detect several contamination peaks, the value of M varies between 2 and 4. These peaks are located near the flow inlets of Ypacarai Lake, the location and number of peaks are randomly selected. The value of N is fixed at 2. The values of the elements of c are random numbers. Figure 5 shows an example of ground truth used in the simulations. It is possible that the peaks are located in prohibited areas.

5.2 Performance Metric

The mean squared error (MSE) is used as a comparison metric to evaluate the performance of the path planners for two of the situations presented in this work, the main objective being the detection of several peaks of water resource contamination, and

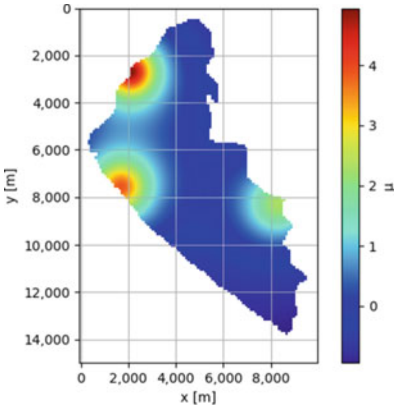


Fig. 5 Example of one of the ten ground truth obtained with the Shekel function used for the simulations of the monitoring system in Ypacarai Lake

for obtaining the most accurate model of the water quality of the water resource. The MSE is often used to evaluate regression models. This is the reason for the selection of this metric as an evaluation method. The Eq. 14 represents the MSE between the real model or ground truth ($f(\mathbf{x})$) and the estimated model or the resulting prediction (y) of the surrogate model to compare the peaks of water resource contamination.

$$\text{MSE}_{peaks}(f(\mathbf{x}), y) = \frac{1}{n_{peaks\_points}} \sum_{k=0}^{n_{peaks\_points}-1} (f(\mathbf{x}_k) - y_k)^2 \tag{14}$$

The main objective is to detect the points of maximum contamination of the water body. Therefore, the MSE between the contamination peaks of the real model or ground truth and the contamination peaks of the estimated model is calculated. As a result, the path planner with the best MSE (low MSE) is considered the optimal path. Then, to obtain the most accurate model of the water quality of the water resources, the MSE between all the coordinates of the actual model and the estimated model is calculated. The Eq. 16 represents the MSE between the real model or ground truth ($f(\mathbf{x})$) and the estimated model or the resulting prediction (y) of the surrogate model to compare the water quality model of the lake.

$$\text{MSE}_{map}(f(\mathbf{x}), y) = \frac{1}{n_{map\_points}} \sum_{k=0}^{n_{map\_points}-1} (f(\mathbf{x}_k) - y_k)^2 \tag{15}$$

Another situation is to detect the highest peak pollution of the water resource. To evaluate the performance of the path planners, the error between maximum ground truth peak pollution and the maximum pollution peak of the estimated model is calculated. The Eq. 16 represents the error between the real model or ground truth ($f(\mathbf{x})$) and the estimated model or the resulting prediction (y) of the surrogate model to compare the highest contamination peak of the water resource.

$$\text{Error}_{highest\_peak}(f(\mathbf{x}), y) = |f(\mathbf{x}_{highest\_peak}) - y_{highest\_peak}| \tag{16}$$

5.3 *Setting Simulation Parameters*

This subsection presents the general settings of the simulations. In addition, the results of the hyper-parametrization carried out for the PSO-based algorithms are shown.

5.3.1 Global Simulation Parameters

The mission will be carried out using four ASVs. Each ASV covers around 15,000 m and travels at a top speed of 2 m per second. The inertia is set to a value of one. In order for the length scale to provide the necessary smoothness, the value should be set to 10% of the length of the search space [25]. Therefore, the value of the length scale is set to 10. The value of λ is set on 0.3. With this value, the ASVs take around 120 samples. The stopping criteria of the path planners is the distance traveled by the ASVs, when an ASV has traveled 15,000 m, the simulation ends.

5.3.2 Results of Hyper-Parametrization of Enhanced GP-Based PSO

In this work, two cases of the Enhanced GP-based PSO have been considered, [34, 35]. The difference between these cases is the main objective of the monitoring task. Because of this, the combination of the coefficient values are different in both cases. The hyper-parametrization of each case is presented below.

Enhanced GP-Based PSO (Exploration)

For the hyper-parametrization presented in [34], the main idea was to obtain the most accurate water quality model. As a consequences, the algorithm focus on the exploration of the water resources surface. Two cases were considered for the hyper-parametrization. In the first case, 200 ground truth were evaluated for the BO. In the second case, the number of ground truth evaluated was 300. For both cases BO will be used for hyper-parameter optimization, with the selected acquisition function being EI. The kernel function used for hyper-parameter optimization was the default function of the Bayesian-Optimization library, Matérn, with parameter values equal to $nu = 2.5$ and $\ell = 1.0$. The number of BO iterations was set to 20, and 10 random starting points were considered.

The range of the PSO hyper-parameters (coefficient acceleration of the local best, c_1; coefficient acceleration of the global best, c_2; coefficient acceleration of the maximum uncertainty, c_3; and coefficient acceleration of the maximum contamination, c_4) is determined according to [28, 31], Table 1. Table 2 shows the results obtained in each case of hyper-parameter optimization for the exploration approach.

The comparison between the MSE results obtained from the monitoring system performance of each case is shown in Table 3. The MSE values are very similar, the difference is 0.003. This is because the values of the coefficients obtained in both optimization cases do not differ much, with c_3 being the dominant coefficient, and c_2 and c_4 being 0 or practically 0. In other words, the monitoring system is driven by the maximum uncertainty of the model and by the best local of each particle. The

Table 1 Range for hyper-parameter optimization

Hyper-parameter	Range
c_1	[0, 4]
c_2	[0, 4]
c_3	[0, 4]
c_4	[0, 4]

Table 2 Hyper-parameter values in each case of the BO for the enhanced GP-based PSO (Exploration)

Hyper-parameter	Case 1	Case 2
	200 ground truth	300 ground truth
c_1	2.0187	3.0970
c_2	0	0.0125
c_3	3.2697	3.5863
c_4	0	0.0382

Table 3 Comparison of the MSE of the BO cases for the enhanced GP-based PSO (exploration)

Case	MSE
1	**0.02650 ± 0.02727**
2	0.02916 ± 0.02952

values of the coefficients for comparison with the other methods will be the values of Case 1, since it has the best MSE.

Enhanced GP-Based PSO (Exploitation)

Unlike the work presented in [34], the hyper-parameterization objective presented in [35] was to find the highest contamination peak. Because of this, the focus of the algorithm was to exploit the areas of high contamination level. The cases considered in this work were the same as the cases presented in Sect. 5.3.2. Since the objective of this work was to obtain the highest contamination peak, the comparison metric used was the error committed between the actual value of the peak and the estimated value of that peak. In the first case, the mean of the error of 200 ground truth was considered for the BO, and for the second case, 300 ground truth was considered. The configurations of the acquisition function, the kernel function and the BO used were also the same as the configurations presented in Sect. 5.3.2. Also, the ranges of the coefficients used for the hyper-parametrization are the same as those presented in the Table 1. Table 4 shows the results obtained in each case of hyper-parameter optimization for the exploitation approach.

The comparison of the errors made between the actual value of the peak and the value estimated by the algorithm are presented in Table 5. The difference between the error values is less that 0.003. The main difference between the two cases is the values of the local and global best coefficients. While in case 1, the value of local best is low and global best is high, in case 2 local best has a higher value than global best. This means that in case 2, the exploitation of the area is more intensive since the local best and maximum contamination terms allow the algorithm to perform the exploitation.

Table 4 Hyper-parameter values in each case of the BO for the Enhanced GP-based PSO (exploitation)

Hyper-parameter	Case 1	Case 2
	200 ground truth	300 ground truth
c_1	0.5683	3.6845
c_2	3.4160	1.5614
c_3	0	0
c_4	3.1262	3.6703

Table 5 Comparison of the MSE of the BO cases for the enhanced GP-based PSO (exploitation)

Case	Error
1	0.00818 ± 0.03139
2	**0.00540 ± 0.01399**

5.3.3 Setting Epsilon Greedy Method

Before making the comparison between the algorithms, four cases of the epsilon greedy method have been compared. In two cases, the values of the coefficients for the exploration stage are: $Explore$: $c_{1explore} = 1$, $c_{2explore} = 4$, $c_{3explore} = 4$, $c_{4explore} = 1$; and for exploitation stage are: $Exploit$: $c_{1exploit} = 4$, $c_{2exploit} = 1$, $c_{3exploit} = 1$, $c_{4exploit} = 4$. The values of the coefficients are determined according to the range given in [28, 31]. For the other two cases, the values for the exploration are the values obtained from the BO in the paper [34]: $Explore$: $c_{1explore} = 2.0187$, $c_{2explore} = 0$, $c_{3explore} = 3.2697$, $c_{4explore} = 0$, and for the exploitation stage, the values are the values obtained from the BO in the paper [35]: $Exploit$: $c_{1exploit} = 3.6845$, $c_{2exploit} = 1.5614$, $c_{3exploit} = 0$, $c_{4exploit} = 3.1262$. The values for the cases are shown in Table 6.

Since the vehicles travel approximately 15,000 m each, for the cases 1 and 3, the values of $d\varepsilon_0$ and $d\varepsilon_f$ were set by dividing the total distance into three equal parts, 5,000 and 10,000 m respectively. For the second part of the algorithm, the value of ε decreases by 0.18 every 1,000 m traveled. For the cases 2 and 4, the distance was increased for the second part, the values of $d\varepsilon_0$ and $d\varepsilon_f$ are 3,000 and 12,000 m, respectively, and the value of ε decreases by 0.116 every 1,000 m traveled.

In addition to detecting all contamination peaks of the Ypacarai Lake (multi-modal problem), two other situations were considered: detecting the highest contamination peak (uni-modal problem), and creating the most accurate model of the lake. The MSE values for all cases and situations are shown in Table 7. For the uni-modal problem, the error between the actual value and the estimated value of the peak was calculate.

The case with the best MSE for detecting all the peaks in the lake map was obtained with case 1, where the values of the coefficients were 1 and 4. However, case 3 also

Table 6 Parameter and hyper-parameter values in each case of the Epsilon Greedy method

Parameter/hyper-parameter	Case 1	Case 2	Case 3	Case 4
$d\varepsilon_0$ (m)	5,000	3,000	5,000	3,000
$d\varepsilon_f$ (m)	10,000	12,000	10,000	12,000
$\Delta\varepsilon$	0.18	0.116	0.18	0.116
$c_{1explore}$	1	1	2.0187	2.0187
$c_{2explore}$	4	4	0	0
$c_{3explore}$	4	4	3.2697	3.2697
$c_{4explore}$	1	1	0	0
$c_{1exploit}$	4	4	3.6845	3.6845
$c_{2exploit}$	1	1	1.5614	1.5614
$c_{3exploit}$	1	1	0	0
$c_{4exploit}$	4	4	3.1262	3.1262

Table 7 Comparison of the MSE and error of the Epsilon Greedy cases

Case	MSE (several contamination peaks)	Error (a contamination peak)	MSE (model of the lake)
1	**1.62302 ± 4.32203**	1.18271 ± 3.48666	0.07334 ± 0.12163
2	1.71359 ± 4.56390	1.27596 ± 3.63597	0.07645 ± 0.13131
3	**1.62643 ± 3.64945**	**0.00984 ± 0.03666**	**0.07018 ± 0.10068**
4	1.65896 ± 4.30441	0.33264 ± 1.67548	0.07327 ± 0.12753

obtained a good MSE, with a difference of about 0.004 with respect to case 1, and it also has the smallest range of variability. As for the situation where there is only one contamination peak, the best MSE is in case 3. Likewise, when the objective is to obtain the most accurate model of the lake, case 3 has the best MSE. Therefore, the coefficient values of case 3 are selected for comparison with the other algorithms.

5.4 Performance Comparison

After selecting the best configuration of PSO-based algorithms, comparisons are made with the other algorithms. The values selected for the Epsilon Greedy method are the values of case 3. These results are compared with the Classical PSO, the Enhanced GP-based PSO using the values obtained from the [34, 35] papers (described in previous subsections), and with the four variants of the Enhanced GP-based PSO. The Tables 8 and 9 show the values of the coefficients for each algorithm [8, 38] were considered to determine the coefficient values for the Classic PSO. Since the objective is to observe the influence of the terms, for the Local Best, Global Best,

Table 8 Acceleration coefficients values

Hyper-parameter	Local best algorithm	Global best algorithm	Uncertainty algorithm	Contamination algorithm	Classic PSO	Enhanced GP-based PSO (exploration)	Enhanced GP-based PSO (exploration)
c_1	3	0	0	0	2	2.0187	3.6845
c_2	0	3	0	0	2	0	1.5614
c_3	0	0	3	0	0	3.2697	0
c_4	0	0	0	3	0	0	3.1262

Table 9 Acceleration coefficients values for the Epsilon Greedy method

Hyper-parameter	Value
$c_{1explore}$	2.0187
$c_{2explore}$	0
$c_{3explore}$	3.2697
$c_{4explore}$	0
$c_{1exploit}$	3.6845
$c_{2exploit}$	1.5614
$c_{3exploit}$	0
$c_{4exploit}$	3.1262

Uncertainty and Contamination algorithms was decided not to set the maximum value of the range.

For the experiments, the following considerations were taken into account: (a) each vehicle starts from a different starting point than the others, however, in all simulations, each vehicle starts from the same initial position; (b) the error and MSE calculations are performed considering 10 different scenarios; (c) the water sample is taken after the vehicles have traveled a distance l (Eq. 7); and (d) the simulation ends when one of the vehicles travels 15,000 m.

The best performing algorithm for detecting contamination peaks is the Enhanced GP-based PSO based on Epsilon Greedy method, where the values of the coefficients are set to the values obtained in [34] for the exploration stage and [35] for the exploitation stage. This can be verified by the results shown in the MSE (Several Contamination Peaks) column of the Table 10. As in Sect. 5.3.3, two more situations were considered, to detect the highest peak of contamination and to generate the most accurate estimated model of lake water quality. As expected, the algorithm that best detects the highest peak is the Enhanced GP-based PSO (Exploitation). By a small difference, the second with the smallest error is the Contamination algorithm. With respect to the generation of the most reliable water quality model, the one that obtained the best result is the Enhanced GP-based PSO based on the Epsilon Greedy method, and in second place, the Enhanced GP-based PSO (Exploration). These results demonstrate that the changes of approach, exploration and then exploitation,

Table 10 Comparison of the MSE and the error of the eight algorithms

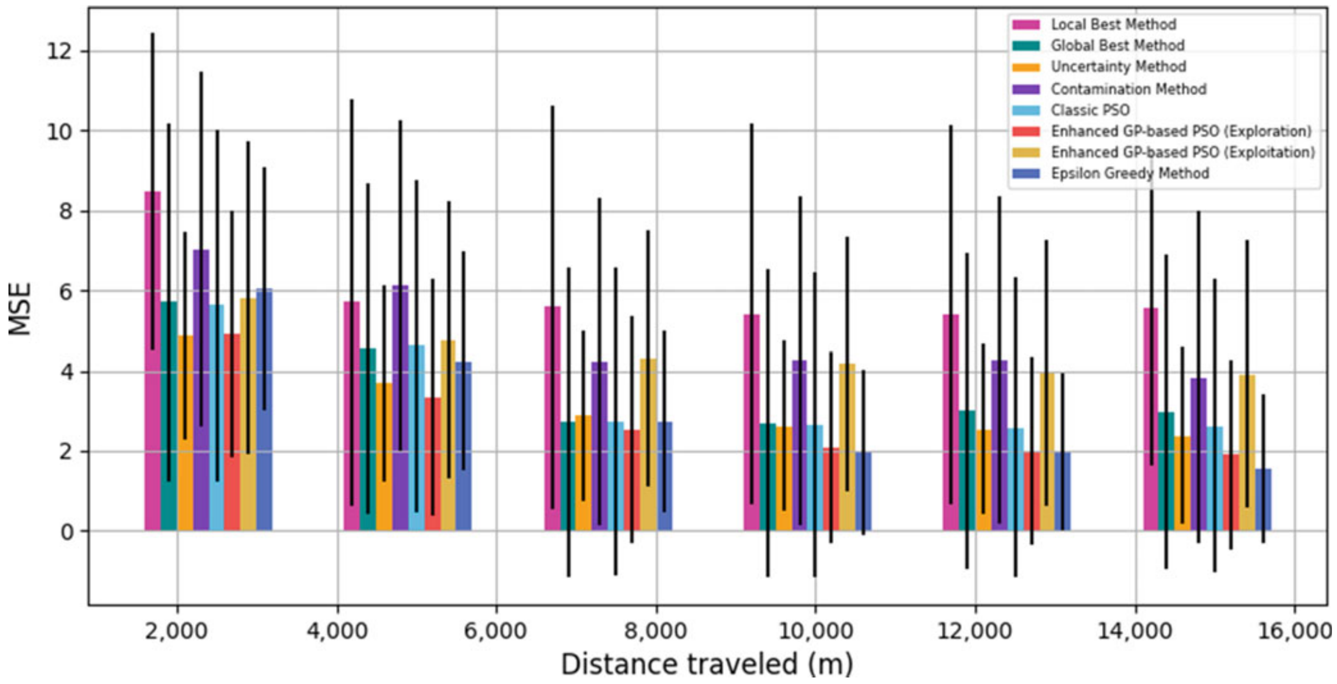
Algorithm	MSE (Several Contamination Peaks)	Error (A Contamination Peak)	MSE (Model of the Lake)
Local Best	2.68687 ± 7.34544	0.89471 ± 3.19382	0.27867 ± 0.60573
Global Best	2.69164 ± 7.43596	2.36964 ± 4.20148	0.17335 ± 0.31163
Uncertainty	2.45316 ± 4.39283	2.11673 ± 3.61938	0.13680 ± 0.22764
Contamination	4.04210 ± 8.18967	0.00224 ± 0.01044	0.29742 ± 0.59782
Classic PSO	2.69269 ± 7.16223	1.73219 ± 4.00487	0.16819 ± 0.29399
Enhanced GP-based PSO (Exploration)	1.94280 ± 4.66284	1.62445 ± 3.92491	0.07465 ± 0.12707
Enhanced GP-based PSO (Exploitation)	3.80756 ± 6.56296	**0.00164 ± 0.00494**	0.21760 ± 0.42043
Epsilon Greedy Method	**1.62643 ± 3.64945**	0.00984 ± 0.03666	**0.07018 ± 0.10068**

during the experiment is the best line to solve multi-modal problems, which is the case of detecting the various peaks of contamination of the water resource.

The comparison between the eight algorithms at different distances traveled is shown in Fig. 6. To improve the visualization of the mean MSE and its 95% confidence interval at the end of the episode, Fig. 6b is shown, which is a zoom of Fig. 6a at the end of the episode (15,000 m traveled). Throughout the episode, the three algorithms with the highest mean of MSE are the Local Best algorithm, the Contamination algorithm, and the Enhanced GP-based PSO (exploitation). This is because these three algorithms focus on exploiting the highest peak, so by not exploring the surface, they are not able to detect all the contamination peaks.

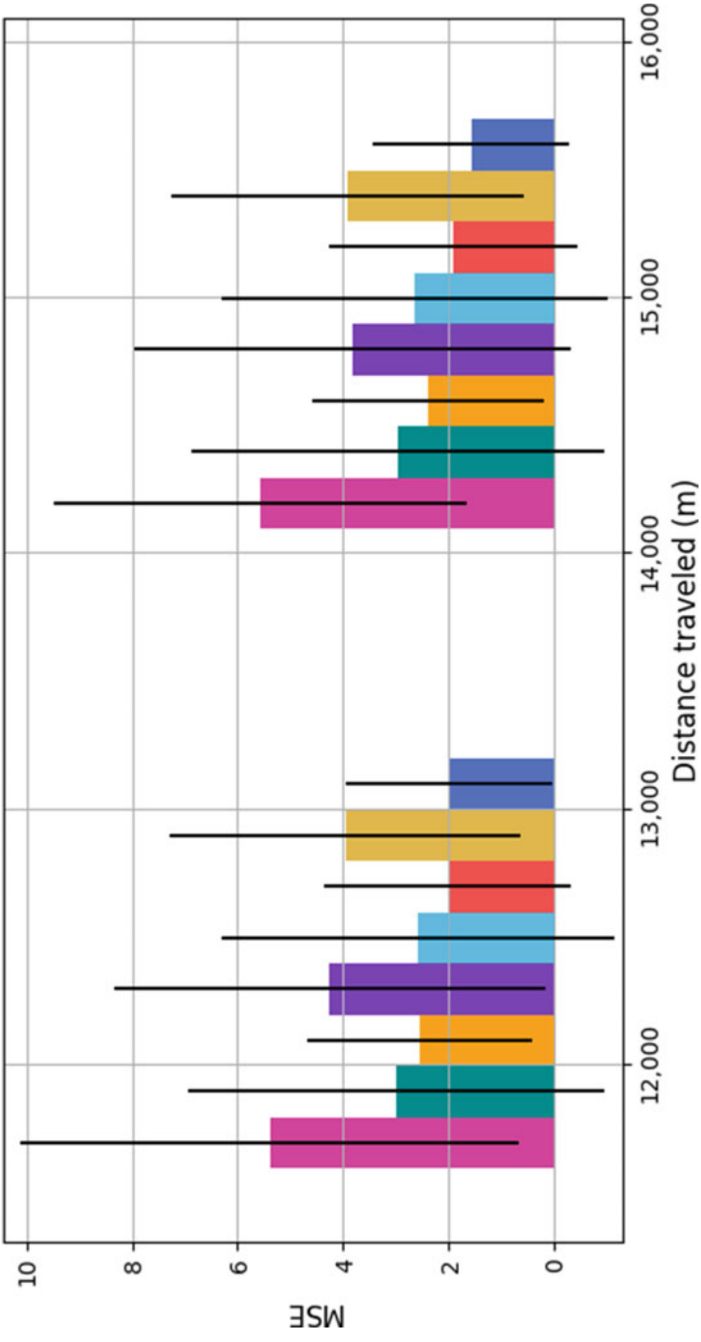
Figure 7 shows the results obtained in the experiments of the different algorithms in one of the scenarios. In the map at the top, the movement made by the vehicles and the uncertainty in the exploration of the lake are shown. The movement of each ASV is represented by a line of a different color from the others. The map at the bottom shows the estimated model of the water quality parameters.

Figure 7a shows the results obtained with the Local Best algorithm. Since in this algorithm only the term corresponding to the best position of the ASV is kept active, the ASV focuses on exploiting the area where the samples it has taken have a high level of contamination. The disadvantage of this algorithm is that the ASVs do not explore, so if two ASVs are near a peak and both are far from a second peak, the two ASVs would go to the peak that is closer, because the water samples would have a higher value near that peak and they would not notice a second peak that is far away. As a result, the ASVs get stuck in a one of the optimum. In Fig. 7b, the ASVs movements are guided by the best position of the swarm. Because of this, all ASVs go to the same zone or peak, so they do not explore the other areas with high contamination level. The vehicle movements in Fig. 7c are guided by the maximum uncertainty term of the surrogate model. This allows the ASVs to explore the entire surface of the lake, being able to go to areas where contamination is high. However,



(a)

Fig. 6 Mean of the MSE and its 95% confidence interval for the detection of zones with high levels of contamination of the Local Best algorithm, the Global Best algorithm, the Uncertainty algorithm, the Contamination algorithm, the Classic PSO, the enhanced GP-based PSO (exploration), the enhanced GP-based PSO (Exploitation), and the Epsilon Greedy method

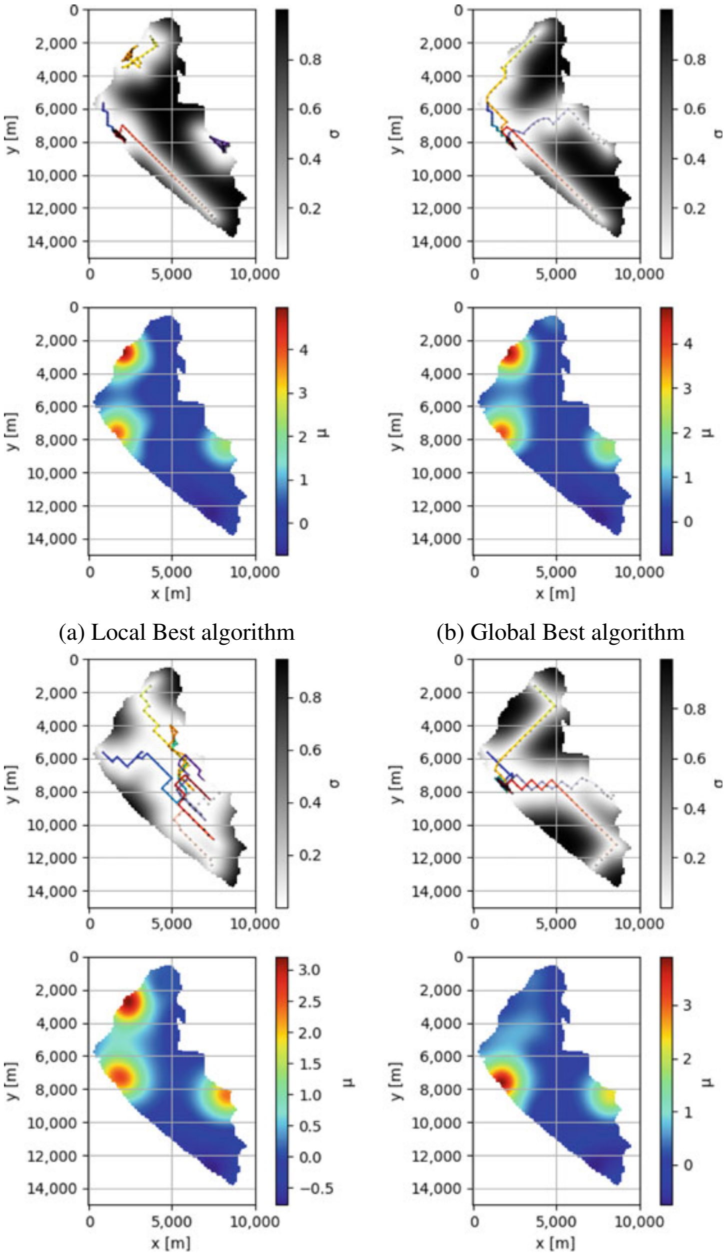


(b)

Fig. 6 (continued)

they do not stop in those areas to exploit pollution peaks. In Fig. 7d, the term that remains active is the maximum contamination term. This allows the ASVs to exploit the zone with the highest level of contamination. However, because the ASVs do not explore the surface lake, the estimated model is not accurate, so the algorithm could have a false global maximum. In other words, the pollution maximum estimated by the algorithm is a local maximum and not a global maximum, as shown in Fig. 7d. If the estimated model is compared with the ground truth (Fig. 5), it will be seen that the global maximum of the ground truth is in the northwest of the lake but the algorithm detects that the maximum is in the west of the lake. This is because the ASVs were not able to explore the surface and there is not enough data to obtain a good estimation of the water quality model. The movement of the ASVs with the Classic PSO is shown in Fig. 7e. The ASVs learn from their own experience and from the swarm, in other words, they consider the positions they have passed through and do not consider the data of the surrogate model. Because of this, after having explored and found a peak, they focus on exploiting that peak and get stuck in that area. This result can be expected since classical PSO algorithm is not intended for multi-modal problems. For the Enhanced GP-based PSO, two cases are considered: one using the coefficient values obtained in [34] (Fig. 7f), where the objective was to obtain the most accurate model of the lake water quality, and the second case is using the coefficient values obtained in [35] (Fig. 7g), where the goal was to detect the highest pollution peak. Because in Fig. 7f, the coefficients of the global best and maximum contamination are 0, the movement of the ASVs is guided by the local best and maximum uncertainty terms. This combination allows the exploration of the lake surface because it uses the uncertainty of the surrogate model, as well as the exploitation of the zones through which the ASVs are passing, since it considers the best position of the ASVs. The local best term acts as gradient for the ASV movements. In contrast, in Fig. 7g, the movement of the ASVs is mainly guided by the terms of local best and maximum contamination, the best global also has a small influence on the movements of the ASVs. Because of this, the ASVs try to reach the area where the maximum pollution peak is located and while heading to that area, they exploit the zones they pass through. However, the same case as for the Contamination algorithm can be seen with this combination of coefficient values. Since there is not enough data to estimate a good model, the ASVs go to one of the local peaks and get stuck in that area. Figure 7h represents the results obtained by applying the Epsilon Greedy method with the values of case 3 (Sect. 5.3.3). At the beginning, the exploration probability is high, so they explore more lake surface while exploiting the areas they are passing through, this is due to the combination of coefficient values. As the ASVs increase the distance traveled, the exploration probability decreases and they focus more on exploiting the areas with high levels of contamination. Because the algorithm focuses more on exploration at the beginning, the surrogate model was able to generate an accurate model, so when the probability of exploitation is higher, the ASVs target the true global maximum.

Figure 8 shows the movement of the vehicles, with the Epsilon Greedy method, during different distances traveled, as well as the evolution of the uncertainty data and the model estimated by the surrogate model. The initial points of the ASVs



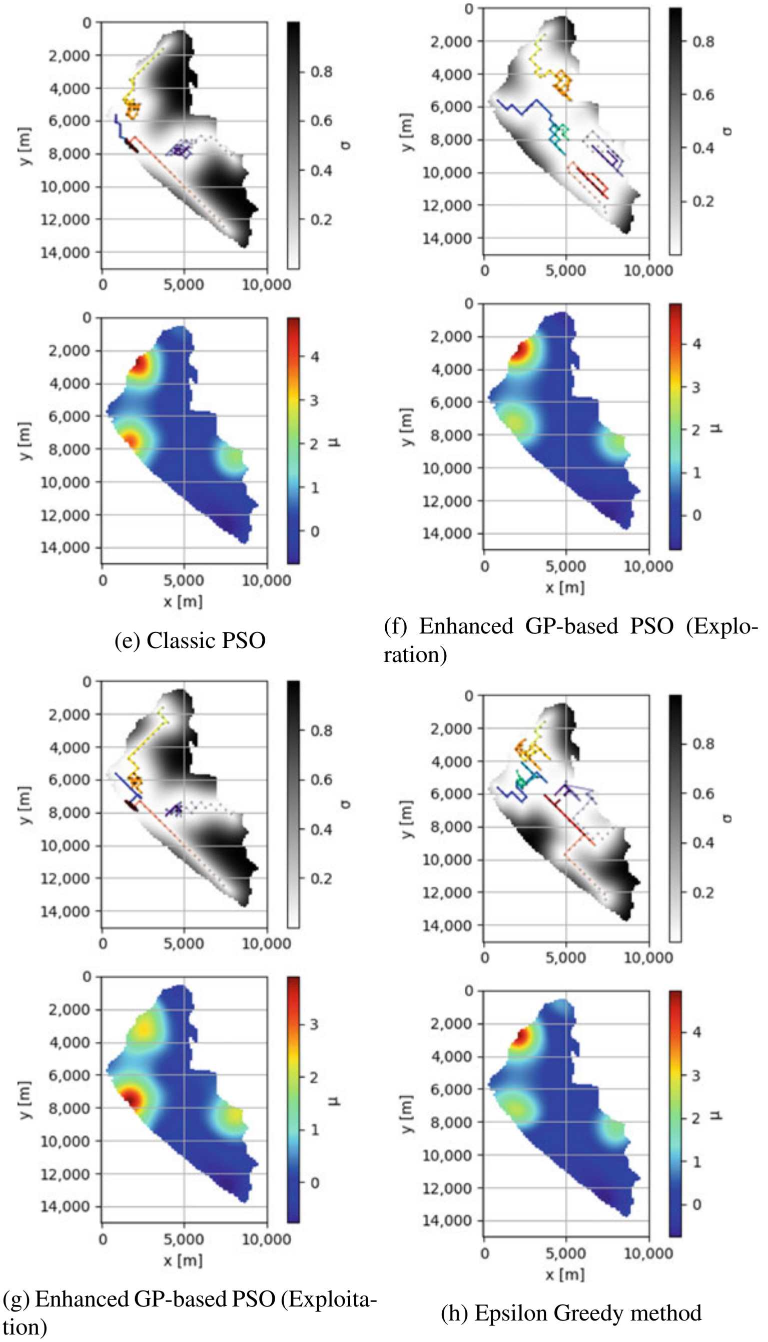
(a) Local Best algorithm

(b) Global Best algorithm

(c) Uncertainty algorithm

(d) Contamination algorithm

Fig. 7 Representation of the Ypacarai Lake monitoring system. The ASV trajectories and model uncertainty are represented in the top graphs (black and white map with coloured line traces), while the surrogate model mean is represented in the bottom graph (coloured map)



(e) Classic PSO

(f) Enhanced GP-based PSO (Exploration)

(g) Enhanced GP-based PSO (Exploitation)

(h) Epsilon Greedy method

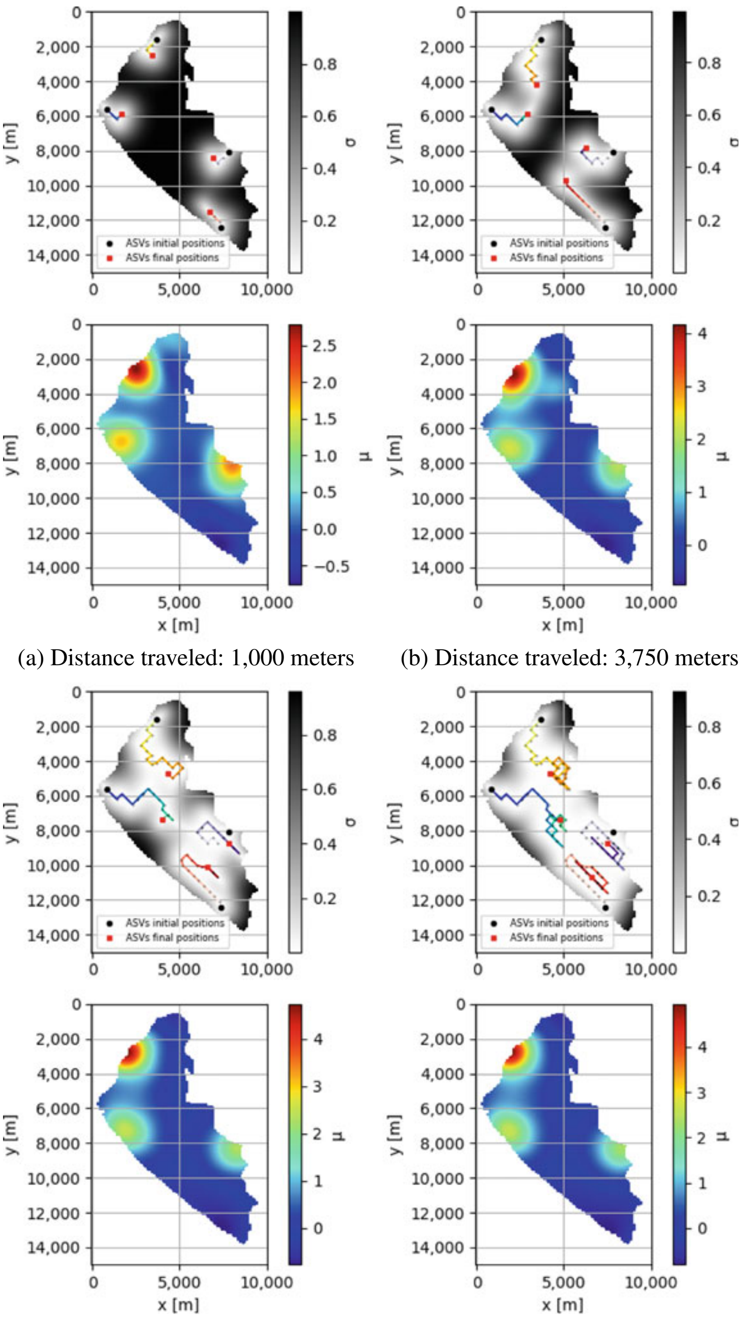
Fig. 7 (continued)

are represented by a black circle, and the end point at that instant is represented by a red square. Figure 8a represents the movement of the vehicles after 1,000 m traveled. The figure above shows how the uncertainty is high in the middle of the lake and the generated model estimates low values compared to the real lake model. As the vehicles move around, the uncertainty decreases and the areas of highest contamination are being better detected in the generated model as shown in Fig. 8b with 3,750 m traveled. In Fig. 8c, the vehicles have traveled 7,500 m, half of the episode. At that instant, the uncertainty is high in only a few sectors of the map and, with the samples taken by the vehicles, it is now possible to define the areas where pollution is high. Finally, Fig. 8d shows the end of the episode, where most of the lake was already explored and the contamination zones could be fully detected.

6 Discussion of the Results

The main findings of this work are discussed below:

- The Local Best algorithm focuses on exploiting the areas through which the ASVs pass. Because of this, if the ASVs detect a peak, they get stuck on that peak because they have no way to escape from that zone. With this algorithm, each ASVs can exploit different areas as in Fig. 7a.
- Unlike the Local Best algorithm, in the Global Best algorithm most of the surface can be explored. However, as all ASVs are guided by a single best position, the fleet targets the same peak and is not able to exploit different areas.
- The Uncertainty algorithm has the advantage that ASVs are able to explore the surface of the water resource and target unexplored areas. However, they cannot exploit areas with high levels of contamination like the Contamination algorithm.
- The advantage of the Contamination algorithm is the ability to exploit the area with the highest contamination peak, but the algorithm is not able to exploit local peaks or explore the surface.
- The Enhanced GP-based PSO with the coefficient values obtained in [34] is able to obtain an accurate water quality model of the water resource. This algorithm combines exploration of unexplored areas and exploitation of the areas through which the ASVs pass. Nonetheless, for solving multi-modal problems in the monitoring system, this algorithm obtained the second best performance, so it needs some adjustments to improve its performance.
- The Enhanced GP-based PSO with the values of the coefficients obtained in [35] has the best performance for detecting the highest contamination peak. Nevertheless, when there is more than one peak, the algorithm is not able to find all the peaks because it does not scan the surface and may get stuck on a local peak.
- The algorithm that has the best performance in detecting water resource contamination peaks is the Enhanced GP-based PSO based on the Epsilon Greedy method. This is due to the ability of the algorithm to switch approaches during the episode, between exploration and exploitation, having a high probability of exploration at



(a) Distance traveled: 1,000 meters

(b) Distance traveled: 3,750 meters

(c) Distance traveled: 7,500 meters

(d) Distance traveled: 15,000 meters

Fig. 8 Representation of the Ypacarai Lake monitoring system in different distances traveled. The black dots represent the initial position of the ASV, while the red square represents the final position of the ASV at that instant

the beginning and a high probability of exploitation at the end. This allows the algorithm to generate an accurate model of the lake, and then, with a good model estimated, to exploit the areas where there are high levels of contamination.
- It is clear that further research is required to determine the optimal dynamic of the Enhanced GP-based PSO algorithms.

7 Conclusions

This work consisted of the simulation and comparison of different path planners based on PSO whose main objective was to detect the different peaks of contamination of a water resource. In addition to detecting the contamination zones, two other situations were analyzed, the performance of the path planners to detect the highest peak of contamination and the ability to generate a water quality model of the water resource as similar as possible to the real model or ground truth. The algorithms that were considered for comparison are: the Classic PSO; the Enhanced GP-based PSO with two different combinations of coefficients, one of them is the one obtained in [34], which focuses on exploring the surface of the water resource and obtaining an accurate model of it, and the other combination is the one obtained in [35], which focuses on exploiting the area with the highest contamination of the water resource; the Enhanced GP-based PSO based on the Epsilon Greedy method, this method allows having dynamic coefficients during the episode, due to this, the path planner can change its focus during the monitoring; and four variants of the Enhanced GP-based PSO, where in each variant the velocity term and one of the exploration (c_2 and c_3) or exploitation (c_1 and c_4) terms were left active with the objective of observing the influence of these terms in the monitoring of the zones with a high level of contamination. The results showed that the path planner that has the best performance to detect the different contamination zones and obtain the most accurate model of the water quality of the water resource is the Enhanced GP-based PSO based on the Epsilon Greedy method using the values of the coefficients obtained in [34, 35]. The algorithm that has the best performance in finding the highest pollution peak is the Enhanced GP-based PSO with the data obtained in [35]. These results show that in order to solve multi-modal problems, such as detecting the different pollution peaks, the focus of the algorithm must change during the search process. From this, it is proposed as a future direction of research, the development of a multi-modal informative path planning based on PSO for monitoring tasks, where the focus of the algorithm will be in setting the appropriated values of the PSO coefficient according to the mission to be carried out. This is different from what is considered in the Epsilon-greedy algorithm that switch between exploration and exploitation randomly.

Acknowledgements This work has been partially funded by the regional government Junta de Andalucía under the Projects “Despliegue Inteligente de una red de Vehículos Acuáticos no Tripulados para la monitorización de Recursos Hídricos US-1257508”, and “Despliegue y Control de una

Red Inteligente de Vehículos Autónomos Acuáticos para la Monitorización de Recursos Hídricos Andaluces PY18-RE0009”.

References

1. Arzamendia M, Espartza I, Reina DG, Toral S, Gregor D (2019) Comparison of Eulerian and Hamiltonian circuits for evolutionary-based path planning of an autonomous surface vehicle for monitoring Ypacarai lake. J Ambient Intell Humaniz Comput 10(4):1495–1507
2. Arzamendia M, Gregor D, Reina DG, Toral SL (2019) An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of Ypacarai lake. Soft Comput 23(5):1723–1734
3. Arzamendia M, Reina DG, Toral S, Gregor D, Asimakopoulou E, Bessis N (2019) Intelligent online learning strategy for an autonomous surface vehicle in lake environments using evolutionary computation. IEEE Intell Transp Syst Mag 11(4):110–125
4. Birattari M, Kacprzyk J (2009) Tuning metaheuristics: a machine learning perspective, vol 197. Springer
5. Blanchard A, Sapsis T (2020) Informative path planning for anomaly detection in environment exploration and monitoring. arXiv:2005.10040
6. Bottarelli L, Bicego M, Blum J, Farinelli A (2019) Orienteering-based informative path planning for environmental monitoring. Eng Appl Artif Intell 77:46–58
7. CfO (2015) Transforming our world: the 2030 agenda for sustainable development. United Nations, New York, NY, USA
8. Cui Y, Zhong J, Yang F, Li S, Li P (2020) Multi-subdomain grouping-based particle swarm optimization for the traveling salesman problem. IEEE Access 8:227,497–227,510
9. Dirección General del Centro Multidisciplinario de Investigaciones Tecnológicas (CEMIT) (2018) Servicios de monitoreo de calidad de agua por campañas de muestreo en el lago ypacaraí. 2016–2018. Technical report, Universidad Nacional de Asunción (UNA)
10. Dirección General del Centro Multidisciplinario de Investigaciones Tecnológicas (CEMIT) (2021) Monitoreo de calidad de agua por campañas de muestreo en el lago ypacaraí 2019–2021. Technical report, Universidad Nacional de Asunción (UNA)
11. Duan H, Loiselle SA, Zhu L, Feng L, Zhang Y, Ma R (2015) Distribution and incidence of algal blooms in lake Taihu. Aquat Sci 77(1):9–16
12. Feurer M, Hutter F (2019) Hyperparameter optimization. Automated machine learning. Springer, Cham, pp 3–33
13. Gul F, Rahiman W, Alhady S, Ali A, Mir I, Jalil A (2021) Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO-GWO optimization algorithm with evolutionary programming. J Ambient Intell Humaniz Comput 12(7):7873–7890
14. Guo B, Hu J, Wu W, Peng Q, Wu F (2019) The Tabu_Genetic algorithm: a novel method for hyper-parameter optimization of learning algorithms. Electronics 8(5):579
15. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95-international conference on neural networks, vol 4. IEEE, pp 1942–1948
16. Kim Y, Chung M (2019) An approach to hyperparameter optimization for the objective function in machine learning. Electronics 8(11):1267
17. Kushner HJ (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise
18. Lessmann S, Caserta M, Arango IM (2011) Tuning metaheuristics: a data mining based approach for particle swarm optimization. Expert Syst Appl 38(10):12,826–12,838
19. López Moreira GA, Hinegk L, Salvadore A, Zolezzi G, Hölker F, Monte Domecq SRA, Bocci M, Carrer S, De Nat L, Escribá J et al (2018) Eutrophication, research and management history of the shallow Ypacaraí lake (Paraguay). Sustainability 10(7):2426

20. Marini F, Walczak B (2015) Particle swarm optimization (PSO). A tutorial. Chemom Intell Lab Syst 149:153–165
21. Michalak AM, Anderson EJ, Beletsky D, Boland S, Bosch NS, Bridgeman TB, Chaffin JD, Cho K, Confesor R, Daloğlu I et al (2013) Record-setting algal bloom in lake Erie caused by agricultural and meteorological trends consistent with expected future conditions. Proc Natl Acad Sci 110(16):6448–6452
22. Mockus J, Tiesis V, Zilinskas A (1978) The application of Bayesian methods for seeking the extremum. Towar Glob Optim 2(117–129):2
23. Peralta F, Arzamendia M, Gregor D, Reina DG, Toral S (2020) A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: the Ypacarai lake case-study. Sensors 20(5):1488
24. Peralta F, Reina DG, Toral S, Arzamendia M, Gregor D (2021) A Bayesian optimization approach for multi-function estimation for environmental monitoring using an autonomous surface vehicle: Ypacarai lake case study. Electronics 10(8):963
25. Peralta F, Reina DG, Toral S, Arzamendia M, Gregor D (2021) A Bayesian optimization approach for water resources monitoring through an autonomous surface vehicle: The Ypacarai lake case study. IEEE Access 9:9163–9179. https://doi.org/10.1109/ACCESS.2021.3050934
26. Peralta F, Yanes S, Reina DG, Toral S (2021) Monitoring water resources through a Bayesian optimization-based approach using multiple surface vehicles: the Ypacarai lake case study. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp 1511–1518
27. Rasmussen CE (2003) Gaussian processes in machine learning. In: Summer school on machine learning. Springer, pp 63–71
28. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput 8(3):240–255. https://doi.org/10.1109/TEVC.2004.826071
29. Roman I, Ceberio J, Mendiburu A, Lozano JA (2016) Bayesian optimization for parameter tuning in evolutionary algorithms. In: 2016 IEEE congress on evolutionary computation (CEC). IEEE, pp 4839–4845
30. Sánchez-García J, Reina DG, Toral S (2019) A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario. Futur Gener Comput Syst 90:129–148
31. Senthil Arumugam M, Rao M, Chandramohan A (2008) A new and improved version of particle swarm optimization algorithm with global-local best parameters. Knowl Inf Syst 16:331–357. https://doi.org/10.1007/s10115-007-0109-z
32. Srinivas N, Krause A, Kakade SM, Seeger M (2009) Gaussian process optimization in the bandit setting: no regret and experimental design. arXiv:0912.3995
33. Ten Kathen MJ, Flores IJ, Reina DG (2021) An informative path planner for a swarm of ASVs based on an enhanced PSO with gaussian surrogate model components intended for water monitoring applications. Electronics 10(13):1605
34. Ten Kathen MJ, Flores IJ, Reina DG (2022) A comparison of PSO-based informative path planners for autonomous surface vehicles for water resource monitoring. In: 2022 7th international conference on machine learning technologies (ICMLT), pp 271–276
35. Ten Kathen MJ, Reina DG, Flores IJ (2022) A comparison of PSO-based informative path planners for detecting pollution peaks of the Ypacarai Lake with autonomous surface vehicles. In: International conference on optimization and learning, OLA'2022
36. Williams C, Rasmussen C (2006) Gaussian processes for machine learning, vol 2. MIT Press, Cambridge, MA
37. Wu J, Chen XY, Zhang H, Xiong LD, Lei H, Deng SH (2019) Hyperparameter optimization for machine learning models based on Bayesian optimization. J Electron Sci Technol 17(1):26–40
38. Xin J, Li S, Sheng J, Zhang Y, Cui Y (2019) Application of improved particle swarm optimization for navigation of unmanned surface vehicles. Sensors 19(14):3096
39. Xu H, Paerl HW, Qin B, Zhu G, Gaoa G (2010) Nitrogen and phosphorus inputs control phytoplankton growth in eutrophic lake Taihu, China. Limnol Oceanogr 55(1):420–432
40. Yanes S, Reina DG, Toral S (2020) A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: the Ypacarai lake case. IEEE Access 8:204,076–204,093

41. Yanes S, Peralta F, Reina DG, Toral S (2021) A sample-efficiency comparison between evolutionary algorithms and deep reinforcement learning for path planning in an environmental patrolling mission. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp 71–78
42. Yanes S, Reina DG, Toral S (2021) A dimensional comparison between evolutionary algorithm and deep reinforcement learning methodologies for autonomous surface vehicles with water quality sensors. Sensors 21(8):2862
43. Yanes S, Reina DG, Toral S (2021) A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: the Ypacarac-lake patrolling case. IEEE Access
44. Yanes S, Peralta F, Córdoba AT, del Nozal ÁR, Marín ST, Reina DG (2022) An evolutionary multi-objective path planning of a fleet of ASVs for patrolling water resources. Eng Appl Artif Intell 112(104):852

Robot-Based Medicine. Robots in Medicine: Mobile Robots Versus Mobile Decision, Necessity Versus Possibility and Future Challenges

Zbigniew Nawrat and Dariusz Krawczyk

Abstract Medicine as a process and practice, i.e., the medical technology of the future, will be based on robotics. Our cooperation with robots, both as service providers and recipients of medical services, will require appropriate behaviour and education from us, including modification of applicable laws and ethics, also. Engineers are facing challenges to make robots cheaper, easier to use and safe. The robot allows us to multiply the work. Telecommunications robots and technology allow us to transfer our competences and possibilities of action to where they are most important at the moment–despite the distance from the patient. So we can contrast with robot mobility–information mobility. Do we already have Robot-Based Medicine today? And do we have any other solution to the problems of humanity and the health service? This chapter reviews issues related to the use of robots in medicine; from technical to ethical. The significant use of robotic mobility in medicine was pointed out. Mobility of robots has different meanings in various fields of medicine and different methods of successfully using robots there are being developed. Robots were created on Earth, by man, in the image of man and for man. This is the next stage in the history of tools and … human evolution.

Keywords Robots · Medical robots · Mobile robots · Surgical robots · Artificial intelligence

Z. Nawrat (✉)
Department of Biophysics, Faculty of Medical Sciences in Zabrze, Medical University of Silesia, Katowice, Poland
e-mail: znawrat@sum.edu.pl; nawrat@frk.pl

Z. Nawrat · D. Krawczyk
Prof. Zbigniew Religa Foundation of Cardiac Surgery Development, Zabrze, Poland

Z. Nawrat
International Society for Medical Robotics, Zabrze, Poland

D. Krawczyk
Department of Biophysics, Doctoral School for Medical Science, Faculty of Medical Sciences in Zabrze, Medical University of Silesia, Katowice, Poland

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_5

List of Notations and Abbreviations

AMRs	Autonomous mobile robots
AI	Artificial intelligence
ML	Machine Learning
MIS	Minimally Invasive Surgery
DOF	Degrees of Freedom
FRK	Fundacja Rozwoju Kardiochirurgii im. prof. Zbigniewa Religi (Foundation of Cardiac Surgery Development)

1 Introduction

Medicine is a contact between a person who has knowledge, tools and medicines with a person who requires help. This is how it is today. In terms of consulting and partly diagnosis, this help can already be performed remotely, using intermediary devices. However, when life is in danger, we send an ambulance to the patient, with appropriate technical equipment and a trained rescue team. Often the patient has to be transported to a competence center, where the operation is performed and the specialist treatment process is continued. This mobility of the medical team (and the patient) is an essential element in the process of providing assistance and treatment.

After the successes of telemedicine, which proved to be indispensable during the last pandemic and during the war in Ukraine, the time has come for the faster development of medical robots. Telemedicine is sending information remotely, if we want to be able to act remotely–we need robots.

It is thanks to the implementation of serial production technology, automation and robotization that the present high level of quality of human life is possible. Medicine struggling with staff problems, growing social expectations and costs exceeding economic possibilities–needs robotization. Just like the introduced machines, automatic machines and robots transformed manufactories into industrial plants, influencing politics and social relations at the turn of the nineteenth and twentieth centuries. Now there is a change in the form of medical services.

The author's conviction about the importance of robots for the good of humans has been shaped by years of work in the team of an outstanding, innovative cardiac surgeon–prof. Zbigniew Religa. The author participated in the development and implementation of a mechanical heart support system and conducted work on the Robin Heart surgical robot (Fig. 1). Thousands of patients around the world who live thanks to artificial organs (artificial heart, dialysis, AI diabetes surveillance systems, etc.), rehabilitated and move around thanks to robots (e.g., skeletons), 1.5 million operations a year performed with the help of robots–it is hard to find more convincing evidence that today robots save lives, care for its quality and change the standards of work performance.

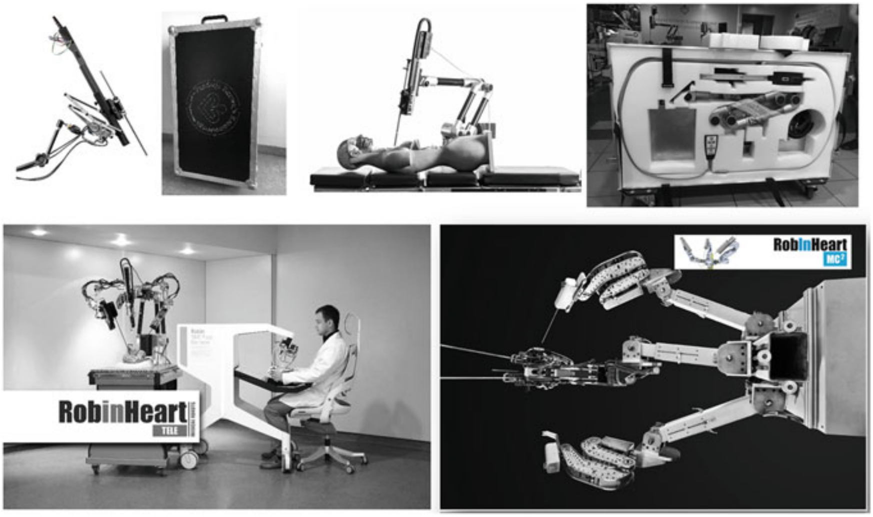


Fig. 1 Polish Medical Robots: Robin Heart family: Robin Heart Pelikan (portable), Robin Heart Port VisionAble (portable), TeleRobinHeart and RobinHeart mc^2

The chapter attempts to answer the following questions: when, where and why to send information over a distance, and when real physical objects should arrive by land, water or air; when, where and why to use remotely controlled robots (telemanipulators) or automatic or autonomous robots; what are the scientific and technical, legal or ethical problems to be solved depending on the adopted method of transport or control. Mobile robots are an important part of medical robotics, necessary to raise standards and efficiency of medical service.

A selection of medical robots that perform a variety of tasks is presented and the opportunities and risks, and limitations specific to medical applications that should be considered when designing the robots, are analyzed. And we will focus on selecting the type of robot control depending on the purpose of the medical application. Importantly, for the design of mobile robots, we consider all potential architectural, service and application issues specific to medical processes.

This chapter is an attempt at presentation of the directory of problems dedicated to choosing the method of providing services using mobile robots; risk–benefit analysis by comparing of mobile robots versus telemedicine methods ("mobile decision"), mobile versus stationary, remote diagnostics, therapy, rehabilitation or rescue operation robots.

2 Work and Medicine

A doctor is concerned with health (now and after, maintaining or restoring, forecasting and preventing) by examining, diagnosing and treating (diseases, injuries, and other physical and mental conditions).

A person can perform medical services in the form of intellectual and physical work. He is equipped with tools, used various physical, chemical, biological phenomena and knowledge. However, the most common work of a doctor is the process of analyzing data and making decisions. In the case of treatment specialties, he also undertakes certain physical activities. The doctor examines, diagnoses, prevents diseases, heals and rehabilitates. Health service also include giving opinions, giving advice and issuing prescriptions or medical certificates.

Medicine is a collection of knowledge about health and disease, about human functioning in the physical and mental aspect. It is the science of health and disease. Medicine is highly interdisciplinary. Its basic sciences include: biology, histology, anatomy, physiology, biochemistry, embryology, biophysics, and immunology. Modern research and diagnostic devices and methods introduced into medicine make it possible to create medical facts for a given patient. Evidence-based medicine (EBM) in the treatment process uses reliable evidence proving the safety and effectiveness of the selected therapeutic procedure. The work of the EBM is as follows: reducing the problem to a question, obtaining the best evidence, critically assessing the consistency of the evidence, applying the results in practice, and evaluating effectiveness. It sounds like a recipe, an algorithm, i.e., we are one step away from being able to define the patient's problem in a digitized, objective way, i.e., suitable for a robot. So, evidence-**based medicine leads to robot-based medicine**.

There are different ways of practicing medicine. The work of a psychologist and a surgeon is definitely different. Medicine deals with man in various situations. There is sports medicine, forensic medicine, emergency medicine, prenatal medicine, palliative medicine, tropical medicine, occupational medicine and aesthetic medicine. Each is a different way of working, and different challenges for the technical support of these services.

There are also many fields of medicine: from allergology through surgery, epidemiology, cardiology, oncology, psychiatry to sexology and dentistry. The analysis of the treatment process in each case is a completely different specificity of activity, intervention, relationship with the patient and the doctor. These are many challenges for biomedical engineers, robotics and AI specialists.

3 Medical Robots

Where has medical robotics been the most successful so far? We can note both economic and practical successes of robotics in the field of surgery and rehabilitation (including exoskeletons).

Surgery. The surgeon's needs are characterized by information about used tools (the global laparoscopic device and accessories market size was valued over $13.7 billion in 2020) and number of performed surgery (over 14 million laparoscopic procedures) [21].

Rehabilitation service needs range reached about 2.41 billion individuals in 2019. Today requires even more specialists and introducing these services as an integral part of primary health care, to reach easier access for more people in need [7].

Who is the robot leader in the soft tissue telemanipulator market? Obviously the da Vinci (Intuitive Surgical) robot, due to the number of medical applications and financial benefits for investors. Currently, almost 6,000 da Vinci systems are in operation, which have performed 8.5 million procedures worldwide (currently over 1 million operations per year). But, after the basic patent lock was removed, competitors, new surgical robots such as HUGO (Medtronic), Versius (Cambridge Medical Robotics CMR), Titan (Titan Medical Inc.) started to appear on the market. The robots also help cardiologists (Coridus Vascular Robotics), neurosurgeons (Renishaw's Neuromate® Gen III), orthopedics (Renaissance by Mazor Robotics), oncologists for lung endoscopy (Monarch by Auris Health) and … hairdressers (Artas–robot platform for hair implantation). We get to know other advantages, new fields of robot application [37].

The Verb Surgical founded in 2015 by Alphabet owned by Google and Johnson & Johnson company presented itself very promisingly. The goal was to create a robot based on the experience of Ethicon and SRI International (Stanford) in the new standard of "digital surgery". Johnson & Johnson (J&J) through its subsidiary Ethicon, after acquiring companies with expertise in machine learning (Verb Surgical) and endoscopic procedures (Auris Health), also plans to expand into the surgical robot market by creating the Ottava robot. Ottava will consist of six robotic arms integrated into the operating table. Johnson & Johnson are also expanding into orthopedics with robot VELYS [31]. Artificial intelligence technology was used to reduce errors, facilitate training and support research results [51].

Trend–**portability.** The portability and ease of use of robotic systems will likely be one of the main advantages of da Vinci's competitors such as the Versius (CMR), which has already been used to perform 5,000 treatments worldwide [8, 31]. Medtronic previously purchased the German Miro (DLR) robot and is now launching the clinical implementation of its HUGO robot [32]. Hugo is not as beautiful as Miro (awarded for design) but he seems to be functional and maybe have a chance (because of compatibility with the globally-trusted Medtronic surgical instrumentation and is ready for AI technology after acquisition of Digital Surgery Ltd) to compete with Versius (similarity: Hugo is modular and have open-console design) and da Vinci. The Senhance Surgical System (Asensus) is a modular, multi-arm telemanipulator system–greater than the Versius and Hugo RAS systems. The 5 mm instruments have no wrist articulation and uses traditional laparoscopic trocars similar to Versius system. Robots use the classic mechanical operation of tools or physical phenomena: laser, electric, thermal, ultrasonic. Let's not forget about extremely useful (synchronizes the work according to the movement of the chest during the breath) Cyberknife System (Accuray). The CyberKnife system (USA) is the first and the only dedicated radiosurgical system designed to treat tumors with adequate accuracy anywhere in the body. Available in over 150 hospitals, about 100,000 patients [37].

Cardiology. Robot-assisted PCI (percutaneous coronary intervention) improve radiation safety for the operators.

Siemens Healthineers bid USD 1.1 billion for Corindus Vascular Robotics, the U.S. company that was one of the first to develop robotic-assisted systems for minimally invasive vascular interventions [41]. Hansen Medical–the Californian company was founded in 2002 and has created two systems for robotic cardiological intervention–transvascular electrophysiology, Sensei Robotic Catheter System and vascular therapy (e.g. endovascular stenting) Magellan Robotic System.

Diagnostic. The remote transmission of the doctor's activity is tested in ultrasound diagnostics. The precursors were Dégoulange with his project Hippocrate [12]. There were important projects: OTELO, ESTELE and TER. They differ in number of degrees of freedom (3–6), the way of a remote manipulation and the presence (or not) of haptic feedback. The ReMeDi project is aimed to develop a robotic system for remote medical diagnosis. The direct examination by the physician is split into two physical locations. On the patient site a dedicated robotic system equipped with a manipulator with the USG transducer and a force sensor, and an active 3D vision system is placed. The robot is supervised by an assistant assuring the proper and safe robot operation. The diagnostician site comprises a control console with a vision system, haptic input device, an USG control panel and a number of foot switches [43]. The procedures have been described in [42]. The XACT ACE™ Robotic System (cleared by the FDA in 2020) is used for CT-guided percutaneous procedures including ablations, biopsies and targeted drug delivery. European project MURAB (MRI and Ultrasound Robotic Assisted Biopsy) improved a minimally invasive technique for early cancer diagnoses.

Endoscopy-Bot. Video capsule endoscopy uses a miniature camera in the shape of an ingested capsule for direct visual and diagnostic evaluation of gastrointestinal diseases since 1981. The pioneers in this field were Gavriel Iddan, an engineer from the Israeli Ministry of Defense, who collaborated with the gastroenterologist Eitan Scapa. PillCam SB3 (Given Imaging) weighs less than 4 g, measures 11–26 mm, has a variable frame rate–up to 6fps when traversing fast areas like the duodenum, and up to 2fps when stationary or moves slowly. It also has a blood detector for bleeding sites identification. Endocapsules have a battery life of approximately to 12 h. Olympus Endocapsule, Micro Cam (IntroMedic) and OMOM (Jinshan Science & Technology) are other capsule endoscopes used all over the world [4]. A team of scientists and doctors from the University of Nebraska for natural orifice translumenal endoscopic surgery (NOTES) techniques proposed to introduce a team of small robots (15 mm in diameter and 70–75 mm long) into the abdominal cavity through openings from the gastrointestinal tract. The robots performed specialized functions: visualization, biopsy, manipulation or lighting robots [44].

Aesthetic medicine. Thanks to robots, we are more efficient and beautiful. There are many robots that are used for physical training and improvement of sport performance. In the field of human image, attempts to develop robots for dentistry and plastic surgery are important. However, the robot assisted surgical hair restoration procedure is gaining the greatest popularity, now. The operation–hair transplantation–using the ARTAS iX robot, equipped with an advanced 3-camera stereoscopic vision system with a resolution of 44 microns and a 7-axis arm, is based on an artificial intelligence algorithm. ARTAS iX is the only device in the world that simultaneously identifies and collects the best quality hair follicles, prepares recipient sites and implants hair [6].

Automatic pharmacies, robots with AI-enabled medicine identifier software reduce the time it takes to identify, match, and distribute medicine to patients in hospitals.

– Aesynt offers drug management solutions for all forms of drugs, including automation and sterile mixing (in over 80 units). It solves security problems associated with manual compounding, ensures connectivity across the company. In 2016, the company was acquired by Omnicell for USD 275 million.
– Innovation Associates offers technology for pharmacies including: ROBOTx, an automatic drug dosing system with a built in dosing capacity range [37].
– There is a growing Polish rival with similar functions and experience in two hospital installations—UniDoseOne (Fig. 2) [46].

Drones. Currently, following the development of technology, many projects have been created regarding the use of robots in medical services. Drone deliveries of first aid kits, emergency medication and blood samples.

Example: EHang–the world's first autonomous drone capable of transporting a human and donated organ;

ZipLine–trial transporting blood for transfusion in Rwanda.

TU Delft Ambulance Drone is a prototype in which a cardiac defibrillator, two-way radio communication and video are integrated. In turn, Alphabet (Google) is developing the Wing project for disaster relief–to provide food, water and medical materials (tested in both the US and Australia) and Vayu drones, which are now

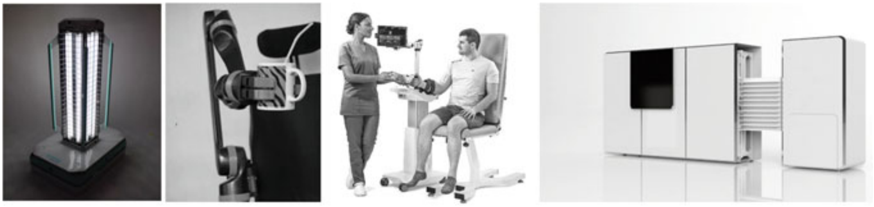


Fig. 2 The DiSiRt disinfection by ACCREA (ACCREA, https://accrea.com/pl/disirt-robot-do-dezynfekcji), ARIA—Wheelchair; ultra-light, safe, and very compact which provides a high level of convenience and independence, ARIA v2: Kinematics 6 DoF, weight 5 kg, payload 1 kg, reach 900 mm, repeatability 1 mm, lightweight carbon fiber structure, joint torque sensors (optional), End Effector 6DoF sensor (optional) (https://accrea.com/aria); Luna EMG uses reactive electromyography to train the sensorimotor cortex by EZGOTECH (Egzotech, https://egzotech.com) and UniDoseOne–robotic pharmacy that prepares doses of medicines in individual sets for hospital patients https://unitdoseone.com

successfully used to transport medical samples–provides faecal and blood samples to the country's central laboratory for testing in Madagascar [14].

Drones are successfully used in rescue operations, most often to supervise the area (e.g., warn against fires) and search for missing people. But more complex systems are also created, such as the concept of creating a Drone Road Rescue System, based on the use of specialized drones to monitor roads and identify road incidents, deliver the necessary medications and equipment to the scene of accidents and transport victims to the hospital.

Service. Service robots set up patient rooms, track supplies and file purchase orders, restock medical supply cabinets, and transport bed linens to and from laundry facilities. Service robots help with cleaning and disinfection. These robots may use ultraviolet (UV) light, hydrogen peroxide vapors, or air filtration to help reduce infection and to sanitize reachable places in a uniform way (example: Akara, Accrea–Fig. 1, [1]). The author of this chapter has been leading for several years as an innovative coach in the European project DIH Healthcare Robotics HERO, consulting for the Polish ACCREA and the Irish company AKARA, which prepares new models of independent disinfection robots. Autonomous mobile robots (AMRs), include service robots and social robots, simplify routine tasks, reduce the physical demands on human workers, and ensure more consistent processes.

Cleaning and disinfection AMRs enable hospital rooms to be sanitized and ready for in-coming patients. Equipped with light detection and ranging (LiDAR) systems, visual compute, or mapping capabilities, AMRs can self-navigate to patient. AMR controlled by a remote specialist allows to tele- consultation performing (so important during pandemic condition) [23].

UV-C light is a proven technology to disinfect environments, commonly [5] used in autonomous AMR robots such ACCREA (Fig. 2) and AKARA. Next example of AMR robots:

– Aethon TUG - common in hospitals (over 140 hospitals in the US) robot provides automated, cost-effective delivery of medicines, laboratory samples or meals to the patient (carrying capacity over 450 kg) [3],
– Xenex is hygiene robot (in 400 hospitals in the US) allows for UV disinfection,
– disinfection robot M2 (Keenon Robotics Co) with multiple disinfection modes: product weight 60 kg, disinfection volume 10 l, UV radiation intensity (measure at 1 m) 200 mikroW/cm^2 and maximum speed 0,8 m/s [26].

The robot can also be dressed for supporting the human locomotor system. Exoskeletons are robotic structures that can help people move and lift huge loads. They can also allow paralyzed people to walk again.

Exoskeletons are also present on the industrial market. Examples of commercial external skeletons are Innophys' Muscle Suit, Activelink's Powerloader Ninja, Cyberdyne's HAL for Labor Support RB3D's HERCULE, Esko Bionics' Esko Vest, Sarcos Robotics' Guardian XO and Noese's Chairless Chair. Unlike powered outer skeletons, non-powered or "passive" exoskeletons increase strength and provide stability by combining a human-controlled bending/stretching and locking mechanism. The maintenance-free versions of Exos for commercial and industrial use include the MAX Exoskeleton Suite from suitX, the Ekso Bionics work vest, StrongArm Technologies FLx ErgoSkeleton technology, Laevo Laevo and Fortis Lockheed Martin's [37].

Example [2, 37]

- Hocoma therapeutic products include medical devices used for automatic training of the neurological treadmill and exoskeleton for the rehabilitation of upper limbs after a stroke or traumatic brain injury. Armeo Spring–a stationary orthosis by the Swiss company Hocoma, used for the neurorehabilitation of the upper limb. By providing complete support to the upper limb so as to minimize the burden on the patient caused by its weight alone, Armeo Spring allows patients to use their own, even residual motor functions, to repeatedly perform upper limb movements.
- Rewalk Robotics offers the ReWalk exoskeleton (received first FDA in US), enabling people with lower limb paralysis to walk again (used by over 1,000 people today around the world.) ReWalk–is a battery-powered bionic orthosis created by the Israeli company ReWalk Robotics. It is powered by four electric motors located at the level of the hip and knee joints, controlled by a computer. The patient initiates the first step by leaning his upper body forward, then controls the movement with subtle changes within his center of gravity. The repetitive sequence of the above-mentioned movements creates an imitation of a natural gait.
- Reha Technology (Swiss company) manufactures robotic gait trainers for the rehabilitation of neurological movement disorders. The G-EO System trainer allows the patient to move his legs through a series of normal walking patterns.
- Keeogo–a product of the Canadian company B-Temia Inc. is an exoskeleton worn on a limb or lower limbs, intended for patients with multiple sclerosis or neurological and musculoskeletal disorders on the basis of other diseases. The main purpose of the device is to support the patient in everyday activities so that he can function independently at home and not be excluded from social life. Unlike the Ekso and ReWalk products, the device can only be used by people who walk independently and only need to correct any motor deficits that make it difficult to move.
- Marsi Active Knee (MAK) is the first robotic knee brace, manufactured by the Spanish company Marsi Bionics, with the function of flexible activation of the knee joint that imitates the activity of the muscles that move the joint in natural conditions. Thanks to the extensive equipment with sensors, it analyzes the biomechanics of the patient's gait and improves it.
- The Hybrid Assistive Limb (HAL) is an electrically powered exoskeleton suit developed by the Japanese Tsukuba University and Cyberdyne. The device offers the possibility of monitoring muscle contractions using EMG electrodes placed on the flexors and extensors of the lower limbs. This allows the patient to make movements according to the will of the patient, even if the motor and sensory nerve pathways are damaged. The impulse from the motor cortex is received and amplified by the HAL system, then even trace nerve impulses from the muscles are amplified and conducted to the sensory cortex of the brain.

Telepresence robots. In many situations in a hospital, it is beneficial (especially during a pandemic) to send a communication robot instead of being close to the patient. Example of telepresence robots (remote controlled) (a) Intouch Health (b) VGo Communications (c) Kuka with the LBR iisy robot–telepresence. They are simple and effective.

Social robots. An important group of robots are social robots, they often have mobile functions and are connected with medical or family caregivers. The most important feature of social robots is presenting the possibility of communication with the person being cared for. They are supposed to be good companions of lonely people and take care of their safety. In developed countries, alienation and loneliness cause serious problems, including health ones. Robots in various forms–humanoid or in the form of animals or other active objects–help in communication with the world (as a security service, entertainment or education), as companions for elderly people and others without the necessary social support.

Example [37]

- Luvozo PBC Established in 2013, the company focuses on developing solutions to improve the quality of life of older people and people with disabilities. In 2015, the company began testing–Sam, a robotic concierge–in a leading senior community in the Washington D.C. area.
- Honda Robotics n 1986, Honda engineers began a two-legged robot project. After 20 years, ASIMO was presented, a full humanoid robot (with excellent mobility) and communication (verbal) skills, as well as facial recognition. ASIMO was planned to provide assistance to the elderly or those with limited mobility (bed or wheelchair bound). In 2018, the project was stopped–after the successes of ATLAS robots, the American company Boston Dynamics (co-financed by Google X in its time, then sold to SoftBank).
- Walking robots delivered by Agility Robotics and Boston Dynamics. Announced in 2019, after 26 years SpotMini is the first robot to enter the Boston Dynamics market.
- AIST A leading Japanese industrial automation pioneer, AIST, has developed PARO, an advanced interactive robot covered in fur (seal-like). This therapeutic robot reduces the stress factor experienced by both patients and their caregivers.
- The most beautiful is definitely the Sophia robot. In the midst of the pandemic, I had the opportunity to talk to David Hanson about what Sophia should offer for healthcare now. May be Hanson's Grace robot–nurse will revolutionize healthcare. The robot Sophia (Hanson Robotics) is granted Saudi Arabian citizenship. It becomes the first robot to be recognized with a gender identity and nationality. Sophia is a human-crafted science fiction character and a platform for advanced robotics and AI research [18].

Robotic psychology (i.e., robopsychology) is a new field that investigate human–robot interactions.

The use of physical robots or intelligent virtual agents as therapeutic tools can be traced to ELIZA [50], one of the first programs capable of "understanding" and generating natural language. The studies presented in [10] show the possible roles of robotic agents in psychotherapy (i.e., Robo-Therapist, Robo-Mediator, and Robo-Assistant) in the following categories:

CBT Virtual Agent for Depressive Patients, Psychological Pills Virtual Agent—Apps PsyPills, Robo-RETMAN (an acronym of Rational Emotive Behavior Therapy professional), Robo-Mediator in the Treatment of ASD Children, Robo-Assistant for Enhancing Therapist Interventions and Robo-Assistant for Psychological Assessment.

Many studies investigate the use of Robo-Assistants and human–robot interaction in elderly people. Especially in comparison to animal assisted therapy which might be helpful for some psychological conditions, but they have several major limitations such hygiene restrictions and animals' unpredictable reactions. The most famous "Robo-Animals" is Paro [39], an animated seal robot with multiple sensors, capable of learning contingencies between its behavior and the reactions of the users (such as stroking or beating) and increasing the frequency of the preferred behavior [9].

Mobile robots. Within the medical environment, transportation is an expensive and time consuming. Mobile robots (autonomously guided vehicles AGV), have been used in such diverse fields as education, commercial cleaning, surveillance, aerospace, defense and manufacturing.

Example [28]

- HelpMate robots (first hospital installation in 1991 and are currently used as cost effective robot in about 70 hospitals) delivers pharmaceuticals, meals, lab results and supplies throughout the hospital. Recently, Pyxis Corporation, a leader in hospital supply inventory management, acquired the assets and robot technology from HelpMate Robotics Inc.
- RoboCart (California Computer Research Inc.) Lighter and smaller, cheaper than HelpMate offers an alternative for intra-laboratory deliveries. RoboCart is the first robot to address the needs of laboratories with narrow corridors.
- Transcar (Swisslog Translogic, Denver) is a computer-controlled vehicle that follows a predetermined path for distribution of materials. Its guidance sensors follow a thin stainless-steel band of tape adhered to the floor surface. Using infrared light, computer control allows Transcar to signal doors to open, operate elevators and interface with multiple vehicles on a common pathway.
- The CyberGuard Robotic Security System (Cybermotion) alerte to the early stages of a fire using flame, smoke and gas detectors. This self-guided autonomous security vehicles uses sonar, microwaves and infrared to detect the presence of people in secure areas. Since 1984, Cybermotion has marketed over 30 robotic security systems.

Mobile medical robots use various techniques to move in space or in the patient's body to reach the place of action. Especially mini, micro and nanorobots are characterized by great ingenuity in accessing the interior of the body (natural and artificial openings are used) and moving inside it (often using the body's natural tracts and transport mechanisms).

Mini robots diagnose the gastrointestinal tract, microrobots will travel within the blood circulation system acting as morphotic components of the blood (e.g. erythrocytes …), nanorobots will repair rebellious cells or repair errors at the level of genes. Among mobile medical robots, it is worth paying attention to the concept of robots performing tasks directly inside the body. A very interesting example is a macroscopic robot–a rover that moves on the surface of the heart…

HeartLander is a miniature mobile robot for minimally invasive therapy on the surface of the beating heart. Under physician control, the robot: enters the chest through an incision below the sternum, adheres to the surface of the heart, travels to the desired location, and administers the therapy. HeartLander uses suction to adhere to the epicardial (outer) surface of the heart (pneumatic system). The inchworm-like locomotion of the crawling robot is generated by alternating the suction to the body sections, while changing the distance be-tween them [19].

Medical robots can be purely mechatronic or hybrid (using biological elements).

Microrobots will travel within the blood circulation system acting as morphotic components of the blood (e.g., bacteria-driven micro swimmers using red blood cells (RBCs; erythrocytes) as autologous cargo carriers for active and guided drug delivery [53].

Example [42]

- Powering mechanism for micro/nanorobots. (a) Magnetically propelled microrobot based on rotating microcoil. (b) Ultrasound propelled microrobot powered by cavitating microbubble. (c) Chemically propelled motor based on zinc microtube, the microrobot converts gastric fluid into gas bubbles that generate propulsion trust. (d) Biohybrid microrobot based on the integration of a sperm with a synthetic structure.
- Microrobot as carrier of individual cells. (a) Chemically powered micro-robot transporting macrophage attached via electrostatic interactions. (b) Bio-hybrid microrobot transporting red blood cell. (c) Magnetically powered micro-helix transporting sperm cell.
- Microrobot based biopsy and sampling. (a) Star-shaped griper collecting tissue. (b) Star gripper collecting red blood cells. (c) Motile microtrap collecting pathogens.
- Micro/nanorobots as mechanical probes. (a) Nanorobot of measuring intracellular mechanical properties. (b) Nanorobot for biofluid rheology. (c) Nanorobot for measuring mechanical forces of a macrophage.

Microbot Medical American company producing micro-robotic technology devices offer ViRob, a crawling, small micro-robot can move in blood vessels, digestive and respiratory systems and TipCat needlelike, self-propelled, flexible and semi-disposable endoscope. Liberty is the surgical robot designed to streamline endovascular procedures [34].

Trol et al. demonstrated novel intravitreal delivery microvehicles, slippery micropropellers, that can be actively propelled through the vitreous humor to reach the retina. The propulsion is achieved by helical magnetic micropropellers that have a liquid layer coating to minimize adhesion to the surrounding biopolymeric network. The submicrometer diameter of the propellers enables the penetration of the biopolymeric network and the propulsion through the porcine vitreous body of the eye over centimeter distances [45].

Researches from University Pennsylvania and Cornell working on "smart" robots that incorporate sensors, clocks and self-controllers (can actually be injected into the body via a hypodermic needle). Each robot's body of this delivery microvehicles consists of the electronic control components and solar cells. When a laser is shined onto the solar cells, the resulting electrical current is alternately applied back and forth between the front and rear legs (made from a 100-atom-thick material, platinum and titanium or graphene layer). The application of that current causes the platinum to expand while the titanium remains rigid, thus bending each leg [9].

Scientists from the Massachusetts Institute of Technology answered the question–how can it get into the brain? The "magnetically controlled robotic thread" has a core made of a nickel-titanium alloy covered with a layer of rubber with the addition of magnetic particles with a layer of soft hydrogel. This allows the robot to be safely placed in a blood vessel and controlled by moving a powerful magnet over the patient's body [40].

This review of medical robotics is of course not complete, but it shows the range of problems and technical solutions used to perform medical services in various variants, macro and microscopic scales of operation, various forms of reaching the area of action inside the patient (access) and performing work.

This review of medical robotics is of course not complete, but it shows the range of problems and technical solutions used to perform medical services in various variants, macro and microscopic scales of operation, various forms of reaching the area of action inside the patient (access) and performing work.

The risk analysis of each of these robots reveals a number of problems to be solved. For example, the use of nanobots requires addressing the toxicity problem and finding a way for them to leave the patient's body once the task is completed. If we supervise or only observe one robot in the hospital or inside the patient's body, it is not a problem for us. But if we have a set of robots performing various logistic orders and various work for the hospital and patients, we must take into account the synergistic control of the set of robots. Modeling the future of healthcare requires introduce smart hospitals, apply AI and analytics in a connected infrastructure, to turn data from devices, robots and workflows into actionable data.

We must start managing an army of robots by defining the strategy, then tactics, and finally defining the appropriate tasks for the appropriate robots. This is a version of the Internet of Things–**Internet of Robots**.

4 Robin Heart

From 2000 several models of the robot have been developed by Robin Heart Team in Foundation of Cardiac Surgery Development (FRK), differing in control systems and mechanical construction:

- Spherical model (2001).
- Robin Heart 0 (2002).
- Robin Heart 1 (2003).
- Robin Heart 2 (2003),
- Robin Heart Vision (2008),
- Robin Heart Junior (2009),
- Robin Heart mc2 and mechatronic tools Robin Heart Uni System (2009),
- Robin Heart PortVisionAble (PVA) model 0,1,2,3 (2012–2015),
- Robin Heart Pelican (2013),
- Tele-Robin (2015).
- Robin 8 (2016) (Fig. 1) [35, 38].

The Robin Heart 0, Robin Heart 1, Robin Heart mc2 have an independent base but the Robin Heart 2, Vision, PVA, Pelican is fixed to the operating table. Models Robin Heart Vision, PVA, Pelican is the one arm robot for endoscope manipulation during laparoscopic video surgery. The biggest Robin, Robin Heart mc2 can replace work of three surgeons at the operating table. This modular robot can be set for various operations in a selected way. Tele Robin Heart robot solution extend of the tool platform concept–on the one arm is mounted fully functional surgical robot with two tools and endoscope. This allows the robot to be easily positioned in positions that are impossible to achieve with the classical design of surgical robots.

The Robin Heart 8 is the only robot that is not a spherical robot based on a parallelogram. This robot was developed for tissue simulator for robots and mechatronic tools tester. Currently, due to its kinematic properties–full freedom of choice of the trajectory–is considered as a robot potentially used in ophthalmology.

Robin Heart **PV**A (license sold to a commercial company in 2019) will replace a human assistant who usually holds the endoscope to enable the observation (**V**ision) of the operative field of laparoscopic instruments, is easy to use and install, and is **P**ortable–carried in a suitcase.

Design of this original Polish robot family started by determining the tool-tissue reaction (mechanical characteristic, the forces for specific operations, dynamic analysis of the work of a tool) and the person-tool/man–machine contact (kinematic analysis of the surgeon's motion).

Ergonomic optimization was considered very important–both in terms of work space and manipulation, control of the work of tools and monitoring and the entire information sphere for the operator. A innovative system of education and operation planning was introduced, including both the use of real model workstations and those using virtual space technology.

As a result, another model of the Robin Heart Shell console is equipped with a consulting program that makes it possible to obtain all patients diagnostic information during the operation, as well as elements of operation planning on the screen. The introduction of feedback from field operations; the implementation of force-feedback loop, and spatial visualization improve working conditions of surgeon. The robot should provide all the information needed not only to the surgeon to correct tool orientation (based on vision sense), but also to take appropriate course of action in this area (selection and tool task).

Robin Heart system experiments using testing apparatus and next, on ani-mals, and also pioneering teleoperations in different conditions and at different distances allowed verifying robots before planned phase of preparing serial production and clinical initiating [37, 38].

Force feedback solution. The surgeon uses the telemanipulator because of the offered increased precision, safety and quality of work. The robot also gives a unique opportunity to control and standardize work. In order to improve the safety of the endoscopic intervention the surgeon must get on-line information about the physical parameters (force measured during interaction endo-scopic tools with the tissue) during operation. The sensor built inside the gripper can measure the strength of which the laparoscope holds the surgical tool. The sensor placed on the tip can provide information about the hardness and surface roughness of the tissue the endoscopic tools touches–facilitates steering and tissue identification and the inside sensors provide appropriate information about the force the grasper holds a needle, surgical sutures or tissue. Due to technical reasons, it is difficult to install sensors directly in the grasper, so robot designers try to solve the problem indirectly in different ways. However, none of the robots in use today, available on the market, have achieved this goal–functionally useful force information in terms of scale: from touching the tool to the bone to tying a knot with a thread.

The effectiveness and safety of the operation on the distance depends on the delay of in-formation coming to the operator from the field of operation. At FRK, advanced work is underway to reduce these threats. The process of adapting the robot to these conditions requires the introduction of several innovations. First, part of the decision-making rights must be transferred to the robot, which in the event of too much delay or loss of Master–Slave contact decides to (1) end the ac-tion movement, or (2) continue the task to the end. This means that the relevant surgery procedures and tasks must be recognizable to the robot in action and must have the appropriate safety status to allow to make by robot the right decision. Methods for displaying the current position of the tool on the operator’s monitor despite the image lag are also being developed. To provide the robot operator with the most up-to-date, least delayed signals, all information paths are used, both related to image transmission and signal paths from motors and encoders located in the axes (robot joints), as well as force measurement transmissions from tool sensors. When the delay is greater than the predetermined limit, e.g., 200 or 500 ms, the appropriate AI-based monitoring program, based on the analysis of the data held, creates the appropriate information for the operator where the tool tip currently touches an obstacle, e.g., tissue of heart or bone (e.g., in a monitor image showing a delayed image displayed, a character indicating the current position of the tool appears).

Teleoperation experience–mobility of action. The first model of teleoperation was performed in December 2010 between two cities Zabrze (FRK) and Katowice (Medical University of Silesia, Center of Medical Experiments on animal) successfully. Next–a remote control experiment of the Robin Heart Pelikan robot at the Guido Mine in Zabrze. Authors [27] describe an experiment undertaken in January 2022–a demonstration of the remote control of the Robin Heart Pelican robot using the public Internet. Several different methods of remote control from Zabrze, a robot located in exhibition halls ArabHealth2022 in Dubai, were developed, applied and tested. Based on the measurements, it was assessed which of the methods of signal and image transmission causes the smallest delay. The robot remote control laboratory tests are carried out, simulating various communication hazards and signal and image transmission delays in order to introduce an effective operator warning system and create a safe teleoperation system. The delay problem is solved using computational algorithms and signal analysis based on which, in the event of an image delay, a colored line appears on the operator monitor in the console indicating the current end of the tool.

Robots' mobility–transport. Certainly, the accompanying devices supporting the surgeon should be easily transported by the surgeon. Robin Heart Team designed and made a robot to control the endoscope position during mini-invasive surgery. The single-armed robot Robin Heart PortVisionAble (a portable vision robot) has passed all tests, and the license for its commercialization was sold by FRK in 2019. The same team also designed and made a prototype of the robot, which is even lighter (we tested the construction with a mechanically internal tensed structure elements). The Robin Heart Pelican (Fig. 1) is a laparoscopic tracking robot that has 4 degrees of freedom (DOF 4), which allows it to navigate the camera inside the animal's abdominal cavity [38]. This endoscopic robot is intended for veterinary use. The robot is easy to transport, so it was at numerous shows, e.g., in Sicily (Erice), on the "Dar Młodzieży" sailing ship in London or at the fair in Dubai. Robots replacing the doctor who holds an endoscope in his hand during the operation to show the surgeon on the screen where the tools work inside the patient's body should meet the following criteria:

(1) lightness (to be mounted at the operating table),
(2) telemanipulation (they should be easily moved–using joystick or voice controlled by the surgeon–and placed stably in a place where we get good visibility of the current operating field) and,
(3) remote counseling (should be adapted to remote control by an advisor at any distance so that he can comment on and pass on important information for the treatment being carried out–education, "teleportation" of tutor).

Mobility? Should a surgical robot be mobile? Of course, it should be available wherever it is needed, in a city, village, at sea or in space. From the very beginning, as part of DARPA projects, the Americans tested the possibility of teleoperation on a car platform on the battlefield, and NASA developed plans for the use of the robot in space during the star wars. Robin Heart Team conducts also research & conceptual studies with the installation of a robot on board a sea or spacecraft and work on a rescue robot that can fulfill the basic functions of caring for the injured person - e.g., as part of the equipment of lifeboats.

5 Mobility in Medicine

That is, about relations in space and time of the medical services performance; when the patient comes to the doctor, and when the doctor comes to the patient. Where and how does the recipient meet with the service provider.

For example: a hospital is first and foremost a team. Each team member is equipped with the appropriate tools. As the instruments and the team are, of course, essential for effective operation, important is continuous training, improvement, cooperation, synergy of employees and technology, human intelligence and algorithms. If people and act exist in a confined space, there is also choreography problem, so that everyone is in the right place and at the right time for realizing a purpose. Important is an architecture, its limitations and advantages. For creation a map with set of goals, must be a strategy, tactic, tasks, i.e., a field of action, in which every point in space–time is appropriate competences and information. The implementation will not be possible without the flow of energy and materials. Patient safety requires caution, i.e., standards of human conduct and used equipment, drugs, and information. Personnel and equipment resources, competence and agility, and economic security are at the heart of all hospital activities.

Thera are different Human Resource management systems; in a nutshell: either a lot of properly equipped specialists or fewer doctors, but an appropriate division of competences into the specialties of middle-level technicians/medics. In order to maintain the quality of the services provided, the fewer professionals there are, the more professional equipment will help people and improve the quality of the service.

Mobile robots, in my understanding, for the purposes of these considerations, they are:

- robots, transport vehicles: for lifting and transporting wounded soldiers, miners or tourists in mountain or marsh conditions, for rescue operations at sea or in space, autonomous wheelchairs (often equipped with an executive arm for opening doors or carrying items or feeding, Fig. 2), drones for transporting drugs or blood, organs for transplants,
- wearable mechanism for rehabilitation (exoskeletons),
- human-carried robots; from the point of view of supervision, because they move using the human locomotive system (all implants, artificial organs e.g., heart assist pumps),
- robots used by humans for independent movement and movements: electromechanical artificial arm or leg (related directly to the patient's ability to move).

A medical service consists of competence and work. And they have to meet at the point of space–time where there is a need to apply them–with the pa-tient. So we have options: either the patient or the service is mobile. In an ur-gent or emergency mode, of course, we should reach the right place with the service in the required time. If there is equipment or medicines on site, it is enough to have mobile competences, e.g., advice and classic telemedicine. A competent person, a medical specialist collects information from the patient and the other way around–passes it on to the patient or caregivers. The development of artificial intelligence methods in most cases can already help. This means that in the near future this type of relationship will be replaced by a ma-chine–a bot. It can be present in the smartphone or in the software delivered with the rescue device or medications (we prescribe the drug along with the smartphone app). But when action is needed, mechanical action, treatment, for example in accidents, strokes; the work performed by a medical team directly on the patient or teleoperation (i.e., a telemanipulator controlled by a competent doctor) is need. If we manage to combine decision-making and executive competences in a robot (AI), we are dealing with a medical autonomous robot. Autonomous vehicles for transporting mail, medicines or as ambulances operated by users, are already being tested today. If we also give them executive functions, we will have fully functional mobile rescue robots. The challenge is how to implement a rescue service in a mine, on a stormy sea or in space.

6 Bio-technical, Hybrid Robot

Direct robot control by the brain. Kevin Warwick, the first human to experience being a cyborg, tested also the direct connection between humans and machines–robots with biological brains. Perhaps the human body, with the support of such technology, can overcome further limitations and achieve more advanced goals. The emergence of a new species of people with intelligence enriched by AI or the ability to act, perform work (e.g., resistance to overload of workers equipped with an exoskeleton) creates a number of new challenges. Warwick tested the ability to control of mini vehicles by the biological neural network (self-learning) [48, 49].

Neurons can be cultured under laboratory conditions on an array of non-invasive electrodes. A robot body can move around in a defined area under the control of such a network/brain. Mobile robots or functional arms can be moved by people with disabilities using human brain signals, but we may also be dealing with really biological brains in a robot body soon.

“When considering robots with biological brains, clearly such an approach allows for ‘complete body engineering’ in which brain size, body size, power, communications and other abilities are optimized for the requirements in hand. Maybe this technique will ultimately open up a future route for human development.. ” [49].

Non-invasive neuroimaging improves the continuous tracking of neurons to steer robots. Brain-computer interfaces (BCIs) utilizing signals acquired with intracortical implants have achieved successful high-dimensional robotic device control useful for completing daily tasks [15].

BCI projects are supported by Chinese, Korean and Russian governments. The Mexican DARPA has the greatest experience in this field. "By creating a more accessible brain-machine interface that doesn't require surgery to use, DARPA could deliver tools that allow mission commanders to remain meaningfully involved in dynamic operations that unfold at rapid speed" [11].

The potential of such a solution has been demonstrated for people using wheelchairs or prostheses, robots to assist with eating and a mind-guided drone.

7 Medical Robots' Market

In the field of robotics, the greatest advances and breakthroughs today are presented in the fields of technology and business rather than science and research. In order to form an opinion on the direction of development and opportunities for further medical applications, it is worth following the reports presented in the Internet media full of current and important facts.

Medical robots are getting better and better business. Surgical robots are particularly advantageous–because they are expensive (unit cost) and require the delivery of materials for each operation (disposable materials). For example, robotic system allows surgeons to measure, plan, and perform knee or spine surgery, which is personalized to the patient's individual anatomy (using directly diagnostic data) in theater. The market is segmented into instrument and accessories, robotic systems, services, and consumables. Since millions of procedures and operations are performed annually, this must generate profits. However, so far only 2% of mini-invasive operations are performed with the help of telemanipulators, because there are the following Market Restraints: High Cost of Surgical Robotic Systems and their Associated Procedures, Shortage of Skilled Professionals and Restrictive Reimbursement Landscape. High cost of robots, along with lack of skilled operators are restraining the growth of the market.

The chances of overcoming these weaknesses lie in: Development of Low-Cost Surgical Robotic Systems, Implementation of Artificial Intelligence (AI) in Surgical Robots and Development of Surgical Simulators for the Training of Professionals. After the end of the term of the basic patents relating to the surgical robot, companies competing against the monopoly in the field of soft tissue surgery–Intuitive Surgical, producing the da Vinci robot, appeared. It is therefore worth following the offer of companies such as: Asensus Surgical, Inc., Avatera Medical GmbH, CMR Surgical Limited, Corin Group, Curexo, Inc., Globus Medical, Inc., Intuitive Surgical, Inc., Johnson & Johnson, Medicaroid Corporation, Medrobotics Corporation, Medtronic plc, Meerecompany Inc. Monteris Medical Corporation, Neocis, Inc., PROCEPT BioRobotics Corporation, Renishaw plc, Siemens Healthineers AG, Smith & Nephew plc, Stereotaxis, Inc., Stryker Corporation, THINK Surgical, Inc., Titan Medical, Inc., Venus Concept, Inc., Vicarious Surgical, Inc., Virtual Incision Corporation and Zimmer Biomet Holdings, Inc.

According to Global Surgical Robotics Market 2021–2031[17], The global surgical robotics market is anticipated to grow with a robust CAGR of 10.2% during the forecast period, 2021–2031. The market for surgical robotics generated $5.46 billion in 2020, in terms of value and is estimated to reach $16.77 billion by 2031.

The range of robots used in medicine also includes diagnostic and rehabilitation robotics and other robots supporting the work of doctors, nurses, cleaners and cooks in the hospital. So, if we add the surgical robots discussed above, we get: "The Medical Robotics Market size was valued at USD 10.88 billion in 2021 and is predicted to reach USD 44.45 billion by 2030 with a CAGR of 17.1% from 2022 to 2030. Medical robots are professional robots that assist surgeons to perform dexterous surgeries, along with streamlining clinical workflow and hospital logistics. They help in assuring complete care of the patient after surgery and ensure productive processes in clinical settings to safeguard environment both for patients and health workers. Also, medical robots are utilized in several applications ranging from pharmaceuticals application to laparoscopy."

From the KEY MARKET SEGMENTS catalog, let's mention only: Soft, Hard Tissue Surgeries, Non-invasive Radio surgery Robotic Systems, Logistics/Handling Robotic Systems, Diagnostic, Rehabilitation, Therapeutic, Assistive and Exoskeleton Robotic Systems.

Medical Robotics Market can be divided, by Application, into: General, Gynecological, Urological, Oncological, Orthopedic, Transoral Surgery, Laparoscopy, Pharmacy Applications, Physical Rehabilitation and External Beam Radiation Therapy. Of course, you should pay attention to where the robots are mounted or in what space they move.

Medical Robotics Market–by End Users: Hospitals, Inpatient (yes, the robot can be swallowed or carried in you, such as an artificial heart), Outpatient, Home Care Settings, Ambulatory Surgery Centers, Rehabilitation Centers [33].

Selected major players in the invasive medical robotics market in the world and their systems: Intuitive Surgical Da Vinci General Surgery, CMR Surgical Versius, Medicaroid Hinotori, Avatera Medical Avatera (in development), Medtronic Hugo, J & J/Verily (Google) Verb Surgical system under development), Orthopedic surgery: Medtronic Mazor X, Globus Medical Excelsius GPS (spine), Stryker Mako (spine, knees), Zimmer Biomet Rosa (knees), ROSA One (brain tumor), Smith & Nephew Navio, J & J/Auris Health Monarch (lung biopsy), Intuitive Surgical Ion (lung biopsy), Siemens Healthineers Cor-Path (cardiology).

Summarizing briefly the analysis of the medical robotics market, attention should be paid to the globalizing trends. There are two reasons:

1. expected profits (modeling on the success of Intuitive Surgery, which monopolized the soft tissue surgical robotics market for 20 years).
2. high costs of developing robots and introducing them to the market and a long time "from the idea to the first profit" (20 years on average).

Therefore, technology companies which have gained financial surpluses in the IT, computer science and commerce on the Internet market, such as Google or Amazon, take over and enrich startups and companies producing robots.

Two examples:

- Amazon acquired the mobile robot company Kiva in 2012 for $775 million. Kiva's automated guided vehicles (AGVs) use computerized barcode stickers on the floor to navigate Amazon's warehouses. Amazon Robotics recently announced its first autonomous mobile robot Proteus. Amazon quietly acquired Dispatch to create Scout, which it announced in early 2019. The six-wheeled sidewalk delivery robots are fully electric and travels at a walking pace. Amazon plans to acquire iRobot for $1.7 billion, in 2022. Amazon announced its $1 billion Amazon Industrial Innovation Fund in April 2022. The fund's first round of investments included Agility Robotics, a company developing a bipedal walking robot, BionicHIVE, a company developing an autonomous robotic solution that can adapt to existing infrastructure in warehouses, and Mantis Robotics, a company developing a tactile robotic arm [22].
- Boston Dynamics, an MIT spin-off, is best known for its innovative robots, including BigDog, Atlas, Stretch, and Spot (last two are available for commercial purchase). Hyundai Motor Group (Hyundai) acquired Boston Dynamics in 2021 from Softbank for about $880 million. Hyundai became the third owner of Boston Dynamics in seven years. It was acquired by Google in 2013 and sold to Softbank Group in 2017. Hyundai announced the launch of the Boston Dynamics AI Institute in 2022 ($400 million goes to make fundamental advances in artificial intelligence (AI), robotics and intelligent machines) [20].

8 Product Life Cycle

From the manufacturer's point of view, the product life cycle begins with its development (conceptual, research and development), the culmination occurs when the product is launched, the cycle ends with the product being withdrawn from the market and the introduction of a newer version or a subsequent product. The time to develop a new solution in medical robotics can be estimated at approximately 4–5 years on average. On average, testing takes about 2–3 years, and the certification required to launch a product takes about 1–2 years. From the user's perspective, the life cycle of a product is related on the one hand to its "lifetime", and on the other hand to the aging of the technologies used to build it. The life cycle of products in the field of medical robotics can be different and depends on the particular robot or its application. Medical robots can be used for as little as 6 to 10 years. Barriers for entrepreneurs. The first barrier identified is the cost-effectiveness of ongoing work on developing new products. Another–related to robot certification and admission to the market of medical products [37].

Standards and definitions

Medical robots are robots and medical products. Proper preparation of such a product is a big challenge because of the need to comply with many laws, regulations and standards. According to ISO (the International Organization for Standardization) [24] medical robots belong to specialist service robots.

Service robot is it robot that performs useful tasks for humans or equipment excluding industrial automation applications (SOURCE: ISO 8373:2012, 2.10) but personal care robot–service robot that performs actions contributing directly towards improvement in the quality of life of humans, excluding medical applications (SOURCE: ISO 13482:2014, 3.13).

ISO 13482 clarifies which robots fall under the definition of personal care robots and what distinguishes personal care robots from robots in other areas, such as medical robots or industrial robots. This document also provides further guidance on the risk assessment and risk reduction process to be conducted for a personal care robot.

Mobile servant robot–personal care robot that is capable of travelling to perform serving tasks in interaction with humans, such as handling objects or exchanging information (SOURCE: ISO 13482:2014, 3.14).

Physical assistant–robot personal care robot that physically assists a user to perform required tasks by providing supplementation or augmentation of personal capabilities (SOURCE: ISO 13482:2014, 3.15).

Person carrier–robot personal care robot (3.2) with the purpose of transporting humans to an intended destination (SOURCE: ISO 13482:2014, 3.16).

Artificial intelligence can be treated as part of medical robotics, that is why it is worth paying attention to ISO/IEC JTC 1/SC 42 Artificial intelligence (https://www.iso.org/committee/6794475. html).

Artificial intelligence can be treated as part of medical robotics, that is why it is worth paying attention to ISO/IEC JTC 1/SC 42 Artificial intelligence [25].

From 26 May 2020, Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices (MDR–Medical Device Regulation) was to apply.

For many years, work has been carried out on a special standard for medical robots. The updated list of standards important for inventors and manufacturers of medical robots can be found on the Digital Innovation Hub HealthCare Robotics HERO portal [13].

9 AI and Robots

In our opinion, artificial intelligence (AI), machine learning is a field of robotics. Its development makes us think more and more boldly about robots becoming independent. Image analysis is one of the medical fields that is a natural area for testing artificial intelligence (from historical point of view machine learning have been developed by looking at, analyzing natural solutions for the perception of the human brain). Medicine should deal with prevention–this requires analyzing a huge number of facts about the patient himself and his environment (from climate to politics, from nature to history). This is a chance for AI to help people.

We live in the space of information, that is, we can assign data and history to specific points in time and space. AI can "broaden" human memory & senses, which will allow him to make better decisions.

In the future, as potential patients, we will have our model, simulator, that is, our "twin" covering the genotype, phenotype, history and effects of diseases and injuries. Doctors will be able to test on this model–a twin–the potential benefits and dangers of using various methods of treatment, the use of drugs, and performing surgery on the twin will facilitate the appropriate planning of the operation. Selecting materials and devices, and even a medical team, and practicing specific surgical movements and medical activities during the procedure.

Artificial intelligence is an element of robotics, because the concept of a robot is derived from the desire to make a machine similar to a human being.

Man, not only uses technology, but is also entangled in technological relationships–practical and also ethically.

Even today, autonomous vehicles travel on the roads of many cities in the USA, Europe and Asia. Legal solutions are going after development of technology. But the direction of changes is already chosen and will not change–we will be surrounded by independent robots. Robots will accompany us at work as trusted partners and home, securing basic cleaning works, during entertainment and will be ready to help in the event of a life-threatening situation. They will be the real guardians of our well-being and life.

AI and machine learning technologies enable robots to sense and respond to their environment. Medical robots will be more and more independent. Thanks to AI and accepted technical solutions guaranteeing security. Corindus, producer of CorPath GRX platform, received permission from the United States Food and Drug Administration (FDA) for the first automated robot movement called "Rotate on Retract (RoR)" in 2018 This was the first step towards introducing autonomous robot operations [37, 38].

But, what do we need to be able to use AI in MIS (Minimally Invasive Surgery)?

1. In advisory programs and planning support systems, AI serves as a method of linking the current situation during a surgical operation with the results obtained earlier in clinics (if they are digitized, i.e., form elements of the Big Data base) or obtained from physical modeling, computer simulations or the use of models theoretical in calculations.
2. In mechatronic surgical instruments, if the instruments are equipped with sensors (determining the features of local surgical intervention), soft & hardware to understand what specific signals mean for the task being performed and monitors presenting data to the surgeon online.

3. In surgical telemanipulators, if robots are equipped with sensors (determining the features of local surgical intervention), supervision and control systems transmitting force information to the operator's hand and other (often processed by computing systems) information on the robot control console monitor (they create the space operations enriched with physical data). Innovative surgical instruments equipped with force sensors and/or performing semi-automatic tasks (such as sewing) play a key role in this process.
4. In independent surgical robots, if the robots have their own computer (strong and fast, respectively) or in an open system in the cloud and an electromechanical system with a control system working in an information loop containing sensor data (imitating the sense of sight, sensation, smell), data from memory resources regarding the practice and theory of selected surgical procedures (imitating the medical knowledge of the surgeon), decision making system (AI) and the implementation of surgical tasks (robot executive tools) [38].

Artificial intelligence can contribute to reducing errors, improving standards, quality of performed operations and increasing patient safety:

1. Indirectly, passively as the basis for the work of the advisory system when planning and performing operations.
2. Indirectly–as an element of surgical telemanipulator or mechatronic tools.
3. Directly–as the basis for the operation of an autonomous surgical robot.

Improving the decision-making process is always associated with access to good, reliable and timely information. The surgeon manually performing a classic operation uses information with all the senses. The surgeon loses access to all this information by moving away from the patient during surgery; endoscopic instruments with a distance of about half a meter and with telemanipulators over a distance of several meters or many kilometers [38].

The main idea: the less invasive the surgical intervention and the greater the distance the surgeon from the operating table is, the greater role of decision support systems (AI) and performance of specific tasks (Robots) is [38]. This principle formulated for MIS can be freely transferred to other activities within medical services. Acting in the area of lack of full knowledge, we will have to rely on analyzes conducted by AI and ML. Thanks to the sensors, robots can inspect in real time, but they have access to a huge, updated database of medical events and facts and build their experience by quickly analyzing the effects of decisions made on models–before making a decision (as AI has proven competing with humans in popular games).

How to determine the advancement of robots' independence? The autonomy of medical robots on a five-point scale (modeled on the SAE J3016 standard). 0. The lowest level, 0, is a lack of autonomy. Level 4, fully automated work, have achieved computer tomography or robotic radio-surgical knives that move and operate in accordance with the planned trajectory and tasks specified before the surgery. Level 5 (means a robot working fully autonomously)–no medical robot has achieved so far [38].

We treat the human body as a biological, physical, chemical and biocybernetic (IT) system.

10 Ethics of the Age of Medical Robots

Ethics concerns the human–human relationship, human–nature. A separate person comes into contact with the environment represented by other people, animals, plants–generally the world, the cosmos. But what if there are human-made intelligent individuals–robots?

One of the interesting aspects in terms of the human–machine relationship is the direct connection. It seems that we are inevitably getting closer to the world of cyborgs. Because it will be possible. And it will be used because it will be optimal and functional. So, we will have the problem of transforming a human into a hybrid individual. If we combine the advantages of a human (sensitivity, subjective relations, commitment proportional to challenges) and a robot (unlimited access to information, objective decisions and standard implementation of tasks)–it will be great. But what if we combine our disadvantages with the unlimited possibilities of robots? 100 years ago, recalling the word "robot", Karel Capek created an amusing theater play on this basis.

Medical robotics offers a paradigm shift in the way medical responders works. Similar changes apply to home care, for instance in postoperative care or rehabilitation. However, it's the medical-social companion robots that will be responsible for the greatest impact on human societies. Social robotics will change both the societal fabric and philosophy.

Robots were created on Earth in the image of man and for man.

Everything indicates that, is impossible to achieve an universally, easily accessible and cheap (?) but high-quality, personal healthcare without medical robots. How will people who are looked after by a robot cope with it? How will we make decisions and how will robots make theirs?

I believe ethics to be the art of making the right choices so let's take a step back and seek advice of the ancient philosophers. For ages, philosophers have been analyzing issues connected with the man versus-man and man-versus-the world relations in order to help us comprehend the reality and find the correct conduct. The term "ethics" comes from the Greek word *ethicos*, meaning a way of conduct accepted in the society, a conduct according to the legal character (ethos character). Today, "ethics" is colloquially understood as "morality", although the Latin word *moralis* denotes more the judgement of the appropriateness of a given action than a person's character. Plato (427–347 BC) believed that most people live in ignorance, and therefore cannot be expected to make the right decision. Knowledge is a virtue; nobody purposefully chooses the wrong way [36].

This situation seems extremely advantageous for the absorption of AI as an intelligent guardian angel. If AI is a safeguard of common sense and knowledge based decision.

In his Nicomachean Ethics, Aristotle (384–322 BC) identified two types of virtues: moral (courage, generosity, modesty, etc.) and intellectual (wisdom, intelligence, reason). We all have the possibility to develop a virtue, however, only few are successful. What kind of virtues we can find in our robot? Which one is the most important? Thomas of Aquin (1225–1274) perceived moral problems in the context of the law of nature and God's commandments. A moral life is life "in accordance with reason". All people are equal and for all people there is a close connection between happiness and a righteous conduct (conscience) [1]. We–the people–create robots. They should respect the rights which we will indicate. Should we build robot conscience to serve as decision control? Immanuel Kant (1724–1804) believed that a man's kindness does not depend on the effects of his actions, since there are too many factors which influence these actions and which we cannot influence. The development of the goodwill is the most important human aim. However, the goodness of the robot will depend solely on the effects of its actions. The purpose of the training will be the best fit to the needs of humans, customized and optimized for specific tasks. Jeremy Bentham (1748–1832) formulated a concept of utilitarianism according to which, one should be useful and act in a way that would cause the greatest good for the greatest number of people. His principle of utility regards "good" as that which produces more pleasure and less pain. Action consistent with the philosophy of utilitarianism means that it is the results or consequences of a given act that decide on its moral value, and not the reasons or motivation behind that act [36].

Robots will be no more and no less moral than its creators. Should a robot be useful only for its caretaker? How should they make decision regarding other people; sometimes what's for good for an individual contradicts what's good for others. A.J. Ayer (1910–1988) and CL Stevenson (1908–) created the theory of emotivism. According to this theory, the scientific, empirical method of verification of statements is inefficient in ethics, whilst morality is only connected with our feelings. However, we won't program our own feelings in robots we may program our preferences. The contemporary British philosopher, Alasdair MacIntyre, in his search of understanding of good and evil delves into history and art, such as: the Homeric virtues (strength, courage, cleverness, friendship), Athens' virtues (courage, friendship, moderation, wisdom), and medieval virtues (bravery, justice, moderation, wisdom, faith, hope and love) [36].

Science fiction offers similar views on robot behaviour. Joseph Fletcher considered moral issues in three ways: • legalistic (based on unknown commands); • autonomic (rejecting rigid norms); • situational (e.g., "love thy neighbour"). His situational ethics suggests that decision making should be based on predicting the consequences of a given act. This, however, is not always possible. Definitely relevant for robots–that decision making should be based on predicting the consequences of a given act. The use of robotics in the treatment process facilitates remote medical care, consultations and the monitoring of a patient's condition. It is therefore a chance for a greater availability and quality of medical services. Robots are a breakthrough in the infrastructure, the organisation of the operating theatre and in the specialised training of surgeons. But how to evaluate a wrong decision or an action of remote physician? How to divide the responsibility for the mistakes of remote robotic devices? The access to information depends on technical resources, software, etc. Therefore, the final effect is influenced by a number of people–engineers, administrators, economists, etc. as well as fortuitous events. Even lower–level robots, let's call them servant robots need to obey the three laws of robotics formulated in the 1950s by

Isaac Asimov in his book entitled: “The Caves of Steel”: • A robot may not intervene in human activities (exception–when the activities may injure a human being). • A robot must obey any orders given to it by human beings (except where such orders conflict with the First Law). • A robot must protect its own existence, as long as such protection does not conflict with the First Law or the Second Law. On principle, the laws of robotics set forth the supremacy of man over robots in every circumstances and, in the case of conflict, the task performed by a”less intelligent” robot should be subordinated to a robot equipped with a more comprehensive system of the assessment of its work environment. As far as medical robots are concerned, their tasks are primarily regarded as intervention into the human condition in rescue operations, surgical procedures or different therapies. Currently, robots are also used for psychological therapy, on the grounds of excellent experience of robotics gained in entertainment functions. Do humans have any obligations or duties towards mechanical devices? No. The answer may not be that straightforward however if those mechanical devices determine our capabilities of helping others. The guarantee of safety may mean that there is a need of proper servicing and maintenance of devices, which are used for saving lives. What if robots starts manufacturing robots? Then we may have to ensure the uninterrupted supply chain of parts. What if by enforcing waste sorting and recycling we will involuntarily introduce “racial segregation” of robots and the need of robot “health (service) care? A fund with spare parts? [36].

Remote patient monitoring, treatment of consultations offers a chance to increase the availability and quality of healthcare. Thanks to the introduction of robotics into the treatment process it is possible the practice “remote” treatments. “Existing telerobotic systems can be classified as two categories: teleprogramming systems or telecontrol systems. In teleprogramming systems, operators rely on the visual information only to instruct robots to execute a motion that has already been programmed offline, and the method is basically an open-loop control. In telecontrol systems operators have direct access to real-time control of the robot at the remote site based on real-time feedback of supermedia information, including images, force, and other information for haptic rendering” [30]. Ethics and our sense morality blindly assumes direct human contact. Our conscience and empathy work differently when faced with abstract terms such as lack of direct correlation between our actions and its effects on a fellow human being. How will we morally and legally assess mistakes made through telemedical systems? Personal, caregiver robots will be our companions in not-too-distant future. The elderly and people living alone will be taken care of by robots supervised by designated remote care centers. Specialized expert-systems will make necessary decisions regarding the medical intervention, should one be required. Without a doubt, such system will boast a great efficiency, however, what will the social implications of such isolation be? After Frankenstein and robots, cyborgs are yet another imaginary creation that is entering reality [36].

The integration of engineering and biology is a fact. This raises the question as to the borderline between a biological organism and a technical device. Where is the beginning of consciousness and intelligence? In my opinion, it is when the question “why” appears between the information from the environment and the actions [36].

To summarize: a robot, as an art object, is created in the likeness of humans to carry out work on their behalf. The robot is an invention that was popularized almost 100 years ago, long before it became a real object. Ethical issues that producers are interested in include neighborliness with robots or hazards arising from their independence or their misuse by humans.

As we can read in “Robot Ethics” by Patrick Lin, Keith Abney and George Bekey. “The emergence of the robotics industry” observed Bill Gates “is developing in much the same way that the computer business did 30 years ago” (2007). As a key architect of the computer industry, his prediction has special weight. In a few decades–or sooner, given exponential progress forecasted by Moore’s Law (that computing power will double every eighteen months or so)–robots in society will be as ubiquitous as computers are today. Gates believes; and we would be hard pressed to find an expert who disagrees” [29].

A personal robot should improve the safety and the level of freedom of humans.

11 Discussion

Robotics will play a key role in the future of our societies. The widespread presence of robots in our environment will change the work culture and the way we “practice” many everyday human activities.

In medical robotics, the fact of direct contact with the patient’s body is important, as it has a direct impact on the life of the human being cared for by the robot. This is not only an engineering challenge: design and execution, but also ethical and economic.

Our cooperation with robots, both as service providers and recipients of medical services, will require appropriate behavior and education from us.

Until now, medical devices and tools were oriented towards service providers, i.e., doctors and nurses. They can now be patient-oriented. It is a change in the way medical services are performed. In this way, we are entering the implementation of a new paradigm of providing medical services. From patient < > doctor to patient> technology> doctor to patient> health technology, in which system the doctor has only a controlling and innovative role. The real breakthrough in health services will come from autonomous robots.

Robotization, automation of a medical service requires, first of all, understanding its essence, selecting, defining individual tasks and dividing the process into repetitive activities or decisions. Similarly to the knowledge of the rules of conduct calculation, “how the counting process works”, as the basis for learning mathematics, caused the possibility to transfer this process to machines. Knowledge of the causes and effects of specific medical activities or, often based on statistics, medical standards constitute the basis for starting the process of automating medical services. Robots with access to patient data and own (sensory) information, controlled by an artificial brain, provide an opportunity to make work independent and personalize the implementation of medical care.

Today we have the technical means to make a choice–to decide whether the performance of work as part of a medical service requires the mobility of information or the medical team, sending information to the patient or performing a procedure on the patient. If the patient needs a drug or drug substrates (if in the future the patient will have a printer producing drugs with him, materials and instructions, a drug development program will suffice), a medical device (e.g., a pulse oximeter, which connected to the network and allows to monitor patients during a pandemic staying at home), then we will send a drone with everything necessary. That is–the mobility of only the therapeutic agent.

We have a lot of evidence of the effectiveness of the use of drones, necessary in many critical situations. Half of the world's population have limited access to healthcare. Help can be provided by drones, because can fly long distances (over dangerous terrain) to deliver medication, vaccines, biological samples, diagnostic kits and food supplies. Remote-controlled transport of materials supported by telemedicine technology of data transmission and providing professional advice in many cases can save lives. Tests of such systems are already very advanced. Examples [34] (1) In 2020, Volansi piloted the first delivery of temperature-controlled vaccines in the US. (2) In 2022, due to Hurricane Ana, drone suppliers (Swoop Aero) are distributing medicines and polio vaccines. (3) In Democratic Republic of the Congo the Ebola bi-directional medical delivery drones (drop-off the Ebola vaccine and then collect biometric data) have been used 4) for villages in the jungle-covered mountains in archipelago Vanuatu–remote food supplies and medicines.

Effective aid requires the involvement of developing countries as equal partners in the creation and dissemination of knowledge. The power of modern communication systems, their decreasing costs and new methods of learning are hope for the future and an opportunity to spread a better standard of health care [47].

What are some disadvantages of medical robots? There is also the possibility of mistakes and malfunctions by direct or indirect care robots: 1. Initial investment is very high. 2. The answer for the question "Who is responsible for mistakes made by robots?" is still open (law and ethics). 3. The use of robots requires new skills for staff–new competences are needed and time for education. 4. Healthcare robots require access to data and create data, imported and exported via the web, are generally "telesensitive". Their use in direct contact with people while replacing or supporting caregivers, surgeons or provide psychological or motor rehabilitation may imply tragic implications of hospital robots caused by programming error or a sabotage. This increases the role of cybersecurity issues. And another cost increase–see point 1. The main task for engineers is therefore: do it cheaper! safer! modify solutions faster, according to emerging new needs! and improve service!

The analysis of the life cycle of robots shows that it is worth taking care of their proper use and efficiency. These are largely IT devices, so they age as fast as computers, smartphones or their software. The cost of buying a robot is high, so the robots should work every day. The most expensive are diagnostic and surgical. Qualified personnel are also valuable. Due to the fact that the number of certified surgeons is limited, it pays to ensure that one surgeon can perform e.g. 4 operations a day (usually one operation lasts less than 2 h). That is why it pays off, for example, to buy two robots for one facility in order to maximize the use of the team.

In this chapter a quick overview of the state of medical robotics and many issues ranging from economic to ethical have been made in order to seek an answer, to the question about the directions of robotization development, very diverse in the way healthcare works. When, where and why to send information over a distance, and when devices, when, where and why to use remotely controlled robots (telemanipulators) or automatic, or autonomous robots.

Due to the Internet network dependence robots, the threat of cyber-attacks is growing. Democratization, universal access means that the criminal world of-ten gets simple tools to commit crimes. However, we are optimists. We cannot afford the lack of such an open, courageous approach. The safety and well-being of citizens around the world depend on it. The situation is somewhat reminiscent of the ease of counterfeiting money, especially today. And yet, by applying very strict laws and controls, we manage to keep our criminal intentions under control. There are also design constraints.

Progress in mechanical, like electronics, depends on advances in material physics (and access to some-times rare minerals). However, unlike electronics, the progress in miniaturization and ease of production of mechanical components (although we have 3D printers) and the ingenuity of solutions is slow and insufficient. To perform work in the dimension needed for the operation, we need forces of 10N, to transport drugs or patients, we need powers of the order of kW. Anyway–based on the genesis of the robot, i.e., its resemblance to humans: the total resting power of the heart related to the activation of blood circulation in small and large circuits is about 1.4 W, and the resting power of a person weighing 60 kg is about 76 W. When heart prosthesis is designed, because of the energy loss, a motor with a power of about 14 W, is need, if the goal is transporting a human, then …. much more, depending on conditions. And yet we mainly use electricity, which we transform into mechanical work in a way that was invented 200 years ago. The use of pneumatic or hydraulic drives in medicine is not widespread for reasons of maintaining an appropriate level of safety. The rapid development of electronics and computerization was possible because we changed the rules of operation, not just materials. There are no such breakthroughs innovation in mechanical and electrical systems. To obtain power from the rotor spinning in the magnetic field, we build motors that are heavy. When mounting motors on a robot structure with many degrees of freedom, we have to oversize its construction elements, due to the safety and the required precision of the effectors–tool tips. Since the robots have to do the right mechanical work, they are heavy and expensive. If we are to move in space, ride, swim or fly, we must use appropriate technical solutions (e.g., wheels or wings) that are rarely suitable for miniaturization or reducing production costs. What a difference to computerization, where computing abilities are not related (so closely) with the weight and volume parameters of the device and the price.

First of all–man is mobile. He changes place and if he needs help, the appropriate medical service should follow him. Only then will it be effective. However, there are different needs and different types of services. Modern technologies come to the rescue: IT, computers, smartphones, MEMs. More and more portable and wearable devices allow some of the diagnostic work, to be done independently or automatically on the patient. In most cases of geographic location, today it is not difficult to send a set of information remotely and receive specialist advice via the Internet or telephone network. Telemedicine already offers excellent support for medical services, when it is possible to "treat with information". In the future, when it is possible to print drugs or orthoses, we will use a distributed network of properly equipped mini-help centers, self-controlled or remotely controlled by specialists.

Every medical service is based on data. Cloud technology allows access to memory-intensive data everywhere. We can move both in geographic and data space at the same time. We can be warned against external threats (e.g., weather) and internal threats (important e.g., for diabetics). AI trainers (including mobile robots accompanying e.g., runners) help in the field of sports medicine. In the field of occupational medicine, professional drivers are inspected. If we believe the authors' belief that artificial intelligence is an element of robotics, it is difficult to find an area where the robot is not treated as an opportunity for progress in the field of democratic, universal, cheap access to appropriate quality services.

Mobility of robots has different meanings in various fields of medicine and different methods of successfully using robots there are being developed. The effective and economically justified application of medical robots are as follows.

In sparsely populated countries, such as Canada, it pays to install surgical telemanipulators to bring them closer to those in need, because competence can be transferred remotely–a surgeon from a reference center in a given field can supervise and control robotic tools inside the patient's body. Moreover, in surgery–the "portability & agility" trend has emerged–robots should be more flexibly adapted to the type of operation and the place where the operation is carried out. Today, a surgical robot can be borrowed or rented for the day of surgery. It is not only economically easy to calculate, but above all possible due to the adoption of appropriate technical solutions by several competing companies (e.g., light functional modules). However, if the star wars paradigm returns, we will return to space hospital installations, as today field hospitals in case of war and qualified medical centers aboard ships are designed. All these medical goals can only be achieved with the right robots. Telemanipulation is associated with the risk of delays and loss of control–that is why autonomous robots are developed, also in the field of surgery.

In rehabilitation, we require a specialist to assess the patient's condition and perform many repetitive body movements. The sensory robotic arm can perform most of the typical rehabilitation tasks. Robots such as the popular Luna (Fig. 2) [16] are transported to the place of use without any problems. This is also IT device, so they can be remotely supervised and remote controlled. If we borrow them home for the duration of convalescence and rehabilitation, both robots and patients, will be cared remotely by the best specialists.

Diagnostic devices are getting smaller, they can be transported in ambulances, stored in places frequented by people: such as airports or stadiums, devices can also have therapeutic and rescue functions (like AEG–Automatic Electronic Defibrillator)–similarly to rescue and firefighting equipment. Digitalized data from stationary diagnostic devices, such as tomography, can be freely collected and transmitted–following the patient.

The pandemic significantly accelerated the process of adopting telemedicine and rehabilitation robots (one specialist may look after more patients). The strongest development, however, is in logistics. For the same reason. The development of means of transport is the result of advances in the technique of building and controlling drones. In terms of hospital robots, their ability to move independently is the result of the advancement of sensory technologies (similar to those used in road vehicles) and machine learning, as well as the miniaturization of computers (weight) and cloud-based data storage technologies. Autonomous emergency vehicles, mobile, automatic pharmacies or robots searching for and collecting the wounded–this is a chance for a safe living space for people.

We are definitely facing a solution to the problem of the organization of the entire healthcare, in which we will have teams of robots specialized in various fields, the coordination of each action, the cooperation of robots and people, will require a completely new system approach.

According to the World Health Organization (WHO), a well-functioning health care system requires a financing mechanism, a well-trained and adequately paid workforce, reliable information on which to base decisions and policies, and well-maintained health facilities to deliver quality medicines and technologies. Let's think–if we have a good financing system for medical robots and a legal system, they can perfectly meet these expectations: diagnostic robots will collect relevant information, digitized data will be sent to therapeutic robots, therapeutic robots will undertake treatment, rehabilitation (after receiving updated data) will take care of returning to patient well-being, in the meantime they will provide appropriate transport and communication services (medicines, advice). Of course, at home, the patient will be able to count on help from social robots. It's perfect. And it all starts with the digitization of data and learning of the robot's behavior in the human life space and controlled interference in the human body if medical treatments are required. And the mobility of information, patient or … service.

The challenges of the future will be no less than those faced by our generation. There is a lack of millions of health care workers to maintain the level of medical assistance provided in future at the present level. However, expectations and needs are much greater. Contact of a competent employee with the patient, i.e., the human–human relationship, is still a determinant of comfort at a time when we get into health problems. Any automation in the economy or administration is favorable to medicine because it frees human resources to play (probably) the most important role–helping people and providing medical care. In the near future, most of the employees will work in health services.

Are we ready? Autonomous vehicles in the US are currently being tested on the road. Responsibility criteria are also tested. In medical robotics, the fact of direct contact with the patient's body is important. This presents a set of design and communication challenges. The possibility of making mistakes provoke asking questions. The answers require an ethical analysis. We are very fascinated by the vision in which robots could be partners of elderly and infirm patients. A personal robot should improve security and the level of human freedom [1]. The world population is already 8 billion. We are accompanied by countless living organisms and several million more and more intelligent robots. Medical robotics will play a key role in the future of societies. However, this role is not without many doubts and controversies. Intelligent and self-aware robots will soon become members (elements) of human society. This is the next step in the evolution of man and his world.

How many types of medical robots will we have? As many types as medical services? Or as much as employees? Health workforce list consist of: Physicians/doctors, Midwives, Nurses, Pharmacists, Dentists, Other health-care providers (including community health workers)–according to WHO [52]. Will they be specialized or multidisciplinary, modular or compact, free or licensed; objective or subjective, empathetic or compassionate, universal and widely available or access to robots will be limited; fully independent and developing without our control or definitely subordinate to us? Our future depends on the answers to these questions.

The robot allows us to multiply the work. It allows us to be more effective and, as a result, more sensitive to the patient (because it frees our time, senses and strength). Telecommunications robots and technology allow us to transfer our competences and possibilities of action to where they are most important at the moment–despite the distance from the patient.

What we create should meet the criteria of beauty (art), good (public activity), truth (science) and utility (inventiveness). A medical robot is today more of an IT tool than a mechanical one.

12 Conclusion

Already today, health services constitute the greatest burden on the human and economic resources of modern countries. It is also difficult to imagine a more important task from the humanitarian and social point of view. Automation and robotization of production and services caused the development of our civilization. Technology influenced the world order and laid the foundations for democratization and the promotion of a good quality of life. It is time to transfer this pattern of technological success to the area of medicine and health services. As a rule, for medical assistance to be effective, it must be delivered at the right time and place where it is needed. This requires the transfer of information and action over a distance. Medicine as a process and practice for preventing disease and caring for people in need must be based on robotics. This is a challenge not only for engineers but for every participant of the system, both for service providers and recipients of medical services. As with any civilization changes, this one will also have a cultural dimension. The universal presence of robots in our lives will require appropriate behavior and education, modification of applicable laws and ethics. Engineers are facing challenges to make robots cheaper, easier to use and safe. The robot will relieve us of many duties, because they allow us to multiply our work. This will allow us to be more human, it will have a positive impact on interpersonal relationships. We can afford it.

Telecommunications robots and technologies allow us to transfer our competences and capabilities to where they are most important at the moment–despite the distance from the patient. Thanks to digitization and telecommunications technologies, we no longer have problems with transmitting information. We can collect and analyze big data (AI). Communication, information transfer is now fully mobile. The rapid development of more and more robotic means of transport, road vehicles and drones mean that today there are no big problems to send the right package to any place on earth, to the moon or a nearby planet. Still, the challenge is to do the job as a medical service in a similar way. Robot mobility has different meanings in different medical fields, and different methods are being developed for the effective use of robots there.

Medical robots are the only chance to provide broad, democratic access to quality healthcare services in the near future. The ongoing demographic changes, the lack of an appropriate number of medical personnel and the need to raise the standards of services provided in hospitals, clinics and homes in need are global trends that force greater involvement in the field of medical robotics. Already today, robots (industrial robots) determine the quality of our lives to a large extent. However, everything indicates that medical robots (including artificial intelligence) will guarantee a safe future for homo sapiens.

It is uncertain whether men are from Mars, women are from Venus, but certainly robots are from Earth. We have a chance to create them in such a way that they will become our chance for a long and happy life for people, fairly and universally. Robots are humans' most powerful tools.

References

1. ACCREA https://accrea.com/pl/disirt-robot-do-dezynfekcji
2. Adamczyk D, Jankowski J (2020) Application of exoskeletons in motor rehabilitation in patients with multiple sclerosis (in polish), Medical Robotics Reports 8–9: 41–46, ISSN 2299–7407
3. Aethon https://aethon.com
4. Akpunonu B, Hummell J, Akpunonu JD, Shahab UD (2022) Capsule endoscopy in gastrointestinal disease: evaluation, diagnosis, and treatment. Clevel Clin J Med 89(4):200–211. https://doi.org/10.3949/ccjm.89a.20061
5. AGV https://www.agvnetwork.com/uvc-disinfection-mobile-robot
6. ARTAS https://roboticrestoration.com/
7. Cieza A, Causey K, Kamenov K, Hanson SW, Chatterji S, Vos T (2021) Global estimates of the need for rehabilitation based on the Global Burden of Disease study 2019: a systematic analysis for the Global Burden of Disease Study 2019. The Lancet 396(10267):2006–2017. https://doi.org/10.1016/S0140-6736(20)32340-0
8. CMR (2022) https://cmrsurgical.com/news/5000-cases
9. Coxworth B (2019) Legged micro-robots designed to walk within the body New Atlas. https://newatlas.com/walking-micro-robots/58797/?fbclid=IwAR1H3wHyX7jh8-oPbEUjPLr3Y80UfAf1k-Xbthhqin4t_6g_ewgs7xHeSWQ

10. David D, Matu , David O (2014) AI Robot-based psychotherapy: concepts development, state of the art, and new directions. Int J Cogn Ther https://www.researchgate.net/publication/264057920_Robot-Based_Psychotherapy_Concepts_Development_State_of_the_Art_and_New_Directions
11. DARPA https://www.darpa.mil/news-events/2019-05-20
12. Dégoulange E, Pierrot F, Dombre E, Urbain L, Caron P, Gariépy J, Jean-louis M (1999) Hippocrate: a safe robot arm for medical applicatioons with force feedback. Med Image Anal 3(3):285–300
13. DIH dih-hero.eu https://services.dih-hero.eu/standards
14. DP (2022) https://doctorpreneurs.com/9-drones-that-will-revolutionise-healthcare
15. Edelman BJ, Meng J, Suma D, Zurn C, Nagarajan E, Baxter BS, Cline CC, He B (2019) Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. Sci Robot 4(31). https://doi.org/10.1126/scirobotics.aaw6844
16. Egzotech (2022) https://egzotech.com
17. Global Surgical Robotics Market 2021–2031. https://www.researchandmarkets.com/r/awdul8
18. Hanson (2022) https://www.hansonrobotics.com/sophia
19. Heartlander (2022) https://www.cs.cmu.edu/~heartlander
20. Hyundai launches Boston Dynamics AI Institute (2022) https://www.therobotreport.com
21. I Data Research (2022) https://idataresearch.com/product/laparoscopic-devices-market/
22. Inside Amazon's robotics ecosystem (2022) https://www.therobotreport.com
23. Intel (2022) https://www.intel.com/content/www/us/en/newsroom/news/autonomous-robot-tech-triages-covid-19-patients.html#gs.akbrv5
24. ISO 1 (2022) www.iso.org
25. ISO 2 (2022) https://www.iso.org/committee/6794475
26. Keenon (2022) https://www.keenonrobot.com
27. Krawczyk D, Nawrat Z (2022) Report–remote control of a robot from the Robin Heart family between Zabrze and Dubai (in polish). Medical Robotics Reports, 10
28. Landers L, Timoney CF, Felder RA (2000) Medical mobile robotics: an industry update. 5(3) SLAS Technology, SAGE Journals. https://journals.sagepub.com/doi/abs/10.1016/S1535-5535%2804%2900070-X; https://doi.org/10.1016/S1535-5535(04)00070-X
29. Lin P, Abney K, Bekey G (2012) Robot ethics. The MIT Press, The ethical and social implications of robotics
30. Lo W, Liu Y, Elhajj IH, Xi N, Wang Y, Fukuda T (2004) Cooperative teleoperation of a multirobot system with force reflection via internet. IEEE/ASME Trans Mechatron 9(4):661–670
31. Mayor N, Coppola ASJ, Challacombe B (2022) Past, present and future of surgical Robotics. Trends Urol Men's Health 13(1):7–10. https://doi.org/10.1002/tre.834
32. Medtronic, HUGO 2022 https://www.medtronic.com/covidien/en-gb/robotic-assisted-surgery/hugo-ras-system.html
33. Medical Robotics Market by Product and Services, by Application, by End-users - Global Opportunity Analysis and Industry Forecast, 2022–2030. https://www.researchandmarkets.com/reports/5626172/medical-robotics-market-by-product-and-services?utm_source=CI&utm_medium=PressRelease&utm_code=wv8wtl&utm_campaign=1730334+-+Global+Medical+Robotics+Market+Report+(2022+to+2030)+-+Opportunity+Analysis+and+Industry+Forecasts&utm_exec=jamu273prd
34. Microbot (2022) https://microbotmedical.comNature, https://www.nature.com/articles/d41591-022-00053-9
35. Nawrat Z (2011) The ethics of artificial organs. ImplantExpert Ed Zbigniew Nawrat Zabrze: M-Studio:127- 135, ISBN 978–83–62023–72–1
36. Nawrat Z, Kawalski M (2013) Question about new ethics – ethics of medical robots era. Medical Robotics Reports 2/2013:22 -25, ISSN 2299–7407
37. Zbigniew N, Krawczyk D (2020) Healthcare Robotics 2020 & HERO. Medical Robotics Reports, 8–9 : 8–14, ISSN 2299–7407

38. Nawrat Z (2020) Robin Heart surgical robot: description and future challenges. Control systems design of bio-robotics and bio-mechatronics with advanced applications. Ed.: Ahmad Taher Azar [London] : Elsevier Academic Press:75–113
39. Paro (2022) http://www.parorobots.com
40. Servick K (2022) Windows of opportunity. Science 257–259 https://www.science.org/doi/epdf/10.1126/science.ada0099
41. Siemens (2022) https://www.siemens-healthineers.com/si/news/mso-robotic-assisted-procedures-in-the-cath-lab.html
42. Soto F, Wang J, Ahmed R, Demirci U (2020) Medical micro/nanorobots in precision. Adv Sci 7:1–34. https://doi.org/10.1126/scirobotics.aar4423
43. Szczęśniak-Stańczyk D (2015) Novel robotic system for remote ultrasonography–from idea to the first prototype. Presentation of the REMEDI Project. Medical Robotics Reports 4:31–38. ISSN 2299–7407
44. Tiwari MM, Reynoso JF, Lehman AC, Tsang AW, Farritor SM, Oleynikov D (2010) In vivo miniature robots for natural orifice surgery: state of the art and future perspectives. World J Gastrointest Surg 2(6):217–223
45. Troll ZW, Wei HJ, Stangfocke M, Wang Z, Dong M, Schnichels S, Fisher P (2018) A swarm of slippery micropropellers penetrates the vitreous body of the eye. Science advances, 4 (11A). https://doi.org/10.1126/sciadv.aat4388
46. UDO (2022) https://unitdoseone.com
47. VPC https://vpchothuegoldenking.com/pl/video-this-robot-penetrates-the-human-brain-and-treats-a-stroke
48. Warwick K, Nasuto S, Becerra V, Whalley B (2010) Experiments with an in-vitro robot brain, Chapter in 'Instinctive Computing', Lecture Notes in Artificial Intelligence, Y. Cai (Ed.) 5987:1–15
49. Warwick K (2013) The diminishing human-machine interface. Med Robot Rep 2:4–11. ISSN 2299–7407
50. Weizenbaum J (1966) ELIZA—A computer program for the study of natural language communication between man and machine. Commun ACM 9(1):36–45. https://doi.org/10.1145/365153.365168
51. Whooley S (2021) Ottava surgical assistant robot finally unveiled by Johnson & Johnson. www.therobotreport.com/ottava-surgical-assistant-robot-finally-unveiled-by-johnson-johnson. Accessed 29 Oct 2021
52. WHO (2022) http://www.wpro.who.int/chips/chip04/definitions.htm
53. Yunus A, Oncay Y, Oliver S, Joshua G, Ahmet FT, Sourjikand V, Siti M (2018) Soft erythrocyte-based bacterial microswimmers for cargo delivery. Sci Robot 3 (17). https://doi.org/10.1126/scirobotics.aar4423; https://doi.org/10.1126/scirobotics.aar4423

Event-Based Robust Control Techniques for Wheel-Based Robots Under Cyber-Attack and Dynamic Quantizer

Mobin Saeedi, Jafar Zarei, Mehrdad Saif, and Allahyar Montazeri

Abstract Nowadays, mobile robots are becoming an increasingly significant part of daily human life. Humanoid robots, wheeled mobile robots, aerial vehicles, mobile manipulators, and more are examples of mobile robots. As opposed to other robots, they are capable of moving autonomously, with sufficient intelligence to make decisions in response to the perceptions they receive from their environment. In today's world, cooperative tasks and the ability to control robots via networks make them a component of cyber-physical systems (CPSs). In this study, mobile robots that are acting as a part of CPSs are examined. Data-network burden, signal quantizers, cyber security, delayed transition, and robust performance are some of the challenges they face. A total of three sections are then devoted to addressing these issues in detail. As a first step, the governing equation for mobile robots is explained, and then their robust and resilient behavior of them is examined by establishing the event-triggered adaptive optimal terminal sliding mode control (AOTSMC) approach for nonlinear uncertain dynamic systems that are subjected to denial-of-service (DoS) cyber attacks. In this case, it is assumed that the conveyed signal is being corrupted randomly by an attacker. In this situation, it is essential to design the closed-loop controller parameters in such a way that the performance can be maintained under malicious attacks while the communication resources are preserved. Due to the unrealistic nature of delayed-free communication, the stability analysis is conducted for a general form of uncertain nonlinear delayed input dynamic systems. The quantization effect on the closed-loop control system is then analyzed in conjunction with

M. Saeedi · J. Zarei
Shiraz University of Technology, Shiraz, Iran
e-mail: m.saeedi@sutech.ac.ir

J. Zarei
e-mail: zarei@sutech.ac.ir

M. Saif
University of Windsor, Windsor, ON N9B2M4, Canada
e-mail: msaif@uwindsor.ca

A. Montazeri (✉)
School of Engineering, Lancaster University, Lancaster LA14YW, England
e-mail: a.montazeri@lancaster.ac.uk

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_6

robust behavior and event-based data transmission. A novel criterion is established to adjust dynamic quantizers' parameters according to the variation of event-triggering error, enabling the quantizer to be more accurate and facilitating implementation procedures. Finally, simulation results validate the presented methodology.

Keywords Mobile robots · Cyber-physical system (CPS) · Event-triggered methodology · Terminal SMC · DoS cyber attack · Dynamic quantizer

1 Introduction

There has been an unstoppable development in cyber-physical systems (CPS) in the engineering field in the last decade. They are constantly under development to become more secure, precise, and capable of making real-time adjustments in the engineering and manufacturing fields. Cyber-based control of robotic mobile devices gives them greater flexibility in terms of their application in different areas of industry, such as warehouse management systems, product assembly, distribution, and hazardous environments [1, 2]. The importance of this issue motivates us to present the current study in which the limitations and considerations of cyber-based mobile robots are addressed. Described in the following two subsections is a review of the CPSs, mobile robots, their dynamic behavior, and their challenges. A typical cyber-based control structure for mobile robots is shown in Fig. 1.

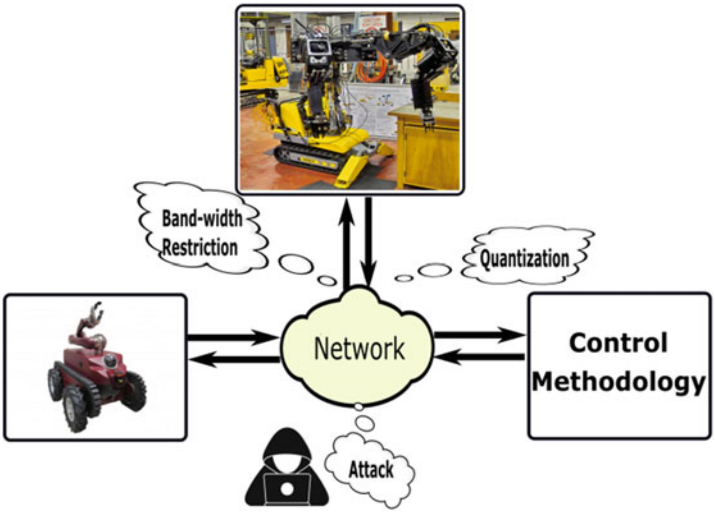


Fig. 1 A typical cyber-based mobile robot control system

1.1 Background

In recent years, there has been a growing interest in CPSs as a result of rapid advancements in digital communications [3]. These new areas require integrating physical systems, microprocessor-based controllers, and sensors through shared cyber layers to achieve desired objectives. As a result of the emergence of such systems, a new research area has emerged that provides control analysis of critical industries, such as oil and gas industry, health care services, as well as the robotic field [4].

Despite the fact that CPS systems offer increased availability and flexibility, there are a number of limitations associated with them, including packet dropout [5, 6], cyber-attacks [7–9], fading data [10], quantization of signals [11–13], and communication delay [14]. Satisfying the limited network bandwidth is an important issue while cyber layer is deployed to transmit data-packets. To cope with this problem, event-triggered data transitions have been developed as an alternative to time-triggered data transitions. Based on the research conducted on the event-triggering schemes, it has been found that it is an effective tool for reducing communication burdens, improving robustness against packet dropouts, and reducing vulnerabilities to cyber attacks [15, 16].

As opposed to traditional time-based methods that cyclically transmit signals within predetermined time instances, an event-triggered terminology would be viewed as a sample-based scheme. There are certain criteria that allow the transmission of measured data at a periodic time intervals that save communication resources while maintaining stability. Although the use of event-based methodology provides an analytical approach to make different control theories applicable to the CPSs in the presence of constraints in the network layer, proving Zeno-free behavior of the designed closed-loop, increased computational load in the design phase [17–21].

When it comes to event-triggering criteria and signal quantization, theory and practice are intertwined in CPSs. It is very important to maintain control precision and stability in physical systems controlled by digital communication, which involves constraints including maximum allowable resolution of analog to digital converters (ADC) [22], quantization errors [23], and the adjustment of sensitivity parameters in dynamic quantizers [24]. It has been shown that most studies by considering certain linear dynamics, present a quantization terminology or demonstrate criteria based on the continuous evolution of the quantizer sensitivity over time [24–26]. Applying these approaches to physical experiments in which time-varying desired responses are included in nonlinear uncertain dynamics does not lead to satisfactory results.

Cyber networks also face a number of challenges when it comes to controlling uncertain dynamic systems. The fulfillment of predefined control tasks for uncertain linear and nonlinear dynamics is always one of the essential researchers' concerns. Then, a lot of robust technics have been established to achieve this target. Among them, one of the most successful control schemes to deal with uncertain dynamic systems is sliding mode control (SMC) approaches [27–29]. Several studies have been conducted in the area of SMC, and various terminologies have been adopted, including event-triggered SMC [25, 30, 31], finite-time convergence [32, 33], switched systems [20], adaptive control [32, 34, 35], and cyber security [36–38]. Under event-triggering criteria, a super-twisting sliding mode controller is proposed in [19, 39, 40]. A sliding mode controller was developed in [41] for the quadruple-tank multi-input multi-output process with input delay. The terminal sliding mode controller is implemented in [32, 33] to find finite-time convergence.

In this work, the problem of event-triggered terminal SMC (ETSMC) in the presence of a malicious DoS attacks for uncertain nonlinear dynamic systems is investigated. Meanwhile, to consider the quantization problem, the new event-based dynamic quantizer is developed to reach maximum accuracy by adjusting the quantizer sensitivity. The novelty of this work would be categorized as:

- Presenting a new event-based optimal terminal SMC to achieve stability in regulation and tracking tasks for a nonlinear dynamic in the presence of uncertain terms under DoS attacks.
- In order to maintain stability under the presented control scheme despite being attacked by cyber criminals, a unique criterion is developed to take into account the DoS attack characteristics. despite being attacked by cyber criminals.
- A new dynamic quantizer approach in the presence of event-based methodology is provided to achieve stability and accuracy in different practical constraints.
  The next subsection investigates the governing dynamic equation of wheeled mobile robots (WMRs).

1.2 Motivations

Throughout the past few decades, a large quantity of investigation is performed on controlling nonholonomic wheeled mobile robots (WMRs). Among the control research topics that have been investigated for nonholonomic systems, the two main problems involved in the research are tracking and stabilization. Taking a look at Kolmanovsky's survey paper [42], one can see the intensive research efforts that are being made in this area. As many researchers have shown [43], stabilizing the equilibrium point of is a difficult problem due to structure of the governing differential equations. Since nonholonomic systems such as wheeled mobile robots, do not satisfy the Brockett condition, the linear control techniques do not provide an effective approach to address this class of systems.

Furthermore, in [44] it is shown that using constant state feedback is not useful in stabilizing nonholonomic systems. As part of the efforts of stabilizing WMRs, several approaches, including discontinuous state feedback [6], time-varying state feedback [45], hybrid controllers [46], and optimal control approaches [47] have been proposed. In fact, majority of methods applied the class of nonholonomic systems successfully are using the concept of chained form to represent the model. There have been a number of control models and strategies developed in order to stabilize uncertain chained systems. In [48], the stabilization problem for nonholonomic single-chained systems is addressed in the context of nonregular feedback linearization. In [49], the nonholonomic systems with nonlinear drifts are globally stabilized. A robust algorithm for nonholonomic robot control is presented in [50]. However, the primary disadvantage of such works is that they do not consider the dynamics of the system.

Since commercially available wheeled mobile robots do not have velocity sensors, using velocity state feedback is not practical and designing an output feedback controller for nonholonomic wheele-based robots is of practical importance. To solve this critical problem, researchers proposed a global output feedback stabilizing controller for a unicycle-type mobile robot based on the backstepping technique. Although, the majority of the proposed studies suffer from complexity and do not consider uncertain dynamic systems.

In the following subsection equations that represent the dynamic of wheeled mobile robots (WMRs) are formulated. The next sections are devoted to the design of an event-based robust controller in the presence of cyber challenges, such as signal quantization and cyber malicious attacks.

1.3 Problem Statement

Consider the nonholonomic wheeled mobile robot's dynamic as follows,

$$M(\varphi)\,\ddot{\varphi} + B(\varphi)\,\tau_d(t) + C(\varphi,\dot{\varphi})\dot{\varphi} + B(\varphi)\,F(\dot{\varphi}) = B(\varphi)\,\tau - A(\varphi)^T\lambda, \tag{1}$$

and,

$$A(\varphi)\,\dot{\varphi} = 0, \tag{2}$$

where $\varphi = [\varphi_1, \ldots \varphi_n]^T$ is a vector of generalized coordinates, actuators' inputs are defined by $\tau \in \mathbb{R}^{(n-m)\times 1}$ $M(\varphi) \in \mathbb{R}^{n\times n}$ expresses an inertia matrix, the Coriolis matrix is denoted by $C(\varphi,\dot{\varphi}) \in \mathbb{R}^{n\times n}$, friction vector is defined by $F(\dot{\varphi}) \in \mathbb{R}^{(n-m)\times 1}$, and $\tau_d(t) \in \mathbb{R}^{(n-m)\times 1}$ expresses external disturbances, $B(\varphi) \in \mathbb{R}^{n\times(n-m)}$ defines the input transformation matrix. Note that $A(\varphi) \in \mathbb{R}^{m\times n}$ is a full-rank matrix and Lagrange multiplier is expressed by $\lambda \in \mathbb{R}^{m\times 1}$ that denotes constraint forces. Suppose $S(\varphi) = [s_1(\varphi), \ldots, s_{n-m}(\varphi)]^T$, where $s_i(\varphi) \in \mathbb{R}^n$, $i = 1, \ldots, n-m$, and $A(\varphi)S(\varphi) = 0$. The pseudo-velocities of the system can now be obtained by considering (2) as $v(t) = [v_1(t), \ \ldots, \ v_{n-m}(t)]^T$ such that

$$\dot{\varphi} = s_1(\varphi)\,v_1 + \cdots + s_{n-m}(\varphi)\,v_{n-m}. \tag{3}$$

Then, by considering (3), (1) can be rewritten as,

$$M_1\dot{v}(t) + C_1(\varphi,\dot{\varphi})\,v(t) + F_1(\dot{\varphi}) + \tau_{d1}(t) = B_1(\varphi)\tau, \tag{4}$$

where $M_1 = S^T(\varphi)M(\varphi)S(\varphi), B_1(\varphi) = S^T(\varphi)B(\varphi), F_1(\dot{\varphi}) = B_1(\varphi)F(\dot{\varphi}), \tau_{d1}(t) = B_1(\varphi)\tau_d(t), C_1(\varphi, \dot{\varphi}) = S^T(\varphi)M(\varphi)\dot{S}(\varphi) + S^T(\varphi)C(\varphi, \dot{\varphi})S(\varphi)$.

In order to take actuators dynamic into account, firstly assume that the robot is being operated by $n - m$ DC motors. It is possible to write the electrical equation of each motor armature in the following way,

$$u_a = L_a \frac{di_a}{dt} + k_b\dot{\theta}_m + R_a i_a, \tag{5}$$

where k_b defines the back electromotive force (EMF) constant, $L_a, \;\; R_a$ introduce the inductance and resistance of the motor armature, respectively, and the voltage input is defined by u_a. Considering torque and armature current in the absence of armature inductance, i.e. $\tau_m = k_\tau \, i_a$, and pre- and post-gear torque-velocity relationships, i.e. $\tau = n \, \tau_m$ and $\dot{\theta}_m = n \, \dot{\theta}$, the delivered torque to the system can be expressed as,

$$\tau = k_1 \, u_a - k_2\dot{\theta}, \tag{6}$$

where $k_1 = (nk_\tau/R_a)$, and, $k_2 = n \, k_b k_1$, n is gear ratio and k_τ is torque constant of the motor. The Eq. (6) can be represented as follows,

$$\tau = k_1 \, u_a - k_2 X_1 v, \tag{7}$$

where $X_1 \in \mathbb{R}^{(n-m)\times(n-m)}$ creates pseudo-velocity vectors from wheels velocities. From (7) and (4), one can obtain,

$$M_1(\varphi)\dot{v}(t) + (C_1(\varphi, \dot{\varphi}) + k_2 B_1(\varphi)X_1(\varphi))\, v(t) + F_1(\dot{\varphi}) + \tau_{d1}(t) = k_1 B_1(\varphi)\, u_a. \tag{8}$$

Note that $M_1(q)$ satisfies inequality $m_1 \leq \|M_1(q)\| \leq m_2$, where m_1 and m_2 are positive scalar constants. The dynamic equation (8) and kinematic model (3) can be defined as follows,

$$\dot{x} = \begin{bmatrix} \dot{\varphi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} Sv \\ -M_1^{-1}((C_1 + k_2 B_1 X_1)\, v(t) + F_1 + \tau_{d1}) \end{bmatrix} + \begin{bmatrix} 0 \\ k_1 M_1^{-1} \, B_1 \end{bmatrix} u_a, \tag{9}$$

where $x \in \mathbb{R}^{(2n-m)}$ is the state vector. Note that at the rest of this study, the following formulation is dropped to investigate the controller design approach,

$$\dot{x}(t) = f(x(t)) + \Delta f(x(t)) + d(x, t) + bu(t), \tag{10}$$

where $x(t) = [x_1(t), \ldots, x_n(t)]^T \in \mathbb{R}^n$ is the system state, $\Delta f(x(t)) \in \mathbb{R}^n$ represents unknown terms, $d(x, t) \in \mathbb{R}^n$ an external disturbance, $u(t)$ defines input signal, and b is a constant parameter.

1.4 Chapter Organisation

After this introduction, the problem of event-based AOTSMC design in the presence DoS attacks for mobile robots is formulated and solved in Sect. 2. The quantization problem and a solution based on event-triggered terminal SMC design for uncertain input-delayed dynamics are formulated and the numerical results are provided in Sect. 3, and finally conclusions come in Sect. 4.

2 Event-Based AOTSMC Design Under DoS Attacks

An event-triggered adaptive optimal terminal sliding mode control (EAOTSMC) design approach under DoS attacks is presented in this section. Attackers are believed to ruin measurement signals randomly. Maintaining the closed-loop stability under malicious DoS attacks is the most important challenge. In order to achieve this goal, the EAOTSMC has been proposed in order to increase the robustness against attacks and reduce the computational load as a result, as shown in Fig. 2. Furthermore, the frequency and duration of DoS attacks are examined explicitly for their effect on the closed-loop stability, as well as on the schedules for controller updates. Thus, the cyber layer's bandwidth is determined to maintain closed-loop stability. Then, in the presence of uncertainties and DoS attacks, designed parameters can be adjusted by the designer. Finally, an evaluation of the proposed methodology is provided by numerical simulations.

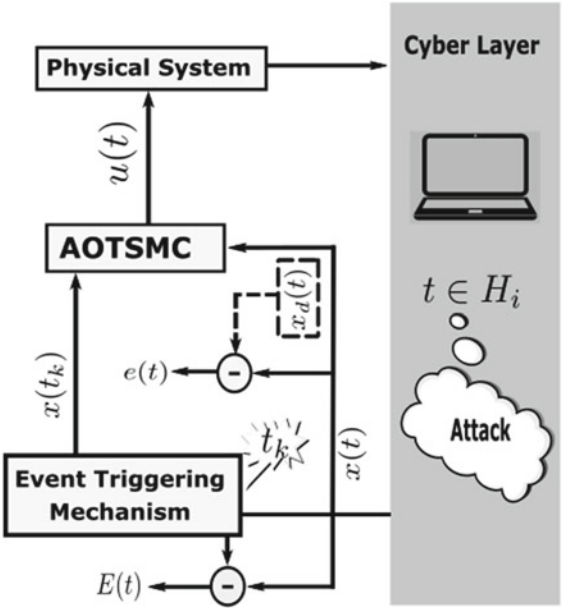


Fig. 2 The schematic representation of the proposed EAOTSMC approach

2.1 Problem Formulation

This section presents an adaptive optimal TSMC (AOTSMC) approach for an uncertain nonlinear dynamic systems. AOTSMC benefits from the fact that there is no reaching phase involved, and it guarantees stability in a fast finite time while fulfilling optimal criteria. Assume the sliding surface is designed as follows

$$S(t) = \vartheta_2\, \Theta(x,t) + \vartheta_1\, e(t), \tag{11}$$

where $e(t) = x(t) - x_d(t)$ expresses tracking error, and the known desired states is defined by $x_d(t)$. $\Theta(x,t)$ is an auxiliary state that is defined in the rest of this section, and ϑ_1, ϑ_2 are positive constants.

Assumption 1 Time-dependent derivatives of $x_d(t)$ exist.

Assumption 2 $x_d(t),\ \dot{x}_d(t),\ \ddot{x}_d(t) \in \mathcal{L}^\infty$ holds for all $t \in R_{\geq 0}$.

Assumption 3 $f(x(t)),\ \dot{f}(x(t))$ satisfy Lipchitz criterion. Then one can obtain

$$L_0 \|x(t_2) - x(t_1)\| - \left\|\dot{f}(x(t_2)) - \dot{f}(x(t_1))\right\| \geq 0, \tag{12}$$

where $L_0 \in R_{>0}$.

Assumption 4 Following inequality holds by $f(x(t))$ for all $t \in R_{\geq 0}$,

$$\bar{\alpha} - \|f(x(t))\|_\infty \geq 0, \tag{13}$$

where $\bar{\alpha} \in R_{>0}$.

Assumption 5 A bound is established on the deviation rate of uncertain dynamic and external disturbance inputs as follows,

$$\bar{\beta}\, \|x(t)\| - \left\|\dot{\Delta} f(x(t)) + \dot{d}(x,t)\right\| \geq 0, \tag{14}$$

where $\bar{\beta} \in R_{>0}$.

Let's rewrite dynamic equation (10) as:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_{n+1}(t) \end{bmatrix} = \begin{bmatrix} f(x(t)) + \Delta f(x(t)) + d(x,t) \\ 0 \end{bmatrix} + \begin{bmatrix} bu(t) \\ \Psi(t) \end{bmatrix}, \tag{15}$$

where $\Psi(t) = \dot{\Theta}(x,t)$ is an auxiliary input, $x_{n+1}(t) = \Theta(x,t)$ is an auxiliary state, where $\Theta(x,0) = -\vartheta_1 e(0)/\vartheta_2$.

From (11) time derivatives of sliding surface is defined as:

$$\dot{S}\ (t) = \vartheta_1 \dot{x}(t) - \vartheta_1 \dot{x}_d(t) + \vartheta_2 \dot{\Theta}(x,\ t), \tag{16}$$

and,

$$\ddot{S}(t) = \vartheta_1 \ddot{x}(t) - \vartheta_1 \ddot{x}_d(t) + \vartheta_2 \ddot{\Theta}(x, t). \tag{17}$$

In this study, the terminal manifold is defined as,

$$\delta(t) = \Gamma \dot{S}^{p\times q^{-1}}(t) + S(t), \tag{18}$$

where $\Gamma \in R_{>0}$, and $p,\ q$ are odd constants that satisfy constraint $1 < p \times q^{-1} < 2$.

Theorem 1 *The control input* (19) *guarantees the finite-time global stability of dynamic system* (10) *while the optimal criteria* (20) *is satisfied.*

$$u(t) = \frac{-\vartheta_2}{\vartheta_1 b} \times \Psi(t) + \int\limits_0^t \frac{q}{\vartheta_1 b p} \dot{S}^{1-p\times q^{-1}}(t)(-\dot{S}(t)) - b^{-1} \times (\ddot{x}_d(t) + \dot{f}(x(t)) + \\ (\overset{\frown}{\beta}(t) + \beta_0)\mathrm{sgn}(\delta(t)) \left\| x(t) \right\|)dt, \tag{19}$$

and,

$$J = \int_0^{\infty} \{R\Psi^2(\tau) + (X^T(\tau)Q(X(\tau))X(\tau))\}d\tau, \tag{20}$$

and,

$$\dot{\hat{\beta}}(t) = \{p \times q^{-1}\}\Gamma \dot{S}^{\{p\times q^{-1}-1\}}(t)\vartheta_1 \left\| \delta(t) \right\| \left\| x(t) \right\|, \tag{21}$$

where $\hat{\beta}(t)$ is an adaptation term, $X(t) = [x(t), \Theta(t)]$ is the auxiliary state vector, Γ, β_0, $R \in R_{>0}$, and $Q(X(t))$ is the positive diagonal matrix. Then, Ψ is expressed as follows,

$$\Psi(t) = R^{-1} B^T \Upsilon X(t), \tag{22}$$

where Υ can be deduced from solving the Riccati equation as follows,

$$A^T(X(t))\Upsilon + \Upsilon A(X(t)) - \Upsilon B R^{-1}(X(t)) B^T \Upsilon = -Q(X(t)), \tag{23}$$

where $B = [0_{n\times 1}, 1]^T \in \mathbb{R}^{(n+1)\times 1}$, $A(X(t)) = [\Xi_{n\times n+1}(t), 0_{1\times n+1}] \in \mathbb{R}^{(n+1)\times(n+1)}$. Note that $\Xi_{n\times n}(t)$ can be derived from following equation,

$$\dot{X}(t) = A(X(t))X(t) + B\Psi(t). \tag{24}$$

Due to functionality of the presented approach to deal with the uncertain terms, and converging states in the closed region $\|x(t)\| < r,\ r \in R_{>0}$, in formulation (24), the uncertain term is ignored.

Now, we are in the position to present the event-based controller by providing Theorem 2.

Theorem 2 *Consider an uncertain nonlinear dynamic system (10). Then, control input (25) under event-triggering rule (27) guarantees ultimately global stability of the closed-loop response. Note that under the presented methodology the minimum inter-sampling time, i.e., $\bar{\Delta}$ satisfies inequality (26).*

$$\begin{aligned} u(t) = & -b^{-1}{\vartheta_1}^{-1}\vartheta_2\Psi(t_k) - \int_{t_k}^{t} {\vartheta_1}^{-1}b^{-1}\Gamma^{-1}qp^{-1}\dot{S}^{2-pq^{-1}}(t_k) + b^{-1}(\dot{f}(x(t_k)) - \ddot{x}_d(t) \\ & + (\overset{\frown}{\beta}\,(t_k) + \beta_0)\,\|x(t_k)\|\, sgn(\delta(t_k)))dt, \end{aligned} \tag{25}$$

and,

$$\bar{\Delta} \leq \frac{1}{L_0}\ln\left(\frac{1}{\frac{k_2}{L_0}(\frac{1}{\bar{\lambda}}+1)} + 1\right), \tag{26}$$

where $k_2 = \max\{\left\|{\vartheta_1}^{-1}\vartheta_2 R^{-1}B^T\Upsilon\right\|_\infty, 1\}$ for all $t \in \left[t_k, t_{k+1}\right)$. The event-triggering rule is defined as,

$$\|E(t)\| - \ \bar{\lambda} \times \{\|X(t)\| + \ (k_0 + k_1)\} \leq 0, \tag{27}$$

where $\bar{\lambda}$ is a positive constant. Triggering-error is described by $\mathrm{E}(t) = x(t_k) - x(t)$ for all $t \in \left[t_k, t_{k+1}\right)$, and $k_0 = \|\dot{x}_d(t)\|_\infty + \|\Delta f(x) + d(x,t)\|_\infty, k_1 = \max(|\varphi_2 + \varphi_1|)$ where $\varphi_1 = \int_{t_k}^{t}(\Gamma^{-1}{\vartheta_1}^{-1}\frac{q}{p}\dot{S}^{2-p/q}(t_k)dt, \varphi_2 = \int_{t_k}^{t}(\beta_0 + \overset{\frown}{\beta}\,(t_k))\,\|x(t_k)\|\, sgn(\delta(t_k)))dt.$

Corollary 1 *Criterion (26) specifies the required minimum inter-sampling-time, and maximum band-width to achieve the closed-loop finite-time stability.*

Proof There are two steps that can be taken to prove the presented theorem. In the first step, it is demonstrated that if (27) holds, then the system dynamic (10) under control input (25) achieves finite-time stability. Then, it is shown that (26) holds under the presented control law and event-triggering rule.

Step 1: We start with the Lyapunov function below

$$V(t) = 0.5 \times {\beta_{\Gamma_0}}^{-1}(\tilde{\beta}^2(t) + \delta^2(t)), \tag{28}$$

where for all $t \in R_{>0}$, $\beta_{\Gamma_0} = \max\{(\tilde{\beta}^2(t) + \delta^2(t))\}$ that guarantees $\|V(t)\| \leq 1$, and $\tilde{\beta}(t) = \overset{\frown}{\beta}\,(t) - \bar{\beta}$. Then, one can obtain:

$$\dot{V}(t) = \delta(t)(\dot{S}(t) + \Gamma p \times q^{-1}\ddot{S}(t)\dot{S}^{\{p\times q^{-1}\}-1}(t)\,) + (\tilde{\beta}(t)\dot{\overset{\frown}{\beta}}(t)), \tag{29}$$

where $x_d(t)$ is a known desired signal for all $t \in R_{>0}$. Then, from (29) and (25) one can obtain,

$$\begin{aligned} &\dot{V}(t) \leq {\beta_{\Gamma_0}}^{-1}\delta(t)(\left\|\{p\times q^{-1}\Gamma\}\dot{S}^{\{p\times q^{-1}\}-1}(t)\right\|(\vartheta_1(\dot{d}(t) + \dot{\Delta}f(x(t)) + \dot{f}(x(t)) \\ &-(\dot{f}(x(t_k)) + (\overset{\frown}{\beta}\,(t_k) + \beta_0)\,\|x(t_k)\|\, sgn(\delta(t_k))) \\ &+\vartheta_2(\ddot{\Theta}(t) - \ddot{\Theta}(t_k)) + q\{p\Gamma\vartheta_1\}^{-1}(\dot{S}^{2-p\times q^{-1}}(t) - \dot{S}^{2-p\times q^{-1}}(t_k))) + {\beta_{\Gamma_0}}^{-1} \times \dot{\overset{\frown}{\beta}}(t)\tilde{\beta}(t). \end{aligned} \tag{30}$$

By considering Assumption 11, one can obtain,

$$\begin{aligned}&\dot{V}(t) \leq \beta_{\Gamma_0}{}^{-1}\delta(t)\left\|\Gamma pq^{-1}\dot{S}^{\{p\times q^{-1}\}-1}(t)\right\|((\vartheta_1 L_0\left\|E(t)\right\| + \vartheta_1\bar{\beta}\left\|x(t)\right\| \\ &-\vartheta_1(\beta_0 + \overset{\frown}{\beta}(t_k))\left\|x(t_k)\right\| sgn(\delta(t_k)) + \vartheta_2\ \ddot{\Theta}(t) + q\{p\Gamma\}^{-1}\|(\dot{S}^{2-p\times q^{-1}}(t) - \dot{S}^{2-p\times q^{-1}}(t_k))\|) \\ &\quad+\beta_{\Gamma_0}{}^{-1}\dot{\overset{\frown}{\beta}}(t)\tilde{\beta}(t).\end{aligned} \tag{31}$$

A new auxiliary variable β_e is defined as follows,

$$\beta_e = \vartheta_1 L_0 \left\|E(t)\right\|, \tag{32}$$

where λ, ϑ_1 are chosen small values such that $(\bar{\beta} + \beta_0)\left\|\delta(t)\right\|\left\|x(t_k)\right\| \gg \beta_e$. Note that according to (21) one can obtain $\hat{\beta}(t) > \beta(t_k)$, $\tilde{\beta}(t) < 0$ for all $t \in [t_k, t_{k+1})$, then,

$$\begin{aligned}&\dot{V}(t) \leq -\underbrace{\beta_{\Gamma_0}{}^{-1}\left\|\{p\times q^{-1}\Gamma\}\dot{S}^{\{p\times q^{-1}\}-1}(t)\right\|(-\beta_e + (\beta_0 + \bar{\beta})\left\|x(t_k)\right\|}_{\mathbb{Z}_0} \\ &\underbrace{+\left|\tilde{\beta}(t)\right|\left\|x(t)\right\| - q\times\{p\Gamma\}^{-1}\|(\dot{S}^{2-\{p\times q^{-1}\}}(t) - \dot{S}^{2-\{p\times q^{-1}\}}(t_k))\| - \|\ddot{\Theta}(t)\|\vartheta_2)}_{\mathbb{Z}_0}\left\|\delta(t)\right\| \\ &-\underbrace{\beta_{\Gamma_0}{}^{-1}\left\|q^{-1}p\Gamma\dot{S}^{\{p\times q^{-1}\}-1}(t)\right\|\left\|x(t_k)\right\|\left\|\delta(t)\right\|}_{\mathbb{Z}_1}\left\|\tilde{\beta}(t)\right\|),\end{aligned} \tag{33}$$

then,

$$V\dot{(}t) \leq -\beta_\Gamma V^{0.5}(t), \tag{34}$$

where $\beta_\Gamma = \min\left\{\sqrt{2}\mathbb{Z}_0, \sqrt{2}\mathbb{Z}_1\right\}$.

Step 2: The closed-loop dynamic system under the proposed triggering rule and control effort for all $t \in [t_k, t_{k+1})$ can be expressed as follows,

$$\begin{aligned}\dot{x}(t) = {}&\Delta f(x) + d(x,t) + f(x(t)) + b(-\vartheta_2\vartheta_1{}^{-1}b^{-1}\Psi(t_k) \\ &+\int_0^t[-\Gamma^{-1}\vartheta_1{}^{-1}b^{-1}\frac{q}{p}\dot{S}^{2-p/q}(t_k) \\ &-b^{-1}(\dot{f}(x(t_k)) - \ddot{x}_d(t) + (\overset{\frown}{\beta}(t_k) + \beta_0)\left\|x(t_k)\right\| sgn(\delta(t_k)))]dt,\end{aligned} \tag{35}$$

where $\{x(t_k)|k \in N_0\}$ represents the transmitted data at time-instances t_k in which triggering-rule is satisfied.

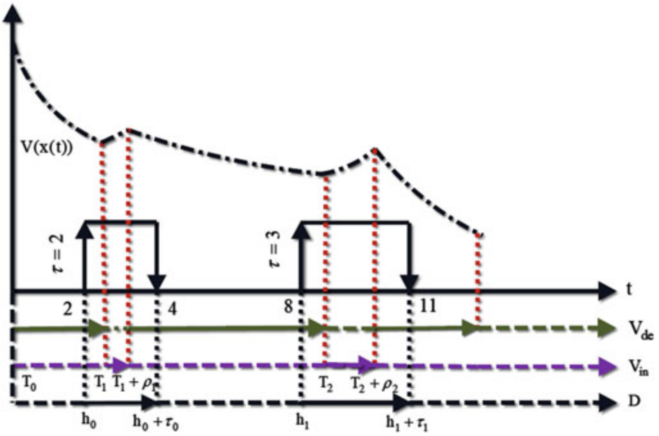


Fig. 3 Typical evolution of Lyapunov function under cyber attack, and event-triggering activation. In this example, $n(1, 9) = 2, n(1, 3) = 1, n(2.5, 9) = 1$ and $D(8, 12) = 3, h = \{2, 8\}, D(1, 3) = 1$, and $H_1 = [2, 4)$

$$\begin{aligned}\dot{x}(t) = \Delta f(x) + d(x, t) + f(x(t)) + b(-\vartheta_2{\vartheta_1}^{-1}b^{-1}\Psi(t_k)\\ + \int_0^t -\Gamma^{-1}{\vartheta_1}^{-1}b^{-1}qp^{-1}\dot{S}^{2-\{p\times q^{-1}\}}(t_k) - b^{-1}(\dot{f}(x(t_k)))\\ -\ddot{x}_d(t) + (\beta_0 + \widehat{\beta})\,\|x(t_k)\|\, sgn(\delta(t_k)))dt.\end{aligned} \tag{36}$$

From (36) and (27), we have,

$$\begin{aligned}\dot{\mathrm{E}}(t) = \Delta f(x) + d(x, t) + f(x(t)) - f(x(t_k)) + \dot{x}_d(t) - {\vartheta_1}^{-1}\vartheta_2\Psi(t_k)\\ -\int_{t_k}^t \Gamma^{-1}{\vartheta_1}^{-1}\frac{q}{p}\dot{S}^{2-p/q}(t_k)dt - \int_{t_k}^t (\widehat{\beta}\ (t_k) + \beta_0)\,\|x(t_k)\|\, sgn(\delta(t_k)))dt,\end{aligned} \tag{37}$$

Considering $\varphi_1 = \int_{t_k}^t (\Gamma^{-1}{\vartheta_1}^{-1}\frac{q}{p}\dot{S}^{2-p/q}(t_k)dt, \varphi_2 = \int_{t_k}^t (\widehat{\beta}\ (t_k) + \beta_0)\,\|x(t_k)\|\, sgn(\delta(t_k)))dt$, and $k_1 = \max(|\varphi_2 + \varphi_1|)$ yield,

$$\dot{\mathrm{E}}(t) \le L_0\,\|E(t)\| - {\vartheta_1}^{-1}\vartheta_2\Psi(t_k) + k_0 + k_1, \tag{38}$$

then,

$$\|E(t)\| \le {L_0}^{-1}(\left\|\vartheta_2{\vartheta_1}^{-1}R^{-1}B^T\Upsilon\right\|_\infty \|X(t_k)\|_\infty + k_0 + k_1) \times \int_{t_k}^t e^{L_0(t-\tau)}d\tau. \tag{39}$$

Now, with considering $\Im(t) = [E(t), 0]$, $k_2 = \max\{1, \left\|{\vartheta_1}^{-1}\vartheta_2 R^{-1}B^T\Upsilon\right\|_\infty\}$, and $X(t_k) = \Im(t) + X(t)$, Eq. (39) can be represented as,

$$\|\Im(t)\| \leq {L_0}^{-1} k_2(\|\Im(t)\| + \|X(t)\| + k_0 + k_1)(e^{L_0 \bar{\Delta}} - 1), \tag{40}$$

In regards to $\bar{\Delta}$, (40) is monotonically increasing. By taking the stability of $X(t)$ into account, and considering $\|E(t)\| = \|\Im(t)\|$, one can obtain,

$$\bar{\Delta} \leq {L_0}^{-1} \ln(\{\frac{k_2}{L_0}(\frac{1}{\bar{\lambda}} + 1)\}^{-1} + 1). \tag{41}$$

This completes the proof.

2.2 Denial-of-Service Attacks

Section 2.1 presents the stability analysis of the proposed event-based terminal SMC approach. In this section, the effect of the DoS cyber attack on the closed-loop stability is expressed, and the explicit relation between the malicious attack properties, i.e., cyber-attacks duration and frequency, with the closed-loop stability is derived.

To interpret cyber attacks' frequency and duration, the following two assumptions regarding the parameters of the DoS attack are taken into consideration without compromising generality [51].

Assumption 6 The following conditions are true for every $\tau, t \in R_{>0}$:

$$n(\tau, t) \leq n_0 + \frac{t - \tau}{\tau_D}, \tag{42}$$

where $\tau_D, n_0 \in R_{>0}$.

Assumption 7 The following conditions are true for every $\tau, t \in R_{>0}$:

$$D(\tau, t) \leq d_0 + \frac{t - \tau}{T}, \tag{43}$$

where $T, d_0 \in R_{>0}$.

In order to clarify the DoS attacks time period, H_n is expressed as,

$$H_n := [h_n \; h_n + \tau_n[\bigcup_{n \in N_0} \{h_n\}, \tag{44}$$

In the DoS attacks occurrence curve, as depicted in Fig. 3, h_n, $n \in N_0$ represents attack initiation instances, and τ_n represents DoS attacks duration, By considering (43), and (44) one can conclude,

$$D(\tau, t) = \bigcup_{n \in N_0} \{[\tau, t[\bigcap H_n\}. \tag{45}$$

Having made that distinction, we also need to find two time-based clusters different viewpoints. The first cluster illustrates the time duration in which a DoS attack occurs, the second group illustrates the time duration in which (27) is violated.

$$A[\tau, t) = [\tau, t)\backslash D[\tau, t), \tag{46}$$

Using $A(\tau, t)$ and $D(\tau, t)$, the time axis is divided into separate clusters, in which DoS attack occurs and in which it does not. In order to clarify the second group, it is necessary to define two sets V_{in} and V_{de}. These sets are used to show the time instances at which (27) does not hold due to a DoS attack and the time stamps at which (27) remains valid.

$$V_{in} = \bigcup_{i \in N_0} [T_i, T_i + \rho_i)\,, \tag{47}$$

and,

$$V_{de} = \bigcup_{i \in N_0} \left[T_i + \rho_i, T_{i+1}\right)\,. \tag{48}$$

In order to determine whether (27) holds true or not in the domain $[t_1, t_2)$, the following sets are defined, respectively,

$$\bar{A}(t_1, t_2) = V_{dec} \bigcap [t_1, t_2), \tag{49}$$

and,

$$\bar{D}(t_1, t_2) = V_{in} \bigcap [t_1, t_2), \tag{50}$$

Assumption 8 It is assumed that no attack occurs at the beginning of communication, thus $T_0 = \rho_0 = 0$.

Figure 3 describes the groups and sets that are proposed. It is now critical to deriving constraints on the parameters of the DoS attack with respect to their impact on V_{in} and V_{de}, as will be discussed in the following subsection.

2.3 Stability and Cyber Attacks

As mentioned in the previous subsections, satisfying (26), and (27) guarantee the closed-loop stability. It is now possible to investigate the situation in which (26), (27) are not valid due to malicious attacks. The solution of (34) is expressed by

$$V(t) \leq V(0)\, e^{-\beta_r t}. \tag{51}$$

Now from (33) we have,

$$\begin{aligned}\dot{V}(t) \leq &-\underbrace{{\beta_{\Gamma_0}}^{-1}\left\| pq^{-1}\Gamma\dot{S}^{\{p\times q^{-1}\}-1}(t)\right\| [((\bar{\beta}+\beta_0)\left\|x(t_k)\right\| - \vartheta_2\|\ddot{\Theta}(t)\|)}_{\chi_3}\\ &\underbrace{+\left|\tilde{\beta}(t)\right| \left\|x(t)\right\| - \{p\Gamma\}^{-1}\times q\|(\dot{S}^{2-p\times q^{-1}}(t)-\dot{S}^{2-p\times q^{-1}}(t_k))\|]}_{\chi_3}\left\|\delta(t)\right\|\\ &-\underbrace{{\beta_{\Gamma_0}}^{-1}\left\|p\Gamma\times q^{-1}\dot{S}^{p/q-1}\right\|\left\|\delta(t)\right\|\left\|x(t_k)\right\|}_{\chi_4}\left\|\tilde{\beta}(t)\right\|\\ &+{\beta_{\Gamma_0}}^{-1}\vartheta_1 L_0\left\|p\Gamma q^{-1}\dot{S}^{\{p\times q^{-1}\}-1}\right\|\left\|E(t)\right\|\left\|\delta(t)\right\|,\end{aligned} \tag{52}$$

with defining $\beta_{\Gamma_2} = \min\left\{\sqrt{2}\chi_3, \sqrt{2}\chi_4\right\}$, $\beta_{\Gamma 3} = \max(\frac{\vartheta_1 L_0}{\beta_{\Gamma_0}}\left\|\Gamma\frac{p}{q}\dot{S}^{p/q-1}\right\|\left\|\delta(t)\right\|)$, and considering $\left\|V(t)\right\| \leq 1$ one can obtain,

$$\dot{V}(t) \leq \beta_{\Gamma 3}\left\|E(t)\right\| - \beta_{\Gamma_2}V(t). \tag{53}$$

The solution of (53) can be restated as,

$$V(t) \leq \beta_{\Gamma 3}\times{\beta_{\Gamma_2}}^{-1}\left\|E(t)\right\| + exp\{-\beta_{\Gamma_2}t\}V(0), \tag{54}$$

By having (27), (51) can be expressed for all $t \in V_{de}$ as,

$$V(t) \leq V(T_i+\rho_i)exp\{-\beta_\Gamma(t-T_i-\rho_i)\}, \tag{55}$$

while (27) does not hold, for all $t \in V_{in}$, (51) results

$$V(t) \leq exp\{-\beta_{\Gamma_2}(t-T_i)\}V(T_i) + \beta_{\Gamma 3}{\beta_{\Gamma_2}}^{-1}\left\|E(t)\right\|, \tag{56}$$

By considering Assumptions 6, and 7 one can obtain,

$$\begin{aligned}&V(t) \leq exp\{-\beta_\Gamma\left|\bar{A}(0,t)\right| - \beta_{\Gamma_2}\left|\bar{D}(0,t)\right|\}\\ &+\beta_{\Gamma 3}\times{\beta_{\Gamma_2}}^{-1}\left\|E(t)\right\|\ (1+\sum_{i\,\in\,N_2 t\,>\,T_i}(exp\{-\beta_\Gamma\left|\bar{A}(T_i,t)\right| - \beta_{\Gamma_2}\left|\bar{D}(T_i,t)\right|\})),\end{aligned} \tag{57}$$

then,

$$\begin{aligned}\left\|E(\tau,t)\right\|_{\max} &< (\alpha\ +k_0)(n_0+\tfrac{t-T_i}{\tau_D})(d_0+\tfrac{t-T_i}{T})\\ &+(1+n)\bar{\Delta} \leq e^{(\alpha\ +k_0)(n_0+\frac{t-T_i}{\tau_D})(d_0+\frac{t-T_i}{T})+(1+n)\bar{\Delta}}.\end{aligned} \tag{58}$$

Considering (49), and (50) yield,

$$-\beta_\Gamma\left|\bar{A}(T_i,t)\right| - \beta_{\Gamma_2}\left|\bar{D}(T_i,t)\right| = (\beta_\Gamma - \beta_{\Gamma_2})\bar{D}(T_i,t) - \beta_\Gamma(t-T_i). \tag{59}$$

The upper bound of right-hand-side of (57) should be specified, then

$$1+\sum_{\substack{i\in N_2\\ t>T_i}} exp\{-\beta_\Gamma \left|\bar{A}(T_i,t)\right| - \beta_{\Gamma_2}\left|\bar{D}(T_i,t)\right|\} \le 1+\sum_{\substack{i\in N_2\\ t>T_i}} e^{\varpi}, \tag{60}$$

where,

$$\varpi = (\beta_\Gamma - \beta_{\Gamma_2})\bar{D}(T_i,t) + (\alpha+\upsilon)(n)\bar{D}(T_i,t) + ((1+n)(\alpha+\upsilon)-\beta_\Gamma)\bar{\Delta}. \tag{61}$$

The stability is guaranteed if $\varpi < 0$, then

$$(\beta_\Gamma - \beta_{\Gamma_2})\bar{D}(T_i,t) + (\alpha+\upsilon)(n)\bar{D}(T_i,t) < (\beta_\Gamma - (\alpha+\upsilon)(1+n))\bar{\Delta}, \tag{62}$$

finally,

$$\bar{D}(T_i,t) < \frac{(\beta_\Gamma - (\alpha+\upsilon)(1+n))}{(\beta_\Gamma - \beta_{\Gamma_2}) + (n)(\alpha+\upsilon)}\bar{\Delta}. \tag{63}$$

2.4 Results and Discussion

The presented resilient AOFTSMC is validated in this section. To achieve this goal, a numerical example is provided as follows. Consider an uncertain nonlinear dynamic system as,

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= -x_1^2(t) + x_2(t) + \Delta f(x(t)) + d(x,t) + u(t).\end{aligned} \tag{64}$$

where $x_1(t), x_2(t)$ are system states, $u(t)$ expresses the control input, unknown system dynamic is defined by $\Delta f(x(t)) = 0.1x_1(t)$, and $d(x,t) = \sin(t)$ represents the disturbance applied to the system. From (33), and (52) one can obtain that $\beta_\Gamma \approx \beta_{\Gamma_2}$ if β_0, ϑ_2 are chosen appropriate values. Then, inequality (63) can be simplified as,

$$\bar{D}(T_i,t) < \lambda_4\bar{\Delta}, \tag{65}$$

where $\lambda_4 = \frac{(\beta_\Gamma-(1+n)(\alpha+\upsilon))}{(\alpha+\upsilon)(n)}$. Consider $\vartheta_1 = 1$, $\beta_0 = 10$, $\vartheta_2 = 0.1$, $\bar{\lambda} = 0.0001$, $q = 5$, $p = 3$, $\alpha = 2$, $k_2 = 1$ and $L_0 = 2$. Then, minimum inter-sampling time $\bar{\Delta} = 9.8 \times 10^{-5}$ is derived through (41). Since (41) decreases as k_2 increases, the worst-case scenario for k_2 is considered during simulation. As illustrated in Fig. 5, none of the inter-sampling duration is less than 0.0001, which verifies the proposed criterion (41). In order to calculate the maximum length of an endurable malicious attack, it would be reasonable to assume that $t - \tau = 1$/s and $n = 1$. It is obvious that increasing n results in decreasing the affordable duration of time that a given attack can last.

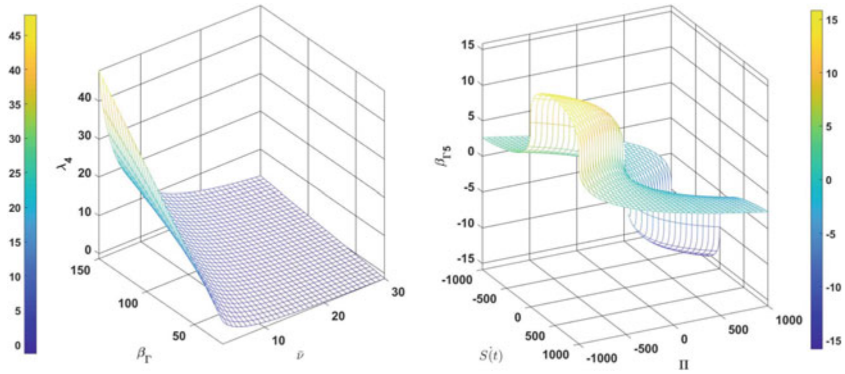


Fig. 4 Left: A numerical analyses of λ_4 and the effect of β_Γ and $\bar{\upsilon} = \alpha + \upsilon$ variations on a tolerable DoS attack's duration are presented. Right: Numerical variations of $\beta_{\Gamma_5} = \dot{S}^{2-p\times q^{-1}}(t) - \dot{S}^{2-p\times q^{-1}}(t_k)$, $\Pi = \dot{S}(t) - \dot{S}(t_k)$ due to the DoS attacks is depicted

Table 1 Variance and mean values of states under cyber-attack

states	DoS attack duration per second (%)	Γ	[Mean/Variance]
$[x_1(t), x_2(t)]$	5	10	[0.014/0.11, −0.01/0.04]
$[x_1(t), x_2(t)]$	10	100	[0.02/0.01, −0.01/0.05]

Figure 5 shows that when states converge to zero, the sampling rate is decreased. Note that when states are stabilized, the inter-sampling time is increased since the event-triggering mechanism is not activated. Figure 6 shows the response of the auxiliary state $x_3(t)$ as well as the augmented input Ψ. Contrary to some studies such as [3, 40, 52] where the focus is on linear dynamic systems, the proposed formulation concerns with the standard nonlinear systems. Also, by specifying the least possible inter-sampling time, it provides the maximum required network bandwidth.

The effect of β_Γ and $\bar{\upsilon} = \upsilon + \alpha$ variations is shown in Fig. 4. Increasing the value of β_Γ results in an increasing convergence rate that increases the endurable cyber-attacks' duration. Meanwhile, the tolerability under malicious DoS attacks during V_{in} is improved while $\bar{\upsilon}$ decreases.

Simulation results are presented in Table 1 under various DoS attack duration with different values of Γ and β_Γ. As shown in this table, the stability is preserved under various conditions.

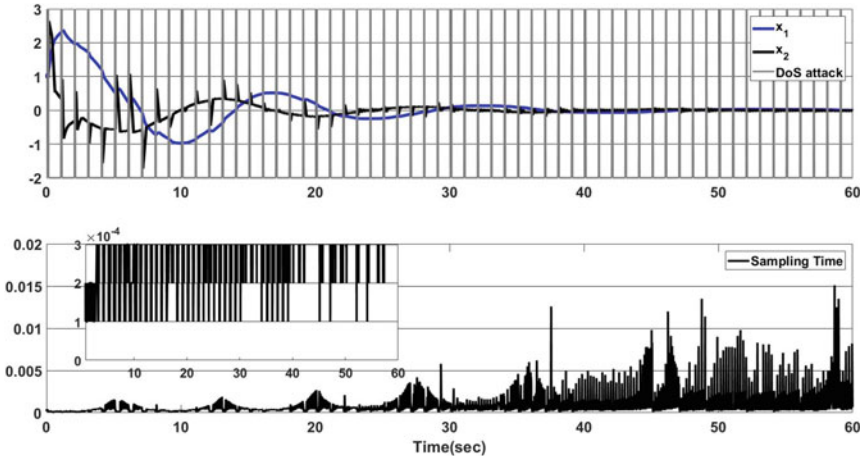


Fig. 5 Simulation results of system in (64) using the control effort (25) and under the cyber attacks. Top: System responses in closed-loop with $x(0) = [1\,,\,1]$. Bottom: inter-sampling times that are produced using the proposed event-triggering method

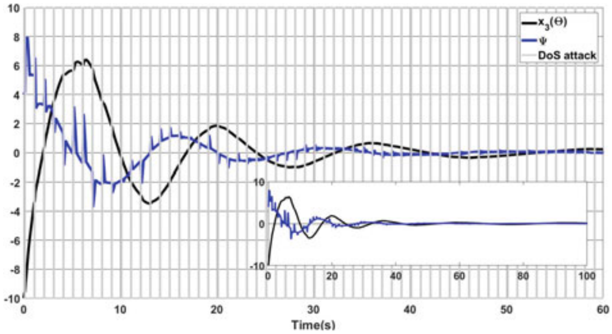


Fig. 6 Evolution of $\Psi(t)$ and $x_3(t)$ using the control effort in (25) and under the malicious cyber-attacks

3 Quantized Event-Triggered Terminal SMC Design for Uncertain Input-Delayed Dynamic

As outlined in Sects. 1 and 2, an overview of event-based approaches, their challenges, and their effects are described in detail. This section examines the effects of quantizers in closed-loop systems with event-based terminal SMC. There is a strong relationship between an event-triggering rule and quantized signal during the development of CPSs, from both a theoretical as well as a practical perspective. Keeping control precision and stability is one of the most significant objectives when a physical system is communicating digitally, with certain considerations and constraints. For instance, the maximum number of bits that can be transmitted through a data-packet over a network layer [22], quantization error [23], and sensitivity adjustment in dynamic quantizers are all critical considerations and limitations that must be considered.

This paper presents a novel design approach for dynamic quantizers that considers event-tagging rules for uncertain input-delayed nonlinear dynamic systems. In contrast, most studies centered on the development of quantization methodologies for certain linear dynamics [24, 26] or they focused on determining the continuous-time evolving of quantizer parameters.

3.1 Problem Formulation

Consider an uncertain input-delayed nonlinear system as follows,

$$\dot{x}(t) = \; f(x(t)) + \Delta f(x(t)) + bu(t - D) + d(x, t), \tag{66}$$

where $x(t) = [x_1(t), \ldots, x_n(t)] \in \mathbb{R}^n$ is the state vector with initial condition $x(0) = x_0$, uncertain dynamics are defined by $\Delta f(x(t)) \in \mathbb{R}^n$, and $d(x, t) \in \mathbb{R}$ shows bounded external disturbance that satisfies $\|d(x, t)\| \leq \bar{d}$, where $\bar{d} \in R_{>0}$. A control input signal is shown by $u(t) \in \mathbb{R}^n$, $b \in R_0$ is a constant parameter, and $D \in R_0$ is the known input delay.

Assumption 9 The dynamic system (66) does not show any finite escape time behavior $t \in [0, D)$ [53].

Assumption 10 $x_d(t)$, $\dot{x}_d(t)$, $\ddot{x}_d(t) \in \mathcal{L}^\infty$ holds for all $t \in R_{\geq 0}$.

Assumption 11 $f(x(t))$, $\dot{f}(x(t))$ satisfy Lipchitz criterion as follows,

$$\left\| \dot{f}(x(t_2)) - \dot{f}(x(t_1)) \right\| \leq L_0 \left\| x(t_2) - x(t_1) \right\|, \tag{67}$$

where $L_0 \in R_{>0}$.

Assumption 12 Following inequality holds by $f(x(t))$ for all $t \in R_{\geq 0}$,

$$\bar{\alpha} - \| f(x(t)) \|_\infty \geq 0, \tag{68}$$

where $\bar{\alpha} \in R_{>0}$.

Assumption 13 A bound is established on the deviation rate of uncertain dynamic and external disturbance inputs as follows, $\Delta f(x(t))$ and $d(x, t)$ satisfy the following condition in $t \in R_{\geq 0}$,

$$\bar{\beta} \left\| x(t) \right\| - \left\| \dot{\Delta} f(x(t)) + \dot{d}(x, t) \right\| \geq 0, \tag{69}$$

where $\bar{\beta} \in R_{>0}$.

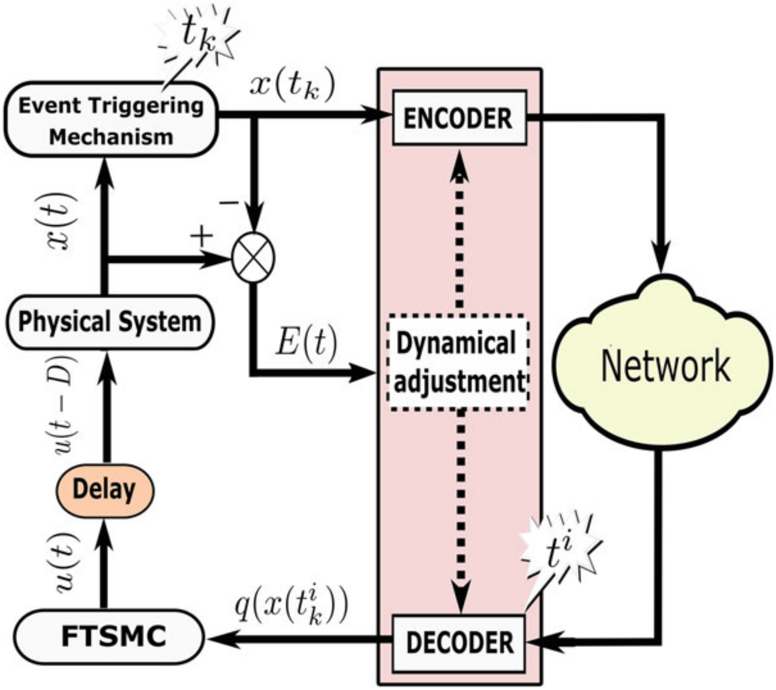


Fig. 7 Block diagram representation of the proposed scheme

It is essential for the design procedure methodology to be guided by the presented assumptions. Lipchitz's criteria are generally satisfied by most physical systems. In the meantime, engineers possess the knowledge of the boundaries of uncertainty terms and disturbances through their practical experience. Then, the proposed assumptions are not restrictive.

It is the purpose of the present section to demonstrate robust scheme for uncertain systems, which use the network layer to convey measurement data despite input delays and quantizers. Accordingly, this study adopts a quantized event-trigger SMC. Figure 7 illustrates the block diagram of the proposed scheme.

According to Fig. 7, the event-trigger mechanism updates the measured states in the time stamps according to the predefined rules that are outlined in the remainder of this section. After being quantized by a dynamic quantizer, measurement data are transmitted through the network layer as part of the digital communication layer. Quantization involves mapping time-varying continuous time signals into piecewise constant signals. The dynamic quantizer described in this study is as follows [24, 54],

$$q_\tau(z(t)) = \tau q(z(t)\tau^{-1}), \tag{70}$$

where $\tau \in R_{>0}$ is adjusted in each quantization time stamp, $z(t)$ is a continues- time signal, and $q(.)$ is defined as follows,

$$\|q(z(t)) - z(t)\| < \mu\,,\ if\ \ \|z(t)\| < M, \tag{71a}$$

$$\|q(z(t))\| > M - \mu,\ if\ \ \|z(t)\| > M, \tag{71b}$$

where $M \in R_{>0}$ is the quantizer saturation level, and $\mu \in R_{>0}$ is quantization error bound parameter. Vector quantization consists of dynamic quantization of each component of the vector, i.e. $q_\tau(Z(t)) = [q_\tau(z_1(t)), q_\tau(z_2(t)), \ldots, q_\tau(z_n(t))]$ where $Z(t) \in \mathbb{R}^{1\times n}$, and a piece-wise constant signal will be created for each component with a value in the set as $\mathbb{N} = \{-\lceil M\rceil\, \tau(t),\ (-\lceil M\rceil + 1)\tau(t), \ldots, (\lceil M\rceil + 1)\tau(t),\ \lceil M\rceil\, \tau(t)\}$. $x(t_k)$ is produced by the event-triggering rule, and passed through an encoder, then transmitted through the network layer. Then measurement data are decoded at time stamps $t^i, i \in N_0$. Finally, the designed FTSMC controller uses $x(t_k^i)$ to produce a control effort signal. $x(t_k^i)$ is defined as:

$$x(t_k^i) = q_{\mu^i(t)}(x(t)), \tag{72}$$

where $t \in \{(t_k, t_{k+1}) \bigcap (t^i, t^{i+1})\}$ and $t_k, k \in N_0$ are triggering time instances. Then, t_{k+1} is expressed as follows.

$$t_{k+1} = \inf\{t \in (t_k, \infty),\ k \in N\}. \tag{73}$$

Note that $t^i, i \in N_0$ start counting in the duration of $[t_k, t_{k+1})$, i.e. $t_0 < t^0 < t^1 < \cdots < t_1 < t^0 < t^1 < \cdots < t_2, \ldots$ According to (70) and (71), $e_k^i(t)$ in the time domain $t \in \{(t_k, t_{k+1}) \bigcap (t^i, t^{i+1})\}$ is expressed as,

$$e_k^i(t) = x(t_k^i) - x(t_k) \tag{74}$$

From (71) and (74), one can obtain,

$$\left\|e_k^i(t)\right\| \le \mu_{t^i}\,,\ if\ \ \|x(t_k)\| < M, \tag{75a}$$

$$\left\|e_k^i(t)\right\| > M - \mu_{t^i},\ if\ \ \|x(t_k)\| > M. \tag{75b}$$

The main conclusion that can be drawn from this subsection is that states are sampled at time instances $t_k, k \in N_0$, and when relaxation time arrives, i.e. $[t_k, t_{k+1})$, states' quantization time instances are represented by stamp t^i, $i \in N_0$. The next subsection presents the controller design approach.

3.2 Control Methodology

The aim of this section is to develop an adaptive fast terminal SMC controller that guarantees finite-time convergence stability is proposed.

Theorem 3 *Consider a dynamic system* (66) *while* $D = 0$. *A control signal (76) grantees the finite-time asymptotically stability.*

$$\begin{aligned} u(t) = -\int_0^t [qp^{-1}{\alpha_1}^{-1} b^{-1}\dot{S}^{2-p/q}(t) + b^{-1}(-\ddot{x}_d(t) + \dot{f}(x(t)) + (\widehat{\beta}\,(t)+ \\ + \beta_0)\,\|x(t)\|\,\operatorname{sgn}(\delta(t)))]dt, \end{aligned} \tag{76}$$

where $S(t)$ is the sliding surface and is considered as,

$$S(t) = \alpha_1 e(t), \tag{77}$$

where $\alpha_1 \in R_{>0}$. $\delta(t)$ is the terminal manifold and is represented as,

$$\delta(t) = k_2 S(t) + k_1 \dot{S}^{p\times q^{-1}}(t), \tag{78}$$

where $k_1 \in R_{>0}$. Tracking error is expressed by $e(t) = x(t) - x_d(t)$, and p, q are two odd numbers that satisfy $1 < \frac{p}{q} < 2$. Finally, adaptation law is defined as follows,

$$\dot{\widehat{\beta}}(t) = k_1 pq^{-1}\dot{S}^{\{p\times q^{-1}\}-1}(t)\,\|x(t)\|\,\|\delta(t)\|\,\alpha_1, \tag{79}$$

where $\widehat{\beta}\,(0) = \beta_0$ is the initial condition and satisfies $\beta_0 \geq \bar{\beta}/2$.

Proof of Theorem 3. A Lyapunov function candidate is chosen as:

$$V(t) = 0.5\{\,\delta(t)^T\delta(t) + \tilde{\beta}^T\tilde{\beta}(t)\}, \tag{80}$$

where adaptation error is considered as $\tilde{\beta}(t) = \widehat{\beta}\,(t) - \bar{\beta}$. From (80), (78) one can obtain,

$$\begin{aligned} \dot{V}(t) = \delta(t)^T\big[\dot{S}(t) + k_1\alpha_1 pq^{-1}\dot{S}^{\{p\times q^{-1}-1\}}(t)\times \\ \big\{\dot{f}(x(t)) + \Delta\dot{f}(x(t)) + \dot{d}(x,t) + b\dot{u}(t) - \ddot{x}_d\big\}\big] + \tilde{\beta}(t)^T\dot{\widehat{\beta}}(t). \end{aligned} \tag{81}$$

Substitution $u(t)$ from (76) yields,

$$\begin{aligned}\dot{V}(t) = \delta(t)\Bigg[& k_1\alpha_1 p \times q^{-1}\dot{S}^{\{p\times q^{-1}-1\}}(t)\times \\ &\Bigg\{\Delta\dot{f}(x(t)) + \dot{d}(x,t) - (\widehat{\beta}(t) + \beta_0)\left\|x(t)\right\| sgn(\delta(t))\Bigg\}\Bigg] + \tilde{\beta}(t)\dot{\widehat{\beta}}(t).\end{aligned} \tag{82}$$

From Assumption 13 and (79) one can obtain,

$$\begin{aligned}&\dot{V}(t) \le k_1\alpha_1\frac{p}{q}\dot{S}^{p/q-1}(t)\left\|\delta(t)\right\|\{\bar{\beta}\left\|x(t)\right\| - (\widehat{\beta}(t) + \beta_0)\left\|x(t)\right\| + \\ &(\widehat{\beta}(t) - \bar{\beta})\left\|x(t)\right\| + \left\|\tilde{\beta}(t)\right\|\left\|x(t)\right\| - \left\|\tilde{\beta}(t)\right\|\left\|x(t)\right\|\}.\end{aligned} \tag{83}$$

Then,

$$\begin{aligned}\dot{V}(t) \le -&\underbrace{\left\{k_1\alpha_1 pq^{-1}\dot{S}^{p\times q^{-1}-1}(t)(\beta_0 - \left\|\tilde{\beta}(t)\right\|)\left\|x(t)\right\|\right\}}_{\xi_1}\left\|\delta(t)\right\| \\ -&\underbrace{\left\{k_1\alpha_1 p\times q^{-1}\dot{S}^{p/q-1}(t)\left\|x(t)\right\|\right\}}_{\xi_2}\left\|\tilde{\beta}(t)\right\|,\end{aligned} \tag{84}$$

then,

$$\begin{aligned}\dot{V}(t) &\le -\min\{\sqrt{2}\xi_1, \sqrt{2}\xi_2\}\frac{1}{\sqrt{2}}\left[\left\|\tilde{\beta}\right\| + \left\|\delta(t)\right\|\right] \\ &\le -\min\{\sqrt{2}\xi_1, \sqrt{2}\xi_2\}V(t)^{1/2}.\end{aligned} \tag{85}$$

3.3 Quantized Event-Triggered Control Design

Compared to the time-triggered mechanism, the event-based approach reduces computation load and provide resilient behavior while data-packets are lost through the network layer. As stated in the introduction section, measurement data are updated when triggering criteria is satisfied at time stamps t_k. Consider event-triggering error as follows,

$$E(t) - x(t) + x(t_k) = 0, \tag{86}$$

where $E(t_k) = 0$ for all $k \in N_0$. The quantized event-based fast terminal SMC is designed for an uncertain input-delayed nonlinear dynamic (66) in this subsection. Consider the sliding surface in the following manner,

$$S(t) = \alpha_1 e(t) + \alpha_2\Theta(t), \tag{87}$$

where $\alpha_2 = b\alpha_1$, $\alpha_1 \in R_{>0}$, where $\Theta(t) = \int\limits_{t-D}^{t} u(\tau)\,d\tau$. Note that initial condition of $\Theta(t)$ is zero, and the terminal sliding manifold is defined according to (78).

Lemma 1 *The area* Ω *in which* $sgn(S(x(t_k^i))) \neq sgn(S(x(t_k)))$ *is the infinitesimally closed region.*

Proof Consider the quantizer that is defined by (70) and (71), where one can obtain $sgn(x(t_k^i)) = sgn(x(t_k))$. Consider (86), (90), and suppose that the last successful transmitted measurement data is $x(t_k)$. Now, suppose that $x(t)$, $t \in \left[t_k, t_{k+1}\right)$ tends toward zero, then, consider t^* represents time instance in which $x(t^*) = 0$, then,

$$\lim_{x(t)\to 0} \|E(t)\| = 0. \tag{88}$$

$\|x(t)\| > 0$ for $t^* < t$ and according to (90), triggering rule is then activated, and $sgn(x(t_k) = sgn(x(t))$. As a result, considering (87) yields $sgn(S(x(t_k^i))) = sgn(S(x(t)))$ in $t \in \left[t_k, t_{k+1}\right)$.

This completes the proof.

Theorem 4 *Consider the dynamic system* (66) *under triggering-rule* (90)*, and dynamic quantizer* (72)*. Then, the control effort* (89) *guarantees globally uniformly ultimately bounded (GUUB) stability.*

$$\begin{aligned} u(t) = -\int\limits_{t^i}^{t} & {k_1}^{-1}{\alpha_1}^{-1}b^{-1}k_2\frac{q}{p}\dot{S}^{2-p/q}(t_i^k) + b^{-1}(\dot{f}(x(t_k^i)) \\ & - \ddot{x}_d(t) + (\bar{\beta} + \beta_0)\left\|x(t_k^i)\right\| \operatorname{sgn}(\delta(t_k^i)))dt. \end{aligned} \tag{89}$$

Note that $u(t) = 0$ for $t \in [0, D)$. Triggering- rule is defined as,

$$\|E(t)\| - \lambda \|x(t)\| \leq 0, \tag{90}$$

for $t \in \{(t_k, t_{k+1}) \bigcap (t^i, t^{i+1}) | i, k \in N_0\}$, $\delta(t_k^i) = q_{\mu^i(t)}(\delta(x(t_k)))$, $\dot{S}(t_k^i) = q_{\mu^i(t)}$ $(\dot{S}(x(t_k)))$, $\overline{\beta} \in R_{>0}$.

Proof of Theorem 4. Lyapunov function candidate is expressed as,

$$V(t) = 0.5\delta(t)^T\,\delta(t). \tag{91}$$

The first time derivation of sliding surface yields,

$$\dot{S}(t) = \alpha_1\dot{x}(t) - \alpha_1\dot{x}_d(t) + \alpha_2\dot{\Theta}(x, t), \tag{92}$$

then,

$$\ddot{S}(t) = \alpha_1(\dot{f}(x(t)) + \Delta\dot{f}(x(t)) + \dot{d}(x, t) + b\dot{u}(t - D)) - \alpha_1\ddot{x}_d(t) + \alpha_2(\dot{u}(t) - \dot{u}(t - D)). \tag{93}$$

From (91), and (78) we have,

$$\dot{V}(t) = \delta(t)^T (k_2 \dot{S}(t) + k_1 \frac{p}{q} \ddot{S}(t) \dot{S}^{p/q-1}(t)\,). \tag{94}$$

Taking (93), and (94) into consideration yields,

$$\begin{aligned} \dot{V}(t) &= \delta(t)^T \{p \times q^{-1}\} k_1 \dot{S}^{\{p \times q^{-1}\}-1}(t) \\ &\quad \times \left\{ q \times \{pk_1\}^{-1} k_2 \dot{S}^{2-p\times q^{-1}}(t) + b\alpha_1 \dot{u}(t) + \alpha_1 \varGamma \right\}, \end{aligned} \tag{95}$$

where $\varGamma = \left[\dot{f}(x(t)) + \dot{\varDelta} f(x(t)) + \dot{d}(t) - \ddot{x}_d(t)\right]$. From (89) and (95) one can conclude,

$$\begin{aligned} \dot{V}(t) \le\; & \delta(t)^T \alpha_1 k_1 p q^{-1} \dot{S}^{\{p\times q^{-1}-1\}}(t) \{\dot{f}(x(t)) + \dot{d}(t) + \dot{\varDelta} f(x(t)) - (\dot{f}(x(t_k^i)) \\ & + (\beta_0 + \bar{\beta}) \left\| x(t_k^i) \right\| sgn(\delta(t_k^i))) \\ & + qk_2 \{k_1 \alpha_1 p\}^{-1} \left\| (\dot{S}^{2-pq^{-1}}(t) - \dot{S}^{2-pq^{-1}}(t_k^i)) \right\| \}. \end{aligned} \tag{96}$$

Then, according to Assumptions 11 and 13 we have,

$$\begin{aligned} \dot{V}(t) \le\; & \delta(t)^T \alpha_1 k_1 p q^{-1} \dot{S}^{p\times q^{-1}-1}(t) \{L_0 \left\| x(t) - x(t_k^i) \right\| + \bar{\beta} \left\| x(t) \right\| \\ & - (\beta_0 + \bar{\beta}) \left\| x(t_i^k) \right\| sgn(\delta(t_k^i)) \\ & + qk_2 \{k_1 \alpha_1 p\}^{-1} \left\| (\dot{S}^{2-p\times q^{-1}}(t) - \dot{S}^{2-p\times q^{-1}}(t_k^i)) \right\| \}, \end{aligned} \tag{97}$$

from Eqs. (74), (86), and Lemma 1 one can obtain,

$$\begin{aligned} & \dot{V}(t) \le \delta^T(t) \alpha_1 k_1 p q^{-1} \dot{S}^{p\times q^{-1}-1}(t) \{L_0(\left\| e_k^i(t) - E(t) \right\|) + \bar{\beta} \left\| x(t) \right\| \\ & - (\beta_0 + \bar{\beta}) \left\| x(t_k^i) \right\| sgn(\delta(t_k)) + qk_2 \{k_1 \alpha_1 p\}^{-1} \left\| (\dot{S}^{2-pq^{-1}}(t) - \dot{S}^{2-pq^{-1}}(t_k^i)) \right\| \}. \end{aligned} \tag{98}$$

Consider $\left\| e(t_k^i) \right\| \le \bar{k}_\mu \left\| E(t) \right\|, \lambda \left\| x(t) \right\| - \left\| E(t) \right\| \ge 0$ for all $t \in [t_k, t_{k+1})$ then

$$\begin{aligned} \dot{V}(t) \le\; & k_1 p q^{-1} \| \dot{S}^{p\times q^{-1}-1}(t) \| \alpha_1 \{(L_0(\bar{k}_\mu + 1)\lambda + \bar{\beta}) \left\| x(t) \right\| \left\| \delta(t) \right\| \\ & - (\bar{\beta} + \beta_0) \left\| x(t_k^i) \right\| \delta(t) sign(\delta(t_k)) + \frac{qk_2}{k_1 \alpha_1 p} \left\| (\dot{S}^{2-p/q}(t) - \dot{S}^{2-p/q}(t_k^i)) \right\| \}. \end{aligned} \tag{99}$$

Now, we need to find $(\bar{k}_\mu + 1)\lambda$ in some way that $L_0(\bar{k}_\mu + 1)\lambda + \bar{\beta}) \left\| x(t) \right\| - (\bar{\beta} + \beta_0) \left\| x(t_k^i) \right\| < 0$ holds. Recalling $\left\| x(t_i^k) - x(t) \right\| - (\bar{k}_\mu + 1)\lambda \left\| x(t) \right\| \le 0$ yields,

$$((L_0(\bar{k}_\mu + 1)\lambda + \bar{\beta}) + (\bar{\beta} + \beta_0)(\bar{k}_\mu + 1)\lambda - (\bar{\beta} + \beta_0)) \left\| x(t) \right\| \le 0, \tag{100}$$

then,

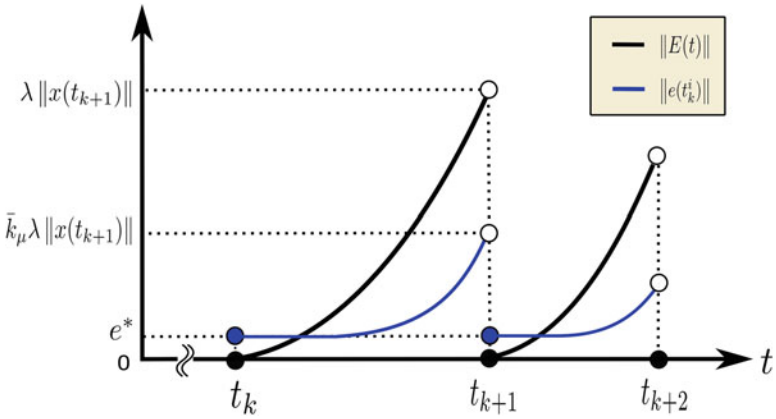


Fig. 8 Evolution of event-triggered and quantization errors in terms of their conceptual behavior. t_k, t_{k+1} and t_{k+2} are time instances in which triggering-rule is satisfied. Based on the ADC convert resolution, e^* describes the minimum possible quantization error

$$(k_\mu + 1)\lambda < \frac{\beta_0}{L_0 + (\bar{\beta} + \beta_0)}. \tag{101}$$

Finally,

$$\dot{V}(t) \leq \frac{(-\Upsilon + \Lambda)}{\sqrt{2}} V(t)^{0.5}, \tag{102}$$

where $\Lambda = k_2 \|\dot{S}^{p/q-1}(t)\| \left\| (\dot{S}^{2-p/q}(t) - \dot{S}^{2-p/q}(t_k^i)) \right\|$, and $\Upsilon = k_1 \frac{p}{q} \|\dot{S}^{p/q-1}(t)\| \alpha_1 \times \left| (\bar{k}_\mu + 1)\lambda - \frac{\beta_0}{L_0 + (\bar{\beta} + \beta_0)} \right|$. Λ is bounded according to Proposition 1. Note that k_2 is chosen to satisfy $\Lambda \ll \Upsilon$. Figure 8 illustrates a concept of how even-triggering and quantization errors evolve over time.

Proposition 1 $\|\Lambda\| < k_2(\bar{\alpha} + \|\Delta f(x) + d(x,t)\| + \|x_d(t)\|)$ *holds for all* $t \in R_{>0}$, *and* $\bar{\alpha}$ *is described through Assumption 12.*

Proof See the proof in the Appendix.

Considering (78), $\delta(t) = 0$ yields $\dot{S}(t) = 0$, and $S(t) = 0$ for $t \geq t^*$, where t^* defines time instance that holds $S(t^*) = 0$. Considering (87), and (66) for all $t \geq t^*$ control signal can be derived as follows after some simplification steps,

$$u(t) = b^{-1}(\dot{x}_d(t) - f(x(t)) - \Delta f(x(t)) - d(x,t)). \tag{103}$$

According to (103) and Assumption 12 one can obtain,

$$\|u(t)\| \leq b^{-1}(\|\dot{x}_d(t)\| + \bar{\alpha} + \bar{\beta}_1), \tag{104}$$

where $\bar{\beta}_1 = \|\Delta f(x) + d(x,t)\|_\infty$. Considering definition of $\Theta(t)$ yields,

$$\|\Theta(t)\| \leq Db^{-1}(\|\dot{x}_d(t)\| + \alpha + \bar{\beta}_1), \tag{105}$$

Finally, considering (87), and $S(t) = 0$, following inequality can be concluded:

$$\|e(t)\| \leq D(\|\dot{x}_d(t)\| + \alpha + \bar{\beta}_1). \tag{106}$$

3.4 Inter-sampling Time Calculation

The computation of minimum inter-sampling and cycle time for a dynamic quantizer are presented in this section.

Assumption 14 For all $t \in \{[t^i, t^{i+1}) | i \in N \ \& \ t^{i+1} \leq t_{k+1}\}$, $e(t_k^i)$ is considered slow varying.

By considering $\Theta(t)$, Eq. (66) can be expressed as,

$$\dot{x}(t) = f(x(t)) + d(x,t) + \Delta f(x(t)) - b\dot{\Theta}(t) + bu(t). \tag{107}$$

Equation (107) can be rewritten in accordance with (87) as follows,

$$\dot{x}(t) = \frac{\alpha_2}{\alpha_2 + \alpha_1}[f(x(t)) + d(x,t) + \Delta f(x(t)) - b{\alpha_2}^{-1}\dot{S}(t) - \alpha_1{\alpha_2}^{-1}\dot{x}_d(t) + bu(t)], \tag{108}$$

Under the presented controller, event-triggered error dynamics can be derived by considering Assumption 14 as follows.

$$\begin{aligned}\dot{E}(t) = \frac{\alpha_2}{\alpha_2 + \alpha_1}\Bigg\{ & f(x(t)) - f(x(t_k^i)) + d(x,t) + \Delta f(x(t)) + (1 - \alpha_1{\alpha_2}^{-1})\dot{x}_d(t) - b{\alpha_2}^{-1}\dot{S}(t) \\ & - \int_{t_k}^{t} {k_1}^{-1}{\alpha_1}^{-1}b^{-1}\frac{q}{p}\dot{S}^{2-p/q}(t_k)dt - \int_{t_k}^{t} (\bar{\beta} + \beta_0) \left\|x(t_k^i)\right\| sgn(\delta(t_k^i)))dt\Bigg\}.\end{aligned} \tag{109}$$

Then, from (109) and recalling Assumptions 9, 10, 11, one can obtain

$$\left\|\dot{E}(t)\right\| \leq \frac{\alpha_2}{\alpha_2 + \alpha_1}\left\{L_0(k_\mu + 1)\left\|E(t)\right\| + \bar{\beta}_1 + \left\|(1 - \frac{\alpha_1}{\alpha_2})\dot{x}_d(t)\right\| + b\frac{1}{\alpha_2}\left\|\dot{S}(t)\right\| + \Gamma\right\}, \tag{110}$$

where $\Gamma = \sup\left\{\int_{t_k}^{t} (\bar{\beta} + \beta_0)\left\|x(t_k^i)\right\| dt, t \in [t_i, t_{i+1})\right\}$, and $\bar{\beta}_1 = \|\Delta f(x(t)) + d(x,t)\|_\infty$. Then,

$$\|E(t)\| \leq \left\{\bar{\beta}_1 + \left\|(1 - \alpha_1{\alpha_2}^{-1})\dot{x}_d(t)\right\| + b\frac{1}{\alpha_2}\left\|\dot{S}(t)\right\| + \Gamma\right\} \times \frac{(e^{\gamma\Delta} - 1)}{\gamma}, \tag{111}$$

where $\gamma = \frac{L_0\alpha_2}{\alpha_2+\alpha_1}(k_\mu + 1)$ and $\bar{\Delta} = \sup\{t_{k+1} - t_k | k \in N_0\}$. Note that (111) is monotonically increasing regard to $\bar{\Delta}$. Considering Proposition 1, and (111) can be restate as,

$$\|E(t)\| \leq \frac{1}{L_0(k_\mu+1)}\left\{\bar{\beta}_1 + \left\|(1-\frac{\alpha_1}{\alpha_2})\dot{x}_d(t)\right\| + \Pi + \Gamma\right\} \times (e^{\gamma\bar{\Delta}}-1), \tag{112}$$

where $\Pi = b\frac{1}{\alpha_2}\left\|\dot{S}(t)\right\|$. Then,

$$\lambda\,\|x(t)\| \leq \left\{\bar{\beta}_1 + \left\|(1-\frac{\alpha_1}{\alpha_2})\dot{x}_d(t)\right\| + \Pi + \Gamma\right\} \times \frac{(e^{\gamma\bar{\Delta}}-1)}{L_0(k_\mu+1)}. \tag{113}$$

Ultimately, $\bar{\Delta}$ is expressed as,

$$\lambda\left(\|x(0)\| + \|x_d(t)\| + \left\{\bar{\alpha}+\bar{\beta}+\bar{\beta}_1\right\}D\right) \leq \frac{(e^{\gamma\bar{\Delta}}-1)}{L_0(k_\mu+1)}\left\{\bar{\beta}_1 + \left\|(1-\frac{\alpha_1}{\alpha_2})\dot{x}_d(t)\right\| + \Pi + \Gamma\right\}. \tag{114}$$

Finally,

$$\bar{\Delta} \geq \frac{1}{\gamma} \times \ln\left(\frac{L_0\lambda(k_\mu+1)(\|x(0)\| + \|x_d(t)\| + \left\{\bar{\alpha}+\bar{\beta}_1\right\}D)}{(\bar{\beta}_1 + \left\|(1-\frac{\alpha_1}{\alpha_2})\dot{x}_d(t)\right\| + \Pi + \Gamma)} + 1\right). \tag{115}$$

It is now possible to specify the sensitivity of the dynamic quantizer that has been designed. To this end, t_q is defined so that to insure $e(t_k^i)$ satisfies the predefined constraint in the duration of $\bar{\Delta} = \min\{t_{k+1} - t_k | k \in N_0\}$. From the predefined criterion $e(t_k^i) \leq \bar{k}_\mu \|E(t)\|$ for all $t \in [t_k, t_{k+1})$ one can obtain,

$$\mu(t^{i+1}) = \min\left\{\frac{\bar{k}_\mu\|E(t)\| - \mu(t^0)}{\bar{\Delta}}t, \bar{k}_\mu\|E(t)\|\right\}, \tag{116}$$

where the lowest encoder/decoder resolution is expressed by $\mu(t^0) = \underline{\mu}$, and

$$t^{i+1} - t^i \leq \frac{\bar{\Delta}}{2}, \tag{117}$$

for all $t \in [0, t_{k+1} - t_k)$. To prevent the dynamic quantizer from saturation, M is selected to satisfy $M\mu(t^i) \geq \|x(0)\| + \|x_d(t)\| + (\bar{\alpha}+\bar{\beta}_1)D$ for $i \in N_0$.

3.5 Numerical Simulations

The numerical simulations demonstrating the effectiveness of an event-trigger FTSMC that is coupled with a dynamic quantizer. Moreover, the theorems and assumptions proposed are discussed in more detail. In the following example, we consider an open-loop unstable dynamic system,

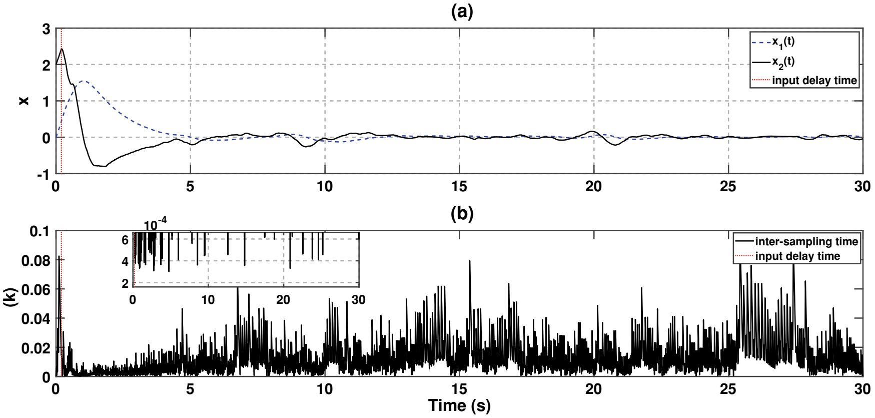


Fig. 9 Under the control effort (66), and in the presence of a dynamic quantizer, simulation results are depicted. **a** shows the closed-loop state responses. **b** the inter-sampling time evolution

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -x_1^2(t) + x_2(t) + \Delta f(x(t)) + d(x,t) + u(t-D),
\end{aligned}
\tag{118}
$$

where $u(t)$ represents the control input, and $x_1(t)$, $x_2(t)$ are system states with initial condition $X(0) = [0, 2]$, $\Delta f(x(t)) = \sin(x_1(t))$ and $d(x,t) = \sin(t)$ represent the dynamic uncertainty and external time-varying disturbance. respectively. Also, a delay of $D = 0.2\,\mathrm{s}$ is considered for the nonlinear system. The Sliding surface dynamic is considered as $S(t) = x_1(t) + x_2(t) + \Theta(t)$. Designed parameters are $p = 7,\ q = 5, k_2 = k_1 = 1, \lambda = 0.001, \bar{k}_\mu = 0.1$. To assess the efficacy of the proposed method the simulation results are plotted in Fig. 9. In accordance with the figure below, when the states of the system are approaching zero, we can observe that the intervals of updating are growing. Zeno-free behavior is verified according to Fig. 9b, where $\bar{\Delta} \approx 2.8 \times 10^{-4}$.

Figure 10 shows the evolution of the quantizer's sensitivity. Compared with prior studies such as [3, 12], the maximum accuracy is guaranteed under the proposed scheme whenever tracking error converges to zero.

As can be seen in some studies such as [24] the dynamic quantizer sensitivity evolves over time, which can lead to decreased effectiveness of the methodology after some time has passed. According to this study, the dynamic treatment of the quantizer is determined by the event-triggering error that preserves performance. Evolution of control effort is shown in Fig. 11.

The proposed methodology is verified in Fig. 12, where the evolution of sliding surface is depicted. Figure 13 demonstrates the dynamic behavior of event-triggering error that is formulated by (90).

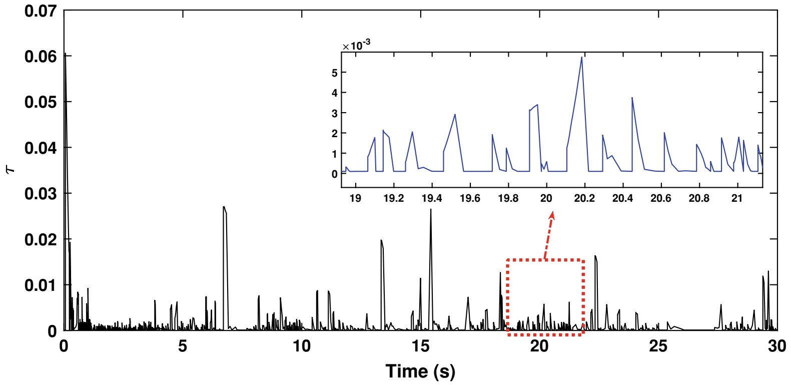


Fig. 10 Quantizer's sensitivity evolution with $\underline{\mu} = 10^{-4}$

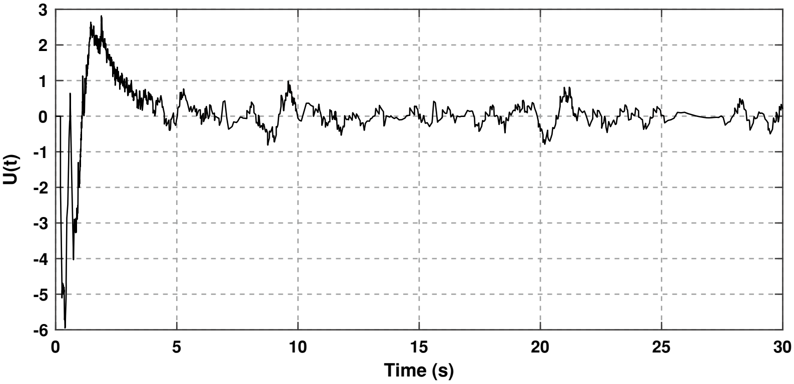


Fig. 11 Control effort $U(t) = u(t - 0.2)$

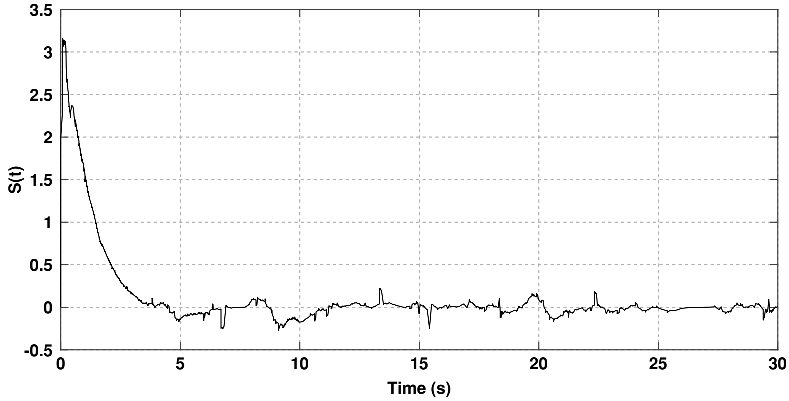


Fig. 12 Sliding surface dynamic, i.e., $S(t)$

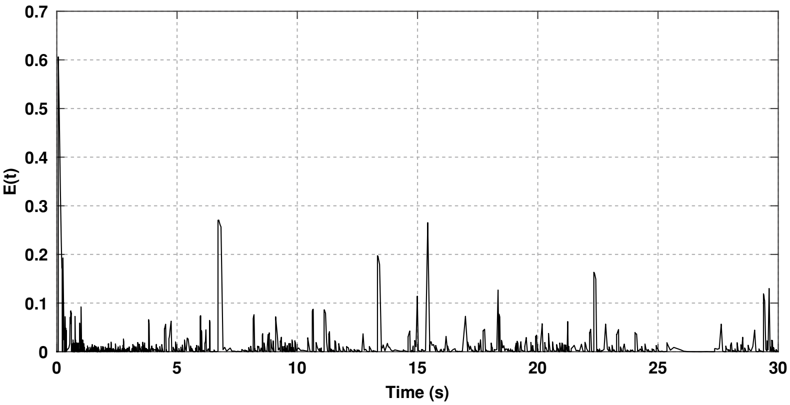


Fig. 13 Event-triggering error dynamic, i.e., $E(t)$

4 Conclusions

It is the purpose of this article to examine the problem of cyber-based control of mobile robots, and what problems it poses. During the first step, a review of wheeled mobile robot dynamic equations, as well as definitions and challenges of cyber-physical systems is discussed. Then, to address the two main challenges of CPSs, i.e., the limitation of communication resources and cyber security, an event-triggered resilient robust control method for a class of uncertain nonlinear dynamic systems is developed, and explicit relationships between malicious attack properties, such as duration and frequency, and controller design parameters are presented. After that, to cope with two other CPSs' challenges, i.e., communication delay and quantization error, an even-based robust control scheme by having a dynamic quantizer for a class of uncertain input-delayed nonlinear dynamic systems is proposed, and novel criteria to design the dynamic quantizer parameters is presented.

Appendix

Proof of Proposition 1. The assumptions made in (9), (11) indicate that system states obey Lipchitz continuity and do not exhibit any finite-escape time behavior. Meanwhile, due to the slow-varying dynamic of $x(t_k^i)$ in the $t \in [t_k, t_{k+1})$, and having stabilizable dynamic outside the region $\|E(t)\| \leq \lambda\, \|x(t)\|$, then one can obtain

$$k_2\left\|\dot{S}(t)\right\| - k_2\|\dot{S}^{pq^{-1}-1}(t)\|\left\|(\dot{S}^{2-pq^{-1}}(t) - \dot{S}^{2-pq^{-1}}(t_k^i))\right\| \geq 0. \tag{119}$$

Now, suppose that $x(t)$ diverges the maximum rate of within the minimum inter-sampling time period.

Now, it is supposed that $x(t)$ diverges with its maximum rate, that would occur during a period when no control inputs are being applied to the system, i.e., $t \in [0, D)$. Then, from (66), (92), and Assumption 13 one can obtain

$$\Lambda < k_2\alpha_1 \{\bar{\alpha} + \|\Delta f(x) + d(x,t)\| + \|x_d(t)\| + \|\dot{x}_d(t)\|\}. \tag{120}$$

References

1. Montazeri A, Ekotuyo J (2016) Development of dynamic model of a 7DOF hydraulically actuated tele-operated robot for decommissioning applications. In: 2016 American control conference (ACC), pp 1209–1214
2. Montazeri A, West C, Monk SD, Taylor CJ (2017) Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm. Int J Control 90(4):661–683
3. Yan Y, Shuanghe Yu, Xinghuo Yu (2019) Quantized super-twisting algorithm based sliding mode control. Automatica 105:43–48
4. Claudio DP, Pietro T (2016) Networked control of nonlinear systems under denial-of-service. Syst Control Lett 96:124–131
5. Jun H, Hongxu Z, Xiaoyang Yu, Hongjian L, Dongyan C (2019) Design of sliding-mode-based control for nonlinear systems with mixed-delays and packet losses under uncertain missing probability. IEEE Trans Syst Man Cybern Syst 51:3217–3228
6. Hossein K, Jafar Z, Roozbeh R-F (2019) Robust fault detection filter design for nonlinear networked control systems with time-varying delays and packet dropout. Circuits Syst Signal Process 38(1):63–84
7. Derui D, Qing-Long H, Yang X, Xiaohua G, Xian-Ming Z (2018) A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing 275:1674–1683
8. Shuai F, Ahmet C, Hideaki I, Pietro T, Claudio DP (2020) Networked control under dos attacks: trade-offs between resilience and data rate. IEEE Trans Autom Control 66:460–467
9. Renjie M, Peng S, Ligang W (2021) Active resilient control for two dimensional systems under denial of service attacks. Int J Robust Nonlinear Control 31(3):759–771
10. Xiaowei G, Tinggang J, Yugang N (2020) Consensus tracking for multi-agent systems subject to channel fading: a sliding mode control method. Int J Syst Sci 51(14):2703–2711
11. Xiao-Heng C, Rui H, Huanqing W, Liang L (2019) Robust design strategy of quantized feedback control. IEEE Trans Circuits Syst II Express Briefs 67(4):730–734
12. Yan Y, Shuanghe Yu, Changyin S (2020) Quantization-based event-triggered sliding mode tracking control of mechanical systems. Inf Sci 523:296–306
13. Saeedi M, Zarei J, Razavi-Far R, Saif M (2022) Event-based fast terminal sliding mode control design for a class of uncertain nonlinear systems with input delay: a quantized feedback control. J Vib Control. https://doi.org/10.1177/10775463211070901
14. Moreno JA (2009) A linear framework for the robust stability analysis of a generalized super-twisting algorithm. In: 6th international conference on electrical engineering, computing science and automatic control (CCE). IEEE, pp 1–6
15. Xu H, Sahoo A, Jagannathan S (2020) 5 joint scheduling and optimal adaptive event-triggered control of distributed cyber-physical systems. In: Principles of cyber-physical systems: an interdisciplinary approach, p 104
16. Narayanan V, Modares H, Jagannathan S, Lewis FL (2020) Event-driven off-policy reinforcement learning for control of interconnected systems. IEEE Trans Cybern
17. Bahreini M, Zarei J, Razavi-Far R, Saif M (2021) Robust and reliable output feedback control for uncertain networked control systems against actuator faults. IEEE Trans Syst Man Cybern Syst

18. Mohsen B, Jafar Z, Roozbeh R-F, Mehrdad S (2019) Robust finite-time stochastic stabilization and fault-tolerant control for uncertain networked control systems considering random delays and probabilistic actuator faults. Trans Inst Meas Control 41(12):3550–3561
19. Kumar V, Mohanty SR, Kumar S (2020) Event trigger super twisting sliding mode control for DC micro grid with matched/unmatched disturbance observer. IEEE Trans Smart Grid
20. Zhao H, Niu Y (2019) Finite-time sliding mode control of switched systems with one-sided Lipschitz nonlinearity. J Frankl Inst
21. Vignesh N, Hamidreza M, Sarangapani J (2021) Event triggered control of input affine nonlinear interconnected systems using multiplayer game. Int J Robust Nonlinear Control 31(3):950–970
22. Liu T, Jiang Z-P (2015) Quantized event-based control of nonlinear systems. In: 54th IEEE conference on decision and control (CDC). IEEE, pp 4806–4811
23. Xueyan X, Hongjun Y, Jinkun L, Shuquan W (2020) Vibration control of nonlinear three-dimensional length-varying string with input quantization. J Vib Control 26(19–20):1835–1847
24. Bo-Chao Z, Xinghuo Yu, Yanmei X (2018) Quantized feedback sliding-mode control: an event-triggered approach. Automatica 91:126–135
25. Bandyopadhyay B, Behera AK (2018) Event-triggered sliding mode control with quantized state measurements. Springer, pp 113–126
26. Yesmin A, Bera MK (2020) Design of event-based sliding mode controller with logarithmic quantized state measurement and delayed control update. ISA Trans
27. Renjie M, Peng S, Ligang W (2020) Dissipativity-based sliding-mode control of cyber-physical systems under denial-of-service attacks. IEEE Trans Cybern 51:2306–2318
28. Song G, Shi P, Agarwal RK (2021) Fixed time sliding mode cooperative control for multiagent networks via event triggered strategy. Int J Robust Nonlinear Control 31(1):21–36
29. Xianqing W, Kexin X, Meizhen L, Xiongxiong H (2020) Disturbance-compensation-based continuous sliding mode control for overhead cranes with disturbances. IEEE Trans Autom Sci Eng 17(4):2182–2189
30. Mustafa A, Dhar NK, Verma NK (2019) Event-triggered sliding mode control for trajectory tracking of nonlinear systems. IEEE/CAA J Autom Sin 7(1):307–314
31. Pengcheng C, Li Yu, Dan Z (2020) Event-triggered sliding mode control of power systems with communication delay and sensor faults. IEEE Trans Circuits Syst I Regul Pap 68:797–807
32. Saim A, Haoping W, Yang T (2019) Adaptive high-order terminal sliding mode control based on time delay estimation for the robotic manipulators with backlash hysteresis. IEEE Trans Syst Man Cybern Syst 51:1128–1137
33. Saleh M (2018) Adaptive global terminal sliding mode control scheme with improved dynamic surface for uncertain nonlinear systems. Int J Control Autom Syst 16(4):1692–1700
34. Qi W, Zong G, Zheng WX (2020) Adaptive event-triggered SMC for stochastic switching systems with semi-Markov process and application to boost converter circuit model. IEEE Trans Circuits Syst I Regul Pap 68:786–796
35. Bangxin J, Jianquan L, Yang L, Jinde C (2019) Periodic event-triggered adaptive control for attitude stabilization under input saturation. IEEE Trans Circuits Syst I Regul Pap 67(1):249–258
36. Li S, Ahn CK, Xiang Z (2020) Decentralized sampled-data control for cyber-physical systems subject to DOS attacks. IEEE Syst J
37. Yuan-Cheng S, Guang-Hong Y (2018) Periodic event-triggered resilient control for cyber-physical systems under denial-of-service attacks. J Franklin Inst 355(13):5613–5631
38. Saeedi M, Zarei J, Razavi-Far R, Saif M (2021) Event-triggered adaptive optimal fast terminal sliding mode control under denial-of-service attacks. IEEE Syst J
39. Bailing T, Jie C, Hanchen L, Lihong L, Qun Z (2020) Attitude control of UAVs based on event-triggered super-twisting algorithm. IEEE Trans Industr Inf 17:1029–1038
40. Singh P, Agrawal P, Nandanwar A, Behera L, Verma NK, Nahavandi S, Jamshidi M (2020) Multivariable event-triggered generalized super-twisting controller for safe navigation of non-holonomic mobile robot. IEEE Syst J 15:454–465
41. Shah DH, Patel DM (2019) Design of sliding mode control for quadruple-tank MIMO process with time delay compensation. J Process Control 76:46–61

42. Kolmanovsky I, Harris McClamroch N (1995) Developments in nonholonomic control problems. IEEE Control Syst Mag 15(6):20–36
43. Lamiraux F, Laumond J-P (1998) A practical approach to feedback control for a mobile robot with trailer. In: Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146), vol 4. IEEE, pp 3291–3296
44. Bloch AM, Harris McClamroch N (1989) Control of mechanical systems with classical nonholonomic constraints. In: Proceedings of the 28th IEEE conference on decision and control. IEEE, pp 201–205
45. Jessivaldo S, André C, Tito S, Humberto A (2018) Remote control of an omnidirectional mobile robot with time-varying delay and noise attenuation. Mechatronics 52:7–21
46. Gharajeh MS, Jond HB (2020) Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation. Robot Auton Syst 134:103669
47. Saleh AL, Hussain MA, Klim SM (2018) Optimal trajectory tracking control for a wheeled mobile robot using fractional order PID controller. J Univ Babylon Eng Sci 26(4):292–306
48. Sun Z, Ge SS, Huo W, Lee TH (2001) Stabilization of nonholonomic chained systems via nonregular feedback linearization. Syst Control Lett 44(4):279–289
49. Zheng X, Yuqiang W (2009) Adaptive output feedback stabilization for nonholonomic systems with strong nonlinear drifts. Nonlinear Anal Theory Methods Appl 70(2):904–920
50. Shihao S, Takahiro E, Fumitoshi M (2018) Iterative learning control based robust distributed algorithm for non-holonomic mobile robots formation. IEEE Access 6:61904–61917
51. Claudio DP, Pietro T (2015) Input-to-state stabilizing control under denial-of-service. IEEE Trans Autom Control 60(11):2930–2944
52. West C, Montazeri A, Monk SD, Taylor CJ (2016) A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator. IFAC-PapersOnLine 49(12):1261–1266
53. Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Springer
54. Daniel L (2003) Hybrid feedback stabilization of systems with quantized signals. Automatica 39(9):1543–1554

Path Optimization and Multi-level Path Planning for the Steam Field Navigation Algorithm

Hussein M. Fawzy, Hisham M. El-Sherif, and Gerd Baumann

Abstract The computational problem of path planning is considered as a core functionality for robotic systems. Physics-based path planning algorithms are developed to generate a path for a robotic agent between two points using a mathematical model as the main building block. In these physical algorithms, the field is not represented as a grid of cells or a set of nodes where a search technique is used to find the path, such as the case for combinatorial and sampling-based algorithms. The Stream Field Navigation (SFN) algorithm is a physical algorithm that is developed to solve the path planning problem by simulating a fluid field. One of the properties of paths generated by the SFN algorithm is that they may take a longer path than required due to the nature of the underlying physical model. To overcome this phenomena, this research proposes a method to optimize the path generated by the SFN algorithm. The proposed method consists of three algorithms that can be successively applied to a path with the aim of making it shorter. The results for the developed path optimization algorithms show an average of 14.5% reduction in path length compared to the original path generated by the SFN algorithm. Such contribution has a favorable effect on energy expenditure by systems using the SFN algorithm due to the direct relation between path length and system energy consumption. This research also proposes a method to allow the SFN algorithm to operate in multi-level environments. This addition enhances the standard SFN algorithm as it is designed for single-plane path planning. The results for the developed multi-level path planning approach show a comparison of the effectiveness of the SFN algorithm with other algorithms in these multi-level room-based environments where the SFN algorithm is shown to produce paths in less computational time and the paths are of higher quality.

H. M. Fawzy (✉) · H. M. El-Sherif · G. Baumann
German University in Cairo, New Cairo, Egypt
e-mail: hussein.fawzy.1@gmail.com

H. M. El-Sherif
e-mail: hisham.elsherif@guc.edu.eg

G. Baumann
e-mail: gerd.baumann@guc.edu.eg

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_7

Keywords Mobile robot · Path planning · Navier-Stokes · Fluid dynamics · Path optimization · Multi-level planning

1 Introduction and Background

Path planning is an important task that is carried out by a robot to find the shortest path—or more general, the optimal path—between two points. This optimal path may be the shortest path or may be the least dangerous for the robot. In essence, optimality is decided according to a certain evaluation criterion that an algorithm uses when finding a path between two points. The robot is of course required to be aware of the field it will be moving in. There has to be a representation of the environment in the computer for the planning algorithm to use.

The path planning algorithm requires that the map is defined. That is, the field boundaries, configuration, obstacles location and obstacle geometries are all known. These definitions can be obtained prior path planning used any mapping technique, such as Simultaneous Localization and Mapping [7, 15, 16].

Different categories of algorithms exist for the path planning problem. For example, there is the Combinatorial Planning techniques [27, 33], Sampling-Based Planning techniques [19, 24] and physical algorithms such as the Artificial Potential Field method (APF) [6, 18, 35] and the Stream Field Navigation algorithm [10, 11].

Algorithms in the combinatorial planning category have a unique property of being complete (A path will be found if it exists). On the other hand, increasing the dimensionality of the field space would cause these algorithms to be inefficient. Sampling-based techniques are not complete but are said to be probabilistically complete; i.e., the probability of finding a path approaches one if the algorithm keeps working without bounds. Artificial potential field methods are known to create local-minima in the field, a problem that causes a stagnation for the robotic agent. However, there are methods to solve the local-minima problem for potential fields [5]. One of these methods is used when the obstacles are cylindrical where fluid stream around these cylinders are calculated using specific potential functions to move around the robot without getting in a stagnation point [29]. Obstacle avoidance is also improved for potential field methods using optimization techniques [28, 31] and Machine learning techniques [23].

The algorithm of interest to this research is the Stream Field Navigation (SFN) algorithm [10]; an enhanced computational-physical path planning algorithm based on fluid stream equations and mechanics with modifications on how the numerical residuals are calculated to improve the computational efficiency. The SFN algorithm can also represent a stagnation-free field, which inherently solves the local-minima problem. There are however other improvements that can be added to the SFN algorithm.

The problem here is that the path calculated using the SFN algorithm may be in-efficient as the underlying fluid dynamics model tends to generate paths that are longer than required. This is a result of simulating the motion of fluid particles in a given medium. This research introduces another layer of efficiency, energy efficiency in this case, with a method to optimize the path generated by the SFN algorithm. The motivation behind this optimization is reducing the carbon footprint of robotic systems and the reductions of operational costs as they are both directly related to energy efficiency. One example can be seen in logistics and shipping where a reduction of one mile per driver per day translates to savings of up to $50 million per year [17]. In previous literature, path optimization is usually part of the path planning problem and not a stand-alone procedure. This makes it difficult to use the path optimization procedure with different path planning algorithms and sometimes even with updated versions of the same path planning algorithm. For example, path planning and optimization can be achieved using reinforcement learning [21] or fuzzy logic [14, 20]. However, the path optimization achieved in previous literature is not separable from path planning, making the optimization only useful to a certain path planning algorithm. The path optimization methodology introduced in this research solves this problem as it is a stand-alone algorithm that can be applied to general path generated by a path planning problem as will be seen later.

Another problem addressed in this research is that the standard SFN algorithm is limited to single plane (Or single level) operation. In multi-level environments, a robotic agent is navigating an environment consisting of multiple levels (floors) with different field boundaries and obstacle distributions for each level. The research also introduces an approach that can be used by the SFN algorithm to operate in multi-level environments. In use cases where the SFN is the algorithm of choice, navigating multi-level environments may be an important addition due to the increasing popularity of autonomous systems that are being deployed in residential buildings, shopping centers, etc. The multi-level path planning method introduced in this research solves this limitation and allows the SFN algorithm to operate in a wider range of situations and environments.

In the next section, the mathematical model and the fundamental SFN approach are explained. Section 3 explains the developed path optimization algorithms and their effectiveness in shortening a path while in Sect. 4, the multi-level path planning approach used by the SFN algorithm is introduced. The results for path optimization and multi-level path planning are included in their respective sections. Section 5 shows a developed ROS-Gazebo simulation environment that was used to simulate multiple aspects of the SFN algorithm followed by the conclusion (Sect. 6) and future work (Sect. 7).

Programming terminology and notations will appear in this book chapter as parts of the code will be explained when required to illustrate how a certain objective is achieved.

2 The Mathematical Model and the Fundamental SFN Approach

The mathematical model that governs the Stream Field Navigation algorithm is based on the Navier-Stokes fluid dynamics equations. The model starts with deriving the continuity equation using the Reynold's Transport Theorem, the Divergence Theorem and Leibniz's rule. A simplification to the equation is also given by assuming the fluid is in-compressible. This assumption greatly helps in simplifying the simulation and reducing the required computation to obtain the velocity vectors of a given field.

Conservation of momentum is then derived using Newton's second law while applying shear and normal stresses to the faces of an infinitesimal cube. Newton's law of viscosity is then used to relate the shear and normal stresses to the flow velocity, pressure and fluid viscosity. Using this derivation, the three equations for the three coordinate frame directions (x, y and z), known as the Navier-Stokes equations, are then formulated.

The equation for the x direction of an in compressible flow on $\Re^3$ is:

$$\rho a_x = \rho g_x - \frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{1}$$

This equation is then generalized to obtain the equations for the y and z directions. The terms in the equation represent the gravity force, pressure force and viscosity force where ρ is the density, a is the acceleration, g is the gravitational acceleration, p is the pressure and μ is the viscosity.

It has been proven that there exists a continuous solution for the Navier-Stokes equations on $R^3x[0, T]$ [32]. The proof is beyond the scope of this research. It however guarantees that a solution exists for a given field given that the domain of the time dimension is $[0, T]$ where $T < \infty$.

The systematic steps of the Stream Field Navigation algorithm are shown in Fig. 1. These steps start with initializing the field parameters such as the kinematic viscosity of the field, the inlet velocity, the mesh accuracy, and other essential parameters. For example, the Reynolds number $Re = \frac{uL}{v}$ is adjusted to keep the flow laminar (turbulence-free) (u is the velocity of the stream vector, v is the kinematic viscosity of the fluid and L is a characteristic linear dimension of the field). Setting the Reynold's number below the range from 10^5 to 10^6 is recommended for keeping the flow laminar [34].

The next step is to initialize the field geometry so that they can be mathematically manipulated. At this step the algorithm splits each obstacle into a set of segments whether this obstacle is polygonal or circular. This process is applied to the start point, goal point and field border so that they can also be mathematically manipulated.

The algorithm now discretizes the field by creating a field mesh. The accuracy of this mesh is controlled by the maximum edge length parameter of the algorithm where increasing the value yields a coarser mesh and decreasing it yields a finer mesh. An experiment is given here (Fig. 2) to show how the accuracy of the mesh

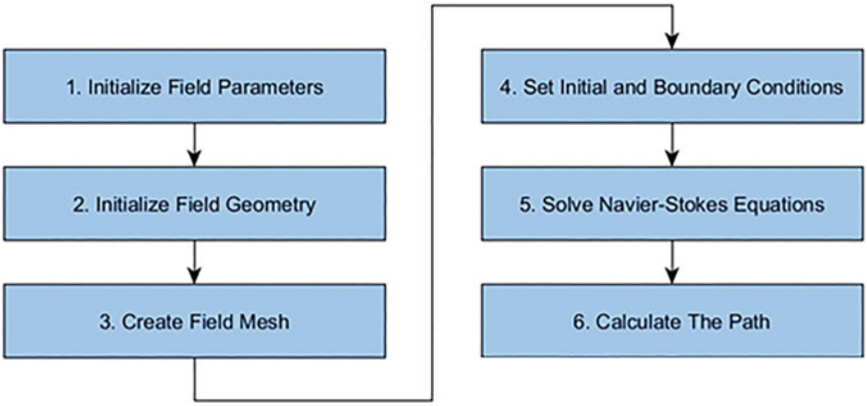

1. Initialize Field Parameters
2. Initialize Field Geometry
3. Create Field Mesh
4. Set Initial and Boundary Conditions
5. Solve Navier-Stokes Equations
6. Calculate The Path

Fig. 1 Stream field navigation algorithm steps [10]

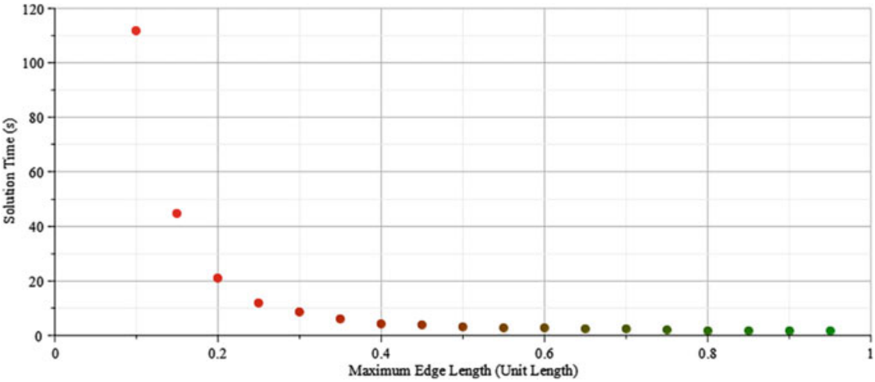


Fig. 2 Mesh maximum edge length versus computations time

affects the computational time required to obtain a numerical solution for a given field. A mesh for a simple navigation field is shown in Fig. 3.

The boundary condition and initial conditions for the discretized field are then defined by the SFN algorithm. These conditions define how the fluid interacts with different segments of the field geometry. The boundaries are mainly defined using a Dirichlet boundary condition and are set to benefit the usage in robotic applications. A comparison between two different sets of boundary conditions is given to show how the effect the flow distribution in the field. In this section, the Stokes flow is also introduced and it was shown why it is not a good model to use for an algorithm that is expected to work in a systematic way. The Stream Field Navigation algorithm then solves the Navier-Stokes equations for the given field. Since the equations do not have a closed-form analytic solution, they solved using numerical methods where the solution is obtained for the discrete points of the field defined by the generated mesh. The solution is obtained using an iterative method and results in a numerical value

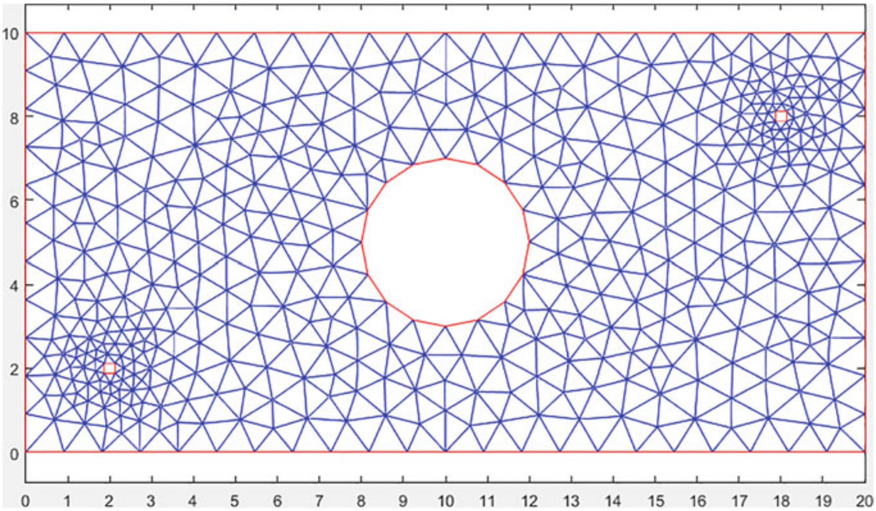


Fig. 3 Mesh of the simple navigation field [10]

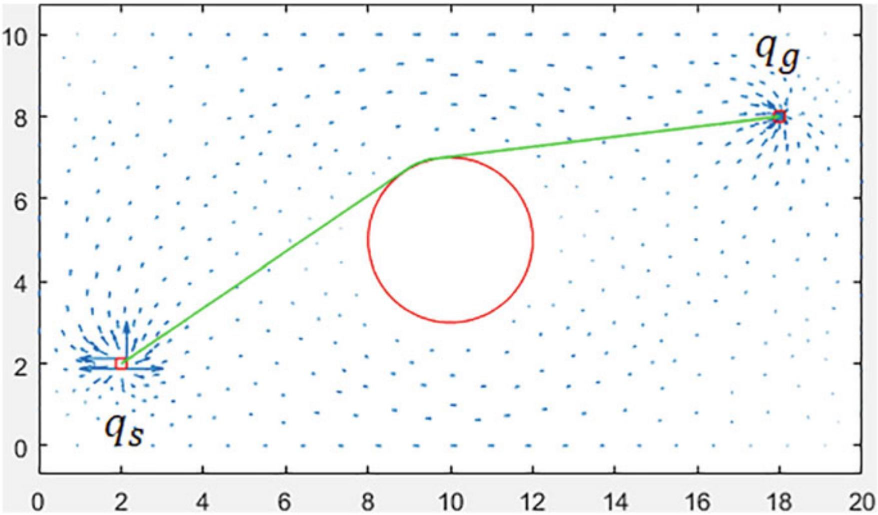


Fig. 4 Path between start and goal points [10]

for each velocity components and for the pressure. With the velocity components available at each discrete points, the algorithm can now start constructing navigation paths. A path is a series of velocity vectors, from from one discrete vector v_n to the next velocity vector v_{n+1}, guiding the agent from the start point q_s to the goal point q_g. Note that the direction of the vector is the value of interest and not the velocity magnitude (Fig. 4).

3 Fluid Path Optimization

Since the final path of the Stream Field Navigation algorithms is a stream of discrete points generated using a meshing algorithm, it can have fluctuations or noise in motion and path shortening may be required. This is a natural behavior of stream fields since, for example, the air flowing between two points in a room with multiple objects may not necessarily take the shortest path between the two points.

Figure 5 shows a field where a path is already calculated and it also highlights some of the imperfections of the path. For example, the velocity vectors starting at the point q_g and the vector v_a can all be replaced by the dashed straight line shown in the figure to reduce the total path length by skipping passing through unnecessary discrete point. The same issue is found for the discrete points inside the oval dashed shape on the figure where the points inside the oval shape can be shortened by a straight line as well. This is part of the task of the path optimization algorithms.

Optimizing the path is done on three levels, L1, L2 and L3. Each level has its own operation, algorithm and path improvement criteria as explained next. The target of each level and the collective target of the three enhancement levels is to minimize the total path length L; where L in its continuous form is:

$$L = \int_{q_s}^{q_g} v\, dt \tag{2}$$

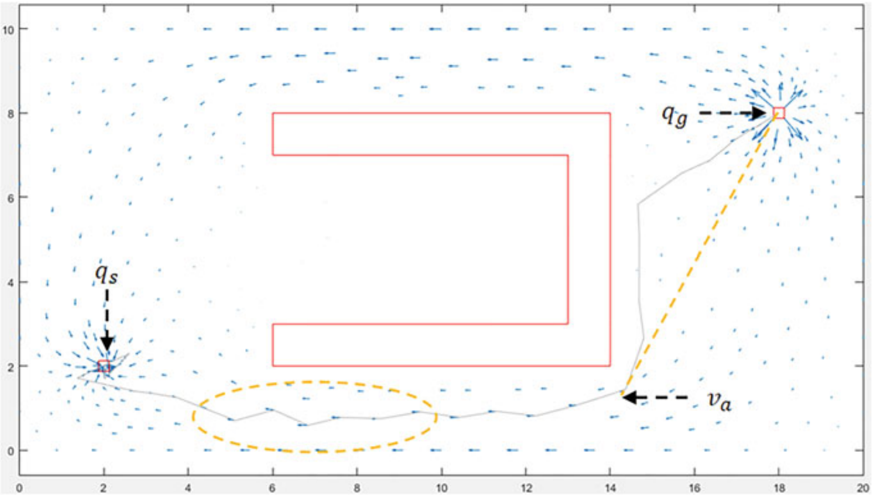


Fig. 5 Stream field navigation path imperfections

Notes on Collision Detection

In the SFN algorithm, field obstacles are represented by straight line and arc segments. Multiple parts of the SFN algorithm are required to check for intersection between segments of the calculated path and the segments representing field obstacles to ensure that a path does not intersect any obstacles and is collision-free. Collision detection is the mathematical problem of evaluating the intersection between a line and a line or between a line and an arc.

- Line-to-Line Intersection

  Assume there are two lines, Line A and Line B; Line A start at (x_1, y_1) and ends at (x_2, y_2) while Line B start at (x_3, y_3) and ends at (x_4, y_4). The equation for each straight line is:

$$y_A = m_A x_A + b_A \quad | \quad b_A = y_1 - m_A x_1 \quad | \quad m_A = \frac{y_2 - y_1}{x_2 - x_1}$$

$$y_B = m_B x_B + b_B \quad | \quad b_B = y_3 - m_B x_3 \quad | \quad m_B = \frac{y_4 - y_3}{x_4 - x_3}$$

  It is now required to find an intersection point (x_i, y_i) between the two lines y_A and y_B so that:

$$m_A x_i + b_A = m_B x_i + b_B$$

$$x_i = \frac{b_B - b_A}{m_A - m_B}$$

  If the value x_i falls within the domain of Line A or B then there is an intersection between The two lines.

  In other words, if $x_i \in [x_1 : x_2]$ and $x_i \in [x_3 : x_4]$ then there is an intersection.

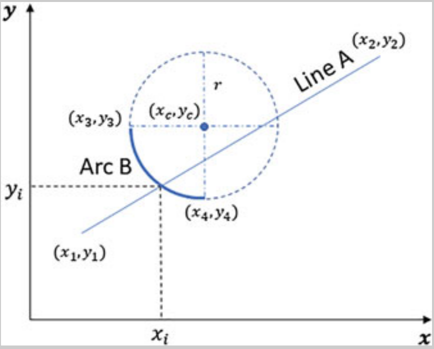


Fig. 6 Line-to-arc intersection

- Line-to-Arc Intersection
  An arc is a section of a circle centered at (x_c, y_c) with radius r (Check Fig. 6)

Assume there is a line, Line A, and an arc, Arc B; Line A start at (x_1, y_1) and ends at (x_2, y_2) while Arc B start at (x_3, y_3), ends at (x_4, y_4), has a radius r and is centered at (x_c, y_c).

The equations for Line A and Arc B are:

$$y_A = m_A x_A + b_A \quad | \quad b_A = y_1 - m_A x_1 \quad | \quad m_A = \frac{y_2 - y_1}{x_2 - x_1}$$

$$(y_B - y_c)^2 + (x_B - x_c)^2 = r^2 \quad | \quad y_B = y_c \pm \sqrt{r^2 - (x_B - x_c)^2}$$

To find an intersection point (x_i, y_i) between the line y_A and the arc y_B, set

$$m_A x_i + b_A = y_c \pm \sqrt{r^2 - (x_i - x_c)^2}$$

Squaring both sides, collecting similar terms and solving for x_i yields:

$$x_i = \frac{-B \pm \sqrt{B^2 - 4\,AC}}{2\,A}$$

$$A = m_A^2 + 1 \quad | \quad B = 2(m_A b_A - m_A y_c - x_c) \quad | \quad C = (b_A - y_c)^2 + x_c^2 - r^2$$

if $x_i \in [x_1 : x_2]$ and $x_i \in [x_3 : x_4]$ then there is an intersection between Line A and Arc B.

3.1 Level 1 Optimization (L1)

A complete path between q_s and q_g is represented - in the SFN algorithm - using a series of connected discrete points or vectors ranging from v_0 to v_{N-1} where N is the number of velocity vectors on the path. The first level of path shortening is required to remove the intermediate vectors between v_0 and v_{N-1} that are not necessary for an obstacle-free path between q_s and q_g.

Programmatically, the L1 Shortening algorithm achieves this by looping over the array of path vectors, with size equal to N, using a nested loop. The first loop loops over the vectors from start to end using the variable i for indexing so that the current vector for the first loop is v_i. The second loop uses the variable j for indexing and loops over the vectors from the end to $i + 1$ so that the current vector for the second loop is v_j. This procedure is outlined in Algorithm 1.

For each index j, the L1 algorithm connects the vector v_j to the vector v_i and checks if the line formed by connecting v_j to v_i intersects any of the field obstacles.

If there are no obstacle intersections, all the vectors between v_j and v_i are removed from the path and then indexer i is increased. Removing the points between two vectors, v_j and v_i, reduces the total number of path point by R where:

$$R = j - i - 1 \tag{3}$$

Algorithm 1 Path optimization L1

```
$i \leftarrow 0$
while $i < N - 1$ do
    for $j$ in $[N - 1 : i + 1]$ do
        if $line(v_i, v_j)$ is collision-free then
            Remove points between $v_i$ and $v_j$
            break
        end if
    end for
    $i \leftarrow i + 1$
end while
```

Figure 7 show the path after applying the L1 algorithm (dotted red line). For the field in the figure, the new path now contains only three discrete points, v_0, v_1 and v_2.

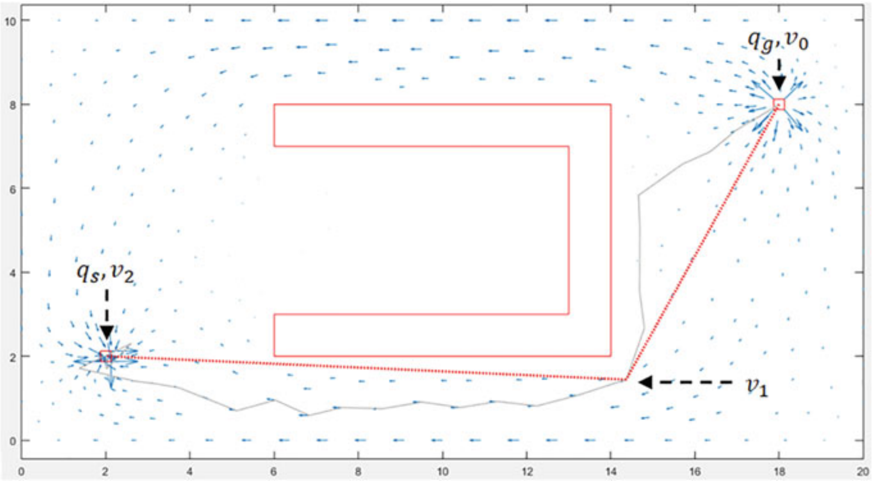


Fig. 7 Path after L1 is applied

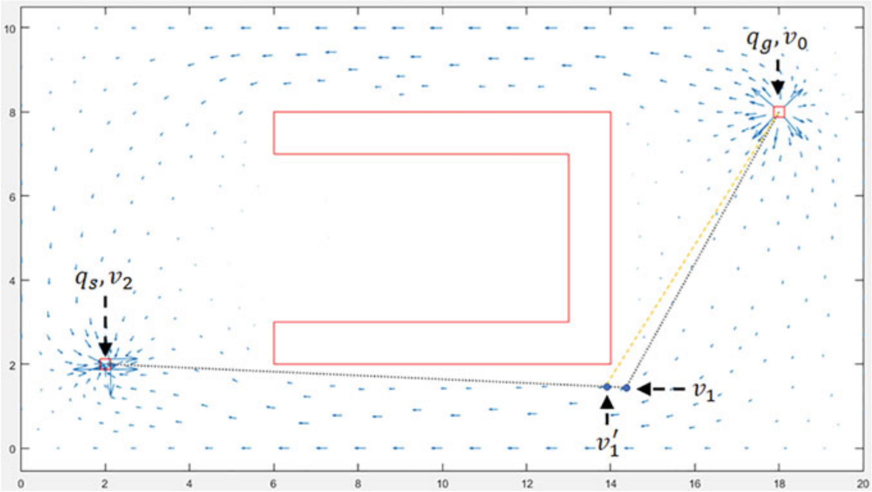


Fig. 8 L2 shortening methodology

3.2 Level 2 Optimization (L2)

The purpose of L2 optimization is to make each segment of the path as short as possible without intersecting with any of the field obstacles. The Level 2 shortening algorithm is applied to the path obtained after applying the Level 1 algorithm.

The path in Fig. 8 has three nodes v_0, v_1, v_2 and two segments $v_0 \rightarrow v_1$, and $v_1 \rightarrow v_2$ where the intermediate point v_1 has two connected segments.

The programmatic methodology of the L2 algorithm in general is to loop over all the points between v_0 and v_{N-1} (N: Number of discrete points on the path) using an indexer i; and for each intermediate point v_i, the algorithm attempts to move the point, using fixed steps with a pre-defined step size, on one of the two adjacent segments $v_{i-1} \rightarrow v_i$ and $v_i \rightarrow v_{i+1}$ and stops the movement when the next step would cause a segment to intersect an obstacle. The same procedure is then carried on the other segment. The implementation of L2 can be seen Algorithm 2.

In Fig. 8, the L2 algorithm moves the point v_1 on the segment $v_1 \rightarrow v_2$ and stops if the next step would intersect a field obstacle. The segment $v_0 \rightarrow v_1$ now becomes $v_0 \rightarrow v_1^{'}$ where $v_1^{'}$ is a temporary point representing v_1. The same process is repeated for the other segment by moving the point $v_1^{'}$ on the segment $v_0 \rightarrow v_1^{'}$. At the end, the location of v_1 is updated to be equal to $v_1^{'}$.

Figure 9 shows the path after applying the Level 2 shortening algorithm (dashed greed line) and a zoom-in for part of the path to show that the path is close to the obstacle but not intersecting it.

Algorithm 2 Path optimization L2

```
$step\_size \leftarrow 0.2$
for $i$ in $[N-2:1]$ do
    //Step 1: calculate number of segments resulting from
    //dividing the length of segment $v_{i+1} \rightarrow v_i$ by $step\_size$
    $N_s \leftarrow ceil\left(\frac{|line(v_{i+1}, v_i)|}{step\_size}\right)$

    //Step 2: calculate step sizes in the $x$ and $y$ directions
    $\triangle x \leftarrow \frac{v_{i,x} - v_{i+1,x}}{N_s}$
    $\triangle y \leftarrow \frac{v_{i,y} - v_{i+1,y}}{N_s}$

    //Step 3: shorten segment $v_{i+1} \rightarrow v_i$ and check for collision
    $v_i^{'} \leftarrow v_i$
    for $j$ in $[1:N_s - 1]$ do
        $v_{i,x}^{'} \leftarrow v_{i,x} - j \triangle x$
        $v_{i,y}^{'} \leftarrow v_{i,y} - j \triangle y$
        if $line(v_i^{'}, v_{i-1})$ is not collision-free then
            break
        end if
    end for
    $v_i \leftarrow v_i^{'}$

    //Repeat steps 1, 2 and 3 for segment $v_i \rightarrow v_{i-1}$
    ...
end for
```

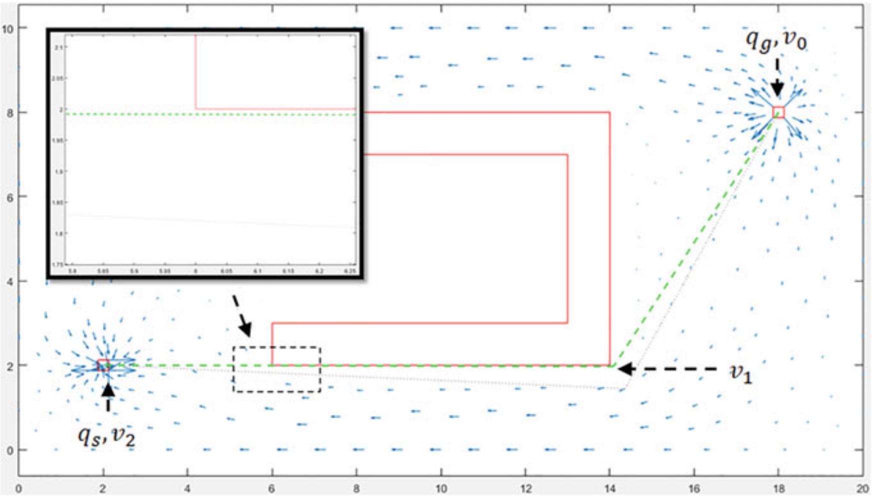


Fig. 9 Path after L2 is applied

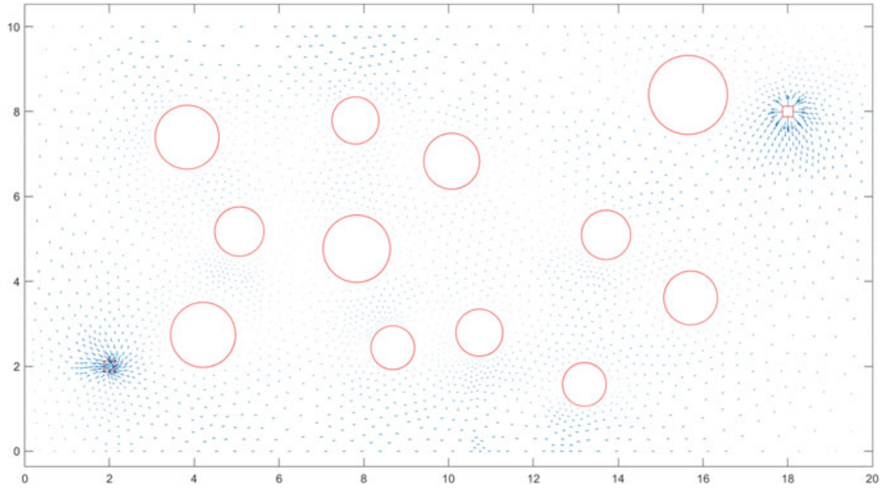


Fig. 10 Navigation field with multiple circular obstacles

3.3 Level 3 Optimization (L3)

An important note about the L1, L2 and L3 algorithms algorithms is that they can be applied to any general discrete path to shorten its length and the improvement percentage of each level would differ from one path to the other. For example, for the field in Fig. 9, L1 and L2 showed improvements for the path while L3 would have no effect on the path. For this reason, another field is introduced in Fig. 10 to show the effects of the L3 optimization algorithm. The navigation field in Fig. 10 is relatively complicated as compared to the navigation field in Fig. 9 and for this field, L1, L2 and L3 would show improvements on the path.

The Level 3 shortening algorithm improves the resulting path from the Level 2 shortening algorithm by making the path shorter, when possible, for certain parts of the path. For each three consecutive nodes (or vectors) in a general path, v_{i-1}, v_i, v_{i+1}, a triangular tip is formed at v_i as a result of the intersection of the segments $v_{i-1} \rightarrow v_i$ and $v_i \rightarrow v_{i+1}$ and under the condition that the three points v_{i-1}, v_i, v_{i+1} are not co-linear as seen in Fig. 11.

When the L3 shortening algorithm is applied on a path with one or more triangular tips, it removes the triangular tips by replacing them with shorter segments as seen in Fig. 12. The new segment is placed successively closer to the obstacle according to a pre-defined step size given that intersection with the obstacle is not allowed.

In this figure, the number of segments is increased to three segments as a result of the new line. The segments are now $v_{i-1} \rightarrow v_i$, $v_i \rightarrow v_{i+1}$ and $v_{i+1} \rightarrow v_{i+2}$.

Theorem 1 *For a general triangle with sides* ***abc****, there exists no side that is equal to or greater in length than the sum of the other two sides.*

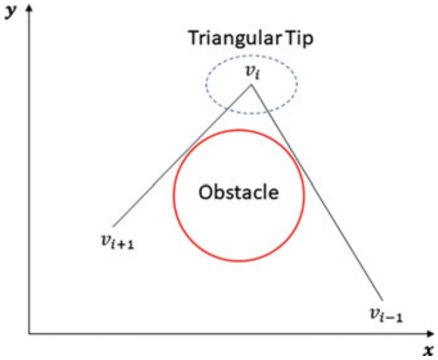


Fig. 11 Triangular tip resulting from three non-colinear path points

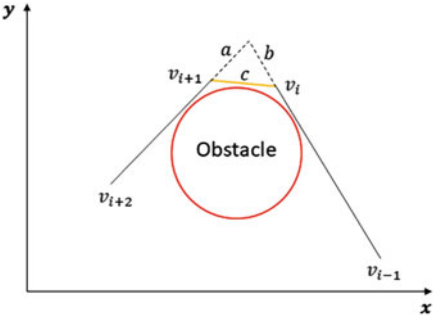


Fig. 12 Triangular tip replaced by a segment

This theorem proves that replacing the replacing the triangular tip in Fig. 11 with a line as in Fig. 12 reduces the total length of the path. Applying this rule to triangle **abc** in Fig. 12 yields:

$$|c| < |a| + |b| \tag{4}$$

The reduction percentage ρ in path length for each triangular tip replacement can be calculated follows:

$$|c|^2 = |a|^2 + |b|^2 + 2\,|a|\,|b|\,\sin(\theta) \tag{5}$$

$$\rho = \frac{\sqrt{|a|^2 + |b|^2 + 2\,|a|\,|b|\,\sin(\theta)}}{|a| + |b|} \tag{6}$$

where θ is the angle between the two sides a and b.

Figure 13 shows the same field in Fig. 10 after finding the path and applying both the L1 (Dotted line) and L2 (Dashed line) algorithms on the original path (Solid line).

Applying the L3 shortening algorithm to the output of L2 for the field in Fig. 13 would result in replacing the triangular tip on the L2 path with a length-reducing segment as seen in Fig. 14.

The programmatic steps of L3 are outlined in Algorithm 3.

Algorithm 3 Path optimization L3

```
$step\_size \leftarrow 0.2$
$i \leftarrow N - 2$
while $i > 0$ do
    //calculate number of segments resulting from dividing
    //the length of segments $v_{i+1} \rightarrow v_i$ and $v_i \rightarrow v_{i-1}$ by $step\_size$
    $N_{v_{i+1} \rightarrow v_i} \leftarrow ceil\left(\frac{|line(v_{i+1}, v_i)|}{step\_size}\right)$
    $N_{v_i \rightarrow v_{i-1}} \leftarrow ceil\left(\frac{|line(v_i, v_{i-1})|}{step\_size}\right)$

    //calculate step sizes in the x and y directions
    //for each segment
    $\triangle x_{v_{i+1} \rightarrow v_i} \leftarrow \frac{v_{i,x} - v_{i+1,x}}{N_{v_{i+1} \rightarrow v_i}}$
    $\triangle y_{v_{i+1} \rightarrow v_i} \leftarrow \frac{v_{i,y} - v_{i+1,y}}{N_{v_{i+1} \rightarrow v_i}}$
    $\triangle x_{v_i \rightarrow v_{i-1}} \leftarrow \frac{v_{i-1,x} - v_{i,x}}{N_{v_i \rightarrow v_{i-1}}}$
    $\triangle y_{v_i \rightarrow v_{i-1}} \leftarrow \frac{v_{i-1,y} - v_{i,y}}{N_{v_i \rightarrow v_{i-1}}}$

    //Replace tip with a new segment (Replace $v_i$ with $v_{ia} \rightarrow v_{ib}$)
    $j \leftarrow 1$ //variable used to loop over the sub-segments of $v_{i+1} \rightarrow v_i$
    $k \leftarrow 1$ //variable used to loop over the sub-segments of $v_i \rightarrow v_{i-1}$
    while $j < N_{v_{i+1} \rightarrow v_i} - 1$ or $k < N_{v_i \rightarrow v_{i-1}} - 1$ do
        $v^{'}_{ia,x} \leftarrow v_{i,x} - j \triangle x_{v_{i+1} \rightarrow v_i}$
        $v^{'}_{ia,y} \leftarrow v_{i,y} - j \triangle y_{v_{i+1} \rightarrow v_i}$
        $v^{'}_{ib,x} \leftarrow v_{i,x} - k \triangle x_{v_i \rightarrow v_{i-1}}$
        $v^{'}_{ib,y} \leftarrow v_{i,y} - k \triangle y_{v_i \rightarrow v_{i-1}}$

        if $line(v^{'}_{ia}, v^{'}_{ib})$ is not collision-free then
            break
        end if

        //increase j and k
        if $j < N_{v_{i+1} \rightarrow v_i} - 1$ then
            $j \leftarrow j + 1$
        end if
        if $k < N_{v_i \rightarrow v_{i-1}} - 1$ then
            $k \leftarrow k + 1$
        end if
    end while
    Replace $v_i$ with $v_{ia}$ and $v_{ib}$
    $i \leftarrow i - 1$
end while
```

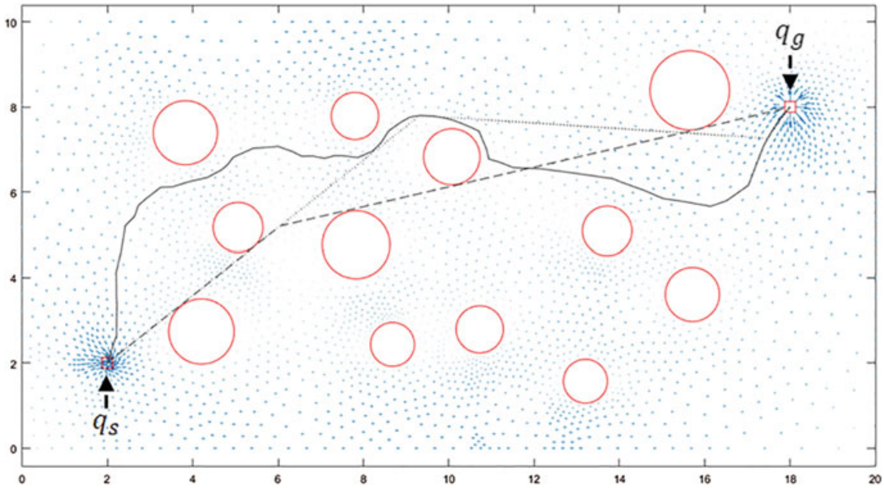


Fig. 13 L1 and L2 algorithms applied on the path

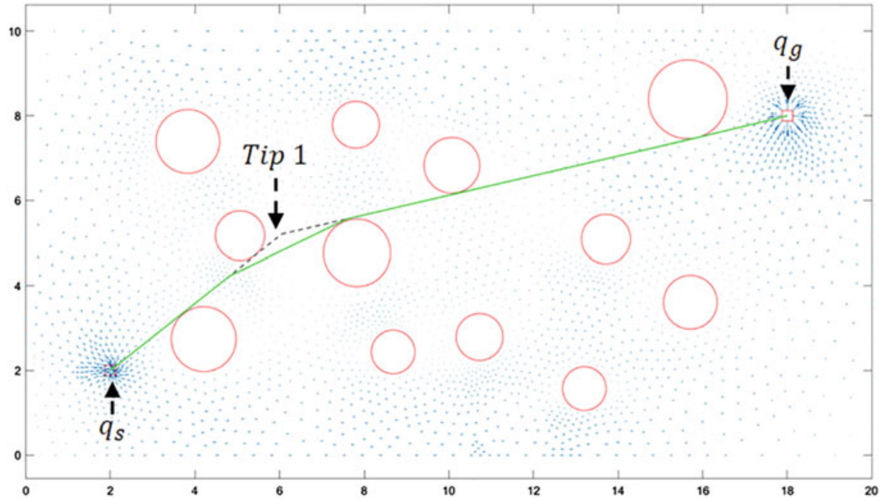


Fig. 14 Path resulting from the L3 shortening algorithm

At this point, the three path optimization algorithms have been concluded. It is worth noting that, although the algorithms were developed to improve the path generated by the SFN algorithm, they can be applied to general paths to benefit from their improvements.

As an example, the RRT algorithm is used to generate a path in the field shown in Fig. 15. In this field, the original RRT path is shown in magenta and the optimized path is shown in green. The original path is optimized using the L1, L2 and L3 algorithms. Table 1 shows a comparison between the original and optimized RRT

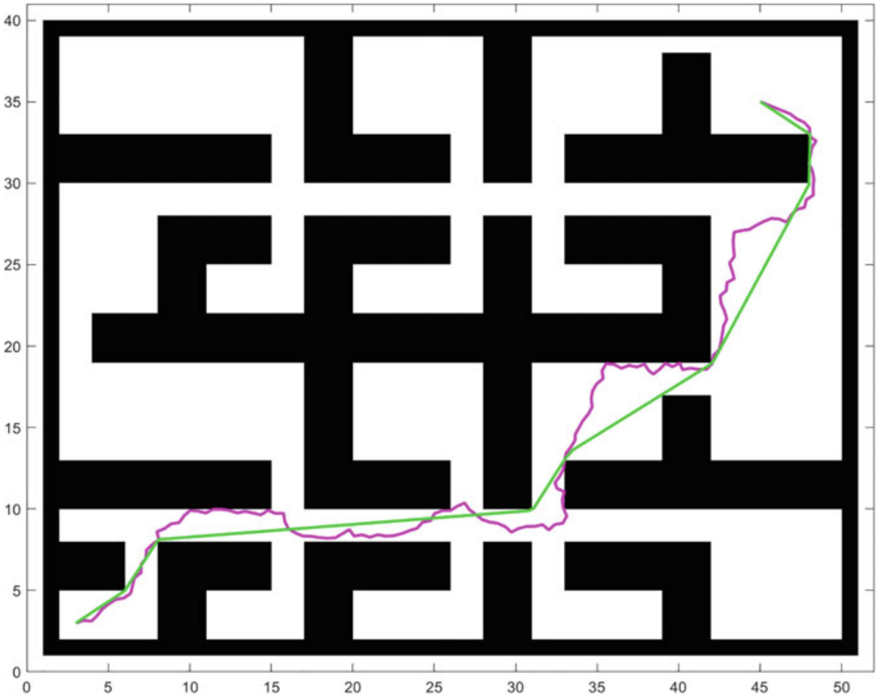


Fig. 15 Path optimization for RRT

Table 1 RRT paths comparison

Algorithm	RRT	RRT + L1–L2–L3
Path length [length unit]	78.75	64.09
Number of turns	159	8

paths in terms of path quality. The number of turns is taken as an assessment for how smooth the generated path is; which is related to the energy consumed by the agent following the path [1]. The optimization algorithms reduced the path length by 18.6% while greatly reducing the number of turns of the path.

3.4 Results

This section includes the results for the path optimization algorithms. Assessing the three optimization algorithms is done by using different types of maps with varying characteristics to show the behavior of the algorithms in different environments. The characteristics vary in terms of:

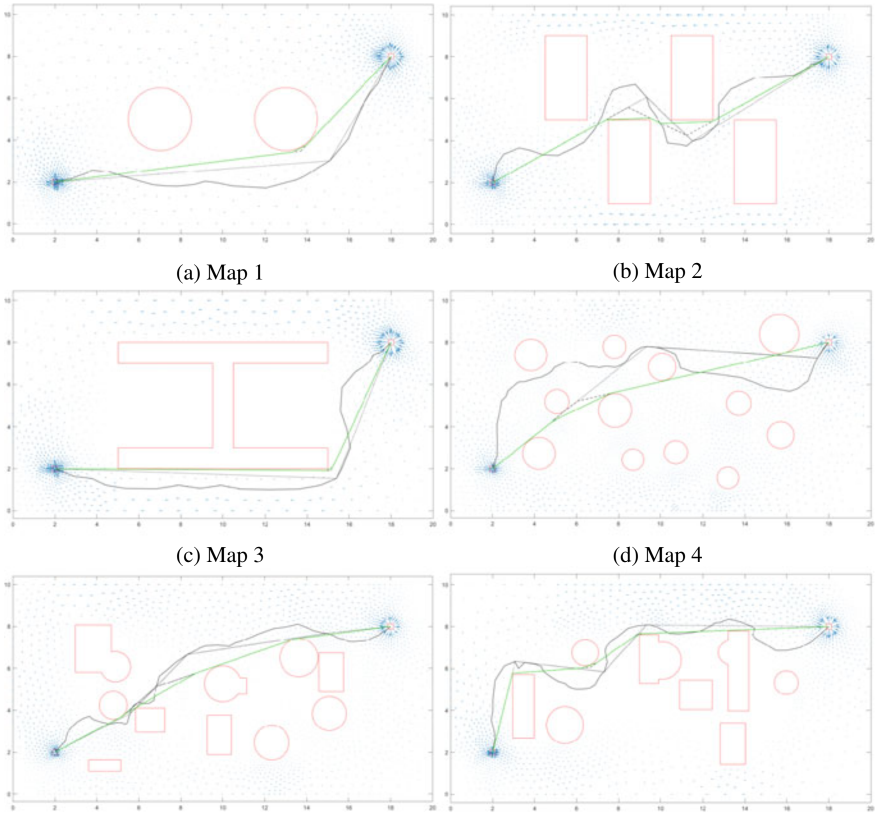
- Number of field obstacles
- Geometry of field obstacles
- Symmetric distribution versus random distribution of the field obstacles.

The tests include six maps as seen in Fig. 16. In each map, the path improvements resulting from the three optimization algorithms and the execution times of the algorithms were recorded.

Table 2 shows the following points that are related to path improvements (Note: The measurement unit used in this table is [length unit] except for percentages, which are unit-less):

Original Length The length of the path before applying any of the three optimization algorithms.

L1–L2–L3 Lengths The path length resulting from applying each of the three optimization algorithms.



(a) Map 1

(b) Map 2

(c) Map 3

(d) Map 4

(e) Map 5

(f) Map 6

Fig. 16 Path optimization maps

Table 2 Path optimization–length improvement

Map	Original length	L1–L2–L3 lengths	Reduction (%)
Map 1	19.44	18.89–18.03–17.99	7.5
Map 2	22.44	19.04–18.21–17.64	21.4
Map 3	21.45	20.37–19.87–19.81	7.7
Map 4	21.91	18.42–17.44–17.36	20.7
Map 5	19.56	17.74–17.38–17.30	11.5
Map 6	23.75	20.42–19.48–19.44	18.2

Table 3 Path optimization–mean execution times

Map	SFN	L1–L2–L3	Increase (%)
Map 1	0.238	0.004–0.006–0.004	5.9
Map 2	0.240	0.003–0.004–0.004	4.6
Map 3	0.201	0.002–0.004–0.004	5.0
Map 4	0.688	0.008–0.017–0.012	5.4
Map 5	0.664	0.017–0.029–0.018	9.6
Map 6	0.386	0.014–0.011–0.006	8.0

Reduction Reduction percentage in the original path length after applying the three optimization algorithms. Reduction is calculated as $\frac{Length_{Original} - Length_{L3}}{Length_{Original}}$.

The execution time experiment is executed 100 times in each map to obtain empirical statistical results of execution times. For each field, the results are expressed as the mean (average) execution time and the standard deviation of the execution times of the 100 runs. The standard deviation is included in the results as the mean by itself is not an effective statistical measure for execution times [26]. All the tests were executed on an Intel 6500U Mobile Processor with 8 GB of RAM.

In Table 3, execution times for the L1, L2 and L3 optimization algorithms are shown (Note: The measurement unit used in this table is [second] except for percentages, which are unit-less).

SFN Execution time to obtain a path from the SFN algorithm without applying any of the three optimization algorithms

L1–L2–L3 Execution time for each of the three optimization algorithms

Increase Time increase percentage when the optimization algorithms are applied. Increase is calculated as $\frac{Time_{L1}+Time_{L2}+Time_{L3}}{Time_{SFN}}$.

Table 4 show the standard deviation (in seconds) for the measurements shown in Table 3.

4 SFN in Multi-level Path Planning

The wide range of indoor applications that can be automated have contributed to the importance of indoor mobile robotics. Robots operating in these environments are required to be autonomous to plan paths and avoid the field obstacles. One special case of indoor environments is the multi-level or multi-floor environment which is characterized by having a different map and distribution of obstacles for each level.

In multi-level environments, a robotic agent may be assigned a task that starts on one level and ends on another. In this case, the robotic agent is required to execute a Multi-Level Path Planning (MLPP) approach to seamlessly plan a path that connects two- or more-floors together.

4.1 MLPP Approach

Path planning algorithms can be categorized by the dimensionality of the space they operate in. In which case, the categorization commonly used for the space is 2D, 2.5D or 3D [37]. 2.5D maps are the result of constraining a 3D map to only planer sections; which is essentially converting a 3D environment to a 2D environment. Accordingly, 2D and 2.5D maps are represented in $\Re^2$ while 3D maps are represented in $\Re^3$. Although there are algorithms that can operate on $\Re^3$, it is generally more computationally heavy due to the increased space dimensionality, compared to $\Re^2$, and the added computational cost of 3D collision-detection.

Constructing a 2.5D map can be done, for example, by using the point cloud generated by a LiDAR sensor [13]. With the 2.5D map available, path planning algorithms can then operate on the obtained maps to find a collision-free path related to a given task. For example, researchers have handled the path planning task in 2.5D maps with the A* algorithm [36] where the 2.5D map represented a navigation graph of the environment.

The robustness of the Stream Field Navigation algorithm allows it to seamlessly generate paths for 3D environments that have been constrained to a 2D movement space where the algorithm essentially operated on $\Re^3$. In Figs. 17 and 18, a building consisting of two levels is introduced to explain how the SFN would operate in the situation of a multi-level environment.

In this building, the levels are not identical, where each level has a different configuration of rooms, and a staircase connects the two levels together. The path planning algorithm is required to navigate from a certain room on the first level to another room on the second level as seen in the figures.

To make the problem more challenging, the doors are kept open to increase the planning space. This essentially increases simulation space for physics-based algorithms and the search space for search-based algorithms.

The given two-level 3D environment is first constrained to only planer movements on the floor of each level (for ground robots). The levels are then separated so that

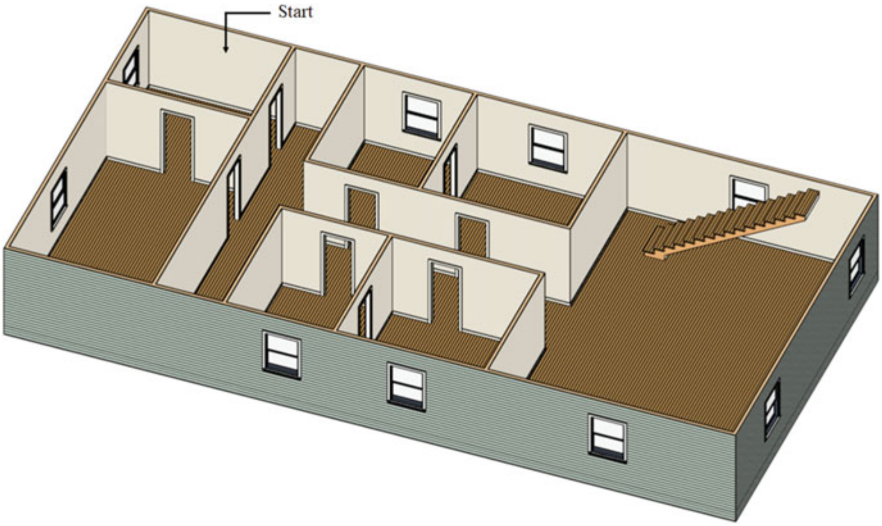


Fig. 17 MLPP building—level 1

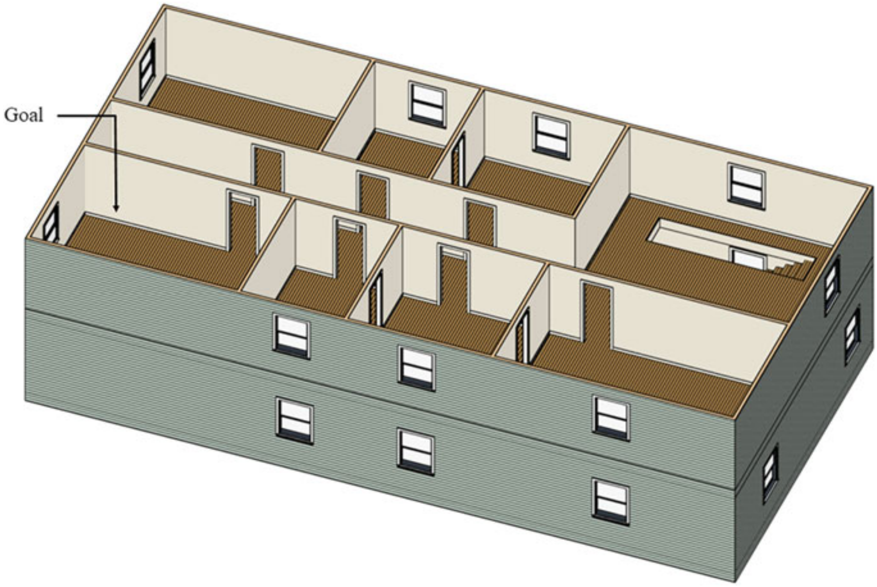


Fig. 18 MLPP building—level 2

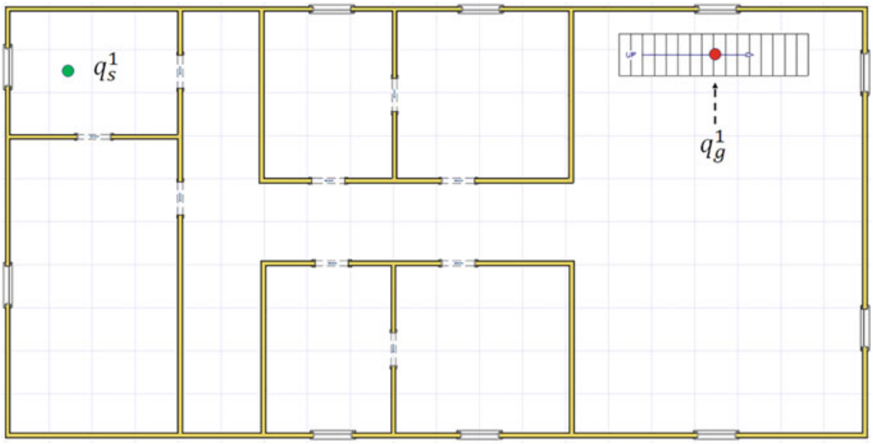


Fig. 19 MLPP building—level 1 schematic

each level has its own 2D schematic and then they are integrated together in a 2.5D map by setting the end point of one level to be the start point of the next level. In other words, q_g of level one overlaps q_s of level two, essentially creating a connection between the two levels.

In general, a multi-level environment is converted to a 2.5D environment by the condition in Eq. 7 for the start and end points of each level

$$q_g^i = q_s^{i-1} \tag{7}$$

where q_g^i is the goal point for level i and q_s^i is the start point for level i. These points would typically be located on stairs or elevators; which are the main transition points between the floors.

Assuming that the agent starts its task on floor m and ends on floor n, it follows that q_s^m and q_g^n are unique and are not required to overlap start or end points of other floors. The 2D schematics for the levels in Figs. 17 and 18 are shown in Figs. 19 and 20, respectively. The schematic figures also show the start and goal points for each floor.

This environment poses a challenge in that, for each level, the staircase as an object of the field is both an obstacle and either a start or goal point; That is, for level one, the robot is allowed to cross the west edge of the staircase while the other edges are not to be crossed. Similarly, for level two, the east edge is open while the other edges are blocked. The SFN algorithm allows different mathematical boundary conditions for segments of the same object in the field. For the staircase, an inlet velocity or constant pressure boundary condition is set for the desired inlet or outlet edge, respectively, while the other edges are set to have a wall boundary condition. These boundaries are modeled with a Dirichlet boundary condition. The SFN algorithm can accordingly calculate a path for each level as seen in Figs. 21 and 22.

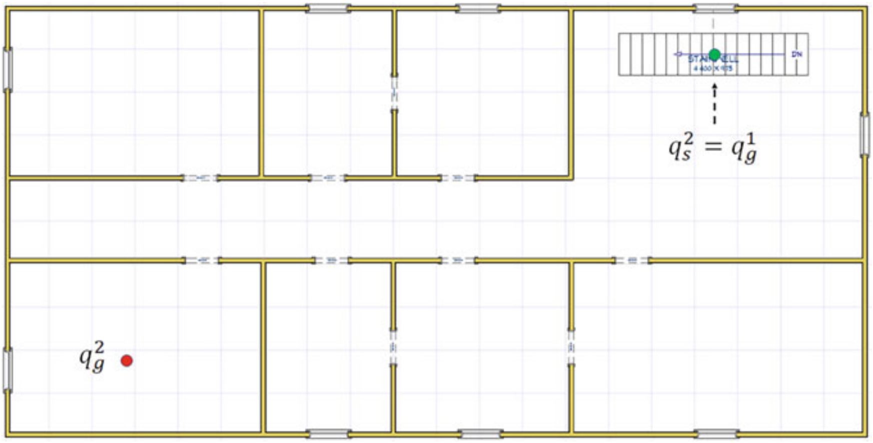


Fig. 20 MLPP building—level 2 schematic

Dirichlet Boundary Condition

This type of boundary condition specifies the value that the unknown function will have along the boundary $\partial\Omega$ of the domain Ω according to the following equation

$$\varphi(x) = f(x) \qquad x \in \partial\Omega$$

where φ is the unknown function, x is the independent variable and f is the scaler function defining the value of φ on $\partial\Omega$ [3, 25].

In the figures, the solid black path is the unoptimized path, the dotted path is the result of L1 optimization, the dashed path is the result of L2 optimization and the solid green path is the result of L3 optimization and the final path.

It can be seen in the figures that the flow magnitudes are dominant in the hallway between the rooms, as denoted by the blue arrows, and weak in rooms that are not part of the path. On each level, the flow is also dominant on one side of the staircase and weak for the other sides as a direct result of the staircase boundary conditions. These observations are governed by the underlying physical model, defined in Sect. 2, and boundary conditions as they force the streams to be dominant in the hallway connecting the staircase to the target room while avoiding eddy flows in neighboring rooms.

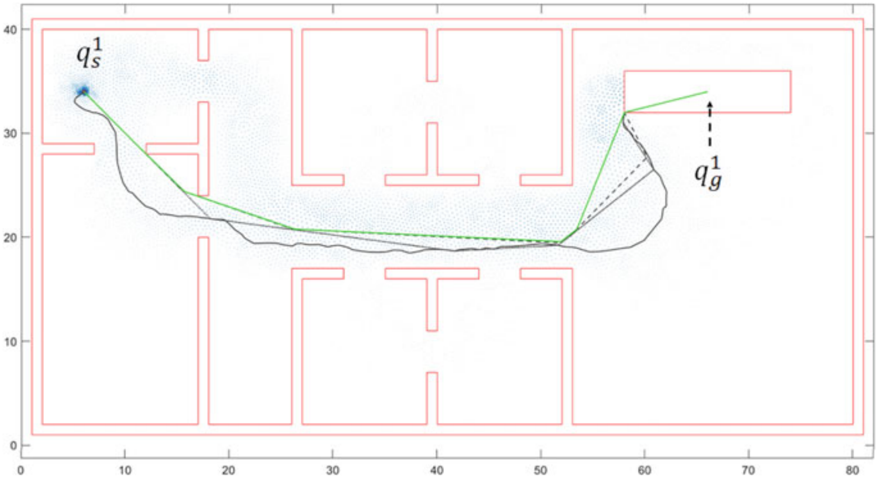


Fig. 21 MLPP building—level 1 path

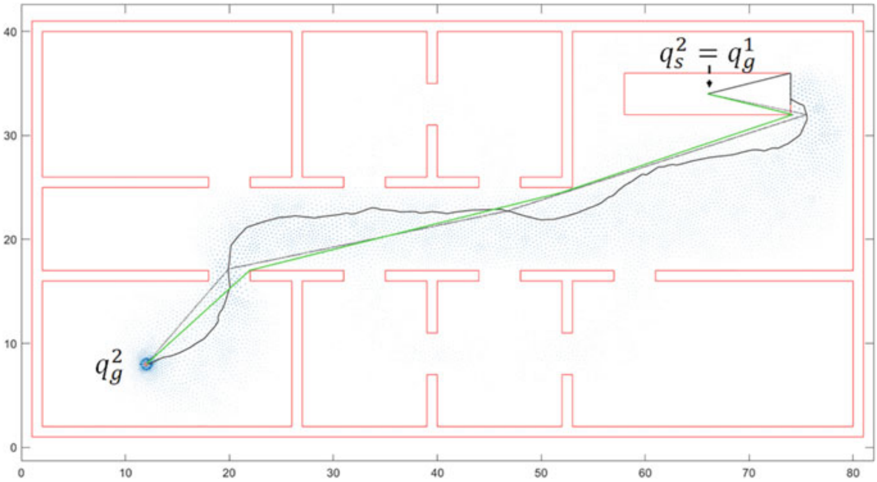


Fig. 22 MLPP building—level 2 path

Table 4 Path optimization–execution time standard deviations

Map	SFN	L1–L2–L3
Map 1	0.014	0.000–0.001–0.001
Map 2	0.011	0.001–0.000–0.000
Map 3	0.081	0.000–0.001–0.000
Map 4	0.076	0.001–0.002–0.002
Map 5	0.051	0.002–0.003–0.002
Map 6	0.018	0.002–0.001–0.001

4.2 Results

In this section, a comparison is given on how differently does the Stream Field Navigation algorithm behave as compared to the RRT and PRM algorithms. The multi-level building introduced in Sect. 4, along with its configuration, start points and points point, will be used for the comparison. These algorithms are selected due to their wide usage for indoor environments and mobile robotics [2, 4, 8, 12, 38].

An important thing to point out is that neither of the RRT and PRM algorithms have the concept of boundary conditions. And therefore, restricting staircase usage to only one side will not be possible without adding more obstacles to represent the other sides of the staircase (Note that for the SFN algorithm, the restricted edges of the staircase are part of the goal location boundaries for level one and part of the start location boundaries for level two). These obstacles will not be added as they will not be an accurate representation of the field and also so that the SFN is compared to the other algorithms on the exact same field.

Constructing the field for the RRT algorithm and generating the path results in the random trees and paths shown in Figs. 23 and 24 for level one and level two, respectively. Similarly, Figs. 25 and 26 show the random nodes and paths for the PRM algorithm for level one and level two, respectively.

The first thing to note is that both the RRT and PRM algorithms explore the rooms that are not part of the path. This is true for both the first and second levels.

A direct result is that it incurs more computational cost when operating in these room-based environments. In fact, only one room on each level is required to be explored while all the rooms are actually explored. This unnecessary exploration of rooms is attributed to the random nature of the algorithms. Moreover, and as mentioned earlier, the RRT and PRM algorithms assume that, by default, all the sides of the staircase are usable. This can be seen in the figures as the direction followed by the algorithms when approaching or leaving the staircase is random; which is also attributed to the random nature of the algorithms.

To obtain the empirical execution times for the algorithms, the experiment is run 100 times in each level for each algorithm while recording the mean execution time and the standard deviation of the recorded execution times. Path and execution comparisons between the SFN algorithm and both the RRT and PRM algorithms are shown in Tables 5 and 6 for level one and level two respectively.

To include path quality factors in the results, the comparisons show the number of turns of the calculated paths and the path length generated by each algorithm. The number of turns is taken as an assessment for how smooth the generated path is; which is related to the energy consumed by the agent following the path [1].

The results show that the SFN algorithm is faster than both the RRT and PRM algorithms on both building levels. This is the result of two points; First, the fact that the SFN algorithm does not require a large number discrete mesh points for these fields; The second point is the weak magnitudes of velocity vectors in rooms that are not part of the generated path which allows for a more straight-forward path

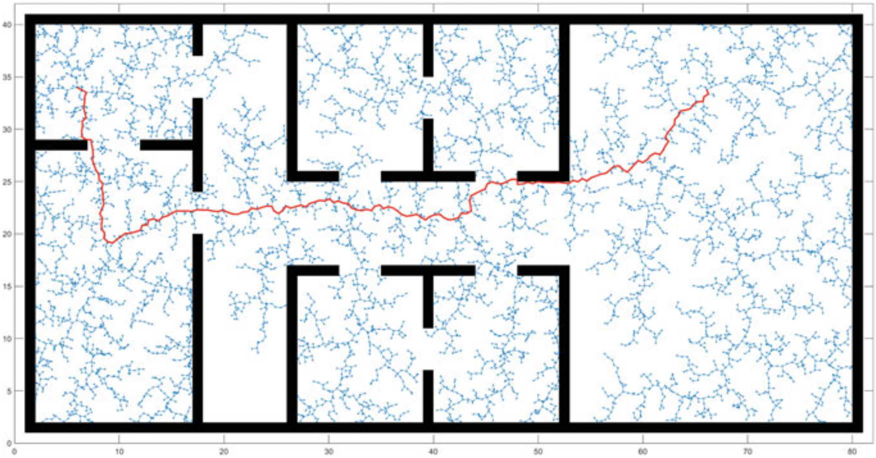


Fig. 23 MLPP building—level 1 RRT path

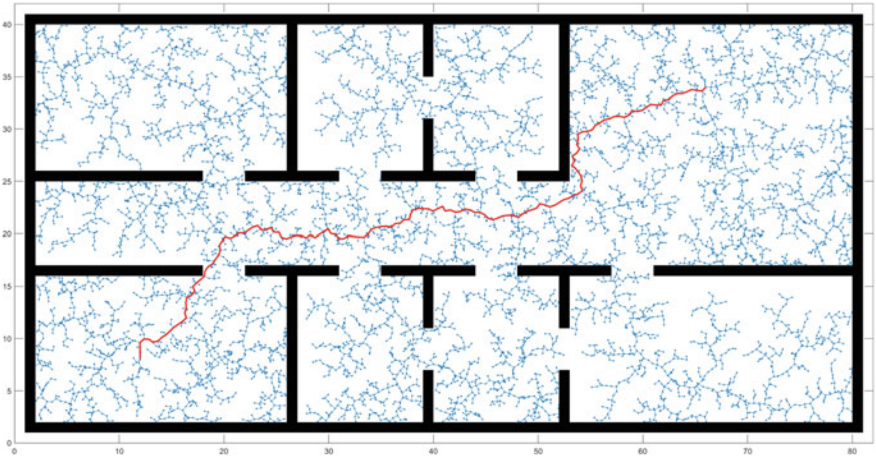


Fig. 24 MLPP building—level 2 RRT path

construction. These two points together greatly reduce the computations required by the SFN algorithm to form a complete path.

Another observation is that the path generated by the SFN algorithm has very few turns as compared to the path generated by the RRT or the PRM algorithm while also being shorter; which is a direct contribution of the L1, L2 and L3 path optimization algorithms. Note that the path generated by the SFN algorithm for level two is longer compared to the path generated by the PRM algorithm because the SFN algorithm is using the far side of the staircase instead of the nearest side as can be seen in Fig. 22.

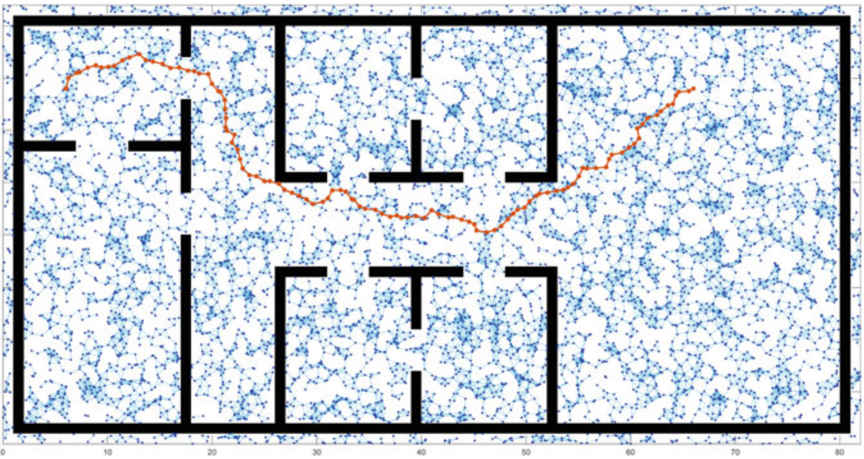


Fig. 25 MLPP building—level 1 RRM path

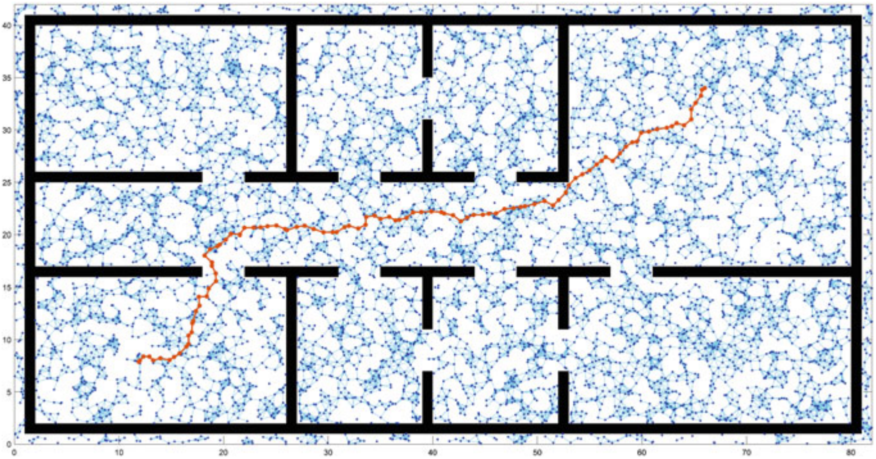


Fig. 26 MLPP building—level 2 PRM path

Table 5 MLPP building—level 1—comparison results

Algorithm	SFN	RRT	PRM
Mean execution time [s]	0.812	8.529	4.162
Standard deviation [s]	0.091	0.422	0.388
Number of turns	6	176	105
Path length [length unit]	72.8	87.71	79.12

Table 6 MLPP building—level 2—comparison results

Algorithm	SFN	RRT	PRM
Mean execution time [s]	0.864	11.174	4.235
Standard deviation [s]	0.94	0.622	0.348
Number of turns	4	152	99
Path length [length unit]	75.99	76.19	73.14

5 ROS Environment

The Robot Operating System (ROS) is a set of software libraries that are widely used for robotic research and development. It is integrated with state-of-the-art robotic algorithms, robot drivers, sensor libraries and mathematical models (Ex. Kalman Filters). Due to its popularity, ROS is widely used in industry, research and virtual verifications. For example, the second version of ROS, ROS2, has been used to develop an architecture for self-driving cars where the main concern is safety [30]. ROS2 is also used for industrial grade virtual commissioning of automation systems in collaborative environments where humans are working alongside robots [9].

- The virtual environment developed for the Stream Field Navigation algorithm is based on ROS2 and Gazebo11. In the remainder of this section, ROS refers to ROS2 While Gazebo refers to Gazebo11.
- Names of ROS libraries and packages are written with lower-case underscore-separated italic words.

A virtual environment architecture is presented in this section to allow the Stream Field Navigation algorithm to operate in the ROS environment. Along with ROS, Gazebo is also integrated to the virtual environment to add physical simulation capabilities to the verification system. ROS is based on the concept of Nodes where each node represents a part of the system and is responsible for a certain task. For example, a node may be responsible for interfacing with a sensor while another node is responsible for operating a certain type of motors.

The developed ROS system is comprised from multiple building blocks and two main sub-systems; a navigation sub-system and a localization sub-system. The architecture of the developed ROS system is shown in Fig. 27. The ROS system includes definitions for a simulated robot, a simulation world and a binary occupancy map of the world. These definitions are used to populate the environment. The *robot_state_publisher* package is responsible for publishing the robot description and the geometrical transformation between the robot joints to other ROS nodes.

Navigation in the ROS system is implemented through the Nav2 framework [22]. This framework is the core component responsible for moving a robot between two

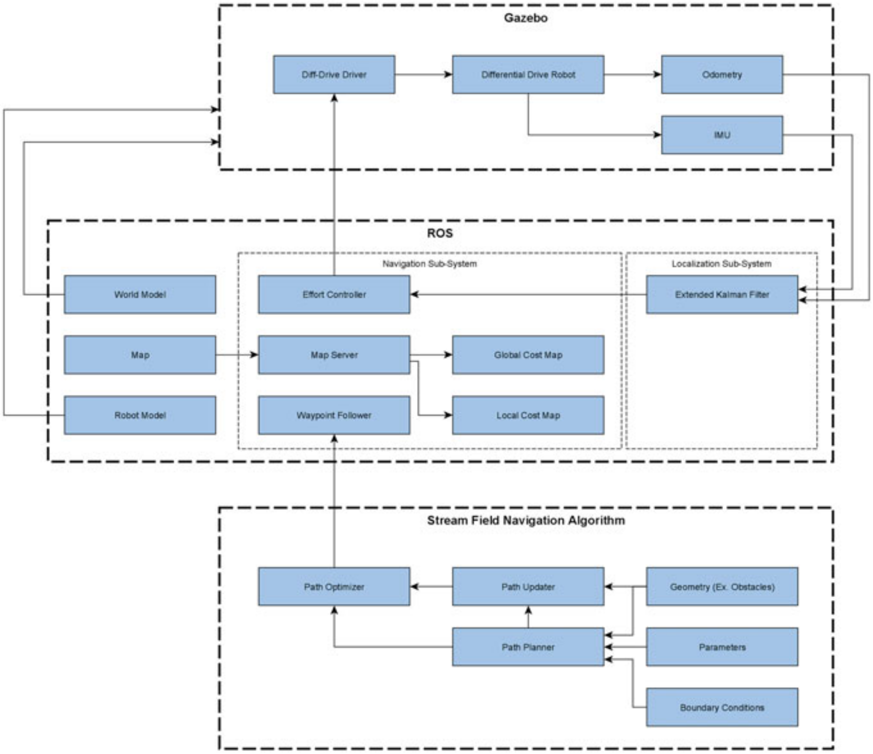


Fig. 27 ROS-Gazebo verification environment

given points. It can also be used for planning a complete path if planning is required. However, in this case, the SFN algorithm is used for planning the complete path between a start and a goal point. Nav2 also contains an effort controller which can control the drive system of a given robot.

In real robotic systems, sensors are not perfect as they may have a noise associated with the signal and their readings may drift overtime. In the ROS environment developed to be integrated with the SFN algorithm, the Gazebo system is implemented with two sensors, and Inertial Measurement Unit (IMU) and a wheel encoder to publish odometry data. These two sensors are used to localize the robot in the map with respect to the initial start point. However, both sensors suffer from an error component where the IMU signal is noisy and the wheel encoders report wheel rotation even if the wheel is rotating in-place. To overcome this issue, an Extended Kalman Filter is integrated into the system to smooth out the sensor reading and output a value that converges to the true value. The Kalman Filter is implemented through the $robot\_localization$ package. The corrected sensor data are then sent to the navigation sub-system to allow for correct guidance to the goal point.

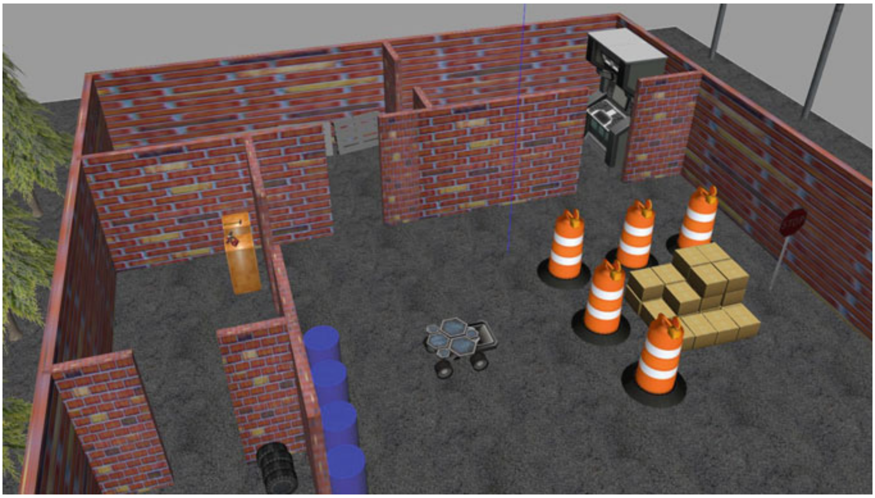


Fig. 28 ROS-Gazebo simulation world

With both the ROS and Gazebo systems implemented, the SFN algorithm can then send the generated paths to the ROS navigation sub-system where it would then control the robot in the Gazebo world. A sample of the Gazebo world designed for this purpose is shown in Fig. 28 as an example.

6 Conclusion

The developed SFN algorithm has been proven to find a collision-free and local-minima-free path in a navigation field. However, since the SFN algorithm is based on the Navier-Stokes equations, it is important that a further path optimization step is added to improve the efficiency of the fluid stream path for robotic applications.

This research work has two targets. First, to design algorithms that optimize the original path generated by the SFN algorithm, and second, to develop an approach for the SFN algorithm to operate in multi-level environments.

Path optimization algorithms were developed as part of this research work to realize the first target where the path calculated by the Local Field Planner is enhanced using three shortening (or optimization) levels, L1, L2 and L3 to convert a longer path to a short path by removing multiple path imperfections. The three shortening algorithms were developed to overcome a shortcoming of fluid streams, which is that fluid streams do not follow the shortest path between two points. Testing of the developed L1, L2 and L3 path optimization algorithms has shown that they results in a 7.5–20.7% reduction in path length at the cost of an average increase of approximately 6.5% to the un-optimized SFN algorithm execution time.

For the second target, a method is introduced to constraint a 3D environment to a 2D movement space; essentially converting the 3D map to a 2.5D map. The method introduced for the SFN algorithm connects different levels of the same building at certain pre-defined points utilizing the mathematical model of the SFN algorithm and its boundary condition. The results show that the path generated by the SFN algorithm is more efficient, in terms of execution time and number of turns, compared to the paths generated by the RRT and PRM algorithms.

7 Future Work

Development of the Stream Field Navigation algorithm is a continuous research work and further research to build upon the current state of the SFN can be done to enhance certain aspects of the algorithm.

One of the proposed ideas is the Local Stream Calculations (LSC); Which can be researched and implemented later to further improve the efficiency of the Stream Field Navigation algorithm in kinematic environments. The idea of LSC is to re-calculate the flow velocities in a certain region of the field instead of doing the calculation for the whole field in case an offline path update is required; For example, in the vicinity of a moved obstacle (Dealing with moving obstacles and the decision to use an offline vs. an online path update are explained in [11]). This would be considered as an improvement in terms of algorithm efficiency; But unlike the efficiency improvement achieved through path optimization (Introduced in Sect. 3), LSC would improve the efficiency for kinematic environments.

The next area of improvement for the Stream Field Navigation algorithm is that it can be integrated with a SLAM algorithm to allow for better operation in outdoor environments. This would also allow the algorithm to seamlessly move between indoor (Where the field is known beforehand) and outdoor environments (Where the field is unexplored and highly variable). Achieving this, along side the capabilities for multi-level path planning (Introduced in Sect. 4), would allow the SFN algorithm to be effective in wider and more varied use cases.

References

1. Ahmed F, Deb K (2013) Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. Soft Comput 1283–1299
2. Al-Dahhan M, Schmidt K (2019) Path planning based on Voronoi diagram and PRM for omnidirectinal mobile robots. In: Digital transformation and smart systems
3. Arfken GB, Weber HJ (2005) Mathematical methods for physicists. Academic Press
4. Bai T, Fan Z, Liu M, Zhang S, Zheng R (2018) Multiple waypoints path planning for a home mobile robot. In: International conference on intelligent control and information processing
5. Chang H (1996) A new technique to handle local minimum for imperfect potential field based motion planning. In: IEEE international conference on robotics and automation

6. Chen Y, Luo G, Mei Y, Yu J, Su X (2016) UAV path planning using artificial potential field method updated by optimal control theory. Int J Syst Sci 47(6)
7. Csorba M, Uhlmann J (1997) A suboptimal algorithm for automatic map building. In: American control conference
8. Dong Y, Camci E, Kayacan E (2017) Faster RRT-based nonholonomic path planning in 2D building environments using Skeleton-constrained path biasing. J Intell Robot Syst 387–401
9. Eros E, Dahl M, Hanna AAA, Falkman P, Bengtsson K (2019) Integrated virtual commissioning of a ROS2-based collaborative and intelligent automation system. In: International conference on emerging technologies and factory automation (ETFA)
10. Fawzy HM, El-Sherif HM, Baumann G (2022) A framework for robotic path planning based on enhanced fluid potential dynamical models. In: International conference on mechatronics and robotics engineering, Munich, Germany
11. Fawzy HM, El-Sherif HM, Baumann G (2022) Computational time reduction in stream field navigation through dynamic path search and selection in kinematic environments. Array 14. https://doi.org/10.1016/j.array.2022.100134
12. Francis A, Faust A, Chiang HTL, Hsu J, Kew JC, Fiser M, Lee TWE (2020) Long-range indoor navigation with PRM-RL. IEEE Trans Robot 1115–1134
13. Gim H, Jeong M, Han S (2021) Autonomous navigation system with obstacle avoidance using 2.5D map generated by point cloud. In: International conference on control, automation and systems (ICCAS)
14. Guo J, Li C, Guo S (2020) Path optimization method for the spherical underwater robot in unknown environment. J Bionic Eng 17. https://doi.org/10.1007/s42235-020-0079-3
15. Jaulin L (2009) A nonlinear set-membership approach for the localization and map building of an underwater robot using interval constraint propagation. IEEE Trans Rob 25:88–98
16. Jaulin L (2011) Range-only SLAM with occupancy maps; a set-membership approach. IEEE Trans Rob 27(5):1004–1010
17. Khamis A (2021) Smart mobility: foundational technologies. Apress, Berkeley, CA
18. Khatib O (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: IEEE international conference on robotics and automation
19. Kingston Z, Mol M, Kavraki L (2018) Sampling-based methods for motion planning with constraints. Ann Rev Control Robot Auton Syst 1:159–185
20. Kumar S, Parhi D, Kashyap A, Muni M (2021) Static and dynamic path optimization of multiple mobile robot using hybridized fuzzy logic-whale optimization algorithm. J Mech Eng Sci 21. https://doi.org/10.1177/0954406220982641
21. Lee H, Jeong J (2021) Mobile robot path optimization technique based on reinforcement learning algorithm in warehouse environment. Appl Sci 11(3). https://doi.org/10.3390/app11031209
22. Macenski S, Martin F, White R, Clavero JG (2020) The Marathon 2: a navigation system. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)
23. Michels J, Saxena A, Ng A (2005) High speed obstacle avoidance using monocular vision and reinforcement learning. In: International conference on machine learning (ICML), Bonn, Germany
24. Morgan S, Branicky M (2004) Sampling-based planning for discrete spaces. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Sendai, Japan
25. Morse PM, Feshbach H, Harnwell GP (1953) Methods of theoretical physics. McGraw-Hill
26. Munoz P, Barrero D, R-Moreno M (2012) Run-time analysis of classical path-planning algorithms. In: Research and development in intelligent systems, pp 137–148
27. Nguyet T, Hoai T, Thi N (2011) Some advanced techniques in reducing time for path planning based on visibility graph. In: International conference on knowledge and systems engineering (IEEE), Hanoi
28. Palm R, Bouguerra A (2013) Particle swarm optimization of potential fields for obstacle avoidance. In: RARM
29. Palm R, Driankov D (2014) Fluid mechanics for path planning and obstacle avoidance of mobile robots. In: AASS, Orebro University

30. Reke M, Peter D, Schulte-Tigges J, Schiffer S, Ferrein A, Walter T, Matheis D (2020) A self-driving car architecture in ROS2. In: Robotics and mechatronics conference (RobMech)
31. RPalm, Bouguerra A (2011) Navigation of mobile robots by potential field methods and market-based optimization. In: European conference on mobile robots, Oerebro, Sweden
32. Stenger F, Tucker D, Baumann G (2016) Navier-Stokes equations on R3x[0, T]. Springer
33. Streinu I (2020) A combinatorial approach to planar non-colliding robot arm motion planning. In: Foundations of computer science (IEEE), CA, USA
34. White F (2002) Fluid mechanics. McGraw Hill
35. Wu Z, Hu G, Feng L, Wu J, Liu S (2016) Collision avoidance for mobile robots based on artificial potential field and obstacle envelope modelling. Assem Autom 36(3)
36. Xie W, Fang X, Wu S (2020) 2.5D navigation graph and improved a-star algorithm for path planning in ship inside virtual environment. In: Prognostics and system health management conference (PHM)
37. Yang L, Qi J, Song D, Xiao J, Han J, Xia Y (2016) Survey of robot 3D path planning algorithms. J Control Sci Eng 1–22
38. Zhang L, Zhang Y, Li Y (2020) Path planning for indoor mobile robot based on deep learning. Optik

Modeling and Simulation of Quadcopter Using Self-tuning Fuzzy-PI Controller

Arezki Fekik, Ahmad Taher Azar, Mohamed Lamine Hamida, Hakim Denoun, Sabrina Mohandsaidi, Amar Bousbaine, Nashwa Ahmad Kamal, Ibraheem Kasim Ibraheem, Amjad J. Humaidi, Ammar K. Al Mhdawi, and Alaa Khamis

Abstract Helicopters, commonly known as quadrotors (UAVs), are popular unmanned aerial vehicles. Despite their small size and high stability, they are used in a variety of applications. This chapter presents the fundamental principles for modeling and controlling quadcopters that will form the basis for future research and development in the field of drones. The problem is addressed on two fronts; first, the mathematical dynamic models are developed, and second, the trajectory of the quadcopter is stabilized and controlled. IMUs (Inertial Measurement Units) consist of accelerometers and gyroscopes and constitute the core of the system. In order to fly the quadcopter in six directions, it is necessary to determine the orientation of the system and control the speed of four BLDC motors. A Matlab/Simulink analysis of the quadcopter is performed. A self-tuning fuzzy-PI regulator is used to control the quadcopter's pitch, roll, and yaw. It was evaluated whether the quadcopter controller was effective and efficient, and the desired outputs were discussed.

A. Fekik (✉)
Department of Electrical Engineering, University Akli Mohand Oulhadj-Bouria, Rue Drissi Yahia Bouira 10000, Algeria
e-mail: a.fekik@univ-bouira.dz

A. Fekik · M. L. Hamida · H. Denoun
Electrical Engineering Advanced Technology Laboratory (LATAGE), Mouloud Mammeri University, Tizi Ouzou, Algeria

A. T. Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia
e-mail: aazar@psu.edu.sa; ahmad.azar@fci.bu.edu.eg; ahmad_t_azar@ieee.org

Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt

S. Mohandsaidi
Department of Electrical Engineering L2CSP Laboratory Mouloud Mammeri University, 15000 Tizi-Ouzou, Algeria

A. Bousbaine
University of Derby, Derby, UK
e-mail: A.Bousbaine@derby.ac.uk

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_8

Keywords Quadcopter · Unmanned aerial vehicles · Inertial measurement unit · Fuzzy-PI controller · Self-tuning

1 Introduction

Everyday life is increasingly dominated by electronic systems. Over the past few years, they have become increasingly integrated into everyday items, such as telephones, electronic diaries, cars, etc. A machine that has embedded systems that simplify our lives requires regulation or control in order to function properly. As a result of their nonlinearity, many of these systems are difficult to control. There are a number of aircraft, airships, and helicopters that are highly non-linear and require appropriate controls. Many of the systems mentioned above are under-actuated, which means that the input control parameters are fewer than their actual degree of freedom. Due to the lack of control parameters, designing controllers for these machines is extremely challenging. The four-propeller drones are among the most complex and difficult to control of these under-actuated systems. Robotics, automation, and aeronautics have all witnessed technical improvements in recent years [16, 19, 43, 67, 71]. As industrial applications have increased, UAVs have become increasingly popular in recent years. One of their advantages is that they expose personnel to less danger while covering a wide range of missions [3, 29, 41, 62, 64, 70].

Generally, they are used in military applications, such as reconnaissance and surveillance missions, where they can assist in the completion of those missions without putting human lives at risk. As a matter of fact, they are ideally suited to missions which could cause danger to crew members, require a permanent presence in an area, or may be laborious for human crew members [14, 46, 52, 63, 74].

N. A. Kamal
Faculty of Engineering, Cairo University, Giza, Egypt
e-mail: nashwa.ahmad.kamal@gmail.com

I. K. Ibraheem
Department of Computer Techniques Engineering, Dijlah University College, Baghdad 10001, Iraq
e-mail: ibraheemki@coeng.uobaghdad.edu.iq

A. J. Humaidi
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
e-mail: amjad.j.humaidi@uotechnology.edu.iq

A. K. Al Mhdawi
Department of Computer Science, Edge Hill University, Ormskirk, UK
e-mail: Al-Mhdaa@edgehill.ac.uk

A. Khamis
General Motors Canada, 500 Wentworth St W, Oshawa, ON L1J 6J2, Canada
e-mail: alaa.khamis@gm.com

Initially, they were used for observation tasks, but their use soon expanded to include target acquisition, electronic warfare, and target destruction. It is becoming increasingly common for civilian applications to be developed, such as highway traffic surveillance, forest fire prevention, meteorological data collection, and the inspection of engineering structures [29, 64].

The size and propulsion mode of drones vary depending on their purpose, which can range from centimeters to several meters. A canopy can be either fixed or rotating depending on whether the aircraft is flying forward or hovering. Unmanned aerial vehicles (UAVs) refer to both large autonomous aircraft and small flying machines that are controlled from the ground via a wireless connection with no pilot required. An unmanned aerial vehicle (UAV) is generally referred to as a drone /cite18. UAVs incorporate new technologies in terms of sensors, microcontrollers, control software, communication hardware, and user interfaces [13, 38]. According to researchers, UAVs can be categorized into fixed-wing and rotorcraft /cite21. A quadrotor is similar to a miniature helicopter with four motors. It is a system with six degrees of freedom, which is generally piloted with power and angle related commands (pitch, roll and yaw) [30, 45, 47]. As a result of its very light structure and sensitivity to external disturbances, such a machine is difficult to pilot. Due to its complexity, the quadcopter has attracted the attention of the research community. In most studies, a basic dynamic model of the quadcopter is used, but more complex aerodynamic characteristics have also been included [32, 34]. There have been various control methods studied, including PID regulators, backstepping control [15], H_∞ nonlinear command [18], LQR regulators [54], and nested saturation nonlinear controllers. For the Control methods to be implemented, it is necessary to perform certain measurements, such as gyroscope, accelerometer, GPS, sonar, or laser sensors, with precise accuracy [5, 27].

As a foundation for further research and development in this area, this chapter presents the fundamentals of quadcopter modeling and control. Two goals are pursued in this process. It is the first goal of the project to develop a dynamic analytical model of the quadcopter. A second objective is to develop a technique for stabilizing and controlling the quadcopter's trajectory. The purpose of this research is to present the differential equations for the dynamics of the quadcopter. Simulating the flight of a quadcopter is used to study the performance of the model. The quadcopter is stabilized using a PI-Flou controller. A PI-Flou controller uses fuzzy logic to adjust the parameters of the classical PI controller automatically. Using Matlab/Simulink and a proposed corrector for stabilizing the quadcopter, a mathematical model of the quadcopter is presented in this work. The pitch, roll and yaw motions of the quadcopter responses are achieved via Matlab/Simulink.

2 Related Work

Recently, there are several studies that dealt with quadrotor, Estimating and dealing with external disturbances, the pairing of the disturbance observer with sliding mode controllers is proposed in [53]. The author of [73] develop a path-following solution for a quad rotor copter with an unknown vehicle characteristics (mass and moment of inertia), as well as external disturbances. The authors of [68], presented the design of a quadrotor using l_1 adaptive controller design, where the control parameters are analytically determined through instinctively chosen performance and robustness metrics established by the designer. In [50], employed a Nonlinear Model Predictive Control (NMPC) to work out the path following issue for UAVs The proposed adaptive control system ensures the following by utilizing the backstepping techniqueis presented in [73], the relative to trajectory tracking, a smoother convergence is reached because the control signals of the actuators for thrust force and torque are limited relative to the position error. This work [54], provid an examination and overall performance of an LQR control algorithm for quadrotor. A robust observer-based nonlinear controller had been established in [33] for quadrotors with consideration of systems' uncertainties and exterior disturbances. The tracking errors are limited by using two dissimilar nonlinear disturbance observers that had been constructed to manage the uncertainties in the system. The paper [4] discusses the UAV's modelling and dynamic control using an interval type-2 (IT2) Takagi-Sugeno (TS) fuzzy system. A quadcopter's dynamics are complicated and nonlinear, as well as underactuated and unpredictable, making modelling and control activities through its whole flying shell difficult. The use of the Newton-Euler method to extract a mathematical model of an undriven six degree of freedom (6 DoF) quadcopter is covered by the author [40]. The actuation forces are also taken into account by simulating the forces and aerodynamic factors. A detailed derivation of the Quadcopter mathematical model is offered. The model of the quadcopter is separated into two sub-systems: the rotational sub-system (roll, pitch, and yaw) and the translational sub-system (Pitch, roll, and yaw) (altitude, x and y positions). A hybrid control system had been designed in [1]to expand the disturbances rejection ability and body jerk recital for a quadrotor model by exploiting an active force control (AFC). The developed intelligent control method incorporated a PID controller and an intelligent active force control (IAFC) component resulting in a robust PID-IAFC system The translational sub-system is underactuated, but the rotating sub-system is fully operational. A PID regulator will be used to control the model. Auto tuning is used to obtain the PID gains, and manual tuning is used to fine tune them. To follow the in- tended trajectory workspace and solve the attitude control problem of a quadcopter UAV, a fractional Active Disturbance Rejection Control (ADRC) is presented in the research [22], The ADRC is a reliable nonlinear control that has lately been utilized to control UAV technologies. It is primarily utilized to address significant critical difficulties such as uncertainties, imprecise modelling, and external disturbances in a basic structure and straightforward manner while setting control parameters. The ADRC control strategy's fundamental idea is to create a fictional state variable that represents the whole disturbance. Then, using an Extended State Observer, this dis- turbance is estimated (ESO). The estimated disturbance is supplied back into the system to build a suitable controller, decoupling it from any uncertainties and dis- turbances affecting the plant.

3 Dynamical Model of the Quadrirotor

There is no doubt that quadrotors are among the most complex flying systems. Due to this, the modelling of the dynamics of the flight of the quad-rotor can be challenging and delicate, as well as a strongly nonlinear dynamic system for interacting between the various states [21]. In the literature, several hypotheses have been proposed to simplify this process, including:

- The mechanical structure of the quadrotor is assumed to be rigid and symmetrical, this gives a diagonal matrix.
- Due to the rigidity of quadcopters' propellers, deformation during rotation should have little effect on them.
- There is a direct correlation between the center of mass of the quadrotor and the origin of the reference frame connected to the earth.
- As the rotational speed of the actuators increases, so do the drag and lift forces.

3.1 Rotational Matrix

According to Fig. 1, two fiducials are required to report the position and orientation of the quadcopter. First, there is the inertial frame (E-frame). A Galilean reference frame is a fixed orthogonal reference frame. In order to depict the orientation of the quadcopter, a second reference frame must be defined. Affixed to the quadcopter's frame, this one moves along with the machine. The mobile reference frame (B-frame) is also known as the mobile reference frame. The moving frame starts off coincident with the fixed frame of reference, then makes a gyratory movement around the x-axis of the angle between the moving frame and the fixed frame of reference as $-2\pi < \phi < 2\pi$, followed by a rotation around the y-axis of an angle between $-2\pi < \theta < 2\pi$, for the pitch, and finally a rotational around the z-axis of an angle between $-2\pi < \psi < 2\pi$ for the yaw [39].

The formula for the rotation matrix R is represented by the equation below [9]:

$$\begin{aligned} R &= Rot_z(\psi)xRot_y(\theta)xRot_x(\phi) \\ &= \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \end{aligned} \tag{1}$$

$$R = \begin{bmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\phi & c\phi s\theta c\psi - s\psi c\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2}$$

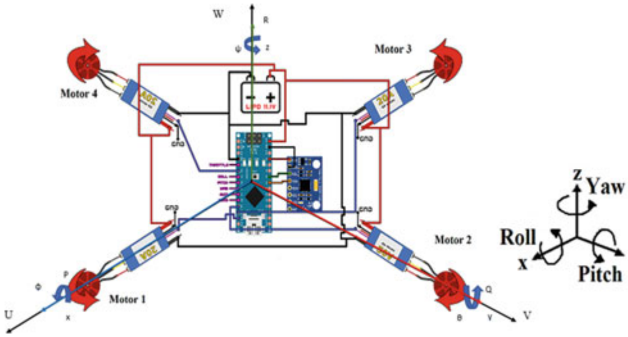


Fig. 1 Tracking of a quadrotor

with:
$c = \cos$
$s = \sin$.

4 Rotational Speed

By utilizing the rotational velocities Ω_1; Ω_2; Ω_3 in the moving frame, we translate the rotational velocities $\dot{\phi}$; $\dot{\theta}$; $\dot{\psi}$ in the fixed frame as follows:

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + Rot_x(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + Rot_y(\theta)Rot_x(\phi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{3}$$

During the period of confusion between the reference frames, the rotation in roll occurs. In order to perform pitching, the vector representing rotation must be expressed in the fixed reference frame: it must therefore be multiplied by $Rot(\phi))^{-1}$. Similarly, the yaw rotation vector must be expressed in the fixed reference frame that has already undergone two revolutions. Therefore, we arrive at the following conclusion:

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} c\phi \\ \dot{\theta} s\phi \end{bmatrix} + \begin{bmatrix} -\dot{\psi} s\theta \\ \dot{\psi} s\phi c\theta \\ \dot{\psi} c\phi c\theta \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} s\theta \\ \dot{\theta} c\phi + c\dot{\psi} s\phi c\theta \\ \dot{\psi} c\phi c\theta - \dot{\theta} s\phi \end{bmatrix} \tag{4}$$

$$\Omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5}$$

So

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi \tan\theta & c\phi \tan\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} * \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \tag{6}$$

When the quadrotor does little rotations, we can make the below approximations
$c\phi = c\theta = c\psi = 1$
$s\phi = s\theta = s\psi = 0$
The relation (7) is obtained:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \left[\Omega_x \ s\Omega_y \ \Omega_z\right]^T \tag{7}$$

4.1 Travelling Speeds

The linear velocities in the fixed frame V_x^b ; V_y^b; V_z^b are expressed in the context of the linear velocity in the moving plan V_x^m; V_y^m; V_z^m by the relation (8):

$$V = \begin{bmatrix} V_x^b \\ V_y^b \\ V_z^b \end{bmatrix} = R * \begin{bmatrix} V_x^m \\ V_y^m \\ V_z^m \end{bmatrix} \tag{8}$$

4.2 Physical Effects Acting on the Quad-Copter

Once the drone is in motion, its dynamics is subject to certain forces and moments as illustrated in Fig. 2.

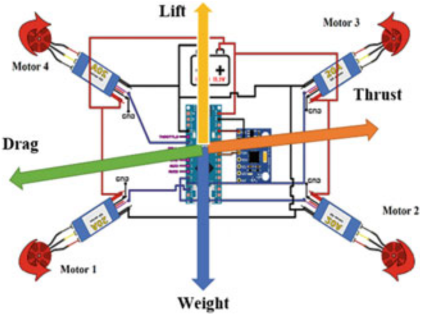


Fig. 2 Forces applied to a drone

4.2.1 Forces

1. Weight Force:
   The weight forces is given by relation (9)
   $$P = m * g \tag{9}$$
   Where m is the quadrotor's overall weight and g is gravity.
2. Thrust forces:
   The forces generated by the rotation of the motors are perpendicular to the axis plane and proportional to their rotational speed:
   $$F_i = b * \omega_i^2 \tag{10}$$
   In this equation, *i* represents the number of motors, whilst *b* represents the coefficient of lift, which depends on the shape of the blades and the density of the air.
3. Drag forces:
   In a heavy fluid, drag acts like friction and opposes the movement of a body. As a result, the system is subject to two drag forces:
   – Propellers' drag: This force acts on the blades and is proportional to the density of the air, the blade size, and the square of the propeller's rotational speed, and is calculated as follows: Eq. 11:
   $$T_i = d * \omega_i^2 \tag{11}$$
   where d is the drag coefficient which is a function of the propeller design.
   – The drag along the x, y, and z alignments: it is due to the movement of the quadrotor copter:
   $$F_t = K_{ft} * V \tag{12}$$
   where K_{ft} is the translational drag coefficient and V the linear velocity.

4.2.2 Moments

There are several moments acting on a quadcopter, which are due to thrust and drag forces and gyroscopic effects.

1. Moments due to thrust forces:
   It is due to the moment produced by the resultant lift forces between the right and left motors that the turning movement around the x-axis occurs, and the moment is given by the following equation:
   $$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \tag{13}$$

where l is the quadcopter's arm's length from its rotor to its centre of gravity.
– There is a rotation around the y axis caused by the resultant of lifting forces differing between the front and rear motors, and the moment is given by the following equation:

$$M_y = l(F_3 - F_1) = lb(\omega_3^2 - \omega_1^2) \tag{14}$$

2. Moments due to drag forces:
– The rotation around the z axis: due to a reactive torque caused by the drag torques in each propeller results in moment is given by Eq. 15:

$$M_z = d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \tag{15}$$

– Aerodynamic friction moment is given by Eq. 16:

$$M_a = k_{fa}\Omega^2 \tag{16}$$

where K_{fa} a is the coefficient of aerodynamic friction and is the angular velocity

4.2.3 Gyroscopic Effects

The gyroscopic effect occurs when an object rotates around an axis. It is the ability of an object to maintain its axis of rotation or, more precisely, its angular momentum. In a quadrotor setup, there are two gyroscopic moments. A gyroscopic moment is produced by the propellers, and a gyroscopic moment is produced by the movement of the quadcopter. The following formula provides the gyroscopic moment of the propellers:

$$M_{gh} = \sum_{i=1}^{4} \Omega \wedge J_r \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+1}\omega_i \end{bmatrix} \tag{17}$$

With J_r is the inertia of the rotors. Gyroscopic moment due to quadrotor movements it is given by the following relation:

$$M_{gh} = \Omega \wedge J\Omega \tag{18}$$

With J is the inertia of the system.

5 Proposed Strategy

5.1 Literature Review

The Proportional, Integral, and Derivative (PID) regulator or corrector is a control algorithm that improves the efficiency of closed-loop control systems [10, 24, 35, 36, 51, 57, 59, 60]. In several fields, its correction properties are applicable to a number of physical quantities [7, 23, 31, 37, 44, 66, 72]. The following are some of the advantages of this controller [6, 55, 56, 61]

- Simple structure;
- Good performance in many processes;
- Reliable, even without a specific model of the control system.

Corrections are based on the observed error, which is typically calculated as the difference between the setpoint and the measurement.

$$e = setpoint - measurement \tag{19}$$

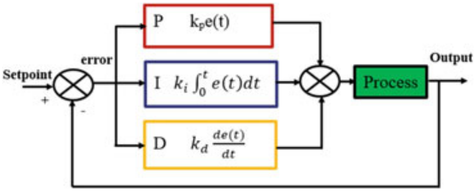
The PID allows three actions depending on this error:

- Increasing the speed of the system by multiplying the error by a gain K_p. As K_p increases, the response time decreases.
- The integral action consists of integrating the error over a time interval s, dividing by a gain K_i, and eliminating the residual error in steady state. This improves accuracy, however it also increases the phase shift.
- It is a derivative action: the error is derived according to a time t, then multiplied by a gain K_d that will accelerate the system's response and improve the loop's stability.

In the following Fig. 3, the classical PID structure is shown: The general expression of the corrector can be expressed by the Eq. 20 form [23, 37].

$$U(t) = k_p e(t) + k_i \int_0^t e(t)\,\mathrm{d}t + k_d \frac{de(t)}{dt} \tag{20}$$

Fig. 3 PID-corrector



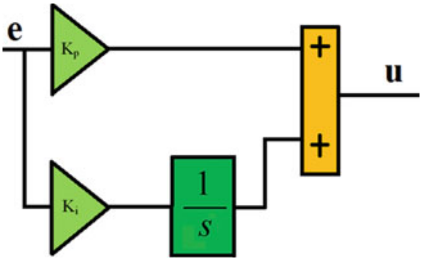


Fig. 4 PI Controller

5.2 PI Controller

According to equation [17], the error between the setpoint and the measurement is integrated with respect to time and multiplied by a constant that must be determined according to the method employed. In this application, the PI-Flou algorithm was used to implement the x, y, and z axes as given by equation [17]:

$$U(t) = k_p e(t) + k_i \int_0^t e(t)\,\mathrm{d}t \tag{21}$$

The following is a representation of the transfer function:

$$C(s) = k_p + \frac{k_i}{s} \tag{22}$$

where the PI controller's block design is shown in Fig. 4.

5.3 Fuzzy-PI Controller with Self-tuning

It is possible to control and carry out the governance of a fuzzy logic-based system using a philosophy known as fuzzy logic control [2, 7, 8, 11, 12, 20, 24, 28, 42, 44, 48, 49, 58, 65, 69]. It is also possible to create nonlinear controller using heuristic data in this manner. It is therefore possible to combine the benefits of both fuzzy and PI controller to provide excellent control performance through the use of one regulator. Based on errors on the three axes (x, y, and z) and fuzzy logic error changes, the fuzzy adaptive PI regulator adjusts the parameters of the PI regulator. The distance error and its variation along the three axes serve as the fuzzy model's input. In the fractional PI regulator, K_p and K_i represent the outputs [25, 26]. Mamdani's fuzzy logic model is a prerequisite for the proposed fuzzy logic PI regulator (Tepljakov 2017; Mamdani Assilian 1999). Figure 5 illustrates the framework for the proposed regulator. Fuzzy inference is used to modify the PI regulator based on the nonlinear mapping between tracking error and its rate of change. Fuzzification, fuzzy rules, aggregation, and

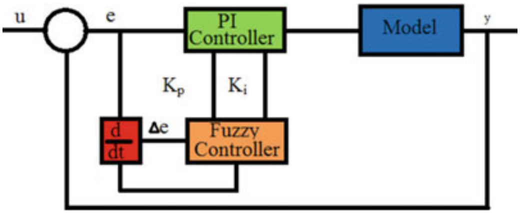


Fig. 5 Design of the Fuzzy PI Corrector

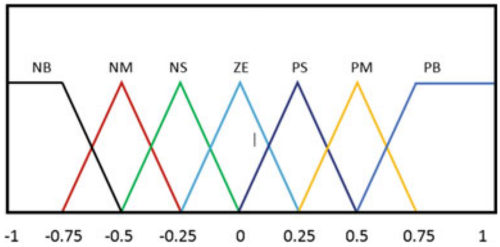


Fig. 6 Fuzzy input membership functions of e and Δe

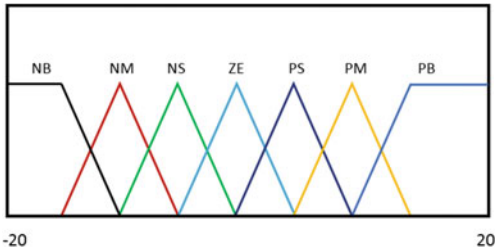


Fig. 7 Fuzzy output membership functions of gains K_p and K_i

defuzzification are the elements of the fuzzy-PI regulator framework. To represent the tracking error e(t) and its derivative e(t) (Positive Big), fuzzy linguistic variables NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZO (Zero), PS (Positive Small), PM (Positive Medium), and PB are used. In Fig. 6, five triangular membership functions and two trapezoidal membership functions are used to define each input.

According to the Mamdani rule, each output of the controller is a gain K_p or K_i, as illustrated in Fig. 7 [25, 26]. The triangle membership functions and trapezoidal membership functions are in the range [−20, 20]. A summary of the fuzzy control rules is presented in Tables 1 and 2. They are composed of the seven language mistake and error variation values [25, 26].

Tables 1 and 2 present a summary of the fuzzy control rules. They are composed of the seven language mistake and error variation values [25, 26].

Table 1 Languistic rules for K_p

$\varDelta E(k)\vert E(k)$	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZE	ZE
NM	PB	PB	PM	PS	PS	ZE	ZE
NS	PM	PM	PM	PM	ZE	ZE	ZE
ZE	PM	PM	PS	ZE	NS	NS	NM
PS	PS	PS	ZE	NS	NS	NM	NM
PM	PS	ZE	NS	NM	NM	NM	NB
PB	ZE	ZE	NM	NM	NM	NB	NB

Table 2 Languistic rules for K_i

$\varDelta E(k)\vert E(k)$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZE	ZE
NM	NB	NB	NM	NS	NS	ZE	ZE
NS	NB	NM	NM	NS	ZE	PS	PS
ZE	NM	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PS	PM	PB
PM	ZE	ZE	PS	PS	PM	PB	PB
PB	ZE	ZE	PS	PM	PM	PB	PB

6 Results and Interpretations

A simulation of the dynamic movements of the quad-copter is conducted using Matlab software incorporating the proposed controllers for the three axes (x, y, z) that create torque rotation angles along the three axes (x, y, z) namely (ϕ, θ, ψ). To demonstrate the efficacy of the proposed controller, two sets of tests are conducted using a conventional PID controller.

6.1 Test1

In this test, the three axes (x, y, z) respond in a fixed manner. Figure 8 illustrates the fixed response of the axis (x) of the quad-rotor drone at a 4 m setting, and the drone follows the instruction perfectly within the acceptable time limits for the axis x that was initially set at 5 m. Furthermore, the static error is zero. Refer to Fig. 9 for the fixed response of the quadrotor UAV over a distance of three meters. Assume that the initial condition for Y was 0 m, and that the static error is zero, the UAV follows the given instruction perfectly along the x axis with an acceptable time.

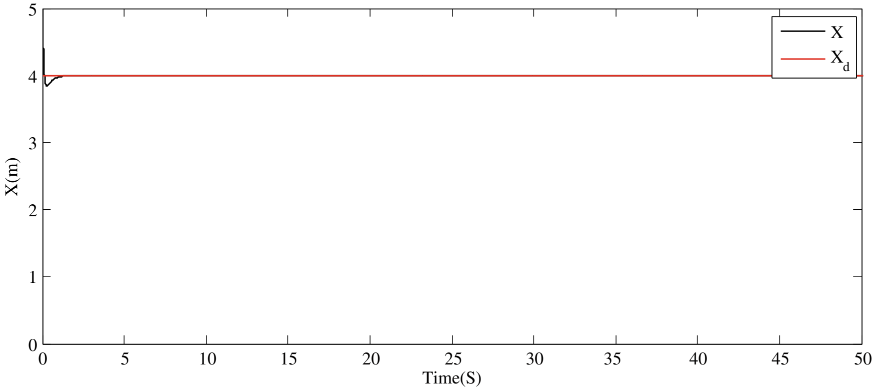


Fig. 8 X-axis response

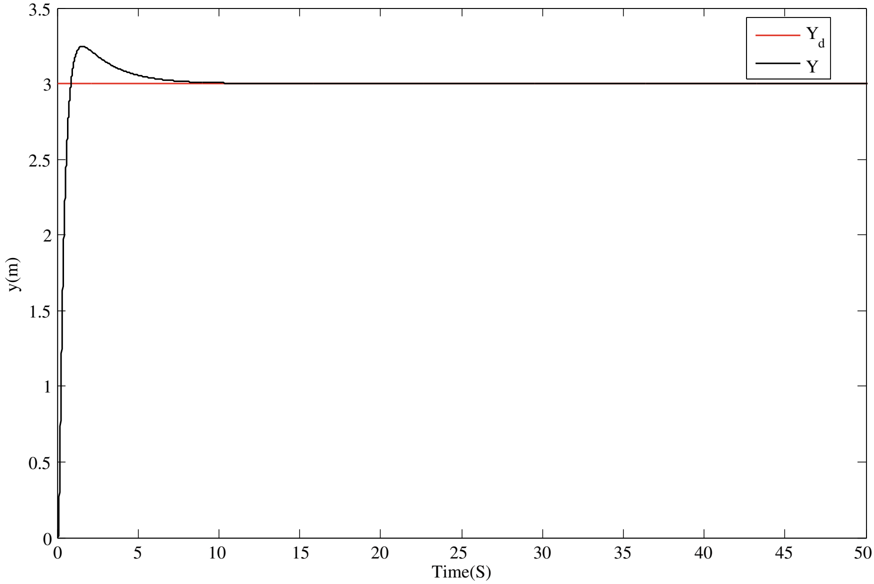


Fig. 9 Y-axis response

According to Fig. 10, the quadrotor UAV follows the given instructions perfectly along the Z axis, knowing that the initial condition along Z was zero m, and that the static error is zero.

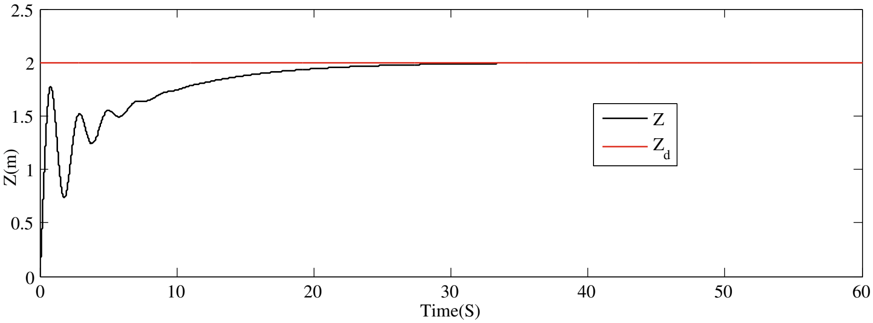


Fig. 10 Z-axis response

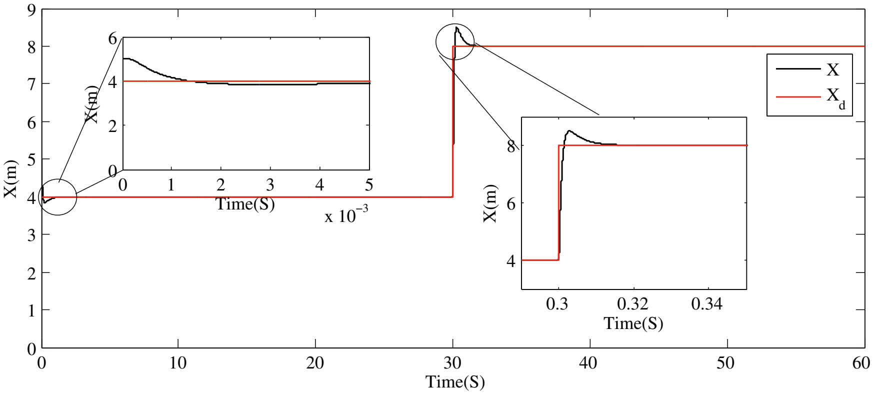


Fig. 11 X-axis response with variation reference

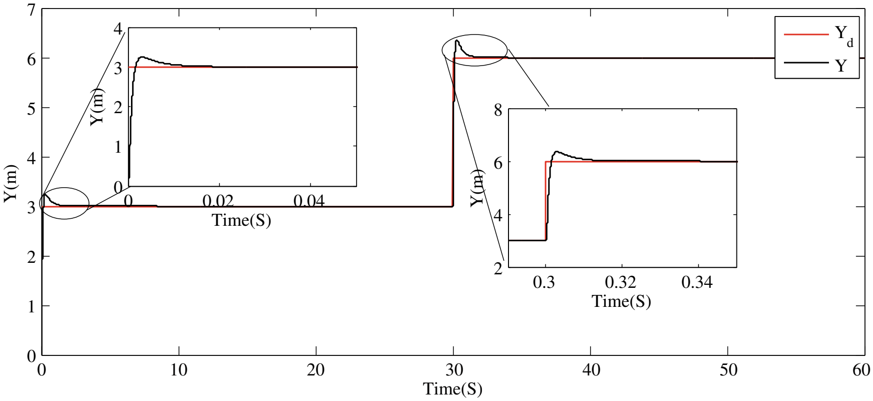


Fig. 12 Y-axis response with variation reference

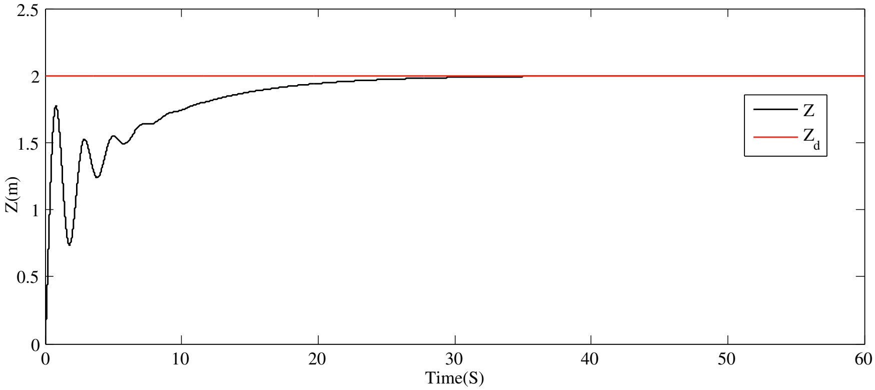


Fig. 13 Z-axis response with variation reference

6.2 *Test2*

To test whether the proposed regulator is effective, the second test focuses on a response to a variation in the references of the three axes (x, y, and z) (see Figs. 11, 12, and 13). During the experiment, the x-axis was varied between 4m and 8m and the y-axis was varied between 3m and 6m, while the z-axis was kept fixed at a value of two meters. According to the results obtained, the settings reached all their references rapidly, and the curves are smooth and do not exhibit oscillations in steady state, despite their appearance in transient regimes, which are an ordinary phenomenon when controlling nonlinear systems of this type. Despite these oscillations, the system continues to function normally, and the maximum overshoots recorded are comparable to the results of previous studies. By comparing the input data with the requested outputs, it is evident that the proposed controller has successfully controlled the system and the errors generated are low on each of the three axes (x, y, z). It is important to note that the time elapsed between the steady state and the stability is insufficient. A multivariable and nonlinear system such as this one exhibits a very small transient regime for the three axes, which is considered to be low for such a system. Thus, we can conclude that these results are very satisfactory and that this technique (PI-flou) has enabled us to produce a well-adapted controller for this type of system and that it can be applied efficiently for the control of complex nonlinear systems where classical controls have been unsuccessful.

7 Conclusions

In recent years, quadcopters have become very popular due to the growing interest in unmanned aerial vehicles (UAVs), vertical take-off and landing aircraft (VTOLs), and the need for lightweight instruments that are unobtrusive. This is the primary motivation for this investigation. An important objective of this study was to develop a mathematical model of minimal complexity and flight control laws that ensure the stability of a quadcopter. To obtain a simulation that is as close to the real behavior as possible, we began this chapter by identifying the parameters of the quadcopter. In order to stabilize and improve the accuracy and speed of the system, we synthesized PI Fuzzy controllers. In terms of response time and static error, the synthesized controls were applied to the simulation model and gave good results. The following tasks can be addressed in future work in order to develop and improve this work:

- Adapt the studied control algorithms (PI-flou) to a faster and more accurate board, such as the Raspberry Pi, FPGA …
- By implementing embedded image processing, a wide range of imaginable activities would be possible (e.g. detecting specific shapes, tracking objects, etc.).

Acknowledgements The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for supporting this work. Special acknowledgement to Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

References

1. Abdelmaksoud SI, Mailah M, Abdallah AM (2020) Robust intelligent self-tuning active force control of a quadrotor with improved body jerk performance. IEEE Access 8:150,037–150,050
2. Abdelmalek S, Azar AT, Dib D (2018) A novel actuator fault-tolerant control strategy of DFIG-based wind turbines using Takagi-Sugeno multiple models. Int J Control Autom Syst 16(3):1415–1424
3. Adepoju O (2022) Drone/unmanned aerial vehicles (UAVs) technology. In: Re-skilling human resources for construction 4.0. Springer, pp 65–89
4. Al-Mahturi A, Santoso F, Garratt MA, Anavatti SG (2021) Modeling and control of a quadrotor unmanned aerial vehicle using type-2 fuzzy systems. In: Unmanned aerial systems. Elsevier, pp 25–46
5. Alaiwi Y, Mutlu A (2018) Modelling, simulation and implementation of autonomous unmanned quadrotor. Mach Technol Mater 12(8):320–325
6. Ammar HH, Azar AT (2020) Robust path tracking of mobile robot using fractional order PID controller. In: Hassanien AE, Azar AT, Gaber T, Bhatnagar R, Tolba MF (eds) The international conference on advanced machine learning technologies and applications (AMLTA2019). Advances in intelligent systems and computing, vol 921. Springer International Publishing, Cham, pp 370–381
7. Ammar HH, Azar AT, Tembi TD, Tony K, Sosa A (2018) Design and implementation of fuzzy PID controller into multi agent smart library system prototype. In: Hassanien AE, Tolba MF, Elhoseny M, Mostafa M (eds) The international conference on advanced machine learning technologies and applications (AMLTA2018). Advances in intelligent systems and computing, vol 723. Springer International Publishing, Cham, pp 127–137
8. Ananth DVN, Kumar LVS, Gorripotu TS, Azar AT (2021) Design of a fuzzy logic controller for short-term load forecasting with randomly varying load. Int J Sociotechnol Knowl Dev (IJSKD) 13(4):32–49
9. Avant T, Lee U, Katona B, Morgansen K (2018) Dynamics, hover configurations, and rotor failure restabilization of a morphing quadrotor. In: 2018 Annual American control conference (ACC). IEEE, pp 4855–4862

10. Azar AT, Serrano FE (2014) Robust IMC-PID tuning for cascade control systems with gain and phase margin specifications. Neural Comput Appl 25(5):983–995
11. Banu PN, Azar AT, Inbarani HH (2017) Fuzzy firefly clustering for tumour and cancer analysis. Int J Modell Identif Control 27(2):92–103. https://www.inderscienceonline.com/doi/pdf/10.1504/IJMIC.2017.082941
12. Barakat MH, Azar AT, Ammar HH (2020) Agricultural service mobile robot modeling and control using artificial fuzzy logic and machine vision. In: Hassanien AE, Azar AT, Gaber T, Bhatnagar R, Tolba MF (eds) The international conference on advanced machine learning technologies and applications (AMLTA2019). Advances in intelligent systems and computing, vol 921. Springer International Publishing, Cham, pp 453–465
13. Boursianis AD, Papadopoulou MS, Diamantoulakis P, Liopa-Tsakalidi A, Barouchas P, Salahas G, Karagiannidis G, Wan S, Goudos SK (2020) Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: a comprehensive review. Internet of Things, p 100187
14. Çetin E, Cano A, Deransy R, Tres S, Barrado C (2022) Implementing mitigations for improving societal acceptance of urban air mobility. Drones 6(2):28
15. Chen F, Lei W, Zhang K, Tao G, Jiang B (2016) A novel nonlinear resilient control for a quadrotor UAV via backstepping control and nonlinear disturbance observer. Nonlinear Dyn 85(2):1281–1295
16. Christensen H, Amato N, Yanco H, Mataric M, Choset H, Drobnis A, Goldberg K, Grizzle J, Hager G, Hollerbach J et al (2021) A roadmap for US robotics–from internet to robotics 2020 edition. Now Publishers
17. Coelho MS et al (2019) Hybrid PI controller constructed with paraconsistent annotated logic. Control Eng Pract 84:112–124
18. Dasgupta R (2018) Adaptive attitude tracking of a quad-rotorcraft using nonlinear control hierarchy. In: 2018 IEEE recent advances in intelligent computational systems (RAICS). IEEE, pp 177–181
19. Dirican C (2015) The impacts of robotics, artificial intelligence on business and economics. Procedia Soc Behav Sci 195:564–573
20. Emary E, Zawbaa HM, Hassanien AE, Schaefer G, Azar AT (2014) Retinal vessel segmentation based on possibilistic fuzzy c-means clustering optimised with cuckoo search. In: 2014 international joint conference on neural networks (IJCNN). IEEE, pp 1792–1796
21. Emran BJ, Najjaran H (2018) A review of quadrotor: an underactuated mechanical system. Annu Rev Control 46:165–180
22. Ennima S, Bourekkadi S, Ourdi A, Elgharad A (2021) Innovation attitude control of a hexacopter platform based on fractional control laws and comparison with the PID and LQR control methods. J Theoret Appl Inf Technol 99(9)
23. Fadel MZ et al (2019) Modeling, simulation and control of a fly-by-wire flight control system using classical PID and modified PI-D controllers. J Eur Syst Autom 52(3):267–276
24. Fekik A, Denoun H, Azar AT, Koubaa A, Kamal NA, Zaouia M, Hamida ML, Yassa N (2020) Adapted fuzzy fractional order proportional-integral controller for DC motor. In: 2020 first international conference of smart systems and emerging technologies (SMARTTECH), pp 1–6. https://doi.org/10.1109/SMART-TECH49988.2020.00019
25. Fekik A, Hamida ML, Houassine H et al (2022) Observability of speed DC motor with self-tuning fuzzy-fractional-order controller. In: Fractional-order design. Academic Press, pp 157–179
26. Fekik A et al (2020) Adapted fuzzy fractional order proportional-integral controller for DC motor. In: 2020 first international conference of smart systems and emerging technologies (SMARTTECH). IEEE, pp 1–6
27. García J, Molina JM, Trincado J (2020) Real evaluation for designing sensor fusion in UAV platforms. Inf Fusion 63:136–152
28. Ghoudelbourk S, Azar AT, Dib D (2021) Three-level (NPC) shunt active power filter based on fuzzy logic and fractional-order PI controller. Int J Autom Control 15(2):149–169

29. Giordan D, Adams MS, Aicardi I, Alicandro M, Allasia P, Baldo M, De Berardinis P, Dominici D, Godone D, Hobbs P et al (2020) The use of unmanned aerial vehicles (UAVs) for engineering geology applications. Bull Eng Geol Env 79(7):3437–3481
30. González-Jorge H, Martínez-Sánchez J, Bueno M et al (2017) Unmanned aerial systems for civil applications: a review. Drones 1(1):2
31. Gorripotu TS, Samalla H, Jagan Mohana Rao C, Azar AT, Pelusi D (2019) TLBO algorithm optimized fractional-order PID controller for AGC of interconnected power system. In: Nayak J, Abraham A, Krishna BM, Chandra Sekhar GT, Das AK (eds) Soft computing in data analytics. Springer, Singapore, pp 847–855
32. Hoffmann G, Huang H, Waslander S, Tomlin C (2007) Quadrotor helicopter flight dynamics and control: theory and experiment. In: AIAA guidance, navigation and control conference and exhibit, p 6461
33. Hua H, Fang Y, Zhang X, Lu B (2020) A novel robust observer-based nonlinear trajectory tracking control strategy for quadrotors. IEEE Trans Control Syst Technol 29(5):1952–1963
34. Huang H, Hoffmann GM, Waslander SL, Tomlin CJ (2009) Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: 2009 IEEE international conference on robotics and automation. IEEE, pp 3277–3282
35. Humaidi AJ, Najem HT, Al-Dujaili AQ, Pereira DA, Ibraheem IK, Azar AT (2021) Social spider optimization algorithm for tuning parameters in PD-like interval type-2 fuzzy logic controller applied to a parallel robot. Meas Control 54(3–4):303–323
36. Ibraheem GAR, Azar AT, Ibraheem IK, Humaidi AJ (2020) A novel design of a neural network-based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization. Complexity 2020:1–18
37. Johnson MH, Michael A et al (2005) PID control
38. Jung D, Tsiotras P (2007) Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. In: AIAA infotech@ aerospace 2007 conference and exhibit, p 2768
39. Kahouadji M, Mokhtari MR, Choukchou-Braham A, Cherki B (2020) Real-time attitude control of 3 DOF quadrotor UAV using modified super twisting algorithm. J Franklin Inst 357(5):2681–2695
40. Kandeel HM et al (2022) Modeling and control of x-shape quadcopter. IOSR J Mech Civil Eng (IOSR-JMCE) 19(1):46–57
41. Khan NA, Jhanjhi N, Brohi SN, Usmani RSA, Nayyar A (2020) Smart traffic monitoring system using unmanned aerial vehicles (UAVs). Comput Commun 157:434–443
42. Khettab K, Bensafia Y, Bourouba B, Azar AT (2018) Enhanced fractional order indirect fuzzy adaptive synchronization of uncertain fractional chaotic systems based on the variable structure control: Robust h∞ design approach. In: Azar AT, Radwan AG, Vaidyanathan S (eds) Mathematical techniques of fractional order systems. Advances in nonlinear dynamics and chaos (ANDC). Elsevier, pp 597–624
43. Koubâa A, Azar AT (2021) Unmanned aerial systems: theoretical foundation and applications. Academic Press
44. Kumar J, Azar AT, Kumar V, Rana KPS (2018) Design of fractional order fuzzy sliding mode controller for nonlinear complex systems. In: Azar AT, Radwan AG, Vaidyanathan S (eds) Mathematical techniques of fractional order systems. Advances in nonlinear dynamics and chaos (ANDC). Elsevier, pp 249–282
45. Kumar KS, Rasheed M, Kumar RMM (2014) Design and implementation of fuzzy logic controller for quad rotor UAV. In: 2nd international conference on research in science, engineering and technology (ICRSET'2014). Dubai, pp 114–120
46. Maddikunta PKR, Hakak S, Alazab M, Bhattacharya S, Gadekallu TR, Khan WZ, Pham QV (2021) Unmanned aerial vehicles in smart agriculture: applications, requirements, and challenges. IEEE Sens J 21(16):17,608–17,619
47. Malpica C, Withrow-Maser S (2020) Handling qualities analysis of blade pitch and rotor speed controlled eVTOL quadrotor concepts for urban air mobility. In: VFS international powered lift conference, pp 21–23

48. Meghni B, Dib D, Azar AT (2017) A second-order sliding mode and fuzzy logic control to optimal energy management in wind turbine with battery storage. Neural Comput Appl 28(6):1417–1434
49. Meghni B, Dib D, Azar AT, Saadoun A (2018) Effective supervisory controller to extend optimal energy management in hybrid wind turbine under energy and reliability constraints. Int J Dyn Control 6(1):369–383
50. Mohammadi Daniali H (2020) Fast nonlinear model predictive control of quadrotors: design and experiments. Master's thesis, University of Waterloo
51. Najm AA, Azar AT, Ibraheem IK, Humaidi AJ (2021) A nonlinear PID controller design for 6-DOF unmanned aerial vehicles. In: Koubaa A, Azar AT (eds) Unmanned aerial systems. Advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 315–343
52. Nex F, Remondino F (2014) UAV for 3D mapping applications: a review. Appl Geomat 6(1):1–15
53. Nguyen NP, Mung NX, Thanh HLNN, Huynh TT, Lam NT, Hong SK (2021) Adaptive sliding mode control for attitude and altitude system of a quadcopter UAV via neural network. IEEE Access 9:40,076–40,085
54. Okyere E, Bousbaine A, Poyi GT, Joseph AK, Andrade JM (2019) LQR controller design for quad-rotor helicopters. J Eng 17:4003–4007
55. Pilla R, Azar AT, Gorripotu TS (2019) Impact of flexible AC transmission system devices on automatic generation control with a metaheuristic based fuzzy PID controller. Energies 12(21):4193
56. Pilla R, Botcha N, Gorripotu TS, Azar AT (2020) Fuzzy PID controller for automatic generation control of interconnected power system tuned by glow-worm swarm optimization. In: Nayak J, Balas VE, Favorskaya MN, Choudhury BB, Rao SKM, Naik B (eds) Applications of robotics in industry using advanced mechanisms. Springer International Publishing, Cham, pp 140–149
57. Pilla R, Gorripotu TS, Azar AT (2021) Design and analysis of search group algorithm-based PD-PID controller plus redox flow battery for automatic generation control problem. Int J Comput Appl Technol 66(1):19–35
58. Pintea CM, Matei O, Ramadan RA, Pavone M, Niazi M, Azar AT (2018) A fuzzy approach of sensitivity for multiple colonies on ant colony optimization. In: Balas VE, Jain LC, Balas MM (eds) Soft computing applications. Springer International Publishing, Cham, pp 87–95
59. Rana K, Kumar V, Sehgal N, George S, Azar AT (2021) Efficient maximum power point tracking in fuel cell using the fractional-order PID controller. In: Azar AT, Kamal NA (eds) Renewable energy systems. Advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 111–132
60. Saidi SM, Mellah R, Fekik A, Azar AT (2022) Real-time fuzzy-PID for mobile robot control and vision-based obstacle avoidance. Int J Serv Sci Manag Eng Technol 13(1):1–32
61. Sallam OK, Azar AT, Guaily A, Ammar HH (2020) Tuning of PID controller using particle swarm optimization for cross flow heat exchanger based on CFD system identification. In: Hassanien AE, Shaalan K, Tolba MF (eds) Proceedings of the international conference on advanced intelligent systems and informatics 2019. Advances in intelligent systems and computing, vol 1058. Springer International Publishing, Cham, pp 300–312
62. Samanta S, Mukherjee A, Ashour AS, Dey N, Tavares JMRS, Abdessalem Karâa WB, Taiar R, Azar AT, Hassanien AE (2018) Log transform based optimal image enhancement using firefly algorithm for autonomous mini unmanned aerial vehicle: an application of aerial photography. Int J Image Graph 18(04):1850019. https://doi.org/10.1142/S0219467818500195
63. Schneier M, Schneier M, Bostelman R (2015) Literature review of mobile robots for manufacturing. US Department of Commerce, National Institute of Standards and Technology
64. Shakhatreh H, Sawalmeh AH, Al-Fuqaha A, Dou Z, Almaita E, Khalil I, Othman NS, Khreishah A, Guizani M (2019) Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. IEEE Access 7:48,572–48,634
65. Shalaby R, Ammar HH, Azar AT, Mahmoud M (2021) Optimal fractional-order fuzzy-MPPT for solar water pumping system. J Intell Fuzzy Syst 40(1):(1):1175–1190

66. Soliman M, Azar AT, Saleh MA, Ammar HH (2020) Path planning control for 3-omni fighting robot using PID and fuzzy logic controller. In: Hassanien AE, Azar AT, Gaber T, Bhatnagar R, Tolba MF (eds) The international conference on advanced machine learning technologies and applications (AMLTA2019). Springer International Publishing, Cham, pp 442–452
67. Sun X, Wandelt S, Zhang A (2021) Technological and educational challenges towards pandemic-resilient aviation. Transp Policy 114:104–115
68. Thu KM, Gavrilov A (2017) Designing and modeling of quadcopter control system using L1 adaptive control. Procedia Comput Sci 103:528–535
69. Vaidyanathan S, Azar AT (2016) Takagi-Sugeno fuzzy logic controller for Liu-Chen four-scroll chaotic system. Int J Intell Eng Inf 4(2):135–150
70. Velusamy P, Rajendran S, Mahendran RK, Naseer S, Shafiq M, Choi JG (2022) Unmanned aerial vehicles (UAV) in precision agriculture: applications and challenges. Energies 15(1):217
71. Vidulich MA, Tsang PS (2019) Improving aviation performance through applying engineering psychology. Advances in aviation psychology, vol 3. CRC Press
72. Wakitani S et al (2019) Design and application of a database-driven PID controller with data-driven updating algorithm. Ind Eng Chem Res 58(26):11,419–11,429
73. Xie W, Cabecinhas D, Cunha R, Silvestre C (2021) Adaptive backstepping control of a quadcopter with uncertain vehicle mass, moment of inertia, and disturbances. IEEE Trans Industr Electron 69(1):549–559
74. Yang T, Li P, Zhang H, Li J, Li Z (2018) Monocular vision slam-based UAV autonomous landing in emergencies and unknown environments. Electronics 7(5):73

Using an Interactive Theorem Prover for Formally Analyzing the Dynamics of the Unmanned Aerial Vehicles

Adnan Rashid, Osman Hasan, and Sa'ed Abed

Abstract The dynamical analysis of Unmanned Aerial Vehicles (UAVs) is based on accessing their performance, stability, and various other control systems properties and it involves modelling their dynamical behavior that is generally captured by a set of differential equations. The state-of-the-art approaches used to study the dynamics of UAVs are analytical and computer-based testing or simulations. However, the inherent limitations of these methods, i.e., human error proneness, sampling-based analysis, approximations of the mathematical results and the presence of unverified algorithms in the core of the associated tools, make them unsuitable for analyzing the UAVs, which are extensively being advocated to be used in many safety-critical applications. Recently, interactive theorem proving, a formal verification technique, has been utilized for analyzing the dynamics of UAVs to overcome the above-stated limitations of the conventional approaches. This chapter briefly overviews these interactive theorem proving based efforts while highlighting their strengths and weaknesses.

Keywords Formal methods · Unmanned aerial vehicles · Dynamics · Interactive theorem proving

A. Rashid (✉) · O. Hasan
School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan
e-mail: adnan.rashid@seecs.edu.pk

O. Hasan
e-mail: osman.hasan@seecs.edu.pk

S. Abed
Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Kuwait City, Kuwait
e-mail: s.abed@ku.edu.kw

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_9

Acronyms

AEoM	Aircraft Equations of Motion
CAA	Civil Aviation Authority
DAIDALUS	Detect and Avoid Alerting Logic for Unmanned Systems
DE	Differential Equation
EoM	Equations of Motion
FD	Frequency-domain
GSN	Goal Structuring Notation
HA	Hybrid Automata
HL	HOL Light
HLMC	Hybrid Logic Model Checker
HLTP	HOL Light theorem prover
LDEoM	Lateral-directional Equations of Motion
LEoM	Longitudinal Equations of Motion
LHA	Linear Hybrid Automata
LT	Laplace Transform
MC	Model Checking
MIMO	Multiple-input Multiple-output
Ocaml	Objective CAML
RPAs	Remotely-Piloted Aircrafts
TALs	Tactical Automated Landing System
TF	Transfer Function
TP	Theorem Proving
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UAVs	Unmanned Aerial Vehicles
UV	Unmanned Vehicles

Notation

x_n	x-axis of the navigation coordinate frame
y_n	y-axis of the navigation coordinate frame
z_n	z-axis of the navigation coordinate frame
x_b	x-axis of the body coordinate frame
y_b	y-axis of the body coordinate frame
z_b	z-axis of the body coordinate frame
O_b	Center of mass of the aircraft
ψ	Yaw angle
ϕ	Roll angle
θ	Pitch angle
U_e	Axial Velocity

V_e	Lateral Velocity
W_e	Normal Velocity
I	Moment of intertia
X	Axial force
Z	Normal force
M	Pitching moment
η	Elevator angle
τ	Thurst

1 Introduction

Unmanned Aerial Vehicles (UAVs) [18] are the aircrafts that are operated, either; (i) by a human operator on ground that are also known as Remotely-Piloted Aircrafts (RPAs) or; (ii) using a built-in computer system, such as autopilot assistance, providing a certain degree of autonomy or; (iii) as a fully autonomous aircraft without any involvement of the human intervention. In all these scenarios, UAVs do not require any pilot aboard. Moreover, they are frequently utilized for completing various tasks, like navigation, guidance and the flight control operations in many domains, ranging from civilian [6] to military [28] applications, such as transportation [6], remote sensing [16], surveillance [38] and rescue missions [7] etc. Moreover, due to their extensive utility in the above-stated civilian and military applications, UAVs are making a remarkable addition to the economic development and safety enhancement.

The continuous dynamics of an Unmanned Aerial Vehicle (UAV) incorporating the effect of different forces on the attitude and speed of the Unmanned Vehicle (UV) with reference to time are analyzed to study stability, performance and several other control characteristics of the UAV. This analysis is based on capturing the dynamical behavior of a UAV in term of general Aircraft Equations of Motion (AEoM) containing the effect of the navigation, force, moment and kinematics equations [41]. Moreover, these AEoM, represented as a system of Differential Equations (DEs), requires modeling a coordinate system describing the motion (relative position and movement) of the aircraft in various frames, such as navigation and aircraft's body-fixed frames [15]. Next, these aircraft equations are solved using the Laplace Transform (LT) to get the corresponding Transfer Function (TF), providing a Frequency-domain (FD) representation, or their solutions in time-domain. These results can be further utilized for computing various properties, such as stability, sensitivity and control of UAVs.

The conventional approaches used to study the dynamics of these UAVs are the analytical [13] and computer based simulation techniques [37]. However, the former is subject to human error, specifically, of the larger system. Similarly, the later is based on several procedures existing in the core od the tools that are not verified and are utilized for the dynamical analysis. Therefore, the computer-based simulation methods may compromise the accuracy of the analysis. For example, according to the 2004 report by US department of transportation, 25% of the accidents happened to the US army aircraft, *Shadow* 200 *(RQ-*7*) Unmanned*, due to the failure of the Tactical Automated Landing System (TALS) [40]. Thus, a rigorous analysis of these UAVs is important, considering the safety-critical nature of UAVs, where these conventional techniques should not be completely relied upon as these may lead to disastrous consequences.

Formal methods [25] are computer-based mathematical methods that can provide a rigorous analysis and have been utilized to cater for the limitations of the above-stated conventional techniques to study the dynamical behaviour of UAVs. Model checking (MC) [11] and Theorem Proving (TP) [23] are two major types of formal methods. MC [11] is based on constructing a mathematical model, using a state diagram, of the given system and the formal verification of its desired behaviour characterized by a temporal logic formula. It has been widely utilized (e.g., [20, 21, 26]) for verifying the dynamics of UAVs. However, it includes discretization of the continuous variables in the model involving differentials and integrals. Moreover, it suffers from the state-space explosion problem [5]. Due to these limitations, the accuracy and completeness of the analysis is somewhat compromised. Higher-order-logic TP [23] includes the construction of a mathematical model of the underlying system in higher-order logic and verification of its desired behaviour using mathematical reasoning within the sound core of a theorem prover. The involvement of the expressive higher-order logic in the construction of a formal model and verification of its corresponding properties of interest within the sound core of a theorem prover guarantees the accuracy and completeness of the associated analysis. Recently, Abed et al. [2] used this method to formally study the continuous dynamics, captured using a set of linear DEs, of UAVs as shown in Fig. 1. In particular, the authors formalized several coordinate frames, such as navigation and aircraft's body-fixed frames based on the library of the complex-valued matrices available in the HOL Light theorem prover (HLTP). Moreover, they formalized the AEoM capturing the continuous dynamics of UAVs and formally verified the TF and the FD solutions of the Equations of Motion (EoM) of UAVs using HOL Light (HL). This chapter presents an overview of these efforts, which includes

- the formalization of the coordinate systems
- the formalization of the AEoM and verification of their solutions
- the formal verification of the stability of CropCam UAV using HL
- the highlights of the strengths and weaknesses of the TP approach to study the continuous dynamics of UAVs

The rest of the chapter is organized as follows: The related work on the formal analysis of UAVs is captured in Sect. 2. Section 3 provides a brief overview of TP, the HLTP, multivariate calculus and the LT theories of HL. The formal verification of various coordinate frames and their transformation, capturing the relative movement and position, of UAVs is presented in Sect. 4. Section 5 provides the formalization of the AEoM, such as the longitudinal and the Lateral-directional Equations of Motion (LDEoM), captured as a set of linear DEs and the formal verification of their frequency and the time-domain solutions using HL. Section 6 presents the formal

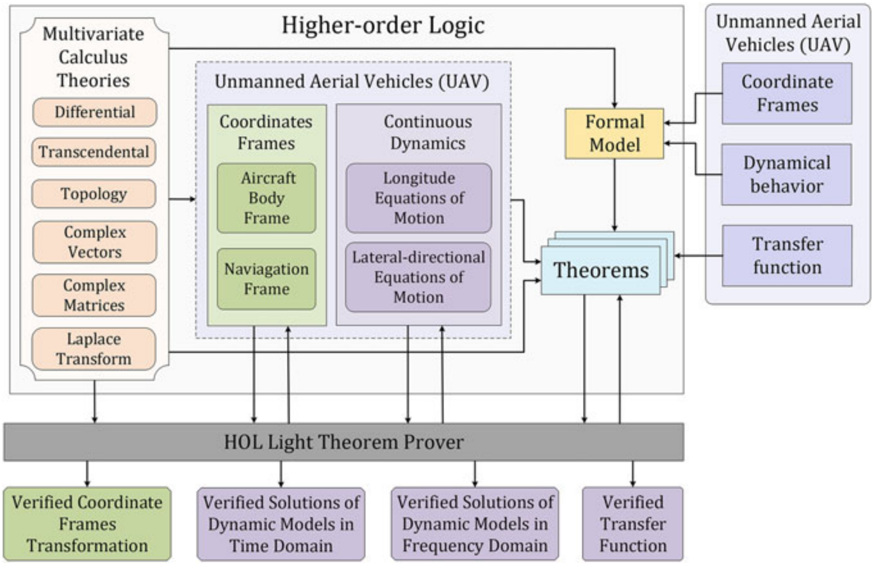

Higher-order Logic
Multivariate Calculus Theories
Differential
Transcendental
Topology
Complex Vectors
Complex Matrices
Laplace Transform
Unmanned Aerial Vehicles (UAV)
Coordinates Frames
Aircraft Body Frame
Naviagation Frame
Continuous Dynamics
Longitude Equations of Motion
Lateral-directional Equations of Motion
Formal Model
Theorems
Unmanned Aerial Vehicles (UAV)
Coordinate Frames
Dynamical behavior
Transfer function
HOL Light Theorem Prover
Verified Coordinate Frames Transformation
Verified Solutions of Dynamic Models in Time Domain
Verified Solutions of Dynamic Models in Frequency Domain
Verified Transfer Function

Fig. 1 Proposed framework

stability analysis of CropCam UAV using HL. Some discussion about the higher-order-logic TP based analysis, while highlighting their strengths and weaknesses, is provided in Sect. 7. Finally, Sect. 8 concludes the chapter by highlighting some future directions.

2 Related Work

Formal methods, such as MC and higher-order-logic TP have been used for the formal analysis of UAVs. Guzey [21] used Hybrid Automaton (HA) to model the consensus-based formation control of a team of the fixed-wing UAVs flying at a fixed altitude that assists in maintaining their predefined formation and keep them on track to their destination. Similarly, Seibel et al. [36] proposed a Linear Hybrid Automata (LHA) based framework for flight and mission planning of the rotary-wing UAVs, which incorporates the flight plan of the aircraft, the region of the flight and the meteorological conditions of the flight. Moreover, the authors formally verified the safety and timeliness properties of the LHA based model that helps in increasing the safe flight operation by avoiding any undesired behaviour. Groza et al. [20] formally analyzed UAVs using Hybrid Logic Model Checker (HLMC). The authors constructed a formal Goal Structuring Notation (GSN) model of the aircraft and utilized the description logic for identification of the assurance deficits in the model. Finally, the identified flaws are verified as properties specification in HLMC. Similarly, Karimoddini et al. [26] provided a hybrid approach for modeling and control design of an unmanned helicopter. The authors designed the control structure using multiple layers, namely, motion planning, supervision and the regulation layers, placed in hierarchical form with each layer using a HA. Finally, these layers are synchronized using a compositional operator that is developed as part of the proposed framework.

AJPF, SPIN and PRISM model checkers have also been used for formally analyzing UAVs. Dennis et al. [14] used the autonomous agent language, Gwendolen, for developing a formal model of the Unmanned Aircraft System (UAS) control system and formally verified the model using the agent model checker AJPF. Similarly, Webster et al. [39] used the SPIN model checker for the formal verification of an autonomous UAS. The authors constructed a fundamental control model of the UAS in PROMELA and used the SPIN model checker to formally verify this model against a few subset of rules defined by Civil Aviation Authority (CAA). The authors also captured the probabilistic aspects of UAS by developing a probabilistic model and verified the same set of rules using PRISM. All the MC based analyses, presented above, consider the discrete time models of the aircraft in the form of automata and are unable to capture their continuous dynamics in true form. Moreover, MC suffers from its inherent state-space explosion problem and thus is not well suited for analyzing systems exhibiting the continuous dynamics, which generally leads to large models.

Higher-order-logic theorem provers, such as PVS and Coq theorem provers have been used for formally verifying UAVs. Munoz et al. [32] proposed a Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS) and used it for developing the self-separation and alerting algorithms that provide an awareness to UAS remote pilots about any situations. Moreover, the authors formally analyzed these algorithms using PVS. Similarly, Munoz and Narkawicz [31] used PVS for formally verifying the extended well-clear boundaries of a UAV that describe the ability of an aircraft to keep itself away from the other aircraft in the airborne traffic to avoid collisions. Similarly, Ghorbal et al. [19] proposed a hybrid TP approach for formally verifying the property of separation between two or more aircraft in the aerospace systems. Narkawicz and Munoz [33] formally verified a conflict detection algorithms for aircraft flying on arbitrary nonlinear trajectories using PVS. Similarly, Carreno and Muñoz [8] formally verified the correctness of an alerting algorithm for aircraft using the PVS theorem prover. Chen and Chen [10] formally verified a control algorithm for automatic landing of a Helicopter using Coq. Similarly, Ma and Chen [29] formally verified the coordinate transformation matrices of the aircraft control system using Coq. Malecha el al. [30] proposed a library, VeriDrone, to formally verify various aspects of CPS. VeriDrone is developed in Coq comprising of theories ranging from floating point numbers to DEs. Similarly, Chan et al. [9] used VeriDrone for formally verifying the stability properties, i.e., Lyapunov and exponential stability, of CPSs.

KeYmaera theorem prover and its various variants have also been used for formally analyzing UAVs. For example, Loos et al. [27] used KeYmaeraD for formally analyzing the control policies for planar aircraft avoidance maneuvers by constructing a formal proof of their safety properties. Arechiga et al. [4] formally verified the closed-loop properties of the control system using KeYmaera. The authors also formally verified the safety of a cooperative intersection collision avoidance system and an intelligent cruise controller. Recently, Abed et al. [2] proposed a higher-order-logic TP based approach for formally analyzing the continuous dynamical behaviour of UAVs. The authors formalized various coordinate frames that captures the relative position and motion of the aircraft, and formally verified their interrelationship using the HLTP. Moreover, they formalized the aircraft's EoM capturing the dynamical behaviour of UAVs and formally verified the TF, and the FD solutions of the EoM of UAVs. The main focus of this chapter is to present the efforts that have been done in this TP based dynamical analysis of UAVs.

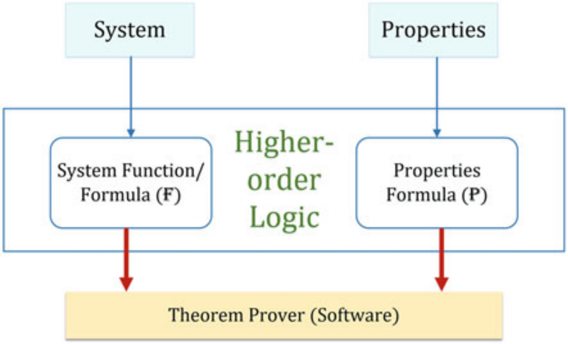
3 Higher-Order-Logic TP and HL

This section includes a brief introduction to the higher-order-logic TP and the HLTP.

3.1 Higher-Order-Logic TP

Higher-order-logic TP [23] is based on developing a mathematical model of the given system and expressing its desired properties using higher-order logic as shown in Fig. 2. In the next step, the relationship between the higher-order logic model and its desired properties is verified as theorems in a theorem prover, which is computer program using deductive reasoning to develop formal proofs using a small set of well-known axioms, inference rules, hypothesis and the already verified theorems. Due to the highly expressiveness of higher-order logic, it enables formal modeling and verification of the continuous dynamics of a system using a higher-order-logic theorem prover. Also, the higher-order logic is undecidable and thus verification of

Fig. 2 Higher-order-logic TP



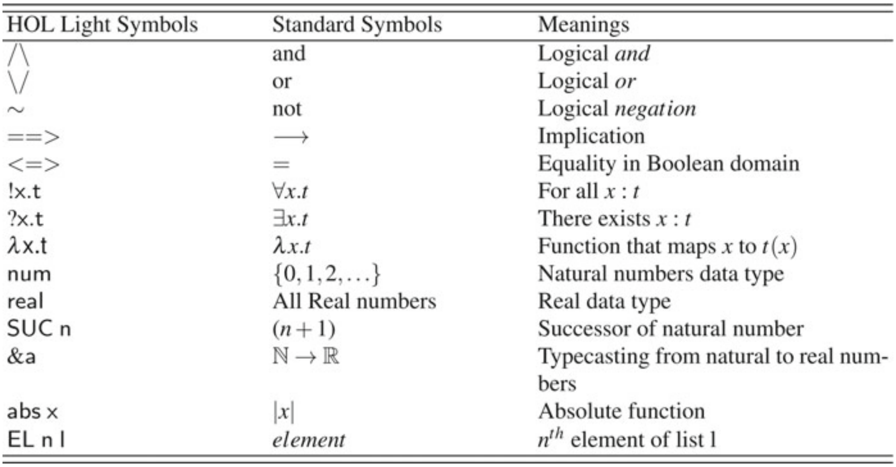
sentences expressed in this logic needs explicit guidance from its user in an interactive manner. Moreover, the involvement of the deductive reasoning in the proof of formal proof development ensure soundness, i.e., every sentence proved in the TP system is actually true.

3.2 HLTP

HL [22] is a higher-order-logic TP tool that guarantees secure TP using a strongly-typed functional programming language Objective CAML (OCaml) [34]. HL users can develop higher-order-logic based model of a system and they can interactively verify theorems, capturing the desired properties, by applying the available proof tactics and procedures. A theory in HL comprises of types, constants, definitions and theorems. HL contains a rich set of formalized libraries, such as real arithmetic, multivariate calculus and the LT, which are extensively used in our formalization. Indeed, one of the main motivations for selecting the HLTP for the proposed framework is the availability of the multivariable calculus and the LT theories in HL. There are many automatic proof procedures [24] available in HL, which assist the user in conducting a proof more efficiently.

Table 1 provides some symbols, i.e., their HL and standard representations, and their meanings, which are commonly used in presenting the formal analysis of the dynamical behavior of UAVs in this chapter.

Table 1 HOL light symbols



HOL Light Symbols	Standard Symbols	Meanings
/\	and	Logical *and*
\/	or	Logical *or*
~	not	Logical *negation*
==>	$\longrightarrow$	Implication
<=>	$=$	Equality in Boolean domain
!x.t	$\forall x.t$	For all $x : t$
?x.t	$\exists x.t$	There exists $x : t$
λx.t	$\lambda x.t$	Function that maps x to $t(x)$
num	$\{0,1,2,\ldots\}$	Natural numbers data type
real	All Real numbers	Real data type
SUC n	$(n+1)$	Successor of natural number
&a	$\mathbb{N} \rightarrow \mathbb{R}$	Typecasting from natural to real numbers
abs x	$\lvert x \rvert$	Absolute function
EL n l	*element*	n^{th} element of list l

4 Formal Modeling of the Coordinate Systems

Coordinate systems are useful for modeling the position and orientation UAVs at any time instant. The navigation and the aircraft's body-fixed frames are the two commonly used coordinate frames for capturing the dynamics of the aircraft. The motion of a UAV is generally represented with respect to the navigation coordinate frame (x_n, y_n, z_n), which is oriented as North, East, Down and is attached to earth's local tangent plane as depicted in Fig. 3. Moreover, a right-handed orthogonal body coordinate frame (x_b, y_b z_b) is also attached to the aircraft as depicted in Fig. 3 and a transformation between these navigation and the aircraft's body-fixed frame is required to efficiently capture the relative motion and the position of the aircraft. The center of mass of the aircraft is located at O_b of the aircraft body-fixed frame. The positive x-axis of the frame is directed forward along the longitudinal axis of the aircraft. Similarly, the positive y-axis is oriented along the right wing of UAV. The positive z-axis is perpendicular to the x and y axes and is pointing downwards from the aircraft as shown in Fig. 3.

A point capturing the position and orientation of a UAV in a coordinate system needs to be formalized in order to model the navigation and the aircraft's body-

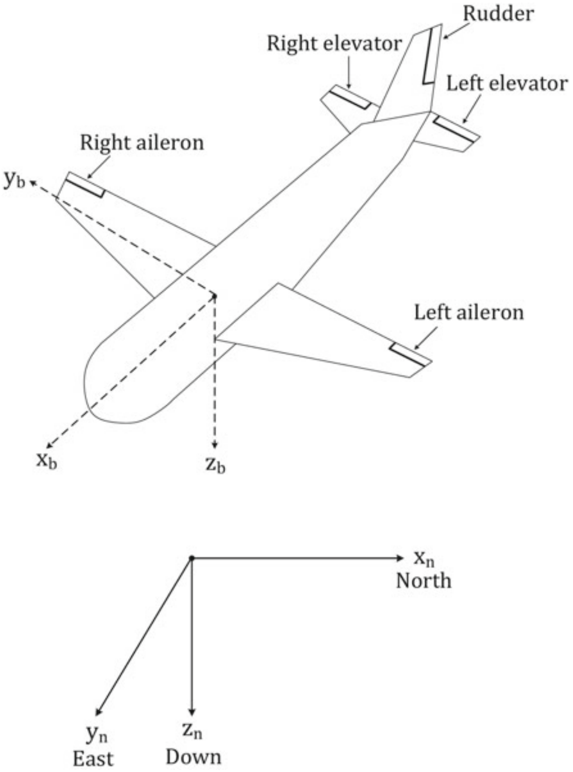


Fig. 3 Aircraft configuration

fixed frames. In this regard, the available types, such as Real ($\mathbb{R}$), Complex ($\mathbb{C}$), One-dimensional real-valued vector ($\mathbb{R}^1$)) can be utilized to abbreviate new types. Therefore, the new types for various points are defined using the feature of type abbreviation in HL as follows:

Definition 1 *Points of a Coordinate System*

nw_typ_abrv ("1d_pnt",: (‘$\mathbb{R}^1 \rightarrow \mathbb{C}$)’)

nw_typ_abrv ("tmd_1d_pnt",‘: (1d_pnt $\times$ $\mathbb{R}^1$)’)

nw_typ_abrv ("2d_pnt",‘: (1d_pnt $\times$1d_pnt)’)

nw_typ_abrv ("tmd_2d_pnt",‘: (2d_pnt $\times$ $\mathbb{R}^1$)’)

nw_typ_abrv ("3d_pnt",‘: (1d_pnt $\times$2d_pnt)’)

nw_typ_abrv ("tmd_3d_pnt",‘: (3d_pnt $\times$ $\mathbb{R}^1$)’)

The type tmd_1d_pnt is a pair, modeling the orientation and position of a UAV in one-dimensional coordinate system that changes with time, where its second element captures the time. Moreover, the notion of the complex-valued matrices and their various classical properties for the formalization of the coordinate frames are required, which are also formalized as a part of our proposed approach. Some of the verified properties of the complex-valued matrices are given in Table 2. Here, the HL operator %%% models the multiplication of a complex-valued vector with a complex-valued matrix and vice versa. Similarly, cmat 0 captures the complex-valued zero matrix (a matrix having all its entries as zero) of order $M \times N$. The function cmat 1 represents complex-valued matrix having all its entries as one. Interested users can find more details about the formalization of the complex-valued matrices and formal proofs of their classical properties at [1].

The navigation and the aircraft's body-fixed coordinate frames are three-dimensional coordinates and are formally defined in HL as follows:

Definition 2 *Three-dimensional Coordinates*

$$\vdash_{def} \forall x\ y\ z\ t.\ \textbf{3d\_cord\_systm}\ (((x,y,z),t){:}\mathrm{tmd\_3d\_pnt}) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}$$

The function 3d_cord_systm accepts a variable of data-type *tmd_3d_pnt* and returns a three-dimensional vector capturing the corresponding coordinate frame.

Next, to transform an orientation of the navigation frame to the aircraft's body-fixed frame, a continuous rotation of the navigation frame is done by the Euler angles, which are defined by performing a rotation about the axes of three-dimensional right-handed coordinate system as shown in Fig. 4. These Euler angles are formalized using the type abbreviation feature in HL, which are further used for verification of the transformation of the navigation frame to aircraft's body-fixed frame.

Definition 3 *Euler Angles*

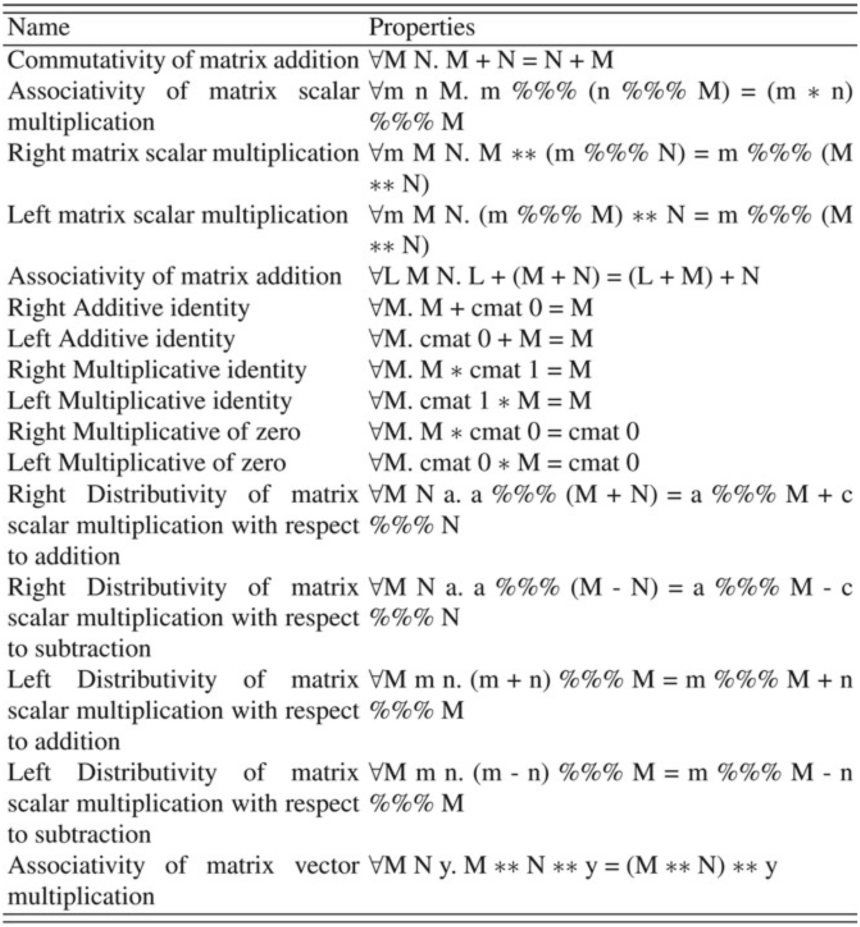
nw_typ_abrv ("θ", ‘:$\mathbb{C}$’)

nw_typ_abrv ("ϕ", ‘:$\mathbb{C}$’)

nw_typ_abrv ("ψ", ‘:$\mathbb{C}$’)

nw_typ_abrv ("eulr_angls", ‘:($\theta \times \phi \times \psi$)’)

Table 2 Classical properties of the complex-valued matrices

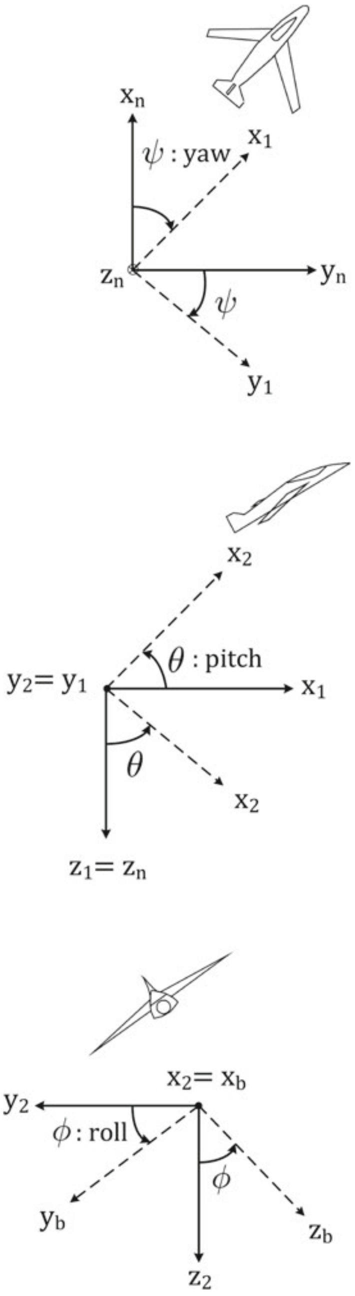


Name	Properties
Commutativity of matrix addition	∀M N. M + N = N + M
Associativity of matrix scalar multiplication	∀m n M. m %%% (n %%% M) = (m ∗ n) %%% M
Right matrix scalar multiplication	∀m M N. M ∗∗ (m %%% N) = m %%% (M ∗∗ N)
Left matrix scalar multiplication	∀m M N. (m %%% M) ∗∗ N = m %%% (M ∗∗ N)
Associativity of matrix addition	∀L M N. L + (M + N) = (L + M) + N
Right Additive identity	∀M. M + cmat 0 = M
Left Additive identity	∀M. cmat 0 + M = M
Right Multiplicative identity	∀M. M ∗ cmat 1 = M
Left Multiplicative identity	∀M. cmat 1 ∗ M = M
Right Multiplicative of zero	∀M. M ∗ cmat 0 = cmat 0
Left Multiplicative of zero	∀M. cmat 0 ∗ M = cmat 0
Right Distributivity of matrix scalar multiplication with respect to addition	∀M N a. a %%% (M + N) = a %%% M + c %%% N
Right Distributivity of matrix scalar multiplication with respect to subtraction	∀M N a. a %%% (M - N) = a %%% M - c %%% N
Left Distributivity of matrix scalar multiplication with respect to addition	∀M m n. (m + n) %%% M = m %%% M + n %%% M
Left Distributivity of matrix scalar multiplication with respect to subtraction	∀M m n. (m - n) %%% M = m %%% M - n %%% M
Associativity of matrix vector multiplication	∀M N y. M ∗∗ N ∗∗ y = (M ∗∗ N) ∗∗ y

The complex data-type ($\mathbb{C}$) in HL is selected to formally define the Euler angles, as this choice enables us to formally verify the stability of UAV, which is based on the placement of the poles in the complex plane, as will be described in Sect. 6 of the chapter. Figure 4 provides the Euler angles and the corresponding frame transformations. A rotation of yaw angle (ψ) about the z_n axes converts the navigation frame to *Intrmdt Frame* 1 defining the aircraft's heading. Next, we rotate the *Intrmdt Frame* 1 by pitch angle (θ) about the new y_1 axis to *Intrmdt Frame* 2. Finally, *Intrmdt Frame* 2 is transformed to the aircraft's body-fixed frame by a rotation of roll angle (ϕ) about the new x_2 axis.

Next, the notion of the rotation matrices for rolling, pitching and yawing is formalized. These matrices provide the orientations of the aircraft with respect to various

Fig. 4 Euler angles and frame transformation



coordinate frames and are used in the verification of the corresponding transformations from navigation to *Intrmdt Frame* 1, *Intrmdt Frame* 1 to *Intrmdt Frame* 2 and *Intrmdt Frame* 2 to the aircraft's body-fixed frame.

Definition 4 *Rotation Matrices*

$\vdash_{def} \forall \theta\ \psi\ \phi.$ **rotat_matrx_rolng** $(\theta,\psi,\phi) =$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\ \phi & \mathrm{csin}\ \phi \\ 0 & -\ \sin\ \phi & \cos\ \phi \end{bmatrix}$$

$\vdash_{def} \forall \phi\ \psi\ \theta.$ **rotat_matrx_ptchng** $(\theta,\psi,\phi) =$

$$\begin{bmatrix} \cos\ \theta & 0 & -\ \sin\ \theta \\ 0 & 1 & 0 \\ \sin\ \theta & 0 & \cos\ \theta \end{bmatrix}$$

$\vdash_{def} \forall \theta\ \phi\ \psi.$ **rotat_matrx_yawng** $(\theta,\psi,\phi) =$

$$\begin{bmatrix} \cos\ \psi & \sin\ \psi & 0 \\ -\ \sin\ \psi & \cos\ \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now, the transformation from the navigation frame to *Intrmdt Frame* 1 is verified in HL as follows:

Theorem 1 Navigation Frame to *Intrmdt Frame 1*

$\vdash_{thm} \forall\ x_n\ y_n\ z_n\ x_1\ y_1\ z_1\ \theta\ \phi\ \psi$ t. trnsfrm_nvgat_frame_to_intrmd_frame1 (x_1,y_1,z_1) $(x_n,y_n,z_n)\ (\theta,\phi,\psi)\ t \Leftrightarrow$

$$\left(\begin{bmatrix} x_1(t) \\ y_1(t) \\ z_1(t) \end{bmatrix} = \begin{bmatrix} x_n(t) * \cos\psi + y_n(t) * \sin\psi \\ -x_n(t) * \sin\psi + y_n(t) * \cos\psi \\ z_n(t) \end{bmatrix}\right)$$

The function trnsfrm_nvgat_frame_to_intrmd_frame1 captures the corresponding transformation. The proof process of Theorem 1 is mainly based on the properties of complex vectors and matrices, and transcendental functions along with some arithmetic reasoning. In a similar manner, the transformations from *Intrmdt Frame* 1 to aircraft's body-fixed frame are verified in HL as given in Table 3.

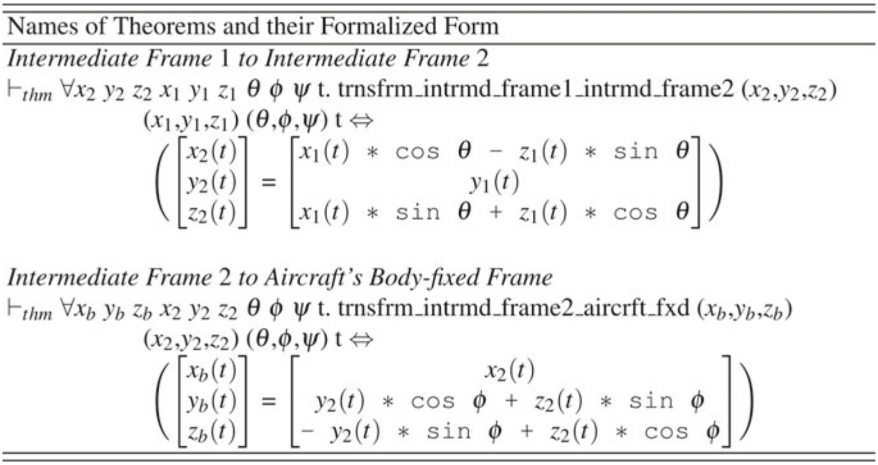
Next, the direction cosine matrix is formalized, which is necessary for the verification of the transformation of the navigation to aircraft's body-fixed frame.

Definition 5 *Direction Cosine Matrix*

$\vdash_{def} \forall\psi\ \phi\ \theta.$ **dirctn_cosne_mtrx** $(\theta,\phi,\psi) = \begin{bmatrix} \mathsf{R} & \mathsf{S} & \mathsf{T} \\ \mathsf{U} & \mathsf{V} & \mathsf{W} \\ \mathsf{X} & \mathsf{Y} & \mathsf{Z} \end{bmatrix}$

where,

Table 3 Verification of transformation matrices



Names of Theorems and their Formalized Form
Intermediate Frame 1 to Intermediate Frame 2
$\vdash_{thm} \forall x_2\ y_2\ z_2\ x_1\ y_1\ z_1\ \theta\ \phi\ \psi$ t. trnsfrm_intrmd_frame1_intrmd_frame2 (x_2,y_2,z_2) (x_1,y_1,z_1) (θ,ϕ,ψ) t $\Leftrightarrow$ $\left(\begin{bmatrix} x_2(t) \\ y_2(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t) * \cos\ \theta - z_1(t) * \sin\ \theta \\ y_1(t) \\ x_1(t) * \sin\ \theta + z_1(t) * \cos\ \theta \end{bmatrix}\right)$
Intermediate Frame 2 to Aircraft's Body-fixed Frame
$\vdash_{thm} \forall x_b\ y_b\ z_b\ x_2\ y_2\ z_2\ \theta\ \phi\ \psi$ t. trnsfrm_intrmd_frame2_aircrft_fxd (x_b,y_b,z_b) (x_2,y_2,z_2) (θ,ϕ,ψ) t $\Leftrightarrow$ $\left(\begin{bmatrix} x_b(t) \\ y_b(t) \\ z_b(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ y_2(t) * \cos\ \phi + z_2(t) * \sin\ \phi \\ - y_2(t) * \sin\ \phi + z_2(t) * \cos\ \phi \end{bmatrix}\right)$

$$
\begin{aligned}
&\mathsf{R} = \cos\ \theta * \cos\ \psi \\
&\mathsf{S} = \cos\ \theta * \cos\ \psi \\
&\mathsf{T} = -\ \sin\ \theta \\
&\mathsf{U} = \sin\ \phi * \sin\ \theta * \cos\ \psi\ -\ \cos\ \phi * \sin\ \psi \\
&\mathsf{V} = \sin\ \phi * \sin\ \theta * \sin\ \psi + \cos\ \phi * \cos\ \psi \\
&\mathsf{W} = \sin\ \phi * \cos\ \theta \\
&\mathsf{X} = \cos\ \phi * \sin\ \theta * \cos\ \psi + \sin\ \phi * \sin\ \psi \\
&\mathsf{Y} = \cos\ \phi * \sin\ \theta * \sin\ \psi\ -\ \sin\ \phi * \cos\ \psi \\
&\mathsf{Z} = \cos\ \phi * \cos\ \theta
\end{aligned}
$$

Now, the direction cosine matrix, formalized as Definition 5), is utilized to formally model the transformation from the navigation frame to aircraft's body-fixed frame in HL as:

Definition 6 *Navigation Frame to Aircraft's Body-fixed Frame*

$\vdash \forall\ x_b\ y_b\ z_b\ x_n\ y_n\ z_n\ \theta\ \phi\ \psi$ t.
    trnsfrm_aircrft_fxd_nvgat_frame (x_b, y_b, z_b) (x_n, y_n, z_n) (θ, ϕ, ψ) t $\Leftrightarrow$
        (3d_cord_systm $((x_b, y_b, z_b)$, t) =
            dirctn_cosne_mtrx (θ, ϕ, ψ) $**$ 3d_cord_systm $((x_n, y_n, z_n)$, t))

Next, the transformation from the navigation to aircraft's body-fixed frame (Definition 6) is formally verified in HL as follows:

Theorem 2 Navigation Frame to Aircraft's Body-fixed Frame

$\vdash \forall\ x_b\ y_b\ z_b\ x_2\ y_2\ z_2\ x_1\ y_1\ z_1\ x_n\ y_n\ z_n\ \theta\ \phi\ \psi$ t.
[C1] trnsfrm_nvgat_frame_to_intrmd_frame1 (x_1,y_1,z_1) (x_n,y_n,z_n) (θ,ϕ,ψ) t $\wedge$
[C2] trnsfrm_intrmd_frame1_intrmd_frame2 (x_2,y_2,z_2) (x_1,y_1,z_1) (θ,ϕ,ψ) t $\wedge$

[C_3] trnsfrm_intrmd_frame2_aircrft_fxd (x_b,y_b,z_b) (x_2,y_2,z_2) (θ,ϕ,ψ) t
$\Rightarrow$ trnsfrm_aircrft_fxd_nvgat_frame (x_b,y_b,z_b) (x_n,y_n,z_n) (θ,ϕ,ψ) t

Conditions C_1, C_2 and C_3 provide the transformations from the navigation frame to *Intrmdt Frame* 1, *Intrmdt Frame* 1 to *Intrmdt Frame* 2 and *Intrmdt Frame* 2 to aircraft's body-fixed frame, respectively. The conclusion captures the transformation of the navigation frame to aircraft's body-fixed frame. The verification of Theorem 2 is mainly based on Theorem 1, theorems given in Table 3 and the properties of complex matrices and vectors alongside some arithmetic reasoning. This concludes the formalization of the coordinate frames and verification of their transformation. Further details about the formalization can be found at [1].

5 Formal Verification of the Dynamics of UAVs

This section provides the formal verification of the continuous dynamical behaviour of UAVs, which includes the formalization of the longitudinal and LDEoM (set of the DEs) capturing the dynamics of the aircraft, and the verification of their FD solutions. These equations are developed by the application of the physical and mathematical laws regarding force and inertia, providing a relationship between the input and output variables of the aircraft as depicted in Fig. 5. Since multiple input and output variables are involved for capturing the dynamics of the underlying system, therefore, it is declared as a MIMO system. To formally verify these equations in the HLTP, first, various parameters involved in the development of these equations, such as velocities, force and the moment of inertia etc needs to be incorporated.

First, the linear and angular disturbance velocities are modelled using the feature of type abbreviation in HL as follows:

Definition 7 *Linear Disturbance Velocities*

nw_typ_abrv ("u",‘ : $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("v",‘ : $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("w",‘ : $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("linr_dstrbnc_veloc",‘:$(\theta \times \phi \times \psi$)‘)

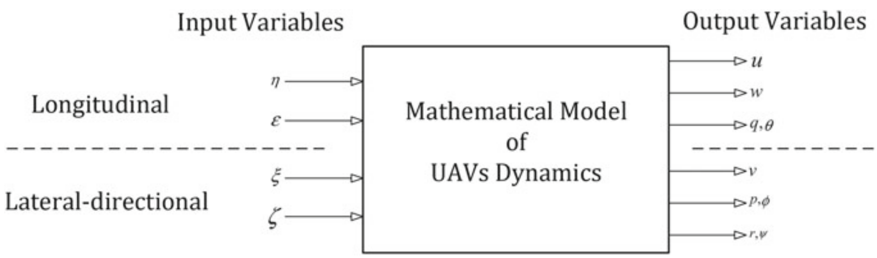


Fig. 5 UAV input–output relationship

Table 4 Aerodynamics stability derivatives and their data-types

Symbol	Type	Symbol	Type	Symbol	Type
X_u	$\mathbb{C}$	Z_w	$\mathbb{C}$	M_u	$\mathbb{C}$
X_w	$\mathbb{C}$	Z_{wd}	$\mathbb{C}$	M_w	$\mathbb{C}$
X_{wd}	$\mathbb{C}$	Z_q	$\mathbb{C}$	M_{wd}	$\mathbb{C}$
X_q	$\mathbb{C}$	Z_u	$\mathbb{C}$	M_q	$\mathbb{C}$
X_v	$\mathbb{C}$	Z_τ	$\mathbb{C}$	M_η	$\mathbb{C}$
X_τ	$\mathbb{C}$	Z_η	$\mathbb{C}$	M_τ	$\mathbb{C}$
X_η	$\mathbb{C}$				

Definition 8 *Angular Disturbance Velocities*

nw_typ_abrv ("p",‘: $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("q",‘: $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("r",‘: $\mathbb{R}^1 \rightarrow \mathbb{C}$‘)
nw_typ_abrv ("anglr_dstrbnc_veloc",‘:$(\theta \times \phi \times \psi\ )$)‘)

where each component of the linear and angular disturbance velocities is a function of type $\mathbb{R}^1 \rightarrow \mathbb{C}$. Similarly, the aerodynamics stability derivatives represent constant values of type $\mathbb{C}$ for the linear LEOM of UAVs, as presented in Table 4. These derivatives for the axial force, normal force and the pitching moment are modeled using type abbreviations in HL as follows:

Definition 9 *Aerodynamics Stability Derivatives*

nw_typ_abrv ("arodyn_stbl_dervt_axl_frce",‘:
$(X_u \times X_v \times X_w \times X_{wd} \times X_q \times X_\eta \times X_\tau\ )$‘)
nw_typ_abrv ("arodyn_stbl_dervt_nrml_frce", ‘:
$(Z_u \times Z_w \times Z_{wd} \times Z_q \times Z_\eta \times Z_\tau\ )$‘)
nw_typ_abrv ("arodyn_stbl_dervt_pitchng_momnt", ‘:
$(M_u \times M_w \times M_{wd} \times M_q \times M_\eta \times M_\tau\ )$‘)

Similarly, the linear velocity and moment of inertia of UAVs are modelled using the type abbreviation feature of HL as follows:

Definition 10 *Linear Velocity*

nw_typ_abrv ("U_e", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("V_e", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("W_e", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("lnr_vlcty", ‘:$(U_e \times V_e \times W_e)$‘)

Definition 11 *Moment of Inertia*

nw_typ_abrv ("I_x", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("I_y", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("I_z", ‘:$\mathbb{C}$‘)
nw_typ_abrv ("momnt_of_inrtia", ‘:$(I_x \times I_y \times I_z)$‘)

where U_e, V_e and W_e capture the axial, lateral and normal velocities, respectively. Similarly, I_x, I_y and I_z model the x, y and z components of the moment of inertia, respectively.

The motion of a UAV is generally described by decoupled equations of motion [12], such as longitudinal and LDEoM. The LEoM for the aircraft are described by the axial force (X), the normal force (Z) and the pitching moment (M) and are mathematical described as [12, 41]:

$$\begin{aligned} m\dot{u} - X_u u - X_{\dot{w}}\dot{w} - X_w w - (X_q - mW_e)q + mg\theta cos\theta_e &= X_\eta \eta + X_\tau \tau \\ -Z_u u + (m - Z_{\dot{w}})\dot{w} - Z_w w - (Z_q + mU_e)q + mg\theta sin\theta_e &= Z_\eta \eta + Z_\tau \tau \\ M_u u - M_{\dot{w}}\dot{w} - M_w w + I_y\dot{q} - M_q q &= M_\eta \eta + M_\tau \tau \end{aligned} \tag{1}$$

where $\dot{u}$, $\dot{w}$ and $\dot{q}$ represent the first order-derivatives of the linear and angular disturbance velocities, respectively. Similarly, m, θ_e and g are the total mass of the UAV, the steady pitch attitude of the UAV and the acceleration due to gravity, respectively. The variables η and τ model the elevator angle and thrust, respectively. Under the conditions $q(t) = \dot{\theta}(t)$ and $\tau(t) = 0$, the above LEoM of the UAV become [12]:

$$\begin{aligned} m\dot{u} - X_u u - X_{\dot{w}}\dot{w} - X_w w - (X_q - mW_e)\dot{\theta} + mg\theta cos\theta_e &= X_\eta \eta \\ -Z_u u + (m - Z_{\dot{w}})\dot{w} - Z_w w - (Z_q + mU_e)\dot{\theta} + mg\theta sin\theta_e &= Z_\eta \eta \\ M_u u - M_{\dot{w}}\dot{w} - M_w w + I_y\ddot{\theta} - M_q\dot{\theta} &= M_\eta \eta \end{aligned} \tag{2}$$

In order to model the above set of linear DEs, first, a linear Differential Equation (DE) of order n is formalized as follows:

Definition 12 *DE of Order n*

$\vdash_{def}$ ∀n lst f t. **n_ord_diffren_equat** n L f t = $\sum_0^n$ (λk. EL k L ∗ $\frac{\mathsf{d^k f(t)}}{\mathsf{dt}}$)

The function n_ord_diffren_equat accepts the order of the DE n, a list of constant coefficients L, a differentiable function f and the differentiation variable t and return a DE of order n. It uses the function EL k L, which returns the kth element of a list L, to generate the DE corresponding to the given parameters.

The first DE of the LEoM (Eq. (2)) is formalized in HL as follows:

Definition 13 *LEoM* 1

$\vdash_{def}$ ∀ X_v X_w X_{wd} X_q X_η X_τ X_u m.

list_u_longtnl_eqtn_motn_fst m (X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ) = [- X_u; m]

$\vdash_{def}$ ∀ X_u X_v X_q X_η X_τ X_w X_{wd}.

list_w_longtnl_eqtn_motn_fst (X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ) = [X_w; X_{wd}]

$\vdash_{def}$ ∀ X_u X_v X_w X_{wd} X_η X_τ U_e V_e g θ_e X_q m W_e.

list_thta_longtnl_eqtn_motn_fst m g θ_e

(X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ) (U_e,V_e,W_e) = [-(m ∗ g ∗ cos θ_e); X_q - m ∗ W_e]

$\vdash_{def}$ ∀ X_u X_v X_w X_{wd} X_q X_τ X_η.

list_eta_longtnl_eqtn_motn_fst $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$ = $[X_\eta]$
$\vdash_{def}$ ∀p q r v u w m g θ_e U_e V_e W_e θ X_u X_v X_w X_{wd} X_q X_η X_τ η t.
longtnl_eqtn_motn_fst $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$
(U_e,V_e,W_e) m g θ_e (p,q,r) (u,v,w) θ η t ⇔
n_ord_diffren_equat 1
(list_u_longtnl_eqtn_motn_fst m $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$) u t -
n_ord_diffren_equat 1
(list_w_longtnl_eqtn_motn_fst $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$) w t -
n_ord_diffren_equat 1
(list_thta_longtnl_eqtn_motn_fst m g thetae $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$
(U_e,V_e,W_e)) θ t =
n_ord_diffren_equat 0
(list_eta_longtnl_eqtn_motn_fst $(X_u,X_v, X_w, X_{wd}, X_q, X_\eta, X_\tau)$) η t

where the function longtnl_eqtn_motn_fst accepts the function variables u, w, θ and η and the lists of coefficients list_u_longtnl_eqtn_motn_fst, list_w_longtnl_eqtn_motn_fst, list_thta_longtnl_eqtn_motn_fst and list_eta_longtnl_eqtn_motn_fst and returns the corresponding DE. Similarly, the other two DEs of the LEoM (Eq. (2)) are modelled as follows:

Definition 14 *LEoM* 2
$\vdash_{def}$ ∀ Z_w Z_{wd} Z_q Z_η Z_τ Z_u.
list_u_longtnl_eqtn_motn_snd $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$ = [- Z_u]
$\vdash_{def}$ ∀ Z_u Z_q Z_η Z_τ Z_w m Z_{wd}.
list_w_longtnl_eqtn_motn_snd m $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$ = [Z_w; -(m - Z_{wd})]
$\vdash_{def}$ ∀ Z_u Z_w Z_{wd} Z_η Z_τ U_e W_e g θ_e Z_q m V_e.
list_thta_longtnl_eqtn_motn_snd m g θ_e $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$
(U_e,V_e,W_e) = [-(m ∗ g ∗ sin θ_e); Z_q + m ∗ V_e]
$\vdash_{def}$ ∀ Z_u Z_w Z_{wd} Z_q Z_η Z_τ.
list_eta_longtnl_eqtn_motn_snd $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$ = [Z_η]
$\vdash_{def}$ ∀p q r v u w m g θ_e U_e V_e W_e θ Z_u Z_w Z_{wd} Z_q Z_η Z_τ η t.
longtnl_eqtn_motn_snd
$(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$ (U_e,V_e,W_e) m g θ_e (p,q,r) (u,v,w) θ η t ⇔
n_ord_diffren_equat 0
(list_u_longtnl_eqtn_motn_snd $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$) u t -
n_ord_diffren_equat 1
(list_w_longtnl_eqtn_motn_snd $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$) w t -
n_ord_diffren_equat 1
(list_thta_longtnl_eqtn_motn_snd m g thetae $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$
(U_e,V_e,W_e)) θ t =
n_ord_diffren_equat 0
(list_eta_longtnl_eqtn_motn_snd $(Z_u,Z_w,Z_{wd},Z_q,Z_\eta,Z_\tau)$) η t

Definition 15 *LEoM* 3
$\vdash_{def}$ ∀ M_w M_{wd} M_q M_η M_τ M_u.
list_u_longtnl_eqtn_motn_trd $(M_u,M_w,M_{wd},M_q,M_\eta,M_\tau)$ = [- M_u]

$\vdash_{def}$ ∀ M_w M_{wd} M_q M_η M_τ M_u.
    list_w_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ) = [M_w; M_{wd}]
$\vdash_{def}$ ∀ M_w M_{wd} M_q M_η M_τ M_u I_x I_z I_y.
    list_thta_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ)
      (I_x,I_y,I_z) = [0; - M_q; I_y]
$\vdash_{def}$ ∀ M_w M_{wd} M_q M_η M_τ M_u.
    list_eta_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ) = [M_η]
$\vdash_{def}$ ∀Ue Ve We v u w Ix Iy Iz theta M_w M_{wd} M_q M_η M_τ M_u η t.
    longtnl_eqtn_motn_trd
        (M_u,M_w,M_{wd},M_q,M_η,M_τ) (U_e,V_e,W_e) (u,v,w) (I_x,I_y,I_z) θ η t ⇔
    n_ord_diffren_equat 0
     (list_u_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ)) u t -
    n_ord_diffren_equat 1
     (list_w_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ)) w t +
    n_ord_diffren_equat 2
     (list_thta_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ) (I_x,I_y,I_z)) θ t =
    n_ord_diffren_equat 0
     (list_eta_longtnl_eqtn_motn_trd (M_u,M_w,M_{wd},M_q,M_η,M_τ)) η t

Next, the LT of the LEoM, described by Eq. (2), is taken to compute the TF of the UV corresponding to its various inputs and outputs.

$$\begin{aligned} (ms - X_u)u(s) - (X_{\dot{w}}ws + X_w)w(s) - ((X_q - mW_e)s - mgcos\theta_e)\theta(s) &= X_\eta \eta(s) \\ -Z_u u(s) - ((Z_{\dot{w}} - m)s + Z_w)w(s) - ((Z_q + mU_e)s + mgsin\theta_e)\theta(s) &= Z_\eta \eta(s) \\ M_u u(s) - (M_{\dot{w}}s + M_w)w(s) + (I_y s^2 - M_q s)\theta(s) &= M_\eta \eta(s) \end{aligned} \tag{3}$$

The LT of the LEoM of the aircraft (Eq. 3) is verified as follows:

Theorem 3 LT of the LEoM
$\vdash_{thm}$ ∀ s u v w θ η U_e V_e W_e Z_q Z_τ Z_u Z_w Z_{wd} Z_η g m p q r θ_e.
  [C$_1$] (∀ t. $\frac{\mathrm{du}}{\mathrm{dt}}$ diff_at t) ∧
  [C$_2$] (∀ t. $\frac{\mathrm{dw}}{\mathrm{dt}}$ diff_at t) ∧
  [C$_3$] (∀ t. $\frac{\mathrm{d}^2\theta}{\mathrm{dt}^2}$ diff_att) ∧
  [C$_4$] (∀ t. η diff_att) ∧
  [C$_5$] zero_intl_cndtns 0 u ∧
  [C$_6$] zero_intl_cndtns 0 w ∧
  [C$_7$] zero_intl_cndtns 1 θ ∧
  [C$_8$] &0 < m ∧
  [C$_9$] &0 < g ∧
  [C$_{10}$] laplc_exst $\frac{\mathrm{du}}{\mathrm{dt}}$ s ∧
  [C$_{11}$] laplc_exst $\frac{\mathrm{dw}}{\mathrm{dt}}$ s ∧
  [C$_{12}$] laplc_exst $\frac{\mathrm{d}^2\theta}{\mathrm{dt}^2}$ s ∧

[C$_{13}$] laplc_exst η s $\wedge$
[C$_{14}$](∀ t.longtnl_eqtn_motn_fst(X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ)(U_e,V_e,W_e)
m g θ_e (p, q, r) (u, v, w) θ η $t)$ $\wedge$
[C$_{15}$](∀ t. longtnl_eqtn_motn_snd(Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ)(U_e,V_e,W_e)
m g θ_e (p, q, r)(u, v, w) θ η t) $\wedge$
[C$_{16}$](∀ t.longtnl_eqtn_motn_trd(M_u,M_w,M_{wd},M_q,M_η,M_τ)(U_e,V_e,W_e)
(u, v, w)(I_x,I_y,I_z)θ η t)
$\Rightarrow$ laplc_trnsfm_longtnl_eqtn_motn_fst(X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ)(U_e,V_e,W_e)
m g θ_e(u, v, w)θ ηs$\wedge$
laplc_trnsfm_longtnl_eqtn_motn_snd(Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ)(U_e,V_e,W_e)
m gθ_e (u, v, w) θ η s $\wedge$
laplc_trnsfm_longtnl_eqtn_motn_trd(M_u,M_w,M_{wd},M_q, M_η,M_τ)(U_e,V_e,W_e)
(I_x,I_y,I_z)θ η s

Conditions C$_1$–C$_4$ capture the differentiability of the higher-order derivatives of the input and outputs u, w, θ and η up to the order 1, 1, 2 and 0, respectively. Similarly, Conditions C$_5$–C$_7$ provide the *zero initial conditions* for the functions u, w and θ, respectively. Conditions C$_8$–C$_9$ present the positivity of the mass of the aircraft m and acceleration due to gravity g. Conditions C$_{10}$–C$_{13}$ assert that the LT of the functions u, w, θ and η exist up to the order 1, 1, 2 and 0, respectively. The last three conditions C$_{14}$–C$_{16}$ describe the LEoM of the UAV. The verification of the above theorem is mainly based on the following Lemma along with some arithmetic reasoning.

Lemma 1 LT of the Generalized Equation of Motion of Aircraft
$\vdash_{thm}$ ∀z y x w m n p q fslst snlst trlst ftlst s.
[C$_1$] (∀t. $\frac{d^m w}{dt^m}$ diff_at t) $\wedge$
[C$_2$] (∀t. $\frac{d^n x}{dt^n}$ diff_at t) $\wedge$
[C$_3$] (∀t. $\frac{d^p y}{dt^p}$ diff_at t) $\wedge$
[C$_4$] (∀t. $\frac{d^q z}{dt^q}$ diff_at t) $\wedge$
[C$_5$] (0< m $\Rightarrow$ zero_intl_cndtns (m − 1) w) $\wedge$
[C$_6$] (0< n $\Rightarrow$ zero_intl_cndtns (n − 1) w) $\wedge$
[C$_7$] (0< p $\Rightarrow$ zero_intl_cndtns (p − 1) w) $\wedge$
[C$_8$] (0< q $\Rightarrow$ zero_intl_cndtns (q − 1) w) $\wedge$
[C$_9$] laplc_exst $\frac{d^m w}{dt^m}$ s $\wedge$
[C$_{10}$] laplc_exst $\frac{d^n x}{dt^n}$ s $\wedge$
[C$_{11}$] laplc_exst $\frac{d^p y}{dt^p}$ s $\wedge$
[C$_{12}$] laplc_exst $\frac{d^q z}{dt^q}$ s $\wedge$
[C$_{13}$] (∀ t.diffren_equat_uav_gnrlzd m n p q fslst snlst trlst ftlst w x y z
$\Rightarrow$ laplc_trnsfrm w s ∗ $\sum_0^m$(λ k. EL k fslst ∗ s^k)+
laplc_trnsfrm x s ∗ $\sum_0^n$ (λk. EL k snlst ∗ s^k)+

laplc_trnsfrm y s $*\sum_0^{\mathrm{p}}$ (λk. EL k trlst $* s^k$) =
laplc_trnsfrm z s $*\sum_0^{\mathrm{q}}$ (λk. EL k ftlst $* s^k$)

Conditions C_1–C_4, C_5–C_8 and C_9–C_{12} are similar to Conditions C_1–C_4, C_5–C_7 and C_{10}–C_{13} of Theorem 3. For example, C_9–C_{12} of Lemma 1 provide the Laplace existence condition for the functions w, x, y and z exist up to the order m, n, p and q, respectively. Assumption C_{13} captures the DE modeling a generic equation of motion of UAV acting as a MIMO system and is formalized in HL as Definition 16. The formal proof of the above lemma is mainly based on the classical properties of the LT along with some complex arithmetic reasoning.

Definition 16 *Generic DE of Aircraft (MIMO System)*
$\vdash_{def}$ ∀m fslst w n snlst x p trlst y q ftlst z.
diffren_equat_uav_genrc m n p q fslst snlst trlst ftlst w x y z ⇔
(n_ord_diffren_equat m fslst w t + n_ord_diffren_equat n snlst x t +
n_ord_diffren_equat p trlst y t = n_ord_diffren_equat q ftlst z t)

Next, the LT of the LEoM (Eq. (3)) is written in matrix form as:

$$\begin{bmatrix} (ms - X_u) & -(X_{\dot{w}}ws + X_w) & -((X_q - mW_e)s - mgcos\theta_e) \\ -Z_u & -((Z_{\dot{w}} - m)s + Z_w) & -((Z_q + mU_e)s + mgsin\theta_e) \\ M_u & -(M_{\dot{w}}s + M_w) & (I_y s^2 - M_q s) \end{bmatrix} \begin{bmatrix} u(s) \\ w(s) \\ \theta(s) \end{bmatrix} = \begin{bmatrix} X_\eta \eta(s) \\ Z_\eta \eta(s) \\ M_\eta \eta(s) \end{bmatrix} \tag{4}$$

By applying the Crammer's rule, the TF of the UV for various inputs and outputs corresponding to the LEoM can be mathematically described as:

$$\frac{u(s)}{\eta(s)} \equiv \frac{N_\eta^u(s)}{\Delta_{lon}(s)} \; \frac{w(s)}{\eta(s)} \equiv \frac{N_\eta^w(s)}{\Delta_{lon}(s)} \; \frac{\theta(s)}{\eta(s)} \equiv \frac{N_\eta^\theta(s)}{\Delta_{lon}(s)} \tag{5}$$

where

$$N_\eta^u(s) = \begin{vmatrix} X_\eta & -(X_{\dot{w}}ws + X_w) & -((X_q - mW_e)s - mgcos\theta_e) \\ Z_\eta & -((Z_{\dot{w}} - m)s + Z_w) & -((Z_q + mU_e)s + mgsin\theta_e) \\ M_\eta & -(M_{\dot{w}}s + M_w) & (I_y s^2 - M_q s) \end{vmatrix}$$

$$N_\eta^w(s) = \begin{vmatrix} (ms - X_u) & X_\eta & -((X_q - mW_e)s - mgcos\theta_e) \\ -Z_u & Z_\eta & -((Z_q + mU_e)s + mgsin\theta_e) \\ M_u & M_\eta & (I_y s^2 - M_q s) \end{vmatrix}$$

$$N_\eta^\theta(s) = \begin{vmatrix} (ms - X_u) & -(X_{\dot{w}}ws + X_w) & X_\eta \\ -Z_u & -((Z_{\dot{w}} - m)s + Z_w) & Z_\eta \\ M_u & -(M_{\dot{w}}s + M_w) & M_\eta \end{vmatrix}$$

$$\Delta_{lon}(s) = \begin{vmatrix} (ms - X_u) & -(X_{\dot{w}}ws + X_w) & -((X_q - mW_e)s - mgcos\theta_e) \\ -Z_u & -((Z_{\dot{w}} - m)s + Z_w) & -((Z_q + mU_e)s + mgsin\theta_e) \\ M_u & -(M_{\dot{w}}s + M_w) & (I_ys^2 - M_qs) \end{vmatrix}$$

The TF $\eta(s)/u(s)$ for the input $\eta(t)$ and output $u(t)$ is verified in HL as the following theorem:

Theorem 4 TF $\eta(s)/u(s)$ for the Input $\eta(t)$ and Output $u(t)$

$\vdash_{thm} \forall$ I_x I_y I_z M_η M_q M_τ M_u M_w M_{wd} U_e V_e W_e X_η X_q X_τ X_u X_v X_w X_{wd} Z_η Z_q Z_τ Z_u Z_w Z_{wd} η g m s θ θ_e u v w.

[C$_1$] ($\forall t$. $\frac{\text{du}}{\text{dt}}$ diff_at t) $\wedge$
[C$_2$] ($\forall t$. $\frac{\text{dw}}{\text{dt}}$ diff_at t) $\wedge$
[C$_3$] ($\forall t$. $\frac{\text{d}^2\theta}{\text{dt}^2}$ diff_at t) $\wedge$
[C$_4$] ($\forall t$. η diff_at t) $\wedge$
[C$_5$] zero_intl_cndtns 0 u $\wedge$
[C$_6$] zero_intl_cndtns 0 w $\wedge$
[C$_7$] zero_intl_cndtns 1 θ $\wedge$
[C$_8$] & 0 < m $\wedge$
[C$_9$] & 0 < g $\wedge$
[C$_{10}$] laplc_exst $\frac{\text{du}}{\text{dt}}$ s $\wedge$
[C$_{11}$] laplc_exst $\frac{\text{dw}}{\text{dt}}$ s $\wedge$
[C$_{12}$] laplc_exst $\frac{\text{d}^2\theta}{\text{dt}^2}$ s $\wedge$
[C$_{13}$] laplc_exst η s $\wedge$
[C$_{14}$] nz_dnmntr_cndtn(X_u,X_v,X_w,X_{wd},X_q, X_η, X_τ) (Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ)
(M_u,M_w,M_{wd},M_q,M_η,M_τ) (U_e,V_e,W_e) m g θ_e (u, v, w) (I_x,I_y,I_z) θ η s $\wedge$
[C$_{15}$] ($\forall$ t.longtnl_eqtn_motn_fst(X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ) (U_e,V_e,W_e)
m g θ_e (p, q, r) (u, v, w) θ η t) $\wedge$
[C$_{16}$] ($\forall$ t.($\forall$ t.longtnl_eqtn_motn_snd(Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ) (U_e,V_e,W_e)
m g θ_e (p, q, r) (u, v, w) θ η t) $\wedge$
[C$_{17}$] ($\forall$ t. longtnl_eqtn_motn_trd(M_u,M_w,M_{wd},M_q,M_η,M_τ) (U_e,V_e,W_e)
(u, v, w)(I_x,I_y,I_z) θ η t)

$\Rightarrow$ laplc_trnsfm u s / laplc_trnsfm η s =
cdet (lngtd_nmrtr_polyn_mtrx_u η (X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ)
(Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ) (M_u,M_w,M_{wd},M_q,M_η,M_τ) (U_e,V_e,W_e)
m g θ_e (u, v, w)(I_x,I_y,I_z) θ η s)/
cdet (lt_lngtd_eqtn_mtrx(X_u,X_v, X_w, X_{wd}, X_q, X_η, X_τ) (Z_u,Z_w,Z_{wd},Z_q,Z_η,Z_τ)
(M_u,M_w,M_{wd},M_q,M_η,M_τ) (U_e,V_e,W_e) m g θ_e (u, v, w)(I_x,I_y,I_z) θ η s)

where cdet models the determinant of a complex valued square matrix. Conditions C$_1$–C$_{13}$ are same as that of Theorem 3. Condition C14 ensures that the denominators of the TF expression are non-zero. Conditions C_{15}–C_{17} present the LEoM of the aircraft. The verification of Theorem 4 is mainly based on Theorem 3, properties of complex matrices and vectors along with the following lemma regarding application of the Crammer's rule.

Lemma 2 Application of Crammer's Rule on a Matrix Representation
$\vdash_{thm} \forall\, \mathsf{a\, b\, c\, d\, e\, f\, g\, h\, i\, u\, w\, z\, n\, p\, q\, r}.$

$$[\mathsf{C}_1]\ \begin{bmatrix} \mathtt{a} & \mathtt{b} & \mathtt{c}\\ \mathtt{d} & \mathtt{e} & \mathtt{f}\\ \mathtt{g} & \mathtt{h} & \mathtt{i}\end{bmatrix} ** \begin{bmatrix} \frac{\mathtt{u}}{\mathtt{n}}\\ \frac{\mathtt{w}}{\mathtt{n}}\\ \frac{\mathtt{z}}{\mathtt{n}}\end{bmatrix} = \begin{bmatrix} \mathtt{p}\\ \mathtt{q}\\ \mathtt{r}\end{bmatrix} \land$$

$[\mathsf{C}_2]\ \mathsf{n} \neq 0 \land$
$[\mathsf{C}_3]\ \mathsf{a} * \mathsf{d} \neq 0 \land$
$[\mathsf{C}_4]\ \mathsf{a} * \mathsf{g} \neq 0 \land$
$[\mathsf{C}_5]\ \mathsf{d} * \mathsf{g} \neq 0 \land$
$[\mathsf{C}_6]\ \mathsf{b} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{d} * \mathsf{h} \neq 0 \land$
$[\mathsf{C}_7]\ \mathsf{b} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{e} * \mathsf{g} \neq 0 \land$
$[\mathsf{C}_8]\ \mathsf{a} * \mathsf{e} * \mathsf{i} - \mathsf{a} * \mathsf{f} * \mathsf{h} - \mathsf{b} * \mathsf{d} * \mathsf{i} + \mathsf{b} * \mathsf{f} * \mathsf{g} + \mathsf{c} * \mathsf{d} * \mathsf{h} - \mathsf{c} * \mathsf{e} * \mathsf{g} = 0$
$[\mathsf{C}_9]\ (\mathsf{b} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{d} * \mathsf{h}) * (\mathsf{c} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{f} * \mathsf{g}) - (\mathsf{b} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{e} * \mathsf{g}) * (\mathsf{c} * \mathsf{d} * \mathsf{g} - \mathsf{a} * \mathsf{d} * \mathsf{i}) \neq 0 \land$

$$\Rightarrow \frac{\mathtt{u}}{\mathtt{n}} = \frac{\mathtt{cdet} \begin{bmatrix} \mathtt{p} & \mathtt{b} & \mathtt{c}\\ \mathtt{q} & \mathtt{e} & \mathtt{f}\\ \mathtt{r} & \mathtt{h} & \mathtt{i}\end{bmatrix}}{\mathtt{cdet} \begin{bmatrix} \mathtt{a} & \mathtt{b} & \mathtt{c}\\ \mathtt{d} & \mathtt{e} & \mathtt{f}\\ \mathtt{g} & \mathtt{h} & \mathtt{i}\end{bmatrix}}$$

Condition C_1 captures the matrix representation of the DEs modeling the continuous dynamical behavior of a system. Conditions C_2–C_9 assert the non-zero conditions for the corresponding denominators the complex arithmetic manipulation of the matrix representation of the underlying system. Finally, the conclusion provides the TF $u(s)/n(s)$ by Crammer's rule. The construction of the formal proof of Lemma 2 mainly requires the properties of the complex-valued matrices and vectors along with some arithmetic reasoning. We also formally verified the Transfer Functions (TFs) corresponding to output functions, $w(t)$ and $\theta(t)$, i.e., $w(s)/\eta(s)$ and $\theta(s)/eta(s)$ and the details about their verification can be found at [1]. This concludes our formalization of the LEoM of UAVs, their FD solutions and the associated TFs.

The lateral-directional motion of UAVs is described by the side force (Y), rolling moment (L) and Yawing moment (N). To formalize the LDEoM, we first formalize the aerodynamics stability derivatives corresponding to lateral-directional motion, given in Table 5, using the type abbreviation feature of HL.

Table 5 Aerodynamics stability derivatives and their data-types

Symbol	Type	Symbol	Type	Symbol	Type
Y_v	$\mathbb{C}$	L_v	$\mathbb{C}$	N_v	$\mathbb{C}$
Y_p	$\mathbb{C}$	L_r	$\mathbb{C}$	N_r	$\mathbb{C}$
Y_r	$\mathbb{C}$	L_p	$\mathbb{C}$	N_p	$\mathbb{C}$
Y_ξ	$\mathbb{C}$	L_ξ	$\mathbb{C}$	N_ξ	$\mathbb{C}$

Definition 17 *Aerodynamics Stability Derivatives*

nw_typ_abrv ("arodyn_stbl_dervt_ltrl_frce", ‘:($Y_v \times Y_p \times Y_r \times Y_\xi$)‘)
nw_typ_abrv ("arodyn_stbl_dervt_rll_mmnt", ‘:($L_v \times L_p \times L_r \times L_\xi$)‘)
nw_typ_abrv ("arodyn_stbl_dervt_yw_mmnt", ‘:($N_v \times N_p \times N_r \times N_\xi$)‘)

The LDEoM of UAVs can be mathematically expressed as the following set of DEs [12, 41].

$$\begin{aligned} m\dot{v} - Y_v v - (Y_p + mW_e)p - (Y_r - mU_e)r - mg\phi cos\theta_e - mg\psi sin\theta_e &= Y_\xi \xi + Y_\zeta \zeta \\ -L_v v + I_x \dot{p} - L_p p - I_{xz}\dot{r} - L_r r &= L_\xi \xi + L_\zeta \zeta \\ -N_v v - I_{xz}\dot{p} - N_p p + I_z \dot{r} - N_r r &= N_\xi \xi + N_\zeta \zeta \end{aligned} \quad (6)$$

where $\dot{v}$, $\dot{p}$ and $\dot{r}$ presents the first order-derivatives of the linear and angular disturbance velocities, respectively. Similarly, the variables ξ and ζ model the aileron and rudder angles, respectively. Under the conditions $p(t) = \dot{\phi}(t)$, $r(t) = \dot{\psi}(t)$ and $\zeta(t) = 0$, the above LDEoM become [12]:

$$\begin{aligned} m\dot{v} - Y_v v - (Y_p + mW_e)\dot{\phi} - (Y_r - mU_e)\dot{\psi} - mg\phi cos\theta_e - mg\psi sin\theta_e &= Y_\xi \xi \\ -L_v v + I_x \ddot{\phi} - L_p \dot{\phi} - I_{xz}\ddot{\psi} - L_r \dot{\psi} &= L_\xi \xi \\ -N_v v - I_{xz}\ddot{\phi} - N_p \dot{\phi} + I_z \ddot{\psi} - N_r \dot{\psi} &= N_\xi \xi \end{aligned} \quad (7)$$

The first DE of the LDEoM of the UAV (Eq. (7)) is formalized as:

Definition 18 *LDEoM* 1

$\vdash_{def}$ ∀ Y_p Y_r Y_ξ Y_v m. **list_v_latrl_eqtn_motn_fst** m (Y_v,Y_p,Y_r,Y_ξ) = [- Y_v; m]
$\vdash_{def}$ ∀ U_e V_e Y_v Y_r Y_ξ g θ_e Y_p m W_e.
list_phi_latrl_eqtn_motn_fst m g θ_e (U_e,V_e,W_e) (Y_v,Y_p,Y_r,Y_ξ) =
[m ∗ g ∗ cos θ_e; Y_p + m ∗ W_e]
$\vdash_{def}$ ∀ Y_v Y_p Y_r Y_ξ V_e W_e g θ_e Y_r m U_e.
list_psi_latrl_eqtn_motn_fst m g θ_e (Y_v,Y_p,Y_r,Y_ξ) (U_e,V_e,W_e) =
[m ∗ g ∗ sin θ_e; Y_r - m ∗ U_e]
$\vdash_{def}$ ∀ Y_v Y_p Y_r Y_ξ. **list_xi_latrl_eqtn_motn_fst** (Y_v,Y_p,Y_r,Y_ξ) = [Y_ξ]
$\vdash_{def}$ ∀u w v ϕ m g θ_e U_e V_e W_e ψ Y_v Y_p Y_r Y_ξ ξ t.
latrl_eqtn_motn_fst (Y_v,Y_p,Y_r,Y_ξ) (U_e,V_e,W_e) m g θ_e (u,v,w) ϕ ψ ξ t ⇔

n_ord_diffren_equat 1 (list_v_latrl_eqtn_motn_fst m (Y_v,Y_p,Y_r,Y_ξ)) v t -
n_ord_diffren_equat 1 (list_phi_latrl_eqtn_motn_fst m g θ_e (U_e,V_e,W_e) (Y_v,Y_p,Y_r,Y_ξ)) ϕ
t -
n_ord_diffren_equat 1 (list_psi_latrl_eqtn_motn_fst m g θ_e (Y_v,Y_p,Y_r,Y_ξ) (U_e,V_e,W_e)) ψ
t =
n_ord_diffren_equat 0 (list_xi_latrl_eqtn_motn_fst (Y_v,Y_p,Y_r,Y_ξ)) ξ t

where the function latrl_eqtn_motn_fst accepts the function variables v, ϕ, ψ and ξ and the lists of coefficients list_v_latrl_eqtn_motn_fst, list_phi_latrl_eqtn_motn_fst, list_psi_latrl_eqtn_motn_fst and list_xi_latrl_eqtn_motn_fst and returns the corresponding DE modeling the dynamical behavior of the UAVs. Similarly, the other two DEs of the LDEoM of the UAV (Eq. (7)) are modelled as:

Definition 19 *LDEoM* 2

$\vdash_{def}$ $\forall$ L_p L_r L_ξ L_v. **list_v_latrl_eqtn_motn_snd** (L_v,L_p,L_r,L_ξ) = [-L_v]
$\vdash_{def}$ $\forall$ I_y I_z L_v L_r L_ξ L_p I_x. **list_phi_latrl_eqtn_motn_snd** (I_x,I_y,I_z) (L_v,L_p,L_r,L_ξ) = [0; L_p; - I_x]
$\vdash_{def}$ $\forall$ L_p L_r L_ξ L_v I_{xz}. **list_psi_latrl_eqtn_motn_snd** I_{xz} (L_v,L_p,L_r,L_ξ) = [0; L_r; I_{xz}]
$\vdash_{def}$ $\forall$ L_p L_r L_ξ L_v. **list_xi_latrl_eqtn_motn_snd** (L_v,L_p,L_r,L_ξ) = [L_ξ]
$\vdash_{def}$ $\forall$u w v I_x I_y I_z ϕ I_{xz} ψ L_v L_p L_r L_ξ ξ t.
latrl_eqtn_motn_snd (L_v,L_p,L_r,L_ξ) I_{xz} (I_x,I_y,I_z) (u,v,w) ϕ ψ ξ t $\Leftrightarrow$
n_ord_diffren_equat 0 (list_v_latrl_eqtn_motn_snd (L_v,L_p,L_r,L_ξ)) v t -
n_ord_diffren_equat 2 (list_phi_latrl_eqtn_motn_snd (I_x,I_y,I_z) (L_v,L_p,L_r,L_ξ)) ϕ t -
n_ord_diffren_equat 2 (list_psi_latrl_eqtn_motn_snd I_{xz} (L_v,L_p,L_r,L_ξ)) ψ t =
n_ord_diffren_equat 0 (list_xi_latrl_eqtn_motn_snd (L_v,L_p,L_r,L_ξ)) ξ t

Definition 20 *LDEoM* 3

$\vdash_{def}$ $\forall$ N_p N_r N_ξ N_v. **list_v_latrl_eqtn_motn_trd** (N_v,N_p,N_r,N_ξ) = [-N_v]
$\vdash_{def}$ $\forall$ N_p N_r N_ξ N_v I_{xz}. **list_phi_latrl_eqtn_motn_trd** I_{xz} (N_v,N_p,N_r,N_ξ) = [0; N_p; I_{xz}]
$\vdash_{def}$ $\forall$ N_p N_r N_ξ N_v I_x I_y I_z. **list_psi_latrl_eqtn_motn_trd** (N_v,N_p,N_r,N_ξ) (I_x,I_y,I_z) = [0; N_r; -I_z]
$\vdash_{def}$ $\forall$ N_p N_r N_ξ N_v. **list_xi_latrl_eqtn_motn_trd** (N_v,N_p,N_r,N_ξ) = [N_ξ]
$\vdash_{def}$ $\forall$u w v I_{xz} ϕ I_x I_y I_z ψ N_v N_p N_r N_ξ ξ t. **latrl_eqtn_motn_trd**
(N_v,N_p,N_r,N_ξ) (I_x,I_y,I_z) I_{xz} (u,v,w) ϕ ψ ξ t $\Leftrightarrow$
n_ord_diffren_equat 0 (list_v_latrl_eqtn_motn_trd (N_v,N_p,N_r,N_ξ)) v t -
n_ord_diffren_equat 2 (list_phi_latrl_eqtn_motn_trd I_{xz} (N_v,N_p,N_r,N_ξ)) ϕ t -
n_ord_diffren_equat 2 (list_psi_latrl_eqtn_motn_trd (N_v,N_p,N_r,N_ξ) (I_x,I_y,I_z)) ψ t =
n_ord_diffren_equat 0 (list_xi_latrl_eqtn_motn_trd (N_v,N_p,N_r,N_ξ)) ξ t

Next, in order to find out the TFs of the UV corresponding to its various inputs and outputs, the LT of the LDEoM is taken as follows:

$$\begin{aligned}(ms - Y_v)v(s) - ((Y_p + mW_e)s + mgcos\theta_e)\phi(s) - ((Y_r - mV_e)s + mgsin\theta_e)\psi(s) = \\ Y_\xi \xi(s) \\ -L_v v(s) + (I_x s^2 - L_p s)\phi(s) - (I_{xz} + L_r s)\psi(s) = \\ L_\xi \xi(s) \\ -N_v v(s) - (I_{xz} s^2 + N_p s)\phi(s) - (I_z - N_r s)\psi(s) = \\ L_\xi \xi(s)\end{aligned} \tag{8}$$

The LT of the LDEoM, i.e, Eq. (8) and various TFs, such as $v(s)/\xi(s)$, $\Delta(s)/\xi(s)$ and $\theta(s)/\xi(s)$ are also formally verified. The details about their verification can be found at [1].

6 Formal Stability Analysis of CropCam UAV

CropCam UAV [3, 35] is an autopilot aircraft that captures the GPS based digital images by flying at low-altitude. It provides the high resolutions images that are somewhat impossible to take using the conventional ways, such as satellite. It is widely utilized for remote sensing in various applications, such as large scale topographic mapping, agriculture and georeferencing etc. Stability is an important control characteristic of UAVs that smoothes the motion of the aircraft by dampening out any oscillation caused by various disturbances and thus ensures the equilibrium flight conditions [17]. It is based on the TFs of the UAV that are obtained by analyzing the dynamical behavior of the aircraft captures as the aircraft's equations of motion.

The TF of a system is mathematically expressed as follows:

$$\frac{Y(s)}{X(s)} = \frac{Numer(s)}{Denomer(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + ... + b_0}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{9}$$

where $X(s)$ and $Y(s)$ present the LT of the input function $x(t)$ and output function $y(t)$, respectively. Similarly, $Numer(s)$ and $Denomer(s)$ are complex-valued polynomials. The mathematical equation $Denomer(s) = 0$ is known as the characteristic equation and its roots are called the poles of the system. The locations of these poles in the complex plane presents important information about the stability of the corresponding system. A system is said to be stable if all the poles are located on the left half of the complex plane [17].

The notion of the stability of a UAV is formalized in HL as:

Definition 21 *Stability of a UAV*
$\vdash_{def}$ ∀Y. uav_stble Y = {s | Y s = 0 ∧ Re s < 0} ≠ { }

where uav_stble accepts the denominator Y: $\mathbb{C} \rightarrow \mathbb{C}$ of the TF corresponding to the aircraft's equations of motion of a UAV and returns a stable UAV. Similarly, s: $\mathbb{C}$ provides the root of the characteristic equation. The conjunct Y s = 0 provides the characteristic equation. Similarly, $\mathsf{Re}\ \mathsf{s} < 0$ ensures the location of the poles in the left half complex plane.

The LEoM for the CropCam UAV are mathematically modelled as below:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} z_u & z_w & 0 \\ m_u & m_w & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \end{bmatrix} = \begin{bmatrix} z_\eta \\ m_\eta \\ 0 \end{bmatrix} \eta \tag{10}$$

Alternatively, Eq. (10) can be written as follows:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} z_u & z_w \\ m_u & m_w \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} z_\eta \\ m_\eta \end{bmatrix} \eta \tag{11}$$

Next, the LT of above equation is taken for extracting the TFs of the CropCam UAV corresponding to its various inputs and outputs.

$$\begin{bmatrix} s - z_u & -z_w \\ s - m_u & -m_w \end{bmatrix} \begin{bmatrix} u(s) \\ w(s) \end{bmatrix} = \begin{bmatrix} z_\eta \eta(s) \\ m_\eta \eta(s) \end{bmatrix} \tag{12}$$

The TF $u(s)/\eta(s)$ corresponding to the LEoM for the CropCam UAV is given as follows [41]:

$$\frac{u(s)}{\eta(s)} = \frac{X_\eta \left(s + Z_w + X_w \frac{Z_\eta}{X_\eta} \right)}{s^2 - (Z_w + X_u)s + (Z_w X_u - Z_u X_w)} \tag{13}$$

Next, the stability of the CropCam UAV is formally verified by utilizing the denominator of the TF $(u(s)/\eta(s))$ as follows:

Theorem 5 Stability of CropCam UAV

$\vdash_{thm} \forall \alpha\ \mathtt{Z_w}\ \mathtt{Z_u}\ \mathtt{X_w}\ \mathtt{X_u}.$

$[\mathsf{C_1}]\ ((\mathtt{Re}\ \mathtt{Z_w} + \mathtt{Re}\ \mathtt{X_u} < 0 \land$

$((\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u})^2 - 4 * (\mathsf{Re}\ \mathtt{Z_w} * \mathsf{Re}\ \mathtt{X_u} - \mathsf{Re}\ \mathtt{Z_u} * \mathsf{Re}\ \mathtt{X_w}) < 0 \lor$

$(\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u})^2 - 4 * (\mathsf{Re}\ \mathtt{Z_w} * \mathsf{Re}\ \mathtt{X_u} - \mathsf{Re}\ \mathtt{Z_u} * \mathsf{Re}\ \mathtt{X_w}) = 0)) \lor$

$(0 < (\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u})^2 - 4 * (\mathsf{Re}\ \mathtt{Z_w} * \mathsf{Re}\ \mathtt{X_u} - \mathsf{Re}\ \mathtt{Z_u} * \mathsf{Re}\ \mathtt{X_w}) \land$

$(\sqrt{(\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u})^2 - 4 * (\mathsf{Re}\ \mathtt{Z_w} * \mathsf{Re}\ \mathtt{X_u} - \mathsf{Re}\ \mathtt{Z_u} * \mathsf{Re}\ \mathtt{X_w})} < -(\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u}) \lor$

$(\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u}) < \sqrt{(\mathsf{Re}\ \mathtt{Z_w} + \mathsf{Re}\ \mathtt{X_u})^2 - 4 * (\mathsf{Re}\ \mathtt{Z_w} * \mathsf{Re}\ \mathtt{X_u} - \mathsf{Re}\ \mathtt{Z_u} * \mathsf{Re}\ \mathtt{X_w})}))) \land$

$[\mathsf{C_2}]\ \mathsf{Im}\ \mathtt{Z_w} = 0 \land$

$[\mathsf{C_3}]\ \mathsf{Im}\ \mathtt{Z_u} = 0 \land$

$[\mathsf{C_4}]\ \mathsf{Im}\ \mathtt{X_w} = 0 \land$

$[\mathsf{C_5}]\ \mathsf{Im}\ \mathtt{X_u} = 0 \land$

$\Rightarrow \mathsf{uav\_stble}\ (\lambda\ \mathsf{s}.\ \mathtt{s}^2 - \mathtt{s} * (\mathtt{Z_w} + \mathtt{X_u}) + \mathtt{Z_w} * \mathtt{X_u} - \mathtt{Z_u} * \mathtt{X_w})$

Conditions $\mathsf{C_1}$–$\mathsf{C_5}$ assert constraints for the stability of the CropCam UAV. The proof process of the above theorem is mainly based on Definition 21 along with some

Table 6 Comparison of techniques for analyzing continuous dynamics of UAVs

	Paper-and-pencil proof	Computer-based simulations	Model checking	Theorem proving
Expressiveness	✓	✓		✓
Accuracy	✓(?)		✓	✓
Automation		✓	✓	

arithmetic reasoning. Similarly, the stability of the CropCam UAV based on its TF $w(s)/\eta(s)$ corresponding to its LEoM is formally verified. More details about the analysis can be found in the corresponding HL proof script at [1].

7 Discussions

The higher-order-logic TP based approach for analyzing the continuous dynamical behaviour of UAVs provided in this chapter allow us to clearly define all of the parameters along with their types that contribute to the dynamics of the aircraft. On the other hand, in the conventional approaches, like paper-and-pencil-proof and computer based techniques, there is always a chance of misinterpreting any of these parameters in analyzing the dynamics of UAVs. Moreover, all of the verified theorems and lemmas are of generic nature, i.e., they are verified for the universally quantified functions and variables and thus can be specialized for a particular scenario. Whereas, in computer based simulations, each of the cases are modelled individually. Moreover, the inherent soundness of our proposed approach ensures that all the required assumptions are explicitly mentioned along with the theorem and are incorporated in the corresponding proof. Table 6 provides a comparison of various methods used for analyzing the continuous dynamical behavior of UAVs and summarizes their strength and weaknesses. This comparison is performed based on various parameters, such as accuracy, expressiveness, accuracy and automation. For example, in higher-order-logic TP, we can truly model the continuous dynamics of UAVs, such as the DEs based aircraft's equations of motion and the corresponding TF in their true continuous form, whereas, in the MC based analysis, they are mostly discretized, which may compromise the accuracy of the analysis.

8 Conclusions

This chapter proposes a higher-order-logic TP based framework for formally analyzing the continuous dynamical behaviour of UAVs. Various coordinate frames, such as navigation and aircraft's body-fixed frames are formalized and their their associated transformations are also verified. This ensure the correctness of the relative movement and position of the aircraft. The longitudinal and LDEoM for UAVs are formalized their various properties, such as TF and stability, and solutions in FD are also formally verified. Finally, the stability analysis of the CropCam UAV is performed. In future, the formalization of the coordinate frames can be extended by incorporating other coordinates systems, such as quaternion. Another future direction is to formally verify some other interesting control system properties of UAVs, such as sensitivity.

References

1. Abed SE, Rashid A, Hasan O (2022) Formal analysis of unmanned aerial vehicles using higher-order-logic theorem proving: project webpage. http://save.seecs.nust.edu.pk/fauav/
2. Abed S, Rashid A, Hasan O (2020) Formal analysis of unmanned aerial vehicles using higher-order-logic theorem proving. J Aerosp Inf Syst 17(9):481–495
3. Ahmad A, Samad AM (2010) Aerial mapping using high resolution digital camera and unmanned aerial vehicle for geographical information system. In: International colloquium on signal processing and its applications. IEEE, pp 1–6
4. Aréchiga N, Loos SM, Platzer A, Krogh BH (2012) Using theorem provers to guarantee closed-loop system properties. In: American control conference. IEEE, pp 3573–3580
5. Baier C, Katoen JP (2008) Principles of model checking. MIT Press
6. Barmpounakis EN, Vlahogianni EI, Golias JC (2016) Unmanned aerial aircraft systems for transportation engineering: current practice and future challenges. Int J Transp Sci Technol 5(3):111–122
7. Birk A, Wiggerich B, Bülow H, Pfingsthorn M, Schwertfeger S (2011) Safety, security, and rescue missions with an unmanned aerial vehicle (UAV). J Intell Robot Syst 64(1):57–76
8. Carreno V, Muñoz C (2000) Aircraft trajectory modeling and alerting algorithm verification. In: Theorem proving in higher order logics. LNCS, vol 1869. Springer, pp 90–105
9. Chan M, Ricketts D, Lerner S, Malecha G (2016) Formal verification of stability properties of cyber-physical systems. In: Coq for programming languages
10. Chen X, Chen G (2018) Formal verification of helicopter automatic landing control algorithm in theorem prover Coq. Int J Performab Eng 14(9)
11. Clarke EM, Zuliani P (2011) Statistical model checking for cyber-physical systems. In: Automated technology for verification and analysis. LNCS, vol 6996. Springer, pp 1–12
12. Cook MV (2012) Flight dynamics principles: a linear systems approach to aircraft stability and control. Butterworth-Heinemann
13. Cooper J, Goodrich MA (2008) Towards combining UAV and sensor operator roles in UAV-enabled visual search. In: Human robot interaction. ACM, pp 351–358
14. Dennis LA, Fisher M, Webster MP, Bordini RH (2012) Model checking agent programming languages. Autom Softw Eng 19(1):5–63
15. Ducard GJ (2009) Fault-tolerant flight control and guidance systems: practical methods for small unmanned aerial vehicles. Springer Science & Business Media
16. Everaerts J et al (2008) The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping. Int Arch Photogramm Remote Sens Spat Inf Sci 37(2008):1187–1192
17. Foster T, Bowman J (2005) Dynamic stability and handling qualities of small unmanned-aerial vehicles. In: AIAA aerospace sciences meeting and exhibit, p 1023
18. Gallington RW, Berman H, Entzminger J, Francis MS, Palmore P, Stratakes J (1997) Unmanned aerial vehicles. In: Future aeronautical and space systems, AIAA, progress in astronautics and aeronautics, vol 172, pp 251–295
19. Ghorbal K, Jeannin JB, Zawadzki E, Platzer A, Gordon GJ, Capell P (2014) Hybrid theorem proving of aerospace systems: applications and challenges. J Aerosp Inf Syst 11(10):702–713

20. Groza A, Letia IA, Goron A, Zaporojan S (2015) A formal approach for identifying assurance deficits in unmanned aerial vehicle software. In: Progress in systems engineering, vol 366. Springer, pp 233–239
21. Guzey HM (2017) Hybrid consensus-based formation control of fixed-wing MUAVs. Cybern Syst 48(2):71–83
22. Harrison J (1996) HOL light: a tutorial introduction. In: Formal methods in computer-aided design. LNCS, vol 1166. Springer, pp 265–269
23. Harrison J (2009) Handbook of practical logic and automated reasoning. Cambridge University Press
24. Harrison J et al (1996) Formalized mathematics. Technical report 36, Turku Centre for Computer Science, Finland
25. Hasan O, Tahar S (2015) Formal verification methods. In: Encyclopedia of information science and technology. IGI Global Publication, pp 7162–7170
26. Karimoddini A, Lin H, Chen BM, Lee TH (2014) Hierarchical hybrid modelling and control of an unmanned helicopter. Int J Control 87(9):1779–1793
27. Loos SM, Renshaw D, Platzer A (2013) Formal verification of distributed aircraft controllers. In: International conference on hybrid systems: computation and control. ACM, pp 125–130
28. Lyon DH (2004) A military perspective on small unmanned aerial vehicles. Instrum Measur Mag 7(3):27–31
29. Ma Z, Chen G (2017) Formal derivation and verification of coordinate transformations in theorem prover Coq. In: International conference on dependable systems and their applications, pp 127–136
30. Malecha G, Ricketts D, Alvarez MM, Lerner S (2016) Towards foundational verification of cyber-physical systems. In: Science of security for cyber-physical systems. IEEE, pp 1–5
31. Munoz C, Narkawicz A (2016) Formal analysis of extended well-clear boundaries for unmanned aircraft. In: NASA formal methods symposium. LNCS, vol 9690. Springer, pp 221–226
32. Munoz C, Narkawicz A, Hagen G, Upchurch J, Dutle A, Consiglio M, Chamberlain J (2015) DAIDALUS: detect and avoid alerting logic for unmanned systems. In: Digital avionics systems conference. IEEE, pp 5A1–1–5A1–12
33. Narkawicz A, Munoz C (2012) Formal verification of conflict detection algorithms for arbitrary trajectories. Reliable Comput 17(2):209–237
34. Paulson LC (1996) ML for the working programmer. Cambridge University Press
35. Ping JTK, Ling AE, Quan TJ, Dat CY (2012) Generic unmanned aerial vehicle (UAV) for civilian application-a feasibility assessment and market survey on civilian application for aerial imaging. In: Sustainable utilization and development in engineering and technology. IEEE, pp 289–294
36. Seibel CW, Farines JM, Cury JE (1997) Towards using hybrid automata for the mission planning of unmanned aerial vehicles. In: International hybrid systems workshop. LNCS, vol 1567. Springer, pp 324–340
37. Shim D, Kim H, Sastry S (2000) Hierarchical control system synthesis for rotorcraft-based unmanned aerial vehicles. In: AIAA guidance, navigation, and control conference and exhibit, pp 1–9
38. Valavanis KP (2008) Advances in unmanned aerial vehicles: state of the art and the road to autonomy. Springer Science & Business Media
39. Webster M, Fisher M, Cameron N, Jump M (2011) Formal methods for the certification of autonomous unmanned aircraft systems. In: Computer safety, reliability, and security, LNPSE, vol 6894. Springer, pp 228–242
40. Williams KW (2004) A summary of unmanned aircraft accident/incident data: human factors implications. Technical report, Federal Aviation Administration
41. Xu Jr K (2014) Frequency domain system identification of fixed-wing unmanned aerial vehicles

Adaptive Fault-Tolerant Control Design for Multi-linked Two-Wheel Drive Mobile Robots

Ayad Al-Dujaili, Vincent Cocquempot, Maan El Badaoui El Najjar, Daniel Pereira, and Amjad Humaidi

Abstract This chapter presents the algorithmic design of an adaptive Fault Tolerant Control (FTC) to address several tasks needed for n-linked mobile robots subjected to actuator faults and friction phenomena. Firstly, it presents the development of kinematic and dynamic models for n-linked wheel drive mobile robot systems. Then, the kinematic model is transformed into a chained form and an approach to generate desired feasible trajectories is proposed. After that, a Lyapunov kinematic control law to control each state of the system and stabilize the tracking error is presented. In practice, the multi-robot system is affected by some disturbances, as for instance friction phenomena. Moreover, actuator faults may occur during the system life. Therefore, an adaptive law was designed to estimate the unknown friction coefficients and faults to adapt the control law online, leading to a multi-design integration-based adaptive controller. The asymptotical stability of the closed-loop is proven applying the Lyapunov theory. Simulations are performed in MATLAB/SIMULINK with different faulty cases, showing the efficiency of this method.

Keywords Fault tolerant control · n-linked mobile robots · Chained form · Actuator fault · Adaptive compensation control · Two-wheel drive robots

A. Al-Dujaili (✉)
Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
e-mail: ayad.qasim@mtu.edu.iq

V. Cocquempot · M. E. B. E. Najjar
UMR 9189, University of Lille, CRIStAL (Research Centre in Computer Science, Signal and Automatic Control of Lille), 59655 Villeneuve d'Ascq, France
e-mail: vincent.cocquempot@univ-lille.fr

M. E. B. E. Najjar
e-mail: maan.el-badaoui-el-najjar@univ-lille.fr

D. Pereira
Department of Automatics, Federal University of Lavras (UFLA), Lavras-MG, Brazil
e-mail: danielpereira@ufla.br

A. Humaidi
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
e-mail: Amjad.J.Humaidi@uotechnology.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_10

1 Introduction

The Wheeled Mobile Robots (WMR) are intensively utilized in applications like search and rescue, and operations in risky environments. Mobile robots with two-wheel drives (2WD) are frequently used in multiple cooperative robotic systems because of their simplicity, efficiency, and flexibility. However, this robot becomes uncontrollable when an actuator fails, causing free rolling of the wheel or blocking in a fixed position with undesirable frictions with the ground. To avoid these types of situations, additional actuators are needed. Physically linking multiple 2WD mobile robots is a good strategy to make the global system over-actuated. The robots may be linked initially or when needed, for instance when a component fault is detected.

A multi-linked 2WD mobile robot is an articulated mobile robot which is similar to a tractor-trailer system. These robotic systems can accomplish difficult tasks that may not be performed by just one mobile robot. Connecting several robots results in increasing the global system power which leads to the ability to carry heavy and large objects or to move in loose soil and rough terrain. As pointed out before, the additional redundancy in actuators and sensors can enhance fault tolerance.

The objective of the work presented in this chapter is to design FTC schemes for multi-linked 2WD mobile robots. The first purpose of the control design is the trajectory tracking, even in case of actuator faults. This study will adopt a multi-design integration-based adaptive approach to address all possible faulty cases.

The contributions of this chapter can be outlined as follows:

1. Firstly, kinematic and dynamic models of n-linked 2WD (for $n > 1$) mobile robots are derived.
2. Secondly, a compensation method based on FTC is proposed for n-linked 2WD mobile robots subjected faults in their actuators. The transformation of kinematic model for mobile robots into chained form is the first step of the FTC design. This transformation works to transform the nonlinear model in an equivalent linear one with a change of coordinates and a feedback transformation. Thus, the kinematic control law is designed using a recursive algorithm in order to provide that the system's states converge to reference trajectories. Then, dynamic control laws were designed, each of them using one possible failure pattern matrix.
3. In order to solve the problem of faults in multiple actuators and to cope with uncertain system friction, a multi-design integration-based adaptive FTC is proposed to ensure overall system stability and to guarantee asymptotic tracking properties.
4. An approach to generate feasible trajectories to be tracked by multi-linked mobile robots is proposed. Tracking these trajectories ensures the controllability of the system and avoids singularity situations.

This chapter is structured as follows: Firstly, in the Sect. 2, related work will be presented. The mathematical modelling is presented Sect. 3. Then, the transformation method of the system model into the chained form is presented in Sect. 4. The actuator fault model and the trajectory generation method are also presented in this section. Section 5 is dedicated to the design of the kinematic and dynamic controllers. In

Sect. 6, simulation results are presented to illustrate the effectiveness of this scheme. Finally, in the Sect. 7, the conclusions are made.

2 Related Works

This section relates the main literature about control design methods for wheeled mobile robots and multi-linked wheeled mobile robots so as to lay out the bases for the present work and justify the new developments made in this chapter. First, a review on trajectory tracking methods of WMR will be presented, followed by a brief review on trajectory tracking methods of multiple mobile robots. The section ends by mentioning several FTC algorithms with application to mobile robots and multi-linked mobile robots.

Several control problems for the autonomous navigation of WMR, such as trajectory tracking, and point stabilization, were studied and reported in the literature. This chapter focuses on trajectory tracking control for multi-linked WMR. The tracking problem of a WMR means that may the robot coordinates converge to a reference trajectory that may be fixed or generated by a moving target.

The 2WD mobile robots are nonholonomic systems, that have the particularity that the kinematic constraints cannot be integrated, which means that it is not possible to write these constraints as derivatives of functions of the generalized coordinates [1–3].

Depending on whether a dynamic or a kinematic model is adopted to represent the system behaviour, it is as a dynamic or a kinematic tracking control problem is considered. Taking kinematic models on nonholonomic WMR, different approaches like neural networks [4], neural-fuzzy systems [5], and backstepping [6], were applied to solve tracking control problems. These solutions rely on the consideration that the control inputs instantly assume the desired velocities. Considering dynamic feedback linearization and the concept of differential flatness, dynamic control laws are given by [7] and [8]. This approach doesn't consider robot dynamics and is well-known as perfect velocity tracking [9].

On the other hand, dynamic control strategies must consider a fully dynamic model that must capture the precise behaviour of the WMR in its totality by addressing the dynamic effects related to mass, inertia, and friction, which are generally not taken into account in kinematic controls. In [10], a dynamic reference trajectory tracking problem of a WMR is considered, and a neural network-based control law is derived. A dynamic control in a nonholonomic WMR with uncertainties in the kinematics and dynamics is presented by [11]. References [12]–[13] considered the trajectory tracking for nonholonomic systems with uncertainty in the parameters.

For a multi-body unicycle-like tractor with multiple trailers, a great review with many control schemes is presented in [14]. [15] presented a controllability study for these linked systems; the flatness of a multibody mobile robot is discussed in [16]; and nonholonomic motion planning is considered in [17]. Reference [18] presents a method based on a coordinate transformation considering the chained form.

Several authors have talked about kinematic control design methods. Reference [19] presents an alternative idea to stabilize nonholonomic systems, based on so-called transverse functions. In [20], the kinematic and dynamic control scheme has been proposed for a tractor with one trailer, while a tractor with two trailers have been used in [21]. However, actuator faults were not considered in these references.

Two bibliographical reviews on FTC design methods may be found in [22, 23], and many successful design schemes are presented for many different applications in [24–26].

Many FTC schemes were proposed for WMR, see for example [27–31] and the references therein. Besides FTC, Fault Diagnosis (FD) was discussed in [27, 28]. Reference [29] presented a model-based fault diagnosis method for skid and loss of performance. A sliding mode FTC method combined with a fault estimation system was presented in [30]. A two-step FTC method is presented in [31] to guarantee additional steering with a new use of the nominal controller that results in a driving model resembling a crab-like motion.

The faults occurring in the motion of WMR have a critical effect on the functionality of these robots [32]. A robust adaptive FTC approach for WMR is proposed in [33]. In [34], it is proposed a hybrid adaptive FTC that accommodates partial faults in 2WD robots. In [35], an adaptive FTC is designed for 2WD mobile robots subjected to faults on the actuated wheels. In [36], kinematic and dynamic models are used for fault detection taking into account the uncertainties on the mechanical parameters of the 2WD mobile robots. Instrumentation faults are considered in [37], where a fault-tolerant controller for a WMR in agriculture was designed. References [38–40] discuss FTC design for mobile robots with four-wheel drive (4WD).

Advanced control approaches have been proposed in the past few years [41, 42] for the tracking problem of WMR. In general, these methods solve a kinematic or dynamic control problem. In many applications, a kinematic controller applied to a WMR is enough for tracking a reference trajectory with satisfactory performance. Although, when there are actuator faults changing the input torques, a dynamic control law should also be considered. The backstepping approach [5] has been successfully used to derive the kinematic controller. Improved performance of the backstepping controller can be achieved with the adaptive kinematic controller proposed in [43]. References [44–47] present other kinematic control schemes that are neither adaptive nor backstepping controllers.

Kinematic controllers are designed as follows. For a simple case of two-linked robots, the kinematic controller is designed using a simple diffeomorphism. For more than two-linked robots, a chained form model derived from the initial kinematic model is used, as shown in Sect. 4, and the controller is designed using a recursive method based on the backstepping technique [48].

There are different types of dynamic controllers. In [49–51], the proposed adaptive dynamic controllers are extensions of kinematic backstepping controllers designed considering the unknown parameters. Robust dynamic control laws can be designed to take into account disturbances and uncertainties [52] using techniques such as sliding mode [53, 54] and H_∞ [55] control laws. Some intelligent and sliding mode controllers [56, 57] approaches can combine adaptive and robust properties. The feedback linearization control can be applied for the compensation of the nonlinear system dynamics (see for instance [58]).

A dynamic controller can be designed to make the mobile robot velocities converge to the reference values. An adaptive control algorithm whose principle may be found in reference [59] adopts two alternative fault compensation schemes: switching between multiple dynamic controllers, each one designed for the compensation of a specific actuator fault; designing an adaptive controller based on a multi-integration mechanism that can compensate all the actuator faults in a single control law.

There are very few studies in the literature on fault accommodation for multi-linked WMR. An FTC method was derived in [60–63] for mobile robots with two-linked 2WD. Furthermore, [64, 65] proposed a multiple dynamic controller to overcome actuator faults in three-linked 2WD mobile robots. Based on the review of the literature on FTC and the application of multi-linked mobile robots, one can notice that friction phenomena have not been addressed in the aforementioned research work. Therefore, one of the significant contributions of this study is the derive an FTC method that includes friction models and uncertain faults.

3 Mathematical Modelling

3.1 Modelling of n-linked 2WD Mobile Robots

The models used in the system are presented in this section. The front passive caster of the robots is omitted, and there are two actuated wheels on the rear. The physical links and the orientations of the robots are compatible, however, the robot 1 has an independent orientation. The following notations are used in the sequel. For the i th) robot: P_i is the middle point between the wheels, C_i is the center of mass, a_i is the distance from P_i to C_i, b_i is the half of the distance from one actuated wheel to the other, r_i is the wheel radius, θ_i is the robot orientation, and τ_{il} and τ_{ir} are the input torque signals on the wheels. The frame OXY represents the inertial frame, d_i denotes the length of the links, and (x, y) denotes the coordinates of the middle point of the rear wheels of robot n (the last one), represented by point P_n. The system, as shown in Fig. 1, can be represented by the generalized coordinates vector

$$q = \left[x, y, \theta_n, \theta_{n-1}, \ldots, \theta_1\right]^T \tag{1}$$

3.1.1 Kinematic Modelling

In WMR, the nonholonomic constraint is a very important characteristic that is related to non-slipping rolling. These system constraints can be represented by:

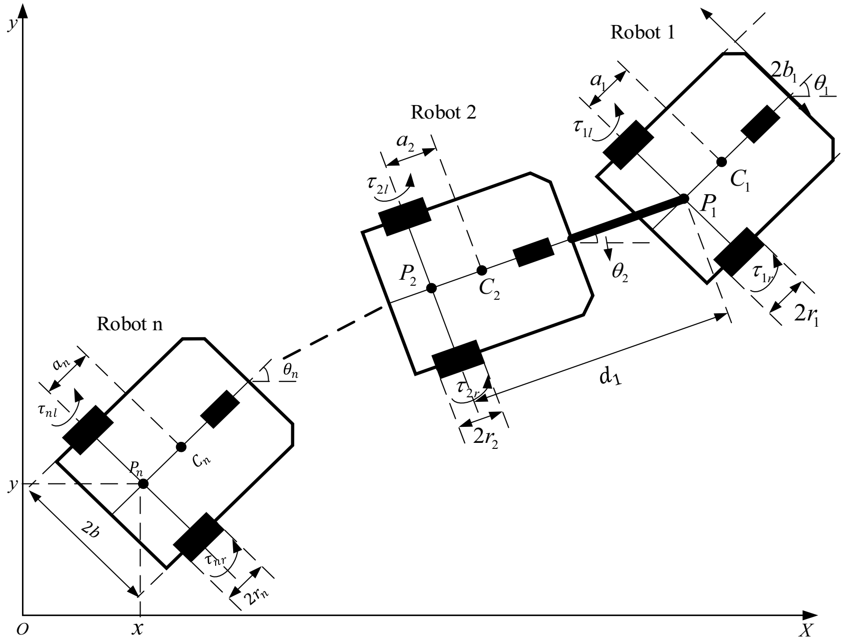


Fig. 1 Multi-linked 2WD mobile robots

$$A(q)\dot{q} = 0 \tag{2}$$

where

$$A(q) = \left[\begin{array}{ccc} sin\theta_n & -cos\theta_n & 0 \\ sin\theta_{n-1} & -cos\theta_{n-1} & -d_n cos(\theta_{n-1} - \theta_n) \\ sin\theta_{n-2} & -cos\theta_{n-2} & -d_n cos(\theta_{n-2} - \theta_n) \\ \vdots & \vdots & \vdots \\ sin\theta_2 & -cos\theta_2 & -d_n cos(\theta_2 - \theta_n) \\ sin\theta_1 & -cos\theta_1 & -d_n cos(\theta_1 - \theta_n) \end{array} \right.$$

$$\left. \begin{array}{cccc} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & -d_2 cos(\theta_2 - \theta_3) & 0 & 0 \\ 0 & -d_2 cos(\theta_1 - \theta_3) & -d_1 cos(\theta_1 - \theta_2) & 0 \end{array} \right] \tag{3}$$

The kinematic equations are given by

$$\dot{q} = S(q)\eta \tag{4}$$

where $\eta = \begin{bmatrix} \upsilon_n & \omega_1 \end{bmatrix}^T$, υ_n and ω_1 represent the linear velocity of robot n and rotational velocity of robot 1, respectively. The matrix $S(q)$ is given by

$$S(q) = \begin{bmatrix} \cos\theta_n & 0 \\ \sin\theta_n & 0 \\ \frac{1}{d_{n-1}} \tan(\theta_{n-1} - \theta_n) & 0 \\ \vdots & \vdots \\ \frac{1}{d_{n-k-1}} \tan(\theta_{n-k-1} - \theta_{n-k}) / \prod_{i=n-k}^{n-1} \cos(\theta_i - \theta_{i+1}) & 0 \\ \vdots & \vdots \\ \frac{1}{d_1} \tan(\theta_1 - \theta_2) / \prod_{i=1}^{n-1} \cos(\theta_i - \theta_{i+1}) & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

where $(k = 2, 3, \ldots, n - 2)$.

An important property is that

$$S(q)^T \cdot A(q)^T = 0 \tag{6}$$

3.1.2 Dynamic Modelling

The dynamic model can be derived by Lagrange formulation:

$$M(q) \cdot \dot{q} + C(q, \dot{q}) = B(q)\tau + A(q)^T \cdot \lambda \tag{7}$$

where, $\tau \in R^{(2n)\times(1)}$ is the vector of control torque, $\lambda \in R^{n\times 1}$ is the vector of Lagrange multipliers, $A(q) \in R^{n\times(n+2)}$ is the system constraint matrix, $B(q) \in R^{(n+2)\times(2n)}$ is the input transformation matrix, $C(q, q) \in R^{(n+2)\times(1)}$ is the centripetal Coriolis matrix, and $M(q) \in R^{(n+2)\times(n+2)}$ is the symmetric, positive definite matrix. The vectors and matrices in (7) are given by

$$M(q) = \left[m_{ij}\right],\ i \wedge j \in \{1, 2, ..., n + 2\} \tag{8}$$

$$C(q, q) = [c_i],\ i \in \{1, 2, ..., n + 2\} \tag{9}$$

$$B(q) = \left[b_{ij}\right], i \in \{1, 2, ..., n + 2\},\ j \in \{1, 2, ..., 2n\} \tag{10}$$

In Appendix 1 and 2, detailed descriptions and derivations of $M(q)$, $C(q, q)$ and $B(q)$ are presented.

The substitution of the time derivative of (4) in (7), and the multiplication by $S(q)^T$, remove the Lagrange multipliers, providing:

$$\overline{\boldsymbol{M}}_1(\boldsymbol{q})\dot{\boldsymbol{\eta}} + \overline{\boldsymbol{M}}_2(\boldsymbol{q})\boldsymbol{\eta} + \overline{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \overline{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\tau} \tag{11}$$

where,

$$\overline{M}_1(q) = S(q)^T \cdot M(q) \cdot S(q) \tag{12}$$

$$\overline{M}_2(q) = S(q)^T \cdot M(q) \cdot S(q) \tag{13}$$

$$\overline{B}(q) = S(q)^T \cdot B(q) \tag{14}$$

$$\overline{C}(q, q) = S(q)^T \cdot C(q, q) \tag{15}$$

3.2 Modelling of Three-Linked 2WD Mobile Robots

Let us first consider the case of three-linked robots. The front passive casters of the robots are omitted, and the two rear wheels are actuated. The leader robot is robot 1, while the followers are robots 2 and 3. The followers have orientations that are compatible with the links, but the leader has an independent orientation. The whole system has a configuration described by $q = [x, y, \theta_3, \theta_2, \theta_1]$.

3.2.1 Kinematic Modelling

As previously mentioned, the non-holonomic feature of a WMR is related to rolling without slipping. The system constraints can be represented by

$$A(q) \cdot q = 0 \tag{16}$$

where

$$A(q) = \begin{bmatrix} sin\theta_3 & -cos\theta_3 & 0 & 0 & 0 \\ sin\theta_2 & -cos\theta_2 & -d_2cos(\theta_2 - \theta_3) & 0 & 0 \\ sin\theta_1 & -cos\theta_1 & -d_2cos(\theta_1 - \theta_3) & -d_1cos(\theta_1 - \theta_2) & 0 \end{bmatrix} \tag{17}$$

The kinematic equations of the multi-linked system presented in Fig. 2 are

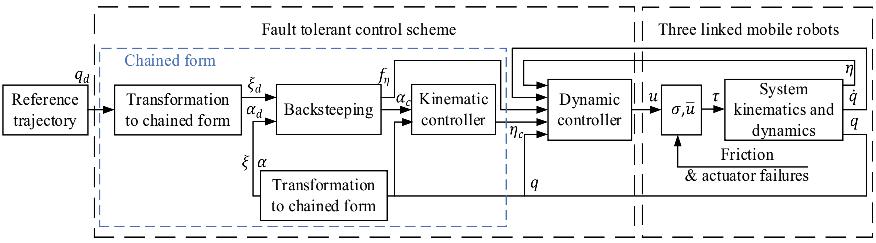


Fig. 2 Block diagram of the fault compensation control with coordinate and feedback transformation

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q}) \cdot \boldsymbol{\eta} \tag{18}$$

where $\eta = \begin{bmatrix} \upsilon_3 \ \omega_1 \end{bmatrix}^T$, ω_1 and υ_3 represent the rotational velocity of robot 1 and the linear velocity of robot 3, and is the. The matrix $S(q)$ is given by:

$$S(q) = \begin{bmatrix} cos\theta_3 & sin\theta_3 & \frac{1}{d_2}tan(\theta_2 - \theta_3) & \frac{1}{d_1}tan(\theta_1 - \theta_2)/cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{19}$$

Moreover, an important property is the orthogonal complement represented by

$$S(q)^T \cdot A(q)^T = 0 \tag{20}$$

3.2.2 Dynamic Modelling

In case of three-linked 2WD mobile robots, the Lagrange formulation can be applied to write the dynamic model as

$$M(q)q + C(q, q) = B(q) \cdot \tau + A(q)^T \cdot \lambda \tag{21}$$

where, $\lambda \in R^{3\times1}$ represents the Lagrange multipliers vector, $A(q) \in R^{3\times5}$ is the system constraint matrix, $\tau \in R^{6\times1}$ is the vector of control torques, $B(q) \in R^{5\times6}$ is the input transformation matrix, $C(q, \dot{q}) \in R^{5\times1}$ is the vector for centripetal and Coriolis forces, $M(q) \in R^{5\times5}$ is the system inertia matrix. The matrices $M(q)$, $C(q, \dot{q})$ and $B(q)$, appearing in (21), are given by

$$M(q) = \left[\begin{matrix} m_1 + m_2 + m_3 & 0 \\ 0 & m_1 + m_2 + m_3 \\ -[(m_1 + m_2)d_2 + a_3m_3]\sin\theta_3 & [(m_1 + m_2)d_2 + a_3m_3]\cos\theta_3 \\ -[d_1m_1 + a_2m_2]\sin\theta_2 & [d_1m_1 + a_2m_2]\cos\theta_2 \\ a_1m_1\sin\theta_1 & a_1m_1\cos\theta_1 \end{matrix} \right.$$

$$\left.\begin{matrix} -[(m_1+m_2)d_2+a_3m_3]\sin\theta_3 & -[d_1m_1+a_2m_2]\sin\theta_2 & a_1m_1\sin\theta_1 \\ [(m_1+m_2)d_2+a_3m_3]cos\theta_3 & [d_1m_1+a_2m_2]cos\theta_2 & a_1m_1cos\theta_1 \\ (m_1+m_2)d_2^2+m_3a_3^2+I_{m3} & (d_1m_1+a_2m_2)d_2cos(\theta_2-\theta_3) & a_1m_1d_2cos(\theta_1-\theta_3) \\ (d_1m_1+a_2m_2)d_2cos(\theta_2-\theta_3) & m_1d_1^2+m_2a_2^2+I_{m2} & a_1m_1d_1cos(\theta_1-\theta_2) \\ a_1m_1d_2cos(\theta_1-\theta_3) & a_1m_1d_2cos(\theta_1-\theta_3) & m_1a_1^2+I_{m1} \end{matrix}\right]$$

$$B(q)=\begin{bmatrix} \frac{\cos\theta_1}{r_1} & \frac{\cos\theta_1}{r_1} & \frac{\cos\theta_2}{r_2} & \frac{\cos\theta_2}{r_2} & \frac{\cos\theta_3}{r_3} & \frac{\cos\theta_3}{r_3} \\ \frac{\sin\theta_1}{r_1} & \frac{\sin\theta_1}{r_1} & \frac{\sin\theta_2}{r_2} & \frac{\sin\theta_2}{r_2} & \frac{\sin\theta_3}{r_3} & \frac{\sin\theta_3}{r_3} \\ \frac{d_2\sin(\theta_1-\theta_3)}{r_1} & \frac{d_2\sin(\theta_1-\theta_3)}{r_1} & \frac{d_2\sin(\theta_2-\theta_3)}{r_2} & \frac{d_2\sin(\theta_2-\theta_3)}{r_2} & \frac{b_3}{r_3} & \frac{b_3}{r_3} \\ \frac{d_1\sin(\theta_1-\theta_3)}{r_1} & \frac{d_1\sin(\theta_1-\theta_3)}{r_1} & \frac{-b_2}{r_2} & \frac{-b_2}{r_2} & 0 & 0 \\ \frac{b_1}{r_1} & \frac{-b_1}{r_1} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C(q,\dot{q})=\left[C_{ij}\right]_{5\times1}$$

where

$$c_{11}=-[(m_1+m_2)d_2+a_3m_3]\dot{\theta}_3^2\cos\theta_3(d_1m_1+a_2m_2)\dot{\theta}_2^2\cos\theta_2-a_1m_1\dot{\theta}_1^2\cos\theta_1$$

$$\begin{aligned} c_{21}=&-(d_1m_1+a_2m_2)d_2\dot{\theta}_2^2\sin(\theta_2-\theta_3)-a_1m_1d_2\dot{\theta}_1^2\sin(\theta_1-\theta_3)\\ &+2\big(d_1m_1+a_2m_2d_2\dot{\theta}_2\dot{\theta}_3\sin\big). \end{aligned}$$

$$\begin{aligned} c_{31}=&-\,(d_1m_1+a_2m_2)d_2\dot{\theta}_2^2\sin(\theta_2-\theta_3)-a_1m_1d_2\dot{\theta}_1^2\sin(\theta_1-\theta_3)+\\ &2+a_2m_2d_2\dot{\theta}_2\dot{\theta}_3\sin(\theta_2-\theta_3)+2a_1m_1d_2\dot{\theta}_1\dot{\theta}_3\sin(\theta_1-\theta_3) \end{aligned}$$

$$\begin{aligned} c_{41}=&-\,(d_1m_1+a_2m_2)d_2\dot{\theta}_3^2\sin(\theta_2-\theta_3)-a_1m_1d_2\dot{\theta}_1^2\sin(\theta_1-\theta_2)\\ &+2m_1d_1\dot{\theta}_1\dot{\theta}_3\sin(\theta_1-\theta_2) \end{aligned}$$

$$c_{51}=a_1m_1d_2\dot{\theta}_3^2\sin(\theta_1-\theta_2)+a_1m_1d_1\dot{\theta}_2^2\sin(\theta_1-\theta_2)$$

The Lagrange multipliers can be removed after the substitution of the derivative of (18) in (21), and the multiplication by $S(q)^T$, which provides

$$\overline{\boldsymbol{M}}_1(\boldsymbol{q})\cdot\dot{\boldsymbol{\eta}}+\overline{\boldsymbol{M}}_2(\boldsymbol{q})\cdot\boldsymbol{\eta}+\overline{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})=\overline{\boldsymbol{B}}(\boldsymbol{q})\cdot\boldsymbol{\tau}) \tag{22}$$

where,

$$\overline{M}_1(q)=S(q)^T\cdot M(q)\cdot S(q) \tag{23}$$

$$\overline{M}_2(q)=S(q)^T\cdot M(q)\cdot\dot{S}(q) \tag{24}$$

$$\overline{B}(q) = S(q)^T \cdot B(q) \tag{25}$$

$$\overline{C}(q, \dot{q}) = S(q)^T \cdot C(q, \dot{q}) \tag{26}$$

When the objective is to control a system using two models (kinematic and dynamic models for the same system), a link (in the form of a diffeomorphism) must be found between the control vectors of the two models to elaborate the control signals.

In case of two-linked mobile robots, it is simple to find the diffeomorphism. But in the case of more than two robots, it is much more difficult. One solution is to convert the kinematic model into the canonical chained form, making it relatively simple to find the diffeomorphism. The present work used the canonical chained form in order not only to find the needed diffeomorphism but also to design the control law more easily.

4 Conversion of the Model into the Canonical Chained Form

The commonly used approach to design the controller for a nonholonomic system is a transformation into a canonical form that makes the controller design easier. One of the commonly used canonical forms for robotic systems is the so-called chained form, where the time derivative of a system state relies on the state above it [66–69].

A special chained form is the two-input chained form, which is expressed as follows

$$\begin{aligned} \dot{\xi}_1 &= \alpha_1 \\ \dot{\xi}_2 &= \alpha_2 \\ \dot{\xi}_3 &= \xi_2 \alpha_1 \\ &\vdots \\ \dot{\xi}_m &= \xi_{m-1} \alpha_1 \end{aligned} \tag{27}$$

where α_1, α_2 are the two virtual control inputs, $\xi = (\xi_1, \xi_2, \ldots, \xi_m)$ is the state vector, and m denotes the number of the states.

In practice, the two-input condition is broad enough to deal with the majority of the kinematic models of the WMR. A more detailed discussion of the multi-chained forms can be found in [68], which is the original reference for these canonical forms.

The motivations for the conversion of the initial state model into the canonical chained form are:

1. To determine a diffeomorphism to link kinematic and dynamic controllers for mobile robots with n-linked 2WD ($n \geq 2$).
2. To develop backstepping control law that permits controlling each state of the system.
3. The next subsection, inspired by the exact linearization method of a nonlinear model presented in [67], presents a theorem which expresses the way to make the transformation of the kinematic model of n-linked robot systems into the chained form.

4.1 Kinematic Model Transformation of n-linked Robots into the Chained Form

This subsection presents a theorem used to derive the diffeomorphism used to transform the multi-linked 2WD mobile robots showed in Eq. (4) into a chained form.

In [18], it is demonstrated how to transform the kinematic model of a vehicle with n-trailers into a chained form. The method is used here for a n-linked 2WD mobile robot system.

Assumption 1 The states of state vector $q = \left[x, y, \theta_n, \theta_{n-1}, \ldots, \theta_1\right]^T$ are in a neighbourhood D of the origin, where D is given by

$$(x, y) \in R^2$$

$$\theta_i \in \left(\frac{-\pi}{4} + \in, \frac{\pi}{4} - \in\right), \; i \in \{1, \ldots, n\}$$

where $\in$ is a small constant. A transformation is performed on the input υ:

$$\upsilon = cos\theta_n \cdot \upsilon_n = cos\theta_n \prod_{j=2}^{n} cos\left(\theta_{j-1} - \theta_j\right) \cdot \upsilon_1 \tag{28}$$

The transformed variable υ is the velocity of robot n in the x-direction. The transformation is non-singular and smooth in D, being υ_1 is the linear velocity of robot 1, which is considered as a virtual input. The angular velocity of robot 1, ω_1, is a virtual input as well.

The linear velocity of robot i,υ_i can be written as

$$\upsilon_i = cos(\theta_{i-1} - \theta_i)\upsilon_{i-1} = \prod_{j=2}^{i} cos\left(\theta_{j-1} - \theta_j\right)\upsilon_1 \tag{29}$$

for $i \in \{2, \nu, n\}$. Then Eq. (29) can be rewritten using Eq. (28) as follows

$$\upsilon_i = \frac{1}{cos\theta_n \prod_{j=i+1}^{n} cos\left(\theta_{j-1} - \theta_j\right)}\upsilon = \frac{1}{p_i(\theta_i)}\upsilon \tag{30}$$

for $i \in \{1, \nu, n\}$

where

$$\theta_i \triangleq [\theta_i, ..., \theta_n]^T \tag{31}$$

$$p_i(\theta_i) \triangleq cos\theta_n \prod_{j=i+1}^{n} cos\left(\theta_{j-1} - \theta_j\right) = \prod_{j=i}^{n} cos\left(\theta_j - \theta_{j+1}\right) \tag{32}$$

where $\theta_{n+1}\nu 0$.

Equation (30) then gives $\upsilon = p_1(\theta_1)\upsilon_1$.

Inspired by [18], next theorem can thus be given:

Theorem 1 *The following nonlinear change of coordinates, $\xi = \phi(q)$, and feedback transformation, $\alpha = G(q)\eta$, locally convert the kinematic model (4) of the n-linked 2WD mobile robot system into the chained form (27).*

The change of coordinates, $\xi = \phi(q)$ is given by

$$\xi_1 = x \tag{33}$$

$$\xi_2 = tan(\theta_1 - \theta_2)/c_2(\theta_2) + r_2(\theta_2) \tag{34}$$

$$\vdots$$

$$\xi_i = tan(\theta_{i-2} - \theta_{i-1})/c_i(\theta_{i-1}) + r_i(\theta_{i-1}) \tag{35}$$

$$\vdots$$

$$\xi_n = tan(\theta_{n-1} - \theta_n)/d_{n-1}cos^3\theta_n \tag{36}$$

$$\xi_{n+1} = tan\theta_n \tag{37}$$

$$\xi_{n+2} = y \tag{38}$$

where

$$c_i(\theta_{i-1}) = \prod_{j=2}^{n+1} cos^{j-i+3}\big(\theta_{j-1} - \theta_j\big) d_{n+i-j} = p_{i-1}^2(\theta_{i-1}) \prod_{j=i-1}^{n} d_j p_j\big(\theta_j\big) \tag{39}$$

$$r_i(\theta_{i-1}) = \frac{\partial \xi_{i+1}}{\partial \theta_i} f_i(\theta_{i-1}) \tag{40}$$

$$f_i(\theta_{i-1}) = \frac{1}{d_i} tan(\theta_{i-1} - \theta_i)/p_i(\theta_i) \tag{41}$$

The feedback transformation of inputs $\alpha = G(q)\eta$ is given by

$$\alpha_1 = p_1(\theta_1)\upsilon_1 \tag{42}$$

$$\alpha_2 = 1/cos^2(\theta_1 - \theta_2)c_2(\theta_2)\omega_1 + r_1(\theta_1)p_1(\theta_1)\upsilon_1 \tag{43}$$

The proof of *Theorem* 1 is the following:

The kinematic model for n-linked 2WD mobile robots in Eq. (4) were derived in Sect. 3 and can be represented (locally) as follows

$$\dot{x} = \upsilon$$

$$\dot{\theta}_1 = \omega_1$$

$$\dot{\theta}_2 = \frac{\upsilon}{d_2}[\tan(\theta_1 - \theta_2)/p_2(\theta_2)]$$

$$\vdots$$

$$\dot{\theta}_i = \frac{\upsilon}{d_i}\big[\tan(\theta_{i-1} - \theta_i)/p_i(\theta_i)\big]$$

$$\dot{y} = \tan\theta_n \upsilon \tag{44}$$

for $i \in \{1, \nu, n\}$, where υ is given by (28). If $i \in \{1, ..., n\}$ is denoted by

$$f_i(\theta_{i-1}) = \frac{1}{d_i} \cdot \big[tan(\theta_{i-1} - \theta_i)/p_i(\theta_i)\big] \tag{45}$$

This means that we can write

$$\dot{\theta}_i = f_i(\theta_{i-1})\upsilon \tag{46}$$

After reordering the state variables, the state is given by vector

$$\left[\xi_1, \xi_2, \xi_{n+2}\right]^T = \left[x, y, \theta_n, \theta_{n-1}, \ldots, \theta_1\right]^T$$

whose dimension is $n + 2$.

From (44) it can be seen that this kinematic model has a special triangular structure where $\dot{\xi}_i$ is not a function of $\xi_1, \ldots, \xi_{i-2}$, where $i \in \{3, \ \ldots \ , n+2\}$ and Let us define.

$$\xi_{n+3} = y \tag{47}$$

Using (44), the differentiation of (47) with respect to time gives

$$\dot{\xi}_{n+3} = \dot{y} = tan\theta_n\upsilon \tag{48}$$

We have $\dot{\xi}_{n+3} = \xi_{n+2}\upsilon$ by choosing

$$\xi_{n+2} = tan\theta_n \tag{49}$$

Using (44), the differentiation of (49) with respect to time gives

$$\dot{\xi}_{n+2} = \frac{1}{cos^2\theta_n}\dot{\theta}_n = \frac{tan(\theta_{n-1} - \theta_n)}{d_n cos^3\theta_n}\upsilon \tag{50}$$

We obtain $\dot{\xi}_{n+2} = \xi_{n+1}\upsilon$ by choosing

$$\xi_{n+1} = \frac{tan(\theta_{n-1} - \theta_n)}{d_n cos^3\theta_n} \tag{51}$$

It can be shown by induction that $\dot{\xi}_{i+2} = \xi_i\upsilon$ by choosing

$$\xi_i = \frac{tan(\theta_{i-2} - \theta_{i-1})}{c_i(\theta_{i-1})} + r_i(\theta_{i-1}) \tag{52}$$

for $i \in \{2, ..., n\}$ where

$$c_i(\theta_{i-1}) = \prod_{j=1}^{n+1} \cos^{j-i+3}\left(\theta_{j-1} - \theta_j\right)d_{n+i-j} = p_{i-1}^2(\theta_{i-1}) \prod_{j=i-1}^{n} d_j p_j\left(\theta_j\right) \tag{53}$$

$$r(\theta_{i-1}) = \frac{\partial\xi_{i+1}}{\partial\theta_i} f_i(\theta_{i-1}) \tag{54}$$

where $i \in \{1, ..., n\}$. This means that $\xi_i = \xi_i(\theta_{i-2})$.

Assuming that (52) is satisfied for $i = m$, Eqs. (32), (44), (53) and (54) imply

$$\dot{\xi}_m = \frac{\partial \xi_m}{\partial \theta_{m-2}} \cdot \dot{\theta}_{m-2} + \frac{\partial \xi_m}{\partial \theta_{m-1}} \cdot \dot{\theta}_{m-1}$$

$$= \frac{1}{cos^2(\theta_{m-2} - \theta_{m-1}) c_m(\theta_{m-1})}$$

$$= \frac{1}{d_{m-2}} \frac{tan(\theta_{m-3} - \theta_{m-2})}{p_{m-2}(\theta_{m-2})} \upsilon + \frac{\partial \xi_m}{\partial \theta_{m-1}} f_{m-1}(\theta_{m-2}) \upsilon$$

$$= \left(\frac{tan(\theta_{m-3} - \theta_{m-2})}{c_{m-1}(\theta_{m-2})} + r_{m-1}(\theta_{m-2}) \right) \upsilon$$

Note from (32) and (52) that

$$c_{m-1} = cos^2 \alpha_{m-2} c_m d_{m-2} p_{m-2}$$

$$= cos^2 \alpha_{m-2} p_{m-1}^2 \left(\prod_{j=m-1}^{n} d_j p_j \right) d_{m-2} p_{m-2}$$

$$= p_{m-2}^2(\theta_{m-2}) \prod_{j=m-2}^{n} d_j p_j(\theta_j)$$

where $\alpha_{m-2} = \theta_{m-2} - \theta_{m-1}$, $c_m = c_m(\theta_{m-1})$, and $p_j = p_j(\theta_j)$.

We have thus shown that if ξ_i is given by (51) for $i = m$, then

$$\dot{\xi}_m = \xi_{m-1} \upsilon$$

If ξ_{m-1} is chosen as in (51) with $i = m - 1$, it remains to show that if ξ_n is given by (50) with $i = n$ and ξ_{n-1} is given by (50), then

$$\dot{\xi}_{n+1} = \xi_n \upsilon$$

From (44), (51), (53) and (54), we get

$$\dot{\xi}_{n+1} = \frac{\partial \xi_{n+1}}{\partial \theta_{n-1}} \dot{\theta}_{n-2} + \frac{\partial \xi_{n+1}}{\partial \theta_n} \dot{\theta}_n$$

$$= \frac{1}{cos^2(\theta_{n-1} - \theta_n) d_n cos^3 \theta_n}$$

$$= \frac{1}{d_{n-1}} \cdot \frac{tan(\theta_{n-2} - \theta_{n-1})}{cos\theta_n cos(\theta_{n-1} - \theta_n)} \cdot \upsilon + \frac{\partial \xi_{n+1}}{\partial \theta_n} \cdot f_n(\theta_{n-1}) \cdot \upsilon$$

$$= \left(\frac{tan(\theta_{n-2} - \theta_{n-1})}{c_n(\theta_{n-1})} + r_n(\theta_{n-1}) \right) \cdot \upsilon$$

This means that

$$\dot{\xi}_{n+1} = \xi_n \upsilon$$

by choosing

$$\xi_n = tan(\theta_{n-2} - \theta_{n-1})/c_n(\theta_{n-1}) + r_n(\theta_{n-1}) \tag{55}$$

Therefore, ξ_i is given by (51) for all $i \in \{1, \nu, n\}$ and the transformation (33)–(43) (see Sect. 3) implies that

$$\dot{\xi}_i = \xi_{i-1}\upsilon, \ \ \forall i \in \{3, \nu, n+3\}$$

To complete the proof, we have shown that

$$\dot{\xi}_2 = \alpha_2, \ \ \dot{\xi}_1 = \alpha_1$$

ξ_2 is given by (51) with $i = 2$.

By differentiating (51), we obtain

$$\dot{\xi}_2 = \frac{\partial \xi_2}{\partial \theta_0}\dot{\theta}_1 + \frac{\partial \xi_2}{\partial \theta_1}\dot{\theta}_2$$

$$= \frac{1}{cos^2(\theta_1 - \theta_2)c_2(\theta_2)}\omega + \frac{\partial \xi_2}{\partial \theta_2} f_1(\theta_1)\upsilon$$

$$= \frac{1}{cos^2(\theta_1 - \theta_2)c_2(\theta_1)}\omega + r_1(\theta_1) f_1(\theta_1)\upsilon_1$$

since $\upsilon = p_1(\theta_1)\upsilon_1$. This implies that the transformation (43) implies

$$\dot{\xi}_2 = \alpha_2 \tag{56}$$

From (48), it follows directly that

$$\dot{\xi}_1 = \alpha_1 \tag{57}$$

Let us choose $\xi_1 = x$ and $\alpha_1 = \upsilon = p_1(\theta_1)\upsilon_1$.

Finally, one can conclude that the transformation (33)–(43) implies that

$$\dot{\xi}_1 = \alpha_1$$

$$\dot{\xi}_2 = \alpha_2$$

$$\dot{\xi}_1 = \xi_2 \alpha_1$$

$$\vdots$$

$$\dot{\xi}_n = \xi_n \alpha_1 \tag{58}$$

As a result, the model given by (27) can be built.

4.2 Illustration for Three-Linked Robots

In this case study, the kinematic model of three-linked 2WD mobile robots has been considered based on (18). This model has a triangular structure, that can be represented by a transformation into the chained form. The Eq. (18) is subsequently showed:

$$\dot{q} = \boldsymbol{g}_1(q) \cdot \upsilon_3 + \boldsymbol{g}_2(q) \cdot \omega_1 \tag{59}$$

where, $q = [x, y, \theta_3, \theta_2, \theta_1]^T$, and $\boldsymbol{g}_1(q)$ and $\boldsymbol{g}_2(q)$ are given by

$$\boldsymbol{g}_1(q) = \left[cos\theta_3, sin\theta_3, \frac{1}{d_2} tan(\theta_2 - \theta_3), \frac{1}{d_1} \frac{tan(\theta_1 - \theta_2)}{cos(\theta_2 - \theta_3)}, 0 \right]^T$$

$$\boldsymbol{g}_2(q) = [0, 0, 0, 0, 1]^T$$

where υ_3 and ω_1 represent, respectively, the linear velocity of robot 3 and rotational velocity of robot 1, and $\boldsymbol{g}_1(q)$, $\boldsymbol{g}_2(q)$ are smooth linearly independent vector fields.

Applying *Theorem* 1, the following change of coordinates is considered

$$\xi_1 = x \tag{60}$$

$$\begin{aligned} \xi_2 = {} & \frac{tan(\theta_1 - \theta_2)}{d_1 d_2 cos^4\theta_3 cos^3(\theta_2 - \theta_3)} + \frac{1}{d_2^2 cos^4\theta_3} \cdot tan(\theta_2 - \theta_3) \\ & \left[3tan\theta_3 tan\,(\theta_2 - \theta_3) \cdot sec^2(\theta_2 - \theta_3) \right] \end{aligned} \tag{61}$$

$$\xi_3 = \frac{tan(\theta_2 - \theta_3)}{d_2 cos^3\theta_3} \tag{62}$$

$$\xi_4 = tan\theta_3 \tag{63}$$

$$\xi_5 = y \tag{65}$$

together with the following input transformations

$$\alpha_1 = \upsilon_3 cos\theta_3 \tag{65}$$

$$\alpha_2 = \beta_1 \upsilon_3 + \beta_2 \omega_1 \tag{66}$$

where

$$\begin{aligned}\beta_1 =& \frac{1}{d_1 d_2} sec^4\theta_3 tan(\theta_1 - \theta_2) sec^4(\theta_2 - \theta_3) \\ &+ \frac{1}{d_2}(6tan\theta_3 \, sin(\theta_2 - \theta_3) - 2tan(\theta_2 - \theta_3) sin(\theta_2 - \theta_3) - sec(\theta_2 \quad - \theta_3))] \\ &+ \frac{1}{d_1 d_2} sec^4\theta_3 sec^2(\theta_1 - \theta_2) sec^3(\theta_2 - \theta_3)\end{aligned}$$

$$\beta_2 = \frac{1}{d_1 d_2} sec^4\theta_3 sec^2(\theta_1 - \theta_2) sec^3(\theta_2 - \theta_3)$$

This transformed form represents local diffeomorphism around the configurations, where the angle of the robot and the angles between each adjacent robot are different from $(\pi/2)$.

When applying the transformation to the system modelled in (59), we get the following system in chained form:

$$\begin{aligned}\dot{\xi}_1 &= \alpha_1, \\ \dot{\xi}_2 &= \alpha_2, \\ \dot{\xi}_3 &= \xi_2 \alpha_1, \\ \dot{\xi}_4 &= \xi_3 \alpha_1, \\ \dot{\xi}_5 &= \xi_4 \alpha_1\end{aligned} \tag{67}$$

4.3 The Fault Model of Actuator for Multi-linked 2WD Mobile Robots

The faults of actuators of mobile robot can be characterized by:

1. An extra friction can be introduced due to power loss of some actuated wheels,
2. A partial loss in the effectiveness of wheel motor;

The model of actuator faults of each motor is described by

$$\tau_j(t) = \sigma_j(t)u_j(t) + \overline{u}_j, \quad t \geq t_j, \tag{68}$$

for $j = 1r, 1l, ..., nr, nl$, $\overline{u}_j$ represents the unknown value of friction due to the fault, u_j denotes the signal of control input, $0 \leq \sigma_j \leq 1$ represents the effectiveness of unknown actuation effectiveness, $t_j > 0$ represents the unknown instant of fault occurrence. The time duration of each fault is assumed to be sufficiently long, and $\overline{u}_j$ and σ_j are piecewise constants.

One can remark that Eq. (68) is composed of two terms, one related to the direct effect of the faults, the second one caused by the resulting frictions.

The proposed fault model for the actuators (68) can represent many practical faulty scenarios. For example, $\sigma_j = 1$ and $\overline{u}_j = 0$ represents the no-fault situation; $\overline{u}_j = 0$ and $\sigma_j = 0$ indicates that there is a loss power in motor, but the motor can still have free rotational motion; $\overline{u}_j \neq 0$ and $\sigma_j = 0$ represents the extra friction, which is added by the motor between the ground and the wheel; $\overline{u}_j \neq 0$ represents the friction due to bearing; $0 < \sigma_j < 1$ represents the partial loss due to effectiveness faults which is produced either by reduction in busbar voltage, or by inter-turn faults. Also, this proposed fault model can represent two conventional fault components; one component $\overline{u}_j$ is an additive fault, and the other element σ_j is a multiplicative fault.

Considering that any actuators can be subjected to a fault, the input torque vector in Eq. (11) is expressed by

$$\tau(t) = \sigma(t) \cdot u(t) + \overline{u}(t) \tag{69}$$

where $u = [u_{1r}, u_{1l}, u_{2r}, u_{2l}, ..., u_{nr}, u_{nl}]^T$ is the control signal, $\tau = [\tau_{1r}, \tau_{1l}, \tau_{2r}, \tau_{2l}, ..., \tau_{nr}, \tau_{nl}]^T$ represents the torque signal, $\sigma = \text{diag}\{\sigma_{1r}, \sigma_{1l}, ..., \sigma_{nr}, \sigma_{nl}\}$ represents the uncertain effectiveness matrix, and $\overline{u} = [\overline{u}_{1r}, \overline{u}_{1l}, \overline{u}_{2r}, \overline{u}_{2l}, ..., \overline{u}_{nr}, \overline{u}_{nl}]^T$ represents the friction due to actuator faults.

Assumption 2 *The dynamic subsystem in* (11) *for n-linked 2WD mobile robots has two control inputs* (υ_n, ω_1) *to be controlled. Then, to design the fault compensation, it is required the actuation redundancy is required. This is represented by*

$$\boldsymbol{rank}\left(\overline{\boldsymbol{B}}\boldsymbol{\sigma}\right) = \mathbf{2} \tag{70}$$

υ_n and ω_1 represent, respectively, the linear velocity of robot n and rotational velocity of robot 1

Remark 1 *For the Two-linked 2WD mobile robots of case study, with $\overline{B}(q)$, the fault scenarios satisfying the redundancy condition in (70) are:*

1. Four non-faulty wheels,
2. One actuator is faulty, but the other three actuators are healthy,
3. Two actuators are faulty, but the other two actuators are healthy.

In the third fault scenario, the system is like a tractor-trailer, where both faults reside in robot 2 [20], and the faults may have no effect; if there is just one fault in each robot, then it can be compensated; but if there are two faults in robot 1, then the faults cannot be compensated, because in this case $rank\left(\overline{B}\sigma\right) = 1$ with $\sigma = \mathrm{diag}\{\sigma_{1r}, \sigma_{1l}, \sigma_{2r}, \sigma_{2l}\}$ and ω_1 is uncontrollable.

Key technical issues: The failure time t_j, the failure pattern σ_j, and the friction $\overline{u}_j$ are unknown, but the feedback controller must guarantee a good system performance.

4.4 Feasible Trajectories and Controllability

In this section, a feasible trajectory path is firstly generated for multi-linked 2WD mobile robots. Secondly, the tracking capability of the trajectory is guaranteed while ensuring the controllability of the system by imposing the conditions of system controllability and avoiding system singularity as characteristics of the trajectory planning.

The feasible desired trajectories are expressed in terms of position point P_3 (the middle point of last robot rear wheels), represented in Cartesian coordinates,

$$x_d = x_d(t),\ \ y_d = y_d(t),\ \ t \geq 0, \tag{71}$$

Consequently, the state-space variable q and the kinematic variable η can be written as a function of flat outputs and their derivatives [16]. A desired output trajectory (14) is feasible when it fulfils equations of kinematics model (4) with reference to n-linked 2WD mobile robots [17].

Let us consider one given desired trajectory. This trajectory is defined by:

$$\dot{x}_d = \mathrm{v}_{nd} \cdot cos\theta_{nd} \tag{72}$$

$$\dot{y}_d = \mathrm{v}_{nd} \cdot sin\theta_{nd} \tag{73}$$

$$\dot{\theta}_{nd} = (\mathrm{v}_{nd}/d_{n-1}) \cdot \tan\left(\theta_{(n-1)d} - \theta_{nd}\right) \tag{74}$$

$$\vdots$$

$$\dot{\theta}_{(n-1)d} = \mathrm{v}_{nd} \cdot \frac{tan\big(\theta_{(n-k-1)d} - \theta_{(n-k)d}\big)}{d_{(n-k-1)} \cdot \prod_{i=n-k}^{n-1} cos\big(\theta_{id} - \theta_{(i+1)d}\big)} \tag{75}$$

$$\vdots$$

$$\dot{\theta}_{2d} = \mathrm{v}_{nd} \cdot \frac{\tan(\theta_{1d} - \theta_{2d})}{d_1 \cdot \prod_{i=2}^{n-1} \cos\big(\theta_{id} - \theta_{(i+1)d}\big)} \tag{76}$$

$$\dot{\theta}_{1d} = \omega_{1d} \tag{77}$$

One can remark that trajectories are feasible when the difference between angles $\big(\theta_{(n-1)d} - \theta_{nd}\big)$ and $\big(\theta_{(n-2)d} - \theta_{(n-1)d}\big)$ are not equal to $(\pi/2)$, and the value of $v_{nd} \neq 0$.

From Eqs. (72) and (73), it can be concluded that:

$$\theta_{nd} = atan2\{\dot{y}_d, \dot{x}_d\} \tag{78}$$

Using Eq. (74), one can get:

$$\theta_{(n-1)d} = \theta_{nd} + \mathrm{atan}(d_{n-1}\dot{\theta}_{nd}/v_{nd}) \tag{79}$$

Using Eq. (75), one can get:

$$\theta_{(n-2)d} = \theta_{(n-1)d} + atan\Big(\big(d_{(n-2)}\dot{\theta}_{2d} cos\big(\theta_{(n-1)d} - \theta_{nd}\big)\big)/\mathrm{v}_{nd}\Big) \tag{80}$$

Also, from Eqs. (72) and (73), the following can be obtained:

$$v_{nd} = \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \tag{81}$$

Differentiating Eqs. (72) and (73), and rearranging, one obtains:

$$\dot{\theta}_{nd} = (\ddot{y}_d \cdot \dot{x}_d - \ddot{x}_d \cdot \dot{y}_d)/\mathrm{v}_{nd}^2 \tag{82}$$

Placing Eqs. (78) and (82) into Eq. (79), one gets:

$$\theta_{(n-2)d} = atan2\{\dot{y}_d, \dot{x}_d\} + a\tan\big(d_2(\ddot{y}_d\dot{x}_d - \ddot{x}_d\dot{y}_d)/v_{3d}^3\big) \tag{83}$$

Differentiating Eq. (83), one obtains:

$$\dot{\theta}_{(n-2)d} = \dot{\theta}_{nd} + \frac{d_{(n-2)}v_{nd}\left(\dddot{y}_d\dot{x}_d - \dddot{x}_d\dot{y}_d\right)v_{nd}^2 - 3(\ddot{y}_d\dot{x}_d - \ddot{x}_d\dot{y}_d)(\dot{x}_d\ddot{x}_d + \dot{y}_d\ddot{y}_d)}{v_{nd}^6 + d_{(n-2)}^2\left(\dddot{y}_d\dot{x}_d - \dddot{x}_d\dot{y}_d\right)^2} \tag{84}$$

Differentiating Eq. (80), one obtains:

$$\dot{\theta}_{(n-2)d} = \dot{\theta}_{(n-1)d} + \frac{d}{dt}\,atan\left(\left(d_{(n-2)}\dot{\theta}_{2d}cos\left(\theta_{(n-1)d} - \theta_{nd}\right)\right)/\mathrm{v}_{nd}\right) \tag{85}$$

Equations (78–85) provide the feasibility conditions that the states and the input trajectories needed to fulfil to reproduce the desired feasible trajectories.

Assumption 3 A reference trajectory x_d, y_d, θ_{nd}, $\theta_{(n-1)d}$, v, θ_{1d}, the reference velocities and their time derivatives are continuous and uniformly bounded. Also, the reference velocities may not converge to zero as $t \to \infty$.

Design issues: A block diagram of the proposed fault compensation scheme is given can be seen in Fig. 2. For the controller design, the following steps are necessary:

1. To facilitate the control design, a transformation of robot kinematic into chained form (canonical form) has to be performed. Generalized coordinates q_d, q are used and the model equation has to be transformed into chained form blocks. The outputs are the new coordinates ξ_d, ξ and the new inputs α_d, α.
2. A recursive approach using the popular backstepping method is used to formulate the virtual kinematic control law α_c.
3. The desired performance is guaranteed desired by appropriate design of kinematic control law η_c.
4. Based on kinematic control law η_c, the dynamic control law u is designed. This connection between kinematic control and dynamic control is rendered possible by putting the system into canonical form using the chained form transformation. This chained form is used to find a diffeomorphism which makes it possible to connect control laws.
5. Multiple dynamic controllers are designed, each one taking into account a specific failure pattern. If the fault considered in the design matches the current fault, then the control signal can guarantee that $\eta \to \eta_c$.

5 Design of Adaptive Fault-Tolerant Control

5.1 Design of Kinematic Control Law

In this part, the kinematic model of multi-linked 2-WD mobile robots, has been considered. Applying the algorithm proposed in [18], using the change of coordinates presented in Sect. 4, the kinematic model can be represented in chained form as

$$\dot{\xi}_1 = \alpha_1,$$

$$\dot{\xi}_2 = \alpha_2,$$

$$\dot{\xi}_3 = \xi_2\alpha_1,$$

$$\dot{\xi}_4 = \xi_3\alpha_1,$$

$$\dot{\xi}_5 = \xi_4\alpha_1 \tag{86}$$

where, α_1 and α_2 are the two control inputs, and $\xi = (\xi_1, \xi_2, \ldots, \xi_5)$ is the state vector.

A change of coordinates can be applied to the desired trajectory $\xi_d = (\xi_{1d}, \xi_{2d}, \ldots, \xi_{5d})$, providing

$$\dot{\xi}_{1d} = c\alpha_{1d},$$

$$\dot{\xi}_{2d} = \alpha_{2d},$$

$$\dot{\xi}_{3d} = \xi_{2d}\alpha_{1d},$$

$$\dot{\xi}_{4d} = \xi_{3d}\alpha_{1d},$$

$$\dot{\xi}_{5d} = \xi_{4d}\alpha_{1d} \tag{87}$$

where $\alpha_d = (\alpha_{1d}, \alpha_{2d})$ are the reference control inputs.

The tracking error is denoted as $\xi_e = \xi - \xi_d$. It can be established that the dynamics errors respect the following differential equations

$$\dot{\xi}_{1e} = \alpha_1 - \alpha_{1d}$$

$$\dot{\xi}_{2e} = \alpha_2 - \alpha_{2d}$$

$$\dot{\xi}_{3e} = \xi_{2e}\alpha_{1d} + \xi_2(\alpha_1 - \alpha_{1d})$$

$$\dot{\xi}_{4e} = \xi_{3e}\alpha_{1d} + \xi_3(\alpha_1 - \alpha_{1d})$$

$$\vdots$$

$$\dot{\xi}_{ne} = \xi_{(n-1)e}\alpha_{1d} + \xi_{(n-1)}(\alpha_1 - \alpha_{1d}) \tag{88}$$

Definition 1 The control design will be said to be semi globally solvable [54] *for* (*86*) *if, for a compact set $S \in R^n$ involving the origin, continuous time-varying state feedback controllers can be designed considering the form $\alpha_1 = \mu_1(\xi)$ and $\alpha_2 = \mu_2(\xi)$, so that for any initial tracking error $\xi_e(0) = \xi(0) - \xi_d(0)$ in S, all the signals of the closed loop system* (*88*) *α_1 and α_2 are uniformly bounded. Furthermore*

$$\lim_{t\to\infty} |\xi(t) - \xi_d(t)| = 0 \tag{89}$$

5.1.1 Kinematic Control Law

This subsection aims the design of the kinematic control law α_1 and α_2 that must ensure trajectory tracking. Firstly, an error dynamic is derived. Secondly, a change in coordinates is derived and the system (88) is organized in a triangular form suitable for the application of the integrator backstepping.

Let us denote $\tilde{\xi}_d = (\xi_{2d}, \nu, \xi_{4d})$ and let $\Phi_1\left(.; \tilde{\xi}_d\right)$: $R^5 \to R^5$ be the mapping defined by

$$z_i = \xi_{(n-i+1)e} - \left[\xi_{(n-i)e} + \xi_{(n-i)d}\right]\xi_{1e}, \, 1 \le i \le n-2$$

$$z_4 = \xi_{2e}$$

$$z_5 = \xi_{1e} \tag{90}$$

Note that $\Phi_1\left(.; \tilde{\xi}_d\right)$ is a global diffeomorphism for $\tilde{\xi}_d \in R^3$ [54] and its inverse $\Phi_1^{-1}\left(z; \tilde{\xi}_d\right)$ is given by

$$\xi_{1e} = z_5$$

$$\xi_{2e} = z_4$$

$$\xi_{3e} = z_3 + [z_4 + \xi_{2d}]z_5$$

$$\xi_{4e} = z_2 + [z_{3+}z_4 z_5 + \xi_{2d} z_5 + \xi_{3d}]z_5$$

$$\vdots$$

$$\xi_{ne} = z_1 + [z_2 + (z_{3+}z_{(n-2)}z_n + \xi_{(n-a)d}z_n + \xi_{(n-2)d)}z_n + \xi_{(n-1)d}]z_n \tag{91}$$

After a transformation, the dynamics of the tracking error can be written as

$$\begin{gathered}
\dot{z}_1 = \alpha_{1d}z_2 - \xi_{n-2}(\alpha_1 - \alpha_{1d})z_n \\
\vdots \\
\dot{z}_{n-3} = \alpha_{1d}z_{n-2} - \xi_2(\alpha_1 - \alpha_{1d})z_n \\
\dot{z}_{n-2} = \alpha_{1d}z_{n-1} - \alpha_2 z_n \\
\dot{z}_{n-1} = \alpha_2 - \alpha_{2d} \\
\dot{z}_5 = \alpha_1 - \alpha_{1d}
\end{gathered} \tag{92}$$

The signals α_1 and α_2 must provide stable error dynamics. The algorithm proposed in [34] is the basis for the work developed in the following.

Considering the system (92), the virtual control functions $\beta_i(1 \leq i \leq n-2)$ should be considered for the backstepping design.

Step (1): Let z_1 be a subsystem of (92)

$$\dot{z}_1 = \alpha_{1d}z_2 - \xi_{n-2}(\alpha_1 - \alpha_{1d})z_n \tag{93}$$

where, α_{1d} and z_n are time-varying functions and the virtual control input signal is assigned to variable z_2. Let $\bar{z}_1 = z_1$, the time derivative of

$$V_1(\bar{z}_1) = \frac{1}{2}\bar{z}_1^2 \tag{94}$$

Satisfies

$$\dot{V}_1(\bar{z}_1) = \alpha_{1d}\bar{z}_1 z_2 - \bar{z}_1\xi_{n-2}(\alpha_1 - \alpha_{1d})z_n \tag{95}$$

Choosing $\beta_1(z_1)$ as follows

$$\beta_1(z_1) = z_2 - \bar{z}_2 \tag{96}$$

with $\beta_1(z_1) = 0$ stabilizing (93) if $z_n = 0$.

Equation (95) becomes

$$\dot{V}_1(\bar{z}_1) = \alpha_{1d}\bar{z}_1\bar{z}_2 - \bar{z}_1\xi_{n-2}(\alpha_1 - \alpha_{1d})z_n \tag{97}$$

The approach derived in step 1 is recursively applied to the subsystem of system (92), where $i = 2, \ldots, n-2$.

***Step** (n − 1)*: Consider the $(z_1, ..., z_{n-1})$ subsystem of (92) and the function

$$V_{n-1}(\bar{z}_1, \cdots, \bar{z}_{n-1}) = V_{n-2}(\bar{z}_1, \cdots, \bar{z}_{n-2}) + \frac{1}{2}\bar{z}_{n-1}^2 \tag{98}$$

From (97), the derivative of (53) from the solutions of (98) must satisfy

$$\begin{gathered}\dot{V}_{n-1} = \bar{z}_{n-1}\left[\alpha_{1d}\bar{z}_{n-2} + \alpha_2 - \alpha_{2d} - \sum_{i=1}^{n-2}\frac{\partial\beta_{n-2}}{\partial z_i}\alpha_{1d}z_{i+1}\right] - \left(\bar{z}_{n-2} - \bar{z}_{n-1}\frac{\partial\beta_{n-2}}{\partial z_{n-2}}\right) \\ \alpha_2 z_n - \left[\sum_{i=1}^{n-3}\bar{z}_i\xi_{n-i-1} - \sum_{i=1}^{n-3}\sum_{k=1}^{i}\bar{z}_{i+1}\frac{\partial\beta_i}{\partial z_k}\xi_{n-k-1} - \sum_{i=1}^{n-3}\bar{z}_{n-1}\frac{\partial\beta_{n-2}}{\partial z_i}\xi_{n-i-1}\right] \\ (\alpha_1 - \alpha_{1d})z_n\end{gathered} \tag{99}$$

Applying the following control law

$$\alpha_2 = \alpha_{2d} - c_{n-1}\bar{z}_{n-1} - \alpha_{1d}\bar{z}_{n-2} + \sum_{i=1}^{n-2}\frac{\partial\beta_{n-2}}{\partial z_i}\alpha_{1d}z_{i+1} \tag{100}$$

where $c_{n-1} > 0$ are chosen to be constant, obtaining

$$\dot{V}_{n-1} = -c_{n-1}\bar{z}_{n-1}^2 - \left(\bar{z}_{n-2} - \bar{z}_{n-1}\frac{\partial\beta_{n-2}}{\partial z_{n-2}}\right)\alpha_2 z_n - \Delta_1(\alpha_1 - \alpha_{1d})z_n \tag{101}$$

where $\Delta_1\left(z_1, z_2, \nu, z_{n-1}, \tilde{\xi}_d\right)$ is a smooth function given by

$$\Delta_1\left(z_1, z_2, ..., z_{n-1}, \tilde{\xi}_d\right) = \sum_{i=1}^{n-3}[\bar{z}_i\xi_{n-i-1} - \sum_{k=1}^{i}\bar{z}_{i+1}\frac{\partial\beta_i}{\partial z_k}\xi_{n-k-1} - \bar{z}_{n-1}\frac{\partial\beta_{n-2}}{\partial z_i}\xi_{n-i-1}] \tag{102}$$

***Step** (n)*: the final goal is to design α_1 so that $z_1, \ldots, z_n$ asymptotically converge to zero. For this, a proper and positive-definite function V_n is selected as a candidate of the Lyapunov function for the system (92)

$$V_n(\overline{z}) = \frac{1}{2}\overline{z}_1^2 + \cdots + \frac{1}{2}\overline{z}_{n-1}^2 + \frac{\kappa}{2}z_n^2 \tag{103}$$

where $\kappa > 0$ is a design parameter. The differentiation of V_n from the solutions of (92) results in

$$\dot{V}_n = -c_{n-1}\overline{z}_{n-1}^2 + z_n\left[(\kappa - \Delta_1)(\alpha_1 - \alpha_{1d}) - \left(\overline{z}_{n-2} - \overline{z}_{n-1}\frac{\partial\beta_{n-2}}{\partial z_{n-2}}\right)\alpha_2\right] \tag{104}$$

By choosing the following control law

$$\alpha_1 = \alpha_{1d} + (\kappa - \Delta_1)^{-1}\alpha_2 \tag{105}$$

where $c_n > 0$ is a chosen constant, leading to

$$\dot{V}_n(\overline{z}) = -c_{n-1}\overline{z}_{n-1}^2 - c_n z_n^2 \tag{106}$$

Equation (106) shows that $\dot{V}_n$ is negative, providing stability of (92).

Performance analysis

Proposition 1 Assume that $\xi_{id}(2 \leq i \leq n-1), \alpha_d$, and $\dot{\alpha}_{1d}$ are bounded. Then, the problem can be semi globally solved for (86). A transformation of coordinates (90) and the application of the proposed design method to system (92), given any compact neighborhood S of the origin in R^n, it is possible to find a $\kappa > 0$ so that, all the solutions of the closed loop system (88), (100), and (105) are uniformly bounded if the initial conditions $\xi_e(0)$ are inside S. Furthermore, if zero-convergence of $\alpha_{1d}(t)$ does not occur, then $\lim_{t\to\infty}|\xi(t) - \xi_d(t)| = 0$.

The proof of proposition 1 is derived in [54].

As mentioned before, the objective is the design of the kinematic control laws α_1 and α_2 in (100) and (105) to achieve trajectory tracking control of the system considered in (92).

$$\alpha = T_\alpha \eta \tag{107}$$

Define a virtual kinematic control law

$$\alpha_c = T_\alpha \eta_c \tag{108}$$

and the velocity tracking error

$$\eta_e = \eta - \eta_c \tag{109}$$

We obtain

$$\alpha_e = \alpha - \alpha_c = T_\alpha \eta_e \tag{110}$$

Accordingly, one can design the following virtual kinematic control law α_c,

$$\alpha_{2c} = \alpha_{2d} - c_{n-1}\overline{z}_{n-1} - \alpha_{1d}\overline{z}_{n-2} + \sum_{i=1}^{n-2} \frac{\partial \beta_{n-2}}{\partial z_i}\alpha_{1d} z_{\mathrm{i+1}} \tag{111}$$

$$\alpha_{1c} = \alpha_{1d} + (\kappa - \Delta_1)^{-1}[-c_n z_n + \left(\overline{z}_{n-2} - \overline{z}_{n-1}\frac{\partial \beta_{n-2}}{\partial z_{n-2}}\right)\alpha_{2c}] \tag{112}$$

The kinematic control law is derived from Eq. (108)

$$\eta_c = T_\alpha^{-1}\alpha_c \tag{113}$$

Substituting (110), (111) and (112) into Eq. (106). Then, letting $f_\eta = [z_n(\kappa - \Delta_1), z_n\left(\overline{z}_{n-1}\frac{\partial \beta_{n-2}}{\partial z_{n-2}}\overline{z}_{n-2}\right)]$ results in

$$\dot{V}_n(\overline{z}) = -c_{n-1}\overline{z}_{n-1}^2 - c_n z_n^2 + f_\eta^T T_\alpha \eta_e \tag{114}$$

Note that, if $f_\eta^T T_\alpha \eta_e \rightarrow 0$, then $\dot{V}_n \leq 0$, and the system is stable. A dynamic controller will be designed to ensure the stability and also a desired performance.

5.2 *Dynamic Control Design*

5.2.1 Multi Design Integration

Substituting (68) in (11), the dynamic equations including the actuator faults are

$$\dot{\eta} = \overline{M}_1^{-1}\overline{M}_2\eta - \overline{M}_1^{-1}\overline{C} + \overline{M}_1^{-1}\overline{B}\sigma u + \overline{M}_1^{-1}\overline{B}\overline{u}_f \tag{115}$$

where, $\overline{u}_f$ is defined by $\overline{u}_f = (I_N - \sigma)\overline{u}$, and $u = [u_{1r}, u_{1l}, u_{2r}, u_{2l}, u_{3r}, u_{3l}]^T$ represents the vector of control signals, $\sigma_{(k)}, k = 1, 2, \ldots, N$ denotes the k th possible pattern fault in the faults pattern matrix which satisfies the redundancy expressed in (70), N denotes the total number of possible fault patterns.

A specific control signal $u_{(k)}$ can be designed for each $\sigma_{(k)}$ such that the system performance will be guaranteed if $\sigma = \sigma_{(k)}$. and $u = u_{(k)}$. In order to cover the set of all possible patterns, a nominal multi-design integration-based dynamic signal can be expressed by

$$u^* = \sum_{k=1}^{N} \chi_{(k)} u_{(k)} \tag{116}$$

where

$$\chi_{(k)} = \begin{cases} 1 \; if \;\; \sigma = \sigma_{(k)} \\ 0 \;\; otherwise \end{cases} \tag{117}$$

with $\chi_{(k)}$ as 1 or 0, and thus Eq. (116) can be rewritten as

$$u^* = \sum_{k=1}^{N} diag\{\chi_{(k)1r}, \chi_{(k)1l}, \cdots, \chi_{(k)nr}, \chi_{(k)nl}\} u_{(k)} \tag{118}$$

where,

$$\chi_{(k)1r} = \chi_{(k)1l} = \chi_{(k)2r} = \chi_{(k)2l} \cdots = \chi_{(k)nr} = \chi_{(k)nl}$$

5.2.2 Design of Dynamic Control Law

For each index $k = 1, 2, \ldots, N$, $\chi_{(k)}$ is uncertain if the current failure pattern σ is uncertain. In order to cope with this uncertainty, the following dynamic control law is proposed:

$$u^* = \sum_{k=1}^{N} \hat{\chi}_{(k)} \cdot u_{(k)} \tag{119}$$

where

$$\hat{\chi}_{(k)} = \text{diag}\left\{\hat{\chi}_{(k)1r}, \hat{\chi}_{(k)1l}, ..., \hat{\chi}_{(k)nr}, \hat{\chi}_{(k)nl}\right\}$$

is the estimate matrix.

The derivative of the error in (109) is

$$\dot{\eta}_e = \dot{\eta} - \dot{\eta}_c \tag{120}$$

Substituting (115) into (120), the following equation is found

$$\dot{\eta}_e = -\overline{M}_1^{-1}\overline{M}_2\eta - \overline{M}_1^{-1}\overline{C} + \overline{M}_1^{-1}\overline{B}\sigma u + \overline{M}_1^{-1}\overline{B}\overline{u}_f - \dot{\eta}_c \tag{121}$$

If $\hat{\overline{u}}_f$ represents the estimate of $\overline{u}_f$, then, the dynamic control signal $u_{(k)}$ can be proposed as

$$u_{(k)} = \left(\overline{M}_1^{-1}\overline{B}\sigma_{(k)}\right)^{+}\left[-c_{n+1}\eta_e - T_\alpha^T f_\eta + \overline{M}_1^{-1}\overline{M}_2\eta + \overline{M}_1^{-1}\overline{C} - \overline{M}_1^{-1}\overline{B}\widehat{\overline{u}}_f + \dot{\eta}_c\right] \tag{122}$$

for $\sigma_{(k)}$, $k = 1, 2, \ldots, N$, where $c_{n+1} > 0$ is selected as a constant, and $\left(\overline{M}_1^{-1}\overline{B}\sigma_{(k)}\right)^{+}$ is a generalized inverse matrix satisfying $\overline{M}_1^{-1}\overline{B}\sigma_{(k)}\left(\overline{M}_1^{-1}\overline{B}\sigma_{(k)}\right)^{+} = I_N$. For $k = 1, 2, \ldots, N$, one can define the estimation errors as $\widetilde{\overline{u}}_f = \overline{u}_f - \widehat{\overline{u}}_f$,

$$\tilde{\chi}_{(k)1r} = \chi_{(k)} - \hat{\chi}_{(k)1r}, \tilde{\chi}_{(k)1l} = \chi_{(k)} - \hat{\chi}_{(k)1l},$$

$$\tilde{\chi}_{(k)2r} = \chi_{(k)} - \hat{\chi}_{(k)2r}, \tilde{\chi}_{(k)2l} = \chi_{(k)} - \hat{\chi}_{(k)2l},$$

$$\vdots$$

$$\tilde{\chi}_{(k)nr} = \chi_{(k)} - \hat{\chi}_{(k)nr}, \tilde{\chi}_{(k)nl} = \chi_{(k)} - \hat{\chi}_{(k)nl} \tag{123}$$

Then, substituting (116), (118) and (77) into (121), results in

$$\dot{\eta}_e = -\overline{M}_1^{-1}\overline{M}_2\eta - \overline{M}_1^{-1}\overline{C} + \overline{M}_1^{-1}\overline{B}\sigma u^* + \overline{M}_1^{-1}\overline{B}\overline{u}_f - \dot{\eta}_c + \overline{M}_1^{-1}\overline{B}\sigma\left(u - u^*\right)$$

$$\begin{aligned}\dot{\eta}_e = {} & -c_6 \cdot \eta_e - T_\alpha^T \cdot f_\eta + \overline{M}_1^{-1} \cdot \overline{B} \cdot \tilde{\overline{u}}_f - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{c1} \cdot \sigma_{1r} \cdot \tilde{\chi}_{(k)1r} \cdot u_{(k)1r} \\ & - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{c2}\sigma_{1l} \cdot \tilde{\chi}_{(k)1l} \cdot u_{(k)1l} - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{c3} \cdot \sigma_{2r} \cdot \tilde{\chi}_{(k)2r} \cdot u_{(k)2r} \\ & - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{c4} \cdot \sigma_{2l} \cdot \tilde{\chi}_{(k)2l} \cdot u_{(k)2l} \\ & \vdots \\ & - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{c(n-1)}\sigma_{nr}\tilde{\chi}_{(k)nr}u_{(k)nr} - \sum_{k=1}^{N}\left(\overline{M}_1^{-1}\overline{B}\right)_{cn}\sigma_{nl}\tilde{\chi}_{(k)nl}u_{(k)nl}\end{aligned} \tag{124}$$

where, $\left(\overline{M}_1^{-1} \cdot \overline{B}\right)_{ci}$ $(i = 1, \ldots, N)$ denotes the i th column vector of matrix $\overline{M}_1^{-1} \cdot \overline{B}$.

5.2.3 Adaptive Law

According to the error equation in (124) and (122), the following adaptive law can be deduced

$$\tilde{\bar{u}}_f = \left(\overline{M}_1^{-1}\overline{B}\right)^T \eta_e + \Gamma_f \tag{125}$$

where, the positive-definite matrix $\Gamma_f = \Gamma_f^T \in R^{n\times n}$ has constant real values.

In order to develop the dynamic control law u in (124), one can deduce the following adaptive laws $\hat{\chi}_{(k)1r}$, $\hat{\chi}_{(k)1l}$, ν, $\hat{\chi}_{(k)nr}$ and $\hat{\chi}_{(k)nl}$:

$$\dot{\hat{\chi}}_{(k)1r} = f_{k1r} - \gamma_{k1r} u_{(k)1r} \eta_e^T \left(\overline{M}_1^{-1}\overline{B}\right)_{c1}$$

$$\dot{\hat{\chi}}_{(k)1l} = f_{k1l} - \gamma_{k1l} u_{(k)1l} \eta_e^T \left(\overline{M}_1^{-1}\overline{B}\right)_{c2}$$

$$\dot{\hat{\chi}}_{(k)2r} = f_{k2r} - \gamma_{k2r} u_{(k)2r} \eta_e^T \left(\overline{M}^{-1}\overline{B}\right)_{c3}$$

$$\dot{\hat{\chi}}_{(k)2l} = f_{k2l} - \gamma_{k2l} u_{(k)2l} \eta_e^T \left(\overline{M}_1^{-1}\overline{B}\right)_{c4}$$

$$\vdots$$

$$\dot{\hat{\chi}}_{(k)nr} = f_{knr} - \gamma_{knr} u_{(k)nr} \eta_e^T \left(\overline{M}_1^{-1}\overline{B}\right)_{c(n-1)}$$

$$\dot{\hat{\chi}}_{(k)nl} = f_{knl} - \gamma_{k3l} u_{(k)nl} \eta_e^T \left(\overline{M}_1^{-1}\overline{B}\right)_{cn} \tag{126}$$

where, $\gamma_{k1r} > 0$, $\gamma_{k1l} > 0$, $\gamma_{k2r} > 0$, $\gamma_{k2r} > 0$, $\ldots$ $\gamma_{knr} > 0$, and $\gamma_{knl} > 0$, denotes the adaptation gains for $k = 1, 2, ..., N$.

Standard parameter projection functions [70] $f_{k1r,} f_{k1l,} f_{k2r,} f_{k2l,} \ldots f_{knr}$ and f_{knl} are set to ensure the boundedness of the parameters errors $\tilde{\chi}_{(k)1r}$, $\tilde{\chi}_{(k)1l}$, $\tilde{\chi}_{(k)2r}$, $\tilde{\chi}_{(k)2l}$, $\ldots$ $\tilde{\chi}_{(k)nr}$, $\tilde{\chi}_{(k)nl}$.

Let us define

$$p_{k1r} = -\gamma_{k1r} \cdot u_{(k)1r} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c1,}$$

$$p_{k1l} = -\gamma_{k1l} \cdot u_{(k)1l} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c2,}$$

$$p_{k2r} = -\gamma_{k2r} \cdot u_{(k)2r} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c3,}$$

$$p_{k2l} = -\gamma_{k2l} \cdot u_{(k)2l} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c4,}$$

$$\vdots$$

$$p_{knr} = -\gamma_{knr} \cdot u_{(k)nr} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c5,}$$

$$p_{knl} = -\gamma_{knl} \cdot u_{(k)nl} \cdot \eta_e^T \cdot \left(\overline{M}_1^{-1}\overline{B}\right)_{c6}, \tag{127}$$

The following projection functions are used to bound the adaptive gains

$$f_{kj} = \begin{cases} & \text{if } \hat{\chi}_{(k)j} \in (0, 1), \text{ or} \\ 0, & \text{if } \hat{\chi}_{(k)j} = 0 \text{ and } p_{kj} \geq 0, \text{ or} \\ & \text{if } \hat{\chi}_{(k)j} = 1 \text{ and } p_{kj} \leq 0, \\ -p_{kj}, & \text{otherwise} \end{cases} \tag{128}$$

Lemma 1 Based on the adaptive laws in (126) together with projection functions in (128) for $k = 1, 2, ..., N$ to ensure.

1. For $k = 1, 2, ..., N$, $\tilde{\chi}_{(k)1r} f_{k1r} \geq 0$, $\tilde{\chi}_{(k)1l} f_{k1l} \geq 0$, $\tilde{\chi}_{(k)2r} f_{k2r} \geq 0$, $\tilde{\chi}_{(k)2l} f_{k2l} \geq 0$, ν, $\tilde{\chi}_{(k)nr} f_{knr} \geq 0$, and $\tilde{\chi}_{(k)nl} f_{knl,} \geq 0$.
2. $\hat{\chi}_{(k)1r}, \hat{\chi}_{(k)1l}, ..., \hat{\chi}_{(k)nr}, \hat{\chi}_{(k)nl} \in [0, 1]$.

Proof *The same proofs can be followed for the 2n adaptive laws. One may take* $\hat{\chi}_{(\mathrm{k})1\mathrm{r}}$ *as an example.*

Firstly, the initial value of estimate can be set as $\hat{\chi}_{(k)1r}(0) \in (0, 1)$. With $j = 1r$, the projection functions in (128) will ensure $\hat{\chi}_{(k)1r}(t) \in (0, 1)$,

$$\left(\chi_{(k)1r} - \hat{\chi}_{(k)1r}\right) \cdot f_{k1r} = \tilde{\chi}_{(k)1r} \cdot_{k1r} \geq 0 \tag{129}$$

The following three cases can be analyzed

1. If $\hat{\chi}_{(k)1r} = 1$ and $p_{k1r} > 0$, then $\chi_{(k)1r} - \hat{\chi}_{(k)1r} \leq 0$ and $f_{k1r} = -p_{k1r} < 0$, which also means (128) is ensured,
2. If $\hat{\chi}_{(k)1r} = 0$ and $p_{k1r} < 0$, then $\chi_{(k)1r} - \hat{\chi}_{(k)1r} \geq 0$ and $f_{k1r} = -p_{k1r} > 0$, which means (128) is ensured,
3. Furthermore, with $f_{k1r} = 0$, then (128) is ensured.

Similarly, the same properties can also be obtained for the other five adaptive laws. The proof is completed.

5.3 Stability Analysis

In this section, an analysis of the system's stability and performance is presented.

Theorem 2 For *n-linked mobile robots subject to actuator failure modelled as* (69)-(70), *the proposed control scheme consists of the following elements: the kinematic control law in* (113), *dynamic control law in* (119) *based on multiple control signals in* (122) *and the updated projection adaptive laws, which are described in* (125) *and* (126).

Proof According to Remark 2, there are $(2n - 2)$ compensable cases: no fault, and cases where there is one faulty actuator and up to four faulty actuators simultaneously. Therefore, the system is given as follows.

I. **No-fault Condition**: the proposed Lyapunov candidate function is

$$\begin{aligned} V_{n+1} = V_n + \frac{1}{2}\eta_e^T \eta_e + \frac{1}{2}\tilde{\tilde{u}}_f^T \Gamma_f^{-1} \tilde{\tilde{u}}_f + \frac{1}{2}\Bigg[\sum_{k=1}^{N} \sigma_{1r}\gamma_{k1r}^{-1}\tilde{\chi}_{(k)1r}^2 + \sum_{k=1}^{N} \sigma_{1l}\gamma_{k1l}^{-1}\tilde{\chi}_{(k)1l}^2 \\ + \sum_{k=1}^{N} \sigma_{2r}\gamma_{k2r}^{-1}\tilde{\chi}_{(k)2r}^2 \sum_{k=1}^{N} \sigma_{2l}\gamma_{k2l}^{-1}\tilde{\chi}_{(k)2l}^2 \\ \vdots \\ + \sum_{k=1}^{N} \sigma_{nr}\gamma_{knr}^{-1}\tilde{\chi}_{(k)nr}^2 + \sum_{k=1}^{N} \sigma_{nl}\gamma_{knl}^{-1}\tilde{\chi}_{(k)nl}^2\Bigg] \end{aligned} \tag{130}$$

Then, the derivative of V_{n+1} is

$$\begin{aligned} \dot{V}_{n+1} =& \dot{V}_n + \tilde{\tilde{u}}_f^T \cdot \Gamma_f^{-1} \cdot \dot{\tilde{\tilde{u}}}_f + \eta_e^T \cdot \dot{\eta}_e + \Bigg[\sum_{k=1}^{N} \gamma_{k1r}^{-1} \cdot \tilde{\chi}_{(k)1r} \cdot \dot{\tilde{\chi}}_{(k)1r} + \sum_{k=1}^{N} \gamma_{k1l}^{-1} \cdot \tilde{\chi}_{(k)1l} \cdot \\ & \dot{\tilde{\chi}}_{(k)1l} + \sum_{k=1}^{N} \gamma_{k2r}^{-1}\tilde{\chi}_{(k)2r}\dot{\tilde{\chi}}_{(k)2r} + \sum_{k=1}^{N} \gamma_{k2l}^{-1}\tilde{\chi}_{(k)2l}\dot{\tilde{\chi}}_{(k)2l} \\ & \vdots \\ & + \sum_{k=1}^{N} \gamma_{knr}^{-1} \cdot \tilde{\chi}_{(k)nr} \cdot \dot{\tilde{\chi}}_{(k)nr} + \sum_{k=1}^{N} \gamma_{knl}^{-1} \cdot \tilde{\chi}_{(k)nl} \cdot \dot{\tilde{\chi}}_{(k)nl}\Bigg] \end{aligned} \tag{131}$$

Substituting (114) and (120)–(126) into (131), and using $\dot{\tilde{u}}_f = -\dot{\hat{u}}$, $\dot{\tilde{\chi}}_{(k)1r} = -\dot{\hat{\chi}}_{(k)1r}$, $\dot{\tilde{\chi}}_{(k)1l} = -\dot{\hat{\chi}}_{(k)1l}$, $\dot{\tilde{\chi}}_{(k)2r} = -\dot{\hat{\chi}}_{(k)2r}$, $\dot{\tilde{\chi}}_{(k)2l} = -\dot{\hat{\chi}}_{(k)2l}$, ν, $\dot{\tilde{\chi}}_{(k)nr} = -\dot{\hat{\chi}}_{(k)nr}$, and $\dot{\tilde{\chi}}_{(k)nl} = -\dot{\hat{\chi}}_{(k)nl}$, to have

$$\begin{aligned}\dot{V}_{n+1} =& c_{n-2}\bar{z}_{n-2}^2 - c_n\eta_e^T\eta_e - c_{n-1}z_{n-1}^2 - \sum_{k=1}^{N}\gamma_{k1r}^{-1}\tilde{\chi}_{(k)1r}f_{k1r} - \sum_{k=1}^{N}\gamma_{k1l}^{-1}\tilde{\chi}_{(k)1l}f_{k1l}\\ &- \sum_{k=1}^{N}\gamma_{k2r}^{-1}\tilde{\chi}_{(k)2r}f_{k2r} - \sum_{k=1}^{N}\gamma_{k2l}^{-1}\tilde{\chi}_{(k)2l}f_{k2l}\\ &\vdots\\ &- \sum_{k=1}^{N}\gamma_{knr}^{-1}\tilde{\chi}_{(k)nr}f_{knr} - \sum_{k=1}^{N}\gamma_{knl}^{-1}\tilde{\chi}_{(k)nl}f_{knl}\Bigg]\end{aligned} \tag{132}$$

If this is combined with second property of Lemma 1, one can have

$$\dot{V}_{n+1} \leq -c_{n-1}\bar{z}_{n-1}^2 - c_n z_n^2 - c_{n+1}\eta_e^T\eta_e \leq 0 \tag{133}$$

Equation (133) indicates that:$\eta_e \in L^{\infty}$, z_1, z_2, z_3, ..., z_n, $\bar{z}_1$, $\bar{z}_2$, $\ldots$, $\bar{z}_{n-1}$, and $\bar{z}_4$, z_5, $\eta_e \in L^2$ and all estimates are bounded. One can find the definitions of L^{∞} and L^2 in [70–72].

Then, based on Eqs. (28), (31), and also from (35)–(41) and (45)–(52), then using Barbalat Lemma, one can conclude that the boundness of all closed loop system signals are guaranteed, and $\lim_{t\to\infty}\bar{z}_{n-1} = 0$, $\lim_{t\to\infty}z_n = 0$ and $\lim_{t\to\infty}\eta_e = 0$, which also implies $\lim_{t\to\infty}\alpha_e = 0$, with Eq. (40). and also, we have $\lim_{t\to\infty}z_1 = 0$, ν, $\lim_{t\to\infty}z_n = 0$, $\lim_{t\to\infty}\bar{z}_1 = 0$, ν, $\lim_{t\to\infty}\bar{z}_{n-1} = 0$, and $|\alpha_{1d}| \geq 0$ [73–76].

Then, one can further deduce $\lim_{t\to\infty}\alpha_{2c} = \alpha_{2d}$ based on Eq. (37), this leads to $\lim_{t\to\infty}\alpha_2 = \alpha_{2d}$, and also $\lim_{t\to\infty}\alpha_{1c} = \alpha_{1d}$. Consequently, according to Eq. (38), it follows that $\lim_{t\to\infty}\alpha_1 = \alpha_{1d}$.

The signals of closed loop system are uniformly bounded, furthermore $\lim_{t\to\infty}z_i(t) = 0, (i = 1, \ldots n)$ and $\lim_{t\to\infty}\xi_{ie}(t) = 0, (i = 1, \ldots n)$ $\lim_{t\to\infty}(\eta(t) - \eta_c(t)) = 0$, which also means $\lim_{t\to\infty}(q(t) - q_d(t)) = 0$.

II. **Failure of one actuator***:* If the left actuator of robot 1 id faulty; that is, $\sigma = \mathit{diag}\{1, 0, ..., 1, 1\}$ with $\sigma_{(1l)} = 0$. In this case, the Lyapunov function candidate is considered as

$$V_{n+1} = V_n + \frac{1}{2}\eta_e^T\eta_e + \frac{1}{2}\tilde{u}_f^T\Gamma_f^{-1}\tilde{u}_f + \frac{1}{2}\Bigg[\sum_{k=1}^{N}\sigma_{1r}\gamma_{k1r}^{-1}\tilde{\chi}_{(k)1r}^2 + \sum_{k=1}^{N}\sigma_{1l}\gamma_{k1l}^{-1}\tilde{\chi}_{(k)1l}^2$$

$$+\sum_{k=1}^{N}\sigma_{2r}\gamma_{k2r}^{-1}\tilde{\chi}_{(k)2r}^{2}\sum_{k=1}^{N}\sigma_{2l}\gamma_{k2l}^{-1}\tilde{\chi}_{(k)2l}^{2}$$

$$\vdots$$

$$\left.+\sum_{k=1}^{N}\sigma_{nr}\gamma_{knr}^{-1}\tilde{\chi}_{(k)nr}^{2}+\sum_{k=1}^{N}\sigma_{nl}\gamma_{knl}^{-1}\tilde{\chi}_{(k)nl}^{2}\right] \tag{134}$$

Then, according to Lemma 1 and Eqs. (44), (54)–(56), one can obtain,

$$\dot{V}_{n+1} \leq -c_{n-1}\bar{z}_{n-1}^{2} - c_{n}z_{n}^{2} - c_{n+1}\eta_{e}^{T}\eta_{e} \leq 0 \tag{135}$$

It is worthy to note that, there is no $\tilde{\chi}_{(k)1l}$ for $k = 1, 2, \ldots, N$, in (64). However, they are bounded due to the first property of Lemma 1. Then, from (135), it can be summed up that system stability and asymptotic trajectory tracking are established when compared to the first case (fault free case). The performance analysis of the system for other cases of one faulty actuator is similar.

III. **Failure in** $(2n-2)$ **actuators***:* If a fault has occurred in the left wheel of robot 1, the right wheel of robot 2, and the right and left wheel of robot 3. In this case, the vector matrix σ is defined as $\sigma = \boldsymbol{diag}\{1, 0, 0, 1, ..., 0, 0\}$ with $\sigma_{(1l)} = 0$, $\sigma_{(2r)} = 0$, $\sigma_{(3r)} = 0$, $\sigma_{(3l)} = 0$. Then, the Lyapunov function candidate will be considered as

$$\begin{aligned}V_{n+1} =&V_{n} + \frac{1}{2}\tilde{\bar{u}}_{f}^{T}\Gamma_{f}^{-1}\tilde{\bar{u}}_{f} + \frac{1}{2}\eta_{e}^{T}\eta_{e} + \frac{1}{2}\left[\sum_{k=1}^{N}\sigma_{1l}\gamma_{k1l}^{-1}\tilde{\chi}_{(k)1l}^{2} + \sum_{k=1}^{N}\sigma_{1r}\gamma_{k1r}^{-1}\tilde{\chi}_{(k)1r}^{2}\right. \\ &+ \frac{1}{2}\left[\sum_{k=1}^{N}\sigma_{2l}\gamma_{k2l}^{-1}\tilde{\chi}_{(k)2l}^{2} + \sum_{k=1}^{N}\sigma_{1r}\gamma_{k1r}^{-1}\tilde{\chi}_{(k)1r}^{2}\right]\end{aligned} \tag{136}$$

Then, according to (44), (54)–(56) and Lemma 1,

$$\dot{V}_{n+1} \leq -c_{n-1}\bar{z}_{4}^{2} - c_{n}z_{5}^{2} - c_{n+1}\eta_{e}^{T}\eta_{e} \leq 0 \tag{137}$$

From Eq. (137), it can be concluded that the system is stable in closed-loop and the trajectory can be asymptotically tracked.

6 Simulations

For the simulations, it is considered a system with three-physically linked 2WD mobile robots, with the parameters of the mobile robots as described in [12]: $b_1 = b_2 = b_3 = 0.75m$, $r_1 = r_2 = r_3 = 0.15m$, $a_1 = a_2 = a_3 = 0.3m, m_1 = m_2 = m_3 = 30\,\text{kg}$, $I_{m1} = I_{m2} = I_{m3} = 15.625\,\text{kg}/m^2$.

The link between two robots has length $d = 1m$. The reference trajectory is generated by (78)–(85) with $x_d(0) = y_d(0) = \theta_{3d}(0) = \theta_{2d}(0) = \theta_{1d}(0) = 0$.

The fault conditions that are selected to check the effectiveness of the proposed fault compensation scheme are:

- No fault: $\sigma_{(1)} = \boldsymbol{diag}\{1, 1, 1, 1, 1, 1\}$, $0 \leq t < 20s$
- τ_{1r}, fails: $\tau_{1r} = 0$, $\sigma_{(2)} = \boldsymbol{diag}\{0, 1, 1, 1, 1, 1\}20s \leq t < 40s$
- τ_{1r},τ_{3l}, fail: $\tau_{1r} = 0$, $\tau_{3l} = -2Nm$, $\sigma_{(3)} = \boldsymbol{diag}\{0, 1, 1, 1, 1, 0\}$, $40s \leq t < 60s$
- τ_{3l}, fails:$\tau_{3l} = -2Nm$, $\sigma_{(4)} = \boldsymbol{diag}\{1, 1, 1, 1, 1, 0\}$, $60s \leq t < 80s$
- τ_{3r}, τ_{3l}, fail: $\tau_{3r} = -1Nm$, $\tau_{3l} = -2Nm$, $\sigma_{(5)} = \boldsymbol{diag}\{1, 1, 1, 1, 0, 0\}$, $t \geq 80s$

Five failure patterns have been taken ino account to satisfy the redundancy condition given by (13). The failure patterns could cope with the fault conditions: non-faulty; a fault in one actuator; a fault in one actuator of robot 1 and one actuator of robot 3; a fault in one actuator of robot 3; faults in both actuators of robots 3.

The initial conditions are: $x(0) = y(0) = 0.5, \theta_3(0) = \theta_2(0) = \theta_1(0) = 30\circ$.

The adaptation gains are set as: $\Gamma_f = I_6$, and $\gamma_{k1r} = \gamma_{k1l} = \gamma_{k2r} = \gamma_{k2l} = \gamma_{k3r} = \gamma_{k3l} = 1$, for $k = 1, 2, 3, 4, 5$. The control gains are set as: $C_4 = C_5 = C_6 = 2$ and $\kappa = 5$.

Figure 3 presents the positions of the three robots and the reference trajectory. Figure 4 presents the actuating control torques generated by wheels of the three robots, showing that the fault in the actuators are compatible with the fault conditions considered for the simulation.

Figure 5 shows the tracking errors of the states. Figure 6 shows the orientation error between two robots. It can be seen that the stability and the asymptotic trajectory tracking are guaranteed, in spite of the faults in the actuators.

Algorithmic Steps for Adaptive FTC Design of Robotic System

This section summarizes the steps to design the passive scheme of FTC for n-linked 2WD mobile robots ($n > 1$).

1. A transformation of coordinates using the chained form is applied to kinematic the model of multi-linked 2WD mobile robots. This consists in transforming the nonlinear model into an equivalent linear one through a change of variables and feedback transformation. The chained form for n-linked robots was presented in Sect. 4, page 8 (Theorem 1). Its application for three-linked robots was shown in Sect. 4, page 13.
2. Establishing recursive algorithm to represent kinematic control design for n ($n > 1$) linked robots to ensure the system's states asymptotical convergence to the reference trajectories [65]. This algorithm is presented in Sect. 7, page 18,

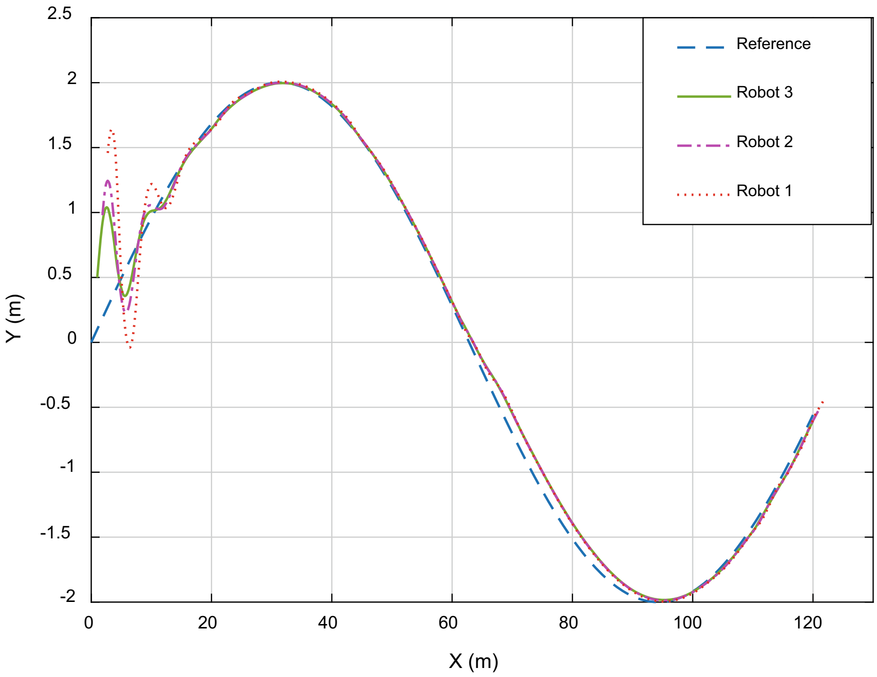


Fig. 3 Robot trajectories in (x, y) plane

for three-linked robots. The same methodology may be applied easily for any n-linked robots.

3. Finally, general dynamic control laws including actuator faults are designed, each one designed for a specific fault pattern. If the actual fault is compatible with a pattern used in the control design, then the related control signal may guarantee the desired performance. Considering different and possible fault patterns, a multi-design integration-based adaptive dynamic controller has been designed to solve the control problems due to these faults.

7 Conclusions

This chapter presented a trajectory tracking control method of multi-linked 2WD mobile robots subjected to actuator faults. The proposed adaptive FTC strategies are based on common theoretical tools related to WMR kinematic and dynamic models, and also nonlinear adaptive control. The theoretical developments were validated in a simulation using three-linked 2WD robots.

One of the contributions concerns the development of a general control strategy for multi-linked 2WD mobile robots to compensate for actuator faults, with a generalization to the configuration of n $(n > 1)$ physically-linked 2WD mobile robots.

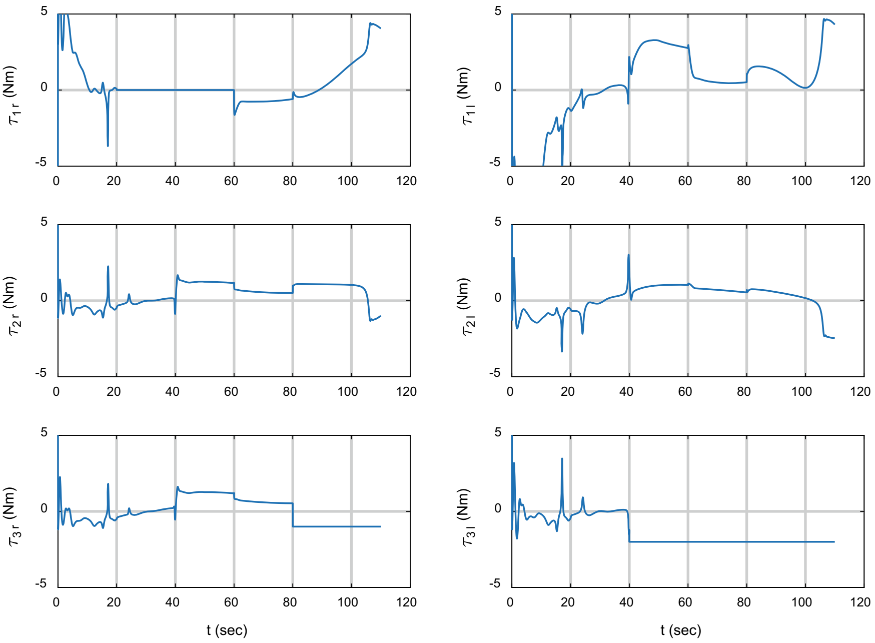


Fig. 4 Control torque inputs produced by the three robots

Kinematic and dynamic models applied to multi-linked 2WD mobile robots were proposed, and a generalization of the FTC compensation algorithm for mobile robotic system has been established. Using a chained form representation ensure that the system's states convergence to the reference trajectories is guaranteed. Furthermore, a general dynamic control law including actuator faults was designed considering an unknown failure pattern matrix. Thus, a multi-design integration-based adaptive control algorithm subjected to failures of actuators faults has been developed to avoid the control switching mechanism and also to conveniently deal with the friction.

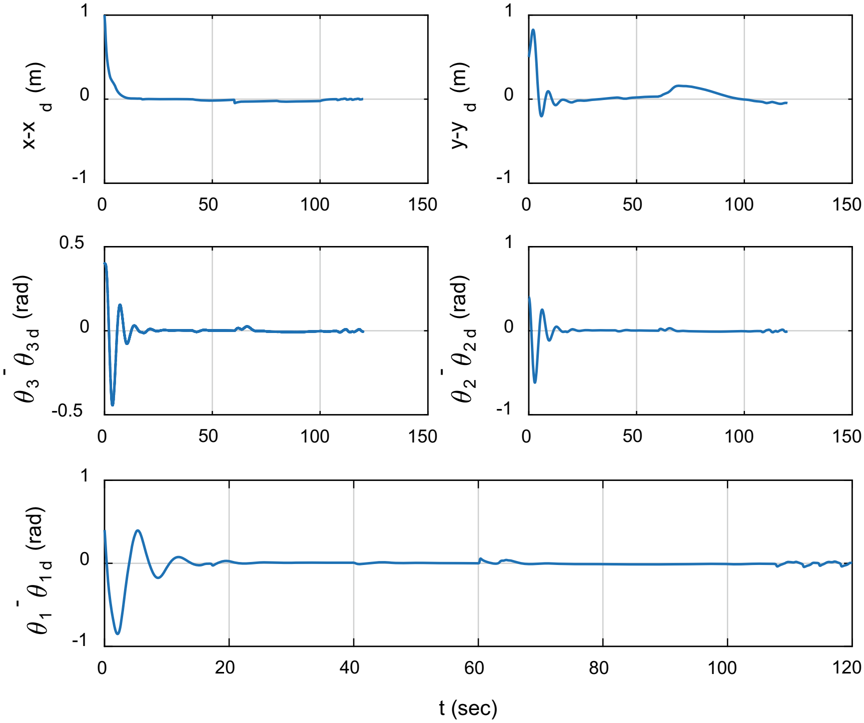


Fig. 5 Tracking errors

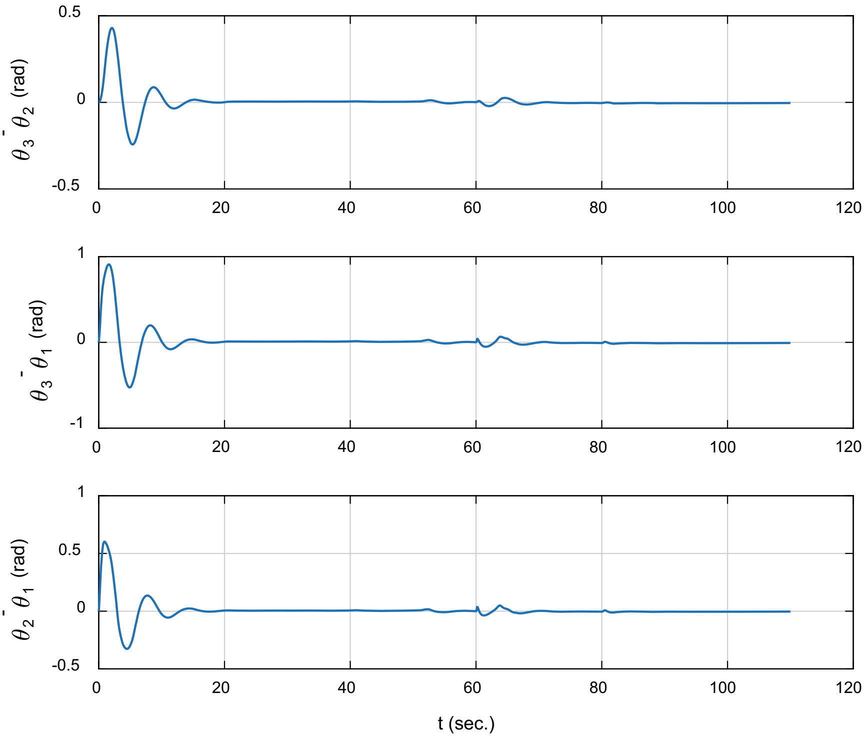


Fig. 6 Orientation errors between the two robots

8. Appendix

8.1. Input Transformation Matrix B(q)

The directions of the n-robots in the forward motion can be expressed by the following unit vectors

$$U_1 = [cos\theta_1, sin\theta_1]^T, U_2 = [cos\theta_2, sin\theta_2]^T, ..., U_n = [cos\theta_n, sin\theta_n]^T \tag{138}$$

where $cos\theta_i$ and $sin\theta_1$ $(i = 1, 2, \cdots, n)$ are the components in the X and Y directions, respectively.

The control force and torque vectors generated by actuators are

$$F_1 = \frac{\tau_{1r} + \tau_{1l}}{r_1}U_1, F_2 = \frac{\tau_{2r} + \tau_{2l}}{r_2}U_2, ..., F_N = \frac{\tau_{nr} + \tau_{nl}}{r_n}U_n$$

$$M_{t1} = \frac{\tau_{1r} - \tau_{1l}}{r_1}b_1, ..., M_{t2} = \frac{\tau_{2r} - \tau_{2l}}{r_2}b_2, ...M_{tN} = \frac{\tau_{nr} - \tau_{nl}}{r_n}b_n \tag{139}$$

where, the vectors M_{ti} $(1, 2, \cdots, n)$ are oriented in the direction of Z-axis. The position vector r_{p1} can be expressed as

$$r_{p1} = \left(x + \sum_{i=1}^{n} d_i cos\theta_i \right) \hat{I} + \left(y + \sum_{i=1}^{n} d_i sin\theta_i \right) \hat{J}, ..., r_{pn} = [x, y]^T \tag{140}$$

The variation in position vector r_{p1} can be described by

$$\delta r_{p1} = \left(\delta x - \sum_{i=1}^{n} d_i sin\theta_i \delta\theta_i \right) \hat{I} + \left(\delta y + \sum_{i=1}^{n} d_i cos\theta_i \delta\theta_i \right) \hat{J},\ \delta r_{pn} = [\delta x, \delta y]^T \tag{141}$$

Also, the variation in the total work due to applied generalized forces is given by

$$\delta W = F_1^T \delta r_{p1} + F_2^T \delta r_{p2} + \cdots + F_N^T \delta r_{pn} + M_{t1}\delta\theta_1 + M_{t2}\delta\theta_2 + \cdots + M_{tN}\delta\theta_n \tag{142}$$

where δr_{p1}, δr_{p2}, $\cdots$, δr_{pn} and $\delta\theta_1$, $\delta\theta_2$, $\cdots$, $\delta\theta_n$ are variations of $r_{p1}, \cdots, r_{pn}$, and $\theta_1, \cdots, \theta_n$. Since q is defined by $q = \left[x, y, \theta_n, \theta_{n-1}, \ldots, \theta_1\right]^T$, one can rewrite Equation (144) as follows:

$$\begin{aligned} \delta W = F_1^T \cdot \frac{\partial r_{p1}}{\partial q} \cdot \delta q + F_2^T \cdot \frac{\partial r_{p2}}{\partial q} \cdot \delta q + ... + F_n^T \cdot \frac{\partial r_{pn}}{\partial q} \cdot \delta q \\ + M_{t1} \cdot \frac{\partial \theta_1}{\partial q} + M_{t2} \cdot \frac{\partial \theta_2}{\partial q} + ... + M_{tn} \cdot \frac{\partial \theta_n}{\partial q} Q^T \cdot \delta q \end{aligned} \tag{143}$$

where δq represents the incremental variation of q, and the vector Q, which represents the generalized forces that corresponds to generalized system coordinates, can be written as,

$$Q = \left(\frac{\partial r_{p1}}{\partial q} \right)^T \cdot F_1 + \nu + \left(\frac{\partial r_{pn}}{\partial q} \right)^T \cdot F_n + \left(\frac{\partial \theta_1}{\partial q} \right)^T \cdot M_{t1} + \nu + \left(\frac{\partial \theta_n}{\partial q} \right)^T \cdot M_{tn} \tag{144}$$

One can define the injection matrix $B(q)$ as

$$B(q) = \frac{\partial Q}{\partial \tau} \tag{145}$$

where,τ represents the control torque vector $\tau = [\tau_{1r}, \tau_{1l}, \tau_{2r}, \tau_{2l}, \cdots, \tau_{nr}, \tau_{nl}]$.

9. Appendix

9.2. Inertia Matrix $M(q)$ and Matrix of Coriolis Forces $C(q, \dot{q})$ for Robotic System

The position vectors $C_1, \ldots, C_n$ can be defined by

$$\begin{aligned} r_{c1} =& r_{p1} + a_1 U_1 \\ &= [x + dcos\theta_2 + a_1 cos\theta_1, y + dsin\theta_2 + a_1 sin\theta_1]^T \\ r_{cn} &= r_{pn} + a_n U_n = [x + a_n cos\theta_n, y + a_n sin\theta_n]^T \end{aligned} \tag{146}$$

The kinematic energy of the system is defined by

$$T(q, \dot{q}) = \frac{1}{2}\sum\nolimits_{i=0}^{n} \{I_{mi}\dot{\theta}_i^2 + m_i \dot{r}_{ci}^T \dot{r}_{ci}\} \tag{147}$$

Taking the time derivative of r_{c1}...., r_{cn}, to have

$$\dot{r}_{c1} = \frac{\partial r_{c1}}{\partial q} \cdot \dot{q} = T_1 \cdot \dot{q}, ..., \dot{r}_{cn} = \frac{\partial r_{cn}}{\partial q} \cdot \dot{q} = T_n \cdot \dot{q} \tag{148}$$

where, the matrices, $T_1, \ldots, T_n$ in (B.3) represents Jacobian matrices. Accordingly, one can rewrite the kinematic energy of robotic system as follows:

$$T(q, \dot{q}) = \frac{1}{2}\sum\nolimits_{i=0}^{n} \{I_{mi}\dot{\theta}_i^2 + m_i \dot{q}^T T_i^T T_i \dot{q}\} \tag{149}$$

where

$$M(q) = diag(\{0, 0, I_{m1}, I_{m2}, \cdots, I_{mn}\}) + \sum\nolimits_{i=0}^{n} \{m_i T_i^T T_i\} \tag{150}$$

where, $M(q)$ represents inertia matrix of robotic system, which is symmetric and positive-definite matrix.

One can proceed by writing the Lagrange formulation of the robotic system as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = Q + A^T(q)\lambda \tag{151}$$

From (149),

$$\frac{\partial T}{\partial \dot{q}} = M\dot{q} \tag{152}$$

Then,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = \dot{M}\dot{q} + M\ddot{q} - \frac{\partial T}{\partial q}$$
$$= M\ddot{q} + C(q, \dot{q}) \tag{153}$$

where the centripetal Coriolis vector $C(q, \dot{q})$ can be expressed as

$$C(q, \dot{q}) = \dot{M}\dot{q} - \frac{\partial T}{\partial q} \tag{154}$$

References

1. Verma V, Gordon G, Simmons R et al (2004) Real-time fault diagnosis [robot fault diagnosis]. IEEE Robot Autom Mag 11(2):56–66
2. Yuan J, Sun F, Huang Y (2015) Trajectory generation and tracking control for double-steering tractor-trailer mobile robots with on-axle hitching. IEEE Trans Industr Electron 62(12):7665–7677
3. Kolmanovsky I, McClamroch N (1995) Developments in nonholonomic control problem. IEEE Control Syst Mag 15(6):20–36
4. Yuan G, Yang S, Mittal G (2001) Tracking control of a mobile robot using a neural dynamic based approach. In: Proceedings 2001 ICRA, IEEE International conference on robotics and automation. Seoul, South Korea, pp 163–168
5. Hu Y, Yang S (2003) A fuzzy neural dynamics-based tracking controller for a nonholonomic mobile robot. In: Proceedings 2003 IEEE/ASME International conference on advanced intelligent mechatronics (AIM 2003). Kobe, Japan, pp 205–210
6. Kanayama Y, Kimura Y, Miyazaki F et al (1990) A stable tracking control method for an autonomous mobile robot. Proceedings, IEEE International Conference on Robotics and Automation, Cincinnati, USA 1:384–389
7. Chen B, Lee T, Chang W (1996) A robust H infinity model reference tracking design for nonholonomic mechanical control systems. Int J Control 63(2):283–306
8. Fliess M, Levine J, Martin P, Rouchon P (2007) Flatness and defect of non-linear systems: introductory theory and examples. Int J Control 61(6):1327–1361
9. Fierro R, Lewis F (1997) Control of a nonholonomic mobile robot: back stepping kinematics into dynamics. J Robot Syst 14:149–163
10. Fierro R, Lewis F (1998) Control of a nonholonomic mobile robot using neural networks. IEEE Trans Neural Networks 9(4):589–600
11. Dong W, Kuhnert K (2005) Robust adaptive control of nonholonomic mobile robot with parameter and non-parameter uncertainties. IEEE Trans Rob 21(2):261–266
12. Fukao T, Nakagawa H, Adachi N (2000) Adaptive tracking control of a nonholonomic mobile robot. IEEE Trans Robot Autom 16(6):609–615
13. Dong W, Huo W, Tso S et al (2000) Tracking control of uncertain dynamic nonholonomic system and its application to wheeled mobile robots. IEEE Trans Robot Autom 16(6):870–874
14. David J, Manivannan P (2014) Control of truck-trailer mobile robots: a survey. Intel Serv Robot 7(4):245–258
15. Laumond J (1993) Controllability of a multibody mobile robot. IEEE Trans Robot Autom 9(6), pp 755–763

16. Lamiraux F, Laumond J (2000) Flatness and small-time controllability of multibody mobile robots. Application to motion planning IEEE Transactions on Automatic Control 45(10):1878–1881
17. Laumond J, Sekhavat S, Lamiraux F (1998) Guidelines in nonholonomic motion planning for mobile robots. Robot Motion Planning and Control, Berlin, Germany: Springer-Verlag 229:1–54
18. Sordalen O (1993) Conversion of the kinematics of a car with n trailers into a chained form. IEEE Conf Robot Autom:382–387
19. Morin P, Samson C (2008) Transverse function control of a class of non-invariant drift less systems. Application to vehicles with trailers. In: Proceedings 47th IEEE conference on decision and control. Cancun, pp 4312–4319
20. Khalaji A, Moosavian S (2014) Robust adaptive controller for a tractor-trailer mobile robot. IEEE/ASME Trans Mechatron 19(3):943–953
21. Khalaji A, Bidgoli M, Moosavian S (2014) Non-model-based control for a wheeled mobile robot towing two trailers. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics 229(1):943–953
22. Zhang Y, Jiang J (2008) Bibliographical review on reconfigurable fault-tolerant control systems. Annu Rev Control 32(2):229–252
23. Yu X, Jiang J (2015) A survey of fault-tolerant controllers based on safety-related issues. Annu Rev Control 39:46–57
24. Ye D, Yang G (2006) Adaptive fault-tolerant tracking control against actuator faults with application to flight control. IEEE Trans Control Syst Technol 14(6):1088–1096
25. Fang J, Li W, Li H et al (2015) Online inverter fault diagnosis of buck-converter BLDC motor combinations. IEEE Trans Power Electron 30(5):2674–2688
26. Zhang K, Jiang B, Cocquempot V (2016) Fuzzy unknown input observer-based robust fault estimation design for discrete-time fuzzy system. Signal Processing, Elsevier 128(5):40–47
27. Luo M, Wang D, Pham M et al. (2005) Model-based fault diagnosis/prognosis for wheeled mobile robots: a review. In: Proceedings of 31st annual conference of IEEE industrial electronics society. pp 6–12
28. Uan Z, Cai Z, Yu J (2005) Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: a survey. In: Proceedings of the IEEE International conference on robotics and automation. pp 3428–3433
29. Şahin O, Dede M (2022) Model-based detection and isolation of the wheel slippage and actuator faults of a holonomic mobile robot. Industrial Robot, ahead-of-print (ahead-of-print)
30. Qina M, Dian S et al. (2022) Fractional-order SMC controller for mobile robot trajectory tracking under actuator fault. Syst Sci & Control Eng J 10(1)
31. Leite A, Schafer B, Souza M (2012) Fault-tolerant control strategy for steering failures in wheeled planetary rovers. Journal of Robotics 2012:15
32. Stavrou D, Eliades D, Panayiotou C et al (2016) Fault detection for service mobile robots using model-based method. Springer, Autonomous Robots 40(2):383–394
33. Shen Z, Ma Y, Song Y (2016) Robust adaptive fault-tolerant control of mobile robots with varying center of mass. IEEE Trans Ind Electron PP (99)
34. Ji M, Zhang Z, Biswas G et al (2003) Hybrid fault adaptive control of a wheeled mobile robot. IEEE/ASME Trans Mechatron 8(2):226–233
35. Chang Y, Wu C, Yang CY (2015) Adaptive output-feedback fault-tolerant tracking control for mobile robots under partial loss of actuator effectiveness. In: 2015 54th IEEE conference on decision and control. Osaka, pp 6306–6311
36. Dixon W, Walker I, Dawson D (2001) Fault detection for wheeled mobile robots with parametric uncertainty. In: 2001 IEEE/ASME International conference on advanced intelligent mechatronics, vol 2. pp 1245–1250
37. Bisgaard M, Vinther D, Østergaard K (2004) Modelling and fault-tolerant control of an autonomous wheeled robot. University of Aalbog, Institute of Control Engineering
38. Rotondo D, Puig V, Nejjari F et al (2014) A fault-hiding approach for the switching Quasi-LPV fault-tolerant control of a four-wheeled omnidirectional mobile robot. IEEE Trans Industr Electron 62(6):3932–3944

39. Koh M, Noton M, Khoo S (2012) Robust fault-tolerant leader-follower control of four-wheel-steering mobile robots using terminal sliding mode. Aust J Electr Electron Eng 9(4):247–254
40. Kim T, Park J, Kim H (2015) Actuator reconfiguration control of a robotic vehicle with four independent wheel driving. In: The 15th International conference on control, automation and systems. Busan, Korea, pp 1767–1770
41. Mohareri O, Daouadi R, Rad A (2012) Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks. Neurocomputing 88:54–66
42. Yang J, Kim J (1999) Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. IEEE Trans Robot Autom 15(3):578–587
43. Bushnell L, Tilbury M, Sastry S (1995) Steering three-input nonholonomic systems: the fire truck example. Int J Robot Res 14(4)
44. Isidori A (1989) Nonlinear control systems: an introduction, 2nd edn. Springer-Verlag New York, Inc. New York, USA
45. Murray R, Sastry S (1993) Nonholonomic motion planning: steering using sinusoids. IEEE Trans Autom Control 38(5):700–716
46. Murray R, Sastry S (1991) Steering nonholonomic systems in chained forms. In: Proceedings of the 30th Conference on decision and control, vol 2. Brighton, England, pp 1121–1126
47. Oriolo G, Luca A, Vendittelli M (2002) WMR control via dynamic feedback linearization: design, implementation, and experimental validation. IEEE Transaction on Control Systems Technology 10(6):835–852
48. Dong W, Xu W (2001) Adaptive tracking control of uncertain nonholonomic dynamic system. IEEE Trans Autom Control 46(3):450–454
49. Kuc T, Baek S, Park K (2001) Adaptive learning controller for autonomous mobile robots. IEE Proc Control Theory Appl 148(1): 49–54
50. Morin P, Samson C (2009) Control of nonholonomic mobile robots based on the transverse function approach. IEEE Trans Rob 25(5):1058–1073
51. Buccieri D, Perritaz D, Mullaupt P et al (2009) Velocity-scheduling control for a unicycle mobile robot: theory and experiments. IEEE Trans Rob 25(2):451–458
52. Dixon W, Dawson D, Zhang F et al (2000) Global exponential tracking control of a mobile robot system via a PE condition. IEEE Trans Syst Man Cybern 30(1):129–142
53. Ortigoza R, Ortigoza G, Guzman V et al (2008) Trajectory tracking in a mobile robot without using velocity measurement for control of wheels. IEEE Latin America Transaction 6(7):598–607
54. Jiang Z, Nijmeijer H (1999) A recursive technique for tracking control of non-holonomic systems in chained form. IEEE Trans Autom Control 44(2):265–279
55. Huang T (2009) Adaptive tracking control of high-order nonholonomic mobile robot systems. IET Control Theory & Appl 3(6):681–690
56. Dixon W, Dawson D, Zargeroglu E et al (2001) Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. IEEE Trans Syst Man Cybern 31:341–352
57. Dixon W, Queiroz M, Dawson D et al (2004) Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. IEEE Trans Control Syst Technol 12(1):138–147
58. Coelho P, Nunes U (2005) Path-following control of mobile robots in presence of uncertainties. IEEE Trans Rob 21(2):252–261
59. Yang J, Kim J (1999) Sliding mode motion control of nonholonomic mobile robots. IEEE Control Syst 19(2):15–23
60. Ma Y, Cocquempot V, el Najjar M, Jiang B (2017) Multi design integration based adaptive actuator failure compensation control for two linked 2WD mobile robots. IEEE/ASME Trans Mechatron 22(5):2174–2185
61. Ma Y, Cocquempot V, El Najjar M, Jiang B (2017) Adaptive compensation of multiple actuator faults for two physically linked 2WD robots. IEEE Trans Robot PP (99):1–8
62. Ma Y, Cocquempot V, El Najjar M, Jiang B (2017) Actuator failure compensation for two linked 2WD mobile robots based on multiple-model control. Int J Appl Math Comput Sci (AMCS) 27(4)

63. Ma Y, AL-Dujaili A, Cocquempot V, EL Najjar M (2016) An adaptive actuator failure compensation scheme for two linked 2WD mobile robots. Adv Control Diagn ACD
64. AL-Dujaili A, Ma Y, EL Najjar M, Cocquempot V (2017) Actuator fault compensation in three linked 2WD mobile robots using multiple dynamic controllers. IFAC WC, Toulouse
65. AL-Dujaili A, Cocquempot V, EL Najjar M, Ma Y (2017) Actuator fault compensation tracking control for multi linked 2WD mobile robots. In: IEEE MED 2017, 25th Mediterranean Conference on control and automation
66. Ryu J, Agrawal S, Franch J (2008) Motion planning and control of a tractor with a steerable trailer using differential flatness. J Comput Nonlinear Dyn 3(3)
67. Khalaji A, Moosavian S (2016) Dynamic modeling and tracking control of a car with trailers. Multibody SysDyn 37(2):211–225
68. Matos A (2011) Optimization and control of nonholonomic vehicles and vehicles formations. Ph.D. Thesis, University of Porto
69. Murray R, LI Z, Sastry S (1994) A mathematical introduction to robotic manipulation. Textbook by CRC Press. ISBN 9780849379819, p 480
70. Tao G, Shuhao C, Tag X, Joshi S (2004) Adaptive control of system with actuator failures. Springer, New York
71. Liao J, Chen Z, Yao B (2017) Performance-oriented coordinated adaptive robust control for four-wheel independently driven skid steer mobile robot. IEEE Access 5:19048–19057
72. Park S, Yoo S, Park J, Choi H (2009) "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty. IEEE Trans Control Syst Technol 17(1):207–214
73. Hao X, Zheng J, Rifai C, Hung N (2021) Robust tracking control of a differential drive wheeled mobile robot using fast nonsingular terminal sliding mode. Comput & Electr Eng J 96(Part A):207–214
74. Labbadi M, Boubaker S, Djemai M et al. (2022) Fixed-time fractional-order global sliding mode control for nonholonomic mobile robot systems under external disturbances. Fractal Fract J 6(177)
75. Yang H, Guo M, Xia Y, Cheng L: Trajectory tracking for wheeled mobile robots via model predictive control with softening constraints. IET Control Theory & Appl 12(2):206–214
76. Al-Dujaili A, Amjad H, Pereira A, Kasim I (2021) "Adaptive backstepping control design for ball and beam system". Int Rev Appl Sci Eng 12(3):211–221

Design and Implementation of a Robust 6-DOF Quadrotor Controller Based on Kalman Filter for Position Control

Amar Bousbaine, Abdelkader Fareha, Ajay K. Josaph, Arezki Fekik, Ahmad Taher Azar, Riad Moualek, Nabil Benyahia, Nacereddine Benamrouche, Nashwa Ahmad Kamal, Ammar K. Al Mhdawi, Amjad J. Humaidi, and Ibraheem Kasim Ibraheem

Abstract The objective of this chapter is to develop quadcopter flight control algorithms using a PID controller enhanced by a Kalman Filter (KF) using an experimental approach to extract the physical and aerodynamic settings of the quadcopter. It is first necessary to present the current state of the quadcopter analytical dynamics model in order to achieve an effective design. A second step involves the development of the quadcopter's hardware and software, as well as the development of a full thrust test rig to extract the parameters of the propulsion system and the linearisation approximations between the different variables. Using the quadcopter's 6-DOF analytical dynamic model, the controller's control parameters are determined using a PID design enhanced with KF. Test results were assessed using dynamic response curves and 3D Matlab visualisations. In order to evaluate the performance of the PID controllers, we measured the time response, overshoot, and settling time with and without the KF. After the SIMULINK model's results for the drone were accepted, a C++ code was produced. Uploading the generated code into the Pixhawk autopilot

A. Bousbaine
College of Science and Engineering, University of Derby, Derby, UK
e-mail: a.bousbaine@derby.ac.uk

A. Fareha · A. K. Josaph
University of Derby, Derby, UK
e-mail: a.koikarathil-joseph1@unimail.derby.ac.uk

A. Fekik (✉)
Department of Electrical Engineering, University Akli Mohand Oulhadj-Bouria, Rue Drissi Yahia Bouira 10000, Algeria
e-mail: a.fekik@univ-bouira.dz

A. Fekik · R. Moualek · N. Benyahia · N. Benamrouche
Electrical Engineering Advanced Technology Laboratory (LATAGE), Mouloud Mammeri University, Tizi Ouzou, Algeria
e-mail: nacereddine.benamrouche@ummto.dz

A. T. Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia
e-mail: aazar@psu.edu.sa; ahmad.azar@fci.bu.edu.eg; ahmad_t_azar@ieee.org

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_11

was accomplished through a Simulink application in the autopilot firmware. Based on the Pixhawk autopilot, we present a quick and real-time test solution for drone controllers. Further enhancements are provided by near-real-time tuning of the control settings. This research uses the Embedded Coder Tool to develop SIMULINK-generated code for the Pixhawk autopilot board.

Keywords Quadcopter · Simulink · Pixhawk · Kalman filter · PID controller · Dynamic model

1 Introduction

The term drone refers to an Unmanned Aerial Vehicle (UAV) that does not have a pilot. In most cases, they are controlled by a ground control station. Depending on the algorithm used by the onboard computers, the flight may be controlled autonomously or remotely from the ground. UAVs are often preferred over manned aircraft due to their flexibility, endurance, and cost-effectiveness. Drones or unmanned aerial vehicles (UAVs) are aircraft, helicopters, etc. that are controlled remotely with a remote control or are autonomously operated by an algorithm. A UAV is generally classified based on its weight, altitude range, and endurance. The main types of aircraft are vertical take-off and landing (VITOL) aircraft such as quadcopters and helicopters as well as flapping wing aircraft [7, 8, 15, 46, 57]. However, with respect to the other principles of flight, VTOL systems have particular properties that make it possible to execute applications that would be difficult or impossible with other

Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh 11586, Saudi Arabia

Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

N. A. Kamal
Faculty of Engineering, Cairo University, Giza, Egypt
e-mail: nashwa.ahmad.kamal@gmail.com

A. K. Al Mhdawi
Department of Computer Science and Engineering, Edge Hill University, Ormskirk, UK
e-mail: Al-Mhdaa@edgehill.ac.uk

Robotics and Autonomous Systems Group (RAS). Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

A. J. Humaidi
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
e-mail: amjad.j.humaidi@uotechnology.edu.iq

I. K. Ibraheem
Department of Computer Techniques Engineering, Dijlah University College, Baghdad 10001, Iraq
e-mail: ibraheemki@coeng.uobaghdad.edu.iq

systems. VTOL aircraft are capable of hovering and flying at low speeds in addition to taking off and landing vertically. As a result of the simplicity of control, this method has a significant advantage. VTOL, in its various configurations, represents one of the most compact and capable flying concepts available today [2, 3]. As a result of the complexity of drone control, they have generated considerable academic interest. Because of their complex dynamics, they are suitable for both indoor and outdoor applications [9, 10, 31, 54]. As a result of their emergence as operative solutions to achieving a wide range of civil or military missions over the past year, drones or Unmanned Aerial Vehicles (UAVs) have soared in popularity in the field of aviation. These include military attacks with arms payloads, land surveying with cameras and other tools, and even delivering goods. However, the persisting issues of autopilot maneuverability and controllability in noisy and harsh environments haves affected the autopilot controllability during the flight and limit the aerial maneuverability operation [15, 24].

Specifically, the research proposed focuses on the navigation system of quadcopters due to its affordability (low financial impact) and simplicity of use (no need for supporting infrastructure). Based on the Kalman Filter algorithm, the system is able to generate accurate and precise measurements for a more effective autonomous navigation system [15, 24, 29, 36, 42].

Experimental extrapolation on a thrust test bench is used to determine the parameters for calibrating the actuator. As a result, it is possible to construct a complete monitoring system for the quadcopter within the given specifications. Matlab/Simulink was used to create the mathematical model of a 6-DOF quadcopter based on the experimentally extracted parameters. Through the identification of settings, the inverse dynamic control, which is employed with the PID regulator to identify the required inputs to the model, can be approximated linearly. It is the role of the Kalman filter to estimate the actual detector measurements that will be used as feedback information for the flight controller (in this case, the PID controller) [25, 32, 34, 55, 56]. As a result of converting the monitoring system into C++ code, the flight regulator system is obtained which includes the Simulink model of the fully planned monitoring system [14, 35].

Our objective in this chapter is to develop quadcopter flight control algorithms based on an experimental technique that is designed to extract physical and aerodynamic settings of a quadcopter based on a PID regulator enhanced by a Kalman filter. Initially, an analytical dynamics model of the quadcopter is developed in order to develop an effective design. The second objective is to demonstrate the material and software of the quadcopter as well as a full thrust test setup for extracting data from the propulsion system and making linear approximations between the variables. Using the quadcopter's 6-DOF analytical dynamic model, the KF enhances the PID-designed regulator and determines the corrector control settings. A 3D Matlab visualization and dynamic response curves were used to test the feasibility of the simulation tests. Overshoot, settling time, and response time were measured as indicators of the effectiveness of the PID controllers, both with and without the KF. By creating a Simulink application within the autopilot firmware, the produced regulator is converted into C++ and loaded into the autopilot of the Pixhawk following simulation validation of the Simulink model controller results. In addition, a real-time test solution for a quadcopter monitoring system based on the Pixhawk autopilot is presented, as well as further adjustments to the real-time control settings. With the implementation of Embedded Coder, Simulink code is deployed onto the autopilot board of the Pixhawk.

2 Analytical Model of Quadcopter

2.1 Basic Concept

The mechanical arrangement of the quadcopter consists of four propellers and motors. As can be seen in Fig. 1, the parallel blades and fixed pitch of the propellers cause lift forces which are directed upward by providing a downward flow of air. As long as the structure is rigid, these forces are determined by the propeller velocity. As can be seen from the rotating configuration of the propeller pair, the front and rear propellers rotate in the opposite direction to the left and right propellers [12, 13, 18, 48]. It is not possible to control a variable in all six degrees of freedom of a quadrotor due to its four propellers and six degrees of freedom. As thrust, roll, pitch, and yaw are the four fundamental movements of the drones, a four-degree of freedom has been used as an alternative method [12, 13, 18, 48].

2.2 Dynamic Modelling of the Quadcopter

According to [12, 13, 18, 48], a Newton-Euler formalism has been used to model the dynamics of the UAV system. Quadcopter motion equations are derived from kinematic equations, aerodynamic forces, and moments. In this method, the four control settings (thrust, roll, pitch, and yaw) are designated as the outputs and the tension or vector moments as the inputs. This model is based on the following assumptions:

- Assume that the propellers and structure are rigid and symmetrical;
- A body's fixed frame origin and its Center of Gravity (CoG) should be aligned;
- A body's fixed frame origin should coincide with its center of gravity (CoG);
- Inertia matrices possess time invariant properties.

According to [12, 13, 18, 48], in order to express the quadcopter dynamic system, it was first expressed in terms of the body's frame (B-frame) and then in terms of the ground frame (E-frame) using a hypothetical hybrid frame (H-frame) according to the quadcopter '+' configuration, as shown in Fig. 1. Based on the formalism and assumptions provided above, the dynamics model of a quadrotor takes into account all physical impacts as well as the two coordinate systems (B-frame and E-frame). Among these considerations are:

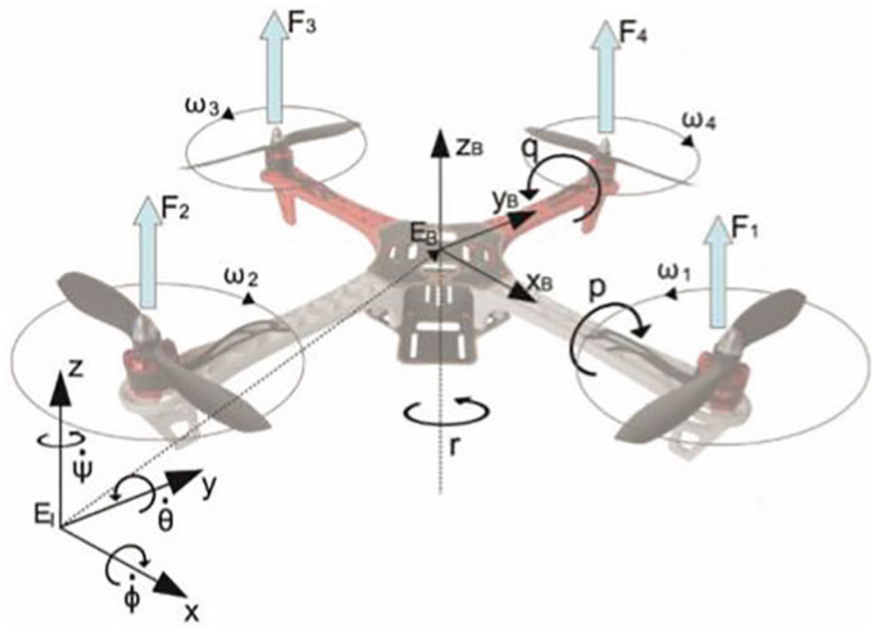


Fig. 1 Quadcopter corps and inertial frame

The inertial reference is the ground inertial frame (E-frame), which is displayed by the numbers (OE, XE, YE, ZE). OE represents the origin, XE represents the north, YE represents the west, and ZE represents the upward direction. The following coordinates (OB, XB, YB, ZB) indicate the vehicle's body-fixed frame (B-frame). XB represents the front of the quadcopter, YB represents the left of the quadcopter, ZB represents the top, and OB represents the origin of the axis, which coincides with the center of the quadcopter's crossbar. According to Eq. 1, motion is represented as a non-matrix formula in the H-frame [12, 13, 18, 48].

$$\begin{gathered}
\ddot{X} = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) + \tfrac{U_1}{m} \\
\ddot{Y} = (\sin\psi\sin\theta\cos\phi - \cos\psi\cos\phi) + \tfrac{U_1}{m} \\
\ddot{Z} = (\cos\theta\cos\phi)\tfrac{U_1}{m} - g \\
\ddot{P} = \tfrac{I_{yy}-I_{zz}}{I_{xx}}q_r - \tfrac{I_{TP}}{I_{xx}}q\Omega + \tfrac{U_2}{I_{xx}} \\
\ddot{q} = \tfrac{I_{zz}-I_{xx}}{I_{yy}}P_r - \tfrac{I_{TP}}{I_{yy}}P\Omega + \tfrac{U_3}{I_{yy}} \\
\ddot{r} = \tfrac{I_{xx}-I_{yy}}{I_{zz}}Pq - + \tfrac{U_4}{I_{zz}}
\end{gathered} \tag{1}$$

where X, Y, and Z are the linear displacements and p, q, and r are the angular velocities, respectively;ϕ, θ, and ψ are the Euler angles about the x, y, and z alignments; the lift force is represented by U_1; and U_2, U_3, and U_4 are the torques about the x, y, and z axes that are coupled to I_{xx}, I_{yy}, and I_{zz}. The relationship between lift force and torques produced by propeller velocity is the link formed by Eq. (2).

$$\begin{gathered} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(\Omega_2^2 - \Omega_4^2) \\ U_3 = lb(\Omega_3^2 - \Omega_1^2 \\ U_4 = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ \Omega = (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \end{gathered} \tag{2}$$

2.3 Dynamic of the Actuator

It is necessary to establish the relationship between the propellers' velocity and the tension provided to the actuator and the dynamic response of the rotor system. In consideration of Kirchhoff's voltage law and ignoring the inductance effect in a DC motor, the relationship between motor voltage and angular speed can be expressed by the following equation [24]:

$$v = Zi + K_e \Omega \tag{3}$$

where Z: Motor impedance (resistance) [Ω], v: Electric Tension applied to the motor [V], I: Electric current [A], and K_e: motor angular velocity [rad/s] and Ω: motor velocity constant [V s/rad]. According to [24], Eq. 4 can be utilized to express the equation between angular velocity and Tension as follows [24]:

$$V = \frac{ZD\Omega^2}{K_q} + \frac{ZJ\dot{\Omega}}{K_q} + K_e \Omega \tag{4}$$

Motor settings include: Z: motor impedance [Ω], J: total motor moment of inertia [N ms^2], $\dot{\Omega}$: motor angular acceleration [rad/s^2], and the physical constants K_e and K_q have the same value [Nm/A] and D: the drag coefficient [24]. Equation 5 supply the relationship between Tension and thrust, according to [24].

$$F = 2\rho A \left[\frac{f \eta b}{K_q} \right]^2 V^2 \tag{5}$$

where F stands for thrust force [N], V for the electric Tension utilized by the motor [V], ρ for air density [Kg m^3], f for figure of merit [NIL], b for torque constant [m], A for the surface of the propeller plate [m^2], K_q for motor velocity constant [Nm/A], and η for motor effectiveness [%]. By linearizing the impedance and the motor angular velocity constant using the developed test bench, the major parameters that contribute to the modeling of quadrotors are obtained.

3 Controller Design

There are numerous components to the controller design process [30, 39, 40, 45, 47]. To estimate the signals that have been contaminated by noise in the environment and measurement detectors, the first step is to develop the mathematical analytical models that will be used to create a linearized control model with feedback. A PID regulator augmented by the KF forms the basis of the flight monitoring system. Using the experimentally retrieved settings, the existing Matlab/Simulink model is calibrated and the regulator is created.

3.1 PID Controller

Proportional Integrator Derivative (PID) regulators are commonly known for their simplicity. These regulators are used in various industrial sectors for process control [1, 5, 6, 20, 22, 23, 26–28]. The PID regulator is controlled by three actions proportional (P), integral (I), and derivative (D) as shown in Fig. 2. Each setting's action has different role leading to a minimized error between the required value or the command and the measured value from the process [4, 11, 37, 38, 41, 43, 44]. It is the proportional action that reduces the difference between the regulated and measured values (the error), the integral action that integrates the error, and the derivative action that determines the rate. Depending on the variables to be controlled and the dynamics of the process, there are four compositions of the regulator. The regulator may be composed of proportional-integral-derivative (PID), proportional-integral (PI), proportional-derivative (PD) or proportional only (P). In order to enable or adjust the actions, the right gains must be determined for K_p, K_i, and K_d [4, 17, 33].

There are four major specifications to be considered when tuning a closed-loop step response system at steady state. They include: (i) the rise time when the output

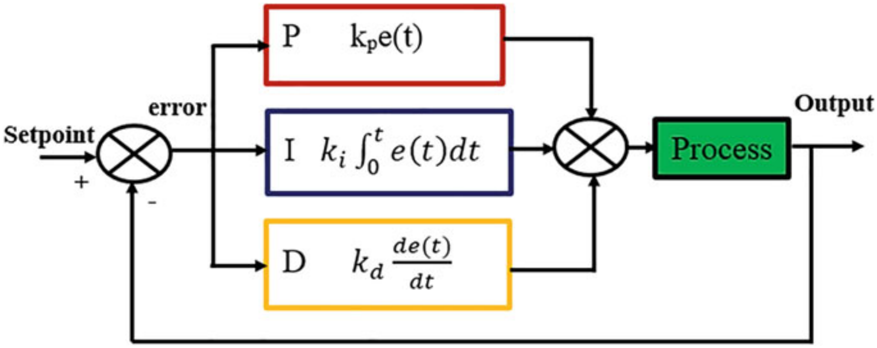


Fig. 2 PID regulator structure

Table 1 Effects of PID gains on the specifications of the response

	Rise time	Overshoot	Settling time	Equilibrium state error
K_p	Reduction	Growthly	Minor modification	Reduction
K_d	Minor modification	Reduction	Reduction	Minor modification
K_i	Reduction	Growthly	Growthly	Clear

reaches 90% of the required value, (ii) the overshoot, which is the deviation above steady state, (iii) the time response, which is how long it takes to converge to the steady state, and iv) the steady-state error, or difference between steady state and desired value [16, 21, 49]. For the purpose of achieving the specifications required, the actions of gains' effects are described in the Table 1. Consequently, the three tuning parameters must be balanced in order to adjust a system.

In order to tune the PID loop, the proportional gain (K_p) is the first to start with, so it brings the system to oscillate around the required value. Following that, it is necessary to activate the derivative gain K_d to minimize the oscillation and settling time and augmentation of rise time. In the last step, it is necessary to adjust K_i in order to eliminate the offset between the required value and the process value.

3.2 Kalman Filter (KF)

In the KF, the state-space variable x is calculated based on the linear-stochastic difference formula below, which is used to estimate the value of the discrete time process (6):

$$\begin{gathered} X_k = AX_{k-1} + Bu_k + W_{k-1} \\ Y_k = CX_k + V_k \end{gathered} \tag{6}$$

Matrix A, B, and C are used to link the time step state variable X_{k-1} to the present state variable vector X_k. It is assumed that the random variable of the process is represented by W, and that the measurement noise is represented by V, which are both considered to be white noise and Gaussian noise, respectively, and are independent of each other. A regulator's input is represented by u, the state vector is represented by x, and the output state measurement is represented by y.

3.3 Linear State Space Representation

A mathematical model based on the non-matrix relation of motion is used to represent an accurate representation of the dynamic and control operational behavior of a quadcopter. A proper set up of these equations is required when designing monitoring systems and their representations in state space.

As a result, the transition state of the system must be defined in terms of these state-space vectors. $X = (\dot{z}, z, \dot{\Phi}, \Phi, \dot{y}, y, \dot{\theta}, \theta, \dot{x}, x, \dot{\Psi}, \Psi)^T$ and input vectors $U = (u_1, u_2, u_3, u_4)$ During the operation of the quadcopter, each of the sub-systems (Height, roll, pitch, and yaw) is supposed to operate independently from the others. Each of the sub-systems (Height, roll, pitch, and yaw) should act independently of each other.

$$\begin{aligned} \dot{X} &= \tfrac{U_1}{m} = -g\theta \\ \dot{Y} &= \tfrac{U_1}{m} = g\Phi \\ \dot{Z} &= \tfrac{U_1}{m} - g \\ \dot{P} &= \tfrac{U_2}{I_{xx}} \\ \dot{q} &= \tfrac{U_3}{I_{yy}} \\ \dot{r} &= \tfrac{U_4}{I_{zz}} \end{aligned} \tag{7}$$

where the Euler angles (Φ, θ, Ψ) are, respectively, the rates of p, q, and r. Therefore, the state space display is divided into four movements that can be executed vertically, longitudinally, and radially. The quadcopter's dynamics may therefore be analyzed as separate systems since they are disconnected from one another, with the state vector X = $(Vertical, Directional, Latitude, Longitudinal)^T$ and the state space display being, respectively, as follows

$$\begin{bmatrix} \ddot{Z} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{Z} \\ Z \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} U_1 \tag{8}$$

$$\begin{bmatrix} \ddot{\Psi} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{\Psi} \\ \Psi \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{yy}} \\ 0 \end{bmatrix} U_4 \tag{9}$$

$$\begin{bmatrix} \ddot{\Phi} \\ \dot{\Phi} \\ \ddot{Y} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{\Phi} \\ \Phi \\ \dot{Y} \\ Y \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} \\ 0 \\ 0 \\ 0 \end{bmatrix} U_3 \tag{10}$$

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \\ \ddot{X} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} \dot{\theta} \\ \theta \\ \dot{X} \\ X \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{zz}} \\ 0 \\ 0 \\ 0 \end{bmatrix} U_2 \tag{11}$$

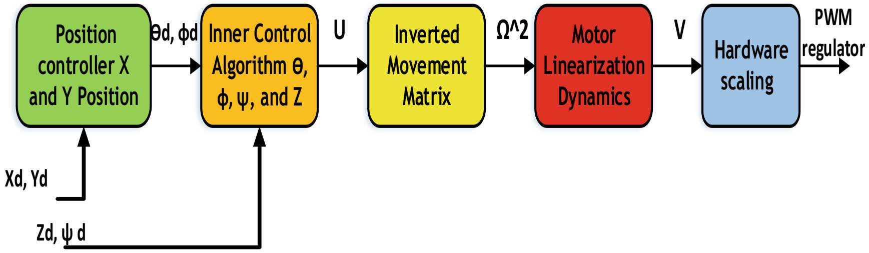

Position controller X and Y Position
θd, ϕd
Inner Control Algorithm θ, ϕ, ψ, and Z
U
Inverted Movement Matrix
Ω^2
Motor Linearization Dynamics
V
Hardware scaling
PWM regulator
Xd, Yd
Zd, ψ d

Fig. 3 Flight regulator block diagram

3.4 *Flight Control Algorithm*

As part of the flight control algorithm, the KF, measurement predictor, provides inputs which are compared to the required value of the trajectory generator. Different computation stages are required to determine the required trajectory during which a large number of constants and variables are evaluated. The regulator block diagram is shown in Fig. 3 that supply a description of the inverse dynamics of the quadrotor. Flight regulators are comprised of two cascading PID controllers, one for position and the other for altitude and attitude.

3.4.1 Control Technique

Based on the traditional PID structure associated with the Quadcopter motor mixer, the control technique adopted is based on the relationship between the PID regulator error and the required input voltage for the system to achieve the set point. In addition to the PID, the quadcopter motor mixer is enhanced by a saturation block that determines the linear zone of error so that any undesirable effects or derivates, such as integral winding, can be avoided.

3.4.2 Outer Loop

This loop runs at a lower frequency than the inner loop, and provides autonomous control of the quadcopter by introducing the X [m] and Y [m] coordinates, which are then converted into attitude commands in terms of Roll and Pitch [50]. A description of the PID controllers developed for X and Y can be found in Fig. 4.

Xd [m] and Yd [m] represent the required coordinates, and X [m] and Y [m] are the estimated coordinates from KF. The error provided by the PID is divided by [m s^2] according to Eq. 6 and results in the required output $\Phi_d \theta_d$.

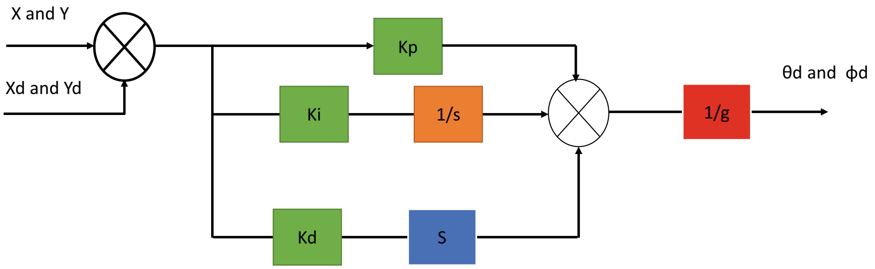


Fig. 4 X and Y PID regulator designed

3.4.3 Inner Loop

The inner loop is the main controller for the quadcopter, as it is responsible for directly transferring angular accelerations into translational movements by establishing torque around the x and y-axis [52]. The control technique utilized in this loop is a PID enhanced by KF to control the height Z, Roll Φ, Pitch θ and Yaw Ψ.

- Throttle controller
  Height of the vehicle is controlled by the throttle PID regulator. A saturation block has been added at the output to improve the performance. Based on the calibration of the model, the max and min values of the saturator have been extracted, Fig. 5. Z_d: represents the required throttle [m], Z: is the throttle planned from the KF [m], e_z: is the throttle error [m], g: gravity acceleration [m s^2], m: mass of the vehicle [kg], and U1: required thrust [N].
- Roll and Pitch controller
  Roll and Pitch PID controllers are similar in structure as shown in block diagram, Fig. 6. The rolls acts in the similar manner as the pitch, with action being around the x-axis.
  Φ_d [rad], θ_d [rad] represents the required Roll and Pitch angles (introduces manually or supplied by the outer loop), Φ [rad] and θ [rad] are the measured angles, e_Φ [rad] and e_Θ [rad]: Roll and Pitch errors, I_{xx} [N m] and I_{yy} [N m]: the inertia corps moment around the x-axis and y-axis, and the last U_2 [N m] and U_3 [N m]: the required Roll and Pitch torques, are. This design comes from Eq. 6 which relates the Roll and Pitch control to U_2 and U_3.
- Yaw controller
  Figure 7 represents the block scheme of the PID yaw control Ψ_d [rad s^1]: signifies the required yaw angle, Ψ [rad]: signifies the yaw angle estimated from Kalman Filter, e_Ψ [rad]: the yaw error, I_{zz} [N m]: the inertia corps moment around the z-axis, and U_4 [N m] the required yaw torque. This design comes from Eq. 6 which relates the Roll and Pitch control to U_4.

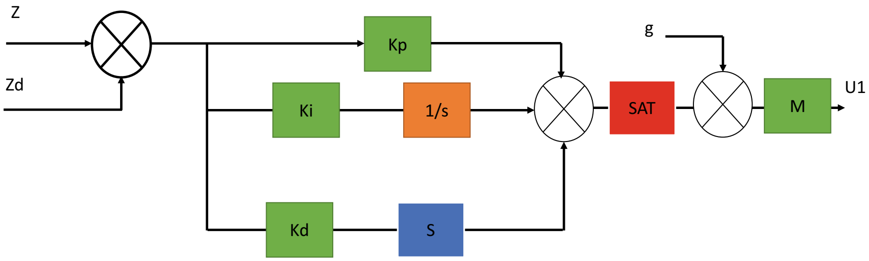


Fig. 5 Throttle PID regulator designed

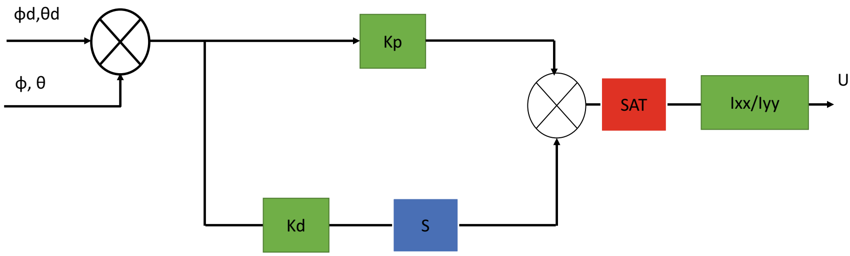


Fig. 6 Pitch and Roll PD controllers Structure

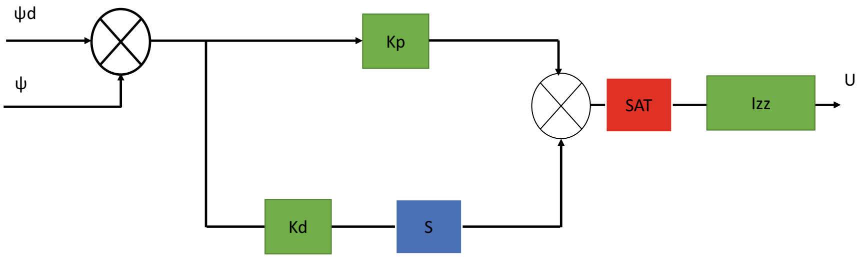


Fig. 7 Yaw PD controllers structure

3.4.4 Inverted Movements

As shown in Eq. 12, the matrix block, whose settings were established through experimentation, allows the input state vector to be converted into the squared normal propellers' squared velocity.

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & 0 & -\frac{1}{2bl} & -\frac{1}{4d} \\ \frac{1}{4b} & -\frac{1}{2bl} & 0 & \frac{1}{4d} \\ \frac{1}{4b} & 0 & \frac{1}{2bl} & -\frac{1}{4d} \\ \frac{1}{4b} & \frac{1}{2bl} & 0 & \frac{1}{4d} \end{bmatrix} * \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{12}$$

Table 2 PID settings for controllers

	Throttle	Roll angle	Pitch angle	Yaw
K_p	75	12	−8.5	−20
K_i	2.8	0.007	0.007	0
K_d	140	−12	−10.5	−100

4 Simulation Results

During this section, three iterations will be performed on the Simulink block. The first step is to demonstrate the efficiency of the PID regulator with and without KF. Using the PID parameters in Table 2, certain limits may be set for the throttle, roll, pitch, and yaw independently. This second iteration will demonstrate the effectiveness of the required command based on a number of trajectory scenarios. A third, final iteration of the simulation demonstrates the ability of the quadcopter to maintain its altitude and attitude. In this simulation, the aim is to demonstrate the effectiveness of the corrector in terms of tracking differences and to compare the controller's behavior with and without KF for the 6-DOF and in 3D.

4.1 Iteration 1: Monitoring System Effectiveness

- Altitude
  In order to evaluate a quadcopter's control effectiveness, the throttle is one of the simplest flight commands. As described earlier, the aircraft maintains a fixed position in the air with the same voltage applied to the four motors during this movement. With the corresponding settings for PID, Table 2, and the optimal matrix coefficients Q and R, the initial value of the state variable is estimated to be zero at the beginning and the matrix is considered time invariant after validation of the PID regulator and KF in the simulation. Figure 8, The altitude response of a quadcopter with a PID controller with and without KF is shown in blue and red respectively. In the presence of two noises—white Gaussian noise, factory noise, and measurement noise—the throttle command (z) was set at 10 m during an infinite time of control. Table 3, illustrates the PID effectiveness settings with and without KF. As can be seen, KF is operating as an ideal observer, determining the new altitude value accurately after reducing the noise error using diagonal matrices Q and R in accordance with the values in Eqs. 13 and 14.

$$Q = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.001 \end{bmatrix} \tag{13}$$

$$R = 0.0001 \tag{14}$$

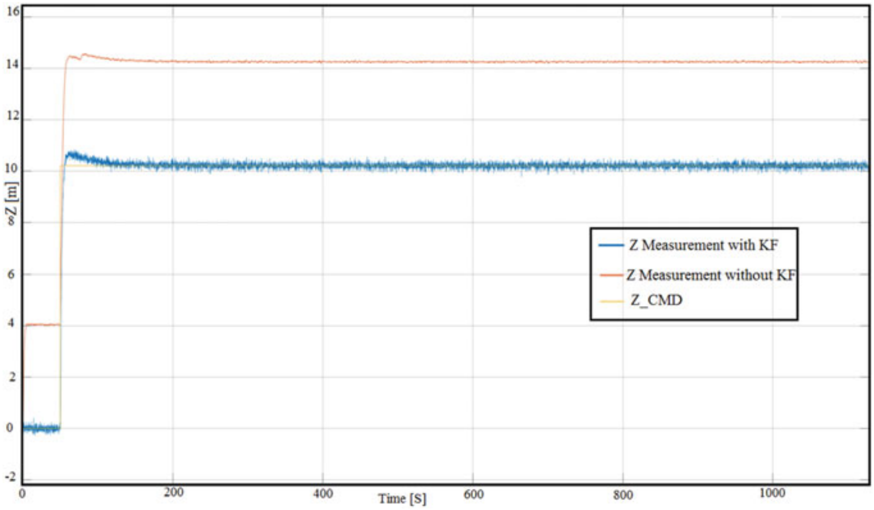


Fig. 8 The PID controller's throttle response with and without Kalman Filter

Table 3 PID effectiveness for height

Controller	Time rise (S)	Overshoot (%)	Offset (m)	Settling time (s)
PID without KF	0.8	5	4	10
PID with KF	0.8	5	0.2	10

- Pitch and Roll

  X and Y axis represent the horizontal movement, respectively, while tilting around the axis produces linear movement along the horizontal axis. In order to achieve attitude stability, the enhanced PD regulator is used, with the maintainability settings described in Table 4. The simulation results is shown in the Fig. 9, which represents the response of the system for the PD regulator with KF (blue) and without KF (red). The command for Pitch and Roll, were set to $+10\ \theta$ and $-10\ \theta$, the simulation time was set to 100 s, with the assumption of the presence of white Gaussian noise $\pm 0.5\%$ for measurement and $\pm 1\%$ for the factory noise. The Table 4, presents the effectiveness response with and without KF, and Eqs. 15 and 16 provide the optimal Q and R matrices for the KF utilized in Pitch and Roll, respectively.

$$Q = \begin{bmatrix} 0.08 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{15}$$

$$R = \begin{bmatrix} 0.0006 & 0 \\ 0 & 0.006 \end{bmatrix} \tag{16}$$

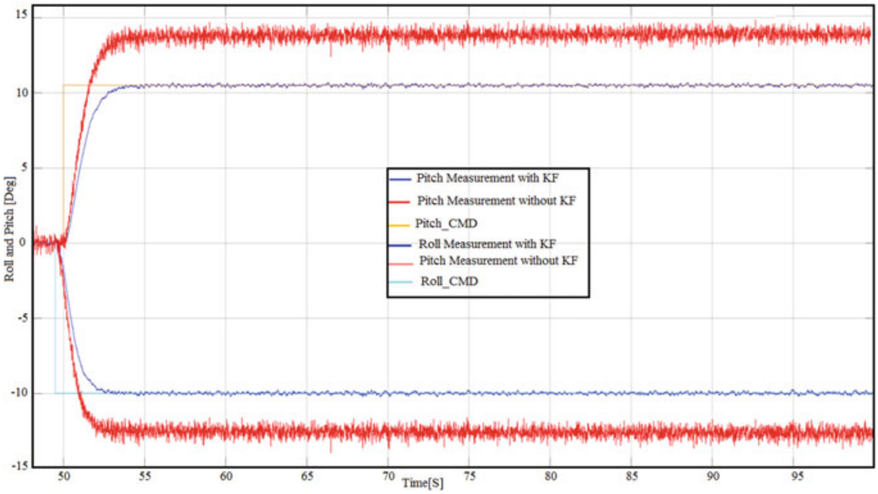


Fig. 9 PID controllers with and without KF respond differently to roll and pitch

- Directional movement (YAW)

A quadcopter's directional movement is determined by its rotation around itself. The control of this manoeuvre is provided by the PD regulator, which is set as indicated in Table 5. Figure 10, illustrates the response to the command at the location of 90° rotation around the z-axis, with the red line representing the response without KF and the blue line representing the response with KF. As with pitch and roll, the noise is similar. Clearly, the effectiveness of the maneuver differs between the quadcopter with and without KF, especially when it comes to the offset command. The effectiveness is validated for this command and the Q and R matrix for the KF and it is evident that KF is acting as an optimal observer, estimating the new altitude value accurately by reducing the noise error using R and Q diagonal matrices according to agreed elements in Eqs. 17 and 18 observer, estimating the new altitude value correctly by reducing the noise error utilizing Q and R diagonal matrices with the according to agreed elements Eqs. 17 and 18.

$$Q = \begin{bmatrix} 0.0035 & 0 \\ 0 & 0.001 \end{bmatrix} \tag{17}$$

$$R = 0.001 \tag{18}$$

Table 4 PID effectiveness for Roll and Pitch

Regulator	Movement	Time rise (s)	Overshoot (%)	Offset (°)	Settling time (s)
PID without KF	Roll	3	0	3	3
	Pitch	3	0	3	3
PID with KF	Roll	3	5	0.2	3
	Pitch	3	5	0.2	3

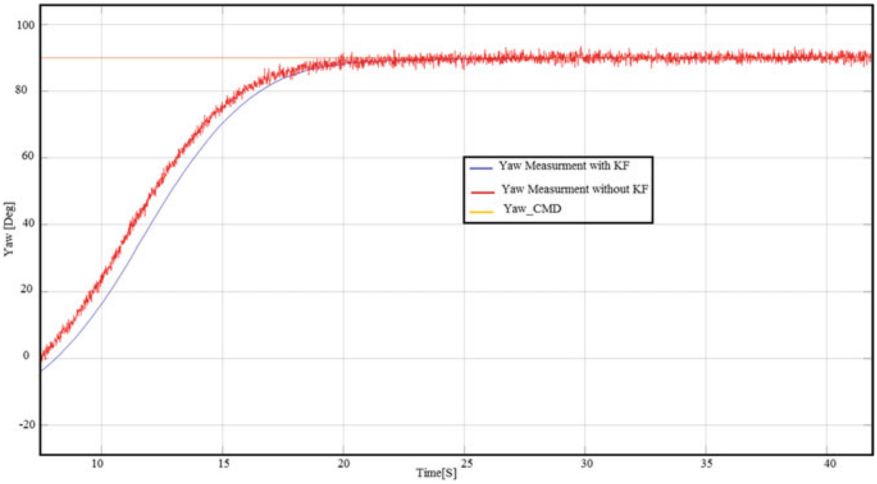


Fig. 10 Yaw response for PID regulators with and without KF (blue) (red)

Table 5 PID effectiveness for Yaw

Regulator	Time rise (s)	Overshoot (%)	Offset (m)	Settling time (s)
PID without KF	4	0	0.8	15
PID with KF	4	0	0.01	15

4.2 Iteration 2: Trajectory Test

After validating all controller settings for the four maneuvers and testing separately, the system is now tested using a trajectory generator with aggressive maneuvers in all directions, using white Gaussian noise assumed for the factory and the measurement detector. The system responses for the PID regulator with KF during a continuous time simulation of 300 s are shown in Fig. 11 using Simulink. In accordance with the sequences, each lasting fifty seconds, the throttle (z), pitch (phi), and roll (theta) commands were used:

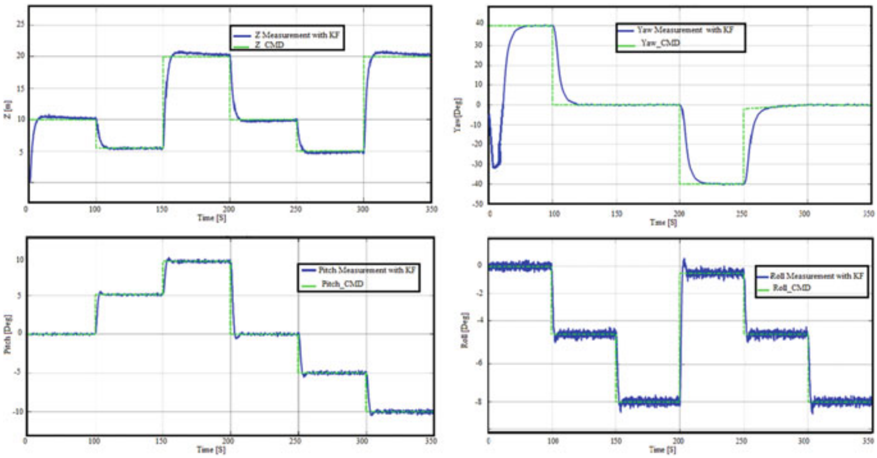


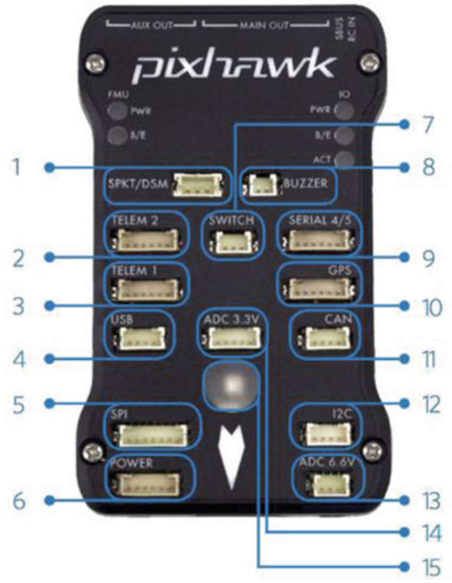
Fig. 11 Simulation: Investigation of the PID and Kalman Filter for complete time sequence. Reference (green), Output (blue)

- Z = 10 m, Pitch = 0°, Roll = 0°, and Yaw = 45;
- Z = 5 m, Pitch = 5°, Roll = −5°, and Yaw = 0;
- Z = 20 m, Pitch = 10°, Roll = −10°, and Yaw = 0;
- Z = 10 m, Pitch = −5°, Roll = 5°, and Yaw = −45;
- Z = 5 m, Pitch = −10°, Roll = 10°, and Yaw = 0;
- Z = 0 m, Pitch = 0°, Roll = 0°, and Yaw = 0.

Despite aggressive movements, the system displays a satisfactory response and follows instructions. This is an excellent evaluation of the four responses of the settings Attitude, Pitch Roll, and Yaw. Using the errors computed by the R and Q diagonal matrices, the KF acts as an ideal observer by accurately predicting the updated values of the states.

5 The Autopilot PIXHAWK's Description

A PIXHAWK autopilot module is an effective on-board autopilot module that can be integrated with a wide range of unmanned aerial vehicles, unmanned ground vehicles, ships, and other moving robotic platforms [19, 55]. The PX4IO (Airplane/Rover Servo and I/O Module) is a single board that integrates the autopilot and flight management unit. There are several platforms that use this technology, including those aimed at high-end research, amateurism, and industry. In the case of the autopilot, it is an integrated board that is controlled by the NuttX Real Time Operating System (RTOS). On a Linux platform, it enables multitasking and programming. The system utilizes ST Microelec-



1	Spektrum DSM receiver
2	Telemetry (radio telemetry)
3	Telemetry (on-screen display)
4	USB
5	SPI (serial peripheral interface) bus
6	Power module
7	Safety switch button
8	Buzzer
9	Serial
10	GPS module
11	CAN (controller area network) bus
12	I²C splitter or compass module
13	Analog to digital converter 6.6 V
14	Analog to digital converter 3.3 V
15	LED indicator

Fig. 12 PIXHAWK flight regulator connector description

Table 6 PID effectiveness for Yaw

Central section	Specifications
Processor	32bit STM32F427 CortexM4 core with FPU
	168 MHz, 256 KB RAM, and 2 MB Flash
	32-bit STM32F103 failsafe co-processor
Detectors section	ST Micro L3GD20H 16 bit gyroscope
	ST Micro LSM303D 14 bit accelerometer/magnetometer
	Invensense MPU 6000 3-axis accelerometer/gyroscope
	MEAS MS5611 barometer
Interface section	5x UART (serial ports), one high-power capable, 2x with HW flow control
	2x CAN (one with internal 3.3V transceiver, one on expansion connector)
	Spektrum DSM/DSM2/DSM-X® Satellite compatible input
	Futaba S.BUS® compatible input and output
	PPM sum signal input and RSSI (PWM or Tension) input
	I2C and SPI serial communication
	3.3 and 6.6 V ADC inputs
	Internal microUSB port and external microUSB port extension

tronics® detectors and ARM Cortex M4 CPUs. Further, it allows the creation and upload of C/C++ programs to the Pixhawk [19, 53] (Fig. 12).

Table 6 supplied a description of the PIXHAWK controller's full specification.

6 Software Development

In this section, we describe the program that has been created to communicate with Pixhawk's autopilot material so that the code for the autopilot can be created using Matlab/Simulink. In order to model the quadcopter regulator for this operation, Matlab/Simulink was used. In order to integrate the entire developed model into the PX4 autopilot, the Pixhawk support software package is required. Aside from that, the requirements for the program to be used and specifications favor the Linux platform over the Windows operating system. Consequently, the programming development for this operation has been divided into four parts: the installation of Matlab R2016b, the installation of the operating system's dependencies, Ubuntu 16.04 LTS, and the installation of Pixhawk's support package. This version is compatible with the Matlab R2016b used to compile the PX4 Firmware [53].

6.1 Matlab/SIMULINK Environment

In the Simulink environment, a quadcopter dynamic model with six degrees of freedom was developed and a linear dynamic inverse regulator was tested. In addition, it can be used as a means of simulating and interacting with material, thus enabling the user to assess the effectiveness of the regulator and to adjust its control settings more precisely. In addition, Simulink models can provide an exact simulation of the dynamic vehicle with 3D animations to visualize the operations due to its high mathematical calculation operation [53]. To achieve all of these operations, the Pixhawk FMUv2 requirements specify which versions of the MATLAB/SIMULINK toolboxes must be used: Aerospace Blockset; Embedded Coder; Simulink Coder; MATLAB R2016b.

6.2 Operational System Dependencies

This application was developed using the PX4 autopilot supplier's recommended operating system, Ubuntu Linux 16.04 LTS, as it makes it possible to develop Linux-based material, simulation software, ROS, and Pixhawk Target NuttX-based material. The official Pixhawk's support package for installation, instructions and the platform open terminal were utilized to install the requirements. As a result, the software requires Git, CMake 3.5.1, arm-none-eabi-gcc (GNU Tools for ARM Embedded Processors), Python 2.7.12 with the Additional Package, and GCC ARM compiler version 4.8.

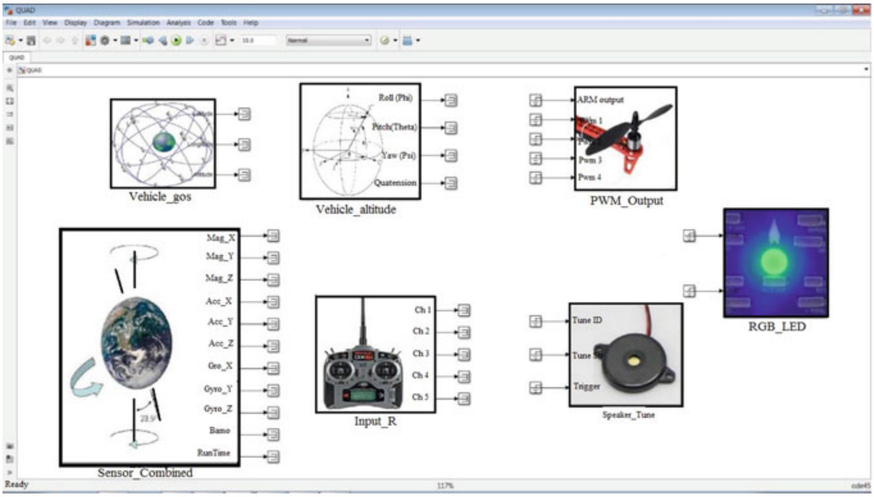


Fig. 13 Pixhawk block toolset in MATLAB/SIMULINK

6.3 *Pixhawk Support Package (PSP)*

Pixhawk's support package is used because it is simple to integrate the Simulink model into the px4fmu Version 2 software that is the main tool for generating the code for the Pixhawk-FMUv2 for the Simulink Flight Management Unit models. With the inclusion of Pixhawk Simulink block-set to the Library, Fig. 13, this makes the system easily realizable. Additionally, it permits communication with the built-in inertial detectors on the board, GPS, an LED, a PWM output, and serial Rx/Tx [19]. A Flight Management Unit, FMU, Firmware is built by copying the forked version from the Pixhawk Website developer into the designated folder using the Ubuntu open terminal command. Based on this procedure, the Firmware is created in accordance with the version selected for this study, px4 f mu-v2default, which was recommended by MathWorks developers because it represents the default configuration.

7 Analytical 6DOF Quadcopter Control Model Using MatLab/Simulink

In this study, the objective was to design analytical models that would enable the construction of a linearized control system that incorporates a Kalman filter in the feedback system so that a genuine predictor of the impact of internal and external noise on the feedback signal could be developed. The Matlab/Simulink models generated [12] are used to extract the true values of the propulsion system and utilized in the analysis to test quadcopter Fig. 14 Pixhawk block toolset in Matlab/Simulink.

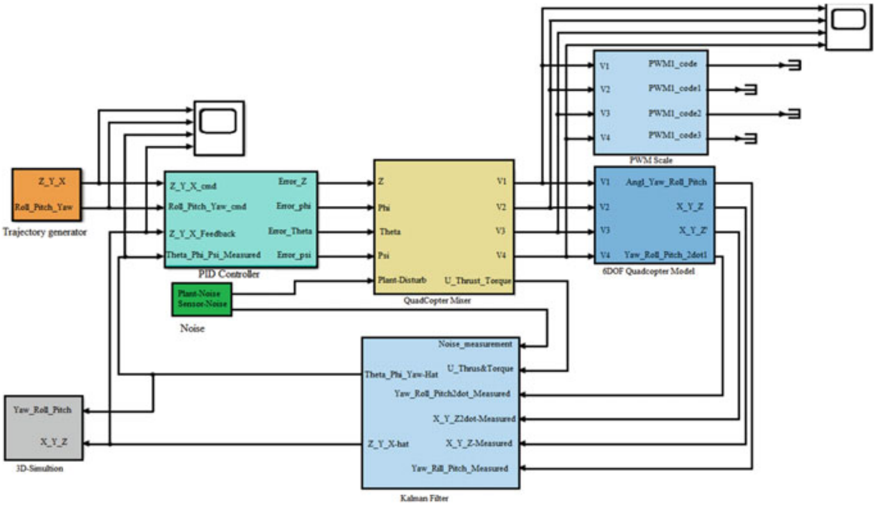


Fig. 14 A 6DOF quadcopter Simulink model with control algorithms as a complete

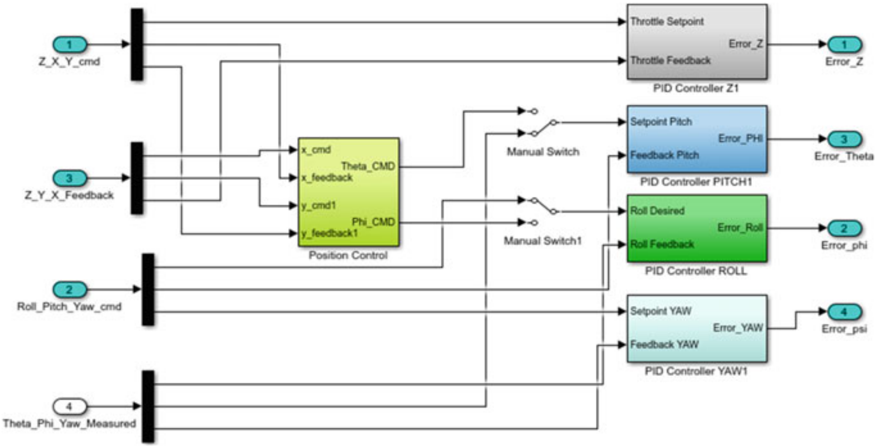


Fig. 15 Block for the PID regulator from MATLAB SIMULINK

7.1 Modelling of Flight Corrector

This model includes a quad-mixer, scaling blocks, and a PID controller. It is necessary to calculate the required motor voltages for the intended command and compare them to the actual measurements in order to stabilize the quadcopter [12]. A screenshot of this Simulink implementation may be seen in Fig. 15.

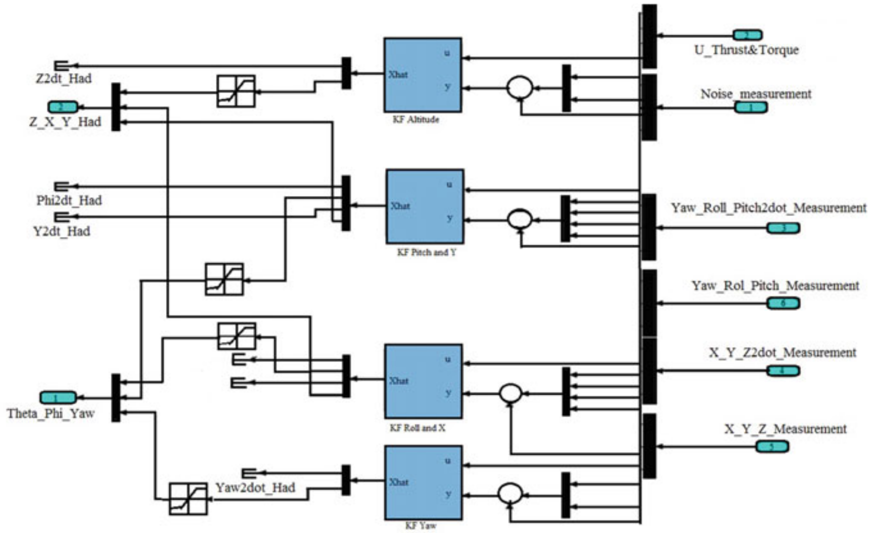


Fig. 16 Simulink block diagram for the Kalman filter for height, pitch, roll, and yaw

7.2 Modelling of Kalman Filter

By estimating the quadcopter's true output using Simulink, the SKF Simulink model attempts to filter all noise encountered during processing. One of the critical phases in this application is the implementation of the KF in Simulink, Fig. 16. The filter is created by utilizing the KF block and block input settings from Simulink's monitoring system Toolbox package [12].

In the constructed KF, the primary settings that determine how the filter behaves are the initial estimate of state vector X[0] and the covariance matrices R and Q [12] in the constructed KF. Each movement of the quadcopter requires an independent filter implementation which takes into account all system-related parameters (such as moments of inertia, mass, detector biases). For the purpose of verifying the KF's effectiveness, the effectiveness specifications were taken from a real commercialized quadcopter used in developing Simulink models [12].

7.3 Mixer for a Quadcopter

As a result of the quadcopter mixer block being constructed in two steps, the first of which uses specifications obtained from a genuine quadcopter, including fixed torque, drag, and the length of the quad arms [12]. The second step involves applying linear regression to determine the relationship between the Tension [V] signals given to the

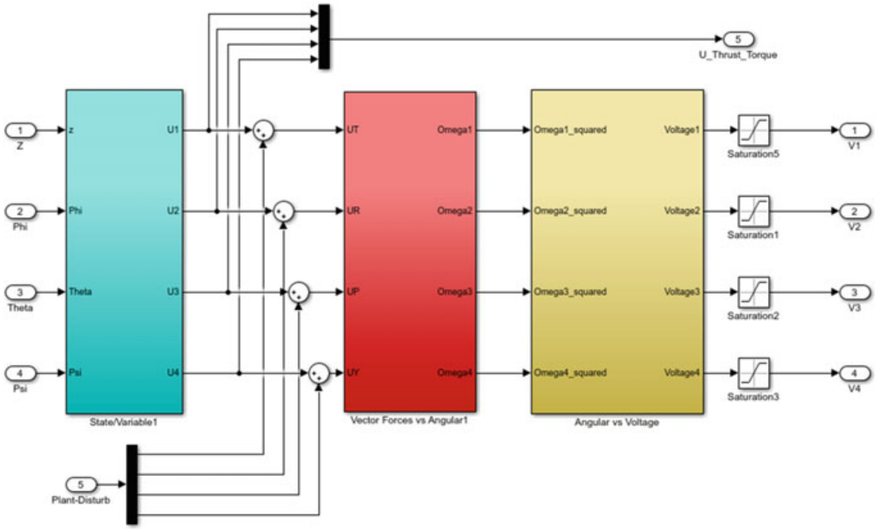


Fig. 17 Quadcopter mixer SIMULINK block

motors and the actual squared angular speed Ω^2 [rad/s]^2. Figure 17 depicts a quick image of the Simulink scheme for the quad-mixer.

7.4 Scaling of the PWM

There have been several blocks that have been used to scale the required voltages for the commands and transform them into 16-bit integers for use in the MCU's PWM output. The linear characteristic coefficient, shown in Fig. 18, is used to represent a snapshot of the block in the scaling linearity.

8 Testing of the System

As soon as the designed regulator's 3D animation was successfully simulated during testing, it was time to begin the actual execution of the regulator. The application building developed in Simulink and download into the PIXHAWK is activated with the necessary data that is needed to construct the regulator designed is shown in Fig. 19, using both sources Pixhawk Website developer and MathWorks developer [12].

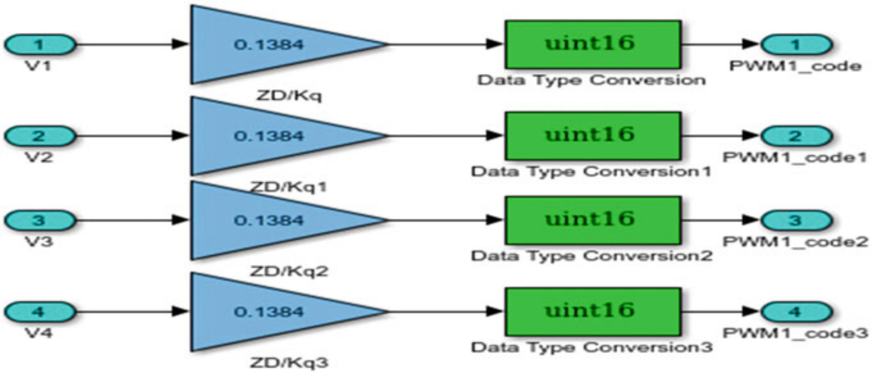
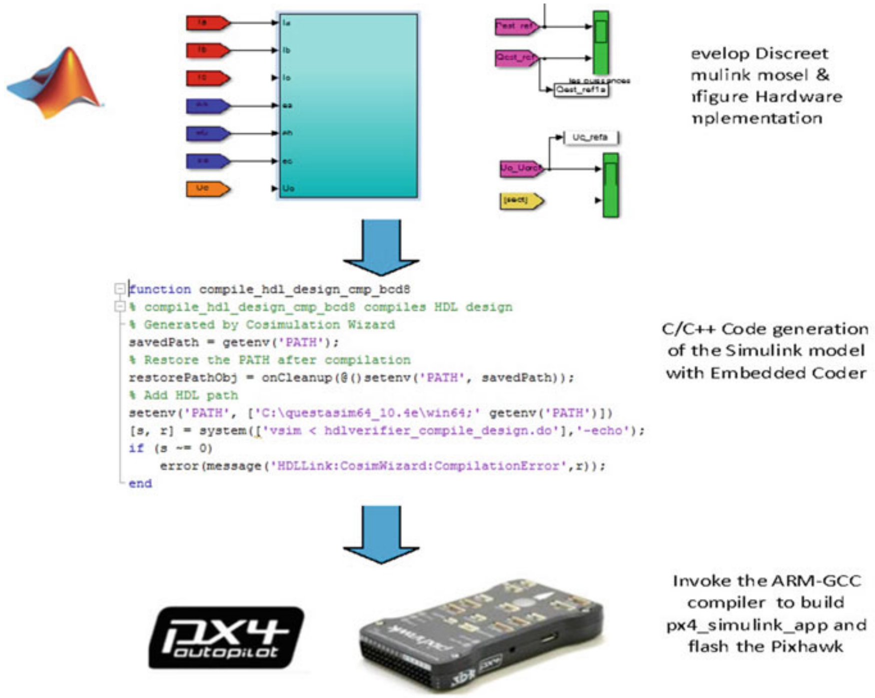


Fig. 18 Scaling PWM SIMULINK block

8.1 Setting of the Software

Simulink regulators were designed and connected to Pixhawk's Simulink target block tools after the entire model had been discretized. The following is a description of the primary Pixhawk Target block:

- Input-rc block:
  The command instructions from the Radio transmitter, which convey the commands for Height, Pitch, Roll, and Yaw, are accessible, Fig. 20. The 5 Channels, Ch1-Ch5, are set up to create the signal for the Height, Pitch, Roll, and Yaw, and RGB LEDs, respectively, in order to link this block. Unsigned Integer 16-bit PWM is utilized to express all the transmitter's data [12].
- PWM Output block:
  Figure 21 demonstrates how the regulator may communicate motor commands to the ESC, which regulates motor velocity, in unsigned 16-bit format at PWM a frequency rates 50, 125, 250, 300, 400 Hz [12].
- detector combined block:
  Access to the several detectors that are accessible on the Pixhawk Board material is made clear by Fig. 22. The quadcopter's altitude is controlled using the barometer signal included in on barometric pressure [12].
- vehicle attitude block:
  The Roll, Pitch, and Yaw may be calculated for the drone's attitude using the combination shown in Fig. 23 [12].
- Drone's controller interface:
  The interface between the sub-blocks and the quadcopter regulator is shown in Fig. 24. The planned continuous system is digitalized at a rate of 4 ms utilizing the zero-order hold.



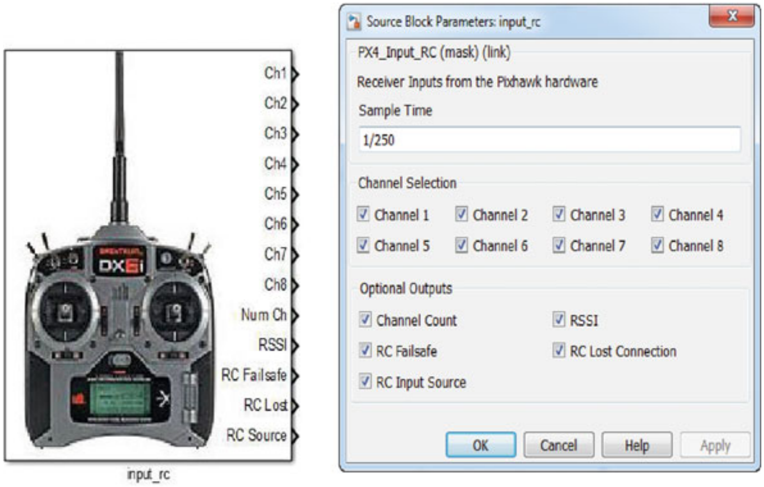
evelop Discreet
mulink mosel &
ıfigure Hardware
nplementation

```
function compile_hdl_design_cmp_bcd8
% compile_hdl_design_cmp_bcd8 compiles HDL design
% Generated by Cosimulation Wizard
savedPath = getenv('PATH');
% Restore the PATH after compilation
restorePathObj = onCleanup(@()setenv('PATH', savedPath));
% Add HDL path
setenv('PATH', ['C:\questasim64_10.4e\win64;' getenv('PATH')])
[s, r] = system(['vsim < hdlverifier_compile_design.do'],'-echo');
if (s ~= 0)
    error(message('HDLLink:CosimWizard:CompilationError',r));
end
```

C/C++ Code generation
of the Simulink model
with Embedded Coder

Invoke the ARM-GCC
compiler to build
px4_simulink_app and
flash the Pixhawk

Fig. 19 Block diagram for the implementation procedures



Source Block Parameters: input_rc

PX4_Input_RC (mask) (link)

Receiver Inputs from the Pixhawk hardware

Sample Time

1/250

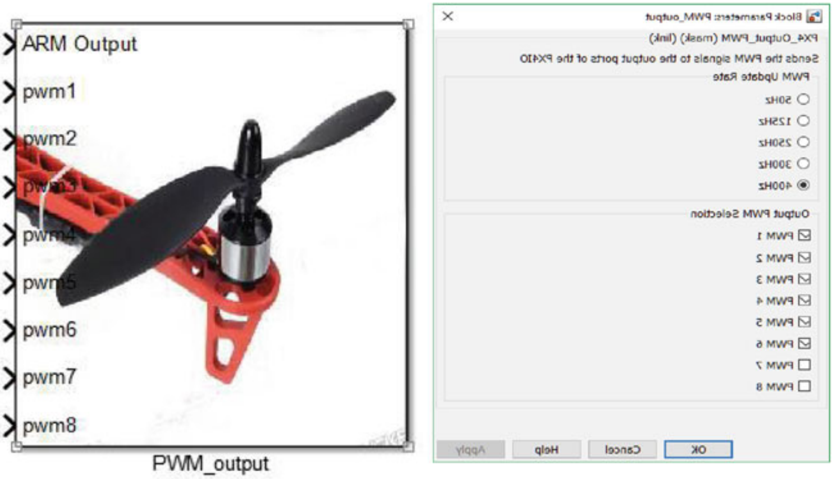
Channel Selection

☑ Channel 1 ☑ Channel 2 ☑ Channel 3 ☑ Channel 4
☑ Channel 5 ☑ Channel 6 ☑ Channel 7 ☑ Channel 8

Optional Outputs

☑ Channel Count ☑ RSSI
☑ RC Failsafe ☑ RC Lost Connection
☑ RC Input Source

Fig. 20 Radio control SIMULINK block



Block Parameters: PWM_output

PX4_Output_PWM (mask) (link)

Sends the PWM signals to the output ports of the PX4IO

PWM Update Rate

○ 50Hz
○ 125Hz
○ 250Hz
○ 300Hz
● 400Hz

Output PWM Selection

☑ PWM 1
☑ PWM 2
☑ PWM 3
☑ PWM 4
☑ PWM 5
☑ PWM 6
□ PWM 7
□ PWM 8

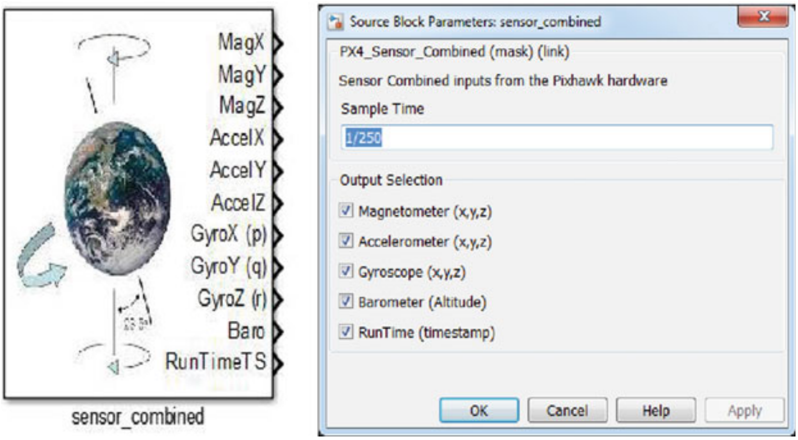
Fig. 21 PWM SIMULIK generator for Pixhawk

8.2 *Program Creation Procedure*

In order to target the Pixhawk PX4 and create the required C/C++ program, the Simulink model is set up using the appropriate program generation settings in the material application page of the Pixhawk PX4 as the material board target. Using Build Tool Integration, Fig. 25 on the Simulink command window, the relevant code is generated automatically. Matlab can call the ARM-GCC compiler through the Build Tool Integration (BTI) feature of MathWorks' ert.tlc, which is included with Embedded Coder [51], and must be utilized as the system target file. The user is then able to select the material and toolchain. When the target material is set to Pixhawk, the appropriate toolchain (Pixhawk) will be automatically selected.

8.3 *Testing of Pixhawk*

After the detection calibration and the setting of all Simulink model regulator configurations, the program was successfully down-loaded to the Pixhawk autopilot [12], Fig. 26. For the execution of the Matlab/Simulink model part of the board, a setup script is required and the script must be transferred to the micro-SD card used by the PX4 to run the px4simulinka pp. A customized version of the program has been developed for this application and is taken from the Pixhawk Support MATLAB Guide [12]. PWM signals were measured using the test quad and scope set up for the altitude corrector test.



Source Block Parameters: sensor_combined

PX4_Sensor_Combined (mask) (link)

Sensor Combined inputs from the Pixhawk hardware

Sample Time

1/250

Output Selection

☑ Magnetometer (x,y,z)

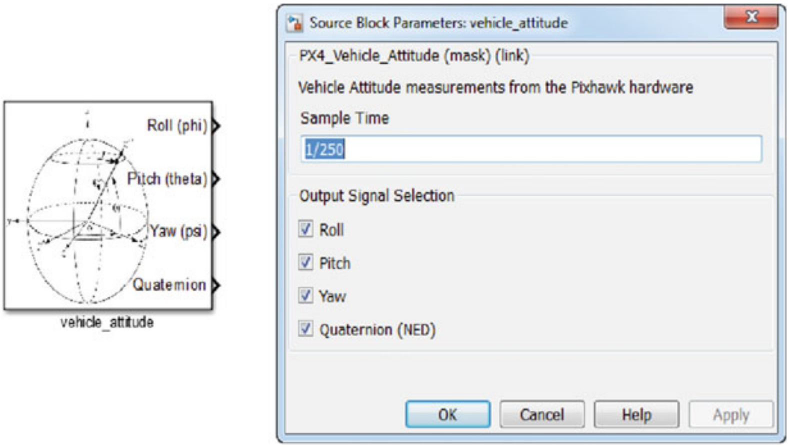
☑ Accelerometer (x,y,z)

☑ Gyroscope (x,y,z)

☑ Barometer (Altitude)

☑ RunTime (timestamp)

Fig. 22 Detector combined block



Source Block Parameters: vehicle_attitude

PX4_Vehicle_Attitude (mask) (link)

Vehicle Attitude measurements from the Pixhawk hardware

Sample Time

1/250

Output Signal Selection

☑ Roll

☑ Pitch

☑ Yaw

☑ Quaternion (NED)

Fig. 23 The Pixhawk's detector block

9 Results and Simulation of the Drone

The model shown in Fig. 27 consists of a complete Matlab/Simulink model based on a 6-DOF quad-copter model, KF, PID regulator, and quad-copter motor mixer. The White Gaussian noise is estimated to be about 5% in the model. Therefore, numerous experiments have been conducted in order to obtain the results. Figure 28 uses Simulink to simulate a system for 300 s while displaying the roll, throttle, and pitch responses for the PID regulator with KF (blue) and without KF (red). In the following sequences, each lasting 50 s, the following commands were set for throttle (z), pitch (phi), and roll (theta):

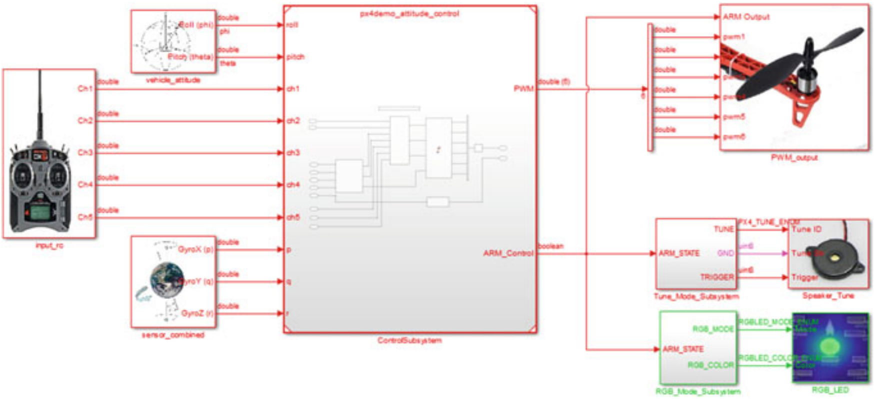


Fig. 24 Quad regulator link with the Pixhawk SIMULINK block set [12]

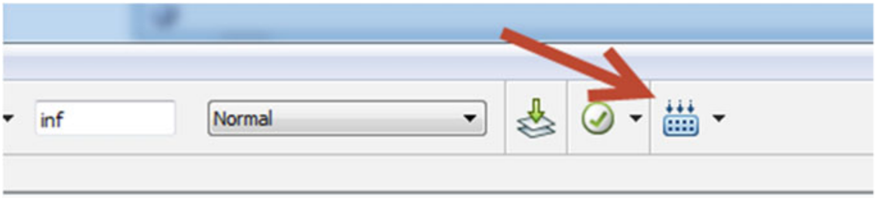
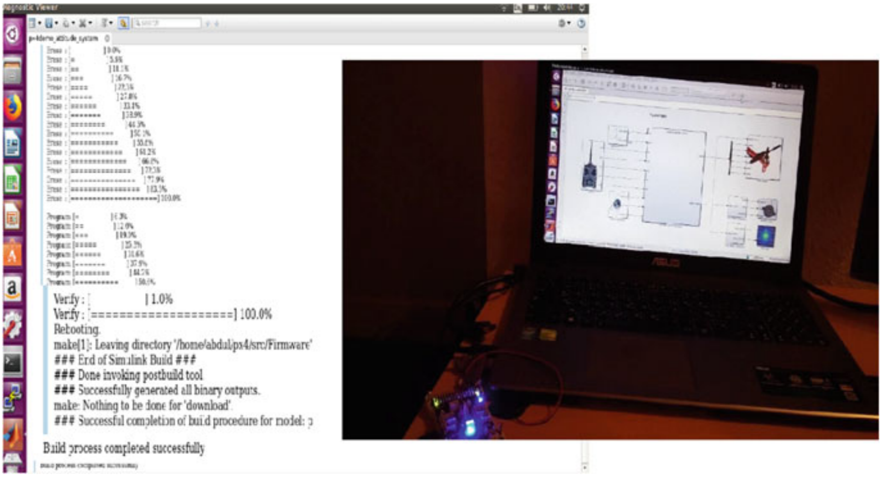


Fig. 25 Design the embedded icon from the Simulink window [12]



```
Erase : [                   ] 0.0%
Erase : [=                  ] 5.5%
Erase : [==                 ] 11.1%
Erase : [===                ] 16.7%
Erase : [====               ] 22.3%
Erase : [=====              ] 27.8%
Erase : [======             ] 33.3%
Erase : [=======            ] 38.9%
Erase : [========           ] 44.3%
Erase : [==========         ] 50.1%
Erase : [===========        ] 55.6%
Erase : [============       ] 61.2%
Erase : [=============      ] 66.6%
Erase : [==============     ] 72.3%
Erase : [===============    ] 77.9%
Erase : [================   ] 83.3%
Erase : [===================] 100.0%

Program: [=         ] 6.3%
Program: [==        ] 12.6%
Program: [===       ] 18.9%
Program: [=====     ] 25.2%
Program: [======    ] 31.6%
Program: [=======   ] 37.9%
Program: [========  ] 44.2%
Program: [==========] 50.5%
 Verify : [                    ] 1.0%
 Verify : [====================] 100.0%
 Rebooting.
 make[1]: Leaving directory '/home/abdul/px4/src/Firmware'
 ### End of Simulink Build ###
 ### Done invoking postbuild tool
 ### Successfully generated all binary outputs.
 make: Nothing to be done for 'download'.
 ### Successful completion of build procedure for model: p

Build process completed successfully
Build process completed successfully
```

Fig. 26 Photo taken while downloading the SIMULINK model into the Pixhawk [12]

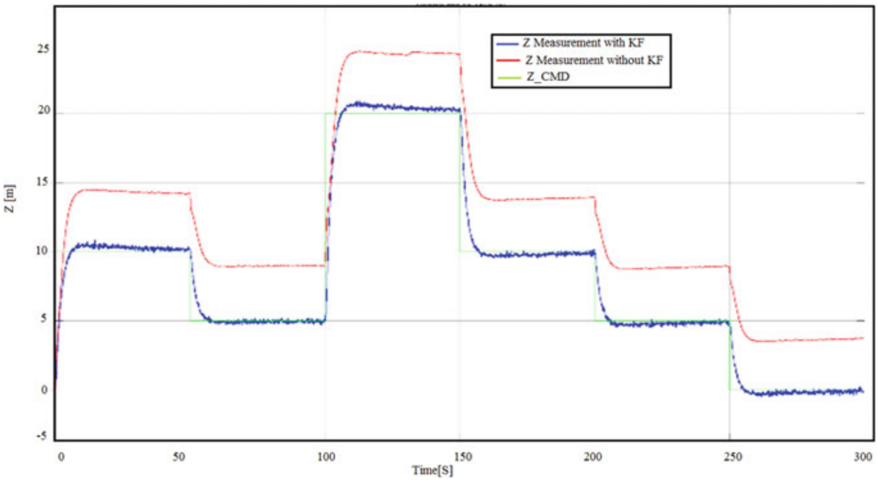


Fig. 27 The PID controller's throttle response with and without KF

- Z = 10 m, Phi = 0°, and Theta = 0°;
- Z = 5 m, Phi = 10°, and Theta = −10°;
- Z = 20 m, Phi = 20°, and Theta = −20°;
- Z = 10 m, Phi = 20°, and Theta = −20°;
- Z = 5 m, Phi = 10°, and Theta = −10°;
- Z = 0 m, Phi = 0°, and Theta = 0°.

In spite of forceful maneuvers, the system exhibits good tracking of the command and accurate estimation of the three responses for altitude, roll, and pitch. A slight divergence is observed in the response without KF as a result of noise and irregularities in the measurement process. Using the PID settings indicated in Table 2, the effectiveness of the system with KF is determined.

10 Conclusion

Using the Matlab/Simulink software, the controller was implemented using PID augmented with KF, along with the 6-DOF dynamic model. Testing was performed using time-varying trajectories and a white Gaussian noise. In the course of the test, a comparison has been made between the PID control with KF and the PID control without KF. In order to demonstrate the effectiveness of the controller using serve maneuverability, graphic plot representations of the command and 3D simulations were established. This phase consisted of providing unique flight control commands using the Simulink model controller created for the material autopilot (Pixhawk). Through the build in command in Simulink, the entire controller design was translated into C/C++ code and uploaded to the board. Using the chosen method of control

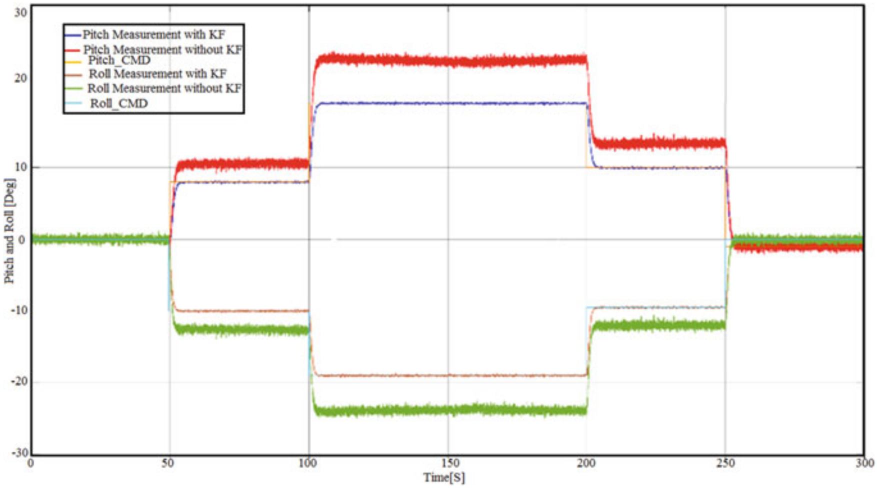


Fig. 28 The PID controller's throttle response with and without KF

design for this study, we were able to validate the experimental methods we developed to simplify the control and identify the quadrotor's linear zone that was utilized to reduce power intake and assure maximum control. Through the use of Embedded Coder® generator, the control system developed using Simulink could be converted and integrated into the available Flight Boards, such as Pixhawk Pilot Support Package autopilot. For any future work, the following points will be taken into account in order to design a monitoring system for a quadcopter based on simple approaches extracted from experiment.

- The physical and aerodynamic actuators of a quadrotor model developed using a faithful CAD model should be identified.
- Consider alternative control techniques instead of PID controllers, such as infinity control, backstepping control, slide mode control, artificial neural network, and linear quadratic techniques.

Acknowledgements The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for supporting this work. Special acknowledgement to Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

References

1. Aboelhassan A, Abdelgeliel M, Zakzouk EE, Galea M (2020) Design and implementation of model predictive control based PID controller for industrial applications. Energies 13(24):6594
2. Ajel AR, Humaidi AJ, Ibraheem IK, Azar AT (2021) Robust model reference adaptive control for tail-sitter VTOL aircraft. Actuators 10(7):1–19. https://doi.org/10.3390/act10070162
3. Al-Qassar A, Abdulkareem A, Hasan A, Humaidi A, Ibraheem I, Azar A, Hameed A (2021) Grey-wolf optimization better enhances the dynamic performance of roll motion for tail-sitter VTOL aircraft guided and controlled by STSMC. J Eng Sci Technol 16(3):1932–1950
4. Alkhafaji FS, Hasan WW, Isa M, Sulaiman N (2018) A novel method for tuning PID controller. J Telecommun Electron Comput Eng (JTEC) 10(1–12):33–38
5. Ammar HH, Azar AT (2020) Robust path tracking of mobile robot using fractional order PID controller. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). Advances in intelligent systems and computing, vol 921. Springer International Publishing, Cham, pp 370–381
6. Ammar HH, Azar AT, Tembi TD, Tony K, Sosa A (2018) Design and implementation of fuzzy PID controller into multi agent smart library system prototype. In: The international conference on advanced machine learning technologies and applications (AMLTA2018). Advances in intelligent systems and computing, vol 723. Springer International Publishing, Cham, pp 127–137
7. Azar AT, Serrano FE, Kamal NA, Koubaa A (2020) Robust kinematic control of unmanned aerial vehicles with non-holonomic constraints. In: International conference on advanced intelligent systems and informatics. Springer, pp 839–850
8. Azar AT, Serrano FE, Koubaa A, Kamal NA (2020) Backstepping h-infinity control of unmanned aerial vehicles with time varying disturbances. In: 2020 first international conference of smart systems and emerging technologies (SMARTTECH). IEEE, pp 243–248
9. Azar AT, Koubaa A, Ali Mohamed N, Ibrahim HA, Ibrahim ZF, Kazim M, Ammar A, Benjdira B, Khamis AM, Hameed IA et al (2021) Drone deep reinforcement learning: a review. Electronics 10(9):999
10. Azar AT, Serrano FE, Kamal NA, Koubaa A, Ammar A (2021) Dynamic integral PID sliding mode attitude-position control of unmanned aerial vehicles. In: Advanced machine learning technologies and applications. Springer International Publishing, Cham, pp 651–661
11. Borase RP, Maghade D, Sondkar S, Pawar S (2021) A review of PID control, tuning methods and applications. Int J Dyn Control 9(2):818–827
12. Bulut N (2019) Modeling, simulation, and control of a quadrotor having a 2-DOF robotic arm. Master's thesis, Middle East Technical University
13. Cano AEJ (2019) Modelling and control of aerial manipulators. PhD thesis, Universidad de Sevilla
14. Cols Margenet M, Schaub H, Piggott S (2021) Flight software development, migration, and testing in desktop and embedded environments. J Aerosp Inf Syst 18(4):157–174
15. DeGarmo MT (2004) Issues concerning integration of unmanned aerial vehicles in civil airspace. Center for Advanced Aviation System Development, 4
16. Erenturk K, Erenturk S (2022) Enhanced fractional-order piλdμ control for a forced circulation evaporator system via advanced disturbance observer. ISA Trans
17. Fadel MZ, Rabie MG, Youssef AM (2019) Modeling, simulation and control of a fly-by-wire flight control system using classical PID and modified PI-D controllers. J Eur Syst Autom 52(3):267–276
18. Fareha A, Bousbaine A, Josaph AK (2018) Experimental characterisation of quad rotor controller based on Kalman filter. In: 2018 53rd international universities power engineering conference (UPEC). IEEE, pp 1–6
19. Fareha A, Bousbaine A, Josaph AK (2020) A hardware implementation of 6dof quadcopter Matlab/Simulink controller algorithm to an autopilot. In: The 10th international conference on power electronics, machines and drives (PEMD 2020), vol 2020, pp 485–490. https://doi.org/10.1049/icp.2021.1078

20. Fekik A, Denoun H, Azar AT, Koubaa A, Kamal NA, Zaouia M, Hamida ML, Yassa N (2020) Adapted fuzzy fractional order proportional-integral controller for dc motor. In: 2020 first international conference of smart systems and emerging technologies (SMARTTECH), pp 1–6. https://doi.org/10.1109/SMART-TECH49988.2020.00019
21. Femi R, Sree Renga Raja T, Shenbagalakshmi R (2022) Performance comparison of optimization algorithm tuned PID controllers in positive output re-lift Luo converter operation for electric vehicle applications. IETE J Res 1–19
22. Filo M, Kumar S, Khammash M (2022) A hierarchy of biomolecular proportional-integral-derivative feedback controllers for robust perfect adaptation and dynamic performance. Nat Commun 13(1):1–19
23. Gorripotu TS, Samalla H, Jagan Mohana Rao C, Azar AT, Pelusi D (2019) TLBO algorithm optimized fractional-order PID controller for AGC of interconnected power system. In: Nayak J, Abraham A, Krishna BM, Chandra Sekhar GT, Das AK (eds) Soft computing in data analytics. Springer, Singapore, pp 847–855
24. Gundlach J, Gundlach J (2012) Designing unmanned aircraft systems: a comprehensive approach, vol 34. American Institute of Aeronautics and Astronautics Reston
25. Guo K, Ye Z, Liu D, Peng X (2021) UAV flight control sensing enhancement with a data-driven adaptive fusion model. Reliab Eng Syst Saf 213(107):654
26. Hanani N, Syazwanadira F, Fakharulrazi NA, Yakub F, Rasid ZA, Sarip S (2019) Full control of quadrotor unmanned aerial vehicle using multivariable proportional integral derivative controller. In: 2019 IEEE 9th international conference on system engineering and technology (ICSET). IEEE, pp 447–452
27. Humaidi AJ, Najem HT, Al-Dujaili AQ, Pereira DA, Ibraheem IK, Azar AT (2021) Social spider optimization algorithm for tuning parameters in PD-like interval type-2 fuzzy logic controller applied to a parallel robot. Meas Control 54(3–4):303–323
28. Ibraheem GAR, Azar AT, Ibraheem IK, Humaidi AJ (2020) A novel design of a neural network-based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization. Complexity 2020:1–18
29. Kazim M, Azar AT, Koubaa A, Zaidi A (2021) Disturbance-rejection-based optimized robust adaptive controllers for UAVs. IEEE Syst J 15(2):3097–3108. https://doi.org/10.1109/JSYST.2020.3006059
30. Kumar J, Azar AT, Kumar V, Rana KPS (2018) Design of fractional order fuzzy sliding mode controller for nonlinear complex systems. In: Mathematical techniques of fractional order systems. Advances in nonlinear dynamics and chaos (ANDC). Elsevier, pp 249–282
31. Lewis FL, Dawson DM, Abdallah CT (2003) Robot manipulator control: theory and practice. CRC Press
32. Liang S, Xu B, Ren J (2021) Kalman-filter-based robust control for hypersonic flight vehicle with measurement noises. Aerosp Sci Technol 112(106):566
33. López LFdM, García FS, Naranjo Hernández JE, Blas NG (2021) Speed proportional integrative derivative controller: optimization functions in metaheuristic algorithms. J Adv Transp 2021
34. Lu F, Gao T, Huang J, Qiu X (2019) A novel distributed extended Kalman filter for aircraft engine gas-path health estimation with sensor fusion uncertainty. Aerosp Sci Technol 84:90–106
35. Luo Y, Awal M, Yu W, Husain I (2021) FPGA implementation for rapid prototyping of high performance voltage source inverters. CPSS Trans Power Electron Appl 6(4):320–331
36. Najm AA, Ibraheem IK, Azar AT, Humaidi AJ (2020) Genetic optimization-based consensus control of multi-agent 6-DOF UAV system. Sensors 20(12):3576
37. Najm AA, Azar AT, Ibraheem IK, Humaidi AJ (2021) A nonlinear PID controller design for 6-DOF unmanned aerial vehicles. In: Koubaa A, Azar AT (eds) Unmanned aerial systems. Advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 315–343
38. Palaniyappan T, Yadav V, Tayal VK, Choudekar P et al (2018) PID control design for a temperature control system. In: 2018 international conference on power energy, environment and intelligent control (PEEIC). IEEE, pp 632–637

39. Pilla R, Azar AT, Gorripotu TS (2019) Impact of flexible AC transmission system devices on automatic generation control with a metaheuristic based fuzzy PID controller. Energies 12(21):4193
40. Pilla R, Botcha N, Gorripotu TS, Azar AT (2020) Fuzzy PID controller for automatic generation control of interconnected power system tuned by glow-worm swarm optimization. In: Nayak J, Balas VE, Favorskaya MN, Choudhury BB, Rao SKM, Naik B (eds) Applications of robotics in industry using advanced mechanisms. Springer International Publishing, Cham, pp 140–149
41. Pilla R, Gorripotu TS, Azar AT (2021) Design and analysis of search group algorithm-based PD-PID controller plus redox flow battery for automatic generation control problem. Int J Comput Appl Technol 66(1):19–35
42. Pilla R, Gorripotu TS, Azar AT (2021) Tuning of extended Kalman filter using grey wolf optimisation for speed control of permanent magnet synchronous motor drive. Int J Autom Control 15(4–5):563–584
43. Rana K, Kumar V, Sehgal N, George S, Azar AT (2021) Efficient maximum power point tracking in fuel cell using the fractional-order PID controller. In: Azar AT, Kamal NA (eds) Renewable energy systems. Advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 111–132
44. Saidi SM, Mellah R, Fekik A, Azar AT (2022) Real-time fuzzy-PID for mobile robot control and vision-based obstacle avoidance. Int J Ser Sci Manag Eng Technol 13(1):1–32
45. Sallam OK, Azar AT, Guaily A, Ammar HH (2020) Tuning of PID controller using particle swarm optimization for cross flow heat exchanger based on CFD system identification. In: Proceedings of the international conference on advanced intelligent systems and informatics 2019. Advances in intelligent systems and computing, vol 1058. Springer International Publishing, Cham, pp 300–312
46. Samanta S, Mukherjee A, Ashour AS, Dey N, Tavares JMRS, Abdessalem Karâa WB, Taiar R, Azar AT, Hassanien AE (2018) Log transform based optimal image enhancement using firefly algorithm for autonomous mini unmanned aerial vehicle: an application of aerial photography. Int J Image Graph 18(04):1850019
47. Soliman M, Azar AT, Saleh MA, Ammar HH (2020) Path planning control for 3-omni fighting robot using PID and fuzzy logic controller. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). Springer International Publishing, Cham, pp 442–452
48. Suarez A, Heredia G, Ollero A (2018) Design of an anthropomorphic, compliant, and lightweight dual arm for aerial manipulation. IEEE Access 6:29,173–29,189
49. Suid M, Ahmad M (2021) Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system. ISA Trans
50. Takahashi MD, Fujizawa BT, Lusardi JA, Goerzen CL, Cleary MJ, Carr JP IV, Waldman DW (2022) Comparison of autonomous flight control performance between partial-and full-authority helicopters. J Guid Control Dyn 45(5):885–901
51. Veyna U, Garcia-Nieto S, Simarro R, Salcedo JV (2021) Quadcopters testing platform for educational environments. Sensors 21(12):4134
52. Wæringsaasen S (2021) Tethered quadcopter control, simulation and modeling platform for a small USV. Master's thesis, UiT Norges arktiske universitet
53. Wu Y, Hu K, Sun XM (2018) Modeling and control design for quadrotors: a controlled Hamiltonian systems approach. IEEE Trans Veh Technol 67(12):11,365–11,376
54. Yasin JN, Mohamed SA, Haghbayan MH, Heikkonen J, Tenhunen H, Plosila J (2020) Unmanned aerial vehicles (UAVs): collision avoidance systems and approaches. IEEE Access 8:105,139–105,155
55. Zhang Q, Xu Y, Wang X, Yu Z, Deng T (2021) Real-time wind field estimation and pitot tube calibration using an extended Kalman filter. Mathematics 9(6):646. https://doi.org/10.3390/math9060646. https://www.mdpi.com/2227-7390/9/6/646
56. Zhang Y, Liu L, Peng Y, Liu D (2018) An electro-mechanical actuator motor voltage estimation method with a feature-aided Kalman filter. Sensors 18(12):4190
57. Zolanvari M, Jain R, Salman T (2020) Potential data link candidates for civilian unmanned aircraft systems: a survey. IEEE Commun Surv Tutor 22(1):292–319

Wireless Sensor Network Based Mobile Robot Applications

Muayad Sadik Croock

Abstract The mobile robots and their applications have been widely considered in different fields including industry, military, and civilian functions. This is due to the noted accuracy enhancement in performing the required jobs in using the mobile robots. This chapter emphases the concepts of Wireless Sensor Network (WSN) and the adopted categories as well as the protocols. It also considers the deep-learning in terms of methods and related advantage and disadvantages. Moreover, mobile robotic applications are adopted including mobile robotic wheelchair controlling using gaze direction and Electroencephalography (EEG) signals as well as indoor localization for a mobile robotic car. These applications consider WSN a backbone that is managed by traditional smart and deep-learning based techniques. The deep-learning helps in enhancing the performance of the indoor localization. In addition, prototypes for these applications are introduced using embedded systems in combination with WSN. The presented results show that the claims of the mentioned applications are achieved.

Keywords Robotic car · WSN · WSN protocols · Deep-learning · Gaze direction · EEG

1 Introduction

The speedy development in the living environment in terms of electronic and physical sides encourage the researchers around the world to spend more time in providing modern applications that can offer satisfied services. The researchers focus on using the Wireless Sensor Network (WSN) as a backbone of proposing the current technologies [7]. WSNs are used in different applications for numerous technologies as they prove their efficiency, particularly in monitoring and tracking systems as well as the smart schemes. WSN adopts a number of criteria in designing the structure

M. S. Croock (✉)
University of Technology-Iraq, Baghdad, Iraq
e-mail: muayad.s.croock@uotechnology.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_12

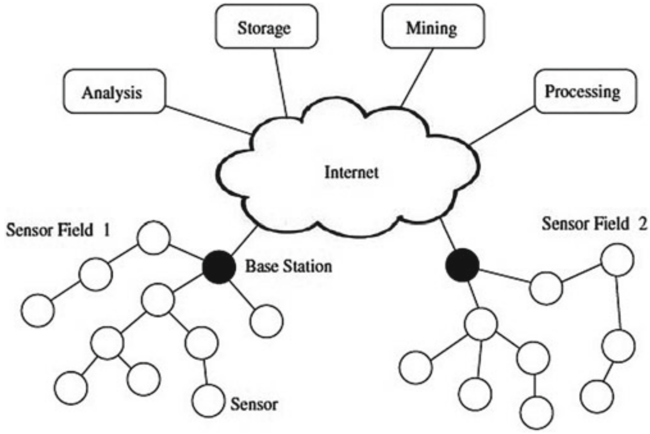


Fig. 1 WSN structure [7]

and protocols for managing the data exchange between nodes within the network [15]. It normally includes different types of sensor nodes, sink node and server or base station that sends the collected information throughout the internet to the cloud for possible processing, such as analysis, storage, data mining and mathematical processing, as shown in Fig. 1 [7]. The selection of the suitable topology is based on the adopted application that needs special management of the exchanged data and information.

In order to increase the fields of WSN applications as well as ensuring the efficiency of performance for them, the intelligent methods are used. Depending on the utilization of these methods, different sides of WSN and related applications are improved. They can be employed to enhance the optimization of the resource allocation, in which the requirements are achieved. Moreover, the intelligent methods are also can be utilized for obtaining the classification of particular problems. They include numerous types, such as deep and machine learning, as well as the Genetic, swarm, etc. Fig. 2 illustrates the work flow of the combination of WSNN and intelligent methods. It shows the two sides of using the intelligent methods and how each side can obtain its own data that is used for performing the function.

The combination of WSN and intelligent methods is commonly used in designing the robots that are employed for different types applications. This is to improve the performance of such robots in terms of accuracy, power efficiency, and so on. WSN works in collecting the information from the surrounding area that is formulated to be understandable for the robot that is responsible on processing this information to take the right decision [21]. They look like the five senses in human to help the robots in different activation. At the other hand, the intelligent methods are used for increasing the performance efficiency of the robots in numerous parts. These parts include the guidance of robots, the accuracy in implementing the required functions. Each part adopts a particular type of intelligent methods depending on the requirements and

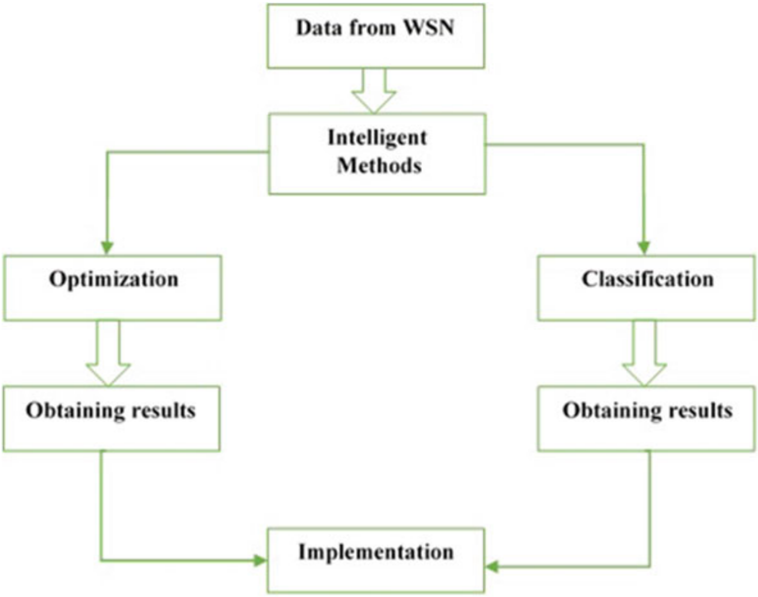


Fig. 2 WSN with intelligent methods

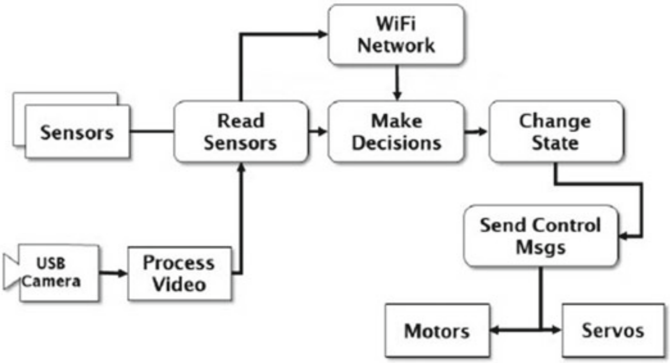


Fig. 3 General diagram of robots [14]

applications. Figure 3 shows the general block diagram of a robot in terms of sensors and actuators to explain the working steps. The intelligent side can be performed in the making decision block as well as the video processing [14].

The robots are widely used in modern life fields and in other different applications, such as industrial machines, discovery missions, guiding and driving vehicles and airplanes, etc. The employment of these robots can be played an important role in replacing the human efforts with high accuracy and efficiency. This is to overcome the human errors and mistakes and come up with massive with high quality productions.

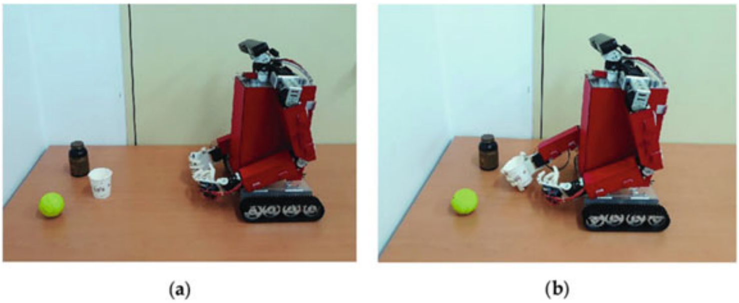


Fig. 4 Object recognition using robot [28]

The robots with WSN and intelligent methods are particularly used in applications with high accuracy, such as medical surgery and microelectronic devices made. These applications suffer from human errors and might take a long time to be in the required measurements. Thus, the intelligent and smart robots are the modern and coming trends in the scientific and applied life style. Figure 4 explains one of the smart application of using robot that can recognize the required object and then catch it to be carried to a specific place [28].

This chapter considers the applications of WSN and intelligent methods, particularly deep-learning in the robotic field to be used for different implementations of robotic cars. These applications include the indoor localization for robotic cars, guiding the electrical wheelchair using eye site and EEG signals, and obstacle avoidance that might face the robotic cars. It also includes a brief information of WSN in terms of topologies and protocols, as well as intelligent methods and diving on the deep-learning.

2 Related Work

The robotic systems and the used techniques were considered in different presented research work in literature. Most of researchers looked for obtaining the best solutions for the underlying problems in industrial and product sides. These problems were considered for different applications, in which the solutions should come up with overcoming to them under the satisfactory of the requirements. The automation of driving robots was addressed using the gaze direction detection and tracking. This was performed by using different types of techniques and proposing algorithms in [6, 27, 28]. These techniques included hardware and software sides. The hardware side contained numerous embedded system components, such as sensors of camera, robotic cars, and Raspberry Pi. Moreover, the software side used the open CV library and mobile android application as well as proposing the working algorithm. The tracking of gaze could drive the robotic car smoothly. The obtained results promised the performance well in achieving the required target efficiently. At the other hand, the brain Electroencephalography (EEG) signals was also utilized in driving the robotic cars that can be employed for paralyzed people. A survey and simulation were presented in [3] that considered five channels of EEG signals read from emotive insight headset. These five signals read the reaction of the face muscles and eyes that were employed in directing the robotic car throughout the driving [4]. Different directional control signals were adopted to guide the robotic car. This application was presented for people suffers from Quadriplegia paralyze. The presented system included two sides of hardware and software. The mentioned headset, and Raspberry Pi were the most important parts of the hardware side, while the attached simulation of the headset as well as C programming and proposed algorithm were introduced to translate the sensor readings to action of driving the car. The simulation and implementation results proved the efficiency of the proposed robotic car system. The localization of robotic cars inside the buildings cannot laid on the Global Position System (GPS) signal due to the weakness of signals and the notified errors. Therefore, the indoor localization system for the robotic car was presented based on deep-learning techniques [18, 19]. The intelligent techniques were used to increase the resolution of the robotic allocation inside the building. The dataset of images was collected from the allocated building using a Raspberry Pi camera that was fixed on the robotic car. Different conditions were considered, such as the accuracy, distances between the car and the obstacles as well as the height of them. The idea was to catch the location of the car by identifying the real-time picture on which location was taken using the presented deep-learning model. The results expressed that the presented localization system was performed with high accuracy. The navigation of mobile robots was adopted based on WSN. The local and global location systems were presented in [23]. The local one calculated the position of the robot and the orientation as a start point for navigation throughout the driving of robots and updating the position in real-time working. While the global location was focused on obtaining the position and orientation without any information about the previous states. WSN was adopted not just in navigation systems for robots, but also in localization and to be combined with intelligent methods, such as fuzzy logic and computer vision for increasing the accuracy [1]. Different models of fuzzy logic were presented to deal with multi-targets that the blimp robot and easing the autonomy of it. WSN was particularly used for measuring the distances between robot and the targeted objects that need to be tracked and allocated. Whilst, the robot was employed as a mobile WSN to sense a specific area and surrounding circumstances [12]. As an example of the robotic WSN is the router one that was used for measuring the speed and the communication quality between sender and receiver. Moreover, the localization of robot in the building that suffered from bad path planning with high danger factors was improved by combining WSN and feature clustering method. The position of the robot was estimated using the feature clustering method, while the WSN was utilized for tracking the robot inside the building. Kalman filter was also adopted to quantize the right location of robot based on the feature clustering and WSN information [9]. It is important to note that the robot path planning throughout its movement was considered as the biggest issue of autonomy. The path planning of robots needed numerous parameters related to the nodes for finding the best path that can save power, time and satisfying the requirements in terms of data transmission and applying the jobs in well form [5]. Each challenge was tackled by proposing a suitable algorithm that can address the expected problems. The combination of the robots with the Internet of Things (IoT) can generate a new version of scientific product, called Internet of Robotic Thing (IoRT) that is used for different applications [20]. This combination led to be the active tool in the fourth industry revolution in the fields of agriculture, industry, health, etc. Moreover, the robots with WSN have been utilized to manage the house holding jobs in smart way [10]. This robot used different types of sensors for measuring the surrounding environment inside the house to take the right actions that can solve any occurred problem. The ZigBee technique was adopted as a wireless communication system between robot and the server. The intelligent approaches have been widely used with the WSN and robots for different applications to enhance the efficiency of performance. Deep-learning technique was used with the robot to apply numerous applications with a wide range study of the ability of learning in robots [26]. It also listed the possible challenges for implementing the deep-learning models in the embedded systems of robots to work in the same efficiency of simulation. In addition, two applications for robots with deep-learning were addressed in [16] that considered the issues and challenges faced the implementation and the difference between the platform and real-time performance.

3 Concepts of WSN

With the development of the human and industry life style that is based mainly on the modern technologies, which uses recent hardware and software techniques. This development needs devices to work in efficient performance, in which the requirements are satisfied. These requirements motivate the researchers for inventing technologies that can be compensate the speedy interactions of users with the related devices. One of the most important technologies is the WSN that adopts different types of sensor nodes to measure the physical factors of the environment. As mentioned above, WSN includes main components, such as the sensor nodes, base station/gateway/sink node, wireless communication technology as explained in Fig. 1 that shows the structure of WSN. Normally, the sensor nodes are distributed in random or planned way to cover the underlying area to be monitored. These nodes are grouped based different techniques and algorithms to called these groups by clusters. Each cluster has one or more cluster head that delivers the collected data from nodes to the base station. The most common clustering techniques are Low-energy adaptive clustering hierarchy (LEACH) and LEACH-C (Central) that are consider the available power of included nodes in selecting the cluster head. The nodes are grouped based on the closest one in terms of distances [11]. Figure 5 shows the structure of the clustered WSN that illustrates the data flow between nodes and base station through the cluster head [17]. The clustering of the sensor nodes can reduce the consumed power

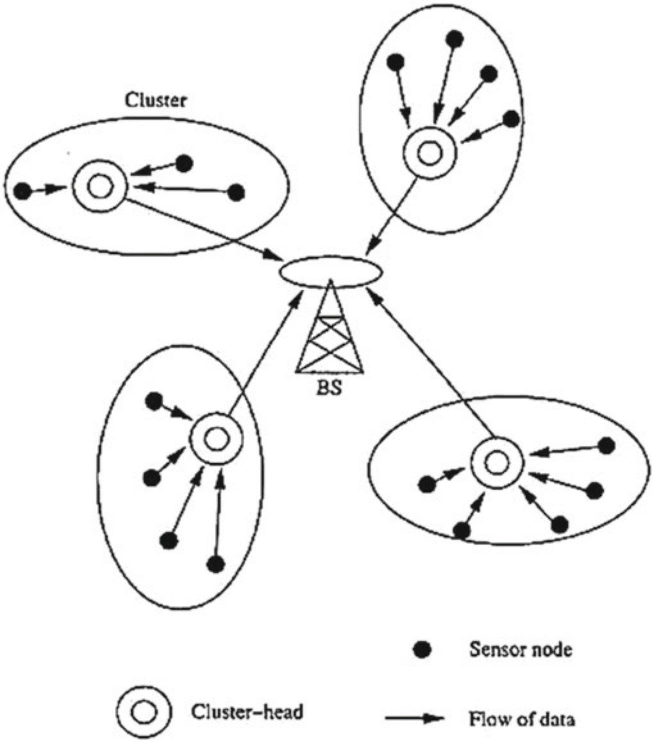


Fig. 5 Clustering of WSN structure [17]

of the sensor nodes, which in turn increasing the life time for WSN. This reduction in power is the result of decreasing the communication range between nodes and the availability of cluster head that manage the network work flow, including makes some nodes in sleeping mode. Moreover, the clustering offers the full exploring to the available nodes in the covered area by minimizing the number of ignored nodes due to the far distances with the nearest node.

At the other hand, each senor node includes, processing/controller unit for formulating the sensor reading data, memory unit for storage, sensors, actuators, power supply/battery, and communication unit for data exchange [7] as shown in Fig. 6 that explains the structure of the sensor node [15]. Each unit has been employed to perform a specific function.

Sensors and Actuators: As explained earlier, the physical phenomena can be measured using devices that contains sensors. The idea of sensors comes from the human sensing system that includes different sensing parts, such as ears for measuring the acoustic signals, eyes for optical signals, and nose for different smells, and so on. The sensors in WSN are electronic devices used for calculating the particular physical factor, for example temperature, air pressure, moisture, etc. The electronic circuit inside the sensor converts the physical factor to a signal with voltage and current

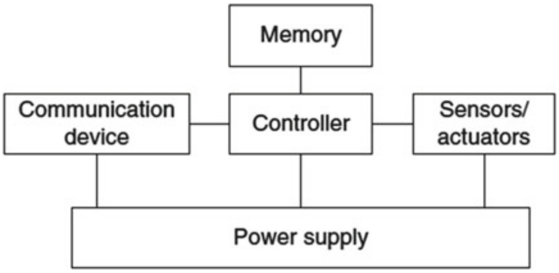


Fig. 6 Sensor node structure [15]

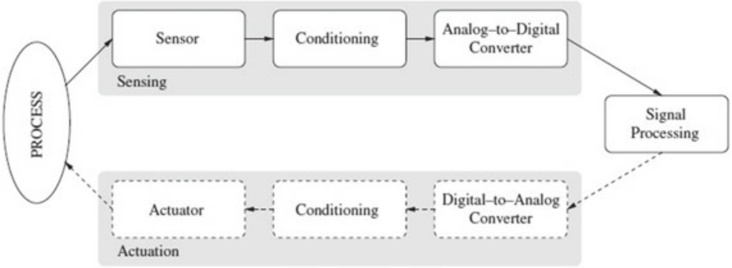


Fig. 7 The data acquisition system for sensors and actuators [7]

variances. The signal that is come from the sensor mostly is not ready to be used in digital systems, such as Raspberry Pi. Therefore, Analog to Digital Converter (ADC) is adopted to change the analog measurement signal to digital ones compatible with digital systems. The ADC is a part of the used data acquisition system inside the sensor. In the same procedure, the actuators are different types of devices that can be electrical based ones. These devices work under the analog dimension as they are performing physical actions, such as motors, fans, air conditioner, etc. Therefore, the Digital to Analog Converter (DAC) is used for converting the data form to be compatible with the actuators. Figure 7 explains the data acquisition system of the relation flow between sensors and actuators [7].

The sensors are classified into passive and active ones. The passive sensors have no power supply, as they sense the power signals from the surrounding environment. While the active sensors require external power supply for supporting the sensing life cycle [7]. These types are adopted in different systems based on the application, for example, the attending system of students in a class using Radio Frequency Identification (RFID) sensors that include receiver and tags. The tags can be passive and active depending on the required distances that should be covered between tags and receivers.

Processing/Controller: The collected data that represents the physical factor measurement is not valid to be used by the connected system due to the redundancy, errors and faults that can happen in hardware or software sides. This unit can be a

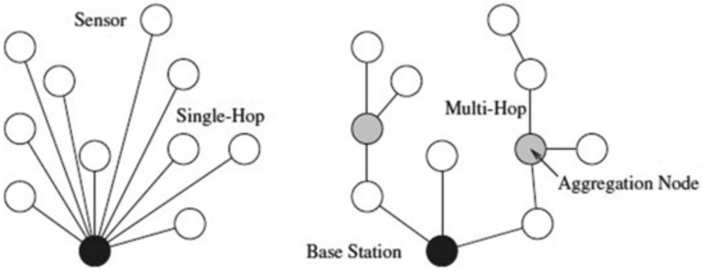


Fig. 8 Single-hop and multi-hop WSN communication ways [7]

microprocessor that is embedded inside the programmable systems, such as raspberry Pi and Arduino kits. It responsible for reformulating the collected data and sometimes taking the decision that is sent to the actuators. Thus, it works with the sensors and actuators in terms of data exchange and sending. The processing/control speed depends on the clock of the used processor that is related with the cost as a trade of between performance and accuracy [15].

Communication Devices: They are adopted in the WSN to guarantee the transmission between the nodes and base station. Different wireless communication system can be applied depending on the amount of data and the environment that follows the standards of IEEE, such as IEEE.802.11a-g that manage the wireless media including the WiFi technology, Orthogonal Frequency Division Multiplexing (OFDM), Orthogonal Frequency Division Multiple Access (OFDMA), and Filtered OFDM (F-OFDM) for the fourth to sixth generations with frequencies between 2.4 and 5 GHz. At the other hand, the data is transmitted using two ways: single-hop and multi-hop that mean the jumping of data between node and others. Figure 8 shows the mentioned data transmission ways. The multi-hop is used for ensuring the connection between the nodes and guaranteeing the involving of all nodes in terms of data delivery [7].

Memory: All sensor nodes have data from readings the physical factors and these data needs a temporary storage unit, which is called memory. It can store a small amount of data for a time period, while preparing the transmitted data packet. In addition, this unit is employed in saving the embedded programming codes for managing the embedded systems [15]. Two types of memory are adopted I sensor nodes: Random Access Memory (RAM) for temporary saving throughout the working time and Read Only Memory (ROM) for permanent saving. Some modern types are considered, such as Erasable Programming ROM (EPROM), which is semi-permanent memory unit.

Power Supply: All the mentioned units in the sensor nodes require a power for operation including the initialization stages for them. Normally, in WSN the power supply for a node is battery that can be chargeable or disposable. The amount of supply voltage and current is based mainly on the utilized units of sensors/actuators, processing, communication, and memory [15].

4 Topologies and Protocols of WSN

WSN as a network needs topologies to be form in a standard infrastructure that can explains the work flow of the network and related data transmission. Different topologies are presented to satisfy the requirements of distinct applications and available resources.

Bus Topology: In this topology, a node in the WSN broadcasts a message for all connected nodes, but only the distention one can receive this message to be analyzed and processed. This type is simple and easy in installing as shown in Fig. 9, yet suffering from congestion problem, particularly with a large number of nodes. Many research work were presented for developing this topology, but the congestion problem still active [22].

Tree Topology: This topology includes three main levels of root node, central hub, and nodes. The central hub collects the data from the level of nodes to be delivered to the upper one, which is the root node. The relation between levels is a parent and children, where the message is passed through lower to upper levels. The paths from a single node to the root node should be selected to be short and efficient in data delivery and each one can be single-hop or multi-hop path depending on the distances between the nodes in a single one. Tree topology includes a star and peer-to-peer topologies as illustrated in Fig. 10. The issue that may face this topology is the link breaking between nodes, especially in the fat trees that is affected even with one link breaking [22].

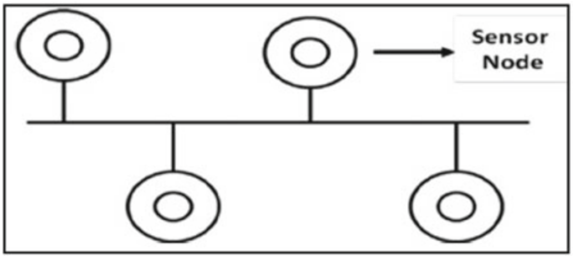


Fig. 9 Bus topology for WSN [22]

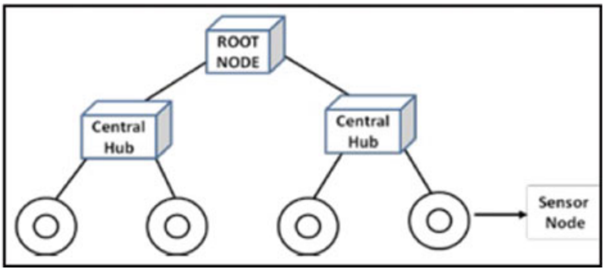


Fig. 10 Tree topology of WSN [22]

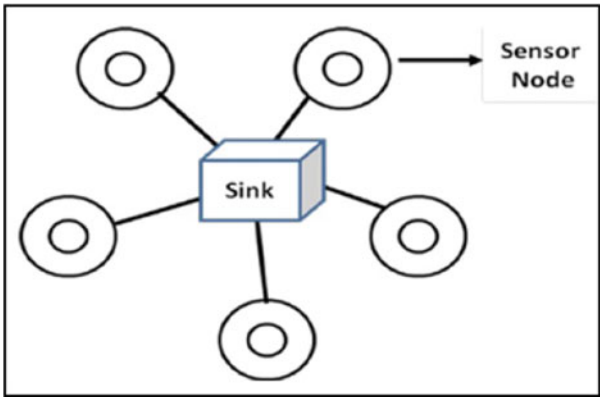
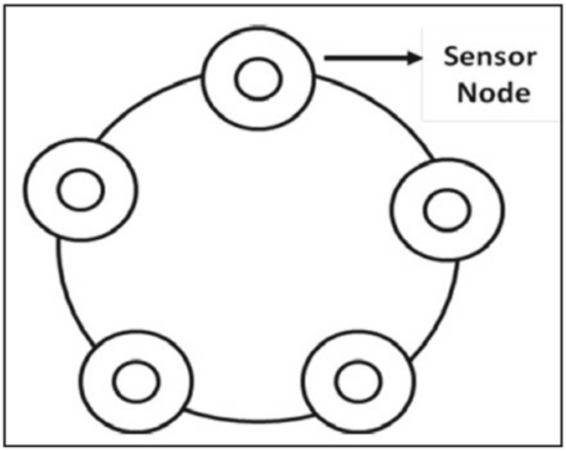


Fig. 11 Star topology of WSN [22]

Fig. 12 Ring topology of WSN [22]



Start Topology: The nodes here are connected to a central node, called sink as shown in Fig. 11. The nodes cannot be connected together directly, but through the sink node. The main problems of this topology is the congestion in the sink node, high power consumption and a single link communication. Different research work enhanced the performance of these nodes, yet the problems are still occurred with huge number of nodes in the WSN [22].

Ring Topology: It offers two neighbor nodes for each node in the WSN as a ring as shown on Fig. 12, and the data is transmitted in one direction, either clockwise or anti-clockwise. The problem of this topology is that each node is disconnected for any reason, the network is down. At the same time, the congestion and traffic problems are solved in this type [22].

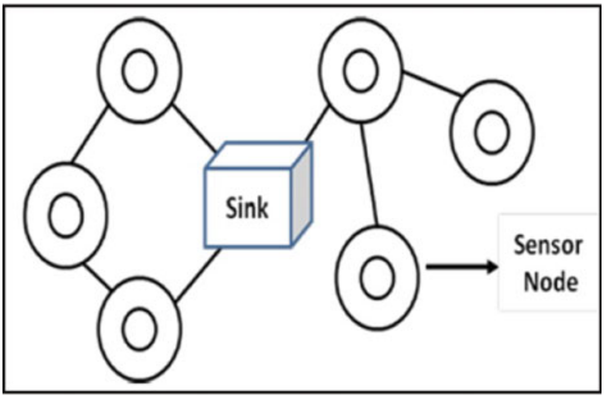


Fig. 13 Mesh topology of WSN [22]

Mesh Topology: In this topology, the nodes are connected together and can freely send as well as receive the messages between them. Two types of mesh are presented, full-mesh and partial-mesh. The full one offers full connections to all nodes, while the partial can connect all nodes through a sink node that works as a bridge between to clusters or groups of nodes within the WSN as explained in Fig. 13 [22].

Circular topology: This topology adopts the circular arranging, called tiers, of node inside the WSN with one central sink as illustrated in Fig. 14. All data is transmitted from nodes to the sink through single-hop and multi-hop paths of the diagonal ones that receives data from random nodes [22].

Grid Topology: The covered area is divided in this topology into equal size squares, each of which includes the same number of nodes. Each grid has one active sensor node, while the other are turned to sleep for increasing the lifetime of the WSN. Therefore, the nodes are working in turn, yet the condition is the nodes should be from the same type. The data is transmitted from source to destination though the grids using the grid heads as shown in Fig. 15 [22].

As mentioned above, the listed topologies are used depending on the applications, environments, and sources. It also depends on the adopted methods of node distribution on the coverage area that can be random or arranged, each of which uses a topology that satisfy the requirements.

At the other hand, the WSN includes five adopted layers in open system interaction as illustrated in Fig. 16 [7]. In addition, three crossed layers are presented, power management plane, mobility management plane, and task management plane. These crossed layers are involved in each of five ones to guarantee that they are performed in well managed in terms of power, mobility, and task. This is done using different protocols for each layer to manage the above crossed ones.

Different protocols are working on the data link layer, particularly MAC sublayer. The main job of these protocols is to decrease the consumed power, caused by collisions, idle listening, overhearing, and excessive overhead. Normally, they are categorized into schedule and contention protocols [25].

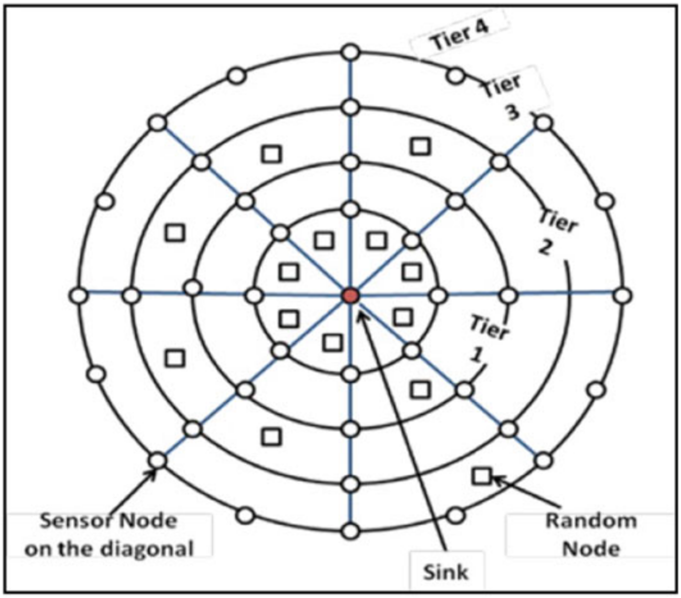


Fig. 14 Circular topology of WSN [12]

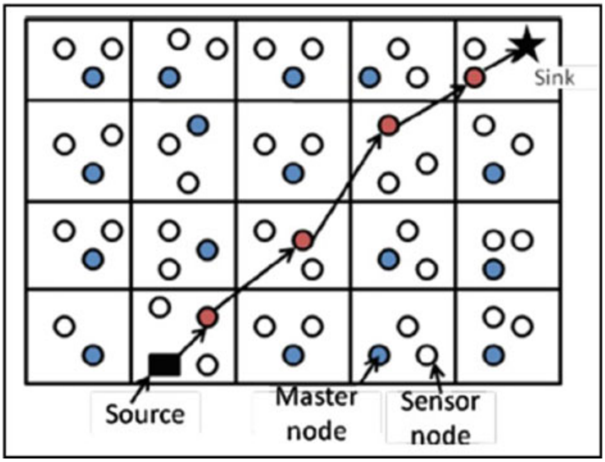


Fig. 15 Grid topology of WSN [22]

Scheduled based protocols aim to control the consumed power throughout the work of WSN, in which the life time for the network is increased. In addition, these protocols provide the WSN with more adaptively and flexibility as well as scalability, for the change in producing the services and the size of WSN. Self-Organizing Medium Access Control for Sensor-nets (SMACS) protocol is one of scheduled

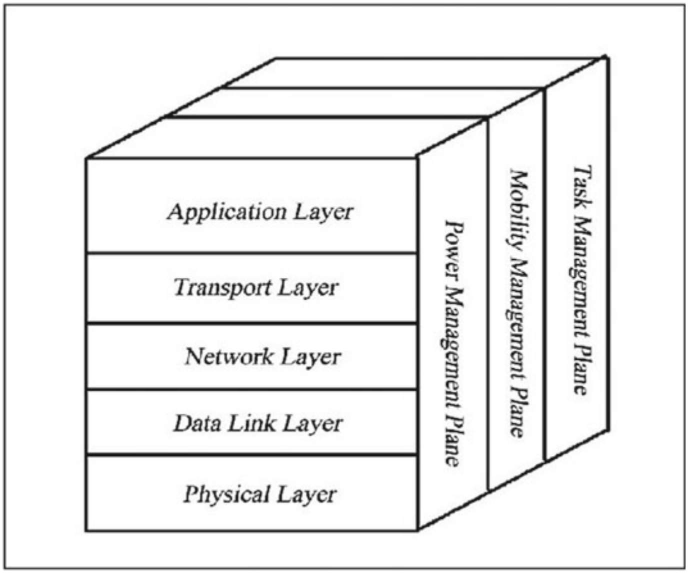


Fig. 16 WSN layer structure [7]

based ones that adopts the random network topologies without the need for synchronization establishment. This protocol uses the communication protocols of Time Division Multiple Access (TDMA), Codded Division Multiple Access (CDMA), and Frequency Division Multiple Access (FDMA). Each of which exploits its ability inn delivering the data in save with minimum power consumed [25]. At the other hand, Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol adopts a hierarchical approach in distributing the nodes inside the WSN amongst the allocated clusters. Each cluster has at least one cluster head that uses the TDMA communication protocol. The TDMA can guarantee the safe receiving of sent data with lowest errors and packet loss as well as delay. It also prevents the interconnection between the sent packets between the nodes within the cluster and from one to others.

The contention access based protocols are not produced to the WSN in particular, however, they can be used as they adopt the collision prevention, Request to Send (RTS), Clear to Send (CTS) methods to enhance the performance of WSN. This is done by overcoming the hidden layer problem in more robustness. In addition, these protocols keep the consumed energy in low level for the same reasons mentioned in LEACH protocol. The Power Aware Multiple-Access Protocol with Signaling (PAMAS) reduces the overhearing between neighboring nodes by using a separate communication signaling channel to avoid the busy RTS and CTS channels. These signals control the activation of nodes to receive or send data. The Sparse Topology and Energy Management (STEM) protocol in the same way uses two communication channels between nodes, known as data radio and wake up radio. It uses the data channel in sending data based on RTS and CTS, while it employs the wake up radio for sending the control signals. The wake up signals switch off/on the nodes to receive the data on demand. In the same procedure, the IEEE 802.11-inspired contention-based protocols work to guarantee the data delivery. The Timeout-MAC (T-MAC) protocol uses for applications with low data rate and not important delay. Moreover, the RTS and CTS are employed in data transmission for offering the safe delivery. In addition, The Berkeley Media Access Control (B-MAC) protocol is a low power with high efficiency one [25].

The routing protocols are the most important ones that work in the network layer of WSN. They manage the finding of the suitable paths between the source and destination nodes inside the network. Different techniques are adopted for performing the aim of these protocols that use parameters, such as bandwidth, bit-rate, latency as a Quality of Service (QoS) ones for selecting the right path. Generally, these protocols are classified into three categories: proactive, reactive and hybrid.

The proactive protocols update the information tables regularly that show the current state of the underlying network. Two types of the proactive ones: flat and hierarchical. The flat uses the overhead computations to find out the optimal path selection. While the hierarchical protocols utilize the trade off formula for optimizing the suitable path. In addition, the reactive protocols use a group of destination nodes in demand for establishing routs (paths). They do not update the information table, yet they depend on the dynamic movement of WSN for finding new paths. At the same time, the hybrid protocols divide the coverage area into clusters for controlling the paths between nodes within each cluster. The data is transmitted between clusters using nodes allocated for this purpose [25].

5 Concepts of Deep-Learning

Deep-learning methods are part of machine learning techniques, the difference in the number of hidden layers and other attributes. They prove their efficiency by providing high accuracy ratios for both training and testing stages. They are also used for two main purposes: feature extraction and classification. The feature extraction is used to obtain the trained weighted values that are employed for different application, such as identification. While the classifier is used for arranging the input material into classes that are trained before based on dataset [13, 24, 29]. The deep-learning model includes a number of layers and blocks. This is because the deep-learning is a class of neural network intelligent methods. The neural contains groups of neurons that perform the following:

- Consider the dataset as input.
- Calculate the scalar output.
- Updating the internal parameters for adapting the change throughout the work.

The neurons are grouped to generate the layers of deep-learning. These layers are arranged in sequence manner and each of which includes tuned parameters that control the change in dataset fitting by updating their case depending on the input parameters/feature. The output of each layer is the input for the next layer in sequence

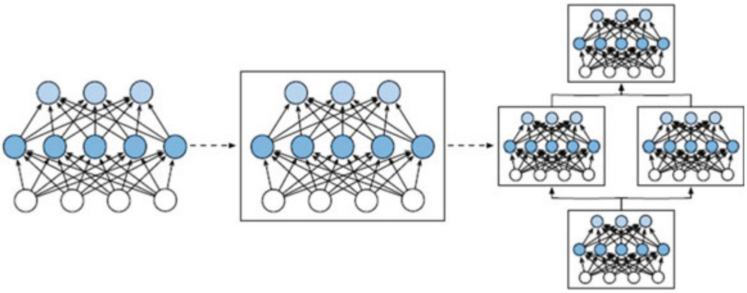


Fig. 17 The construction of block from multiple layers [29]

manner [29]. These layers do not provide the output results high accuracy, therefore, the term block is adopted in neural network ad deep-learning in turn. This block can be formulated using a single layer, multiple layers, the entire model of deep-learning. Figure 17 represents the combination of multiple layers in generating the block. The block in deep-learning represents the class that is designed for classification of the input dataset. Each one has a forward propagation function for transforming the input to output with saving the necessary parameters. In addition, the back-propagation function is also utilized for evaluating the gradients.

6 Methods of Deep-Learning

As mentioned earlier, the deep-learning is a class of machine learning technique. Thus, most of the used algorithms and methods are either the same or developed in some fields. The deep-learning methods can be categorized into three main one [8]:

- **Unsupervised Learning Network or Generative Learning**: It is used for allocating the high-order correlation of the input dataset that is used for pattern analysis and the purpose of synthesis. This is beneficial for the cases when the target class is not known well.
- **Supervised Learning Network**: Here, the target class is available and the method attend to describe the discriminate power for the underlying pattern in direct way.
- **Hybrid Learning Network**: It is used for applications that need models with different methods (supervised and unsupervised) to achieve the target in optimal way. This method can accomplish the requirements of allocating the discriminate power values for the patterns with availability of target class or not.

Throughout these categories, there are different methods for deep-learning that are adopted based on the required applications and the availability of the datasets [8]:

- **General Deep-Learning**: It is a type of machine learning to include numerous layers of processing the information that are arranged in hierarchical supervised structure. This structure works on the input unsupervised features for learning the model in classifying the input data. The concept of this method is to compute the high level correlation of features with the low level ones that are identified using the used iterative neural network.

- **Deep Belief Network (DBN)**: This method adopted different layers from generative models as shown in Fig. 18. The first two layers are connected in indirect way but symmetric. The output of these two layers is passed to the lower layers that are connected in direct way and symmetric.
- **Boltzmann Machine (BM)**: It adopts the neural network that includes symmetric connected layers, which provide the decision of the network to be On or Off as shown in Fig. 19.
- **Restricted Boltzmann Machine (RBM)**: It is a lass of BM, yet it includes hidden and visible layers that are connected in different orientation to prevent the connection between hidden and hidden as well as the visible and visible as shown in Fig. 20.
- **Deep Neural Network (DNN)**: This method contains a multi-layer structure that are classified into hidden and visible as shown in Fig. 21. The weights of these layers are connected under the supervised training model.
- **Deep Auto-encoder**: It is a DNN type with adoption of the input dataset as output targets instead of class labels. Thus, it employs the unsupervised training technique as explained in Fig. 22. In the de-noising training, the deep auto-encoder is a generative model.
- **Distributed Representation**: This method considers the distribution concept by employing the hidden factors to be connected and used for generating a new network configuration. The connection represents the distribution that are occurred in the produced pattern of activity over the units.

7 Mobile Robotic Wheelchair Based on Gaze Direction Signals for Quadriplegic People

The WSN and the robots are combined in different applications that satisfy the user and institute requirements. These applications can serve some of human cases that need special care, such as quadriplegic people, who suffers from severe limitation in the movement. The electrical wheelchair solves the movement problem for paralyzed people, but in case of quadriplegic the matter is different and needs more efforts for adapting a solution can overcome this problem. The robotic car concepts can be adopted in driving the wheelchair based on the gaze direction. The idea behind is to detect and track the pupil of the eye to get the aimed direction to be reflected on the control signals, responsible on driving the robotic car [28]. This system can be programmed as a mobile application to perform the gaze tracking for four directions. Figure 23 shows the interface of the mobile application that was presented in [28].

Fig. 18 DBN method [8]

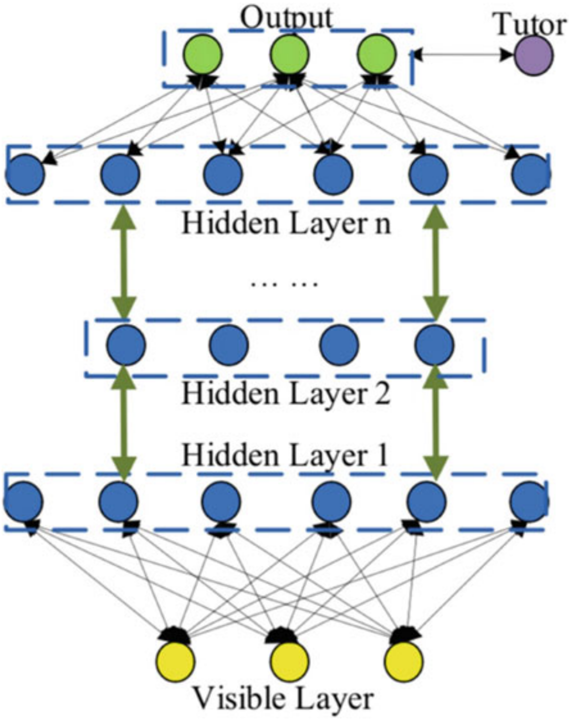
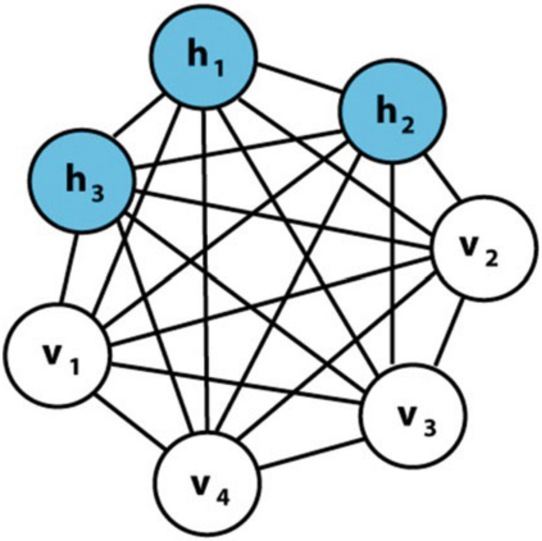


Fig. 19 BM method [8]



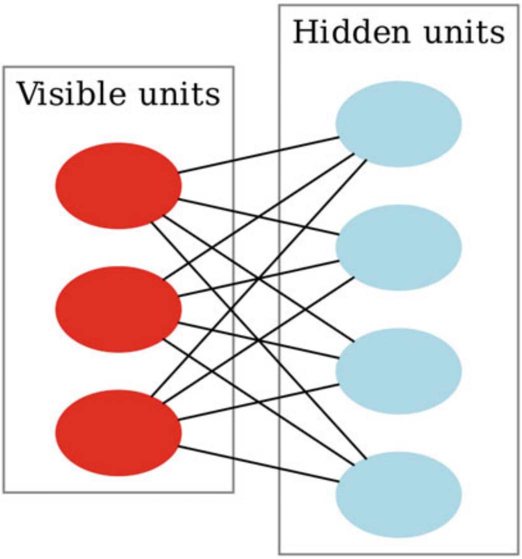


Fig. 20 RBM method [8]

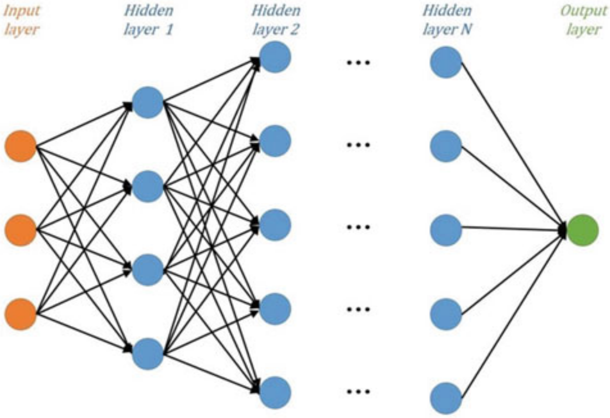


Fig. 21 DNN method [8]

The detection and tracking of the eye's pupil was performed using open CV library, particularly the Haar cascade classifier. This classifier starts with face detection of the patient and then allocated the area of interest, which is the eyes, by taking the first third of the face to focus on the pupil of eyes for tracking purpose. This has been done using the formulas of finding the width and height of the eye area. These formulas were implemented for left and right eyes. The grid is applied for the allocated eyes area based on five rows and five columns to ease the tracking of pupil as shown in

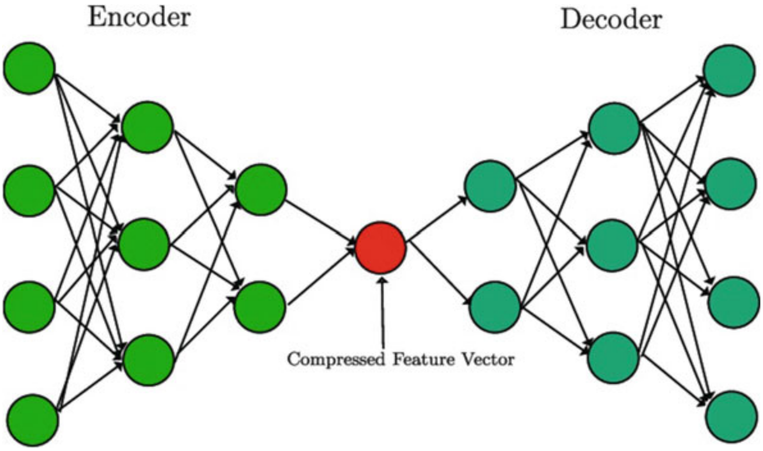
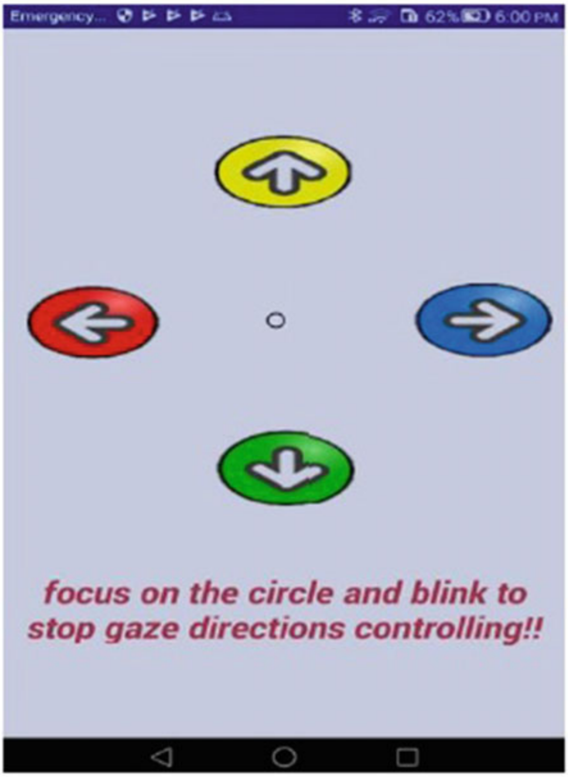


Fig. 22 Deep auto-encoder method [8]

Fig. 23 Mobile application interface [28]



focus on the circle and blink to stop gaze directions controlling!!

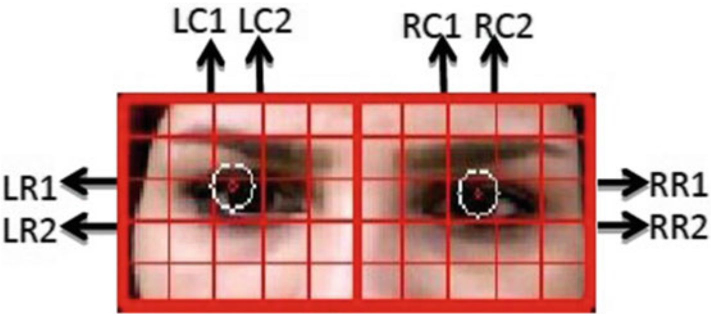


Fig. 24 The grid of eyes area [28]

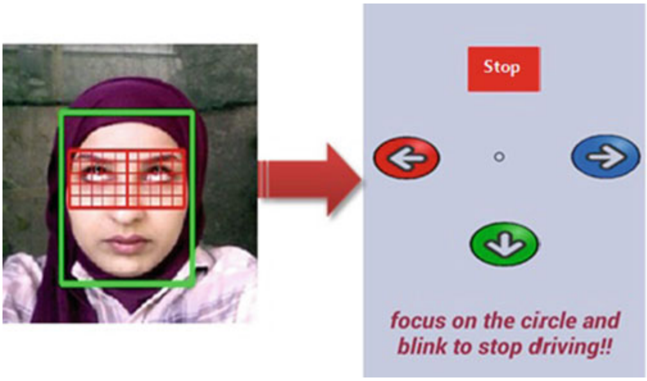


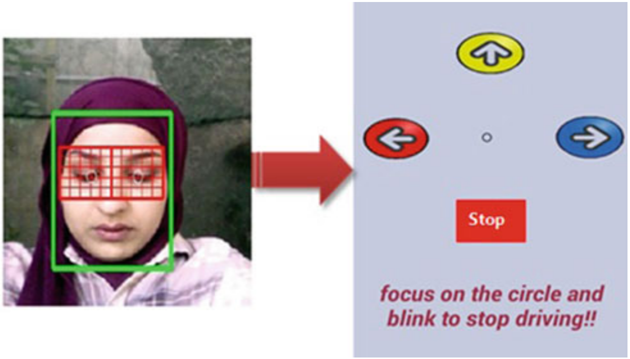
Fig. 25 Up direction reaction [28]

Fig. 24. It can be seen that the pupils of both eyes are allocated and highlighted in white circles [28].

The interaction between the mobile application and the movement of eyes. Figures 25, 26, 27 and 28 explain the movement of eyes in up, down, left and right direction as well as the reaction of mobile application after detecting the direction of pupils. While, Fig. 29 explains the case of double eye blinking, where the system is inactivated. Another double eye blinking can activate the mobile application again.

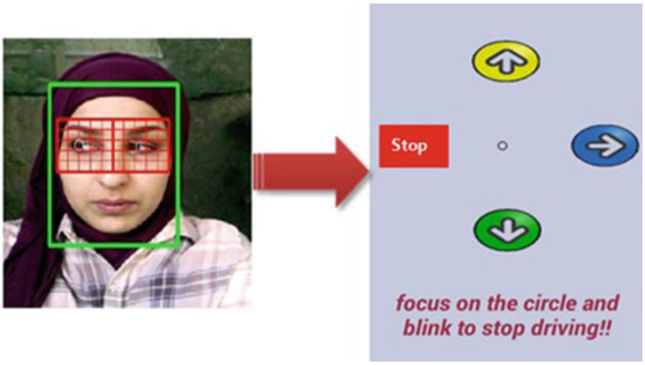
This application is applied for robotic car as a prototype of electrical wheelchair. The system requires different hardware components as shown in Fig. 30 [6]:

- Robotic car.
- An Android smartphone.
- Raspberry Pi microcomputer.
- Two ultrasonic sensors.
- Step down voltage regulator.
- Battery.



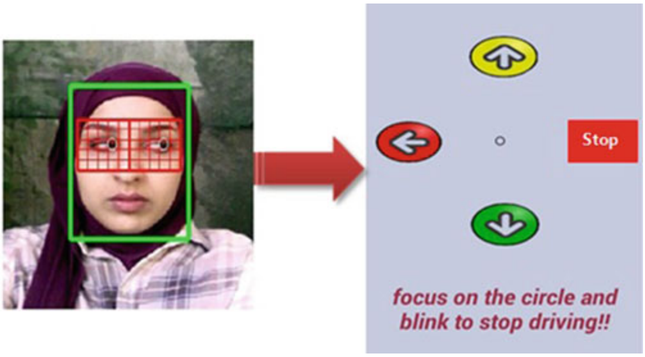
focus on the circle and blink to stop driving!!

Fig. 26 Down direction reaction [28]



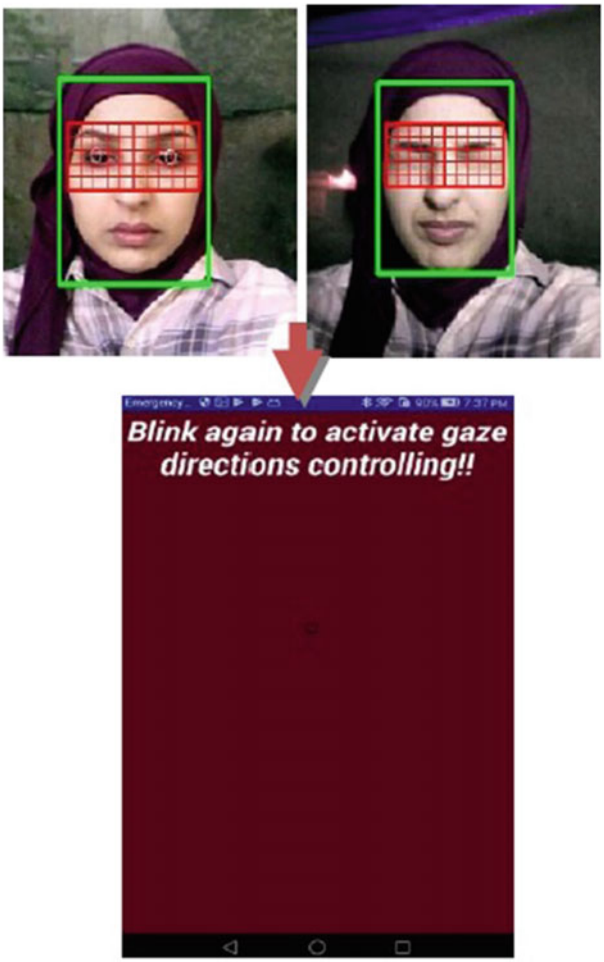
focus on the circle and blink to stop driving!!

Fig. 27 Left direction reaction [28]



focus on the circle and blink to stop driving!!

Fig. 28 Right direction reaction [28]



Blink again to activate gaze directions controlling!!

Fig. 29 Double eye blinking reaction [28]

It also needs the following software environment of Android Studio (2.3.3) IDE, and OpenCV (2.4.9) library.

After applying the mobile application of the designed robotic car, the obtained results for all moving directions prove the efficiency of performance in terms of accuracy and robust reaction. Figure 31 is a sample example that represents the left direction movement, while the blue LED is on to show the directional movement of a car to left.

The ultrasound sensors are allocated in the front and back of a car for stopping the car (wheelchair) in case there is an obstacle, as shown in Fig. 32 that explains the stop of a car in occurring of obstacle with turning on the blue LED to prevent any expected accident for the wheelchair.

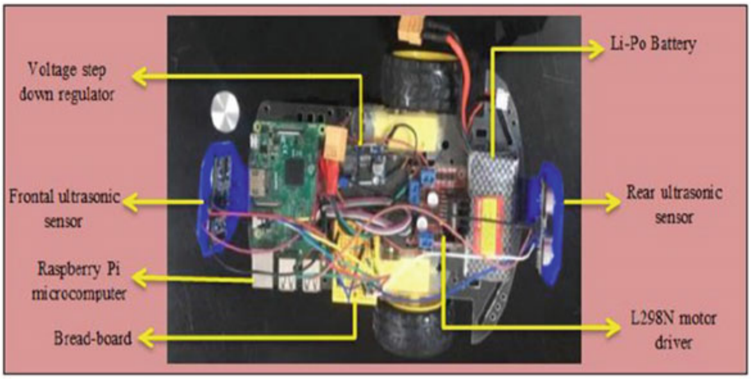

Voltage step down regulator
Frontal ultrasonic sensor
Raspberry Pi microcomputer
Bread-board
Li-Po Battery
Rear ultrasonic sensor
L298N motor driver

Fig. 30 Hardware components connected to robotic car [6]

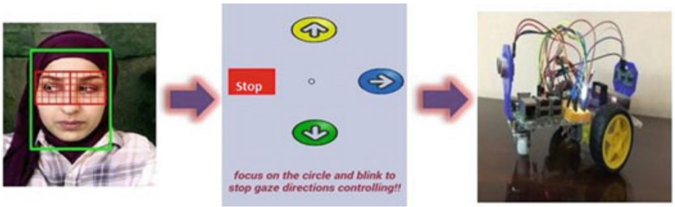

Stop
focus on the circle and blink to stop gaze directions controlling!!

Fig. 31 The left directional movement of a car [6]

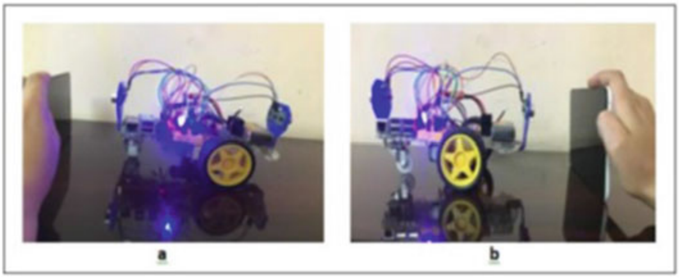


Fig. 32 Obstacle occurring [6]

At the other hand, the accuracy for the system is evaluated based on two working environments: indoor and outdoor and applied for Galaxy phones. Table 1) lists the accuracy of the system in terms of directional movements over the adopted environments. It is shown from this figure that the indoor accuracy proves the validity of the system in indoor. The reason behind the drop in the accuracy in outdoor is the effect of light that is reflected negatively on the camera as a noise.

Table 1 System accuracy over different cases [6]

Action	Indoor accuracy percentage	Outdoor accuracy percentage
Left gaze direction	90	70
Right gaze direction	100	80
Up gaze direction	80	70
Down gaze direction	70	70
Blinking	70	90
Total accuracy	82	76

8 Mobile Robotic Wheelchair Based on EEG Signals for Quadriplegic People

It is important in the scientific research to produce systems can serve different cases of human disability, in which they are able to be a part of society without the need for major assistance from others. As mentioned earlier, the quadriplegic people suffer from a severe disability that prevent them from even moving and part of their body expect the face muscles. Therefore, the need for producing smart driving wheelchair represents the high priority of helping them in moving. In addition to use the gaze tracking highlighted in the previous section, the EEG signals of the brain can be exploited in well-designed system that can dive the wheelchair for quadriplegic people in efficient way.

Figure 33 shows the block diagram of the EEG based wheelchair control system. It is explained that the movement/actions of the patient's face can be measured as EEG signals using the Emotive headset. These signals are sent to the main controller, Raspberry Pi, to be translated as a directional movement in the wheelchair, which is represented as a robotic car [4]. Different face actions are considered to cover the directional movements, stop and activation functions including blinking, smiling, frowning, left wink, right wink, and surprising as shown in Fig. 34. These actions are considered due to the notified change of EEG signals between them. This can increase the response accuracy of the system, which is reflected positively on the performance efficiency.

The Emotive headset, shown in Fig. 35, is a five channel EEG reader that expresses these signals in three main classes: mental commends, performance metrics, and facial expressions [3]. The facial expression is exploited due to easy use by patients and need less efforts of training by patient as well. This headset is simple, effective with acceptable accuracy and easy in use without any requirements of assistance.

This application is implemented using the emotive headset and the robotic car with the Raspberry Pi controller and this system requires the following hardware components as shown in Fig. 36 [9–1]:

- Robotic car.
- Raspberry Pi microcomputer.

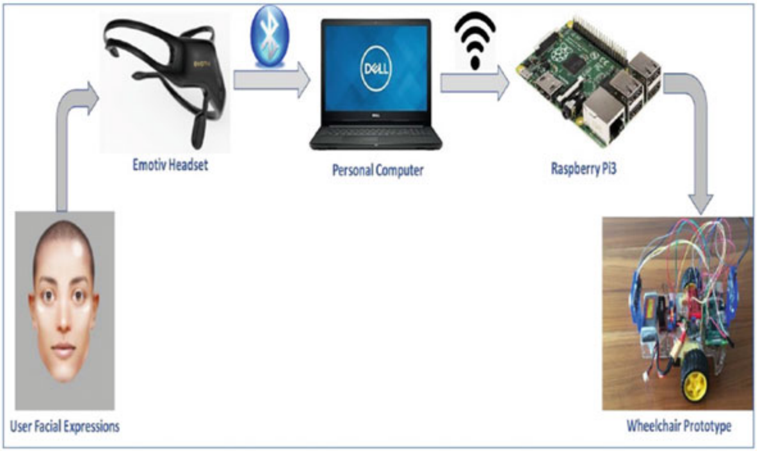


Fig. 33 EEG based wheelchair control system [4]

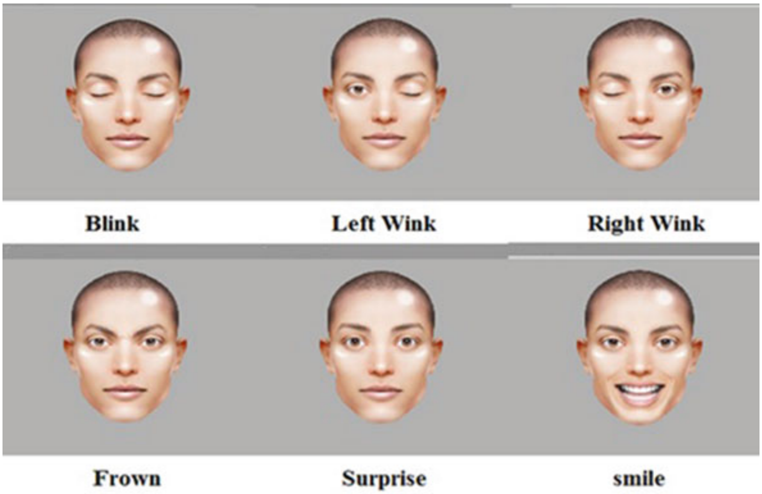


Fig. 34 Adopted face reactions [4]

- Two ultrasonic sensors.
- Step down voltage regulator.
- Battery.

Moreover, the software needed here is the Emotive control panel and Raspberry Pi environment.

Fig. 35 Emotive five channels headset [3]

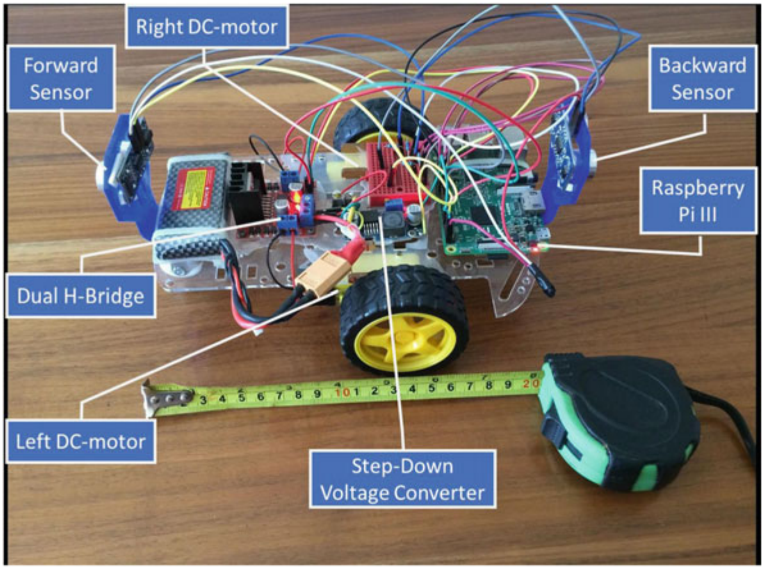


Fig. 36 Hardware components for EEG based robotic car [2]

Different case studies are considered for obtaining the results that prove the effectiveness of the EEG based wheelchair/robotic car system for quadriplegic people. Figure 37 expresses the blinking face action that activate and deactivate the system, which means double blinking for activation, and next double blinking for deactivation.

Figures 38, 39, 40 and 41 show the face actions and related car reactions as a moments to forward, backward, left and right based on surprising, furrowing, left wink, and right wink, respectively. Each action is represented in figures in a different

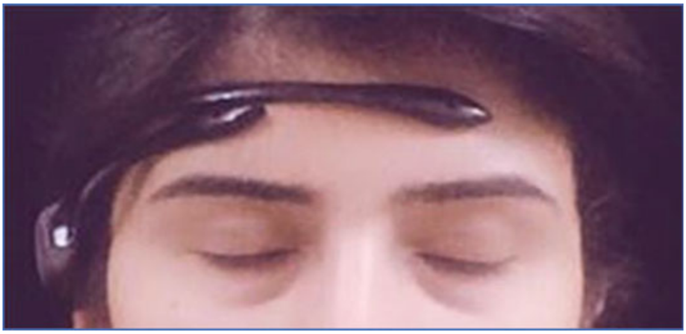


Fig. 37 Double blinking face action [2]

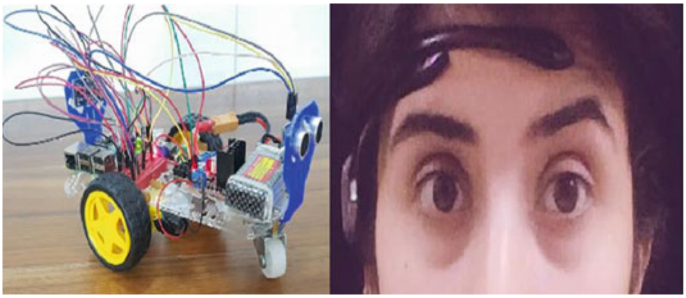


Fig. 38 Forward movement action [2]

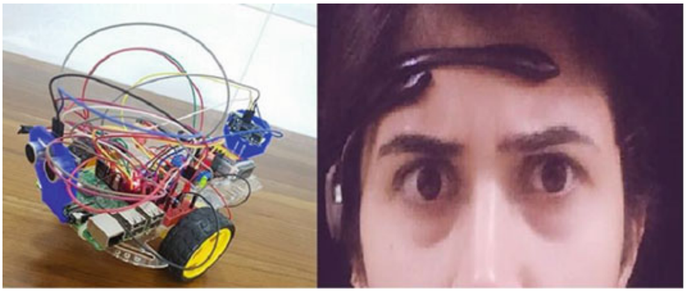


Fig. 39 Backward movement action [2]

color LED to indicate the receipt of control order. It is important to note that each face expression or action should be repeated five times within five seconds time period to ensure that it is the request of patients and it is not purposed action. From these figures, we can see the robust responding of the robotic car and the high accuracy of obeying the patient's order.

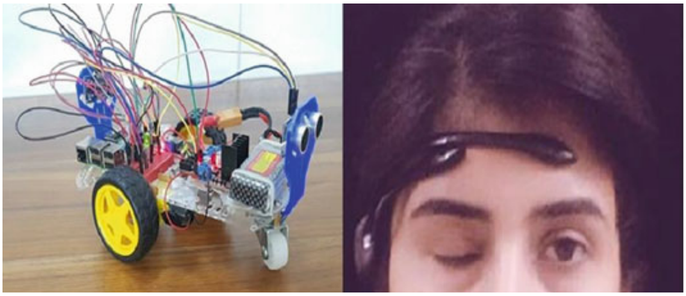


Fig. 40 Left movement action [2]

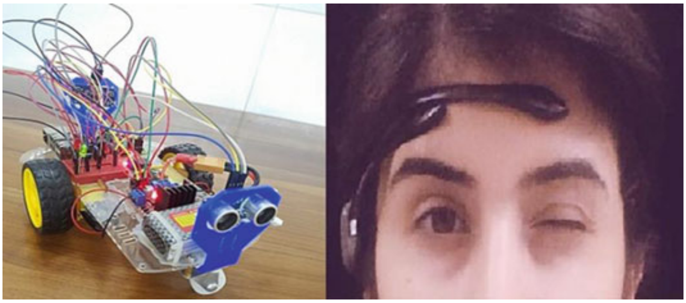


Fig. 41 Right movement action [2]

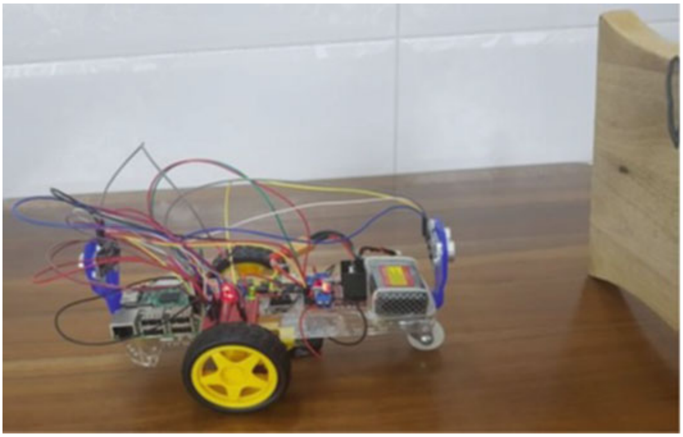


Fig. 42 Obstacle detection and stop reaction [2]

In addition, the obstacle stop action is performed using the front and back ultrasound sensors as shown in Fig. 42. These sensors detect the obstacles in front of or behind the wheelchair/robotic car to prevent any accident could be happened.

Table 2 Accuracy of the facial action and robot reaction [2]

Facial action	Robot reaction	Accuracy percentage
Surprising	Forward	80
Frowning	Backward	70
Left wink	Left turn	40
Right wink	Right turn	40
Blinking	Activation/deactivation	80
Total accuracy		62

The accuracy of the system performance is measured for the adopted actions as listed in Table 2. It is shown from this table that the left and right winking facial actions suffers from weakness in accuracy due to the closeness to the blinking action.

9 Intelligent Indoor Localization of Mobile Robotic Car

It is well known that the GPS signals is either weak with error or missed inside the buildings, particularly with numerous floors, but the indoor localization is required for performing different functions, such as finding survivors in case of disasters. Therefore, the replacement is to apply the WSN with robots to address the indoor localization with high accuracy. Moreover, the intelligent systems are added to the robots with WSN to optimize the performance, which leads to enhancement in accuracy of localization. Different methods and sensor types are employed for achieving the target by different researchers including the RFID, cameras, ultrasound, etc. In terms of intelligent systems, the machine learning and deep-learning are dominated on this application due to the efficient performance of images and raw sensor data analysis.

The combination of deep-learning model and WSN is adopted to localize the robotic car that is moved inside the building. This is done using camera and ultrasound sensors, where the read data is processed and classified using deep-learning model. The utilized model needs dataset for training, thus, an off-line data is collected from the underlying building using the same robotic car with its camera by recording movies for all rooms and halls inside such buildings. Figure 43 expresses the general block diagram that shows the working flow of the system from capturing the images till obtaining the location of robotic car [18, 19]. The offline collected data is sent to the deep-learning model for training, and the results is then sent through the website to the robotic car Raspberry PI. The robotic car captures the images from the nearest wall and this images is processed by the trained deep-learning model for classifying the location of the car.

The adopted deep-learning model is built based on Convolutional Neural Network (CNN) method that include the softmax filter, convolutional layer, pooling layer, and

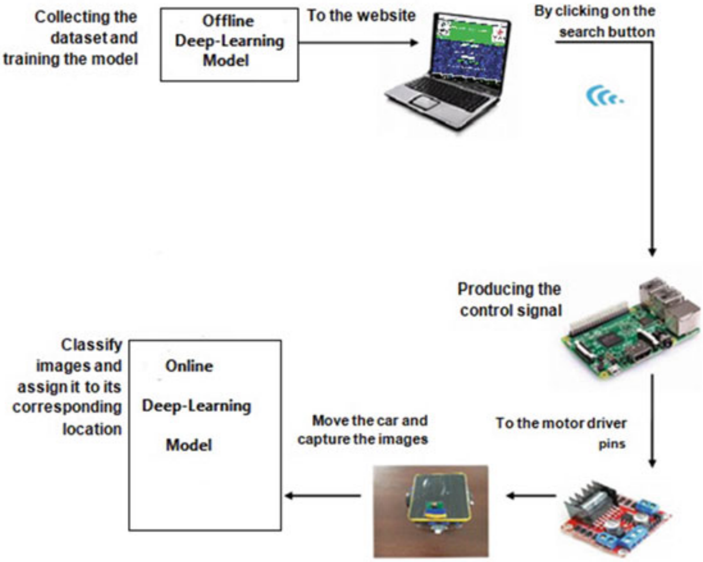

Collecting the dataset and training the model
Offline Deep-Learning Model
To the website
By clicking on the search button
Producing the control signal
Classify images and assign it to its corresponding location
Online Deep-Learning Model
Move the car and capture the images
To the motor driver pins

Fig. 43 General block diagram of indoor localization [18]

fully connected layer as shown in Fig. 44 as a circular shape [19]. The images are passed through the convolutional layers for extracting the features from them, while the job of pooling layers is to reduce the obtained parameters from the convolutional layers to be more optimized. The obtained parameters ate converted to single dimension array for feeding the fully connected layers to perform the neural network procedure. The softmax filter is the final stage of classification deep-learning. The designed deep-learning model is shown in Fig. 45 as a layers [19].

The implementation of the mentioned indoor localization system can be performed using the following hardware components:

- Robotic car.
- Raspberry PI.
- 4 megapixel RPI camera for Raspberry PI.
- Two ultrasound sensors.
- Battery.

Figure 46 explains the integrated robotic car system and each part is assigned with a title. While the software side includes the python language and Raspberry PI environment. The robotic car is the moving object inside the building, and its location is requested by the system at any time. The trained deep-learning model is already installed inside the Raspberry PI of the robotic car, thus, the car takes picture for the nearest wall to be processed and classified using the deep-learning model to classify the location. This location is sent back to the system for allocating the robotic car. The pictures are taken when the robotic car is 50 cm in distance from the wall, where

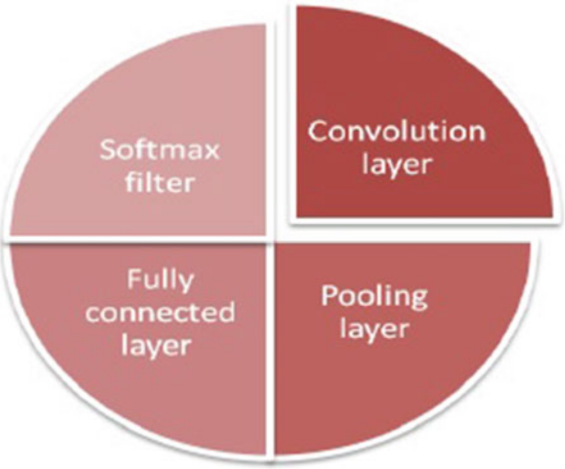


Fig. 44 The used layers in deep-learning model [19]

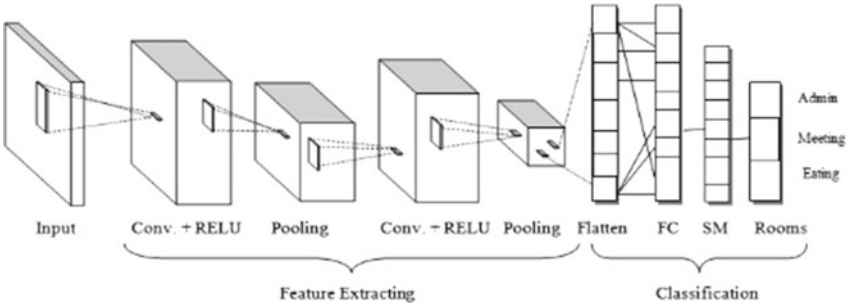


Fig. 45 The layered design deep-learning model [19]

the camera is fixed in a suitable angle. In case, the taken picture is not classified, the robotic car moves with 15° angle to right and left to achieve a picture can be classified by the system as shown in Fig. 47 [19].

The indoor localization system robotic car is provided with obstacle avoidance algorithm that is detected using the front and back ultrasound sensors. The distance of more than 90 cm is considered to be the maximum allowed one for moving the car. If the detected distance between obstacle and the car is less than or equal to 90 cm, the car is stopped and the yellow LED is set on as shown in Fig. 48 [19]. After detecting the obstacle, the car waits for 5 s to check if the obstacle is still there or moved. In case of the availability of the obstacle, the car is turned to right and go forward to be near a wall. In case there is an obstacle in the new way, the car takes the same action sequence in avoiding the obstacle [19].

In terms of the efficiency of the adopted deep-learning, the accuracy and loss of the training and validation phases are shown in Fig. 49. It is explained that the accuracy

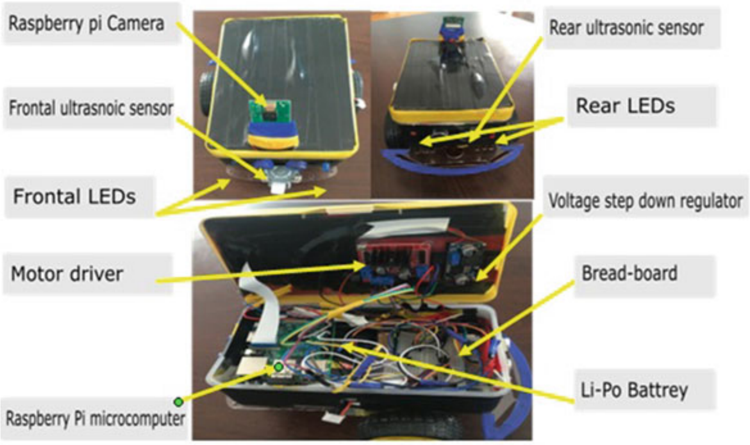


Fig. 46 Hardware components of the robotic car for indoor localization system [19]

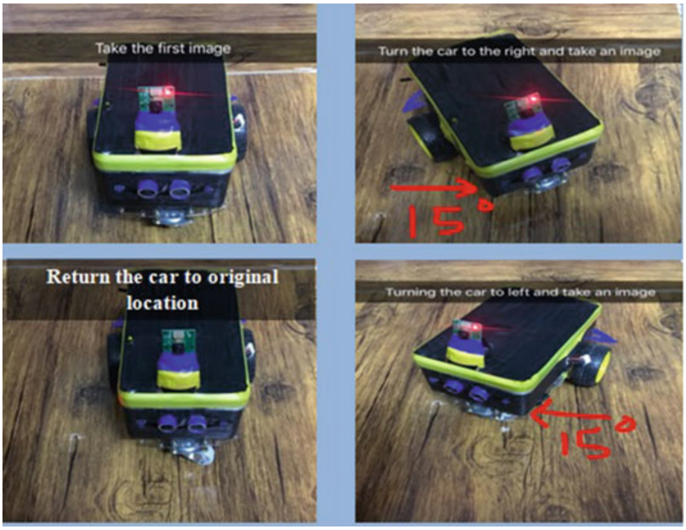

Take the first image
Turn the car to the right and take an image
15°
Return the car to original location
Turning the car to left and take an image
15°

Fig. 47 The robotic car movement to obtained a classified picture [19]

of the training reaches 99.6–100%, while the accuracy of validation is 99–100%. The loss for training and validation records low ratio. The mentioned accuracy and loss results prove the efficiency of the proposed system, which means the allocation of the robotic car in high accuracy. Moreover, the low number of epoch to get the high accuracy is called the early learning model that reaches the 100% accuracy within the 20 epoch. This is due to clear dataset as well as the optimal structure design of the deep-learning model.

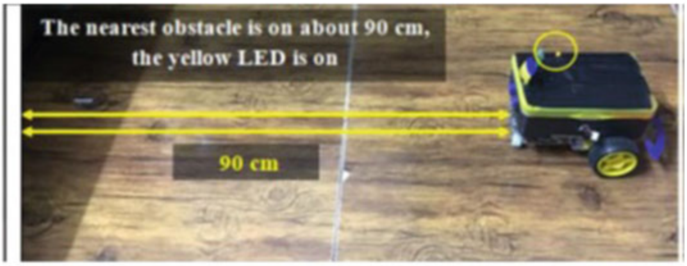


Fig. 48 Obstacle detection and car stoop [19]

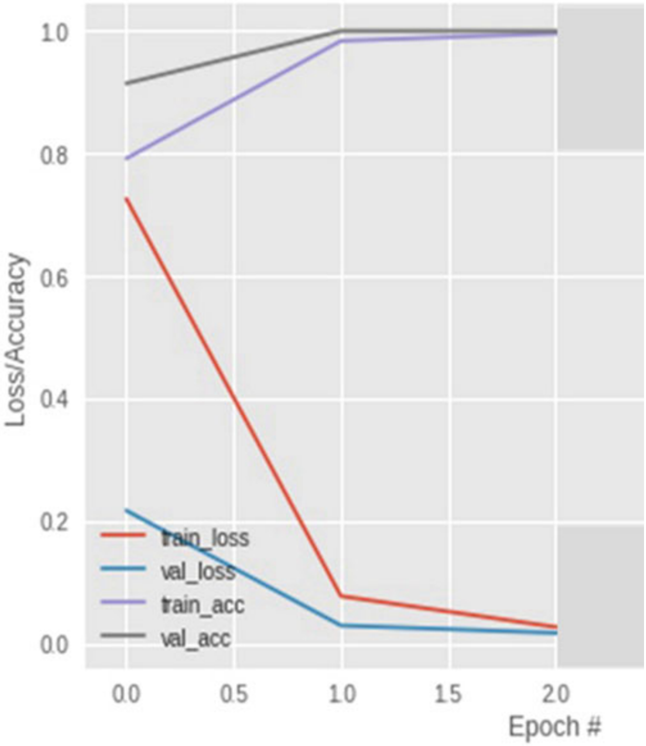


Fig. 49 Accuracy and loss of the training and validation of deep-learning model [19]

10 Discussions

In this section, the obtained results from the considered applications are discussed. The results of the Mobile Robotic Wheelchair Based on Gaze Direction Signals for Quadriplegic People application, shown in Figs. 23, 24, 25, 26, 27, 28 and 29, expressed the successful working steps in testing the proposed system. The performance recorded a high efficiency in terms of catching the gaze direction of patient's eyes and the fast response action of moving the robotic wheelchair. Numerous actions were considered in this system to cover the most possible scenarios of patient's requests.

Moreover, the Mobile Robotic Wheelchair Based on EEG Signals for Quadriplegic People applications recorded different results, as shown in Figs. 33, 34, 35, 36, 37, 38, 39, 40, 41 and 42. These results illustrated the testing steps to present the expected performance of the proposed system that used Emotive device for recording the EEG signals in synchronization with the face reactions of patients different face reactions, particularly the eyes ones, had been considered to cover the possible patient's actions. The readings of EEG signals were used to move the robotic wheelchair. The obtained results proved the efficiency of the proposed system in the Fields of order accuracy and right response.

The intelligent indoor localization of mobile robotic car application results of Figs. 43, 44, 45, 46, 47, 48 and 49 showed the practical implementation of this system. In addition, they were introduced to explain the effects of using deep-learning in detecting the places that were used for localization of robotic cars. Thee results illustrated the steps of working to get the location using the WSN to get the readings of pictures and deep-learning for classifying the location in exact to ratio up to 100% of accuracy.

At the other hand, the mentioned applications of mobile robotic were applied in prototypes for realizing the design implementation and being close to the real-time performance. The utilized embedded components and sensors were explained in the mentioned figures as well to make a clear visual picture to these applications.

11 Conclusions

This chapter presented a view on the theory background of WSN and deep-learning technologies that were adopted in presenting three mobile robotic applications. The gaze direction based robotic wheelchair application adopted the WSN and opeCV library for controlling the movement of the wheelchair for Quadriplegic People. While the EEG based robotic wheelchair application used the WSN, represented in Emotive device and the prototype, for managing the movement of robotic wheelchair for Quadriplegic People. In addition, the WSN and deep-learning model was used by the indoor localization for robotic car. The readings of WSN were employed in training the deep-learning model for allocating the position of a car. The obtained results of testing the presented applications proved the efficiency in terms of working performance and responses.

12 Summary

The robots perform different services in numerous fields of life, such as medical, engineering, and emergency cases. The enhancement of this performance is the aim of the researchers over the presented proposals and works. The enhancement can be achieved using the modern technologies that can increase the efficiency of robots in significant way. These technologies include the WSN and intelligent systems. This combination affects the performance of robot, in which the provided services are enlarged in all fields. WSN improves the controlling side of the robots and help them in indoor allocation, medical and human services. Moreover, the deep-learning models as intelligent systems are combined with the robots and WSN to enhance the robot performance in terms of accuracy and decreasing the required time. These models are selected and designed well to be adaptive with the variation of the input datasets that contains different variables, such as integers, reals, text, images, etc.

References

1. Al-Jarrah R, Al-Jarrah M, Roth H (2016) WSNs and on-board visual fuzzy servoing on blimp robot for tracking purposes. Int J Comput Commun Eng 5:215–221
2. Alawi S (2018) Prototype wheelchair design for quadriplegia people based on EEG signals. MSc Thesis. University of Technology-Iraq
3. Alawi S, Al-Qaraawi S, Croock M (2018) Electroencephalography signals based face interaction system: survey and simulation. Int J Adv Res Comput Eng Technol 7:307–310
4. Al-Qaraawi S, Croock M, Alawi S (2018) Electroencephalography signals based face interaction for directional control system. Int J Comput Netw Technol 6:49–55
5. Amaro J, Caldeira J, Soares V, Gaspar P (2020) Wireless sensor network for building robotic path: a survey of problems and restrictions. Int J Emerg Eng Res Technol 8:11–22
6. Croock M, Al-Qaraawi S, Taban R (2018) Gaze direction based mobile application for quadriplegia wheelchair control system. Int J Adv Comput Sci Appl 9:415–426
7. Dargie W, Poellabauer C (2010) Fundamentals of wireless sensor networks. Wiley series on wireless communications and mobile computing
8. Deng L, Yu D (2014) Deep learning: methods and applications. River Publishers
9. Dong X, Su B, Jiang R (2018) Indoor robot localization combining feature clustering with wireless sensor network. EUROSIP J Commun Netw 175:3–8
10. Gat K, Puranik V (2020) Zigbee based WSN and service robot into an intelligent home system using ARM7. Int J Electron Commun Eng Technol 6:48–56
11. Geetha V, Kallapurb V, Tellajeera S (2012) Clustering in wireless sensor networks: performance comparison of LEACH & LEACH-C protocols using NS2. Procedia Technol 4:163–170
12. Ghosh P, Gasparr A, Jin J, Krishnamachari B (2018) Robotic Wirel Sens Netw 2:1–59
13. Goodfellow I, Bengio Y, Courville A (2016) Deep learning
14. Govers F (2018) Artificial intelligence for robotics. Packet Publishing
15. Karl H, Willig A (2005) Protocols and architecture for wireless sensor networks. Wiley
16. Lenz I (2016) Deep learning for robotics. PhD thesis, Cornell University
17. Reka N (2015) Wireless sensor networks (WSN). Int J Comput Sci Inf Technol 6:3706–3708
18. Resan Z, Croock M (2019) Indoor localization using deep-learning and smartphone. Iraqi J Comput Commun Control Syst Eng 19:40–49
19. Resan Z, Croock M (2020) Software engineering model based smart indoor localization system using deep-learning. TELKOMNIKA Telecommun Comput Electron Control 18:1904–1916

20. Romeo L, Petitti A, Marani R, Milella A (2020) Internet of robotic things in smart domains: applications and challenges. Sensors 20:1–23
21. Russell S, Norvig P (2010) Artificial intelligence a modern approach, 3rd edn. Pearson Education Inc
22. Sharma D, Verma S, Sharma K (2013) Network topologies in wireless sensor networks: a review. Int J Electron Commun Technol 4:93–97
23. Simon J, Martinović G (2013) Navigation of mobile robots using WSN's RSSI parameter and potential field method. Acta Polytechnica Hungarica 10:107–118
24. Sobh T, Veltman A (2015) Advances in intelligent robotics and collaborative automation. River Publishers
25. Sohraby K, Minolt D, Znati T (2007) Wireless sensor networks: technology, protocols, and applications. Wiley
26. Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, Leitner J, Upcroft B, Abbeel P, Burgard W, Milford M, Corke P (2020) The limits and potentials of deep learning for robotics. Int J Robot Res 37:405–420
27. Taban R, Croock M (2018) Eye tracking based directional control system using mobile applications. Int J Comput Digital Syst 7:365–374
28. Taban R, Croock M, Korial A (2018) Eye tracking based mobile application. Int J Adv Res Comput Eng Technol 7:246–250
29. Zhang A, Lipton Z, Li M, Smola A (2022) Dive into deep learning

Fault Diagnosis and Fault Tolerant Control for *n*-Linked Two Wheel Drive Mobile Robots

Ayad Al-Dujaili, Vincent Cocquempot, Maan E. EI Najjar, Daniel Pereira, and Amjad Humaidi

Abstract This chapter presents Fault Diagnosis (FD) and Fault Tolerant Control (FTC) schemes for multiple physically linked mobile robots. A nonlinear dynamic observer is designed not only to estimate the actuator fault signal (to perform the Fault Diagnosis) but also to estimate the states that are needed in the feedback control law. Firstly, a system with three two-wheel drive (2WD) mobile robots subjected to multiplicative and additive actuators faults was considered. Secondly, the method was generalized for nth order mobile robots. A case study is presented by considering the problem of trajectory tracking of three physically linked 2WD mobile robots. The simulations were performed using the MATLAB/SIMULINK software, and it is shown that the nonlinear adaptive observer is suitable for the estimation of both the system state-space variables and the parameter associated with the actuator faults.

Keywords Fault tolerant control · Fault diagnosis · Linked mobile robots · Nonlinear observer · Two-wheel drive robots

A. Al-Dujaili (✉)
Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
e-mail: ayad.qasim@mtu.edu.iq

V. Cocquempot · M. E. E. Najjar
University of Lille, CRIStAL (Research Centre in Computer Science, Signal and Automatic Control of Lille), UMR 9189, 59655 Villeneuve d'Ascq, France
e-mail: vincent.cocquempot@univ-lille.fr

M. E. E. Najjar
e-mail: maan.el-badaoui-el-najjar@univ-lille.fr

D. Pereira
Department of Automatics, Federal University of Lavras (UFLA), Lavras-MG, Brazil
e-mail: danielpereira@ufla.br

A. Humaidi
Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq
e-mail: Amjad.J.Humaidi@uotechnology.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_13

1 Introduction

Wheeled Mobile Robots (WMR) have been increasingly used in recent years, with applications such as exploration of other planets, demining operations, and applications in risky environments. However, these complex systems, which have many instruments, may be subjected to system faults, that can result in failures of the missions and avoid them to be accomplished [1]. So, dedicated actions and strategies must be developed to attenuate the results of these faults.

Because these robots generally evolve in harsh environments, it is difficult, even impossible, to send human operators to repair faulty robots. However, when there are incipient failures in some components, a group of robots can operate at lower efficiencies or complete parts of the task with a reduced set of resources. The identification of the faults and the compensation for their effects are needed in order to ensure that the missions will be accomplished satisfactorily. The failures must be independently detected and isolated by the robots themselves. Then, the still-existing functionalities need to be utilized in a maximized manner in order to maintain the robot working.

The two-wheel drive (2WD) mobile robots are used in diverse cooperative robotic systems because of their reduced cost, efficiency, and flexibility. However, this type of robot can become out of control when an actuator is totally faulty. This may cause the wheels to roll freely, or for the robot to become stuck in a fixed position where there are undesirable frictions with the ground. To avoid situations as these, additional actuators are needed.

Physical links between 2WD mobile robots can be adopted as a solution to this problem. Adding physical interaction between robots makes the global system more tolerant to total failures of actuators, thus providing them with the ability to complete their tasks even when there are faulty actuators, rather than aborting the entire mission. The robots may be linked initially or when needed, for instance when an actuator fault is detected.

Multi-linked 2WD mobile robots are similar to a tractor-trailer [2]. These robotic systems can accomplish dexterous and complex tasks, improving performance and creating many advantages in comparison to a single robot. The potential advantages of this system are flexibility (robots may be attached or separated easily), improvement of the disturbance-rejection capabilities, robustness to failure, adaptability, intrinsic system redundancy, ease of motion in harsh environments and fault tolerance. As an over-actuated system, a linked-robot systems can reconfigure itself when partial faults occur, to be able to continue the mission.

In the present work, 2WD mobile robots are physically linked to deal with faulty actuators. With this configuration, even the faulty robots can reach at their targets, guaranteeing that the instruments they carry will continue to be available in the mission. Physically linked 2WD mobile robot systems are a kind of articulated mobile robot.

The physical link can help to solve some problems in the missions. For example, the power of the system will grow and heavier loads may be carried. Also, gaps in rough grounds can be crossed. More importantly, there will be redundancy in sensors and actuators, and the system will have improved fault tolerance.

Operations with robots can reduce the interventions performed by humans when there are accidents or disasters. In this situation, it is very important to guarantee that the global system can operate with fault tolerance. This chapter proposes the use of multi-linked 2WD mobile robots to overcome this challenge. The goal of these multi-robot systems to ensure a proper trajectory tracking even when the actuators are faulty.

An actuator faults introduce extra uncertainties into the system, mainly in the control gain. A nonlinear adaptive observer is used to deal with the uncertainties, being part of the proposed Fault Diagnosis (FD) and Fault Tolerant Control (FTC) schemes.

So, the contributions of this chapter can be outlined as follows:

- An adaptive nonlinear observer is proposed to estimate actuator faults, which are modelled as multiplicative and additive faults changing the torque inputs on the wheels, and also to estimate the states of the system, making non-measured states available for the feedback control law.
- An adaptive dynamic control law is proposed, which takes into account the estimated actuator gains and state variables provided by the adaptive nonlinear observer.

The chapter is organized as follows. Section 2 presents the Related Works. The Methodology and Mathematical Modelling are presented in Sect. 3. The Proposed System, which is a generalized FD and FTC scheme for the case of n-linked mobile robots, is presented in Sect. 4. Section 5 contains Simulation Studies. The Conclusion is presented in Sect. 6.

2 Related Works

In harsh environments, the probability of multiple faults in mobile robots may increase. In these situations, with multiple cooperative mobile robots, an actuator fault in a single 2WD mobile robot may reduce the performance or even abort the entire mission. So, a strategy is needed to reduce the effects of the faults.

When there are actuators' faults, a Fault Tolerant Control (FTC) method should be considered for an efficient tracking of the reference trajectory. Some important FTC schemes are presented in [3, 4]. FTC methods for mobile robots are proposed in [5, 6]. In [7], a reconfiguration method is developed where the non-faulty robots assume the tasks that should be performed by the faulty robots.

Fault Diagnosis (FD) must be also taken into account, either for maintenance and monitoring issues or to enhance the control system autonomously in all practical situation. In general, FD includes fault detection, fault isolation, and fault estimation [8]. The fault detection indicates the presence and also the specific time instant when the fault occurred. The fault isolation indicates the location of the fault. The fault estimation provides information such as the size and type of the fault.

Several books ([9–11]) and papers ([12–14]) have reviewed FD approaches and methods. In [15], the following classification of methods is presented: model-based, signal processing, knowledge-based, hybrid methods, and active fault diagnosis.

Observer-based methods are the most commonly used in model-based fault diagnosis. They were applied to mobile robots in [16], where a FD observer is designed based on the work presented in [17]. In [18], modeling is performed by local networks, and residues are generated by change-detection algorithms. This is a powerful approach for FD that is validated on the wheels of a mobile robot. A model-based FD for a four-wheel skid steering WMR subjected to the actuator's faults was presented in [19]. Reference [20] proposed a multiple FD method for a WMR system with a hiding effect. Reference [21] developed a diagnosis scheme for a discrete event system that can deal with two different faulty situations in an autonomous WMR mission. A fault diagnosis algorithm that estimates abnormal conditions before a breakdown occurs is presented in [22].

In [23], Kalman filters are used to estimate the states of multiple robot models, where each model is related to a fault in one actuator or sensor. Reference [24] presents a model-based Fault Detection, Isolation and Path Correction Module (FDI-PC) for 2WD robots. The simulations showed that navigation in a building can be well performed after the occurrence of a fault on a wheel by the fault compensation on the input signals. In [25], the non-faulty operation of the robot was used to train a neural network and then different faulty scenarios were considered to train again the neural network. In [26], a mobile robot of unicycle type was used in an application of data fusion for localization purposes, where the tolerance of multiple sensors faults are considered in the data fusion algorithm.

In the last decades, FD methods have been extensively studied by the control systems community. In general, reliability and fault tolerance are the major goals, particularly achieved by the detection, isolation, identification, and accommodation of faults [27–29]. The necessity of providing combined estimation of states that are not measured, parameters and faults by state observers originated researches in the field of adaptive and fault tolerant control [28, 30].

FTC design is performed for the compensation of the fault effects. With the fault estimation it is possible to change the control input and accommodate the system [31, 32].

There are very few studies on fault accommodation for multi-linked WMR in the literature. Here, the configuration of the physically-linked 2WD mobile robots is the same presented in [33–38], where two strategies for fault compensation were proposed: multiple models switching control, where each dynamic control law is designed for the compensation of a known set of actuator faults; and an adaptive control approach, with a single control law designed considering a multi-integration scheme that compensates the pre-established faults.

This chapter focuses on the fault estimation and compensation in physically linked 2WD mobile robots. The formulation starts with a system composed three-linked 2WD mobile robots, and then it is proposed a generalized for the case of n-linked

2WD mobile robots. The approach proposed by [36] is adopted as the basis for the development of the control strategy, which is based on a kinematic controller and a dynamic controller designed for trajectory tracking. In this work, one contribution is that the dynamic control law is designed considering the actuator fault compensation, what can be done by the integrating the fault estimations. The actuator faults, considered to be multiplicative faults, are estimated by a nonlinear adaptive observer, which also estimates the system states that are not measured and are needed in the feedback control law.

Based on the literature review on fault tolerant control for multi-linked mobile robots, one can notice that friction phenomena were not addressed in the aforementioned research work. Thus, this work further contributes by designing a fault tolerant control scheme that includes friction models and uncertain faults.

3 Methodology and Mathematical Modelling

3.1 Modelling of Three-Linked 2WD Mobile Robots

Let us consider the case of three-linked robots. For every mobile robot, there is actuation in the two rear wheels, but the front passive caster is omitted. The leader robot is robot 1, while the followers are robots 2 and 3. Robots 2 and 3 are oriented according to the physical links, however, robot 1 is independent. The configuration of the system, as shown in Fig. 1, is given by the vector of generalized coordinates $q = [x, y, \theta_3, \theta_2, \theta_1]$.

3.1.1 Kinematic Modelling

Mobile robots are considered to be of nonholonomic nature, being subjected to nonholonomic constraints involving velocity. The system constraints are given by

$$A(q)\dot{q} = 0 \tag{1}$$

where

$$A(q) = \begin{bmatrix} sin\theta_3 & -cos\theta_3 & 0 & 0 & 0 \\ sin\theta_2 & -cos\theta_2 & -d_2cos(\theta_2 - \theta_3) & 0 & 0 \\ sin\theta_1 & -cos\theta_1 & -d_2cos(\theta_1 - \theta_3) & -d_1cos(\theta_1 - \theta_2) & 0 \end{bmatrix} \tag{2}$$

The equations for the kinematics of the three-linked 2WD mobile robots, as shown in Fig. 1, are

$$\boldsymbol{\dot{q} = S(q)\eta} \tag{3}$$

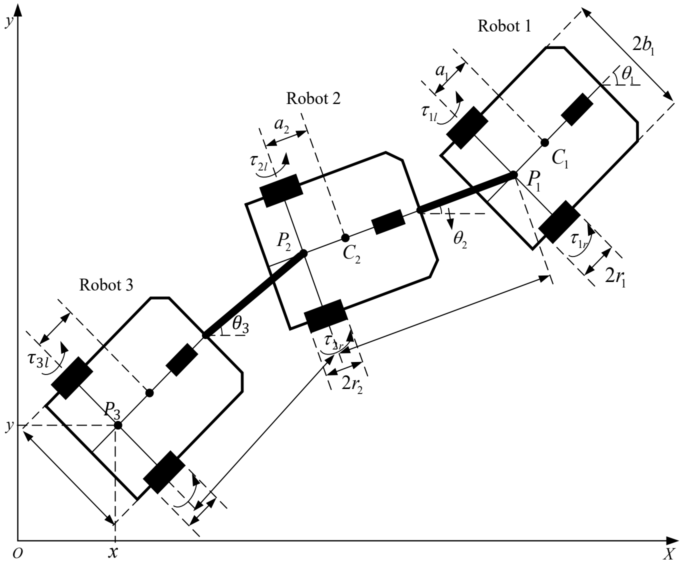


Fig. 1 Three-linked 2WD mobile robots

where $\eta = \begin{bmatrix} \upsilon_3 \ \omega_1 \end{bmatrix}^T$, υ_3 is the linear velocity of robot 3, and ω_1 is the angular velocity of robot 1. Matrix $S(q)$ is as follows

$$S(q) = \begin{bmatrix} cos\theta_3 & sin\theta_3 & \frac{1}{d_2}tan(\theta_2 - \theta_3) & \frac{1}{d_1}tan(\theta_1 - \theta_2)/cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{4}$$

Moreover, the following equation describes the natural orthogonal complement property

$$S(q)^T A(q)^T = 0 \tag{5}$$

3.1.2 Dynamic modelling

Applying the Lagrange method and considering fixed links, the dynamics of the three-linked system is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}\big(\boldsymbol{q}, \dot{\boldsymbol{q}}\big) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{A}(\boldsymbol{q})^{\boldsymbol{T}}\boldsymbol{\lambda} \tag{6}$$

being $M(q) \in R^{5\times5}$ the system inertia matrix, $C) \in R^{5\times1}$ is the centripetal and Coriolis forces vector, $B(q) \in R^{5\times6}$ is the input transformation matrix, $\tau \in R^{6\times1}$

is the vector of control torques $A(q) \in R^{3\times 5}$ is the system constraint matrix, and $\lambda \in R^{3\times 1}$ is the vector of Lagrange multipliers.

Matrices $M(q)$, $C(\boldsymbol{q}, \boldsymbol{q})$ and $B(q)$ in (21) are

$$M(q) = \begin{bmatrix} m_1 + m_2 + m_3 & 0 \\ 0 & m_1 + m_2 + m_3 \\ -\big[(m_1 + m_2)d_2 + a_3 m_3\big] sin\theta_3 & \big[(m_1 + m_2)d_2 + a_3 m_3\big] cos\theta_3 \\ -\big[d_1 m|1 + a_2 m_2\big] sin\theta_2 \big[d_1 m|1 + a_2 m_2\big] cos\theta_2 a_1 m_1 sin\theta_1 & a_1 m_1 cos\theta_1 \end{bmatrix}$$

$$M(q) = \begin{bmatrix} m_1 + m_2 + m_3 \\ 0 \\ m_1 + m_2 + m_3 \\ -[d_1 m|1 + a_2 m_2] sin\theta_2 [d_1 m|1 + a_2 m_2] cos\theta_2 a_1 m_1 sin\theta \\ 0 \\ m_1 + m_2 + m_3 \\ [(m_1 + m_2)d_2 + a_3 m_3] cos\theta_3 \\ a_1 m_1 cos\theta_1 \end{bmatrix}$$

$$B(q) = \begin{bmatrix} \frac{cos\theta_1}{r_1} & \frac{cos\theta_1}{r_1} & \frac{cos\theta_2}{r_2} & \frac{cos\theta_2}{r_2} & \frac{cos\theta_3}{r_3} & \frac{cos\theta_3}{r_3} \\ \frac{sin\theta_1}{r_1} & \frac{sin\theta_1}{r_1} & \frac{sin\theta_2}{r_2} & \frac{sin\theta_2}{r_2} & \frac{sin\theta_3}{r_3} & \frac{sin\theta_3}{r_3} \\ \frac{d_2 sin(\theta_1 - \theta_3)}{r_1} & \frac{d_2 sin(\theta_1 - \theta_3)}{r_1} & \frac{d_2 sin(\theta_2 - \theta_3)}{r_2} & \frac{d_2 sin(\theta_2 - \theta_3)}{r_2} & \frac{b_3}{r_3} & \frac{b_3}{r_3} \\ \frac{d_1 sin(\theta_1 - \theta_3)}{r_1} & \frac{d_1 sin(\theta_1 - \theta_3)}{r_1} & \frac{-b_2}{r_2} & \frac{-b_2}{r_2} & 0 & 0 \\ \frac{b_1}{r_1} & \frac{-b_1}{r_1} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C(q, \dot{q}) = \big[C_{ij}\big]_{5\times 1}$$

where

$$c_{11} = -[(m_1 + m_2)d_2 + a_3 m_3]\dot{\theta}_3^2 \cos\theta_3 (d_1 m_1 + a_2 m_2)\dot{\theta}_2^2 \cos\theta_2 - a_1 m_1 \dot{\theta}_1^2 \cos\theta_1$$

$$\begin{aligned} c_{21} = &- (d_1 m_1 + a_2 m_2) d_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) - a_1 m_1 d_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) \\ &+ 2\big(d_1 m_1 + a_2 m_2 d_2 \dot{\theta}_2 \dot{\theta}_3 sin\big). \end{aligned}$$

$$\begin{aligned} c_{31} = &- (d_1 m_1 + a_2 m_2) d_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) - a_1 m_1 d_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) + 2(d_1 m_1 \\ &+ a_2 m_2) d_2 \dot{\theta}_2 \dot{\theta}_3 sin(\theta_2 - \ \theta_3) + 2 a_1 m_1 d_2 \dot{\theta}_1 \dot{\theta}_3 \sin(\theta_1 - \theta_3) \end{aligned}$$

$$\begin{aligned} c_{41} = &- (d_1 m_1 + a_2 m_2) d_2 \dot{\theta}_3^2 \sin(\theta_2 - \theta_3) - a_1 m_1 d_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \\ &+ 2 m_1 d_1 \dot{\theta}_1 \dot{\theta}_3 \sin(\theta_1 - \theta_2) \end{aligned}$$

$$c_{51} = a_1 m_1 d_2 \dot{\theta}_3^2 sin(\theta_1 - \theta_2) + a_1 m_1 d_1 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2)$$

The Lagrange multipliers can be removed if the derivative of (3) is substituted into (6) and multiplied by $S(q)^T$, which gives.

$$\boldsymbol{\overline{M}_1(q)\dot{\eta} + \overline{M}_2(q)\eta + \overline{C}(q, \dot{q}) = \overline{B}(q)\tau} \tag{7}$$

where

$$\overline{M}_1(q) = S(q)^T M(q) S(q) \tag{8}$$

$$\overline{M}_2(q) = S(q)^T M(q) \dot{S}(q) \tag{9}$$

$$\overline{B}(q) = S(q)^T B(q) \tag{10}$$

$$\overline{C}(q, \dot{q}) = S(q)^T C(q, \dot{q}) \tag{11}$$

3.1.3 Actuator Fault Modeling

The actuator faults under consideration are:

(1) Partial loss of effectiveness of the wheel motor.
(2) Complete loss of power in some motors which introduces extra friction.

These faults are multiplicative and additive, and can be represented by

$$\tau(t) = \sigma(t)u(t) + \overline{u}(t) \tag{12}$$

where $\tau = [\tau_{1r}, \tau_{1l}, \tau_{2r}, \tau_{2l}, \tau_{3r}, \tau_{3l}]^T$ is the input torque provided by the motors, $u = [u_{1r}, u_{1l}, u_{2r}, u_{2l}, u_{3r}, u_{3l}]^T$ is the control input, $\sigma = \text{diag}\{\sigma_{1\text{r}}, \sigma_{1\text{l}}, \sigma_{3\text{r}}, \sigma_{3\text{l}}\}$ is the uncertain control effectiveness matrix, and $\overline{u} = [\overline{u}_{1r}, \overline{u}_{1l}, \overline{u}_{2r}, \overline{u}_{2l}, \overline{u}_{3r}, \overline{u}_{3l}]^T$ is the friction related to the actuator faults.

One can remark that Eq. (12) is composed of two terms, one related to the direct effect of the faults, the other caused by the resulting frictions.

Some fault scenarios are discussed below, based on the proposed actuator fault model (12). Let j denote one of the actuators $j \in \{1r, 1l, 2r, 2l, 3r, 3l\}$. The fault-free situation is represented by $\sigma_j = 1$ and $\overline{u}_j = 0$; the loss of control is given by $\sigma_j = 0$ and is associated with a freely rotating wheel or a stuck motor, represented by $\overline{u}_j = 0$ or $\overline{u}_j \neq 0$, respectively; $0 < \sigma_j < 1$ denotes partial loss of effectiveness of the actuator.

Assumption 1: The dynamic subsystem in (11) for n-linked 2WD mobile robots has two control inputs $(\upsilon_n, \omega_1)$to be controlled. Then, the design for fault compensation requires an actuation redundancy represented by

$$\mathbf{rank}\left(\overline{\mathbf{B}}\sigma\right) = \mathbf{2} \tag{13}$$

3.2 *Fault Estimation and Compensation*

In this section, the fault estimation and compensation scheme proposed for the three-linked 2WD mobile robot system, is presented. The chained form is used in conjugation with a recursive technique based on the integrator backstepping to derive the kinematic control law that guarantees the trajectory tracking [39, 40]. There is also a dynamic controller to ensure the system stability. The dynamic control law incorporates the multiplicative actuator faults that are estimated by a nonlinear adaptive observer, providing fault compensation.

The proposed FTC scheme is presented in Fig. 2. There is a nonlinear adaptive observer that estimates the faults and a dynamic control law that can compensate the faults by using these estimations. It is important to notice that the measurement of the generalized coordinates is available for feedback, but their derivatives (velocities) must be estimated by the observer in order to be used in the feedback law. The observer provides estimates of the state space vector and the parameters associated to the actuator faults, which are incorporated in the dynamic control law and can also be used for FD.

3.2.1 State-Space Representation

Consider the following notation to represent a nonlinear system in state-space

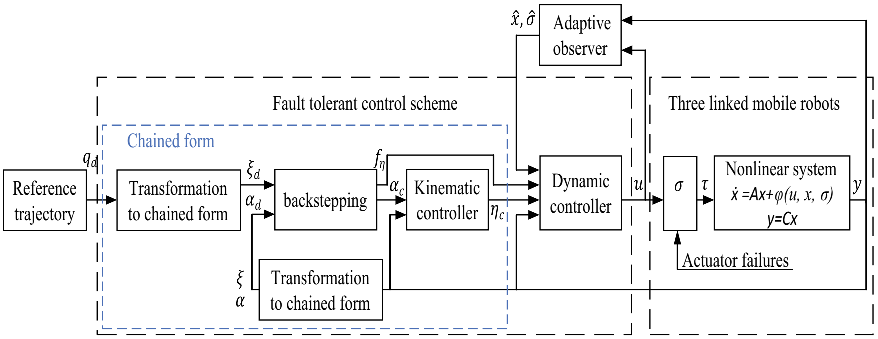


Fig. 2 Active fault tolerant control scheme for fault estimation

$$\begin{aligned}\dot{x} &= Ax(t) + \varphi(u(t), x(t), \overline{\rho}) \\ y(t) &= Cx(t)\end{aligned} \tag{14}$$

where $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the control input, $y(t) \in R^p$ is the system output, $\varphi(u(t), x(t), \overline{\rho}) \in R^n$ is a nonlinear function; and $A \in R^{n\times n}$ and $C \in R^{p\times n}$ are time-independent system matrices.

The state variables $x_1 = q$ and $x_2 = \dot{q}$ can be defined so that the state vector is $x = \left[\, x_1^T \; x_2^T \,\right]^T$. The output vector of the system is $y = q = x_1$.

From (3), (7) and (12), the system representation in the state-space form given by (14) requires

$$A = \begin{bmatrix} 0_{5\times5} & I_{5\times5} \\ 0_{5\times5} & 0_{5\times5} \end{bmatrix}, C = \left[\, I_{5\times5} \; 0_{5\times5} \,\right] \tag{15}$$

$$\varphi(u, x, \overline{\rho}) = \left[\, \varphi_1(u, x, \overline{\rho})^T \;\; \varphi_2(u, x, \overline{\rho})^T \,\right]^T \tag{16}$$

where $\overline{\rho} = \left[\rho^T, \overline{u}^T\right]^T$, $I_{5\times5}$ is a 5 × 5 identity matrix, $0_{5\times5}$ is a null square matrix of dimension 5, and

$$\varphi_1(u, x, \overline{\rho}) = 0_{5\times1}$$

$$\begin{aligned}\varphi_2(u, x, \overline{\rho}) = &S(x_1)(\overline{M}_1^{-1}(x_1)\big(-\overline{M}_2(x_1)\big(S^+(x_1)x_2 - \overline{C}(x_1, x_2) + \overline{B}(x_1)\overline{\sigma}v\big)\big) \\ &+ \dot{S}(x_1)S^+(x_1)x_2\end{aligned} \tag{17}$$

where $\tilde{\sigma} = [\mathrm{diag}(\rho), \mathrm{diag}(\overline{u})]$ and $\mathrm{v} = \left[u^T, 1, 1, 1, 1, 1, 1\right]^T$.

Note that $S^{+(.)}$ represents the pseudo-inverse of $S(.)$.

3.2.2 Nonlinear Adaptive Observer

It was presented in [31] a nonlinear adaptive observer for the estimation of the state-space vector x and a parameter vector ρ' in a nonlinear system (14) respecting *Assumption* 1.

Assumption 1: Considering the nonlinear system represented by (14):

(a) *The state vector x, the control signal u and the unknown parameter vector ρ are bounded.*
(b) *The function $\varphi(u, x, \overline{\rho})$ is Lipschitz with respect to x and $\overline{\rho}$, uniformly in u.*
(c) *The nonlinear parameterization $\varphi(u, x, .)$ is one to one.*

Remark 1 [31] *: The following notations are adopted for the subsequent definition of the observer equations:*

(a) *Given* $\Theta > 0$*, it is defined:*

$$\Delta_\Theta = diag\left[I_P, \frac{1}{\Theta}I_P, ..., \frac{1}{\Theta^{q-1}}I_p\right] \tag{18}$$

(b) *Considering A and C as in (4.4),* S_o *is the solution of:*

$$S_o + A^T S_o + S_o A - C^T C = 0 \tag{19}$$

(c) $\forall \tilde{y}(t)\epsilon\mathbb{R}^m$*,* $K(\tilde{y}(t)) = k_1\tanh(k_0\tilde{y}(t))$ with $k_1, k_0 > 0$, is a function that satisfies

.

$$\tilde{y}^T(t)K(\tilde{y}(t)) \geq \frac{1}{2}\tilde{y}^T(t)\tilde{y}(t) \tag{20}$$

.

Theorem 1 [31] *: Considering the class of nonlinear systems* (14) *with Assumption 1 and adopting the notations and definitions in Remark 1, estimations* $\widehat{x}$ *of states* x *and* $\widehat{\rho}$ *of the parameters* $\overline{\rho}$ *can be given by the nonlinear adaptive:*

$$\hat{x}'(t) = A\hat{x}(t) + \varphi\big(u(t), \hat{x}(t), \hat{\rho}(t)\big) - \Theta\Delta_\theta^{-1}\big(S_o^{-1} + \Upsilon(t)P(t)\Upsilon^T(t)\big)C^T K(\tilde{y}(t))$$

$$\dot{\hat{\rho}} = -\Theta P(t)\Upsilon^T(t)C^T K(\tilde{y}(t))$$

$$\dot{\Upsilon}(t) = \Theta\big(A - S_o^{-1}C^T C\big)\Upsilon(t) + \Delta_\Theta\frac{\partial\varphi}{\partial\rho}\big(u(t), \widehat{x}(t), \widehat{\rho}\big)$$

$$\dot{P}(t) = -\Theta P(t)\Upsilon^T(t)C^T C\Upsilon(t)P(t) + \Theta P(t) \tag{21}$$

with $\Upsilon(t) \in R^{n\times m}$, $P(t)\in R^{m\times m}$, $\Upsilon(0) = 0$, $P(0) = P^T(0) > 0$, and $\tilde{y}(t) = y(t) - C\widehat{x}(t)$.

The proof of *Theorem 1* may be found in [39].

3.2.3 Fault Estimation

It was described in Sect. 3.1.3 that the actuator faults are represented by the actuator gains ρ and additive terms μ'. The estimation of $\overline{\rho} = [\rho^T, \overline{u}^T]^T$ given by the observer

(21) is the fault estimation. Multiplicative and additive faults can be simultaneously estimated. However, there is a minimum number of actuators that are necessary to provide enough torque for the motion, limiting the set of faults that are possible to occur.

3.3 Fault Tolerant Control

3.3.1 Kinematic Controller

In this work, the derivation of the kinematic control law is based, firstly, on a transformation of the kinematic model that is performed applying the chained form [39]. In the following sequence, a recursive technique based on the classical backstepping method [40] is applied to ensure the trajectory tracking through the kinematic controller.

$$\eta_c = T_\alpha^{-1}\alpha_c \tag{22}$$

where T_α and α_c are described as follows

$$T_\alpha = \begin{bmatrix} T_{\alpha 11} & T_{\alpha 12} \\ T_{\alpha 21} & T_{\alpha 22} \end{bmatrix} \tag{23}$$

$$\alpha_c = \begin{bmatrix} \alpha_{1c} \\ \alpha_{2c} \end{bmatrix} \tag{24}$$

where

$$T_{\alpha 11} = cos\theta_3, \qquad T_{\alpha 12} = 0,$$

$$\begin{aligned} T_{\alpha 21} =& \frac{1}{d_1 d_2} sec^4\theta_3 tan(\theta_1 - \theta_2) sec^4(\theta_2 - \theta_3) \\ &+ \frac{1}{d_2}(6tan\theta_3 sin(\theta_2 - \theta_3) - 2tan(\theta_2 - \theta_3) sin(\theta_2 - \theta_3) - sec(\theta_2 - \theta_3))], \end{aligned}$$

$$T_{\alpha 22} = (1/d_1 d_2) sec^4\theta_3 sec^2(\theta_1 - \theta_2) sec^3(\theta_2 - \theta_3)$$

$$\alpha_{2c} = \alpha_{2d} - c_{n-1}\bar{z}_{n-1} - \alpha_{1d}\bar{z}_{n-2} + \sum_{i=1}^{n-2} \frac{\partial \beta_{n-2}}{\partial z_i} \alpha_{1d} z_{\mathrm{i}+1} \tag{25}$$

$$\alpha_{1c} = \alpha_{1d} + (\kappa - \nu_1)^{-1} + \left(z'_{n-2} - z'_{n-1}\frac{\partial \beta_{n-2}}{\partial z_{n-2}}\right)\alpha_{2c} \tag{26}$$

This kinematic control law is a function of the measured generalized coordinates q and the desired generalized coordinates q_d.

3.3.2 Dynamic Controller

The substitution of (12) into (7), provides

$$\dot{\eta} = \bar{M}_1^{-1}\left(-\overline{C}\eta + \overline{B}\sigma u\right) \tag{27}$$

Taking the velocity tracking error, $\eta_e = \eta - \eta_c$, the error dynamics is given by

$$\dot{\eta}_e = -\overline{M}_1^{-1}\overline{C}\eta + \overline{M}_1^{-1}\overline{B}\sigma u - \dot{\eta}_c \tag{28}$$

Then, the dynamic control law is

$$u = \left(\overline{M}_1^{-1}\overline{B}\hat{\sigma}\right)^{+}\left[-c_6\hat{\eta}_e - T_\alpha^T f_\alpha + \overline{M}_1^{-1}\overline{M}_2\hat{\eta} + \overline{M}_1^{-1}\overline{C} - \overline{M}_1^{-1}\overline{B}\hat{\overline{u}} + \dot{\eta}_c\right] \tag{29}$$

where

$$f_\eta = [z_5(\kappa - (z_3 + 2z_1)\xi_3 - (2z_4 + 5z_1)\xi_2), z_5(-z_1 - z_3)]^T$$

The details to obtain κ, z_1, z_2, z_3, z_4, z_5, ξ_2 and ξ_3, are given in [38].

Control law (29) depends on unknown parameters σ and $\overline{u}$.

The nonlinear adaptive observer (21) gives estimations of σ and $\overline{u}$ that may be available for the dynamic control law.

From (3) it can be seen that $x_2 = S(x_1)\eta$, which means that $\eta = S^{+}(x_1)x_2$. It is assumed that just the state x_1 is measured ($y = x_1$). Also, an estimation of the state x_2 is provided by the observer. So, an estimation of η can be given by $\widehat{\eta} = S^{+}(x_1)\widehat{x}_2$. Based on the estimated signals $\widehat{x}_2$, $\widehat{\eta}$ and also $\widehat{\sigma}$ and $\widehat{\overline{u}}$ the dynamic control law can be reformulated as:

$$\hat{\eta}_e = \hat{\eta} - \eta_c$$

4 Generalization for n-Linked 2WD Mobile Robots

The FTC method derived for three-linked 2WD mobile robots can be generalized for n-linked 2WD mobile robots. The kinematic control law proposed in Sect. 3 aimed the trajectory tracking, and the dynamic control law dealt with fault compensation.

An adaptive nonlinear observer can estimate the actuator faults and the non-measured states of the system, which are used in the dynamic controller.

Considering a n-linked 2WD mobile robots, it is possible to represent the model in the state-space adopting the form presented in (21) with

$$\dot{x} = Ax(t) + \varphi(u(t), x(t), \overline{\rho})$$

$$y(t) = Cx(t) \tag{30}$$

where

$$A = \begin{bmatrix} 0_{N\times N} & I_N \\ 0_{N\times N} & 0_{N\times N} \end{bmatrix}, C = \begin{bmatrix} I_N & 0_{N\times N} \end{bmatrix} \tag{31}$$

$$\varphi(u, x, \rho) = \begin{bmatrix} \varphi_1(u, x, \rho)^T & \varphi_2(u, x, \rho)^T \end{bmatrix}^T \tag{32}$$

where I_N is a $N \times N$ identity matrix, the zeros in A and C are matrices of proper dimensions such that $A \in R^{2N\times 2N}$ and $C \in R^{N\times 2N}$,

and

$$\varphi_1(u, x, \rho) = 0_{N\times 1}$$

$$\begin{aligned} &\varphi_2(u, x, \rho)S(x_1)(\overline{M}_1^{-1}(x_1)\big(-\overline{M}_2(x_1)\big(S^{+}(x_1)x_2 - \overline{C}(x_1, x_2) + \overline{B}(x_1)\sigma u\big)\big) \\ &+\dot{S}(x_1)S^{+}(x_1)x_2 \end{aligned} \tag{33}$$

where N is the dimension of the vector of the generalized coordinates of system q, and $N = n + 2$.

4.1 Nonlinear Adaptive Observer

A nonlinear adaptive observer, as derived in Sect. 3.2.2, can be used for the estimation of the state-space vector x and the parameter vector $\overline{\rho}$ in a nonlinear system that respects the properties given in *Assumption 1*, *Remark 1*, and *Theorem 1*, presented in Sect. 3.

4.2 Fault Estimation

As mentioned in Sect. 3, actuator faults are interpreted as multiplicative terms ρ and additive terms $\overline{u}$. Thus, the estimation of $\overline{\rho} = [\rho^T, \overline{u}^T]^T$ given by the observer

(21) can be also seen as a fault estimation. Notice that simultaneous faults in all the actuators can be estimated. Although, as mentioned in Sect. 3, there is a limit for the set of possible faults.

4.3 *Fault Tolerant Control Design*

The control design is based on the approach proposed in Sect. 3. For the kinematic control law, a generalization of the FTC for n-linked 2WD mobile robots can be directly performed. The kinematic controller is designed using a transformation of coordinates (chained form) [39] in combination with the recursive method (backstepping) [40].

5 Simulation Studies

The case of two-linked 2WD mobile robots.

A two-linked 2WD mobile robot system was considered, being the 2WD mobile robots presented in [41]. The robots are characterized by the following parameters: $a_1 = a_2 = 0.3m$, $b_1 = b_2 = 0.75m$, $r_1 = r_2 = 0.15m$, $m_1 = m_2 = 30kg$, $I_{m1} = I_{m2} = 15.625kg{\cdot}m^2$. The link connecting the two robots has length $d = 1m$.

The actuator faults are presented in Table 1. Only actuator $1r$ is non-faulty when the simulation starts, but at time 35 (s) this actuator completely loses effectiveness.

The parameters related to the kinematic and dynamic control laws are set as $k_1 = 1$ $k_2 = 1$, $k_3 = 0.2$, and $k_4 = 20$. The initial conditions are $x_1(0) = q(0) = [0, 1, 0]^T$ and $x_2(0) = \dot{q}(0) = [0, 1, 0]^T$. For the observer, the initial conditions are$P(0) = 0.1 I_{4\times 4}$, $x(0) = [0, 0, 0, 0,]^T$, $\rho(0) = [1, 1, 1, 1]$, $\overline{u} = [0, 0, 0, 0]$; the other parameters are $k_0 = k_2 = 1\, and\ \Theta = 1.6$. Parameter Θ is freely selected in

Table 1 Actuator faults

$0 \le t < 35$	$35 \le t \le 70$
$\rho_{1r} = 1$	$\rho_{1r} = 1$
$\rho_{2l} = 1$	$\rho_{2l} = 0$
$\rho_{2r} = 1$	$\rho_{2r} = 0.8$
$\rho_{2l} = 0.5$	$\rho_{2l} = 0.5$
$\overline{u}_{1r} = 0$	$\overline{u}_{1r} = -0.5$
$\overline{u}_{1l} = 0$	$\overline{u}_{1l} = 0$
$\overline{u}_{2r} = 0$	$\overline{u}_{2r} = 0$
$\overline{u}_{2l} = 0$	$\overline{u}_{2l} = 0$

order to provide convergence. The MATLAB/SIMULINK was used in the simulations considering simulation time$T = 70[s]$, sampling time$T_s = 10[ms]$, and solver ode4 (Runge–Kutta).

The reference trajectory for robot 2 in plane (x, y) can be seen in Fig. 3a. The obtained trajectories can also be seen in Fig. 3a. Even in case of severe faults, the controlled system can track the reference trajectory. Tracking error $q_d - q$ can be seen in Fig. 3b.

The state estimation provided by the observer can be seen in Fig. 4a. At time $t = 35[s]$, there is a remarkable transient due to faults, which can be seen in Fig. 4b.

The multiplicative and additive actuators faults estimated by the observer are shown in Fig. 5a and b, respectively. In steady state, the estimated values of ρ and $\overline{u}$ converge to the values presented in Table 1.

The simulations were performed considering a saturation of the torque signals at 5 [N. m]. So, the increasing values during the transients, as shown in Figs. 6 and 7, are truncated at 5 [N. m], but rapidly decrease to small values. Figures 8 and 9 show the linear and angular velocities of each robot.

The case of three-linked 2WD mobile robots

Consider now a three-linked 2WD mobile robots' system, where each 2WD mobile robots is characterized as in [41]. The parameters are: $a_1 = a_2 = a_3 = 0.3m$, $b_1 = b_2 = b_3 = 0.75m$, $r_1 = r_2 = r_3 = 0.15m$, $m_1 = m_2 = m_3 = 30kg$, $I_{m1} = I_{m2} = I_{m3} = 15.625kg{\cdot}m^2$. The link connecting the robots has length $d = 1m$.

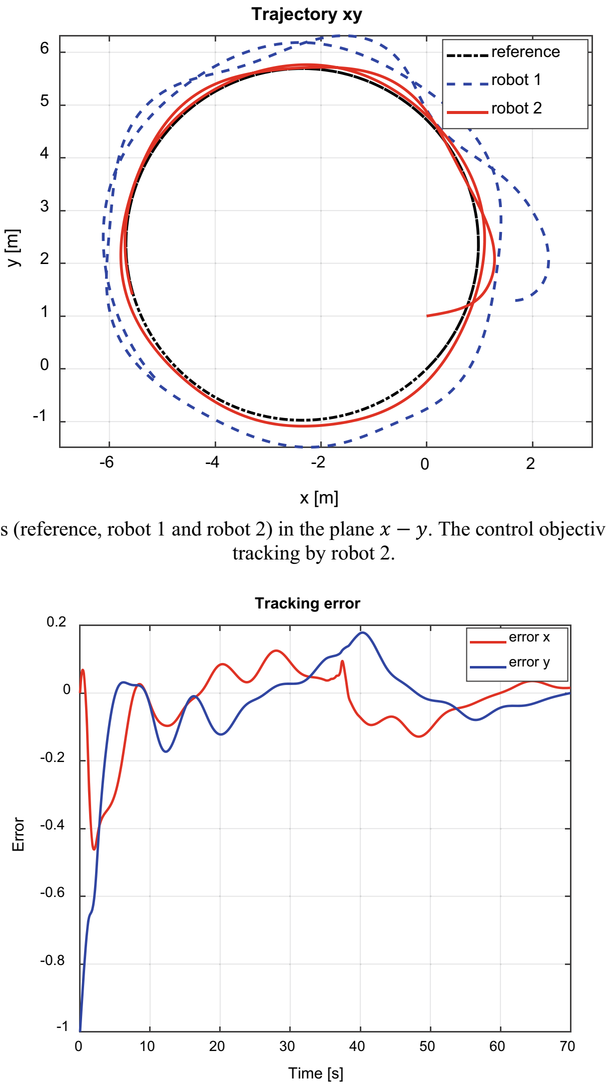
The actuator faults are presented in Table 2. A fault condition characterized by a severe loss of effectiveness (multiplicative fault), without considering friction (additive fault), is taken into consideration. The control objective is that each robot should track a sinusoidal trajectory. Only actuator $1r$ is non-faulty when the simulation starts, but at time 1100 (s) there is a fault.

The parameters of the kinematic and dynamic control laws are chosen as $C_4 = 2$, $C_5 = 2$, $C_6 = 5$ and $\kappa = 5$. For the observer, the chosen parameters are $\theta = 0.9$ and $k_0 = k_1 = 1$. Parameter Θ is selected in order to provide convergence.

In this simulation, a sinusoidal trajectory should be tracked by the generalized coordinates q. Figure 10a presents the trajectories in planes (x, y): desired, simulated and estimated. Despite the severe faults, the controlled system can track the reference trajectory. Considering q and the desired coordinates $q_d = (x_d, y_d, \theta_{3d}, \theta_{2d}, \theta_{1d})$, a detailed view of the tracking error is presented in Fig. 10b.

The estimates of the state variables are presented in Fig. 11a, where the state estimation error is shown during the transient response at the beginning of the simulation $(t < 50(s))$. At time $t = 220(s)$, due to the faults, there is a transient response, which is shown in Fig. 11b.

The estimation of the gains that characterize the actuators' faults are shown in Figs. 12 and 13. The estimates of ρ converge to the values presented in Table 2. Figure 14 shows the filtered version of the estimations, using a first-order low-pass filter whose transfer function is $1/(30s + 1)$. The estimations are filtered to make the signal smooth and reduce rapid oscillations. It will also be useful to reduce



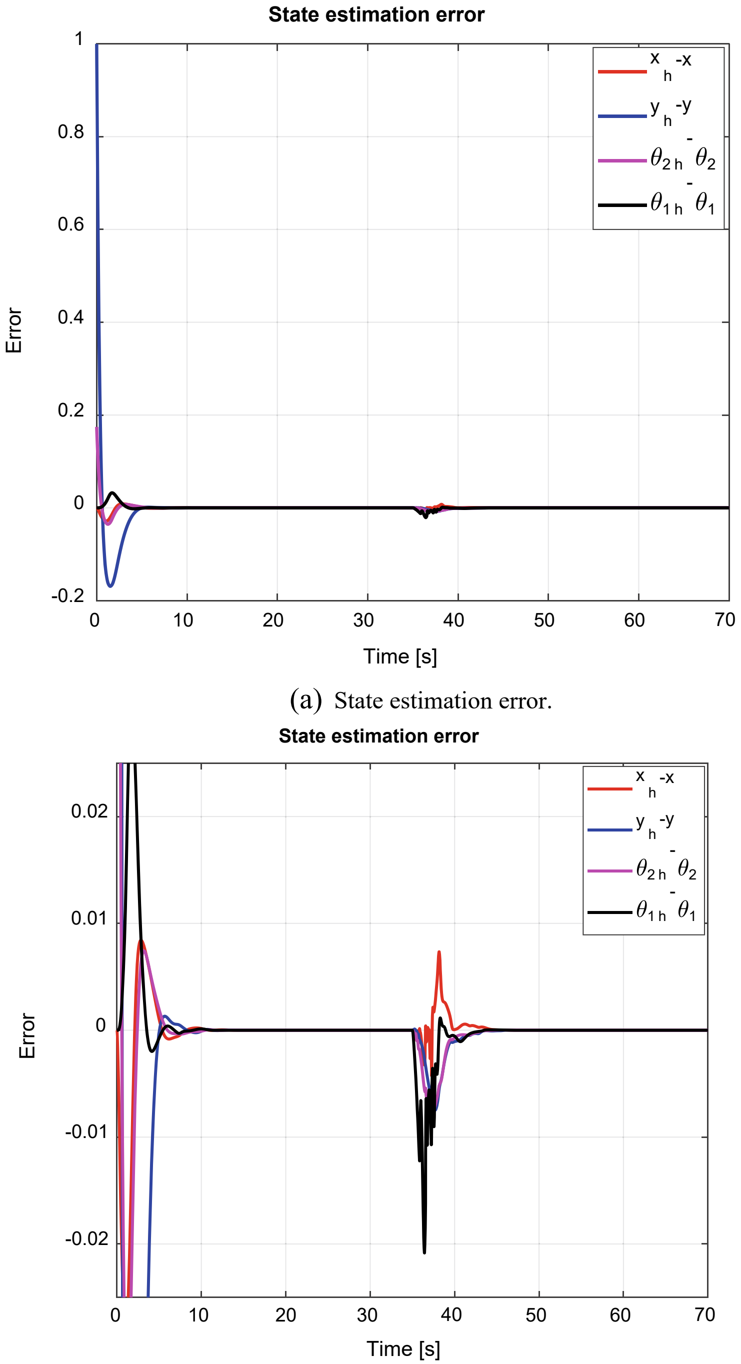
(a) Trajectories (reference, robot 1 and robot 2) in the plane $x - y$. The control objective is the reference tracking by robot 2.

(b) Transient behavior of the tracking error between reference and simulated trajectories.
Figure 3: Trajectories of the mobile robots and tracking errors.

Fig. 3 Trajectories of the mobile robots and tracking errors

noise influence and high-frequency disturbances in practical situations. These filtered estimations are used in the feedback controller.

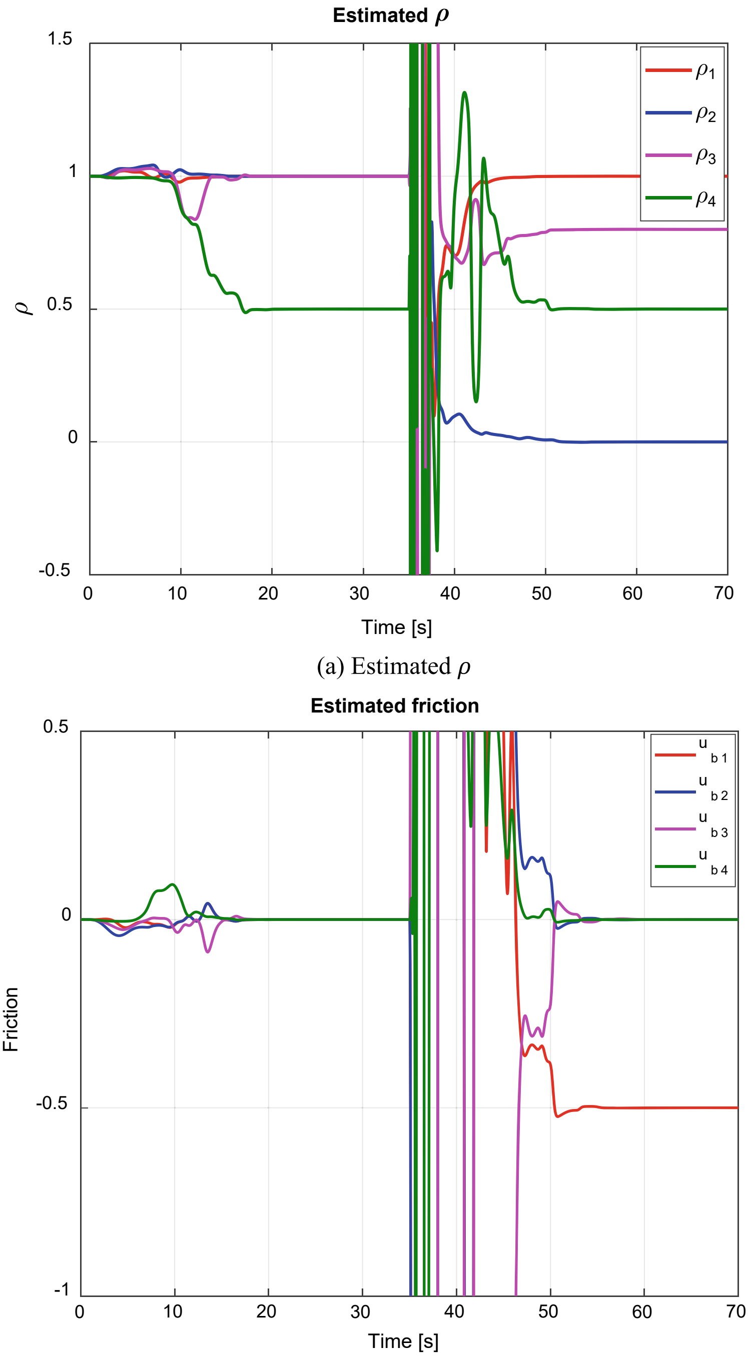
The control signals are considerably high in the beginning, as shown in Figs. 15, 16 and 17, but rapidly decrease to small values. The linear and angular velocities of each 2WD mobile robot are presented in Figs. 18 and 19.



(a) State estimation error.

(b) Zoom-in of the state estimation error.

Fig. 4 State estimation error



(a) Estimated ρ

(b) Estimated $\bar{u}$.

Fig. 5 Estimations of ρ and $\overline{u}$

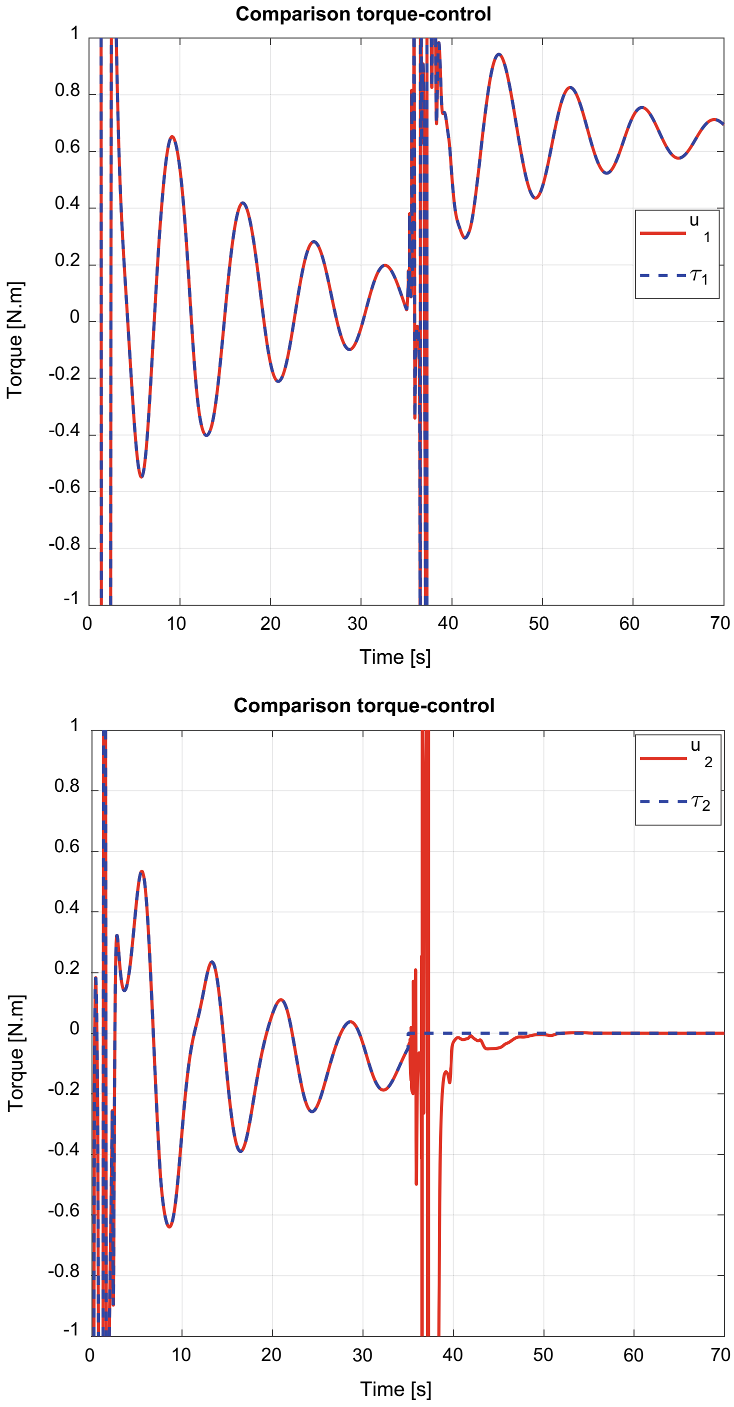
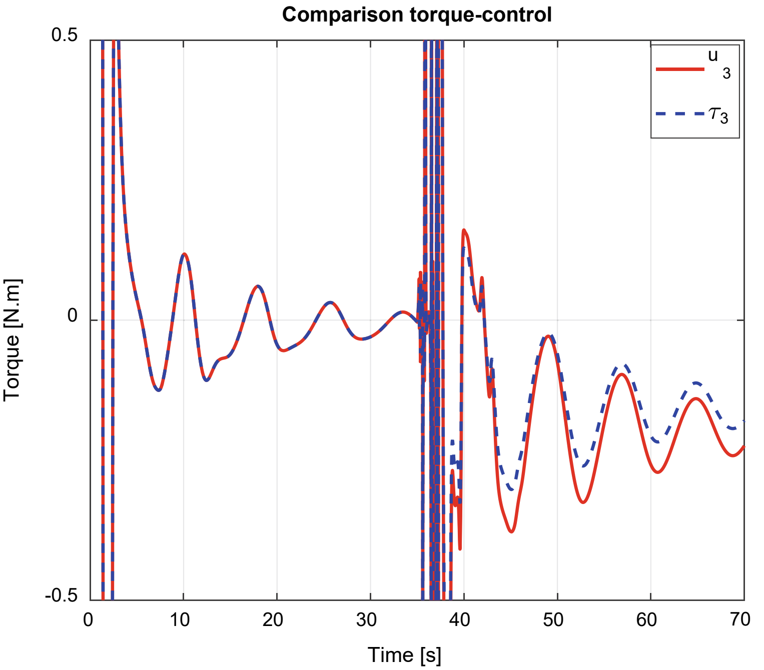


Fig. 6 Applied torques and control signals of robot 1



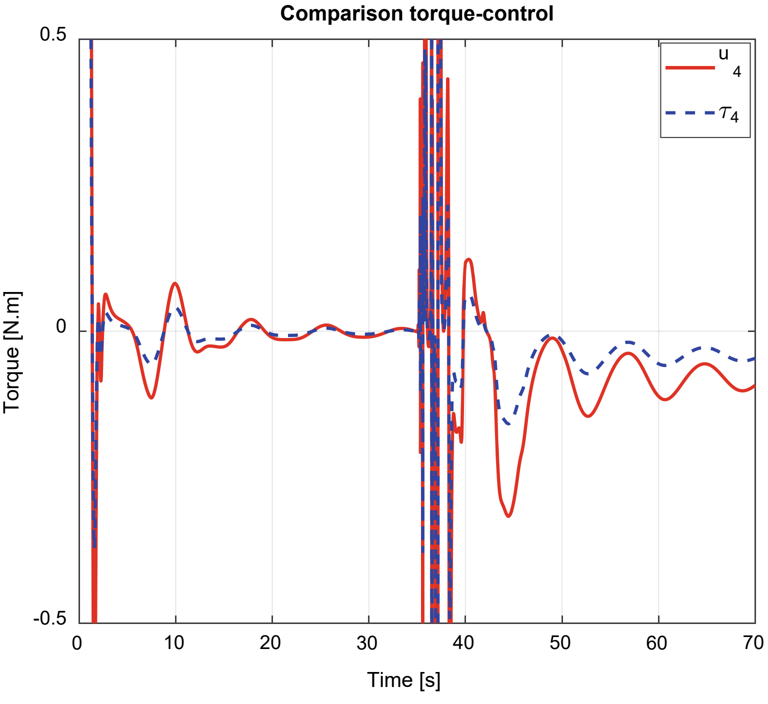


Fig. 7 Applied torques and control signals of robot 2

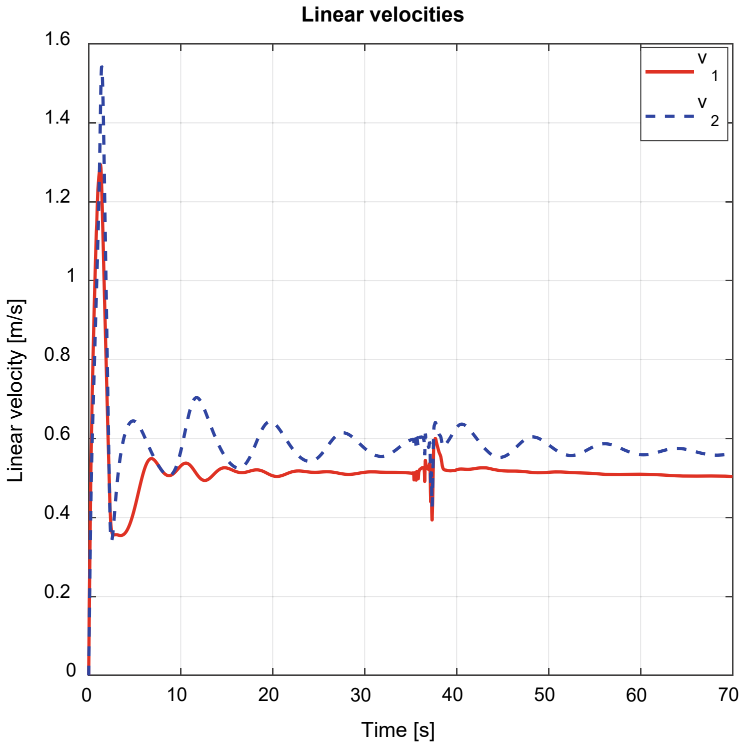


Fig. 8 Linear velocities of robot 1 and robot 2

6 Conclusion

The nonlinear adaptive observer adopted in this work was originally developed for the estimation of both the system state-space variables and a parameter vector. The work presented in this chapter considered an application with n physically linked 2WD mobile robots, where the adaptive observer is used to estimate the state-space vector and also the actuator faults, which can be either multiplicative or additive constant faults. The simulated results, considering two physically linked 2WD mobile robots and also three physically linked 2WD mobile robots, showed that the observer provides good estimates of the state-space variables and also provides estimated fault signals with small steady-state estimation errors, proving that this nonlinear adaptive observer can be a good choice for fault estimation.

In a control scheme composed of a kinematic and a dynamic controller designed to ensure good trajectory tracking for the n physically linked 2WD mobile robots, estimated actuator faults and estimated state-space variables are used to update the dynamic control law, ensuring trajectory tracking with fault tolerance. With this scheme, it is not necessary to measure the whole state-space vector in order to perform

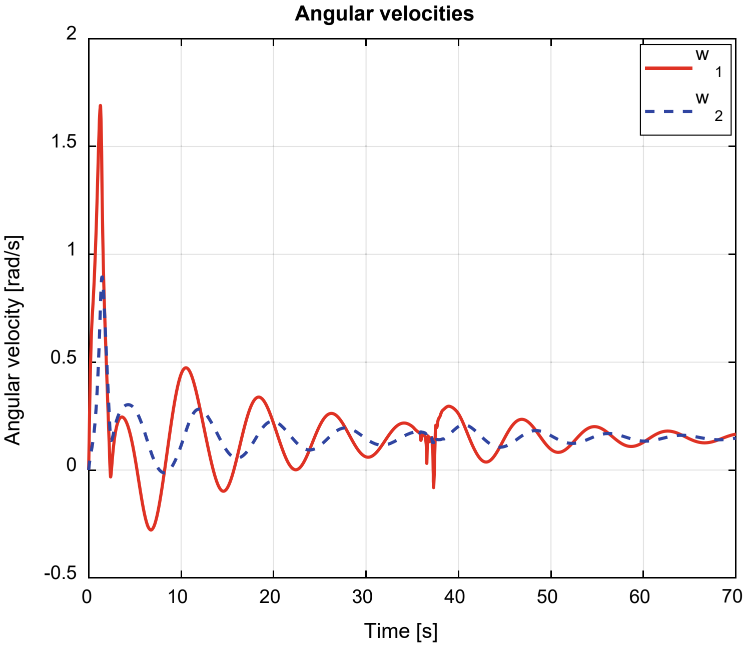


Fig. 9 Angular velocities of robot 1 and robot 2

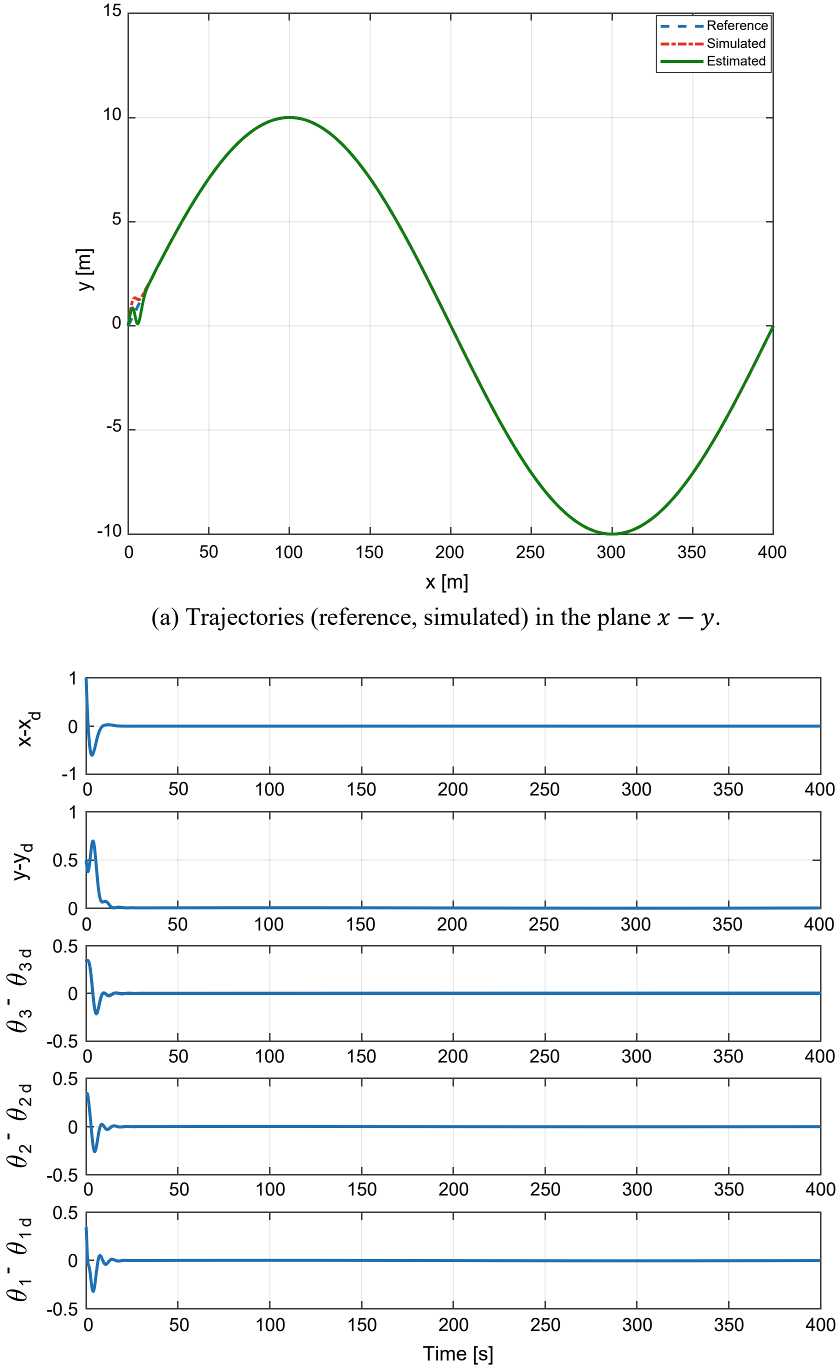
Table 2 Actuator faults in two-linked 2WD mobile robots

$0 \leq t < 220$	$220 \leq t \leq 400$
$\rho_1 = 1$	$\rho_1 = 0.8$
$\rho_2 = 0.3$	$\rho_2 = 0.3$
$\rho_3 = 0.4$	$\rho_3 = 0.4$
$\rho_4 = 0.2$	$\rho_4 = 0.2$
$\rho_5 = 0.1$	$\rho_5 = 0.1$
$\rho_6 = 0.7$	$\rho_6 = 0.7$

the state feedback, avoiding the requirement of additional sensors for the implementation of the control law. The effectiveness of the overall approach was illustrated by simulations for two-linked and three-linked robots.

In this work, only simulations were performed, which means that a natural extension of the research is the development of lab experiments with the physically linked 2WR mobile robot system. There are many mobile robot academic platforms that could be adapted to be used as a linked system, for example, the TurtleBot, which was already tested by the team that worked on this research.

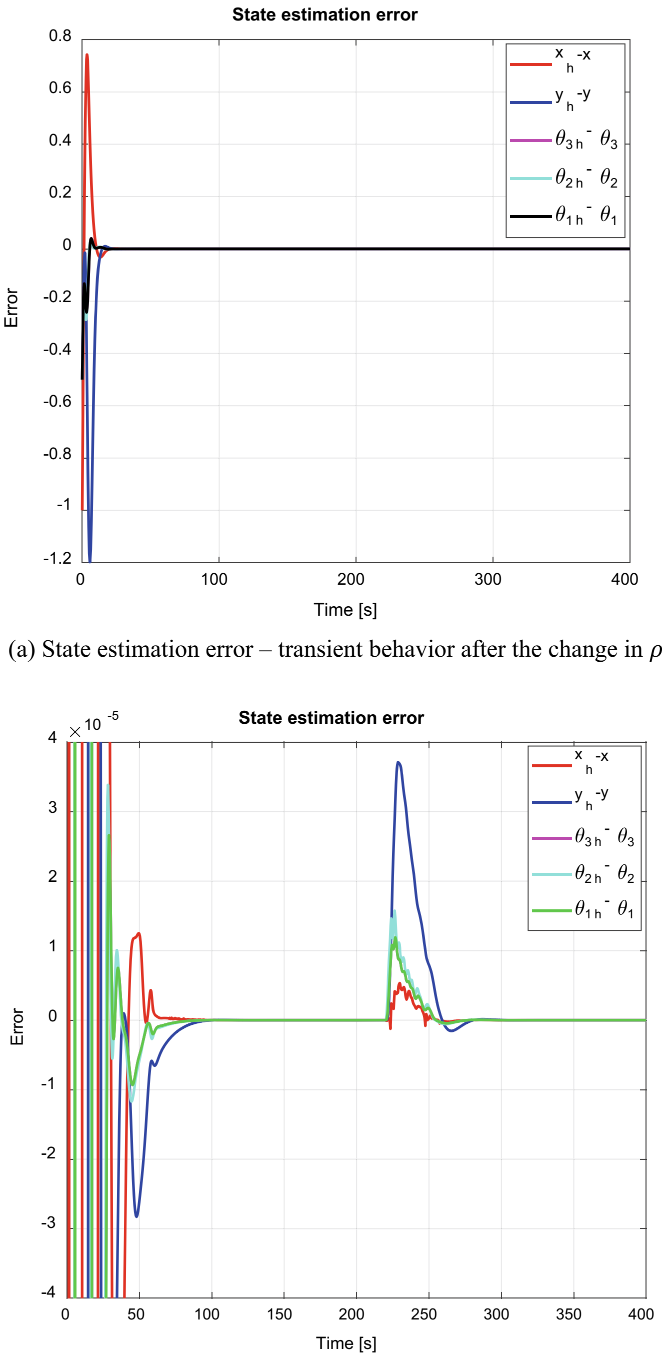
The robustness of the controllers is another issue that must be considered in the future. The kinematic and dynamic control laws were developed based on the nonlinear system model without considering the model uncertainties [42, 43]. So,



(a) Trajectories (reference, simulated) in the plane $x - y$.

(b) Tracking error.

Fig. 10 Trajectories and tracking errors



(a) State estimation error – transient behavior after the change in ρ_1.

(b) State estimation error – zoomed in on the transient behavior at the initial time.

Fig. 11 State estimation error

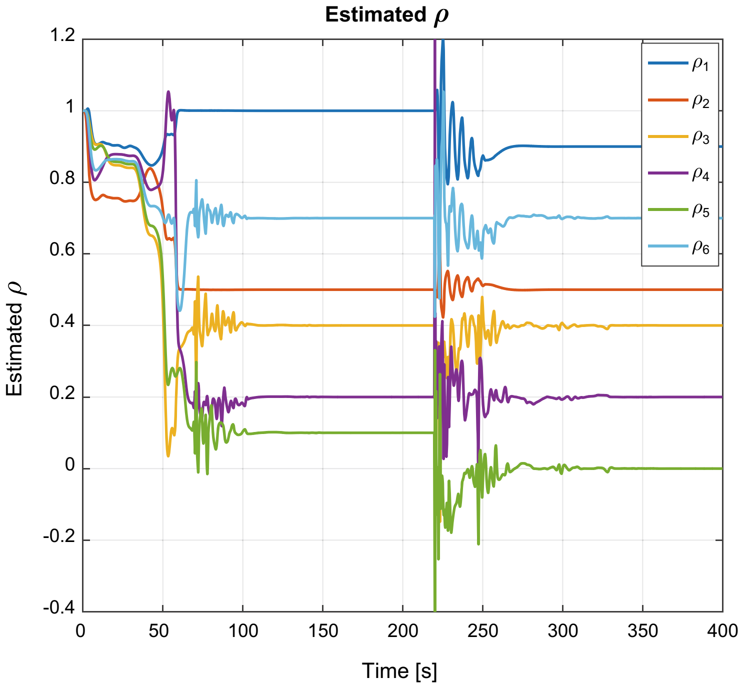


Fig. 12 Estimated ρ (unfiltered)

control design methods with compensation for model uncertainties should be applied. The model-based nonlinear observer design should also be adapted in order to take the model uncertainties into account.

The fault estimation performed in this work considered just the case of constant faults and provided good results in a steady state. As this estimation is used in the adaptive control law, better estimations in the transient behaviors are necessary to enhance the performance. Time-dependent faults should also be considered.

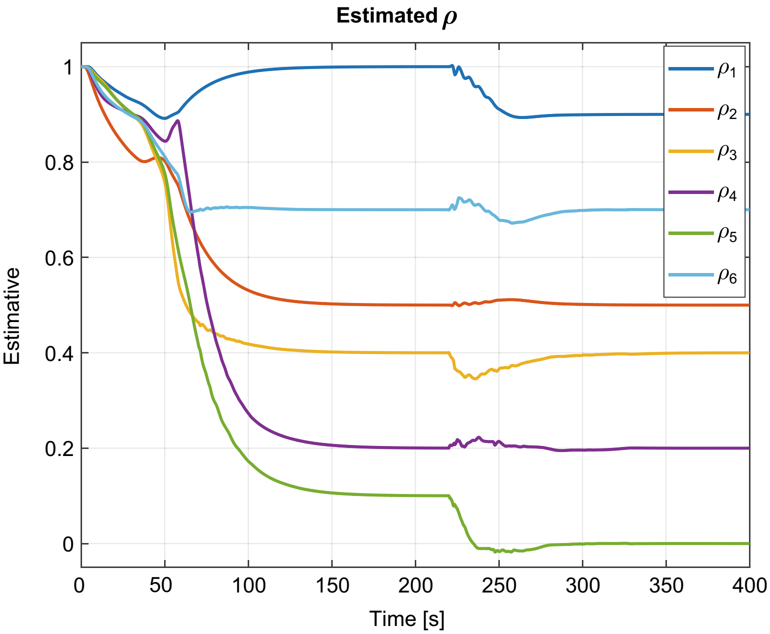
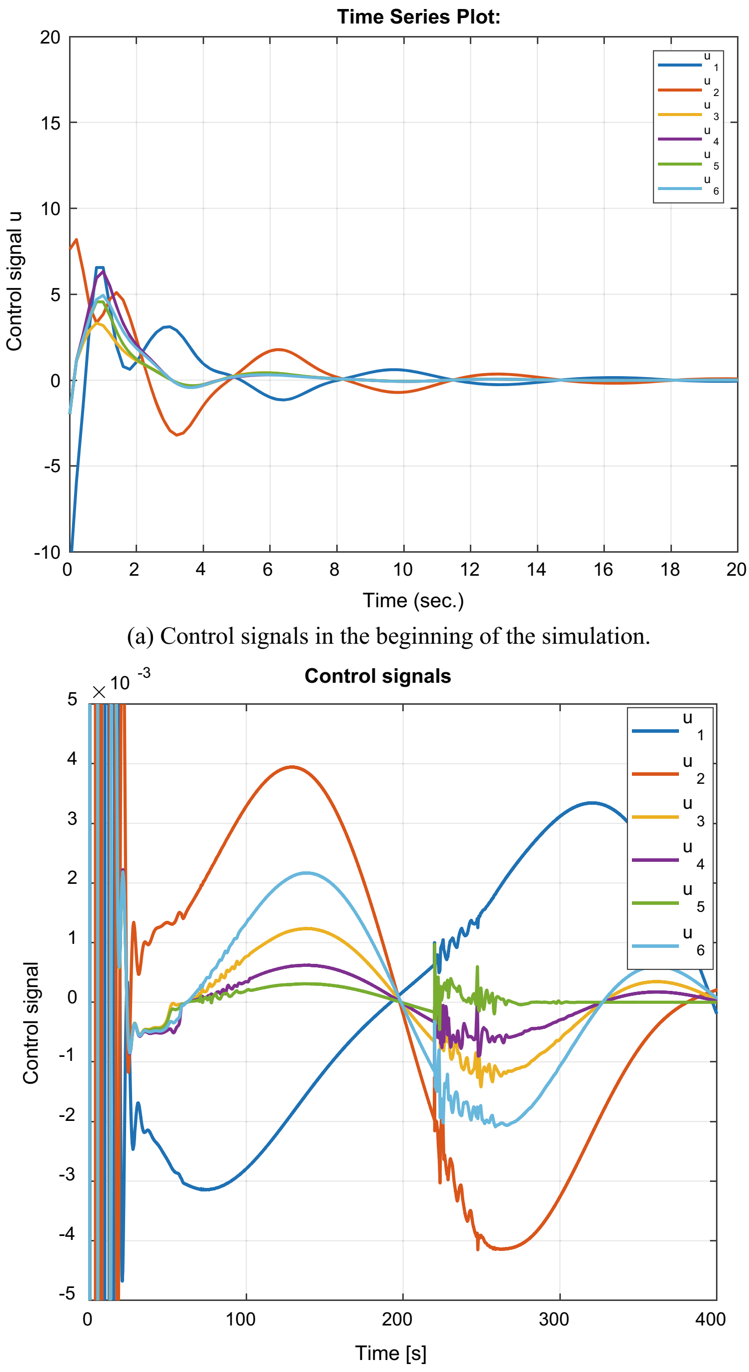


Fig. 13 Estimated ρ (filtered)



(a) Control signals in the beginning of the simulation.

(b) Control signals, zoomed in on the y-axis.

Fig. 14 Control signal

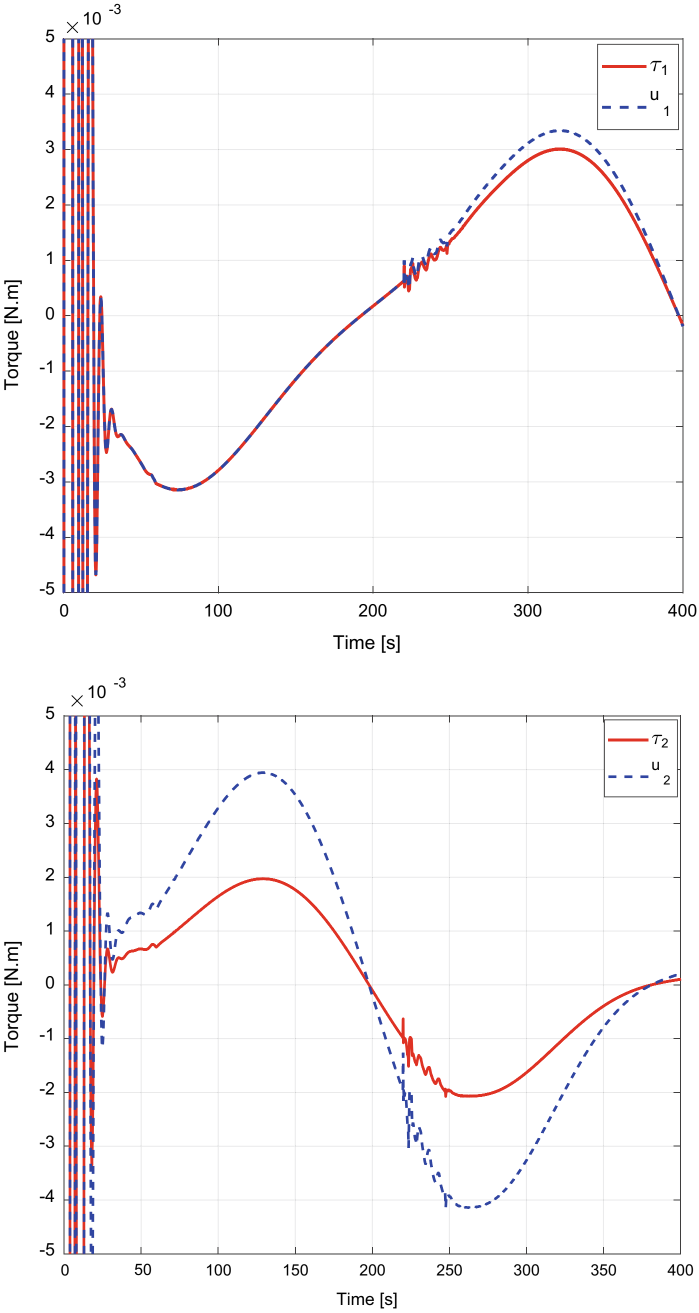


Fig. 15 Comparison torque-control of robot 1

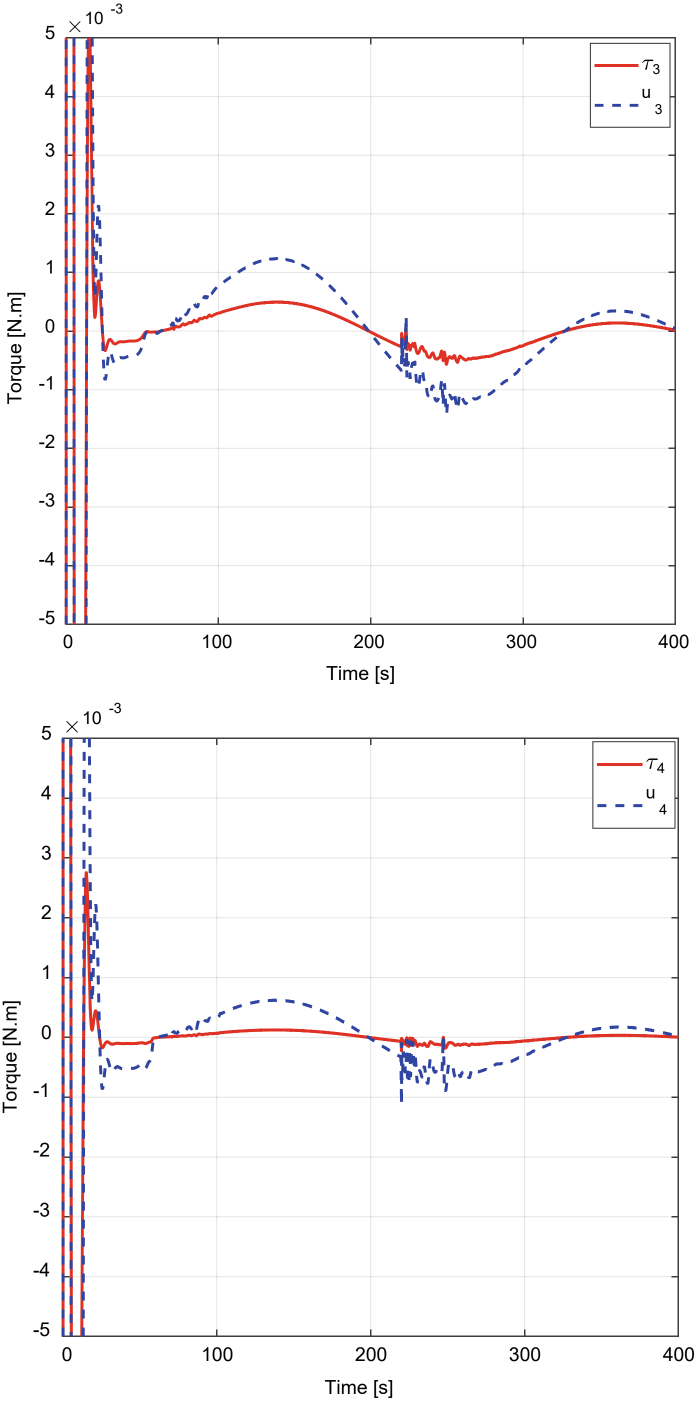


Fig. 16 Comparison torque-control of robot 2

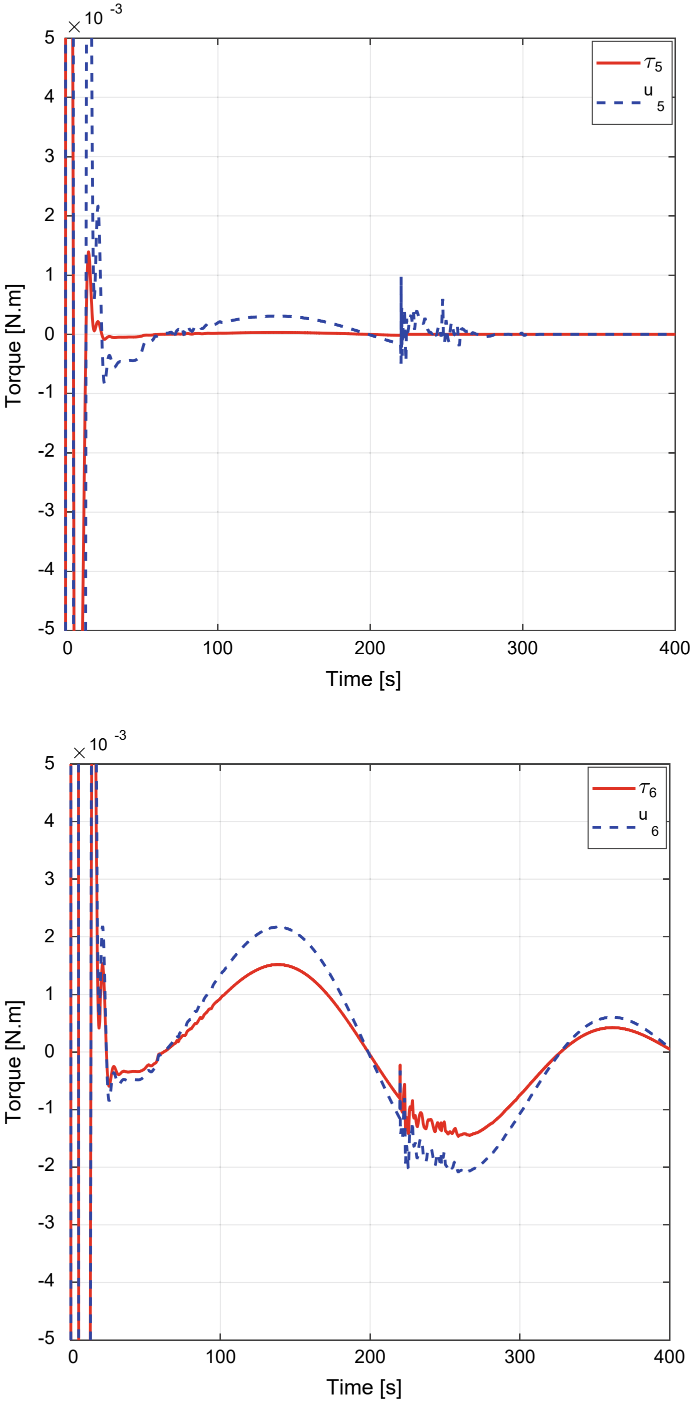


Fig. 17 Comparison torque-control of robot 3

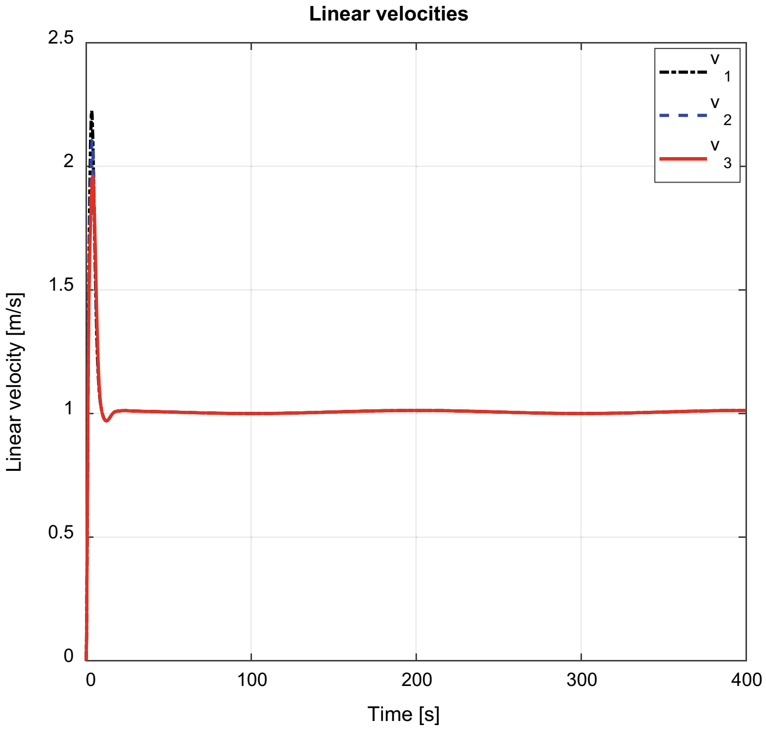


Fig. 18 Linear velocities of robots 1, 2 and 3

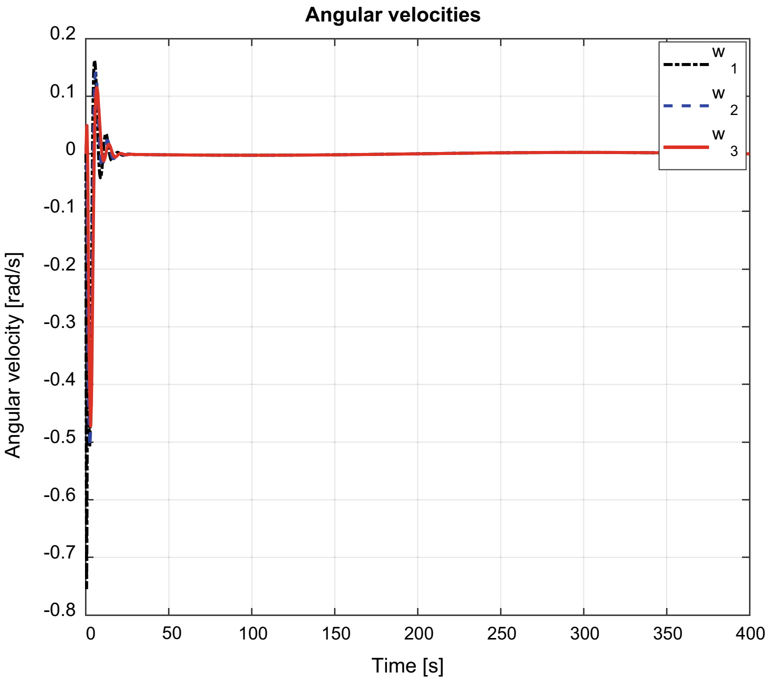


Fig. 19 Angular velocities of robots 1, 2 and 3

References

1. Verma V, Gordon G, Simmons R, Thrun S (2004) Real-time fault diagnosis [robot fault diagnosis]. IEEE Robot Autom Mag 11(2):56–66
2. Yuan J, Sun F, Huang Y (2015) Trajectory generation and tracking control for double-steering tractor-trailer mobile robots with on-axle hitching. IEEE Trans Industr Electron 62(12):7665–7677
3. Zhang Y, Jiang J (2008) Bibliographical review on reconfigurable fault-tolerant control systems. Annu Rev Control 32(2):229–252
4. Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2016) Diagnosis and fault-tolerant control, 3rd edn. Springer, Verlag, Berlin Heidelberg
5. Rotondo D, Puig V, Nejjari F, Romera J (2015) A fault-hiding approach for the switching Quasi-LPV fault-tolerant control of a four-wheeled omnidirectional mobile robot. IEEE Trans Industr Electron 62(6):3932–3944
6. Ji M, Zhang Z, Biswas G, Sarkar N (2003) Hybrid fault adaptive control of a wheeled mobile robot. IEEE/ASME Trans Mechatron 8(2):226–233
7. Kamel MA, Yu X, Zhang Y (2017) Fault-tolerant cooperative control design of multiple wheeled mobile robots. IEEE Transactions on Control Systems Technology, PP (99), 1–9
8. Rotondo D, Cristofaro A, Johansen T, Nejjari F, Puig V (2017) Diagnosis of icing and actuator faults in UAVs using LPV unknown input observers. J Intell & Robot Syst: 1–15
9. Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2016) Diagnosis and Fault-Tolerant Control, 3rd edn. Springer-Verlag, Berlin Heidelberg
10. Ding S (2013) “Model-Based Fault Diagnosis Techniques: Design Schemes Algorithms and Tools”, *2nd edition, series Advances in Industrial Control*. Springer-Verlag, London

11. Chen J, Patton R (2012) Robust Model-Based Fault Diagnosis for Dynamic Systems. Springer Publishing Company, Incorporated
12. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN (2003) A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. Comput Chem Eng 27(3):293–311
13. Venkatasubramanian V, Rengaswamy R, Kavuri SN (2003) A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. Comput Chem Eng 27(3):313–326
14. Venkatasubramanian V, Rengaswamy R, Kavuri SN, Yin K (2003) A review of process fault detection and diagnosis: Part iii: Process history-based methods. Comput Chem Eng 27(3):327–346
15. Gao Z, Cecati C, Ding SX (2015) A survey of fault diagnosis and fault-tolerant techniques - part ii: Fault diagnosis with model-based and signal-based approaches. IEEE Trans Industr Electron 62(6):3757–3767
16. Hoang N, Kang H (2014) Model-based fault diagnosis scheme for wheeled mobile robots. Int J Control Autom Syst 12(3):637–651
17. Zhang X, Polycarpou M, Parisini T (2002) A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. IEEE Trans Autom Control 47(4):576–593
18. Skoundrianos EN, Tzafestas SG (2004) Finding fault - fault diagnosis on the wheels of a mobile robot using local model neural networks. IEEE Robot Autom Mag 11(3):83–90
19. Fourlas GK, Karkanis S, Karras GC, Kyriakopoulos KJ (2014) Model based actuator fault diagnosis for a mobile robot. In: 2014 IEEE International Conference on Industrial Technology (ICIT). Busan, South Korea, pp 79–84
20. Yu M, Chen S, Xia H, Li M, Wang H (2016) Intelligent multiple-fault diagnosis of a mobile robot system in the presence of hide effect. In: 2016 IEEE International conference on mechatronics and automation. Harbin, China, pp 1572–1577
21. de-Gabriel JMG, Mandow A, Lozano JF, Cerezo AG (2015) Mobile robot lab project to introduce engineering students to fault diagnosis in mechatronic systems. IEEE Trans Educ 58(3):187–193
22. Li Z, Jiang W (2013) Active fault-tolerant control for two-wheeled differential drive mobile robot based on fault compensation method. In: 2013 Chinese automation congress. Changsha, pp 359–363
23. Roumeliotis SI, Sukhatme GS, Bekey GA (1998) Sensor fault detection and identification in a mobile robot. In: Proceedings. 1998 IEEE/RSJ International conference on intelligent robots and systems, innovations in theory, practice and applications, vol 3. Victoria, BC, pp 1383–1388
24. Stavrou D, Eliades DG, Panayiotou CG, Polycarpou M (2013) A path correction module for two-wheeled service robots under actuator faults. In: 21st Mediterranean conference on control and automation. Chania, Greece, pp 1119–1126
25. Christensen AL, O'Grady R, Birattari M, Dorigo M (2008) Fault detection in autonomous robots based on fault injection and learning. Auton Robot 24(1):49–67
26. Hachemi L, Guiatni M, Nemra A (2021) Fault diagnosis and reconfiguration for mobile robot localization based on multi-sensors data fusion. World Scientific
27. Chen H, Ma MM, Wang H, Liu ZY, Cai ZX (2009) Moving horizon tracking control of wheeled mobile robots with actuator saturation. IEEE Trans Control Syst Technol 17(2):449–457
28. Akhavan S, Jamshidi M (2000) A NN-based sliding mode control for nonholonomic mobile robots. In: Proceedings of the IEEE International conference on control applications. Anchorage. pp 664–667
29. Martins NA, Bertol D, De Pieri ER, Castelan EB, Dias MM (2008) Neural dynamic control of a nonholonomic mobile robot incorporating the actuator dynamics. In: Proceedings of the international conference on computational intelligence for modeling control and automation. Vienna, pp 563–568
30. Ye J (2008) Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique. Neurocomputing 71(16–18):3373–3378
31. Tao G (2003) Adaptive control design and analysis. John Wiley & Sons, New Jersey

32. Narendra KS, Balakrishnan J (1997) Adaptive control using multiple models. IEEE Trans Autom Control 42(2):171–187
33. Ma Y, Cocquempot V, El Najjar M, Jiang B (2017) Multi design integration based adaptive actuator failure compensation control for two linked 2WD mobile robots. IEEE/ASME Trans Mechatron 22(5):2174–2185
34. Ma Y, Cocquempot V, El Najjar M, Jiang B (2017) Adaptive compensation of multiple actuator faults for two physically linked 2WD robots. In: IEEE transactions on robotics PP (99):1–8
35. Ma Y, Cocquempot V, El Najjar M, Jiang B (2017) Actuator failure compensation for two linked 2WD mobile robots based on multiple-model control. Int J Appl Math Comput Sci (AMCS) 27(4)
36. Ma Y, AL-Dujaili A, Cocquempot V, El Najjar M (2016) An adaptive actuator failure compensation scheme for two linked 2WD mobile robots. In: Advanced Control and Diagnosis, ACD 2016. Lille
37. AL-Dujaili A, Ma Y, El Najjar M, Cocquempot V (2017) Actuator fault compensation in three linked 2WD mobile robots using multiple dynamic controllers. IFAC WC, Toulouse
38. AL-Dujaili A, Cocquempot V, El Najjar M, Ma Y (2017) Actuator fault compensation tracking control for multi linked 2WD mobile robots. In: IEEE MED 2017, 25th Mediterranean conference on control and automation. Valletta, Malta
39. Sørdalen OJ (1993) Conversion of the kinematics of a car with n trailers into a chained form. In: IEEE conference on robotics and automation. pp 382–387
40. Jiang ZP, Nijmeijer H (1999) A recursive technique for tracking control of non-holonomic systems in chained form. IEEE Trans Autom Control 44(2):265–279
41. Fukao T, Nakagawa H, Adachi N (2000) Adaptive tracking control of a nonholonomic mobile robot. IEEE Trans Robot Autom 16(6):609–615
42. Al-Dujaili A, Amjad H, Pereira A, Kasim I (2021) Adaptive backstepping control design for ball and beam system. Int Rev Appl Sci Eng 12(3):211–221
43. Hussein EQ, Al-Dujaili AQ, Ajel AR (2020, June) Design of sliding mode control for overhead crane systems. IOP conference series. Mater Sci Eng 881 Art no 012084

Fractional Order Extended State Observer Enhances the Performance of Controlled Tri-copter UAV Based on Active Disturbance Rejection Control

Alaq F. Hasan, Amjad J. Humaidi, Abdulkareem Sh. Mahdi Al-Obaidi, Ahmad Taher Azar, Ibraheem Kasim Ibraheem, Ayad Q. Al-Dujaili, Ammar K. Al-Mhdawi, and Farah Ayad Abdulmajeed

Abstract In this chapter, an Active Disturbance Rejection Control (ADRC) is developed to trajectory tracking control of a Tricopter Unmanned Aerial Vehicle (UAV). The extended state observer (ESO) is an essential part of ADRC. The performance of ESO has direct impact on the performance of controlled system. This study addresses the design of ADRC for Tricopter UAV based on three types of extended state observers; one is the fractional order extended state observer (FOESO) and the two others are based on nonlinear extended state observer (NESO) and Super

A. F. Hasan
Technical Engineering College, Middle Technical University, Baghdad, Iraq
e-mail: alaaqf.hasan@mtu.edu.iq

A. J. Humaidi (✉)
Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq
e-mail: amjad.j.humaidi@uotechnology.edu.iq

A. Sh. M. Al-Obaidi
Taylor's University, Subang Jaya, Taylor's Lakeside Campus, Selangor DE, Malaysia
e-mail: abdulkareem.mahdi@taylors.edu.my

A. T. Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia
e-mail: aazar@psu.edu.sa; ahmad.azar@fci.bu.edu.eg; ahmad_t_azar@ieee.org

Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

I. K. Ibraheem
Department of Computer Techniques Engineering, Dijlah University College, Baghdad 10022, Iraq
e-mail: ibraheemki@coeng.uobaghdad.edu.iq

A. Q. Al-Dujaili
Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
e-mail: ayad.qasim@mtu.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_14

Twisting Extended State Observer (STESO). A performance comparison has been made between the observers in terms of robustness against variation of parameters and the capability to reject applied disturbance. Tunicate Swarm Algorithm Method (TSA) is used to tune the parameters of ADRC to reach minimum error in order to further improve the dynamic performance of the controlled system. Numerical simulations have been conducted to assess the effectiveness of proposed ADRC based on FOESO in the presence of uncertainty and external disturbance.

Keywords Fractional order extended state observer · Active disturbance rejection control · Tunicate Swarm Algorithm Method · Tricopter UAV

Nomenclature

φ	Roll angle *(rad)*
θ	Pitch angle *(rad)*
ψ	Yaw angle *(rad)*
x	The position of the Tricopter on *x-axis (m)*
y	The position of the Tricopter on *y-axis (m)*
z	The position of the Tricopter on *z-axis (m)*
Ω_i	The angular speed of rotor *i*
U_1	Thrust force *(N)*
U_2	Torque about *x-axis (N/m)*
U_3	Torque about *y-axis (N/m)*
U_4	Torque about *z-axis (N/m)*
k_t	The drag factor
k_f	The thrust factor
l	Distance between the center of a rotor and the center of the Tricopter.

1 Introduction

The unmanned aerial vehicle (UAV) is an aircraft that can be remotely and autonomously piloted or flown by manipulating the aerodynamic forces with the absence of onboard human operator. The UAVs are capable to bear payloads

A. K. Al-Mhdawi
Department of Computer Science, Edge Hill University, Ormskirk, UK
e-mail: Al-Mhdaa@edgehill.ac.uk

F. A. Abdulmajeed
Aeronautical Department, College of Technical Engineering, Alfarahidi University,
Baghdad 10070, Iraq
e-mail: f.ayad@uoalfarahidi.edu.iq

according to aircraft power [1–10]. A vertical take-off and landing (VTOL) aircraft is UAV which is capable of moving from static or dynamic position and it has different movement phases including vertical take-off, hovering, lateral motion and landing. Such VTOL aircrafts can make transitions from movement phase to another. The VTOLs, which have recently emerged, are multi-rotor UAVs and they can activate one or more than one rotor to perform specified task and function [11, 12]. The Tricopter is a special type of multi-rotor aircraft that includes three actuating rotors. There are two categories which can classify the Tri-rotor VTOL UAVs. In the first category, three coaxial rotors are utilized for the flight of UAVs. In order to cancel the yaw motion in this configuration, one motor within each coaxial rotor assembly rotates in the direction opposite to direction of its twin [13]. The other category of the UAV is characterized by having three rotors and a single servo motor is equipped with one of the rotors. The servo motor is responsible for changing the lifting angle of the corresponding single rotor. The yaw motion of the vehicle can be changed by turning the motor in clockwise or counter clockwise direction by the proposed controller. However, the main goal in both configurations is how to stabilize the yaw moment of the UAV [14]. The schematic representation of a Tricopter UAV is illustrated in Fig. 1.

The Tricopter is a six DOF system. Six variables can be detected to express the position of UAV in space (x, y, z, φ, θ and ψ). The variables x, y and z represent the distances from center of mass, along the x, y and z axes, from the fixed Earth inertial frame. The rotation of Tricopter is represented by Euler angle φ, θ and ψ. The roll angle φ, pitch angle θ, and yaw angle ψ denote the rotation around x-axis, y-axis and z-axis, respectively. Due to aerodynamics of three rotors, the Tricopter is typically nonlinear, underactuated, and coupled system. Moreover, the Tricopter is susceptible to wind gust during the flight in low attitude, which influences the flight performance and may lead to instability problems.

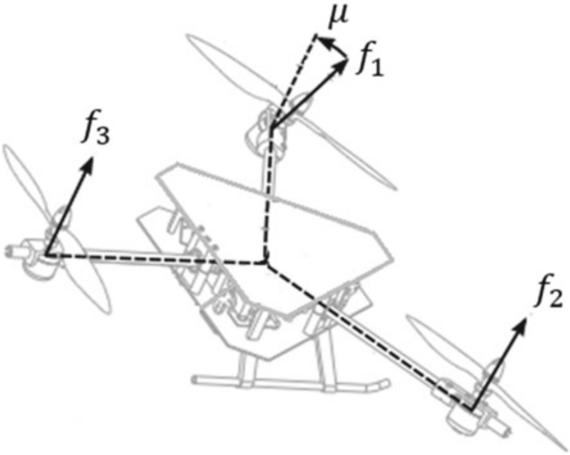


Fig. 1 Schematic representation of a Tricopter UAV

There are four control modes that governing the movements of Tricopter UAV. The first control mode is responsible for attitude control by equally decreasing and increasing the angular velocities of actuating brushless motors as indicated in Fig. 2a. The second control action governs the roll moment by changing the angular velocities of two front actuating rotors, while fixing the speed of the third rotor. This can be embodied by changing the speed of second and third rotors, while fixing the first speed of rotor as shown in Fig. 2b. The pitch movement can be generated by the third mode of control, which varies the velocity of first rotor, while fixing the front rotors angular velocities. This scenario can be illustrated in Fig. 2c. The four-control mode is used to activate the yaw moment by tilting the servo motor in the tail by a small angle as indicated in Fig. 2d. The Tricopter UAV can be guided to fly ahead by increasing the velocity of tail-rotor and reducing the velocities of front rotors so that the frame can be inclined forwards. The UAV can be flown to right side by slowing down the right rotor and speeding up the left rotor and vice versa [13].

The Active Disturbance Rejection Control (ADRC) is a powerful control technique which can deal with very complicated uncertainties of nonlinear systems. This approach has been proposed by Han, during the late 1980s and 1990s, and it can cope with the coupling of unmodeled dynamics, external disturbances, unknown model of zero dynamics and control input. The Extended State Observer (ESO) is the

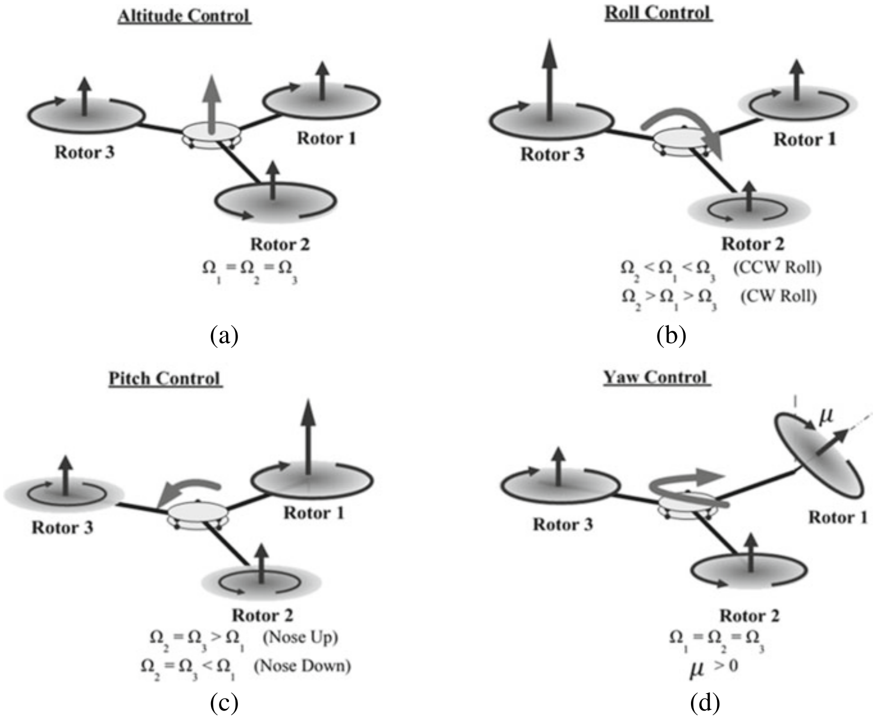

Altitude Control
Rotor 3
Rotor 1
Rotor 2
$\Omega_1 = \Omega_2 = \Omega_3$
(a)
Roll Control
Rotor 3
Rotor 1
Rotor 2
$\Omega_2 < \Omega_1 < \Omega_3$ (CCW Roll)
$\Omega_2 > \Omega_1 > \Omega_3$ (CW Roll)
(b)
Pitch Control
Rotor 3
Rotor 1
Rotor 2
$\Omega_2 = \Omega_3 > \Omega_1$ (Nose Up)
$\Omega_2 = \Omega_3 < \Omega_1$ (Nose Down)
(c)
Yaw Control
μ
Rotor 3
Rotor 1
Rotor 2
$\Omega_1 = \Omega_2 = \Omega_3$
$\mu > 0$
(d)

Fig. 2 The movement actions of Tricopter UAV due to changes in rotor speeds [3]

crucial part of ADRC, which is designed to estimate the total uncertainties. Recently, this control approach has been applied in many control problems and it has shown its outstanding PID nature independent of mathematical model and irrespective of control accuracy [15, 16].

This chapter is organized as follows; Sect. 2 focuses on previous relevant. The Sect. 3 presents the modeling of UAV Tricopter system. A novel design and stability analysis of ADRC based on FOESO is discussed in Sect. 4. The control scheme of the UAV Tricopter system and the optimization problem and solution based on TSA is presented in Sect. 5. The simulated results are presented in Sect. 6. In Sect. 7 highlighted the main concluded points in the numerical simulation.

2 Related Works

Many control design approaches have been developed to address the control of linear mechanical systems [17–23]. Within the defined operating range [24–27], these controllers will function effectively. The linearized models could not provide an exact solution to the control problem due to inherited nonlinearities like as saturation, Coulomb friction, hysteresis, and dead-zones. Because of such strong nonlinearities, state variables are represented in a discontinuous and non-smooth manner. Furthermore, accurate linear approximations to these nonlinearities are impossible to obtain. As a result, in order to achieve acceptable system performance with these nonlinearities, nonlinear control design must be used rather than linear design version [24, 28–39]. Over the last few decades, there have been numerous notable advances in the design of nonlinear systems for diverse practical applications [40–54].

Several control methodologies have been proposed to control the motion of Tricopter system. In [55], Ali et al. have presented the fuzzy regulation, pole-placement, and tracking (RST) control algorithm for controlling the attitude and altitude of Tricopter UAV. Compared to conventional RST control, the fuzzy based RST control showed less tracking error and higher robustness. In [56], Prach et al. proposed the usage of a multiloop structure that comprises a conventional PID control method and linear MPC. The outer-loop PID controller handles the Tricopter position, the inner-loop MPC controller handles the attitude and vertical body velocity of the Tricopter. The results demonstrate the effective tracking of the desired trajectory in the presence of the constraints imposed on the states and control inputs. In [57], NAM et al. has proposed PID controller and PI controller for control of Tricopter UAV with individually tilted main wings for flight maneuvers. The results of simulations and experiments demonstrate the proposed controller, including the roll rate 80% larger than that of the conventional UAV with ailerons as control surfaces. In [58], Song et al. have presented the back-stepping control strategy to design attitude controller. The results demonstrate the effective maneuvering flight of the Tricopter. In [59], Tran et al. developed GA adaptive fuzzy gain scheduling controller for controlling the following parameters of the single tilt Tricopter UAV device: DC motor speed, DC servo motor position, and the altitude of the Tricopter. Compared to optimal adaptive fuzzy gain scheduling controller, GA adaptive fuzzy gain scheduling controller shows better dynamic performance in terms of required saving the operation time, robustness, and reliability. In [60], Yoon et al., applied LQR controller for the attitude control of Tricopter UAV in the presence of disturbance during the hovering statues. The simulation results validated the performance of the designed controller with fast yaw motion control performance.

3 Methodology

In this section, the nonlinear kinematic and dynamic models for the Tri-copter system has been developed based on Newton–Euler method**.** There are number of assumptions which have been made to formulate the dynamic model of the Tricopter UAV for preparing to control design [1]:

Assumption 1: A rigid body of Tricopter structure,

Assumption 2: A symmetric structure of Tricopter is assumed,

Assumption 3: The coincidence of mass center and body center,

Assumption 4: The drag and trust forces are directly related to speed of rotors.

3.1 Kinematic Model

In order to describe the kinematics model of Tricopter kinematics model, one has to define two reference frames: Body-fixed reference frame (B-frame) and Earth-inertial reference frame (E-frame) [60]. The B-frame is attached on the Tricopter's body such that the frame origin is fixed at the mass center. The X-axis of the frame points to forward direction, the Z-axis points to upward direction of Tricopter body, while the Y-axis points to direction which accomplishes the right-hand system. The E-frame is fixed at specified point on ground level and it described by x_E-y_E-z_E notation. Their axes, respectively, points forward, leftward and upwards as indicated in Fig. 3. The position of the mass center of the Tricopter $r = \begin{bmatrix} x & y & z \end{bmatrix}^T$ represents the distance between B-frame and E-frame. The orientation of B-frame with respect to E-frame represents the angular attitude (position) of the Tricopter system.

The roll, pitch and yaw angles (φ, θ and ψ) represent the orientations of Tricopter around the x, y, and z-axes, respectively. If the order of rotation is assumed to be roll (φ), pitch (θ) and yaw (ψ), then according to sequence of rotation principle, the rotation matrix $R_{B \to E}$ can be obtained [61, 62]:

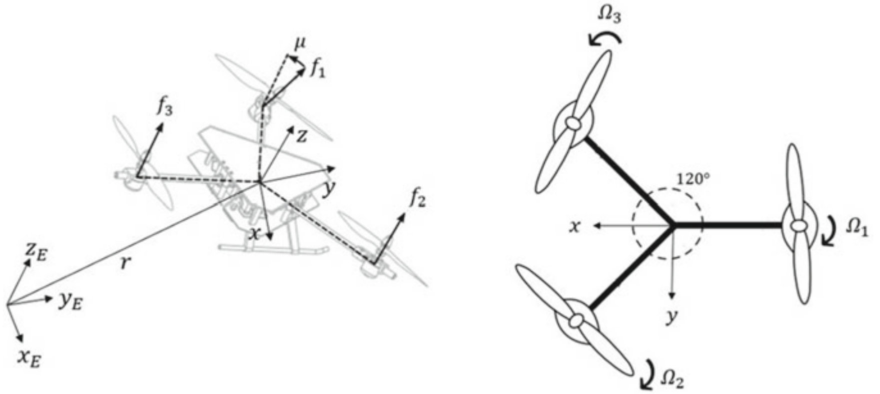


Fig. 3 Configuration of a Tricopter UAV with Earth-inertial reference and body-fixed frames

$$R_{B\to E} = R(z,\psi)\times R(y,\theta)\times R(x,\varphi)$$

$$R_{B\to E} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix}$$

$$R_{B\to E} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} \tag{1}$$

where the abbreviations *s and c* stands for *sin and cos* respectively. One can write the translational kinematic model as follows:

$$\dot{\Gamma}_E = R_{B\to E} V \tag{2}$$

where, the variables V and $\dot{\Gamma}_E$ represent the vector of linear speed with respect to the body B-frame and Earth E-frame, respectively.

3.2 Dynamic Model

One can divide the motion of Tricopter into two sub-motions: the translational motion (linear motion) and the rotational motion (angular motion). Therefore, one can define two motion vectors in the dynamic model to describe these subsystems of motions:

$$\Gamma_E = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{3}$$

$$\Theta_B = \left[\varphi\ \theta\ \psi\right]^T \tag{4}$$

It is instructive to represent the rotational motion equations of UAV with respect to B-frame and the translational motion equations with respect to the E-frame. The translational dynamic model of rigid body of Tricopter can be established based on Euler's first law of motion,

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_{B\rightarrow E}F_B \tag{5}$$

where *m, g* and F_B represent the mass of Tricopter, the acceleration of gravity, and the force of body, respectively. The vector $\ddot{r}$ represents the linear acceleration of body.

$$F_B = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ F_1\sin(\mu) \\ F_2 + F_3 + F_1\cos(\mu) \end{bmatrix}$$

The force component f_y is responsible for eliminating the moment reaction due to tail rotor and it has small value. Setting this component to zero, the force vector becomes

$$F_B = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F_2 + F_3 + F_1\cos(\mu) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ k_f({\Omega_2}^2 + {\Omega_3}^2 + {\Omega_1}^2\cos(\mu)) \end{bmatrix} \tag{6}$$

where k_f denotes the thrust factor and the angular speed of rotor i is denoted by Ω_i. According to the Newton–Euler method, one can derive the equations of rotational subsystem:

$$J\ddot{\Theta}_B = \tau_B - \dot{\Theta}_B \times J\dot{\Theta}_B \tag{7}$$

where the elements of vector τ_B represents the torques acting on the Tricopter in the B-frame, and the matrix J denotes the inertia-matrix of UAV, which can be expressed as follows:

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{8}$$

$$\tau_B = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} l k_f \left({\Omega_2}^2 - {\Omega_3}^2 \right) \\ \frac{1}{2} l k_f \left({\Omega_2}^2 + {\Omega_3}^2 \right) - l k_f {\Omega_1}^2 \cos(\mu) + k_t {\Omega_1}^2 \sin(\mu) \\ - l k_f {\Omega_1}^2 \sin(\mu) - k_t {\Omega_1}^2 \cos(\mu) - k_t {\Omega_2}^2 - k_t {\Omega_3}^2 \end{bmatrix} \tag{9}$$

where k_t denotes the drag factor, l represents the distance between the center of a rotor and the center of the Tricopter. According to Eqs. (5) and (7), the dynamic equations of Tricopter system can be expressed as follows

$$\begin{cases} \ddot{\varphi} = \frac{U_2}{I_x} + \frac{\dot{\theta}\dot{\psi}(I_y - I_z)}{I_x} \\ \ddot{\varphi} = \frac{U_3}{I_y} + \frac{\dot{\varphi}\dot{\psi}(I_z - I_x)}{I_y} \\ \ddot{\psi} = \frac{U_4}{I_z} + \frac{\dot{\theta}\dot{\varphi}(I_x - I_y)}{I_z} \\ \ddot{z} = -g + \frac{1}{m}(cos\theta cos\varphi)U_1 \\ \ddot{x} = (cos\psi sin\theta cos\varphi + sin\psi sin\varphi)\frac{U_1}{m} \\ \ddot{y} = (sin\psi sin\theta cos\varphi - cos\psi sin\varphi)\frac{U_1}{m} \end{cases} \tag{10}$$

4 Analysis of Active Disturbance Rejection Control Schemes

As shown in Fig. 4, the ADRC consists of three main elements, represented by tracking differentiator (TD), extended state observer (ESO) and linear state error feed-back controller (LSEFC) [62, 63]. Indicated in the figure, the task of TD is to quickly supply both original reference signal and its differentiation version as reference input to ADRC structure. The objective of ESO is to estimate the extended state which represent the lumped uncertainties of the system (external disturbances and internal uncertainty). The LSEFC performs the linear combination of the state error to realize the effective control [64].

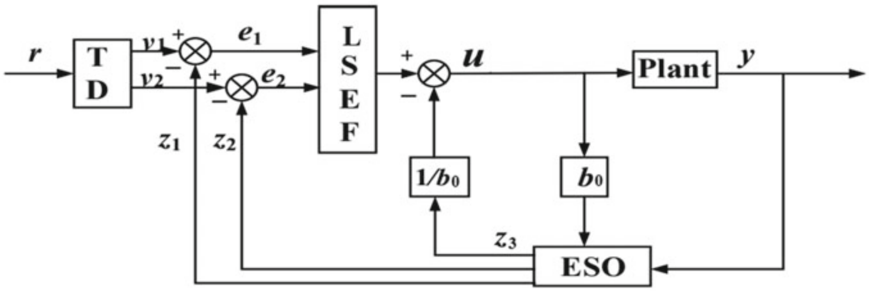


Fig. 4 Structure of ADRC

4.1 Tracking Differentiator (TD)

The tracking differentiator, in principle, is an estimator, which is model-independent. The differentiation dynamic of a certain signal $v(s)$ can be approximated by a linear operator which is expressed by $\frac{s}{\tau s+1}v(s)$. Due to presence of the factor $1/\tau$, this approximation is immune to the noise corrupting the reference signal $v(t)$. The objective of TD is to solve the problem of tracking dynamic differentiation such that to the estimate the elements of input reference signal $v(t)$, represented by v_1, v_2, ..., and v_n, and their derivatives up to the order $(n-1)$ [65].

In this study, a second order nonlinear differentiator has been proposed. The dynamics of TD can be described as [66, 67]:

$$\begin{cases} \dot{v}_1 = v_2 \\ \dot{v}_2 = fhan(v_1 - v, v_2, r, h) \end{cases} \tag{11}$$

where r and h are the parameters of differentiator and h is the sampling period. The function $fhan(v_1 - v, v_2, r, h)$ is described by

$$fhan(v_1 - v, v_2, r, h) = \begin{cases} -r.sign(a), & a > d \\ -r.\frac{a}{d}, & a \le d \end{cases} \tag{12}$$

$$a = \begin{cases} v_2 + \frac{a_o - d}{2} sign(y), & |y| > d_o \\ v_2 + \frac{y}{h}, & |y| \le d_o \end{cases} \tag{13}$$

$$\begin{cases} d = rh \\ d_o = dh \\ a_o = \sqrt{d^2 + 8r|y|} \\ y = v_1 - v + hv_2 \end{cases} \tag{14}$$

4.2 Extended State Observer

4.2.1 Fractional Order Extended State Observer (FOESO)

Fractional calculus has a 300-year history as an extension of conventional calculus. It has been discovered that the fractional order system theory may accurately describe the behavior of many physical systems [68–82]. In fact, real-world processes are almost always fractional order systems [83–107]. Additionally, fractional order controllers have been applied to increase the performance and resilience of closed loop control systems [108–118]. In the control structure of ADRC, there is a core part, represented by ESO. This part is responsible for estimating the lumped uncertainties (parametric and non-parametric uncertainty). To establish the structure of ESO, the state variables of the system has been extended or expanded such that the new state or augmented state has to represent the lumped uncertainties to be estimated by ESO. This constitutes the basic idea of ESO. Therefore, the estimation process does not rely on the model which generates the uncertainty, and the uncertainty of the system can be estimated or observed without the need of direct measurement.

In this study, fractional calculus is applied in ADRC by considering the following system [119, 120]:

$$y^{2\alpha} + a_2 y^{\alpha} + a_1 y = w + bu$$

where w represents the system uncertainty. If there is uncertainty within the input matrix b, the above equation can be written as

$$y^{(2\alpha)} = -a_2 y^{(\alpha)} - a_1 y + w + (b_0 + \Delta b)u$$

The above dynamic equation can be expanded into the following state space form by taking the second-order position loop as the control object,

$$\begin{cases} \zeta^{(\alpha)} = A\zeta + Bu + Eh(t) \\ y = C\zeta \end{cases} \tag{15}$$

where the total uncertainties are assigned by $g(.) = h(t)$, or $g(.) = -a_2 y^{(\alpha)} - a_1 y + w + \Delta b$. This uncertainty function $g(.)$ involves the external and internal disturbances of the system and this is assigned to the expanded state variables $\zeta_3{}^{(\alpha)} = g(.)$. The extended state variable becomes

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix}, A = \begin{bmatrix} 0\ 1\ 0 \\ 0\ 0\ 1 \\ 0\ 0\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1\ 0\ 0 \end{bmatrix}$$

One can propose the following structure of fractional-order extended state observer according to the extended state variable of the system:

$$\begin{cases} \widehat{\zeta}_1{}^{(\alpha)} = \widehat{\zeta}_2 + \beta_1\left(\zeta_1 - \widehat{\zeta}_1\right) \\ \widehat{\zeta}_2{}^{(\alpha)} = \widehat{\zeta}_3 + b_0 u + \beta_2\left(\zeta_1 - \widehat{\zeta}_1\right) \\ \widehat{\zeta}_3{}^{(\alpha)} = \beta_3\left(\zeta_1 - \widehat{\zeta}_1\right) \end{cases} \tag{16}$$

The variables $\widehat{\zeta}_1$, $\widehat{\zeta}_2$, $\widehat{\zeta}_3$ represent, respectively, the estimates of the actual states of ζ_1, ζ_2, ζ_3. The gain coefficients of the extended state observer are represented by the parameters β_1, β_2, β_3. If the estimation errors are defined by $e_1 = \zeta_1 - \hat{\zeta}_1$, $e_2 = \zeta_2 - \hat{\zeta}_2$ and $e_3 = \zeta_3 - \hat{\zeta}_3$, the dynamic of estimation error can be obtained based on Eq. (16),

$$\begin{bmatrix} {e_1}^{(\alpha)} \\ {e_2}^{(\alpha)} \\ {e_3}^{(\alpha)} \end{bmatrix} = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} h(t) \tag{17}$$

The mapping of fractional-order system to the integer-order system of w-plane is used based on the mapping condition $w = s^{(\alpha)}$.

To conduct the stability analysis of observer dynamic, the classical root-locus method can be used,

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} w = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} h(t) \tag{18}$$

In order to tune the parameters of linear FOESO, the bandwidth tuning method of integer-order ESO can be used for this purpose. The idea of observer bandwidth ω_o was firstly proposed by Gao Zhiqiang. One can assign the poles of characteristic polynomial for error dynamics at the same multiple roots. According to Eq. (18), the characteristic equation of the ESO can be given by

$$s^{3\alpha} + \beta_2 s^{2\alpha} + \beta_2 s^{\alpha} + \beta_3 = (s^{\alpha} + w_0)^3 \tag{19}$$

The gains of observer can be parameterized as $[\beta_1, \beta_2, \beta_3] = \left[3w_o, 3w_o^2, w_o^3\right]$. The process of parameter tuning of fractional-order ESO can be simplified if these gains are chosen to be uniquely related to the observer bandwidth ω_o.

4.2.2 Nonlinear Extended State Observer (NESO)

If the observation states of system states ζ_i in Eq. (15) are defined by $\hat{\zeta}_i$, $i = 1, 2, \ldots, n + 1$, and the observation errors are described by $e = \hat{\zeta}_1 - \zeta_1$, then, the structure of nonlinear ESO becomes [121],

$$\begin{cases} e = \hat{\zeta}_1 - \zeta_1 \\ \dot{\hat{\zeta}}_1 = \hat{\zeta}_2 - c_1 e \\ \dot{\hat{\zeta}}_2 = \hat{\zeta}_3 - c_2 fal(e, \alpha_1, \delta) + b_0 u \\ \dot{\hat{\zeta}}_3 = -c_3 fal(e, \alpha_2, \delta) \end{cases} \tag{20}$$

where c_1, c_2 and c_3 represent the gains of observer. The power function $fal(.)$ is a continuous and has linear characteristics near the origin. This description of this function can be expressed as follows:

$$fal = \begin{cases} e\delta^{\alpha-1} & |e| \leq \delta \\ |e|^{\alpha} sgn(e) & |e| > \delta \end{cases} \tag{21}$$

where, α is control parameter and δ is the threshold between smooth and non-smooth intervals. In the interval, where $|e| < \delta$, the function fal is linear and the high-frequency fluctuations, due to high gains in sign function, can be considerably prevented at this region. The other part of the function fal, $|e| > \delta$, has non-smooth characteristics, where the control parameter α can be adjusted within the range $(0 < \alpha < 1)$.

The fal function, described by Eq. (21), can be replaced by some trigonometric continuous functions to effectively reduce the chattering problem. One replacement is to use the nonlinear hyperbolic tangent function $\tanh(fal(e, \beta, c))$, which is much smoother than fal function [122],

$$\tanh(fal(e, \beta, c)) = \beta \tanh(ce) \tag{22}$$

where β and c are the design parameters to be adjusted, and e is the input estimation error. The super-twisting extended observer (STESO) can be expressed as Eq. (23)

$$\begin{cases} e = \hat{\zeta}_1 - \zeta_1 \\ \dot{\hat{\zeta}}_1 = \hat{\zeta}_2 - k_1|e|^{\frac{2}{3}} sign(e) \\ \dot{\hat{\zeta}}_2 = \hat{\zeta}_3 - k_2|e|^{\frac{1}{3}} sign(e) + b_0 u \\ \dot{\hat{\zeta}}_3 = -\tanh(fal(e, k_3, c)) \end{cases} \tag{23}$$

Remark 1 The nonlinear function $g(.)$, which represents the extended state and the lumped uncertainty in the expanded system, has to be continuously differentiable and bounded. However, such assumption is reasonable in most real systems. If this nonlinear characteristic $g(.)$ represents, for example, the friction in the system, then, this value of friction is always bounded. Generally speaking, the system disturbances and uncertainties in most practical applications cannot be infinite.

4.3 Linear State Error Feedback (LSEF)

The block of state error feedback receives the errors between the reference input, or the output of TD, and the output of ESO is defined by [123, 124],

$$\begin{cases} e_1 = v_1 - \widehat{\zeta}_1 \\ e_2 = v_2 - \widehat{\zeta}_2 \end{cases} \tag{24}$$

Based on Eq. (24), the LSEF will generate the control signal u_o, which combines the error signal and its differential as follows:

$$u_o = k_p e_1 + k_d e_2 \tag{25}$$

One can establish the feedback control law according to the following equation

$$u = \frac{u_o - \widehat{\zeta}_3}{b_0} \tag{26}$$

5 Proposed Control System

According to dynamic characteristics of Tricopter model, there are four channels of motion, represented by motions in z-channel (attitude), y-channel, x-channel and the yaw channel (angle) [125]. Therefore, one can represent the double closed-loop control scheme as shown in Fig. 5.

If U_x and U_y are considered as the orientations of the total thrust force U_1, where U_1 is responsible for the motion of UAV in both x and y directions. Then, φ_d and θ_d can be found from Eqs. (27) and (28):

$$\varphi_d = \arcsin\big((U_x sin\psi_d - U_y cos\psi_d)/U_1\big) \tag{27}$$

$$\theta_d = \arcsin\big((U_x cos\psi_d + U_y sin\psi_d)/cos\varphi_d U_1\big) \tag{28}$$

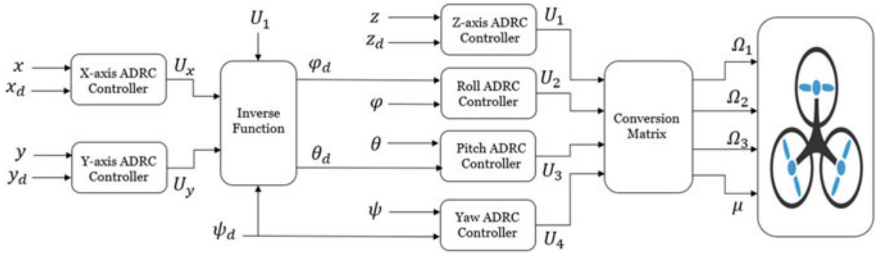


Fig. 5 Control scheme

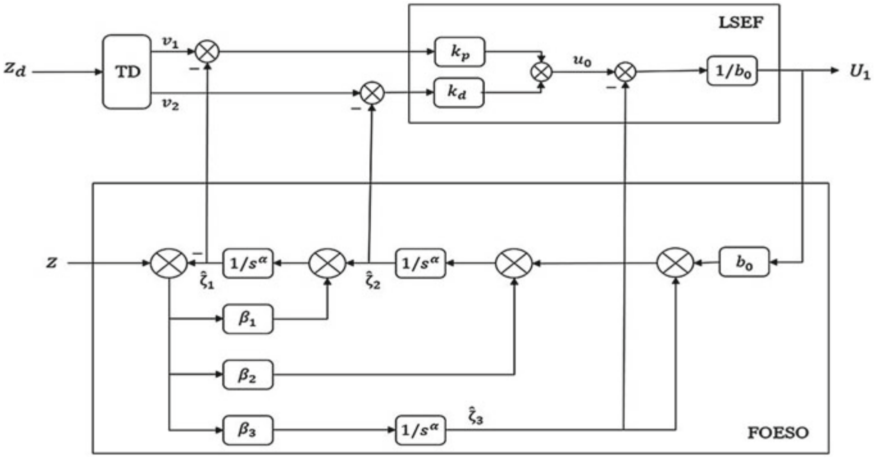


Fig. 6 ADRC based on FOESO scheme

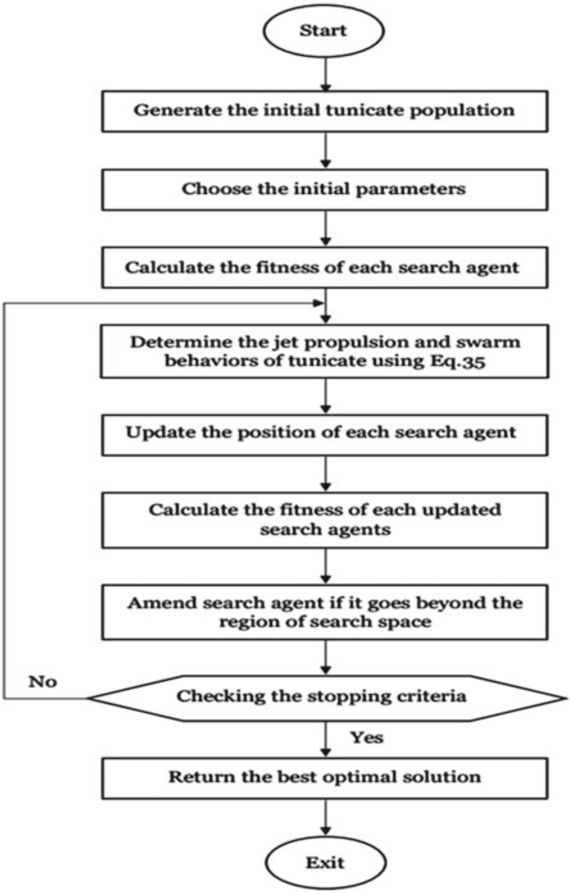
In each channel, the controller is designed in the same manner. For example, Fig. 6 shows the schematic representation of designed controller based on ADRC in the attitude z-channel. This applies to all motion control in the other channels of Tricopter. In this paper, Tunicate Swarm Algorithm will be applied to found the optimal values of the design parameters in such a way an optimal performance of the closed-system dynamic could be reached.

5.1 Tunicate Swarm Algorithm

The Tunicate Swarm Algorithm (TSA) is a bio-inspired optimization algorithm. This metaheuristic algorithm mimics the jet-propulsive motion and swarm behaviors of tunicates during their navigation for foraging in order to survive at ocean depths. The tunicates are cylindrical-shaped with bright bio-luminescence. They live at depth of 500–800 m and grow up to a few millimeters in size. The tunicate is closed at one end and opened at the other end, which produces jet propulsion of water via atrial syphons. This jet propulsion helps these animals to move and mitigate around the ocean [126, 127].

In order to develop the mathematical model that represents and mimics the behaviors of jet propulsion, the tunicate has to satisfy the three conditions: collision-avoidance between candidate solutions, proceeding and stepping towards best solution, and sticking near best solution. The flowchart which depicts the step of TSA is indicated in Fig. 7 depicts the steps of process flow chart of TSA for parameter tuning.

Fig. 7 Flowchart of the TSA algorithm [24]


Start
Generate the initial tunicate population
Choose the initial parameters
Calculate the fitness of each search agent
Determine the jet propulsion and swarm behaviors of tunicate using Eq.35
Update the position of each search agent
Calculate the fitness of each updated search agents
Amend search agent if it goes beyond the region of search space
No
Checking the stopping criteria
Yes
Return the best optimal solution
Exit

5.1.1 Prevent Collisions Between Candidate Solutions

In order to prevent collision between candidate solutions (tunicates), one has to determine the vector $\overrightarrow{A}$ which is applied to calculate the position of new search agent

$$\overrightarrow{A} = \frac{\overrightarrow{G}}{\overrightarrow{M}} \tag{29}$$

$$\overrightarrow{G} = c_2 + c_3 - \overrightarrow{F} \tag{30}$$

$$\overrightarrow{F} = 2 \times c_1 \tag{31}$$

where, $\overrightarrow{F}$ represents the advection of water flow in the ocean, $\overrightarrow{G}$ denotes the gravity force. The variables c_i, $i = 1, 2, 3$, represents random real numbers lying in the range [0,1]. Next, one has to calculate the social force $\overrightarrow{M}$ between the search agents based on the following equation

$$\overrightarrow{M} = [P_{min} + c_1 \times P_{max} - P_{min}] \tag{32}$$

where P_{min} and P_{max} denotes the initial speed and subordinate speed, respectively, which are set to $P_{min} = 1$ and P_{max}.

5.1.2 Step More Toward the Location of the Best Solution

After ensuring successful conflict-avoidance with neighbors, the search agents proceed towards the finest neighbors:

$$\overrightarrow{PD} = \left|\overrightarrow{FS} - rand \times \overrightarrow{P}_p(x)\right| \tag{33}$$

where the rand-function is used to generate random real number in the range [0, 1], $\overrightarrow{PD}$ represents the total distance between the food source and the search agent, $\overrightarrow{FS}$ denotes the position of food source, $\overrightarrow{P}_p(x)$ denotes the positions of tunicates, and x indicates to current iteration.

5.1.3 Stick Close to the Best Solution

The position of search agent could be established as the leading search agent,

$$\vec{P}_p(x) = \begin{cases} \overrightarrow{FS} + \vec{A} \times \overrightarrow{PD}, \, if \, rand \geq 0.5 \\ \overrightarrow{FS} - \vec{A} \times \overrightarrow{PD}, \, if \, rand < 0.5 \end{cases} \tag{34}$$

To update the positions of all tunicates with respect to positions of the first two tunicates, the following update equation is used:

$$\overrightarrow{P}_p(x+1) = \frac{\overrightarrow{P}_p(x) + \overrightarrow{P}_p(x+1)}{2 + c_1} \tag{35}$$

where the updated position of the tunicates is denoted by $\overrightarrow{P}_p(x + 1)$.

6 Results and Discussion

This section conducted a comparison study among three schemes of ADRC. The first scheme considered the case of optimal ADRC based on FOESO, the second scheme considered the case of optimal ADRC based on NESO, while the last control structure addressed the case of optimal ADRC based on STESO. The performances of these proposed control schemes have evaluated according to computer simulations, which has been implemented in MATLAB programming format. The Ode45 has been used as the numerical solver to obtain the simulated results. Table 1 listed the system parameters of UAV (Tricopter) [128].

The desired inputs (x_d, y_d, z_d, ψ_d) is assumed to be determined from the Eq. (36):

$$\begin{cases} x_d = 0.5\cos(\frac{pi}{2}t) \\ y_d = 0.5\sin(\frac{pi}{2}t) \\ z_d = 2 - \cos(\frac{pi}{2}t) \\ \psi_d = 0 \end{cases} \tag{36}$$

The optimization algorithm based on TSA results in optimal ADRC algorithm based on FOESO, optimal ADRC algorithm based on NESO and optimal ADRC algorithm based on STESO. In order to evaluate the performance of proposed controllers under optimization algorithm, the Root Mean Square Error (RMSE) has been adopted as index of performance evaluation and it has been included as the cost function within the algorithm of optimizer. The minimization of cost function was the basis of optimization search and problem. Table 2 lists the optimal values of design parameters for each element of ADRC for various schemes based on TSA.

In this chapter, there are three scenarios have to be considered: the first scenario focuses on disturbance-free and uncertainty-free and cases, the second scenario takes into account the variation in parameters of Tricopter system of about 5% over their

Table 1 The parameters of Tricopter system

Description of parameter	Symbol	Value (unit)
Mass of tricopter	m	0.85 kg
Moment of inertia for pitch	I_y	0.0480 kg.m^2
Moment of inertia for roll	I_x	0.0430 kg.m^2
Moment of inertia for yaw	I_z	0.0770 kg.m^2
Arm length	l	0.180 m
Rotational moment of inertia for motor	I_M	1.97 × 10^{-6} kg.m^2
Ground acceleration	g	9.81 m/s^2
Aerodynamic moment constant	k_t	2.88 × 10^{-7} N.m.s^2
Aerodynamic force constant	k_f	1.970 × 10^{-6} N.s^2

Table 2 The optimized design parameters of ADRC based on FOESO, NESO and STESO using TSA for optimization

	φ	θ	ψ	x	y	z	
TD	R_h	–	–	1021.7	1114.1	1250.3	1121.9
	h	–	–	0.001	0.001	0.001	0.001
LSEF	k_p	55.05	55.05	12.46	4.84	4.84	25.46
	k_d	11.5	14.86	7.128	3.23	3.23	13.37
	b_0	11.416	11.207	11.9	0.581	0.581	0.98
FOESO	w_o	76.6005	76.6005	58.70	91.5337	91.5337	99.2650
	α	0.9930	0.9930	0.9860	0.9992	0.9992	0.9974
NESO	c_1	497.506	497.3798	297.38	599.1240	743.3354	78.8029
	c_2	255.066	214.3201	233.72	684.5629	491.4868	634.5752
	c_3	306.648	133.6726	133.6726	700.9868	689.4817	449.0615
	α_1	0.5047	0.4730	0.573	0.9388	0.7900	0.1985
	α_2	0.4529	0.1608	0.278	0.9680	0.9412	0.2389
	δ	93.114	174.7876	76.79	1046.5	606.7838	0.8114
STESO	k_1	16.3004	34.9866	26.400	27.2567	27.2567	104.7019
	k_2	190.5999	310.0170	270.600	160.4990	162.4990	46.9838
	k_3	440.9333	79.1049	320.94	131.1760	131.1760	457.1791
	c	66.9175	28.6446	46.92	3.1141	3.1141	536.1739

nominal value, while the third scenario considers disturbance application on the Tricopter system.

Figures 8, 9, 10, 11 and 12 show the response for each controlled variable for the uncertainty-free and disturbance-free case based on ADRC with fractional order state observer, nonlinear state observer and super twisting state observer. The control effort due to optimal ADRCs based on FOESO, NESO and STESO is indicated in Figs. 13, 14, 15 and 16. Figures 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 and 31 show the estimation errors of the controlled variable and their rate of change and the estimation of the system dynamics for the uncertainty-free and disturbance-free case based on the proposed observers. The simulation results of 2D and 3D position tracking for the designed controllers are shown in Fig. 32. Table 3 reports numerically the evaluation of performances for proposed controllers. The RMSE, ISE and ITAE have been chosen as the indices for performance evaluation. It can be noticed from these figures that the ADRC algorithm based on FOESO achieves high motion precision as compared to ADRC algorithm based on NESO and STESO and that's because of the ADRC algorithm based on FOESO's superiority in the estimation process. Where the FOESO can estimate the controlled variable, the rate of change of the controlled variable and dynamics system much better than the NESO and STESO.

In the second scenario, the above simulation is repeated in the presence of uncertainty. The uncertainty has been represented by the uncertainty in parameters of

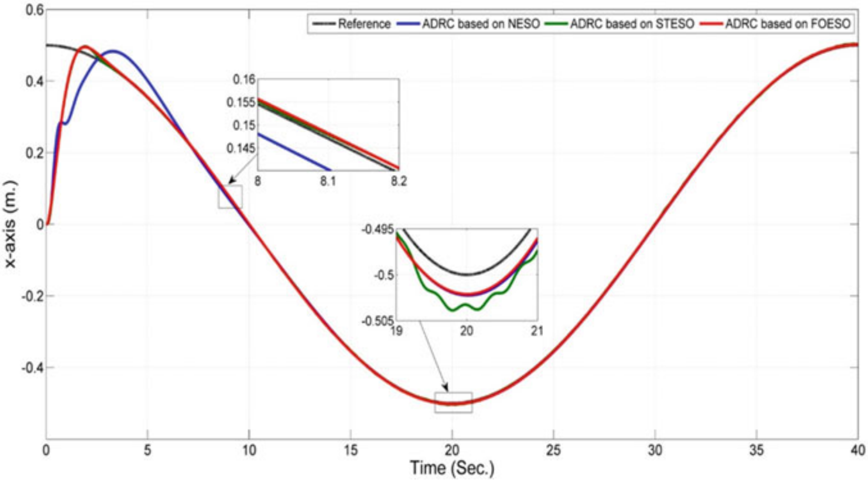


Fig. 8 The responses of x-axis due to ADRC algorithm based on FOESO, NESO and STESO

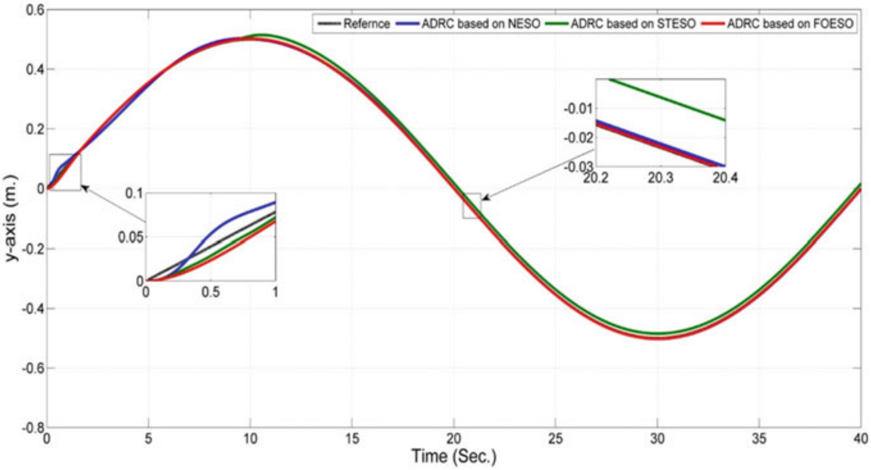


Fig. 9 The responses of y-axis due to ADRC algorithm based on FOESO, NESO and STESO

system equal to 5% over their nominal values. Figures 33, 34 and 35 show the response for each controlled variable, for the controlled system based on ADRC using FOESO, NESO and STESO. The control signals generated by ADRC due to using proposed observers are depicted in Figs. 36, 37, 38 and 39. The simulation results of 2D position tracking for the designed controllers are shown in Fig. 40.

In the last scenario, the disturbance due to wind gust is taken into account and the performances of optimal ADRC based on FOESO, NESO and STESO are evaluated based on computer simulation. The behavior of applied disturbance is described in

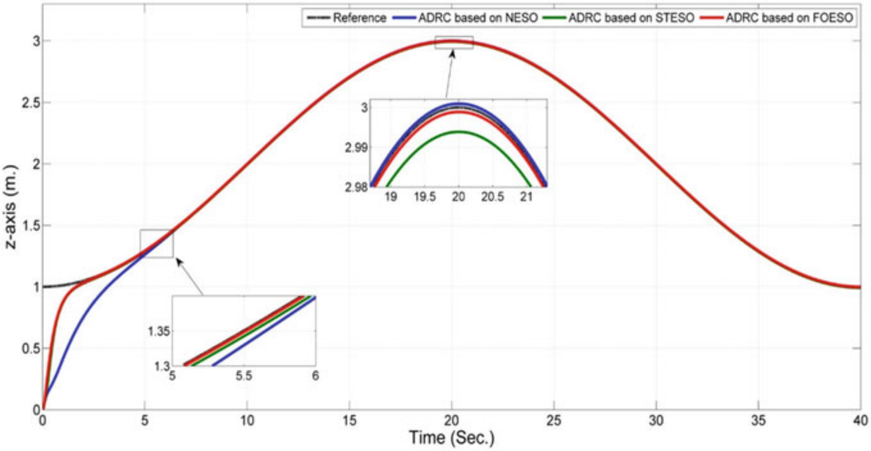


Fig. 10 The responses of z-axis due to ADRC algorithm based on FOESO, NESO and STESO

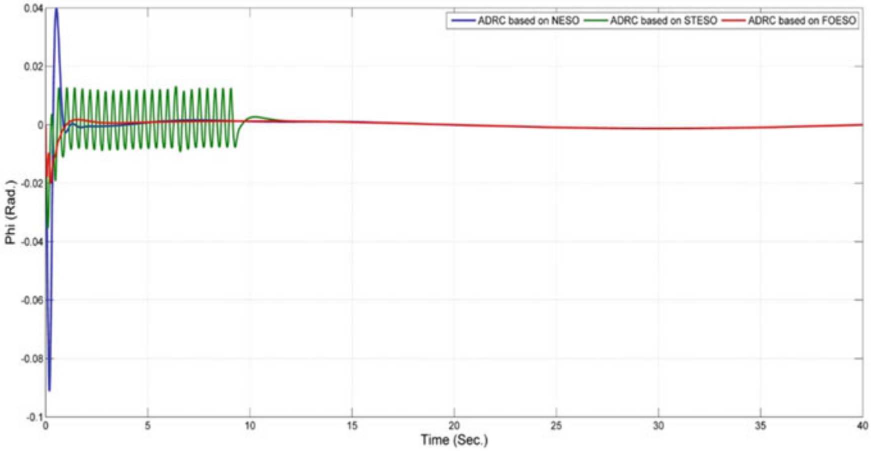


Fig. 11 The responses of φ signal due to ADRC algorithm based on FOESO, NESO and STESO

Eq. (37). The response of controlled variables based on optimal controllers in the presence of wind disturbance has been illustrated in Figs. 41, 42 and 43. Figures 44, 45, 46, 47 and 48 shows the control effort and 2D position tracking for the designed controllers under the action of disturbance. The previous figures show that the ADRC based on FOESO has a good dynamic performance under the applied disturbance and a perceived failure in the performance of ADRC based on NESO and STESO algorithms.

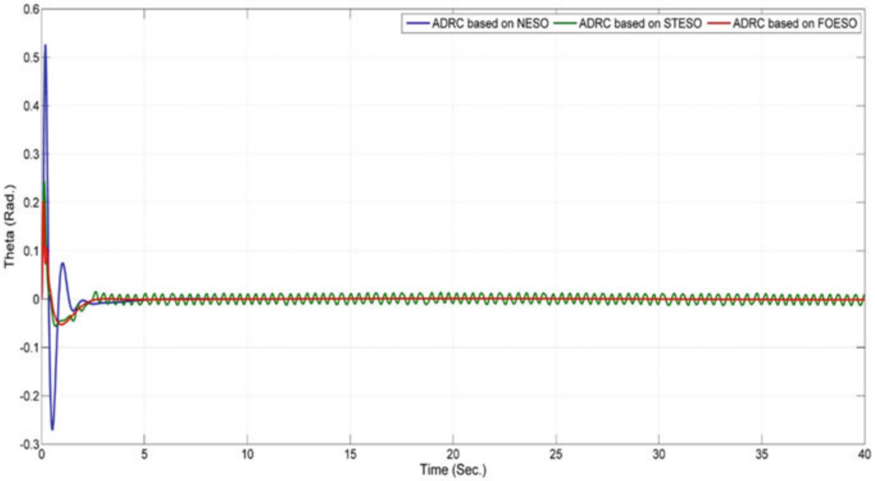


Fig. 12 The responses of θ signal due to ADRC algorithm based on FOESO, NESO and STESO

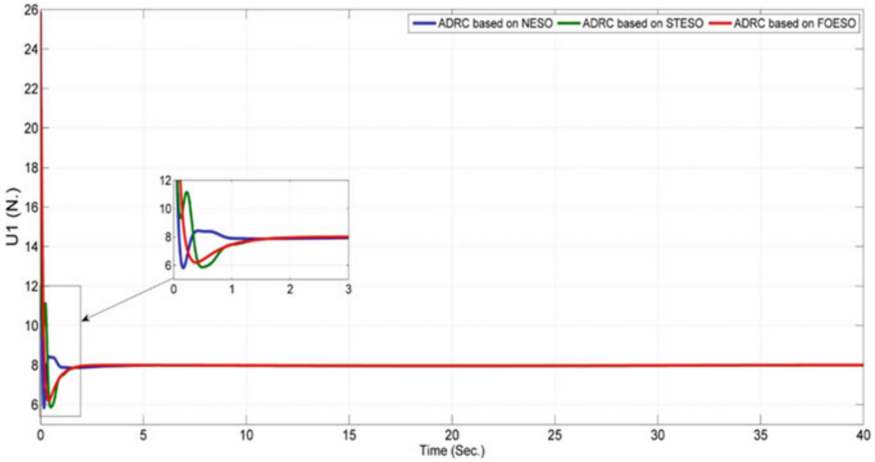


Fig. 13 Shows the control effort (U_1) due to both optimal ADRC based on FOESO, NESO and STESO

$$\begin{cases} D_\varphi = 0.2\sin(0.4t) \\ D_\theta = 0.2\cos(0.5t) \\ D_\psi = 0.2\cos(0.7t) \\ D_X = 0.2\cos(0.4t) \\ D_Y = 0.2\sin(0.5t) \\ D_Z = 0.2\sin(0.7t) \end{cases} \tag{37}$$

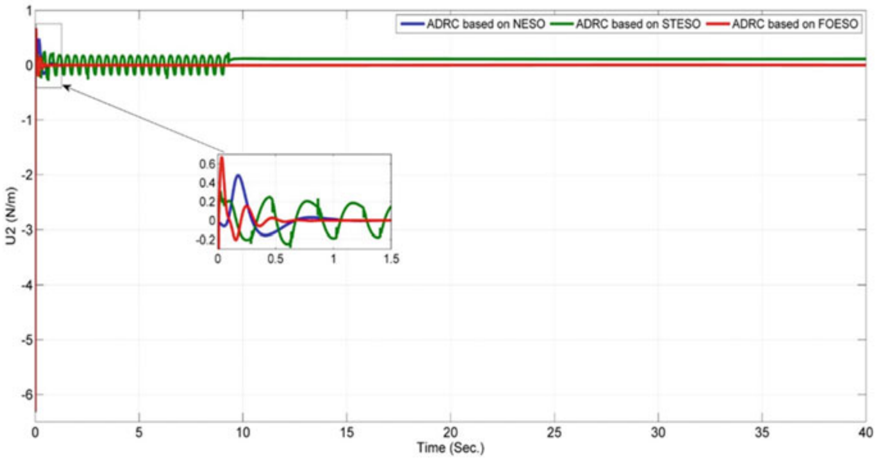


Fig. 14 Shows the control effort (U_2) due to both optimal ADRC based on FOESO, NESO and STESO

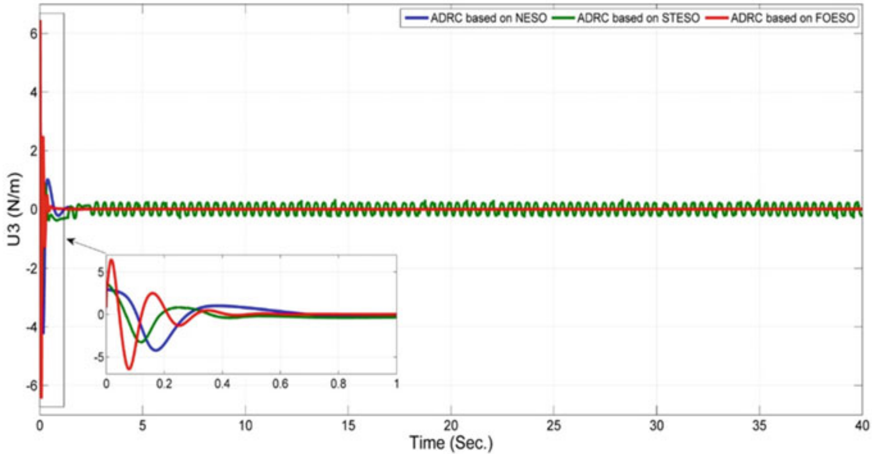


Fig. 15 Shows the control effort (U_3) due to both optimal ADRC based on FOESO, NESO and STESO

7 Conclusion

In this chapter, the ADRC algorithm based FOESO, NESO and STESO have been developed to control the Tricopter system. In order to improve the performance of proposed versions of ADRC, their design parameters has been optimally tuned based on Tunicate Swarm Algorithm. A comparison study in performance has been made between the optimal ADRC based FOESO and the other schemes of ADRC, represented by optimal ADRC based on NESO and optimal ADRC based on STESO.

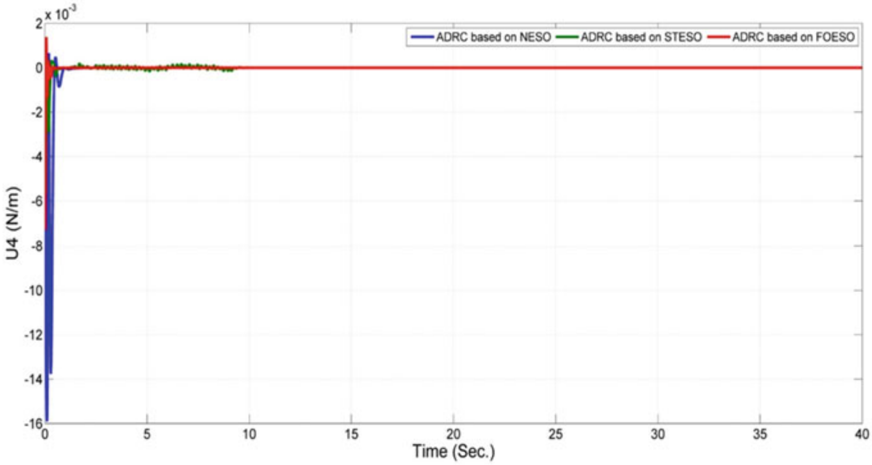


Fig. 16 Shows the control effort (U_4) due to both optimal ADRC based on FOESO, NESO and STESO

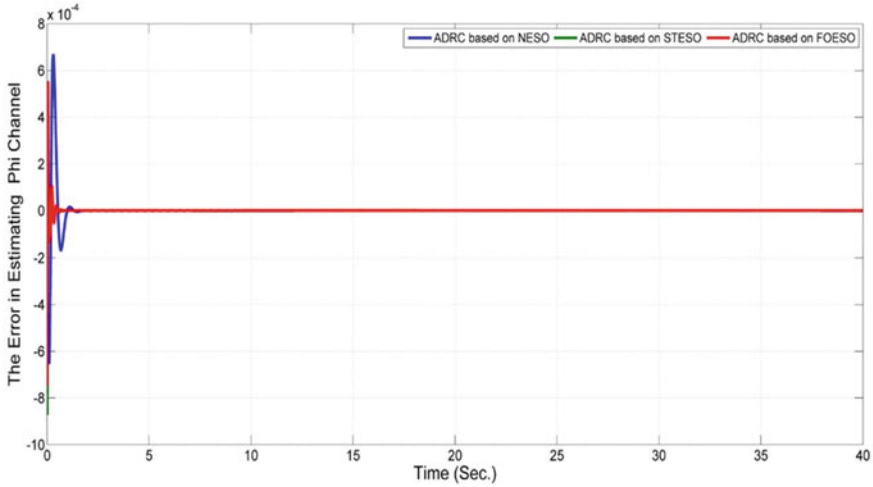


Fig. 17 The estimation errors of φ signal due to FOESO, NESO and STESO

The evaluation of controlled Tricopter based on the three control schemes have been verified via numerical simulation within the environment of MATLAB programming format. The simulated results showed that the optimal ADRC based on FOESO has better dynamic performance than that based on optimal ADRC based on NESO and optimal ADRC based on STESO in terms of tracking error and transient characteristics. In order to extend this study for future work, other optimization techniques, like Butterfly Optimization Algorithm, Grey-Wolf Optimization [129], Social Spider

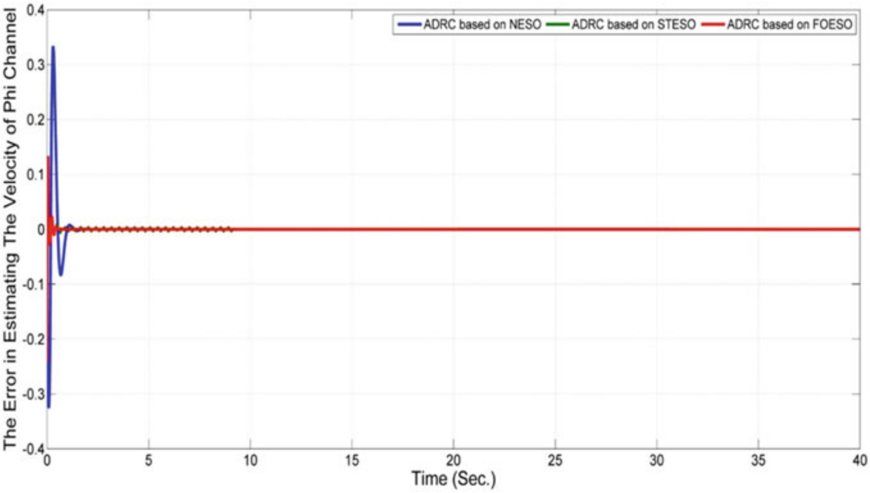


Fig. 18 The estimation errors of the rate of change of φ signal due to FOESO, NESO and STESO

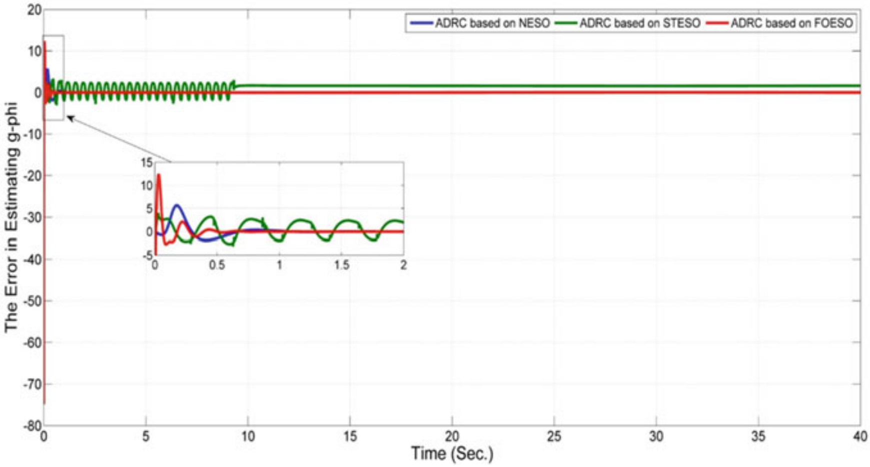


Fig. 19 The estimation errors of g_{φ} due to FOESO, NESO and STESO

Optimization [27], Whale Optimization Algorithm [130, 131], and Particle Swarm Optimization can be applied to tune the design parameters of ADRC schemes to show their effects on performance of these control schemes [132].

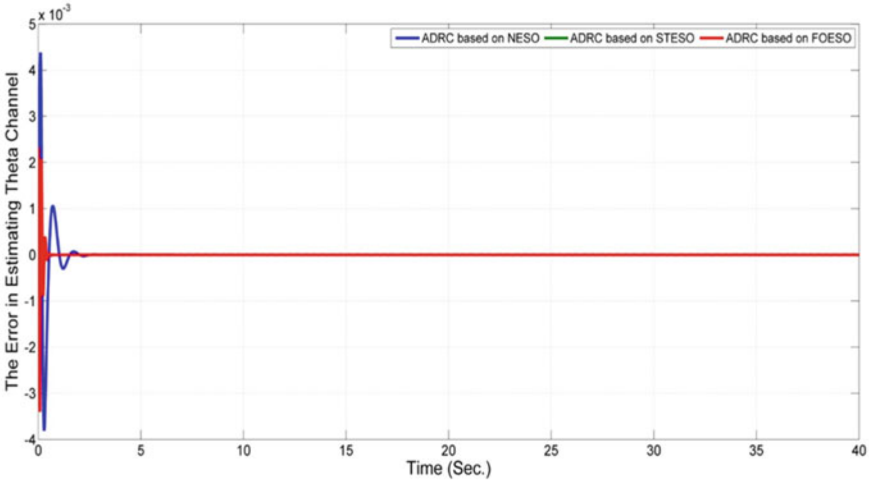


Fig. 20 The estimation errors of θ signal due to FOESO, NESO and STESO

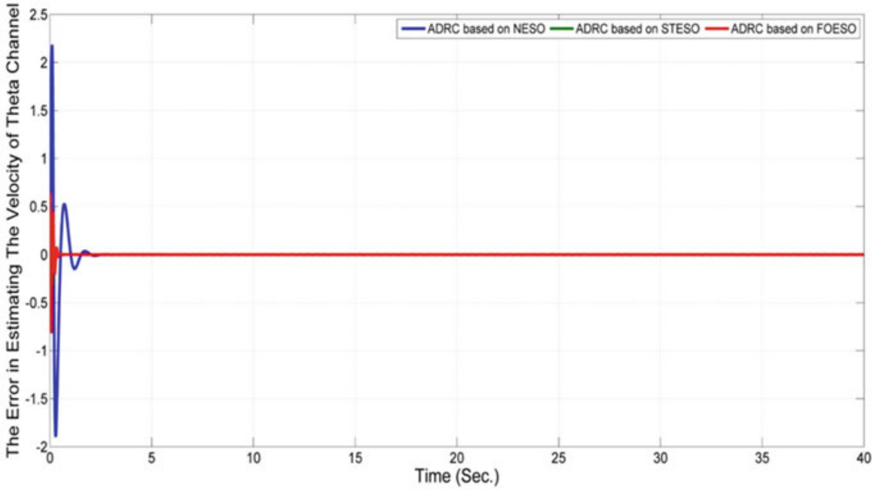


Fig. 21 The estimation errors of the rate of change of θ signal due to FOESO, NESO and STESO

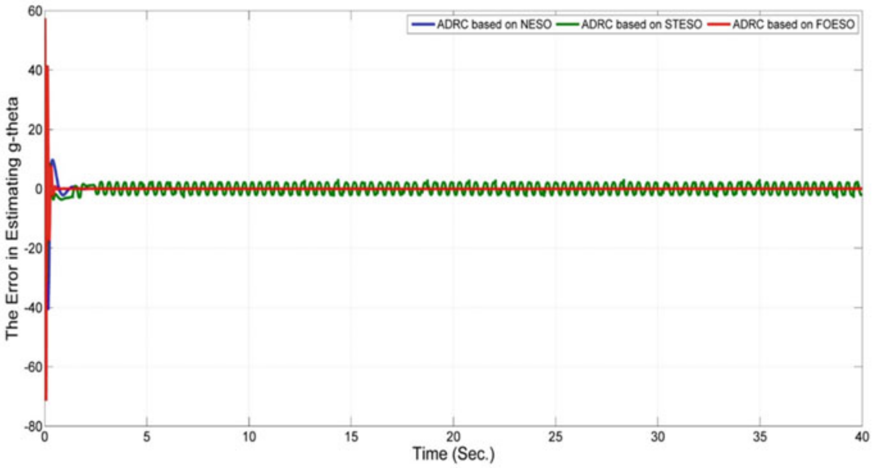


Fig. 22 The estimation errors of g_θ due to FOESO, NESO and STESO

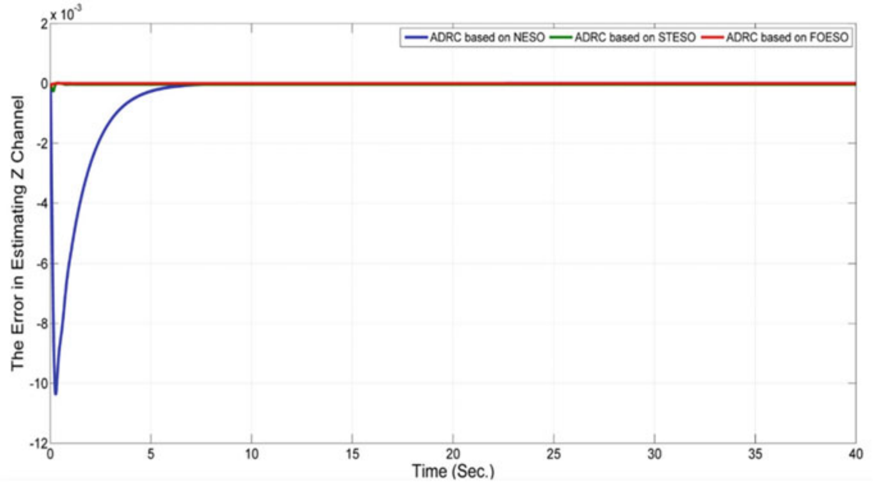


Fig. 23 The estimation errors of z signal due to FOESO, NESO and STESO

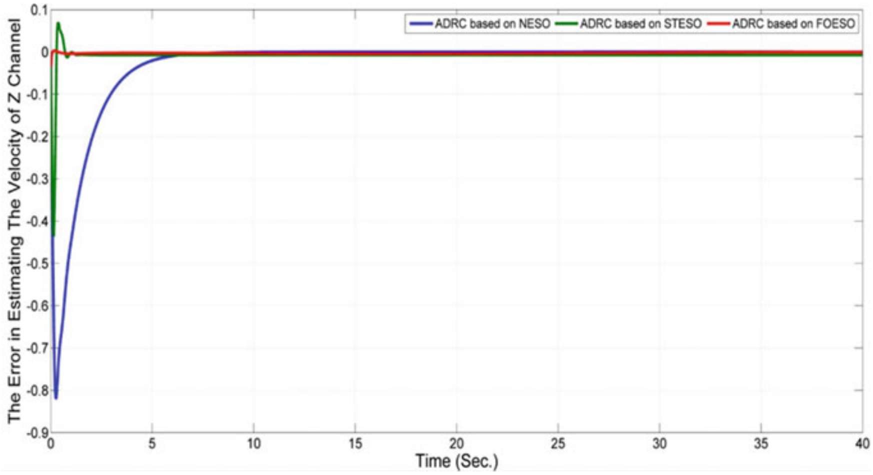


Fig. 24 The estimation errors of the rate of change of z signal due to FOESO, NESO and STESO

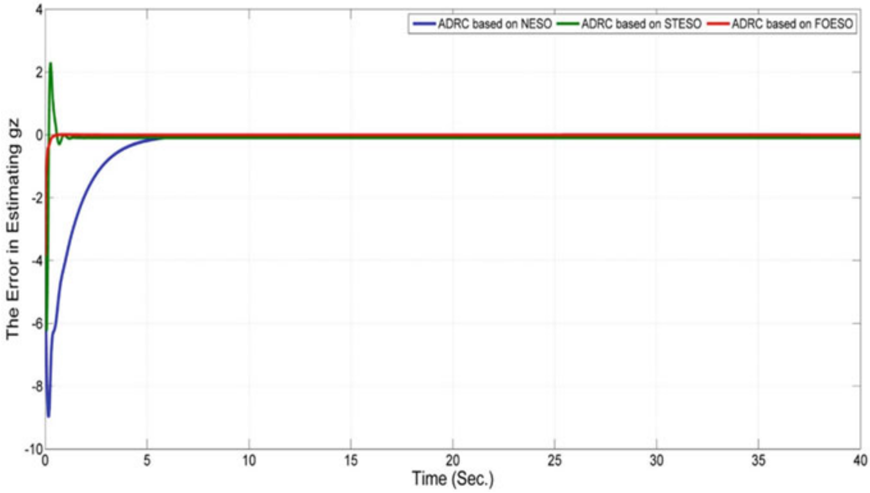


Fig. 25 The estimation errors of g_z due to FOESO, NESO and STESO

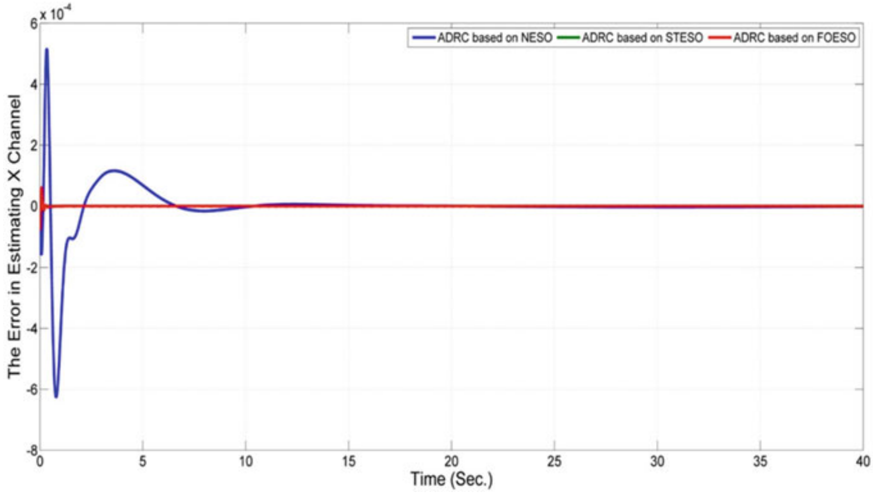


Fig. 26 The estimation errors of x signal due to FOESO, NESO and STESO

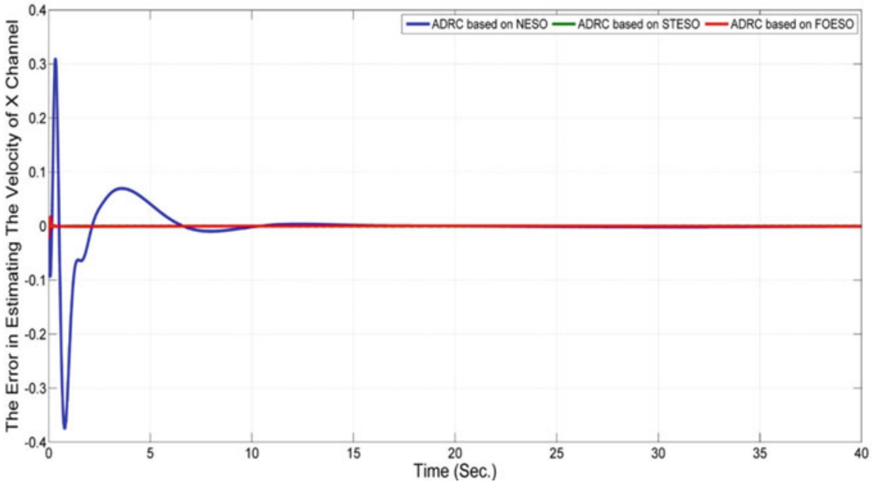


Fig. 27 The estimation errors of the rate of change of x signal due to FOESO, NESO and STESO

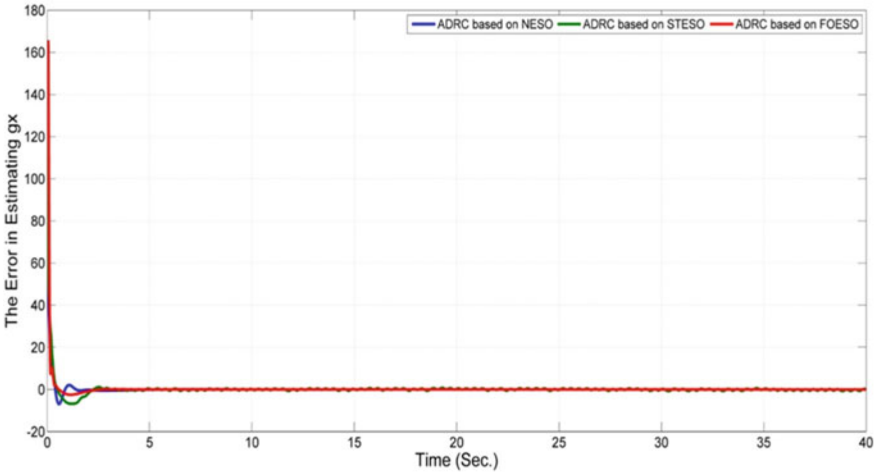


Fig. 28 The estimation errors of g_x due to FOESO, NESO and STESO

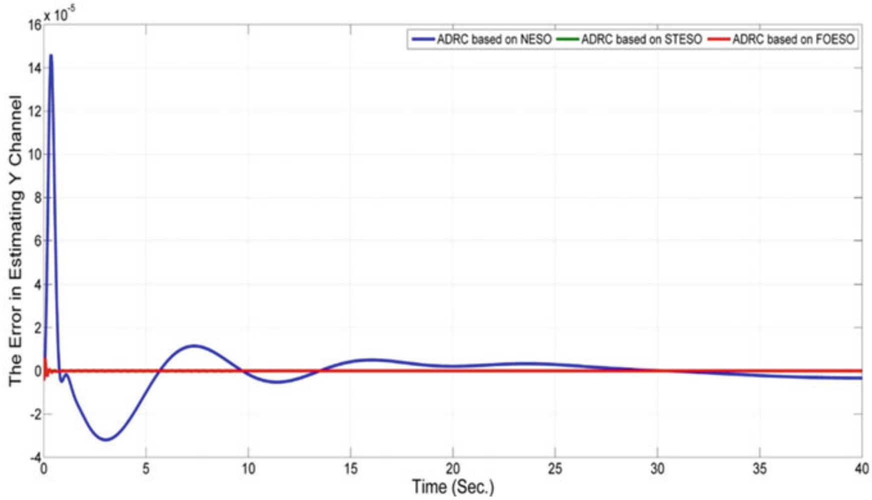


Fig. 29 The estimation errors of y signal due to FOESO, NESO and STESO

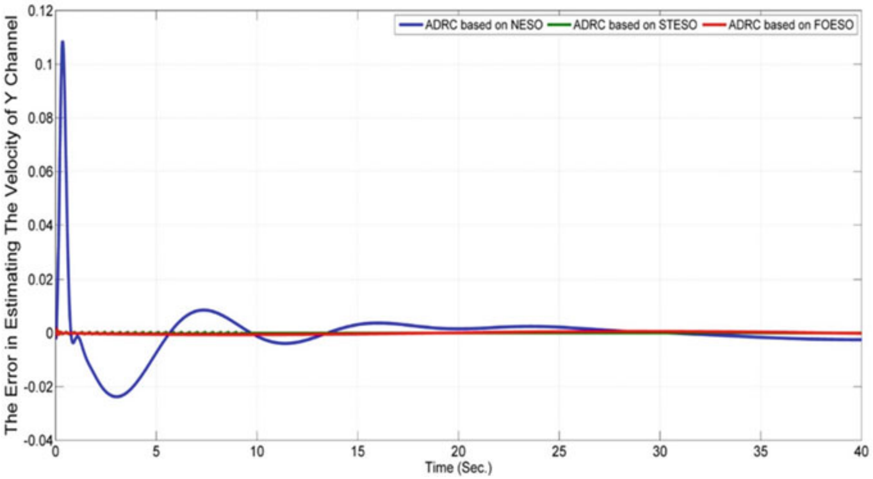


Fig. 30 The estimation errors of the rate of change of y signal due to FOESO, NESO and STESO

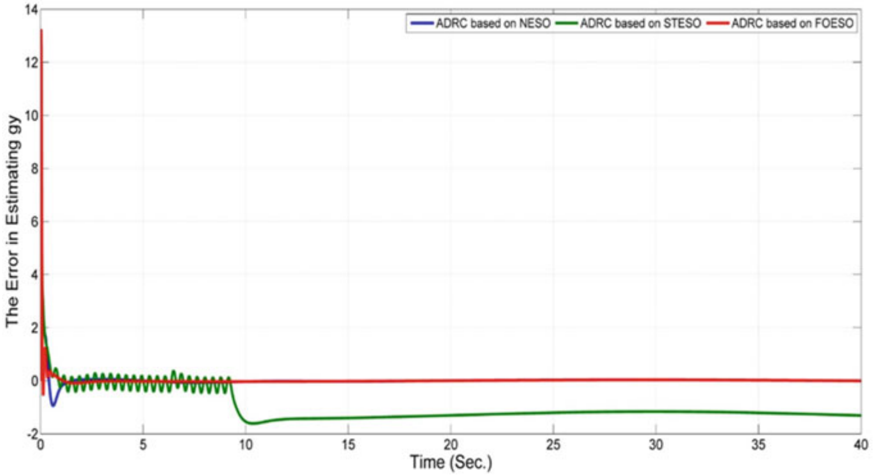
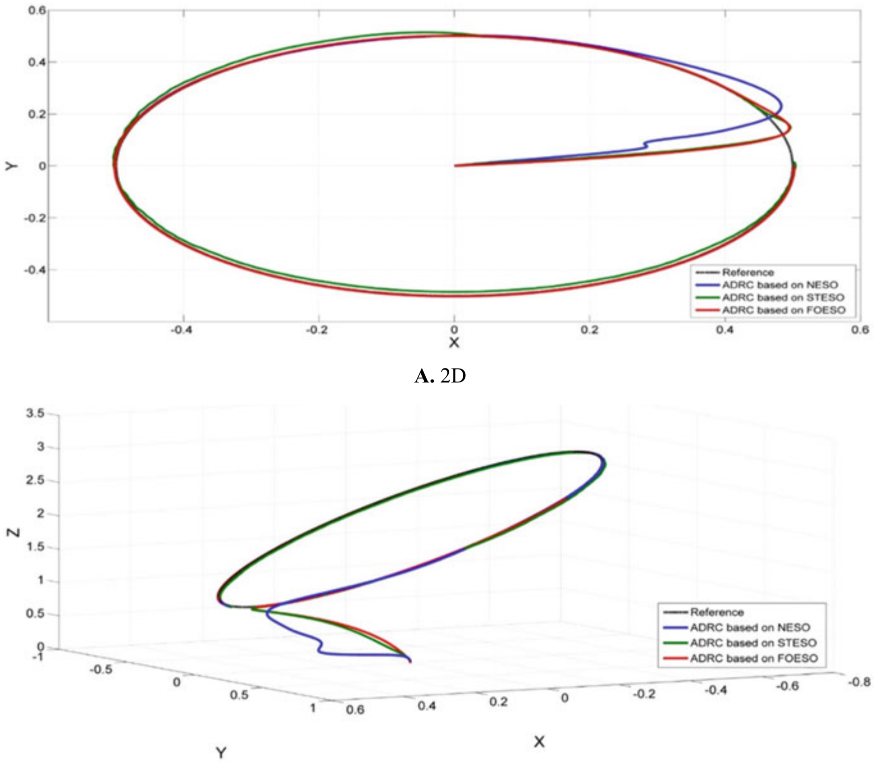


Fig. 31 The estimation errors of g_y due to FOESO, NESO and STESO



A. 2D

B. 3D.

Fig. 32 Desired trajectory and actual Tricopter trajectory due to ADRC algorithm based on FOESO, NESO and STESO

Table 3 The parameters of transient responses due to controlled system using ADRC based on FOESO, NESO and STESO under uncertainty-free and disturbance-free case

	ADRC based on FOESO			ADRC based on STESO			ADRC based on NESO		
	ISE	ITAE	RMSE	ISE	ITAE	RMSE	ISE	ITAE	RMSE
$\boldsymbol{\varphi}$	0.036	356.8	0.0302	10.1	13,547	0.502	0.486	355.2	0.110
$\boldsymbol{\theta}$	6.058	461.22	0.389	10.22	5976.9	0.505	16.39	503.7	0.640
$\boldsymbol{\psi}$	$9\,e^{-7}$	0.022	$1.5\,e^{-4}$	$2.8\,e^{-6}$	0.158	$2.6\,e^{-4}$	$7.4\,e^{-5}$	0.264	0.0014
$\boldsymbol{x}$	122.2	1261.9	1.74	123.6	1483.1	1.758	143.26	2424.27	1.893
$\boldsymbol{y}$	0.2822	1162.5	0.084	8.354	12236.2	0.457	1.025	1941.61	0.160
$\boldsymbol{z}$	310.3	1093.3	2.785	350.8	5887	2.961	859.8	2688.2	4.63

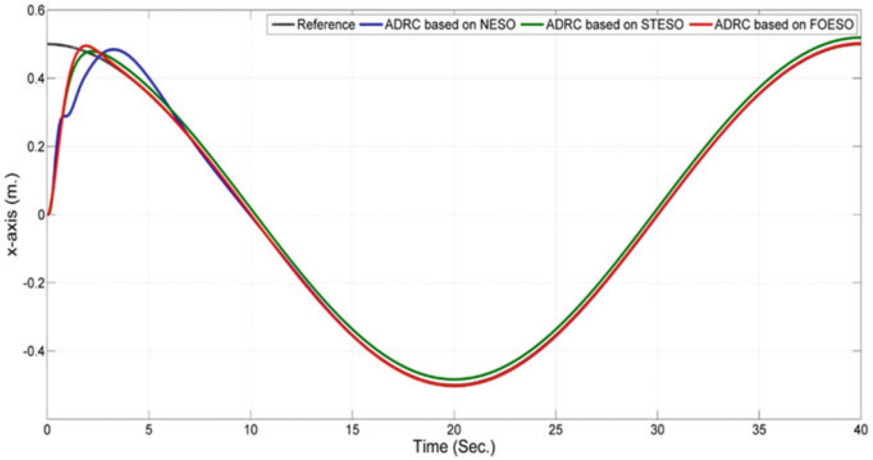


Fig. 33 The responses of x-axis due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

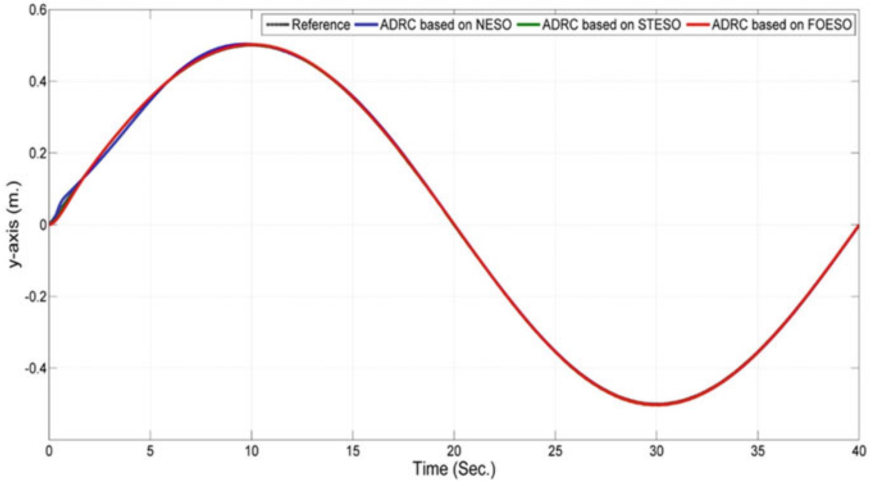


Fig. 34 The responses of y-axis due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

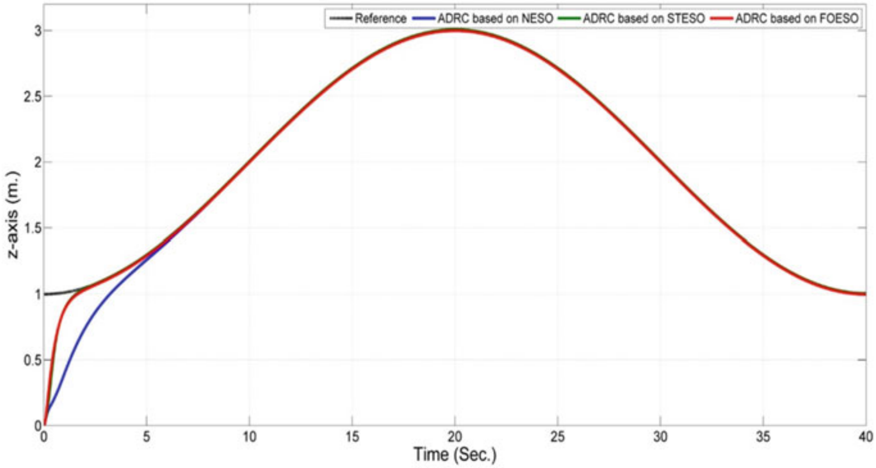


Fig. 35 The responses of z-axis due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

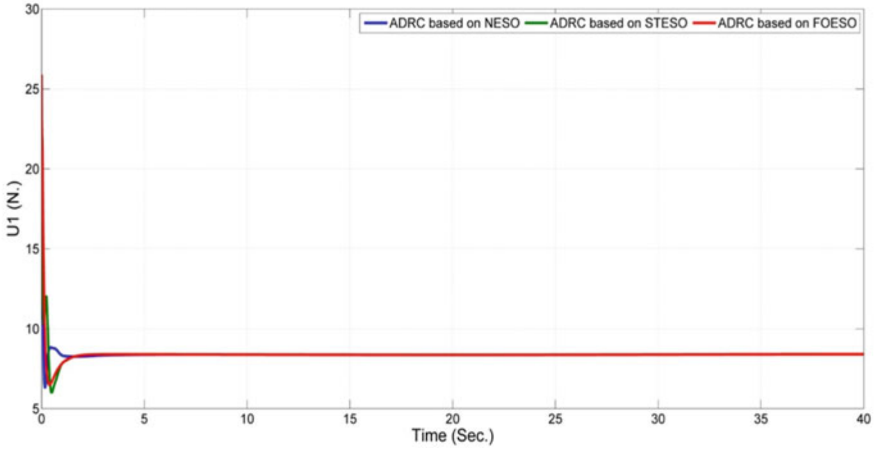


Fig. 36 Shows the control effort (U_1) due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

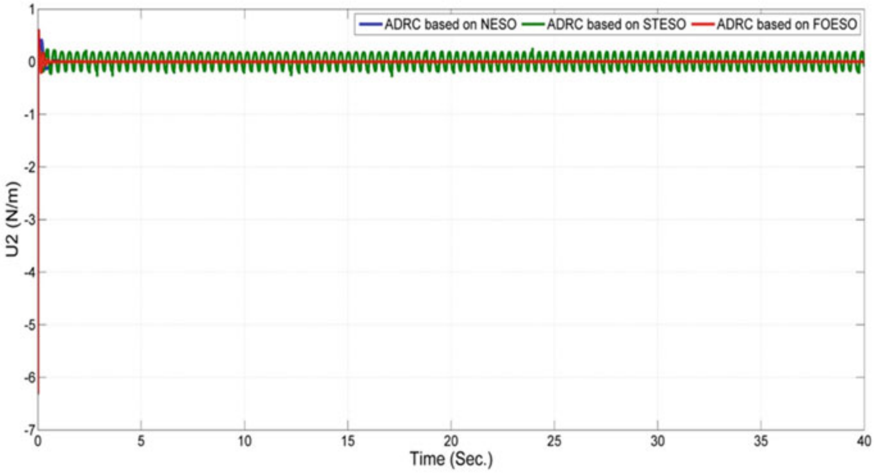


Fig. 37 Shows the control effort (U_2) due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

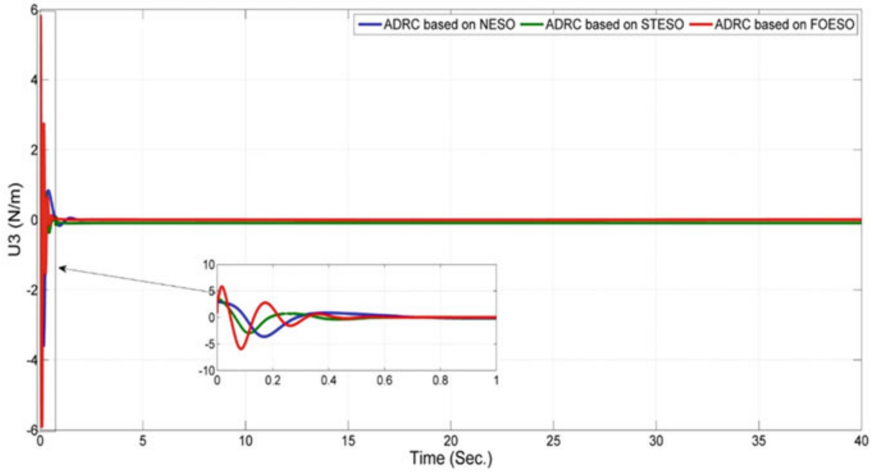


Fig. 38 Shows the control effort (U_3) due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

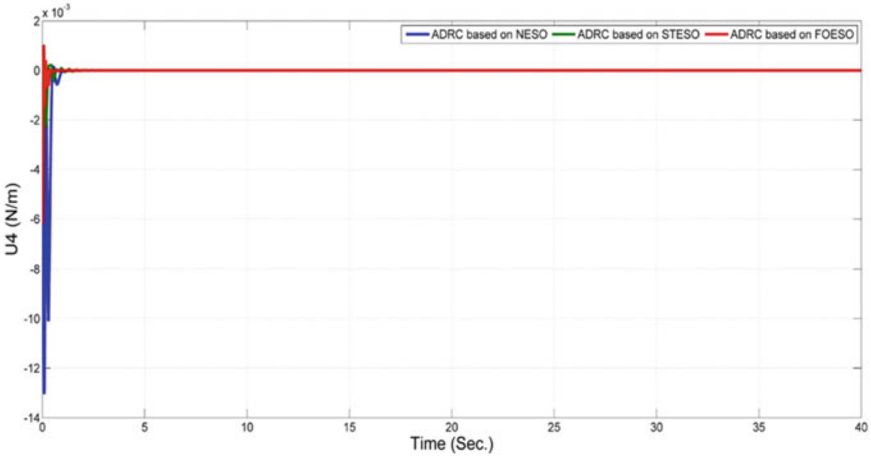


Fig. 39 Shows the control effort (U_4) due to ADRC algorithm based FOESO, NESO and STESO in the uncertainty case

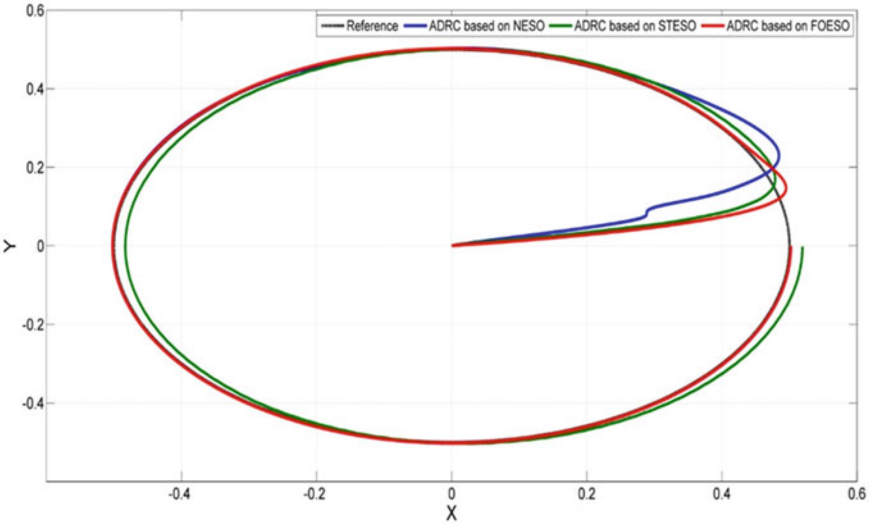


Fig. 40 Desired trajectory and actual Tricopter trajectory due to ADRC algorithm based on FOESO, NESO and STESO in the uncertainty case

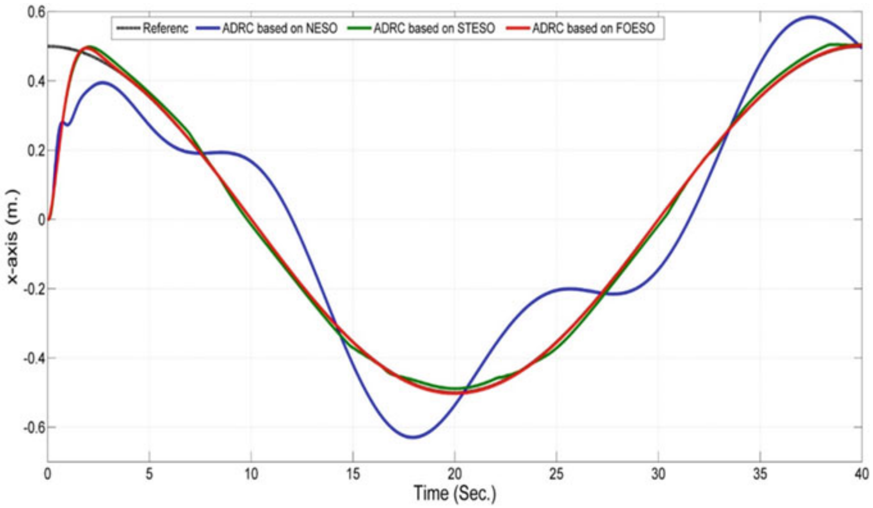


Fig. 41 The responses of x-axis due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

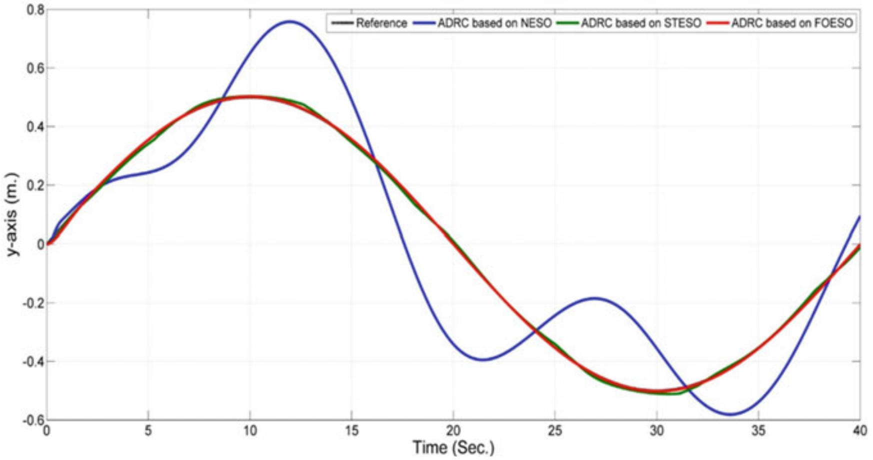


Fig. 42 The responses of y-axis due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

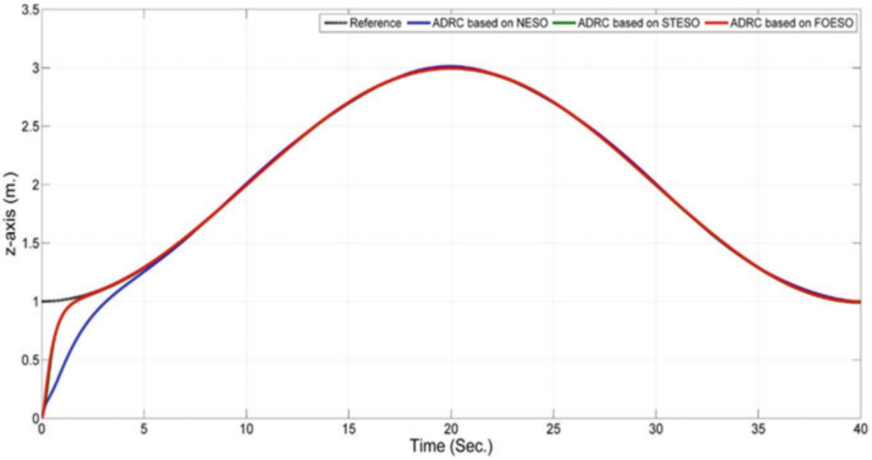


Fig. 43 The responses of z-axis due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

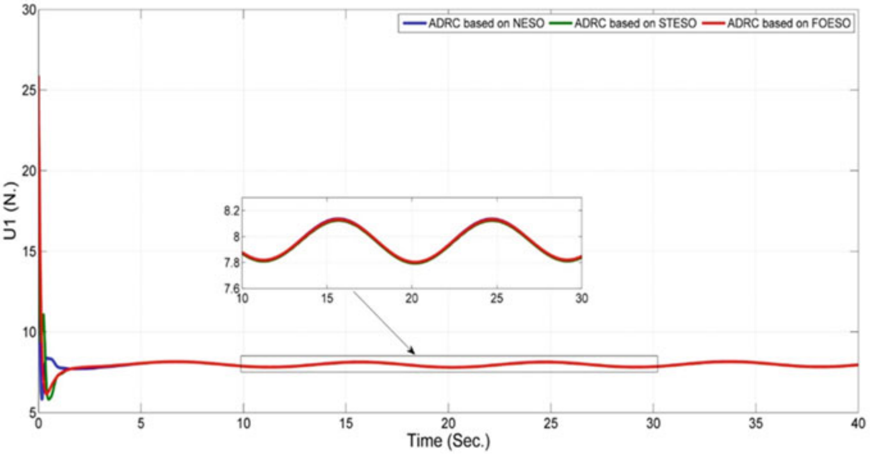


Fig. 44 Shows the control effort (U_1) due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

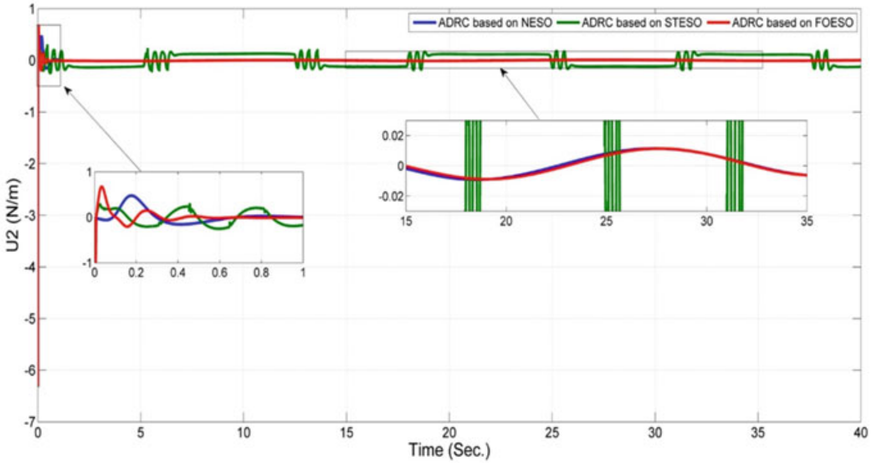


Fig. 45 Shows the control effort (U_2) due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

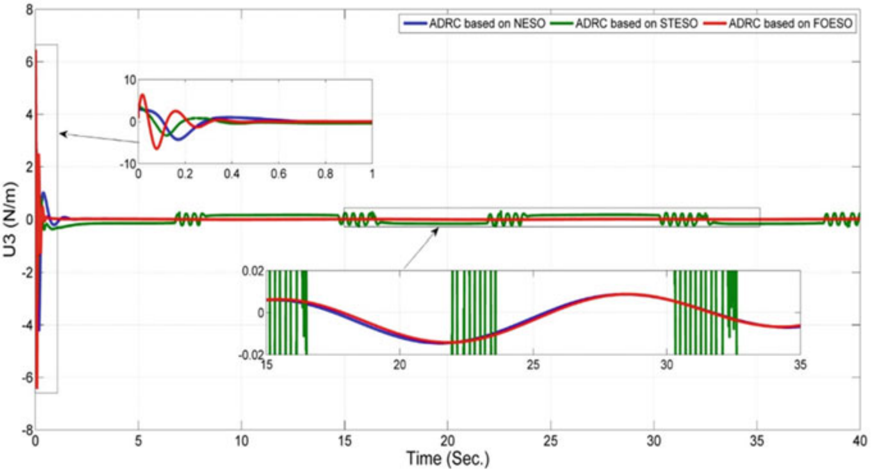


Fig. 46 Shows the control effort (U_3) due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

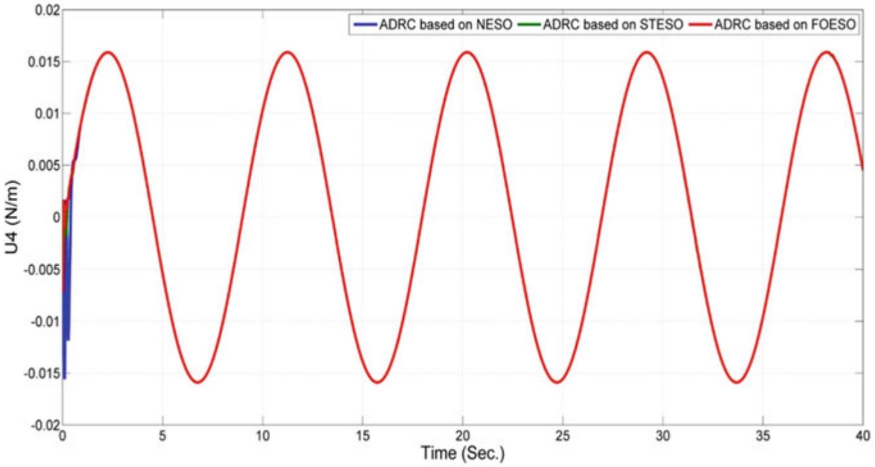


Fig. 47 Shows the control effort (U_4) due to ADRC algorithm based FOESO, NESO and STESO in the disturbance case

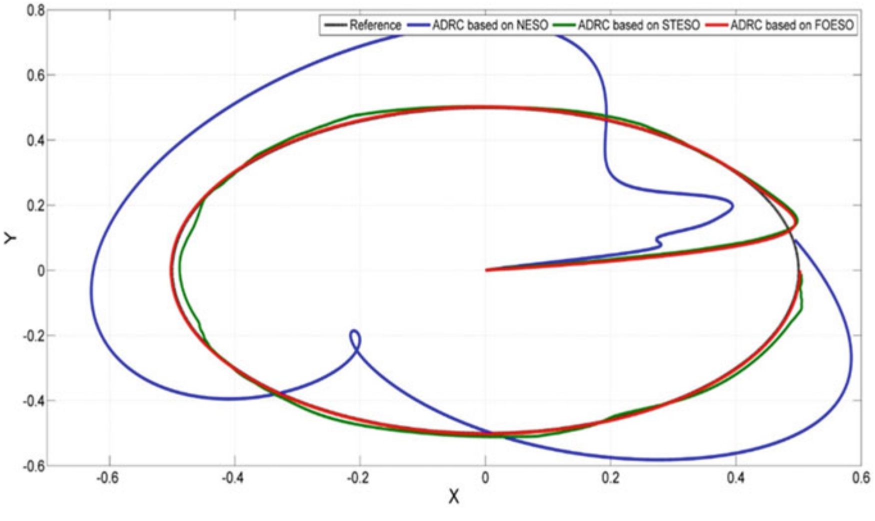


Fig. 48 Desired trajectory and actual Tricopter trajectory due to ADRC algorithm based on FOESO, NESO and STESO in the disturbance case

Acknowledgements The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for supporting this work. Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

References

1. Najm AA, Ibraheem IK, Humaidi AJ, Azar AT (2022) Output tracking and feedback stabilization for 6-DoF UAV using an enhanced active disturbance rejection control. Int J Intell Unmanned Syst 10(4):330–345. https://doi.org/10.1108/IJIUS-09-2020-0059
2. Kazim M, Azar AT, Koubaa A, Zaidi A (2021) Disturbance rejection based optimized robust adaptive controller for UAVs. IEEE Syst J 15(2):3097–3108. [ISI Q1, IF: 3.987]. https://doi.org/10.1109/JSYST.2020.3006059
3. Najm AA, Ibraheem IK, Azar AT, Humaidi AJ (2020) Genetic optimization-based consensus control of multi-agent 6-DoF UAV system. Sensors 20(12):3576. https://doi.org/10.3390/s20123576
4. Kazim M, Azar AT, Abdelkader M, Koubaa A (2022) Adaptive backstepping based linear parameter varying model predictive control multi-rotor UAVs. In: 2022 2nd international conference of smart systems and emerging technologies (SMARTTECH), May 9–11, 2022, Riyadh, Saudi Arabia, pp 166–171. https://doi.org/10.1109/SMARTTECH54121.2022.00045
5. Azar AT, Serrano FE, Koubaa A, Ibrahim HA, Kamal NA, Khamis A, Ibraheem IK, Humaidi AJ, Precup RE (2021) Robust fractional-order sliding mode control design for UAVs subjected to atmospheric disturbances. In: Koubaa A, Azar AT (eds) Unmanned aerial systems, advances in nonlinear dynamics and Chaos (ANDC). Academic Press, pp 103–128
6. Najm AA, Ibraheem IK, Azar AT, Humaidi AJ (2021) On the stabilization of 6-DOF UAV quadrotor system using modified active disturbance rejection control. In: Koubaa A, Azar AT (eds) Unmanned aerial systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 257–287
7. Azar AT, Serrano FE, Kamal NA, Koubaa A (2021) Robust kinematic control of unmanned aerial vehicles with non-holonomic constraints. In: Hassanien AE, Slowik A, Snášel V, El-Deeb H, Tolba FM (eds) Proceedings of the international conference on advanced intelligent systems and informatics 2020. AISI 2020. Advances in intelligent systems and computing, vol 1261. Springer, Cham. https://doi.org/10.1007/978-3-030-58669-0_74
8. Kazim M, Azar AT, Koubaa A, Ibrahim ZF, Zaidi A, Zhang L (2021) Event-driven programming-based path planning and navigation of UAVs around a complex urban environment. In: Koubaa A, Azar AT (eds) Unmanned aerial systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 531–565
9. Serrano FE, Azar AT, Kamal NA, Koubaa A, Abdelkader M (2022) Robust dynamic surface control of unmanned aerial vehicles with constrained inputs and unmodelled dynamics. In: 2022 2nd international conference of smart systems and emerging technologies (SMARTTECH), May 9–11, 2022, Riyadh, Saudi Arabia, pp 178–183. https://doi.org/10.1109/SMARTTECH54121.2022.00047
10. Azar AT, Serrano FE, Kamal NA, Koubaa A (2021) Leader-follower control of unmanned aerial vehicles with state dependent switching. In: Hassanien AE, Slowik A, Snášel V, El-Deeb H, Tolba FM (eds) Proceedings of the international conference on advanced intelligent systems and informatics 2020. AISI 2020. Advances in intelligent systems and computing, vol 1261. Springer, Cham. https://doi.org/10.1007/978-3-030-58669-0_76
11. Sababha BH, Zu'bi HMA, Rawashdeh OA (2015) A rotor-Tilt-free Tricopter UAV: design, modelling, and stability control. Int J Mech Autom 5(2–3):107–113. https://doi.org/10.1504/IJMA.2015.075956
12. Zou JT, Su KL, Tso H (2012) The modeling and implementation of tri-rotor flying robot. Artif Life Robot 17(1):86–91. https://doi.org/10.1007/s10015-012-0028-2
13. Yoo D-W et al (2010) dynamic modeling and stabilization techniques for tri-rotor unmanned aerial vehicles. Int J Aeronaut Space Sci 11(3):167–174. https://doi.org/10.5139/ijass.2010.11.3.167
14. Źrebiec J (2016) Modelling of unmanned aerial vehicle–tricopter. Automatyka/Automatics 20(1):7. https://doi.org/10.7494/automat.2016.20.1.7
15. Han J (2009) From PID to active disturbance rejection control. IEEE Trans Ind Electron 56(3):900–906. https://doi.org/10.1109/TIE.2008.2011621

16. Abdul-Kareem AI, Hasan AF, Al-Qassar AA, Humaidi AJ, Hassan RF, Ibraheem IK, Azar AT (2022) Rejection of wing-rock motion in delta wing aircrafts based on optimal LADRC schemes with butterfly optimization algorithm. J Eng Sci Technol 17(4):2476–2495
17. Toumi I, Meghni B, Hachana O, Azar AT, Boulmaiz A, Humaidi AJ, Ibraheem IK, Kamal NA, Zhu Q, Fusco G, Bahgaat NK (2022) Robust variable-step perturb-and-observe sliding mode controller for grid-connected wind-energy-conversion systems. Entropy 24:731. https://doi.org/10.3390/e24050731
18. Daraz A, Malik SA, Azar AT, Aslam S, Alkhalifah T, Alturise F (2022) Optimized fractional order integral-tilt derivative controller for frequency regulation of interconnected diverse renewable energy resources. IEEE Access 10:43514–43527. https://doi.org/10.1109/ACCESS.2022.3167811
19. Ali T, Malik SA, Hameed IA, Daraz A, Mujlid H, Azar AT (2022) Load frequency control and automatic voltage regulation in a multi-area interconnected power system using nature-inspired computation-based control methodology. Sustainability 14(19):12162. https://doi.org/10.3390/su141912162
20. Abdul-Adheem WR, Ibraheem IK, Azar AT, Humaidi AJ (2021) Design and analysis of a novel generalized continuous tracking differentiator. Ain Shams Eng J 13(4):101656
21. Sain C, Banerjee A, Biswas PK, Azar AT, Babu TS (2022) Design and optimisation of a fuzzy-PI controlled modified inverter-based PMSM drive employed in a light weight electric vehicle. Int J Autom Control 16(3/4):459–488
22. Ali MO, Abou-Loukh SJ, Al-Dujaili AQ, Alkhayyat A, Abdulkareem AI, Ibraheem IK, Humaidi AJ, Al-Qassar AA, Azar AT (2022) Radial basis function neural networks-based short term electric power load forecasting for super high voltage power grid. J Eng Sci Technol 17(1):0361–0378
23. Ajel AR, Humaidi AJ, Ibraheem IK, Azar AT (2021) Robust model reference adaptive control for tail-sitter VTOL aircraft. Actuators 10:162. https://doi.org/10.3390/act10070162
24. Drhorhi I, El Fadili A, Berrahal C, Lajouad R, El Magri A, Giri F, Azar AT, Vaidyanathan S (2021) Adaptive backstepping controller for DFIG-based wind energy conversion system. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 235–260
25. Daraz A, Malik SA, Waseem A, Azar AT, Haq IU, Ullah Z, Aslam S (2021) Automatic generation control of multi-source interconnected power system using FOI-TD controller. Energies 14(18):5867. https://doi.org/10.3390/en14185867
26. Pilla R, Gorripotu TS, Azar AT (2021) Tuning of extended Kalman filter using grey wolf optimisation for speed control of permanent magnet synchronous motor drive. Int J Autom Control 15(4–5):563–584
27. Humaidi AJ, Najem HT, Al-Dujaili AQ, Pereira DA, Ibraheem IK, Azar AT (2021) Social spider optimization algorithm for tuning parameters in PD-like Interval Type-2 Fuzzy Logic Controller applied to a parallel robot. Meas Control 54(3–4):303–323. https://doi.org/10.1177/0020294021997483
28. Abdul-Adheem W, Ibraheem IK, Humaidi AJ, Azar AT (2021) Model-free active input-output feedback linearization of a single-link flexible joint manipulator: an improved active disturbance rejection control approach. Meas Control 54(5–6):856–871. https://doi.org/10.1177/0020294020917171
29. Pilla R, Gorripotu TS, Azar AT (2021) Design and analysis of search group algorithm based PD-PID controller plus redox flow battery for automatic generation control problem. Int J Comput Appl Technol 66(1):19–35
30. Al-Qassar AA, Al-Dujaili AQ, Hasan AF, Humaidi AJ, Ibraheem IK, Azar AT (2021) Stabilization of single-axis propeller-powered system for aircraft applications based on optimal adaptive control design. J Eng Sci Technol (JESTEC) 16(3):1851–1869
31. Abdul-Adheem WR, Ibraheem IK, Azar AT, Humaidi AJ (2020) Improved active disturbance rejection-based decentralized control for MIMO nonlinear systems: comparison with the decoupled control scheme. Appl Sci 10(7):2515. https://doi.org/10.3390/app10072515

32. Liu L, Ma D, Azar AT, Zhu Q (2020) Neural computing enhanced parameter estimation for multi-input and multi-output total nonlinear dynamic models. Entropy 22(5):510. https://doi.org/10.3390/e22050510
33. Kammogne AST, Kountchou MN, Kengne R, Azar AT, Fotsin HB, Ouagni STM (2020) Polynomial robust observer implementation based-passive synchronization of nonlinear fractional-order systems with structural disturbances. Front Inf Technol Electron Eng 21(9):1369–1386
34. Djeddi A, Dib D, Azar AT, Abdelmalek S (2019) Fractional order unknown inputs fuzzy observer for Takagi-Sugeno systems with unmeasurable premise variables. Mathematics 7(10):984. https://doi.org/10.3390/math7100984
35. Pilla R, Azar AT, Gorripotu TS (2019) Impact of flexible ac transmission system devices on automatic generation control with a metaheuristic based fuzzy PID controller. Energies 12(21):4193. https://doi.org/10.3390/en12214193
36. Ammar HH, Azar AT, Shalaby R, Mahmoud MI (2019) Metaheuristic optimization of fractional order incremental conductance (FO-INC) maximum power point tracking (MPPT). Complexity 2019:1–13. Article ID 7687891. https://doi.org/10.1155/2019/7687891
37. Radwan AG, Emira AA, Abdelaty A, Azar AT (2018) Modeling and analysis of fractional order DC-DC converter. ISA Trans 82(2018):184–1991
38. Meghni B, Dib D, Azar AT, Saadoun A (2018) Effective supervisory controller to extend optimal energy management in hybrid wind turbine under energy and reliability constraints. Int J Dyn Control 6(1):369–383. https://doi.org/10.1007/s40435-016-0296-0. Springer
39. Meghni B, Dib D, Azar AT (2017) A second-order sliding mode and fuzzy logic control to Optimal Energy Management in PMSG Wind Turbine with Battery Storage. Neural Comput Appl 28(6):1417–1434. https://doi.org/10.1007/s00521-015-2161-z
40. Fekik A, Denoun H, Azar AT, Kamal NA, Zaouia M, Benyahia N, Hamida ML, Benamrouche N, Vaidyanathan S (2021) Direct power control of three-phase PWM-rectifier with backstepping control. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 215–234
41. Humaidi AJ, Sadiq ME, Abdulkareem AI, Ibraheem IK, Azar AT (2022) Adaptive backstepping sliding mode control design for vibration suppression of earth-quaked building supported by magneto-rheological damper. J Low Freq Noise Vib Act Control 41(2):768–783. https://doi.org/10.1177/14613484211064659
42. Vaidyanathan S, Jafari S, Pham VT, Azar AT, Alsaadi FE (2018) A 4-D chaotic hyperjerk system with a hidden attractor, adaptive backstepping control and circuit design. Arch Control Sci 28(2):239–254
43. Pham VT, Vaidyanathan S, Azar AT, Duy VH (2021) A new chaotic system without linear term, its backstepping control, and circuit design. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 33–52
44. Vaidyanathan S, Pham VT, Azar AT (2021) A new chaotic jerk system with egg-shaped strange attractor, its dynamical analysis, backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 53–71
45. Sambas A, Vaidyanathan S, Zhang S, Mohamed MA, Zeng Y, Azar AT (2021) A new 4-D chaotic hyperjerk system with coexisting attractors, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 73–94
46. Sambas A, Vaidyanathan S, Zhang S, Mohamed MA, Zeng Y, Azar AT (2021) A new 3-D chaotic jerk system with a saddle-focus rest point at the origin, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 95–114
47. Vaidyanathan S, Sambas A, Azar AT, Rana KPS, Kumar V (2021) A new 5-D hyperchaotic four-wing system with multistability and hidden attractor, its backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 115–138

48. Vaidyanathan S, Sambas A, Azar AT, Rana KPS, Kumar V (2021) A new 4-D hyperchaotic temperature variations system with multistability and strange attractor, bifurcation analysis, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 139–164
49. Alimi M, Rhif A, Rebai A, Vaidyanathan S, Azar AT (2021) Optimal adaptive backstepping control for chaos synchronization of nonlinear dynamical systems. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 291–345
50. Kumar V, Rana KPS, Azar AT, Vaidyanathan S (2021) Backstepping controller for nonlinear active suspension system. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 347–374
51. Bansal N, Bisht A, Paluri S, Kumar V, Rana KPS, Azar AT, Vaidyanathan S (2021) Single-link flexible joint manipulator control using backstepping technique. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 375–406
52. Singh S, Mathpal S, Azar AT, Vaidyanathan S, Kamal NA (2021) Multi-switching synchronization of nonlinear hyperchaotic systems via backstepping control. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 425–447
53. Vaidyanathan S, Sambas A, Azar AT (2021) A 5-D hyperchaotic dynamo system with multistability, its dynamical analysis, active backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 449–471
54. Sambas A, Vaidyanathan S, Azar AT, Hidayat Y, Gundara G, Mohamad MA (2021) A novel chaotic system with a closed curve of four quarter-circles of equilibrium points: dynamics, active backstepping control, and electronic circuit implementation. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC). Academic Press, pp 485–507
55. Ali ZA et al (2016) Attitude and altitude control of trirotor UAV by using adaptive hybrid controller. J Control Sci Eng. https://doi.org/10.1155/2016/6459891
56. Prach A, Kayacan E (2018) An MPC-based position controller for a tilt-rotor tricopter VTOL UAV. Optim Control Appl Methods 39(1):343–356. https://doi.org/10.1002/oca.2350
57. Nam KJ, Joung J, Har D (2020) Tri-copter UAV with individually tilted main wings for flight maneuvers. IEEE Access, 8:46753–46772. https://doi.org/10.1109/ACCESS.2020.2978578
58. Song Z et al (2017) Modeling and maneuvering control for tricopter based on the backstepping method. In: CGNCC 2016-2016 IEEE Chinese guidance, navigation and control conference, (rotor 1), pp 889–894. https://doi.org/10.1109/CGNCC.2016.7828903
59. Tran HK et al (2019) Adaptive fuzzy control method for a single tilt tricopter. IEEE Access 7:161741–161747. https://doi.org/10.1109/ACCESS.2019.2950895
60. Yoon S et al. (2013) Design and flight test of small tri-rotor unmanned vehicle with LQR based onboard attitude control system. INter
61. Zhang Y et al (2018) A novel control scheme for quadrotor UAV based upon active disturbance rejection control. Aerosp Sci Technol 79:601–609. https://doi.org/10.1016/j.ast.2018.06.017
62. Tran HK, Chiou JS, Peng ST (2016) Design Genetic Algorithm Optimization education software based fuzzy controller for a tricopter fly path planning. Eurasia J Math Sci Technol Educ 12(5):1303–1312. https://doi.org/10.12973/eurasia.2016.1514a
63. Mohamed MK, Lanzon A (2012) Design and control of novel tri-rotor UAV. In: Proceedings of the 2012 UKACC international conference on control, CONTROL 2012, September, pp 304–309. https://doi.org/10.1109/CONTROL.2012.6334647.
64. Humaidi AJ, Badr HM (2018) Linear and nonlinear active disturbance rejection controllers for single-link flexible joint robot manipulator based on PSO tuner. J Eng Sci Technol Rev 11(3):133–138. https://doi.org/10.25103/jestr.113.18

65. Humaidi AJ, Badr HM, Ajil AR (2018) Design of active disturbance rejection control for single-link flexible joint robot manipulator. In: 2018 22nd international conference on system theory, control and computing, ICSTCC 2018-Proceedings, pp 452–457. https://doi.org/10.1109/ICSTCC.2018.8540652
66. Yang Y et al (2014) Mechanical analysis and modeling for tricopter. Appl Mech Mater 455:304–309. https://doi.org/10.4028/www.scientific.net/AMM.455.304
67. Huang Y, Xue W (2014) Active disturbance rejection control: methodology and theoretical analysis. ISA Trans 53(4):963–976. https://doi.org/10.1016/j.isatra.2014.03.003
68. Humaidi AJ, Badr HM, Hameed AH (2018) PSO-based active disturbance rejection control for position control of magnetic levitation system. In: 2018 5th international conference on control, decision and information technologies, CoDIT 2018, pp 922–928. https://doi.org/10.1109/CoDIT.2018.8394955
69. Azar AT, Radwan AG and Vaidyanathan S (2018) Fractional order systems: optimization, control, circuit realizations and applications. Elsevier. ISBN: 9780128161524
70. Azar AT, Radwan AG, Vaidyanathan S (2018) Mathematical techniques of fractional order systems. Elsevier. ISBN: 9780128135921
71. Abed AM, Rashid ZN, Abedi F, Zeebaree SRM, Sahib MA, Mohamad Jawad AJ, Redha Ibraheem GA, Maher RA, Abdulkareem AI, Ibraheem IK, Azar AT, Al-khaykan A (2022) Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization. Meas Control.https://doi.org/10.1177/00202940221092134
72. Ghoudelbourk S, Azar AT, Dib D, Rechach A (2022) Fractional order control of switched reluctance motor. Int J Adv Intell Paradig 21(3/4):247–266. https://doi.org/10.1504/IJAIP.2018.10024488
73. Shalaby R, Ammar HH, Azar AT, Mahmoud MI (2021) Optimal Fractional-Order Fuzzy-MPPT for solar water pumping system. J Intell Fuzzy Syst 40(1):1175–1190. https://doi.org/10.3233/JIFS-201538
74. Ghoudelbourk S, Azar AT, Dib D (2021) Three-level (NPC) shunt active power filter based on fuzzy logic and fractional-order PI controller. Int J Autom Control 15(2):149–169
75. Ibraheem GAR, Azar AT, Ibraheem IK, Humaidi AJ (2020) A novel design of a neural network based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization, complexity, vol 2020, pp 1–18. Article ID: 3067024. https://doi.org/10.1155/2020/3067024
76. Khennaoui AA, Ouannas A, Boulaaras S, Pham VT, Azar AT (2020) A fractional map with hidden attractors: chaos and control. Eur Phys J Spec Top 229:1083–1093
77. Kammogne AST, Kountchou MN, Kengne R, Azar AT, Fotsin HB, Ouagni STM (2020) Polynomial robust observer implementation based-passive synchronization of nonlinear fractional-order systems with structural disturbances. Front Inf Technol Electron Eng 21(9):1369–1386
78. Alain KST, Azar AT, Kengne R, Bertrand FH (2020) Stability analysis and robust synchronisation of fractional-order modified Colpitts oscillators. Int J Autom Control (IJAAC) 14(1):52–79
79. Ouannas A, Azar AT, Ziar T (2020) Fractional inverse full state hybrid projective synchronisation. Int J Adv Intell Paradig 17(3–4):270–298
80. Tolba MF, AbdelAty AM, Soliman NS, Said LA, Madian AH, Azar AT, Radwan AG (2017) FPGA implementation of two fractional order chaotic systems. Int J Electron Commun 28(2017):162–172
81. Ouannas A, Azar AT, Vaidyanathan S (2017) A robust method for new fractional hybrid chaos synchronization. Math Methods Appl Sci 40(5):1804–1812. https://doi.org/10.1002/mma.4099
82. Ouannas A, Azar AT, Vaidyanathan S (2017) A new fractional hybrid chaos synchronization. Int J Model Identif Control 27(4):314–322
83. Ghoudelbourk S, Dib D, Omeiri A, Azar AT (2016) MPPT Control in wind energy conversion systems and the application of fractional control (PI^{α}) in pitch wind turbine. Int J Model Identif Control (IJMIC) 26(2):140–151

84. Rana KPS, Kumar V, Sehgal N, George S, Azar AT (2021) Efficient maximum power point tracking in fuel cell using the fractional-order PID controller. In: Advances in nonlinear dynamics and chaos (ANDC), renewable energy systems. Academic Press, pp 111–132. https://doi.org/10.1016/B978-0-12-820004-9.00017-6

85. Fekik A, Azar AT, Kamal NA, Denoun H, Almustafa KM, Hamida ML, Zaouia M (2021) Fractional-order control of a fuel cell-boost converter system. In: Hassanien A, Bhatnagar R, Darwish A (eds) Advanced machine learning technologies and applications. AMLTA 2020. Advances in intelligent systems and computing, vol 1141. Springer, Singapore, pp 713–724

86. Ouannas A, Grassi G, Azar AT, Khennaoui AA (2021) Synchronization control in fractional discrete-time systems with chaotic hidden attractors. In: Hassanien A, Bhatnagar R, Darwish A (eds) Advanced machine learning technologies and applications. AMLTA 2020. Advances in intelligent systems and computing, vol 1141, Springer, Singapore, pp 661–669

87. Mittal S, Azar AT, Kamal NA (2021) Nonlinear fractional order system synchronization via combination-combination multi-switching. In: Hassanien AE, Slowik A, Snášel V, El-Deeb H, Tolba FM (eds) Proceedings of the international conference on advanced intelligent systems and informatics 2020. AISI 2020. Advances in intelligent systems and computing, vol 1261. Springer, Cham. https://doi.org/10.1007/978-3-030-58669-0_75

88. Ammar H, Ibrahim M, Azar AT, Shalaby R (2020) Gray wolf optimization of fractional order control of 3-omni wheels mobile robot: experimental study. In: 2020 16th international computer engineering conference (ICENCO), 29–30 Dec 2020, Cairo, Egypt, pp 147–152. https://doi.org/10.1109/ICENCO49778.2020.9357384

89. Singh S, Azar AT (2020) Multi-switching combination synchronization of fractional order chaotic systems. In: Hassanien AE, Azar A, Gaber T, Oliva D, Tolba F (eds) Proceedings of the international conference on artificial intelligence and computer vision (AICV2020). AICV 2020. Advances in intelligent systems and computing, vol 1153, Springer, Cham, pp 655–664

90. Khennaoui AA, Ouannas A, Grassi G, Azar AT (2020) Dynamic analysis of a fractional map with hidden attractor. In: Hassanien AE, Azar A, Gaber T, Oliva D, Tolba F (eds) Proceedings of the international conference on artificial intelligence and computer vision (AICV2020). AICV 2020. Advances in intelligent systems and computing, vol 1153, Springer, Cham, pp 731–739

91. Ammar HH, Azar AT (2020) Robust path tracking of mobile robot using fractional order PID controller. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in intelligent systems and computing, vol 921. Springer, pp 370–381

92. Khan A, Singh S, Azar AT (2020) Synchronization between a novel integer-order hyperchaotic system and a fractional-order hyperchaotic system using tracking control. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in intelligent systems and computing, vol 921. Springer, Cham, pp 382–391

93. Khan A, Singh S, Azar AT (2020) Combination-combination anti-synchronization of four fractional order identical hyperchaotic systems. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in intelligent systems and computing, vol 921. Springer, Cham, pp 406–414

94. Ouannas A, Grassi G, Azar AT (2020) A new generalized synchronization scheme to control fractional chaotic systems with non-identical dimensions and different orders. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in intelligent systems and computing, vol 921. Springer, Cham, pp 415–424

95. Ouannas A, Grassi G, Azar AT (2020) Fractional-order control scheme for Q-S chaos synchronization. In: The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in intelligent systems and computing, vol 921. Springer, Cham, pp 434–441

96. Ouannas A, Grassi G, Azar AT, Khennaouia AA, Pham VT (2020) Chaotic control in fractional-order discrete-time systems. AISI 2019. In: The international conference on advanced intelligent systems and informatics AISI 2019. Advances in intelligent systems and computing, vol 1058, Springer, pp 207–217
97. Ouannas A, Grassi G, Azar AT, Khennaouia AA, Pham VT (2020) Synchronization of fractional-order discrete-time chaotic systems. In: The international conference on advanced intelligent systems and informatics AISI 2019. Advances in intelligent systems and computing, vol 1058. Springer, pp 218–228
98. Fekik A, Denoun H, Azar AT, Koubaa A, Kamal NA, Zaouia M, Hamida ML, Yassa N (2020) Adapted fuzzy fractional order proportional-integral controller for DC motor. In: The first international conference of smart systems and emerging technologies (SMART TECH 2020), November 3–5, 2020, Riyadh, Saudi Arabia, pp 1–6. https://doi.org/10.1109/SMART-TECH49988.2020.00019
99. Gorripotu TS, Samalla H, Jagan Mohana Rao C, Azar AT, Pelusi D (2019) TLBO algorithm optimized fractional-order PID controller for AGC of interconnected power system. In: Nayak J, Abraham A, Krishna B, Chandra Sekhar G, Das A (eds) Soft computing in data analytics. Advances in intelligent systems and computing, vol 758. Springer, Singapore
100. Ouannas A, Grassi G, Azar AT, Singh S (2019) New control schemes for fractional chaos synchronization. In: Proceedings of the international conference on advanced intelligent systems and informatics 2018. AISI 2018. Advances in intelligent systems and computing, vol 845. Springer, Cham
101. Kumar J, Azar AT, Kumar V, Rana KPS (2018) Design of fractional order fuzzy sliding mode controller for nonlinear complex systems. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series, Elsevier, pp 249–282
102. Singh S, Azar AT, Vaidyanathan S, Ouannas A, Bhat MA (2018). Multiswitching synchronization of commensurate fractional order hyperchaotic systems via active control. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series, Elsevier, pp 319–345
103. AbdelAty AM, Azar AT, Vaidyanathan S, Ouannas A, Radwan AG (2018) Applications of continuous-time fractional order chaotic systems. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series. Elsevier, pp 409–449
104. Pham VT, Gokul PM, Kapitaniak T, Volos C, Azar AT (2018) Dynamics, synchronization and fractional order form of a chaotic system with infinite equilibria. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series, Elsevier, pp 475–502
105. Shukla MK, Sharma BB, Azar AT (2018) Control and synchronization of a fractional order hyperchaotic system via backstepping and active backstepping approach. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series. Elsevier, pp 597–624
106. Khettab K, Bensafia Y, Bourouba B, Azar AT (2018) Enhanced fractional order indirect fuzzy adaptive synchronization of uncertain fractional chaotic systems based on the variable structure control: robust H? Design approach. In: Mathematical techniques of fractional order systems, advances in nonlinear dynamics and chaos (ANDC) series. Elsevier, pp 559–595
107. Alain KST, Romanic K, Azar AT, Vaidyanathan S, Bertrand FH, Adele NM (2018) Dynamics analysis and synchronization in relay coupled fractional order Colpitts oscillators. In: Advances in system dynamics and control. IGI-Global, USA, pp 317–356. https://doi.org/10.4018/978-1-5225-4077-9.ch011
108. Pham VT, Vaidyanathan S, Volos CK, Azar AT, Hoang TM, Yem VV (2017) A three-dimensional no-equilibrium chaotic system: analysis, synchronization and its fractional order form. Studies in computational intelligence, vol 688. Springer, Germany, pp 449–470
109. Meghni B, Dib D, Azar AT, Ghoudelbourk S, Saadoun A (2017) Robust adaptive supervisory fractional order controller for optimal energy management in wind turbine with battery storage. Studies in computational intelligence, vol 688. Springer, Germany, pp 165–202

110. Lamamra K, Azar AT, Ben Salah C (2017) Chaotic system modelling using a neural network with optimized structure. Studies in computational intelligence, vol 688, Springer, Germany, pp 833–856
111. Ouannas A, Azar AT, Ziar T and Vaidyanathan S (2017) On new fractional inverse matrix projective synchronization schemes. Studies in computational intelligence, vol 688. Springer, Germany, pp 497–524
112. Ouannas A, Azar AT, Ziar T, Vaidyanathan S (2017) Fractional inverse generalized chaos synchronization between different dimensional systems. Studies in computational intelligence, vol 688. Springer, Germany, pp 525–551
113. Ouannas A, Azar AT, Ziar T, Vaidyanathan S (2017) A new method to synchronize fractional chaotic systems with different dimensions. Studies in computational intelligence, vol 688, Springer, Germany, pp 581–611
114. Ouannas A, Azar AT, Ziar T, Radwan AG (2017) A study on coexistence of different types of synchronization between different dimensional fractional chaotic systems. Studies in computational intelligence, vol 688, Springer, Germany, pp 637–669
115. Ouannas A, Azar AT, Ziar T, Radwan AG (2017) Generalized synchronization of different dimensional integer-order and fractional order chaotic systems. Studies in computational intelligence, vol 688. Springer, Germany, pp 671–697
116. Pham VT, Volos CK, Vaidyanathan S, Azar AT (2017) Dynamics, synchronization and fractional order form of a chaotic system without equilibrium. In: Volos CK (ed) Nonlinear systems: design, applications and analysis. Nova Science Publishers
117. Soliman NS, Said LA, Azar AT, Madian AH, Radwan AG, Ouannas A (2017). Fractional controllable multi-scroll V-shape attractor with parameters effect. In: The 6th international conference on modern circuits and systems technologies (MOCAST), 4–6 May 2017, Thessaloniki Greece
118. Boulkroune A, Bouzeriba A., Bouden T, Azar AT (2016) Fuzzy adaptive synchronization of uncertain fractional-order chaotic systems. Studies in fuzziness and soft computing, vol 337. Springer, Germany, pp 681–697
119. Ouannas A, Azar AT, Radwan AG (2016) On inverse problem of generalized synchronization between different dimensional integer-order and fractional-order chaotic systems. In: The 28th international conference on microelectronics, IEEE, December 17–20, 2016, Cairo, Egypt. https://doi.org/10.1109/ICM.2016.7847942
120. Zhonghua M (2008) Tracking differentiator in the application of signal processing and theory research. In: Proceedings-2008 international seminar on future biomedical information engineering, FBIE 2008, vol 57(1), pp 382–384. https://doi.org/10.1109/FBIE.2008.57
121. Huang J et al (2020) Research on position servo system based on fractional-order extended state observer. IEEE Access 8:102748–102756. https://doi.org/10.1109/ACCESS.2020.2997407
122. Fareh R (2019) Control of a single flexible link manipulator using fractional active disturbance rejection control. In: 2019 6th international conference on control, decision and information technologies, CoDIT 2019, pp 900–905. https://doi.org/10.1109/CoDIT.2019.8820708
123. Wang C et al (2016) Design of PID and ADRC based quadrotor helicopter control system. In: Proceedings of the 28th Chinese control and decision conference, CCDC 2016, pp 5860–5865. https://doi.org/10.1109/CCDC.2016.7532046
124. Shi DF et al (2021) A novel active disturbance rejection control with a super-twisting observer for the rocket launcher servo system. Shock Vib. https://doi.org/10.1155/2021/6617599
125. Alawad NA, Humaidi AJ, Al-Araji AS (2022) Improved active disturbance rejection control for the knee joint motion model. Math Model Eng Probl 9(2):477–483. https://doi.org/10.18280/mmep.090225
126. Peng C et al (2013) ADRC trajectory tracking control based on PSO algorithm for a quad-rotor. In: Proceedings of the 2013 IEEE 8th conference on industrial electronics and applications, ICIEA 2013, pp 800–805. https://doi.org/10.1109/ICIEA.2013.6566476
127. Kaur S et al (2020) Tunicate Swarm Algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intell 90(February):103541. https://doi.org/10.1016/j.engappai.2020.103541

128. Sharma A et al (2021) Parameter extraction of photovoltaic module using tunicate swarm algorithm. Electronics (Switzerland) 10(8)https://doi.org/10.3390/electronics10080878
129. Raheema SJ, Saleh MH (2021) An experimental research on design and development diversified controllers for tri-copter stability comparison. In: IOP conference series: materials science and engineering, vol 1105(1), pp 012019. https://doi.org/10.1088/1757-899x/1105/1/012019
130. Al-Qassar AA, Abdulkareem AI et al (2021) Grey-Wolf optimization better enhances the dynamic performance of roll motion for tail-sitter VTOL aircraft guided and controlled by STSMC. J Eng Sci Technol 16(3):1932–1950
131. Al-Qassar AA, Al-Obaidi ASM et al (2021) Finite-time control of wing-rock motion for delta wing aircraft based on whale-optimization algorithm. Indones J Sci Technol 6(3):441–456. https://doi.org/10.17509/ijost.v6i3.37922
132. Waheed ZA, Humaidi AJ (2022) Design of optimal sliding mode control of elbow wearable exoskeleton system based on whale optimization algorithm. Journal Européen des Systèmes
133. Humaidi AJ, Hasan AF (2019) Particle swarm optimization–based adaptive super-twisting sliding mode control design for 2-degree-of-freedom helicopter. Meas Control (United Kingdom), 52(9–10):1403–1419. https://doi.org/10.1177/0020294019866863

Robust Adaptive Sliding Mode Controllers Design for a Non-holonomic Mobile Robot

Shibly Ahmad Al-Samarrai, Yasir Khudhair Al-Nadawi, Taif Ghadban Hama, and Taha Adel Al-Gadery

Abstract This chapter considers the design of robust adaptive sliding mode controllers (ASMCs) for mobile robot trajectory tracking as an example of a nonholonomic mechanical system. Unlike the classical SMC, the ASMC does not need to know the bounds on the system parameters variation and on the disturbances. The mobile platform mathematical model has been derived here, with considering the existence of one holonomic and two nonholonomic constraints in addition to the uncertainty in the system model. The input–output dynamic model was obtained using partial feedback linearization method which was considered uncertain, and subjected to friction torques on the wheels. With variable sliding mode control gain which is adapted according to two adaptation laws, the platform will be enforced to follow the desired trajectory utilizing the proposed controllers. The first is based on the classical sliding mode control (ACSMC), where two control laws were derived with two adapted gain parameters, while the second control approach is based on the unity sliding mode control (AUSMC) where only one control gain adaptation is needed. Theoretical analysis confirms that the closed-loop control system variables will enter a positively invariant set in a finite time. The size of the steady-state error represented by the positively invariant set is found function to a boundary layer whose value can be adjusted by a suitable selection of the adaptation law parameters. The performance of the two adaptive controllers and the classical SMC were verified and compared by implementing in MATLAB. Simulations illustrate the ability of the

S. A. Al-Samarrai (✉)
University of Technology, Baghdad, Iraq
e-mail: 60132@uotechnology.edu.iq

Y. K. Al-Nadawi
Michigan State University, East Lansing, MI 48824, USA

T. G. Hama
Ministry of Health and Environment Medical City, Baghdad, Iraq

T. A. Al-Gadery
Tikrit University, Tikrit, Iraq
e-mail: Taha.a.tahir@tu.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_15

proposed ASMCs in forcing the mobile platform to track efficiently the desired trajectory for various path shapes. Finally, in spite the simplicity of designing AUSMC where only one gain needed to be adapted, the CASMC showed better performance as compared to the unity sliding mode controller in terms of the transient and tracking error. Moreover, both controllers induced chattering with small amplitude especially when an approximation was used.

Keywords Mobile robot · Non-linear control · Sliding mode control · Adaptive control

Nomenclature

Q	System's configuration space
n	Dimension of configuration space
SMC	Sliding Mode Control
ACSMC	Adaptive Classical Sliding Mode Control
AUSMC	Adaptive Unity Sliding Mode Control
VSS	Variable Structure Systems
L(r, $\dot{r}$, t)	Euler Lagrange equation
$q, \dot{q}, \ddot{q}$	Position, velocity, and acceleration vector of mobile platform
q	System Dynamic configuration
r	The radius of the mobile platform wheel
S	Sine
2b	Distance between the two wheels
v(q, $\dot{q}$, t)	Velocity of particles
Po	Point in the middle of the axis that connects the two wheels
T(q, $\dot{q}$, t)	Kinetic energy
Pc	The point on the platform chassis
U(q, t)	Potential Energy
d	Distance between P_o and P_c
C	Integral constant
C	Cosine
ρ	Disk radius
v_x^{prime}	Linear velocity for P_o on the x-axis
mc	Chassis mass
v_y^{prime}	Linear velocity for P_o on the y-axis
mw	Wheels mass
v_{w1}^{prime}	The linear velocity of the first wheel
xo, yo, zo	Inertial (reference) frame
v_{w2}^{prime}	The linear velocity of the second wheel
$x1, y1, z1$	Attached frame to the platform on the point P_o
$A(q)$	Non-holonomic matrix
$x2, y2, z2$	Attached frame to the platform on the point P_c

$N(q)$	Kernel space matrix
φ	Chassis rotation about the z-axis
$M(\ddot{q})$	Positive-definite matrix(mass and inertia)
$\theta 1$	The left wheel rotation angle
$D(q, q')$	Carioles matrix
$\theta 2$	The right wheel rotation angle
$G(q)$	Potential energy matrix
w	Rotation velocity
λ	Lagrange multiplier
$\delta(e, u)$	Perturbation term
V_{mx}	The linear velocity of the left wheel on the x-axis
V_{w1y}	The linear velocity of the left wheel on the y-axis
V_{w2x}	The linear velocity of the right wheel on the x-axis
V_{w2y}	The linear velocity of the right wheel on the y-axis
W_{x1}	Rotation velocity for the left wheel on the x-axis
W_{y1}	Rotation velocity for the left wheel on the y-axis
W_{z1}	Rotation velocity for the left wheel on the z-axis
W_{x2}	Rotation velocity for the right wheel on the x-axis
W_{y2}	Rotation velocity for the right wheel on the y-axis
W_{z2}	Rotation velocity for the right wheel on the z-axis

1 Introduction

A mobile robot is a controlled machine that can move in its environment and/or interact with it. The continuously increasing popularity of mobile robots and their wide range of applications in different fields such as exploration, transportation etc. have earned mobile robotics a rapidly growing interest in modern literature. In this chapter, two sliding mode-based adaptive control schemes are proposed for the lateral movement tracking problem of a non-holonomic differential drive mobile robot. But, before stating the problem statements and the motivation behind this work, the following three subjects in mechanics and control theory should be briefly introduced:

- **Holonomic and Non-holonomic Constraints**

In mechanical literature, the constraints of dynamics are often seen as restrictions on the position and velocity of a system [1]. There are two types of constraints or restrictions; holonomic and nonholonomic constraints.

Nonholonomic systems are present in a great variety of environments like robotics, wheeled vehicle, satellite dynamics, manipulation devices, and locomotion systems [2]. A Nonholonomic system is a mechanical system with Nonholonomic constraints. A constraint is a condition imposed on the possible motions of a system. A penny is an example when rolling without slipping over the floor. The point of contact with the surface has a zero linear velocity, otherwise, the penny would slip. The robotic manipulator with various links is another example. For the robot manipulator each link can be thought of is a rigid body which has the ability to move arbitrarily as long as it maintains contact with the other links imposed by the joints. When only the configuration variables exist in the constraint's expression, they named as Holonomic, as in the case with the robotic manipulator. When it is not possible to express the constraints without involving the velocities of the system, those constraints are Nonholonomic, i.e. can not express in terms of configuration variables only. This is the case with the rolling penny for example [2].

The study of mechanical systems was traditionally carried out from two points of view. The Classical Mechanics forms the first point of view which focuses on more theoretically oriented problems such as the role of dynamics, the analysis of symmetry and related subjects like reduction. In the second point of view, the discipline of Nonlinear Control Theory tries to answer more practically oriented questions such as the accessibility and controllability, the motion and trajectory planning, the motions with the least amount of time or energy spent by the system (optimal control), also how to follow a desired trajectory (trajectory tracking), and the stability of the system. Of course, both viewpoints are complementary and mutually interacted.

- **Robust Control**

Robust control is an aspect of modern control theory which mainly concern with the uncertainties of a controller design. The methods for robust control are designated to work efficiently where there are uncertain perturbations within some sets. Robust methods aim to get good stability and good performance and achieved the objective of the controller with the existence of limited modelling errors. The design of traditional control methods based on the frequency domain approach was considerably robust although the robustness of the earlier introduced state-space methods was questionable [3], prompting the need for their improvement. This marked the initiation of the robust control theory which started in the 80s and 90s and is still in use now. The robust control obligations, unlike the adaptive control obligations, are static and do not adapt to variation measurements. The design of the controller enables it to work with the assumption that some variables may not be known or bounded [4, 5]. Informally, a designed controller for any class of parameters is considered robust if it can efficiently perform well on another class of parameters. In the early 1970s and 1980s, the beginning of the modern theory of robust control was observed, but later, several techniques were developed for handling the uncertainty of bounded systems [6, 7].

- **Sliding Mode Control**

One type of robust control approaches for the Variable Structure Systems (VSS) on differential equations with a discontinuous control input was introduced in the Soviet Union in the 1950s after the efforts of Filippov [8] and Popovski [9]. Over these years, the VSS has been continuously exploited for non-continuous dynamic systems control or for accepting non-continuous inputs [10, 11]. The control parameters of

VSS can vary, allowing the maintenance of certain conditions in the phase space. The closed loop system on a considered hypersurface evolves in a sliding mode, and in this condition, the sensitivity of the system to outer constraints as well as changes in system parameters is reduced [12]. The efficiency of this control system has been proven in tracking control and stabilization control problems [12, 13] but has often attracted much criticism as a result of the chattering phenomenon due to the non-continuous control type. There are suggestions for improvements, as evidenced in the equivalent control [13, 14] where the equivalent control law can be designed as a continuous function [15], as well as the generalized control [12, 16]. Other approaches include controls with higher orders of structure and sliding mode [17].

The most cited English text on SMC was reported by Itkis (1976) [18]. The major significance of the SMC theory was completed in 1980 and later cited in a monograph published in 1981 in the Russian language, and later translated to the English language [14] by Utkin. There are several structures in these systems, with rules for real-time switching of structures to achieve good system performance. The major concept of the variable structure control with the sliding mode is first applied to CT (continuous time) systems. Briefly, the state is first driven to a sliding subspace by the control and enforces it to remain there. This way of motion is referred to as a reduced-order sliding mode. A suitable design confers several beneficial features such as minimizing disturbances, reducing desired dynamics and optimal behaviour, as well as robust stability to the sliding motion.

One of the most famous applications for Nonholonomic mechanical systems is the mobile platform. This system contains both holonomic and Nonholonomic constraints. these constraints, together with several factors such as the type of nonlinearities in such systems, the disturbances, the presence of non-smoothness situations like friction, and the MIMO problem, add difficulty to the lateral movement tracking control problem for the mobile platform. Furthermore, the range of uncertainties in the model and the external disturbances usually lead to a conservativeness problem in the control design, especially when assigning the gains for the controller with VSS controllers which could lead to higher levels of chattering and reduces the life span of the actuators in the system.

The importance of the mobile platform robots, together with the control problems mentioned above, drives the motivation behind this chapter to design robust sliding mode control for the mobile platform, with minimum control gain and attenuated chattering.

The contribution of this chapter can be summarized by two points: The use of a new modelling approach in which the mathematical model of the mobile robot was driven by eliminating the Lagrange multiplier using the Pfaffian matrix and then using partial feedback control to reduce the system model into one that satisfies the matching conditions and more suited for the application of sliding mode based controller. And the proposition of two adaptive control algorithm schemes. The first one is the ACSMC approach and the second one is the AUSMC approach. With the adaptation capability, the parametric uncertainty and input disturbance bounds need not to be known in advance since the adaptation laws are designed to estimate the switching gains to compensate for the effect of the lumped perturbations in both approaches. The AUSMC approach was more suited for the MIMO systems; furthermore, the number of control design parameters was fewer in the AUSMC which gave it a simplicity advantage over the ACSMC approach. The two controllers were applied to a mobile robot platform path-tracking problem under the presence of model parameter uncertainties and disturbance input due to friction torques. The performance of both controllers is validated in simulation and compared with classical sliding mode controllers under various desired routes. The results demonstrate comparable results in terms of the achieved tracking error.

The organization of this chapter is as follows: Sect. 2 presents the state of the art on Nonholonomic differential drive mobile robot control. In Sect. 3, the derivation of the Nonholonomic mobile robot mathematical model is presented. Section 4, focused on the controller design aspect, as well as the coupling and de-coupling problems, chattering problems and present ways of eliminating them. Additionally, present the control system design using the Adaptive Classical SMC (ACSMC) and the Adaptive Unity SMC (AUSMC), which is based on the Unity SMC (USMC) for the mobile platform. The closed loop system analysis is also presented to show that both stability and robustness of the system are guaranteed in the presence of system parametric uncertainty and the disturbance inputs due to the existence of friction torque. The validity of the proposed approach is confirmed by simulation under various reference inputs in Sect. 5. Finally, the chapter ends with the conclusions.

2 Related Work

In the following, a brief review of the state of the arts on Nonholonomic differential drive mobile robot control.

In [19], a robust (min–max) H_∞ as a linear control approach was proposed. This control technique has been applied to achieve trajectory tracking for Nonholonomic two-wheeled mobile robots in the presence of system parameter uncertainty and input disturbance. In their design, the authors considered the kinematic models of the system to design a robust H_∞ controller and considers that the mobile system has 3 spaces (D.O.F). Then, the controller was designed using the linearized model under the assumption of nominal initial conditions. The main drawback of this approach is the conservativeness problems that may arise when estimating the bounds of the perturbation term in the model.

A four-wheeled mobile robot trajectory tracking controller was developed by [20] based on kinematics. In this article, a mathematical model for a mobile platform was developed and a backstepping controller was proposed to control the platform while considering the non-holonomic constraints. The proposed controller solved the tracking problem; However, the parametric uncertainty and external disturbance were not considered in the controller design.

In [21], a robust control system based on ASMC was proposed for achieving path-tracking control of a non-holonomic mobile manipulator mounted on a wheeled chassis with parametric uncertainties and disturbances. A kinematic controller was designed with 6 DOF, (4 for the platform and 2 for the two link manipulators). The effectiveness of the proposed controller was verified via simulation; However, the article neglected the effect of the coupling problems that may associate with multi-input systems.

A novel modelling approach for wheeled mobile platforms with differential chassis has been proposed by [22]. This method included the effects of friction forces in the model which increased the complexity of the model but achieved more accurate results.

A hierarchy sliding mode controller for underactuated systems was proposed in [23]. The aim is to force a differential-drive mobile robot to track a certain trajectory in the presence of friction and uncertainties. The simulations results showed the effectiveness of the proposed approach, but the high levels of switching were a drawback.

A novel sliding mode controller was proposed for trajectory tracking of a differential-drive mobile robot was proposed in [24]. The sliding surface uses a Proportional-Integral form for trajectory tracking while for velocity tracking another sliding surface uses the Proportional-Integral-Derivative (PID) form. The chattering problems were mitigated by the proposed sliding surface function design.

To ensure the robust and accurate tracking of a planned path, a robust flatness-based feedback controller was proposed in [25]. The path is planned using a decoupling planning approach.

The non-singular fast terminal sliding mode controller was proposed in [26] to design a novel controller for the tracking problem of a differential drive mobile robot. In this work, the tracking error was driven to zero in a finite time. Moreover, inside the region of interest, the obstacle avoidance algorithm was incorporated to avoid collision.

In [27], a novel flexible lateral control scheme was proposed. To evaluate the centroid slip angle and yaw rate, this scheme used cubature Kalman algorithm; additionally, a preview angle-enhanced sliding mode controller with fuzzy compensation was utilized to improve the tracking accuracy and robustness. While the proposed approach achieved high tracking accuracy and stability, the use of the Kalman algorithm and the fuzzy compensation added to the complexity of the approach.

In [28], a robust tracking controller was proposed using Lyapunov redesign for the Nonholonomic mobile robot. In this work, a modified Kanayama transformation was used to transform the system model into a form in which the matching condition is satisfied. While the proposed approach achieved accuracy and robustness, the bounds on perturbation should be known in advance which could lead to conservativeness issues.

In [29, 30], intelligent approaches were proposed for mobile platform lateral movement tracking problem. While these approaches achieved the desired objectives, these algorithms were relatively complex compared with the previous approaches since they considered the formation control of multiple mobile robot agents, this by itself is more difficult than controlling a single robot.

3 Mathematical Model of Mobile Platform

In this section, the mathematical model for the mobile platform is derived via the Euler–Lagrange equations. The homogenous transformation matrix method is used to obtain the holonomic and non-holonomic constraints, and their effect is represented in the model with Lagrange multiplier. By eliminating the Lagrange multiplier from the model, a reduced-order model is obtained with four differentiable DOFs equations.

3.1 Analytical Mechanics

In analytical mechanics, the motions along a generalized set of coordinates are described in terms of system's configuration space and denoted as Q [31]. It represents a collection of variables with values uniquely specified by the 3-D position coordinates of each physical point of the mechanism. A generalized set of coordinates known as the number of degrees of freedom of a system is the basic requirement for the determination of the location of all the points in the mechanism. Fewer variables may not be enough to determine these locations [31].

3.2 The Euler–Lagrange Equation

The principle of Least Action is expressed in the form of a function $L(q, \dot{q}, t)$ referred to as the Lagrangian, which it is defined as the difference between the kinetic energy and its potential energy. For example, for a particle of mass m with position r, the Lagrangian L can be represented as:

$$L(r, \dot{r}, t) = \frac{m}{2}|\dot{r}|^2 - u(r, t) \tag{3.1}$$

where $\frac{m}{2}|\dot{r}|^2$ is the kinetic energy of m and $u(r, t)$ is the potential energy. This implies the conservation of energy. The Lagrangian function of a simple mechanical system of N particles is constructed by executing the following steps [32]:

1. Defining the generalized coordinates: This requires an instant configuration variable that describe N-particles at a time $(q_1(t), \ldots, q_n(t))$.
2. Construction of each particle's position vector $r_i(q)$ and its related velocity in Cartesian coordinates $v_i(q, \dot{q}, \mathrm{t})$ for$i = 1, \ldots, N$.
3. Construction of the kinetic and potential energies of the system $T(q, \dot{q}, t)$ and $U(q, t)$ respectively.
4. Combination of the kinetic and potential energies to obtain the Lagrangian $L(q, \dot{q}, t)$.

The Euler–Lagrange equation is obtained from the Lagrangian for each generalized coordinate q

$$L(q, \dot{q}, t) = T(q, \dot{q}, t) - U(q, t) \tag{3.2}$$

From this Lagrangian $L(q, \dot{q}, t)$, the Euler–Lagrange equation is given as [33];

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = 0, j = 1, \ldots, n \tag{3.3}$$

3.3 Constraints in Mechanical Systems

Generally speaking, the motion can be restricted with kinematic constraints, which directions in which system can take. The function of a mechanical system is guided by a correct application of these constraints to meet the system's operational requirements. The constraint is determined by the velocity-dependent coordinates and orientation of the position. Several constraint-dependent positions and orientations called holonomic, while constraints that depend on velocity are referred to as non-holonomic [34]. Systemic motions are vulnerable to constraints which may come from the mechanical structure itself or from the manner the system is controlled or actuated [34].

Constraints, whether unilateral (one sided) or bilateral (two sided), can be grouped into two groups as follows:

3.3.1 Holonomic Constraint

Holonomic constraints are only dependent on the present position of the material, and can be presented either through the amendment of the configuration space (done by forcing the holonomic constraints) or through a presentation of the holonomic constraint as a linear non-holonomic constraint (via extending the constraint manifold tangent bundle to the span for tangent vector to the motion manifold on the configuration space). The distribution (span for tangent vector to the motion manifold on the configuration space) on smooth manifold M is known as a smooth vector sub-bundle of the tangent bundle TM of M, therefore, the original or holonomic constraint manifold is an integral aspect of this span for tangent vector to motion manifold. Assume D to be a tangent vector of motion manifold, and Q as the linear non-holonomic constraint; a system motion $t \rightarrow q(t)$ is dynamically allowed if its velocity $\dot{q}$(t) at time t is represented in $D_q(t)$. Should D be integrable, then, the system is holonomic where each integral manifold $M of D$ represents a holonomic constraint [35]. If all the constraints in a system are holonomic, the system is referred to as a holonomic system.

$$h_i(q) = 0 \;\; i = 1, 2, \ldots, k < n \tag{3.4}$$

where n is the dimension of the configuration space $\mathcal{Q}$.

Holonomic constraints aim at reserving the achievable system configurations to a $(n - k)$ dimensional smooth sub-manifold of $\mathcal{Q}$. Holonomic constraints can be directly worked with, via the implicit function theorem that made it possible to solve Eq. (3.4) in terms of $(n - k)$ generalized coordinates in order to reduce the remaining k variables emanating from the problem. One limitation of this approach, it may generate algebraic singularities. The system configuration can be accurately specified by defining the $(n - k)$ new coordinates on the limited submanifold which marked the actual systemic degree of freedom.

3.3.2 Non-holonomic Constraints

Non-holonomic mechanics are found in systemic motions that are prone to nonintegrable constraints. Non integrable constraints are constraints on velocities, such as those found in rolling and skating motions. Nonholonomic mechanics are not easily fitted into classical mechanics due to their non variational nature [36].

There are two forms of nonholonomic systems, the first form is those with dynamic nonholonomic constraints. These are constraints which can be represented by Euler–Lagrange equations such as angular momentum, or the more common momentum maps. These constraints are internally generated (not introduced from an external source); they may result from the equations of motion and are sometimes more ideal to be handled as conservation laws [31]. The second form is those that are introduced by kinematics (known as kinematic nonholonomic constraints), such as rolling constraints [32]. Kinematic constraints can be presented in terms of generalized coordinates and velocities $(A_i(q, \dot{q}) = 0, i = 1, \ldots, k < n)$. They restrict the permitted systemic motions via a limitation of the generalized velocities obtainable at a given configuration. Usually such constraints are represented in mechanics in the Pfaffian form (i.e. linear in the generalized velocities) [32]

$$a_i^T(q)\dot{q} = 0, i = 1, \ldots, k < n$$

or in a compact matrix form

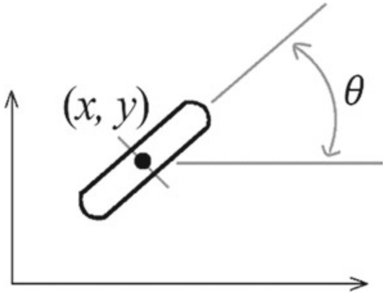
$$a^T(q)\dot{q} = 0 \tag{3.5}$$

The vectors $a_i \colon Q \to \mathbb{R}^n$ are considered smooth and linearly independent.

The holonomic constraints (3.4) indicate the presence of kinematic constraints. It can be represented as:

$$\frac{\partial h_i}{\partial q}\dot{q} = 0, i = 1, \ldots, k. \tag{3.6}$$

Fig. 1 Unicycle configuration



However, this is not always true kinematic constraints (Eq. 3.5) which it may not always be integrable (i.e. cannot be represented as in Eq. (3.4)). The nonholonomic constraints, when compared to the holonomic constraints, completely restricts the mobility of the system in a different manner. This can be better explained by assuming a single Pfaffian constraint $a^T(\dot{q})q = 0$.

Constraint (3.6) can be integrated as $h(q) = c$, if it is holonomic; where $\frac{\partial h}{\partial q} = a^t(q)$ and c are the integration constants. The system motion will therefore be restricted to a specified surface level of h based on the initial condition through the value of $c = h(q(0))$. Now, let constraint (3.6) be considered as nonholonomic; even though the instantaneous system mobility is restricted to $(n - 1)$-dimensional space, still, it can reach any configuration in Q. The number of differentiable degrees of freedom will consequently be reduced to $n - 1$ though there will be no reduction in the generalized coordinates number. Generally, it can be said that the access to the complete configuration space of any mechanical system with k nonholonomic constraints and n generalized coordinates is preserved even when the generalized velocities are restricted to $(n - k)$-dimensional subspace.

Let us take the simplified kinematic model of a unicycle as a typical example of a nonholonomic system. Here, the unicycle configuration is expressed by the planar coordinates (x, y) of the wheel's contact point with the ground, the configuration $q = (x, y, \theta)$, and by the wheel's orientation with respect to the x-axis represented by angle θ. The wheel is subjected to a rolling-without-slipping constraint, written as;

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$$

3.4 Differential Drive Mobile Robot Description

Mobile platforms are platforms that can automatically move from place to place without any human or external assistance. Unlike industrial robots, mobile platforms are designed to freely move within a predefined work space to achieve a designated target. The mobility of mobile platforms made them ideal for use in both structured and unstructured scenarios.

Wheeled mobile robot (WMR) drives are classified into differential, tricycle, omni directional, synchro, Ackerman steering, and skid steering drives [37] The differential drive is made up of two independently driven fixed powered wheels mounted on either side (L and R) of the robot platform. The balance and stability of the wheels are maintained by one or two passive castor wheels. The differential drive is the simplest form of mechanical drives because its operation does not require the rotation of a driven axis. The robot is moved straight forward or backward by a simultaneous rotation of the wheels. Moving one of the wheels at a faster rate than the other makes the robot to follow a curved path along the arc of an instantaneous circle, but moving both wheels at an equal velocity but in opposite directions makes the robot to turn around the midpoint of the two driving wheels. These movements are depicted in Fig. 2. This form of WMR clearly cannot turn on the spot [37].

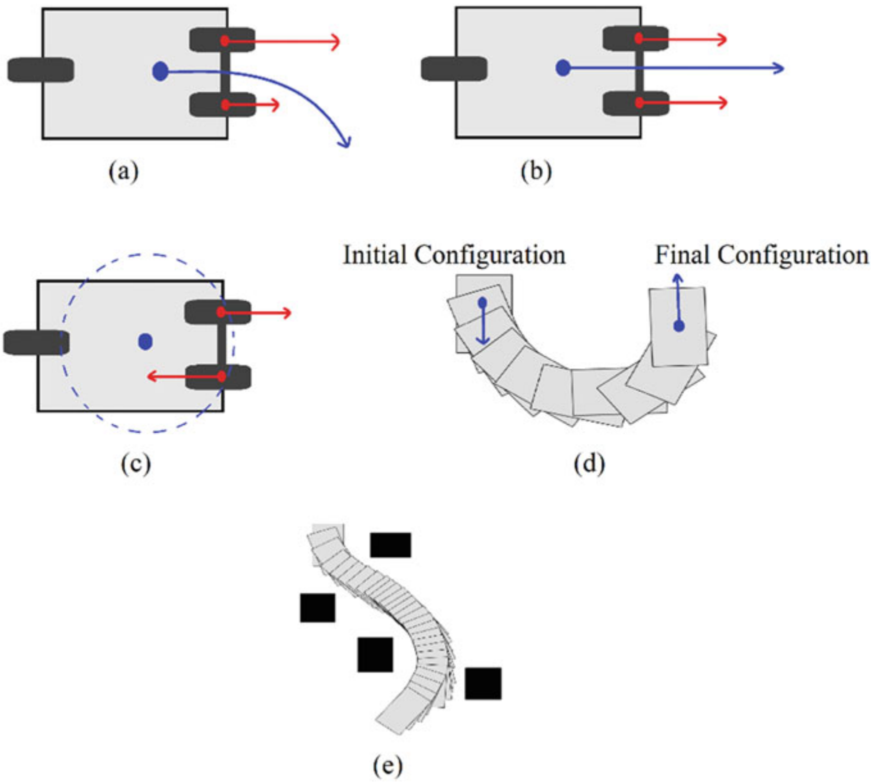


Fig. 2 The possible movements of a differential drive, **a** curved path, **b** straight path, **c** circular path, **d** obstacle-free maneuvering when moving from an initial to a final pose, and **e** maneuvering when moving from an initial to a final pose in the presence of obstacles

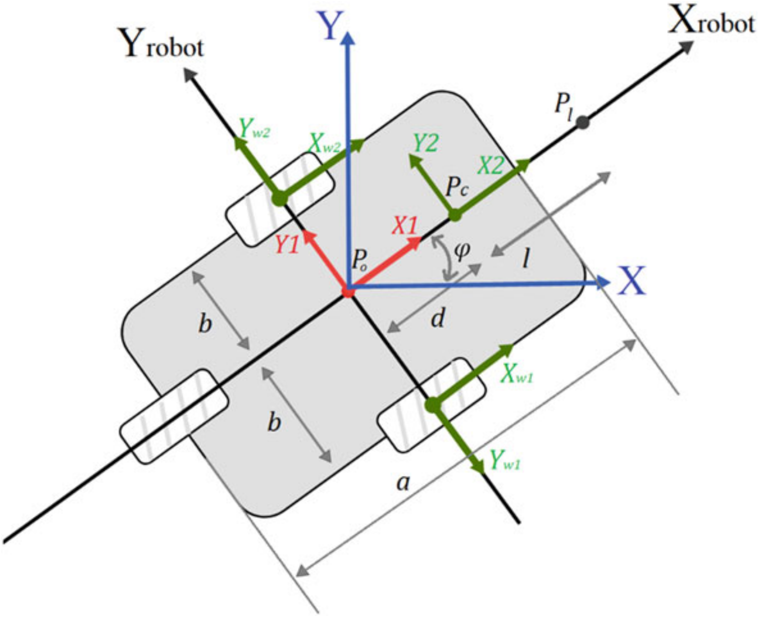


Fig. 3 Differential-drive mobile robot configration

The mobile platform considered in this study is a differential drive mobile robot as defined in [37]. In the dynamic control systems, the two-wheeled systems are the commonest but challenging case study due to the presence of nonholonomic constraints, as well as their related stability problems. A mobile platform is a famous and common application for nonholonomic mechanical systems. It is comprised of two wheels driven by two DC motor and rigid cart. As presented in Fig. 3, $2b$ represents the distance between the two wheels, d is the distance between p_o which is located in the middle of the axis that connects the two wheels, and p_c is the point on the platform. Point p_o is essential when computing nonholonomic constraints.

3.5 *Mathematical Model of a Differential-Drive Mobile Platform*

3.5.1 Kinematic Equation

Platform kinematics implies dealing with robotic configurations in their workspace, the relationship between their geometric parameters and dimensions, as well as the constraints assumed in their paths. Kinematic equations are mainly developed based on the dimensions and geometrical structure of the working platform. Mobile platforms may have one or more wheels that may or may not have motion constraints

[37]. Kinematic studies are basically required for studying the stability features, dynamics, and control of the working platform. Therefore, kinematic equations can be referred to as equations that mathematically represent the motion of the system (position and velocity) without regard to the forces that oppose the motion.

In this section, the kinematic equations of the chassis and wheels of a mobile platform are presented. These equations are computed using homogenous transformation matrices under the following assumptions [37]:

- The deformation due to the gravity effects is negligible.
- The mobile chassis is assumed to have uniform material properties.
- The wheels are assumed to move in a pure rolling motion with non slipping.

The coordinates shown in Fig. 3 were used; X_0, Y_0, Z_0 is the inertial reference frame. The kinematic equation was computed based on these coordinates. The frame $x_1y_1z_1$ was attached to the chassis positioned on the axis between the wheels. The frame $x_2y_2z_2$ was attached to the chassis at point p_c. Angle φ is a representation of the chassis rotation about z_0 and (x, y) represents the distance between the centers of x_0, y_0, z_0 and of $x_1y_1z_1$, and r is the mobile wheel's radius. As a result, the system configurational variables are described as $[x, y, \varphi, \theta_1, \theta_2]$, were φ is the platform's rotational angle (chassis) measured from the inertial x-axis, θ_1 and θ_2 represents the angle of rotation of the first and second wheels, respectively. The chassis dynamics is nonlinear and it was coupled with the wheel dynamics, while the body of the chassis was taken to be symmetric, requiring no slip condition.

Homogeneous Transformation Matrices

This is one of the ways of finding the kinematic equations of homogenous matrix systems $\in R^{4\times 4}$. Square matrix comprise of the rotating part that represents the rotation movement of the system, and is represented by first 3×3 square matrix. The last column in this matrix represents the system's linear motion [38]. The linear and angular velocity for the chassis and the wheels are presented below as follows:

1. Chassis: The linear velocity of chassis at point (P_o) is given by

$$\left.\begin{aligned} v_{xp_o} &= \dot{x} \\ v_{yp_o} &= \dot{y} \end{aligned}\right\} \tag{3.7}$$

   While the angular velocity of chassis is;

$$\omega = \dot{\varphi} \tag{3.8}$$

   which rotates around the z-direction.
2. Wheels: Firstly, the linear velocity, with respect to the inertial frame, of the wheels are obtained as:

$$\mathrm{v}_{w_1\mathrm{c}} = \begin{bmatrix} bC(\varphi)\dot{\varphi} + \dot{x} \\ bS(\varphi)\dot{\varphi} + \dot{y} \\ 0 \\ 0 \end{bmatrix} \text{and } \mathrm{v}_{w_2\mathrm{c}} = \begin{bmatrix} -bC(\varphi)\dot{\varphi} + \dot{x} \\ -bS(\varphi)\dot{\varphi} + \dot{y} \\ 0 \\ 0 \end{bmatrix}$$

where $\mathrm{v}_{w_1\mathrm{c}}$ and $\mathrm{v}_{w_2\mathrm{c}}$ are the velocity of the first and the second wheels centers respectively. Also $S(\varphi)$ and $C(\varphi)$ refer to $\sin(\varphi)$ and $\cos(\varphi)$ respectively.

Secondly, the wheels angular velocity are obtained as follows;

$$S_{w_1} = \begin{bmatrix} 0 & -\dot{\varphi} & C\varphi\dot{\theta}_1 \\ \dot{\varphi} & 0 & S\varphi\dot{\theta}_1 \\ -C\varphi\dot{\theta}_1 & -S\varphi\dot{\theta}_1 & 0 \end{bmatrix} \text{and } S_{w_2} = \begin{bmatrix} 0 & -\dot{\varphi} & C\varphi\dot{\theta}_2 \\ \dot{\varphi} & 0 & S\varphi\dot{\theta}_2 \\ -C\varphi\dot{\theta}_2 & -S\varphi\dot{\theta}_2 & 0 \end{bmatrix}$$

where $S(\omega)$ is a 3×3 skew symmetric matrix from which the angular velocity is computed with the aid of the general form of $S(\omega)$, which it is given by;

$$S(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The calculations details are given in Appendix A.

Nonholonomic Constraints Equations

Two type of nonholonomic constraints imposed on the mobile platform; Lateral and no-slip constraints. These two types are derives here as in the following items:

- ***The lateral constraint***: the platform motion has constraints at each instant from moving in the lateral direction, i.e., the instantaneous platform lateral velocity is equal to zero. Therefore, we need to derive the wheels velocity in the lateral direction, and then equating it to zero. At p_o the velocity in the tangential and lateral velocity are obtained as follows;

$$\mathrm{v}_{p_o} = \begin{bmatrix} \mathrm{v}_{xo} \\ \mathrm{v}_{yo} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} C\varphi & -S\varphi \\ S\varphi & C\varphi \end{bmatrix} \begin{bmatrix} \acute{\mathrm{v}}_{xo} \\ \acute{\mathrm{v}}_{yo} \end{bmatrix} \Rightarrow \acute{\mathrm{v}}_{x,y} = \begin{bmatrix} \acute{\mathrm{v}}_{\dot{x}} \\ \acute{\mathrm{v}}_{\dot{y}} \end{bmatrix} = \begin{bmatrix} C\varphi & S\varphi \\ -S\varphi & C\varphi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

The lateral velocity $\acute{\mathrm{v}}_{yo}$ from above is;

$$\acute{\mathrm{v}}_{\dot{y}} = -S\varphi\dot{x} + C\varphi\dot{y}$$

Accordingly, the lateral nonholonomic constraints, as can be seen from Fig. 4, is obtained by equation $\acute{\mathrm{v}}_{\dot{y}}$ to zero, which represent the first noholonomic constraint;

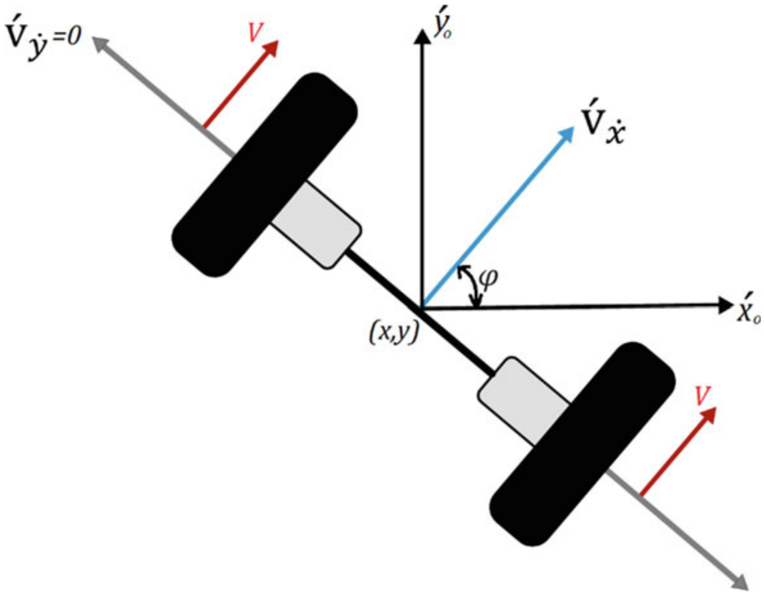


Fig. 4 Illustration of lateral nonholonomic constraints

$$-S\varphi\dot{x} + C\varphi\dot{y} = 0 \tag{3.9}$$

- ***The no-slipping constraint***: two no-slipping constraints at the wheels of the mobile platform are imposed. These constraints are derived by making the longitudinal velocity of the wheel center, which computed according to the platform longitudinal velocity, equal to the wheel velocity which computed based on wheel angular velocity. With the aid of Fig. 5, the wheel velocity in the inertial frame, a tangential $\acute{\mathrm{v}}_{wt_1\mathrm{c}}$ and normal velocity components with respect to the wheel, is obtained as follows;

$$\acute{\mathrm{v}}_{w_1\mathrm{c}} = \begin{bmatrix} C\varphi & S\varphi \\ -S\varphi & C\varphi \end{bmatrix} \mathrm{v}_{w_1\mathrm{c}} = \begin{bmatrix} C\varphi & S\varphi \\ -S\varphi & C\varphi \end{bmatrix} \begin{bmatrix} \dot{x} + bC\varphi\dot{\varphi} \\ \dot{y} + bS\varphi\dot{\varphi} \end{bmatrix}$$

$$\Rightarrow \acute{\mathrm{v}}_{wt_1\mathrm{c}} = \dot{x}C\varphi + \dot{y}S\varphi + b\dot{\varphi}\left(S^2\varphi + C^2\varphi\right) = \dot{x}C\varphi + \dot{y}S\varphi + b\dot{\varphi}$$

but with respect to the wheel angular velocity, $\acute{\mathrm{v}}_{wt_1\mathrm{c}}$ is given by

$$\acute{\mathrm{v}}_{wt_1\mathrm{c}} = r\dot{\theta}_1$$

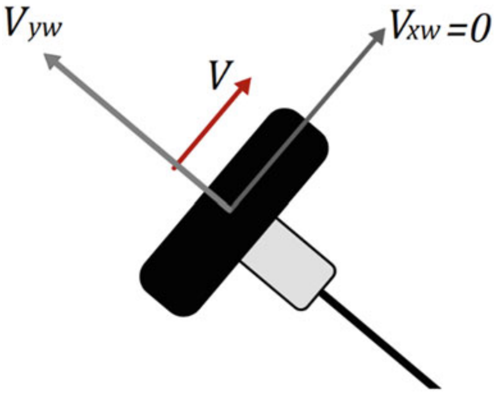


Fig. 5 Illustration of the no-Slipping condition

therefore, the no-slipping condition on the first wheel becomes;

$$\dot{x}C\varphi + \dot{y}S\varphi + b\dot{\varphi} = r\dot{\theta}_1 \tag{3.10}$$

In a similar way, for the second wheel, the no-slipping condition is obtained as:

$$\dot{x}C\varphi + \dot{y}S\varphi - b\dot{\varphi} = r\dot{\theta}_2 \tag{3.11}$$

From the above no-slipping constraints, the second noholonomic constraint is obtained by summing Eqs. (3.10) and (3.11);

$$\dot{x}C\varphi + \dot{y}S\varphi - (r/2)\dot{\theta}_1 - (r/2)\dot{\theta}_2 = 0 \tag{3.12}$$

Collecting the nonholonomic constraints in Eqs. (3.9) and (3.12) in a matrix form, the Nonholonomic constraint matrix is obtained as;

$$A(q)\dot{q} = \begin{bmatrix} -\sin\varphi & \cos\varphi & 0 & 0 \\ \cos\varphi & \sin\varphi & -r/2 & -r/2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{3.13}$$

where $\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T$ are the differentiable DOF for the platform. The nonholonomic constraint is non integrable constraint, and Eq. (3.13) is it's Pfaffian form.

Holonomic Constraint Equation

There is one holonomic constraint imposed to the motion of the mobile platform. It can be obtained by subtracting Eq. (3.11) from Eq. (3.10);

$$\dot{\varphi} = r\left(\dot{\theta}_1 - \dot{\theta}_1\right)/2b \tag{3.14}$$

By integrating both sides;

$$\varphi = \left(\frac{r}{2b}\right)(\theta_1 - \theta_2) + \varphi(0) \tag{3.15}$$

where $\varphi(0)$ is the initial chassis rotational angle. So φ can be expressed in terms of θ_1 and θ_2, hence the number of degrees of freedom to express the system is reduced to 4, i.e., $q = \begin{bmatrix} x & y & \theta_1 & \theta_2 \end{bmatrix}^T$.

3.5.2 Dynamic Model of the Platform

This section is devoted to obtaining the platform dynamical model with holonomic and nonholonomic constraints. The Lagrange method was used to mathematically represent the system, where the difference between the kinetic and potential energy is represented by the Lagrangian L.

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) \tag{3.16}$$

$T(q, \dot{q})$ represents the kinetic energy, while $U(q)$ is the potential energy of the platform. Based on deriving the Lagrangian for the mobile platform in Appendix B, and by using Euler–Lagrange equation, the dynamical model is obtained as;

$$M(q)\ddot{q} + D(q, \dot{q})\dot{q} + G(q) = A(q)^T\lambda + E(q)u \tag{3.17}$$

where $M(q)$ is the inertia matrix, $D(q, \dot{q})$ is the carioles matrix, $G(q)$ is the potential energy, $A(q)$ is the Pfaffian matrix, λ is the Lagrangian multiplier vector, $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ is the control torque vector, and finally $E(q)$ is matrix given by;

$$\mathrm{E}(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The calculations for the above matrices and components are found in detail in Appendix B. Additionally, the carioles matrix and the potential energy are equal to zero for the mobile platform, hence Eq. (3.17) becomes;

$$M(q)\ddot{q} = A(q)^T \lambda + E(q)u \tag{3.18}$$

In order to make the Lagrangian multiplier vector λ disappear in Eq. (3.18), a multiplication by a matrix $N(q)$ is needed, which has the following property;

$$A(q)N(q) = 0$$

i.e., the row vectors of matrix $N(q)$ lie in the kernal space of matrix $A(q)$. Accordingly, $N(q)$ takes the following form (please refer to Appendix B);

$$N(q)^T = \begin{bmatrix} r/2\cos\varphi & r/2\sin\varphi & 1 & 0 \\ r/2\cos\varphi & r/2\sin\varphi & 0 & 1 \end{bmatrix} \tag{3.19}$$

As a result, multiplying Eq. (3.18) by $N(q)^T$, yield;

$$\left.\begin{aligned} N(q)^T M(q)\ddot{q} &= N(q)^T A(q)^T \lambda + N(q)^T E(q)u \\ &= (A(q)N(q))^T \lambda + N(q)^T E(q)u = N(q)^T E(q)u \end{aligned}\right\} \tag{3.20}$$

where $N(q)^T E(q) = I_{2\times 2}$. To this end, it is needed to define the following coordinate transformation;

$$\dot{q} = N(q)\rho \tag{3.21}$$

where $\rho \in R^2$, is a new coordinate for the kernal space of $A(q)$, i. e.,

$$A(q)\dot{q} = 0 = A(q)N(q)\rho$$

The total number of states that describe the dynamics of the mobile platform is six, where there are four states ($q \in R^4$) in addition to the two states due to ρ vector. In a state vector form, the mobile platform state vector is given by;

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} q \\ \rho \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta_1 \\ \theta_2 \\ \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{3.22}$$

and the state vector derivative is;

$$\dot{\hat{x}} = \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} \dot{q} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} \tag{3.23}$$

where from Eq. (3.23), $\dot{q}$ is given by;

$$\dot{q} = \dot{\hat{x}}_1 = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = N(q)\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \tag{3.24}$$

while $\dot{\rho}$ is derived from Eqs. (3.21) and (3.20) as follows;

$$\ddot{q} = \frac{d}{dt}(N(q)\rho) = \dot{N}(q)\rho + N(q)\dot{\rho}$$

Substituting $\ddot{q}$ in Eq. (3.20), yield;

$$N(q)^T M(q)\big(\dot{N}(q)\rho + N(q)\dot{\rho}\big) = u$$

Then by solving for $\dot{\rho}$:

$$\begin{aligned} \dot{\rho} = \dot{\hat{x}}_2 = \begin{bmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} &= \Big[N(q)^T M(q) N(q)\Big]^{-1}\Big(-N(q)^T M(q)\dot{N}(q)\rho + u\Big) \\ &= -\Big[N(q)^T M(q) N(q)\Big]^{-1}\Big[N(q)^T M(q)\dot{N}(q)\Big]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + \Big[N(q)^T M(q) N(q)\Big]^{-1} u \end{aligned} \tag{3.25}$$

Eventually, the proposed mobile platform dynamical model is given by;

$$\dot{\hat{x}} = \begin{cases} N(q)\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \\ -\Big[N(q)^T M(q) N(q)\Big]^{-1}\Big[N(q)^T M(q)\dot{N}(q)\Big]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + \Big[N(q)^T M(q) N(q)\Big]^{-1}(u + T_f) \end{cases} \tag{3.26}$$

where: u represents the actuating torques applied to the platform wheels, T_f is frictional torques affecting the mobile platform at the wheels and $M(q)$ represents the mass and inertia matrix (see Appendix B).

4 Sliding Mode Control Design

In this section, the concept of SMC was presented, followed by an illustration of the attendant features of performance specification and robustness. After that the ASMC and AUSMC approaches will be introduced. The major disadvantage of this approach (i.e. the chattering behavior) and how to overcome it will be discussed. Finally, a robust controller that will achieve trajectory tracking using both ASMC and AUSMC approaches for the nonholonomic mechanical system is designed.

4.1 Sliding Mode Control (SMC)

The development of any control system comes with typical differences between the real plant and mathematical model (which the controller will be designed based on this model). These differences may be attributed to the unmodeled dynamics, changes in the system parameters, or complex plant behavior approximation by a straightforward model [39]. Engineers must ensure the resulting controller has a practical applicability to producing the intended performance levels irrespective of the discrepancies. This has increased the interest in effective control methods development that can address this issue. A major approach to the design of robust controllers is the SMC methodology.

An essential aspect of the SMC is that its control action is discontinuous in nature. Each feedback channels primarily works as a switch between two significantly different systems in a way that a new type of motion of system that is, known as a sliding mode, is established in the manifold. This specific feature is believed to enhance the system performance in terms of its insensitivity to parameter changes, as well as a complete annulling of disturbances.

The SMC is a specialized form of variable structure control where the control systems are designed to drive the state of the system and constrain it to be within the neighborhood of the sliding manifold. This approach has two major merits, firstly, the system's dynamic behavior can be designed by the special selection of its sliding variable; secondly, the closed loop system response is completely insensitive to a specific class of uncertainty. This invariance feature ascertained the appropriateness of this method for robust control. In addition, performance specification directly makes the SMC attractive from the design point of view.

In designing a sliding mode controller, there are two aspects: first, it involves selecting the sliding variable to meet the design specifications; second, the control law selection which will enhance the attractiveness of the sliding manifold to the system state.

The main benefits of the classic SMC include its convergence with respect to finite time, compensated dynamics with reduced order, and robustness [40]. There are different methodologies of the SMC approach, and these methodologies are classified according to the type of function which represents the sliding variable. This work focuses on the adaptive conventional ASMC and adaptive unity AUSMC.

4.2 Conventional SMC

The major aspect of traditional SMC theory had been concluded, but as stated later by Utkin [41]. The two-step approach towards the design of the SMC involves the sliding manifold design, and the design of the discontinuous controllers to ensure the sliding modes.

The major benefits of the SMC include a reduced order of sliding equations, an exact compensation of the insensitivity bounded with uncertainties, and a convergence to the sliding manifold at a finite time.

Meanwhile, its major drawbacks include insensitivity to only matched uncertainties, chattering, and an underperformance of the closed-loop with the existence of disturbing dynamics, noise, or discretization.

Assume a single linear input/single output dynamic system (affine nonlinear system) expressed as follows:

$$\dot{x}(t) = f(x,t) + \boldsymbol{g}(x,t)u(x,t) \tag{4.1}$$

where $x \in R^n$ is the state vector, $u \in R$ is the control input, $f(x,t)$ and $\boldsymbol{g}(x,t)$ are the vector fields.

There is a restriction in selecting the sliding variable, which it must have a relative degree one with respect to control. That is for a selected sliding variable $s(x)$, the control input u must appears after the first differentiation of $s(x)$, i.e.,

$$\dot{s} = \dot{s}(t,x,u) \tag{4.2}$$

Now let the control variable $u(x,t)$ to have undergone a discontinuity motion on the sliding surface $(s(x) = 0)$;

$$u(x,t) = \begin{cases} u^+, (x,t)\ s(x) < 0 \\ u^-, (x,t)\ s(x) > 0 \end{cases} \tag{4.3}$$

Definition 4.1 ([42]) Equations 4.2 and 4.3 defined the system called a variable structure system. If the convergence condition $\dot{s}(x)s(x) < 0$ is satisfied by the control variable $u(x,t)$, it will further satisfy the control and said to be in a sliding mode.

An ideal sliding mode is achieved when $s(x) = 0$ and $\dot{s}(x) = 0$. Furthermore, a sliding mode is real when only the convergence criteria is verified, i.e. $\dot{s}(x)s(x) < 0$ [43]. This is depicted in Fig. 6, and the motion is therefore said to be in a real sliding mode.

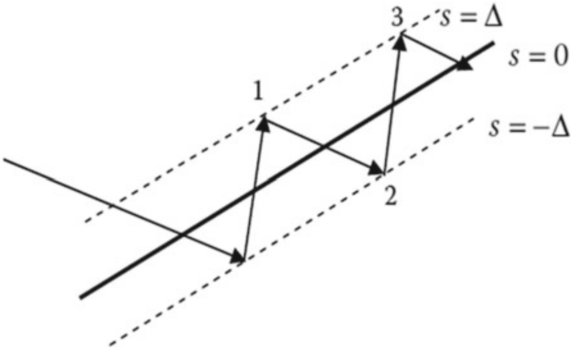


Fig. 6 Oscillations around the switching surface ($s = 0$)

4.2.1 Conditions for Sliding Mode Stability

The sliding mode existence conditions can be derived by following the stability analysis in nonlinear systems, and in trying to establish a Lyapunov function. Similarly, it should be remembered that the motion's right hand side equation is non-continuous, and is not defined in the points where the signum functions is equal to zero [43].

To this end let the candidate Lyapunov function be selected as a positive definite function of the sliding variable s:

$$V(s) = \frac{1}{2}s^2 \tag{4.4}$$

And $\dot{V}(s)$ is obtained as:

$$\dot{V}(s) = s\dot{s} \tag{4.5}$$

Therefore, the sliding manifold ($s = 0$) is attractive iff $\dot{V}(s) < 0, \forall s \neq 0$ [44]. This means that:

$$s\dot{s} < 0 \tag{4.6}$$

is the stability condition which ensure that the state directed towards the sliding manifold and maintained at this location for all future time (Fig. 7). Condition (4.6) is also considered as the stability condition for the SMC system.

Remark 4.1 The stability condition in inequality (4.6) is derived for a system with one control input and consequently one sliding variable. In the case of multi-input control system the stability condition (sliding condition) is derived below via unit sliding mode.

Remark 4.2 The sliding manifold for the case of n-input control system is the intersection of n-hypersurfaces, where each hypersurface defined by $s_i = 0, i = 1, 2, \ldots, n$. For $n = 2$ the sliding manifold is depicted in Fig. 8.

Fig. 7 Sliding manifold

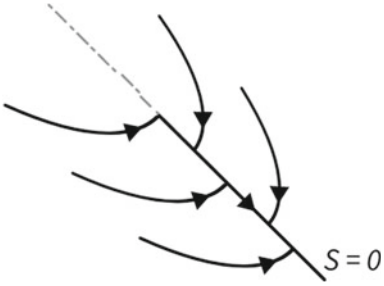
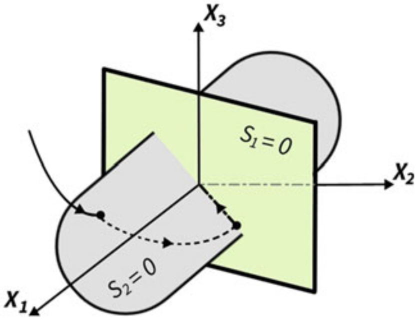


Fig. 8 Two-dimensional sliding mode

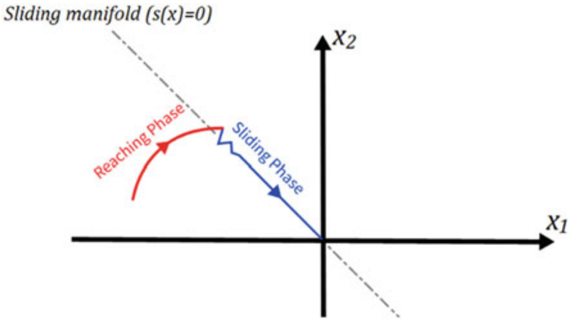


4.2.2 Reaching Phase and Sliding Phase

There are two parts of the state trajectory modes, one is the reaching phase, which is the point from the primary (initial) state to the meeting point with the sliding surface, and the other is the sliding phase, which is the point from the point of meeting with the sliding surface back to the initial point. Both phases are depicted in Fig. 9 in terms of a second-order system, where the convergence condition $\dot{s}(x)s(x) < 0$ during the reaching phase and sliding phase is satisfied [43].

Within the sliding phase, the closed loop system is a linear state function, will have a similar behavior to a linear system with a reduced or free order. For an n

Fig. 9 Illustration of reaching and sliding modes for second-order system example



system order the system dynamics will be reduced to $(n - m)$ system order where m represents the number of control input u.

4.3 Unit Sliding Mode Control

In this section, designing a non-continuous control complying sliding mode in selected manifolds is introduced without the need for individually selecting each control component as a non-continuous function. This way comprises of designing the control dependent on a Lyapunov function chosen for a nominal system. The control will be found in a way that the Lyapunov function will have negative time derivative along the system trajectories, with disturbances caused by uncertainties inherent in the environment and in the plant model. This approach is based on the evidence presented by Gutman and Leitmann (1970) [45, 46]. The design concept for an affine system may be explained as:

$$\dot{x} = f(x, t) + B(x, t)u + h(x, t) \tag{4.7}$$

where $x \in R^n$ and $u \in R^m$ are the state and control vectors respectively, $f(x, t)$ and $h(x, t)$ are the state-dependent vectors, and $B(x, t) \in R^{n \times m}$ is the control input matrix. The vector $h(x, t)$ is the perturbation term whose effect on the control process must be eliminated. The equation

$$\dot{x} = f(x, t) \tag{4.8}$$

depicts a presumed asymptomatically stable open-loop nominal system with a defined Lyapunov function [39].

$$V(x) > 0$$

$$W_o = \left.\frac{dV}{dt}\right|_{h=0,u=0} = grad(V)^T f(x, t) < 0,$$

$$grad(V)^T = \left[\frac{\partial V}{\partial x_1} \cdots \frac{\partial V}{\partial x_n}\right] \tag{4.9}$$

Let the perturbation vector $h(x, t)$, satisfies the matching condition.

$$h(x, t) \in span(B(x, t)) \tag{4.10}$$

then there exist a vector $\gamma(x, t) \in R^m$, such that

$$h(x, t) = B(x, t)\gamma(x, t) \tag{4.11}$$

$\gamma(x, t)$ may be regarded as an unknown vector with a known upper bound $\gamma_o(x, t)$; such that

$$\|\gamma(x, t)\| < \gamma_o(x, t) \tag{4.12}$$

The $\dot{V}(x)$ along the perturbed system trajectories (Eq. 4.7) can be calculated as

$$W = dV/dt = W_o + \boldsymbol{grad}(V)^T B(u + \gamma) \tag{4.13}$$

Control u is selected based on the upper limit of the unknown disturbance as

$$u = -\psi(x, t)\frac{B^T \boldsymbol{grad}(V)}{\|B^T \boldsymbol{grad}(V)\|} \tag{4.14}$$

with a scalar function $\psi(x, t) > \gamma_o(x, t)$. Since $\psi(x, t)$ represent the (USMC) gain, and

$$\|B^T \boldsymbol{grad}(V)^T\|^2 = (\boldsymbol{grad}(V)^T B)(B^T \boldsymbol{grad}(V)$$

then Eq. (4.13) becomes

$$\begin{aligned} W &= W_o - \psi(x, t)\|B^T \boldsymbol{grad}(V)\| + \boldsymbol{grad}(V)^T B\gamma(x, t) \\ &< W_\circ - \|B^T \boldsymbol{grad}(V)\|[\psi(x, t) - \gamma_\circ(x, t)] \\ &< W_o - \|B^T \boldsymbol{grad}(V)\|(\psi(x, t) - \gamma(x, t)) < 0 \end{aligned}$$

In the USMC, $s(x) = B^T \boldsymbol{grad}(V)$ is the variable which will have a negative time derivative, suggesting the asymptomatic stability of the perturbed system with control.

For the system with control, two basic features should be considered;

1. The control law (Eq. 4.14) in $(n - m)$ dimensional manifold $s(x) = B^T \boldsymbol{grad}(V) = 0$ undergoes discontinuities, and beyond this manifold, it is a continuous state function. This distinguished the control law as given in Eq. (4.14) from all the control inputs in the SMC where each control component has individual sliding variable.
2. Because of the enforcement of the sliding motion (SM) at the sliding manifold $s(x) = 0$, the disturbance $h(x, t)$ is rejected. The disturbance (Eq. 4.11) is rejected, control u should then be equal to $-\gamma(x, t)$ and this is not always the case when $s(x) = B^T \boldsymbol{grad}(V) \neq 0$, i.e., the state outside the sliding manifold. This implies that a sliding mode exists in the manifold $s = 0$, and the corresponding value of the equivalent control equal to $-\gamma(x, t)$.

The norm of control with the gain $\psi(x, t) = 1$ for any value of the state vector is given by;

$$\|u\|_{\psi(x,t)=1} = \|\frac{B^T grad(v)}{\|B^T grad(v)\|}\|$$

This provides an explanation of the term "Unit control" for the control (Eq. 4.14).

4.4 ASMC

The application of the SMC is mainly faced with the problem of two inter-related incidents: these are chattering and high activity of control action. The chattering amplitude is proportional to the discontinuous control magnitude. These problems are solved simultaneously if there is a reduction in the magnitude to a lowest level determined by the conditions necessary for SM to exist. An adaptation method is mentioned here for achieving the minimum possible control value based on two recently developed approaches [47]. Meanwhile, its design demands the understanding of the bounds on the uncertainties which could be conservatively estimated. In other words, the bound is often overestimated and usually result in an excessive gain. The chattering phenomenon, a well-known problem of SMC (see [48] for its analysis), is important and could cause damage to the systems and the actuators. Chattering can be minimized by using the boundary layer approach. In this case, many ways have suggested adequate controller gains tuning [49]. Another way of minimizing chattering phenomenon is by using a higher order SMC [50–55]. In both control approaches, there is a need to know the bound on the uncertainties. Being that the non-requirement of the uncertainties bound is the main objective, another way involves the use of ASMC with the aim of ensuring obtain as small as possible control gain but still enough to annul the perturbations and uncertainties.

In this section, the adaptation method to find the discontinuous control gain $k(t)$ that will offer a minimal discontinuity value which will result in chattering effect reduction. Gain dynamics is mainly dependent on the tracking error; since the sliding mode is not established, the control gain keeps increasing and, in this situation, until the state hits the boundary layer for the sliding manifold. After that, $k(t)$ will decrease until it reaches a certain minimal value. The adaptation law for the $k(t)$ is presented blow.

4.4.1 The ASMC Law

In this section, an approach ensures the presence of only a real SM is presented as follows; Let ASMC be assumed as follow

$$u(t,s) = -K(t)\, sign(s(x(t),t)) \tag{4.15}$$

with the gain coefficient $K(t)$ satisfies [47]

$$\dot{K} = \begin{cases} \alpha|s(x(t), t)|\, sign(|s(x(t), t)| - \varepsilon) & \text{if } K > \mu \\ 0 & \text{if } K \leq \mu \end{cases} \tag{4.16}$$

where $s(x(t), t)$ is the sliding variable with $\alpha > 0$, $\varepsilon > 0$ and a positive small value of μ, which it is incorporated to maintain K positive value. Once there is an establishment of a SM with respect to $s(t, x(t))$, the suggested gain adaptation law (4.16) declines the gain K while $|s(x(t), t)| < \varepsilon$. Similarly, gain K will be maintained at the least level that will allow the stabilization of a given accuracy of s. That means an adequate gain is allowed to obtained with respect to the magnitude of the uncertainties and perturbations based on this adaptation law.

Eventually, the state will be maintained in a positively invariant set as can be seen in the following; First, the sliding variable dynamic from Eqs. (4.1) and (4.2) is given by

$$\dot{s} = \frac{\partial s}{\partial x}\dot{x}(t) = \frac{\partial s}{\partial x}\{f(x, t) + \boldsymbol{g}(x, t)u(x, t)\} = \Psi(x, t) + \Gamma(x, t)u \tag{4.17}$$

where $\Psi(x, t) = \frac{\partial s}{\partial x} f(x, t)$, and $\Gamma(x, t) = \frac{\partial s}{\partial x}\boldsymbol{g}(x, t)$. Both functions $\Psi(x, t)$ and $\Gamma(x, t)$ are supposed assumed bounded for $x \in R^n$ and all $t \geq 0$.

$$\Psi(x, t) \leq \Psi_M, 0 < \Gamma(x, t) \leq \Gamma_M$$

Theorem 4.1 ([56]) *Given the nonlinear uncertain system* (4.1) *with the sliding variable* $s(t, x(t))$ *dynamics* (4.17) *controlled by* (4.15) *and* (4.16)*, then a finite time* t_F *exist such that a real sliding mode is established for all* $t > t_F$*, i.e.,*

$$|s(x(t), t)| < \delta_s$$

for all $t > t_F$ *with*

$$\delta_s = \sqrt{\varepsilon^2 + \frac{\Psi_M^2}{\alpha\Gamma_m}}$$

Proof [56].

Remark 4.3 According to the above theorem, the state is eventually converged in a finite time to the following positively invariant set

$$\left\{ x \in R^n \,\middle|\, |s(x(t), t)| < \sqrt{\varepsilon^2 + \frac{\Psi_M^2}{\alpha\Gamma_m}} \right\} \tag{4.18}$$

Moreover, since the selection of the ensures the asymptotic convergence to the origin, so by using the ASMC, the state convergences to a positively invariant set function to δ [56].

4.5 The Chattering Problem

Chattering is an incident of oscillations with a finite frequency and amplitude that appears in many sliding mode implementations. A high-frequency switching of the SM controller which excites the unmolded dynamics in a closed loop is the cause of these oscillations. The unmolded dynamics may result from actuators and sensors which were ignored during the fundamental process of modeling due to their significantly faster dynamics compared to that of the main system. Meanwhile, there is a need to consider all system dynamics in an ideal sliding mode system since all its dynamics are infinitely fast. Interestingly, the control of chattering doesn't usually require intensive modeling of the complete system components, but instead, an SMC can first be designed under idealized presumptions that there is no unmolded dynamic. In the next design step, potential chattering can be avoided using any of the methods presented below. Finding a solution to the chattering phenomenon is essential, especially when exploiting the advantages of an SMC in a real-life system. Note that switching in itself as a major component of the continuous-time SMC, isn't referred to as chattering due to that, ideally, switching is an intended process with its frequency tending to infinity, but chattering is an unwanted system oscillation with finite frequency, which is caused by imperfections in the system [42].

- ***Signum function approximation***

An obvious reason for making the control function smooth or continuous, the non-continuous function $sign(\sigma)$ is approximated to some continuous smooth functions. It could, for instance, be replaced with a sigmoid function

$$sign(\sigma) \approx \frac{\sigma}{|\sigma| + \varepsilon} \tag{4.19}$$

where $\lim\limits_{\varepsilon \to 0} \frac{\sigma}{|\sigma|+\varepsilon} = sign(\sigma)$ and $\in$ represents positive scalar of a small value. It can point-wise be observed that for $\sigma \neq 0$, ε value should be chosen to trade-off the criteria for maintaining an ideal system performance and ensure a smooth control action.

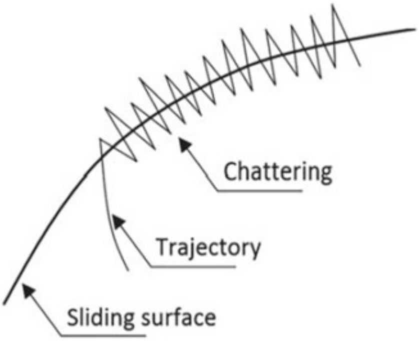


Fig. 10 The chattering phenomena

- ***Boundary-Layer Approximation***

A boundary layer approximation of discontinuous control can be used as an additional approximation function for the reduction of the chattering phenomenon [48, 57] around the sliding surface. The idea of this approach is illustrated by using continuous functions like hyperbolic tangent (*tan*), saturation (*sat*), or inverse tangent (*arctan*) functions rather than using discontinuous functions in an SMC law to annul unwanted chattering effects. The sat function and the $arctan$ function are defined in Eqs. (4.20) and (4.21) respectively;

$$sat(\sigma, \varepsilon) = \begin{cases} sign(\sigma) & \text{if} |\sigma| > \varepsilon \\ \frac{\sigma}{\varepsilon} & \text{if} |\sigma| \leq \varepsilon \end{cases} \tag{4.20}$$

$$arctan(\sigma, \rho) = \frac{2}{\pi} \tan^{-1}(\rho\sigma),\ \rho \geq 1 \tag{4.21}$$

where $\in > 0$, $\rho \geq 1$ that defines the width of the boundary layer.

4.6 Coupling and Decoupling

Coupling is a big challenge in control theory in the case of multi–input–multi–output which is the case of our application in this work. Mechanical systems often consist of a set of second-order differential equations in the case of multi-input, where the number of controllers is greater than one. Actually, these controllers $(u_1, \ldots, u_n)$ may appear together in every channel and this creates the coupling situation. In case the control matrix B (or $\boldsymbol{g}(x)$ for nonlinear systems) is certain, the problem is solved and the system will be under a decoupling situation, but the challenge of coupling appears if B (or $\boldsymbol{g}(x)$) is uncertain. There are many methods for solving this problem, and these methods have been proven mathematically. These are the control hierarchy, the unity siding mode control approach, and the component-wise method as illustrated in [41]. The drawbacks of this method are that the obtained value of control gain is high, and the chattering problem arises again.

4.7 Control Design of Nonholonomic Mobile Robot

4.7.1 Partial Feedback Linearization

Feedback linearization is an approach for the transformation of nonlinear systems to their equivalent linear systems [44] When an input/output linearization map is made, the state equation is partially linearized by introducing the concepts of relative degree and zero dynamics with a minimum phase system. The desired system output y_d dynamics may not contain all the state variables, where the linearized feedback control has made some of the system state to be unobservable from the output function. When designing the tracking control (the objective of this chapter), we should ensure that the state variable which is not observable by y_d is well-behaved and bounded Relative degree is the order of differential equation until the control element u is achieved in the result. In this work, we have designed a trajectory tracking controller, and our objective is to enforce the output function, the point pc on the mobile platform, to follow a certain desired trajectory with guaranteed minimum-phase property [58].

As a first step the location of the point (pc) is obtained as follows;

$$T_{PC}^{\circ} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & x \\ \sin\varphi & \cos\varphi & 0 & y \\ 0 & O & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & O & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_{Po}^{\circ} T_{PC}^{po}$$

$$= \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & x + d\cos\varphi \\ \sin\varphi & \cos\varphi & 0 & y + d\sin\varphi \\ 0 & O & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow P_{pc}^{0} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & x + d\cos\varphi \\ \sin\varphi & \cos\varphi & 0 & y + d\sin\varphi \\ 0 & O & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \begin{bmatrix} x + d\cos\varphi \\ y + d\sin\varphi \\ 0 \\ 1 \end{bmatrix}$$

where T_{PC}° is the homogeneous transformation matrix. From P_{pc}^{0} the required outputs are

$$\left.\begin{array}{l} y_1 = p_{pcx}^{0} = x + d\cos(\varphi) \\ y_2 = p_{pcy}^{0} = y + d\sin(\varphi) \end{array}\right\} \tag{4.22}$$

And the time derivative of outputs y_1 and y_2 are;

$$\left.\begin{array}{l} \dot{y}_1 = \dot{x} - d\sin(\varphi) * \dot{\varphi} \\ \quad = \left[\, (r/2)\cos\varphi - cd\sin\varphi \;\; (r/2)\cos\varphi + cd\sin\varphi \,\right] \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \end{array}\right\} \tag{4.23}$$

$$\left.\begin{array}{l} \dot{y}_2 = \dot{y} + d\cos(\varphi) * \dot{\varphi} \\ \quad = \left[\, (r/2)\sin\varphi + cd\cos\varphi \;\; (r/2)\sin\varphi - cd\cos\varphi \,\right] \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \end{array}\right\} \tag{4.24}$$

where $\dot{x}$, $\dot{y}$ and $\dot{\varphi}$ are (see Eqs. 3.14 and 3.24)

$$\left.\begin{aligned}\dot{x} &= \left[\,(r/2)\cos\varphi \;\; (r/2)\cos\varphi\,\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \\ \dot{y} &= \left[\,(r/2)\sin\varphi \;\; (r/2)\sin\varphi\,\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \\ \dot{\varphi} &= c\big(\dot{\theta}_1 - \dot{\theta}_2\big) = \left[\,c \;\; -c\,\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix}\end{aligned}\right\} \tag{4.25}$$

Finally, $\ddot{y}_1$ and $\ddot{y}_2$ are obtained as

$$\left.\begin{aligned}\ddot{y}_1 &= f_{y1}(x) + \mathrm{g}_{\mathrm{y}1}(x)u \\ \ddot{y}_2 &= f_{y2}(x) + \mathrm{g}_{\mathrm{y}2}(x)u\end{aligned}\right\} \tag{4.26}$$

where;

$$\begin{aligned}f_{y1}(x) = {} & \left[\,-(r/2)\sin(\varphi)\dot{\varphi} - cd\cos(\varphi)\dot{\varphi} \;\; -(r/2)\sin(\varphi)\dot{\varphi} + cd\cos(\varphi)\dot{\varphi}\,\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \\ & + \left[\,(r/2)\cos(\varphi) - cd\sin(\varphi) \;\; r/2\cos(\varphi) + cd\sin(\varphi)\,\right] \\ & * \left[\left[N^T(q)M(q)N(q)\right]^{-1}\left[-N^T(q)M(q)\dot{N}(q)\begin{bmatrix} x_5 \\ x_6 \end{bmatrix}\right]\right]\end{aligned}$$

$$\begin{aligned}f_{y2}(x) = {} & \left[\,(r/2)\cos(\varphi)\dot{\varphi} \;\; (r/2)\cos(\varphi)\dot{\varphi}\,\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} - d\sin(\varphi)\dot{\varphi}^2 \\ & + \left[\,\left(\tfrac{r}{2}\right)\sin(\varphi) + cd\cos(\varphi) \;\; \left(\tfrac{r}{2}\right)\sin(\varphi) - cd\cos(\varphi)\,\right] \\ & * \left[\left[N^T(q)M(q)N(q)\right]^{-1}\left[-N^T(q)M(q)\dot{N}(q)\begin{bmatrix} x_5 \\ x_6 \end{bmatrix}\right]\right]\end{aligned}$$

$$\mathrm{g}_{\mathrm{y}1}(x) = \left[\,(\mathrm{r}/2)\cos(\varphi) - \mathrm{cd}\sin(\varphi) \;\; \tfrac{r}{2}\cos(\varphi) + \mathrm{cd}\sin(\varphi)\,\right]\left[N^T(q)M(q)N(q)\right]^{-1}$$

$$\mathrm{g}_{\mathrm{y}2}(x) = \left[\,(\mathrm{r}/2)\cos(\varphi) + \mathrm{cd}\sin(\varphi) \;\; \tfrac{r}{2}\sin(\varphi) - \mathrm{cd}\cos(\varphi)\,\right]\left[N^T(q)M(q)N(q)\right]^{-1}$$

Now define the error functions $e_1 = y_1 - y_{1d}$ and $e_2 = y_2 - y_{2d}$. Where y_{1d} and y_{2d} represent the desired position for y_1 and y_2 respectively. Accordingly;

$$\left.\begin{aligned}\dot{e}_1 &= \left[\,(r/2)\cos\varphi - \mathrm{cd}\sin\varphi \;\; (r/2)\cos\varphi + \mathrm{cd}\sin\varphi\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} - \dot{y}_{1d} \\ \dot{e}_2 &= \left[\,(r/2)\sin\varphi + \mathrm{cd}\cos\varphi \;\; (r/2)\sin\varphi - \mathrm{cd}\cos\varphi\right]\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} - \dot{y}_{2d}\end{aligned}\right\} \tag{4.27}$$

and $\ddot{e}_1$ and $\ddot{e}_2$ from Eqs. (4.26) and (4.27) are

$$\left.\begin{aligned}\ddot{e}_1 &= F_1(x) + G_1(x)u \\ \ddot{e}_2 &= F_2(x) + G_2(x)u\end{aligned}\right\} \tag{4.28}$$

where: $F_1(x) = f_{y1}(x) - \ddot{y}_{1d}$, $F_2(x) = f_{y2}(x) - \ddot{y}_{2d}$, $G_1(x) = \mathrm{g}_{y1}(x)$ and $G_2(x) = \mathrm{g}_{y2}(x)$. Eventually, (4.28) in a vector form is given by;

$$\ddot{e} = \begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} = \begin{bmatrix} F_1(x) \\ F_2(x) \end{bmatrix} + \begin{bmatrix} G_1(x) \\ G_2(x) \end{bmatrix} u = F(x) + G(x)u \tag{4.29}$$

The error function dynamics can also be putted considering the uncertainty in the system model as

$$\ddot{e} = F_o(x) + G_o(x)u + \delta(x, u) \tag{4.30}$$

where $F_o(x)$ and $G_o(x)$ are the nominal $F(x)$ and $G(x)$ i.e., considering the nominal parameters value, while $\delta(x, u)$ is the perturbation term. Mathematically $\delta(x, u)$ is the difference between the right-hand side of Eq. (4.29) and $F_o(x) + G_o(x)u$, i.e.,

$$\begin{aligned} \delta(x, u) &= F_y(x) + G_y(x)u + F_r - \big(F_{yo}(x) + G_{yo}(x)u\big) \\ &= \Delta F(x) + \Delta G(x)u + F_r \end{aligned} \tag{4.31}$$

where

$$F_r = -N^T * [F_v * \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + F_s * sign\begin{bmatrix} x_5 \\ x_6 \end{bmatrix}]$$

Here F_v and F_s are viscous and Coulomb friction coefficients matrices respectively, while the $sign\begin{bmatrix} x_5 \\ x_6 \end{bmatrix}$ function is defined by;

$$sign\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} sign(x_5) \\ sign(x_6) \end{bmatrix}$$

In the next section, a robust nonlinear controller utilizing the ASMC theory is designed to make the mobile platform follow the desired trajectory in spite of system dynamic model uncertainty (the perturbation terms $\delta(x, u)$).

4.7.2 Sliding Variables Selection and Nominal Control Design

The selection of the sliding variables is the first step in the SMC design. In this work, the switching function vector $s = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T$ is derived as follows;

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \Lambda \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \dot{e} + \Lambda e \tag{4.32}$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \lambda_1, \lambda_2 > 0$$

Also, $\dot{s}$ is obtained as;

$$\begin{aligned} \dot{s} = \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} + \Lambda \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \ddot{e} + \Lambda\dot{e} \\ = F_o(x) + G_o(x)u + \delta(x, u) + \Lambda\dot{e} \end{aligned} \tag{4.33}$$

Let the control input vector be given as

$$u = G_o^{-1}(x)(u_o + u_s), |G_o(x)| \neq 0 \forall x \tag{4.34}$$

where u_o is the nominal control vector, while u_s is the discontinuous control vector term. Accordingly $\dot{s}$ becomes;

$$\dot{s} = F_o(x) + u_o + u_s + \delta(x, u) + \Lambda\dot{e}$$

The nominal control u_o is selected here as

$$u_o = -F_o(x) - \Lambda\dot{e} \tag{4.35}$$

So $\dot{s}$ in terms of discontinuous control vector becomes

$$\dot{s} = u_s + \delta(x, u) \tag{4.36}$$

Next, the discontinuous control u_s was designed based on the classical ASMC and then according to the AUSMC. For both design approaches, the designed discontinuous controller will cancel the effect of $\delta(x, u)$, and hence the total control u enforces the output y to follow the desired trajectory.

4.7.3 Classical ACSMC Law with a Friction Force

The adaptation law suggested in this study tunes the parameter of control gain of the first and second discontinuous controller functions (k_1, k_2), which will attenuate the chattering effect. The ASMC uses three conditions for limiting the adaptive gain at the minimum required value with respect to uncertainty and perturbation magnitude (Eq. 4.31). As mentioned in Sect. 4.4.1, the discontinuous control term and the adaptation law are given here as follows;

$$\left.\begin{array}{c} u_{si} = -k_i(t)sign(s_i)i = 1, 2 \\ \dot{\mu}_i(t) = \eta_i * |s_i| * sign(|s_i| - \varepsilon_i) \\ k_i(t) = \begin{pmatrix} \mu_i & if K_{imin} < \mu_i < K_{imax} \\ K_{imin} & if \mu_i \le K_{imin} \\ K_{imax} & if \mu_i \ge K_{imax} \end{pmatrix} \end{array}\right\} \tag{4.37}$$

where ε_i is a small positive constant, $\eta_i > 0, \varepsilon_i > 0$, and K_{min} is the minimum possible value of $K(t)$, while K_{max} is the maximum possible value of $K(t)$.

To this end, the steady-state error for the error functions can be estimated according to Remark 4.3 by the following inequalities;

$$|e_i| \le \frac{\delta_{si}}{\lambda_i} = \frac{1}{\lambda_i}\left(\varepsilon^2 + \frac{\Psi_M^2}{\alpha\Gamma_m}\right)^{1/2}$$

So,

$$|e| \le \left(\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2}\right)^{1/2}\left(\varepsilon^2 + \frac{\Psi_M^2}{\alpha\Gamma_m}\right)^{1/4} \tag{4.38}$$

4.7.4 Design of Adaptive Unity Sliding Mode Control (AUSMC)

The adaptive method was applied here to aid in the reduction of the chattering phenomena by tuning the gain control associated with the bounded perturbation. The presence of a boundary layer helps to reduce the chattering phenomena. To design a USMC, a sort of linearization is recommended and that to design a linear controller for the nominal part of the system. This is done via rewriting the error dynamics in terms of a new state variable $z = [z_1 z_2 z_3 z_4]^T$. The error system dynamics, which are described in Eq. (4.28) are rewritten as;

$$\left.\begin{array}{l} \dot{z}_1 = z_3 \\ \dot{z}_2 = z_4 \\ \dot{z}_3 = F_1(x) + G_1(x)u \\ \dot{z}_4 = F_2(x) + G_2(x)u \end{array}\right\} \tag{4.39}$$

where $z_1 = e_1 = y_1 - y_{d1}$, $z_2 = e_2 = y_2 - y_{d2}$, and $F_1(x)$, $F_2(x)$, $G_1(x)$ and $G_2(x)$ are as defined in the previous section. In a matrix form, Eq. (4.39) can be written as;

$$\dot{z} = \widehat{F}(x) + \widehat{G}(x)u \tag{4.40}$$

where

$$\widehat{F}(x) = \begin{bmatrix} z_3 \\ z_4 \\ F(x) \end{bmatrix} = Az + \begin{bmatrix} 0_{2\times 1} \\ F(x) \end{bmatrix}, \text{ and } \widehat{G}(x) = \begin{bmatrix} 0_{2\times 2} \\ G(x) \end{bmatrix}$$

and $F(x)$ and $G(x)$ are given in Eq. (4.29), while A is a 4×4 matrix given by;

$$A = \begin{bmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix}$$

Additionally, the error dynamics (Eq. 4.40) can also be given in terms of nominal and perturbation terms as follows;

$$\dot{z} = \widehat{F}_o(x) + \widehat{G}_o(x)u + D\delta \tag{4.41}$$

where

$$\widehat{F}_o(x) = Az + \begin{bmatrix} 0_{2\times 1} \\ F_o(x) \end{bmatrix}, \widehat{G}_o(x) = \begin{bmatrix} 0_{2\times 2} \\ G_o(x) \end{bmatrix} \text{and } D = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix}$$

Here $I_{2\times 2}$ refers to a 2×2 identity matrix, while $0_{l\times p}$ is the zero matrices of size $\times p$. To this end, let the control$u = G_o^{-1}(x)(u_o + u_s)$, where u_o is nominal control and u_s is the discontinuous unity control. Then Eq. (4.41) becomes;

$$\left.\begin{aligned} \dot{z} &= \widehat{F}_o(x) + \widehat{G}_o(x)G_o^{-1}(x)(u_o + u_s) + D\delta \\ &= Az + \begin{bmatrix} 0_{2\times 1} \\ F_o(x) \end{bmatrix} + B(u_o + u_s) + D\delta \end{aligned}\right\} \tag{4.42}$$

where $B = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix} = D$. This shows that the matching condition is satisfied.

Now let the nominal control term u_o be selected as;

$$u_o = -F_o(x) - Kz \tag{4.43}$$

Then the Eq. (4.42) becomes as below

$$\dot{z} = (A - BK)z + B(u_s + \delta) \tag{4.44}$$

where K is the gain of a linear state feedback term, which is selected such that the matrix $(A - BK)$ is Hurwitz.

In the next step in USMC design, the following Lyapunov function $V(z)$ is candidate;

$$V = z^T P z \tag{4.45}$$

Since the matrix $(A - BK)$ is Hurwitz, the matrix P is a positive definite matrix. The matrix P it is obtained by solving the following Lyapunov equation [59];

$$P(A - BK) + (A - BK)^T P = -I_{4\times 4}$$

From Eq. (4.45), the time derivative of V is:

$$\dot{V} = \dot{z^T} pz + z^T p\dot{z}$$

Since $\dot{z^T} pz = (pz)^T \dot{z}$, $(pz)^T \dot{z} = z^T p\dot{z}$ and p is a symmetric positive definite matrix, then

$$\dot{V} = z^T p\dot{z} + z^T p\dot{z} = 2z^T p\dot{z} = \nabla V^T \dot{z} \tag{4.46}$$

where

$$\nabla V^T = 2z^T p$$

$$\begin{aligned}\dot{V} &= \nabla V^T \dot{z} = \nabla V^T \{(A - BK)z + B(u_s + \delta)\}\\ &= \nabla V^T (A - BK)z + \nabla V^T B(u_s + \delta)\\ &= -\|z\|_2^2 + \nabla V^T B(u_s + \delta)\end{aligned} \tag{4.47}$$

where

$$\nabla V^T (A - BK)z = 2z^T P(A - BK)z = -z^T Iz = -\|z\|_2^2 \leq 0$$

Hence $\dot{V}$ will satisfy the following inequality;

$$\dot{V} = -\|z\|^2 + \nabla V^T B(u_s + \delta) < \nabla V^T B(u_s + \delta) \tag{4.48}$$

Consider the following discontinuous unity control vector

$$u_s = -\gamma \frac{S}{\|S\|} \tag{4.49}$$

where $S = B^T \nabla V$ is the sliding vector. Substitute the selected u_s (Eq. 4.49) in the inequality (4.48) yield

$$\left.\begin{aligned}\dot{V} &< \nabla V^T B\left(-\gamma \frac{S}{\|S\|} + \delta\right) = S^T\left(-\gamma \frac{S}{\|S\|} + \delta\right)\\ &\leq -\gamma\|S\| + \|S\|\|\delta\| = -\|S\|(\gamma - \|\delta\|)\end{aligned}\right\} \tag{4.50}$$

In classical USMC design, the control gain γ is selected such that the inequality (4.50) is satisfied i.e., $-\gamma \|S\| + \|S\| \|\delta\| < 0$. However, the chattering is again induced due to the discontinuity in control law and with large amplitude if γ is selected as a large value. Additionally determining γ required a known bound on system model uncertainty. As mentioned in this work, the ASMC can be used to calculate the discontinuous control gain γ without knowing the bound on system uncertainty. As a result, the chattering is attenuated since γ will be near the minimum value required to induce sliding mode motion.

In this work the gain γ is determined according to the following proposed adaptation law;

$$\dot{\eta} = \xi * max(V(z,t), \beta) * sign(V(z,t) - \varepsilon) \tag{4.51}$$

where ξ and ε are positive constants. As for the case of ASMC, in the previous subsection, the gain γ is determined according to the following rules;

$$\gamma = \begin{pmatrix} \eta & \text{if } \Gamma_{min} < \eta < \Gamma_{max} \\ \Gamma_{min} & \text{if } \eta \leq \Gamma_{min} \\ \Gamma_{max} & \text{if } \eta \geq \Gamma_{max} \end{pmatrix} \tag{4.52}$$

where Γ_{max} and Γ_{min} are the maximum and minimum gain values, respectively. Eventually, the proposed control law for the mobile platform is given by

$$u = G_o^{-1}(x)\left(-F_o(x) - Kz + -\gamma \frac{S}{\|S\|}\right) \tag{4.53}$$

Additionally, the use of the proposed ASMC for the unit SMC will induce a positively invariant as stated in Theorem 4.1 [56] given by

$$\left\{ x \in R^n \middle| V < \sqrt{\varepsilon^2 + \frac{\Psi_{VM}^2}{\xi \Gamma_{Vm}}} \right\} \tag{4.54}$$

For the unit SMC case, Ψ_{VM} and Γ_{Vm} are determined from Eqs. (4.40) and (4.46) as follows;

$$\dot{V} = \nabla V^T \dot{z} = \nabla V^T \widehat{F}(x) + \nabla V^T \widehat{G}(x) u = \Psi_V + \Gamma_V u \tag{4.55}$$

where $\Psi_V = \nabla V^T \widehat{F}(x)$, and $\Gamma_V = \nabla V^T \widehat{G}(x)$. Both functions Ψ_V and Γ_V are supposed assumed bounded for $x \in R^n$ and all $t \geq 0$.

$$\Psi_V \leq \Psi_{VM}, 0 < \Gamma_{Vm} \leq \Gamma_V \leq \Gamma_{VM}$$

So, after a finite time, the error state will stay within the following set;

$$|e| \leq \frac{1}{\lambda_{min}(P)}\left(\varepsilon^2 + \frac{\Psi_{VM}^2}{\xi\Gamma_{Vm}}\right)^{1/4} \tag{4.56}$$

5 Simulation Results

This section introduces the simulation results for the mobile platform dynamical system, using two control systems which were designed based on the ASMC. The mobile platform dynamics are described by the nonlinear model given in Eq. (3.26), while the proposed controller; ACSMS and AUSMC are as given in Sect. 4.7. The task of the derived controllers is to enforce the platform trajectory, through the point P_c, to follow three different trajectory shapes. The desired shapes are the infinity shape, the circular shape, and the fixed point shape. They are given by;

- ***Infinity shape***:

$$\left.\begin{array}{r} x_d = y_{1d} = a * \sin(t) \\ y_d = y_{2d} = a * \sin(t) * \cos(t) \end{array}\right\} \tag{5.1}$$

- ***Circular shape:***

$$\left.\begin{array}{r} x_d = y_{1d} = 5 * \cos(t) \\ y_d = y_{2d} = 5 * \sin(t) \end{array}\right\} \tag{5.2}$$

- ***Fixed point shape***:

$$\left.\begin{array}{r} x_d = y_{1d} = 5 \\ y_d = y_{2d} = 6 \end{array}\right\} \tag{5.3}$$

Finally, Table 1 presents the nominal mobile platform parameters, while Table 2 gives the parameters with 100% uncertainty and also the external friction torques.

5.1 Simulation Results Based on ACSMC

This section presents the simulation results for the mobile platform trajectory tracking control system based on the ACSMC proposed in Sect. 4. The mobile platform parameters which were used in the simulations are presented in Table 2, while the proposed control parameters which were derived in Eqs. (4.37) and (4.38) with signum approximation function in Eq. (4.19) are given in Table 3. Three different desired trajectories are used in the present work to demonstrate the ability of the designed controller in forcing the mobile platform to follow them through P_c which

Table 1 The mobile platform nominal parameters [58]

Parameter	Description	Values	units
r_a	The radius of each driving wheel	0.075	m
b	Distance between the driving wheels and the axis of symmetry	0.171	m
d	Distance between point $\boldsymbol{p}^{\circ}$ and point $\boldsymbol{p_c}$ on the plat form surface	0.4	m
I_{xx}	The moment for the inertia of chassis in—x-direction	0	Kg m^2
I_{yy}	The moment for the inertia of wheels in—y-direction	0.139	Kg m^2
I_{zzc}	Moment of inertia of chassis in—z-direction	6.609	Kg m^2
I_{zzw}	Moment of inertia of wheels in—z-direction	0.010	Kg m^2
m_c	Chassis mass	94	Kg
m_w	Wheel mass	5	Kg

Table 2 Mobile platform parameter with 100%; uncertainty and friction coefficients

Parameter	Definition	Value	Unit
m_c	The maximum chassis mass value	188	Kg
I_{zzc}	The maximum moment of inertia of chassis in—z-direction	13,218	Kg m^2
F_{V1}	The viscous friction coefficient of the first wheel friction torque	25	Nm/rd/s
F_{V2}	The viscous friction coefficient of the second wheel friction torque	15	Nm/rd/s
F_{S1}	The Coulomb friction coefficient of the first wheel friction torque	55	Nm
F_{S2}	The Coulomb friction coefficient of the second wheel friction torque	70	Nm

is the selected point on the platform. The desired trajectories are the infinity shape, the circular and the fixed point.

- Infinity trajectory shape

The simulation results for the mobile robot with an infinity shape for the desired trajectory are presented here. The ability of the proposed controller to robustly forcing P_c to follow the infinity shape in presence of frictional torques on its wheels and the uncertainty in model parameters is demonstrated in Fig. 12. The mobile robot tracked the desired trajectory under the ACSMC in less than 2 s. with an ultimate bound on $|e|$ of around 0.04, as can be seen in Fig. 13. The evolution of the adaptive gains during the simulation are shown in Fig. 14. From which it can be seen that the adaptive gains, k_1 and k_2, start to increase, from their initial values, to meet the set demand on s ($\varepsilon = 0.04$) in the adaptation law until they reach the minimum values that satisfy the demand (at $k_1 \cong 180$ and $k_2 \cong 160$) after approximately 400 s. In addition, it can be seen from Fig. 15 that the chattering problem was avoided by the use of the signum approximation function in Eq. (4.19).

Table 3 Control adaptation law parameter and control gain

Parameter	Definition	Value	Units
η_1	A positive number in the adaptation law for first control signal gain Eq. (4.37)	20	
η_2	A positive number in the adaptation law for second control signal gain Eq. (4.37)	15	
ϵ_1	Small Positive number In the adaptation law for the first control signal gain Eq. (4.37)	0.05	
ϵ_2	Small Positive number In the adaptation law for the second control signal gain Eq. (4.37)	0.05	
ε	Small Positive number In the approximation law for first and second control signal gain Eq. (4.19)	0.1	
k_{1min}	The minimum value for the first control signal with signum approximation function	10	N m
k_{1max}	Maximum value for first control signal with signum approximation function	750	N m
k_{2min}	The minimum value for the second control signal with signum approximation function	10	N m
k_{2max}	The maximum value for the second control signal with signum approximation function	1000	N m
$k_{1\circ}$	The initial value for first gain with signum approximation function	50	N m
$k_{2\circ}$	The initial value for first gain with signum approximation function	100	N m

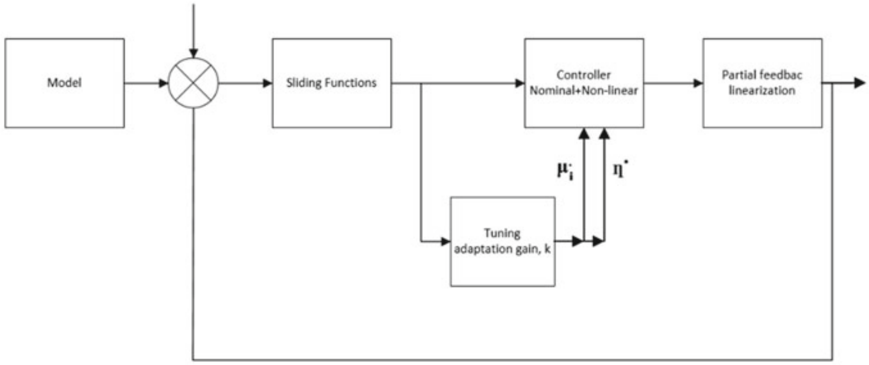


Fig. 11 The block diagram for the sliding mode control with adaptive control gain

- Circular trajectory shape

The simulation results for the mobile robot with a circular shape for the desired trajectory (Eq. 5.2) are presented here. The platform trajectory is plotted in Fig. 16, while Fig. 17 show the tracking errors. In this figure, a similar performance to the previous case is shown in which the mobile robot tracked the desired in less than 2 s.

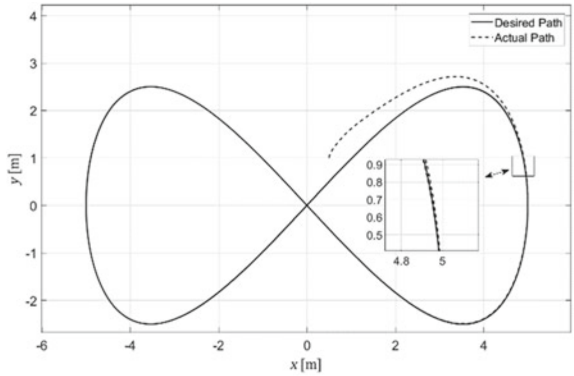


Fig. 12 Trajectory tracking

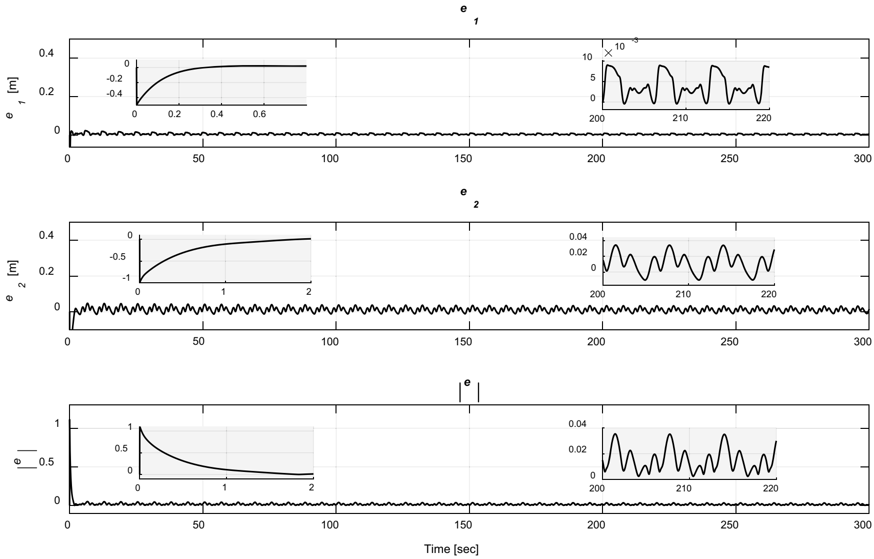


Fig. 13 Tracking errors

with an ultimate bound on $|e|$ of around 0.03. The evolution of the adaptive gains during the simulation are clarified in Fig. 18. from which it is clear that the adaptive gains, k_1 increased, at first, to 230 then start to gradually decay till it reaches around 195 after 400 s. while k_1 increase from the initial value to 145 after approximately 800 s. Finally, Fig. 19 shows the control torques.

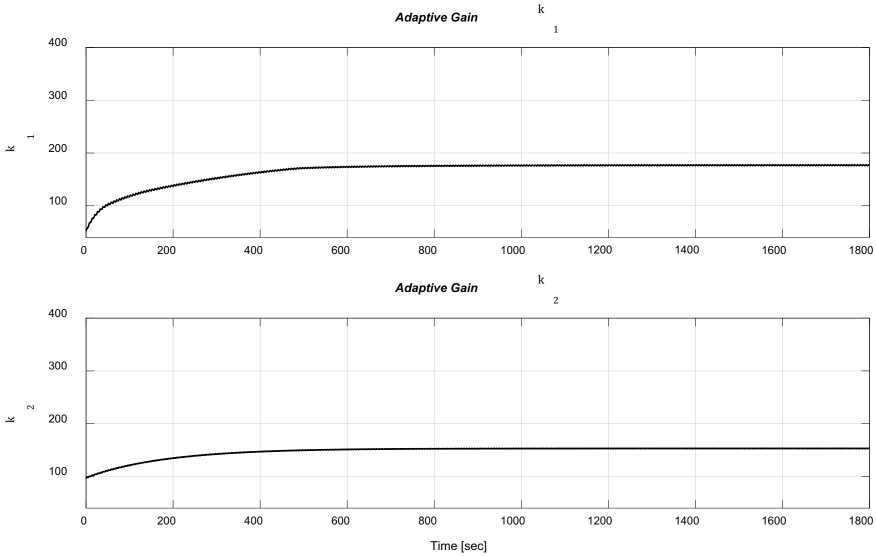


Fig. 14 Adaptive gains

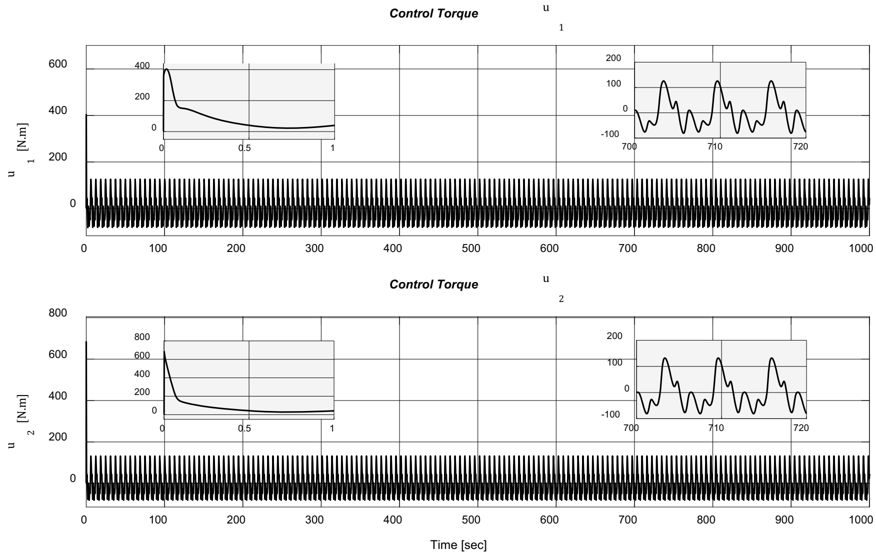


Fig. 15 Control torques

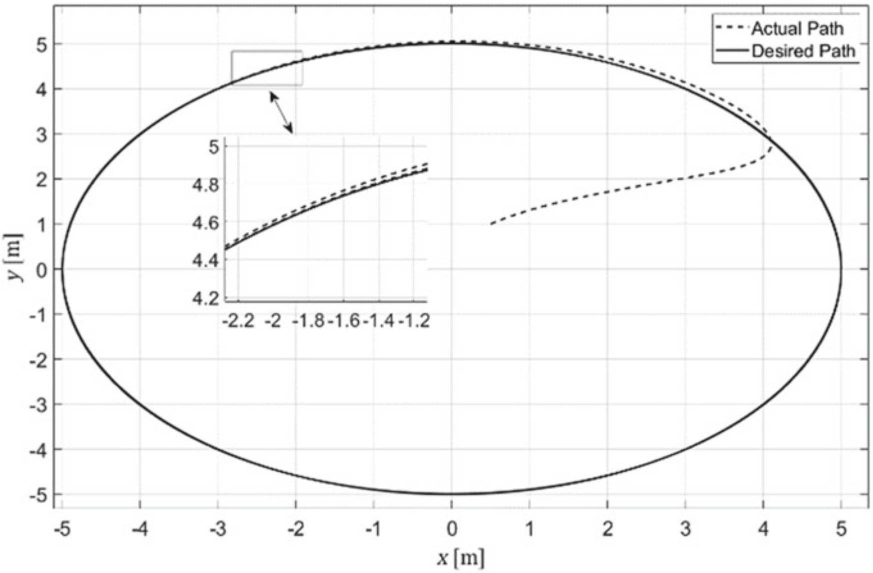


Fig. 16 Trajectory tracking

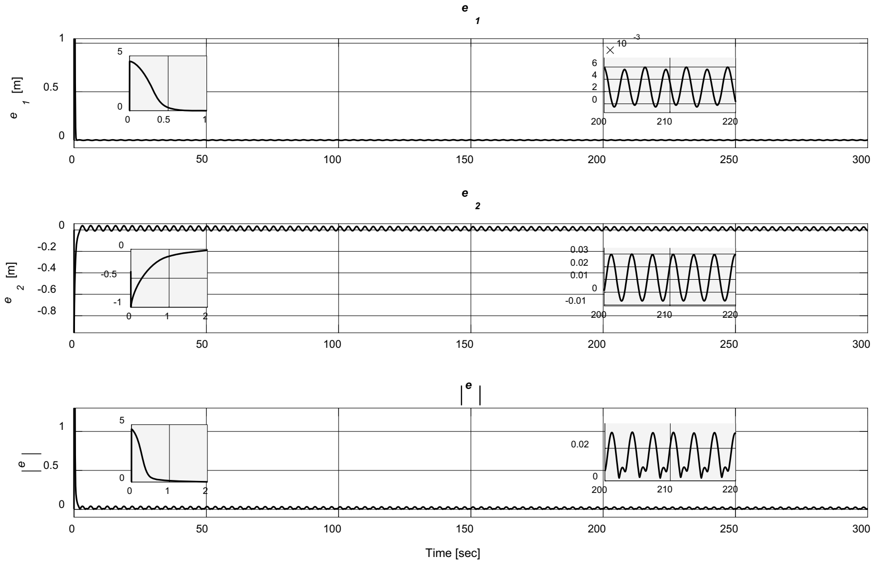


Fig. 17 Tracking errors

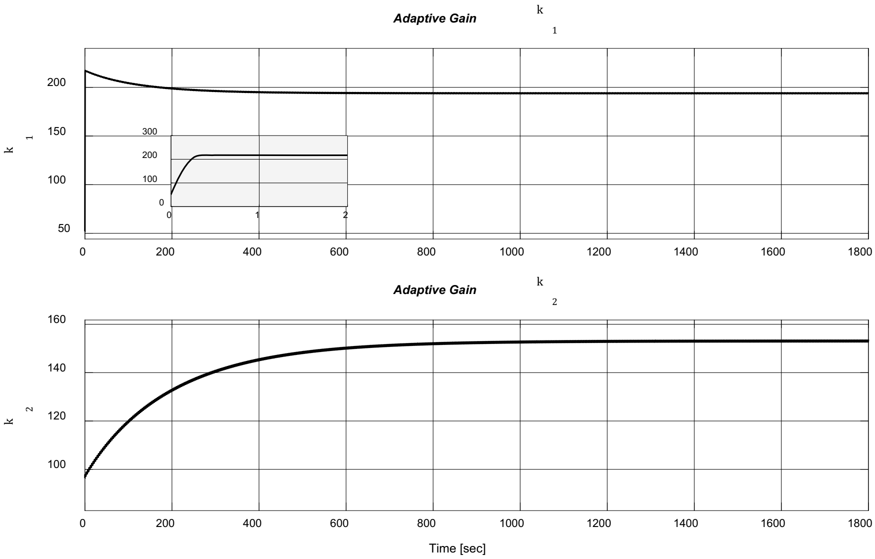


Fig. 18 Adaptive gains

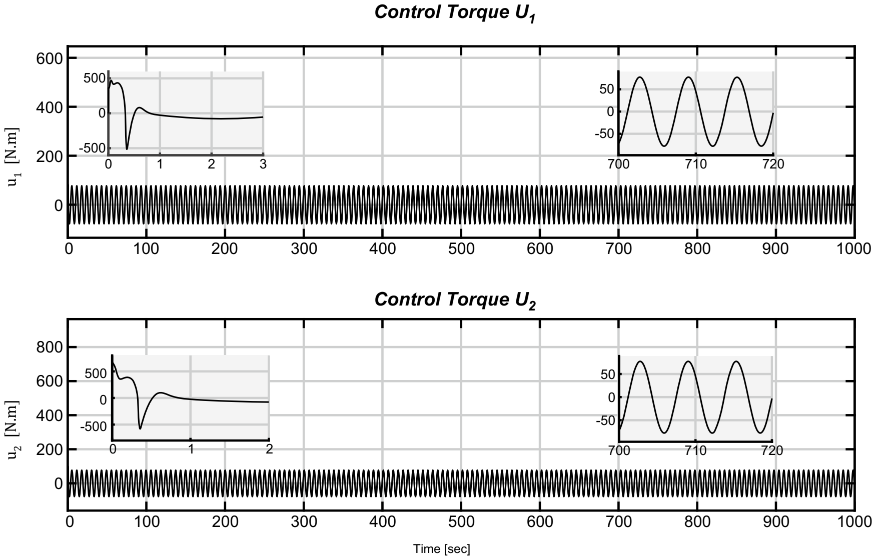


Fig. 19 Control torques

- The fixed point trajectory

Figures 20, 21 and 22 show the simulation results for a fixed point trajectory. The results show again the ability of the ACSMC in directing the platform to converge to the desired trajectory and then remain close to it.

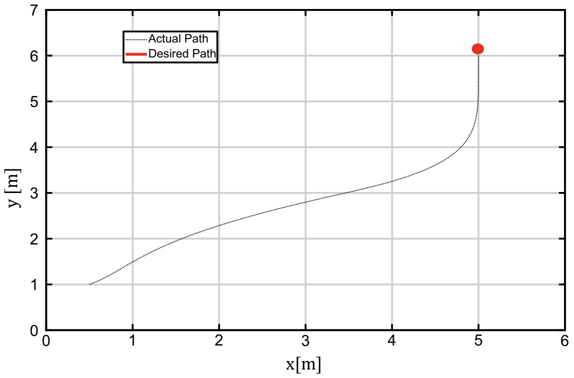


Fig. 20 Trajectory tracking

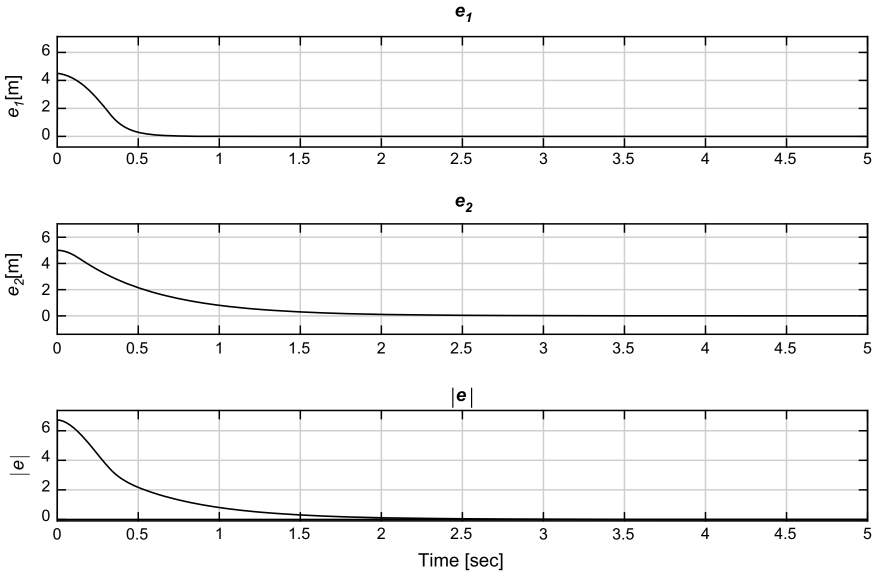


Fig. 21 Tracking errors

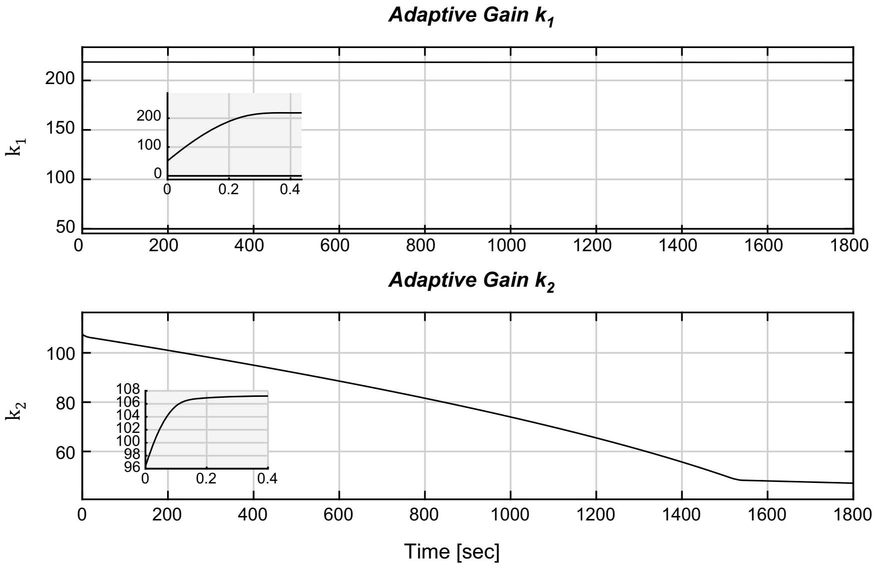


Fig. 22 Adaptive gains

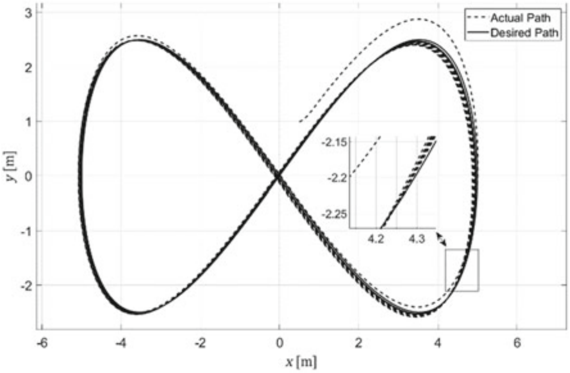
5.2 Simulation Results Based on AUSMC

This simulation results for the mobile platform trajectory tracking control system are presented in this section based on the AUSMC proposed in Sect. 4. The design parameters for the AUSMC in Eq. (4.53) with signum approximation function in Eq. (4.19) are given in Table 4 with the P and K matrices given by:

Table 4 Control adaptation law parameter and control gain for the AUSMC approach

Parameter	Definition	Value	Units
ξ	A positive number in the adaptation law Eq. (4.51)	10	
β	A positive number in the adaptation law Eq. (4.51)	0.2	
ε	A Small Positive number in the adaptation law Eq. (4.19)	0.001	
ε	A Positive number in the approximation law Eq. (4.51)	0.1	
k_{min}	The Minimum value of k	10	N m
k_{max}	The maximum value of k	750	N m
$k_{\circ}$	The Initial value of k	70	N m

Fig. 23 Trajectory tracking



$$P = \begin{bmatrix} 1.3917 & 0 & 0.0250 & 0 \\ 0 & 1.6128 & 0 & 0.0263 \\ 0.0250 & 0 & 0.0583 & 0 \\ 0 & 0.0263 & 0 & 0.0752 \end{bmatrix}$$

$$K = \begin{bmatrix} 20 & 0 & 9 & 0 \\ 0 & 19 & 0 & 7 \end{bmatrix}$$

Additionally, the system parameters, as for the ACSMC, are given in Table 4. Again, the control system simulations considered three desired trajectories. These are; the infinity shape, the circular shape, and the fixed point.

- Infinity trajectory shape

The simulation results for the mobile robot with an infinity shape for the desired trajectory in Eq. (5.1) are given in Figs. 23, 24, 25 and 26. It can be noted from Figs. 23 and 24 that the mobile robot successfully tracks the desired trajectory with an ultimate bound on $|e|$ of around 0.06 which agrees with Eq. (4.56).

The evolution of the adaptive gains during the simulation is shown in Fig. 25. From which the adaptive gain k starts to increase, from its initial values, at a constant rate until V drops below ε after around 120 s. then V start to oscillate around ε. While k continues to increase, the magnitude of this oscillation gradually decays until k reaches around 280 after 600 s. In addition, it can be seen from Fig. 26 that the chattering problem was avoided by the use of the signum approximation function in Eq. (4.19).

- Circular trajectory shape

The AUSMC is tested again, but now with a circular shape (Eq. 5.2) as the desired trajectory. Figures 27, 28, 29 and 30 show the simulation results. The platform trajectory is plotted in Fig. 27 while the error functions are plotted in Fig. 28. It can be seen that within less than 10 s. the mobile robot tracks the desired trajectory. The ultimate bound on $|e|$ was less than 0.04.

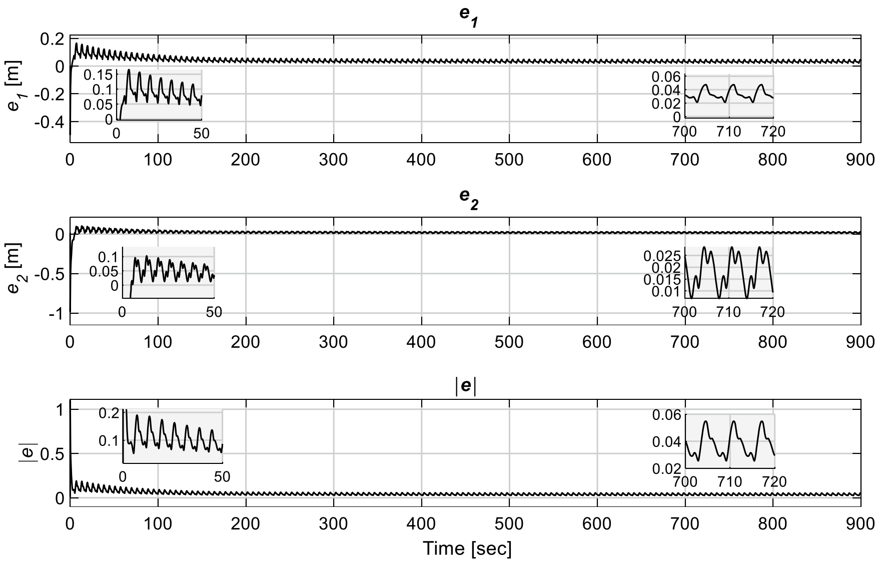


Fig. 24 Tracking errors

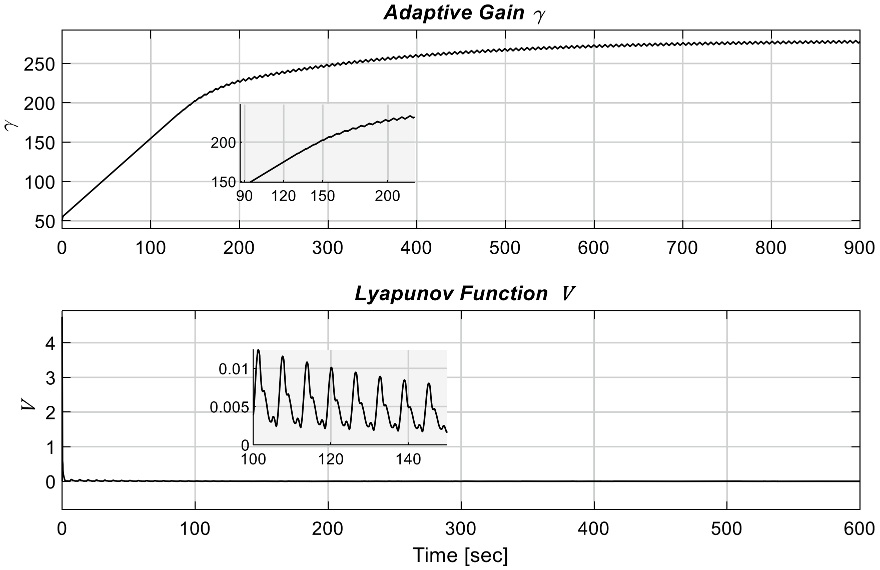


Fig. 25 Adaptive gains

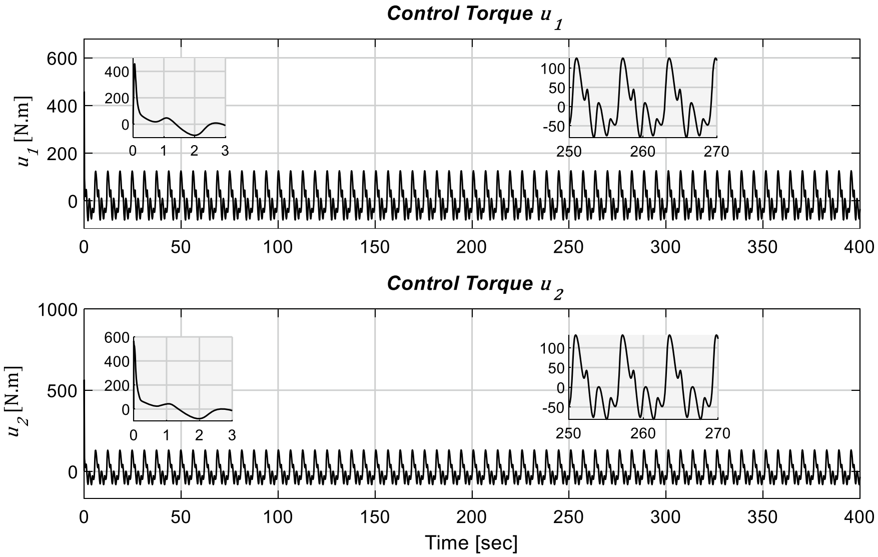


Fig. 26 Control torques

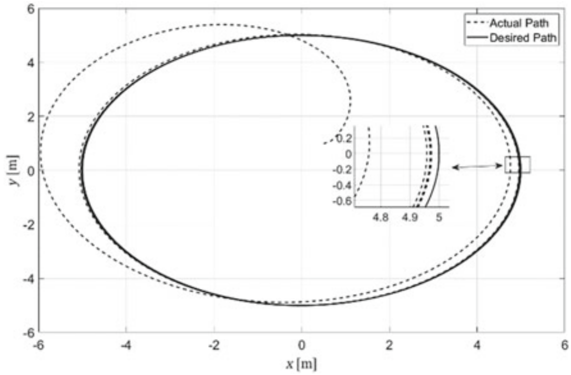


Fig. 27 Trajectory tracking

The evolution of the adaptive gain during the simulation can be seen in Fig. 29. From which it can be seen that the adaptive gain, k start to increase until it reaches around 420 after 200 s.

- Fixed point trajectory

Figures 31, 32 and 33 show the simulation results for a fixed point trajectory. The results show again the robustness of the proposed approach in tracking the desired trajectory.

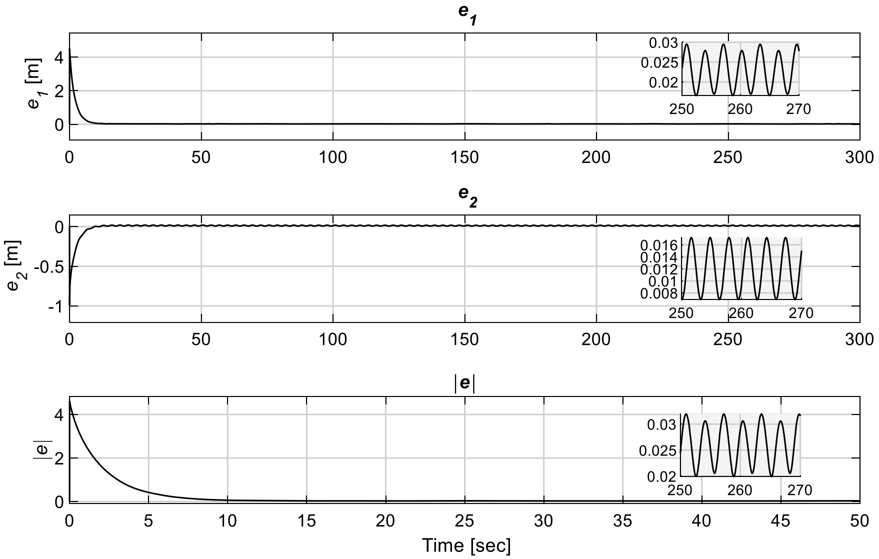


Fig. 28 Tracking errors

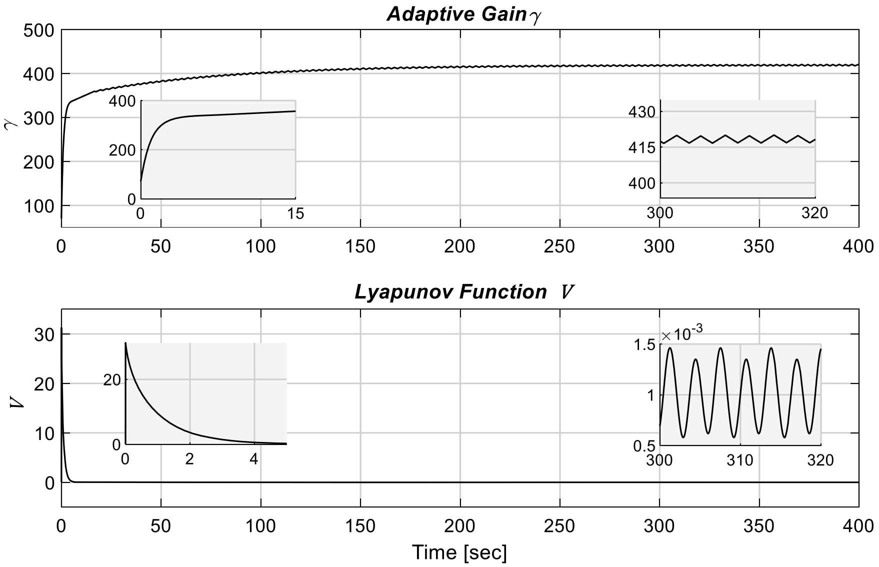


Fig. 29 Adaptive gains

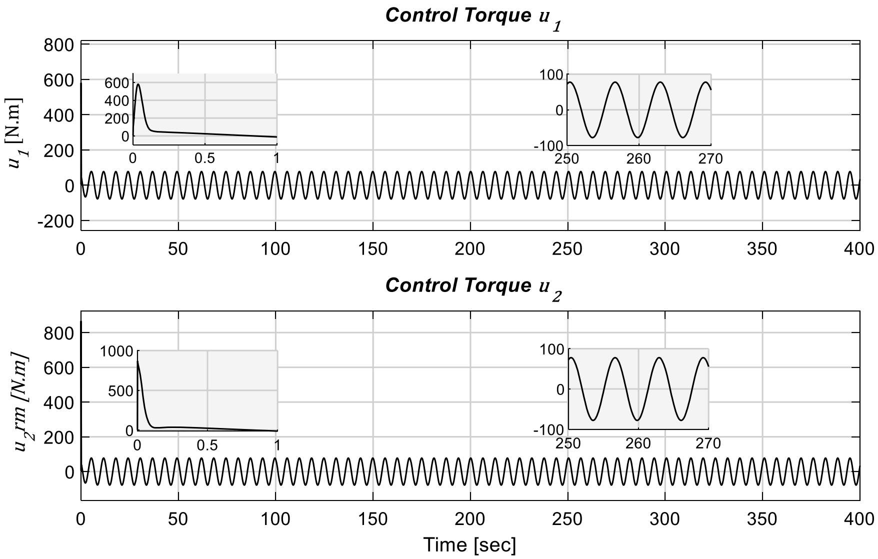
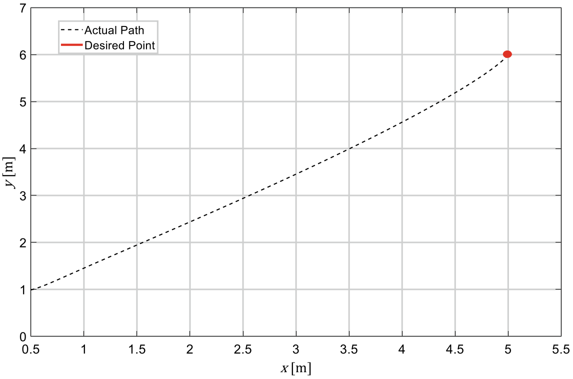


Fig. 30 Control torques

Fig. 31 Trajectory tracking



5.3 Performance Comparison

For comparison purposes, Figs. 34 and 35 show the simulation results for the mobile robot with parameter values from Tables 1 and 2 for a circular trajectory shape in Eq. (5.2) using three different control algorithms:

- A classic SMC:

$$\begin{aligned} u &= G_o^{-1}(x)(u_o + u_s) \\ u_o &= -F_o(x) - \Lambda\dot{e} \\ u_{si} &= -k_i sign(s_i) \end{aligned} \tag{5.4}$$

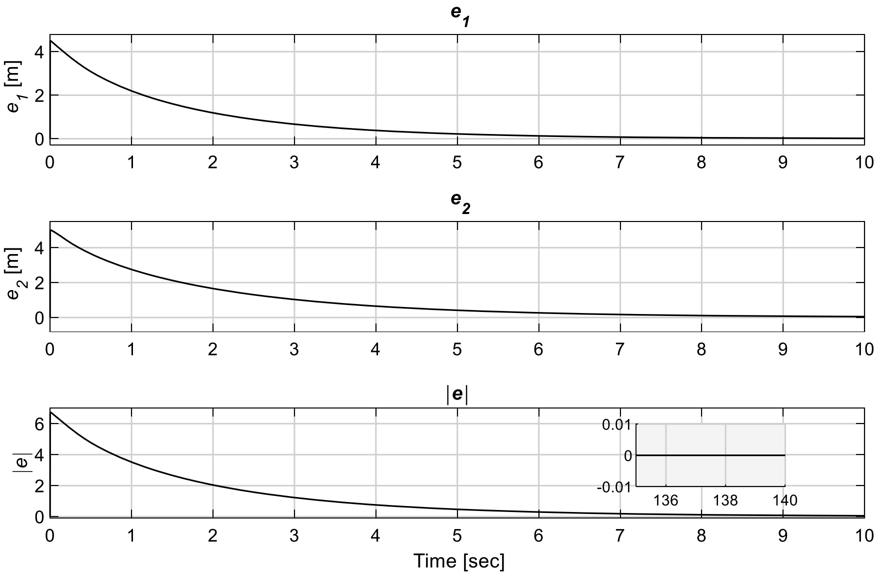


Fig. 32 Tracking errors

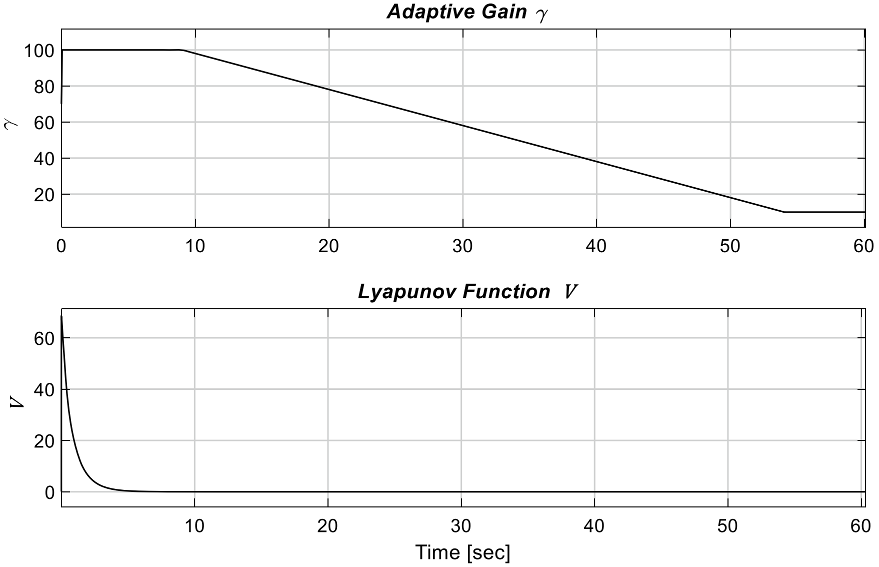


Fig. 33 Adaptive gains

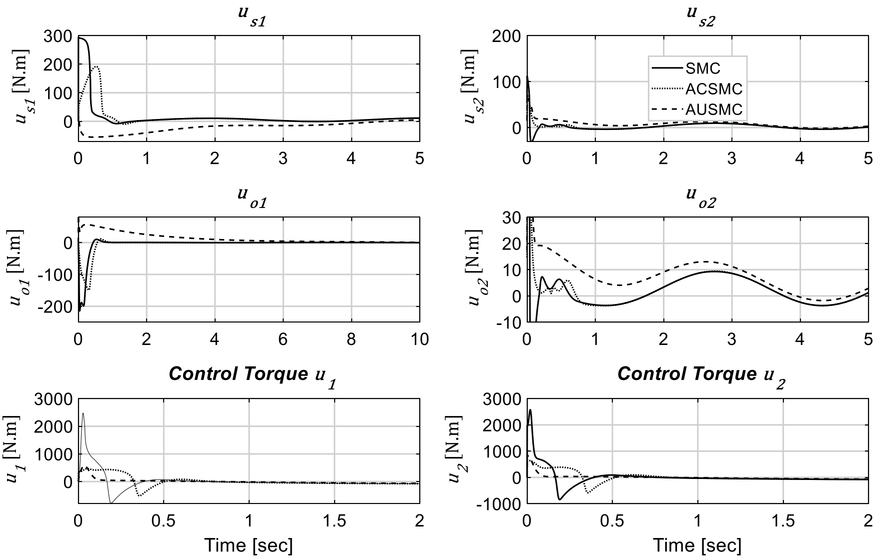


Fig. 34 Control signals

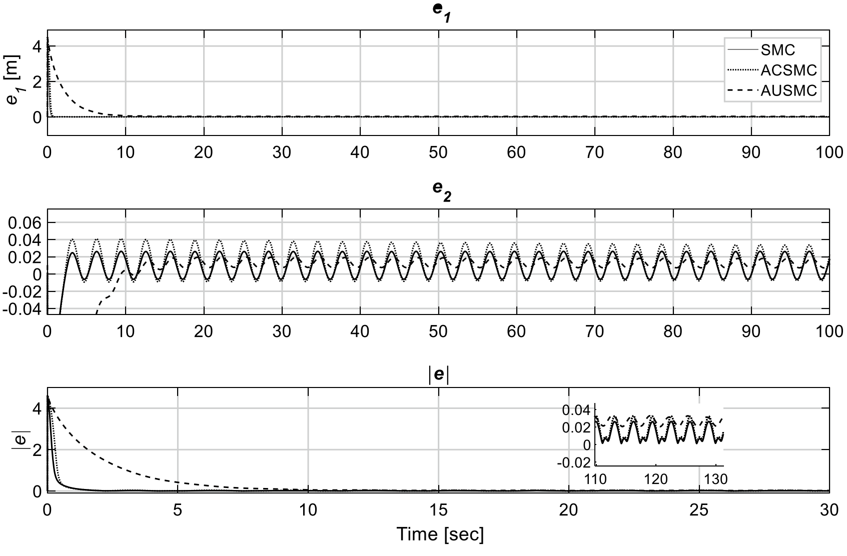


Fig. 35 Tracking errors

with $k_1 = 300$ and $k_1 = 150$.

- ACSMC with design parameter in Table 3.
- AUSMC with design parameters in Table 4.

From the results, it can be seen that all three algorithms achieved comparable results in terms of the tracking errors. The classic SMC seems to have faster convergence and smaller error but this is the trade-off with the conservativeness problem in selecting the control gain values. Overall, the AUSMC is superior to the other two in terms of simplicity and the number of design parameters.

5.4 Future Work

In the future, the following directions are considered:

1. An adaptive second order SMC can be applied for the error function model and the comparison can be made with the two adaptive SMCs proposed in this work.
2. The high order sliding mode control for non-holonomic mechanical systems can also be used to design a robust finite time stable control system.
3. A perturbation estimator based on the SMC theory can be used in order to estimate the perturbations which subjected to the platform and then use it to cancel the perturbation effect. This control type will reduce the discontinuous control gain and hence reduce the chattering in system response.

6 Conclusions

In this work, two adaptive controllers have been proposed for the mobile platform where the bound on model uncertainty, which is the main requirement for designing a classical sliding mode control, was not required here. The first controller was based on the classical SMC, while the second controller was proposed based on the USMC. The obtained controllers in addition to a CSMC are applied to the derived uncertain model for the mobile platform, which also subjected to the friction torques on the wheels. The results obtained via simulations for the proposed controllers reveal the robustness and ability of the proposed controllers in forcing the mobile platform to track the trajectory for various path shapes. Additionally, the following points can be deduced.

1. In spite that the ACSMC and the AUSMC can be applied for nonholonomic mechanical systems, a pioneering feature of the AUSMC is that it required only one control gain and consequently one adaptation law. This will greatly simplify the selection of the adaptation law parameters.
2. The control system performance when using the ACSMC method is better than the performance when using the AUSMC method, where the time required for the platform to reach the desired path is much smaller for the ACSMC. This

result because the adaption process is performed for two control gains which added more flexibility, while for AUSMC the adaptation is performed for only one control gain. Accordingly, the ACSMC is adapted efficiently for eliminating the system model uncertainty and making the outputs track the desired path.

3. The chattering was induced in mobile platform system response when applying the proposed controllers. However, by using the adapted gain in the SMC law, the chattering amplitude is reduced as can be seen when compared with the fixed gain value in classical sliding mode control. Additionally, further reduction was obtained by approximating the signum function.

Appendix A

- ***Computing the kinematic equation of mobile platform***

- ***Computing velocities by homogenous transformation matrices***

The homogenous transformation matrix is a square matrix $\in \mathrm{R}^{4\times 4}$ that consists of a rotation part which represents the rotation movement of the system (represented by the first 3 × 3 square matrix) and the fourth column in this matrix represents the linear motion of the system as illustrated below:

$$\begin{pmatrix} \mathrm{R} & \mathrm{L} \\ 0 & 1 \end{pmatrix}$$

Chassis linear velocity at point P_0 (Referring to the internal reference frame $(\mathrm{x}_0, \mathrm{y}_0)$):

$$\mathrm{T}^{\circ}_{\mathrm{P}_{circ}} = \begin{bmatrix} \mathrm{C}\varphi & -\mathrm{S}\varphi & 0 & \mathrm{x} \\ \mathrm{S}\varphi & \mathrm{C}\varphi & 0 & \mathrm{y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathrm{T}^{\circ}_{1}$$

$$\dot{\mathrm{T}}^{\circ}_{1} = \begin{bmatrix} -\mathrm{S}\varphi\dot{\varphi} & -\mathrm{C}\varphi\dot{\varphi} & 0 & \dot{\mathrm{x}} \\ \mathrm{C}\varphi\dot{\varphi} & -\mathrm{S}\varphi\dot{\varphi} & 0 & \dot{\mathrm{y}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{Velocity at } \dot{P}^{\circ}_{1} = \dot{T}^{\circ}_{1} P^{1} = \begin{bmatrix} -S\varphi\dot{\varphi} & -C\varphi\dot{\varphi} & 0 & \dot{x} \\ C\varphi\dot{\varphi} & -S\varphi\dot{\varphi} & 0 & \dot{y} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \\ 0 \end{bmatrix}$$

$$V_{xp^{\circ}} = \dot{x} \tag{A.1}$$

$$\mathrm{V}_{yp^{\circ}} = \dot{y} \tag{A.2}$$

$V_{p^{\circ}}$: velocity at P_0(the mid-point between the two wheels).

- ***Kinematic equations of two wheels***:

For the kinematic analysis of two wheels, the following coordinates for the wheels were defined as follows:

The coordinate of the first wheel as $\mathrm{x_{w1}y_{w1}z_{w1}}$ which is achieved by translating by $(-b)$ through $\mathrm{y_1}$ and then rotating by θ_1 about $\mathrm{y_{w1}}$.

The coordinate of the second wheel as $\mathrm{x_{w2}y_{w2}z_{w2}}$ which is achieved by translating by (b) through $\mathrm{y_1}$ and then rotating by θ_2 about $\mathrm{y_{w2}}$.

- ***Wheels linear velocity***:

$$\dot{P}^{w}_{Pw1,2,c} = \dot{T}P^{w}_{P_{\circ}}$$

$$V_{w1,2} = \dot{P}^{\circ}_{2,1} = \dot{T}^{\circ}_{2,w1}P^{2,w1}_{2,1}$$

$$T^{\circ}_{w1} = T^{\circ}_{2_{w1}}$$

$$= \begin{bmatrix} C\varphi & -S\varphi & 0 & x \\ S\varphi & C\varphi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\varphi & -S\varphi & 0 & x + bS\varphi \\ S\varphi & C\varphi & 0 & y - bC\varphi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\varphi C\theta_1 & -S\varphi & C\varphi S\theta_1 & bS\varphi + x \\ S\varphi C\theta_1 & C\varphi & S\varphi S\theta_1 & -bC\varphi + y \\ -S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$v_{w1c} = \dot{T}^{\circ}_{2\_1}P^{w}_{wc} = \dot{T}^{\circ}_{2}P^{w1}$$

$$= \begin{bmatrix} -S\varphi C\theta\dot{\varphi} - C\varphi S\theta\dot{\theta} & -C\varphi\dot{\varphi} & -S\varphi S\theta\dot{\varphi} + C\varphi C\theta\dot{\theta} & bC\varphi\dot{\varphi} + \dot{x} \\ C\varphi C\theta\dot{\varphi} - S\varphi S\theta\dot{\theta} & -S\varphi\dot{\varphi} & C\varphi S\theta\dot{\varphi} + S\varphi C\theta\dot{\theta} & bS\varphi\dot{\varphi} + \dot{y} \\ -C\theta\dot{\theta} & 0 & -S\theta\dot{\theta} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} bC\varphi\dot{\varphi} + \dot{x} \\ bS\varphi\dot{\varphi} + \dot{y} \\ 0 \\ 0 \end{bmatrix}$$

$$\left.\begin{aligned} V_{w1x} &= bC\varphi\dot{\varphi} + \dot{x} \\ V_{w1y} &= bS\varphi\dot{\varphi} + \dot{y} \end{aligned}\right\} \tag{A.3}$$

$$T^{\circ}_{w2} = T^{\circ}_{2_{w2}}$$

$$= \begin{bmatrix} C\varphi & -S\varphi & 0 & x \\ S\varphi & C\varphi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_2 & 0 & S\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta_2 & 0 & C\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$V_{w2} = \dot{T}^{\circ}_{2_{w2}} P^{w2} =$$

$$\begin{bmatrix} -S\varphi C\theta_2\dot{\varphi} - C\varphi S\theta_2\dot{\theta}_2 & -C\varphi\dot{\varphi} & -S\varphi S\theta_2\dot{\varphi} + C\varphi C\theta_2\dot{\theta}_2 & -bC\varphi\dot{\varphi} + \dot{x} \\ C\varphi C\theta_2\dot{\varphi} - S\varphi S\theta_2\dot{\theta}_2 & -S\varphi\dot{\varphi} & C\varphi S\theta_2\dot{\varphi} + S\varphi C\theta_2\dot{\theta}_2 & -bS\varphi\dot{\varphi} + \dot{y} \\ -C\theta_2\dot{\theta}_2 & 0 & -S\theta_2\dot{\theta}_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -bC\varphi\dot{\varphi} + \dot{x} \\ -bS\varphi\dot{\varphi} + \dot{y} \\ 0 \\ 0 \end{bmatrix}$$

$$\left.\begin{aligned} \mathrm{V_{w1x}} &= bC\varphi\dot{\varphi} + \dot{\mathrm{x}} \\ \mathrm{V_{w1y}} &= bS\varphi\varphi + \dot{\mathrm{y}} \end{aligned}\right\} \tag{A.4}$$

- ***Angular Velocity***:

Angular velocity has been evaluated by using the Homogenous transformation method depending on the SQ matrix and its rule as shown bellow:

$$S(\omega) = \dot{R}^{\circ}_{1} R^{\circ\,-1}_{1}$$

- ***Angular Velocity of chassis***:

$$S(\omega) = \dot{R}^{\circ}_{1} R^{\circ\,-1}_{1} = \begin{bmatrix} -S\varphi\dot{\varphi} & -C\varphi\dot{\varphi} & 0 \\ C\varphi\dot{\varphi} & -S\varphi\dot{\varphi} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} C\varphi & S\varphi & 0 \\ -S\varphi & C\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -\mathrm{S}\varphi\mathrm{C}\varphi\dot{\varphi}-\mathrm{C}\varphi\mathrm{S}\varphi\dot{\varphi} & -(\mathrm{S}\varphi^2+\mathrm{C}\varphi^2)\dot{\varphi} & 0 \\ (\mathrm{C}\varphi^2+\mathrm{S}\varphi^2)\dot{\varphi} & (\mathrm{S}\varphi\mathrm{C}\varphi-\mathrm{S}\varphi\mathrm{C}\varphi)\dot{\varphi} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{S}(\omega)=\begin{bmatrix} 0 & -\dot{\varphi} & 0 \\ \dot{\varphi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\therefore w=\dot{\varphi}k \tag{A.5}$$

- ***Angular velocity of wheels***:

$$S(\omega)=\dot{R}_2^{\circ}R_2^{\circ-1}=\dot{R}R^t$$

since the matrix(R) is an orthogonal rotation matrix:

$$\overset{\circ}{R_2}=\begin{bmatrix} \mathrm{C}\varphi & -\mathrm{S}\varphi & 0 \\ \mathrm{S}\varphi & \mathrm{C}\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \mathrm{C}\theta & 0 & \mathrm{S}\theta \\ 0 & 1 & 0 \\ -\mathrm{S}\theta & 0 & \mathrm{C}\theta \end{bmatrix}$$

$$=\begin{bmatrix} Col_1 & Col_2 & Col_3 \end{bmatrix}$$

where:

$$Col_1=\begin{bmatrix} -S\varphi C\theta^2C\varphi\dot{\varphi}-C\varphi^2S\theta C\theta\dot{\theta}+C\varphi S\varphi\dot{\varphi}-C\varphi S\varphi S\theta^2\dot{\varphi}+C\varphi^2S\theta C\theta\dot{\theta} \\ C\theta^2C\varphi^2\dot{\varphi}+S\varphi^2\dot{\varphi}+C\varphi S\varphi C\theta S\theta\dot{\theta}+C\varphi^2S\theta^2\dot{\varphi}-S\varphi C\varphi S\theta C\theta\dot{\theta} \\ -C\varphi C\theta^2\dot{\theta}-S\theta^2C\varphi\dot{\theta} \end{bmatrix}$$

$$Col_2=\begin{bmatrix} -C\varphi^2\dot{\varphi}-C\varphi^2C\theta^2\dot{\varphi}+S\varphi C\varphi S\theta C\theta\dot{\theta}-S\varphi^2C\theta^{2\dot{\varphi}}-S\varphi C\varphi S\theta C\theta\dot{\theta} \\ C\varphi S\varphi\dot{\varphi}-S\varphi C\varphi C\theta^2\dot{\varphi}-S\varphi^2S\theta C\theta\dot{\theta}-C\varphi S\varphi S\theta^2\dot{\varphi}+S\varphi^2C\theta S\theta\dot{\theta} \\ C\varphi C\theta^2\dot{\theta}+C\varphi S\theta^2\dot{\theta} \end{bmatrix}$$

$$Col_3=\begin{bmatrix} \mathrm{S}\varphi\mathrm{C}\theta\mathrm{S}\theta\dot{\varphi}+C\varphi S\theta^2\dot{\theta}-S\varphi C\theta S\theta\dot{\varphi}+C\varphi C\theta^2\dot{\theta} \\ -\mathrm{C}\varphi\mathrm{S}\theta\mathrm{C}\theta\dot{\varphi}+S\varphi S\theta^2\dot{\theta}+C\varphi S\theta C\theta\dot{\varphi}+S\varphi C\theta^2\dot{\theta} \\ -\mathrm{C}\theta\mathrm{S}\theta\dot{\theta}+C\theta S\theta\dot{\theta} \end{bmatrix}$$

$$S(w1)=\begin{bmatrix} 0 & -\dot{\varphi} & C\varphi\dot{\theta}_1 \\ \dot{\varphi} & 0 & S\varphi\dot{\theta}_1 \\ -C\varphi\dot{\theta}_1 & -S\varphi\dot{\theta}_1 & 0 \end{bmatrix}$$

$$\left.\begin{aligned} w_{x1}&=-S\varphi\dot{\theta}_1 \\ w_{y1}&=C\varphi\dot{\theta}_1 \\ w_{z1}&=\dot{\varphi} \end{aligned}\right\} \tag{A.6}$$

$$S(w2) = \begin{bmatrix} 0 & -\dot{\varphi} & C\varphi\dot{\theta}_2 \\ \dot{\varphi} & 0 & S\varphi\dot{\theta}_2 \\ -C\varphi\dot{\theta}_2 & -S\varphi\dot{\theta}_2 & 0 \end{bmatrix}$$

$$\left.\begin{aligned} w_{x2} &= -S\varphi\dot{\theta}_2 \\ w_{y2} &= C\varphi\dot{\theta}_2 \\ w_{z2} &= \dot{\varphi} \end{aligned}\right\} \tag{A.7}$$

(S) and (C) represent sin and cosine functions respectively.

Appendix B

The dynamic model of the n-dimensional platform with H holonomic constraints and k non-holonomic constraints was driven in this section. The Lagrange method was used to represent the system mathematically. The Lagrangian L is defined as the difference between the system kinetic and potential energy:

$$L(q, \dot{q}) = T(q, \dot{q}) - u(q) = 1/2\dot{q}^t M(q)\dot{q} - u(q) \tag{B.1}$$

where $T(q, \dot{q})$ is the kinetic energy and $u(q)$ is potential energy.

The Euler Lagrange equation of motion is

$$L = L = \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i}; \text{ where } \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} i = 1, 2 \ldots, n$$

Computing kinetic energy

$$\left.\begin{aligned} T &= T_W + T_C \\ T_W &= T_{W1} + T_{W2} \\ T_C &= \tfrac{1}{2} m v_c^2 + \tfrac{1}{2} I_{zc} {w_{zc}}^2 \end{aligned}\right\} \tag{B.2}$$

Since T, T_W, T_C are total kinetic energy, wheels and chassis kinetic energy respectively

$$T_{W1} = \frac{1}{2} m_{wa}(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} I_{y1}\dot{\theta}_1^2 + \frac{1}{2} I_{z1}\dot{\varphi}^2$$

$$T_{W2} = \frac{1}{2} m_{w2}(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} I_{y2}\dot{\theta}_2^2 + \frac{1}{2} I_{z2}\dot{\varphi}^2$$

$$
\begin{aligned}
T &= \frac{1}{2}mv_c^2 + \frac{1}{2}I_{zc}\dot{\varphi}^2 + \frac{1}{2}m_w(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}(I_{y1}\dot{\theta}_1^2) + \frac{1}{2}I_{z1}\dot{\varphi}^2 \\
&\quad + \frac{1}{2}m_w(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}I_{y2}\dot{\theta}_2^2 + \frac{1}{2}I_{z2}(\dot{\varphi}^2) \\
&= \frac{1}{2}\mathrm{m_c}(\dot{\mathrm{x}}^2 + \dot{\mathrm{y}}^2) + \frac{1}{2}\mathrm{I_{zc}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{m_w}(\dot{\mathrm{x}}^2\_2\mathrm{b}\cos(\varphi)\dot{\varphi}\dot{\mathrm{x}} + \mathrm{b}^2(\cos(\varphi))^2\dot{\varphi}^2 \\
&\quad + \dot{\mathrm{y}}^2 - 2\mathrm{b}\sin(\varphi)\dot{\varphi}\dot{\mathrm{y}} + \mathrm{b}^2(\sin(\varphi))^2\dot{\varphi}^2 + \dot{\mathrm{x}}^2 + 2\mathrm{b}\cos(\varphi)\dot{\varphi}\dot{\mathrm{x}} + \mathrm{b}^2(\cos(\varphi))^2\dot{\varphi}^2 \\
&\quad + \dot{\mathrm{y}}^2 + 2\mathrm{b}\sin(\varphi)\dot{\mathrm{y}} + \mathrm{b}^2(\sin(\varphi))^2\dot{\varphi}^2) + \frac{1}{2}(\mathrm{I_{y1}}\dot{\theta}_1^2) + \frac{1}{2}\mathrm{I_{z1}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{I_{y2}}\dot{\theta}_2^2 + \frac{1}{2}\mathrm{I_{z2}}(\dot{\varphi}^2)
\end{aligned}
\tag{B.3}
$$

$$
\begin{aligned}
T &= \frac{1}{2}\mathrm{m_c}(\dot{\mathrm{x}}^2 + \dot{\mathrm{y}}^2) + \frac{1}{2}\mathrm{I_{zc}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{m_w}(2\dot{\mathrm{x}}^2 + 2\dot{\mathrm{Y}}^2 + 2\mathrm{b}^2\dot{\varphi}^2\big[(\cos(\varphi))^2 + (\sin(\varphi))^2\big] \\
&\quad + \frac{1}{2}(\mathrm{I_{y1}}\dot{\theta}_1^2) + \frac{1}{2}\mathrm{I_{z1}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{I_{y2}}\dot{\theta}_2^2 + \frac{1}{2}\mathrm{I_{z2}}(\dot{\varphi}^2)
\end{aligned}
$$

$$
\begin{aligned}
T &= \frac{1}{2}\mathrm{m_C}(\dot{\mathrm{x}}^2 + \dot{\mathrm{y}}^2) + \frac{1}{2}\mathrm{I_{zc}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{m_w}(2\dot{\mathrm{x}}^2 + 2\dot{\mathrm{Y}}^2 + 2\mathrm{b}^2\dot{\varphi}^2) \\
&\quad + \frac{1}{2}\mathrm{I_{y1}}\dot{\theta}_1^2 + \frac{1}{2}\mathrm{I_{z1}}\dot{\varphi}^2 + \frac{1}{2}\mathrm{I_{y2}}\dot{\theta}_2^2 + \frac{1}{2}\mathrm{I_{z2}}\dot{\varphi}^2
\end{aligned}
\tag{B.4}
$$

where m_c, m_w, I_y, I_z represent chassis mass, wheels mass, the moment of inertia about the y axis and z axis respectively,

By applying Eq. (3.13) (holonomic constraint) on Eq. (B.4):

$$
\begin{aligned}
T &= \frac{1}{2}m_c(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_{zc}c^2(\dot{\theta}_1 - \dot{\theta}_2)^2 + m_w(\dot{x}^2 + \dot{y}^2 + b^2c^2(\dot{\theta}_1 - \dot{\theta}_2)^2) \\
&\quad + \frac{1}{2}I_{ya}\dot{\theta}_1^2 + \frac{1}{2}I_{Za}c^2(\dot{\theta}_1 - \dot{\theta}_2)^2 + \frac{1}{2}I_{yb}\dot{\theta}_2^2 + \frac{1}{2}I_{zb}c^2(\dot{\theta}_1 - \dot{\theta}_2)^2
\end{aligned}
\tag{B.5}
$$

$$
L(q, \dot{q}) = T(q, \dot{q}) - u(q) = \frac{1}{2}(\dot{q}t)M(q)\dot{q} - u(q)
$$

$$
L = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{qi}}\right) - \frac{\partial L}{\partial qi};
$$

where:

$$
\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} i = 1, 2, \ldots, n
$$

$$
L = M(q)\ddot{q} + D(q, \dot{q}) + G(q) = A(q)t\lambda + E(q)U
\tag{B.6}
$$

where L is the lagrangian equation, $M(q)$ is the mass and inertia positive definite matrix, and $D(q, q^{cicr})$ is the carioles matrix.

$$M(q)\ddot{q} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$m_{11} = \frac{d}{dt}\left(\frac{\partial L}{\partial \,\ddot{x}\dot{i}}\right) = (m_c + 2m_w)$$

$$m_{12} = \frac{d}{dt}(\frac{\partial L}{\partial \dot{y}i}) - \frac{\partial L}{\partial yi} = 0;$$

$$m_{13} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}i}\right) - \frac{\partial L}{\partial xi} = 0$$

$$m_{14} = 0$$

$$m_{21} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}i}\right) - \frac{\partial L}{\partial yi} = 0$$

$$m_{22} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}i}\right) - \frac{\partial L}{\partial yi} = (m_c + 2m_w)$$

$$m_{23} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}i}\right) - \frac{\partial L}{\partial yi} = 0$$

$$m_{42} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}i}\right) - \frac{\partial L}{\partial yi} = 0$$

$$m_{31} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}i}\right) - \frac{\partial L}{\partial \theta i} = 0$$

$$m_{32} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}i}\right) - \frac{\partial L}{\partial \theta i} = 0$$

$$m_{33} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}i}\right) - \frac{\partial L}{\partial \theta i} = I_{zzc}c^2 + +2m_w b^2 c^2 + I_{zzw}c^2 + I_{zw}c^2 + I'_{yy}$$

$$m_{34} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}1}\right) - \frac{\partial L}{\partial \theta 1} = -I_{zc}c^2 - 2m_W b^2 c^2 - I_{ZAW}c^2 + I_{zbw}c^2$$

$$m_{41} = \frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}2}) - \frac{\partial L}{\partial \theta 2} = 0$$

$$m_{42} = \frac{\mathrm{d}}{\mathrm{dt}}(\frac{\partial \mathrm{L}}{\partial \dot{\theta 2}}) - \frac{\partial \mathrm{L}}{\partial \theta 2} = 0$$

$$m_{43} = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathrm{L}}{\partial \dot{\theta 2}}\right) - \frac{\partial \mathrm{L}}{\partial \theta 2} == -I_{zc}c^2 - 2m_W b^2c^2 - I_{ZAW}c^2 + I_{zbw}c^2$$

$$m_{44} = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathrm{L}}{\partial \dot{\theta 2}}\right) - \frac{\partial \mathrm{L}}{\partial \theta 2} = I_{zc}C^2 + 2m_W b^2c^2 + I_{zaw}c^2 + I_{zbw}c^2 + I_{ya}$$

Since $M(q)$ positive definite symmetric matrix, $D(q, \dot{q})\dot{q} = 0$. $G(q) = 0$ since the potential energy is (zero) (horizontal motion) where $G(q)$ is potential forces matrix.

It cab be noticed that the nonholonomic constraint matrix $A(q)\dot{q}$ Eq. (3.12) is not full rank. its dimension $(m \times n)$ since $m = n - k = 2, n = 4$; So, the matrix $A(q)$ has a kernel space $A(q) \neq [0]$. Where n, k are the system configuration after holonomic constraint, the nonholonomic constrain number and m is the number of columns for the null space matrix respectively.

$$Let\, kernel\; A(q) = \begin{bmatrix} a & b & c & d \\ e & f & \boldsymbol{g} & h \end{bmatrix}, so\; N(q)A(q)^t = 0$$

$$\begin{bmatrix} a & b & c & d \\ e & f & \boldsymbol{g} & h \end{bmatrix} \begin{bmatrix} -\sin\varphi & \cos\varphi \\ \cos\varphi & \sin\varphi \\ 0 & -r/2 \\ 0 & -r/2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.7}$$

since kernel space of $A(q)$ is orthogonal on $A(q)$, first raw:

$$-a\sin\varphi + b\cos\varphi = 0;\; \text{let}\, a = (r/2)\cos\varphi$$

Therefore:

$$b = r\sin\varphi\cos\varphi/2(\cos\varphi) = (r/2)\sin\varphi$$

$$r/2(\cos\varphi)^2 + r/2(\sin\varphi)^2 - r/2 \times C - r/2 \times d = 0$$

let $d = 0$, *then*:

$$r/2((\cos\varphi)^2 + r/2((\sin\varphi)^2 - r/2 \times C$$

$$= r/2((\cos\varphi)^2 + (\sin\varphi)^2) - r/2 \times C$$

$$Then\; C = 1$$

Second raw:

$$Let\, e = r/2 \cos\omega,\ f = r/2 \sin\omega$$

$$= e\cos\varphi + f\sin\varphi - r \times \boldsymbol{g} - r \times h$$

$$r/2(\cos\varphi)^2 + r/2(\sin\varphi)^2 - r/2 \times \boldsymbol{g} - r/2 \times h = 0;$$

let $\boldsymbol{g} = 0$, then

$$r/2(\cos\varphi)^2 + r/2(\sin\varphi)^2 - r/2 \times h = r/2(\cos\varphi^2 + \sin\varphi^2) - r/2 \times h$$

Then $h = 1$

$$kernel\, A(q) = \begin{bmatrix} r/2\cos\varphi & r/2\sin\varphi & 1 & 0 \\ r/2\cos\varphi & r/2\sin\varphi & 0 & 1 \end{bmatrix} = N(q)^T \tag{B.8}$$

Consider a matrix $N(q)$ which columns form the basis for the kernel space of $A(q)$. So $N(q)^T A(q) = 0$. $E(q)$ is an $n \times m$ matrix which maps the $m = n - k$ external inputs u into forces /torques that perform work on q, $\lambda \varepsilon R^m$ is the vector of unknown Lagrange multiplier and the term $A(q)^T\lambda$ represent the vector of constrain forces

$$= N(q)^T[M(q)q^\circ + D(q, q^\circ) + G(q) = A(q)^T\lambda + E(q)U]$$

Multiply by $N(q)^T$ is needed in order to cancel the unknown Lagrange multiplier vector λ since $N(q)^T$ multiply by $A(q)^T$ equal to zero.

Our system has 4 (D.O.F) and because of the presence of non-holonomic constraints, the number of (the state in control theory) is equal to $(6 = n + k)$ as demonstrated below.

Change of variables: Since

$$A(q)\dot{q} = 0, A(q)N(q)\rho = 0$$

Therefore

$$\dot{q} = N(q)\rho$$

where ρ is a vector $\in R^m$, $N(q) \in R^{(n\times m)}$

$$\dot{q} = N(q)\rho \text{ (kinematic velocity)} \tag{B.9}$$

$$\ddot{q} = \dot{N}(q)\rho + N(q)\dot{\rho} \tag{B.10}$$

So, the Eq. (B6) become as below:

$$N(q)^T[M(q)\big(\dot{N}(q)\rho + N(q)\dot{\rho}\big] + D(q,\dot{q}) + G(q) = A(q)^t\lambda + E(q)U \quad \text{(B.11)}$$

$$\mathrm{E}(q) = \begin{bmatrix} 0\ 0 \\ 0\ 0 \\ 1\ 0 \\ 0\ 1 \end{bmatrix}$$

$$N(q)^T A(q)^T \lambda = 0$$

$$N(q)^T\mathrm{E}(q) = \begin{bmatrix} r/2\cos\varphi\ r/2\sin\varphi\ 1\ 0 \\ r/2\cos\varphi\ r/2\sin\varphi\ 0\ 1 \end{bmatrix}\begin{bmatrix} 0\ 0 \\ 0\ 0 \\ 1\ 0 \\ 0\ 1 \end{bmatrix} = \begin{bmatrix} 1\ 0 \\ 0\ 1 \end{bmatrix}$$

Then Eq. (B.9):

$$\rho^\circ = [N(q)^t M(q)N(q))]^{-1}[-N(q)^t(M(q)N^\circ(q)\rho - N(q)tD(q,q^\circ)) + [\begin{bmatrix} 1\ 0 \\ 0\ 1 \end{bmatrix}][\begin{matrix} u_1 \\ u_2 \end{matrix}]]\ \text{(Differintiable velocity)}$$

$$\text{(B.12)}$$

$$\begin{bmatrix} N(q)\rho \\ [\big(N(q)^T M(q)N(q)\big)^{-1}][-N(q)^T(M(q)\dot{N}(q)\rho - N(q)^T D(q,\dot{q})] + \begin{bmatrix} 1\ 0 \\ 0\ 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \end{bmatrix}$$

Let:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} r/2\cos\varphi & r/2\cos\varphi \\ r/2\sin\varphi & r/2\sin\varphi \\ 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_5 \\ x_6 \end{bmatrix} \quad \text{(B.13)}$$

$$\dot{\rho} = \begin{bmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \end{bmatrix} = \begin{bmatrix} \dot{x_5} \\ \dot{x_6} \end{bmatrix}$$

$$= \begin{bmatrix} f_1(q,\rho) \\ f_2(q,\rho) + \boldsymbol{g}(q,\rho)u \end{bmatrix}$$

where

$$f_1(q,\rho) = N(q)\big[x_5\ x_6\big]^T,$$

$$f_2(q,\rho) = -\big[N^T(q)M(q)N(q)\big]^{-1}N^T(q)M(q)\dot{N}(q)\rho \text{ and}$$

$$\boldsymbol{g}(q,\rho) = \big[N^T(q)M(q)N(q)\big]^{-1}N^T(q)E(q)$$

as it can be seen above the effect of non-holonomic constraints it was appeared by multiplying the dynamic equation by the kernel space of the non-holonomic constraint matrix to cancel the term which contains the unknown $lagrange\, multiplier$ (λ) and this has caused change of variable and eliminate the number of differentiable velocity space from 4 to $2 = \begin{bmatrix} \dot{x_5} \\ \dot{x_6} \end{bmatrix}$ and this caused elimination the number of system state from (8 to 6) as note there are 6 state to represent the Mobil plat form first four state $[\dot{q}]$ are kinematic velocity and $\begin{bmatrix} \dot{x_5} \\ \dot{x_6} \end{bmatrix}$ are the dynamic or differentiable velocity so the system can be considered as fully actuated although it has 4(D.O.F) and two input (torque) if the system was looked at from this point and change the system from under actuator to full actuator if looked at in the since of number of differentiable velocity which equal to the input torque and this achieved because of nonholonomic constraints.

References

1. Rabior P, Rheinboldt W (2000) Non-holonomic in motion of rigid mechanical systems from ADE. In: View point, other titles in applied mathematics. SIMA, Philadelphia, USA. https://doi.org/10.1137/1.9780898719536
2. Monforte JC (2002) Geometric control and numerical aspects of nonholonomic systems. Lecture notes in mathematics, vol 1793. Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/b84020
3. Athans M (1971) Editional on the LQG problem. IEEE Trans Autom Control 16(6):528–528. https://doi.org/10.1109/TAC.1971.1099845
4. Askerman J (1993) Robuste regelung. Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-09777-9
5. Morari M, Zafiriou E (1989) Robust process control theory. Prentice Hall, Englewood Cliffs, NJ, USA
6. Safonov MG (1980) Stability and Robustness of multi variable feedback systems. MIT Press, Cambridge, M.A., USA
7. Zhou K, Doyle J (1999) Essentials of robust control. Prentice Hall, N.Y., USA
8. Filippov A (1988) Differential equations with discontinuous righthand sides – control systems. Mathematics and its applications, vol 18. Springer Dordrecht, Berlin, Heidelberg. https://doi.org/10.1007/978-94-015-7793-9
9. Popovski MA (1950) Linarization of sliding operation mode for a constant speed controller. Automatics itelemekhanika 11(3):161–163
10. Utkin VI (1972) Equations of sliding mode in discontinuous systems. Autom Remote Control 2(2):211–219
11. Neymark YI (1957) On sliding modes in relay control systems. Automatiks itelemekhanika 18(1):27–33
12. Lopez P, Nouri AS (2006) Theorie elemtaire et pratique deal command par les regimes glissants. Mathematiques et applications, vol 55. Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/3-540-34480-2
13. Utkin VI (1978) Discontinuous control systems: state of art in theory and application. IFAC Proc 5(1): 25–44. https://doi.org/10.1016/S1474-6670(17)55534-2
14. Utkin VI (1992) Sliding mode in control and optimization. Springer-Verlag, Berlin, Heidelberg, Communications and control engineering. https://doi.org/10.1007/978-3-642-84379-2

15. Soltine JJE, Li W (1991) Applied non-linear control. Prentice Hall, Englewood Cliffs, NJ, USA
16. Fliess M, Massager F (1991) Method of nonlinear discontinuous stabilization. In: Byrnes CI, Kurzhansky AB (eds) Nonlinear synthesis. Progress in systems and control theory, vol 9. Birkhauser, Boston, MA. https://doi.org/10.1007/978-1-4757-2135-5_8
17. Fridman L, Levant A (1999) Higher order sliding modes. Sliding modes in automatic control; Int. School in Automatic control of Lille
18. Itkis Y (1976) Control systems of variable structure. Wiley, NY, USA
19. Nawash N (2001) H-infinity control of an autonomous mobile robot, Master's thesis. Cleveland State University; May 2001
20. Kumar U, Sukavanam N (2008) Back stepping based trajectory tracking control for four wheeled mobile robot. Int J Adv Robot Syst 5(4):403–410. https://doi.org/10.5772/6224
21. Mohamed B, Tarak D, Mohamed J (2011) Robust adaptive control for mobile manipulators. Int J Autom Comput 8(1):8–13. https://doi.org/10.1007/s11633-010-0548-y
22. Cerkala J, Jadcoeska A (2014) Mobile robot dynamics with friction Simulink. In: 22th annual conference proceedings of the international scientific conference. Technical Computing Bratislava
23. Hwang CL, Wu HM (2013) Trajectory tracking of a mobile robot with frictions and uncertainties using hierarchical sliding-mode under-actuated control. IET Control Theory 7(7): 952–965. https://doi.org/10.1049/iet-cta.2012.0750
24. Goswami NK, Padhy PK (2018) Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance. Comput Electric Eng 72:307–323. https://doi.org/10.1016/j.compeleceng.2018.09.02
25. Boutalbi O, Benmahammed K, Henni K et al (2019) A high-performance control algorithm based on a curvature-dependent decoupled planning approach and flatness concepts for non-holonomic mobile robots. Intel Serv Robot 12(1):181–196. https://doi.org/10.1007/s11370-018-00270-7
26. Yousuf BM, Khan AS, Noor A (2019) Multi-agent tracking of non-holonomic mobile robots via non-singular terminal sliding mode control. Robotica 38(11):1–17. https://doi.org/10.1017/S0263574719001772
27. Li J, Wang J, Peng H, Hu Y, Su H (2022) Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot. IEEE Trans Syst Man Cybern Syst 52(4):2491–2500. https://doi.org/10.1109/TSMC.2021.3050616
28. Shafiei MH, Monfared F (2019) Design of a robust tracking controller for a nonholonomic mobile robot with side slipping based on Lyapunov redesign and nonlinear H∞ methods. Syst Sci Control Eng 7(1):1–11. https://doi.org/10.1080/21642583.2018.1555061
29. Han X, Ge M, Cui J, Wang H, Zhuang W (2019) Motion modeling of non-holonomic wheeled mobile robot based on trajectory tracking control. Trans Can Soc Mech Eng 44(2):228–233. https://doi.org/10.1139/tcsme-2019-0083
30. Hou R, Cui L, Bu X, Yang J (2021) Distributed formation control for multiple non-holonomic wheeled mobile robots with velocity constraint by using improved data-driven iterative learning. Appl Math Comput 395. https://doi.org/10.1016/j.amc.2020.125829
31. Bloch AM, Baillieul J, Crouch P, Mar J (2003) Non-holonomic mechanical and control system. Springer
32. Brizard AJ (2014) Introduction to Lagrangian and Hamiltonian mechanics. Department of Chemistry and Physics Saint Michael's College,14 July 2014
33. Pfeifferd F, Wriggers P (2008) Mechanical system dynamic. Lecture notes in applied and computational mechanics, vol 40. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-79436-3
34. Luca AD, Oriolo G (1995) Modelling and control of nonholonomic mechanical systems. In: Angeles J, Kecskeméthy A (eds) Kinematics and dynamics of multi-body systems. CISM international centre for mechanical sciences, vol 360. Springer, Vienna. https://doi.org/10.1007/978-3-7091-4362-9_7
35. Cushman R, Duistermaat H, Sniatysky J (2010) Geometry of non-holonomically constrained systems. World Scientific Publishing Co. Pte. Ltd.

36. Jean F (2014) Control of non-holonomic system from sub-Riemanniah geometry to motion planning. Springer Briefs in mathematics. Springer, Cham. https://doi.org/10.1007/978-3-319-08690-3
37. Tzafestas SG (2014) Introduction to mobile robot control. Elsevier, Athens, Greece
38. Spong MW, Hutchinson S (2006) Robot modeling and control. Wiley
39. Lui J, Wang X (2011) Advanced sliding mode control for mechanical systems. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20907-9
40. Shtessel Y, Edwards C, Fridman L, Levant A (2014) Sliding mode control and observation. Springer, NY, USA. https://doi.org/10.1007/978-0-8176-4893-0
41. Utkin V, Guldner J, Xin Shi J (2009) Sliding mode control in electro-mechanical systems, 2nd ed. CRC Priss, Boca, Raton. https://doi.org/10.1201/9781420065619
42. Boilco I (2009) Discontinuous control system - frequency domain analysis and design. Birkhäuser Boston, MA. https://doi.org/10.1007/978-0-8176-4753-7
43. Derbel N, Ghommam J, Quan MZ (2017) Application of sliding mode control. Studies in systems, decision and control, vol 79. Springer, Singapore. https://doi.org/10.1007/978-981-10-2374-3
44. Khalil HK (2002) Nonlinear systems, 3rd ed. Prentice Hall Inc.
45. Gutman S (1979) Uncertain dynamic systems: a Lyaponov Min-Max Approach. IEEE Trans Autom Control 24(3):437–449. https://doi.org/10.1109/TAC.1979.1102073
46. Custman S, Leitmann G (1976) Stabilizing feedback control for dynamic systems with bounded uncertainty. In: IEEE conference on decision and control including the 15th symposium on adaptive processes. IEEE. https://doi.org/10.1109/CDC.1976.267709
47. Padhyay BB, Janardhanan S, Spurgeon SK (eds) (2013) Advance in sliding mode control – concept, theory and implementation. Lecture notes in control and information sciences, vol 440. Springer, Berlien, Heidelberg. https://doi.org/10.1007/978-3-642-36986-5
48. Boiko I, Fridman L (2005) Analysis of chattering in continuous sliding mode controllers. IEEE Trans Autom Control 50(9):1442–1446. https://doi.org/10.1109/TAC.2005.854655
49. Boiko I, Fridman L, Pisano A, Usai E (2007) Analysis of chattering in systems with second order sliding modes. IEEE Trans Autom Control 52(11):2085–2102. https://doi.org/10.1109/TAC.2007.908319.
50. Levant A (1993) Sliding order and sliding accuracy in sliding mode control. Int J Control 58(6):1247–1263. https://doi.org/10.1080/00207179308923053
51. Levant A (2001) Universal SISO sliding-mode controllers with finite-time convergence. IEEE Trans Autom Control 46(9):1447–1451. 10.1109/ 9.948475
52. Levant A (2007) Principles of 2-sliding mode design. Automatica 43(4):576–586. https://doi.org/10.1016/j.automatica.2006.10.008
53. Bartolini G, Ferrara A, Utkin VI (2000) On multi input chattering-free second-order siding mode control. IEEE Trans Autom Control 45(9):1711–1717. https://doi.org/10.1109/9.880629
54. Laghrouche S, Polstan P, Lumineou GA (2007) Higher order sliding mode control based on integral sliding surface. Automatica 43(3):531–537. https://doi.org/10.1016/j.automatica.2006.09.017
55. Plestan F, Glumineau A, Laghrouche S (2008) A new algorithm for high-order sliding mode control. Int J Robust Nonlinear Control 18(4):441–453. https://doi.org/10.1002/rnc.1234
56. AL-Samarraie SA (2013) Invariant sets in sliding mode control theory with application to servo actuator system with friction. WSEAS Trans Syst Control 8(2):33–45
57. Isodori A (2000) Nonlinear control systems. Communications and control engineering. Springer, London, UK. https://doi.org/10.1007/978-1-84628-615-5
58. Lee T (2010) Nonholonomic constrained mobile manipulator control using multilayer neural network. School of Information and Mechatronics Gwangju Institute of Science and Technology; 2010.
59. Burton JA, Izinobor AS (1986) Continuous approximation of variable structure control. Int J Syst Sci 17(6):875–885. https://doi.org/10.1080/00207728608926853

Algorithmic Design of Block Backstepping Motion and Stabilization Control for Segway Mobile Robot

Amjad J. Humaidi, Mustafa R. Hameed, Alaq F. Hasan, Abdulkareem Sh. Mahdi Al-Obaidi, Ahmad Taher Azar, Ibraheem Kasim Ibraheem, Ayad Q. Al-Dujaili, Ammar K. Al Mhdawi, and Farah Ayad Abdulmajeed

Abstract This chapter presents a novel control algorithm based on block backstepping control approach for trajectory tracking and balancing of Segway Mobile Robot system. Two versions of block backstepping control design have been presented. The first control algorithmic design is established for nonlinear model of SMR, while the

A. J. Humaidi (✉)
Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq
e-mail: amjad.j.humaidi@uotechnology.edu.iq

M. R. Hameed
State Company for Oil Projects-Ministry of Oil-Iraq, Baghdad, Iraq
e-mail: cse.60448@uotechnology.edu.iq

A. F. Hasan
Technical Engineering College, Middle Technical University, Baghdad, Iraq
e-mail: alaaqf.hasan@mtu.edu.iq

A. Sh. M. Al-Obaidi
Taylor's University, Taylor's Lakeside Campus, Subang Jaya, Selangor, Malaysia
e-mail: abdulkareem.mahdi@taylors.edu.my

A. T. Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia
e-mail: aazar@psu.edu.sa; ahmad.azar@fci.bu.edu.eg; ahmad_t_azar@ieee.org

Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

I. K. Ibraheem
Department of Computer Techniques Engineering, Dijlah University College, Baghdad 10022, Iraq
e-mail: ibraheemki@coeng.uobaghdad.edu.iq

A. Q. Al-Dujaili
Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
e-mail: ayad.qasim@mtu.edu.iq

A. K. Al Mhdawi
Department of Computer Science, Edge Hill University, Ormskirk, UK
e-mail: Al-Mhdaa@edgehill.ac.uk

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_16

other design is devoted for linearized model. The control design objective of underactuated SMR is to keep the inverted pendulum at the up-right position and to make the cart follow desired trajectory. Stability analysis based on zero dynamic criteria and Lyapunov theorem has been conducted such that global asymptotic stability (GAS) for the controlled SMR can be ensured. Moreover, a modification in block backstepping control design for nonlinear model has been added by introducing the integral action in the developed control laws. According to this Integral-based block backstepping control, the robustness and dynamic performance of controlled SMR has been investigated via numerical simulation. Based on numerical simulation, the results that the proposed block backstepping controllers could successfully stabilize both linearized and nonlinear underactuated systems for both regulation and tracking problems. However, the design of linear backstepping controller could keep controlled system stable with maximum initial condition of 0.523 rad, while the nonlinear control design could keep controlled system stable with maximum initial condition of 1.22 rad. Furthermore, the effect of integral action could improve both tracking performance and robust characteristics. The nonlinear block backstepping controller showed better performance in terms of dynamic behavior and error accuracy than the linear version for both regulation and tracking scenarios.

Keywords Block backstepping control · Segway robot · Stability analysis · Zero dynamic

Nomenclature

c_i	The i-th design constant with positive real values.
d_e	The error in the distance between the desired and actual values—m,
D	The separated distance between the wheels' contact patches—m
e	The error between desired and actual signals.
E_i	The actuating control signal for SMR system—V.
F	The applied force to the cart—N
g	The acceleration of gravity—m/s^2
h	The distance between wheel center and the pendulum's center of gravity and—m,
I_p	The chassis's inertia—$kg \cdot m^2$
I_{pdel}	Inertia due to rotation of chassis—$kg \cdot m^2$
$\mathrm{I_w}$	The wheel Inertia $-kg \cdot m^2$
K	Kinetic energy—$Joule$,
k_e	The constant of Back emf,

F. A. Abdulmajeed
Aeronautical Department, College of Technical Engineering, Alfarahidi University,
Baghdad 10070, Iraq
e-mail: f.ayad@uoalfarahidi.edu.iq

k_f	The frictional constant,
k_i	The controller's design parameters,
k_m	The motor's constant—$N \cdot m/A$,
M_p	The chassis's mass—kg,
M_w	The wheel's mass—kg,
R	The resistance of armature coil of DC motor—Ω,
r	The radius of cart's wheel—m,
T	The generated torque due to actuating motors—$N \cdot m$,
V_p	The potential energy—$Joule$,
$V(.)$	The Lyapunov function (L.F.),
X_c	The linear displacement of SMR—m,
z_1	The regulated variable in the block backstepping control design,
α	The stabilization function,
δ_d	The error of heading angle,
λ_i	The design constant of arbitrary positive-real value,
φ	The angle of SMR rotation—rad,
θ	The angular position of pendulum w.r.t vertical axis—rad.

1 Introduction

There are many controls design approaches which have addressed the control of linear mechanical systems [1–10]. These controllers will work properly within specified operating range [11–15]. Due to inherited nonlinearities like saturation, Coulomb friction, hysteresis and dead-zones, the linearized models could not provide exact solution to the control problem. Such severe nonlinearities lead to discontinuous and non-smooth representation of state variables. In addition, faithful linear approximations of these nonlinearities are impossible to obtain. Therefore, in order to satisfy acceptable performances of system that include these nonlinearities, it is necessary to use nonlinear control design instead of linear design version [16–20]. Numerous significant advancements in the design of nonlinear systems for various practical applications have taken place over the last few decades [21–29].

The backstepping control methodology is one of the most important types in nonlinear control design of nonlinear system [11, 30–34]. In this control approach, a recursive algorithm based on Lyapunov theorem has been applied. It uses the powerful feedback linearization in the control design of complex nonlinear systems. This control algorithmic technique is the most commonly control design method in control of nonlinear models. Recently, this nonlinear control design showed highly effectiveness in solution of both regulation and tracking problems in control of many nonlinear systems [35–40]. Breaking the complete controlled system into lower-order subsystems is the main key of this control design method. Each lower-order subsystem becomes a minor design problem to be solved sequentially [41–48].

The backstepping control strategy could solve the problem of cancelling-out all nonlinearities of the system, which was commonly carried out by feedback linearization technique. This control methodology focuses on cancellation of unwanted nonlinearities and keep the useful ones to be utilized throughout the design. However, a prerequisition of this control method is to transform or re-represent the system model into controllable canonical form. The characteristic feature of backstepping control is that it could tackle the mismatched uncertainties, which is a critical problem in most control design approaches. It has been shown that this control algorithm could effectively stabilize nonlinear delayed systems subjected to uncertainties and perturbation. Moreover, the control method has the ability to improve the dynamic performance by appropriate choice of design parameters. The pre-requisition of strictly feedback form of the dynamic system to be controlled, is the main drawback of this control approach. Added to this drawback, the presence of un-avoided phenomenon, named as "explosion of complexity", which is another weak point of this algorithmic control design [49–51].

The researchers Koditschek [52], Sussmann and Sonntag [53], Tsinias [54], Byrnes and IsidoriIn [55] were the pioneers in introducing and developing the concept of integrator backstepping control method in the years 1987 to 1989. Moreover, Sussmann and Kokotovic presented in a "literature of integrator backstepping" the idea of "*Feedback from Positive Real (FPR) transfer functions*" [56]. However, the pioneers in the field of back-stepping control design are Kokotovic and his colleagues. In 1991, the lecture presented by Bode, at the conference "IEEE Conference on Decision and Control", has been devoted to advances in this control field. After that, Kokotovic published this work a year later [57]. In 1992, Kanellakopoulos et al. proposed an efficient mathematical toolkit for developing and implementing the control algorithm based on bacstepping control laws for many nonlinear systems and for various applications [58]. He then proposed the concept of combining the backstepping control laws with integral action, which has signficantly enhanced the steady-state and dynamic charcteristics of controlled system dynamics based on this control hybridization [59].

Many researches and textbooks on the backstepping control approach have appeared during these years. In the field of backstepping control, one of the most famous references is the book entitled "Adaptive and nonlinear backstepping Design" which had been authored by Kritic and issued in 1995. This reference has attracted the intention of most researchers who are interesting in this trend of control design. In this work, the structure of simplified approaches has been presented for control purposes and parameter's adaptation. Moreover, the modular approach has been developed and used in the design based on full-state feedback control and output feedback control (observer-based backstepping control) [60].

Fossen and Berge introduced the concept of vectorial backstepping control in 1997. This modification in backstepping control approach had considerably simplified the analysis and control design by simply taking the benefits of structural properties of nonlinear Multi-Input Multi-Output dynamic system in control design [61]. In 1999, a progress report on the advanced in backstepping control had been presented by Kokotovic at the IFAC World Congress, which took place in Beijing [62].

Because there are many mechanical systems in real-world applications that are not fully actuated or underactuated, like underwater robots, rotary inverted pendulum, free-flying space robots, parallel robots, Quadcopters, and surface vessels, it necessitated the development of control design to tackle such types of dynamic systems to meet the control objectives [63, 64]. The new advances in field of backstepping control was the development of what is so called "block backstepping control". This technique was developed by Rudra in 2013, to solve the control problem of dynamic systems which have non-strictly feedback form. This control approach had been applied to many applications with underactuated systems, like Pendubot, Vertical Take-Off and Landing aircraft (VTOL), Acrobat, Inertia Wheel Pendulum (IWP), underactuated Surface Vessels (USV). In this control technique, the non-strictly feedback dynamic equations are transformed into a block of strictly feedback form by an appropriate choice of regulated variables such that the global asymptotic stability of controlled underactuated system is guranteed [65, 66].

The motivation behind this study was that the model of underactuated system, represented by the Segway Mobile Robot (SMR), is high complexity and it is not strictly feedback form. As such, the presence of unknown uncertainites and disturbances may lead to instability problems. Therefore, the block backstepping control design has been presented to solve this problem. However, the block backstepping control design of this underactuated system is a challenging problem and hence the aim of this work is to develop the control algorithm based on block backstepping control methodology.

Therefore, the main contribution of this study is to develop a novel block-backstepping design for solving the control problem of non-strict under-actuated systems and dealing with unknown disturbances. One can summarize the contribution of the work as follows:

- The zero-dynamics concept has been introduced to transform the non-strict form system into a reduced-order strict feedback form,
- Stability analysis based on zero dynamic analysis and Lyapunov theorem has been conducted to ensure global asymptotic stability (GAS) for both nonlinear and linearized model of controlled SMR.
- The control laws have been developed to establish block-backstepping control algorithm to cope with unknown uncertainties and disturbances.
- The integral action has been fuzed with control laws of nonlinear block backstepping controller to enhance both dynamic performance and robust chateristics of proposed controller.

This chapter is arranged in the following order: Sect. 2 introduces the related works. Section 3 presents the dynamic modeling of the under-actuated system, represented by SRM system. In Sect. 4. a novel block-backstepping design algorithms are developed for both linear and linearized model of SMR system. The simulated results are presented in Sect. 5. In Sect. 6, conclusions based on the simulated results are drawn.

2 Related Works

In recent years, robotics researchers that are familiar with tractor robots have attempted to concentrate their efforts on a number of the robots' features, such as path tracking, stabilization, steering control, dynamics modeling, and traction [67–77]. The application of a Segway-like mobile robot has attracted many researchers' interests in control and stabilization. In what follows, many related and recent works are briefly interviewed, which highlight different control schemes for this type of robotic system:

Music et al. in [78] presented the design of a networked controller based on a model predictive approach to remotely control a two-wheeled inverted pendulum robot using W-LAN communication. The controller is responsible for keeping the pendulum at the upright position in the presence of uncertainties and delays due to incorporating wireless communication in the control loop. The proposed networked control scheme employs model predictive techniques and deliberately extends delays in order to make them constant and deterministic.

Key et al. in [79] presented a robust control design for a two-wheeled inverted pendulum mobile robot moving on uniform slopes. For stabilizing the body posture of the mobile robot on slopes, a 2-degree-of-freedom (DOF) model, obtained by restricting the spinning motion in a 3-DOF motion dynamic equation, was used. In order to design a robust controller guaranteeing the stability of the robot driving on slopes, a sliding mode control was proposed based on the theory of variable structure systems and design a sliding surface using the theory of linear quadratic regulation (LQR).

Nawawi et al. in [80] designed a robust controller for a two-wheel inverted pendulum system based on proportional integral sliding mode control (PISMC). The objective of the controller is to achieve robust stabilization of the inverted pendulum and to reject the applied disturbance.

Mansour et al. in [81] proposed control architecture based on a predictive approach for a non-holonomic Two-Wheeled Inverted Pendulum Mobile Robot. The control structure consists of two loops. The first-speed inner loop is based on the state feedback technique to ensure the stability of the inverted structure of the robot. The second loop is devoted to assisting the robot's navigation along the desired trajectory. This study utilized image-based techniques for obstacle recognition and avoidance.

Villacrés et al. in [82] conducted the design and real-time implementation of three control schemes to stabilize the two-wheeled inverted pendulum. The proposed control strategies are the proportional-Integral-Derivative (PID) control, Linear-Quadratic Regulator (LQR) control, and Sliding Mode Control. The study made a comparison in performance in terms of trajectory tracking error, rejection of applied disturbance, and robustness against uncertainties in model parameters.

Goher and Fadlallah in [83] developed two control designs based on PID controller and fuzzy logic control (FLC) for stabilizing five degrees of freedom (DOF) two-wheeled robot.

Odry et al. in [84] presented a design of a cascade control structure based on fuzzy logic (FL) control to stabilize of under-actuated two-wheeled mobile pendulum system. The design can perform trajectory tracking and reduce body oscillation by applying three FL controllers. Particle Swarm Optimization (PSO) has been applied for tuning the elements of FL controllers to improve the dynamic performance of both mobile and pendulum. The proposed controller could eliminate the vibration in motion and limit the energy consumption of actuating power.

Feriyonika and Hidayat in [85] addressed the optimality of states and power consumption for the Two-Wheeled Balancing Robot utilizing a Linear Quadratic Regulator (LQR). In order to estimate noise-corrupted states, the Kalman filter has been applied to eliminate the noise in the measure states, which are acquired from the Accelero-Gyro sensor. The proposed controller has been verified and assessed based on an experimental setup, and the results showed the effectiveness of the controller in terms of stability and disturbance rejection capability.

Isdaryani et al. in [86] developed Model Reference Adaptive Control (MRAC) to stabilize and control a two-wheeled robot system subjected to uncertainty and external disturbance. The study conducted a stability analysis of a controlled system based on the Lyapunov theorem to deduce the adaptive laws for estimating uncertainties. The convergence of both tracking and estimation errors has been proved experimentally.

Nabil in [87] applied an intelligent controller based on Supervised Feed-Forward Neural Networks (SFFNN) to stabilize a two-wheel inverted pendulum in a real-time environment. The control design is firstly initiated by using state-feedback control to stabilize the nonlinear system. Then, the real-time measurements of system states and motor control signals from the state-feedback control phase have been used to train the SFFNN structure based on backpropagation Levenberg–Marquardt optimization.

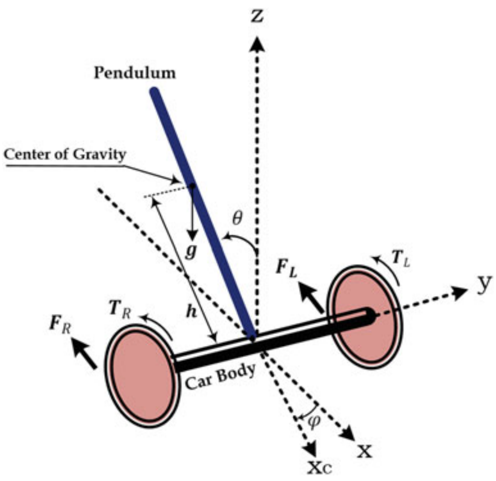
Kharola in [88] has designed a fuzzy logic controller to stabilize the robot chassis during the motion of a two-wheeled robot (SMR). The controller is designed to stabilize the system such as to achieve desired transient parameters like peak overshoot, steady-state error, and rise time.

Zhou et al. in [89] presented a robust controller based on optimal integral sliding mode control (SMC) for following a planar motion of a two-wheeled inverted pendulum subjected to an input delay. The control objective is to keep the pendulum stabilized in the upright position during the mobile motion. The optimal SMC has been designed based on LQR to stabilize the inverted pendulum while tracking the motion of the mobile robot.

3 Mathematical Model of Segway Mobile Robot System

Figure 1 shows the configuration design of Segway Mobile Robot. The system consists of three rigid parts: The wheels, chassis and inverted pendulum. In order to move the robot directly, the robot is supported by two separate motors such that

Fig. 1 The configuration of Segway mobile robot system [35]



they actuate each wheel on the same axle. According to Fig. 1, $\mathrm{T_L}$ and $\mathrm{T_R}$ represent the torque generated by left and right DC motors, respectively. $\mathrm{F_L}$ ($\mathrm{F_R}$) denotes the reaction force between the chassis and the left (right) wheel. The angles θ and φ denote the pendulum angular position and the cart rotational angle, respectively. The parameters X_c and h represent displacement of chassis motion and the distance between the center of gravity of pendulum mass and the wheel, respectively [90].

The dynamic equations of motion have been developed based on Newton–Euler formulation. The total dynamic analysis of SMR model can be decomposed into four separate sub-dynamic systems. These subsystems include dynamic of DC motor, dynamic of the robot wheel, dynamic of robot chassis, dynamic of inverted pendulum. The total dynamic model of SMR is the sum-up of all these sub-dynamic systems.

3.1 Dynamic Model of DC Motor

The DC motor is responsible for actuating the robot wheels such as to move the robot chassis at desired trajectory and in the time to balancing the inverted pendulum in the up-right position. In Fig. 2, the circuit diagram of DC motor is indicated, where L and R represent the inductance and resistance of armature coil, respectively. The variables V_e and V_a denotes the back emf voltage and the terminal voltage applied to Dc motor, I_R represnts the inertia of motor armature. The variables τ_m and τ_a represent the developed motor torque and the load friction torque, respectively, and the coefficient k_f denotes frictional constant.

The current i is generated through the coil of motor armature when a voltage is applied to the terminals of motor. According to this current, a torque τ_m will be produced which is directly related to the current flowing into motor windings:

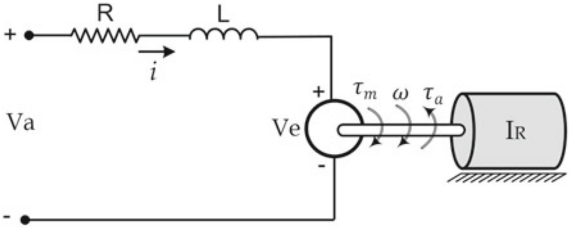


Fig. 2 The circuit diagram of actuating DC motor attached to SMR wheel [91]

$$\tau_m = k_m \cdot i \tag{1}$$

where k_m denotes the torque constant of the motor. A back emf voltage will be generated upon the rotation of armature winding, which is proportional to the velocity of motor shaft and expressed by the following equation:

$$V_e = k_e \omega \tag{2}$$

where, ω and k_e represents the speed of motor shaft and the back-emf constant, respectively. Using Kirchhoff's voltage law based on Fig. 2, the following equation can be obtained

$$V_e = V_a - Ri - L\frac{di}{dt} \tag{3}$$

Using Newton's 1st law of motion, the torque generated by motor is equal to sum of load torques exerted on the shaft of motor,

$$I_R \dot{\omega} = \tau_m - \tau_a - k_f \omega \tag{4}$$

If the motor friction and inductance of armature coil are neglected, one can obtain the following formulas:

$$i = -\frac{k_e \omega}{R} + \frac{V_a}{R} \tag{5}$$

$$\dot{\omega} = \frac{k_m i}{I_R} - \frac{\tau_a}{I_R} \tag{6}$$

The final dynamic model of DC motor can be obtained using Eq. (5) and Eq. (6),

$$\dot{\omega} = -\frac{k_m k_e \omega}{I_R R} + \frac{k_m V_a}{I_R R} - \frac{\tau_a}{I_R} \tag{7}$$

3.2 Dynamic Model of Wheel

Figure 3 shows the diagram of robot wheel. The forces F, F_g and F_p represents the reaction force between the chassis and the wheel, the friction force between the ground and the wheel, the normal force of the wheel on the ground, respectively. On the hand, the variables θ_w, M_w and T [35]. In the analysis of wheel dynamics, two motions have been detected: rotational motion and translation motion [92].

The following equation of linear motion has been obtained by applying Newton's 2nd law of linear motion [33]:

$$M_w\ddot{x} = F_g - F \tag{8}$$

On the other hand, applying Newton's first law of inertia, the following equations can be deduced,

$$I_w \cdot \ddot{\theta}_w = T - F_g \cdot r \tag{9}$$

where, r and I_w represent the radius of wheel and rotational inertia of the wheel, respectively. The developed torque of motor can be expressed as;

$$\tau_m = I_R \cdot \dot{\omega} + \tau_a \tag{10}$$

One can rewrite Eq. (7) as,

$$I_R \cdot \dot{\omega} = -\frac{k_m k_e \dot{\theta}_w}{R} + \frac{k_m V_a}{R} - \tau_a \tag{11}$$

Letting $T = \tau_m$, using Eqs. (9) and (10), one can express Eq. (9) as follows:

$$F_g = -\frac{k_m k_e \dot{\theta}_w}{R \cdot r} + \frac{k_m V_a}{R \cdot r} - \frac{I_w \cdot \ddot{\theta}_w}{r} \tag{12}$$

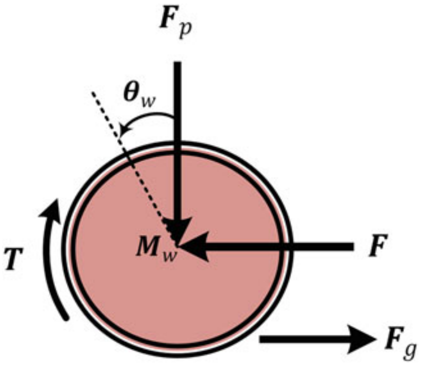


Fig. 3 Diagram of free wheel [91]

One can express the dynamic motions of left and right wheels as follows:

$$M_w\ddot{x}_L = -\frac{k_m k_e \dot{\theta}_L}{Rr} + \frac{k_m V_{aL}}{Rr} - \frac{I_w \ddot{\theta}_L}{r} - F_L \tag{13}$$

$$M_w\ddot{x}_R = -\frac{k_m k_e \dot{\theta}_R}{Rr} + \frac{k_m V_{aR}}{Rr} - \frac{I_w \ddot{\theta}_R}{r} - F_R \tag{14}$$

where, V_{aL} and V_{aR} denotes the voltage applied on the terminal of left DC motor, and the terminal of right DC motor, respectively. Equations (13) and (14) can be rewritten based on linear transformation,

$$M_w\ddot{x}_L = -\frac{k_m k_e \dot{x}_L}{Rr^2} + \frac{k_m V_{aL}}{Rr} - \frac{I_w \ddot{x}_L}{r^2} - F_L \tag{15}$$

$$M_w\ddot{x}_R = -\frac{k_m k_e \dot{x}_R}{Rr^2} + \frac{k_m V_{aR}}{Rr} - \frac{I_w \ddot{x}_R}{r^2} - F_R \tag{16}$$

One can combine Eqs. (15) and (16) into Eq. (17) by utilizing the expression $x_c = (x_R + x_L)/2$,

$$2\left(M_w + \frac{I_w}{r^2}\right)\ddot{x}_c = -\frac{2k_m k_e \dot{x}_c}{Rr^2} + \frac{k_m (V_{aR} + V_{aL})}{Rr} - (F_R + F_L) \tag{17}$$

3.3 Dynamic Analysis of Inverted Pendulum

Figure 4 describes the schematic diagram of Inverted pendulum attached at the surface of robot chassis. In the horizontal direction, the sum of forces is established based on Newton's 2$^{\text{nd}}$ law of motion [34],

$$M_p\ddot{x}_c = M_p h\dot{\theta}^2 \sin\theta - M_p h\ddot{\theta}\cos\theta + (F_R + F_L) \tag{18}$$

The total of forces perpendicular to the pendulum, as defined by Newton's second law of motion, is is expressed as

$$M_p\ddot{x}_c\cos\theta = (F_{PR} + F_{PL})\sin\theta + (F_R + F_L)\cos\theta - M_p g\sin\theta - M_p h\ddot{\theta} \tag{19}$$

where, M_p is the chassis mass, and the left and right reaction forces between the chassis and the wheel are represented by the variables F_{PL} and F_{PR}, respectively. The sum of moments around mass center pendulum is expressed as;

$$I_p\ddot{\theta} = -(F_R + F_L)h\cos\theta - (F_{PR} + F_{PL})h\sin\theta - (T_R + T_L) \tag{20}$$

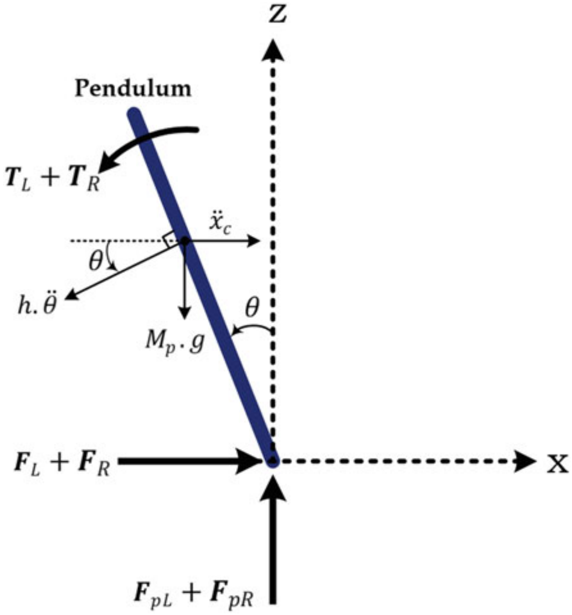


Fig. 4 Schematic diagram of Inverted pendulum analysis [91]

If E_1 is expressed as $E_1 = V_{aR} + V_{aL}$, then the following equation can be deduced:

$$-(M_p \ddot{x}_c \cos\theta) = \left(I_p + M_p h^2\right) \cdot \ddot{\theta} - \frac{2k_m k_e \dot{x}_c}{Rr} + M_p gh \sin\theta + \frac{k_m}{R} E_1 \quad (21)$$

$$\frac{k_m}{Rr} E_1 = \left(2M_w + \frac{2I_w}{r^2} + M_p\right) \cdot \ddot{x}_c + \frac{2k_m k_e \dot{x}_c}{Rr^2} + M_p h \ddot{\theta} \cos\theta - M_p h \dot{\theta}^2 \sin\theta \quad (22)$$

According to Fig. 4, the moment of inertia due to rotation of robot chassis can be expressed in terms of the reaction forces generated by the wheels

$$\ddot{\varphi} I_{pdel} = \frac{D(F_L - F_R)}{2} \quad (23)$$

where, D denotes the distance between the wheels' contact patches, I_{pdel} represents the rotational inertia of chassis. The equation for heading control is described by,

$$F_L - F_R = \frac{k_m \cdot V_{aR}}{Rr} - \frac{k_m \cdot V_{aL}}{Rr} \quad (24)$$

Let $E_2 = V_{aR} - V_{aL}$, then Eq. (23) becomes

$$\ddot{\varphi} = -\frac{k_m D}{2RrI_{pdel}} \cdot E_2 \quad (25)$$

In order to express the dynamic equation in state variable form, one has to assign each physical quantity by corresponding state variables such as the following state vector is obtained:

$$\begin{bmatrix} q_1 & p_1 & q_2 & p_2 & q_3 & p_3 \end{bmatrix} = \begin{bmatrix} x_c & \dot{x}_c & \theta & \dot{\theta} & \varphi & \dot{\varphi} \end{bmatrix}$$

Accordingly, the following state variable representation will be obtained

$$\begin{aligned}
\dot{q}_1 &= p_1 \\
\dot{p}_1 &= \frac{-2k_m k_e\left(b + M_p hr\cos q_2\right)}{Rr^2 w} p_1 + \frac{M_p^2 g h^2 \sin q_2 \cos q_2}{w} \\
&\quad + \frac{k_m\left(b + M_p hr\cos q_2\right)}{Rrw} E_1 + \frac{M_p h p_2^2 b \sin q_2}{w} \\
\dot{q}_2 &= p_2 \\
\dot{p}_2 &= \frac{2k_m k_e\left(ar + M_p h\cos q_2\right)}{Rr^2 w} p_1 - \frac{M_p^2 h^2 p_2^2 \sin q_2 \cos q_2}{w} \\
&\quad - \frac{k_m\left(ar + M_p h\cos q_2\right)}{Rrw} E_1 - \frac{M_p g h a \sin q_2}{w} \\
\dot{q}_3 &= p_3 \\
\dot{p}_3 &= -\frac{k_m D}{2RrI_{pdel}} E_2
\end{aligned} \tag{26}$$

where, $a = 2M_w + 2I_w/r^2 + M_p$, $b = I_p + M_p h^2$, $w = ab - M_p^2 h^2 \cos^2 x_3$

4 Block Backstepping Control Design of Segway Mobile Robot

The block backstepping control design of SMR system has to perform two main goals. The stabilization of pendulum at up-right position is the first objective of proposed controller, while trajectory tracking control of the robot cart is the other objective. Designing controllers that will ensure both goals is the main topic of this section. Firstly, the block backstepping control algorithm will be developed for the linearized model of SMR system. In this case, the controller is called as “linear blockstepping controller”. Secondly, the block backstepping control algorithm will be designed to the nonlinear system of SMR and the controller is named as “nonlinear block backstepping controller”.

4.1 Block Back-Stepping for Linearized Model

To start the control design based on block backstepping control scheme, one has to linearize the nonlinear system described by Eq. (26) about the equilibrium point $(q, p) = (0, 0)$. the resultant model is given by,

$$\begin{gathered}
\dot{q}_1 = p_1 \\
\dot{p}_1 = -\frac{2k_m k_e\big(b - M_p hr\big)}{Rr^2\beta} p_1 + \frac{M_p^2 gh^2}{\beta} q_2 + \frac{k_m\big(b - M_p hr\big)}{Rr\beta} E_1 \\
\dot{q}_2 = p_2 \\
\dot{p}_2 = \frac{2k_m k_e\big(ar - M_p h\big)}{Rr^2\beta} p_1 - \frac{k_m\big(ar - M_p h\big)}{Rr\beta} E_1 + \frac{M_p hga}{\beta} q_2 \\
\dot{q}_3 = p_3 \\
\dot{p}_3 = -\frac{k_m D}{2RrI_{pdel}} E_2
\end{gathered} \tag{27}$$

where, $\beta = ab - M_p^2 h^2$.

The block backstepping-based algorithm is applied to the linearized model such as to fulfill the required control objectives within zone of stabilization surrounding the equilibrium point. There two scenarios will be adopted to achieve the algorithmic block backstepping design technique [93];

Scenario I

Step 1: The algorithm is started by introducing the regulated variable z_1, which is defined by:

$$z_1 = q_1 - q_{1d} \tag{28}$$

where, q_{1d} is the reference displacement. The time-derivative of z_1 is given by,

$$\dot{z}_1 = p_1 - \dot{q}_{1d} \tag{29}$$

In order to realize the desired value of virtual input, p_2, for the first sub-system, one can define the following suitable stabilizing function;

$$\alpha_1 = -c_1 z_1 + \dot{q}_{1d} \tag{30}$$

where, the design constant c_1 has a positive real value. In light of this, the appropriate state variable error can be expressed as;

$$z_2 = p_1 - \alpha_1 \tag{31}$$

Step 2: Taking the time-derivative of error z_2 to have:

$$\dot{z}_2 = \dot{p}_1 - \dot{\alpha}_1$$

$$\dot{z}_2 = \psi_1 E_1 + \phi_1 + c_1 \dot{z}_1 \tag{32}$$

where,

$$\psi_1 = \frac{k_m \left(b - M_p hr\right)}{Rr\beta}$$

$$\phi_1 = -\frac{2k_m k_e \left(b - M_p hr\right)}{Rr^2\beta} p_1 + \frac{M_p^2 gh^2}{\beta} q_2 - \ddot{q}_{1d}$$

One can express the desired dynamics of z_2 by the following dynamic equation;

$$\dot{z}_{2d} = -z_1 - c_2 z_2 \tag{33}$$

For solving the control signal, Eq. (33) is substituted into Eq. (32) to obtain the linear control law, which responsible for satisfying the desired behaviors of state variables z_1 and z_2,

$$E_1 = \psi_1^{-1}\left(-\phi_1 - \left(1 - c_1^2\right) z_1 - (c_1 + c_2) z_2\right) \tag{34}$$

where, the design constant c_2 has a positive real value.

Step 3: The regulated variable z_3 can be introduced and defined by

$$z_3 = q_3 - q_{3d} \tag{35}$$

Taking the time derivative of z_3,

$$\dot{z}_3 = p_3 - \dot{q}_{3d} \tag{36}$$

where, p_3 represents the virtual control variable. One can choose the stabilizing function to be

$$\alpha_2 = -c_3 z_3 + \dot{q}_{3d} \tag{37}$$

where, c_3 is a design parameter with real positive value. The variable for error state is selected as

$$z_4 = p_3 - \alpha_2 \tag{38}$$

Step 4: Taking the time-derivative of Eq. (38), one can have

$$\dot{z}_4 = \dot{p}_3 - \dot{\alpha}_2 \tag{39}$$

$$\dot{z}_4 = \psi_2 E_2 + \phi_2 + c_3\dot{z}_3 \tag{40}$$

where,
$\psi_2 = -\frac{k_m D}{2RrI_{pdel}}$, $\phi_2 = -\ddot{q}_{3d}$

One can express the desired dynamics of z_4 using

$$\dot{z}_{4d} = -z_3 - c_4 z_4 \tag{41}$$

For solving the control signal, Eq. (41) is substituted into Eq. (40) to obtain the linear control law, which responsible for satisfying the desired behaviors of state variables z_1 and z_2,

$$E_2 = \psi_2^{-1}\left(-\phi_2 - \left(1 - c_3^2\right)z_3 - (c_3 + c_4)z_4\right) \tag{42}$$

where, the design constant c_4 has positive-real value. The stability of controlled system will be analyzed by firstly selecting an appropriate Lyapunov function (L. F.)

$$V = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2}z_3^2 + \frac{1}{2}z_4^2 \tag{43}$$

Taking the time derivative of Eq. (43), the following equation is deduced:

$$\dot{V} = -c_1 \cdot z_1^2 - c_2 \cdot z_2^2 - \mathrm{c}_3 \cdot \mathrm{z}_3^2 - \mathrm{c}_4 {\cdot} \mathrm{z}_4^2 \tag{44}$$

According to Eq. (44), $\dot{V} < 0$ (negative definite), then in light of Lyapunov stability theorem the error states variables, represented by z_1, z_2, z_3, and z_4, will eventually converge to zero. In order to ensure globally asymptotic convergence of equilibrium states, the criteria of zero dynamic have to be applied. After simple calculation, the zero dynamics of the system can be represented by the following state equation,

$$\begin{bmatrix} \dot{q}_2 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & 0 \end{bmatrix}\begin{bmatrix} q_2 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma \end{bmatrix} \tag{45}$$

where,

$$a_1 = \frac{M_p gh}{b - M_p hr}, \quad \gamma = -\frac{\big(ar - M_p h\big)\ddot{q}_{1d}}{b - M_p hr}$$

According to Eq. (45), It is evident that the matrix is not Hurwitz, which means that the system is unstable. This means that the states q_2 and p_2 cannot converge to zero. To solve the problem of unacceptable phenomenon, redesigning of regulated variables has to be conducted.

Scenario II

Step 1: Firstly, one has to define the regulated variable z_1 as

$$z_1 = e + k_2(M_1\dot{e} + M_2 p_2) + k_1 q_2 \tag{46}$$

where the error and its derivative are expressed by,

$$e = q_1 - q_{1d} \tag{47}$$

$$\dot{e} = p_1 - \dot{q}_{1d} \tag{48}$$

$$M_1 = ar - M_p h, \quad M_2 = b - M_p hr$$

where, q_{1d} denotes the desired path of cart, k_1 and k_2 are design parameters of real value, and e is the difference between the actual and desired cart's displacement. The addition of variables p_2 and $\dot{e}$ is intended to eliminating the unwanted phenomenon. The derivative of z_1 is then calculated as,

$$\dot{z}_1 = \dot{e} + k_2(M_1\ddot{e} + M_2\dot{p}_2) + k_1 p_2 \tag{49}$$

After simple calculations, Eq. (49) results in

$$\dot{z}_1 = \dot{e} - k_2 M_1 \ddot{q}_{1d} + k_2\left(\frac{M_p gh}{\beta}\big(M_1 M_p h + M_2 a\big)\right)q_2 + k_1 p_2 \tag{50}$$

For the first subsystem, one can realize the desired value of the virtual input by defining an appropriate stabilizing function. By using this stabilization function, the original state model of the system can be converted into a strict feedback form. To serve this purpose, the following stabilizing function has been chosen,

$$\alpha_1 = -k_1 p_2 + k_2 M_1 \ddot{q}_{1d} - k_2\left(\frac{M_p gh}{\beta}\big(M_2 a + M_1 M_p h\big)\right)q_2 - c_1 z_1 - \lambda_1 \chi_1 \tag{51}$$

where, the design constants c_1 and λ_1 have positive-real value. One firstly has to define the integral action of regulated variable z_1,

$$\chi_1 = \int_0^t z_1 dt \tag{52}$$

Defining z_2 to be the error in state variable,

$$z_2 = \dot{e} - \alpha_1 \tag{53}$$

The time derivative of variable z_1 can be expressed as,

$$\dot{z}_1 = z_2 - c_1 z_1 - \lambda_1 \chi_1 \tag{54}$$

Step 2: Then, the time-derivative of z_2 is taken

$$\dot{z}_2 = \ddot{e} - \dot{\alpha}_1 \tag{55}$$

$$\dot{z}_2 = \psi_1 E_1 + \phi_1 + c_1 \dot{z}_1 + \lambda_1 z_1 \tag{56}$$

where,

$$\psi_1 = \frac{k_m}{Rr\beta}(M_2 - k_1 M_1)$$

$$\begin{aligned}\phi_1 = &-\frac{2\,k_m\,k_e}{R\,r^2\,\beta}(M_2 - k_1\,M_1)\,p_1 + \frac{M_p\,g\,h}{\beta}\big(M_p\,h + k_1\,a\big)\,q_2 \\ &- \ddot{q}_{1d} - k_2\,M_1\,\dddot{q}_{1d} + \frac{k_2\,M_p\,g\,h}{\beta}\big(M_p\,M_1\,h + M_2\,a\big)\,p_2\end{aligned}.$$

One can define the following desired dynamics of z_2

$$\dot{z}_{2d} = -z_1 - c_2 z_2 \tag{57}$$

To satisfy the desired dynamics of z_1 and z_2, the linear control law has to be derived by substituting Eq. (57) into Eq. (56)

$$E_1 = \psi_1^{-1}\big(-\phi_1 - \big(1 - c_1^2 + \lambda_1\big)z_1 + \lambda_1 c_1 \chi_1 - (c_1 + c_2)z_2\big) \tag{58}$$

where, ψ_1 is invertible and the design constant c_2 has positive-real value.

Step 3: One can define the first regulated variable z_3 as

$$z_3 = q_3 - q_{3\mathrm{d}} \tag{59}$$

Then, the time derivative of z_3 is taken

$$\dot{z}_3 = p_3 - \dot{q}_{3d} \tag{60}$$

The stabilizing function is chosen by assigning the state p_3 as the virtual control variable

$$\alpha_2 = -c_3 z_3 + \dot{q}_{3d} \tag{61}$$

where, the design constant c_3 has a positive-real value. Accordingly, the error state variable is established

$$z_4 = p_3 - \alpha_2 \tag{62}$$

Step 4: Then, the time-derivative of z_4 is taken as

$$\dot{z}_4 = \dot{p}_3 - \dot{\alpha}_2 \tag{63}$$

$$\dot{z}_4 = \psi_2 E_2 + \phi_2 + c_3 \dot{z}_3 \tag{64}$$

where,

$$\psi_2 = -\frac{k_m D}{2 R r I_{pdel}}, \phi_2 = -\ddot{q}_{3d}$$

The desired dynamics of z_4 can be expressed as

$$\dot{z}_{4d} = -z_3 - c_4 z_4 \tag{65}$$

To satisfy the desired dynamics of z_3 and z_4, one has to use Eq. (65) and Eq. (64) to yield the control law for the linearized model

$$E_2 = \psi_2^{-1}\left(-\phi_2 - \left(1 - c_3^2\right)z_3 - (c_3 + c_4)z_4\right) \tag{66}$$

where,c_4 represents design constant of positive-real value and ψ_2 is invertible. In order to analyze the stability of controlled system, the following L. F. has to be chosen:

$$V = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2}z_3^2 + \frac{1}{2}z_4^2 + \frac{1}{2}\lambda_1 \chi_1^2 \tag{67}$$

After simple manipulation, the time derivative of Eq. (67) becomes

$$\dot{V} = -c_1 z_1^2 - c_2 z_2^2 - \mathrm{c}_3 \mathrm{z}_3^2 - \mathrm{c}_4 \mathrm{z}_4^2 \le 0 \tag{68}$$

According to Eq. (68), the inequality $V(t) \le V(0)$ is satisfied. Hence, it can be concluded that the states z_1, z_2, z_3, z_4, and χ_1 are bounded, and consequently the boundness of $\dot{z}_1$, $\dot{z}_2$, $\dot{z}_3$ and $\dot{z}_4$ are proved. Taking the second time-derivative of L. F., one can obtain

$$\ddot{V} = -2c_1 z_1 \dot{z}_1 - 2c_2 z_2 \dot{z}_2 - 2c_3 z_3 \dot{z}_3 - 2c_4 z_4 \dot{z}_4 \tag{69}$$

It is clear that $\ddot{V}$ is bounded since all variables constituting Eq. (69) are bounded; that is, the variables z_1, z_2, z_3, z_4, $\dot{z}_1$, $\dot{z}_2$, $\dot{z}_3$ and $\dot{z}_4$ are bounded. Based on Barbalat's Lemma, one can show that if $t \to \infty$, all the variables z_1, z_2, z_3 and z_4 eventually converge to zero.

By using simple calculations, one can obtain the following zero dynamics of the system:

$$\begin{bmatrix} \dot{q}_2 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \begin{bmatrix} q_2 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma \end{bmatrix} \tag{70}$$

where,

$$\gamma = \frac{M_1}{M_2 - k_1 M_1}(-\ddot{q}_{1d} - k_2 M_1 \ddot{q}_{1d}),\ a_1 = \frac{M_p h g}{M_2 - k_1 M_1},$$
$$a_2 = \frac{M_p g h M_1 k_2}{M_2 - k_1 M_1}$$

If a_1 and a_2 have values less than zero, the system described Eq. (70) has Hurwitz matrix and the stability can be guaranteed and consequently the states q_2 and p_2 will converge to zero as $t \to \infty$. To reach this end, the constants k_1 and k_2 have to be selected so that the following inequalities have to be satisfied;

$$k_1 > \frac{M_2}{M_1}, k_2 > 0$$

According to above argument, one can conclude that the proposed control law could ensure global stabilization of controlled SMR system based on Block Backstepping Design. Figure 5 shows the realization of block-backstepping controller for linearize model of SMR system.

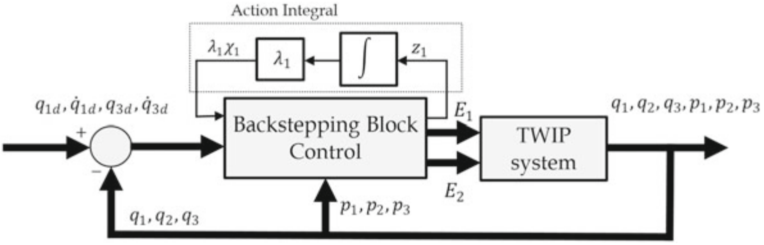


Fig. 5 Closed-loop of SMR system

4.2 Block Backstepping Design of Nonlinear Model

In order to develop a block backstepping controller for linearized model of Eq. (27), stability analysis based on Lyapunov method has been applied. However, if one uses the same control laws, described by Eqs. (58) and (66), to the nonlinear system given by Eq. (26), then the controller performs well within the stabilization zone (around the equilibrium points). As such, a novel block backstepping controller has to be synthesized to cope with the nonlinear system and to fulfill the control objectives irrespective of how the solution is far from the equilibrium points. Therefore, this section addresses the extension of design conception applied for the case of linearized model. However, the steps of previous algorithm will be followed concerning the choice of regulated variables. In what follows, the steps for this new design procedure can be described and analyzed in details:

Step 1: The regulated variable has been chosen according to the following equation

$$z_1 = e + k_1 q_2 + k_2(M_1 \dot{e} + M_2 p_2) \tag{71}$$

where,

$$\mathrm{e} = q_1 - q_{1d} \tag{72}$$

$$\dot{e} = p_1 - \dot{q}_{1d} \tag{73}$$

$$M_1 = ar + M_p h \cos q_2, \; M_2 = b + M_p hr \cos q_2$$

where, the design constants k_1 and k_2 have real values. The time-derivative of state variable z_1 is calculated as:

$$\dot{z}_1 = \dot{e} + k_1 p_2 + k_2\left(\dot{M}_1 \dot{e} + M_1 \ddot{e} + M_2 \dot{p}_2 + \dot{M}_2 p_2\right) \tag{74}$$

Equation (74) can be arranged to be written as follows;

$$\dot{z}_1 = \left(1 + k_2 \dot{M}_1\right)\dot{e} + \left(k_1 + k_2 \dot{M}_2\right)p_2 + k_2\left(M_p h p_2^2 r \sin q_2 - M_p h g \sin q_2 - M_1 \ddot{q}_{1d}\right) \tag{75}$$

In this design procedure, the stabilizing function can be chosen as;

$$\alpha_1 = -\left(k_1 + k_2 \dot{M}_2\right)p_2 - k_2\left(M_p h p_2^2 r \sin q_2 - M_p h g \sin q_2 - M_1 \ddot{q}_{1d}\right) - c_1 z_1 - \lambda_1 \chi_1 \tag{76}$$

where, the design constants c_1 and λ_1 have positive real-value. The regulated variable's integral action is described as

$$\chi_1 = \int_0^t z_1 dt \tag{77}$$

The following equation can be assigned to error variable z_2:

$$z_2 = \left(1 + k_2\dot{M}_1\right) \cdot \dot{e} - \alpha_1 \tag{78}$$

Next, the time derivative of variable z_3 is taken

$$\dot{z}_3 = z_4 - c_3 z_3 - \lambda_2 \chi_1 \tag{79}$$

Step 2: If time-derivative of z_2 is taken, one can have

$$\dot{z}_2 = \left(1 + k_2\dot{M}_1\right)\ddot{e} - \dot{\alpha}_1 + k_2\ddot{M}_1\dot{e} \tag{80}$$

After simple calculation and reshuffling, Eq. (80) becomes

$$\dot{z}_2 = \phi_1 + \psi_1 E_1 + \lambda_1 z_1 + c_1\dot{z}_1 \tag{81}$$

where,

$$\psi_1 = \frac{k_m}{Rrw}\left(M_2\left(1 + k_2\dot{M}_1\right) - M_1k_1 - M_1k_2\left(M_p h p_2 r \sin q_2 - M_p h\dot{e}\sin q_2 + \dot{M}_2\right)\right) \tag{82}$$

$$\begin{aligned}\phi_1 =& \left(1 + k_2\ \dot{M}_1\right)\left(-\ddot{q}_{1d} - \frac{2\ k_m\ k_e\ M_2}{R\ r^2\ w}p_1 + \frac{M_p^2\ g\ h^2 \sin q_2 \cos q_2}{w} + \frac{M_p\ h\ p_2^2\ b \sin q_2}{w}\right)\\ & - k_2\ M_p\ h\ p_2^2\ \dot{e} \cos q_2\\ & + k_2\ \left(2\ M_p\ h\ r\ p_2^3 \cos q_2 - \dot{M}_1\ \ddot{q}_{1d} - M_1\ \dddot{q}_{1d} - M_p\ h\ g\ p_2 \cos q_2\right)\\ & +\ \left(k_1 + k_2\ \left(\dot{M}_2 + M_p\ h\ p_2\ r \sin q_2 - M_p\ h\ \dot{e} \sin q_2\right)\right)\\ & \left(\frac{2\ k_m\ k_e\ M_1}{R\ r^2\ w}p_1 - \frac{M_p^2\ h^2\ p_2^2 \sin q_2 \cos q_2}{w} - \frac{M_p\ h\ g\ a \sin q_2}{w}\right)\end{aligned} \tag{83}$$

$$\ddot{M}_1 = -M_p h\dot{p}_2 \sin q_2 - M_p h p_2^2 \cos q_2$$

$$\ddot{M}_2 = -M_p h r\dot{p}_2 \sin q_2 - M_p h r p_2^2 \cos q_2$$

The desired dynamics of z_2 is expressed as follows;

$$\dot{z}_{2d} = -z_1 - c_2 z_2 \tag{84}$$

Solving for nonlinear controller E_1 in terms of desired dynamics of z_3 and z_4 to have,

$$E_1 = \psi_1^{-1}\big(-\phi_1 - \big(1 - c_1^2 + \lambda_1\big)z_1 - (c_1 + c_2)z_2 + \lambda_1 c_1 \chi_1\big) \tag{85}$$

where, the design constant c_2 has positive-real value.

Step 3: Introducing a new regulated variable z_3 defined by

$$z_3 = q_3 - q_{3d} \tag{86}$$

Taking the time derivative of z_3 as

$$\dot{z}_3 = p_3 - \dot{q}_{3d} \tag{87}$$

One may choose the following stabilizing function

$$\alpha_2 = -c_3 z_3 + \dot{q}_{3d} \tag{88}$$

where, the design constant c_3 has positive-real value. If the virtual control variable is assigned to the state variable p_3, and the error state variable z_4 can be defined as

$$z_4 = p_3 - \alpha_2 \tag{89}$$

Step 4: Again, taking the time-derivative of state variable z_4 as,

$$\dot{z}_4 = \dot{p}_3 - \dot{\alpha}_2 \tag{90}$$

$$\dot{z}_4 = \psi_2 E_2 + \phi_2 + c_3 \dot{z}_3 \tag{91}$$

where,

$$\psi_2 = -\frac{k_m D}{2 R r I_{pdel}}, \ \phi_2 = -\ddot{q}_{3d}$$

One can express the desired dynamics of state variable z_4 as,

$$\dot{z}_{4d} = -z_3 - c_4 z_4 \tag{92}$$

Solving for the control signal E_2 in terms of desired dynamics of z_3 and z_4,

$$E_2 = \psi_2^{-1}\big(-\phi_2 - \big(1 - c_3^2\big)z_3 - (c_3 + c_4)z_4\big) \tag{93}$$

where, the design constant c_4 has positive real value.

4.3 Analysis Based on Zero Dynamics

The stability of transformed variables z_1 and z_2 may not be guaranteed by applying the control input of Eq. (85). Also, the nonlinear dynamic equations of the system can be reduced into a reduced-order state model in terms of z_1 and z_2 by utilizing state transformation equations, Eq. (71) and Eq. (78). Consequently, such state transformation leads to internal dynamics of second order [94].

Considering the variable z_1 as the system output and second differentiable, one can have

$$\ddot{z}_1 = \dot{z}_2 - c_1\dot{z}_1 - \lambda_2 z_1 \tag{94}$$

One can rewrite Eq. (94) as follows:

$$\ddot{z}_1 = \psi E_1 + \lambda_2 z_1 + c_1\dot{z}_1 + \phi - c_1\dot{z}_1 - \lambda_2 z_1 \tag{95}$$

or,

$$\ddot{z}_1 = \psi E_1 + \phi \tag{96}$$

Setting $z_1 = 0$, then from Eq. (71), one can get;

$$z_1 = e + k_1 q_2 + k_2(M_1\dot{e} + M_2 p_2) = 0 \tag{97}$$

or,

$$\text{e} = -k_1 q_2 - k_2(M_1\dot{e} + M_2 p_2) \tag{98}$$

According to Eq. (75), and since $z_1 = 0$ and its derivative $\dot{z}_1 = 0$, the following expression can be deduced

$$\dot{e} = -\frac{\left(k_1 + k_2\dot{M}_2\right)}{\left(1 + k_2\dot{M}_1\right)}p_2 - \frac{k_2}{\left(1 + k_2\dot{M}_1\right)}\left(M_p h p_2^2 r \sin q_2 - M_p h g \sin q_2 - M_1\ddot{q}_{1d}\right) \tag{99}$$

Since $\ddot{z}_1 = 0$, one can have

$$\ddot{z}_1 = \psi E_1 + \phi = 0 \tag{100}$$

or,

$$E_1 = -\psi^{-1}\phi \tag{101}$$

Based on Eq. (101), one can have the following dynamics using E_1 as the input of the subsystem

$$\dot{q}_2 = p_2$$

$$\dot{p}_2 = f_2 + g_2 E_1 \Rightarrow f_2 - g_2\big(\psi^{-1}\phi\big) = F(q_2, p_2)|z_1 = 0 \tag{102}$$

Referring to the expressions of ψ, ϕ, q_2 and p_2 in Eqs. (81), (83), (98), and(99), respectively, it is clear that the parameter k plays a vital role in these expressions. To ensure desired characteristics of internal stability, it is evident that Eqs. (82), (83), (98), (99), and the zero dynamics in Eq. (102) are relying uniquely on the parameter k.

4.4 Stability Analysis

Lemma 1

For the nonlinear Segway robotic system, the trajectory tracking of prescribed reference trajectory is well-performed by applying the control input described in Eqs. (85) and (93). If the state variables are started with initial conditions $[q_1(0)p_1(0)q_2(0)p_2(0)q_3(0)p_3(0)]$, then under the control input described by Eqs. (85) and (93), global asymptotic stability are guaranteed such that the tracking errors $[e(t)\dot{e}(t)q_2(t)p_2(t)z_3(t)\dot{z}_3(t)]$ convergence to their corresponding equilibrium points as $t \to \infty$.

Proof

One can decompose the proof of Lemma 1 into three stages. The first stage has to show how the control law could guarantee the asymptotic stability of controlled system described by Eqs. (76), (81), (91), and (93). In the second stage, one has to show how the state variables of nonlinear dynamic model, represented by Eq. (26), have to converge to zero as $t \to \infty$. In the last stage, globally asymptotic convergence to zero has to be proved.

The stability analysis is initialized by appropriate choice of L. F.

$$\mathrm{V} = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2}z_3^2 + \frac{1}{2}z_4^2 + \frac{1}{2}\lambda_2\chi_1^2 \tag{103}$$

Taking the time derivative of above L.F. along the solutions of dynamic system, described by Eqs. (77), (79), (80), (87), and Eq. (91), one can obtain;

$$\dot{V} = -c_1 z_1^2 - c_2 z_2^2 - c_3 z_3^2 - c_4 z_4^2 \leq 0 \tag{104}$$

Based on Eq. (104), the inequality $V(t) \leq V(0)$ is satisfied, and this indicates that the state variables χ_1,z_1, z_2, z_3 and z_4 are bounded, and hence the boundenss of $\dot{z}_1$, $\dot{z}_2$ $\dot{z}_3$, $\dot{z}_4$ are ensured. Taking the second derivative of L.F., one can have

$$\ddot{V} = -2c_1 z_1 \dot{z}_1 - 2c_2 z_2 \dot{z}_2 - 2c_3 z_3 \dot{z}_3 - 2c_4 z_4 \dot{z}_4 \tag{105}$$

According to above equation, the second derivative of L.F. ($\ddot{V}$) is bounded due to due to boundness of state variables z_1, z_2 z_3, z_4, $\dot{z}_1$, $\dot{z}_2$, $\dot{z}_3$ and $\dot{z}_4$. One can utilize Barbalat's Lemma to show the convergence of states z_1, z_2, z_3 and z_4 to zero as $t \to \infty$. Since the zero-convergence of state z_1 was proven, the behavior of zero dynamics of Eq. (102) indicates the asymptotic convergence of states q_2 and p_2.

Based on Eq. (71), one can conclude

$$\underset{t\to\infty}{lim}\, e = \underset{t\to\infty}{lim}\,[-k_1 q_2 - k_2(M_1 \dot{e} + M_2 p_2)] = 0 \tag{106}$$

Using Eq. (76) and Eq. (78), the following reasoning can be reached

$$\underset{t\to\infty}{lim}\left(1 + k_2 \dot{M}_1\right)\dot{e} = \underset{t\to\infty}{lim}\, \alpha = 0 \tag{107}$$

The zero-convergence proof of states q_2 and p_2 leads to the fact;

$$\mathrm{e} + k_2\left(ar + M_p h\right)\dot{e} \to 0 \tag{108}$$

Due to the orthogonally of q_1 and p_1 and taking the limits of e and $\dot{e}$ as $t \to \infty$ gives

$$\underset{t\to\infty}{lim}\, e = \underset{t\to\infty}{lim}\,[q_1 - q_{1d}] = 0 \tag{109}$$

$$\underset{t\to\infty}{lim}\, \dot{e} = \underset{t\to\infty}{lim}\,[p_1 - \dot{q}_{1d}] = 0 \tag{110}$$

The zero-convergence of z_1 leads to zero-convergence of Eq. (108). Accordingly, the states q_1 and p_1 of Eqs. (109) and (110) will converge asymptotically to the desired reference trajectory. Also, the zero-convergence of z_3, which has previously proved, results in $\dot{z}_3 = 0$. In addition, based on Eqs. (86) and (87), the states q_3 and p_3 will converge asymptotically to the desired reference trajectories. Therefore, the global stabilization of the SMR system has been guaranteed by applying the proposed control law.

5 Simulation and Results

This part is devoted to MATLAB modeling of controlled SMR system based on a block backstepping control approach. The effectiveness and dynamic behaviors of the backstepping controller for the robotic system is verified via numerical simulation based on MATLAB programming format. The numerical simulation has been implemented using M-files coded with MATLAB instructions and applied the 4th-order Runge–Kutta as numerical solver. In addition, a step size of 0.01 s has been chosen as a suitable sampling time to keep reasonable plot resolution of behaviors and to guarantee the stability of numerical solution for controlled SMR system. The performances of controlled SMR system based on linear and nonlinear block backstepping controllers has been assessed and a comparison study has been conducted. Also, the effect of excluding and including the integral action on performance of block backstepping controller for nonlinear SMR system has been considered. Table 1 gives of values of parameters describing the SMR system.

Table 2 lists the setting of design parameters that have been used through the design of block backstepping control.

5.1 Regulator Path

To start the numerical simulation of controlled system based on linear and nonlinear block backstepping controllers, the following initial conditions have to be used:

Table 1 The values of parameters for SMR system [95, 96]

Descriptions of parameters	Value
The wheel's mass (M_w)	0.025 (kg)
The chassis's mass (M_p)	0.525 (kg)
The resistance of armature coil of DC motor (R)	2.66 (ohm)
The acceleration of gravity (g)	9.81 (m/sec^2)
The distance between the pendulum's center of gravity and wheel center (h)	0.055 (m)
The chassis Inertia (I_p)	0.0103 (kg · m^2)
Inertia due to rotation of chassis (I_{pdel})	0.00018 (kg · m^2)
The radius of cart's wheel (r)	0.031 (m)
The constant of Back emf (k_e)	0.006
The wheel Inertia (I_w)	0.000012 (kg · m^2)
Maximum volt of DC motor	12 (volt)
The motor's constant (k_m)	0.00498
The separated distance between the wheels' contact patches (D)	0.2 (m)
Maximum volt of control input (E_{max})	24 (volt)

Table 2 The setting values of design constants in the design of block back-stepping control

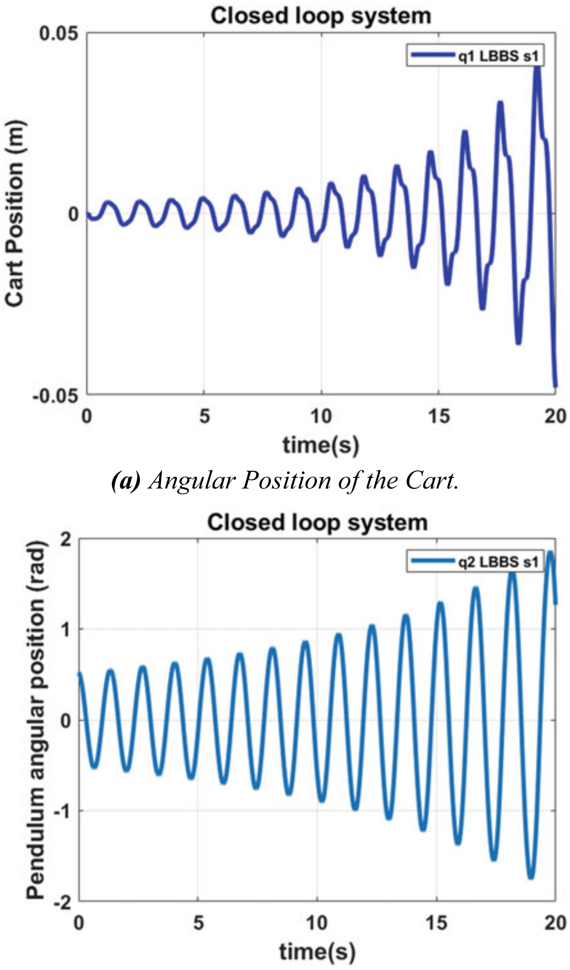
The setting values of design constants for nonlinear controller	Value	The setting values of design constants for nonlinear controller	Value
c_1	5.00	c_1	5.00
c_2	5.00	c_2	5.00
c_3	1.00	c_3	1.00
c_4	1.00	c_4	1.00
k_1	−0.10	k_1	−0.11
k_2	1.10	k_2	1.00
λ_2	10.00	λ_1	1.00

$$[q_1(0)p_1(0)q_2(0)p_2(0)q_3(0)p_3(0)]^T = [0, 0, 0.523, 0, 0, 0]^T.$$

In the first scenario of linearized model the open-loop test has been conducted as shown in Figs. 6a, b. It is evident from the figure that the pendulum cannot be balanced at the upright position and the cart cannot track the reference path.

In the second scenario of linearized model and the scenario of nonlinear model the cart of Segway Mobile Robot has been commanded with step-input orientation of 1 rad height as shown Fig. 7. In response to this step input orientation, the cart will react with linear motion to balance the pendulum at the upright position such that the cart will settles at zero equilibrium point. The responses of linear positions and velocities for the cart are depicted in Fig. 7a, b. The behaviors of angular position and velocity of the pendulum part for controlled SMR system, based on linear and nonlinear block backstepping controllers, are shown in Fig. 7c, d. Moreover, the dynamic responses of the cart angular position and angular velocities under I rad step input are shown in Fig. 7e, f. According to the figure, it has been reported that the responses of linear motion of the cart and the angular position of the pendulum reach their steady-states in 8 and 7 s, respectively. On the other hand, the velocity responses of cart linear motion and pendulum angular positions reach their steady-states in 9 and 8 s, respectively. In addition, the control laws E_1 and E_2, which developed by the proposed controllers and described by Eqs. (58), (66), (85) and (93), are also simulated in the Fig. 7g, h, respectively. The figure shows that the control laws for both linear and nonlinear controllers could successfully give good balancing of pendulum and give good orientation tracking performance of the cart. Referring to behaviors of control laws, one can conclude that their maximum excursions does not exceed the 9-V of supplied battery devoted for actuating motors in both channels of motions. This indicates reasonable and feasible control actions are developed by linear and nonlinear controllers. Table 3 gives the performance report resulting from both control strategies.

Fig. 6 Responses of the linear controller in Eq. (34) with $q_2(0) = 0.532\text{rad}(30^{\circ})$



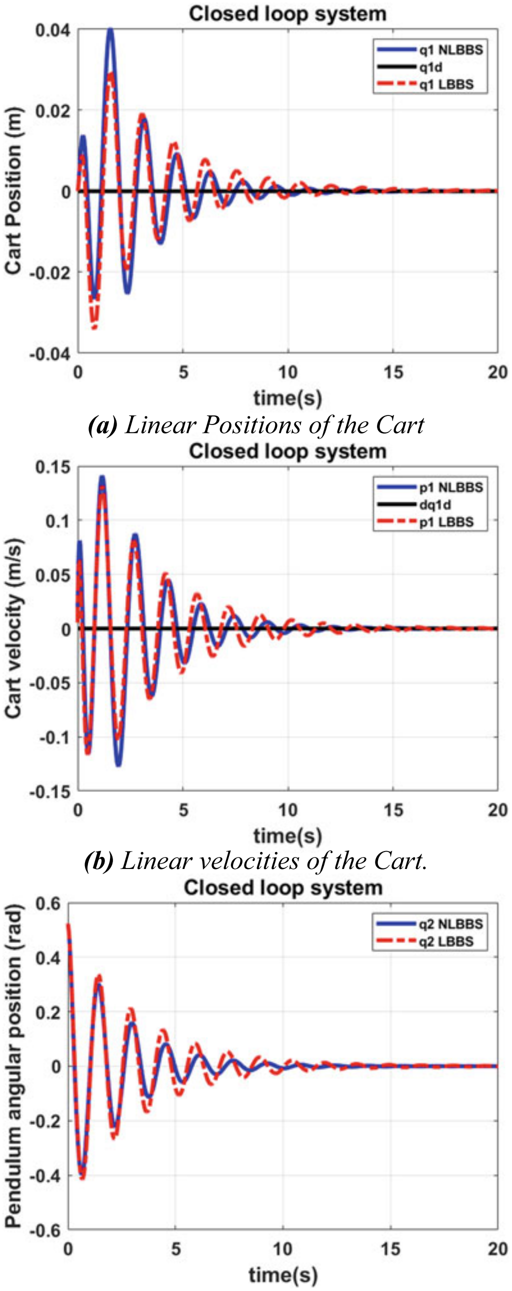
(a) *Angular Position of the Cart.*

(b) *Angular position of Pendulum.*

5.2 Tracking Path

In this part, the cart has been conducted to follow desired trajectory of sinusoidal waveform as shown in Fig. 8. The behaviors of linear motion for the cart are illustrated in Fig. 8a, b based on both proposed controllers. In addition, the behaviors of angular motions for the pendulum are shown in Fig. 8c, d. The figure indicates that the developed control laws described by Eqs. (58), (66), (85) and (93) could show both excellent tracking performance of the cart and good upright stabilizing of pendulum for both nonlinear and linearized model of SMR system. The behavior of control signals is depicted in Fig. 8e. It is evident from the figure that the maximum voltages developed by both controllers do not exceed the supply voltage (9 V.). However, such

Fig. 7 The behaviors of various states and control inputs for SMR system under rotational step input of the Cart

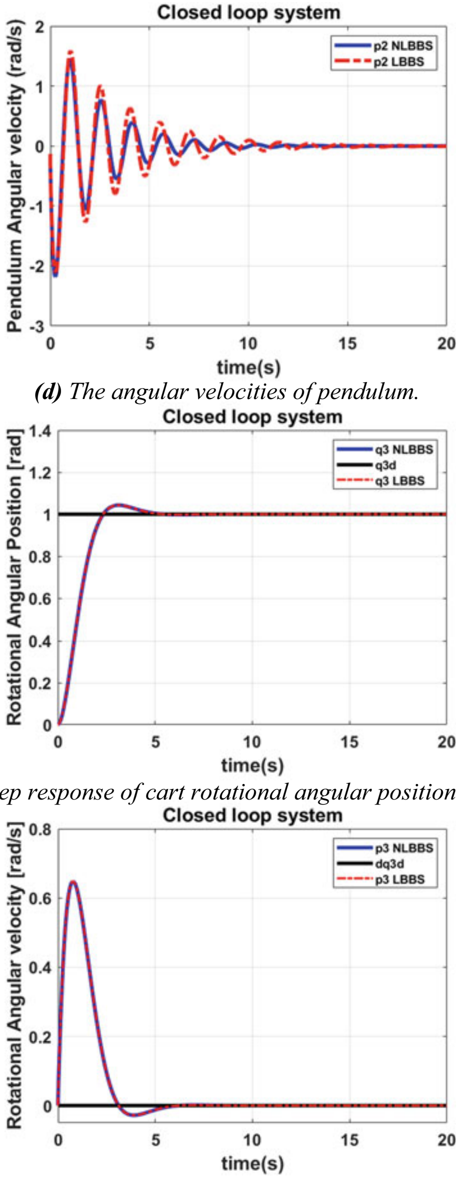


(a) *Linear Positions of the Cart*

(b) *Linear velocities of the Cart.*

(c) *The angular positions of pendulum.*

Fig. 7 (continued)

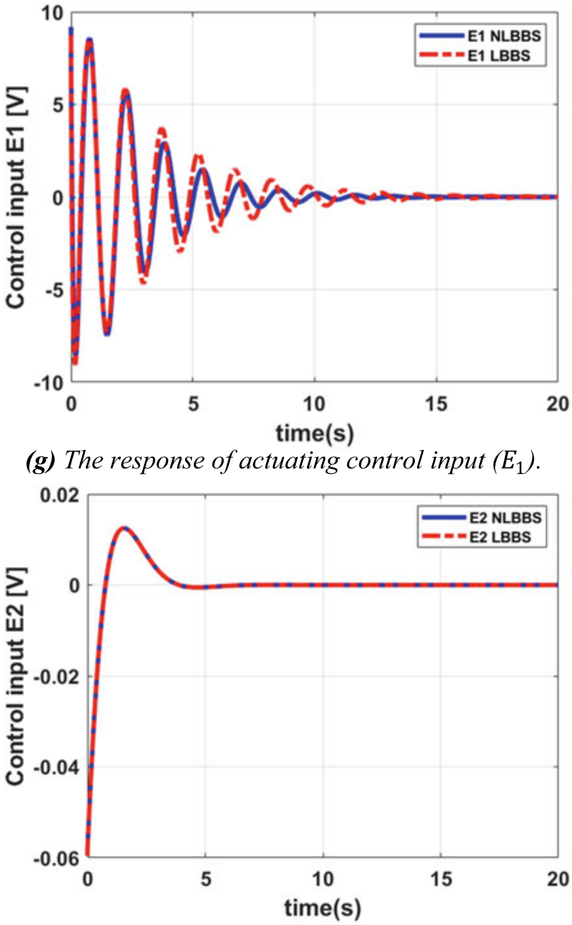


(d) *The angular velocities of pendulum.*

(e) *Step response of cart rotational angular position (q_3).*

(f) *Responses of cart rotational angular velocity (p_3).*

Fig. 7 (continued)



(g) *The response of actuating control input (E_1).*

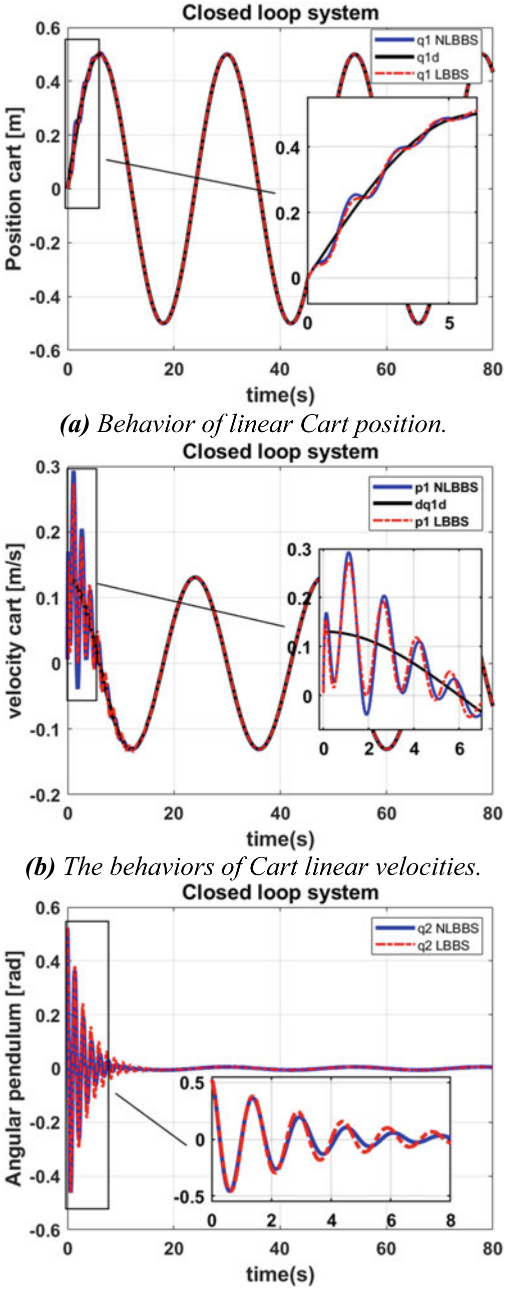
(h) *The response of actuating control input (E_2).*

satisfactory performance is assessed only for specified selected initial conditions, where both controllers showed equivalent performance. In other words, the linear block back-stepping controller may fail to give such performance when the initial conditions are set at large values.

.

Table 4 reports the dynamic performances of both controllers (LBBS and NLBBS).

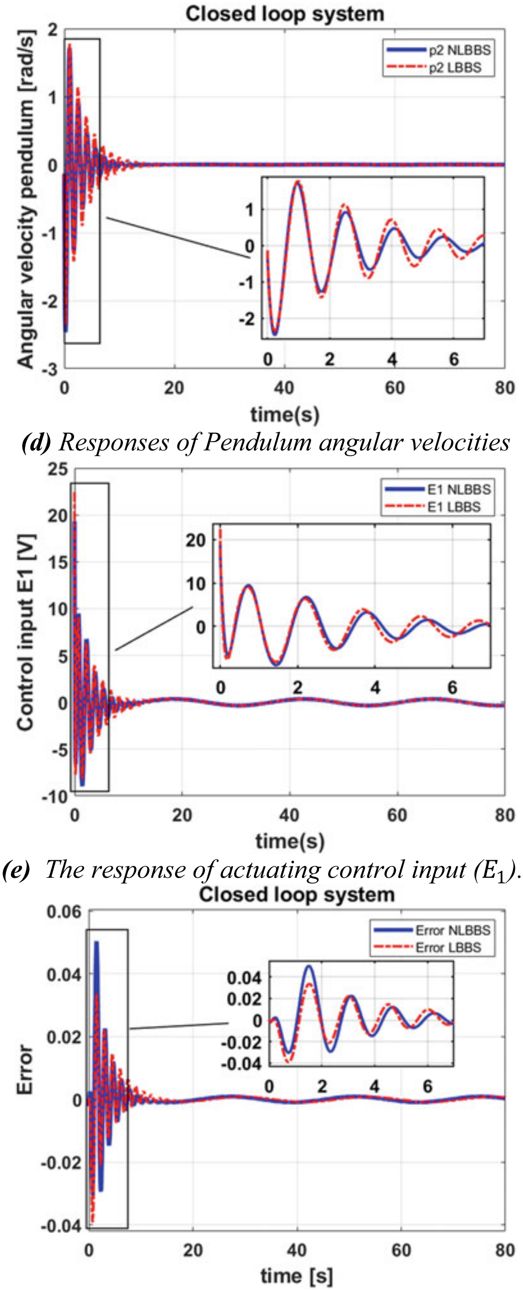
Fig. 8 The behaviors of various states and control inputs for SMR system under sinusoidal desired trajectory



(a) *Behavior of linear Cart position.*

(b) *The behaviors of Cart linear velocities.*

(c) *Responses of Pendulum angular positions.*



(d) *Responses of Pendulum angular velocities*

(e) *The response of actuating control input (E_1).*

(f) *The dynamic errors between the desired and actual positions of cart.*

Fig. 8 (continued)

Table 3 Performance report of proposed controllers for the SMR system

Controller Type	State Variable	RMSE	Settling Time (sec)	Max peak
LBBS	q_1	0.008017	8	−0.033 (m)
	q_2	0.105084	8	0.523 (rad)
	q_3	0.1938	4	0.05 (rad)
NLBBS	q_1	0.008616	8	0.04 (m)
	q_2	0.09346	7	0.523 (rad)
	q_3	0.1938	4	0.05 (rad)

Table 4 The dynamic performance of controlled SMR system using LBBS and NLBBS

Type of Controller	State variables	Max peak	RMSE
Linear Backstepping Controller (LBBS)	q_1	−0.04 (m)	0.004621
	q_2	0.523 (rad)	0.057288
Nonlinear Backstepping Controller (NLBBS)	q_1	−0.043 (m)	0.005247
	q_2	0.523 (rad)	0.053073

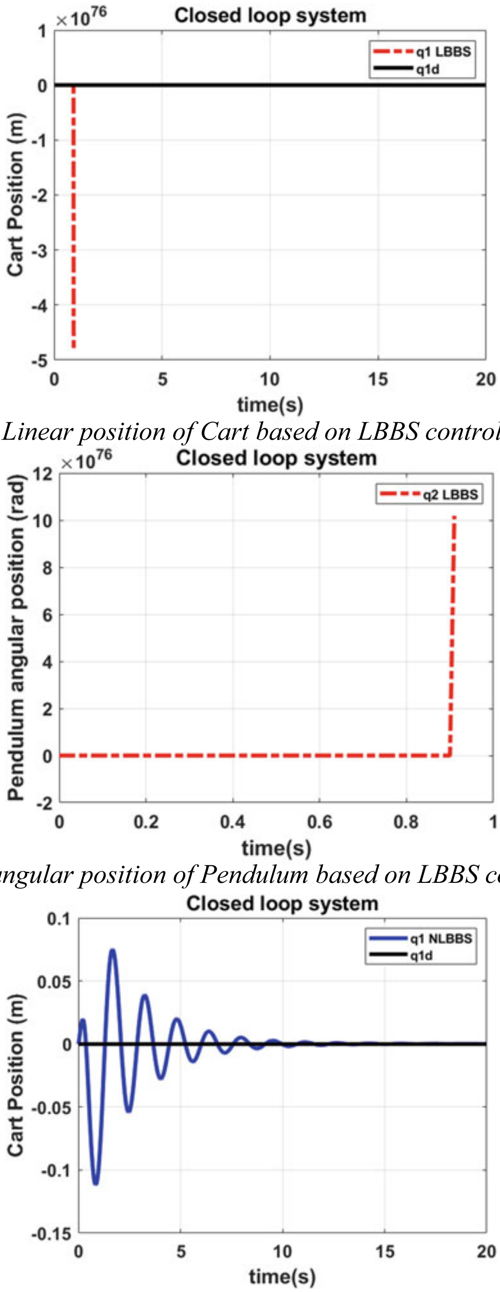
5.3 Performance Assessment Under Large Initial Conditions

In this part, the performances of proposed controllers with control laws, described by Eqs. (58) and (85), have been assessed under large deviated initial states. In the present case, the initial value of the state $q_2(0)$ concerning the pendulum has been set to a value $q_2(0) = 1.22$ rad as defined by the initial state vector below

$$[0, 0, 1.22, 0, 0, 0]^T = [q_1(0), p_1(0), q_2(0), p_2(0), q_3(0), p_3(0)]^T$$

According to above state vector, one can notice that the initial state of pendulum angular position $q_2(0) = 1.22$ is far away from zero equilibrium state $q_2(0) = 0$. The responses of various states of SMR system based on linear and nonlinear block backstepping controllers as shown in Fig. 9. The figure shows that the linear block back-stepping controller fails to keep the stability of balancing and tracking as indicated in Fig. 9a, b. The controlled system shows unstable characteristics in terms of tracking and balancing. On the other hand, the figure shows that the nonlinear block backstepping controller could keep the desired performance in the presence of this large initial condition; as with this controller the states of controlled system could successfully been brought to the equilibrium point and stability of controlled system has been ensured as shown in Fig. 9c, d.

Table 5 gives the numeric characteristics of controlled SMR system based on nonlinear block backstepping controller with large initial deviation of pendulum angular position $q_2 = 1.22 rad$.

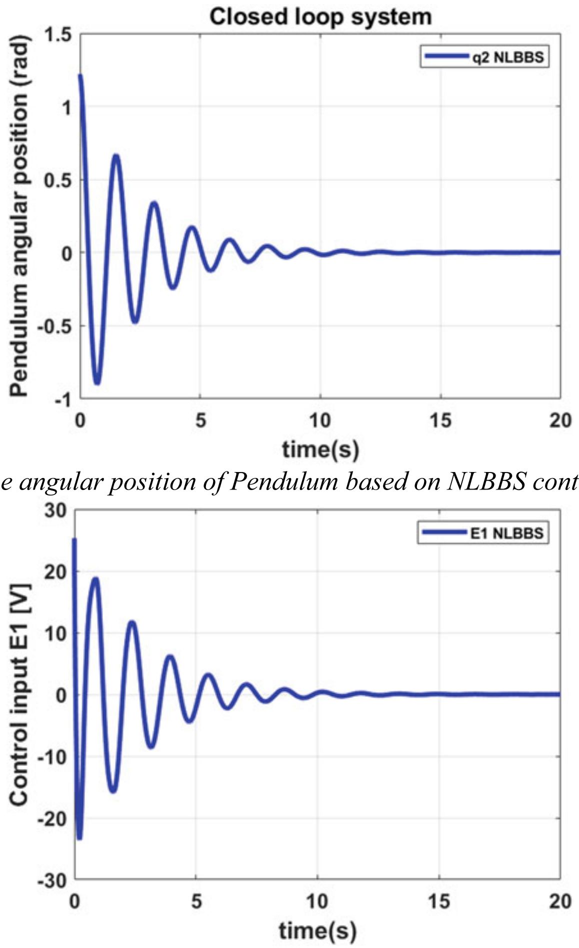


***(a)** Linear position of Cart based on LBBS controller.*

***(b)** The angular position of Pendulum based on LBBS controller.*

***(c)** Linear position of Cart based on NLBBS controller.*

Fig. 9 The dynamic performances both linear and nonlinear block backstepping controllers (with large initial condition of pendulum angle)



(d) *The angular position of Pendulum based on NLBBS controller.*

(e) *Actuating control input* E_1

Fig. 9 (continued)

Table 5 Performance report of nonlinear block backstepping controller for SMR system

State Variables	RMSE	Settling Time (sec)	Max peak
q_1	0.023425	9	0.11(m)
q_2	0.22475	8	1.22(rad)

5.4 Simulation Results Based on Integral Action and Disturbance Effects

Figures 10 and 11 shows the effect of including the integral action in the design of nonlinear block backstepping controller on dynamic and robustness characteristics under load and load-free cases. In both figures, the initial condition $q_2(0) = 0.523$ has been considered, while the other initial values for other states are set to zeros. In Fig. 10, the effect of excluding and including the integral part in the control design of nonlinear controller has been investigated in the no-load case. The figure clearly shows how the inclusion of an integral component in the control design could reduce the maximum peak of pendulum angular position and cart linear position to lower levels. During the time period (10–10.5) seconds, 0.5 N height disturbance has been applied at the gravity center of pendulum mass, and the centers of right and left wheels. According to Fig. 11, the inclusion of integral term in the control design could improve the robustness characteristics of the controlled SMR system based on nonlinear block backstepping controller. Table 6 reports numerically the performance of nonlinear controller in the presence of integral action in terms of steady-state and transient characteristics.

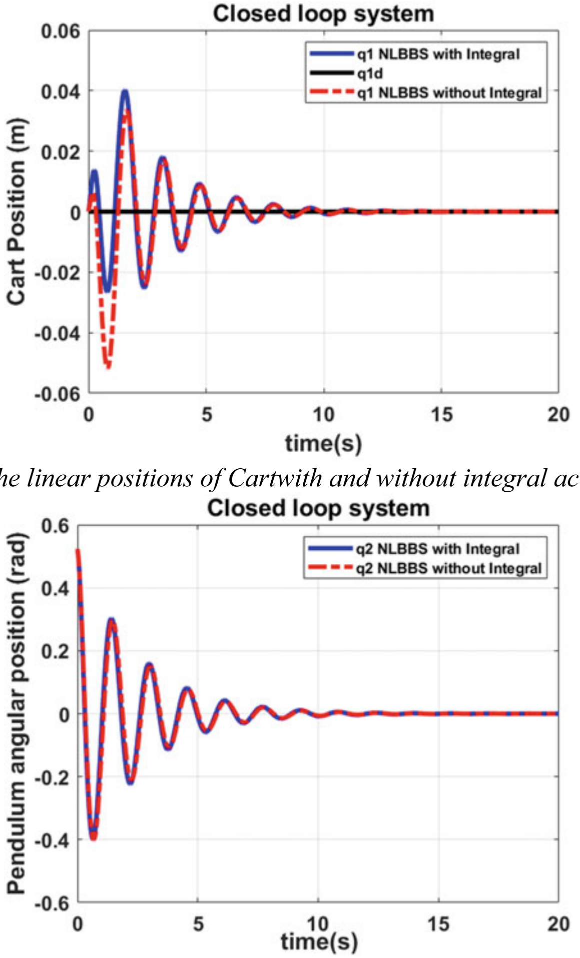
5.5 Simulation Results for Circle Path

Figure 12 shows the tracking performance of linear and nonlinear block backstepping controllers, represented by control laws Eqs. (58), (66), (85), and (93), for desired trajectories: $X_r = \cos t$, $Y_r = \sin t$

In this scenario, the initial conditions for state variables are given by initial vector:

$$[q_1(0) p_1(0) q_2(0) p_2(0) q_3(0) p_3(0)]^T = [1.4, 0, 0.523, 0, 0, 0]^T.$$

Also, this scenario considered the case of disturbance application with height 0.5 N during the time period (10–10.5) seconds at the center of gravity of pendulum, and at the centers of left and right wheels. According to Fig. 12, the results showed that the nonlinear block backstepping controller could give well trajectory tracking performance and the nonlinear controller could satisfy the control objectives with minimum errors of both steady-state and transient behavior. As indicated in Fig. 12a, the response of cart linear position reaches the steady-state in 4 s in the case of nonlinear block backstepping controller, while it reaches the steady-state in 7 s in the case of linear version of controller. In Fig. 12b, it easy to deduce that the behavior of the pendulum angular reaches the equilibrium state in 4 s for the nonlinear controller, while it reaches the equilibrium point in 7 s for the linear controller. In Fig. 12c, the responses of rotational angles for cart chassis for both controllers are shown. In Fig. 12d, e, the actuating signals of controller E_1 and E_2 are demonstated. As shown in Fig. 12d, the control law due to nonlinear controller, Eq. (85), is used to synthesize

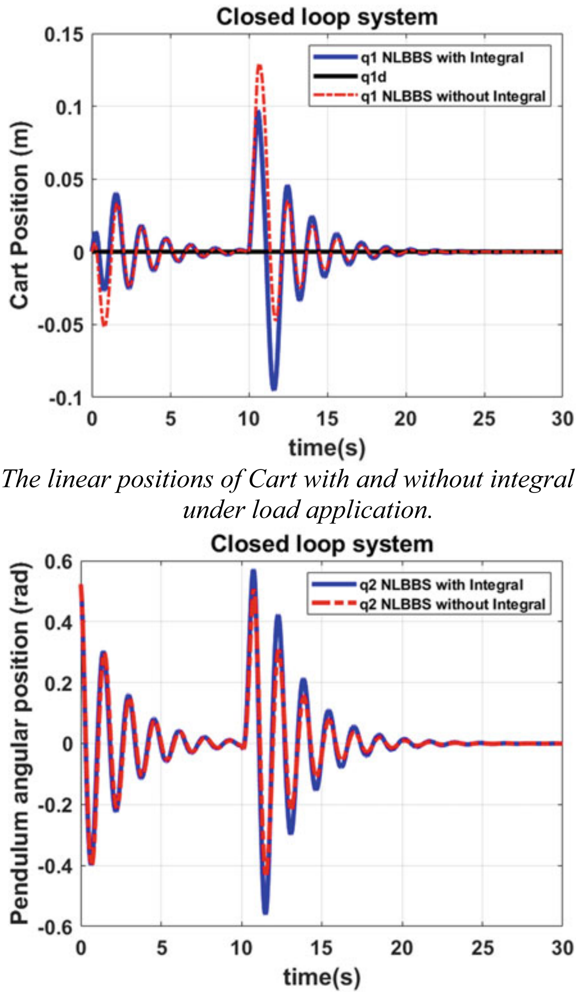


(a) *The linear positions of Cartwith and without integral action.*

(b) *The angular positions of Pendulum with and without integral action.*

Fig. 10 The dynamic performances of controlled SMR system based on nonlinear block backstepping controller with and without integral action

the control signal for left and right actuating motors; i.e., $E_1 = V_{aR} + V_{aL}$. Also, one can see that the maximum voltage produced by the controller may reach to 25 V, and this is reasonable voltage which can be produced by available batteries. Hence, one can conclude that the proposed nonlinear block backstepping controller is feasible and applicable in real world applications. However, a higher maximum voltage (62 V) has been observed in actuating signal associated with linear version of controller. This indicates that the nonlinear controller could save more energy than its counterpart. Figure 12f represents the trajectory path of controlled SMR system along above circular reference path. Figure 12g shows the behaviors of distance errors



(a) *The linear positions of Cart with and without integral action under load application.*

(b) The angular positions of Pendulum with and without integral action under load application..

Fig. 11 The effect of Integral action on dynamic and robustness characteristics of nonlinear block backstepping controller with load exertion

due to both proposed controllers. It is clear that from the figure that the nonlinear controller gives better dynamic performance than the linear version of controller. The error behavior of cart heading angle is shown in Fig. 12h.

Table 7 reports the dynamic performances of both controllers (LBBS and NLBBS).

Table 6 Performance of nonlinear block backstepping controller for SMR system

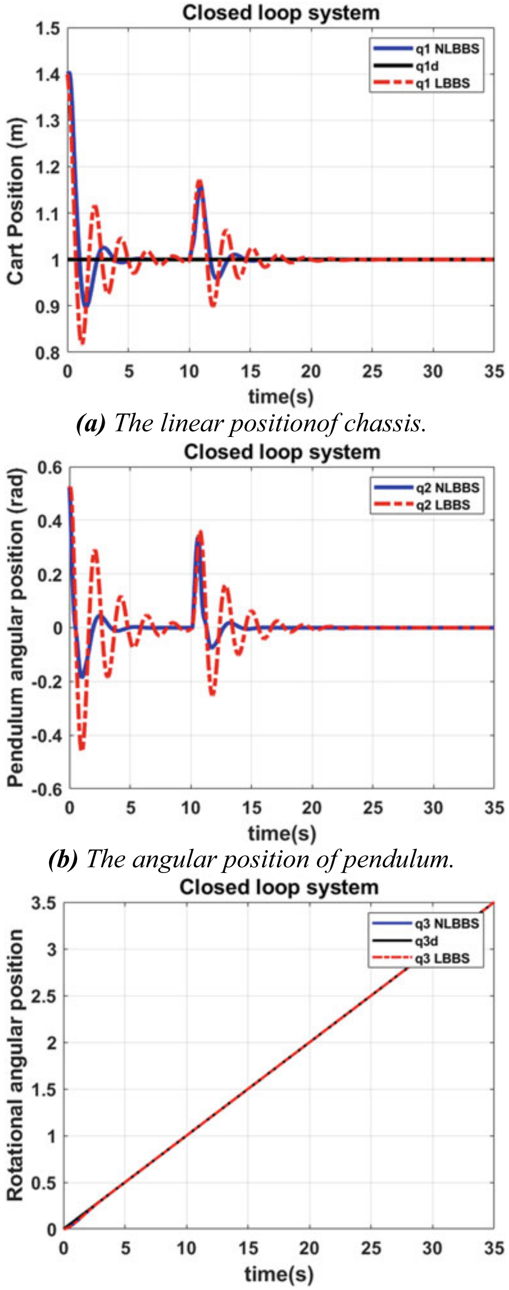
Nonlinear block backstepping Controller	State variables	Max peak	RMSE	Steady-State error
Integral action included (Disturbance free)	q_1	−0.05	0.008616	3.035×10^{-4}
	q_2	0.523	0.09346	1.15×10^{-4}
Integral action Excluded (Disturbance free)	q_1	0.04	0.010247	3.534×10^{-4}
	q_2	0.523	0.0934	1.6×10^{-4}
Integral action included (with Disturbance)	q_1	0.08	0.017502	1.34×10^{-4}
	q_2	0.523	0.130796	3.47×10^{-4}
Integral action Excluded (with Disturbance)	q_1	0.13	0.021668	4.977×10^{-4}
	q_2	0.523	0.12506	9.53×10^{-4}

6 Conclusion

This chapter presented and developed two versions of block backstepping control design; one for linearized model (linear block backstepping controller) and the other for nonlinear model (nonlinear block backstopping controller) for trajectory tracking and balancing of Segway Mobile Robot system. The control design objective for controlled underactuated SMR system is to keep the pendulum in an upright posture and to make the cart track desired trajectory (path). Based on the proposed control design, a stability analysis using Lyapunov theory and zero dynamic criterion has been conducted to ensure global asymptotic stability for the controlled SMR system. According to the observations of numerical results, the following concluded points can be drawn: Both linear and nonlinear controllers could perform good stabilizing performance when they are tested for cart-orientation scenario (regulation case). In case of tacking scenario with small initial states, the nonlinear controller gives better tracking and stabilization performances than linear controller. In addition, the both controllers give reasonable control signals which make them applicable in real-world applications. In case of tacking scenario with large initial states, the linear controller shows unstable characteristics; on the contrary of nonlinear controller which could keep good tracking and stabilizing performance with this large initial condition. In case of including the integral action in control design of nonlinear controller, the results showed that the integral could bound the maximum excursions of pendulum angular position and cart linear position to lower limits. Moreover, the addition of integral part could improve the dynamic and robustness characteristics of nonlinear controller to large extent. In the case of circular-path tracking, the nonlinear controller shows better performance than linear controller in terms of dynamic errors and accuracy.

In order to extend the present study for future work, the following suggestions can be pursued. In this, the design of block backstepping controllers has been conducted based on state feedback control approach. If one or more state is not measured, then one may introduce the observer to propose "Observer-based block

Fig. 12 Responses of the system for path trajectory

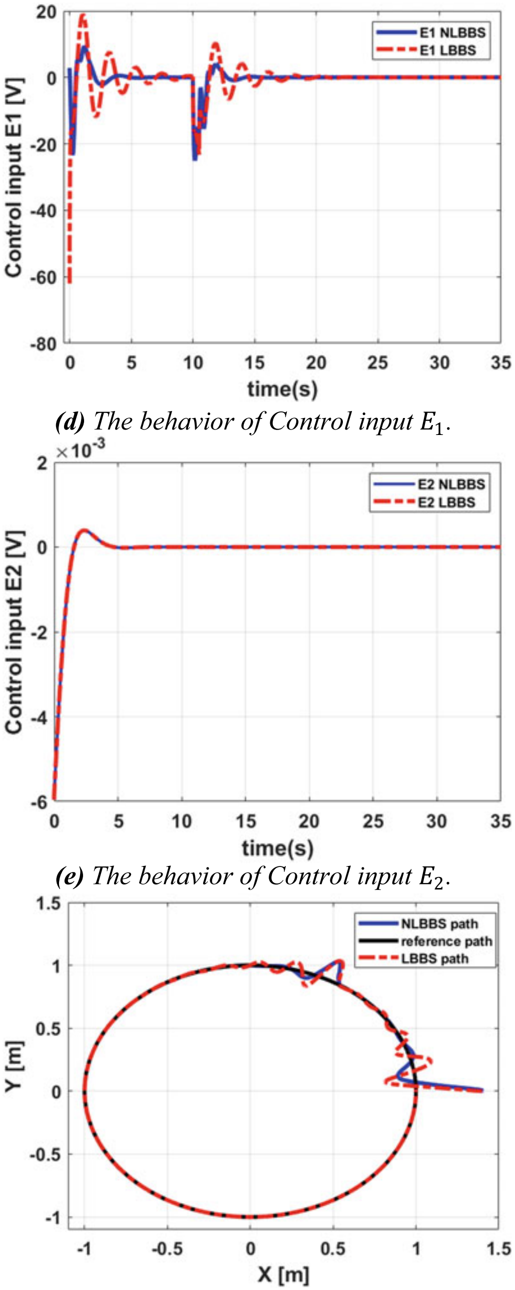


(a) *The linear positionof chassis.*

(b) *The angular position of pendulum.*

(c) *The behavior of rotational angular position.*

Fig. 12 (continued)

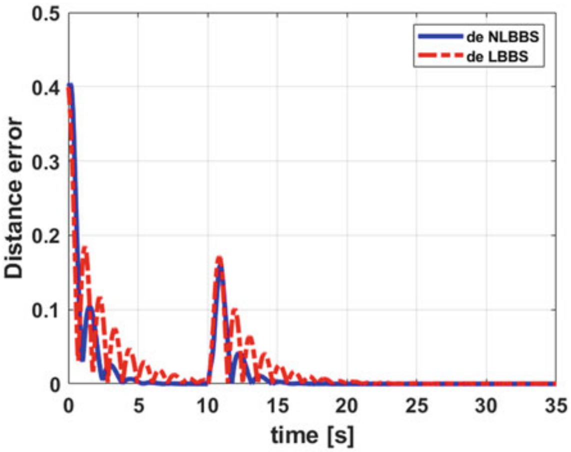


(d) *The behavior of Control input* E_1.

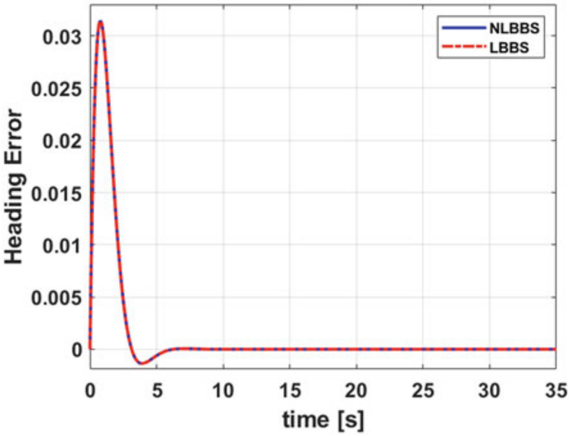
(e) *The behavior of Control input* E_2.

(f) *Desired and actualtrajectoryof cart.*

Fig. 12 (continued)



(g) *The error in distance.*



(g) *The heading angular error of cart.*

Table 7 Numerical reports of performances for LBBS and NLBBS controllers

Type of Controller	Output	RMSE	Max peak
LBBS controller	q_1	0.0559	0.19
	q_2	0.1059	0.47
	d_e	0.0561	–
NLBBS controller	q_1	0.0552	0.16
	q_2	0.05774	0.34
	d_e	0.05555	–

backstepping Control" [97–99]. In addition, to show the effectiveness of block backstepping controller to other under-actuated systems, one may suggest other under-actuated system like ships, airplanes, quad-copter, and parallel robots [100–103]. Moreover, one other critical problem in control design based on block backstepping control approach is how to choose the design parameters. The try-and-error procedure is conventionally used for this purpose. One may suggest modern optimization methods like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Butterfly Optimization Algorithm (BOA), Grey-Wolf Optimization (GWO), and Whale-Optimization Algorithm (WOA) to tune the design parameters of block backstepping control design [104–106].

Acknowledgements The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for supporting this work. Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

References

1. Batista DIG (2012) Solvability of the direct Lyapunov first matching condition in terms of the generalized coordinates. Kansas State University
2. Toumi I, Meghni B, Hachana O, Azar AT, Boulmaiz A, Humaidi AJ, Ibraheem IK, Kamal NA, Zhu Q, Fusco G, Bahgaat NK (2022) Robust variable-step perturb-and-observe sliding mode controller for grid-connected wind-energy-conversion systems. Entropy 24:731. https://doi.org/10.3390/e24050731
3. Daraz A, Malik SA, Azar AT, Aslam S, Alkhalifah T, Alturise F (2022) Optimized fractional order integral-tilt derivative controller for frequency regulation of interconnected diverse renewable energy resources. IEEE Access 10:43514–43527. https://doi.org/10.1109/ACCESS.2022.3167811
4. Ali T, Malik SA, Hameed IA, Daraz A, Mujlid H, Azar AT, Load frequency control and automatic voltage regulation in a multi-area interconnected power system using nature-inspired computation-based control methodology. Sustainability 14(19):12162. https://doi.org/10.3390/su141912162
5. Abdul-Adheem WR, Ibraheem IK, Azar AT, Humaidi AJ (2021) Design and analysis of a novel generalized continuous tracking differentiator. Ain Shams Eng J 13(4):101656
6. Abdul-Kareem AI, Hasan AF, Al-Qassar AA, Humaidi AJ, Hassan RF, Ibraheem IK, Azar AT (2022) Rejection of wing-rock motion in delta wing aircrafts based on optimal LADRC schemes with butterfly optimization algorithm. J Eng Sci Technol 17(4):2476–2495
7. Sain C, Banerjee A, Biswas PK, Azar AT, Babu TS (2022) Design and optimisation of a fuzzy-PI controlled modified inverter-based PMSM drive employed in a light weight electric vehicle. Int J Autom Control 16(3/4):459–488
8. Ali MO, Abou-Loukh SJ, Al-Dujaili AQ, Alkhayyat A, Abdulkareem AI, Ibraheem IK, Humaidi AJ, Al-Qassar AA, Azar AT (2022) Radial basis function neural networks-based short term electric power load forecasting for super high voltage power grid. J Eng Sci Technol 17(1):0361–0378
9. Kazim M, Azar AT, Koubaa A, Zaidi A (2021) Disturbance rejection based optimized robust adaptive controller for UAVs. IEEE Syst J 15(2):3097–3108
10. Ajel AR, Humaidi AJ, Ibraheem IK, Azar AT (2021) Robust model reference adaptive control for tail-sitter VTOL aircraft. Actuators 10:162. https://doi.org/10.3390/act10070162

11. Drhorhi I, El Fadili A, Berrahal C, Lajouad R, El Magri A, Giri F, Azar AT, Vaidyanathan S (2021) Adaptive backstepping controller for DFIG-based wind energy conversion system. In: Backstepping control of nonlinear dynamical systems, advances in nonlinear dynamics and chaos (ANDC), Academic Press, pp 235–260
12. Daraz A, Malik SA, Waseem A, Azar AT, Haq IU, Ullah Z, Aslam S (2021) Automatic generation control of multi-source interconnected power system using FOI-TD controller. Energies 14(18):5867. https://doi.org/10.3390/en14185867
13. Pilla R, Gorripotu TS, Azar AT (2021) Tuning of extended Kalman filter using grey wolf optimisation for speed control of permanent magnet synchronous motor drive. Int J Autom Control 15(4–5):563–584
14. Al-Qassar AA, Abdulkareem AI, Hasan AF, Humaidi AJ, Ibraheem IK, Azar AT, Hameed AH (2021) Grey-wolf optimization better enhances the dynamic performance of roll motion for tail-sitter VTOL aircraft guided and controlled By STSMC. J Eng Sci Technol 16(3):1932–1950
15. Humaidi AJ, Najem HT, Al-Dujaili AQ, Pereira DA, Ibraheem IK, Azar AT (2021) Social spider optimization algorithm for tuning parameters in PD-like Interval Type-2 fuzzy logic controller applied to a parallel robot. Measur Control 54(3–4):303–323. https://doi.org/10.1177/0020294021997483
16. Abdul-Adheem W, Ibraheem IK, Humaidi AJ, Azar AT (2021) Model-free active input-output feedback linearization of a single-link flexible joint manipulator: an improved active disturbance rejection control approach. Measur Control 54(5–6):856–871. https://doi.org/10.1177/0020294020917171
17. Pilla R, Gorripotu TS, Azar AT (2021) Design and analysis of search group algorithm based PD-PID controller plus redox flow battery for automatic generation control problem. Int J Comput Appl Technol 66(1):19–35
18. Najm AA, Ibraheem IK, Humaidi AJ, Azar AT (2021) Output tracking and feedback stabilization for 6-DoF UAV using an enhanced active disturbance rejection control. Int J Intell Unmanned Syst. https://doi.org/10.1108/IJIUS-09-2020-0059
19. Al-Qassar AA, Al-Dujaili AQ, Hasan AF, Humaidi AJ, Ibraheem IK, Azar AT (2021) Stabilization of single-axis propeller-powered system for aircraft applications based on optimal adaptive control design. J EngSci Technol (JESTEC) 16(3):1851–1869
20. Abdul-Adheem WR, Ibraheem IK, Azar AT, Humaidi AJ (2020) Improved active disturbance rejection-based decentralized control for MIMO nonlinear systems: comparison with the decoupled control scheme. Appl Sci 10(7):2515. https://doi.org/10.3390/app10072515
21. Abdul-Adheem WR, Azar AT, Ibraheem IK, Humaidi AJ (2020) Novel active disturbance rejection control based on nested linear extended state observers. Appl Sci 10(12):4069. https://doi.org/10.3390/app10124069
22. Liu L, Ma D, Azar AT, Zhu Q (2020) Neural computing enhanced parameter estimation for multi-input and multi-output total nonlinear dynamic models. Entropy 22(5):510. https://doi.org/10.3390/e22050510
23. Kammogne AST, Kountchou MN, Kengne R, Azar AT, Fotsin HB, Ouagni STM (2020) polynomial robust observer implementation based-passive synchronization of nonlinear fractional-order systems with structural disturbances. Front Inf Technol Electron Eng 21(9):1369–1386
24. Djeddi A, Dib D, Azar AT, Abdelmalek S (2019) Fractional order unknown inputs fuzzy observer for Takagi-Sugeno systems with unmeasurable premise variables. Mathematics 7(10):984. https://doi.org/10.3390/math7100984
25. Pilla R, Azar AT, Gorripotu TS (2019) Impact of flexible AC transmission system devices on automatic generation control with a metaheuristic based Fuzzy PID controller. Energies 12(21):4193. https://doi.org/10.3390/en12214193
26. Ammar HH, Azar AT, Shalaby R, Mahmoud MI (2019) Metaheuristic optimization of fractional order incremental conductance (FO-INC) maximum power point tracking (MPPT). Complexity, Vol 2019, Article ID 7687891, 1–13. https://doi.org/10.1155/2019/7687891

27. Radwan AG, Emira AA, Abdelaty A, Azar AT (2018) Modeling and analysis of fractional order DC-DC converter. ISA Trans 82(2018):184–1991
28. Meghni B, Dib D, Azar AT, Saadoun A (2018) Effective supervisory controller to extend optimal energy management in hybrid wind turbine under energy and reliability constraints. Int J Dyn Control 6(1):369--383, Springer. https://doi.org/10.1007/s40435-016-0296-0
29. Meghni B, Dib D, Azar AT (2017) A Second-order sliding mode and fuzzy logic control to optimal energy management in PMSG Wind turbine with battery storage. Neural Comput Appl 28(6):1417–1434. https://doi.org/10.1007/s00521-015-2161-z
30. Fekik A, Denoun H, Azar AT, Kamal NA, Zaouia M, Benyahia N, Hamida ML, Benamrouche N, Vaidyanathan S (2021) Direct power control of three-phase PWM-rectifier with backstepping control. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 215–234
31. Humaidi AJ, Sadiq ME, Abdulkareem AI, Ibraheem IK, Azar AT, Adaptive backstepping sliding mode control design for vibration suppression of earth-quaked building supported by magneto-rheological damper. J Low Frequency Noise Vib Active Control 41(2):768–783. https://doi.org/10.1177/14613484211064659
32. Nail B, Bekhiti B, Hafaifa A et al (2015) Feedback linearization and backstepping controllers for coupled tanks. Inf Eng Int J (IEIJ) 3(4):135
33. Ghaderi P, Amini F (2017) Adaptive block backstepping control for civil structures with unknown parameters subjected to seismic excitation. Struct Control Health Monitoring 24(2)
34. Humaidi A, Hameed M (2019) Development of a new adaptive backstepping control design for a non-strict and under-actuated system based on a PSO tuner. Information 10(2):38
35. Vaidyanathan S, Jafari S, Pham VT, Azar AT, E. Alsaadi F (2018) A 4-D chaotic hyperjerk system with a hidden attractor, adaptive backstepping control and circuit design. Arch Control Sci 28(2):239–254
36. Kazim M, Azar AT, Abdelkader M, Koubaa A (2022) Adaptive Backstepping based linear parameter varying model predictive control multi-rotor UAVs. 2022 2nd international conference of smart systems and emerging technologies (SMARTTECH), May 9–11, 2022, Riyadh, Saudi Arabia, pp 166–171, https://doi.org/10.1109/SMARTTECH54121.2022.00045
37. Pham VT, Vaidyanathan S, Azar AT, Duy VH (2021) A new chaotic system without linear term, its backstepping control, and circuit design. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 33–52
38. Vaidyanathan S, Pham VT, Azar AT (2021) A new chaotic jerk system with egg-shaped strange attractor, its dynamical analysis, backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 53–71
39. Sambas A, Vaidyanathan S, Zhang S, Mohamed MA, Zeng Y, Azar AT (2021) A new 4-D chaotic hyperjerk system with coexisting attractors, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 73–94
40. Sambas A, Vaidyanathan S, Zhang S, Mohamed MA, Zeng Y, Azar AT (2021) A new 3-D chaotic jerk system with a saddle-focus rest point at the origin, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 95–114
41. Vaidyanathan S, Sambas A, Azar AT, Rana KPS, Kumar V (2021) A new 5-D hyperchaotic four-wing system with multistability and hidden attractor, its backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 115–138
42. Vaidyanathan S, Sambas A, Azar AT, Rana KPS, Kumar V (2021) A new 4-D hyperchaotic temperature variations system with multistability and strange attractor, bifurcation analysis, its active backstepping control, and circuit realization. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 139–164

43. Alimi M, Rhif A, Rebai A, Vaidyanathan S, Azar AT (2021) Optimal adaptive backstepping control for chaos synchronization of nonlinear dynamical systems. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 291–345
44. Kumar V, Rana KPS, Azar AT, Vaidyanathan S (2021) Backstepping controller for nonlinear active suspension system. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 347–374
45. Bansal N, Bisht A, Paluri S, Kumar V, Rana KPS, Azar AT, Vaidyanathan S (2021) Single-link flexible joint manipulator control using backstepping technique. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 375–406
46. Singh S, Mathpal S, Azar AT, Vaidyanathan S, Kamal NA (2021) Multi-switching synchronization of nonlinear hyperchaotic systems via backstepping control. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 425–447
47. Vaidyanathan S, Sambas A, Azar AT (2021) A 5-D hyperchaotic dynamo system with multistability, its dynamical analysis, active backstepping control, and circuit simulation. In: Backstepping control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 449–471
48. Sambas A, Vaidyanathan S, Sukono, Azar AT, Hidayat Y, Gundara G, Mohamed MA (2021) A novel chaotic system with a closed curve of four quarter-circles of equilibrium points: dynamics, active backstepping control, and electronic circuit implementation. In: Backstepping Control of nonlinear dynamical systems, Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp 485–507
49. Rozali SM, Kamarudin MN, Rahmat MFA et al (2013) Asymptotic tracking position control for nonlinear systems using backstepping technique. Procedia Eng 53:255–263
50. Cheng CC, Chiang YC, Huang PC (2017) Design of adaptive block backstepping controllers with perturbations estimation for nonlinear state-delayed systems in semi-strict feedback form. Asian J Control 19(3):856–873
51. Humaidi AJ, Tala'at EN, Hameed MR et al (2019) Design of adaptive observer-based backstepping control of cart-pole pendulum system. 2019 IEEE international conference on electrical, computer and communication technologies (ICECCT). IEEE, 1–5
52. Koditschek DE (1987) Adaptive techniques for mechanical systems. Presented at the proceedings of the 5th workshop on adaptive systems. New Haven
53. Sontag ED, Sussmann HJ (1989) Further comments on the stabilizability of the angular velocity of a rigid body. Syst Control Lett 12(3):213–217
54. Tsinias J (1989) Sufficient Lyapunov-like conditions for stabilization. Math Control Signals Syst 2(4):343–357
55. Byrnes CI, Isidori A (1989) New results and examples in nonlinear feedback stabilization. Syst Control Lett 12(5):437–442
56. Kokotovic PV, Sussmann HJ (1989) A positive real condition for global stabilization of nonlinear systems. Syst Control Lett 13(2):125–133
57. Kokotovic P, Krstic M, Kanellakopoulos I (1992) Backstepping to passivity: recursive design of adaptive systems. Proceedings of the 31st IEEE conference on decision and control, Tucson, Arizona. IEEE, 3276–3280.
58. Kanellakopoulos I, Kokotovic P, Morse A (1992) A toolkit for nonlinear feedback design. Syst Control Lett 18(2):83–92
59. Kanellakopoulos I, Krein P (1993) Integral-action nonlinear control of induction motors. IFAC Proc Volumes 26(2):117–120
60. Krstic M, Kokotovic PV, Kanellakopoulos I (1995) Nonlinear and adaptive control design. Wiley, Canada
61. Fossen TI, Berge SP (1997) Nonlinear vectorial backstepping design for global exponential tracking of marine vessels in the presence of actuator dynamics. Proceedings of the 36th IEEE conference on decision and control, San Diego, California USA. IEEE, 5, 4237–4242

62. Kokotovic P (1999) Constructive nonlinear control: Progress in the 90's. In: Proceedings 14th IFACWorld Congr, Bejjing, China. Plenary and Index Volume, 49–77
63. Tsai F-K, Lin J-S (2003) Nonlinear control design of 360-degree inverted pendulum systems. 4th International conference on control and automation proceedings, Montreal, Canada. IEEE Computer Society, 634–638.
64. Fu Y-C, Lin J-S (2005) Nonlinear backstepping control design of the furuta pendulum. Proceedings of 2005 IEEE conference on control applications, Toronto, Canada. IEEE, 96–101
65. Rudra S, Barai RK, Maitra M (2017) Applications of the block backstepping algorithm on 2-DOF Underactuated mechanical systems: some case studies. Block Backstepping design of nonlinear state feedback control law for Underactuated Mechanical Systems. Springer, pp 53–108
66. Rudra S, Barai RK, Maitra M (2017) Applications of the block backstepping algorithm on underactuated mechanical systems with higher degrees of freedom: some case studies. Block Backstepping design of nonlinear state feedback control law for underactuated mechanical systems. Springer, pp 109–144
67. Ibraheem GAR, Azar AT, Ibraheem IK, Humaidi AJ (2020) A novel design of a neural network based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization. Complexity, vol 2020, Article ID 3067024, pp 1–18, https://doi.org/10.1155/2020/3067024
68. Mahdi SM, Yousif NQ, Oglah AA, Sadiq ME, Humaidi AJ, Azar AT (2022) Adaptive synergetic motion control for wearable knee-assistive system: a rehabilitation of disabled patients. Actuators 11(7):176. https://doi.org/10.3390/act11070176
69. Ajeil FH, Ibraheem IK, Azar AT, Humaidi AJ (2020) Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. Sensors 20(7):1880. https://doi.org/10.3390/s20071880
70. Najm AA, Ibraheem IK, Azar AT, Humaidi AJ (2020) Genetic optimization-based consensus control of multi-agent 6-DoF UAV system. Sensors 20(12):3576. https://doi.org/10.3390/s20123576
71. Humaidi AJ, Ibraheem IK, Azar AT, Sadiq ME (2020) A new adaptive synergetic control design for single link robot arm actuated by pneumatic muscles. Entropy 22(7):723. https://doi.org/10.3390/e22070723
72. Ajeil F, Ibraheem IK, Azar AT, Humaidi AJ (2020) Autonomous Navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment. Int J Adv Rob Syst 17(3):1–15. https://doi.org/10.1177/1729881420929498
73. Khamis A, Meng J, Wang J, Azar AT, Prestes E, Li H, Hameed IA, Haidegger T (2022) AI and robotics in the fight against COVID-19 Pandemic. In: Azar AT, Hassanien AE (eds) Modeling, control and drug development for COVID-19 outbreak prevention. Studies in Systems, Decision and Control, vol 366. Springer, Cham. https://doi.org/10.1007/978-3-030-72834-2_3
74. Ammar H, Ibrahim M, Azar A.T., Shalaby R. (2020) Gray wolf optimization of fractional order control of 3-omni wheels mobile robot: experimental study. In: 2020 16th international computer engineering conference (ICENCO), 29–30 Dec. 2020, Cairo, Egypt, pp 147–152. https://doi.org/10.1109/ICENCO49778.2020.9357384
75. Soliman M, Azar AT, Saleh MA, Ammar HH (2020) Path planning control for 3-Omni fighting robot using PID and fuzzy logic controller. In: Hassanien A, Azar A, Gaber T, Bhatnagar R, Tolba F M (eds) The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in Intelligent Systems and Computing, vol 921, pp 442–452, Springer, Cham
76. 1Barakat MH, Azar AT, Ammar HH (2020) Agricultural service mobile robot modeling and control using artificial fuzzy logic and machine vision. In: Hassanien A, Azar A, Gaber T, Bhatnagar R, Tolba FM (eds) The international conference on advanced machine learning technologies and applications (AMLTA2019). AMLTA 2019. Advances in Intelligent Systems and Computing, vol 921, pp 453–465, Springer, Cham

77. Azar AT, Ammar HH, Mayra Beb MY, Garces SR, Boubakarig A (2020) Optimal design of PID controller for 2-DOF drawing robot using bat-inspired algorithm. The international conference on advanced intelligent systems and informatics AISI 2019. Advances in Intelligent Systems and Computing, vol 1058, pp 175–186, Springer
78. Music Z, Molinari F, Gallenmüller S et al (2019) Design of a networked controller for a two-wheeled inverted pendulum robot. IFAC-Papers OnLine 52(20):169–174
79. Key M-S, Jeon C-G, Yoo DS (2012) Sliding mode control for a two-wheeled inverted pendulum mobile robot driving on uniform slopes. 2012 12th international conference on control, automation and systems. IEEE, 2159–2162
80. Nawawi S, Ahmad M, Osman J et al (2006) Controller design for two-wheels inverted pendulum mobile robot using PISMC. 2006 4th student conference on research and development. IEEE, 194–199
81. Mansour S-MB, Ghommam J, Naceur S-M (2016) Design and visio control of two-wheeled inverted pendulum mobile robot. 2016 2nd international conference on advanced technologies for signal and image processing (ATSIP). IEEE, 437–442
82. Villacrés J, Viscaíno M, Herrera M et al (2016) Real-time implementation of different controllers for a two-wheeled inverted pendulum. Int J Circuits Syst Signal Process 10:281–290
83. Goher KM, Fadlallah SO (2019) Control of a two-wheeled machine with two-directions handling mechanism using PID and PD-FLC algorithms. Int J Autom Comput 16(4):511–533
84. Odry Á, Kecskés I, Burkus E et al (2017) Protective fuzzy control of a two-wheeled mobile pendulum robot: design and optimization. WSEAS Trans Syst Control 12:297–306
85. Feriyonika F, Hidayat A (2020) Balancing control of two-wheeled robot by using linear quadratic gaussian (LQG). J Telecommun Electron Comput Eng (JTEC) 12(3):55–59
86. Isdaryani F, Salam R, Feriyonika F (2021) Design and implementation of two-wheeled robot control using MRAC. J Telecommun Electron Comput Eng (JTEC) 13(3):25–30
87. Nabil H (2020) Supervised neural network control of real-time two wheel inverted pendulum. J Adv Eng Trends 38(2):131–146
88. Kharola A (2015) The control of two-wheeled inverted pendulum robot (TWIPR) using fuzzy logic. 2015 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, pp 1–4
89. Zhou Y, Wang Z, Chung K-W (2019) Turning motion control design of a two-wheeled inverted pendulum using curvature tracking and optimal control theory. J Optim Theory Appl 181(2):634–652
90. Kang MT, Vo HD, Kim HK et al (2007) Control system design for a mobile inverted pendulum via sliding mode technique. 2007 IEEE international conference on mechatronics. IEEE, pp 1–6
91. Wang C (2014) Adaptive tracking control of uncertain MIMO switched nonlinear systems. Int J Innov Comput Inf Control 10(3):1149–1159
92. Grasser F, D'arrigo A, Colombi S et al (2002) JOE: a mobile, inverted pendulum. IEEE Trans Industr Electron 49(1):107–114
93. Humaidi A, Hameed M, Hameed A (2018) Design of block-backstepping controller to ball and arc system based on zero dynamic theory. J Eng Sci Technol 13(7):2084–2105
94. Rudra S, Barai RK, Maitra M (2014) Nonlinear state feedback controller design for underactuated mechanical system: a modified block backstepping approach. ISA Trans 53(2):317–326
95. Yang CH (2014) DSP-Based Control of Two-Wheeled Inverted Pendulum. Department of Electrical Engineering, National Taiwan Ocean University, Master's degree
96. Humaidi AJ, Hameed MR, Ajel AR et al (2021) Block Backstepping control design of two-wheeled inverted pendulum via zero dynamic analysis. 2021 IEEE 12th control and system graduate research colloquium (ICSGRC). IEEE, pp 87–92.
97. Hassan M, Humaidi A, Hamza M (2020) On the design of backstepping controller for Acrobot system based on adaptive observer. Int Rev Electr Eng (IREE) 15(4):328–335

98. Aljuboury AS, Hameed AH, Ajel AR et al (2022) Robust adaptive control of knee exoskeleton-assistant system based on nonlinear disturbance observer. Actuators. MDPI 11(3):78
99. Humaidi AJ, Tala'at EN, Al-Dujaili AQ et al (2020) Design of backstepping control based on adaptive observer for underactuated system. 2020 7th international conference on control, decision and information technologies (CoDIT). IEEE, 1, 288–293
100. Humaidi AJ, Mohammed AA, Hameed AH et al (2020) State estimation of rotary inverted pendulum: a comparative study of observers performance. 2020 IEEE Congreso Bienal de Argentina (ARGENCON). IEEE, pp 1–7
101. Humaidi AJ, Hasan AF (2019) Particle swarm optimization–based adaptive super-twisting sliding mode control design for 2-degree-of-freedom helicopter. Measur Control 52(9–10):1403–1419
102. Humaidi AJ, Abdulkareem AI (2019) Design of augmented nonlinear PD controller of Delta/Par4-like robot. J Control Sci Eng
103. Humaidi AJ, Hussein HA (2019) Adaptive control of parallel manipulator in Cartesian space. 2019 IEEE international conference on electrical, computer and communication technologies (ICECCT). IEEE, pp 1–8
104. Ghanim T, Ajel AR (2020) Optimal fuzzy logic control for temperature control based on social spider optimization. IOP conference series: materials science and engineering. IOP Publishing, 745(1):012099
105. Humaidi AJ, Kadhim SK, Gataa AS (2022) Optimal adaptive magnetic suspension control of rotary impeller for artificial heart pump. Cybern Syst 53(1):141–167
106. Al-Qassar AA, Al-Obaidi ASM, Hasan AF et al (2021) Finite-time control of wing-rock motion for delta wing aircraft based on whale-optimization algorithm. Indonesian J Sci Technol 6(3):441–456

Extremum Seeking Based PID Control of Quadrotor System

Safanah M. Raafat, Abdullah N. Muhsen, and Ahmed A. A. Alawsai

Abstract The control of quadrotor has been recognized for its difficulty due to the significance of the maneuverability, presence of nonlinearity and strong coupling of variables. The quadrotor features a sophisticated dynamic model with six degrees of freedom, making it naturally unstable. As a result, it is necessary to ensure the system stability around the intended reference state, or to adjust it, regardless the presence of internal or external disturbances. Moreover, it will be interesting to investigate the development of an efficient robust non-model-based control approach. The online Extremum Seeking (ES) optimization algorithm is a non- model approach that minimizes a cost function. It can iteratively modify controller parameters based on the change of a selected cost function to reach a local minimum. On the other hand, Proportional-Integral-Derivative (PID) controller is well known for its simple structure, good performance, and configuration even without a precise model of the controlled system. Though, it requires efficient tuning for better performance during processing. For this purpose, ES is developed in this chapter to automate the tuning of the PID controller. Furthermore, the ES tuned PI-D controller is compared to a different tuned PID controller that uses particle swarm optimization (PSO) to predict the PID parameters. The simulation results show that the ES-PID controller provides better stability, real- time optimization, and consumes less time to reach the optimal value as compared to the PSO-PID controller.

Keywords Quadrotor · Proportional-Integral-Derivative (PID) control · Extremum Seeking (ES) · Particle Swarm Optimization (PSO)

S. M. Raafat (✉) · A. N. Muhsen
University of Technology-Iraq, Baghdad, Iraq
e-mail: Safanah.m.raafat@uotechnology.edu.iq

A. A. A. Alawsai
University of Wasit, Wasit, Iraq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_17

1 Introduction

The quadrotor is considered as one of the most recent advanced technologies that has become a part of the available tools for everyday human use, and is still being under research. Unmanned Aerial Vehicles (UAV)s are aircraft that flies without a pilot, under radio control from a ground station, or independently according to pre-programmed flight plans [1].

Recently, many applications based on UAV have been developed for hazardous environments, catastrophes, and rescue [2–4]. Quadrotor has been extensively utilized in agriculture [5, 6], and useful jobs like waitress quadcopter for serving [7]. Many quadrotor entertainment applications have been illustrated in [8]. Additionally, many studies on the multi-agent systems and formation control can be found in [9, 10].

The quadrotor has four arms and four rotors. Each two of these rotors are fixed on the end of opposite arms and rotate clockwise, whereas the other two rotors are also fixed on the other two arms but rotate counterclockwise. Since the quadrotor is an under-actuated system, the controller design is a difficult challenging task.

Various types of controllers were developed by many researchers to investigate controller design and development. The PID controller is widely used due to its simplicity and ease of application implementation [11, 12]. Meanwhile, the optimization of the PID controllers' parameters, which is a time-consuming operation, is usually included in the usual PID tuning method. A lot of systematic tuning techniques have been established to address some problems [13–16]. A large number of these methods needs previous familiarity of the plant, where the employed model of the system has a significant importance in developing the auto-tuning task [17].

There are various advantages of using a system model with a definite interpretation, like transfer function or a state transition matrix. Consequently, a controller-settings can be directly evaluated through a calculation; as in [18], where the Linear Quadratic Regulator (LQR) feedback control coefficients are directly calculated using the identified open-loop system equation. Another method is based on the analysis of the frequency response plots of the systems using the plant's specified interpretation, where PID controller is required to be designed with the appropriate amplitude and phase characteristics [19]. However, in many cases, an accurate system model cannot be easily obtained and requires complicated procedures. As a result, a basic approximated model can be helpful for determining the parameters of the controller. Nevertheless, when the optimal parameters are implemented for a real-world system, the desired result is rarely achieved. On the other hand, some researchers examined tuning the control parameters without using a model for the system; this strategy may be summarized as determining the best control parameters by weighing a sequence of practicable experiments outcomes equivalent to various system configurations [20]. This approach is a good choice for systems that have a complicated identification procedure in order to avoid excessive impractical oscillations at numerous points during the open-loop operation. One good use of this concept is relay feedback tuning [21]. Most systems will oscillate when briefly replacing the feedback controller with a relay. Hence, a point on the Nyquist plot can be attained, and the designer can determine the best PID parameters for the corresponding gain and phase margins. This strategy has also been used in the cascade controller system [22].

Considering these issues, this chapter aims to develop an auto-tuning technique of PID controllers that are used to control a quadrotor system, based on extremum seeking (ES) algorithm. With the implementation of a discrete form of ES algorithm, we propose a strategy for improving the performance of a PID controlled system. ES, in particular, will minimize a performance index (or a cost function) while improving the PID controller's performance [23]. The non-model-based approach of ES technique can iteratively modify the PID control parameters until the result of the cost function hits a local minimum or maximum. The features of step response performance afford suitable measures for controller design [24]. And since the technique of optimal parameters seeking is based on altering the controller parameters and gathering the reaction performance, it is not possible to have unstable parameter values for the control of the quadrotor.

The remaining of the chapter presents some related works in Sect. 2, illustration of the quadrotor concepts of motion in Sect. 3, description of the PI-D control design strategy in Sect. 4, and the application of the (ES) algorithm for tuning of the PI-D controller in Sect. 5. For comparison with another optimization technique, PSO has been introduced in Sect. 6, Simulation results are provided in Sect. 7, and finally some important comments are concluded in Sect. 8.

2 Related Works

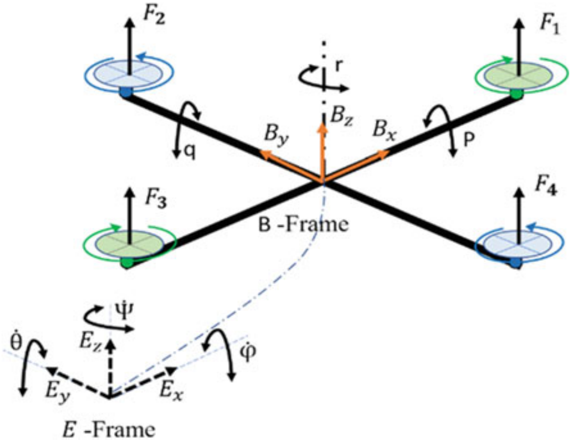
Recently, there are several studies that dealt with quadrotor controller design. In 2017 Islam et al. [25], considered the control and dynamics of a quadrotor using Model Predictive Control (MPC) method. Thua and Gavrilova [26], presented the design of a quadrotor using L_1 adaptive controller design, where the control parameters are analytically determined through instinctively chosen performance and robustness metrics established by the designer. Okyere et al. [27], provided an examination and overall performance of an LQR control algorithm for quadrotor. Consequently, an adaptive state feedback control based on Classical Multiple Model Adaptive Control was developed in [28]. Daniali [29], employed a Nonlinear Model Predictive Control (NMPC) to work out the path following issue for UAVs. A hybrid control system had been designed in [30] to expand the disturbances rejection ability and body jerk recital for a quadrotor model by exploiting an active force control (AFC). The developed intelligent control method incorporated a PID controller and an intelligent active force control (IAFC) component resulting in a robust PID-IAFC system. A robust observer-based nonlinear controller had been established in [31] for quadrotors with consideration of systems' uncertainties and exterior disturbances. The tracking errors are limited by using two dissimilar nonlinear disturbance observers that had been constructed to manage the uncertainties in the system.

On the other hand, many applications of extremum seeking control (ESC) have been well recognized, as in the followings: to create control of maximum friction between the tire and the road during emergency braking for Anti-lock brake systems (ABS) [32]. In [33], ESC had been utilized to maneuver autonomous vehicle in order to find a radiation source, and in [34, 35] for vehicle planar control. In a thermoacoustic combustion chamber, ESC had also been employed to decrease thermoacoustic instabilities. The ESC algorithm is used to identify the best gain and time delay parameters in the actuation of the process [36]. In [37], ESC is used to maximize the biomass production rate for a bioprocess in order to optimize the steady-state operation. ESC has been extensively utilized for maximum power point tracking (MPPT) technique; in photovoltaic systems [38] under various solar irradiation and temperature conditions. Also, in [39], a multivariable gradient-based ES design to abstract maximum power from Photovoltaic PV modules. The developed multivariable design improves the flexibility of tuning the parameters of the controller, and includes the effects of interactions between PV modules. It has been also implemented for wind energy systems to modify the angular velocity of wind turbines under a variety of wind speeds [40].

3 Mathematical Model of Quadrotor Dynamic

Figure 1 depicts the quadrotor's free body diagram with the coordinates. The angular speeds of the four electric motors control the motion of the quadrotor. The basic inputs of the thrust U$_{\text{thrust}}$, roll U$_{\text{roll}}$, pitch U$_{\text{pitch}}$, and yaw t U$_{\text{yaw}}$ are generated by the combination of thrust and torque m_i ($i = 1, 2, 3, 4$). The quadrotor is subjected to two reference frames: the earth-fixed frame $\{O_E X_E Y_E Z_E\}$ and the body-fixed frame $\{O_B X_B Y_B Z_B\}$. These two frames can be used to control (x, y, z) location, attitude (pitch θ, roll ϕ and yaw ψ), and the motion.

Fig. 1 Quadrotor degrees of freedom. Boday and earth inertial frame [41]



The mathematical model of the quadrotor system is developed based on the following assumptions [42, 43]:

1. The distribution of mass of quadrotors is symmetrical.
2. Propellers are inflexible.
3. The thrust and drag forces generated by the quadrotor propulsion system (propellers-rotors) are proportionate to the square of the rotor speed.
4. The quadrotor system does not flip in the planned flight missions, and the total lifting force $\boldsymbol{U}_1$ is always positive.

$$\text{Roll and Pitch angles:} -\frac{\pi}{2} < \theta, \phi < \frac{\pi}{2} \tag{1}$$

$$\text{Yaw angle:} -\pi < \psi < \pi \tag{2}$$

$$\text{Thrust force:} \mathrm{U}_1 > 0 \tag{3}$$

5. The angular movements of quadrotors are expected to be small, therefore Euler angles and the angular velocities time derivative are the same.

The model of quadrotor's six DoF dynamics can be presented as [44]:
The inertial frame linear accelerations:

$$\ddot{\boldsymbol{x}} = \frac{\boldsymbol{U}_{thrust}}{\boldsymbol{m}}\left(\boldsymbol{c}_{\psi}\boldsymbol{s}_{\theta}\boldsymbol{c}_{\phi} + \boldsymbol{s}_{\psi}\boldsymbol{s}_{\phi}\right) - \frac{\boldsymbol{a}_x}{\boldsymbol{m}}\dot{\boldsymbol{x}} \tag{4}$$

$$\ddot{y} = \frac{U_{thrust}}{m}(s_{\psi}s_{\theta}c_{\phi} - c_{\psi}s_{\phi}) - \frac{a_y}{m}\dot{y} \tag{5}$$

$$\ddot{\boldsymbol{z}} = \frac{\boldsymbol{U}_{thrust}}{\boldsymbol{m}}(\boldsymbol{c}_{\phi}\boldsymbol{c}_{\theta} - \boldsymbol{g}\boldsymbol{m}) - \frac{\boldsymbol{a}_z}{\boldsymbol{m}}\dot{\boldsymbol{z}} \tag{6}$$

The Angular acceleration in body frame:

$$\ddot{\boldsymbol{\phi}} = \frac{\boldsymbol{I}_{yy} - \boldsymbol{I}_{zz}}{\boldsymbol{I}_{xx}}\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\psi}} + \frac{\boldsymbol{U}_{pitch}}{\boldsymbol{I}_{xx}} \tag{7}$$

$$\ddot{\boldsymbol{\theta}} = \frac{\boldsymbol{I}_{zz} - \boldsymbol{I}_{xx}}{\boldsymbol{I}_{yy}}\dot{\boldsymbol{\phi}}\dot{\boldsymbol{\psi}} + \frac{\boldsymbol{U}_{roll}}{\boldsymbol{I}_{yy}} \tag{8}$$

$$\ddot{\boldsymbol{\psi}} = \frac{\boldsymbol{I}_{xx} - \boldsymbol{I}_{yy}}{\boldsymbol{I}_{zz}}\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\phi}} + \frac{\boldsymbol{U}_{yaw}}{\boldsymbol{I}_{zz}} \tag{9}$$

$C = cos$, $S = sin$, m is the system mass.

Meanwhile, the thrust $\boldsymbol{T}_{Bi}$ is given by:

$$\boldsymbol{T}_{\boldsymbol{Bi}} = \boldsymbol{k}\sum_{i=1}^{4} \boldsymbol{\omega}_{\boldsymbol{i}}{}^{2} = \boldsymbol{k}(\boldsymbol{\omega}_1{}^2 + \boldsymbol{\omega}_2{}^2 + \boldsymbol{\omega}_3{}^2 + \boldsymbol{\omega}_4{}^2) \tag{10}$$

where $\boldsymbol{k}$ is the constant of friction and $\boldsymbol{\omega}_{\boldsymbol{i}}$ is the angular velocity of the motors, i = 1,4.

The rates of roll, pitch and yaw are related to the angular velocity vector as in the following:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_{\phi}t_{\theta} & c_{\phi}t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi}/c_{\theta} & c_{\phi}/c_{\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{11}$$

where t = tan and p,q,r are the angular velocities.

4 Applied Control Strategy

As can be observed in Eqs. (4)–(9), the dynamic behavior of the quadrotor is underactuated and strongly coupled, since the translational motion hangs on the change of attitude angles. Accordingly, the control system can be divided into the outer-loop for translation controller and inner-loop for attitude controller that are cascaded in a close-loop feedback circuit, as shown in Fig. 2.

4.1 Position Controller

The part of control contains the control laws for positioning the quadrotor as shown in Fig. 3. The path command r(t), which is the state (x, y, z), as well as the output y(t) from the entire block, are the inputs to this system. This controller's output is the attitude command, which is passed to the attitude controller.

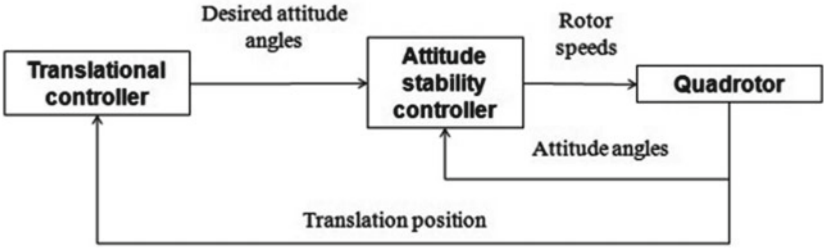


Fig. 2 Attitude and position control loops [45]

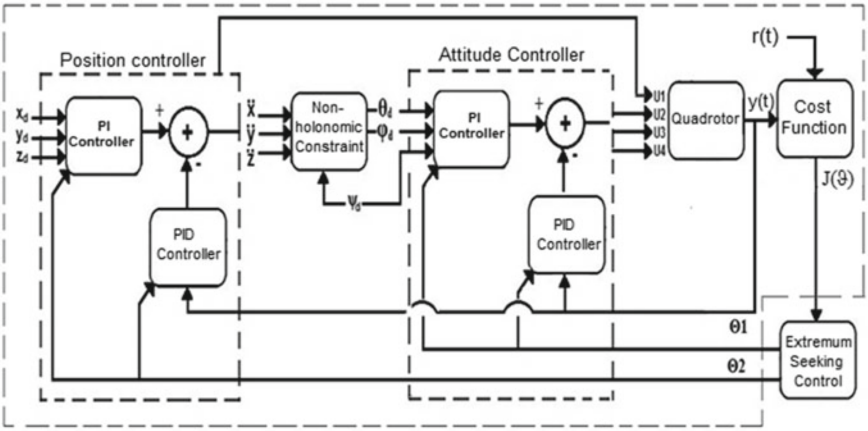


Fig. 3 Block diagram of the ES-PID quadrotor-controlled system, including the position controller, altitude controller, and ESC

As the system response is fed to the cost function, the performance of the system is will reduce to a single number. The ES algorithm adjusts **Θ1** (for position controller) and **Θ2** (for attitude controller) in order to minimize the cost function. where **Θ1** and **Θ2** represent two sets of gains of $[K_P, K_I, K_D]$ for each controller.

The error is determined using the PID control principles, which is then used to optimize the quadrotor's position controller. This controller is known as a lower-level controller, and it is responsible for quadrotor navigation in the x, y, and z coordinates. The non-Holonomic constraint subsystem uses the output from this controller to compute the desired rotational angles.

4.2 *Non-holonomic Constraints*

A non-holonomic system is a system with at minimal one non-holonomic constraint, i.e. a constraint imposed on its velocities $(f(\dot{x}, \dot{y}, \dots) = 0)$ that cannot be integrated and thus transformed into a constraint forced on coordinates $(f(x, y, \dots) = 0)$. Nonholonomy's origins can be classified into: (1) The bodies of motion that are in contact with one another and roll/move without slippage. (2) Moment conservation in a multi-body system is related to under-actuated control.

It is assumed that there is no side slippage in the UAV model since the engine thrust is always aligned with the longitudinal direction of the body. Holonomic constraints are readily dealt with since they may be solved by integrating the constraint equation and then clearing one of the coordinates as a function of the others, thus lowering the system order and releasing the constraint. For non-holonomic constraints, such a technique is impossible [46].

This part receives the output from the position subsystem and calculates the desired rotation angles. This subsystem's output is subsequently supplied into the quadrotor's higher-level controller for attitude stabilization.

4.3 Attitude Controller

For attitude stabilization and quadrotor control, it is necessary to employ effective control laws. The altitude command from the position controller is the controller input. This subsystem is made up of three sub-blocks, each of which implements an ES-PI-D controller using the control rules for three different angles of rotation. In addition to altitude, each of these sub-blocks offers a corrected value of rotational angles (roll, pitch, and yaw). These are the outputs of this block, which are then forwarded to the quadrotor dynamics block for further processing. The quadrotor controller is completed in this block. Following that, quadrotor dynamics are used to move and follow the specified path.

On the other hand, it worth to mention here that the standard PID controller may cause problems. If the preferred input is a step function, the derivative output will be an impulsive signal, which could cause the motors to overheat and push the system out of the controlled region. To avoid this issue, a redesigned PI-D controller has been implemented. The error signal is the input signal for the integral and proportional lines in Fig. 4, while the derivative term is determined directly from sensor data [45].

In order to apply the designated controller for the quadrotor system, it is required to control the six single-input and single-output signals (ϕ, θ and ψ) (heading), in addition to the movement in the x, y, z directions. The implemented PI-D controller is applied as follows:

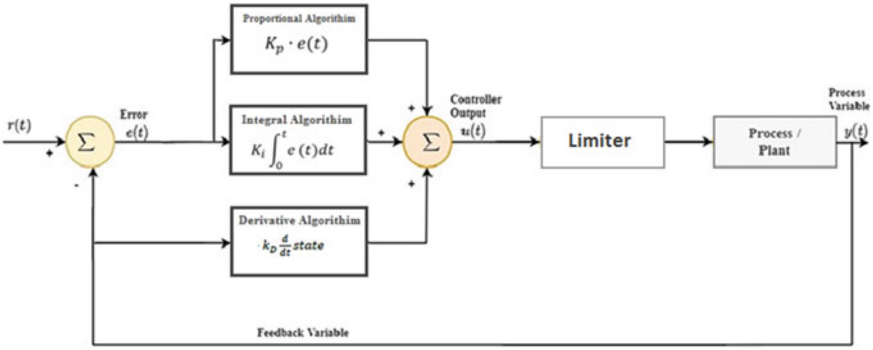


Fig. 4 Block diagram for PI-D controller [47]

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt}\phi(t) \tag{12}$$

where e denotes the error between the desired reference input and the measured output while ϕ is the state measured from the output.

The active vertical net force F_{net} can be obtained by using the $F_{net} = mg_y$ relationship, where (m) is the quadrotor's mass and (g_y) is the vertical acceleration. Then, the net force can be divided into two different vertical forces:

(1) the upward thrust force (F_{thrust}), and
(2) the downward gravitational force (mg) so

$$F_{net} = F_{thrust} - mg \tag{13}$$

where g is the gravity of earth. And by rearranging leads to

$$F_{thrust} = F_{net} + mg \tag{14}$$

The desired suited thrust force, U_{thrust}, can be calculated by [45]:

$$U_{thrust} = K_{Pz}e_z + K_{Iz}\int e_z - K_{Dz}\dot{z}_{meas} + mg \tag{15}$$

where, K_{Pz}, K_{Iz}, and K_{dz} are the gains of the altitude PI-D controller, and e_z is the error due to altitude.

The following equations are used to calculate the control signals:

Roll Control

The torque U_{roll} that rotates the quadrotor around the *x*-axis is as follows:

$$U_{roll} = K_{p\phi}e_\phi + K_{I\phi}\int e_\phi - K_{D\phi}\dot{\phi}_{meas} \tag{16}$$

where: $K_{p\phi}$, $K_{I\phi}$ and $K_{D\phi}$ are the gains of the roll angle PI-D controller; e_ϕ is roll angle error.

Pitch Control

The torque U_{pitch} that rotates the quadrotor around the *y*-axis is as follows:

$$U_{pitch} = K_{p\theta}e_\theta + K_{I\theta}\int e_\theta - K_{D\theta}\dot{\theta}_{meas} \tag{17}$$

where: $K_{p\theta}$, $K_{I\theta}$ and $K_{D\theta}$ are the gains of the pitch angle PI-D controller; e_θ is pitch angle error.

Yaw Control

The torque U_{yaw} that rotates the quadrotor around the z-axis is as follows:

$$U_{yaw} = K_{p\psi} e_{\psi} + K_{I\psi} \int e_{\psi} - K_{D\psi} \dot{\psi}_{meas} \tag{18}$$

where: $K_{p\psi}$, $K_{I\psi}$ and $K_{D\psi}$ are the gains of the yaw angle PI-D controller; e_{ψ} is pitch angle error.

Equations (10) and (11) can be used to compute the correct angular velocities of rotors ω_i using PI-D values from Eqs. (15)–(18)

$$\begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_t} & 0 & \frac{-1}{2k_t L} & \frac{-1}{4b} \\ \frac{1}{4k_t} & \frac{-1}{2k_t L} & 0 & \frac{1}{4b} \\ \frac{1}{4k_t} & 0 & \frac{1}{2k_t L} & \frac{-1}{4b} \\ \frac{1}{4k_t} & \frac{1}{2k_t L} & 0 & \frac{1}{4b} \end{bmatrix} \begin{bmatrix} U_{thrust} \\ U_{pitch} \\ U_{roll} \\ U_{yaw} \end{bmatrix} \tag{19}$$

where $\boldsymbol{L}$ is the distance from the center of quadrotor to the motor, and k_t is the thrust factor.

As the quadrotor has 6DOF and only four actuators, it is difficult to control each of these DOFs directly. Therefore, the control Eqs. (15)–(18) are employed to control 4 DOF (z, ϕ, θ, ψ) directly. While, the roll (ϕ) and pitch (θ) angles cause the quadrotor moves towards the required x and y directions. Thus, the desired x and y are going to be applied first to resolve the desired Roll and pitch angles, and to control positions in x and y directions indirectly.

In the meantime, the equations that are exploited to determine the desired roll and pitch angles are usually evolved in terms of favored x and y accelerations [48].

Define the position error as:

$$\mathbf{P}_{\boldsymbol{e}} = \mathbf{P}_{\boldsymbol{des}} - \mathbf{P} \tag{20}$$

where the vector $\mathrm{P}_{des} = [x_{des}, y_{des}, z_{des}]$ is the position desired vector.

By replacing $\ddot{\mathbf{P}}$ with $\ddot{\mathbf{P}}_{\boldsymbol{des}}$ in Eq. (21), where $\ddot{\mathbf{P}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{\mathrm{R}(\Theta)}{m} T_B - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{\Gamma_D}{m} \tag{21}$$

we obtain:

$$\boldsymbol{R}^{\boldsymbol{T}} \left(\ddot{\mathbf{P}}_{\boldsymbol{des}} + g z_e \right) = \frac{T_B z_e}{m} \tag{22}$$

Then we got:

$$\begin{bmatrix} c_\psi c_\theta & c_\psi c_\theta & -s_\theta \\ s_\phi c_\psi s_\theta - s_\phi s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\theta s_\phi \\ s_\phi s_\psi + c_\phi c_\psi s_\theta & c_\psi c_\phi s_\theta - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} \ddot{x}_{des} \\ \ddot{y}_{des} \\ \ddot{z}_{des} + g \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T_B \end{bmatrix} \tag{23}$$

The following equations are acquired:

$$\ddot{x}_{des} c_\psi c_\theta + \ddot{y}_{des} c_\psi c_\theta - (\ddot{z}_{des} + g) s_\theta = 0 \tag{24}$$

$$\ddot{x}_{des} (s_\phi c_\psi s_\theta - s_\phi s_\psi) + \ddot{y}_{des} \left(c_\phi c_\psi + s_\phi s_\theta s_\psi \right) + (\ddot{z}_{des} + g) c_\theta s_\phi = 0 \tag{25}$$

$$\ddot{x}_{des} (s_\phi s_\psi + c_\phi c_\psi s_\theta) + \ddot{y}_{des} \left(c_\psi c_\phi s_\theta - c_\psi s_\phi \right) + (\ddot{z}_{des} + g) c_\theta c_\phi = \frac{T_B}{m} \tag{26}$$

In consequence of the fact that c_θ is non-zero, divide both sides of Eq. (24) by c_θ and then the desired pitch attitude can be computed as:

$$\theta_{des} = \tan^{-1} \left(\frac{\ddot{x}_{des} c_\psi - \ddot{y}_{des} s_\psi}{\ddot{z}_{des} + g} \right) \tag{27}$$

Equation (26) multiplied by s_θ, subtracted from (Eq. (25), multiplied by c_ϕ) yields

$$\frac{1}{m} T_B s_\phi = \ddot{x}_{des} s_\psi - \ddot{y}_{des} c_\psi \tag{28}$$

Utilizing Eq. (22), the following relationship will have:

$$[\frac{1}{m} T_B z_e]^T \left[\frac{1}{m} T_B z_e \right] = [\ddot{p_{des}} + g z_e]^T [\ddot{\mathbf{P}}_{\boldsymbol{des}} + g z_e] \tag{29}$$

which is equivalent to:

$$\ddot{x}_{des}^2 + \ddot{y}_{des}^2 + (\ddot{z}_{des} + g)^2 = \left(\frac{1}{m} T_B \right)^2 \tag{30}$$

Combining Eq. (30) with Eq. (28) gives the roll attitude expression:

$$\phi_{des} = \sin^{-1} \left(\frac{\ddot{x}_{des} s_\psi - \ddot{y}_{des} c_\psi}{\sqrt{\ddot{x}_{des}^2 + \ddot{y}_{des} + (\ddot{z}_{des} + g)^2}} \right) \tag{31}$$

In regard to the earth-fixed reference frame, the non-holonomic constraints between the roll, pitch and yaw angles, and the linear accelerations, that related the objective control with the control variables are provided by Eqs. (27) and (31). The angles (ϕ, θ) in Eqs. (27) and (31) depend on the yaw angle and on the movements along (x, y) axes with their dynamics.

where yaw is the desired yaw attitude $\psi = 0$, the zero value constraints the net aerodynamic torque, and thus the angular acceleration about the yaw axis, and make it equal exactly zero.

Thus, for the preferred acceleration through preferred *x, y*, and *z* directions, the following equations have been used [46]:

$$\ddot{x}_{des} = \frac{K_{Px}e_x + K_{Ix}\int e_x - K_{Dx}\dot{x}_{meas}}{m} \tag{32}$$

$$\ddot{y}_{des} = \frac{K_{Py}e_y + K_{Iy}\int e_y - K_{Dy}\dot{y}_{meas}}{m} \tag{33}$$

$$\ddot{z}_{des} = \frac{K_{Pz}e_z + K_{Iz}\int e_z - K_{Dz}\dot{z}_{meas}}{m} \tag{34}$$

A translational movement in (*x, y,* and *z*) directions as well as rotation over (roll, pitch, and yaw) axes has been accomplished by controlling the velocity and, as a result, the thrust of each rotor. Because the quadrotors are under-actuated, significant coupling between rotary and translational movements occurs. Consequently, controlling the quadrotor's trajectory tracking or position is a difficult challenge.

5 Extremum Seeking (ES) Control

Extremum Seeking (ES) is a highly effective real-time optimization technique. Although the first ES-based power transmission mechanism was invented in 1922 [49], understanding the behavior of ES plants from a system theory perspective often necessitates the use of modern technologies. The study [50] in 2000 provided the first rigorous stability analysis of extremum searching, shedding new light on ES through the use of averaging theory.

Different kinds of extremum searching algorithms [51] extensions to discrete time [52], and semi-global stability results [53] have all been developed since then. Numerous applications, such as engine calibration [54], drag reduction [55], and plasma management in tokamaks [56, 57], have lately been proposed as a result of these theoretical discoveries.

Extremum searching is a non-model based, real-time optimization approach that develops a gradient estimate using zero-mean perturbations. The state variable is then shifted closer to the extremum using the gradient estimate. Some applications, as tracking can benefit from the predictability associated with periodic perturbations, e.g. a vehicle is navigated by a periodic extremum seeking algorithm, so it can become a predictable target. Meanwhile biological systems, as bacterial chemotaxis, are rarely use periodicity in locating optima. Furthermore, if the system is complex, the orthogonality prerequisites for the members of the periodic perturbation vector can be difficult to meet. Thus, exploring the application of stochastic perturbations inside the ES architecture is valuable.

5.1 Auto-Tuning of PI-D Controller Based on Extremum Seeking Algorithm

ES minimizes a cost function that measures the PI-D controller's performance, iteratively altering the cost function's arguments (the PI-D parameters) until the output hits a minimizer of $J\left(\overrightarrow{\vartheta}\right)$, which is not necessarily a global point. ES is used to adjust a controller's PI-D parameters in order to optimize the response map in the smallest period of time. Even though it can be utilized with a typical PI-D controller, the derivative term affects the measured output of the plant and does not affect the reference signal. Because the derivative output will be an impulse signal when the input is a step signal, so, it keeps the system in the controlled region because it prevents motors reaching to saturation mode.

The cost function is derived based on error signal of the closed-loop system. It is usually used by ES algorithm to improve the controller parameters. The input of cost function is modified frequently by a non-model-based strategy of the ES algorithm until the output of cost function reaches maximum or local minimum. ES algorithm perform these requirements by disturbing the input variables of system to create an online estimation of the output gradient as a function of inputs. Then, these gradient data are used to frequently change the input parameters toward the output's minimum [58].

The cost function $J\left(\overrightarrow{\vartheta}\right)$ that is used to measure the efficiency of a given PI-D controller, is rated during the application of the ES-PI-D tuning experiment. The integral square error (ISE) cost function is used in this study [59]:

$$J\left(\overrightarrow{\vartheta}\right) = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} e(t, \overrightarrow{\vartheta})^2 dt \tag{35}$$

where the error $e\left(t, \overrightarrow{\vartheta}\right) = r(t) - y(t, \vartheta)$ represents the difference between the reference signal and the output signal that belongs to the closed-loop system and includes the effect of PI-D parameters. The error is taken into account by the cost function $J\left(\overrightarrow{\vartheta}\right)$ described in Eq. (35) across the time span $[t_0, t_f]$.

By choosing t_0 to reach the moment T peak at which the closed-loop system's step response hits the first peak, the cost function $J\left(\overrightarrow{\vartheta}\right)$ effectively sets zero weighting on the initial transient component of the response. As a result, the controller is

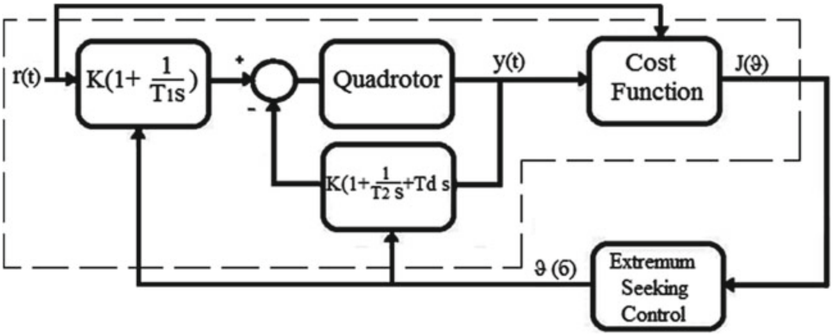


Fig. 5 The block diagram of the applied ES-PID Quadrotor controlled system

configured to reduce error beyond T_{peak} while ignoring initial transient restrictions. For the simulations, t_f was set to the simulation duration.

The PI-D controller is built depending on Eq. 12, where K, T_i, and T_d are given by:

$$\overrightarrow{\vartheta} \triangleq [K_P, K_I, K_D]^T \tag{36}$$

Figure 5 shows the performed closed-loop controlled system and the controller is parameterized as [60]

$$C_r(s) = K\left(1 + \frac{1}{T_{i1}s}\right) \tag{37}$$

$$C_y(s) = K\left(1 + \frac{1}{T_{i2}s} + T_d s\right) \tag{38}$$

where r is the reference signal, y is the output signal and u, is the control signal. The y-signal is regulated to the r signal by the 2DOF controller, C_r and C_y.

The implementation of the extremum seeking algorithm can be described as follows [60]:

$$\zeta(6) = -h\zeta(6-1) + J\left(\overrightarrow{\vartheta}(6-1)\right) \tag{39}$$

$$\widehat{\vartheta}_i(6+1) = \widehat{\vartheta}_i(6) - \gamma_i\alpha_i cos(\omega_i 6)[J\left(\overrightarrow{\vartheta}(6)\right) - (1+h)\zeta(6)] \tag{40}$$

where $\zeta(6)$ it is a scalar, i is the subscript of i_{th} entry of a relevant vector, α_i is the perturbation amplitude while 6 is the iteration number and γ_i is the adaptation gain. The input parameters $\widehat{\vartheta}_i(6)$ are perturbed by the signal $\alpha_i cos(\omega_i 6)$. The output from the cost function $J\left(\overrightarrow{\vartheta}(6)\right)$ is filtered by high pass filter, demodulated, and then, filtered by low-pass filter to get new input parameters.

The values of α affect the Convergence and stability of the system. This, in turn, affects cost function $J\left(\overrightarrow{\vartheta}\right)$ shape to reach the minimum. The modulation frequency ω_i is selected in a way that $\omega_i = a^i\pi$, where a must be rational, by satisfies $0 < a < 1$ to guarantee Nyquist criterion satisfied. Moreover, the high-pass filter (HPF) $(z-1)/(z+h)$ is designed with $0 < h < 1$ with a cut-off frequency lower than the modulation frequency ω_i.

The PI-D controller parameters ϑ (k) are updated by ES algorithm while minimizing the cost function $J\left(\overrightarrow{\vartheta}\right)$. The ES algorithm utilizes the cost function value $J\left(\overrightarrow{\vartheta}(6)\right)$ so that the new controller parameter $\overrightarrow{\vartheta}(6)$ is calculated. The structure of proposed PID control system consists of 2DOF controllers for each parameter (x, y, z, roll, pitch, and yaw) in both feedback and feedforward at the reference signal, as shown in Fig. 5.

5.2 Stability Analysis

Extremum seeking control was subjected to some early stability analyses, such as in [61]. In [50], the first significant study of ESC stability analysis for generalized non-linear dynamics was reported, and it will be briefly described in the following form:

$$\dot{x} = f(x, u) \tag{41}$$

$$y = h(x) \tag{42}$$

it is assumed that there is a smooth control law:

$$u = \alpha(x, \vartheta) \tag{43}$$

The control law u can be described as a function of ϑ, where it can be assumed to act just like a static state feedback law. Consequently, Eq. (41) can be written as:

$$\dot{x} = f(x, \alpha(x, \vartheta)) \tag{44}$$

and is parametrized by ϑ.

The following assumptions are made about the existence and stability of an equilibrium point [50]:

Assumption 1 There exists a smooth function l: $R \rightarrow R_m$ such that.

$$f(x, \alpha(x, \vartheta)) = 0 \, \text{if and only if} x = l(\vartheta) \tag{45}$$

Assumption 2 For each $\vartheta \in R$, the equilibrium $x = l(\vartheta)$ of the system $\dot{x} = f(x, \alpha(x, \vartheta))$ is locally exponentially stable with decay and overshoot constants uniform in ϑ.

The function composition $(h \circ l)(\vartheta) = h(l(\vartheta))$ represents the system's steady-state static map and is the subject of this assumption, which is required for the static map to have a maximum.

Assumption 3 There exists $\vartheta^* \in R$ such that:

$$\frac{\partial}{\partial\vartheta}(h \circ l)(\vartheta^*) = 0 \tag{46}$$

$$\frac{\partial^2}{\partial\vartheta^2}(h \circ l)\left(\vartheta^*\right) < 0 \tag{47}$$

The standard Band Pass Filter (BPF) filter technique is known to be locally stable under these assumptions, with a minimal perturbation signal frequency, perturbation signal amplitude, and integrator gain (i.e., exponentially converges to the neighborhood of the optimal point).

The equilibrium point for the system in Eq. (41) is computed, and the Jacobian of the reduced system is determined to be Hurwitz at the equilibrium. This indicates that the reduced system is locally stable (exponentially converges to the neighborhood of the optimal point).

The boundary layer model is then subjected to a singular perturbation, and exponential convergence will be obtained. Then, by using Lyapunov's Theorem [62], overall system stability can be determined by combining the reduced system's stability with the system's boundary layer stability. That will only guarantee the system convergence to near the optimum if it is begun in a sufficiently small neighborhood of it and the region of attraction is unknown. Furthermore, it does not guarantee that the maximum of the static map is unique; it just assures that there is a ϑ^*, which signifies a maximum. Accordingly, the next critical step in the ESC stability investigation is considered, where the first non-local stability analysis of extremum seeking control for the traditional BPF method, as well as various ESC variations like without pass filters and without a high pass filter, have been examined [63]. Accordingly, an additional assumption is introduced to ensure the maximum of the static map at ϑ^*.

Assumption 4 There exists $\vartheta^* \in R$ such that:

$$\frac{\partial}{\partial\vartheta}(h \circ l)(\vartheta^*) = 0, \frac{\partial^2}{\partial\theta^2}(h \circ l)\left(\vartheta^*\right) < 0 \tag{48}$$

$$\frac{\partial}{\partial \vartheta}(h \circ l)\big(\vartheta^* + \vartheta_\Delta\big) < 0 \forall \vartheta_\Delta \neq 0 \tag{49}$$

It's worth noting that Eq. (49) is an important condition for the unique maximum and one crucial point ($\frac{\partial}{\partial \vartheta}(h \circ l)(\vartheta^*) = 0$ only when $\vartheta = \vartheta^*$).

6 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) incorporates aspects inspired by natural swarm social behavior and linkages with evolutionary computation. PSO has been widely used in difficult optimization domains and is a popular area of research, providing an alternative to more established evolutionary computation approaches that can be used in many applications. It is an offline population that is based on the artificial intelligence (AI) optimization technique that Kennedy and R. Eberhart first proposed [64]. It is a heuristic approach that can be used to solve nonlinear and noncontiguous problems with continuous variables. To perform optimization, the algorithm was simplified and observed. The PSO algorithm is capable of searching a huge variety of available solutions. Metaheuristics like PSO, on the other hand, do not guarantee that the best solution will be identified.

The particles fly in a multidimensional search space of a problem in the PSO algorithm, starting with a random value to start the population. Every particle, based on its own experience, evaluates the cost function at its current position (x) to identify the optimum position (P_{best}), after which the particle can estimate its velocity (V) using the search space. The global best solution (G_{best}) for each generation will be discovered by modifying the flying experience of each particle about surrounding particles [65] . After all, particles have been transferred, the new iteration begins.

The position and velocity of a particle are computed as follows:

$$x^{q+1} = x^q + 6 * V^{q+1} q = 1, 2, \ldots n \tag{50}$$

$$V^{q+1} = w_P * V^q + c_1 * rand * \big(P_{best} - x^q\big) + c_2 * rand * \big(G_{best} - x^q\big) \tag{51}$$

where x^q, x^{q+1}, V^q *and* V^{q+1}: are respectively the current position, modified position, current velocity, and modified velocity. c_1, c_2, $rand$, w_P, P_{best}, G_{best} and 6: are respectively two positive constants, randomly generated numbers within [0,1] range, inertia weight, particle best-known position, swarm best-known position and constant equals 1 which make the addition possible between position and velocity.

The Algorithm

1. Within the search space, input the starting random velocities and locations of the population array particles in D-dimensions.
2. Begin looping.

3. For each particle, determine the needed cost function.
4. Update the particle cost analysis. If the current value is greater than the previous one, it is kept; otherwise, it is moved on.
5. Using the P for each particle and the overall generation, determine the best success inside the search space.
6. Based on Eqs. (52) and (53), update the velocity and position of the particle.

$$ISE = \int_{0}^{\infty} e(t)^2 dt \tag{52}$$

$$ITSE = \int_{0}^{\infty} te(t)^2 dt \tag{53}$$

7. In case the criterion is met (which is either sufficiently good fitness or reaches a maximum number of iterations), then exit the loop. Otherwise go to step 2.
8. End loop.

7 Simulation Results

The simulation of the proposed approach to control the quadrotor and tune the PI-D controller as well as the obtained results will be presented in this Section. MATLAB/ Simulink has been implemented for this purpose.

By using the quadrotor dynamics as described in Eqs. (4, 5, 6, 27 and 31), a Simulink model has been developed, as shown in Fig. 6, to simulate the attitude stabilization and position tracking of a quadrotor using different paths. In addition to utilizing ES to auto-tuning the PID parameters. In the followings, each block will be described.

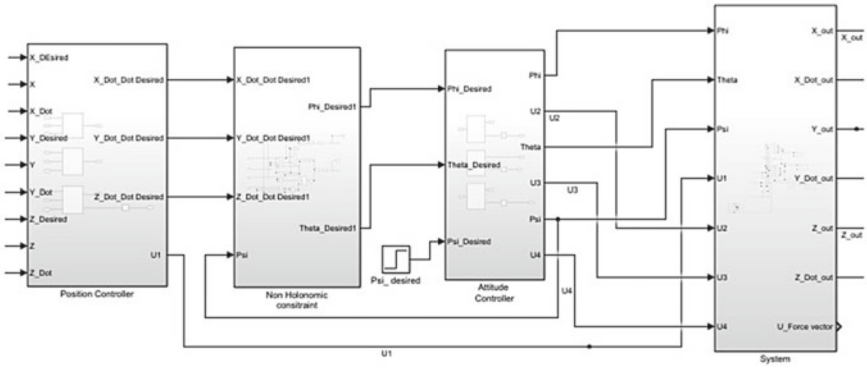


Fig. 6 Developed control system model for the quadrotor using Simulink

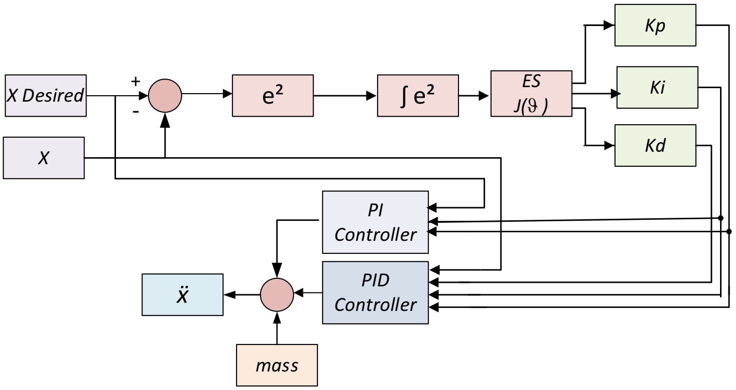


Fig. 7 Position control subsystem of *x*-axis

7.1 Position Control

The “position controller” subsystem presents the control law (U_1), as described in Eq. (13). The inputs of this block are the state inputs and path command for three axes (x, y, z), and the outputs are the altitude command that is then fed to the attitude controller. Figure 7 shows the entire position control for x-axis. Similarly, the subsystems for (y and z) axes are developed. It can be seen that the controller computes the feedback error by using Eq. (10).

These PI-D control laws, are used to improve the quadrotor position controller, and they are considered as the lower-level of control, that is utilized for steering of the quadrotor in x, y, and z axes. The controller output is then returned back into the non-Holonomic constraint subsystem to compute the desired rotational angles.

7.2 Non-Holonomic Constraint

The second block of Fig. 7 is the “Non-Holonomic Constraint” subsystem that implements the position output and computes the desired rotation angles. The subsystem is built by using Eqs. (27) and (31). The controller output is then input to the higher-level controller for quadrotor attitude stabilization.

Attitude controller

The third subsystem is the “attitude controller” which implements the remaining control laws for stabilization of the attitude. The subsystem consists of three sub-blocks, for three angles of rotation, each subsystem implements an ES-PI-D controller Eqs. (27) and (31). Each of these blocks generates an adjusted value of roll, pitch, and yaw, as well as altitude. All these values represented the controller outputs that are sent to the quadrotor dynamics subsystem for further processing. Figure 8 shows the roll angle with ES-PID controller. The subsystems for (pitch and yaw) angles are analogous to the subsystem of the roll-angle, so its diagram has been crossed.

The main quadrotors' parameters that are utilized in the simulation environment are presented in Table 1. While the selecting parameters for ES tuning described in Table 2.

All simulations have used the values of (10, 13, and 6) m as the desired position for x, y, and z coordinates. The quadrotor was driven to the desired location using the previously discussed PI-D controller, and the controller's driving error and

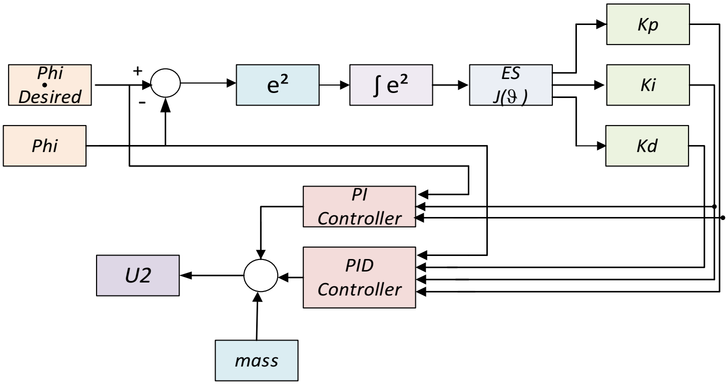


Fig. 8 Attitude Control Subsystem for 2DOF controller with ES of roll angle

Table 1 The quadrotor simulation parameters

Name of parameter	Value and unit
Time of simulation	20 s
Mass (m)	0.8 kg
Gravity (g)	$9.81(\text{m/s}^2)$
Length of the rods (L)	0.25 m
Propellers drag coefficient (b)	$3.13 * 10^{-5}$
Inertia matrix (I)	$[5 * 10^{-3} 5 * 10^{-3} 10 * 10^{-3}]$

Table 2 The ES-PID control parameters [66]

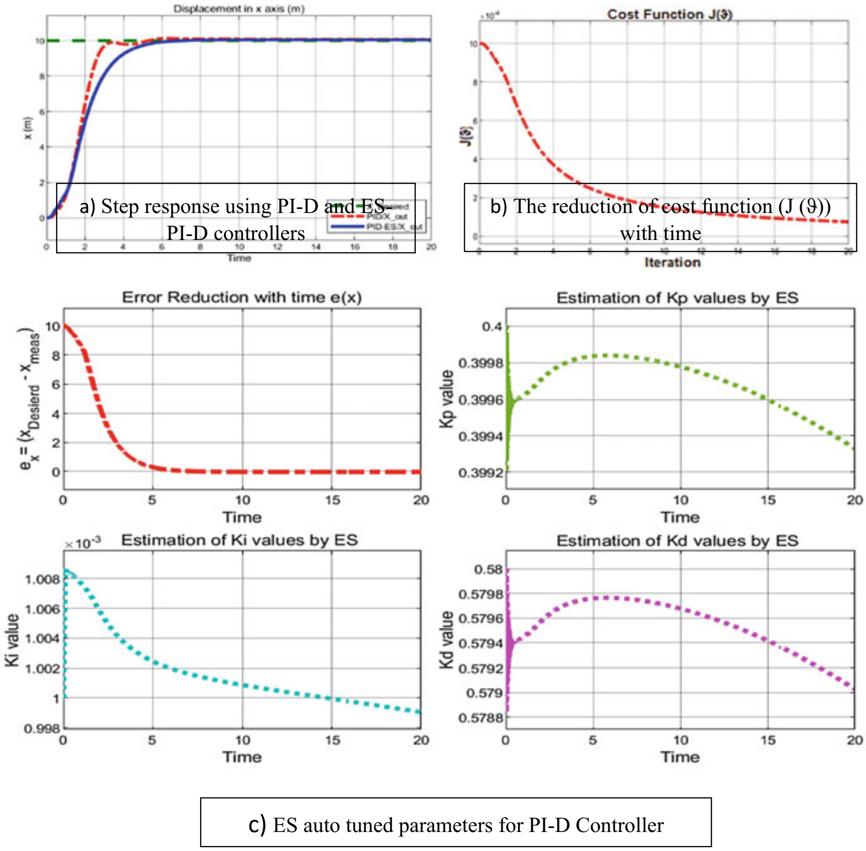
Name of parameter	Value and unit
Adaptation gain (γ_i)	[2 1 2]
Perturbation amplitude (α_i)	[0.4 0.001 0.58]
$[to\,tf]$	[020]s
Modulation frequency (ω_i)	$[a\pi a^2\pi a^3\pi]$
Scalar (a)	0.009

response time were compared. Additionally, disturbance and uncertainty, two turbulence elements that might impact the behavior of quadrotors (or most systems), have been taken into account.

Firstly, the improvement of the PI-D controller has been investigated when combined with ES. Three cases have been taken into consideration. These are: displacement results, disturbance rejection performance, and effect of the system uncertainty.

7.3 Displacement Simulation Results

The quadrotor displacement has been simulated and comparison has been established between the results using only PI-D controllers and that with ES-PI-D controllers. The results are shown in Figs. 9, 10 and 11.

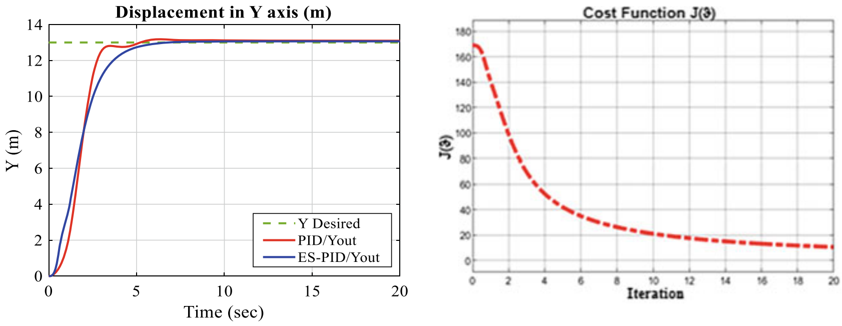


a) Step response using PI-D and ES-PI-D controllers

b) The reduction of cost function (J (ϑ)) with time

c) ES auto tuned parameters for PI-D Controller

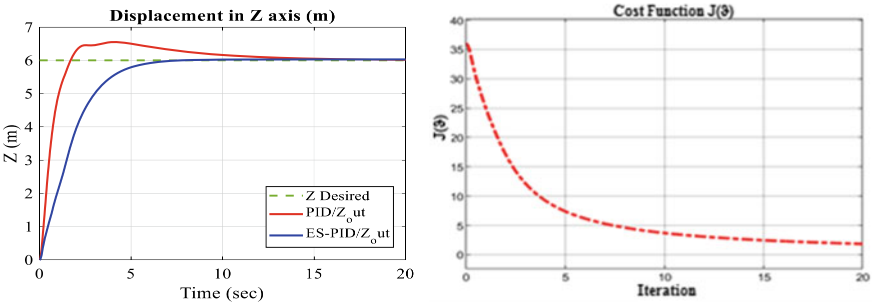
Fig. 9 Quadrotor displacement in the x-axis using PI-D and ES-PI-D controllers



a) Step response using PI-D and ES-PI-D controllers

b) The reduction of cost function J ϑ with time

Fig. 10 Quadrotor displacement in the y-axis using PI-D and ES-PI-D controllers



a) Step response using PI-D and ES-PI-D controller

b) The reduction of cost function J ϑ with time

Fig. 11 Quadrotor displacement in the z-axis using PI-D and ES-PI-D controllers

As can be noticed in Figs. 9a, 10a and 11a, each of the implemented controllers steer the quadrotor from initial position x, y, z = [0, 0, 0] to the target of [6, 10, 13]. Nevertheless, the application of the ES-PI-D controller can provide more accurate results than using only PI-D controller. In addition, the ES-PI-D control is uniformly raised until reaching the final steady state, which is about 10.05 m (i.e., steady state error equals to 0.05 m) within 9 s, whereas the results when using PI-D controller is more varying and takes about 12 s to reach steady state of 10.08 m (i.e., steady state error about 0.08 m). Consequently, Figs. 9b, 10b and 11b show that the controller didn't achieve a minimum cost function. This is an inherent drawback of ES, the algorithm never achieves the true global optimum. Figure 12c, illustrates the tuning and correction of the parameters (K_p, K_i, K_d) of the PI-D controllers via ES until reaching steady state conditions.

According to Fig. 12a, that shows the ES-PI-D control's reaction, the angle of roll oscillates consistently for around 4.7 s before becoming stable at the desired angle. The pitch angle is oscillating until it stabilizes at the appropriate angle, which takes around 4.6 s, according to Fig. 12b. Figure 12c also shows that the psi angle remains at zero after a brief amount of time, roughly 0.2 s. The control signals in Fig. 13a–d have responses that are not at all consistent with the behaviours of the parameter.

Similarly, Fig. 14 shows the results when using sinusoidal reference signals for the *x, y* and *z* directions. It can be seen that by using both of the controllers the quadrotor tracks the desired position of (sin2t). However, it can be noticed that ES-PI-D controlled quadrotor performs more accurately than when using only PI-D control. Accordingly, the ES-PI-D controller has better angle tracking capability than a traditional PI-D controller. The ES-PI-D displays higher oscillation inhibition, less tracking error, and more significant anticipated angle tracking performance, demonstrating improved resilience and stability. While, in Fig. 15, it is clear how the control signal is changing with response for the desired displacements.

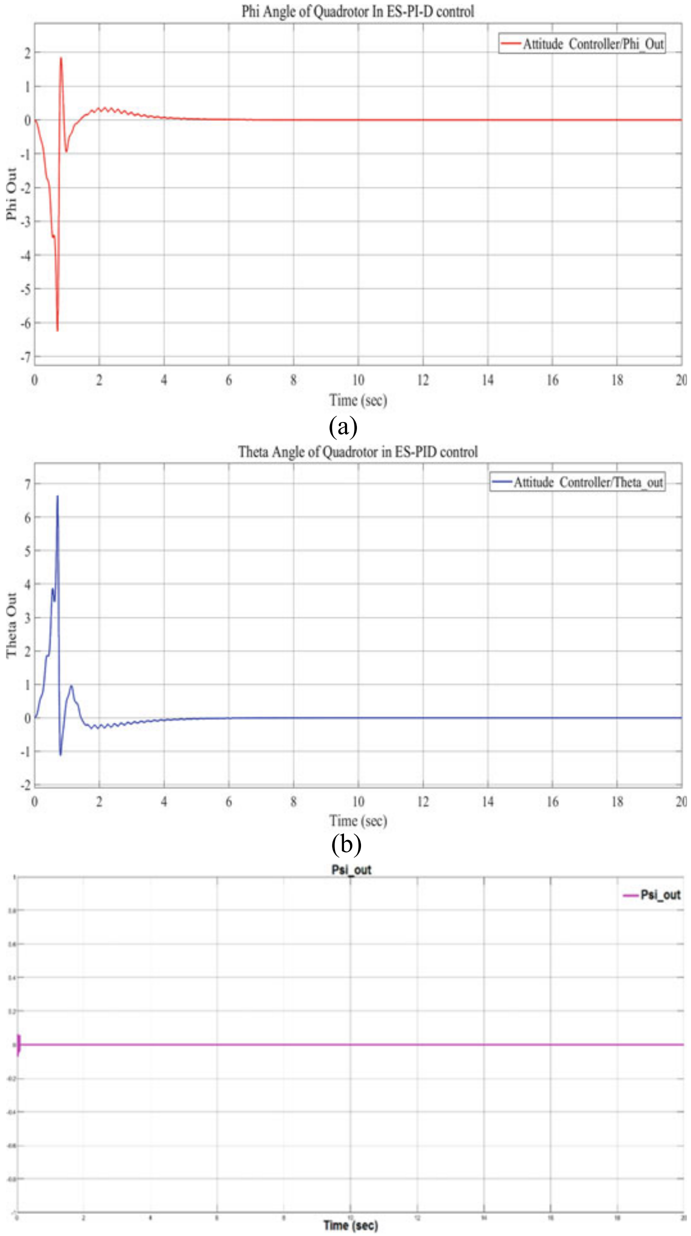
7.4 Disturbance Rejection Performance of the Controllers

The robustness of the proposed control approach is tested in this part against a sudden disturbance effect at time 10 s. The test includes applying both PI-D and ES PI-D controllers. The results are presented in Fig. 16.

From the simulation results in Fig. 16, it can be seen that the influence of disturbance starts from the 10th second. Each of the controllers has overcome the disturbance effect and drives the quadrotor to target. Meanwhile, it can be seen that ES-PI-D is faster than PI-D in attenuating the disturbance effects, as in Fig. 17a, b. In Fig. 17c, both of the ES-PI-D and the PI-D controlled systems are starting to eliminate the disturbance effect from about 10th second and reach the desired value in the 17th second. It can be summarized that the ES-PI-D controller performs better in disturbance rejection than PI-D only controller.

7.5 Robustness Against system's Uncertainties

Uncertainty is a challenging issue to the design of a robust controller for a system, especially when there is a limited information on the type and features of the possible uncertainties. For the system under study, the mass can be considered as a combination of a nominal value and a certain amount of uncertainty. Accordingly, a significantly large values of variations within (20-90%) of the nominal value of the 0.8 kg

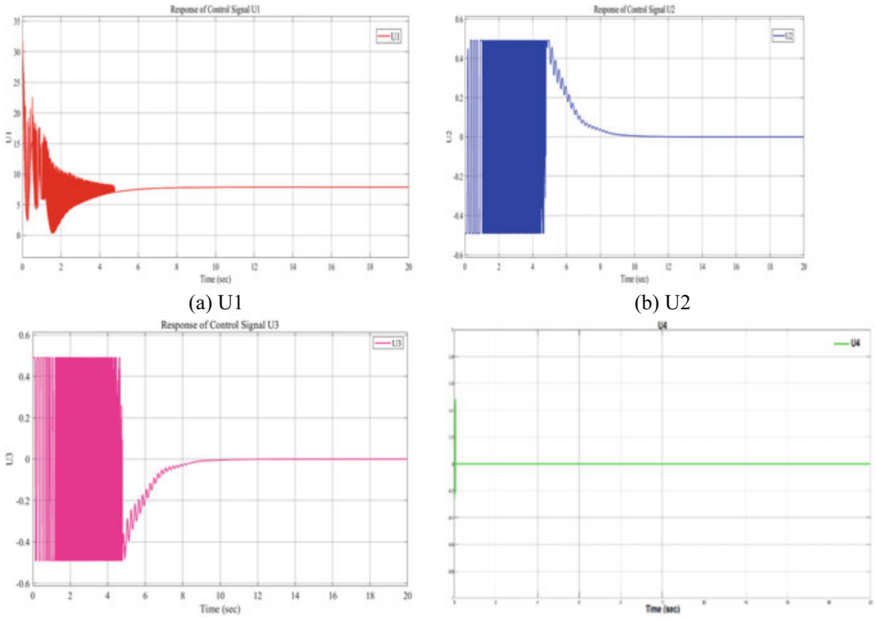


(a)

(b)

(c)

Fig. 12 Simulation results for a quadrotor rotation for ES-PI-D control. (a) $\boldsymbol{\phi}$, (b) $\boldsymbol{\theta}$, (c) ψ



(a) U1 (b) U2

(c) U3 (d) U4

Fig. 13 The simulated control signals

mass has been considered here. The results shown in Fig. 17 illustrates the effects of including the uncertainties while applying the previously designed PI-D and ES-PI-D controllers.

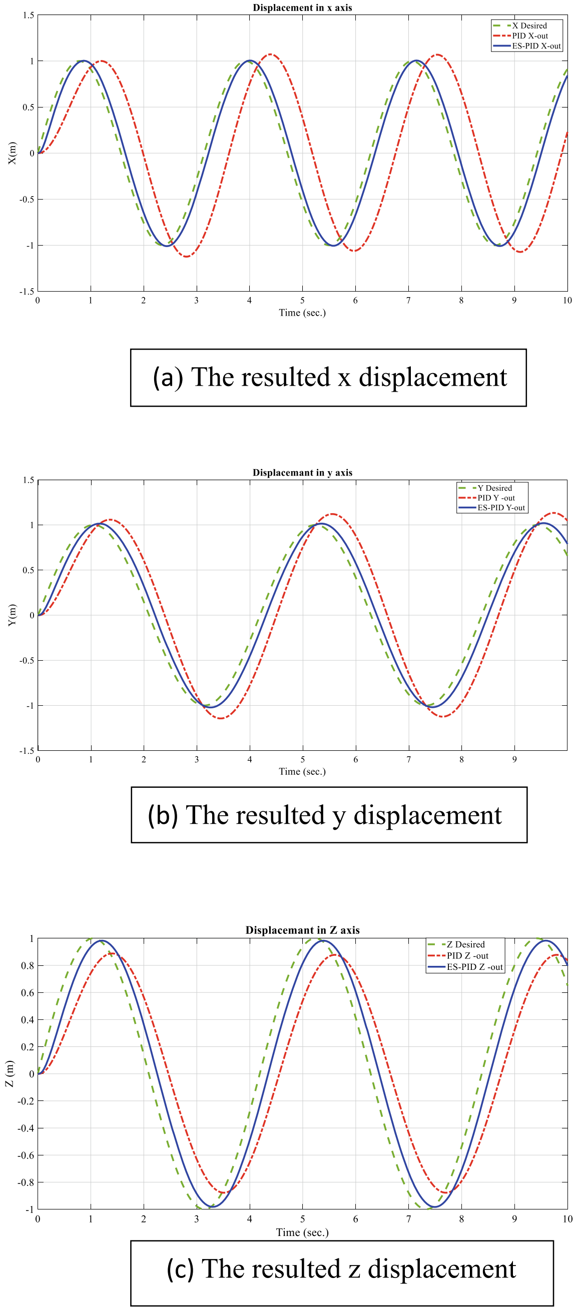
It can be clearly shown from Fig. 17 that both of controllers can derive the quadrotor to the desired position. Nevertheless, it is clear that ES- PI-D controller provides better performance as compared with ordinary PI-D controller. Table 3 presents a description of the main differences between PI-D, and the ES-PI-D controller in terms of steady-State error E_{ss} and settling time T_s.

7.6 PID tuning using PSO

A comparison of the proposed ES method with that utilizing PSO algorithm for auto-tuning PID controller is made by using the same quadcopter system. The simulation results are shown in Fig. 18.

As can be seen from Fig. 18, the PID-PSO model is also successfully tuned (K_p, K_i, K_d) of PI-D controller, but the simulation takes a longer time to reach steady state performance, where it took 8.25 s in the x-direction, about 10 s in y and z-direction. Nevertheless, although the PSO algorithm has the power to make auto-tune for PID controller parameters, it has a slow rate of convergence to the optimal solution with

Fig. 14 The resulted performance of the quadrotor using PI-D and ES-PI-D controllers; **a** x displacement **b** y displacement **c** z displacement



(a) The resulted x displacement

(b) The resulted y displacement

(c) The resulted z displacement

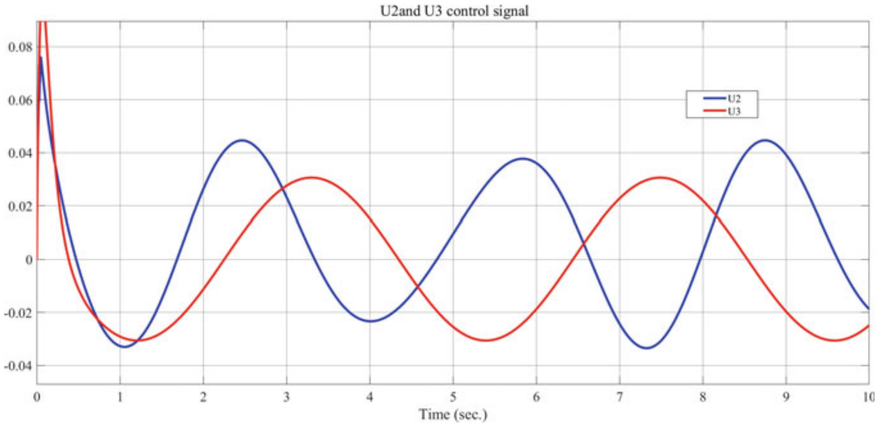
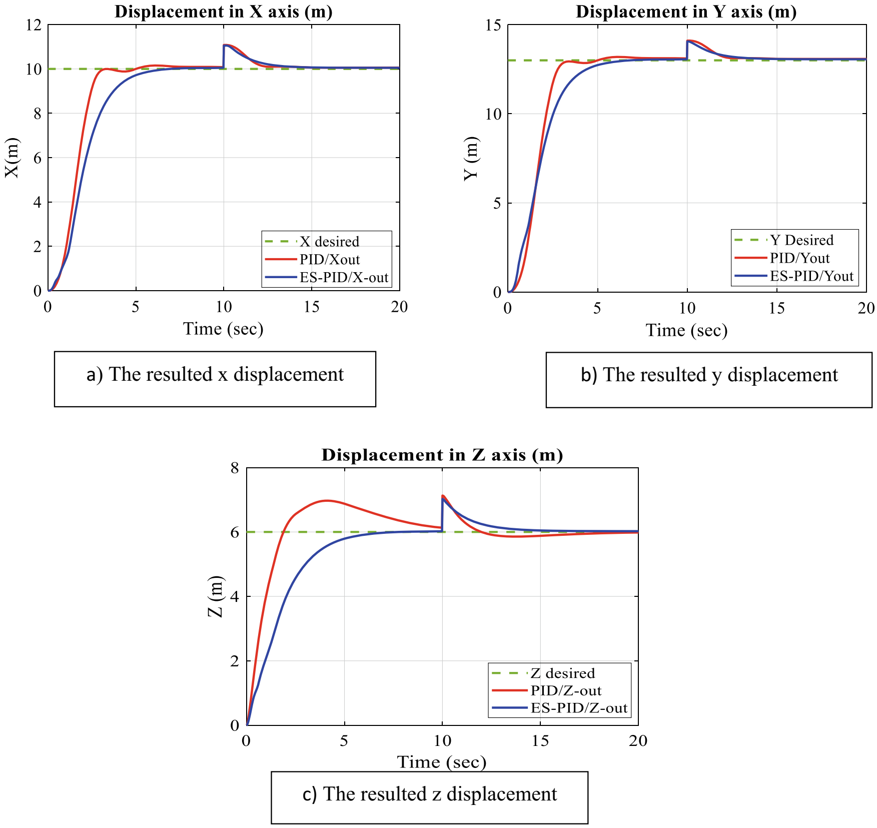


Fig. 15 The control signals U2 and U3

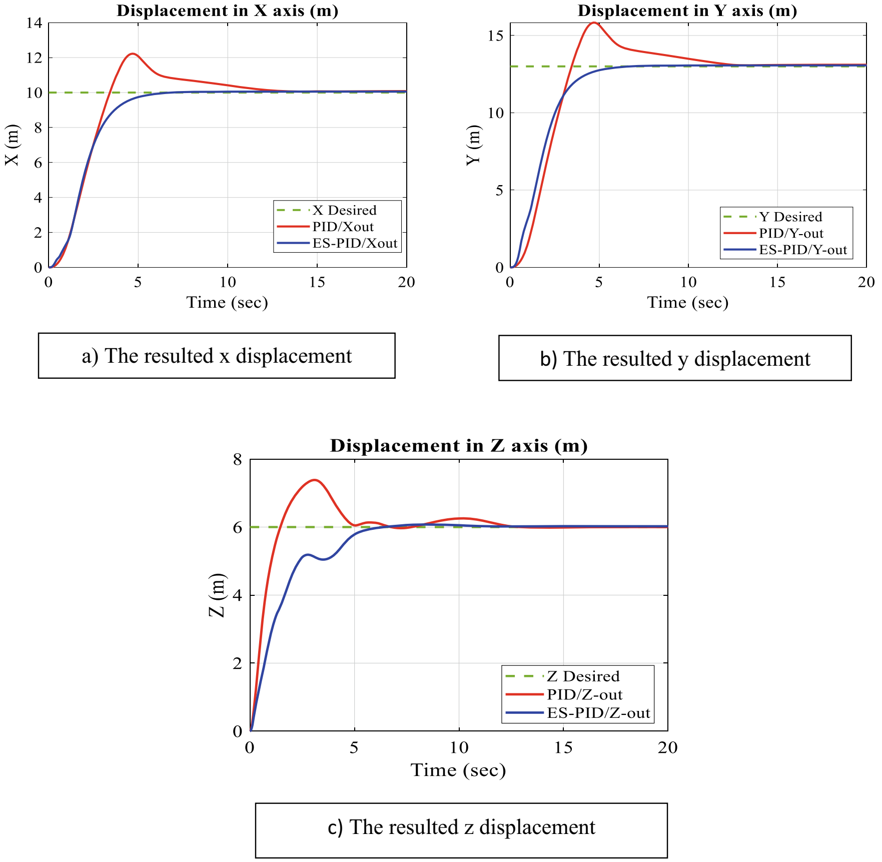


a) The resulted x displacement

b) The resulted y displacement

c) The resulted z displacement

Fig. 16 Simulation Results of the Quadrotor Controlled System Under Disturbance Effects, Using PI-D and ES-PI-D controllers



a) The resulted x displacement

b) The resulted y displacement

c) The resulted z displacement

Fig. 17 The performance of the Quadrotor in presence of uncertainty in the mass, under PI-D and ES-PI-D controllers: **a** x-axis, **b** y-axis, **c** z-axis

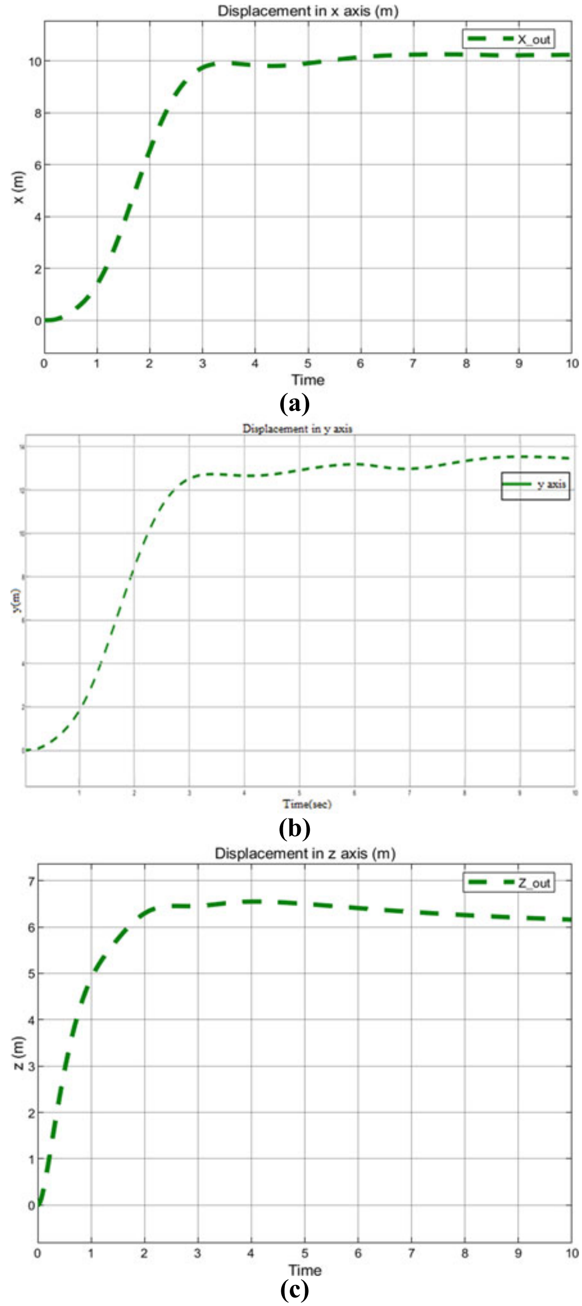
Table 3 Comparison in presence of parameter uncertainty between the ES-PI-D and PI-D controllers

	PI-D		ES-PI-D	
	E_{ss} (m)	T_s (s)	E_{ss} (m)	T_s (s)
x	0.06	8	0.02	7
y	0.02	8	0.02	7
z	0.2	10	0.02	7

offline tuning as compared with the ES algorithm that can effectively optimize the parameters online.

Table 4 illustrates the differences between the three applied controllers; PI-D, PID-PSO, and the ES PI-D, in terms of overshoot, steady state error, and settling time.

Fig. 18 Simulation results for quadrotor displacement auto-tuned by PSO algorithm in **a** *x*-direction, **b** *y*-direction, and **c** *z*-direction



(a)

(b)

(c)

Table 4 The Comparison of the slandered results and disturbance rejection results between PI-D, PID-PSO, and ES-PI-D

		Normal operation			Operation in presence of disturban		
		x	y	z	x	y	z
PI-D	Overshoot value (m)	0	0.2	0.7	0.2	0.2	0.6
	SteadyState error(m)	0.08	0.1	0.02	0.06	0.06	0.02
	Setting time(s)	12	11	16	11	10	18
PID-PSO	Overshoot value (m)	0	0.2	0.6	0.1	0.1	0.4
	SteadyState error(m)	0.08	0.09	0.02	0.06	0.06	0.02
	Settling time (s)	12	11	15	9	9.2	16
ES-PI-D	Overshoot value (m)	**0**	**0**	**0**	**0**	**0**	**0**
	SteadyState error(m)	**0.05**	**0.05**	**0.02**	**0.02**	**0.02**	**0.02**
	Settling time(s)	**9**	**9**	**10**	**7.6**	**7.6**	**12**

It can be clearly noticed the effect of using the ES in improving the performance of the quadrotor-controlled system.

8 Conclusion

In this chapter, the design of a stable quadrotor control system has been presented, where the parameters of PI-D controller can be tuned in real-time operation via ES optimization algorithm. Stability, robustness, and improved performance have been observed. The simulation results showed that the developed ES-PI-D controller is considerably more effective than only PI-D controller at transient and steady-state performance, even in presence of uncertainty and disturbances. This is due to the effect of the ES cost function that is related to the square error. where, faster reduction of total error can be achieved. As a comparison with the implementation of another efficient optimization algorithm, PSO algorithm has been utilized to tune PID controller as well. It has been shown that ES approach has achieved better stability and performance results as compared with PSO method. Further study will investigate including constraints in the ES optimization problem. In addition, another nonlinear controller as the sliding mode controller will be combined with the ES algorithm to further improve the performance and stability of the controlled system.

References

1. Hussain R, Zeadally S (2019) Autonomous cars: research results, issues, and future challenges. IEEE Commun Surv Tutorials 21(2):1275–1313. https://doi.org/10.1109/COMST.2018.2869360
2. Kumar PV, Challa A, Ashok J, Narayanan GL (2016) GIS based fire rescue system for industries using Quad copter—A novel approach. 2015 international conference on microwave, optical and communication engineering, ICMOCE 2015. IEEE, pp 72–75, https://doi.org/10.1109/ICMOCE.2015.7489693
3. Bashi OID, Wan Hasan WZ, Azis N, Shafie S, Wagatsuma H (2017) Quadcopter sensing system for risky area. Proceedings of the 2017 IEEE regional symposium on micro and Nanoelectronics, RSM 2017. IEEE, pp 216–219, https://doi.org/10.1109/RSM.2017.8069152
4. Saha HN et al (2018) A low cost fully autonomous GPS (Global Positioning System) based quadcopter for disaster management. 2018 IEEE 8th annual computing and communication workshop and conference, CCWC 2018. vol 2018-Janua. IEEE, pp 654–660, https://doi.org/10.1109/CCWC.2018.8301782
5. Navia J, Mondragon I, Patino D, Colorado J (2016) Multispectral mapping in agriculture: Terrain mosaic using an autonomous quadcopter UAV. 2016 international conference on unmanned aircraft systems, ICUAS 2016. IEEE, pp 1351–1358, https://doi.org/10.1109/ICUAS.2016.7502606
6. Daroya R, Ramos M (2018) NDVI image extraction of an agricultural land using an autonomous quadcopter with a filter-modified camera. Proceedings—7th IEEE international conference on control system, computing and engineering, ICCSCE 2017, vol 2017-Novem. IEEE, pp 110–114, https://doi.org/10.1109/ICCSCE.2017.8284389
7. Camci E, Kayacan E (2017) Waitress quadcopter explores how to serve drinks by reinforcement learning. IEEE Region 10 annual international conference, proceedings/TENCON. IEEE, pp 28–32, https://doi.org/10.1109/TENCON.2016.7847952
8. Kim SJ, Jeong Y, Park S, Ryu K, Oh G (2018) “A survey of drone use for entertainment and AVR (Augmented and Virtual Reality),” Augmented reality and virtual reality. Springer International Publishing, pp 339–352, https://doi.org/10.1007/978-3-319-64027-3_23
9. Ghamry KA, Zhang Y (2016) Cooperative control of multiple UAVs for forest fire monitoring and detection. MESA 2016—12th IEEE/ASME international conference on mechatronic and embedded systems and applications—conference proceedings. IEEE, https://doi.org/10.1109/MESA.2016.7587184
10. Hou Z, Wang W, Zhang G, Han C (2017) A survey on the formation control of multiple quadrotors. 2017 14th international conference on ubiquitous robots and ambient intelligence, URAI 2017. IEEE, pp 219–225, https://doi.org/10.1109/URAI.2017.7992717
11. Hadi N, Ramz A (2017) Tuning of PID controllers for quadcopter system using hybrid memory based gravitational search algorithm—particle swarm optimization. Int J Comput Appl 172(4):9–18. https://doi.org/10.5120/ijca2017915125
12. Alsharif MA, Arslantas YE, Holzel MS (2017) A comparison between advanced model-free PID and model-based LQI attitude control of a quadcopter using asynchronous android flight data. 2017 25th mediterranean conference on control and automation (MED). IEEE, https://doi.org/10.1109/med.2017.7984252
13. Åström KJ, Hägglund T (1984) Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica 20(5):645–651. https://doi.org/10.1016/0005-1098(84)90014-1
14. Rivera DE (2006) Pid control: New identification and design methods-[book review]. IEEE Control Syst Mag 26(1):95–97
15. Atherton DP (2000) Autotuning of PID controllers: relay feedback approach. Int J Adapt Control Signal Process 14(5):559–562. https://doi.org/10.1002/1099-1115(200008)14:5%3c559::aid-acs597%3e3.0.co;2-h
16. Hang CC, Astrom KJ, Ho WK (1991) Refinements of the Ziegler-Nichols tuning formula. IEE Proc D Control Theory Appl 138(2):111–118. https://doi.org/10.1049/ip-d.1991.0015

17. Åström KJ, Hägglund T, Hang CC, Ho WK (1993) Automatic tuning and adaptation for PID controllers—a survey. Control Eng Pract 1(4):699–714. https://doi.org/10.1016/0967-0661(93)91394-C
18. Lian FL, Moyne J, Tilbury D (2001) Analysis and modeling of networked control systems: MIMO case with multiple time delays. Proceedings of the American control conference, vol 6. IEEE, pp 4306–4312, https://doi.org/10.1109/ACC.2001.945654
19. Ho WK, Hang CC, Cao LS (1995) Tuning of PID controllers based on gain and phase margin specifications. Automatica 31(3):497–502. https://doi.org/10.1016/0005-1098(94)00130-B
20. Koivo HN, Tanttu JT (1991) Tuning of PID Conrollers: survey of SISO and MIMO techniques. IFAC Proc. 24(1):75–80. https://doi.org/10.1016/s1474-6670(17)51299-9
21. Voda AA, Landau ID (1995) A method for the auto-calibration of PID controllers. Automatica 31(1):41–53. https://doi.org/10.1016/0005-1098(94)00067-S
22. Chen X, Wang L (2016) Quadrotor cascade PID controller automatic tuning. 2016 Australian control conference, AuCC 2016. IEEE, pp 311–316, https://doi.org/10.1109/AUCC.2016.7868208
23. Lequin O, Gevers M, Mossberg M, Bosmans E, Triest L (2003) Iterative feedback tuning of PID parameters: comparison with classical tuning rules. Control Eng Pract 11(9):1023–1033. https://doi.org/10.1016/S0967-0661(02)00303-9
24. Zhuang M, Atherton DP (1993) Automatic tuning of optimum PID controllers. IEE Proc D Control Theory Appl 140(3):216–224. https://doi.org/10.1049/ip-d.1993.0030
25. Islam M, Okasha M, Idres MM (2017) Dynamics and control of quadcopter using linear model predictive control approach. IOP Conf Ser Mater Sci Eng 270(1):12007. https://doi.org/10.1088/1757-899X/270/1/012007
26. Thu KM, Gavrilov AI (2017) Designing and modeling of quadcopter control system using L1 adaptive control. Procedia Comput. Sci. 103:528–535. https://doi.org/10.1016/j.procs.2017.01.046
27. Okyere E, Bousbaine A, Poyi GT, Joseph AK, Andrade JM (2019) LQR controller design for quad-rotor helicopters. J Eng 2019(17):4003–4007. https://doi.org/10.1049/joe.2018.8126
28. Raafat S, Mahmoud Z (2018) Robust multiple model adaptive control for dynamic positioning of quadrotor helicopter system. Eng. Technol. J. 36(12A):1249–1259. https://doi.org/10.30684/etj.36.12a.6
29. Daniali HM, Azad NL (2020) Fast nonlinear model predictive control of quadrotors. Progress in Canadian mechanical engineering. Volume 3. University of Prince Edward Island. Robertson Library, https://doi.org/10.32393/csme.2020.1180
30. Abdelmaksoud SI, Mailah M, Abdallah AM (2020) Robust intelligent self-tuning active force control of a quadrotor with improved body jerk performance. IEEE Access 8:150037–150050. https://doi.org/10.1109/ACCESS.2020.3015101
31. Hua H, Fang Y, Zhang X, Lu B (2021) A novel robust observer-based nonlinear trajectory tracking control strategy for quadrotors. IEEE Trans Control Syst Technol 29(5):1952–1963. https://doi.org/10.1109/TCST.2020.3024805
32. Dix P, Ashrafi B, Drakunov S, Özguner U (1995) ABS control using optimum search via sliding modes. IEEE Trans Control Syst Technol 3(1):79–85. https://doi.org/10.1109/87.370698
33. Elor Y, Bruckstein AM (2012) Two-robot source seeking with point measurements. Theor Comput Sci 457:76–85. https://doi.org/10.1016/j.tcs.2012.06.036
34. Dürr HB, Stanković MS, Ebenbauer C, Johansson KH (2013) Lie bracket approximation of extremum seeking systems. Automatica 49(6):1538–1552. https://doi.org/10.1016/j.automatica.2013.02.016
35. Stankovic MS, Stipanovic DM (2009) Discrete time extremum seeking by autonomous vehicles in a stochastic environment. Proceedings of the 48h IEEE Conference on decision and control (CDC) held jointly with 2009 28th Chinese Control Conference. IEEE, https://doi.org/10.1109/cdc.2009.5400504
36. Gelbert G, Moeck JP, Paschereit CO, King R (2012) Advanced algorithms for gradient estimation in one- and two-parameter extremum seeking controllers. J Process Control 22(4):700–709. https://doi.org/10.1016/j.jprocont.2012.01.022

37. Wang HH, Krstić M, Bastin G (1999) Optimizing bioreactors by extremum seeking. Int J Adapt Control Signal Process 13(8):651–669. https://doi.org/10.1002/(SICI)1099-1115(199912)13:8%3c651::AID-ACS563%3e3.0.CO;2-8
38. Leyva R, Alonso C, Queinnec I, Cid-Pastor A, Lagrange D, Martínez-Salamero L (2006) MPPT of photovoltaic systems using extremum—Seeking control. IEEE Trans Aerosp Electron Syst 42(1):249–258. https://doi.org/10.1109/TAES.2006.1603420
39. Ghaffari A, Seshagiri S, Krstić M (2015) Multivariable maximum power point tracking for photovoltaic micro-converters using extremum seeking. Control Eng Pract 35:83–91. https://doi.org/10.1016/j.conengprac.2014.11.007
40. Tinglong P, Zhicheng J, Zhenhua J (2008) Maximum power point tracking of wind energy conversion systems based on sliding mode extremum seeking control. 2008 IEEE Energy 2030 conference, ENERGY 2008. IEEE, https://doi.org/10.1109/ENERGY.2008.4781032
41. Hassani H, Mansouri A, Ahaitouf A (2020) Robust autonomous flight for quadrotor UAV based on adaptive nonsingular fast terminal sliding mode control. Int J Dyn Control 9(2):619–635. https://doi.org/10.1007/s40435-020-00666-3
42. Hassani H, Mansouri A, Ahaitouf A (2019) Control system of a quadrotor UAV with an optimized Backstepping controller. Proceedings—2019 international conference on intelligent systems and advanced computing sciences, ISACS 2019. IEEE, https://doi.org/10.1109/ISACS48493.2019.9068874
43. Zhou L, Zhang J, She H, Jin H (2019) Quadrotor uav flight control via a novel saturation integral backstepping controller. Automatika 60(2):193–206. https://doi.org/10.1080/00051144.2019.1610838
44. Guo Y, Jiang B, Zhang Y (2018) A novel robust attitude control for quadrotor aircraft subject to actuator faults and wind gusts. IEEE/CAA J. Autom. Sin. 5(1):292–300. https://doi.org/10.1109/JAS.2017.7510679
45. Kotarski D, Benic Z, Krznar M (2016) Control design for unmanned aerial vehicles with four rotors. Interdiscip Descr Complex Syst 14(2):236–245. https://doi.org/10.7906/indecs.14.2.12
46. Fanni M, Khalifa A (2017) A new 6-DOF quadrotor manipulation system: Design, kinematics, dynamics, and control. IEEE/ASME Trans. Mechatronics 22(3):1315–1326. https://doi.org/10.1109/TMECH.2017.2681179
47. Magsino ER, Dollosa CM, Gavinio S, Hermoso G, Laco N, Roberto LA (2014) Stabilizing quadrotor altitude and attitude through speed and torque control of BLDC motors. 2014 13th international conference on control automation robotics and vision, ICARCV 2014. IEEE, pp 438–443, https://doi.org/10.1109/ICARCV.2014.7064345
48. Luukkonen T (2011) Modelling and control of quadcopter. Indep Res Proj Appl Math Espoo 22:22
49. Tan Y, Moase WH, Manzie C, Nešić D, Mareels IMY (2010) Extremum seeking from 1922 to 2010. In: Proceedings of the 29th Chinese control conference, pp 14–26
50. Krstić M, Wang HH (2000) Stability of extremum seeking feedback for general nonlinear dynamic systems. Automatica 36(4):595–601. https://doi.org/10.1016/S0005-1098(99)00183-1
51. Ariyur KB, Krstić M (2004) Real-time optimization by extremum-seeking control. Real-Time Optim by Extrem Control. https://doi.org/10.1002/0471669784
52. Choi J-Y, Krstic M, Ariyur KB, Lee JS (2002) Extremum seeking control for discrete-time systems. IEEE Trans Automat Contr 47(2):318–323. https://doi.org/10.1109/9.983370
53. Chin DC (1997) Comparative study of stochastic algorithms for system optimization based on gradient approximations. IEEE Trans Syst Man Cybern Part B Cybern 27(2):244–249, https://doi.org/10.1109/3477.558808
54. Popović D, Janković M, Magner S, Teel AR (2006) Extremum seeking methods for optimization of variable cam timing engine operation. IEEE Trans Control Syst Technol 14(3):398–407. https://doi.org/10.1109/TCST.2005.863660
55. Beaudoin JF, Cadot O, Aider JL, Wesfreid JE (2006) Bluff-body drag reduction by extremum-seeking control. J Fluids Struct 22(6–7):973–978. https://doi.org/10.1016/j.jfluidstructs.2006.04.010

56. Centioli C et al (2005) Extremum seeking applied to the plasma control system of the Frascati Tokamak Upgrade. Proceedings of the 44th IEEE conference on decision and control, and the European control conference, CDC-ECC '05, vol 2005. IEEE, pp 8227–8232, https://doi.org/10.1109/CDC.2005.1583494
57. Ou Y, Xu C, Schuster E, Luce TC, Ferron JR, Walker ML (2007) Extremum-seeking finite-time optimal control of plasma current profile at the DIII-D tokamak. Proceedings of the American control conference. IEEE, pp 4015–4020, https://doi.org/10.1109/ACC.2007.4283102
58. Guay M, Dochain D (2017) A proportional-integral extremum-seeking controller design technique. Automatica 77:61–67. https://doi.org/10.1016/j.automatica.2016.11.018
59. La Salle J, Lefschetz S (2012) Stability by Liapunov's direct method with applications by Joseph L Salle and Solomon Lefschetz. Elsevier
60. Killingsworth NJ, Krstić M (2006) PID tuning using extremum Seeki: online, model-free performance optimization. IEEE Control Syst 26(1):70–79. https://doi.org/10.1109/MCS.2006.1580155
61. Hammond PH, Duckenfield MJ (1963) Automatic optimization by continuous perturbation of parameters. Automatica 1(2–3):147–175. https://doi.org/10.1016/0005-1098(63)90020-7
62. Khalil HK (2002) Nonlinear systems, Third edit
63. Tan Y, Nešić D, Mareels I (2006) On non-local stability properties of extremum seeking control. Automatica 42(6):889–903. https://doi.org/10.1016/j.automatica.2006.01.014
64. Kennedy J, Eberhart R, Particle swarm optimization. Proceedings of ICNN'95—international conference on neural networks. IEEE, https://doi.org/10.1109/icnn.1995.488968
65. Wang D, Tan D, Liu L (2018) Particle swarm optimization algorithm: an overview. Soft Comput 22(2):387–408. https://doi.org/10.1007/s00500-016-2474-6
66. Muhsen A, Raafat S (2021) Optimized PID control of quadrotor system using extremum seeking algorithm. Eng. Technol. J. 39(6):996–1010. https://doi.org/10.30684/etj.v39i6.1850

Wireless Sensor Network for Robot Navigation

Ekhlas Kadum Hamza, Kawther Dawood Salman, and Shahad Nafea Jaafar

Abstract In smart factories and industrial automation, mobile robots were majorly used. Various mobile robot types collaborate in the workplace to execute various difficult tasks. In addition, Wireless Sensor Networks (WSNs) offer a virtual layer through which computational systems could access information regarding the physical world. Also, mobile robots have long played an important part in WSNs, and great research efforts were gone into figuring out how to use them to improve system performance over the last few decades. Depending on the tasks which mobile robots do in WSNs, they are categorized into 3 categories: Collection, in which mobile robots gather the data from sensor nodes; Delivery, in which mobile robots send things like energy to the sensor nodes; and Combination, in which the mobile robots gather and deliver at the same time. In this work, we'll look at a few major ideas for dealing with problems by integrating approaches from other fields. The study introduces a wireless network paradigm for data exchange between nodes of the mobile robot. It can be utilized for applications such as control and monitoring. Communication hardware and sensors are installed in each one of the nodes. Nodes are collecting data from the sensors and relay it to a central server. Which could be a host computer that was connected to a cloud network. It's called a WSN. The primary aim is to discover a solution for reducing robot node computation power and energy consumption.

Keywords Wireless sensor network · Mobile robots · Application mobile robot

E. K. Hamza (✉) · S. N. Jaafar
Control and System Engineering Department, University of Technology, Baghdad, Iraq
e-mail: Ekhlas.K.Hamza@uotechnology.edu.iq

S. N. Jaafar
e-mail: shahadnafaa_1994@yahoo.com

K. D. Salman
Computer Technology Engineering Department, Al-Hikma University College, Baghdad, Iraq
e-mail: cse.19.02@grad.uotechnology.edu.iq

A. T. Azar et al. (eds.), *Mobile Robot: Motion Control and Path Planning*,
Studies in Computational Intelligence 1090,
https://doi.org/10.1007/978-3-031-26564-8_18

1 Introduction

Smart technologies are critical for long-term economic growth. They turn offices, residences, cities, and even factories into self-contained, autonomous systems without the need for human involvement [1, 2]. When wireless sensors and actuators, the latest technological developments (large random-access memory chips, high-power processors, grid computing, and digital signal processing) are combined to open the door to a new generation of inexpensive actuators and sensors capable of high-grade temporal and spatial resolution. The world's economy is being boosted by the latest automation trends and the expanding use of modern technologies [3, 4].

WSNs, which often offer significant field coverage, are becoming common for applications like precision agriculture, area monitoring, and wildlife conservation. Also, WSNs could be quickly deployed to handle unforeseen circumstances like man-made or natural disasters. WSNs are an interesting addition to mobile robots. Mobile robotics and WSNs have been combined for various uses. By providing mobile robots with environmental data, a WSN could increase their sensory capabilities. The mobile robots could use such information for a variety of tasks, including navigation, mapping, localization, and mission planning. Additionally, mobile robots might improve WSN applications by setting up, swapping out, and collecting data from sensor nodes, offering sensor node locations, and offering gateway services to sensor networks.

In such modernization, the WSN and the Internet of Things (IoT) play critical roles [5, 6]. IoT is an area of engineering that focuses on providing thousands of miniature, physically connected objects that can work together for achieving a common purpose. Because of the widespread use of such small networked devices, the Internet of Things has gained a lot of traction. These are intelligent yet simple devices that could communicate and sense wirelessly [7, 8]. As depicted in (Fig. 1), a WSN may be defined as a set of sensor and routing nodes that can be used in the environment to anticipate physical factors like temperature, wind, and various others. Those networks take data from small nodes, process it, and after that send it to operators. Sensor networks have been utilized in a range of control systems, such as home automation, environmental monitoring, biological and chemical assault detections, smart grid deployments, surveillance, and much more, as shown in (Fig. 2). Data collecting, catastrophe avoidance, offshore exploration, pollution monitoring, and tactical surveillance are all areas [9] where WSN plays an important part in the oil industry and aquaculture [10, 11].

WSNs are frequently used in remote areas in which human involvement for post-deployment maintenance isn't possible. As a result, attempts are being conducted to improve their durability and effectiveness [12]. Various obstacles to WSN deployment exist, including long-distance deployment and power consumption. Those limitations are no longer barriers to large-scale remote deployments because of applications developed and automation trends. WSN uses a star topology to reduce the possibility of network failure through the connection of all of the systems to one central node. Ad-hoc networks, on the other hand, have a mesh topology with every

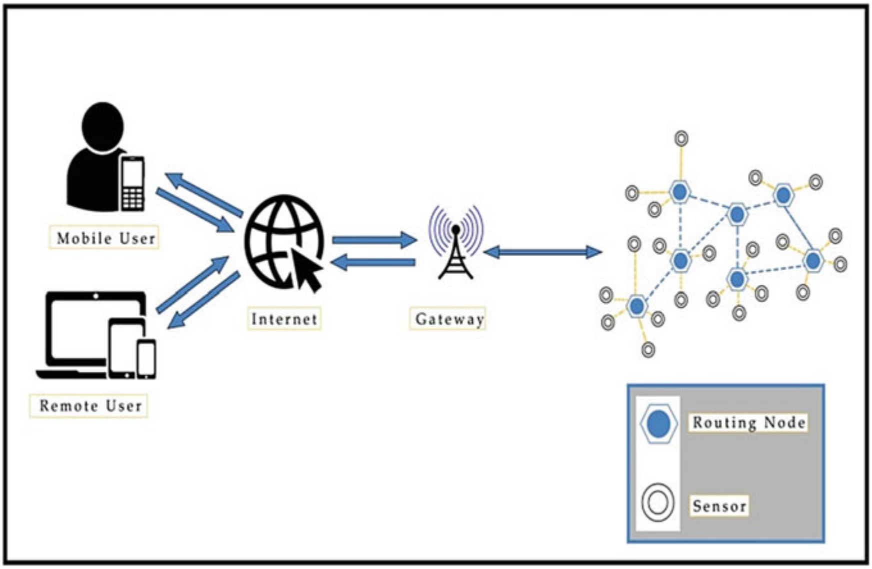


Fig. 1 Architecture of WSN

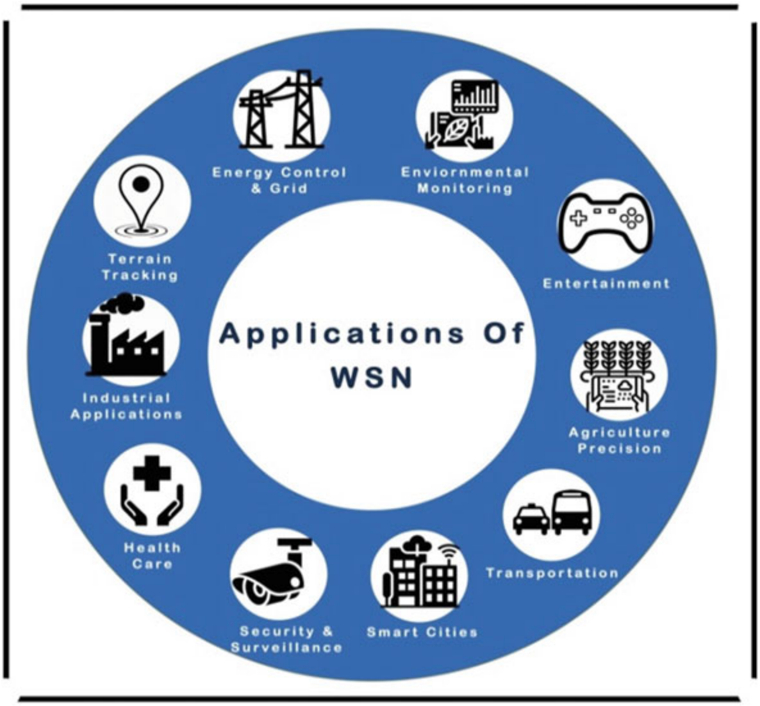


Fig. 2 Application of WSN

one of the nodes being human-driven [13–15]. WSN applications, at the same time, make daily life more comfortable and easier, like healthcare applications that increase longevity and health. In addition to WSN, the Internet of Things has played a significant role in human life. The digital age and IoT are critical in reducing physical and social obstacles and presenting individuals with mobility and ease, leading to enhanced and equitable opportunities and information access [16–18]. As demonstrated in (Fig. 2), IoT has various applications in climate, agribusiness, education, clinical care, finance, and transportation.

One of the Robot Sensor Network applications in health care is the remote monitoring of physiological data. Inside a hospital, doctors and patients are monitored and tracked, along with pharmaceutical administration and assistance to the elderly. The two domains that have led to the merging of Mobile (Tele-) Robotics and WSNs are robotics and wireless sensor networks (WSNs). Researchers of WSNs began exploring ways to make the network's energy use more effective while also doing certain additional tasks in the monitored field. WSN researchers have therefore examined including static actuators and/or mobile robots in the network [18]. This path arose from the requirement for more resource-rich nodes to fulfill a variety of functions while extending the WSN's lifetime. Robotics researchers, have shifted their focus from single-robot applications to multi-robot applications, and are looking for approaches for the efficient coordination of several robots while also increasing their knowledge of this area. As a result, researchers in the area of robotics had considered incorporating networking and multi-media sensing abilities into mobile robots, resulting in the creation of Network Robot Systems (NRIs). This line of inquiry investigated the placement of stationary sensors as well in the area so that mobile robots could interface with them and get extra-sensory data about their surroundings. Despite their initial differences, those two paths finally converged to create a system that integrates WSN and NRS in addition to human users for teleoperation activities [15]. As in Wireless Sensor and Actuator/Actuator Networks (WSAN), this model could be generalized to incorporate static actuators. The interactions among sensors, base stations, and mobile robots, on the other hand, must be kept to a minimum to reduce the level of interaction complexity. One of the major issues in WSRNT is coordinating decision-making among various components of such a complex system. The Decision-making process could be classified into 3 classes: teleoperation, hybrid and autonomous. Autonomous decision-making represents a distributed method that relies on the networks to determine which robots will be responding to which sensor requests. Since a human operator controls the robot's behavior, teleoperation decision-making is a more centralized technique. Instead of passively monitoring the environment, teleoperation allows operators to interact with it [10]. The hybrid technique aims to efficiently combine autonomous decision-making.

Interest in mobile robots has increased recently. Generally speaking, such robots are finding more and more uses. It is a difficult and occasionally risky task for humans. Examples of this include radiation following nuclear power plant failure, air monitoring, and widespread land pollution. Multiple measurements must be taken in various areas for getting a precise map of the level of pollution. This task is carried out over a long period, which implies that the findings may not reflect the true value in the event of extremely dynamic processes like the spread of contaminations or radioactive particles in the air. Utilizing networked mobile robots with sensors for data collecting is the solution to this issue. The flying or mobile robots could be arranged in a grid-like pattern. Large (and various) areas could be covered in this way. The cost of implementation is one of the key elements in developing this system. Low-power, small mobile robots that are controlled by a primary robot should be used to accomplish this. The most computationally intensive tasks are carried out on the cloud. This gives someone network access to robotic resources. Particularly presenting a model for robot communications within the platform. The study provides a WSN extension for establishing communications between the platform server and the robot over IEEE 802.15.4, rather than IEEE 802.11 (Wi-Fi). It's concentrated on networked sensor development and configuration regarding robot control and monitoring. A survey of several works reveals a range of communication protocols and networks. The majority of attempts are concentrated on issues with network delay, packet loss, and address channel limits associated with packet rates. It might be inferred that there is still no perfect solution.

1.1 The Robotic Map Problem

The challenge with robotic mapping is building a spatial model of the robot's surroundings. Robots frequently utilize maps for navigation [7].

Robots need sensors to be able to perceive the outside world to obtain a map. For this purpose, sensors including cameras, range finders using sonar, laser, and IR technologies, radar, touch sensors, GPS, and compasses are frequently used. All of such sensors, on the other hand, are prone to mistakes, which is sometimes referred to as measurement noise. More crucially, most robot sensors have strict range restrictions. For instance, light and sound can't pass through the walls. Due to such range restrictions, a robot must navigate its surroundings for creating a map. When constructing maps, the motion instructions (controls) used throughout environment exploration provide crucial information regarding the locations where various sensor data were gathered. Controls by themselves are not sufficient to establish a robot's pose (orientation and location from its environment's point of view) because robot mobility is also susceptible to errors.

The high dimensionality of the entities which are being mapped presents a challenging component of the robot mapping problem. Consider how many numbers it might take for describing an environment similar to home for the reader to grasp the problem's multidimensionality. A few dozen numbers can be sufficient if one limits oneself to the description of significant topological structures like doors, rooms, corridors, and intersections. As a standard depiction of robotic maps, a thorough 2D floor layout sometimes calls for thousands of integers. Yet, it might be simple to need millions of numbers to create a detailed D visual map of a building (or of the ocean floor). Each of these figures can be viewed from a statistical perspective as a dimension of the underlying estimate problem. As a result, the mapping problem may have a very high dimension.

The correspondence problem, sometimes referred to as the data association problem, is the third and maybe most challenging problem with robotic mapping. Also, the correspondence problem is the challenge of establishing whether sensor readings made at various times in the past correspond to the same real-world physical object. where a robot is attempting to map a large cyclic environment. The robot must determine its location on the map it had previously constructed before ending the cycle. This problem is made more difficult by the possibility that the robot's accumulated pose error will be unboundedly large at cycle closing. Due to the exponential growth of potential hypotheses over time, the correspondence problem is challenging. After a lengthy period during which the correspondence problem was mostly overlooked by the robot mapping community, the majority of scientific work on the problem has occurred in the last five years.

The environment changes with time. A few changes might occur gradually, such as how a tree's look varies with the seasons or the structural modifications that most office buildings experience over time. Others, like changing the door's state or finding furniture pieces, such as chairs, are quicker. The movement of other environmental agents like people or cars could occur even more quickly. The dynamic nature of robot settings poses a significant challenge since it gives another explanation for sensor results that appear to be inconsistent. To see, picture a robot standing in front of a door that was formerly open but is now closed. Two possibilities can account for such a finding: either the robot is not where it thinks it is, or the door status has changed. Unfortunately, very few mapping algorithms are capable of learning accurate maps of dynamic environments. The dominant paradigm is based on the premise that the world is static and that the robot is the only time-variant variable (and everything else that moves is just noise). As a result, the majority of approaches are only used within brief time frames when the surrounding settings are static. Robots having to choose their way while mapping presents a fifth and final challenge. Robotic exploration is the term used to describe the process of causing a robot to move to create a map. While completely described settings have optimal robot motion reasonably well established, exploring robots must deal with incomplete and partial models. Any effective exploration approach must therefore be able to account for potential surprises and eventualities that could happen throughout map acquisition. Because of this, exploration is a difficult planning problem that is frequently tackled using simple heuristics. The predicted gain in map information, the energy and time required to obtain such information, the potential loss of pose information along the way, and other factors must all be traded off when deciding where to proceed. Another requirement that excludes many present methods is that the underlying map estimating technology should be able to produce maps in real time.

1.2 Objective of Chapter

Robotics in WSNs can be utilized for solving problems as well as increasing performance, including data mulling, node distribution, sensor failure response, and data aggregation. Nodes attached to mobile robots are often utilized to help investigate the impacts of node mobility in sensor networks. A full-coverage network is made up of partial ones for reducing sensor overuse. Due to various constraints, a full-coverage network isn't ideal for real-world problems. The properties of probabilistic or partial coverage issues have been explored in this work, and they are put into comparison with problems of full coverage. In addition, this study will help in the overview of unresolved coverage issues. Relevant models depending on partial and probabilistic coverage problems, like network, detection, and deployment models, are also summarized. Through deploying, selecting or scheduling, and regulating sensors and other focus on the use of MEs for collecting and/or relaying data, this study examines 3 major aims: the way for the maximization of the network lifetime, the way for maximizing coverage quality, and the way for minimizing the number of sensors for the problems of the coverage with uncertain characteristics. Unlike them, this study compiles previous research under new taxonomy, which is: Collection, Delivery, and Combination, which encompasses a variety of functions carried out via MEs in WSNs.

The other parts of this study are provided as follows: Sect. 2 is presenting a previous study on Wireless Sensor Networks (WSNs). Section 3 provides a brief description of some types of wireless networks, summarizes the differences between them, and in the Sect. 4, explains the Li-Fi architecture. The components of Li-Fi are explained in Sect. 5, while Sect. 6 shows its operation. Li-Fi technology advantages and a review of Li-Fi features over Wi-Fi are provided in Sect. 7, respectively. In Sect. 8, problems with Li-Fi are explained. Finally, conclusions are drawn in Sect. 9 and the sources in Sect. 10.

2 Related Work

There is no precise and comprehensive definition of the Smart Grid but a large number of definitions that complemental 1 present suggested solutions regarding work that has been undertaken by many authors, as well as limitations and contributions of their work. The purpose was to identify research gaps in this field so that other academics may benefit. Researchers will be able to devise solutions and new approaches to address the research gaps, which will aid in the completion of the missing piece.

A mobile robot's motion is controlled by a method described using data obtained from WSNs and a sensor node that is attached to the robot that manages its navigation. With the use of WSN, this node connects with another node that is connected to a PC. The PC receives the robot's motion and position as well as information about other nodes in the surroundings. To enable the WSN to track the robot's location, it periodically sends beacons. A trilateration technique is used to determine the robot's position. This implies that a minimum of three WSN nodes should be aware of both their positions and the distances between them and the robot. This distance is calculated using RSSI (Radio Signal Strength Indicator). A new approach to the navigation problem is put out. It is suggested that a distributed planning framework be used to determine suitable courses for mobile robots while considering "obstacles" observed via WSN. Put differently, the wireless sensor network creates a roadmap to direct robots throughout the environment. Sensors need to have laser rangefinders that can map out their sensing region and detect obstacles. A map of the monitored region is created by combining the local maps. Roadmaps for the robot are then produced using this map. The environment needs to be equipped with landmarks, referred to as relay points, to merge the local maps produced via sensors. The local maps are later pieced together using such relay points. A WSN-based indoor navigation system for mobile autonomous robots is provided. In addition, this system is built upon a sensor network that has already been set up, with no less than three nodes in each room having known locations. Those nodes relay their positions to the robot through communication. The robot can after that use RSSI for determining its position. WSNs are used to address localization problems as well. Using an 806 Anais system determines the localization regarding a moving target traveling through a WSN-covered area. In this case, a mobile robot that is connected to the WSN through a sensor node becomes a mobile sensor node. All of the sensor network's static nodes are thought to have known positions. Trilateration is used to pinpoint the robot's position as it moves across the WSN depending on the static nodes' positions and RSSI measurements between mobile and static nodes. Using mobile robots in conjunction with WSN to gather or aggregate data from sensor nodes is another option. presents a system that utilizes mobile robots to gather data from sensor nodes. In the case when monitored locations are spread apart and information needs to be integrated, this solution becomes intriguing. In this case, it would be more interesting (for example, about energy savings) to utilize a mobile robot that could gather data from one area and after that go to the next area to gather the data rather than installing nodes to link the two locations. According to, a mobile node that moves within a WSN could also be utilized to localize the nodes regarding a deployed network. In this study, a mobile robot with a sensor node moves randomly inside a WSN. Periodically, the robot broadcasts beacons indicating its location, which are picked up by nodes in the area. The sensor nodes can determine the robot's distance with the use of such beacons, and they may then utilize that information for creating and maintaining their position estimates. An alternate approach for mapping a WSN is provided. In this instance, several sensor nodes with cameras are placed on the surroundings without any intercommunication between the nodes. A mobile robot that can navigate around the environment and communicate its position to the nodes is outfitted with a sensor node and a well-known landmark that the cameras can identify. The nodes can calculate the distance between them and the robot with the use of the landmark, and they can calculate their positions utilizing the robot's position data. Typically, solutions for the combination of WSN and mobile robotics are created to address particular issues in particular contexts. WSN-based navigation algorithms often depend on the fact that all network nodes' positions are known or may be inferred. To identify node or robot positions, localization systems frequently use RSSI readings, which are unreliable in dynamic situations.

As stated in Table 1, this section presents a comparison of past ideas and research linked to WSNs for Robots.

Table 1 Presents a comparison of past ideas and research linked to WSNs for Robots

Authors	Years	Research concepts
Sharma et al. [19]	2020	They developed relay node architectures and load balancing for WSN to increase network lifetime metrics. They created an IoT-based WSN to give coal miners premature warnings regarding natural disasters
Arslan et al. [20]	2020	They created a computer vision-based method for weed and non-image classification. Wireless sensor nodes use image processing for detecting weed images and collecting data on temperature, light, moisture, and humidity
Haque et al. [21]	2020	Discussed the assess the vibration or earthquake measurement, they devised a computer-based monitoring system. WSNs are utilized for sensing structural damage and locating its exact location. Also, they suggested a network architecture analysis system based on topology. Their system has been an energy-saving one that turned off nodes when there was no traffic identified
Majid et al. [1]	2022	The Authorsfocuses on research solutions and new techniques to automate Industry 4.0, they analyzed over 130 articles from 2014 until 2021 and covers several aspects of Industry 4.0, from the designing phase to security needs, from the deployment stage to the classification of the network, the difficulties, challenges, and future directions
Kim et al. [5]	2022	The Authors proposes a novel method for multiple-robot localization in wireless sensor networks. The proposed method is theoretically based on the finite memory estimation and utilizes relative distance and angle measurements between robots. Thus, the proposed method is referred to as distributed finite memory estimation from relative measurements (DFMERM). Due to the finite memory structure, the DFMERM has inherent robustness against computational and modeling errors. Moreover, the novel distributed localization method using relative measurements shows the robustness against missing measurements. Robust DFMERM localization performance is experimentally demonstrated using multiple mobile robots under the harsh conditions
Majid et al. [22]	2022	The authors investigated Industry 4.0 automation options and novel approaches, covering numerous areas of Industry 4.0, from the planning process to security concerns, from the deployment stage to network classification, as well as challenges, difficulties, and prospects, based on an analysis of 130 publications published between 2014 and 2021

(continued)

Table 1 (continued)

Authors	Years	Research concepts
Majid et al. [23]	2022	Explained a WSNs, the authors suggested a unique approach for multiple-robot localization. This approach uses relative distance and angle readings between the robots and is conceptually based on a nite memory estimate. As a result, distributed nite memory estimation from relative measurements (DFMERM) was coined. DFMERM offers inherent robustness towards computational and modeling errors because of its nite memory structure. Furthermore, the innovative distributed localization approach based on relative measures demonstrates robustness in the face of missing measurements. Multiple mobile robots are used to demonstrate robust DFMERM localization ability under extreme situations

3 Applications and History of Wireless Sensor Networks

WSNs are made up of numerous sensor nodes that may connect over wireless channels to send data to the base station (BS) [27]. Wildlife [28], volcano [29, 30], environmental [31], structure monitoring [32], and water [33, 34], along with health care and fire detection [35, 36], have all been proposed as applications for such networks. Those networks can collect data about the surroundings (for example, temperature or light readings) and send it to BS. The advantage is that such sensors could be dispersed in areas where human monitoring can be excessively dangerous or expensive. Sensor Networks have been first developed in the 1980s, yet comparable systems date back to Cold War and as far back as 1949 with SOSUS (i.e. Sound Surveillance System) [37]. In addition, SOSUS can be defined as one of the ocean-floor acoustic sensor systems that were originally designed to track Soviet submarines. SOSUS began development in 1949 and was first used for tracking an American submarine in 1961. In June of the following year, it made its 1st successful Soviet submarine detection [16]. With applications like SOSUS and networks of air defense radars, defense applications represent one of the primary drivers for sensor network development and research. The components of the technology regarding a Distributed Sensor Network (DSN) have been detected at a DNS workshop in 1978 [19], which launched modern sensor network research. With their DSNs initiative in the late 1970s and early 1980s, Defense Advanced Research Projects Agency (DARPA) was another driving force behind this early research [20]. DARPA launched the Sensor Information Technology (SensIT) program to enhance the usage of WSNs more recently [24]. WSN applications that have been developed and deployed in practice come from a wide range of sectors and are as varied as they are numerous. The field of environmental monitoring was one of the most common and attractive pathways for scholars. Everything from ice and glacier monitoring [38] to fire detection [35] and volcano monitoring [31] has been covered. A WSN with mobile robots could be used for any monitoring application, like flood detection or pollution detection and warning systems. In reality, one such system, which is known as the ALERT system, is currently in use in the United States [39]. Various security and military applications, like border monitoring apps or combat applications of monitoring, can also be explored [30]. The majority of research had concentrated on the fixed WSNs.

Where sensors are not allowed to move after being deployed in the field. Researchers recently began to include mobile sensor nodes in distributed sensing systems including habitat monitoring, surveillance, urban search and rescue, and wildfire detection. A mobile sensor node (such as a mobile robot) can change its position on its own depending on its mission. Robomote, for example, is a small, mobile robot created to investigate challenges in large-scale sensor networks and distributed robots. Some other examples of low-cost robots that have been utilized in the robotics and sensor networks are CotsBots MICAbot [37] represents a small, low-cost robot that explores large-scale networks. MICAbot is designed in the same way as the Robomote, but to improve the modularity, functionality, and structural stability of MICAbot without increasing costs. In WSNs, portability has been demonstrated to improve network connectivity, energy efficiency, and reliability. The ability to collect data more efficiently is the most common feature of mobile robots. By complementing data collection with mobile bots, energy use can be reduced and evenly distributed across the network. Multihop forwarding can be reduced to a few hops or even one hop with mobile bots, and that means that sensors only have to transmit their data instead of the data of other sensors. Besides saving energy, lowering the forwarding of multiple stores in the network helps avoid switching effects, as the nodes that are nearest to the base station die sooner, and as a result, the network is disconnected sooner than might be anticipated. Mobile bots also have other advantages, such as enhancing connectivity in sparse WSNs. Allowing the offline network to be functioning as a normal fully connected network. Security measures within WSN can also be improved by botnets using methods such as key distribution [30]. Mobile bots will improve and strengthen monitoring at the application layer. All of these benefits prompted researchers to focus their efforts on mobile botnets and take advantage of these benefits to enhance entire wireless networks. WSNs may be utilized to help solve various difficulties in robotics, and bots may be utilized for helping to solve problems in WSN wireless networks [22]. These issues include everything from WSN deployment to bot path planning. Data aggregation, data mules, and even localization can all be done by bots. Coordination between bots, as well as sensing by bots, can be improved by WSNs. Many new benefits and difficulties emerged as a result of this navigation, including cruise control and lane planning. Researchers have explored the use of several coordinated robots, yet a great deal of early research uses only a single, controlled robotic robot [35].

4 Mobile Wireless Sensor Networks

In this section, a brief MWSN taxonomy is provided, which includes differences between the WSNs and the MWSNs, and the benefits of the addition of mobility.

A. MWSN Architectures

Mobile sensor networks may be classified as flat, 2-tier, or 3-tier hierarchical architectures [5], as has been depicted in Fig. 3, and described below.

A set of heterogeneous devices which interact in an ad hoc manner make up flat, or planar, network architecture. The gadgets can all communicate over the same network, whether they are stationary or mobile. Figure 1a depicts the flat architecture of basic navigation systems [26]. A set of mobile nodes and a set of stationary nodes make up the two-tier architecture. The mobile nodes aid the network by acting as overlay networks or as data mules. Mobile devices with more powerful processors, better bandwidth, and wider communication ranges can be a part of the overlay network. Additionally, the density of the overlay network can be such that all of the nodes remain connected at all times or that the network disjoint. In the latter scenario, mobile entities can move into position for re-establishing the connectivity, making sure that network packets get to where they're supposed to. This is how the voting system [17] operates. Figure 1b shows the 2-tier architecture in detail. In a three-tier design, data is passed from stationary sensor nodes to mobile devices, which after that send it to access points. This heterogeneous network has been intended for covering a large area and working with multiple applications at once. Consider a sensor network application that checks the availability of parking spaces in the parking garage for instance. Sensor network's (1st tier) availability updates are broadcast to passing mobile devices (2nd tier), like cellphones or PDAs, which are compatible. Cell phones then transmit these available data to third-tier access points (cell towers), where they are uploaded into one centralized database server. The database is after that available to anyone who wants to find a parking space. Figure 1c shows the

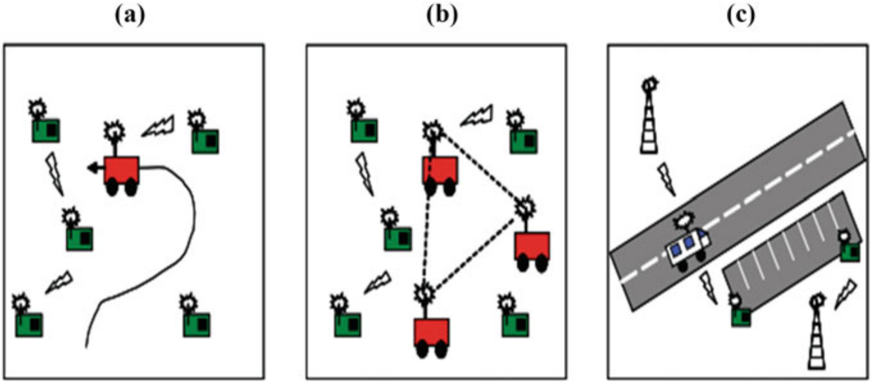


Fig. 3 **a** Flat, **b** 2-Tier, and **c** 3-Tier MWSN architectures

three-tier architecture. Mobile wireless sensors could be categorized at the node level according to their function inside the network: Mobile Embedded Sensor. The movement of mobile embedded nodes isn't under their control; instead, it is controlled by an external force, like in the case where they are tethered to an animal or shipping container [20, 21], and are examples of common embedded sensors. Mobile Actuated Sensor. To move across a sensing zone, sensor nodes might additionally have the capacity to move. This kind of controlled mobility allows for more precise deployment specifications, increased coverage, and the targeting and monitoring of particular occurrences. Data Mule The sensors might or might not need to be mobile, yet they sometimes do to collect their data and send it to a base station (BS). Data mule is the term used to describe such mobile entities [16]. It is commonly believed that data mules can automatically recharge their power supply. Access Point. Mobile nodes could position themselves for maintaining network connectivity in sparse networks or in the case where a node disconnects from the network [27]. They act as network access points in this instance.

4.1 Differences Between WSNs and MWSNs

It's crucial to comprehend how the typical presumptions about statically deployed WSNs change in the case where the mobile entities are added before concentrating on the mobility feature of WSNs. Localization. Node position could be established once throughout initialization in networks that have been statically deployed. Yet, when they move about the sensing region, mobile nodes should constantly obtain their position. This necessitates more time and effort in addition to the availability of a quick localization service. Dynamic Network Topology. Routing tables or recent route histories are frequently used in conventional WSN routing protocols [23] that define the way to send messages via the network so they would most possibly reach their destinations. Table data rapidly become obsolete in dynamic topologies, and it is costly in time, power, and bandwidth to continuously perform route discovery. Fortunately, routing in mobile ad hoc networks (MANET) is a topic of ongoing research, and MWSNs could learn from this work [32]. Power Consumption. The models of power consumption [20] for MWSNs and WSNs are very different. Wireless communication has a high energy cost for both types of networks and needs to be handled wisely. Mobile entities, on the other hand, need more power for mobility, hence they frequently have much greater energy reserves or can recharge themselves by plugging into the power grid. Network sink. Sensor data is sent to a base station in centralized WSN applications so that it can be handled in a resource-intensive manner. The cost of data aggregation and routing might be high. A few MWSNs have mobile BSs [19] that move around the sensing area to gather data or place themselves in such a way that the sensor nodes experience the fewest amount of transmission hops.

5 Localizations in the MWSNs

In this section, give a taxonomy of the MWSN localization techniques and survey a few works that are typical of MWSN localization. And a lot of such methods could be used with MWSNs. The algorithms, hardware, and signal modalities used by the localization techniques might be characterized along several dimensions in which MWSN localization is normally carried out. To start localization, several nodes coordinate. After that, at least one node emit a signal, and at least one receiver observe a few characteristics of the signal (such as arrival time, signal strength, phase, etc.). A localization algorithm is after that used to convert signal measurements into estimates of the position to determine the node's location. It is frequently required for enlisting the aid of cooperating sensor nodes which were placed into the environment at predetermined locations for determining position. Infrastructure, anchor, and seed nodes are the names given to such devices. The satellites which orbit the earth, for instance, represent the infrastructure nodes for GPS. In the case when the positions of anchor nodes have been known in terms of some global coordinate system, absolute coordinates can be determined instead of the position estimate being relative to a group of the stationary anchor nodes at known positions in the local coordinate system (i.e., through using GPS) [22].

5.1 The Impacts of the Environment on Localization

The environment significantly affects how well a localization technique works. There is therefore no single localization technique that will be accurate. For all circumstances, a survey on the localization for the mobile WSNs is available at 247. Several environmental elements are listed below, along with their impact on the aforementioned localization techniques. Since such factors have an immediate effect on the crystal oscillator in the transceiver, pressure, ambient temperature, and humidity can have an impact on the accuracy of the localization. It is well known that precipitation, especially air moisture, affects radio wave propagation, thus localization methods that rely on RF measurements may suffer in such circumstances [33]. The fact that GPS is unreliable indoors, in water, and even in overcast conditions is one of its main issues. This is because the line of sight to up to 4 satellites orbiting the earth is necessary for the GPS receiver [14]. A significant effort is currently being made to provide precise localization techniques for indoor environments. The majority of propagation and measurement approaches suffer from multi-path effects [24], in which obstacles (such as furniture, people, walls, etc.) create signal reflections interfering with one another, making it difficult for indoor applications requiring the estimate of the node position. Additionally, many of the localization methods currently in use, which offer decent accuracy outside, are ineffective indoors.

6 MWSN Applications with Localization

Even though MWSNs remain in their infancy, several different applications were already created in which localization is crucial. The applications can be divided into four primary areas: environmental, commercial, military, and civil. Most, though, cross more than one of such categories.

6.1 Commercial

Anticipate a burst in commercial applications that need a certain form of position data as MWSNs gain in popularity.

- Service Industry. One such area is the service industry. For service robots that carry out activities like security and maintenance in office buildings, basic patient care in nursing homes, and food and concierge service in hotels and restaurants, businesses like Skilligent [35] are developing software protocols. A technique for position estimate is needed for each of such applications. A visual localization approach depending on pattern matching is utilized by Skilligent. When landmarks and objects are put into the system either a priori or dynamically as needed during operation. By comparing video images with landmark information, the robot determines its location.
- Housekeeping. An automated vacuum cleaning robot for homes is iRobot Roomba [26]. The Roomba uses feedback from various bumper and optical sensors to map the room as it goes. The entire room may be covered thanks to the run-time position information provided by wheel encoders. The Roomba utilizes a self-docking station as well for the automatic recharge of its batteries. 248I. Amundson and X. Koutsoukos

6.2 Environmental

The MWSNs became one the valuable assets for environmental monitoring. This is partial because of their capability in being utilized in remote areas and for their capability of gathering the data of broad areas of interest.

- Wildlife Tracking. An early MWSN called ZebraNet [18] involved fitting zebras with small wireless sensors to follow their movement. Data was sent to mobile base stations via the peer-to-peer network as there was no cellphone service because of the area's remoteness. The zebras have not been constrained to specific areas, and other than small devices that are attached to their bodies are left undisturbed. Without the usage of MWSNs, this kind of tracking wouldn't be possible.
- Pollution Monitoring. A mobile system of air quality monitoring has been presented in [37]. Sensor nodes measuring specific pollutants in the air are fixed

on vehicles, as they move along roadways, the sensors perform the sampling of the air, in addition to recording the concentrations of different pollutants along with the time and location. In the case where the sensors are in proximity of the access points, then, data are uploaded to a server and after that, they are published on the web.

7 Robot Navigate

The different types of environments where a robot can navigate are listed below.

7.1 Structured Environments

Quite a few very firm assumptions might be made regarding environmental characteristics. A classic example is environments that could be reliably divided into purely navigable open spaces and obstacles. The majority of environments fall under this category, as would an engineered outdoor setting with only flat ground and positive obstacles. Also, the environment will be altered to support any assumptions made, which implies a high level of environmental engineering.

7.2 Partially Structured Environments

Pre-mapped terrain with potentially significant elevation changes, despite being locally smooth, could be assumed to have a few characteristics. Urban environments are the typical example; it is considered that there are specific appropriate places to handle problems that are simple to recognize, yet could be occupied by dynamic obstacles.

The system has 3 key capabilities, which are:

(i) Highly accurate tracking of the stated paths,
(ii) Reliable detection of small obstacles,
(iii) planning coverage patterns for the complete coverage of a specified area. It’s usually the case that an environment has been originally exaggeratedly created. on the other hand, continuous reformation for the enforcement of strong environmental assumptions doesn’t happen.

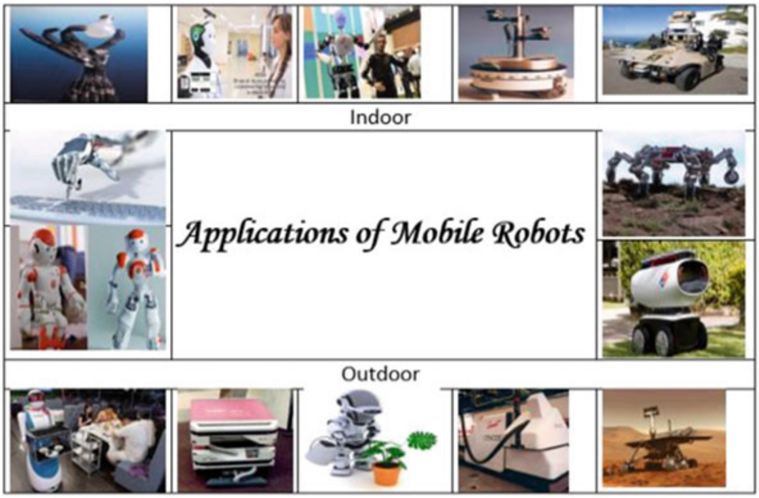


Fig. 4 Different mobile robot applications

7.3 *Unstructured Environments*

There are no established presumptions regarding the environment that can be followed for navigation. There are often two types of scenarios in such situations: i. Like terrestrial vehicles, the robot roams the area at random [24]. The robot completes a task with a goal position. In this scenario, both a localization method and a map of the robot's movement patterns are required. The "Robotic All-Terrain Lunar Explorer Rover" [15] is an impressive illustration of a mapping and positioning system. ii. "Mars Pathfinder" is made up of two parts: a drifter and a lander. The former is a mobile component that explores the environment, whereas the latter is a static component that takes pictures of Mars' surface using a stereo camera. Human operators at the earth control station choose goal points in 3D representations of previously taken photos of the landscape to establish the drifter mission path. Dead reckoning methods are used for determining the position; in any instance, no error may be presumed to have occurred (Fig. 4).

8 Mobile Robot Navigation in Indoor Environments

In indoor environments, mobile robots can have disturbances caused by revealing conditions and stationary obstacles. Under these conditions, the lower-level functions of robust and real-time landmark processing and trajectory generation must be established so that the system can be easily applied to an autonomous mobile robot. Mobile robots have no external guiding systems [33]. The indoor environment at the center mostly consists of a few static obstacles as well as fast-fluctuating dynamic areas with many movable obstacles. It provides a typical indoor environment. Therefore, the system developed here can be applied to similar environments without much alteration. The goal of the robot may be reaching a prescribed destination or following as closely as possible finding and mapping an area for later use [37]. In some indoor environments, it is difficult to detect the movable area because of the complexity of environmental structures. However, when there is a plan to use mobile robots in artificially organized environments, the environmental conditions may be modified according to their complexity [28]. For mobile robots working in indoor environments, a simple landmark model that is indicated by a circle, square, or rectangle, and a practical localization algorithm that uses a single landmark in a single image, which runs in real-time scenarios [39] are usually employed. Navigation in the indoor environment can be achieved by goal-oriented techniques. A map of the environment is generally a-priori known and a route to a goal position can be pre-planned. Then, the navigation task of the mobile robot, during the route consists of, continuously, determining its location in the environment to verify the vehicle is on the planned path [22]. Generally, mobile robot navigation issues are separated from designing artificial landmarks.

9 Mobile Robot Navigation in the Outdoor Environments

For mobile robots used in various civic and military applications, like exploration, personal assistance, material delivery, and human–robot interaction, autonomous outdoor navigation capability is required [19]. Robot mobility can be enhanced and their application in various daily activities could increase with the help of such navigation systems which could perceive the environment with compatible sensors. Autonomous mobile robots are intelligent machines created to accomplish certain goals by making decisions, effectively allocating resources, and upholding their integrity [33]. In addition, mobile robots are frequently used to complete tasks requiring navigational knowledge, and they are anticipated to move precisely and smoothly by making use of their sensory capabilities. Self-localization algorithm for outdoor navigation has been suggested as a solution. Constructing the environment with recognizable objects in recognized locations is one approach. Another method of fixing the robot is to dead-reckon its position after integrating its trajectory information. The final method is to compare active sensor readings to an established model of the environment or outside world [33]. It entails a fairly vast subject that was researched for a while. Navigation could be broken down into two issues in the context of mobile robotics: path planning and static and dynamic obstacle avoidance. Making the best choice for achieving a specific objective is the process of path planning. In the literature on mobile robots, several path-planning approaches were put forth. These path-planning algorithms could be evaluated using a variety of metrics. The majority of path-planning methods are created to operate in static environments with complete access to environmental information. For mobile autonomous robots, such as those used in manufacturing, mining, space missions, underwater research, and service robots, obstacle avoidance could be discussed. Mobile robot agents must be able to maneuver and engage with various settings. Autonomous robots can plot their paths and avoid obstacles [24]. The robot can avoid unexpected and dynamic obstacles in this way, including people strolling by it. The mobile robotics community has written extensively on this subject.

10 Robot Navigations in Complex Environments

The deployment of autonomous mobile robots in situations with many barriers presents several difficult navigational problems. In the case where the configuration of obstacles is unknown a priori, the control task becomes more challenging [25]. The most well-known techniques of controlling such systems have been based upon reactive local navigation approaches tightly coupling actions of the robot with sensory data. The environmental transformation has led to the projection of fuzzy behavior systems. With the help of a multi-valve judgment network and a predilection-based fuzzy behavior system, mobile robot vehicles can plan their routes to avoid obstacles effectively, allowing them to go through chaotic environments with relative ease. Preference logic is used to solve the hardest problem in the application of fuzzy-reactive-behavior-based navigation control systems, which is that of fusing or arbitrating personal behavior reactions [38]. In goal-directed robotic vehicle applications, the safe directing of autonomous ground vehicles (AGVs) in unstructured complex settings that are densely crowded with obstacles remains a significant difficulty. As a result of the lack of secrecy and destruction present in open environments, navigation through a forest is a form of challenge that particularly interests the military [37]. The target of this navigation challenge is to make sure that the robot not just gets to its destination without running into any obstacles, yet also does so at safe speeds which guarantee stability. Due to the possibility that a few of the navigational objectives may conflict with one another, the issue is extremely challenging. The navigation problem in a cluttered environment can be broken down into two categories: (1) Speed control; (2) Heading control. Two factors are used for speed control: (1) Overturning avoidance 2. Avoid the obstacles. The heading control carries out the following four traits: Avoid the obstacle on the front side, the obstacle on the right side, and the obstacle on the left side. (4) Target seeking. The goal-seeking behavior employs compass measurements to establish the direction of the goal, the overturning avoidance behavior uses a speedometer reading to estimate the robot's speed, and the obstacle avoidance behaviors use range-finding sensors to determine distances to the nearest obstacles [37]. The primary problem with autonomous navigation for mobile robots is local obstacle avoidance. The majority of navigation approaches used in partially known environments combine local and global navigation techniques to identify a feasible, free path to the desired destination while avoiding obstacles [28]. Local navigation strategies are crucial for a mobile robot to accomplish its objectives whenever it has to cope with a partially or unknown environment [30].

11 Research Methodology

This section presents the suggested taxonomy, along with the characteristics of sensor nodes and MEs in various applications. They explore shared goals of current methods for the use of MEs in WSNs, as well as the numerous constraints they take into account. Those discussions will help in identifying the key similarities and distinctions among the various methods in the sections that follow.

a. Planning Review

The review has been organized around research questions as well as their aims. IoT and WSN analyses have been carried out to demonstrate their significance in everyday life. In this subject, the usage of appropriate and smart devices was examined, yet further research is needed. As a result, it's critical to publish a systematic review article in this field, particularly for WSN and IoT, to demonstrate future directions. As a result, for the studies obtained from various repositories, created the search approach, search strings, quality assessment criteria, and inclusion/exclusion criteria.

b. Localization and Mapping

The mapping technique is closely linked to the localization stage related to mobile robots since it's critical that an agent could estimate the relative pose while mapping the environment and keeping track of everything that's around it. Such intuition generates a new problem, namely, mobile robots collecting data and charging batteries at the same time (Fig. 5) [33]. For many years, WSN localization was explored, and new abilities of mobility were discovered to make this localization simpler. As previously mentioned, SLAM 8 was widely utilized for mapping and localizing an area as well as the robots. Utilizing received signal strength indicators (RSSI) is another technique to work on localization. They utilize RSSI and SLAM by using robots that use an extended Kalman filter that has been customized to use RSSI distance estimates. Another alternative localization technique considers a robot's movement instead of its precise position [35]. If you know a robot's time traveled, speed, and angular rotation, you can determine the robot's new location at any point in time. the sensor as well as robot localization. They localize the sensors only through interactions of the sensor to sensor through establishing a network with self-localizing reference nodes interspersed among numerous normal nodes. Self-localizing nodes can after that triangulate the other nodes to assist in their localization. The authors discuss robot localization in the same work. The robot sends a request for the localization service to the sensors, localized sensors transmit localization information back to the robot, which utilizes this information to determine its position while maintaining the travel direction and speed in mind. however, they concentrate on assessing error that is imposed by distance from Distributed Intelligent Network Devices (DIND). Authors discover the directional error of a maximum of 3.5° as well as the positional error of a maximum of 17 cm, yet just in the case where a robot is seen for over 75% of the time. The GPS may be utilized for the accurate location of specific robots for easy localization [22]. The issue concerning this method is that the GPS consumes

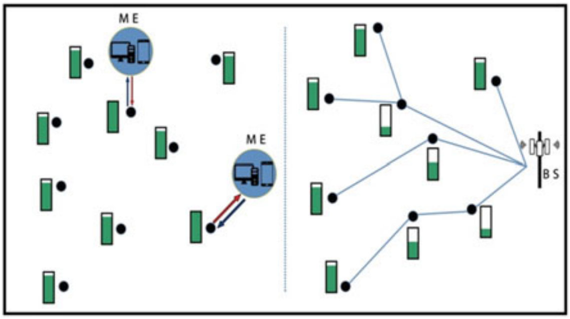


Fig. 5 Utilizing me for the simultaneous collection of the data and charge the sensor nodes against conventional routing in the WSN

a lot of energy, which could shorten a robot's lifetime. Another option in the literature is using anchor nodes, however, this needs particular robots or sensors to know their position ahead of time. They employ trigonometry to localize some unlocalized sensor about a sensor that is already localized in the first method, which they call semi-localization. The second method is rigidlocalization, which involves locating an unlocalized sensor using two distinct localized sensors to ensure that the graph is rigid globally. Those authors were capable of demonstrating through simulation that in the case where inter-radio distance is sufficient, this technique is capable of accurately localizing nodes. This is only a summary of some of the work on the topic of localization [33].

Estimating the mobile robot's orientation and position in its environment is referred to as localization [27]. As a necessary step before path planning, position estimate is a crucial task for mobile robots [33]. Numerous technologies, divided into absolute localization and relative localization, are employed in mobile robot localization. The robot's pose is estimated via relative (local) localization, which makes use of internal sensors like inertial sensors, an encoder, and a digital compass [36]. By utilizing external sensors, absolute (global) localization enables a robot to locate itself from within its domain. Passive or active landmarks, navigation, satellite signals, and map matching are all utilized in the techniques.

At high-level what every one of the interactions means and in the sections below, will be referring back to those interactions while discussing communications:

I. Sensor-Sensor

Those interactions have been found similar to the WSNs. A lot of problems concerning WSNs are still being researched and will continue to be problems with this system type. Routing, localization, QoS, and connectivity are examples of these. WSNs need efficient energy management to ensure the network's overall longevity. The limited energy source exacerbates these problems, necessitating creative solutions, as stated in previous sections.

II. Robot-Robot

The dynamic nature of robot-to-robot interactions is one of the most significant difficulties. At one point, a robot may be connected to some other robot, but not the next. All of the challenges of WSNs are also found in MRNs, albeit with the increased mobility of the robots, solutions could be discovered in a variety of ways. MANETs are a similar network type that has been investigated for a long time.

III. **Sensor-Robot**

Sensor-to-robot coordination and communication were proven to extend network lifetime since robots have more abilities compared to sensors and could assume more responsibility from them, allowing sensors to save energy by not being required to forward as many packets as in strictly WSNs. Robots' additional abilities in a network make tasks like connectivity and localization simpler than those in solely WSNs.

IV. **Base Station-WSRNT**

Because the differences are just nominal, the BS to the WSN and BS to MRN interactions are not included. A channel of communication between robots and human operators in the environment is required for teleoperation. A dedicated direct link from human operators to every one of the robots could be used in a small-scale system. Generally, inexpensive solutions are needed that allow a human operator to interface with mobile robots via an MRN. The majority of teleoperation problems in WSRNT will be comparable with issues in Internet-based teleoperation as a result of the use of MRNs combined with other variable delay communication routes to connect the operator to the MRN. There are two reasons for this. First, the link between the BS and WSRNT in the field might be made via the internet. Second, because of the noise problem in wireless communications, the connections inside WSRNT will be wireless and hence subject to variable delays, similar to internet-based communications.

V. **Sensors-Based Localization**

Gyroscopes and other inertial sensors are frequently employed in localization. However, inertial sensors, which are frequently made of inexpensive MicroElectroMechanical Systems (MEMS), include errors, which reduce the localization accuracy [32]. By preventing breakdowns brought by excessive noise, this method improves the system's performance and accuracy. Both beacon-based and signature-based approaches for localization in sensor-based technology fall into these two categories. To estimate location by observing node neighbors, signature-based algorithms [34] presuppose non-uniform distribution of nodes that are to be utilized as a signature. Beacon nodes must use GPS or manual configuration to determine their absolute positions in beacon-based approaches [35]. Utilizing distance and angle measurements from beacon nodes, succeeded by triangulation or multilateration, the other nodes' locations are estimated. A key component of mobile systems, including entertainment robots, autonomous navigation, military robots, and service robots, is sensor-based localization [37]. By using wheel speed sensors, for example, dead reckoning-based odometry could be used in such contexts to locate mobile nodes. Also of significant interest are emergency sensor networks (ESN) in situations like wildfires, natural disasters (storms, hurricanes), and terrorist strikes [38]. In emergency applications, the placement of sensor nodes is very localized, with the size of the node group depending on the intensity of the observed event at that location.

The Collection, Delivery, and Combination, specified in the following way, are the 3 major tasks of MEs in WSNs that were extensively investigated:

- Collection: MEs obtain something from the sensor nodes.
- Delivery: MEs deliver something to the sensor nodes.
- Combination: MEs obtain something from the sensor nodes, then deliver it to them. The most prevalent MEs task in WSNs is the collection, which mostly indicates the sensory data collection as seen in (Fig. 6). In the monitoring of the habitat [24], for instance, a group of the nodes is fitted with at least one sensor, a receiver, and a transmitter. The sensory data from the nodes is collected by a ME. Various applications are also compatible with delivery.

In sensor networks, localization can be defined as the process of defining the sensor's location information in search space. Because of its impact on minimizing power consumption in the case when indicating the sensor node location, localization is a significant activity in sensor networks [24]. Because of its work relativity, localization enhances both routing and clustering processes. Localization is used by various monitoring systems to detect sensor locations within search space. The

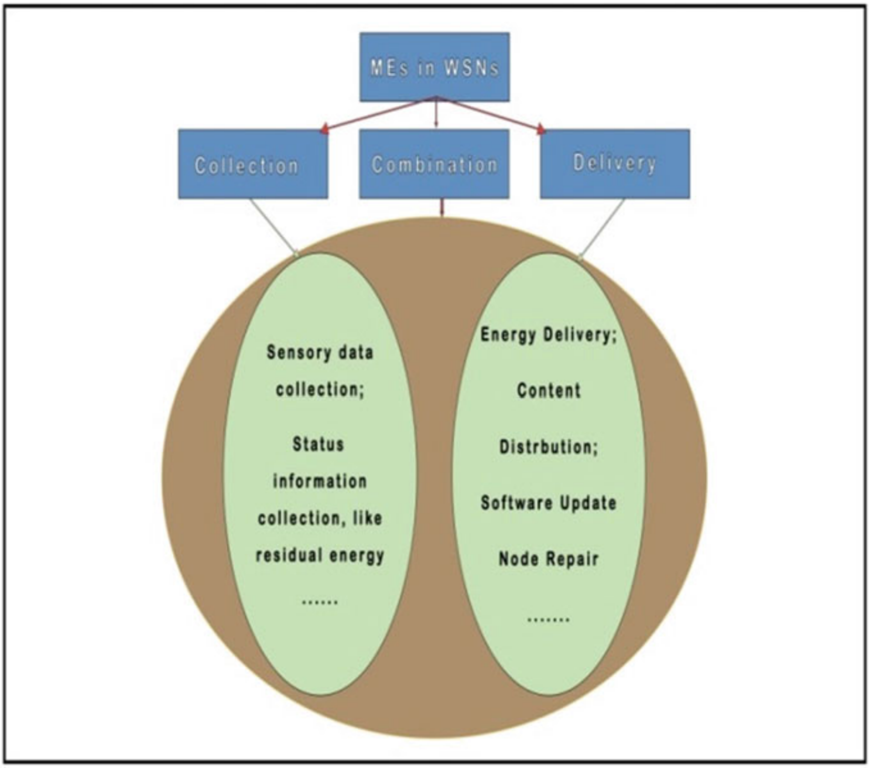


Fig. 6 Applications of MEs in WSNs

requirement for finding a sensor node in a system of sensor networks arose from the fact that a sensor node must detect its location and prepare for any further activities, particularly in the case when the routing algorithm or clustering algorithm requests location information. For determining the sensor locations, the localization procedure primarily employs an optimization-based method.

Simultaneous Localization and Mapping (SLAM) represents the common name for the problem. The Extended Kalman Filter, FastSLAM [20], Smoothing, and GraphSLAM are just a few of the solutions that were investigated. For this specific scenario, an offline SLAM paired with an online localization system could be used to do continuous localization and mapping, as shown in prior research [32]. This method is appropriate for mapping topological and natural features. A few approaches were tested to include the artificial beacons, in which the RFID tags' locations are approximated using the maximal likelihood path regarding a robot evaluated via FastSLAM algorithm and recursive Bayesian altering/Monte Carlo localization. To use the Trilateration approach for robot localization, there is a need for knowing the positions of at least three beacons. Because GPS signals may be absent or because the positions of noisy robots and beacons are not very exact, a SLAM must be utilized again. Because of the measurement inaccuracy, the circle interceptions frequently result in a series of points rather than a single point. To reduce the severity of the problem, two solutions must be considered:

- Apply probabilistic filters to the most possible region/points, like Histogram Filters (Grid Based), Kalman Filter, or other Particle Filters;
- Position beacons carefully. They must be set as high as feasible (to minimize interferences) and never on straight lines to ensure that they cover all the directions. GPS's geometric dilution of precision (GDOP) works on the same principle. The Kalman Filter is preferable under linear and optimum conditions, yet in an agricultural setting, Particle Filters, Histogram Filters, or Extended Kalman filters might be more appropriate. The fundamental concept behind Histogram based is using statistics from the surrounding area to estimate the most likely location. It entails the discretization related to space into a grid map where each one of the cells is allocated a likelihood of containing the object [22]. The Particle Filters use the distribution of the particles with a likelihood of being close to a robot that is updated constantly as measurements have been taken. The distribution is uniformly initialized, and particles that have a lower likelihood are removed at each step. The estimates are based on the particle position centroid.

c. **Strategies and Solutions**

The major optimization goals of the coverage problems have been categorized as:

1. Maximizing the quality of the coverage;
2. Maximizing network lifetime;
3. Maximizing the number of sensors.

Basic techniques for various coverage goals, scheduling, deployment, and movement are only a few of the options for achieving their goal. The sensors' allocation to ensure coverage quality regarding the basic network is referred to as deployment. Scheduling is the process of selecting a subset of the sensors that are to be enabled for a specified period while other sensors are put to sleep. The term "selection strategy" indicates one time choice for a sub-set of sensors throughout a period. The movement approach deals with sensor reallocation. Coverage Issues in WSNs: A Survey 161 Deployment methods This original position could be used to calculate the initial position regarding each one of the sensors and sensors could then be accordingly deployed. However, there's a chance that sensors will deviate from their designated position. A few deployment approach applications are discussed.

- Increasing the quality of the coverage; increasing the lifetime of the network; reducing the number of sensors. Limited sensors are used to collect as many ROI features as feasible [26].
- Increasing network lifetime: Individual sensor lifetime is determined by battery life, but WSN lifetime indicates the amount of time that installed sensors could provide appropriate coverage quality. The placement of sensors can considerably lengthen the network's lifetime [22].
- Reducing the number of sensors: By deploying radar sensors, the authors of [38] consider barrier coverage in the bi-static radar networks.

The latter, on the other hand, takes into consideration the cost differential between receiver and transmitter sensors. Strategies for selection or scheduling. can establish a deployment strategy if know the sensors' initial positions. Utilize a network builder to distribute a large number of sensors in a random pattern to meet various coverage quality requirements. Sensors are set to function alternately for optimal usage of redundant sensors. This approach is utilized to extend the lifetime of a network and is simple to deploy. Adjustment or movement techniques. Sensors are currently integrated with mobility features. Those mobile components take advantage of movement strategy. may alter the position of the components because they move. One of the primary issues in this regard is energy consumption, for which decreasing movement distance becomes critical. It's also crucial to modify sensor characteristics like angle, detecting range, and so on. The authors of [39] propose a three-stage optimization strategy for deploying such sensors, including network connectivity restoration, pattern node selection, and coverage hole repair, to maximize the covered volume ratio while maintaining connectivity.

The problem of area coverage with the rotatable directional sensors is tackled in [30]. Staged Rotation and Motion Control (SRMC) and Concurrent Rotation and Motion Control (CRMC) are two methods that have been proposed.

12 Conclusions

The addition of mobility increases the performance of WSNs in a variety of ways. There have already been certain survey articles in the domain of WSNs with MEs, and a lot of attempts have gone into network design considering the different properties of WSNs. Those surveys, on the other hand, primarily focus on data relay and/or data collection, which represents a sub-set of suggested taxonomy depending on ME tasks, Collection, Delivery, and Combination. A basic difficulty in robotics is safe navigation. It is a necessary prerequisite for industrial autonomous mobile robots, particularly in smart factories. Industrial mobile robots had become commonly utilized in numerous manufacturing areas, like electronics and automotive, as a result of the advancement of industrial automation. Industrial mobile robots must be capable to roam on the factory floor while avoiding equipment, walls, walking people, and other robots in a smart factory. With the rapid advancement of industrial automation, there is a growing demand for effective, simple, and integrated robot navigation systems for a variety of robots in smart factories.

13 Future Directions

Wireless sensor network robot has been considered as a prospective area as a result of the complementary natures of the WSNs and robotics. A system that integrates both may be utilized to replace human presence in areas that are too dangerous to send humans. The applications that potentially benefit from these implementations, like military operations or search and rescue, demand further research due to their complementary nature. The development of these networks will be pushed ahead by studies into energy-efficient real-time protocols and control mechanisms, which will lead to more extensive and useful applications. The migration of the Mobile Robot WSN system from TinyOS to embedded Linux OS, the implementation of a discovery algorithm to detect robots arriving and exiting the area, and the migration of the microserver architecture to embedded Linux OS, thereby removing the robotic framework overhead. Additionally, are looking into WSN routing algorithms that are better suited for dynamic topologies, such as in the scenario where several WSN nodes are mobile nodes.

References

1. Majid M, Habib Sh, Javed A, Rizwan M, Srivastava G, Gadekallu Th, Wei Lin J (2022) Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: a systematic literature review. Sensors
2. Salman K, Hamza E (2021) Visible light fidelity technology: survey. Iraqi J Comput Commun Control Syst Eng
3. Reis R, Mendes J, Santos F, Morais R, Ferraz N, Santos L, Sousa A (2018) Redundant robot localization system based in wireless sensor network. In: 2018 IEEE international conference on autonomous robot systems and competitions (ICARSC). IEEE
4. Saba A, Ekhlas K, Qiang N, Alagan A (2017) Industrial internet of things driven by SDN platform for smart grid resiliency. IEEE Internet Things J
5. kim Y, Kang H, Lee S, Pak J, Ahn CH (2022) Distributed finite memory estimation from relative measurements for multiple-robot localization in wireless sensor networks. IEEE Access

6. Nafea Sh, Hamza E (2020) Path loss optimization in WIMAX network using genetic algorithm: Int J Comput Commun Control (IJCCC)
7. Maheshwari A, Chand N (2019) :A survey on wireless sensor networks coverage problems. In: Proceedings of 2nd international conference on communication, computing and networking. Springer, Singapore
8. Salman K, Hamza E (2022) Indoor positioning systems based on Li-Fi technology using RSS-triangulation with assisted by DNN. INASS
9. Al-Mousawi A (2020) Evolutionary intelligence in wireless sensor network: routing, clustering, localization and coverage. Wireless Networks
10. Ryu Ji, Rfan M, Reyaz A (2015) A review on sensor network issues and robotics. J Sensors
11. Sam S, Conrad J (2013) A survey of robotic applications in wireless sensor networks. In: 2013 proceedings of IEEE Southeastcon. IEEE
12. Kazala R, Taneva A, Petrov M, Penkov St (2015) Wireless network for mobile robot applications. In: IFAC-PapersOnLine
13. Huang H, Savkin A, Ding M, Huang Ch (2019) Mobile robots in wireless sensor networks: a survey on tasks. In: Computer Networks
14. Ekhlas K, Russul H (2018) Performance analysis of IEEE 802.15.4 transceiver system under adaptive white Gaussian channel. Int J Electr Comput Eng
15. Hamza E, Al-asady H (2018) Indoor localization system using wireless sensor network. Iraqi J Comput Commun Control Syst Eng (IJCCCE)
16. Sayed A, Ammar H (2020) Centralized multi-agent mobile robots SLAM and navigation for COVID-19 field hospitals. In: 2020 2nd novel intelligent and leading emerging sciences conference (NILES) in IEEE
17. Kumar B, Sharma L, Wu Sh (2018) Job allocation schemes for mobile service robots in hospitals. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM)
18. Hamid H, Kadhum E, Ismail W, Singh M (2009) Analyze BER performance of wireless FSK system. Microwave and RF
19. Sharma R, Prakash S (2020) Enhancement of relay nodes communication approach in WSN-IoT for underground coal mine. J Inf Optim Sci
20. Khan A, Aziz S, Bashir M, Khan M (2020) IoT and wireless sensor network based autonomous farming robot. In: 2020 international conference on emerging trends in smart technologies (ICETST). IEEE
21. Haque Md, Asikuzzaman Md, Khan I, Ra In-Ho, Hossain Md, Shah S (2020) Comparative study of IoT-based topology maintenance protocol in a wireless sensor network for structural health monitoring. Remote Sensing
22. Majid M, Habib S, J aved A, Rizwan M, Srivastava G, Gadekallu Th, Lin J. (2022) Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: aA systematic literature review. Sensors
23. Majid M, Habib Sh, Javed A, Rizwan M, Srivastava G, Gadekallu Th, Lin J (2022) Distributed finite memory estimation from relative measurements for multiple-robot localization. In: Wireless Sensor Networks, IEEE Access
24. Wu Y, Dai H, Wang H (2020) Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. IEEE Internet Things J
25. Wua Y, Taa X, Xiaoa R, Weia Y, Ana D, Lia D (2019) Survey of underwater robot positioning navigation. In: Applied ocean research
26. Zhu J, Wang B (2014) Sensor placement algorithms for confident information coverage in wireless sensor networks. In: 2014 23rd international conference on computer communication and networks (ICCCN), IEEE
27. Zhang L, Li D, Zhu H, Cui L (2012) Open: an optimisation scheme of n-node coverage in wireless sensor networks. IET Wirel Sens Syst
28. Chen J, Wang B, Liu W (2015) Constructing perimeter barrier coverage with bistatic radar sensors. J Netw Comput Appl (JNCA)
29. Gong X, Zhang J. Cochran D, Xing K (2016) Optimal placement for barrier coverage in bistatic radar sensor networks. ACM/IEEE Trans Netw

30. Wang Z, Wang B (2017) A novel node sinking algorithm for 3d coverage and connectivity in underwater sensor networks. Ad Hoc Netw
31. Li W, Chen Z, Gao X, Liu W, Wang J (2019) Multimodel framework for indoor localization under mobile edge computing environment. IEEE Internet Things
32. Calderoni L, Ferrara M, Franco A, Maio D (2015) Indoor localization in a hospital environment using random forest classifiers. Expert Syst Appl
33. Davidson P, Piché R (2017) A survey of selected indoor positioning methods for smartphones. IEEE Commun Surv Tutor
34. Roy P, Chowdhury C (2021) A survey of machine learning techniques for indoor localization and navigation systems. J Intell Robot Syst
35. Roy P, Chowdhury C (2021) WiFi CSI-based device-free sensing: from Fresnel zone model to CSI-ratio model. CCF Trans. Pervasive Comput Interact
36. Yassin A, Nasser Y, Awad M, Al-Dubai A, Liu R, Yuen C, Raulefs R, Aboutanios E (2016) Recent advances in indoor localization: a survey on theoretical approaches and applications. IEEE Commun Surv Tutor
37. Kokkonen H et al (2022) Autonomy and intelligence in the computing continuum: challenges, enablers, and future directions for orchestration. arXiv preprint arXiv
38. Zafari F, Gkelias A, Leung K (2019) A survey of indoor localization systems and technologies. IEEE Commun Surv Tutor
39. Zhang Z, Nie W, Wang Y, Xie L (2020) Channel state information based indoor localization error bound leveraging pedestrian random motion. IEEE Access