Systems Engineering
Methods, Developments
and Technology

A Closer Look at Fault-Tolerant Control

Jeremy M. Hutton
Editor

NOVA

SYSTEMS ENGINEERING METHODS, DEVELOPMENTS AND TECHNOLOGY

A CLOSER LOOK AT FAULT-TOLERANT CONTROL

SYSTEMS ENGINEERING METHODS, DEVELOPMENTS AND TECHNOLOGY

SYSTEMS ENGINEERING METHODS, DEVELOPMENTS AND TECHNOLOGY

A CLOSER LOOK AT FAULT-TOLERANT CONTROL

JEREMY M. HUTTON
EDITOR

nova
science publishers
New York

NOTICE TO THE READER

The Publisher has taken reasonable care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained in this book. The Publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the readers' use of, or reliance upon, this material. Any parts of this book based on government reports are so indicated and copyright is claimed for those parts to the extent applicable to compilations of such works.

Independent verification should be sought for any data, advice or recommendations contained in this book. In addition, no responsibility is assumed by the Publisher for any injury and/or damage to persons or property arising from any methods, products, instructions, ideas or otherwise contained in this publication.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered herein. It is sold with the clear understanding that the Publisher is not engaged in rendering legal or any other professional services. If legal or any other expert assistance is required, the services of a competent person should be sought. FROM A DECLARATION OF PARTICIPANTS JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Additional color graphics may be available in the e-book version of this book.

Library of Congress Cataloging-in-Publication Data

Names: Hutton, Jeremy M., editor.
Title: A closer look at fault-tolerant control / Jeremy M. Hutton.
Description: New York : Nova Science Publishers, 2020. | Series: Systems engineering methods, developments and technology | Includes bibliographical references and index. |
Identifiers: LCCN 2020005567 (print) | LCCN 2020005568 (ebook) | ISBN 9781536175288 (hardcover) | ISBN 9781536175295 (adobe pdf)
Subjects: LCSH: Automatic control. | Fault tolerance (Engineering) | Control theory.
Classification: LCC TJ213 .C55 2020 (print) | LCC TJ213 (ebook) | DDC 629.8--dc23
LC record available at https://lccn.loc.gov/2020005567
LC ebook record available at https://lccn.loc.gov/2020005568

Published by Nova Science Publishers, Inc. † New York

CONTENTS

PREFACE

A Closer Look at Fault-Tolerant Control first presents the application of a fault tolerant control system on distillation processes, with automatic actuator faults containment capabilities and an atmospheric crude distillation unit.

Following this, model-based fault-tolerant control and fault accommodation algorithms are presented for two challenging classes of distributed systems: a spatially distributed system that can be decomposed into interconnected subsystems, and a distributed parameter system where the system state is distributed over a continuous range of space.

The authors present recent research on fault-tolerant control systems for unmanned aerial systems, particularly for multirotor-type vehicles commonly known as drones.

An overview of tools for the analysis of the fundamental properties of an automated system is provided, allowing for any inherent redundancy in the controlled process to be utilised to maintain availability.

Additionally, a reconfigurable fault-tolerant flight control system is proposed to combat sensor/actuator faults for autonomous underwater vehicles.

The reconfigurable design and operation of complex systems is addressed, with emphasis on autonomous systems, building upon concepts of autonomy, incipient failure diagnosis and prognosis algorithms.

The authors present a fault detection filter for induction motors speed as a class of nonlinear system in networked control systems subject to induced time delays. The multi-model approach for the modeling of induction motors is described using a set of linear models.

In the concluding study, the construction of an induction motor is presented, and a review of induction motor failures is discussed.

Chapter 1 - It is unimaginable these days that any facility will be built or retrofitted in the oil and gas industry without a considerable level of automation. There is an increase in the complexity and sophistication of modern control systems deployed in the industries, especially on safety-critical systems. This growing complexity comes with some level of inherent susceptibilities, part of which is the potential failure in some of the components that make up the control system, such as actuators and sensors. The risk is even higher in complex chemical plants like refinery with hundreds to thousands of sensors and actuators. The interplay between these components and the control system needs to have some built-in robustness to guarantee high level of safety and reliability of the plant, which is fundamental to the operation of the system. More so, meeting the economic and operational targets of the system requires its continued safe operation even in the presence of faults in the system or some of its control system components. This chapter presents the application of frugally designed fault tolerant control system (FTCS) with automatic actuator faults containment capabilities on distillation processes, particularly atmospheric crude distillation unit. A simple active actuator FTCS that uses backup feedback signal, switchable references and restructurable PID controllers was designed and implemented on distillation processes with varying complexities – the benchmark Shell heavy oil fractionator and an interactive dynamic crude distillation unit (CDU) to accommodate actuator faults.

Chapter 2 - Faults are inevitable and even incipient faults that progress very slowly can downgrade the performance of the system. In cases where a fault is not critical, the system performance can be kept at an acceptable level by mitigating the effect of fault. In this chapter, model-based fault-tolerant control and fault accommodation algorithm are presented for two challenging classes of distributed systems; first a spatially distributed system that can be decomposed into interconnected subsystems, and second a distributed parameter system where the system state is distributed over a continuous range of space. The design of a decentralized fault tolerant controller (DFTC) is presented for interconnected nonlinear continuous-time systems by using local subsystem state vector alone in contrast with traditional distributed fault tolerant controllers or fault accommodation schemes where the measured or estimated state vector of the overall system is needed. The decentralized controller uses local state and input vectors in each subsystem and minimizes the fault effects on the entire system. The DFTC in each subsystem includes a traditional controller and a neural network based online approximator which is used to deal with the unknown parts of the system dynamics, such as fault and interconnection terms. The stability of the overall system with DFTC is investigated by using Lyapunov approach and the boundedness of all signals is guaranteed in the presence of a fault. Therefore, the proposed controller enables the system to continue its normal operation after the occurrence of a fault, as long as it does not cause failure or break- down of a component. Next, a model-based fault accommodation scheme is introduced for a class of linear distributed parameter systems (DPS) represented by partial differential equations (PDEs) in the presence of both actuator and sensor faults. A filter-based observer on the basis of the linear PDE model of the DPS is designed with output measurements. The estimated output from the observer and the measured outputs are utilized to generate a residual for fault detection. Upon detection, the fault function is estimated by using an unknown parameter vector and a known basis function. Update laws are introduced to estimate the unknown fault parameter vector for actuator and sensor faults. These estimates will then be used to modify the nominal controller in order to accommodate the actuator and sensor faults.

Chapter 3 - This chapter presents some recent results on fault-tolerant control systems for unmanned aerial systems, in particular for multirotor-type vehicles, commonly known as drones. Over the last years, these vehicles have become widely popular. Simplicity and cost-effectiveness have turned out to be very appealing and, as a consequence, an increasing number of applications have risen in many fields such as agriculture, surveillance, and photography, among others. As mission requirements become more demanding, the matter of fault tolerance emerges as a key challenge, especially if system certification is sought.

Here, the focus is placed particularly on rotor failures in multirotor vehicles, and a specific definition for fault tolerance is considered based on the maneuverability capabilities in case of a failure. A geometricanalysis is presented to evaluate the fault tolerant capabilities of a given vehicle, together with an experimental validation. Then, the limitations of this conceptareanalyzed. Finally, a novel reconfigurable structure is proposed for a fault-tolerant hexarotor, that presents good flight performance in failure cases, together with experimental results.

Chapter 4 - Faults in automated processes will often cause undesired reactions and shutdown of a controlled plant, and the consequences could be damage to technical parts of the plant, to personnel or the environment. Fault tolerant control combines diagnosis with control methods to handle faults in an intelligent way. The aim is to prevent that simple faults develop into serious failure and hence increase plant availability and reduce the risk of safety hazards. Fault-tolerant control merges several disciplines into a common framework to achieve these goals. The desired features are obtained through online fault diagnosis, automatic condition assessment and calculation of appropriate remedial actions to avoid certain consequences of a fault. The envelope of the possible remedial actions is very wide. Sometimes, simple re–tuning can suffice. In other cases, accommodation of the fault could be achieved by replacing a measurement from a faulty sensor by an estimate. In yet other situations, complex reconfiguration or online controller redesign is required. This chapter gives an overview of well–established and more recent tools to analyse and explore structure and other fundamental properties of an automated system such that any inherent redundancy in the controlled process can be fully utilised to maintain availability, even though faults may occur. On the other hand, the effectiveness of the analysed solutions has been verified when applied to a wind turbine system. In fact, wind turbine plants are complex dynamic and uncertain processes driven by stochastic inputs and disturbances, as well as different loads represented by gyroscopic, centrifugal, and gravitational forces. Moreover, as their aerodynamic models are nonlinear, both modelling and control become challenging problems. On one hand, high–fidelity simulators should contain different parameters and variables in order to accurately describe the main dynamic system behaviour. Therefore, the development of fault tolerant control solutions for wind turbine systems should consider these complexity aspects. On the other hand, these solutions have to include the main wind turbine dynamic characteristics without becoming too complicated. The second point of this chapter is thus to provide practical examples of the development of robust fault tolerant control strategies when applied to a simulated wind turbine plant. Experiments with the wind turbine simulator represent the instruments for assessing the main aspects of the developed control methodologies.

Chapter 5 - In this study, a reconfigurable fault-tolerant flight control system against sensor/actuator faults for autonomous underwater vehicles (AUVs) is proposed. First, an approach for detecting and isolating AUV sensor/actuator faults affecting the mean of the Kalman filter (KF) innovation sequence is proposed. Second, an augmented Kalman filter is used to isolate the sensor and actuator faults and estimate the control derivatives corresponding to the faulty actuator. In the case of a sensor fault, the robust Kalman filter algorithm with the filter gain correction is used. With the use of defined variables named as measurement noise scale factor, the faulty measurements are taken into consideration with a small weight and the estimations are corrected without affecting the characteristic of the accurate ones. In case of an actuator fault, fault isolation and identification are performed using the augmented KF. The control reconfiguration procedure is executed by utilizing the identified control distribution matrix. The parameters of the feedback controller are tuned by the control reconfiguration procedure. In the simulations, the steering subsystem dynamics of the AUV model is considered, and the sensor/actuator fault detection and isolation are examined. Some simulation results for the reconfigurable active fault tolerant control against actuator faults are given.

Chapter 6 - This contribution addresses the reconfigurable design and operation of complex systems, with emphasis on autonomous systems, building upon concepts of autonomy, incipient failure diagnosis and prognosis algorithms, while introducing a novel methodology for reconfigurable design, control and/or operation formulated as an optimization problem where new or reconfigured designs and their operational characteristics are optimized to perform as designed/desired. The innovative feature of the adverse event mitigation architecture is the utility of real-time prognostic information in the design of the control algorithms. Given accurate on-line prognostic information in terms of estimates of the Remaining Useful Life (RUL) or Time to Failure (TTF) of a failing component/subsystem, the proactive fault accommodation system manages the accumulation of further damage through control actions until major flight/mission objectives are achieved although the system is in an impaired state. This approach constitutes a major paradigm shift in the way fault-tolerant systems are designed and operated. The implications to system survivability, safety and availability to complete a critical flight/mission are significant. Existing/published research focuses either on single component (i.e., navigation controller), or specific system (i.e., single type of Unmanned Aerial Vehicle UAV) reconfiguration, or reconfigurable control [without providing the fundamentals of a general and justifiable methodology for overall (hardware/software components) system reconfiguration. Reconfiguration is achieved based on metrics related to measures of effectiveness and performance. Once metrics are defined, graph-based (dependency, directed graphs) and non-homogeneous Markov-based modeling approaches are followed to arrive at different system configurations and choose the best alternative according to mission requirements.

The authors introduce two complementary approaches to fault tolerance or reconfigurable control of complex unmanned systems. Both assume that an incipient failure or fault is detected and the failing component's remaining useful life is estimated. The fault to failure evolution allows a sufficient period for the application of the reconfiguration strategy. A self-organization method is introduced as a compensatory measure to maintain system functionality under the presence of failure modes. It is noted that resilience requirements refer to severe disturbances, i.e., failure modes compared to usual disturbances compensated by conventional technologies such as robust or PID control. A typical unmanned autonomous ground vehicle – the hexapod – is employed as the testbed for the development and validation of the self-organizing strategy. Methods to understand system behavior include data acquisition, system modeling, and proper construction of performance metrics; the strategy includes a policy to address the changing system conditions and success criteria to evaluate the optimal action. The physical, functional, nonlinear dynamic, and graph theoretic models will be considered to examine system behaviors under both normal and faulty conditions. Then, the self-organization strategy is introduced in the form of a Markov Decision Process (MDP) with dynamic programming for optimal performance. Finally, the success criteria for the control method are constructed with Lyapunov stability conditions so that the self-organization strategy can be modified throughout the system operation for system resilience regarding stability and resource limitations. Simulation results are presented to demonstrate the efficacy of the approach. The second approach introduces a design methodology for resilient-based control reconfiguration of Unmanned Autonomous Systems (UAS) when extreme disturbances, such as a largely growing fault or a component failure mode occur. An optimal control approach with Differential Dynamic Programming (DDP) and Model Predictive Control (MPC) is deployed as a means for control authority redistribution and reconfiguration; the system continues performing its mission while compensating for the impact of the extreme disturbances. Prognostic knowledge is considered in a quadratic cost function of the optimal control problem as a soft constraint. A trade-off parameter is introduced between the prognostic constraint and the terminal cost. An autonomous ground operable under-actuated hovercraft is employed to demonstrate the efficacy of the proposed reconfiguration strategy.

Chapter 7 - In this chapter, the authors present a fault detection filter for the induction motor speed as a class of nonlinear system in networked control systems (NCSs) subject to induced time delays. The authors used the multi-model approach for modeling of induction motor described by a set of linear models. Recent research shows that the multi-model approach is a powerful tool to deal with nonlinear system. Thus, the authors were interested particularly in electric machine, especially in induction motor as a strongly nonlinear system. The necessity to assure the induction motor safety operation implicates protective supervision process based on fault diagnosis techniques. The first focus of this chapter is to describe the induction motor via an interpolation of a set of linear local models. This representation require a strategy of four steps that are database acquisition, cluster estimation, structural and parametric identification and local models combination. Then, an adaptive state filter is presented which can provide the information of faults and states of induction motor. In reality, certain observations may be missing possibly due to network-induced delay, random packet dropout; access constraints, etc. Therefore, in this work, an approach is proposed to perform estimation in network-induced delay. The induced time delays are from the controller to the plant and from the sensor to the controller. An example is included to show the efficiency of the proposed method.

Chapter 8 - The chapter deals with a diagnosis of an induction motor followed by sensor failure modes. First, construction of induction motor has been presented. Then a review of induction motor failures has been discussed. The third part studies the problem of diagnosis strategy for an induction motor sensor faults. This strategy is based on unknown input proportional integral (PI) multiobserver. The need of a sensorless drive requires soft sensors such as estimators or observers. The convergence of the estimation error is guaranteed by using the Lyapunov’s based theory. The proposed diagnosis approach is experimentally validated on a 1 kW Induction motor. Obtained simulation results confirm that the adaptive PI multi-observer consent to accomplish the detection, isolation and fault identification tasks with high dynamic performances.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8

Chapter 1

ACTUATOR FAULT TOLERANT CONTROL SYSTEM FOR DISTILLATION PROCESSES

***Sulaiman A. Lawal*[1,2] *and Jie Zhang*[1,*]**
[1]School of Engineering, Merz Court, Newcastle University,
Newcastle upon Tyne, UK
[2]Chemical Engineering Department, University of Lagos,
Lagos, Nigeria

ABSTRACT

It is unimaginable these days that any facility will be built or retrofitted in the oil and gas industry without a considerable level of automation. There is an increase in the complexity and sophistication of modern control systems deployed in the industries, especially on safety-critical systems. This growing complexity comes with some level of inherent susceptibilities, part of which is the potential failure in some of the components that make up the control system, such as actuators and sensors. The risk is even higher in complex chemical plants like refinery with hundreds to thousands of sensors and actuators. The interplay between these components and the control system needs to have some built-in robustness to guarantee high level of safety and reliability of the plant, which is fundamental to the operation of the system. More so, meeting the economic and operational targets of the system requires its continued safe operation even in the presence of faults in the system or some of its control system components. This chapter presents the application of frugally designed fault tolerant control system (FTCS) with automatic actuator faults containment capabilities on distillation processes, particularly atmospheric crude distillation unit. A simple active actuator FTCS that uses backup feedback signal, switchable references and restructurable PID controllers was designed and implemented on distillation processes with varying complexities – the benchmark

[*] Corresponding Author's Email: jie.zhang@newcastle.ac.uk.

Shell heavy oil fractionator and an interactive dynamic crude distillation unit (CDU) to accommodate actuator faults.

Keywords: fault tolerant control, principal component analysis, crude distillation unit, actuator fault

1. INTRODUCTION

In spite of the successes recorded in the last four decades or so with the use of computers for conventional and advanced process control systems in our various industries, the task of responding to abnormal situations (i.e., faults) is mostly performed manually. Billions of dollars are lost in the industries every year due to low productivity, loss of operational hours, occupational injuries and illnesses resulting from major and common minor accidents occurring on a daily basis [1-3]. It was reported by Nimmo [4] that United States petrochemical industry alone incurs approximately 20 billion US dollars in annual losses, while United Kingdom records up to 27 billion US dollars losses every year [5] due to poor abnormal event management (AEM). It is also interesting to know that about 70% of industrial accidents are caused by human errors [6]. Despite advances in computer-based control applications in the industries, the fact that some of the worst chemical and nuclear power plants accidents, namely Nuclear Tsunami of March 2011 (though caused by unforeseen natural disaster) that had devastating effect on Japanese economy; Santrach's LNG plant explosion (Skikda, Algeria) on January 19, 2004 where 27 people died, and 56 were injured; Kuwait Petrochemical's Mina Al-Almedi refinery in June 2000; Occidental Petroleum's Piper Alpha accident [7] on July 6, 1988 that resulted in the death of 162 employees of the company; Chornobyl Nuclear Power Plant on April 26, 1986; Union Carbide's Bhopal, India, accident of December 3, 1984 that caused 3,800 deaths and approximately 11,000 disabilities [8], just to mention a few, all happened in the last three decades or so. It is inevitable that some processing equipment including actuators, sensors and control systems will breakdown or malfunction at some point during their operational life span. Hence, it will be desirable to have a control system that can accommodate those potential failures during operation while still maintaining acceptable level of performance, albeit with some graceful degradation. Having smart control systems with some fault tolerant capabilities on these plants would have offered some robustness in the overall control architecture, and ultimately give sufficient time to repair the impaired systems.

The increasing availability and application of intelligent actuators and sensors with built-in diagnostic capabilities in several industries, oil and gas inclusive also supports the efforts towards the development of smart plants. The demand for development and application of smart controllers with built-in diagnostics and reconfigurable capabilities for optimal operation and management of plants during normal and abnormal situations in the process industries is therefore on the increase. These smart controllers could be referred to as fault tolerant control systems (FTCS). FTCS is an advanced control system with automatic components containment capabilities. It is necessitated by the increasing demand for higher performance, improved safety, reliability and availability of control systems in the event of malfunctions in actuators, sensors and or other system components. FTCS is also expected to provide desirable performance on complex automated facilities when process equipment, actuators, and sensors breakdown or malfunction during operation.

FTCS has received a great deal of interest in both the industry and in the academia, but its actualization has faced some challenges in terms of its applicability in the industry. FTCS has two major components – fault detection and diagnosis (FDD) and fault tolerant controllers (FTC). FDD is a matured research area. Researches in this area span over four decades with different and diverse techniques employed [9-23]. FDD mainly detects fault, isolates and estimates its magnitude, and feeds the information to FTC, which then in real-time reconfigures as appropriate to ensure acceptable performance in the impaired system.

There are some commercial equipment monitoring and health management packages in use in the industries, such as Profit Sensor from Honeywell, Plant Triage from Expertune and AMS from Emerson. These packages employ techniques such as Multiple Linear Regression (MLR) and Principal Components Analysis (PCA) in monitoring the process variables and health of the system components, but have no integrated fault tolerant controllers to take corrective actions when faults are detected. The architecture and integration of FDD and FTC to form FTCS sounds pretty straightforward theoretically, but in actual fact, its actualization has faced numerous challenges as most of the developed FDD techniques are for monitoring purposes rather than control purposes. Admittedly, significant effort has been made recently in FTCS, where many algorithms and methods have been developed in different application areas [14-16, 24-40]. However, there are still issues to be addressed in the application of FTCS to oil and gas processes. Some of these challenges include the ability of the FDD component to quickly and accurately detect and diagnose different faults (actuator, sensor and component faults); the mechanism for effective integration of FDD and FTC; the suitability of the FTCS to handle non-linear systems; its robustness to noise and uncertainties and the complexity of computation required during implementation.

Hence, with the challenges listed above, this chapter contributes to the furtherance of the development and application of FTCS to the oil and gas processes, particularly the distillation processing units with special focus in its computational complexity, ease of implementation and effective FDD and FTC integration mechanism. This involves the development and application of simple restructurable feedback controllers with backup feedback signals and switchable reference points to tolerate actuator faults in fractionator and crude distillation units.

The chapter is organised as follows. Next section presents a brief review of the different components of fault tolerant control systems. Relevant state of the art fault detection and diagnosis techniques are summarily reviewed to assess their suitability for the development of the FTCS for complex chemical plants. Different techniques employed under model-based and data-based fault detection and diagnosis are also outlined in this section. The different approaches that have been researched in the development of fault-tolerant controllers for complex systems including model predictive control are assessed. Detailed design of the proposed simplified actuator fault-tolerant control system is presented thereafter. The control strategies and the tools employed in identifying and analysing different control loops pairing pre and post-fault era in order to achieve a seamless switching and stability in the system post-fault era are then presented. The last section in this chapter focuses on the implementation of the developed FTCS for actuator faults accommodation on two distillation processes – the Shell heavy oil fractionator unit and a crude distillation unit.

2. Review of Fault Tolerant Control System

A significant number of researches has been carried out in FTCS, leading to the multiplicity of a wide range of techniques in different application areas [11, 14, 15, 24, 25, 27, 28, 31-37, 39-41]. FTCS is broadly classified into two types – passive and active fault-tolerant control systems (PFTCS and AFTCS). The classification is functional, based on how the controllers handle faults in systems. Passive FTCS have predesigned control laws that are made insensitive to some known faults and have limited capabilities on the range and magnitude of faults they can handle. Active FTCS, on the other hand, have built-in fault monitoring diagnostic component that can detect the occurrence of faults in real-time and relay the information to the reconfigurable controller component of the control system to act, maintaining some level of acceptable performance in the system despite the fault. Further considerations on the classes of FTCS and their many different components, as well as the major relevant state of the art techniques that have been applied in the field of FTCS will be discussed in this section.

2.1. Passive Fault Tolerant Control Systems

Passive fault tolerant controllers are also referred to as reliable controllers. Passive controllers usually have fixed structure and are without built-in diagnostics to detect and diagnose faults in any system, as such, they are not referred to as smart controllers.

Several authors have worked on PFTCS. Liang et al. [42] worked on state feedback controllers that can accommodate a predefined set of actuator faults for nonlinear systems using Hamilton-Jacobi inequality without any fault diagnostic component. Hsieh [43] proposed a unified gain margin constraint approach to develop a reliable, guaranteed cost controller using two-stage linear quadratic (LQ) reliable control technique. Veillette et al. [44] presented the design of reliable centralised and decentralised control systems that guarantee stability and H-infinity performance pre and post-fault era for sensor or actuator faults in the centralised control system, and for control channel faults in the decentralised case. Siljak [45] and Yang et al. [46] considered reliable control system design through the use of multiple identical controllers that guarantee internal stability and H-infinity performance before, during and after the occurrence of a sensor and or an actuator fault. Yang et al. [47] considered the application of reliable LQ state-feedback regulators to provide stability for discrete-time systems with actuator failures. They also presented procedures for designing reliable H-infinity controllers that guarantee asymptotic stability and H-infinity performance during normal operation and in the presence of faults in sensors and actuators for linear systems. Zhao and Jiang [48] proposed robust pole region assignment techniques using a dynamic pre-compensator to modify the dynamic characteristics of the redundant actuator control channels and offer reliable control performance. Yang and Zhang [49] discussed a method that guarantees closed-loop stability and an H-infinity-norm bound using multiple similar controllers based on algebraic Riccati equation approach to accommodate actuator faults. All these techniques need neither FDD system nor reconfigurable controllers to function, hence have limited capabilities in handling more serious faults. More insight and a brief review of researches in PFTCS can be found in Yu and Jiang [50].

2.2. Active Fault Tolerant Control Systems

Active FTCS is an advanced control system with automatic components containment capabilities that provides desirable performance on complex automated facilities whether faults are present or not. There are many stakeholders in AFTCS research field, especially from within the academia which reflects its multidisciplinary nature. The improved consideration the field has received recently was necessitated by the need to achieve a higher level of reliability, maintainability and performance in situations where controlled systems can have potentially damaging effects on the personnel, plant and the environment if faults occur in its or other system components [51]. Modern control systems are becoming increasingly complex and control algorithms even more sophisticated. Consequently, the issues of availability, cost efficiency, reliability, operating safety and environmental protection are of major importance [52]. Active Fault Tolerant Control System (AFTCS) aims to prevent catastrophic consequences of fault by reconfiguring the control system to maintain satisfactory operational performance even with severe faults. Control actions are generated based on the observed faulty situations to achieve the process objectives using the process information from the remaining functional sensors and manipulating the available healthy actuators [25].

AFTCS is of significant practical importance to the oil and gas industry – the focus of this chapter. It offers benefits in addition to those offered by advanced control systems through possession of diagnostics features that provide accurate timely information on the occurrence of faults, such as sensor and actuator faults and the capabilities to manage such failures in the control system components thereby maintaining the integrity of not just the control system, but of the entire operation. Fault tolerance and diagnostics capabilities are considered as some of the features of intelligent systems. According to Stengel [36] "Fault tolerant control systems, by design or implementation are intelligent systems." Astrom [53] is of the same opinion that fault diagnostic capability is an essential ingredient property of an intelligent system. One could argue that the use of FTCS in our various industries, especially in chemical and petroleum processing industry could be the norm in the next few decades, to take advantage of the increasing use of smart sensors and actuators. The main motivation for the application of FTCS in the chemical and oil and gas processing industries was driven historically by its application in the aircraft flight control systems [35]. However, FTC has not been widely applied to the oil and gas industry, hence the need for this research.

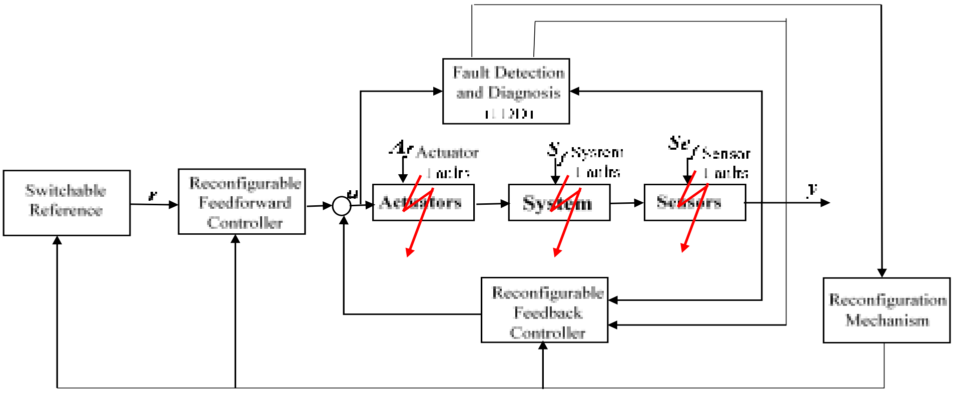


Figure 1. General structure of FTCS.

AFTCS has two major discrete components: fault detection and diagnosis (FDD) and fault tolerant controllers (FTC), and a third – controller reconfiguration and switching mechanism which handles the interplay between FDD and FTC to achieve a seamless AFTCS that meets its design objectives as shown in Figure 1. The effectiveness or otherwise of an appropriate FDD component of the AFTCS, which essentially detects, isolates and identifies faults will in large part determine the success or otherwise of the whole AFTCS. Also, the ease of controller reconfiguration and switching mechanism, in addition to having suitable healthy actuators and alternative measurements sources for input-output restructuring will be crucial for the fault-tolerant effort. Some of the commonly used FDD techniques are discussed next. Beyond this section, AFTCS is referred to as FTCS for simplicity.

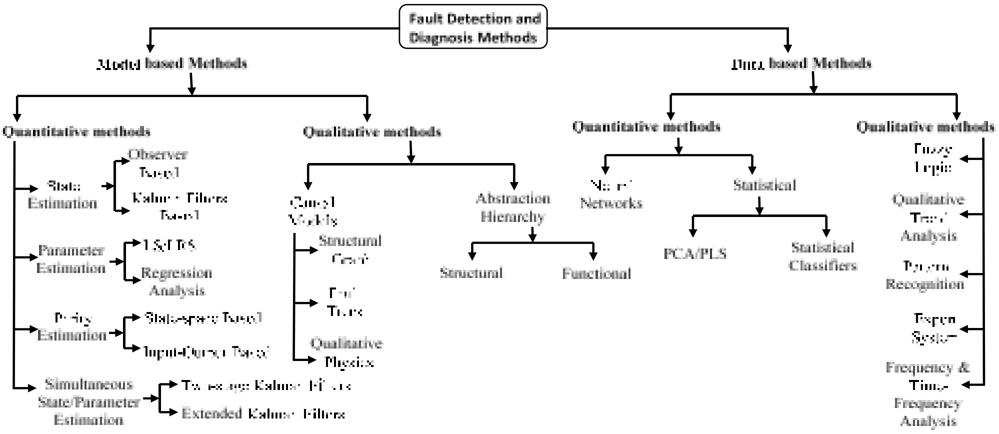
2.3. Fault Detection and Diagnosis

Fault detection and diagnosis has been researched extensively in the last four decades or so and several techniques have been developed. Some of the techniques are quantitative model based approaches [18, 21, 22, 51, 54-58], qualitative model based approaches [59, 60], data based/process history based approaches [9, 12, 13, 23, 61], and knowledge based approaches [62]. Several authors have adopted slightly varied and overlapping classifications of FDD. For example Zhang [12] adopted three broad classifications of FDD into model-based approaches, data analysis based approaches and knowledge-based approaches; Zhang and Jiang [14] used two broad classifications of model-based and data-based methods with each method further classified into quantitative and qualitative methods; while Venkatsubramanian et al. [10] broadly classify FDD into three categories: quantitative model-based methods; qualitative model-based methods and process history based methods. They all almost refer to the same broad classification with slightly different nomenclatures. This chapter adopts the two broad classifications of Zhang and Jiang [14] to summarise available FDD techniques, see Figure 2.

FDD is a crucial component of an FTCS. Its effectiveness determines the applicability, effectiveness and overall functionality of the resulting FTCS. An FDD scheme has three main tasks: (1) fault detection which detects the presence of fault in a system and the time it occurs; (2) fault isolation that determines the kind, location and time of detection of a fault; and (3) fault identification which provides information on the size and time-variant of fault [14, 63]. For the sake of clarity, FDD is used in this chapter to mean a combination of fault detection and isolation (FDI) plus the fault identification function [33]. Fault identification is the determination of type, size, location and time of detection of a fault [63]. There are certain minimum performance criteria a suitable FDD candidate must satisfy to fit into an overall structure of an active FTCS. Such desirable performance indices according to Zhang and Jiang [14] are:

- Ability to handle different type of faults (actuator, sensor and system component faults)
- Ability to produce quick and accurate detection
- Isolability, which is being able to differentiate between different faults
- Identifiability

- Suitability for fault tolerant control system integration
- Identifiability for multiple faults
- Suitability for nonlinear systems
- Robustness to noise and uncertainties
- Computational complexity



Note: LS/LRS – Least Squares/Recursive Least Squares; PCA – Principal Component Analysis; PLS – Partial Least Squares

Figure 2. Classification of FDD techniques.

The performance indices outlined above are the minimum requirements which an FDD scheme must satisfy, at least to a greater extent before one can hope for a practically applicable FTCS in the oil and gas industry. Model-based FDD is discussed next.

2.3.1. Model-Based Fault Detection and Diagnosis

The traditional approach to fault diagnosis is based on hardware or physical redundancy with the application of a voting scheme. The approach employs multiple lanes of sensors, actuators, computers and software to measure and or control a particular variable [52]. Imagine employing this approach in modern complex systems with hundreds, possibly thousands of variables to measure, monitor and control. Indeed, the drawbacks of having extra equipment and the accompanying costs, additional space for installation and the costs of maintenance will be of serious concern. To overcome these problems, analytical redundancy had been developed. It mainly uses the redundant (or functional) relationships between various measured variables of the monitored system. Analytical redundancy is deemed to be potentially more reliable. It does not need additional hardware to generate residual signal. Hence, no additional hardware fault will be introduced [64]. Figure 3 illustrates the concepts of hardware and analytical redundancy.

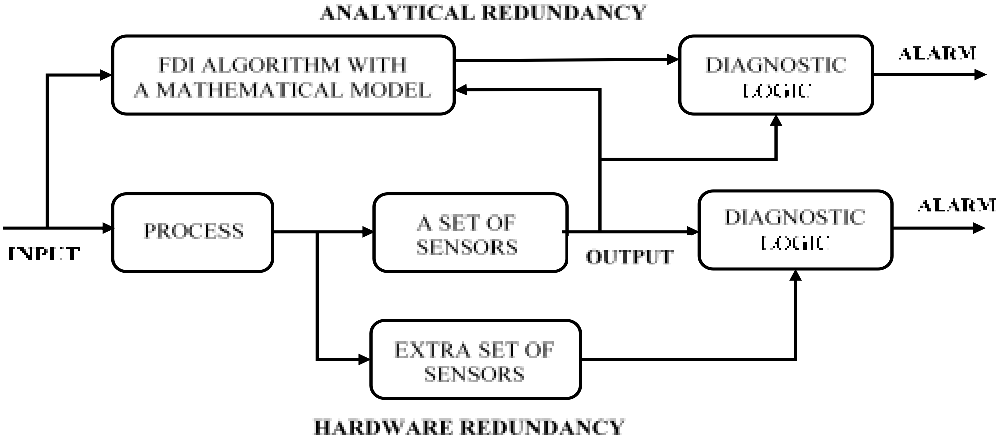


Figure 3. Hardware and analytical redundancy.

Patton and Chen [21] defined model-based fault diagnosis as "the determination of faults in a system by comparing the available system measurements with a priori information represented by the system's mathematical model, through generation of residual quantities and their analysis." Faults are declared when the residuals generated as a result of the difference between the measured variables and their estimates from the mathematical models reach or exceed a set of fixed or variable thresholds on the particular residual. A set of residuals can be designed with each having a unique sensitivity to individual faults occurring in different location in the system. Fault isolation is then achieved with subsequent analysis of each residual after a threshold has been breached. Application of this approach hinges heavily on having a good knowledge of the process and the relationship between faults and model states or parameters.

Also, an accurate mathematical model of the system is required, which is usually a constraint especially for complex chemical and petroleum processing facilities as considered in this chapter. Modelling the dynamics of a system with increasing complexity becomes more difficult due to uncertainties in respect to the system's structure, its parameters and the effect of disturbances on the system. The primary tasks of an FDD will be discussed under state estimation as one of the most frequently used model-based fault detection and isolation techniques. Another point worthy of mentioning is the issue of robustness in model based fault diagnosis. Robustness against modelling uncertainty that results from incomplete knowledge and understanding of the monitored processes is as important as the main objective for which the diagnostic scheme was designed. It has become an important research issue in recent time [21, 52]. Model based fault diagnosis involves two main stages of residual generation and decision making. It was initially proposed by Chow and Willsky [65] and is now generally accepted by the fault diagnosis community.

2.3.1.1. Faulty System Model

Building a mathematical model of the system under investigation is the first step in model-based fault diagnosis. A multiple-input multiple-output linear dynamic system is considered in this chapter. A model linearized around an operating point will be used for non-linear system. For the purpose of modelling a faulty system, an open-loop system is considered, which can be separated into three parts: actuators, sensors and system dynamics. Figure 4 presents the open loop system dynamics with actuator, sensor and component faults under consideration [52].

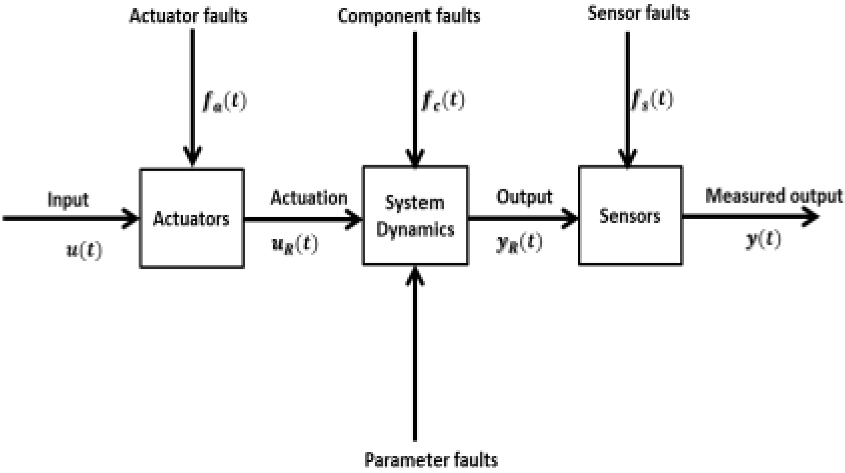


Figure 4. Open loop system dynamics.

The state space model of the system dynamics block in Figure 4 without the fault component is given as:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu_R(t) \\ y_R(t) = Cx(t) + Du_R(t) \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u_R \in \mathbb{R}^r$ is the input vector to the system, $y_R \in \mathbb{R}^m$ is the real system output vector, $f_s \in \mathbb{R}^m$ is the sensor fault vector and $f_a \in \mathbb{R}^r$ is the actuator fault vector, A, B, C and D are the known system matrices with appropriate dimensions. Including component fault in equation 1 above results in:

$$\dot{x}(t) = Ax(t) + Bu_R(t) + f_c(t) \tag{2}$$

The component fault affects the dynamics of the original system and needs to be captured in the model. Representing such a fault as a change in the system parameter, such as a change in the i_{th} row and j_{th} column element of matrix A, then we have:

$$\dot{x}(t) = Ax(t) + Bu_R(t) + I_i\Delta a_{ij}x_j(t) \tag{3}$$

Here, $x_j(t)$ is the j_{th} element of vector $x(t)$ and I_i is an all zero n-dimensional vector except the i_{th} element being 1. The output of the system is described below, with the sensor and actuator dynamics ignored.

A correct choice of the sensor and actuator fault vectors as presented in equations 4 and 5 can describe all sensor and actuator fault situations. Equation 6 describes a system with an unknown input, for instance, an uncontrolled system. Instead, an input sensor is used to measure the input to the actuator.

$$y(t) = y_R(t) + \ f_s(t) \tag{4}$$

$$u_R(t) = u(t) + \ f_a(t) \tag{5}$$

$$u(t) = u_R(t) + \ f_{is}(t) \tag{6}$$

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Bf_a(t) + f_c(t) \\ y(t) = Cx(t) + \ Du(t) + Df_a(t) + f_s(t) \end{cases} \tag{7}$$

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + R_1 f(t) \\ y(t) = Cx(t) + \ Du(t) + R_2 f(t) \end{cases} \tag{8}$$

$$y(s) = G_u(s)u(s) + \ G_f(s)f(s) \tag{9}$$

$$\begin{cases} G_u(s) \ = C(sI - A)^{-1}B \ + D \\ G_f(s) = C(sI - A)^{-1}R_1 + R_2 \end{cases} \qquad 10)$$

Equation 7 presents a system with all possible actuator, component and sensor faults while equation 8 is the compact state space model of a system with all possible faults where $f(t) \in \mathbb{R}^g$ is a fault vector with each $f_i(t)$ $(i = 1, 2, 3, \ldots\ldots, g)$ corresponding to a specific fault. The matrices R_1 and R_2 are the fault entry matrices and they represent the effect of faults on the system. $u(t)$ and $y(t)$ are both known for fault detection and diagnosis purpose. They are the input vector to the system (measured actuation) and the measured output vector respectively. Equations 8 and 9 represent general model representation for faulty system in time domain and frequency domain respectively. The faulty system representations are widely accepted in the fault diagnosis literature [17, 20, 21, 52, 66, 67].

2.3.1.2. State Estimation Approach

State estimation is one of the several approaches employed in the residual generation for fault detection and diagnosis purposes [17, 67-69]. Output observer based residual generation is the most commonly used approach, and it is discussed here as a representative of the state estimation technique for residual generation.

2.3.1.2.1. Observer Based Residual Generation

Our interest here is outputs estimation using an observer to generate residual vectors. Output estimates are sufficient for this purpose, so the use of full state observer is not required. The approach estimates the outputs of the system through the measurements, or a subset of it using either Luenberger observer(s) in a deterministic setting or Kalman filter(s) in a stochastic setting. Then, the residual is the weighted output estimation error (or innovations in the stochastic case) [17, 66, 68, 70]. Figure 5 shows a residual generator via generalised Luenberger observer.

$$\begin{cases} \dot{z}(t) = Fz(t) + Ky(t) + Ju(t) \\ r(t) = L_1 z(t) + L_2 y(t) + L_3 u(t) \end{cases} \tag{11}$$

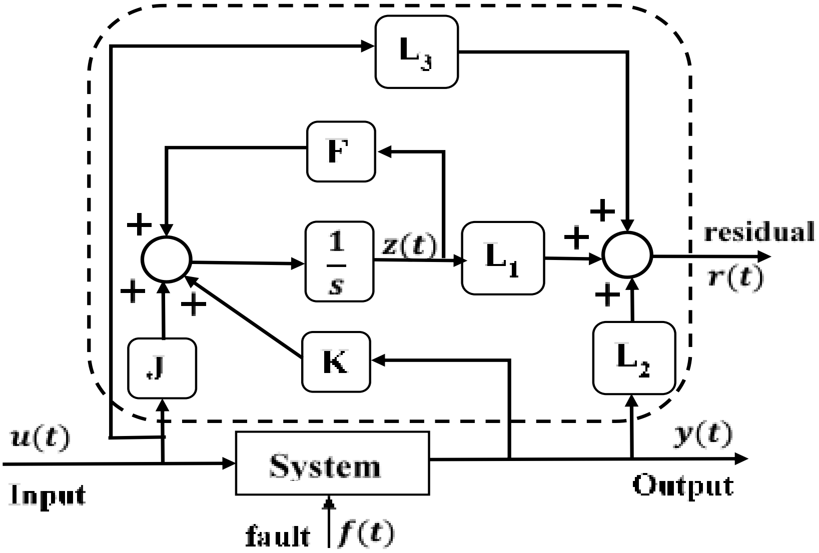


Figure 5. Generalized Luenberger observer residual generator.

The matrices in the equation 11 above should satisfy:

$$\begin{cases} F \text{ has stable eigenvalues} \\ TA - FT = KC \\ J = TB - KD \\ L_1 T + L_2 C = 0 \\ L_3 + L_2 D = 0 \end{cases} \tag{12}$$

When the residual generator, equation (11) is applied to the system, equation (8), the residual is:

$$\begin{cases} \dot{e}(t) = Fe(t) - TR_1 f(t) + KR_2 f(t) \\ \quad r(t) = L_1 e(t) + \; L_2 R_2 f(t) \end{cases} \tag{13}$$

where $e(t) = z(t) - Tx(t)$. It is obvious from the above expressions that the residual depends completely on faults. The other option is to use full order observer with $T = I$.

A single residual is sufficient to detect fault, but a set of residual vectors (structured residual set) or directional residual vector will be required to isolate faults with the observer-based approach. The design of a structured residual set for sensor faults is straightforward. For instance, if the output vector $y = (y_1, \ldots, y_m)$ is replaced with an output vector y $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)$ without the single sensor measurement y_i, the residual will be insensitive to the fault in the i_{th} sensor. However, for isolating an actuator fault, the design of a structured residual set is not as straightforward and can be achieved through the use of unknown input observers [17, 71] or eigenstructure assignment [69, 70]. A fixed residual vector can be designed through “fault detection filter” invented by Beard [72].

2.3.1.2.2. Unknown Input Observer

Mathematical description of any system under consideration is at the heart of model-based fault detection and diagnosis. The more accurately the model represents the system, the better the reliability and performance of the corresponding fault diagnostic scheme. Modelling errors and disturbances are inevitable in such mathematical representation. Hence, there is need to develop robust residual generator. Robust residual generation is the most significant task in model-based fault diagnosis techniques, and unknown input observer (UIO) belongs to such class of robust residual generator. It works on the principle of decoupling the state estimation error from the unknown inputs (disturbances). By so doing, the residual can also get de-coupled from each disturbance; the residual is defined as a weighted output estimation error [52, 70]. Though the unknown input vector is unknown, its distribution matrix is assumed known. The approach was originally proposed by Watanabe and Himmelblau [73], and the design problem of UIO dated back to 1975 [74]. Consider a dynamic system in which its uncertainty can be summarised as an additive unknown disturbance:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \\ \qquad y(t) = Cx(t) \end{cases} \tag{14}$$

Given the structure of a full-order observer described as:

$$\begin{cases} \dot{z}(t) = Fz(t) + TBu(t) + Ky(t) \\ \hat{x}(t) = z(t) + Hy(t) \end{cases} \tag{15}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $y(t) \in \mathbb{R}^m$ is the output vector, $u(t) \in \mathbb{R}^r$ is the known input vector, $d(t) \in \mathbb{R}^q$ is the unknown input (disturbance) vector, $\hat{x} \in \mathbb{R}^n$ is the estimated state vector, $z \in \mathbb{R}^n$ is the state of the full-order observer, A, B, C, E are known matrices with appropriate dimensions and F, T, K, H are matrices to be designed to achieve unknown input de-coupling and other design requirements. The observer described by equation 15 is shown in Figure 6.

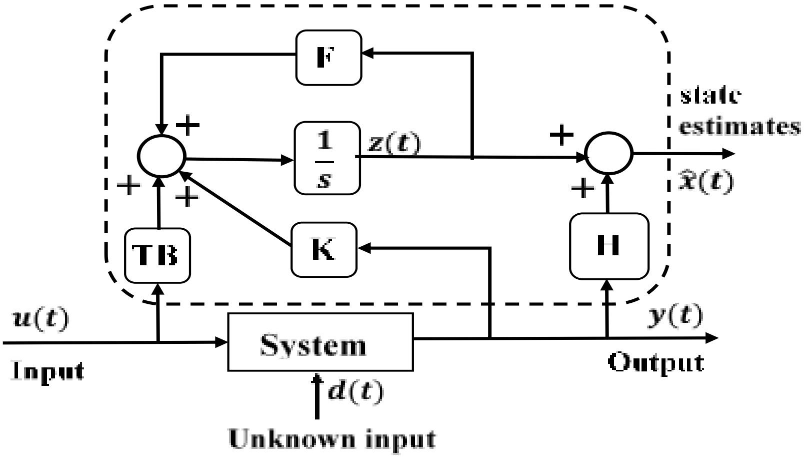


Figure 6. The structure of a full-order unknown input observer.

When the observer (15) is applied to the system (14), the state estimation error $(e(t) = x(t) - \hat{x}(t))$ is governed by the expression

$$\dot{e}(t) = (A - HCA - K_1C)e(t) + [F - (A - HCA - K_1C)]z(t) + [K_2 - (A - HCA - K_1C)H]y(t) + [T - (I - HC)]Bu(t) + (HC - I)Ed(t) \tag{16}$$

where,

$$K = K_1 + K_2 \tag{17}$$

Then, the state estimation error will be:

$$\dot{e}(t) = Fe(t) \tag{18}$$

If the following relations hold true:

$$\begin{cases} (HC - I)E = 0 \\ T = I - HC \\ F = A - HCA - K_1C \\ K_2 = FH \end{cases} \tag{19}$$

If all eigenvalues of F are negative, $e(t)$ will approach zero asymptotically, meaning $\hat{x} \to x$. It means that the observer (15) is an unknown input observer of the system (14). Hence, the design of UIO is to solve equations 17 and 19, and to make sure that all eigenvalues of the system matrix F are stable [52, 70]. The following algorithm summarizes the procedure and the necessary conditions required for the design of an unknown input observer.

Step 1. Check the rank condition for E and CE: If the $rank(CE) \neq rank(E)$, a UIO does not exist, go to step 10.

Step 2. Compute H, T and A_1:

$$H = E[(CE)^T CE]^{-1}(CE)^T; \quad T = I - HC; \quad A_1 = TA.$$

Step 3. Check the observability: If (C, A_1) observable, a UIO exist and K_1 can be computed using pole placement, then go to step 9.

Step 4. Construct a transformation matrix P for the observable canonical decomposition: To select independent $n_1 = rank(W_0)$ (W_0 is the observability matrix of (C, A_1)) row vector $p_1^T, \ldots, p_{n_1}^T$ from W_0, together with other $n - n_1$ row vector $p_{n_1+1}^T, \ldots, p_n^T$ to construct a non-singular matrix as:

$$P = [p_1, \ldots, p_{n_0};\ p_{n_0+1}, \ldots, p_n]^T$$

Step 5. Perform an observable canonical decomposition on (C, A_1):

$$PA_1P^{-1} = \begin{bmatrix} A_{11} & 0 \\ A_{12} & A_{22} \end{bmatrix} \qquad CP^{-1} = [C^* \ 0]$$

Step 6. Check the detectability of (C, A_1): If any one of the eigenvalues of A_{22} is unstable, a UIO does not exist and go to step 10.

Step 7. Select n_1 desirable eigenvalues and assign them to $A_{11} - K_p^1 C^*$ using pole placement.

Step 8. Compute $K_1 = P^{-1}K_p = P^{-1}[(K_p^1)^T\ (K_p^2)^T\ ]^T$, where K_p^2 can be any $(n - n_1) \times m$ matrix.

Step 9. Compute F and K: $F = A_1 - K_1C, \quad K = K_1 + K_2 = K_1 + FH.$

Step 10. STOP

2.3.2. Data-Based Fault Detection and Diagnosis

The difficulties faced in developing detailed first principle models for complex chemical processes with acceptable level of accuracy needed for fault monitoring and accommodation purposes limit the application of model-based FDD to well-understood systems like electro-mechanical systems. Data-based FDD, on the other hand, has been extensively used in the chemical industries for process monitoring and fault diagnosis because of its ability to provide reduced dimensional models for high dimensional processes. Its extensive usage also stems from its simplicity and ability to handle large amount of correlated process measurements. A large amount of process data collected from a system under normal and faulty conditions is required for the data-based FDD techniques. Using the classification of Zhang and Jiang [14], data-based FDD is further classified into quantitative and qualitative methods.

The quantitative data-based approaches extract features from the available process data through multivariate statistical and non-statistical means. Neural networks FDD approach is an example of the non-statistical method while principal component analysis (PCA), statistical pattern classifiers and partial least squares (PLS) are examples of the multivariate statistical methods. The qualitative data based FDD approaches, such as expert systems, fuzzy logic, pattern recognition, qualitative trend analysis and frequency and time frequency analysis, as presented in Figure 2 will not be discussed further in this chapter as our focus is on model-based and data-based FDD methods. Multivariate statistical approaches are powerful tools that are capable of compressing data to reduce its dimensionality and still retain as much variation as contained in the original data set for more straightforward analysis. The multivariate statistical techniques can efficiently handle noise and correlation in the original data during transformation into a much lower dimension. From successful industrial application point of view, multivariate statistical process monitoring techniques are the most widely used techniques for fault diagnostics owing to their fast abnormal events detection, ease of implementation and little effort required for their modelling with very little a priori process knowledge. However, they do not possess ‘fingerprint’ or ‘signature’ properties for diagnosis due to their limited process knowledge.

2.3.2.1. Principal Component Analysis

PCA is a standard multivariate statistical technique that has been used for various analyses stretching over a century. It was originally proposed by Pearson [75] and later developed by Hotelling [76]. Principal component analysis is based on orthogonal decomposition of the covariance matrix of the process variables along direction that explains the maximum variation of the data. Its main function is finding factors that have a much lower dimension than the original data set which accurately describes the major trend in the original data set.

Let X be a $n \times p$ matrix of the scaled measurements of n samples and p measured process variables with covariance matrix Σ. From matrix algebra, Σ may be reduced to a diagonal matrix L by a particular orthonormal $p \times p$ matrix U, i.e.,

$$\Sigma = ULU^T \tag{20}$$

where columns of U are the principal component loading vectors and the diagonal elements of L are the ordered eigenvalues of Σ which defines the amount of variance explained by the corresponding eigenvector. Then, the principal component transformation is given as:

$$T = XU \text{ or } t_i = Xu_i \tag{21}$$

where t_i and u_i are the i^{th} column of T and U respectively. Equivalently, X can be decomposed by PCA as:

$$X = TU^T = \sum_{i=1}^{p} t_i {u'}_i \tag{22}$$

The $n \times p$ matrix $T = (t_1, t_2, \ldots, t_p)$ contains the so-called principal component (PC) scores which are linear combinations of all the p variables. Typically, the first "a" principal components $(a < p)$ will capture the most variation in the original data if they are correlated and can be used to represent the majority of data variation. There are different criteria available for the selection of number of principal components "a." In this chapter however, we select "a" which account for between 75% and 90% variation in the original data set and examine the suitability of different values of "a" for the FDD purpose using appropriate data sets. Equation (22) can be written as

$$X = t_1{u'}_1 + t_2{u'}_2 + \ldots + t_a{u'}_a + E = \hat{X} + E \tag{23}$$

where E and $\hat{X}$ are the residual terms and the PCA model prediction of X respectively. With an in-control model established based on historical data collected during normal operation, process monitoring is achieved by using the Hotelling's T^2 and squared prediction error (SPE) monitoring statistics of the nominal model given below to detect fault from new measurements.

$$T_i^2 = \sum_{j=1}^{a} \frac{t_{i,j}^2}{\lambda_j} \tag{24}$$

where T_i^2 is the Hotelling's T^2 value for sample i, $t_{i,j}$ is the i^{th} element of principal component j, λ_j is the eigenvalue corresponding to principal component j and a is the number of principal components retained. SPE is simply the sum of squares of the difference between the original scaled data and their estimates $(\hat{X})$ from the PCA model. When the process is in normal operation, both SPE and T^2 monitoring statistics should be small and within their control limits. However, when a fault appears in the monitored process, the fault will cause some variables to have larger than normal magnitudes (large T^2 value) and change the variable correlations leading to large SPE values. The T^2 index indicates nonconformity with the expected behaviour of the process as captured by the diagnostic model while the SPE index presents deviations that result from events not described in the diagnostic model [25]. The fault then causes the monitoring statistics to violate their respective limits (thresholds) for some specified periods, before a fault is eventually declared. The control limits for SPE and T^2 are given by (25) and (26) respectively.

$$\begin{cases} SPE_{lim} = \theta_1 \left[\frac{c_\alpha h_0 \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{\frac{1}{h_0}} \\ \theta_i = \sum_{j=a+1}^{p} \lambda_j^i \\ h_0 = 1 - \frac{2\theta_1\theta_2}{3\theta_2} \end{cases} \tag{25}$$

$$T_{lim}^2 = \frac{a(n-1)}{(n-1)} F_{a,n-a;\,\alpha} \tag{26}$$

In (25) and (26), c_α is the value for normal distribution at $100(1-\alpha)\%$ confidence level and $F_{a,n-a;\,\alpha}$ is the F distribution with degrees of freedom, a and n-a, and confidence level α. Upon declaration of a fault, variable contribution plots are obtained for the SPE and the Hotelling's T^2 for further fault diagnosis to identify the component that has developed fault. This is done with some good understanding of the monitored process. The Hotelling's T^2 variable contribution plot can be obtained using (27) [77].

$$cont\_x_{k_{f,j}} = \frac{t_{f,j}}{\lambda_j}\, u_{k,j}\, x_{f,k} \tag{27}$$

where $cont\_x_{k_{f,j}}$ is the contribution of variable x_k to score vector t_j at point f (point of fault declaration); $t_{f,j}$, λ_j and $u_{k,j}$ are the score vector t_j, the corresponding eigenvalue and loading vector for k^{th} variable respectively at the faulty sample f, while $x_{f,k}$ is variable x_k also at point f. The SPE contribution plots can be easily obtained by taking the contributions of each variable to the large SPE value at the point of fault declaration.

2.3.2.2. Dynamic PCA

Dynamic PCA is a variant of PCA technique that incorporates time-lagged measurements in its model to capture the dynamic correlation behaviour of the system for effective fault propagation analysis. The technique is the same as PCA with the only difference being the increased dimension of the process variable p by a factor of l (the number of time lags considered) to give $(l+1)p$ process variables. In essence, this leads to an increase in the columns of X to a new dimension $n \times (l+1)p$ resulting in orthonormal matrix U in (20) having dimension $(l+1)p \times (l+1)p$. Consider an $(n \times p)$ process variable matrix X, the augmented matrix X for DPCA at any time instant t will be:

$$X = [X\,X(t-1)\ \dots\ X(t-l)] \tag{28}$$

If for instance, $p = 3$ and $l = 1$, we have

$$X(t) = [x_1(t) \quad x_2(t) \quad x_3(t) \quad x_1(t-1) \quad x_2(t-1) \quad x_3(t-1)] \tag{29}$$

where $x_i(t)$ and $x_i(t-1)$ are the process variables at time t and at $l = 1$. The procedure for the determination of the number of time lag (l) can be found in Ku et al. [78].

2.3.2.3. Projection to Latent Structure

Projection to latent structure (PLS) originated from the pioneering work of Wold [79] between the mid-1960s and early 1980s and was further developed by Wold and co-workers [80, 81]. PLS, like PCA, conceptually reduces the dimension of correlated process data by projecting them down onto a lower dimensional latent variable space. PLS however, works with additional data matrix Y, process quality variables together with the process variable X. PLS models the relationship between the two sets of data while compressing them simultaneously. It extracts the latent variables that explain the variation in process data X, at the same time the variation in X that is most predictive of the quality data Y. The first PLS latent variable is the linear combination of the process variables that maximises the covariance between them and the quality variable [9]. PLS defines the high dimensional process variables (regressor) and process quality variables (response) (X and Y) in terms of a small number of latent variables (T) that defines the major directions of variation in the process data [25]. The basic model is defined as:

$$X = TU^T + E \tag{30}$$

$$Y = TC^T + F \tag{31}$$

where X and Y are $(n \times p)$ and $(n \times m)$ matrices of observed values, $T = XW^*$ is a $(n \times a)$ matrix of latent variable scores $(a \ll p)$, U, C and W^* are matrices of loading estimated from the data, n is the number of observations, and p and m are the numbers of regressor and response variables respectively. The concept of dependent and independent variables has little place in latent variable model. E and F are errors associated with X and Y respectively. The choice of process variables and process quality variables are user defined [25].

2.4. Fault Tolerant Controllers

This section discusses FTC as the other component of FTCS. Fault tolerant controllers belong to the class of smart or intelligent controllers with built-in diagnostics. They are capable of tolerating failures or malfunctions in system components, actuators and sensors and still deliver satisfactory performance despite those failures. The main task in FTC is to design a controller with suitable structure to achieve stability and satisfactory performance, whether or not all the system components including the control system itself are functioning correctly. An extensive number of researches has been carried out on FTC since the early 1980s [15, 16, 26-28, 33-37, 82]. This was motivated by the need to give aircraft control system much needed control capabilities to accommodate faults within the system and still be able to land the aircraft safely. Interest in the design and application of FTC grew in the other industries due to the increased safety and reliability demand beyond what conventional controllers offer. These industries include aerospace, nuclear power plants, automotive, manufacturing and chemical and process industries [14, 34].

Several techniques have been used in the design of fault tolerant controllers. Zhang and Jiang [14] gave a detailed classification of such techniques. They used criteria such as mathematical design tools, design approaches, reconfiguration mechanism, and the type of systems investigated. It is not surprising that most of the techniques that have been researched in FTC are concentrated in the aerospace and aviation industry due to its historical reasons. Some impressive results on the design and application of FTC have been published lately: application of distributed model predictive control (DMPC) to accommodate actuator faults in a three unit continuous stirred tank reactor [30-32]; the use of adaptive controller for FTC in General Electric XTE46 engine [38]; combined model predictive control (MPC) and H_∞ robust controller [40] and the use of proactive fault tolerant Lyapunov-based MPC [39] rather than reactive FTC that have been the norm over the last two decades or so. Many of the techniques employed in FTC rely on ideas that had been investigated in the past for other control purposes. Though well-known control design techniques are used, they face new challenges and problems that may not appear in the conventional controller design [14]. It is essential that such control methods deliver some good level of performance in the impair system in an online real time manner. Owing to the demand and performance requirement of an FTC, it is not unusual for an FTC to have a combination of different control structures and control design algorithm. This chapter focuses on simple restructurable active FTC that uses backup feedback signal design approaches for actuator fault accommodation.

2.4.1. Fault Tolerant Model Predictive Control (FTMPC)

Model predictive control is a high performing model-based process control strategy with ability to handle multivariable interactions, constraints on control inputs and system states, and optimisation requirements in a systematic manner. It is popular in the process control industry because the actual operating objectives and operating constraint can be represented explicitly in the optimisation problem solved at every control instant [83]. Several researchers have worked and continue to work on FTMPC with interest in the area growing daily. Mhaskar [84] designed a robust model predictive controller to achieve fault-tolerant control of nonlinear systems subject to uncertainties, constraints and actuator fault. He used Lyapunov-based approach to formulate constraints that account for uncertainty explicitly in the predictive control law and also explicitly to allow the characterisation of initial conditions starting from where closed-loop stability is guaranteed. Zhang et al. [85] used state space model predictive fault-tolerant control to accommodate partial actuator faults in batch processes with unknown disturbances. They propose an improved cost index that can aid selection of relevant weighting factors for better control performance. Tao et al. [86] applied state space model predictive control to accommodate partial actuator fault in linear systems. Lao et al. [39] proposed proactive Lyapunov-based fault-tolerant model predictive control to handle effectively incipient actuator fault in chemical processes.

Mirzaee and Salahshoor [40] presented a unified robust fault tolerant control framework to effectively handle changes in unmeasured disturbance and model parameters, biases and drifts in sensors and actuators respectively. This was achieved using adaptive unscented Kalman filters (AUKFs) and fuzzy-based decision making (FDM) algorithm for fault detection and isolation, and actuator and sensor faults diagnostics respectively. The AUKF and FDM schemes were integrated with H∞ optimal robust controller and MPC using a fuzzy switch scheme for switching between MPC and robust controller for effective performance in actuator and sensor faults accommodation.

Generally, the design of an MPC has three main components:

- The model of the system under consideration. This is used essentially for the system open-loop future trajectory prediction and in large part plays a crucial role in the effectiveness or otherwise of the MPC.
- A control objective function to be minimised subject to constraints imposed by the system model, restrictions on control inputs, system states and others.
- A receding horizon scheme that introduces feedback into the control law for disturbances and model-mismatch compensation.

Consider the state space model of a system as given below:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), x(0) = x_0 \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{32}$$

where x, u and y are the state variables, inputs and outputs of the system respectively, A, B, C and D are matrices of appropriate dimensions. A brief description of a typical MPC formulation is given as [87]:

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} \left[\|\tilde{x}_i(\tau)\|_{Q_{c1}}^2 + \|u_i(\tau)\|_{R_{c1}}^2 \right] d\tau \tag{33a}$$

$$s.t. \dot{\tilde{x}} = f(\tilde{x}, u(t)) \tag{33b}$$

$$u_i(t) \in U_i \tag{33c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{33d}$$

where $S(\Delta)$, N and $\tilde{x}$ denote the family of piece-wise constant functions with sampling interval Δ, the prediction horizon and the predicted trajectories of the nominal system in (32) respectively, Q_{c1} and R_{c1} are positive definite symmetric weighting matrices. The objective function in (33a) is to be minimised subject to constraint (33b) which is supposed to have zero uncertainties in model (32) used to predict future trajectories of the system. Constraints (33c) and (33d) take into account the restrictions on the control inputs and the measured system states respectively. The first step of the optimal solution defined by (33), denoted as $u_i^*(t|t_k)$ is implemented and the whole procedure is repeated continuously.

2.4.2. Distributed Model Predictive Control

MPC typically works in a centralised fashion, but when dealing with complex systems, as we have in the chemical and oil and gas industry, for optimality, it may be better to have distributed control schemes where local control inputs are computed using local measurements and reduced-order of the sub-system dynamics. DMPC are used to coordinate the implementation of separate MPC controllers to achieve optimal input trajectories in a distributed manner. It is a developing research area with interest from both the academia and the industry. A review of DMPC by Christofides et al. [88] gave algorithmic details of the different approaches that have been used in the design and implementation of DMPC to provoke further researches in the area. Rawlings and Stewart [89] presented cooperative DMPC to guarantee nominal stability and performance properties with high degree of communication between local controllers by using MPCs with modified objective functions. Mercangöz and Doyle III [90] proposed a DMPC algorithm based on the work of Mutambara [91] and implemented it for level control on an experimental four-tank system. Chilin et al. [31] demonstrated the application of DMPC for actuator faults and their work is briefly outlined below. Consider a nonlinear system described by a state-space model:

$$\dot{x}(t) = f(x(t), u_1(t), u_2(t), d(t)) \tag{34}$$

where $x(t) \in R^{n_x}$ denotes state variables vector, $d \in R^p$ is the model of the set of possible faults, $u_1(t) \in R^{n_{u1}}$ and $u_2(t) \in R^{n_{u2}}$ are the two different sets of possible manipulated inputs. The faults are unknown and d_j, $j = 1 \dots p$, can take any value. The system is controlled by two sets of control input u_1 and u_2 (i.e., $u(t) = u_1(t) + u_2(t)$). They assumed a Lyapunov-based controller $u_1(t) = h(x)$ exists, which renders the origin of the fault-free closed-looped system asymptotically stable with $u_2(t) = 0$.

Then, they designed a DMPC structure (see Figure 7) to achieve closed-loop stability and performance using two Lyapunov-based MPC, LMPC2 and LMPC1 to compute control input trajectories u_2 and u_1 respectively [11, 30]. Consider the expressions for LMPC2 (equation 35a – 35e) and LMPC1 (equation 36a – 36d) below:

$$\min_{u_{d2} \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau) Q_c \tilde{x}(\tau) + u_{d1}^T(\tau) R_{c1} u_{d1}(\tau) + u_{d2}^T(\tau) R_{c2} u_{d2}(\tau)] d\tau \tag{35a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}(\tau), 0) \tag{35b}$$

$$u_{d1}(\tau) = h(\tilde{x}(j\Delta)), \forall\, \tau \in [j\Delta, (j+1)\Delta), j = 0 \dots N-1 \tag{35c}$$

$$\tilde{x}(0) = x(t_k) \tag{35d}$$

$$\frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), u_{d2}(0), 0) \leq \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), 0{,}0) \tag{35e}$$

and:

$$\min_{u_{d1}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{d1}^T(\tau)R_{c1}u_{d1}(\tau) + u_{d2}^{*T}(\tau|t_k)R_{c2}u_{d2}^*(\tau|t_k)]d\tau \tag{36a}$$

$$\dot{\tilde{x}} = f(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}^*(\tau|t_k), 0) \tag{36b}$$

$$\tilde{x}(0) = x(t_k) \tag{36c}$$

$$\frac{\partial V(x)}{\partial x} f(x(t_k), u_{d1}(0), u_{d2}^*(0|t_k), 0) \leq \\ \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), u_{d2}^*(0|t_k), 0) \tag{36d}$$

where V is the Lyapunov function, $\tilde{x}$ is the predicted trajectory for the fault-free system with u_2 being the input trajectory computed by the LMPC2 and u_1 being the Lyapunov-based controller $h(x)$ applied in a sample and hold fashion. The DMPC is implemented thus:

1) Both LMPC1 and LMPC2 receive the state measurement $x(t_k)$ from the sensor at each sampling instant t_k.
2) LMPC2 evaluates the optimal input trajectory of u_2 based on $x(t_k)$ and sends the first step input value to its corresponding actuators and the entire optimal input trajectory to LMPC1.
3) After receiving the entire input trajectory of u_2 together with $x(t_k)$, LMPC1 evaluates the future input trajectory of u_1.
4) LMPC1 then sends the first step input value of u_1 to its corresponding actuators.

$u_{d2}^*(\tau|t_k)$ and $u_{d1}^*(\tau|t_k)$ are the optimal solutions to the optimisation problems of LMPC2 and LMPC1 respectively. Hence, the manipulated inputs to the system are:

$$\begin{cases} u_1(t|x(t_k)) = u_{d1}^*(t - t_k|t_k), \forall\ t \in [t_k, t_{k+1}) \\ u_2(t|x(t_k)) = u_{d2}^*(t - t_k|t_k), \forall\ t \in [t_k, t_{k+1}) \end{cases} \tag{37}$$

A non-zero residual is generated when fault occurs in a system. The residual is generated through this expression ($\hat{x}$ and x are the filter state for the fault-free system and the measured state respectively):

$$r(t) = |\hat{x}(t) - x(t)| \tag{38}$$

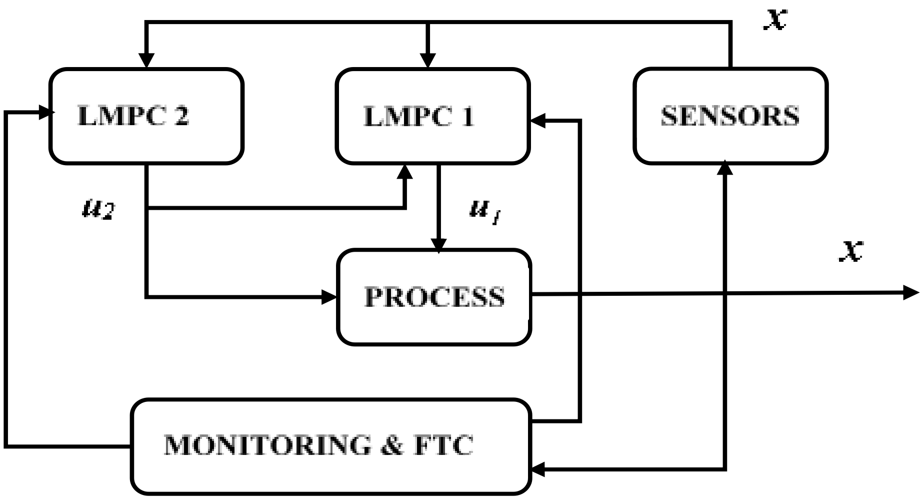


Figure 7. DMPC structure.

The main idea behind this approach is that there is an extra control input u_2 that can be called upon to stabilize the system in the presence of a fault. It is suspected that the effectiveness of the approach will depend on the type and severity of the fault.

3. Actuator Fault Tolerant Controllers

Most of the current fault-tolerant controllers involve techniques with high level of complexity and computational tasks in their design and implementation. The proposed fault-tolerant control technique offers a simplified approach to the design and implementation of controllers capable of tolerating actuator faults in complex systems with acceptable graceful performance degradation. The approach is expected to achieve results comparable to those employing complex computational tasks, though different possible control structures would have to be analysed a priori using tools like relative gain array (RGA) and dynamic relative gain array analysis to select possible switching options. As it is often the case that, for any given process, there are several ways of controlling it, some better than others, so selecting a sub-optimal strategy under faulty condition would be far more acceptable than process shut-down. However, the switchability and restructurability of a fault tolerant controller are process dependent as maintaining acceptable level of performance in some processes may not always be achievable due to lack of suitable controlled and manipulated variable pairing. This must be carefully assessed taking into consideration the remaining healthy actuators and the process variables pairing for control purposes.

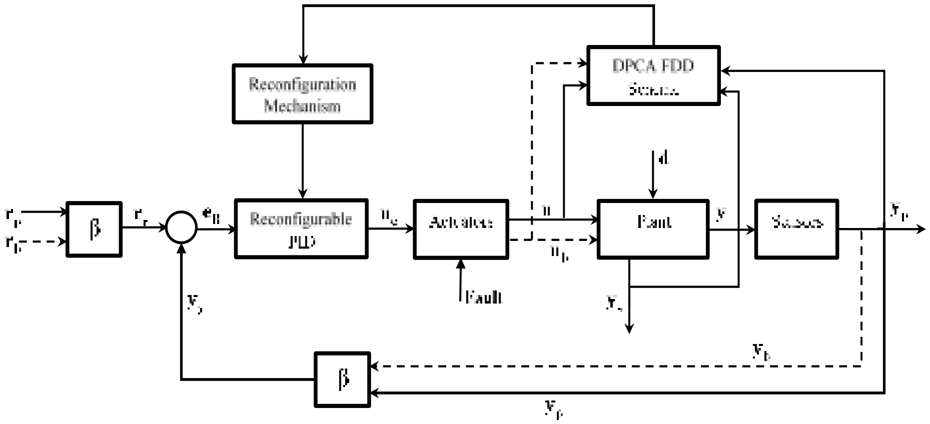


Figure 8. The proposed FTC for actuator faults.

Figure 8 presents the structure of the proposed Actuator FTC for a complex chemical process. It has additional blocks like DPCA FDD scheme, reconfiguration mechanism, two weighting matrices blocks and a reconfigurable PID controller block instead of an ordinary PID controller block. In this figure, *e*, u_c, *u*, *d*, *y*, y_p and y_s are vectors of control errors, controller outputs, manipulated variables, disturbances, actual process outputs, measured primary controlled outputs, and measured uncontrolled secondary variables respectively; while r_b, r_r, u_b, y_b and y_y are vectors of appropriate dimensions for back-up set point signals, switchable references, back-up manipulated variables, controlled variables back-up signals and restructurable controlled outputs respectively. r_p is the reference points for the primary controlled variables. The DPCA FDD scheme of the system deals with process monitoring for timely and accurate detection and diagnosis of actuator faults. The reconfiguration mechanism acts on the fault information received from the FDD scheme. It contains several possible controller switching options designed a priori based on rigorous analysis of a closed set of possible actuator faults using RGA and DGRA, including stability analysis of the entire system. The reconfigurable PID controller implements the selected reconfigurable option by reconfiguring its control structure after isolating the faulty actuator using the back-up signals for reference points and the primary controlled outputs. This is made possible with the use of the weighting matrices blocks for seamless implementation. Procedures involved in some of the major components of the FTCS are presented next.

3.1. DPCA FDD Scheme

Dynamic PCA monitoring technique is used in the integrated actuator FDD scheme to identify possible actuator faults occurrence. In order to avoid repetition, the DPCA procedures earlier presented will be augmented further to highlight how the FDD scheme functions in the whole FTCS system. The matrix of the scaled measurement "X" in equation 28 is given as:

$$X = [u \quad y_p \quad y_s] \tag{39}$$

where X is the matrix of past measurements of all the process variables to be included in the DPCA diagnostic model with dimension n×$(l$+1$)p$, u is an n×$(l$+1$)np$ matrix of manipulated variables, y_p is an n×$(l$+1$)np$ matrix of primary controlled variables, y_s is an n×$(l$+1$)ns$ matrix of measured secondary variables and p (p=2np+ns) is the total number of variables included in the monitoring diagnostic model during normal operation. *n, np, ns* and *l* are the total number of samples, total number of primary controlled variables, number of measured secondary variables and the time lag considered respectively. It is assumed that the manipulated variable 'u' is always available, otherwise it can be obtained from the knowledge of the controller output 'u_c'. The first phase of the FDD scheme is the development of an actuator fault detection scheme. The scheme is then used to monitor the process for possible actuator faults using the computed control limits for the Hotelling's T^2 and the SPE monitoring statistics presented in equations 25 and 26. The second phase of the FDD scheme involves fault diagnosis to identify the faulty actuator using contribution plots of the monitoring statistics.

Contribution plots are simply graphical representations depicting the contributions of each variable in the diagnostic model to the values of the Hotelling's T^2 and SPE monitoring statistics, particularly upon detection of a fault. In this chapter, excess contributions of each variable are used by first computing their total contributions to the monitoring statistics at the point of fault declaration and the following two consecutive sampling periods for proper diagnosis. Average contributions of each variable to the monitoring statistics during normal operation are also obtained and subtracted from the total contributions earlier computed to obtain the variable excess contribution. The variables that contribute the most to the faulty situation are then mapped to a particular actuator fault based on the knowledge of the system. Hotelling's T^2 variable contributions to a faulty actuator are obtained using the following equations [77]. Let r be the number of score vectors that violate their limits ($r \leq a$).

$$cont\_x_{k_{f,j}} = \frac{t_{f,j}}{\lambda_j}\, u_{k,j}\, x_{f,k} \tag{40}$$

where $cont\_x_{k_{f,j}}$ is the contribution of variable x_k to score vector t_j at point f (point of fault declaration and the following two consecutive sampling period), a is the number of principal components, $t_{f,j}$, λ_j and $u_{k,j}$ are the score vector t_j, the corresponding eigenvalue and loading vector for k^{th} variable respectively at the faulty sample f, while

$x_{f,k}$ is variable x_k also at point f. The value of $cont\_x_{k_{f,j}}$ represents $\frac{t_{f,j}{}^2}{\lambda_j}$ which should always be positive, and it is set equal to zero if negative. The total contribution of variable x_k to the detected fault is given as:

$$Cont\_x_k = \sum_{f=1}^{r} \left(cont\_x_{k_{f,j}}\right) \tag{41}$$

Average variable contributions to the monitoring statistics during normal operation at any instance is given as:

$$avg\_x_{k_j} = \frac{t_j}{\lambda_j}\, u_{k,j}\, x_k \tag{42}$$

$avg\_x_{k_j}$ should always be positive and is set to zero if negative. The overall average contributions of each variable to Hotelling's T^2 monitoring statistics pre-fault era is given as:

$$x_{k\_avg} = \frac{\sum_{i=1}^{n}\left(avg\_x_{k_j}\right)}{n} \tag{43}$$

Subtracting equation 43 from equation 41 gives the excess contributions of each variable to the out of control situation, which are then plotted to identify the variables indicative of the fault and then mapped unto a particular actuator fault. After successful detection and diagnosis of an actuator fault and subsequent implementation of the FTC, for continued process monitoring, *np* is reduced by 1 and *p* by 2. The dimension of *X* post-fault era now reduces to *n*×(*l*+1)(*p*-2). This reflects the isolation of the faulty actuator and subsequent removal of a controlled variable for further system monitoring. Output of the FDD monitoring scheme is passed on to the reconfiguration mechanism to reconfigure the input-output pairing for the whole system as appropriate.

3.2. Control Strategies and Loop Pairing Assessment

It is imperative that rigorous process interaction of the multivariable system is undertaken, in order to have good understanding of the effect of variable pairing reconfiguration on the stability of the system, particularly during faults accommodation. Different control strategies during normal operation and faulty conditions are investigated to determine the optimum and sub-optimal controlled variable-manipulated variable pairing for every potential actuator fault in the system. The task involved is non-trivial and it is achieved through the use of RGA and DRGA.

3.2.1. Relative Gain Array

Relative gain array, developed by Bristol [92] and extended by McAvoy [93] and Shinskey [94] is used for the control loop interaction analysis. A brief description of the procedures involved in the analysis is given in this section. RGA gives a quantitative measure of the level of interaction amongst the loops of a multivariable control structure using the system process gains matrix, which defines the steady state open-loop relationship between the inputs and outputs. Let the relationship between outputs and inputs of a multivariable system be presented as below:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ \vdots \\ y_p(s) \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1p} \\ k_{21} & k_{22} & \cdots & k_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ k_{p1} & k_{p2} & \cdots & k_{pp} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_p(s) \end{bmatrix} \tag{44}$$

Equation 44 can be presented in a compact form as:

$$y(s) = K.u(s) \tag{45}$$

where $y(s)$, $u(s)$ and K are controlled outputs, manipulated inputs and the steady state process gain matrix respectively. K can be obtained by independently varying the manipulated inputs of the multivariable system one at a time and then allowing the system to reach a new steady state. Several changes can be made to individual manipulated variable over a reasonably long period of time during the process simulation to gather enough data, which can then be used to obtain a more accurate K matrix and dynamic models, in this case, transfer function models of the system using System Identification Toolbox in MATLAB. The RGA (Λ) of the system can then be obtained using:

$$\Lambda = K.^{*}\, (K^T)^{-1} \tag{46}$$

where $.^{*}$ represents element by element multiplication.

Several Λ for different sets of K matrices will have to be analysed for each possible actuator fault and implemented on the system to assess the stability of the system under various degrees of actuator faults. The RGA analysis could involve several hundreds of different inputs-outputs pairing for all the possible actuator faults, particularly for complex system. This will help to determine an optimum/sub-optimal inputs-outputs pairing during controller reconfiguration in any faulty situation.

3.2.2. Dynamic Relative Gain Array (DRGA)

RGA has some limitations as it does not consider the transient behaviour and effect of presence of disturbances in the system. DRGA is used in conjunction with RGA for a more robust loop pairing and stability analysis. DRGA was first introduced by Witcher and McAvoy [95] and later by Bristol [96] to address the perceived limitations of RGA by using the transfer function models of the system instead of the traditional steady state process gains. It can give more accurately the extent of interactions that is present amongst different loop pairing and more insight into the stability of the system, especially during controller reconfiguration. The denominator of the transfer function models provides an opportunity to evaluate the magnitude of the elements of relative gain at several frequencies by setting $s=j\omega$.

3.3. Reconfigurable PID Controllers

As it is often the case, for any given process, there are several possible sub-optimal control structures (input-output pairing) for the system, some more effective than others. The simple reconfigurable PID controller proposed here leverages on the opportunity of having more than one manipulated variable that can be used to control an output. Several possible control structures will have to be assessed a priori as explained in the last section using RGA and DRGA, and then stored for possible implementation in the event of an actuator fault being identified. Let the control error generated by a conventional feedback control law be:

$$e = r - y_p \tag{47}$$

and the control error with back-up feedback signal for an actuator fault in Figure 8 be

$$e_R = r_r - y_y \tag{48}$$

where

$$r_r = \beta\left[{r_p}^T \quad {r_b}^T\right]^T \text{ and } y_y = \beta\left[{y_p}^T \quad {y_b}^T\right]^T \tag{49}$$

β is a weighting matrix block given as:

$$\beta = diag(\beta_p \quad \beta_b) = \begin{bmatrix} \beta_p & 0 \\ 0 & \beta_b \end{bmatrix} \tag{50}$$

During normal operation, β_p and β_b are identity and zero square weighting matrices with dimension (*np*, *np*) for the primary controlled variables and back-up feedback signals respectively. The weighting matrices are used to deactivate and activate actual and backup feedback signal as appropriate during fault-tolerant controller reconfiguration. Substituting equation 49 and equation 50 into equation 48 gives

$$e_R = \begin{bmatrix} \beta_p & 0 \\ 0 & \beta_b \end{bmatrix} \begin{bmatrix} {r_p}^T & {r_b}^T \end{bmatrix}^T - \begin{bmatrix} \beta_p & 0 \\ 0 & \beta_b \end{bmatrix} \begin{bmatrix} {y_p}^T & {y_b}^T \end{bmatrix}^T \quad (51)$$

Let the reconfigurable PID controller be

$$G_R = [{G_c}^T \quad {G_b}^T]^T \quad (52)$$

where G_c and G_b are the actual controllers used during normal process operation and the pre-assessed backup feedback controllers to accommodate possible actuator fault occurrence respectively. Weighting matrix β is also introduced in equation 52 in order to implement the reconfigurable controller, which now becomes

$$G_{RC} = \beta G_R \quad (53)$$

The control law for the reconfigurable fault tolerant PID controller is then given as

$$u = G_{RC} e_R \quad (54)$$

The different possible manipulated and controlled variable pairing are assessed a priori to decide on the reconfiguration pairing upon detection and identification of a fault. Hence, accommodation of any individual fault is dependent on having a suitable healthy actuator that can provide satisfactory performance in the impaired system. Only a single fault-tolerant control system is considered in this chapter. However, the approach can also be applied to duplex FTCS structure. By single and duplex FTCS, we mean a single and double fault-tolerant control system backup for each pre-assessed actuator fault provided there are suitable restructurable manipulated and controlled variable pairings.

4. Implementation on Distillation Processes

The distillation column is among the most common and energy intensive units in any refinery operation. It is fundamental to the chemical and process industries, which is why its dynamics and control has been studied extensively. Implementation of the actuator faults tolerant control system on distillation processes with varying degrees of complexities under normal operation and faulty circumstances is presented in this section. This demonstrates the flexibility of the approach under various actuator faults. The developed FTCS for actuator fault is first implemented on the Shell heavy oil fractionator with three primary control loops and four measured secondary variables [97], and then on a crude distillation unit with several interactive primary control loops and numerous indirectly controlled secondary variables.

4.1. Application to the Shell Heavy Oil Fractionator

The proposed actuator FTC scheme is applied to the Shell heavy oil fractionator in this section. Figure 9 presents the schematic diagram of the system with interactions amongst its control loops.

The Shell heavy oil fractionator benchmark used here was developed by Shell Company as a test bed for the assessment of new control theories and technologies in 1986 [98, 99]. It is a highly constrained multivariable process with large dead times and very strong interactions amongst its control loops. The original system is slightly modified in this study by relaxing some of its constraints for the purpose of actuator faults accommodation. The heavy oil fractionator has five inputs and seven outputs, and it provides a realistic test bed for control related studies. The process was modelled using a first-order plus dead time transfer function matrix. Three out of the 5 inputs (top draw – u_1, side draw – u_2 and bottom reflux duty – u_3) into the system are

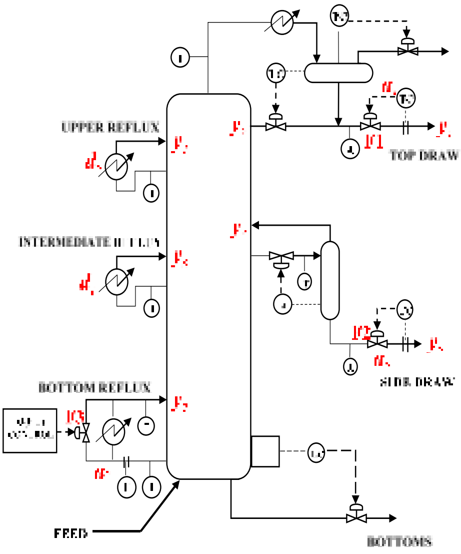


Figure 9. Shell heavy oil fractionator.

Table 1. Variables for the heavy oil fractionator

Variable	**Output variables**
Variable 1	Top end point (y_1)
Variable 2	Side end point (y_2)
Variable 3	Top temperature (y_3)
Variable 4	Upper reflux temp. (y_4)
Variable 5	Side draw temp. (y_5)
Variable 6	Internediate Reflux temp. (y_6)
Variable 7	Bottom reflux temp. (y_7)
	Input variables
Variable 8	Top draw (u_1)
Variable 9	Side draw (u_2)
Variable 10	Bottom reflux duty (u_3)
	Disturbance variables
	Intr. Reflux duty (d_1)
	Upper reflux duty (d_2)

used as manipulated variables, directly maintaining 3 process outputs (top end point – y_1, side draw end point – y_2 and bottom reflux temperature – y_7) at their set points while the remaining 2 inputs – intermediate reflux duty (d_1) and upper reflux duty (d_2) serve as unmeasured disturbances into the system. The other four outputs are not controlled. Table 1 gives the full listing of all the system variables. The manipulated variables are subject to saturation (±0.5) and rate limit (±0.05 per sample time) actuator hard constraints, which introduce non-linearity into the system. The disturbances are bounded within absolute values not more than 0.5. The complete model of the system is given in Table 2 while Figure 10 presents the system with different back-up feedback signals (indicated by dashed lines) for possible implementation of actuator fault tolerant controller. The system is controlled using three reconfigurable PI controllers with integral anti-windup.

Table 2. Shell Heavy oil fractionator transfer function model parameters

	Top draw			Side draw			Bottom reflux			Intr. Reflux duty			Upper reflux duty		
	(u_1)			(u_2)			duty (u_3)			(d_1)			(d_2)		
	K	τ	θ	K	τ	θ	K	τ	θ	K	τ	θ	K	τ	θ
Top end point (y_1)	4.05	50	27	1.77	60	28	5.88	50	27	1.20	45	27	1.44	40	27
Side end point (y_2)	5.39	50	18	5.72	60	14	6.90	40	15	1.52	25	15	1.83	20	15
Top temperature (y_3)	3.66	9	2	1.65	30	20	5.53	40	2	1.16	11	0	1.27	6	0
Upper reflux temp. (y_4)	5.92	12	11	2.54	27	12	8.10	20	2	1.73	5	0	1.79	19	0
Side draw temp. (y_5)	4.13	8	5	2.38	19	7	6.23	10	2	1.31	2	0	1.26	22	0
Inter. Reflux temp. (y_6)	4.06	13	8	4.18	33	4	6.53	9	1	1.19	19	0	1.17	24	0
Bottom reflux temp. (y_7)	4.38	33	20	4.42	44	22	7.20	19	0	1.14	27	0	1.26	32	0

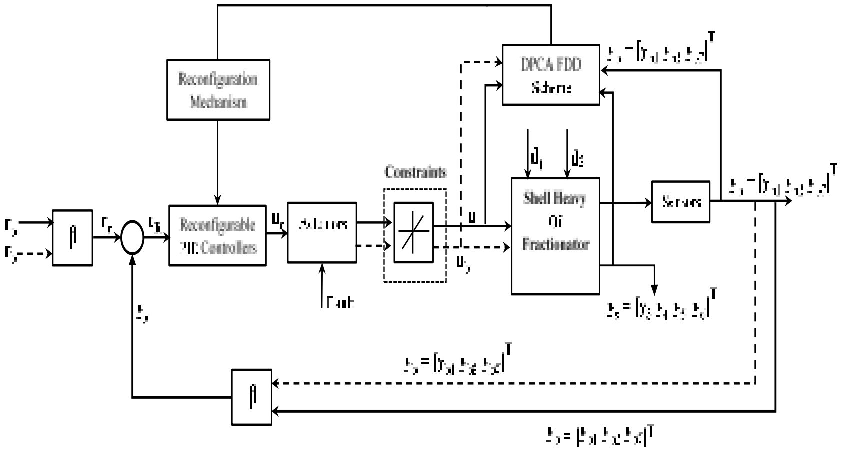

Reconfiguration Mechanism
DPCA FDD Scheme
Reconfigurable PID Controllers
Actuators
Constraints
Shell Heavy Oil Fractionator
Sensors
Fault
u
T

Figure 10. Schematic of the Shell heavy oil fractionator integrated with FTCS.

4.1.1. Process Description and Control Loop Pairing

The input-output selection for the control configuration was achieved after careful analysis of the system coupled with the use of RGA analysis as earlier presented. The transfer function matrix of the system given in equation 55 is used to obtain the steady state RGA for the system as shown in equation 56. Based on the RGA values, the manipulated variables u_1, u_2 and u_3 are used to control y_1, y_2 and y_7 respectively under normal operating conditions, producing a 3×3 control configuration. Possible controller reconfigurations are pre-assessed using the RGA tool for the input-output pairings under different faulty conditions.

$$G(s) = \begin{bmatrix} \frac{4.05e^{-27}}{50s+1} & \frac{1.77e^{-28}}{60s+1} & \frac{5.88e^{-27}}{50s+1} \\ \frac{5.39e^{-18}}{50s+1} & \frac{5.72e^{-14}}{60s+1} & \frac{6.90e^{-15}}{40s+1} \\ \frac{4.38e^{-20}}{33s+1} & \frac{4.42e^{-22}}{44s+1} & \frac{7.20}{19s+1} \end{bmatrix} \tag{55}$$

$$\Lambda = \begin{bmatrix} 2.0757 & -0.7289 & -0.3468 \\ 3.4242 & 0.9348 & -3.3585 \\ -4.4999 & 0.7946 & 4.7053 \end{bmatrix} \tag{56}$$

The input-output pairing for controller reconfiguration of the three actuator faults, F1 – top draw actuator fault; F2 – side draw actuator fault; and F3 – bottom reflux duty actuator fault investigated in this case study is also determined. When a fault is declared and identified, for instance top draw actuator fault (F1), we are left with just two healthy actuators, side draw and bottom reflux duty actuators (u_2 and u_3) to maintain three outputs at set points. This is unrealizable using the conventional PID control strategy. Therefore, only two outputs are controlled directly while the third is uncontrolled. We have chosen the top draw and the side draw end points (y_1 and y_2) as the outputs to control after actuator fault F1 (u_1) was declared by appropriately reconfiguring the remaining healthy actuators. An example of the RGA matrix obtained under F1 is given in equation 57 and Table 3 presents the inputs-outputs pairing for the three fault cases. Table 4 presents the PI controller settings for the reconfigured controllers under normal condition and each faulty actuator.

$$\Lambda_{F1} = \begin{bmatrix} -0.570 & 1.5702 \\ 1.5702 & -0.5702 \end{bmatrix} \tag{57}$$

where Λ_{F1} is the RGA for F1.

Table 3. Controlled and manipulated variables pairing

	Manipulated Inputs			
Controlled Outputs	**Normal**	**F1**	**F2**	**F3**
Top end point (y_1)	u_1	u_3	u_3	--
Side end point (y_2)	u_2	u_2	u_1	u_2
Bot. Reflux Temp. (y_7)	u_3	--	--	u_1

Table 4. Reconfigurable FTC settings (PI controllers)

	Controller parameters							
Controlled Output Loop	**Normal**		**F1**		**F2**		**F3**	
	K_P	T_I	K_P	T_I	K_P	T_I	K_P	T_I
Top end point	0.05	0.0215	0.2	0.004	0.21	0.005	--	--
Side end point	0.45	0.0160	0.45	0.016	0.2	0.001	0.45	0.016
Bot. Reflux Temp.	3	0.005	--	--	--	--	1	0.020

4.1.2. Process Simulation under Fault-Free and Faulty Conditions

The heavy oil fractionator was simulated without actuator faults in Simulink for 2000 minutes with 1-minute sampling time as shown in Figure 11 to collect 2000 samples of the seven outputs and three manipulated variables. Intermediate reflux duty (d_1) and upper reflux duty (d_2) serve as disturbances and were randomly introduced into the system during normal process operation. Gaussian noise of zero mean and 0.003 standard deviation was added to each of the 7 outputs to represent true measurements of the data collected. Figure 12 presents the system actuator outputs under normal operating conditions and their respective outputs responses to changes in set-points and introduction of disturbances. Details of the three actuator fault cases (F1, F2 and F3), one each for the three actuators (u_1, u_2 and u_3) are presented in Table 5. The fault was introduced in each case at 800 minutes as a constant value of 0.5 (i.e., control valve stuck to 0.5). The fault cases were each simulated for 2000 minutes to collect 2000 samples.

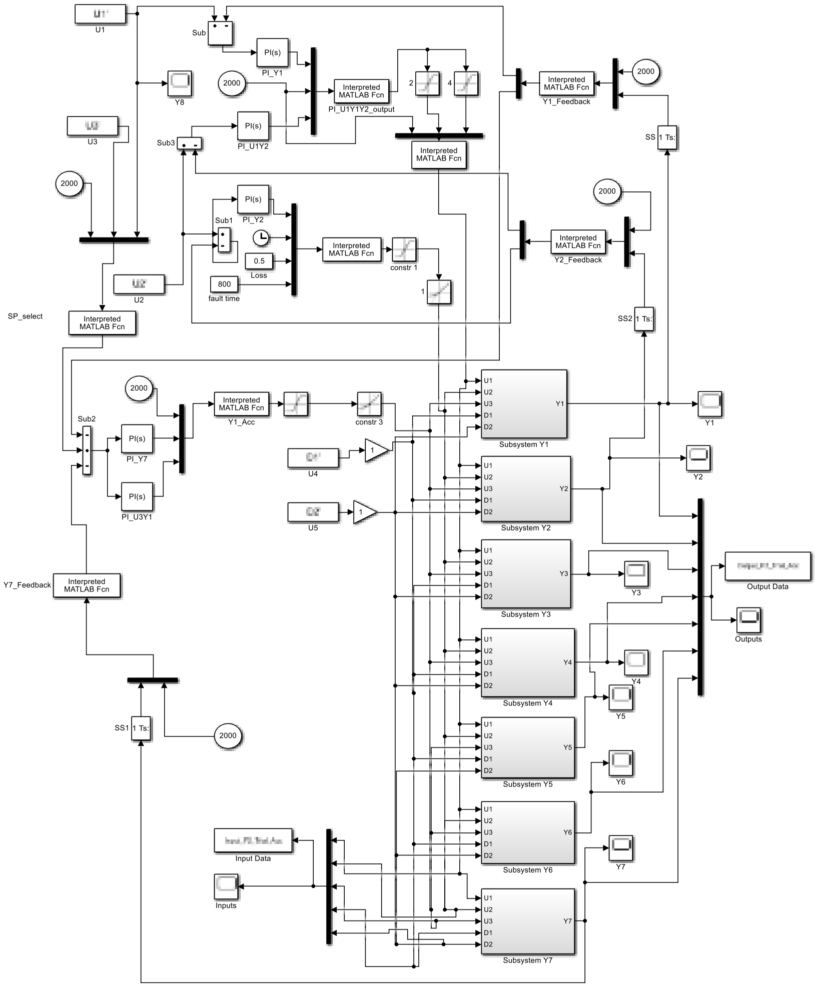


Figure 11. Heavy oil fractionator Simulink model.

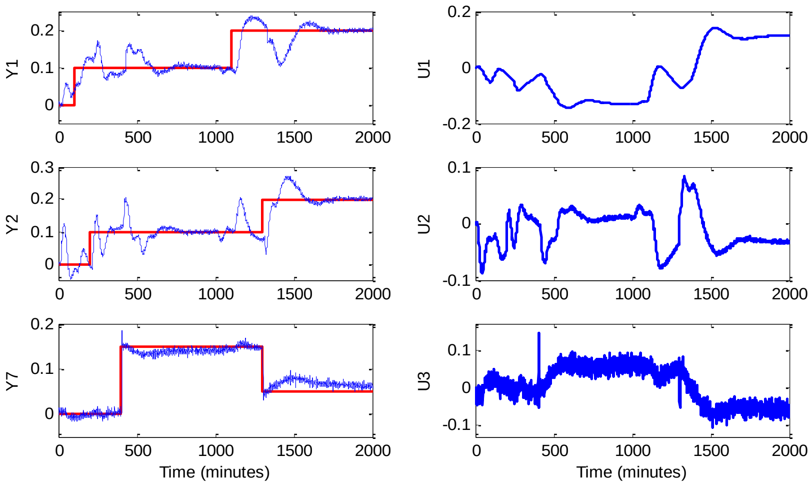


Figure 12. Input and output responses to set-point changes and disturbance.

Table 5. Heavy oil fractionator fault list

Fault	Fault description
F1	Top draw actuator fault – control valve stuck to 0.5
F2	Side draw actuator fault – control valve stuck to 0.5
F3	Bottom reflux duty actuator fault – control valve stuck to 0.5

4.1.3. Actuator Fault Detection and Diagnosis

A DPCA diagnostic model with one time-lagged measurements was developed. 1100 samples of the 2000 samples collected during normal operating conditions were used to develop the DPCA diagnostic model with one time lag while the remaining 900 samples were used for validation. The training data set was scaled to zero mean and unit variance. Three principal components which account for 86.95% variation (a = 3) in the original data are used to develop the DPCA diagnostic model for process monitoring and actuator FDD. Figure 13 shows the process monitoring performance indices for the training and testing data sets. The developed diagnostic model is then applied to the three faulty actuator cases in the Shell heavy oil fractionator to detect possible fault occurrences. A fault is declared when the monitoring indices, T^2 and SPE violate their respective limits for four consecutive sampling times to ensure no false alarm is recorded. Figure 14 presents the Hotelling's T^2 and SPE process monitoring performance for the three faults (F1 – F3). After a fault is declared, its root cause is further investigated through contribution plots which provide information on the contribution of each variable to the faulty scenario thereby aiding its isolation.

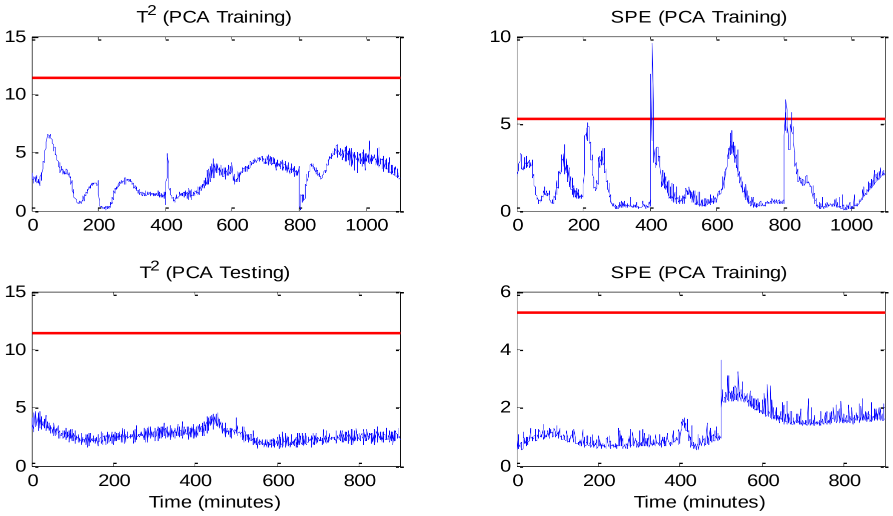


Figure 13. T^2 and SPE monitoring plots for training and testing data.

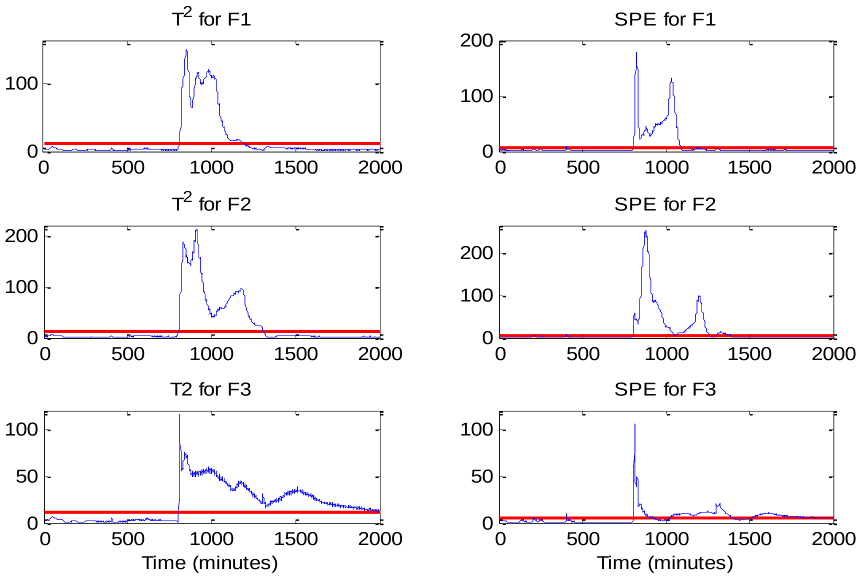


Figure 14. T^2 and SPE monitoring plots for faults F1 – F3.

4.1.4. Implementation of FTC on Identified Actuator Fault

When there is an actuator fault, the 3 by 3 control configuration used for normal process operation will have to be restructured, settings of the reconfigurable controllers retuned, and the set points switched as appropriate upon detection and isolation of an actuator fault in order to maintain the integrity of the system. This is achieved through the feedback and set points backup signals as shown in Figure 10. The different pre-assessed input-output pairings presented in Table 3 and the appropriate reconfigured controller settings presented in Table 4 are implemented, depending on the fault identified. The error vector generated for the reconfigurable controller for the system during normal operation is obtained as equation 58 using equation 47.

$$e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{p1} \\ r_{p2} \\ r_{p3} \\ r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{p1} \\ y_{p2} \\ y_{p3} \\ y_{b1} \\ y_{b2} \\ y_{b3} \end{bmatrix} \tag{58}$$

where r_{p1}, r_{p2}, r_{p3} are the reference points for the system outputs; y_{p1}, y_{p2}, y_{p3} are the outputs; r_{b1}, r_{b2}, r_{b3} are the backup signals for reference point and y_{b1}, y_{b2}, y_{b3} are the corresponding outputs backup feedback signals. When the top draw actuator fault (F1) is declared and the fault tolerant controller reconfigured as appropriate, equation 59 is obtained.

$$e_R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{p1} \\ r_{p2} \\ r_{p3} \\ r_{b1} \\ r_{b2} \\ r_{b3} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{p1} \\ y_{p2} \\ y_{p3} \\ y_{b1} \\ y_{b2} \\ y_{b3} \end{bmatrix} \tag{59}$$

Then, the fault tolerant control law under F1 is given as

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_{b1} \\ u_{b2} \\ u_{b3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{b1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ r_{p2} - y_{p2} \\ 0 \\ r_{b1} - y_{b1} \\ 0 \\ 0 \end{bmatrix} \tag{60}$$

As shown in equation 59 and equation 60 above, the weightings for different signals are activated or deactivated as appropriate to accommodate the fault declared and sub-optimally maintain the system within acceptable operating region.

4.1.5. Results and Discussions

The three actuator faults investigated in this system – top draw actuator fault (F1), side draw actuator fault (F2) and the bottom reflux duty actuator faults (F3) were all detected. The DPCA diagnostic model monitoring statistics, T^2 and SPE detected the top draw actuator fault (F1) 11 minutes and 8 minutes respectively after its introduction as presented in Figure 14. Side draw reflux actuator fault (F2) violated the T^2 and SPE monitoring limits at 809 and 807 minutes respectively while bottom reflux duty actuator fault (F3) was detected at 808 and 806 minutes respectively. Hotelling's T^2 and SPE variable contribution plots are analysed at the point an actuator fault is detected to investigate the root cause of the fault. The variable contribution plots shown in Figure 15 present excess contributions of each variable to the average values of T^2 and SPE that led to the fault being declared. Top temperature (variable 3) and top draw (variable 8) contributed significantly to the fault, as identified by T^2 contribution plot. The SPE contribution plot shows side end point, top temperature, upper reflux temperature, side draw and bottom reflux duty (variables 2, 3, 4, 5, 8 and 10) as the major contributors to the faulty situation recorded. A critical analysis of the effect of top draw actuator fault (F1), depending on the magnitude of the fault, shows a similar effect on the variables identified by the diagnostic model as being responsible for the fault.

Though the T^2 and SPE contribution plots give indications of the likely causes of the fault, however an understanding of the system is still required to make the connections between the fault detected and the variables identified by the isolation technique.

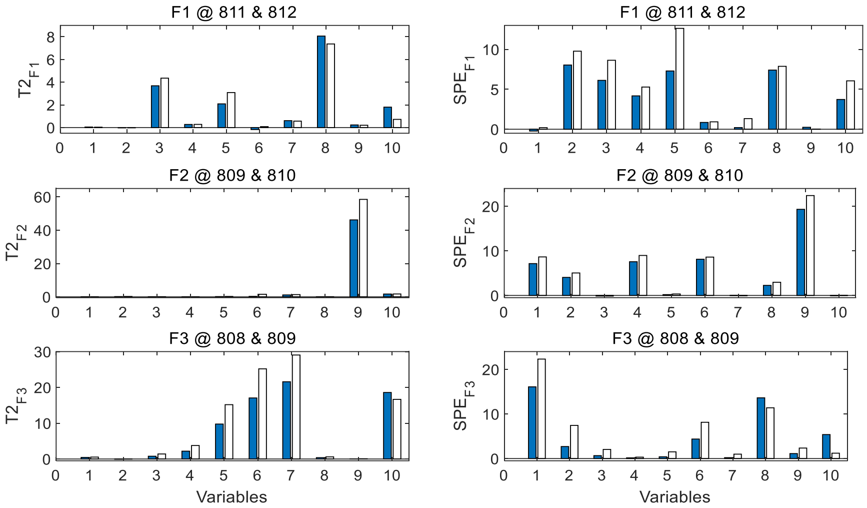


Figure 15. T^2 and SPE excess contribution plots for faults F1 – F3.

The side draw actuator fault (F2) was caused by a significantly large value of side draw (variable 9) which is the output of the faulty actuator as identified by contribution plots. SPE contribution plots in addition to the faulty actuator output also show top end point, side end point, upper reflux temperature and intermediate reflux temperature (variables 1, 2, 4, 6 and 9) as the variables responsible for the fault, as presented in Figure 15. Similarly, side draw temperature, intermediate reflux temperature, bottom reflux temperature and bottom reflux duty actuator output (variables 5, 6, 7, and 10) are identified by the T^2 contribution plots as the variables responsible for the bottom reflux duty actuator fault (F3). SPE contribution plots indicate the top end point and the top draw actuator (variables 1 and 8) as the root causes of the fault.

After the fault is detected and isolated as either being top draw actuator fault (u_1), side draw actuator fault (u_2) or bottom reflux duty actuator fault (u_3), it must be accommodated in order to stabilise the system and ensure its continued safe operation, at least sub-optimally. When top draw actuator fault (u_1) occurs, clearly the 3 by 3 control structure will not be functional and depending on the severity of the fault, one of the remaining two healthy actuators, side draw actuator (u_2) and the bottom reflux duty actuator (u_3) are reconfigured to control the top end point (y_1) as presented in Table 3. Reflux duty actuator (u_3) is reconfigured to maintain the top end point at set point, leaving the bottom reflux temperature (y_7) uncontrolled, as shown in Figure 16. The control structure reconfiguration was achieved through the backup feedback signals presented in Figure 8. Appropriate backup feedback signals, in this case r_{b1} and y_{b1} were activated by changing their weightings from 0 to 1 and at the same time changing the weightings of the corresponding feedback signal to zero, as shown in equation 59. SP, NF, FR and AFR in Figures 16 – 18 represent set points, no faults, fault responses and accommodated fault responses respectively. It can be observed from Figure 16 that the bottom reflux duty was able to maintain the top end point at set point despite the influence of disturbances. Also, the performance of side end point control loop was slightly affected due to the strong interaction in the system.

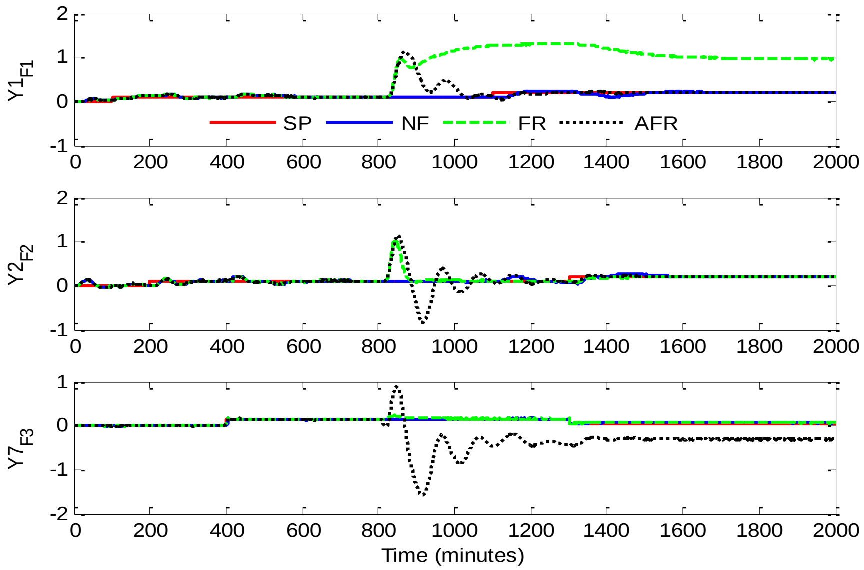


Figure 16. Output responses of accommodated actuator fault 1 (F1).

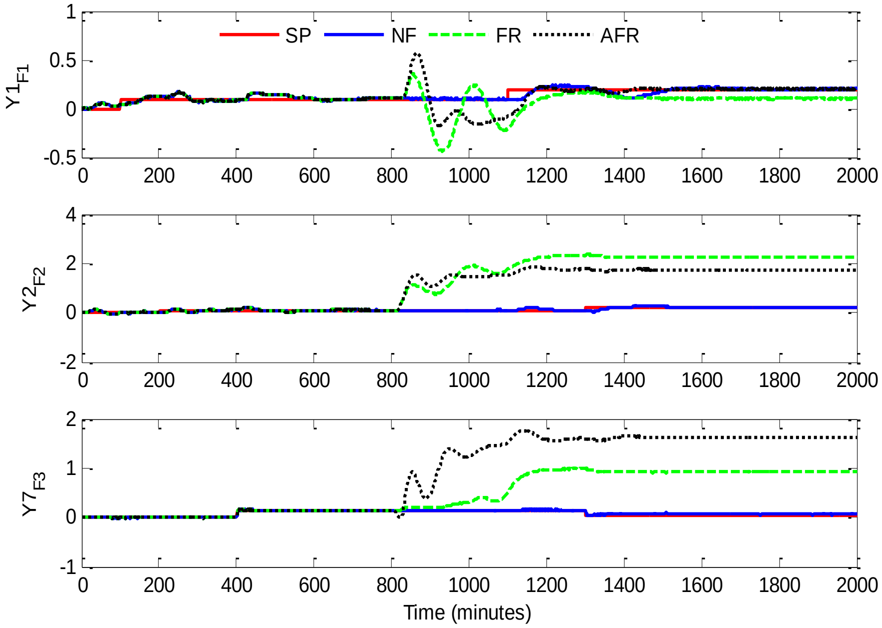


Figure 17. Output responses of accommodated actuator fault 2 (F2).

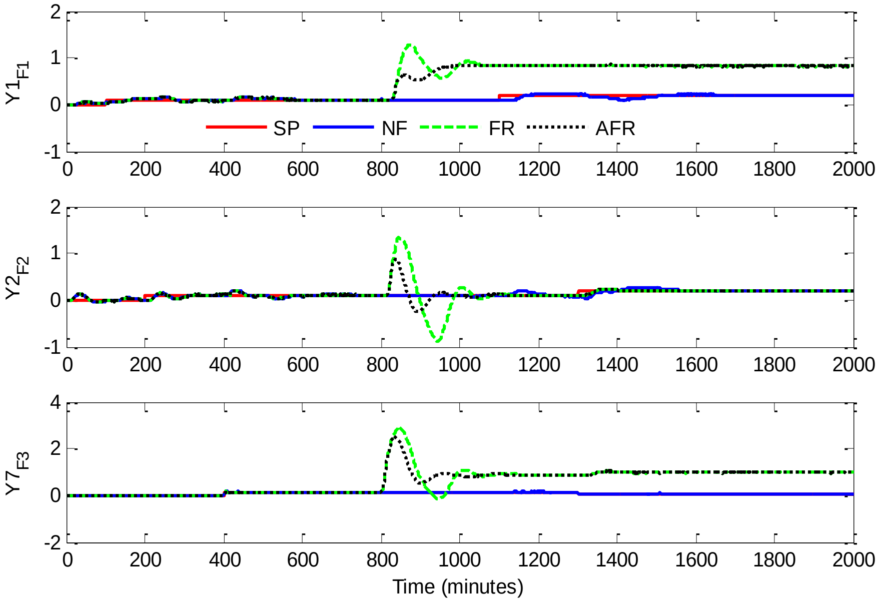


Figure 18. Output responses of accommodated actuator fault 3 (F3).

In the case of side draw actuator fault (F2), none of the two remaining healthy actuators (u_1 and u_3) was able to accommodate the fault. Though the RGA analysis suggests the top draw (u_1) should be able to maintain the side end point (y_2) at set point, however its performance was very poor as can be observed from Figure 17. FTC was reconfigured to a 2 by 2 structure controlling the top end point (y_1) and side end point (y_2) by manipulating bottom reflux duty (u_3) and top draw (u_1) respectively making use of the backup feedback signals and reference point reconfiguration mechanism. The bottom reflux duty was able to keep the top end point at set point. However, the top draw was not effective in maintaining side end point at set point. The bottom reflux temperature is uncontrolled having reduced the control configuration to 2 by 2.

The same scenario was observed when the bottom reflux duty actuator fault (u_3) was declared. Neither of the two remaining healthy actuators, top and side draw actuators (u_1 and u_2) were able to control the bottom reflux temperature. Figure 18 presents the fault tolerant controller performance for the bottom reflux duty actuator fault (F3) where top draw actuator (u_1) was reconfigured to control y_7. Observations from Figure 18 shows that implementation of fault tolerant controller in this case could not improve the system performance.

4.2. Application to Crude Distillation Unit

The preservation of the integrity of crude distillation unit in the presence of actuator faults through the implementation of the proposed actuator FTC is presented in this section. The crude distillation unit is a complex energy intensive industrial distillation process with substantial time lag and severe interaction amongst its control loops. The use of a control system in CDU only ensures system stability and consistent production of quality products as long as no fault occurs. However, in the presence of control system component faults, such as actuator fault, a more robust control system with automatic components containment capabilities will be required to provide desirable performance in the system. Implementation of the proposed actuator fault tolerant control system on such a complex process demonstrates the simplicity, effectiveness and applicability of the accommodating strategy in the presence of actuator faults. A dynamic HYSYS model of the crude distillation unit, including the actuator faults, investigated in this chapter is presented in Figure 19. The dynamic CDU model has been used in previous works by Yu et al. [100] and Zhou et al. [101] to investigate multi-objective optimization of industrial CDU and inferential estimation of kerosene dry point respectively.

4.2.1. Crude Distillation Unit Process Description

The crude distillation unit simulated in HYSYS consists of a train of heat exchangers, an atmospheric CDU with a 3-phase condenser attached, a vacuum CDU, three pumparound cooling circuits, three side draws with stripper attached to each, crude furnace, several separator vessels and 29 control loops. Three different crudes designated as standard, middle and heavy are created in HYSYS. However, only standard crude was used throughout the simulation with random introduction of middle and heavy crudes during simulation, representing approximately 2% of the total volume of crude charged into the system. This introduction serves as disturbances in the system, representing changes in composition of the standard crude. The crude is heated to 185 ^{o}C through series of exchangers by exchange with hot intermediate streams from the crude and vacuum columns before entering the furnace where its temperature is raised to 360 ^{o}C, the temperature at which it enters the atmospheric column flash zone. This chapter focuses mainly on the atmospheric crude unit; hence operations and control of the vacuum unit is not discussed.

The crude mixes with the bottom boil-up vapour and the steam injected into the bottom separator vessel to strip the lightest hydrocarbons from the column bottom residue. As the hot vapour from the flash zone rises, it is contacted by the colder reflux flowing down the column. The pumparound circuits and the overhead condenser provide the reflux that is used to condense the side liquid products. The products from the system are naphtha, kerosene, diesel, atmospheric gas oil (AGO) and the CDU residue. The side liquid products are kerosene, diesel and AGO; and are drawn from column stages 9, 17 and 22 respectively. The side products are transferred into their respective side strippers with attached reboiler and separator vessels with steam injection lines to strip the

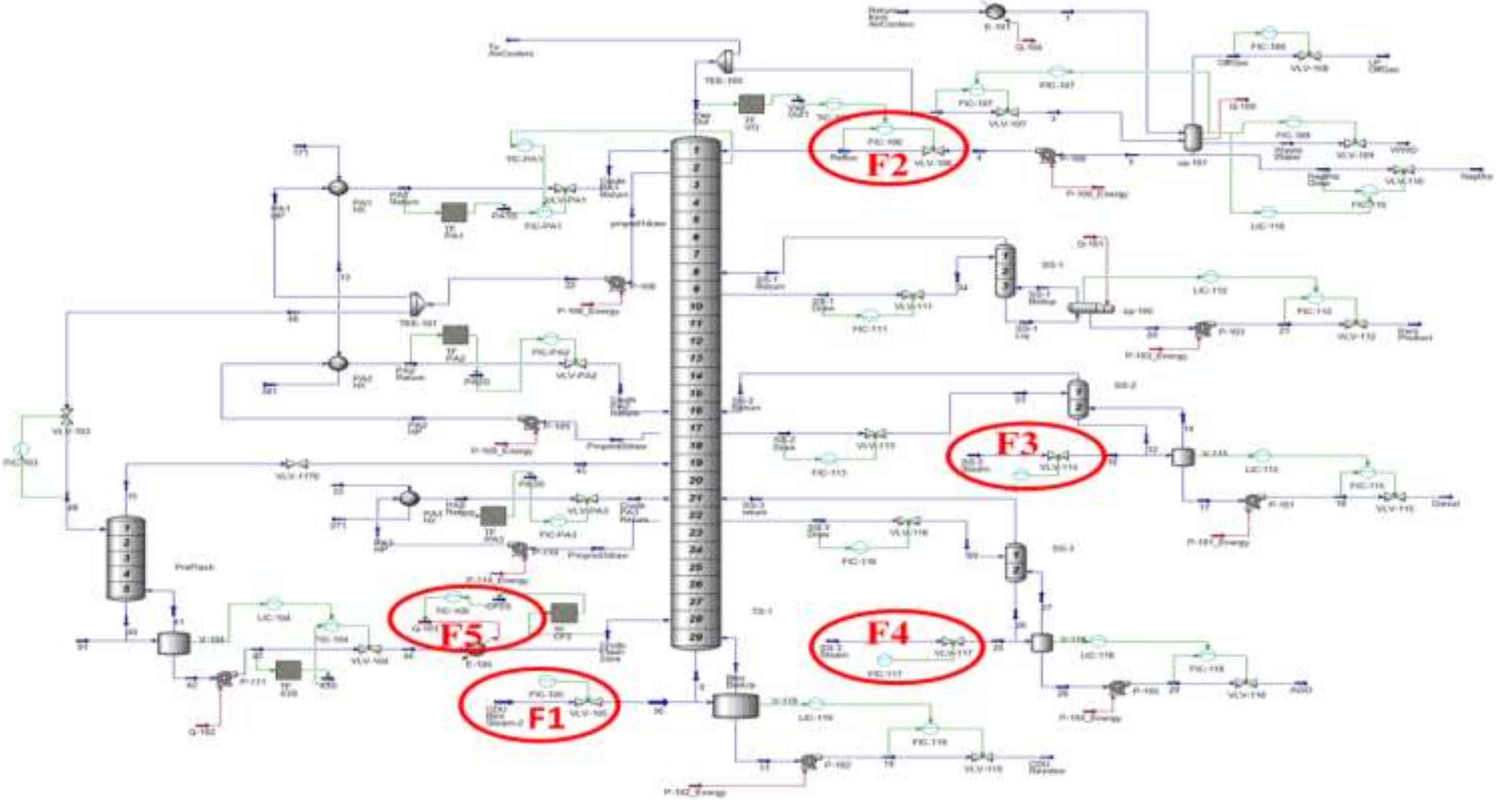


Figure 19. Dynamic CDU model in HYSYS with faults F1 – F5.

products off lighter hydrocarbons. The pumparound streams are drawn from stages 2, 17 and 22; and the cooled pumparound streams are returned to stages 1, 16 and 21 respectively.

Table 6. CDU control structure

No	Controllers	Controlled Variables	Manipulated Variable	Nominal Valve Opening
1	FIC-PA1	1st pumparound mass flow	Valve PA1 desired position	85.70%
2	FIC-PA2	2nd pumparound mass flow	Valve PA2 desired position	77.33%
3	FIC-PA3	3rd pumparound mass flow	Valve PA3 desired position	56.65%
4	TIC-100	Crude flash zone temperature	Q-103 control valve	48.51%
5	FIC-103	Stream 48 mass flow	Valve 103 desired position	32.10%
6	LIC-104	V-104 liquid percent level	Separator V104 liquid level	49.37%
7	FIC-104	Stream 43 mass flow	Valve 104 desired position	41.59%
8	FIC-105	CDU steam mass flow	Valve 105 desired position	50.83%
9	TIC-106	Vap out temperature	FIC-106 SP	34.27%
10	FIC-106	Reflux mass flow	Valve 106 desired position	32.66%
11	PIC-107	OP-101 Condenser Pressure	FIC-107 SP	92.61%
12	FIC-107	Stream 2 mass flow	Valve 107 desired position	90%
13	FIC-108	LP Offgas mass flow	Valve 108 desired position	0%
14	LIC-110	OP-101 liquid percent level	FIC-110 SP	59.50%
15	FIC-110	Naphtha draw flow	Valve 110 desired position	30.97%
16	FIC-111	Side draw 1 mass flow	Valve 111 desired position	39.13%
17	LIC-112	OP-100 reboiler liquid percent level	FIC-112 SP	7.58%
18	FIC-112	Kerosene mass flow	Valve 112 desired position	21.27%
19	FIC-113	Side draw 2 mass flow	Valve 113 desired position	79.07%
20	FIC-114	Side stripper 2 steam flow	Valve 114 desired position	35.35%
21	LIC-115	Separator V-115 liquid level	FIC-115 SP	57.13%
22	FIC-115	Diesel mass flow	Valve 115 desired position	73.34%
23	FIC-116	Side draw 3 mass flow	Valve 116 desired position	74.16%
24	FIC-117	Side stripper 3 steam flow	Valve 117 desired position	76.73%
25	LIC-118	Separator V-118 liquid level	FIC-118 SP	82.62%
26	FIC-118	AGO mass flow	Valve 118 desired position	90.00%
27	LIC-119	Separator V-119 liquid level	FIC-119 SP	47.46%
28	FIC-119	CDU residue mass flow	Valve 119 desired position	42.29%

The column has 29 control loops – 20 flow control loops, 6 level control loops, 2 temperature control loops and 1 pressure control loop. The flow control loops are used to maintain the flowrates of specific streams including pumparound, side draw, steam and products rate. The level controllers are used as master controllers together with some flow controllers in a cascade control setting to control liquid percentage levels in the separators attached to the pre-flash column, the main column, side strippers and liquid level in the overhead condenser. The temperature controllers control the temperature of the crude entering the flash zone and that of the vapour leaving the top of the column, while the pressure controller controls the pressure of the 3-phase condenser. Details of all the controlled variables (y_i) – manipulated variables (u_i) pairing and their respective nominal operating conditions are presented in Table 6 while Table 7 presents the column nominal values for some selected process variables. The controllers are used to maintain the flow, temperature and pressure profile of the column, which in turn maintain the specified product quality variables for the system. The product quality variables used for the CDU are ASTM D1160 cut-points at 0% and 100% for kerosene, ASTM D1160 cut-points at 90% and 95% for diesel, ASTM D93 flash points for kerosene and AGO, and AGO viscosity at 210F.

Table 7. Nominal crude distillation unit operating conditions

Selected Process Variables	**Values**
Crude mass flow	235,200 kg/hr
Crude temperature	15.65^{o}C
Crude flash zone flow	209,200 kg/hr
Crude flash zone temperature	360^{o}C
Reflux flow rate	68,550 kg/hr
Reflux flow temperature	22^{o}C
Vapour out flow	125,800 kg/hr
Vapour out temperature	138^{o}C
Column bottom boil-up	9,931 kg/hr
First pumparound flowrate	121,600 kg/hr
First pumparound return temperature	119^{o}C
Second pumparound flowrate	50,000 kg/hr
Second pumparound return temperature	200^{o}C
Third pumparound flowrate	35,000 kg/hr
Third pumparound return temperature	245^{o}C
First side draw flow rate	15,000 kg/hr
First side draw temperature	199^{o}C
Second side draw flow rate	65,000 kg/hr
Second side draw temperature	245^{o}C
Third side draw flow rate	20,000 kg/hr
Third side draw temperature	327.2^{o}C
Naphtha to crude feed ratio	0.221
Kerosene to crude feed ratio	0.03062
Diesel to crude feed ratio	0.2351
AGO to crude feed ratio	0.06921
CDU residue to crude feed ratio	0.4439
Heat flow to the furnace	1.198e+008 kJ/h

4.2.2. Development and Simulation of Interactive Dynamic Crude Distillation Units

The dynamic crude distillation model in HYSYS is integrated with MATLAB programme to create an interactive dynamic crude distillation simulator, through which effective implementation of the proposed actuator fault tolerant control system could be achieved. This requirement is fundamental to successfully implement the actuator FTC because it allows flow of information between the two applications. The CDU model to monitor and control resides in HYSYS while the actuator FTC system is developed in MATLAB. Hence, the need to automate the operation, monitoring and implementation of the proposed FTCS on the dynamic CDU model in HYSYS. Figure 20 illustrates the interface between the two applications. First, an active connection is created in MATLAB that allows it to connect and simulate the dynamic CDU model in HYSYS for a specified period. Then different sub connections are created in MATLAB to access objects in the CDU HYSYS model that contain the variables of interest. These are process variables that need to be monitored, manipulated and controlled in some cases to ensure that actuator faults are quickly detected, identified and accommodated, depending on the severity of the fault identified.

To minimise the number of object connections created in MATLAB, an appropriate number of spreadsheet objects are created in HYSYS that contain all the different variables and parameters that may need to be adjusted as appropriate during the simulation. The spreadsheets created in HYSYS include spreadsheets for all the process variables of interest, the manipulated variables, controlled variables, disturbance variables, process quality variables, and the percentage maximum control valve openings. Data in the spreadsheets are accessed and stored in MATLAB during simulation. The disturbance variables spreadsheet is used to randomly introduce disturbances into the system during simulation while the actuator fault spreadsheet is used to introduce faults into the system. The process variables collected during the simulation include temperature and flow rate measurements of the crude flash zone, pump-arounds, side draws, reflux stream, and the temperature measurements of all the 29 stages in the column. Flow rates and temperatures of naphtha, kerosene, diesel, AGO, the CDU residue and the ratios of the feed rate to each of the products flow rates are also included. The automated HYSYS-MATLAB CDU model is simulated for 600 minutes with 30 seconds sampling time to collect 1200 data points under normal operating conditions. A total of seventy-one variables including the controlled variables, some manipulated variables and the disturbance variables are monitored during the simulation. Figure 21 presents plots of the product quality variables for the system during normal operating conditions. Table 8 presents the nominal values of the product quality variables. The system is also simulated for 400 minutes with 30 seconds sampling time for the five different actuator faults investigated in this section (F1 – F5) as shown in Figure 19.

Table 8. CDU process quality variables

Process Quality Variables	Values
Kero Cut Pt: ASTM D1160 – Atm (Cut Pt-0.0%)	184.9ºC
Kero Cut Pt: ASTM D1160 – Atm (Cut Pt-100.0%)	241.7ºC
Kero Cut Pt: ASTM D93 Flash Pt	71.28ºC
Diesel Cut Pt: ASTM D1160 – Atm (Cut Pt-90.0%)	341ºC
Diesel Cut Pt: ASTM D1160 – Atm (Cut Pt-95.0%)	354.9ºC
AGO Cut Pt: ASTM D93 Flash Pt	139.1ºC
AGO Cut Pt: Viscosity @ 210F	3.22 cP

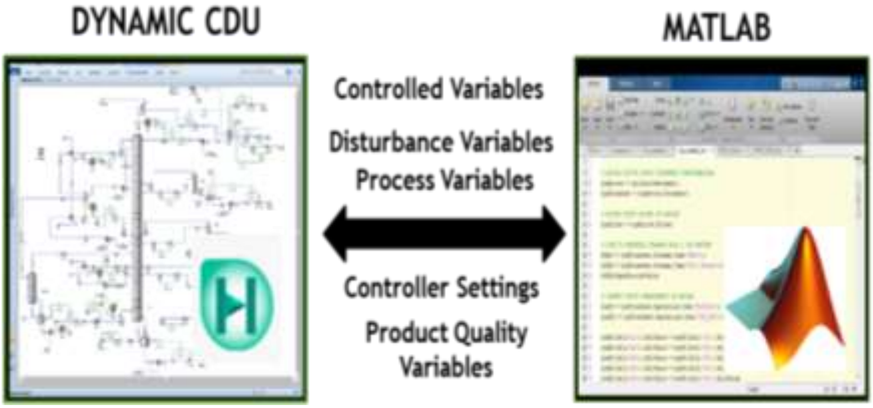


Figure 20. Interactions between HYSYS and MATLAB application.

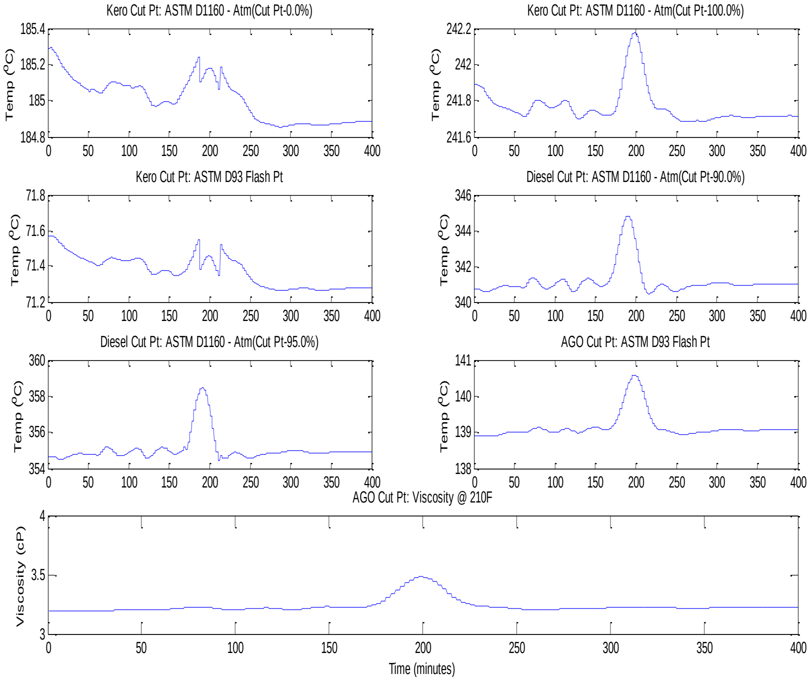


Figure 21. Dynamic CDU product quality variables.

4.2.3. Control Strategies Prior Assessment

The current control structure of the system presented in Figure 19 needs to be modified to allow for implementation of the proposed actuator FTC on the system. Majority of the controlled variables that we are interested in are indirectly controlled by maintaining the flow rates of some of the streams including pump-arounds, side draws, and products draw rates. However, after careful consideration and rigorous analyses of the CDU current control structure, we decided to restructure some of the control loops to directly control some variables using suitable manipulated variables. Reasonable efforts were made to reduce the model to a manageable 5 by 5 control structure in order to implement the proposed actuator FTC on the automated CDU system. More details about the control structure reduction can be found in [102].

Details of the reduced model inputs – outputs pairing is shown in Table 9 and equations 61 and 62 present the FOPDT models and the RGA results respectively. The reduced fault free system has bottom boil-up flow (y_1), stage 1 temperature (y_2), diesel temperature (y_3), AGO temperature (y_4) and crude flash zone temperature (y_5) being directly controlled by CDU bottom steam (u_1), reflux flow rate (u_2), side stripper 2 (SS-2) steam flow rate (u_3), side stripper 3 (SS-3) steam flow rate (u_4) and furnace heat output (u_5) respectively. The reconfigured pairing using the manipulated variables to directly control the selected controlled variables is simulated to ensure effective control of the outputs. The performances of the new control structure using PID controllers tuned with the IMC tuning tool in HYSYS is compared with those of the original structure for the selected controlled variables and presented in Figure 22. Table 10 presents the PID controllers settings used for the reconfigured system during fault free and faulty situations.

Table 9. Reduced 5 by 5 inputs – outputs pairing

No	Controllers	Controlled Variables	Manipulated Variable
1	FIC-PA1	Bottom boil-up flow (9,931 kg/h)	CDU bottom steam 2 (3,500 kg/h)
2	FIC-PA2	Stage 1 temperature (138ºC)	Reflux mass flow (68,550 kg/h)
3	FIC-PA3	Diesel temperature (228.9ºC)	SS-2 steam flow (1,000 kg/h)
4	FIC-104	AGO temperature (309.7ºC)	SS-3 steam flow (500 kg/h)
5	FIC-105	CFZ temperature (360ºC)	Furnace heat flow (1.198e+08 kJ/h)

$$G_5(s) = \begin{bmatrix} \frac{30.85}{s+26.11} & \frac{0.0321e^{-5.34s}}{1+0.086s} & \frac{0.0065e^{-1.5s}}{s+0.054} & \frac{0.00014e^{-1.5s}}{s+0.361} & \frac{6.18e{-}07e^{-12s}}{s+0.0068} \\ \frac{0.0145}{s+0.064} & \frac{-0.055931}{1+1.001s} & \frac{0.0271e^{-0.5s}}{s+0.099} & \frac{0.0247}{s+0.101} & \frac{7.39e{-}06e^{-0.5s}}{s+0.0778} \\ \frac{0.03556}{s+0.098} & \frac{-0.0535e^{-3.59s}}{1+1.414s} & \frac{-0.301}{s+0.575} & \frac{0.0485}{s+0.129} & \frac{1.61e{-}05e^{-2s}}{s+0.1536} \\ \frac{0.03637}{s+0.078} & \frac{-2.095e^{-14.5s}}{s+41.2} & \frac{-0.0055e^{-3.5s}}{s+0.031} & \frac{-0.2837}{s+0.256} & \frac{0.00011e^{-10.5s}}{1+1e{-}06s} \\ \frac{0.17986}{1+2.95s} & \frac{0.00035e^{-6s}}{s+0.0114} & \frac{0.191}{1+1.012s} & \frac{0.182}{1+1.37s} & \frac{0.000173}{s+1.46} \end{bmatrix} \quad (61)$$

$$\Lambda_{G5} = \begin{bmatrix} 1.1472 & 0.0427 & 0.0191 & -0.0015 & -0.2075 \\ 0.0344 & 0.5563 & 0.3548 & 0.0723 & -0.0178 \\ 0.0138 & 0.0647 & 0.6097 & 0.1529 & 0.1589 \\ -0.0473 & 0.0580 & 0.0205 & 0.7805 & 0.1882 \\ -0.1481 & 0.2783 & -0.0042 & -0.0042 & 0.8782 \end{bmatrix} \tag{62}$$

where G_5 and Λ_{G5} are the transfer function models of the reduced 5 by 5 system and the corresponding RGA respectively.

Having established a stable operation of the fault free system with the restructured controllers and achieved effective control of the selected outputs, possible reconfiguration of the control structure is undertaken in the event of an actuator fault occurring in any of the five control loops. RGA and DRGA tools are used a priori to investigate possible control structure reconfiguration upon detection of an actuator fault. For instance, a fault in the CDU bottom steam (u_1) control valve will reduce the system to a 5 by 4 control structure where four manipulated variables are available to maintain five controlled variables at desired set points. Non-squared RGA is first used to eliminate the least effective controlled variable thereby reducing the system to 4 by 4 after which RGA and DRGA are used to select possible input-output pairing. The decision on the controlled variable to leave out in the effect of an actuator fault could also be due to economic reasons, and most importantly what is physically and technically achievable given the circumstance. Equations 63 – 72 show the reduced 4 by 4 models for each of the five faults and their respective RGA results. However, not all the faults can be accommodated by switching the manipulated variables, even for the fault-free system as observed during the initial fault free simulation. Table 11 presents the possible inputs – outputs reconfiguration upon detection and identification of an actuator fault.

G_{F1} and Λ_{F1} are the transfer function models and the corresponding RGA values for the reduced 4 by 4 system under actuator fault one (F1). The resulting y_i – u_i pairings are y_2 – u_2, y_3 – u_3, y_4 – u_4 and y_5 – u_5 after isolating u_1 (CDU bottom steam control valve) as there is no suitable manipulated variable to control y_1.

$$G_{F1}(s) = \begin{bmatrix} \frac{-0.055931}{1+1.001s} & \frac{0.0271e^{-0.5s}}{s+0.099} & \frac{0.0247}{s+0.101} & \frac{7.39e{-}06e^{-0.5s}}{s+0.0778} \\ \frac{-0.0535e^{-3.59s}}{1+1.414s} & \frac{-0.301}{s+0.575} & \frac{0.0485}{s+0.129} & \frac{1.61e{-}05e^{-2s}}{s+0.1536} \\ \frac{-2.095e^{-14.5s}}{s+41.2} & \frac{-0.0055e^{-3.5s}}{s+0.031} & \frac{-0.2837}{s+0.256} & \frac{0.00011e^{-10.5s}}{1+1e{-}06s} \\ \frac{0.00035e^{-6s}}{s+0.0114} & \frac{0.191}{1+1.012s} & \frac{0.182}{1+1.37s} & \frac{0.000173}{s+1.46} \end{bmatrix} \tag{63}$$

$$\Lambda_{\mathrm{F1}} = \begin{bmatrix} 0.5681 & 0.3480 & 0.0685 & 0.0153 \\ 0.0675 & 0.6130 & 0.1515 & 0.1680 \\ 0.0508 & 0.0176 & 0.7690 & 0.1625 \\ 0.3136 & 0.0214 & 0.0110 & 0.6541 \end{bmatrix} \tag{64}$$

G_{F2} and Λ_{F2} are the transfer function models and the corresponding RGA values for the reduced 4 by 4 system under actuator fault two (F2). The reconfigured control structure under F2 for the inputs – outputs pairings are $y_1 - u_1$, $y_2 - u_5$, $y_3 - u_3$ and $y_4 - u_4$ after isolating u_2 (reflux flow control valve fault), leaving y_5 uncontrolled.

$$G_{F2}(s) = \begin{bmatrix} \frac{30.85}{s+26.11} & \frac{0.0065e^{-1.5s}}{s+0.054} & \frac{0.00014e^{-1.5s}}{s+0.361} & \frac{6.18e{-}07e^{-12s}}{s+0.0068} \\ \frac{0.0145}{s+0.064} & \frac{0.0271e^{-0.5s}}{s+0.099} & \frac{0.0247}{s+0.101} & \frac{7.39e{-}06e^{-0.5s}}{s+0.0778} \\ \frac{0.03556}{s+0.098} & \frac{-0.301}{s+0.575} & \frac{0.0485}{s+0.129} & \frac{1.61e{-}05e^{-2s}}{s+0.1536} \\ \frac{0.03637}{s+0.078} & \frac{-0.0055e^{-3.5s}}{s+0.031} & \frac{-0.2837}{s+0.256} & \frac{0.00011e^{-10.5s}}{1+1e{-}06s} \end{bmatrix} \tag{65}$$

$$\Lambda_{F2} = \begin{bmatrix} 1.2927 & 0.0195 & -0.0016 & -0.3107 \\ -0.1711 & 0.3482 & 0.0661 & 0.7568 \\ -0.0262 & 0.6113 & 0.1518 & 0.2632 \\ -0.0954 & 0.0210 & 0.7837 & 0.2907 \end{bmatrix} \tag{66}$$

Table 10. Reconfigurable actuator FTC PID settings

		y_1	y_2	y_3	y_4	y_5
Normal	K_p	0.78	0.50	8.35	8.48	0.51
	T_I	0.04	0.30	1.74	3.90	0.69
	T_D	--	--	--	--	--
F1	K_p	--	0.50	8.35	8.48	0.51
	T_I	--	0.30	1.74	3.90	0.69
	T_D	--	--	--	--	--
F2	K_p	0.78	0.25	8.35	8.48	--
	T_I	0.04	13.10	1.74	3.90	--
	T_D	--	0.25	--	--	--
F3	K_p	0.78	--	0.45	8.48	0.51
	T_I	0.04	--	5.00	3.90	0.69
	T_D	--	--	0.79	--	--
F4	K_p	0.78	0.50	--	0.58	0.51
	T_I	0.04	0.30	--	2.00	0.69
	T_D	--	--	--	1.66	--
F5	K_p	0.78	0.50	8.35	8.48	--
	T_I	0.04	0.30	1.74	3.90	--
	T_D	--	--	--	--	--

Table 11. Possible inputs – outputs reconfiguration

	Manipulated Inputs					
Controlled Outputs	**Normal**	**F1**	**F2**	**F3**	**F4**	**F5**
Bottom boil-up flow (y_1)	u_1	--	u_1	u_1	u_1	u_1
Stage 1 temperature (y_2)	u_2	u_2	u_5	--	u_2	u_2
Diesel temperature (y_3)	u_3	u_3	u_3	u_2	--	u_3
AGO temperature (y_4)	u_4	u_4	u_4	u_4	u_3	u_4
Crude flash zone temp. (y_5)	u_5	u_5	--	u_5	u_5	--

G_{F3} and Λ_{F3} are the transfer function models and the corresponding RGA values for the reduced 4 by 4 system under actuator fault three (F3) where u_1, u_2, u_4 and u_5 are reconfigured to control y_1, y_3, y_4, and y_5 respectively after isolating the faulty control valve (u_3), leaving y_2 uncontrolled.

$$G_{F3}(s) = \begin{bmatrix} \frac{30.85}{s+26.11} & \frac{0.0321e^{-5.34s}}{1+0.086s} & \frac{0.00014e^{-1.5s}}{s+0.361} & \frac{6.18e{-}07e^{-12s}}{s+0.0068} \\ \frac{0.03556}{s+0.098} & \frac{-0.0535e^{-3.59s}}{1+1.414s} & \frac{0.0485}{s+0.129} & \frac{1.61e{-}05e^{-2s}}{s+0.1536} \\ \frac{0.03637}{s+0.078} & \frac{-2.095e^{-14.5s}}{s+41.2} & \frac{-0.2837}{s+0.256} & \frac{0.00011e^{-10.5s}}{1+1e{-}06s} \\ \frac{0.17986}{1+2.95s} & \frac{0.00035e^{-6s}}{s+0.0114} & \frac{0.182}{1+1.37s} & \frac{0.000173}{s+1.46} \end{bmatrix} \quad (67)$$

$$\Lambda_{F3} = \begin{bmatrix} 1.1254 & 0.0810 & -0.0010 & -0.2054 \\ 0.0632 & 0.5421 & 0.2534 & 0.1412 \\ -0.0409 & 0.1038 & 0.7508 & 0.1864 \\ -0.1477 & 0.2731 & -0.0032 & 0.8779 \end{bmatrix} \quad (68)$$

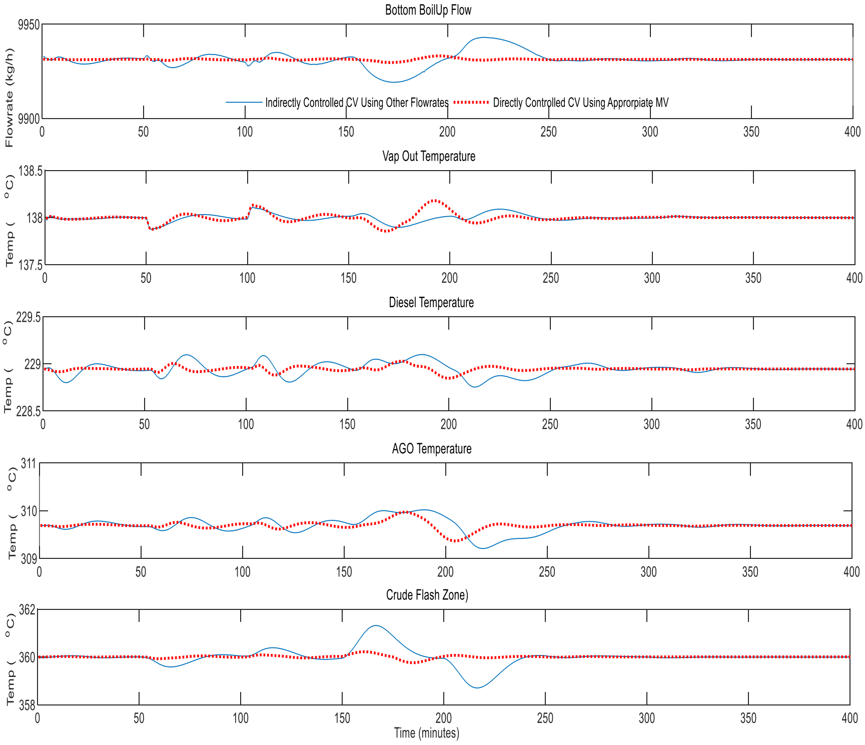


Figure 22. Controlled variable plot for the reconfigured fault-free system.

G_{F4} and Λ_{F4} are the reduced 4 by 4 transfer function models and the corresponding RGA values for the CDU system under actuator fault four (F4) where y_1, y_2, y_4, and y_5 are controlled directly using *u*1, *u*2, *u*3 and *u*5 respectively after control structure reconfiguration with SS-3 steam control valve (*u*4) isolated and *y*3 uncontrolled.

$$G_{F4}(s) = \begin{bmatrix} \frac{30.85}{s+26.11} & \frac{0.0321e^{-5.34s}}{1+0.086s} & \frac{0.0065e^{-1.5s}}{s+0.054} & \frac{6.18e{-}07e^{-12s}}{s+0.0068} \\ \frac{0.0145}{s+0.064} & \frac{-0.055931}{1+1.001s} & \frac{0.0271e^{-0.5s}}{s+0.099} & \frac{7.39e{-}06e^{-0.5s}}{s+0.0778} \\ \frac{0.03637}{s+0.078} & \frac{-2.095e^{-14.5s}}{s+41.2} & \frac{-0.0055e^{-3.5s}}{s+0.031} & \frac{0.00011e^{-10.5s}}{1+1e{-}06s} \\ \frac{0.17986}{1+2.95s} & \frac{0.00035e^{-6s}}{s+0.0114} & \frac{0.191}{1+1.012s} & \frac{0.000173}{s+1.46} \end{bmatrix} \tag{69}$$

$$\Lambda_{F4} = \begin{bmatrix} 1.1363 & 0.0520 & 0.0533 & -0.2417 \\ 0.0281 & 0.5068 & 0.5879 & -0.1228 \\ -0.0167 & 0.1651 & 0.3755 & 0.4761 \\ -0.1478 & 0.2762 & -0.0168 & 0.8884 \end{bmatrix} \tag{70}$$

G_{F5} and Λ_{F5} are the transfer function models and the corresponding RGA values for the CDU system under actuator fault five (F5). There is no suitable manipulated variable that could be used to directly maintain *y*5 at set point and the system control structure is reduced to 4 by 4 with *y*1 – *u*1, *y*2 – *u*2, *y*3 – *u*3 and *y*4 – *u*4 inputs – outputs pairing respectively.

$$G_{F5}(s) = \begin{bmatrix} \frac{30.85}{s+26.11} & \frac{0.0321e^{-5.34s}}{1+0.086s} & \frac{0.0065e^{-1.5s}}{s+0.054} & \frac{0.00014e^{-1.5s}}{s+0.361} \\ \frac{0.0145}{s+0.064} & \frac{-0.055931}{1+1.001s} & \frac{0.0271e^{-0.5s}}{s+0.099} & \frac{0.0247}{s+0.101} \\ \frac{0.03556}{s+0.098} & \frac{-0.0535e^{-3.59s}}{1+1.414s} & \frac{-0.301}{s+0.575} & \frac{0.0485}{s+0.129} \\ \frac{0.03637}{s+0.078} & \frac{-2.095e^{-14.5s}}{s+41.2} & \frac{-0.0055e^{-3.5s}}{s+0.031} & \frac{-0.2837}{s+0.256} \end{bmatrix} \tag{71}$$

$$\Lambda_{F5} = \begin{bmatrix} 0.8544 & 0.1287 & 0.0183 & -0.0014 \\ 0.0297 & 0.5435 & 0.3546 & 0.0721 \\ 0.0747 & 0.1632 & 0.6074 & 0.1547 \\ 0.0412 & 0.1646 & 0.0197 & 0.7746 \end{bmatrix} \tag{72}$$

4.2.4. Introduction of Actuator Faults

The five actuator faults investigated in this CDU case study are presented in Table 12. F1 is the bottom boil-up control valve fault; F2 is the reflux flow control valve fault; F3 is the SS-2 steam control valve fault; F4 is the SS-3 steam control valve fault, and F5 is the furnace heat flow control valve fault. The full range of throttling of the control valves is restricted one at a time to values below their nominal operating conditions on

Table 12. Crude distillation unit fault list

Fault	Fault description
F1: FIC 105	CDU bottom steam control valve maximum opening restricted to 30% (50.84**)
F2: FIC 106	Reflux flow control valve maximum opening restricted to 28% (32.66**)
F3: FIC 114	Side stripper-2 steam control valve maximum opening restricted to 25% (35.35**)
F4: FIC 117	Side stripper-3 steam control valve maximum opening restricted to 50% (76.73**)
F5: TIC 100	Furnace heat flow control valve maximum opening restricted to 44.82% (48.51**)

Note: ** - Nominal operating condition.

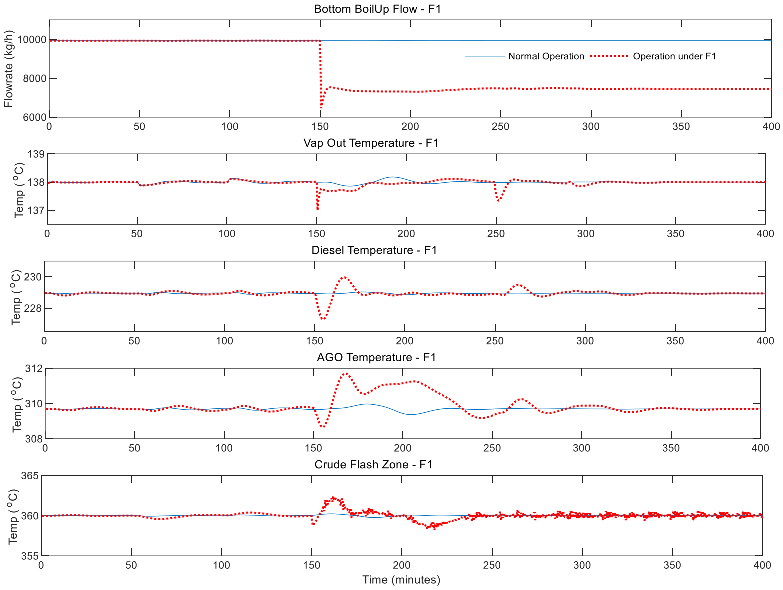


Figure 23. Plots of the controlled variables under F1.

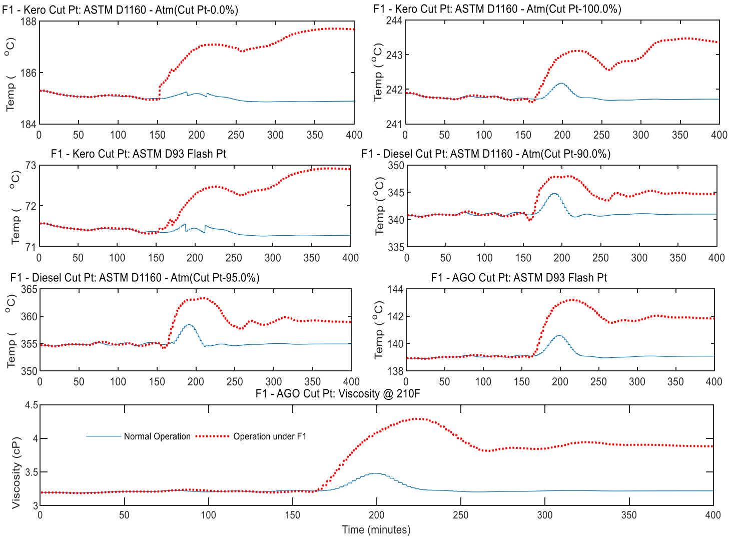


Figure 24. Plots of the product quality variables under F1.

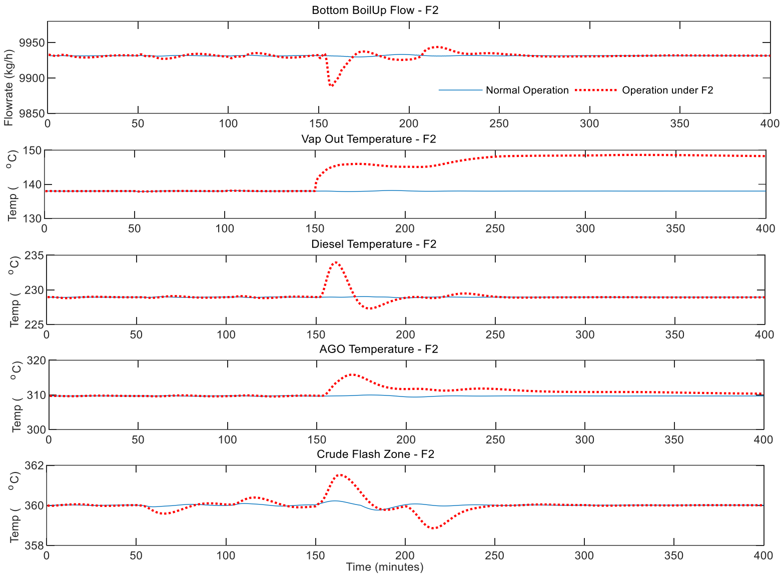


Figure 25. Plots of the controlled variables under F2.

150 minutes (sample 300) during the simulation. These restrictions limit the ability of the individual control valve to maintain their respective controlled variables at set point leading to detection of faults. Effects of the individual faults on the five controlled variables and the seven product quality variables for faults F1 and F2 are shown in Figures 23 to 26. Plots for the other three actuator faults (F3 – F5) are presented under discussion of results section.

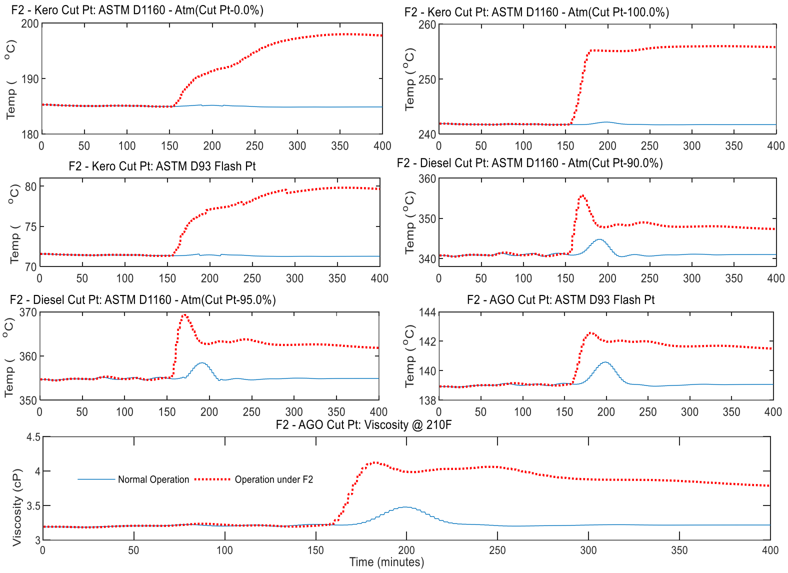


Figure 26. Plots of the product quality variables under F2.

4.2.5. Diagnostic Model Development and Faults Detection and Identification

The 1200 data points collected for all the seventy-one process variables monitored during the CDU fault-free simulation in MATLAB are used to develop a DPCA diagnostic model for the system. First, measurement noises are added to all the variables except the feed charged to product flow ratios. The data is then scaled to zero means and unit variance. 800 samples out of the 1200 collected during normal operating conditions are used to develop the fault detection and diagnostic model, while the remaining 400 data points are used to validate the model. Five principal components which account for 85.85% variation in the original data set with one time lag (l = 1) are sufficient to develop the dynamic PCA diagnostics model. The diagnostic model is then used to monitor the operation of the interactive dynamic CDU system under the five faulty conditions to detect and identify possible occurrence of actuator faults.

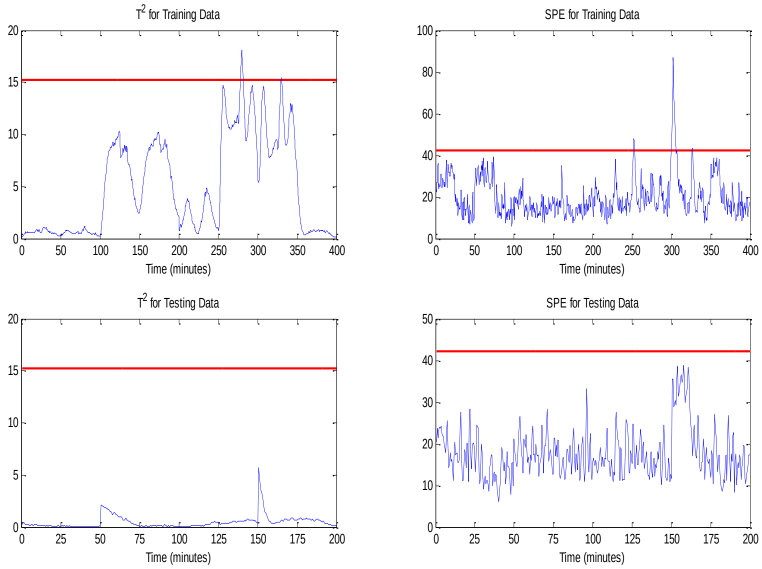


Figure 27. T^2 and SPE plots for the training and validating data sets.

Figures 27 and 28 present the T^2 and SPE monitoring statistics with control limit (red line) for the training and validating data sets and those of the five fault cases (F1 – F5) respectively. The values of the T^2 and SPE monitoring statistics should be small and within their control limits in the absence of fault, but large enough to be detected as fault when one is present. An actuator fault is declared after the limits of both monitoring statistics are violated for eight consecutive sampling times (4 minutes) simultaneously. Actuator faults could also be declared faster if the values of the monitoring statistics are more than double those of their respective limits for two consecutive sampling period. These criteria are appropriate to eliminate declaration of false alarm, and also because of the complexity of the system being investigated.

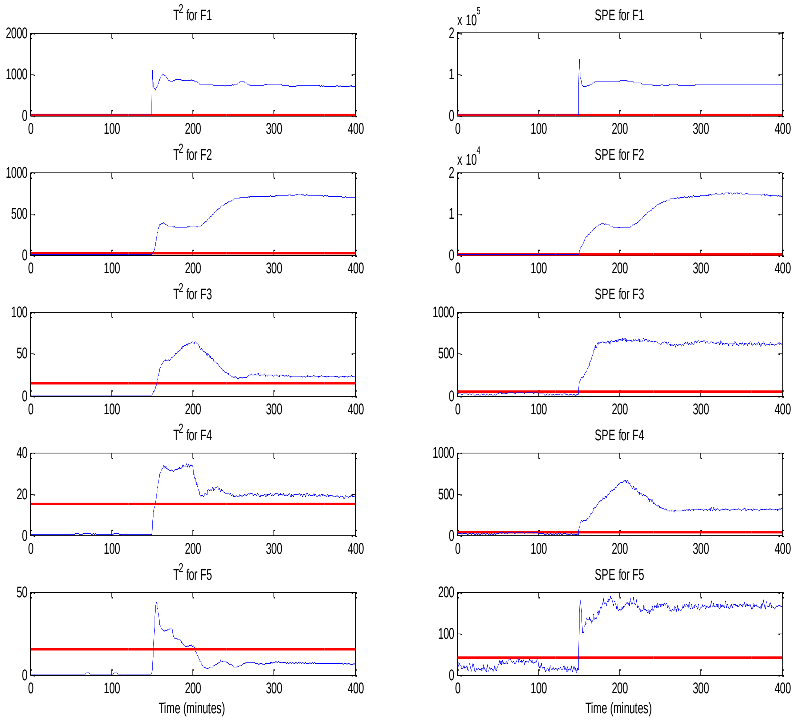


Figure 28. T^2 and SPE plots for faults 1 – 5 (F1 – F5).

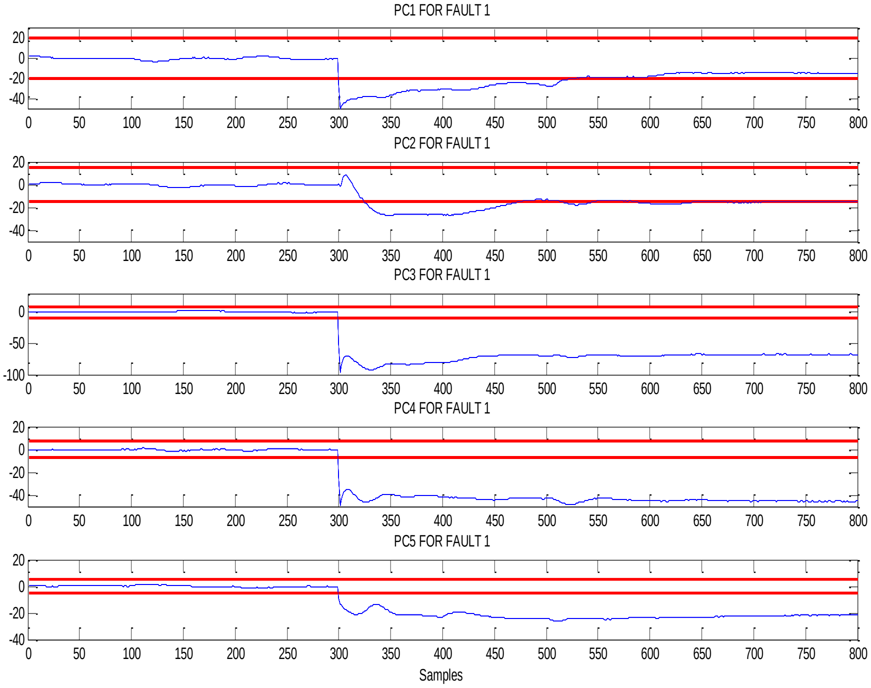


Figure 29. PC plots for fault F1.

The moment an actuator fault is detected, further fault diagnostic is carried out through contribution plots of the monitoring statistics to identify the fault. When the fault is declared, each of the principal components (PC) used to develop the diagnostic model is checked, in this case five principal components, at the point of fault declaration to identify the PC that violates its limit (+/- limit for each PC). An example of the PC plots is shown for fault case F1 in Figure 29. The cumulative effects of the variables responsible for the PCs going outside their bounds are then presented pictorially in the contribution plots. Figures 30 to 34 present the contribution plots of the five fault cases.

4.2.6. Implementation of the Actuator FTC on CDU for the Identified Actuator Faults

The structure of the control system pre-fault era may need to be reconfigured upon declaration and identification of an actuator fault, so as to preserve the integrity of the control system and most importantly ensure the CDU system is continued to be operated safely and economically in spite of the fault. Settings of some of the reconfigured controllers may also need to be adjusted and the back-up feedback signals switched as appropriate. As discussed in the previous sections, not all of the five actuator faults investigated in this case study can be accommodated. This is due to the non-availability of suitable manipulated variable that could be used to accommodate the faults. Faults F1 and F5 could not be accommodated as there are no suitable manipulated variables that could be used to sub-optimally control them. However, fault cases F2 – F4 are sub-optimally accommodated when identified using different manipulated variables as presented in Table 11. To accommodate the F2 actuator fault, the error signal generated by the reconfigurable actuator FTC for the fault-free automated CDU system during normal operation is given as:

$$e=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}\\r_{p2}\\r_{p3}\\r_{p4}\\r_{p5}\\r_{b1}\\r_{b2}\\r_{b3}\\r_{b4}\\r_{b5}\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}y_{p1}\\y_{p2}\\y_{p3}\\y_{p4}\\y_{p5}\\y_{b1}\\y_{b2}\\y_{b3}\\y_{b4}\\y_{b5}\end{bmatrix} \tag{73}$$

where r_{pi} and y_{pi} $(i = 1 \ldots .5)$ are the system reference points and the primary controlled outputs respectively; r_{bi} are the reference points backup signal and y_{bi} are the corresponding outputs backup feedback signals. To accommodate reflux flow actuator fault (F2), the furnace heat flow control valve (u_5) is reconfigured and its controller settings adjusted, as presented in Table 10. The error signal for the reconfigured actuator FTC under F2 fault is obtained as:

$$e_{RF2}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}\\r_{p2}\\r_{p3}\\r_{p4}\\r_{p5}\\r_{b1}\\r_{b2}\\r_{b3}\\r_{b4}\\r_{b5}\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}y_{p1}\\y_{p2}\\y_{p3}\\y_{p4}\\y_{p5}\\y_{b1}\\y_{b2}\\y_{b3}\\y_{b4}\\y_{b5}\end{bmatrix}\qquad(74)$$

The fault tolerant control law to accommodate F2 actuator fault is then given as

$$u_{F2}=\begin{bmatrix}u_1\\u_2\\u_3\\u_4\\u_5\\u_{b1}\\u_{b2}\\u_{b3}\\u_{b4}\\u_{b5}\end{bmatrix}=\begin{bmatrix}G_1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&G_3&0&0&0&0&0&0&0\\0&0&0&G_4&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&G_{b2}&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}-y_{p1}\\0\\r_{p3}-y_{p3}\\r_{p4}-y_{p4}\\0\\0\\r_{b2}-y_{b2}\\0\\0\\0\end{bmatrix}\qquad(75)$$

Similarly, the error signals $(e_{RFi}, i=3{,}4)$ and the fault tolerant control laws (u_{Fi}) under actuator faults F3 and F4 are obtained as

$$e_{RF3}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}\\r_{p2}\\r_{p3}\\r_{p4}\\r_{p5}\\r_{b1}\\r_{b2}\\r_{b3}\\r_{b4}\\r_{b5}\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}y_{p1}\\y_{p2}\\y_{p3}\\y_{p4}\\y_{p5}\\y_{b1}\\y_{b2}\\y_{b3}\\y_{b4}\\y_{b5}\end{bmatrix}\qquad(76)$$

$$u_{F3}=\begin{bmatrix}u_1\\u_2\\u_3\\u_4\\u_5\\u_{b1}\\u_{b2}\\u_{b3}\\u_{b4}\\u_{b5}\end{bmatrix}=\begin{bmatrix}G_1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&G_4&0&0&0&0&0&0\\0&0&0&0&G_5&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&G_{b3}&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}-y_{p1}\\0\\0\\r_{p4}-y_{p4}\\r_{p5}-y_{p5}\\0\\0\\r_{b3}-y_{b3}\\0\\0\end{bmatrix}\qquad(77)$$

$$e_{RF4}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}r_{p1}\\r_{p2}\\r_{p3}\\r_{p4}\\r_{p5}\\r_{b1}\\r_{b2}\\r_{b3}\\r_{b4}\\r_{b5}\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}y_{p1}\\y_{p2}\\y_{p3}\\y_{p4}\\y_{p5}\\y_{b1}\\y_{b2}\\y_{b3}\\y_{b4}\\y_{b5}\end{bmatrix}\qquad(78)$$

$$u_{F4} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_{b1} \\ u_{b2} \\ u_{b3} \\ u_{b4} \\ u_{b5} \end{bmatrix} = \begin{bmatrix} G_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_{b4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{p1} - y_{p1} \\ r_{p2} - y_{p2} \\ 0 \\ 0 \\ r_{p5} - y_{p5} \\ 0 \\ 0 \\ 0 \\ r_{b4} - y_{b4} \\ 0 \end{bmatrix} \tag{79}$$

The weightings for the primary and backup feedback signals are deactivated and activated as appropriate in order to accommodate the identified actuator faults as shown in equations 73 – 79 so as to sub-optimally maintain the system within acceptable operating region and preserve the integrity of both the control system and the process.

4.2.7. Discussion of Results

All the five actuator faults investigated in the dynamic CDU as presented in Table 12 are detected using the DPCA monitoring statistics with data collected during the fault free simulation of the system. Fault case 1 (F1 – bottom boil-up control valve fault) was detected at sample 301, a minute after introduction on both T^2 and SPE monitoring statistics as presented in Figure 28. Fault case 2 (F2 – reflux control valve actuator fault) was detected at sample 305 (2 minutes 30 seconds after introduction) for T^2 monitoring statistic and at sample 302 (1 minute after introduction) for SPE monitoring statistic as presented in Figure 28. Fault case 3 (F3 – SS-2 steam actuator fault) was detected 9 minutes and 1 minute 30 seconds after introduction, at samples 317 and 302 for T^2 and SPE monitoring statistics respectively. Actuator fault F4 (SS-3 steam actuator fault) was detected at samples 315 and 302 for the T^2 and SPE monitoring statistics respectively, 8 minutes and 1 minute 30 second after introduction for the T^2 and SPE monitoring statistics respectively. The fifth actuator fault case (F5 – furnace heat flow actuator fault) was detected 4 minutes and 1 minute 30 seconds after introduction, at samples 308 and 302 for the T^2 and SPE monitoring statistics respectively.

After an actuator fault is declared, the contributions of each variable monitored in the system to the fault recorded are further investigated through the variable contribution plots of the two monitoring statistics. Contributions in excess of normal average variable contributions to the monitoring statistics at the point of fault declaration are examined to identify variables responsible for the fault. Normal average variable contributions are the average contributions recorded for each variable during normal operating conditions. To achieve this, the plot of each PC used to develop the monitoring statistics, in this case 5 PCs are examined to identify the PC that violates their limits. Figure 29 presents the PC plots for fault F1. Table 13 summarises the PCs that violate their bounds (+/- limit for each PC), and the cumulative contributions of the PCs for each actuator fault are presented in Figures 30 to 34.

Figure 30 presents the contribution plots for F1, and observation from the T^2 contribution plot reveals variables 55 and 53, bottom boil-up mass flow (y_1) and the CDU bottom steam flow (u_1) as the two major variables that contributed to the faulty situation. The SPE contribution plot also reveals variables 40, 49 and 69, that is diesel temperature, SS-2 return temperature and AGO feed ratio in addition to the actual controlled variable for the control loop as being responsible for the fault. The fault was easily identified using the two contribution plots due to significantly large changes in the contributions of those variables associated with the control loop, particularly variables 53 and 55. These two variables are directly linked to the faulty actuator. The contribution plots for F2 as presented in Figure 31 show variables 64, 49, 3, 17, 18 and 40; which are temperature of the vapour leaving the column (y_2), SS-2 return temperature, stages 2, 16 and 17 temperatures and diesel temperature respectively as being responsible for the fault according to the T^2 contribution plot. The SPE contribution plot for F2 also shows variable 54; the reflux mass flow (u_2) as variable with significantly large contribution in addition to variables 64, 3, and 49 which were already picked up by the T^2 contribution plots. These variables are closely connected to the faulty control loop. For instance, reduced reflux mass flow (variable 54) is a consequence of the faulty valve which had a direct negative impact on variables 64 and 3 – temperature of stages 1 and 2, leading to the identification of the actuator fault.

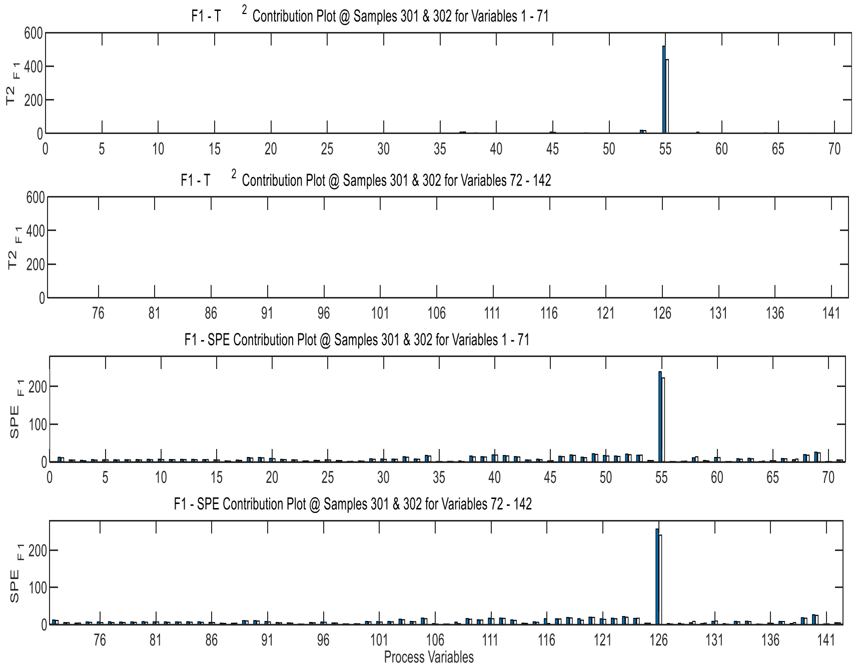


Figure 30. T^2 and SPE excess contribution plots for fault F1.

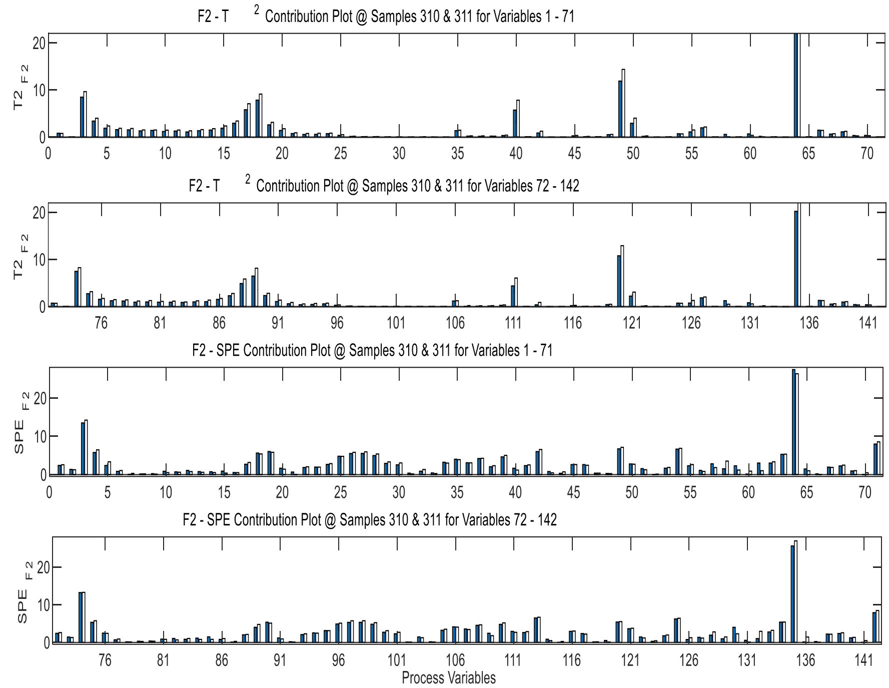


Figure 31. T^2 and SPE excess contribution plots for fault F2.

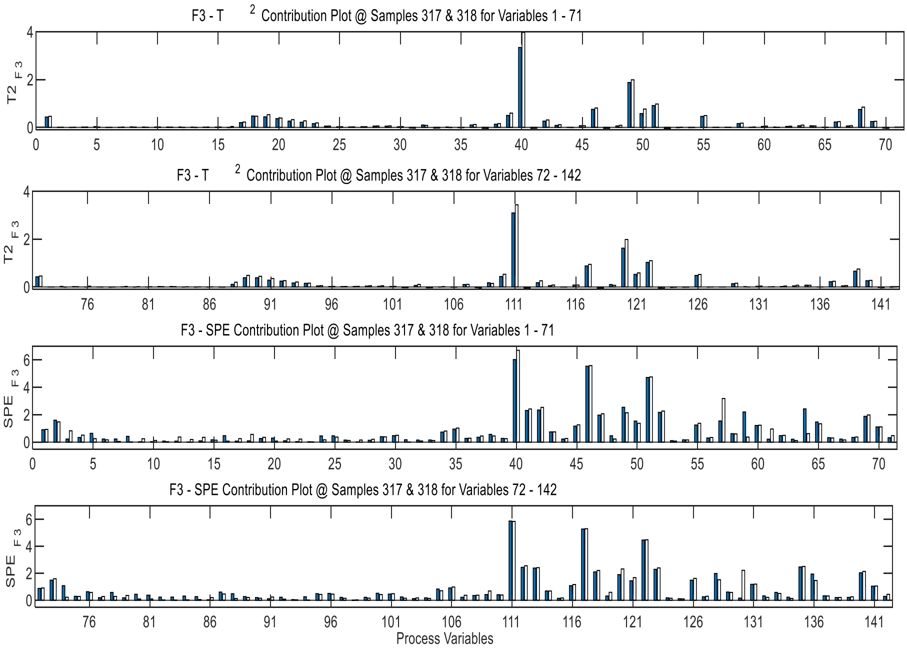


Figure 32. T^2 and SPE excess contribution plots for fault F3.

Table 13. List of PCs that violate their limits for F1 – F5

Faults	Principal Components
F1	PC1, PC3, PC4, PC5
F2	PC1, PC2, PC3, PC4, PC5
F3	PC2, PC3, PC4
F4	PC3, PC4, PC5
F5	PC4, PC5

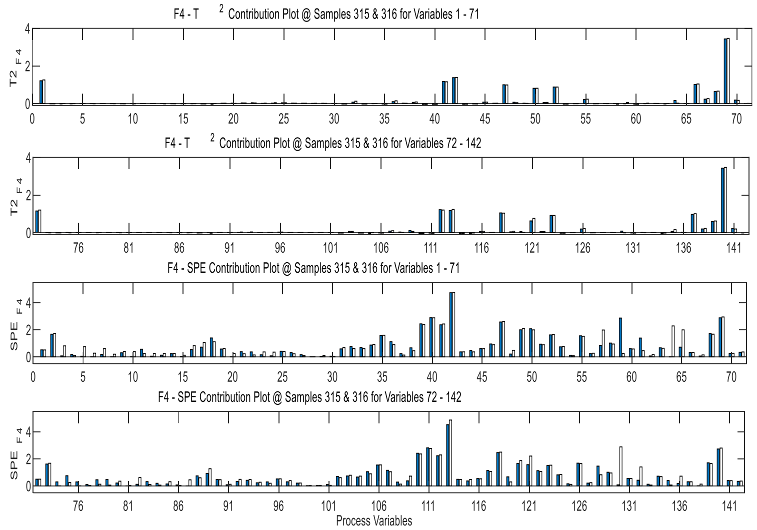


Figure 33. T^2 and SPE excess contribution plots for fault F4.

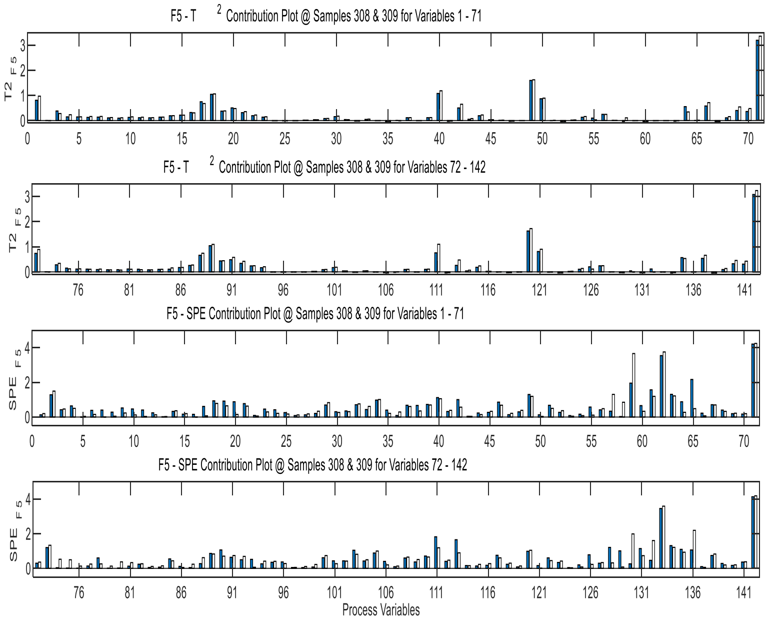


Figure 34. T^2 and SPE excess contribution plots for fault F.

For the fault case 3 (F3), the T^2 contribution plot as shown in Figure 32 identifies variables 40, 49, 46 and 51; which are diesel temperature (y_3), SS-2 return temperature and mass flow, and SS-2 steam flow (u_3) respectively, as the variables with larger than average contributions to the monitoring statistics. The SPE contribution plot also identified the same variables as being responsible for the fault. Again, the identified variables are closely linked to the faulty control valve and can be mapped to the fault. Significant changes in the values of the contributions of these variables to the monitoring statistics are the direct consequence of the actuator fault F3. Figure 33 presents the contribution plots for F4. The T^2 contribution plot reveals variables 1, 69, 41, 42, 47 and 66; the crude oil mass flow, AGO feed ratio, AGO mass flow, AGO temperature (y_4), SS-3 return mass flow and naphtha feed ratio respectively as being responsible for the fault. SPE contribution plot also reveals variables 41, 42, and 69 in addition to variables 40 (diesel temperature), 39 (diesel mass flow), 50 (SS-3 return temperature) and 49 (SS-2 return temperature) as being the contributing variables to the fault. A critical analysis of the effect of fault F4 on those variables shows good cause to associate the variables to the fault as they are closely linked to the faulty control loop. The T^2 and SPE contribution plots for F5 are shown in Figure 34. It reveals variable 71, the furnace heat flow (u_5) to the crude flash line as the major contributor to the fault. The T^2 contribution plot in addition to variable 71 (u_5) also shows variables 40 and 49, diesel temperature and SS-2 return temperature as being contributors while SPE contribution plot reveals variable 62, the crude flash zone temperature (y_5) as another major contributor to the fault recorded. A fault in the furnace heat flow valve directly affects the crude flash zone temperature (y_5), making the variables – fault mapping easily achievable. Adequate knowledge of the system being investigated is still required to make the connections between the variables identified by the contribution plots and the faults declared by the monitoring statistics. Table 14 summarises the list of variables responsible for each fault.

Table 14. Variables responsible for faults F1 – F5

Faults		Variables
F1	T^2	55, 53
	SPE	55, 40, 49, 69
F2	T^2	64, 49, 3, 17, 18, 40
	SPE	64, 3, 71, 49, 54
F3	T^2	40, 49, 46, 51
	SPE	40, 46, 51, 41, 42, 52, 59
F4	T^2	69, 1, 41, 42, 47, 66
	SPE	42, 40, 39, 41, 47, 49, 69
F5	T^2	71, 49, 40, 18, 1
	SPE	71, 62, 59, 2

The identified faults are accommodated according to the possible actuator FTC reconfiguration presented in Table 11. Fault F1 could not be accommodated as there was no suitable manipulated variable that could keep it at set point, even in the absence of any actuator fault, as observed during the system rigorous fault free simulation. When fault F2 (u_2 – faulty reflux flow control valve) is identified and isolated, non-square RGA analysis suggested y_2 (stage 1 temperature) is left uncontrolled and the remaining four controlled loops are maintained. However, further input-output pairing investigation through RGA and DRGA reveals y_2 (stage 1 temperature) could be controlled by manipulating u_5 (furnace heat flow rate) as presented in Table 11. Figures 35 and 36 show the responses of the five controlled variables and the seven products quality variables to the implementation of the actuator FTC on the dynamic CDU system to accommodate F2. The solid blue lines in the figures are the responses of the controlled variables and products quality variables during normal operating conditions; the dashed red lines are the responses of the same variables under faults while the dotted blue lines are the responses of the controlled variables and the product quality variables to the implementation of the actuator FTC strategy. The actuator FTC system is able to accommodate fault F2, maintaining y_2 (stage 1 temperature) at set point and reduced the effects of the fault on other controlled variables while y_5 (crude flash zone temperature) is left uncontrolled, as presented in Figure 35. Similarly, the effects of the fault on the product quality variables are reduced greatly, particularly for the ASTM D1160 cut points at 0% and 100% for kerosene, ASTM D93 flash point for kerosene and diesel, and ASTM D1160 cut points at 90% and 95% for diesel, as presented in Figure 36. No improvement is recorded on the viscosity at 210F for AGO.

Fault F3 (u_3 – faulty SS-2 steam control valve) is accommodated by reconfiguring the actuator FTC using u_2 (reflux flow control valve) to directly maintain y_3 (diesel temperature) at set point and the controller settings tuned as appropriate, as presented in Table 10. Figures 37 and 38 present the responses of the five controlled variables and the seven product quality variables to the implementation of the accommodating actuator FTC. The curves are as previously defined above. The actuator FTC is very effective in accommodating F3 by quickly returning y_3 (diesel temperature) to its set point and reducing the effect of the fault on other controlled variables, except for y_2 (stage 1 temperature) which is uncontrolled, as presented in Figure 37. The strategy is not so effective in reducing the effects of the fault on all the product quality variables as can be observed from Figure 38. The strategy was only able to reduce the fault effect on ASTM D1160 cut points at 90% and 95% for diesel, ASTM D93 flash point for diesel and viscosity at 210F for AGO; while ASTM D1160 cut points at 0% and 100% for kerosene and ASTM D93 flash point for kerosene further drifted away from their respective nominal values. This is because the controller reconfigured to directly maintain y_3 (diesel temperature) at set point is direct acting and increased reflux flow rate (u_2) in order to maintain y_3 as set point. This action led to reduced temperature on the top stages of the column which invariably made the product quality variables to drift further away from their nominal values. This is a decision that will be made based on the economy of the plant.

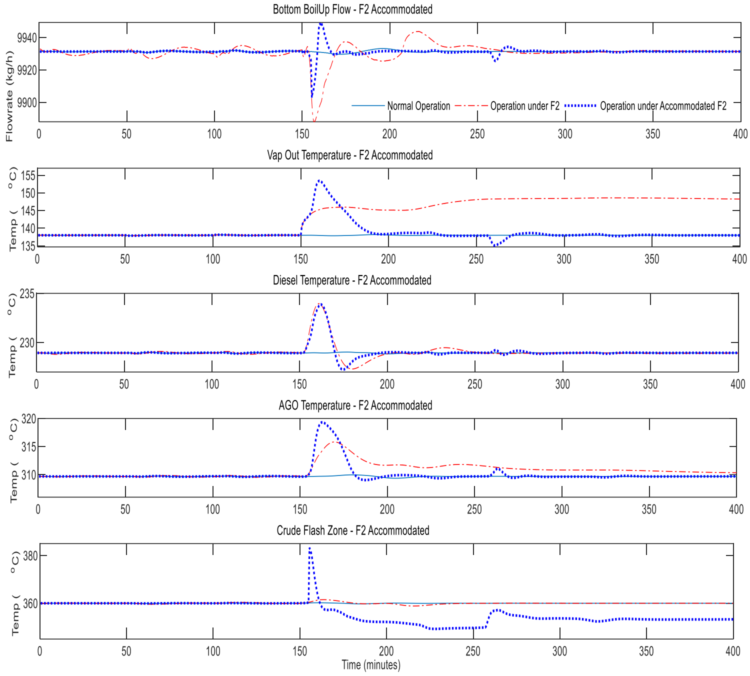


Figure 35. Controlled variables response plot for accommodated F2.

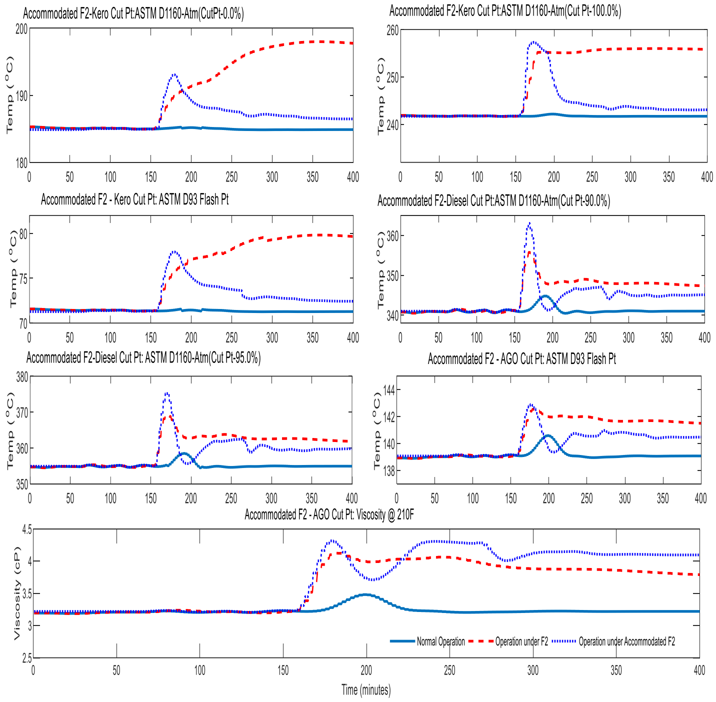


Figure 36. Process quality variables response plot for accommodated F2.

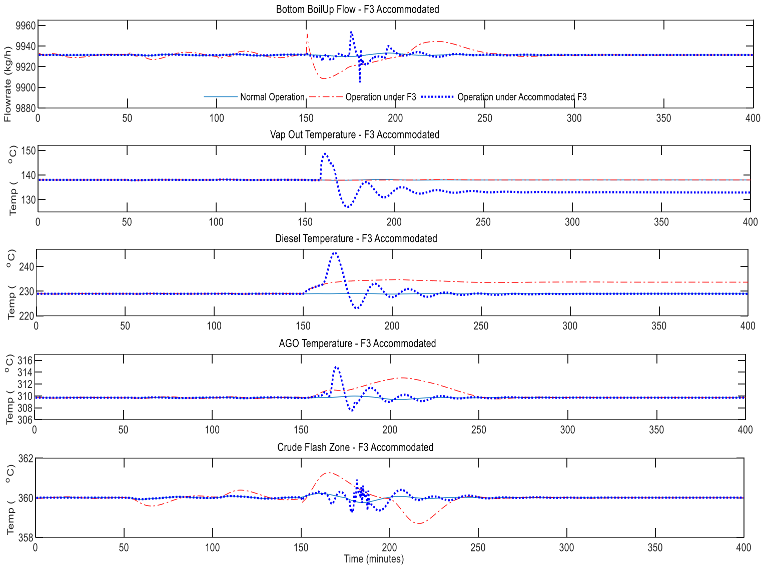


Figure 37. Controlled variables response plot for accommodated fault F3.

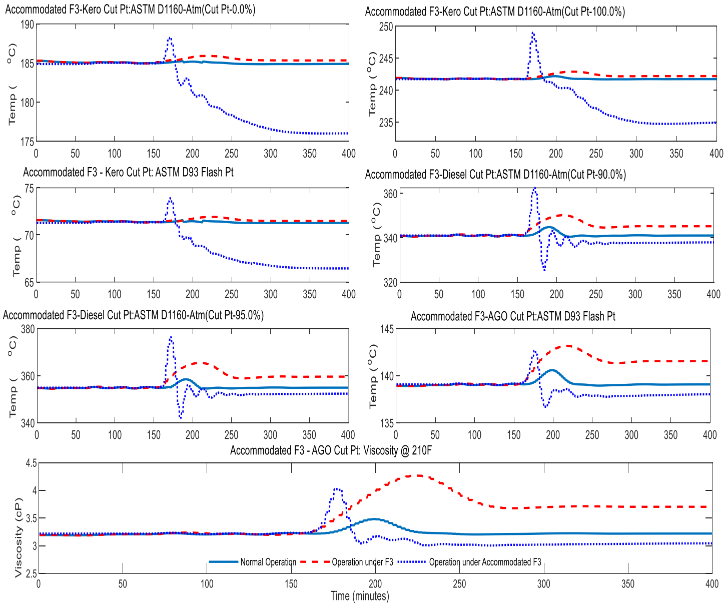


Figure 38. Product quality variables response plot for accommodated fault F3.

Fault case 4 (F4) is accommodated upon identification by reconfiguring the actuator FTC using u_3 (SS-2 steam control valve) to directly maintain y4 (AGO temperature) at set point, leaving y_3 (diesel temperature) uncontrolled. Figures 39 and 40 present responses of the controlled and product quality variables to the actuator fault accommodating strategy respectively. The plots are as previously defined. It can be observed from Figure 39 that the reconfigured actuator FTC was able to maintain y4

(AGO temperature) at set point and reduced the effects of the faults on other controlled variables. However, the results of the accommodating strategy on the product quality variables suggest it is not effective as all the product quality variables further drifted away from their respective nominal values, as presented in Figure 40. The responses of the product quality variables under F4 are better than under the accommodating strategy, and it might be better to leave the fault (F4) uncontrolled as suggested by the inputs – outputs pairing tools, RGA and DRGA. Fault case 5 (F5 – faulty furnace heat flow control valve) could not be accommodated as there is no suitable manipulated variable to reconfigure to keep it at set point.

The actuator FTC works well in situations where there are suitable manipulated variables that could be reconfigured to accommodate the identified actuator fault. However, the reconfiguration is not always possible as observed in this case study where F1 and F5 could not be accommodated due to non-availability of suitable manipulated variable pairing. Hence, the proposed actuator FTC provides an opportunity to sub-optimally maintain the integrity of control systems in the presence of actuator faults by reconfiguring the structure of the control system to minimise the impact of the faults on the system. The sub-optimal actuator FTC strategy is system dependent and it needs no additional hardware but needs the possible control reconfiguration structure pre-assessed and used as back when necessary. The accommodating strategy is not always applicable, not only when there are no suitable manipulated variables, but also when the available pairing during faulty circumstance cannot effectively accommodate the identified fault as evident in the accommodation of F4 (faulty SS-3 steam control valve) using u_3 (SS-2 steam control valve).

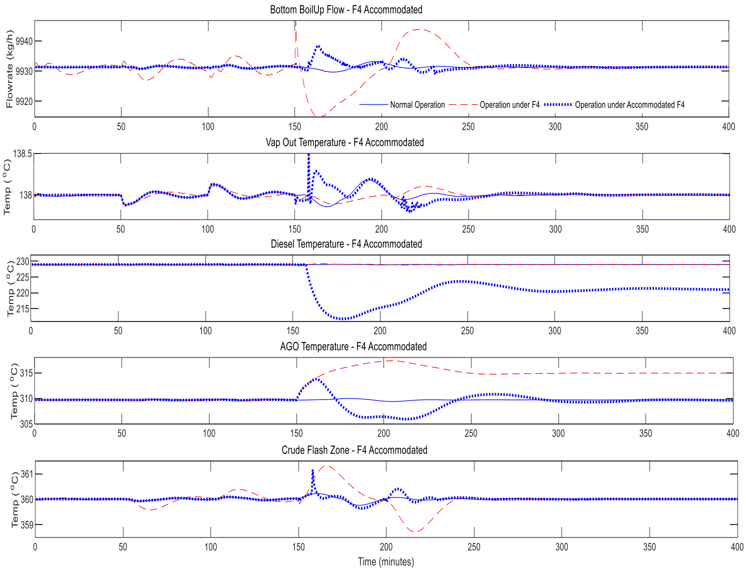


Figure 39. Controlled variables response plot for accommodated fault F4.

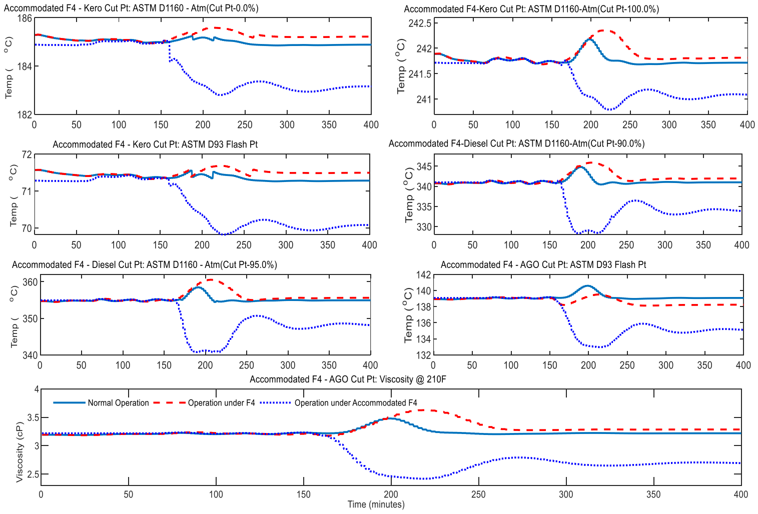


Figure 40. Process quality variables response plot for accommodated fault F4.

Conclusion

This chapter investigated and contributed to the furtherance of the development and application of fault tolerant control system (FTCS) to the oil and gas processes, particularly the distillation processing units. A simple restructurable feedback controller with backup signals and switchable references (actuator FTCS) was developed and applied to two different distillation processes with varying complexities – the Shell heavy oil fractionator and a crude distillation unit to accommodate actuator faults in the systems. The strategy can help improve the availability and performance of control systems in the presence of actuator faults and ultimately prevent avoidable potential disasters in the refinery operation with improved bottom line, profit, by sub-optimally maintaining continued safe operation of the plant during abnormal events.

References

[1] Bureau of Labor Statistics, *Occupational Injuries and Illnesses in the United States by Industry*. 1998, Government Printing Office: Washington, DC.

[2] National Safety Council, *Injury Facts 1999 Edition*. 1999, National Safety Council: Chicago.

[3] McGraw-Hill Economics, *Survey of Investment in Employee Safety and Health*. 1985, McGraw-Hill Publishing Co: New York.

[4] Nimmo, I., Adequately address abnormal situation operations. *Chemical Engineering Progress*, 1995. 91(9): p. 36-45.

[5] Laser, M., *Recent safety and environmental legislation. Transactions of the Institution of Chemical Engineers*, 2000. 78(B): p. 419-422.

[6] Venkatasubramanian, V., R. Rengaswamy, K. Yin, and S. N. Kavuri, A review of process fault detection and diagnosis. Part I. Quantitative Model-Based Methods. *Computers & Chemical Engineering*, 2003. 27(3): p. 293-311.

[7] Lees, F. P., *Loss Prevention in the Process Industries Hazard Identification, Assessment, and Control*. 2nd ed.. ed. 1996, Boston: Boston: Butterworth-Heinemann.

[8] Jackson, B. B., *Union Carbide: Disaster at Bhopal*. 1993, Union Carbide Corporation.

[9] Venkatasubramanian, V., R. Rengaswamy, S. N. Kavuri, and K. Yin, A review of process fault detection and diagnosis Part III: Process history based methods. *Computers & Chemical Engineering*, 2003. 27(3): p. 327-346.

[10] Venkatsubramanian, V., R. Rengaswamy, K. Yin, and S. N. Kavuri, A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 2003. 27(3): p. 293-311.

[11] Chilin, D., J. Liu, X. Chen, and P. D. Christofides, Fault detection and isolation and fault tolerant control of a catalytic alkylation of benzene process. *Chemical Engineering Science*, 2012. 78: p. 155-166.

[12] Zhang, J., Improved on-line process fault diagnosis through information fusion in multiple neural networks. *Computers & Chemical Engineering*, 2006. 30(3): p. 558-571.

[13] Zhang, J., E. B. Martin, and A. J. Morris, Process monitoring using non-linear statistical techniques. *Chemical Engineering Journal*, 1997. 67(3): p. 181-189.

[14] Zhang, Y. and J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 2008. 32(2): p. 229-252.

[15] Blanke, M., M. Kinnaert, J. Lunze, and M. Staroswiecki, Diagnosis and fault-tolerant control. *Robotica*. Vol. 22. 2003, Berlin, Germany: Springer – Verlag. 517.

[16] Blanke, M., M. Staroswiecki, and N. E. Wu, Concepts and methods in fault-tolerant control. *Proceedings of the 2001 American Control Conference*, Vols 1-6, 2001: p. 2606-2620.

[17] Frank, P. M., Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 1990. 26(3): p. 459-474.

[18] Frank, P. M., Analytical and qualitative model-based fault diagnosis – a survey and some new results. *European Journal of Control*, 1996. 2(1): p. 6-28.

[19] Isermann, R., Process fault detection based on modeling and estimation methods—A survey. *Automatica*, 1984. 20(4): p. 387-404.

[20] Patton, R. J. and J. Chen, A Review of Parity Space Approaches to Fault-Diagnosis. *Fault Detection, Supervision and Safety for Technical Processes (Safeprocess 91)*, 1992. p. 65-81.

[21] Patton, R. J. and J. Chen, Observer-based fault detection and isolation: Robustness and applications. *Control Engineering Practice*, 1997. 5(5): p. 671-682.

[22] Sangha, M. S., D. L. Yu, and J. B. Gomm, Robustness assessment and adaptive FDI for car engine. *International Journal of Automation and Computing*, 2008. 5(2): p. 109-118.

[23] Yu, D. L., A. Hamad, J. B. Gomm, and M.S. Sangha, Dynamic fault detection and isolation for automotive engine air path by independent neural network model. *International Journal of Engine Research*, 2014. 15(1): p. 87-100.

[24] Qin, S. J. and T. A. Badgwell, A survey of industrial model predictive control technology. *Control Engineering Practice*, 2003. 11(7): p. 733-764.

[25] MacGregor, J. and A. Cinar, Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers & Chemical Engineering*, 2012. 47: p. 111-120.

[26] Eterno, J. S., J. L. Weiss, D. P. Looze, and A. S. Willsky, Design issues for fault tolerant-restructurable aircraft control, in *Proceedings of the 24th IEEE Conference on Decision and Control*. 1985. p. 900-905.

[27] Blanke, M., M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. 2 ed. 2006, Berlin, Germany: Springer-Verlag Berlin Heidelberg. XIX, 672.

[28] Rauch, H. E., Intelligent fault diagnosis and control reconfiguration. *IEEE Control Systems Magzine*, 1994. 14(3): p. 6-12.

[29] Bruccoleri, M., M. Amico, and G. Perrone, Distributed intelligent control of exceptions in reconfigurable manufacturing systems. *International Journal of Production Research*, 2003. 41: p. 1393-1412.

[30] Chilin, D., J. Liu, J. F. Davis, and P. D. Christofides, Data-based monitoring and reconfiguration of a distributed model predictive control system. *International Journal of Robust and Nonlinear Control*, 2012. 22(1): p. 68-88.

[31] Chilin, D., J. Liu, D. Munoz de la Pena, P. D. Christofides, and J. F. davis, Detection, isolation and handling of actuator faults in distributed model predictive control systems. *Journal of Process Control*, 2010. 20: p. 1059-1075.

[32] Chilin, D., J. F. Liu, D. M. de la Pena, P. D. Christofides, and J. F. Davis, Monitoring and Handling of Actuator Faults in a Distributed Model Predictive Control System. *2010 American Control Conference*, 2010: p. 2847-2854.

[33] Isermann, R., *Fault-diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. 2006, Berlin, Germany: Springer-Verlag Berlin Heidelberg.

[34] Isermann, R., R. Schwarz, and S. Stolzl, Fault-tolerant drive-by-wire systems. *IEEE Control Systems Magazine*, 2002. 22(5): p. 64-81.

[35] Steinberg, M., Historical overview of research in reconfigurable flight control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2005. 219(4): p. 263-275.

[36] Stengel, R. F., Intelligent failure-tolerant control. *IEEE Control Systems Magzine,* 1991. 11(4): p. 14-23.

[37] Zhang, Y. M. and J. Jiang, Fault tolerant control system design with explicit consideration of performance degradation. *IEEE Transactions on Aerospace and Electronic Systems*, 2003. 39(3): p. 838-848.

[38] Diao, Y. and K. M. Passino, Intelligent fault-tolerant control using adaptive and learning methods. *Control Engineering Practice*, 2002. 10(8): p. 801-817.

[39] Lao, L., M. Ellis, and P. D. Christofides, Proactive fault-tolerant model predictive control. *AIChE Journa*l, 2013. 59(8): p. 2810-2820.

[40] Mirzaee, A. and K. Salahshoor, Fault diagnosis and accommodation of nonlinear systems based on multiple-model adaptive unscented Kalman filter and switched MPC and H-infinity loop-shaping controller. *Journal of Process Control*, 2012. 22(3): p. 626-634.

[41] Noura, H., D. Theilliol, and D. Sauter, Actuator fault-tolerant control design: demonstration on a three-tank-system. *International Journal of Systems Science*, 2000. 31(9): p. 1143-1155.

[42] Liang, Y. W., D. C. Liaw, and T. C. Lee, Reliable control of nonlinear systems. *IEEE Transactions on Automatic Control*, 2000. 45(4): p. 706-710.

[43] Hsieh, C. S., Performance gain margins of the two-stage LQ reliable control. *Automatica*, 2002. 38(11): p. 1985-1990.

[44] Veillette, R. J., J. V. Medanic, and W. R. Perkins, Design of reliable control-systems. *IEEE Transactions on Automatic Control*, 1992. 37(3): p. 290-304.

[45] Siljak, D. D., Reliable control using multiple control-systems. *International Journal of Control*, 1980. 31(2): p. 303-329.

[46] Yang, G. H., S. Y. Zhang, J. Lam, and J. L. Wang, Reliable control using redundant controllers. *IEEE Transactions on Automatic Control*, 1998. 43(11): p. 1588-1593.

[47] Yang, G. H., J. L. Wang, and Y. C. Soh, Reliable H(infinity) controller design for linear systems. *Automatica*, 2001. 37(5): p. 717-725.

[48] Zhao, Q. and J. Jiang, Reliable state feedback control system design against actuator failures. *Automatica*, 1998. 34(10): p. 1267-1272.

[49] Yang, G. H. and S. Y. Zhang, Design of reliable control systems by using multiple similar controllers. *Proceedings of the 34th IEEE Conference on Decision and Control*, Vols 1-4, 1995: p. 969-972.

[50] Yu, X. and J. Jiang, A survey of fault-tolerant controller based on safety related issues. A*nnual Review in Control*, 2015. 39: p. 46-57.

[51] Patton, R., Fault-tolerant control system: The 1997 situation, in *Proceedings of SAFEPROCESS*. 1997: Kingston Upon Hull, UK. p. 1033-1054.

[52] Chen, J. and R. J. Patton, Robust model based fault diagnosis for dynamic systems. *Asian Studies in Computer and Information Science*, ed. K. Y. Cai. 1999, Boston, USA: Kluwer Academic Publishers.

[53] Astrom, K. J., Intelligent control, in *Proc. First European Control Conference*. 1991: Grenoble, France. p. 2328-2339.

[54] Himmelblau, D. M., *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*. 1978, New York: Elsevier Scientific Publishing Company.

[55] Isermann, R., Fault diagnosis of machines via parameter estimation and knowledge processing—Tutorial paper. *Automatica*, 1993. 29(4): p. 815-835.

[56] Patton, R. J., Robustness in model-based fault diagnosis: The 1995 situation. *Annual Reviews in Control*, 1997. 21: p. 103-123.

[57] Willsky, A. S., A survey of design methods for failure detection in dynamic systems. *Automatica*, 1976. 12(6): p. 601-611.

[58] Yu, D. L., D. Williams, D. N. Shield, and J. B. Gomm, A parity space method of fault detection for bilinear systems. *Proceedings of the 1995 American Control Conference*, Vols 1-6, 1995: p. 1132-1133.

[59] Dvorak, D. L., *Monitoring and Diagnosis of Continuous Dynamic Systems Using Semi-Quantitative Simulation*. 1992, The University of Texas at Austin, Austin, Texas 78712, USA: Austin, Texas 78712, USA.

[60] Venkatasubramanian, V., R. Rengaswamy, and S. N. Kavuri, A review of process fault detection and diagnosis Part II: Quantitative model and search strategies. *Computers & Chemical Engineering*, 2003. 27(3): p. 313-326.

[61] Zhang, J., E. B. Martin, and A. J. Morris, Fault detection and diagnosis using multivariate statistical techniques. *Chemical Engineering Research & Design*, 1996. 74(A1): p. 89-96.

[62] Tzafestas, S. G., System fault diagnosis using the knowledge-based methodology, in *Fault Diagnosis in Dynamic Systems, Theory and Application*, R. J. Patton, P. M. Frank, and R. N. Clark, Editors. 1989, Prentice Hall. p. 509-572.

[63] Isermann, R. and P. Ballé, Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 1997. 5(5): p. 709-719.

[64] van Schrick, D., *Zustandsschätzer' siemen zur Fehlerkenung, deren Zuverhissigkeit und Anwendung auf den sp gefiuihrten Omnibus*. 1993, University of Wuppertal: Germany.

[65] Chow, E. Y. and A. S. Willsky, *I*ssues in the development of a general algorithm for reliable failure detection, in *Proc. of the 19th Conf. on Decision & Control*. 1980: Albuquerque, NM.

[66] Frank, P. M., Enhancement of robustness in observer-based fault-detection. *International Journal of Control*, 1994. 59(4): p. 955-981.

[67] Gertler, J. J. and M. M. Kunwer, Optimal residual decoupling for robust fault-diagnosis. *International Journal of Control*, 1995. 61(2): p. 395-421.

[68] Patton, R. J. and J. Chen, Optimal selection of unknown input distribution matrix in the design of robust observers for fault-diagnosis. *Fault Detection, Supervision and Safety for Technical Processes (Safeprocess 91)*, 1992. p. 229-234.

[69] Patton, R. J., P. M. Frank, and R. N. Clark, *Fault Diagnosis in Dynamic Systems, Theory and Application*. Control Engineering Series. 1989, New York: Prentice Hall.

[70] Chen, J., *Robust Residual Generation for Model-Based Fault Diagnosis of Dynamic Systems*, in *Department of Electronics*. 1995, University of York, UK: Accessed from: http://etheses.whiterose.ac.uk/2468/ (May, 2014). p. 311.

[71] Viswanadham, N. and R. Srichander, Fault detection using unknown input observers. *Control Theory and Advanced Technology*, 1987. 3(2): p. 91-101.

[72] Beard, R. V., *Failure accommodation in linear systems through self-reorganization*. 1971, Cambridge, Massachusetts: Man Vehicle Lab, MIT.

[73] Watanabe, K. and D. M. Himmelblau, Instrument fault detection in systems with uncertainties. *International Journal of Systems Science*, 1982. 13(2): p. 137-158.

[74] Wang, S. H., E. J. Davison, and P. Dorato, Observing states of systems with unmeasurable disturbances. *IEEE Transactions on Automatic Control*, 1975. 20(5): p. 716-717.

[75] Pearson, K., *On Lines and Planes of Closest Fit to Systems of Points in Space.* Philosophical Magazine, 1901. 2: p. 559-572.

[76] Hotelling, H., Multivariate quality control illustrated by air testing of sample bombsights, in *Techniques of Statistical Analysis*, C. Eisenhart, M. W. Hastay, and W. A. Wallis, Editors. 1947, McGraw Hill: New York. p. 111-184.

[77] Kourti, T. and J. F. MacGregor, Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 1996. 28(4): p. 409-428.

[78] Ku, W. F., R. H. Storer, and C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 1995. 30(1): p. 179-196.

[79] Wold, H., Soft modeling, the basic design and some extensions, in *System under Indirect Observations*, K. Joreskog and H. Wold, Editors. 1982, North Holland: Amsterdam.

[80] Wold, S., K. Esbensen, and P. Geladi, Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 1987. 2(1-3): p. 37-52.

[81] Wold, S., A. Ruhe, H. Wold, and W. J. Dunn, The collinearity problem in linear-regression - the partial least-squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 1984. 5(3): p. 735-743.

[82] Blanke, M., R. Izadi-Zamanabadi, S. A. Bøgh, and C. P. Lunau, Fault-tolerant control systems — A holistic view. *Control Engineering Practice*, 1997. 5(5): p. 693-702.

[83] Camponogara, E., D. Jia, B. H. Krogh, and S. Talukdar, Distributed model predictive control. *IEEE Control Systems Magazine*, 2002. 22(1): p. 44-52.

[84] Mhaskar, P., Robust model predictive control design for fault tolerant control of process systems. *Industrial & Engineering Chemistry Research*, 2006. 45(25): p. 8565−8574.

[85] Zhang, R. D., J. Y. Lu, H. Y. Qu, and F. R. Gao, State space model predictive fault-tolerant control for batch processes with partial actuator failure. *Journal of Process Control*, 2014. 24(5): p. 613-620.

[86] Tao, J., Y. Zhu, and Q. Fan, Improved state space model predictive control design for linear systems with partial actuator failure. *Industrial & Engineering Chemistry Research*, 2014. 53: p. 3578 - 3586.

[87] Garcia, C. E., D. M. Prett, and M. Morari, Model predictive control - theory and practice - a survey. *Automatica*, 1989. 25(3): p. 335-348.

[88] Christofides, P. D., R. Scattolini, D. M. de la Pena, and J. F. Liu, Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 2013. 51: p. 21-41.

[89] Rawlings, J. B. and B. T. Stewart, Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 2008. 18(9): p. 839-845.

[90] Mercangöz, M. and F. J. Doyle III, Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 2007. 17(3): p. 297-308.

[91] Mutambara, A. G. O., *Decentralized Estimation and Control for Multisensor Systems*. 1998, Boca Raton: CRC Press.

[92] Bristol, E., On a new measure of interaction for multivariable process control. *IEEE Trans. Auto. Control*, 1966. AC-11: p. 133 – 134.

[93] McAvoy, T. J., Some results on dynamic interaction analysis of complex control-systems. *Industrial & Engineering Chemistry Process Design and Development*, 1983. 22(1): p. 42-49.

[94] Shinskey, F. G., *Process Control System*. 1988, New York: McGraw-Hill.

[95] Witcher, M. F. and T. J. McAvoy, Interacting control-systems - steady-state and dynamic measurement of interaction. *ISA Transactions*, 1977. 16(3): p. 35-41.

[96] Bristol, E., Recent results on interactions in multivariable process control, in *Presented at 71st Annual AIChE Meeting*. 1979: Houston, TX.

[97] Lawal, S. A. and J. Zhang, Actuator fault monitoring and fault tolerant control in distillation columns. *International Journal of Automation and Computing*, 2017. 14(1): p. 80-92.

[98] Vlachos, C., D. Williams, and J. B. Gomm, Solution to the Shell standard control problem using genetically tuned PID controllers. *Control Engineering Practice*, 2002. 10(2): p. 151-163.

[99] Prett, D. M. and M. Morari., *The Shell Process Control Workshop*. 1987, London: Butterworths. 386.

[100] Yu, X., W. Lu, D. Huang, and Y. Jin, Multi-objective optimization of industrial crude distillation unit based on HYSYS and NSGA-II. *Huagong Xuebao/CIESC Journal*, 2008. 59(7): p. 1646-1649.

[101] Zhou, C., Q. Y. Liu, D. X. Huang, and J. Zhang, Inferential estimation of kerosene dry point in refineries with varying crudes. *Journal of Process Control*, 2012. 22(6): p. 1122-1126.

[102] Lawal, S. A., *Fault Tolerant Control System Design for Distillation Processes*, PhD Thesis. Newcastle University: Newcastle upon Tyne, UK, 2018.

BIOGRAPHICAL SKETCH*ES*

Sulaiman Ayobami Lawal, PhD

Affiliation:

- School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne, UK
- Chemical Engineering Dept., University of Lagos, Lagos, Nigeria

Education:

- BSc Chemical Engineering (2003) from Ladoke Akintola University of Technology, Ogbomoso, Nigeria.
- MSc in Applied Process Control (2006) from Newcastle University, UK.
- MBA (2009) from University of Sunderland, Sunderland, UK.
- PhD (2018) from the School of Engineering, Newcastle University, UK on thesis entitled “Fault Tolerant Control System Design for Distillation Processes.”

Business Address:

Chemical Engineering, School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne, NE1 7RU. Chemical Engineering Department, University of Lagos, Akoka, Yaba, Lagos, Nigeria.

Research and Professional Experience:

1) Research Fellow in Pharmaceutical Process Characterisation and Data Analysis, University of Leeds, UK. Feb. – June 2019.
2) Doctoral Researcher in Fault Tolerant Control System at Chemical Engineering, School of Engineering, Newcastle University, UK. Jan. 2014 – June 2018.
3) Teaching Assistant and Laboratory Demonstrator at the School of Engineering, Newcastle University, UK. Jan 2015 – June 2018.
4) Lecturer at the Chemical Engineering Department, University of Lagos, Nigeria. Aug. 2010 – Present (Current Position).
5) Manager, Operations and Strategy, Slabmark Nigeria Limited, Nigeria. Dec. 2009 – Oct. 2010.
6) Process Engineer at Grand Cereals and Oil Mills Ltd, Jos, Nigeria. Fe. 2004 – Feb. 2005.

Professional Appointments:

Member, Engineering Faculty Board of Studies, University of Lagos. Jan. 2011 – Present.

Member, Engineering Faculty Board of Examiners, University of Lagos, Jan. 2011 – Present.

Reviewer, IEEE Access Journal since 2018.

Reviewer, IEEE Transactions on Industrial Informatics since 2018

Reviewer, IEEE Transactions on Systems, Man and Cybernetics: Systems since 2019

Honors:

- Awarded funds by the Chemical Engineering, School of Engineering, Newcastle University to make a presentation at the CPACT Research Day in Glasgow. (September 2016)
- Won third best presentation at the 2016 Newcastle University Chemical Engineering Annual Postgraduate Research Conference. (July 2016)
- Awarded Partial PhD Scholarship (£26k) by the Petroleum Technology Development Fund (PTDF) for my PhD at Newcastle University, UK. (July 2016)
- Awarded funds by the Chemical Engineering, School of Engineering, Newcastle University to attend an annual CPACT Conference, APACT 16 Conference in Chester, UK. (April 2016)
- Awarded Teaching Assistant Scholarship (£6.5k per year) for the first 3 years of my PhD by the Chemical Engineering, School of Engineering, Newcastle University (January 2014).
- Newcastle University International Postgraduate Scholarship (£1.5k) for my MSc (September 2005).

Publications from the Last 3 Years:

Lawal, S. A. (2018) "Fault tolerant control system design for distillation processes." PhD Dissertation, Newcastle University

Lawal, S. A. and Zhang J. (2017) 'Actuator fault monitoring and fault tolerant control in distillation columns', *International Journal of Automation and Computing*. 14(1), 80–92.

Lawal, S. A. and Zhang J. (2017) 'Actuator and sensor fault tolerant control of a crude distillation unit', in Espuna, A., Graells, M. and Puigianer, L. (eds.) *Computer Aided Process Engineering*. Elsevier, pp. 1705–1710.

Lawal, S. A. and Zhang J. (2016) 'Fault monitoring and fault tolerant control in distillation columns', *Proceedings of the 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 865–870 .

Lawal, S. A. and Zhang J. (2016) 'Sensor Fault Detection and Fault Tolerant Control of a Crude Distillation Unit', in Kravanja, Z. and Bogataj, M. (eds.) *26th European Symposium on Computer Aided Process Engineering*. Amsterdam: Elsevier Science Bv, pp. 2091–2096.

Jie Zhang, PhD

Affiliation: School of Engineering, Newcastle University

Education:

1) September 1982 - July 1986, Division of Industrial Automatic Control, Department of Electrical Engineering, Hebei University of Technology, Tianjin, China, undergraduate studies (BSc) majored in Automatic Control.
2) September 1986 - November 1986, Division of Industrial Automatic Control, Department of Electrical Engineering, Hebei University of Technology, Tianjin, China, postgraduate studies (MSc). (Study terminated by an offer to study in the U.K.)
3) November 1986 - July 1987, Shanghai International Studies University, Shanghai, China, English training.
4) October 1987 - March 1991, Control Engineering Centre, Department of Electrical Electronic and Information Engineering, City University, London, PhD research in Control Engineering.

Business Address:

School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

Research and Professional Experience:

1) May 1991 - March 1996, Research Associate, Department of Chemical & Process Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom.
2) April 1996 - July 2007, Lecturer, School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom.
3) August 2007 – July 2018, Senior Lecturer, School of Chemical Engineering and Advanced Materials, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom.
4) August 2018 - present, Reader in Process Systems Engineering, School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom.

Professional Appointments:

Senior Member of IEEE (elected in August 2002), member of IEEE Computational Intelligence Society, IEEE Control System Society, and IEEE Industrial Electronics Society.

Member of the executive committee of the Chinese Automation and Computer Society in UK (since 1995), General Secretary and President Elect (2017-2018), President (2018-2019).

Honors:

Senior member of IEEE.

Associate editors for the following journals: *Neurocomputing*, *International Journal of Automation and Computing*, *PLOS ONE*, *Open Engineering*

Publications from the Last 3 Years:

1) Lawal, S. A. and Zhang, J. “Actuator Fault Monitoring and Fault Tolerant Control in Distillation Columns,” *International Journal of Automation and Computing*, 2017, Vol. 14, No. 1, PP80-92.
2) Ahmad, Z., Rahim, N. A., Bahadori, A. and Zhang, J. “Improving Water Quality Index Prediction in Perak River Basin Malaysia through the Combination of Multiple Neural Networks,” *The International Journal of River Basin Management*, 2017, Vol.15, No.1, PP79-87.
3) Li, F., Zhang, J., Oko, E. and Wang, M. “Modelling of a Post-combustion CO2 Capture Process Using Extreme Learning Machine,” *International Journal of Coal Science and Technology*, 2017, Vol.4, No.1, PP33-40.

4) Ahmad, Z., Rahim, N. A., Bahadori, A. and Zhang, J. "Air pollution index prediction using multiple neural networks," *IIUM Engineering Journal*, 2017, Vol.18, No.1, PP1-12.
5) See, L. F., Rahim, N. A., Ahmad, Z. and Zhang, J. "Selective combination in multiple neural networks prediction using independent component regression approach," *Chemical Engineering Research Bulletin*, 2017, Vol.19, PP12-19.
6) Osuolale, F., and Zhang, J. "Thermodynamic Optimization of Atmospheric Distillation Unit," *Computers & Chemical Engineering*, 2017, Vol. 103, PP201-209.
7) Gao, X., Zhang, J., Yang, F., Shang, C., and Huang, D. "Robust Proportional−Integral−Derivative (PID) Design for Parameter Uncertain Second-Order Plus Time Delay (SOPTD) Processes Based on Reference Model Approximation," *Ind. Eng. Chem. Res.*, 2017, Vol. 56, No. 41, PP11903-11918.
8) Li, F., Zhang, J., Shang, C., Huang, D., Oko, E., and Wang, M. "Modelling of a Post-combustion CO2 Capture Process Using Deep Belief Network," *Applied Thermal Engineering*, 2018, Vol. 130, PP997-1003.
9) Ahmad, Z., Bahadori, A., Zhang, J. "Prediction of equilibrium water dew point of natural gas in TEG dehydration systems using Bayesian Feedforward Artificial Neural Network (FANN)," *Petroleum Science and Technology*, 2018, Vol. 36, No. 20, PP1620-1626.
10) Zhou, Z., Du, N., Xu, J., Li, Z., Wang, P. and Zhang, J. "Randomized KPCA for Modeling and Monitoring of Nonlinear Industrial Processes with Massive Data," *Ind. Eng. Chem. Res.*, 2019, Vol. 58, No.24, PP10410-10417.
11) Leong, W. C., Bahadori, A., Zhang, J. and Ahmad, Z. "Prediction of Water Quality Index (WQI) using Support Vector Machine (SVM) and Least Square-Support Vector Machine (LS-SVM)," *The International Journal of River Basin Management*, 2019, Vol.17, No., PP-.
12) Yang, Q., Zhang, J. and Yi, Z. "Predicting molten steel endpoint temperature using a feature-weighted model optimized by mutual learning cuckoo search," *Applied Soft Computing*, 2019, Vol. 83, article 105675 (16 pages).
13) Tian, Y., Zhang, J., Chen, L., Geng, Y. and Wang, X. "Single wearable accelerometer-based human activity recognition via kernel discriminant analysis and QPSO-KELM classifier," *IEEE Access*, 2019, Vol.7, PP109216-109227.
14) Tian, Y., Zhang, J., Chen, L., Geng, Y. and Wang, X. "Selective ensemble based on extreme learning machine for sensor-based human activity recognition," *Sensors*, 2019, Vol.19, article 3468 (24 pages).
15) Zhu, C. and Zhang, J. "Developing Soft Sensors for Polymer Melt Index in an Industrial Polymerization Process using Deep Belief Networks," *International Journal of Automation and Computing*, 2019, in press.

16) Qiu, W., Xiong, Z., Zhang, J., Hong, Y., Li, W. “Integrated Predictive Iterative Learning Control Based On Updating Reference Trajectory for Point-to-Point Tracking,” *Journal of Process Control*, 2019, in press.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8

Chapter 2

MODEL-BASED FAULT-TOLERANT CONTROL FOR DISTRIBUTED SYSTEMS

Hasan Ferdowsi[1,*], Jia Cai[2] and Sarangapani Jagannathan[2]
[1]Department of Electrical Engineering,
Northern Illinois University, DeKalb, IL, US
[2]Electrical and Computer Engineering Department,
Missouri University of Science and Technology, Rolla, MO, US

ABSTRACT

Faults are inevitable and even incipient faults that progress very slowly can downgrade the performance of the system. In cases where a fault is not critical, the system performance can be kept at an acceptable level by mitigating the effect of fault. In this chapter, model-based fault-tolerant control and fault accommodation algorithm are presented for two challenging classes of distributed systems; first a spatially distributed system that can be decomposed into interconnected subsystems, and second a distributed parameter system where the system state is distributed over a continuous range of space. The design of a decentralized fault tolerant controller (DFTC) is presented for interconnected nonlinear continuous-time systems by using local subsystem state vector alone in contrast with traditional distributed fault tolerant controllers or fault accommodation schemes where the measured or estimated state vector of the overall system is needed. The decentralized controller uses local state and input vectors in each subsystem and minimizes the fault effects on the entire system. The DFTC in each subsystem includes a traditional controller and a neural network based online approximator which is used to deal with the unknown parts of the system dynamics, such as fault and interconnection terms. The stability of the overall system with DFTC is investigated by using Lyapunov approach and the boundedness of all signals is

[*] Corresponding Author's Email: hferdowsi@niu.edu.

guaranteed in the presence of a fault. Therefore, the proposed controller enables the system to continue its normal operation after the occurrence of a fault, as long as it does not cause failure or break- down of a component. Next, a model-based fault accommodation scheme is introduced for a class of linear distributed parameter systems (DPS) represented by partial differential equations (PDEs) in the presence of both actuator and sensor faults. A filter-based observer on the basis of the linear PDE model of the DPS is designed with output measurements. The estimated output from the observer and the measured outputs are utilized to generate a residual for fault detection. Upon detection, the fault function is estimated by using an unknown parameter vector and a known basis function. Update laws are introduced to estimate the unknown fault parameter vector for actuator and sensor faults. These estimates will then be used to modify the nominal controller in order to accommodate the actuator and sensor faults.

Keywords: fault-tolerant control, fault detection, fault accommodation, adaptive estimation, distributed systems

1. INTRODUCTION

In modern control systems, reliability is as important as maintaining performance. System failures due to unex-pected faults or degradation of the system components may cause a change in the system dynamics leading to un-reliable operation. Faults are inevitable part of any industrial system and can lead to catastrophic failures if not detected and accommodated in the early stages. Therefore, fault accommodation or fault-tolerant control research, which is introduced to mainly mitigate the effect of unexpected incipient faults, has attracted attention [1]. Although certain faults are critical and force the overall system to shut down, other faults at an incipient stage can be accommodated for a limited time, allowing the system to continue operating with an acceptable performance in presence of fault. This is normally done by modifying the control input after detection of fault which is generally called fault accommodation [2] or by using adaptive terms to learn and cancel the fault in the system dynamics without the need for detecting the fault, which is commonly refered to as fault-tolerant control [3-6]. Both of these methods are covered in this chapter. On the other hand, there are many different types of systems that would require such a framework. In this chapter, the focus will be on distributed systems. Two sub-categories of distributed systems will be targeted; first a spatially distributed system that can be decomposed into interconnected subsystems, and second a distributed parameter system where the system state is distributed over a continuous range of space.

Many industrial systems such as power or water distribution networks, telecommunication networks, and many others are complex, large-scale, spatially distributed and interconnected nonlinear systems. The decentralized control for such systems has been an interesting research topic over the past decade, mainly because transmission of data between numerous subsystems in an spatially distributed system is not always possible or can be very expensive both in terms of initial setup and the maintenance costs, not to mention faults or issues that can be initiated from the transmissions. Theoretically speaking, the best performance can be achieved when interconnections between all subsystems are known and measurements from all subsystems are available for controller design in each subsystem. But in a practical scenario, the problems and limitations mentioned above can make centralized controllers undesirable for large-scale systems. This is the motivation behind the development of decentralized controllers [7-10] for such systems. Several distributed fault accommodation schemes and fault tolerant controllers [11-15] have been introduced for such interconnected systems. However, they require either entire state or estimated state vector for each subsystem, since these are merely distributed schemes. As mentioned earlier, it is not always possible to provide the information of the entire state or its estimated value for subsystems of a large-scale spatially distributed system. Even if this is possible, the information will be delayed and outdated, besides being expensive.

In contrast to the aforementioned fault accommodation and fault-tolerant schemes [11-15], in the first part of this chapter, the objective is to design a decentralized fault tolerant controller (DFTC) without the need for interconnection dynamics to be known or the entire system states to be available at all subsystems. For this purpose, the control input at each subsystem is constructed by using the known parts of the subsystem dynamics, stabilizing terms, and a neural network (NN) based online approximator (OLA). The OLA is always active and is utilized to deal with the unknown parts of the system dynamics such as the interconnection term. However, it will not approximate the interconnection term, since the interconnection function depends on nonlocal system state vector, which is not available for the controller design. Instead, all the online approximators in all subsystems can together mitigate the effect of interconnections. Once a fault happens in the system, the output of the OLA in the faulty subsystem will also include an estimation of the fault dynamics, while the OLAs in other subsystems will include the estimation of the fault effect that is propagated to those subsystems through interconnections. Since OLAs are incorporated in the controllers, the effect of fault is mitigated in all subsystems.

The main advantages of the DFTC method include guaranteed stability regardless of the presence of fault, reducing the downtime and improving the performance of the system in the presence of fault without the need for a human operator, ability to perform in the presence of uncertainties as well as unknown interconnection dynamics, and most importantly achieving all of the above by using only the local dynamics and local measurements at each subsystem of a possibly large-scale and spatially distributed system, which will in turn reduce the complexity, cost, and errors associated with modeling interconnections and continuous transmission of large amounts of data across the network.

Further, it is shown that the same DFTC structure can be used for small-scale systems where the state vector is available at each subsystem. In that case, interconnection terms can be used in the controllers to provide better performance. As a result the OLA in the faulty subsystem will provide an estimate of the fault function. Estimated fault dynamics can be very useful in root cause analysis of the fault as well as failure prediction which are discussed in detail in [16, 17].

In the second part of this chapter, the focus will be shifted to distributed parameter systems. In the past two decades, significant level of effort is introduced in the literature [18, 19] on model-based diagnosis and fault-tolerant control of lumped parameter systems (LPS) represented as ordinary differential equations (ODEs). However, many industrial systems such as fluid flows, thermal convection and chemical reaction systems are categorized as distributed parameter systems (DPS) or infinite dimensional systems because the system state changes not only with time but also with space. Fault detection and accommodation for DPS represented by partial differential equations (PDEs) is more involved and challenging when compared to LPS due to the need to estimate the system state at all locations [20-23]. It is not possible to measure the system state of a DPS at all these locations. Though under certain assumptions, the DPS are represented as LPS, the ODE models from LPS representation [24] are no longer suitable to mimic the behavior of DPS accurately.

Because of the distributed nature and complicated dynamics, limited effort is being reported for fault detection and diagnosis of DPS. Recently, fault detection of mechanical and aerospace engineering systems have been studied in [25] and fault tolerant controller was considered in [26-28] with actuator faults. Besides, an adaptive fault detection and accommodation scheme is presented in [29] in order to deal with incipient actuator faults. On the other hand, fault-tolerant control of DPS with control constraints and actuator faults is introduced in [30]. In spite of these exciting results, the detection and accommodation schemes in [26-30] have been developed based on approximate finite dimensional representation of DPS which may lead to false and missed alarms because of the model reduction. Moreover, the system dynamics change in the presence of a fault and thus reduced order models may be inaccurate for fault detection and accommodation in DPS.

Driven by these model reduction considerations, a novel fault accommodation scheme on the basis of the PDE representation for linear DPS with incipient faults is presented in this chapter. A filter-based observer is introduced for generating a residual which is utilized for fault detection. Next the approximation of fault dynamics is carried out by using an adaptive term under the assumption that the fault function is expressed as linear in the unknown parameters. This adaptive term is added to the filter-based observer upon detecting the fault. Both actuator and sensor faults are considered and suitable parameter tuning scheme using the output measurements alone is derived. Then, the fault accommodation is introduced on the basis of estimated fault function. The system stability is demonstrated through Lyapunov analysis. Moreover, upon detecting a fault and by using the tracking error dynamics, estimated time to accommodation (TTA), which is defined as the time needed by the accommodation scheme to recover back to the normal operating regime, can be assessed online. The TTA is particularly useful when compared to the remaining useful life, since it can predict whether or not the accommodation scheme will work before the system reaches failure.

2. METHODS

2.1. Decentralized Fault-Tolerant Control of Spatially Distributed Systems

2.1.1. System Description

Consider a nonlinear continuous-time system that is comprised of N interconnected subsystems. The *i*th subsystem with n_i states is described by

$$\begin{aligned} \dot{x}_{ij}(t) &= x_{i(j+1)}(t) \qquad\qquad j = 1,2,\ldots,n_i - 1 \\ \dot{x}_{in_i}(t) &= f_i(x_i(t)) + g_i(x_i(t))u_i(t) + \omega_i(x(t)) \\ &\quad + \eta_i(x(t)) + \Omega_i(t-t_0)h_i(x_i(t)) \\ y_i(t) &= x_{i1}(t) \end{aligned} \tag{1}$$

where $u_i \in \mathbb{R}$ is the local control input of subsystem i, $x_i \in \mathbb{R}^{n_i}$ is the local state vector of subsystem i, $x = \bigcup_{i=1}^{N} x_i$ is the collection of all subsystem state vector, and $y_i \in \mathbb{R}$ is the output of subsystem i. Moreover, the functions $f_i\colon \mathbb{R}^{n_i} \to \mathbb{R}$ and $g_i\colon \mathbb{R}^{n_i} \to \mathbb{R}$ describe the known local dynamics, $\omega_i\colon \mathbb{R}^{n_i} \to \mathbb{R}$ represent the interconnection dynamics, $\eta_i\colon \mathbb{R}^{n_i} \to \mathbb{R}$ denotes the system uncertainties, and $h_i\colon \mathbb{R}^{n_i} \to \mathbb{R}$ is the local fault function. The time profile of a fault $\Omega_\mathrm{i}(t - t_0)$ is modeled by

$$\Omega_i(t-t_0) = \begin{cases} 0, & if\ t < t_0 \\ 1 - e^{-\bar{\kappa}_i(t-t_0)}, & if\ t \ge t_0 \end{cases} \tag{2}$$

where $\bar{\kappa}_i$ is an unknown constant that represents the rate at which a fault occurs. A larger values of $\bar{\kappa}_i$ indicates that the fault is close to an abrupt fault while small values of $\bar{\kappa}_i$ indicate that the fault is of an incipient type. The use of such time profiles is common in the fault diagnosis literature [2, 13, 17]. Next standard assumptions common in the literature are presented.

Assumption 1: The modeling uncertainty is bounded, i.e., $\|\eta_i(x(t))\| \le \eta_{i_M}, \forall(x,u) \in (\chi \times U), i = 1,2,\ldots,N$, where η_{i_M} is a known positive constant.

Assumption 2: The fault functions can be expressed as nonlinear in the unknown parameters (NLIP) [31]. The NLIP representation for fault functions allows the use of two-layer NNs with nonlinear activation functions.

2.1.2. Decentralized Fault-Tolerant Controller

In theory, a centralized controller can provide desired performance when the interconnection dynamics are known. However, it might be far from a practical solution for large-scale systems due to the difficulties and problems arising from the need for continuous transmission of data between spatially distributed subsystems or between the central controller and all the subsystems. In such cases, a decentralized controller which can provide satisfactory performance is desirable to avoid additional cost and problems associated with the transmission of large amounts of data over the entire network of subsystems.

Centralized controllers are preferred in the case of small-scale systems with small number of states and control inputs, mainly because long distance transmission of large amounts of data would not be required. In this chapter the fault tolerant controller is designed based on a distributed structure. However, it can also be useful for small-scale systems which provide the luxury of using the entire system state vector at all subsystems. In that case, the interconnection dynamics will be included in the controller design which will result in better performance. Based on the above discussion, the DFTC is first designed in the general case where the control input only utilizes local measurements. Then the controller is revisited and customized for the special case where the interconnection terms are available.

2.1.2.1. DFTC Design: General Case

The fault tolerant controller has two main objectives. The first objective is to stabilize the system and make the outputs track their desired trajectories in healthy operating conditions. The second objective is to keep the performance of the system at a satisfactory level in faulty conditions by mitigating the effect of fault through the use of adaptive terms. A fault is normally initiated in one subsystem, but can propagate to neighbor subsystems through the interconnection between the subsystems. Therefore, this fault accommodation has to be performed not only in the faulty subsystem but also in all the other subsystems.

Suppose that the desired output trajectory for subsystem i is given by $y_{id}(t)$ which is continuous and n_i times differentiable. Now the output error in subsystem i can be defined by $e_{i1} = y_i - y_{id}$ and subsequently the error dynamics in healthy conditions can be described by

$$\begin{aligned}
\dot{e}_{ij}(t) &= e_{i(j+1)}(t) \qquad\qquad j=1,2,\ldots,n_i-1\\
\dot{e}_{in_i}(t) &= \dot{x}_{in_i}(t)-y_{id}^{(n_i)}(t)\\
&= f_i(x_i(t))+g_i(x_i(t))u_i(t)+\omega_i(x(t))+\eta_i(x(t))-y_{id}^{(n_i)}(t)
\end{aligned} \tag{3}$$

where $e_{ij}=x_{ij}-y_{id}^{(j-1)}$. In order to write the error dynamics in vector form, define the tracking error vector $e_i=[e_{i1},\ldots,e_{in_i}]^T$ and let A_i and B_i be defined as

$$A_i=\begin{bmatrix}0&1&0\ldots&0\\0&0&1\ldots&0\\\vdots&\vdots&\vdots\ \ddots&\vdots\\0&0&0\ldots&1\\0&0&0\ldots&0\end{bmatrix}\ B_i=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix} \tag{4}$$

Then tracking error dynamics in healthy conditions can be rewritten as

$$\dot{e}_i(t)=A_ie_i(t)+B_i\left[f_i(x_i(t))+g_i(x_i(t))u_i(t)+\omega_i(x(t))+\eta_i(x(t))-y_{id}^{(n_i)}(t)\right] \tag{5}$$

Subsequently, the error dynamics in the presence of fault are described by

$$\begin{aligned}
\dot{e}_i(t)=A_ie_i(t)+B_i\big[&f_i(x_i(t))+g_i(x_i(t))u_i(t)+\eta_i(x(t))\\
&+\omega_i(x(t))+h_i(x_i(t))-y_{id}^{(n_i)}(t)\big]
\end{aligned} \tag{6}$$

By assuming that $\omega_i(x(t))$ and $h_i(x_i(t))$ are both available, the control input $u_i(t)$ can be easily defined by

$$\begin{aligned}
u_i(t)=\ &g_i(x_i(t))^{-1}\left(y_{id}^{(n_i)}(t)-K_i\,e_i(t)-f_i(x_i(t))\right)\\
&+g_i(x_i(t))^{-1}\left(-h_i(x_i(t))-\omega_i(x(t))\right)
\end{aligned} \tag{7}$$

Applying this controller to the error dynamics in (6) leads to

$$\begin{aligned}
\dot{e}_i(t)&=A_ie_i(t)+B_i\left[-K_i\,e_i(t)+\eta_i(x(t))\right]\\
&=(A_i-B_iK_i)e_i(t)+B_i\eta_i(x(t))
\end{aligned} \tag{8}$$

Therefore, with bounded uncertainties, this controller can keep the tracking error bounded if $K_i\in\mathbb{R}^{1\times n_i}$ is selected such that $A_i-B_iK_i$ is Hurwitz. However, $h_i(x_i(t))$ represents the unknown fault dynamics which is uncertain term in the control input. Moreover, decentralized controller can only use the local measurements, hence the interconnection term $\omega_i\big(x(t)\big)$, which includes nonlocal states, cannot be used in the controller development. In order to deal with the unknown terms ω_i and h_i in the subsystem dynamics, an online approximator along with a stabilizing term are added to the control input as follows

$$\begin{aligned} u_i(t) = {} & g_i\left(x_i(t)\right)^{-1}\left(y_{id}^{(n_i)}(t) - K_i\, e_i(t) - f_i\left(x_i(t)\right)\right) \\ & - g_i\left(x_i(t)\right)^{-1}\left(\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right) + l_i B_i^T P_i e_i(t)\right) \end{aligned} \tag{9}$$

where $\hat{q}_i\colon \mathbb{R}^{2n_i} \times \mathbb{R}^{p_i} \to \mathbb{R}$ is the online approximator which will later be defined by a feed-forward neural network (NN), $\hat{\theta}_i \in \mathbb{R}^{p_i}$ is the vector of unknown parameters (neural network weights) tuned by an adaptive update law obtained through Lyapunov design, l_i is a positive constant, and $P_i \in \mathbb{R}^{n_i \times n_i}$ is a symmetric positive definite matrix obtained from the Lyapunov equation $P_i(A_i - B_iK_i) + (A_i - B_iK_i)^T P_i = -Q_i$ where $Q_i \in \mathbb{R}^{n_i \times n_i}$ is an arbitrary symmetric positive definite matrix.

Substituting the control input (9) into the error dynamics in (6), yields

$$\begin{aligned} \dot{e}_i(t) = {} & A_i e_i(t) + B_i\left[\omega_i\left(x(t)\right) + h_i\left(x_i(t)\right) + \eta_i\left(x(t)\right)\right] \\ & - B_i\left[K_i\, e_i(t) + l_i B_i^T P_i e_i(t) + \hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right)\right] \end{aligned} \tag{10}$$

Further, define a scalar error signal $s_i(t) = B_i^T P_i e_i(t)$ to be used in the stability analysis. Next Assumption 3 is introduced which is a common assumption in the decentralized control literature [8, 32, 33], and then the system stability is discussed in Theorem 1.

Assumption 3: The interconnections are unknown but can be expressed as $\omega_i\big(x(t)\big) \le \varsigma_{i0} + \sum_{j=1}^{N} \varsigma_{ij}\big(|s_j(t)|\big), i = 1, 2, \dots, N$, where $\varsigma_{i0}\; for\; i = 1,2,\dots,N$ are unknown constants and $\varsigma_{ij}(.) for\; i, j = 1, 2, \dots, N$ are unknown smooth functions, such that $\varsigma_{ij}(0) = 0$.

Theorem 1 (Performance of the Decentralized Fault Tolerant Controller): Consider the large-scale interconnected system described by (1) with the decentralized local control inputs in the form of (9). If the vectors K_i are chosen such that $A_i - B_iK_i$ is Hurwitz for $i = 1, \dots, N$ and the NN based online approximator is defined by

$$\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right) = \hat{\theta}_i^T(t)\psi_i\left(x_i(t), e_i(t)\right) \tag{11}$$

with parameter update law selected as

$$\dot{\hat{\theta}}_i(t) = \alpha_i \psi_i\left(x_i(t), e_i(t)\right) s_i(t) - \gamma_i \hat{\theta}_i(t) \tag{12}$$

where α_i and γ_i are positive constants, Then the tracking errors $e_i(t)$, and parameter estimation errors $\tilde{\theta}_i(t) = \theta_i - \hat{\theta}_i(t)$ are uniformly ultimately bounded (UUB), if the design parameters are selected such that

$$l_i > 3/2 \qquad for\ i = 1, \ldots, N \tag{13}$$

Proof: Consider the Lyapunov function candidate as $V(t) = \sum_{i=1}^{N} V_i(t)$ where

$$V_i(t) = \frac{1}{2} e_i^T(t) P_i e_i(t) + \frac{1}{2\alpha_i} \tilde{\theta}_i^T(t) \tilde{\theta}_i(t) \tag{14}$$

Then the derivative of the Lyapunov function is given by $\dot{V}(t) = \sum_{i=1}^{N} \left\{ \frac{1}{2} \left(\dot{e}_i^T(t) P_i e_i(t) + e_i^T(t) P_i \dot{e}_i(t) \right) + \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\tilde{\theta}}_i(t) \right\}$. After substituting $\dot{e}_i(t)$ from the tracking error dynamics in (10), we arrive at

$$\begin{aligned} \dot{V}(t) = \sum_{i=1}^{N} \Bigg\{ & \frac{1}{2} e_i^T(t) \left(A_i^T P_i + P_i A_i \right) e_i(t) + \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\tilde{\theta}}_i(t) \\ & + e_i^T(t) P_i B_i \left[\omega_i(x(t)) + h_i(x_i(t)) + \eta_i(x(t)) \right] \\ & - e_i^T(t) P_i B_i \left[K_i e_i(t) + l_i B_i^T P_i e_i(t) \right] \\ & - e_i^T(t) P_i B_i \left[\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t) \right) \right] \Bigg\} \end{aligned} \tag{15}$$

Substituting $s_i(t) = B_i^T P_i e_i(t)$ in the above equation and rearranging the terms results in

$$\begin{aligned} \dot{V}(t) = \sum_{i=1}^{N} \Bigg\{ & \frac{1}{2} e_i^T(t) \left(\left(A_i - B_i K_i \right)^T P_i + P_i \left(A_i - B_i K_i \right) \right) e_i(t) \\ & + s_i(t) \left[\omega_i(x(t)) + h_i(x_i(t)) + \eta_i(x(t)) \right] - l_i s_i^2(t) \\ & - s_i(t) \left[\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t) \right) \right] + \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\tilde{\theta}}_i(t) \Bigg\} \end{aligned} \tag{16}$$

Note that P_i is the solution of the Lyapunov equation $P_i(A_i - B_i K_i) + (A_i - B_i K_i)^T P_i = -Q_i$ where Q_i is a positive definite matrix. Now use this Lyapunov equation along with the result of Assumption 3 and the Cauchy-Schwarz inequality to get

$$\begin{aligned}\dot{V}(t) \le \sum_{i=1}^{N} \Bigg\{ & -\frac{1}{2} e_i^T(t) Q_i e_i(t) - l_i s_i^2(t) - \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\hat{\theta}}_i(t) \\ & + s_i(t) h_i\left(x_i(t)\right) + \frac{s_i^2(t)}{2} + \frac{\varsigma_{i0}^2}{2} + \frac{N}{2} \sum_{j=1}^{N} \varsigma_{ij}^2\left(\left|s_j(t)\right|\right) \\ & + \frac{s_i^2(t)}{2} + \frac{\eta_i^2\left(x(t)\right)}{2} - s_i(t)\left[\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right)\right] \Bigg\}\end{aligned} \tag{17}$$

Since $\varsigma_{ij}(.)$is a smooth function for i, j=1,..., N, there exists another smooth function $\zeta_{ij}(.)$ such that $\varsigma_{ij}(|s_j|) = |s_j|\zeta_{ij}(|s_j|)$ for i, j=1,..., N (Refer to [8, 33] for more details). Applying this result to (17) and changing the order of the summations yields

$$\begin{aligned}\dot{V}(t) \le \sum_{i=1}^{N} \Bigg\{ & -\frac{1}{2} e_i^T(t) Q_i e_i(t) - \left(l_i - 1\right) s_i^2(t) - \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\hat{\theta}}_i(t) \\ & + s_i(t)\left[h_i\left(x_i(t)\right) + \frac{N}{2} s_i(t) \sum_{j=1}^{N} \xi_{ji}^2\left(\left|s_i(t)\right|\right) \right] \\ & + s_i(t)\left[-\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right)\right] + \frac{\eta_{i_M}^2}{2} + \frac{\varsigma_{i0}^2}{2} \Bigg\}\end{aligned} \tag{18}$$

Based on Assumptions 2 and 3, the function $q_i(x_i(t), e_i(t)) = h_i(x_i(t)) + \frac{N}{2} s_i(t) \sum_{j=1}^{N} \zeta_{ji}^2(|s_i(t)|)$ is a smooth function, so it can be approximated by a two-layer NN with bounded activation functions, target weights, and estimation error as $q_i(x_i(t), e_i(t)) = \theta_i^T \psi_i\big(x_i(t), e_i(t)\big) + \varepsilon_i(t)$ where $|\varepsilon_i| \le \varepsilon_{i_M}$ and $\left|\theta_i^T \theta_i\right| \le \theta_{i_M}^2$ with ε_{i_M} and θ_{i_M} being positive constants. Therefore, the online approximator $\hat{q}_i$ which is used to approximate q_i, is constructed by a feedforward NN with one layer of adjustable weights as

$$\hat{q}_i\left(x_i(t), e_i(t); \hat{\theta}_i(t)\right) = \hat{\theta}_i^T(t) \psi_i\left(x_i(t), e_i(t)\right) \tag{19}$$

The vectors x_i and e_i are the NN inputs, $\psi_i: \mathbb{R}^{2n_i} \to \mathbb{R}^{p_i}$ is a vector of basis functions like sigmoid or RBF, and $\hat{\theta}_i$ is the vector of NN weights in the output layer. Now define the NN weight estimation error $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$ and rewrite the derivative of the Lyapunov function as

$$\begin{aligned}\dot{V}(t) \le \sum_{i=1}^{N} \Bigg\{ & -\frac{1}{2} e_i^T(t) Q_i e_i(t) - \left(l_i - 1\right) s_i^2(t) - \frac{1}{\alpha_i} \tilde{\theta}_i^T(t) \dot{\hat{\theta}}_i(t) \\ & + s_i(t) \tilde{\theta}_i^T(t) \psi_i\left(x_i(t), e_i(t)\right) + s_i(t) \varepsilon_i + \frac{\eta_{i_M}^2}{2} + \frac{\varsigma_{i0}^2}{2} \Bigg\}\end{aligned} \tag{20}$$

If the update law is selected as in (12), then

$$\dot{V}(t) \leq \sum_{i=1}^{N} \left\{ -\frac{1}{2} e_i^T(t) Q_i e_i(t) - \left(l_i - \frac{3}{2} \right) s_i^2(t) - \frac{\gamma_i}{2\alpha_i} \tilde{\theta}_i^T(t) \tilde{\theta}_i(t) + \frac{\gamma_i}{2\alpha_i} \theta_{i_M}^2 + \frac{\varepsilon_{i_M}^2}{2} + \frac{\eta_{i_M}^2}{2} + \frac{\varsigma_{i0}^2}{2} \right\} \tag{21}$$

Assuming that user defined parameters are selected such that $l_i > 3/2$, $\dot{V}(t)$ will be less than zero, if one of the following $2N$ conditions is satisfied

$$\|e_i(t)\| > \sqrt{\frac{2D_M}{Q_{i_{\max}}}}, \quad \|\tilde{\theta}_i(t)\| > \sqrt{\frac{2\alpha_i D_M}{\gamma_i}} \qquad i = 1, \ldots, N \tag{22}$$

where $D_M = \sum_{i=1}^{N} \left\{ \frac{\gamma_i}{2\alpha_i} \theta_{i_M}^2 + \frac{\varepsilon_{i_M}^2}{2} + \frac{\eta_{i_M}^2}{2} + \frac{\varsigma_{i0}^2}{2} \right\}$. So all the tracking errors e_i and the parameter estimation errors $\tilde{\theta}_i$ are uniformly ultimately bounded with bounds provided in (22).

Remark 1: Note that, in healthy conditions, the neural network which is generating $\hat{q}_i$ is not estimating the interconnection function ω_i, but it is used to approximate $q_i(e_i(t)) = \frac{N}{2} s_i(t) \sum_{j=1}^{N} \zeta_{ji}^2(|s_i(t)|)$. In fact, all the OLAs must be utilized together to cancel all the interconnection terms.

Remark 2: Based on the proof of Theorem 1, the bound on tracking error can be decreased by choosing Q_i matrices with large eigenvalues. This change will be reflected in the control input through the P_i matrix.

2.1.2.2. DFTC Design: Special Case

In this subsection, fault tolerant control of centralized/small-scale systems with known interconnection terms is investigated as a special case. Although the DFTC in previous subsection is mainly designed for large-scale interconnected systems with unknown interconnections, it can also be applied to systems in which interconnections are known and measurements are available to all subsystems. The performance of the controller can be improved for such systems by making use of the interconnection dynamics and the additional information on nonlocal measurements. To this end, the term $\omega_i(x(t))$ in the ith subsystem error dynamics is known and can be cancelled out by designing the controller as

$$\begin{aligned} u_i(t) = g_i(x_i(t))^{-1}\left(y_{id}^{(n_i)}(t) - K_i e_i(t) - f_i(x_i(t)) - \omega_i(x(t))\right) \\ -g_i(x_i(t))^{-1}\left(\hat{p}_i(x_i(t);\hat{\upsilon}_i(t)) + l_i B_i^T P_i e_i(t)\right) \end{aligned} \tag{23}$$

where $K_i \in \mathbb{R}^{1\times n_i}$ is selected such that $A_i - B_i K_i$ is Hurwitz, $P_i \in \mathbb{R}^{n_i \times n_i}$ is the unique solution to the Lyapunov equation $P_i(A_i - B_i K_i) + (A_i - B_i K_i)^T P_i = -Q_i$ with $Q_i \in \mathbb{R}^{n_i \times n_i}$ being an arbitrary symmetric positive definite matrix, and $\hat{p}_i$ is the output of the online approximator which will later be defined by a neural network with $\hat{\upsilon}_i$ as the vector of its adjustable weights. Applying the controller in (23) to the error dynamics (6) leads to

$$\begin{aligned} \dot{e}_i(t) = A_i e_i(t) + B_i\left[h_i(x_i(t)) + \eta_i(x(t))\right] \\ -B_i\left[K_i e_i(t) + l_i B_i^T P_i e_i(t) + \hat{p}_i(x_i(t);\hat{\upsilon}_i(t))\right] \end{aligned} \tag{24}$$

The stability of the entire system in the presence of fault is investigated in the next theorem.

Theorem 2: Consider the interconnected system described by (1) with known interconnection terms and local control inputs in the form of (23). If the vectors K_i are chosen such that $A_i - B_i K_i$ is Hurwitz for $i = 1, \dots, N$ and the NN based online approximator is defined by

$$\hat{p}_i(x_i(t);\hat{\upsilon}_i(t)) = \hat{\upsilon}_i^T(t)\varphi_i(x_i(t)) \tag{25}$$

with parameter update law selected as

$$\dot{\hat{\upsilon}}_i(t) = \beta_i \varphi_i(x_i(t)) s_i(t) - \lambda_i \hat{\upsilon}_i(t) \tag{26}$$

where β_i and λ_i are positive constants. Then the tracking errors $e_i(t)$, and parameter estimation errors $\tilde{\upsilon}_i(t) = \upsilon_i - \hat{\upsilon}_i(t)$ are uniformly ultimately bounded, if the design parameters are selected such that

$$l_i > 1 \qquad for\ i = 1, \dots, N \tag{27}$$

Proof: Consider the Lyapunov function candidate as

$$V_i(t) = \sum_{i=1}^{N}\left(\frac{1}{2}e_i^T(t)P_i e_i(t) + \frac{1}{2\beta_i}\tilde{\upsilon}_i^T(t)\tilde{\upsilon}_i(t)\right) \tag{28}$$

Taking the derivative of this function, using the error dynamics in (24), and using the Lyapunov equation $P_i(A_i - B_iK_i) + (A_i - B_iK_i)^T P_i = -Q_i$, results in

$$\dot{V}(t) = \sum_{i=1}^{N} \Big\{ -\frac{1}{2} e_i^T(t) Q_i e_i(t) - l_i s_i^2(t) - s_i(t)\hat{p}_i\left(x_i(t); \hat{\upsilon}_i(t)\right) \\ + s_i(t) h_i\left(x_i(t)\right) + s_i(t)\eta_i\left(x(t)\right) + \frac{1}{\beta_i}\tilde{\upsilon}_i^T(t)\dot{\tilde{\upsilon}}_i(t) \Big\} \tag{29}$$

Based on Assumption 2, the fault function h_i can be represented as $h_i(x_i(t)) = \upsilon_i^T \varphi_i(x_i(t)) + \mu_i(t)$ where φ_i is a vector of known basis functions, υ_i is a vector of unknown parameters bounded by $|\upsilon_i^T \upsilon_i| \leq \upsilon_{iM}^2$, and μ_i is the approximation error bounded by $|\mu_i| \leq \mu_{i_M}$. Therefore, in order to cancel h_i in (30), the online approximator $\hat{p}_i$ is constructed as $\hat{p}_i(x_i(t); \hat{\upsilon}_i(t)) = \hat{\upsilon}_i^T(t)\varphi_i(x_i(t))$ which is basically a NN with n_i inputs, one hidden layer, and one output. The activation functions of the hidden layer are determined by the basis function φ_i and the vector of output layer weights is referred to as $\hat{\upsilon}_i$.

Now we use this definition of the NN-based online approximator and apply the Cauchy-Schwarz inequality and the result of Assumption 1 to get

$$\dot{V}(t) \leq \sum_{i=1}^{N} \Big\{ -\frac{1}{2} e_i^T(t) Q_i e_i(t) - \left(l_i - 1\right) s_i^2(t) - \frac{1}{\beta_i}\tilde{\upsilon}_i^T(t)\dot{\hat{\upsilon}}_i(t) \\ + s_i(t)\tilde{\upsilon}_i^T\left(t\right)\varphi_i\left(x_i(t)\right) + \frac{1}{2}\mu_{i_M}^2 + \frac{1}{2}\eta_{i_M}^2 \Big\} \tag{30}$$

By selecting the weight update law as in (26), we arrive at

$$\dot{V}(t) \leq \sum_{i=1}^{N} \Big\{ -\frac{1}{2} e_i^T(t) Q_i e_i(t) - \left(l_i - 1\right) s_i^2(t) - \frac{\lambda_i}{2\beta_i}\tilde{\upsilon}_i^T(t)\tilde{\upsilon}_i(t) \\ + \frac{\lambda_i}{2\beta_i}\upsilon_i^T \upsilon_i + \frac{1}{2}\mu_{i_M}^2 + \frac{1}{2}\eta_{i_M}^2 \Big\} \tag{31}$$

Since Q_i is selected as a positive definite matrix and other user-defined parameters are selected such that β_i and λ_i are positive and $l_i > 1$ for$= 1, \ldots, N$, the derivative of Lyapunov function is less than zero if one of the following conditions is satisfied

$$\|e_i(t)\| > \sqrt{\frac{2\bar{D}_M}{Q_{i_{\max}}}}, \quad \|\tilde{\upsilon}_i(t)\| > \sqrt{\frac{2\beta_i\bar{D}_M}{\lambda_i}} \qquad i = 1, \ldots, N \tag{32}$$

where $\bar{D}_M = \sum_{i=1}^{N}\left\{\frac{\lambda_i}{2\beta_i}\upsilon_{i_M}^2 + \frac{1}{2}\mu_{i_M}^2 + \frac{1}{2}\eta_{i_M}^2\right\}$. So all the tracking errors e_i and weight estimation errors $\tilde{\upsilon}_i$ will be uniformly ultimately bounded with bounds provided in (32).

Remark 3: As expected, the use of additional available information about the interconnection terms in the controller development leads to a better performance in terms of smaller error bounds. Furthermore, unlike the general case, the outputs of the neural networks incorporated in the controllers are useful for further analysis. In fact, the OLA in the controller of each subsystem provides estimation of local fault function in that subsystem. The approximated fault dynamics can be very useful in fault identification as well as failure prediction.

The decentralized fault tolerant controller presented in this chapter is easy to implement on large-scale industrial systems, where significant amount of communication between subsystems due to state vectors is not possible or desirable. With the presented method, all subsystems can be controlled by using only local measurements at each subsystem which is made possible by the online approximators incorporated in the controllers. Once a fault occurs in the system, the online approximators will estimate the effects of fault as well as the interconnections in order to mitigate the fault effects. This will allow the entire system to continue its operation with an acceptable performance in the presence of fault. Therefore, repair or replacement of the faulty component can be postponed to the next scheduled shut-down without a significant loss in the overall performance, hence the system downtime is reduced. Furthermore, the cost and issues of unnecessary transmissions between the subsystems of a possibly large-scale system can be avoided or significantly decreased, due to the decentralized structure of the DFTC. Next, fault accommodation of distributed parameter systems is discussed.

2.2. Fault Accommodation for Distributed Parameter Systems Represented by Parabolic PDEs

2.2.1. System Description

Before presenting the system description, the notation and the norm used throughout this section is given [34]. A scalar function $v_1(x) \in L_2(0,1)$ means it is square integrable on the Hilbert space $L_2(0,1)$ with the corresponding norm

$$\|v_1(\cdot)\|_2 = \sqrt{\int_0^1 v_1^2(x)dx}\ , \tag{33}$$

Now consider

$$[L_2(0,1)]^n = \underbrace{L_2(0,1)\times L_2(0,1)\times\ldots\times L_2(0,1)}_{n\ times}, \tag{34}$$

with the corresponding norm of a vector function $v(x,t) = [v_1(x,t), v_2(x,t), \ldots, v_n(x,t)]^T \in [L_2(0,1)]^n$ defined as

$$\|v(\cdot)\|_{2,n} = \sqrt{\sum_{i=1}^{n}\|v_i(x)\|_2^2} = \sqrt{\int_0^1 v^T(x)v(x)dx}\ . \tag{35}$$

A class of n-dimensional linear DPS, which can be expressed by the following parabolic partial differential equation (PDE), is described by

$$v_t(x,t) = \varepsilon v_{xx}(x,t) + \Lambda v(x,t) + d(v,x,t), \tag{36}$$

with boundary conditions defined by

$$v_x\left(0,t\right) = 0,\ \ v(1,t) = U(t),\ \ y(t) = v(0,t), \tag{37}$$

for $x \in (0,1)$ and $t \geq 0$, where $d(v,x,t) = [d_1(v,x,t), \ldots d_n(v,x,t)] \in \Re^n$ stands for disturbance or uncertainty, $v(x,t) = [v_1(x,t), \ldots, v_n(x,t)]^T \in [L_2(0,1)]^n$ represents the state of the DPS, v_t and v_x denote partial derivatives of $v(x,t)$ or $\partial v(x,t)/\partial t$ and $\partial v(x,t)/\partial x$ respectively, $U(t) = [u_1(x,t), \ldots, u_n(x,t)]^T \in \Re^n$denotes the control input, ε is a positive constant, and $\Lambda \in \Re^{n\times n}$ is a real valued square matrix. In addition, $y(t) \in \Re^n$ is the system output vector measured at the opposite end of both the actuator and controller. For fault accommodation, a controller is required prior to the fault.

Assumption 4: The system uncertainty or disturbance is bounded above such that $\|d(v,x,t)\| \leq \bar{d}$ for all (v,x) and $t \geq 0$, where $\bar{d} > 0$ is a known constant. It is written as a general form in this chapter, a more specific model can be found in [35, 36].

Given a reference output, a full-state desired trajectory satisfying the system dynamics given by (36) and (37) is required in order to design the control input $U(t)$ which in turn allows the system state to follow the trajectory. Given a reference output $v_d(0,t) \in \Re^n$, a desired state trajectory for $0 < x \leq 1$ can be represented as [37]

$$v_d(x,t) = \sum_{k=0}^{\infty} a_k(t)\frac{x^k}{k!}, \tag{38}$$

where $a_k(t) = [a_{k1}(t), a_{k2}(t), \ldots, a_{kn}(t)] \in \Re^n$ represents time-varying coefficients of Taylor series expansion in x. These coefficients are determined by using the reference output and the system description (36), (37).

Next define the state tracking error as $r(x,t) = [r_1(x,t),\dots,r_n(x,t)]^T = v(x,t) - v_d(x,t)$. The state tracking error dynamics can be obtained as

$$r_t(x,t) = \varepsilon r_{xx}(x,t) + \Lambda r(x,t) + d(v,x,t)\,, \tag{39}$$

$$r_x(0,t) = 0\,, \tag{40}$$

where $r_t = \partial r/\partial t$, $r_x = \partial r/\partial x$ and $r_{xx} = \partial^2 r/\partial x^2$. The open-loop system (39) and (40) with $r(1,t) = 0$ is unstable when Λ is positive definite with sufficiently large eigenvalues. Since $\Lambda r(x,t)$ is the source of instability, our aim is to eliminate this term by using both the Volterra integral transformation [34, 37] and a suitable controller. Apply the Volterra integral transformation given by

$$w(x,t) = r(x,t) - \int_0^x K(x,\tau) r(\tau,t) d\tau\,, \tag{41}$$

with feedback control input $U(t) = U_h(t)$

$$U(t) = U_h(t) = v_d(1,t) + \int_0^1 K(1,\tau) r(\tau,t) d\tau\,, \tag{42}$$

along with the boundary condition

$$r(1,t) = \int_0^1 K(1,\tau) r(\tau,t) d\tau\,, \tag{43}$$

to convert the system (39),(40) and (43) into a stable target system described by

$$w_t(x,t) = \varepsilon w_{xx}(x,t) - Cw(x,t) + d(v,x,t) - \int_0^x K(x,\tau) d(v,\tau,t) d\tau\,, \tag{44}$$

$$w_x(0,t) = 0, \quad w(1,t) = 0\,. \tag{45}$$

Here

$$K(x,\tau) = -\sum_{n=0}^{\infty} \frac{(x^2-\tau^2)^n (2x)}{(4\varepsilon)^{n+1} n!(n+1)!} [\sum_{i=0}^{n} \binom{n}{i} C^i (\Lambda + C)\Lambda^{n-i}]\,, \tag{46}$$

is an $n \times n$ controller kernel matrix obtained by using a backstepping approach through the well-posed hyperbolic PDE given by [34]

$$\varepsilon K_{xx}(x,\tau)-\varepsilon K_{\tau\tau}(x,\tau)=K(x,\tau)\Lambda+CK(x,\tau), \tag{47}$$

$$K_{\tau}(x,0)=0 \;\; K(x,x)=-(\Lambda+C)x/2\varepsilon, \tag{48}$$

where $C \in \Re^{n\times n}$ is an arbitrary symmetric positive definite square matrix. Due to the invertability of (41) [37], the boundedness of $w(x,t)$ can guarantee the boundedness of $r(x,t)$.

It is important to note that the controller given in (42) clearly requires the state vector $v(x,t)$ at all positions. Therefore, the output feedback controller will be introduced in Section 2.2.2. Next actuator and sensor fault function, $h_a(t) \in \Re^n$ and $h_s(t) \in \Re^n$, respectively are considered at the boundary of the DPS.

In the presence of actuator and sensor faults, the system description from (36) and (37) can be described by (36) subjected to the new boundary conditions

$$v_x(0,t)=0,\; v(1,t)=U(t)+h_a(t), \tag{49}$$

$$y(t)=v(0,t)+h_s(t), \tag{50}$$

Moreover, the fault functions can be written as

$$h_a(t)=\Omega(t-t_0)\bar{h}_a(t)\,,\; h_s(t)=\Omega(t-t_0)\bar{h}_s(t)\,, \tag{51}$$

where $\Omega_i(t-t_0)$ is the time profile of the fault defined as in (2) for $i=1,\dots,n,\ t_0$ is the fault occurrence time and $\bar{h}_a(t)$ and $\bar{h}_s(t)$ denote actuator and sensor fault function dynamics respectively. For the purpose of accommodation, only incipient faults are considered here. The following assumption is needed in order to proceed.

Assumption 5: The fault function can be expressed as linear in the unknown parameters (LIP)[31]. In other words, the actuator fault function $\bar{h}_a(t)=\Phi_a(U(t),t)\theta_a$ and the sensor fault function $\bar{h}_s(t)=\Phi_s(t)\theta_s$ with $\theta_a \in \Re^n$ and $\theta_s \in \Re^n$ being the unknown fault parameter vector satisfies $\|\theta_a\| \le \theta_{amax}$, $\|\theta_s\| \le \theta_{smax}$, with $\Phi_a(U(t),t)=diag(\sigma_i^{(a)}(U(t),t)) \in \Re^{n\times n}$ for an actuator fault and $\Phi_s(y(t),t)=diag[\sigma_i^{(s)}t)] \in \Re^{n\times n}$ for a sensor fault being known where $\sigma_i^{(a)}(\cdot)$ and $\sigma_i^{(s)}(\cdot) \in \Re(i=1,2,\dots n)$ are smooth bounded function.

2.2.2. Fault Accommodation for DPS with Output Measurements

In this section, a detection observer, which provides the estimated state information, is designed using an input and a couple of output filters. In addition, an adaptive tuning law has to be carefully selected to detect and approximate both the sensor and actuator fault functions using estimated state measurements under the assumption that the type of fault is known. The controller structure from the previous section with state measurements can be utilized with modifications for fault accommodation. The controller for the healthy case is introduced first and it is modified for the purpose of fault accommodation later.

2.2.2.1. Output Feedback Controller Design Under Healthy Conditions

Now assume that the only the boundary value $v(0,t)$ is measured. In order to design the observer and output feedback controller, the DPS from (36) and (37) is first converted to an observable form, by utilizing the following transformation [38] given by

$$z(x,t) = v(x,t) - \int_0^x l(x,\tau)v(\tau,t)d\tau\,, \tag{52}$$

where $l(x,\tau) = -2(x-1)\sum_{n=0}^{\infty}\frac{(-1)^n[(\tau-1)^2-(x-1)^2]^n\Lambda^{n+1}}{(4\varepsilon)^{n+1}n!(n+1)!}$ being the solution of the hyperbolic PDE given by $l_{xx} - l_{\tau\tau} = l(x,\tau)\Lambda/\varepsilon,\ l(1,\tau) = 0$ and $l(x,x) = \Lambda(1-x)/(2\varepsilon)$. The transformation (52) can convert the original system (36)-(37) to the following PDE

$$z_t(x,t) = \varepsilon z_{xx}(x,t) + G(x)z(0,t) + d(v,x,t) + \int_0^x l(x,)d(v,\tau,t)d\tau\,, \tag{53}$$

$$z_x(0,t) = L_0 z(0,t),\ z(1,t) = U(t)\,, \tag{54}$$

$$y(t) = z(0,t)\,, \tag{55}$$

where $L_0 = -\Lambda/(2\varepsilon)$ and $G(x) = -\varepsilon l_\tau(x,0)$. Note $z(0,t)$ is available since $z(0,t) = v(0,t)$. This transformation helps to avoid the unstable term $\Lambda v(x,t)$ from appearing in the design of filters which are introduced next.

The DPS given by (53) and (54) have $U(t)$, $L_0 v(0,t)$ and $G(x)v(0,t)$ as external inputs. According to superposition principle [38] of linear DPS, its solution can be expressed as the sum of the response of the PDEs for each external input acting alone. Therefore, $z(x,t) \in \Re^n$ can be expressed as a combination of the solution to three individual PDEs defined by

$$\Xi_t(x,t) = \varepsilon\Xi_{xx}(x,t),\quad \Xi_x(0,t) = 0,\quad \Xi(1,t) = U(t)\,, \tag{56}$$

where $\Xi(x,t)$ is referred to an input filter since it is derived from the input of the actual system [38]. Next consider

$$A_t(x,t)=\varepsilon A_{xx}(x,t),\ A_x(0,t)=y(t),\ A(1,t)=0\ , \tag{57}$$

where $A(x,t)$ is an output filter since it is derived from output of the actual system, $y(t)$. Finally consider

$$\Pi_t(x,\eta,t)=\varepsilon\Pi_{xx}(x,\eta,t)+\delta(x-\eta)y(t),\ \ \Pi_x(0,\eta,t)=0,\ \ \Pi(1,\eta,t)=0\ . \tag{58}$$

where $\Pi(x,\eta,t)$ is a second output filter.

Define the observer with its state, $\hat{z}(x,t)\in\Re^n$, given by

$$\hat{z}(x,t)=\Xi(x,t)+L_0\mathrm{A}(x,t)+\int_0^1 G(s)\Pi(x,s,t)ds\ , \tag{59}$$

with $\hat{y}(t)=\hat{z}(0,t)$ and $e(t)=y(t)-\hat{y}(t)$.

The observer state estimation error is obtained as $\tilde{z}(x,t)\in\Re^n=z(x,t)-\hat{z}(x,t)$ with its dynamics satisfying

$$\tilde{z}_t(x,t)=\varepsilon\tilde{z}_{xx}(x,t)+d(v,x,t)+\int_0^x l(x,)d(v,\tau,t)d\tau,\ \tilde{z}_x(0,t)=0,\ \tilde{z}(0,t)=0\ . \tag{60}$$

Since (59) provides $\hat{z}$ instead of $\hat{v}$, for the controller design we need the inverse transformation of (52) given by

$$v(x,t)=z(x,t)+\int_0^x M(x,\tau)z(\tau,t)d\tau\ , \tag{61}$$

to obtain the estimated state $\hat{v}(x,t)=\hat{z}(x,t)+\int_0^x M(x,\tau)\hat{z}(\tau,t)d\tau$ where $M(x,\tau)\in\Re^n=-2(x-1)\sum_{n=0}^{\infty}\frac{[(\tau-1)^2-(x-1)^2]^n\Lambda^{n+1}}{(4\varepsilon)^{n+1}n!(n+1)!}$ is a bounded solution to the following hyperbolic PDE $M_{xx}-M_{\tau\tau}=-\frac{M(x,\tau)\Lambda}{\varepsilon}, M(1,\tau)=0, M(x,x)=\frac{\Lambda(1-x)}{2\varepsilon}$. Then the state estimation error is defined in terms of $M(x,\tau)$ and $\tilde{z}(\tau,t)$ as

$$\tilde{v}(x,t)=v(x,t)-\hat{v}(x,t)=\tilde{z}(x,t)+\int_0^x M(x,\tau)\tilde{z}(\tau,t)d\tau\ . \tag{62}$$

Note that the boundedness of $\tilde{v}(x,t)$ is guaranteed due to the boundedness of $\tilde{z}(x,t)$. With the observer defined in (59), the stability of the observable system (53) and (54) as well as the original system (36) and (37) can be demonstrated with the controller designed as

$$\begin{aligned} U(t) = \hat{U}_h(t) &= v_d(1,t) + \int_0^1 K(1,\tau)[\hat{v}(\tau,t) - v_d(\tau,t)]d\tau \\ &= v_d(1,t) + \int_0^1 K(1,\tau) r(\tau,t) d\tau - \int_0^1 K(1,\tau)\tilde{v}(\tau,t)d\tau \end{aligned} \quad . \qquad 63)$$

where $\widehat{U}_h(t)$ is the control input using estimated state vector during healthy conditions. It is important to observe the difference between this controller using the estimated state vector $\hat{v}(x,t)$ and the controller (42) designed by using the measured state vector $v(x,t)$. They will be equivalent when $\tilde{v}(x,t) \to 0$.

Next, apply the controller (63) to the system (36) and (37), the state tracking error dynamics can be obtained as (39) and (40) with the following boundary condition

$$r(1,t) = \int_0^1 K(1,\tau) r(\tau,t) d\tau - \int_0^1 K(1,\tau)\tilde{v}(\tau,t) d\tau \ . \qquad (64)$$

Then by asserting the transformation (41) to the state tracking error dynamics (39),(40) and (64), we get (44) subject to

$$w_x(0,t) = 0, \quad w(1,t) = -\int_0^1 K(1,\tau)\tilde{v}(\tau,t) d\tau \ . \qquad (65)$$

Therefore, from (62) and (65) it can be shown that

$$\begin{aligned} w^T(1,t) w(1,t) &= [\int_0^1 K(1,\tau)\tilde{v}(\tau,t)d\tau]^T [\int_0^1 K(1,\tau)\tilde{v}(\tau,t)d\tau] \\ &\le \int_0^1 [K(1,\tau)\tilde{v}(\tau,t)]^T [K(1,\tau)\tilde{v}(\tau,t)] d\tau \\ &= \int_0^1 \tilde{v}^T(\tau,t) K^T(1,\tau) K(1,\tau)\tilde{v}(\tau,t) d\tau \\ &\le \bar{k}^2 \int_0^1 \tilde{v}^T(\tau,t)\tilde{v}(\tau,t) d\tau = \bar{k}^2 \|\tilde{v}\|_{2,n}^2 \le k_h \|\tilde{z}\|_{2,n}^2 \end{aligned} \quad , \qquad (66)$$

where $\bar{k} = \underset{0 \le x \le 1}{max} \|K(x,\tau)\|_2$, $k_h = 2\bar{k}^2(\bar{m}^2 + 1)$ and $\bar{m} = \underset{0 \le x \le 1}{max} \|M(x,\tau)\|_2$.

To show the performance of the controller under healthy conditions, consider the following Lyapunov function candidate

$$V = \frac{1}{2\varepsilon} \|\tilde{z}(x,t)\|_{2,n}^2 + \frac{1}{2\varepsilon k_h} \int_0^1 \int_0^x w^T(\eta,t) w(\eta,t) d\eta dx \ , \qquad (67)$$

and its derivative of V with respect to t can be obtained as

$$\dot{V} = \frac{1}{\varepsilon}\int_0^1 \tilde{z}^T(x,t)\tilde{z}_t(x,t)dx + \frac{1}{\varepsilon k_h}\int_0^1\int_0^x w^T(\eta,t)w_t(\eta,t)d\eta dx \;. \tag{68}$$

By substituting (60), (44), (65) and (66) into (68) above and applying the integration by parts, the derivative of Lyapunov function can be written as

$$\begin{aligned}\dot{V} &= -\|\tilde{z}_x(x,t)\|_{2,n}^2 + \frac{1}{\varepsilon}\int_0^1 \tilde{z}^T(x,t)\int_0^x l(x,\eta)d(v,\eta,t)d\eta dx + \frac{1}{\varepsilon}\int_0^1 \tilde{z}^T(x,t)d(v,x,t)dx \\ &+\int_0^1 w^T(x,t)w_x(x,t)dx/k_h + \int_0^1\int_0^x w^T(\eta,t)d(v,\eta,t)d\eta dx/(\varepsilon k_h) \\ &-\int_0^1\int_0^x w^T(\eta,t)\int_0^\eta K(\eta,\tau)d(v,\tau,t)d\tau d\eta dx/(\varepsilon k_h) \\ &-[\int_0^1\int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta dx + \int_0^1\int_0^x w^T(\eta,t)Cw(\eta,t)d\eta dx/\varepsilon]/k_h\end{aligned} \;, \tag{69}$$

$$\begin{aligned}\Rightarrow \dot{V} &\le -(\pi^2-2)\|\tilde{z}(x,t)\|_{2,n}^2/4 - \frac{c}{\varepsilon k_h}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx \\ &+\frac{d_l}{\varepsilon}\int_0^1\sqrt{\tilde{z}^T(x,t)\tilde{z}(x,t)}dx + \frac{d_k}{\varepsilon k_h}\int_0^1\int_0^x\sqrt{w^T(\eta,t)w(\eta,t)}d\eta dx\end{aligned} \;, \tag{70}$$

$$\begin{aligned}\Rightarrow \dot{V} &\le -(\pi^2-4)\|z(x,t)\|_{2,n}^2/4 + d_l^2/2\varepsilon^2 + d_k^2/(4c\varepsilon k_h) \\ &-c\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx/(2\varepsilon k_h),\end{aligned} \;, \tag{71}$$

where $d_l = (1+\bar{l})\bar{d}$, $d_k = (1+\bar{k})\bar{d}$ with $\bar{k} = \max\limits_{0\le x\le 1}\|K(x,\tau)\|_2$ and $\bar{l} = \max\limits_{0\le x\le 1}\|l(x,\eta)\|_2$. Then $\dot{V} < 0$ when

$$\begin{aligned}&\|\tilde{z}(x,t)\|_{2,n} > \sqrt{\frac{2d_l^2}{\varepsilon^2(\pi^2-4)} + \frac{d_k^2}{c\varepsilon k_h(\pi^2-4)}} \quad or \\ &\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx > \frac{k_h d_l^2}{\varepsilon c} + \frac{d_k^2}{2c^2}\end{aligned} \;, \tag{72}$$

Therefore, $\tilde{z}$ and $w(x,t)$ will be bounded. The boundedness of $\tilde{v}$ and r are also guaranteed because of (52) and the invertiblity of (41).

2.2.2.2. Actuator Fault Detection and Accommodation

Recall the dynamics of transformed system with an actuator fault represented as (53) subjecting to

$$z_x(0,t) = L_0 z(0,t),\ z(1,t) = U(t) + h_a(t),\ y(t) = z(0,t)\,. \tag{73}$$

In order to approximate the fault dynamics upon detection, the design of the fault filter will be performed based on the observable form (53) which is expressed as $D_t(x,t) = \varepsilon D_{xx}(x,t)$, $D_x(0,t) = 0$, and $D(1,t) = [\sigma_1^{(a)}, \sigma_2^{(a)}, \dots \sigma_n^{(a)}]^T$ where $D(x,t) \in \Re^n$. Then the observer (59) after incorporating the adaptive term becomes

$$\begin{aligned}&\hat{z}(x,t) = \Xi(x,t) + \Gamma(x,t)\hat{\theta}_a(t) + L_0 \mathrm{A}(x,t) + \int_0^1 G(s)\Pi(x,s,t)ds \\ &\hat{y}(t) = \hat{z}(0,t), \quad e(t) = y(t) - \hat{y}(t)\end{aligned}, \tag{74}$$

where $\hat{\theta}_a(t)$ is the estimated fault parameter vector with $\hat{\theta}_a(0) = 0$ since the fault parameter vector under healthy conditions is $\theta_a = 0$ and $\Gamma(x,t) = diag(D(x,t)) \in \Re^{n\times n}$ with $\Gamma(1,t) = \Phi_a(U(t),t)$. Next, an ideal function $\bar{z}(x,t) \in \Re^n$ is introduced with an initial condition same as that of $\hat{z}(x,t)$. This ideal function is viewed as the ultimate target of $\hat{z}(x,t)$ as it gets tuned along with $\hat{\theta}_a(t)$. It is designed as

$$\bar{z}(x,t) = \Xi(x,t) + \Gamma(x,t)\theta_a + L_0 \mathrm{A}(x,t) + \int_0^1 G(s)\Pi(x,s,t)ds\,. \tag{75}$$

Then it is straight-forward to obtain the dynamics of $\bar{z}(x,t)$ as

$$\bar{z}_t(x,t) = \varepsilon \bar{z}_{xx}(x,t) + G(x)z(0,t)\,, \tag{76}$$

$$\bar{z}_x(0,t) = L_0 z(0,t),\ \bar{z}(1,t) = U(t) + \Phi_a(U(t),t)\theta_a\,. \tag{77}$$

Notice $\bar{z}(x,t)$ has the same initial condition as that of $\hat{z}(x,t)$ while it has different initial condition from $z(x,t)$. Because $\bar{z}(x,t)$ has the same dynamics as that of DPS given by the observable form (53), it will be utilized in the proof of fault approximation with filters. The next theorem discusses the performance of this observer as a fault detection observer and provides a suitable parameter tuning law.

Theorem 3 (Actuator Fault Detection and Approximation): Let the observer in (74) be used to monitor the system defined by (53) and (73). Then the magnitude of output detection residual $e(t)$ will increase in the presence of an actuator fault and when it reaches the threshold, a fault is considered detected. Upon detecting a fault, select the parameter tuning law as

$$\dot{\hat{\theta}}_a = \beta\Gamma(0,t)e(t) - \gamma\hat{\theta}_a\,, \tag{78}$$

where $0 < \beta < 2$ is the leaning rate and $\gamma > 0$ is a design parameter. Then the observer estimation error, $\tilde{z}$, and parameter estimation error, $\tilde{\theta}_a = \theta_a - \hat{\theta}_a$, are ultimately bounded (UB).

Proof: This is an extension of the scalar case from [39]. To show the boundedness of observer and parameter estimation errors in the presence of fault, an error is first defined as $\mu(x,t) = z(x,t) - \bar{z}(x,t)$. It is clear that

$$\mu_t(x,t) = \varepsilon\mu_{xx}(x,t) + d(v,x,t) - \int_0^x l(x,\tau)\, d(v,\tau,t)d\tau,\ \mu_x(0,t) = 0,\ \mu(1,t) = 0 \tag{79}$$

Now select a Lyapunov function candidate in the form of

$$V = \|\mu(x,t)\|_{2,n}^2 / (2\varepsilon) + \tilde{\theta}_a^T(t)\tilde{\theta}_a(t)/(2\beta)\ , \tag{80}$$

which is positive definite. Then the derivative of the Lyapunov function with respect to time can be obtained as

$$\dot{V} = \int_0^1 \mu^T(x,t)\mu_t(x,t)dx/\varepsilon + \tilde{\theta}_a^T(t)\dot{\tilde{\theta}}_a(t)/\beta\ . \tag{81}$$

Substituting the update law (78) in the above equation and notice that $e(t) = \tilde{z}(0,t) = \mu(0,t) + \Gamma(0,t)\tilde{\theta}_a(t)$, results in

$$\begin{aligned}\dot{V} \le & -\|\mu_x(x,t)\|_{2,n}^2 - e^T(t)e(t)/2 + \mu^T(0,t)\mu(0,t)/2 \\ & -\gamma[\tilde{\theta}_a^T(t)\tilde{\theta}_a(t) - \theta_{a\max}^2]/2\beta + d_l\int_0^1 \sqrt{\mu^T(x,t)\mu(x,t)}dx/\varepsilon\end{aligned}\ , \tag{82}$$

$$\Rightarrow \dot{V} \le -(\pi^2 - 4)\|\mu(x,t)\|_{2,n}^2/8 - \gamma\tilde{\theta}_a^T(t)\tilde{\theta}_a(t)/(2\beta) + \gamma\theta_{a\max}^2/(2\beta) + d_l^2/(4\varepsilon^2)\ . \tag{83}$$

Therefore, $\dot{V}$ will be less than zero when

$$\|\mu\|_{2,n} > \sqrt{\frac{4\gamma\varepsilon^2\theta_{a\max}^2 + 2\beta d_l^2}{(\pi^2-4)\beta\varepsilon^2}}\ \text{ or } \|\tilde{\theta}_a(t)\| > \sqrt{\theta_{a\max}^2 + \frac{\beta d_l^2}{2\gamma\varepsilon^2}}\ . \tag{84}$$

It is shown that with the parameter tuning law (78), the derivative of this function will be less than zero if μ or $\tilde{\theta}_a$ stays in a bounded region. Note that since $\tilde{z}(x,t) = \mu(x,t) + \Gamma(x,t)\tilde{\theta}_a(t)$, the bound of the observer residual $\tilde{z}$ is guaranteed since $\Gamma(x,t)$ is bounded.

The approximated fault function given by $\Phi_a(U(t),t)\hat{\theta}_a(t)$ is then utilized in the control input for fault accommodation. The overall control input is designed as

$$U(t) = U_{accom}(t) = \hat{U}_h(t) - \Phi_a(t)\hat{\theta}_a(t)\,, \tag{85}$$

yielding

$$z(1,t) = \hat{U}_h(t) + \Phi_a(t)\tilde{\theta}_a(t)\,, \tag{86}$$

in order to mitigate the effect of the fault on the system where $\widehat{U}_h(t)$ is the control input under healthy conditions using filter-based approach as given by (63). Then the dynamics of the transformed tracking error becomes (44) subject to

$$w_x(0,t) = 0, \quad w(1,t) = \Phi_a(U(t),t)\tilde{\theta}_a(t) - \int_0^1 K(1,\tau)\tilde{v}(\tau,t)d\tau\,. \tag{87}$$

Noting that $\Phi_a^T(U(t),t)\Phi_a(U(t),t) \leq \sigma_{amax}^2$ and $\tilde{z}(x,t) = \mu(x,t) + \Gamma(x,t)\tilde{\theta}_a$, it follows that

$$\begin{aligned} w^T(1,t)w(1,t) &\leq 2\sigma_{a\max}^2\tilde{\theta}_a^T\tilde{\theta}_a + 2\bar{k}\,\|\tilde{v}\|_{2,n}^2 \\ &\leq 2\sigma_{a\max}^2\tilde{\theta}_a^T\tilde{\theta}_a + 2k_h\,\|\tilde{z}\|_{2,n}^2 \leq k_c[\tilde{\theta}_a^T\tilde{\theta}_a + \|\mu\|_{2,n}^2] \end{aligned}\,. \tag{88}$$

where $k_c = max\{\,4k_h, (4k_h\bar{D} + 2\sigma_{amax}^2)\}$, $\bar{D} = \max\limits_{0\leq x\leq 1}\|D(x,t)\|^2$ and $\sigma_{a\,max} = sup\{\sqrt{\sum_{i=1}^n[\sigma_i^{(a)}(U(t)]^2,t)}\}$.

The next theorem shows the boundedness of tracking error with the accommodation scheme.

Theorem 4 (Actuator Fault Accommodation): Let the control law in (85) be used upon detecting an actuator fault. Then the parameter estimation, observer estimation and tracking errors are UB.

Proof: Notice that with controller modified as (85) the boundary condition of $\bar{z}(x,t)$ stays the same as $z(x,t)$ satisfying $\bar{z}_x(0,t) = L_0 z(0,t), \bar{z}(1,t) = \widehat{U}(t) + \Phi_a(U(t),t)\tilde{\theta}_a$, thus the dynamics of $\mu(x,t)$ is maintained as (79). Now select a Lyapunov function candidate as

$$V = \frac{R}{2\varepsilon}\|\mu(x,t)\|_{2,n}^2 + \frac{\tilde{\theta}_a^T\tilde{\theta}_a}{2\beta} + \frac{\gamma}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx\,. \tag{89}$$

By taking the derivative of the Lyapunov function with respect to time and applying integration by parts with (44) the following can be obtained

$$\begin{aligned}\dot{V} \leq & -R\|\mu_x(x,t)\|_{2,n}^2 + \frac{d_k}{2\varepsilon k_c}\int_0^1\int_0^x \sqrt{w^T(\eta,t)w(\eta,t)}d\eta dx + \frac{Rd_l}{\varepsilon}\int_0^1 \sqrt{\mu^T(x,t)\mu(x,t)}dx \\ & -\frac{\gamma c\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx}{2\varepsilon k_c} + \frac{\gamma}{2k_c}\int_0^1\int_0^x w^T(\eta,t)w_{\eta\eta}(\eta,t)d\eta dx + \frac{1}{\beta}\tilde{\theta}_a^T(t)\dot{\tilde{\theta}}_a(t)\end{aligned} \quad . \qquad (90)$$

Substituting the parameter update law from (78) and applying Poincare inequality [40] to (90) yields

$$\begin{aligned}\dot{V} \leq & -\frac{R\pi^2}{4}\|\mu(x,t)\|_{2,n}^2 - \tilde{\theta}_a^T(t)\Gamma(0,t)e(t) - \frac{\gamma c}{2\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx \\ & + \frac{Rd_l}{\varepsilon}\int_0^1 \sqrt{\mu^T(x,t)\mu(x,t)}dx + \frac{d_k}{2\varepsilon k_c}\int_0^1\int_0^x \sqrt{w^T(\eta,t)w(\eta,t)}d\eta dx \\ & + \frac{\gamma\tilde{\theta}_a^T(t)\hat{\theta}_a(t)}{\beta} + \frac{\gamma}{2k_c}\int_0^1\int_0^x w^T(\eta,t)w_{\eta\eta}(\eta,t)d\eta dx\end{aligned} \quad . \qquad (91)$$

Because $\tilde{z}(0,t) - \mu(0,t) = \Gamma(0,t)\tilde{\theta}_a(t)$ and $\tilde{z}(0,t) = e(t)$, the derivative of Lyapunov function can be rewritten as

$$\begin{aligned}\dot{V} \leq & -\frac{R(\pi^2-2)}{4}\|\mu(x,t)\|_{2,n}^2 + \mu^T(0,t)e(t) + \frac{d_k^2}{4c\varepsilon k_h} + \frac{Rd_l^2}{2\varepsilon^2} - e^T(t)e(t) \\ & -\frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx + \frac{\gamma}{2k_c}\int_0^1\int_0^x w^T(\eta,t)w_{\eta\eta}(\eta,t)d\eta dx + \frac{\gamma\tilde{\theta}_a^T(t)\hat{\theta}_a(t)}{\beta}\end{aligned} , \qquad (92)$$

$$\begin{aligned}\Rightarrow \dot{V} \leq & -\frac{R(\pi^2-2)}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx - \frac{e^T(t)e(t)}{2} \\ & + \frac{\mu^T(0,t)\mu(0,t)}{2} + \frac{\gamma\theta_{a\max}^2}{2\beta} + \frac{\gamma}{4k_c}\left[w^T(1,t)w(1,t) - w^T 0,t)w(0,t)\right] \\ & + \frac{d_k^2}{4c\varepsilon k_h} + \frac{Rd_l^2}{2\varepsilon^2} - \frac{\gamma\tilde{\theta}_a^T(t)\tilde{\theta}_a(t)}{2\beta} - \frac{\gamma}{2k_c}\int_0^1\int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta dx\end{aligned} \quad . \qquad (93)$$

From (88) it is known that $w^T(1,t)w^T(1,t) \leq k_c(\tilde{\theta}_a^T\tilde{\theta}_a + \|\mu\|_{2,n}^2)$, thus

$$\begin{aligned}\dot{V} \leq & -\frac{R(\pi^2-2)-\gamma-2}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{1}{2}e^T(t)e(t) - \frac{\gamma}{2\beta}\tilde{\theta}_a^T(t)\tilde{\theta}_a(t) \\ & -\frac{\gamma}{2k_c}[\int_0^1\int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta dx + \frac{1}{2}w^T 0,t)w(0,t)] \\ & -\frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx + \frac{\gamma}{2\beta}\theta_{a\max}^2 + \frac{Rd_l^2}{2\varepsilon^2} + \frac{d_k^2}{4c\varepsilon k_h} + \frac{\gamma}{4}\tilde{\theta}_a^T(t)\tilde{\theta}_a(t)\end{aligned} \quad . \qquad (94)$$

By applying Poincare inequality [40], we have $\int_0^1 \int_0^x w^T(\eta,t)w(\eta,t)d\eta\, dx \le 2w^T(0,t)w(0,t) + 4\int_0^1 \int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta\, dx$. Then the first derivative of the Lyapunov function becomes

$$\begin{aligned}\dot{V} \le & -\frac{R(\pi^2-2)-\gamma-2}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{\gamma(2-\beta)}{4\beta}\tilde{\theta}_a^T(t)\tilde{\theta}_a(t) \\ & -\gamma(\varepsilon+2c)\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx / 8\varepsilon k_c + \frac{\gamma\theta_{a\max}^2}{2\beta} + \frac{Rd_l^2}{2\varepsilon^2} + \frac{d_k^2}{4c\varepsilon k_c}\end{aligned}, \tag{95}$$

where $R > \frac{\gamma+2}{\pi^2-2}$ and $c = \lambda_{min}(C)$. Therefore, the derivative of Lyapunov function will be less than zero when

$$\begin{aligned}&\|\mu(x,t)\|_{2,n} > \sqrt{\frac{2\gamma\varepsilon^2 ck_c\theta_{a\max}^2 + 2R\beta ck_c d_l^2 + \beta\varepsilon}{\beta\varepsilon^2 ck_c[R(\pi^2-2)-\gamma-2]}} \quad or \\ &\|\tilde{\theta}_a(t)\| > \sqrt{\frac{2\theta_{a\max}^2}{2-\beta} + \frac{2R\beta d_l^2}{\gamma\varepsilon^2(2-\beta)} + \frac{\beta d_k^2}{\gamma\varepsilon ck_c(2-\beta)}} \quad or \\ &\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx > \frac{4\varepsilon k_c\theta_{a\max}^2}{\beta(\varepsilon+2c)} + \frac{4Rk_c d_l^2}{\gamma\varepsilon^2(\varepsilon+2c)} + \frac{2d_k^2}{4\gamma(\varepsilon+2c)}\end{aligned}. \tag{96}$$

Hence, μ and $\tilde{\theta}_a$ are ultimately bounded with the bounds defined in (96) which can be adjusted by using the designed parameter $c = \lambda_{min}(C)$. Since $\tilde{z}(x,t) = \mu(x,t) + \Gamma(x,t)\tilde{\theta}_a(t)$, $\tilde{z}$ is also bounded due to boundedness of $\Gamma(x,t)$. With the boundedness of $\int_0^1 \int_0^x w^T(\eta,t)w(\eta,t)d\eta\, dx$ proven and because $w(x,t)$ is continuous over $x \in [0,1]$, the transformed tracking error $w(0,t)$ is also bounded. Now, given the transformation (41) it is known that $w(0,t) = r(0,t)$, thus the boundedness of the tracking error $r(0,t)$ is ensured.

2.2.2.3. Sensor Fault Detection and Accommodation

Upon detection of a sensor fault, the following two fault filters will be applied to estimate the fault dynamics

$$\begin{aligned}&F_{1t}(x,t) = \varepsilon F_{1xx}(x,t), \\ &F_{1x}(0) = [\sigma_1^{(s)}, \sigma_2^{(s)}, ...\sigma_n^{(s)}]^T, F_1(1,t) = 0\end{aligned}, \tag{97}$$

$$\begin{aligned}&F_{2t}(x,\eta,t) = \varepsilon F_{2xx}(x,\eta,t) + \delta(x-\eta)[\sigma_1^{(s)}, \sigma_2^{(s)}, ...\sigma_n^{(s)}]^T, \\ &F_{2x}(0,\eta,t) = 0,\ F_2(1,\eta,t) = 0\end{aligned}. \tag{98}$$

The two output filters become

$$\begin{aligned} &\mathrm{A}_t(x,t) = \varepsilon \mathrm{A}_{xx}(x,t), \\ &\mathrm{A}_x(0,t) = z(0,t) + \Phi_s(t)\theta_s,\ \mathrm{A}(1,t) = 0 \end{aligned}, \tag{99}$$

$$\begin{aligned} &\Pi_t(x,\eta,t) = \varepsilon \Pi_{xx}(x,\eta,t) + \delta(x-\eta)[z(0,t) + \Phi_s(t)\theta_s], \\ &\Pi_x(0,\eta,t) = 0,\ \Pi(1,\eta,t) = 0 \end{aligned}. \tag{100}$$

Then the corresponding observer will be redefined as

$$\begin{aligned} \hat{z}(x,t) = \Xi(x,t) + L_0\mathrm{A}(x,t) + \int_0^1 G(s)\Pi(x,s,t)ds \\ -[L_0\Delta(x,t) + \int_0^1 G(s)\Psi(x,s,t)ds]\hat{\theta}_s(t) \end{aligned}, \tag{101}$$

where $\Delta(x,t) = diag(F_1(x,t))$, $\Psi(x,s,t) = diag(F_2(x,s,t))$ and $\hat{\theta}_s(t) \in \Re^n$ is the estimated sensor fault parameter vector. In order to proceed, similar to the actuator fault case, we introduce a variable defined by

$$\begin{aligned} \bar{z}(x,t) = \Xi(x,t) + L_0\mathrm{A}(x,t) + \int_0^1 G(s)\Pi(x,s,t)ds \\ -[L_0\Delta(x,t) + \int_0^1 G(s)\Psi(x,s,t)ds]\theta_s(t) \end{aligned}, \tag{102}$$

and therefore,

$$\hat{z}(x,t) = \bar{z}(x,t) + [L_0\Delta(x,t) + \int_0^1 G(s)\Psi(x,s,t)ds]\tilde{\theta}_s(t), \tag{103}$$

where $\tilde{\theta}_s(t) = \theta_s - \hat{\theta}_s(t)$ is the parameter estimation error. Defining an error signal as $\mu(x,t) = z(x,t) - \bar{z}(x,t)$, it is clear that

$$\mu_t(x,t) = \varepsilon\mu_{xx}(x,t) + d(v,x,t) - \int_0^x l(x,\tau)d(v,\tau,t)d\tau, \tag{104}$$

$$\mu_x(0,t) = 0,\ \mu(1,t) = 0. \tag{105}$$

Then the estimated state error $\tilde{z}(x,t) = z(x,t) - \hat{z}(x,t)$ can be represented as

$$\tilde{z}(x,t) = \mu(x,t) - [L_0\Delta(x,t) + \int_0^1 G(s)\Psi(x,s,t)ds]\tilde{\theta}_s(t). \tag{106}$$

The next theorem evaluates the detection observer and presents an appropriate tuning law to approximate the fault function upon detection of the sensor fault.

Theorem 5 (Sensor Fault Detection and Fault Approximation): Let the observer in (101) be used to monitor the system defined by (53)-(54) and (50). The magnitude of detection residual $e(t)$ will increase in the presence of a sensor fault and when it reaches the detection threshold, a fault is considered detected. Upon detecting a sensor fault, select the parameter tuning law as

$$\dot{\hat{\theta}}_s = \beta[\Phi_s(t) - F(0,t)]^T e(t) - \gamma\hat{\theta}_s, \tag{107}$$

where $F(x,t) = L_0\Delta(x,t) + \int_0^1 G(s)\Psi(x,s,t)ds$, $0 < \beta < 2$ is the leaning rate, $\gamma > 0$ is a design parameter, and $e(t)$ is the detection residual defined as $e(t) = y(t) - \Phi_S(t)\hat{\theta}_S(t) - \hat{z}(0,t) = \tilde{z}(0,t) + \Phi_S(t)\tilde{\theta}_S(t)$. Then the observer residual, $\tilde{z}$, and parameter estimation error, $\tilde{\theta}_S$, are bounded.

Proof: This is an extension of [39] where only scalar actuator fault is considered. We have shown that under healthy condition the observer estimation error $\tilde{z}$ will converge. Note that $\hat{\theta}_S$ is initialized as $\hat{\theta}_S(0) = 0$ and it will not be updated until the detection of a sensor fault. Now select a positive definite Lyapunov function candidate as

$$V = \|\mu(x,t)\|_{2,n}^2 / (2\varepsilon) + \tilde{\theta}_s^T \tilde{\theta}_s / (2\beta), \tag{108}$$

with the update law (107) and the fact that $e(t) = \tilde{z}(0,t) + \Phi_S(t)\tilde{\theta}_S(t) = \mu(0,t) - F(0,t)\tilde{\theta}_S(t) + \Phi_S(t)\tilde{\theta}_S(t)$, the derivative of the Lyapunov function candidate is given by

$$\begin{aligned} \dot{V} = &-\|\mu_x(x,t)\|_{2,n}^2 - \tilde{\theta}_s^T(t)[\Phi_s(t) - F(0,t)]^T e(t) \\ &+\gamma\tilde{\theta}_s^T\hat{\theta}_s / \beta + d_l\int_0^1 \sqrt{\mu^T(x,t)\mu(x,t)}dx / \varepsilon \end{aligned}, \tag{109}$$

$$\Rightarrow \dot{V} \le -(\pi^2 - 4)\|\mu(x,t)\|_{2,n}^2 / 8 - \gamma\tilde{\theta}_s^T\tilde{\theta}_s / (2\beta) + \gamma\theta_{s\max}^2 / (2\beta) + d_l^2 / (4\varepsilon^2). \tag{110}$$

Then, $\dot{V} < 0$ when

$$\|\mu(x,t)\|_{2,n} > \frac{4\gamma\varepsilon^2\theta_{s\max}^2 + 2\beta d_l^2}{(\pi^2 - 4)\beta\varepsilon^2} \;\; or \;\; \|\tilde{\theta}_s(t)\| > \sqrt{\theta_{s\max}^2 + \frac{\beta d_l^2}{2\gamma\varepsilon^2}}. \tag{111}$$

Therefore, μ and $\tilde{\theta}_S$ are ultimately bounded with the bounds defined above. Since $\tilde{z}(x,t) = \mu(x,t) - F(x,t)\tilde{\theta}_S(t)$, $\tilde{z}$ is also bounded because of the boundedness of $F(x,t)$ and thus $\tilde{v}$ is bounded due to (62).

Now, it will be shown that the tracking error $r(x,t)$ will remain bounded, with the following controller

$$\begin{aligned} U(t) &= v_d(1,t) + \int_0^1 K(1,\tau)[r(\tau,t) - \tilde{v}(\tau,t)]d\tau \\ &= v_d(1,t) + \int_0^1 K(1,\tau)[r(\tau,t) - \tilde{z}(\tau,t) - \int_0^\tau M(\tau,\eta)\tilde{z}(\eta,t)d\eta]d\tau \end{aligned}, \tag{112}$$

where $\tilde{v}(x,t)$ is given by (62) with observer state defined by (101). With controller defined in (112) the dynamics of state tracking error with a sensor fault is expressed as (39, 40) and (64).

Theorem 6 (Sensor Fault Accommodation): Let the control law in (112) be used upon detecting the sensor fault. Then the parameter estimation, observer estimation, and tracking errors are UB.

Proof: The dynamics of state tracking error $r(x,t)$ can be obtained as (39),(40) and (64). Applying transformation (41) to (39),(40) and (64) leads to (44) and (65). Now select a Lyapunov function candidate as

$$V = \frac{R}{2\varepsilon}\|\mu(x,t)\|_{2,n}^2 + \frac{\tilde{\theta}_s^T\tilde{\theta}_s}{2\beta} + \frac{\gamma}{4\varepsilon k_s}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx\,. \tag{113}$$

By taking the derivative of the above with respect to time we will arrive at

$$\dot{V} = \frac{R}{\varepsilon}\int_0^1 \mu^T(x,t)\mu_t(x,t)dx + \frac{\tilde{\theta}_s^T(t)\dot{\tilde{\theta}}_s(t)}{\beta} + \frac{\gamma}{4\varepsilon k_s}\int_0^1\int_0^x w^T(\eta,t)w_t(\eta,t)d\eta dx\,. \tag{114}$$

Substituting (104), (44) and the update law (107) in (114) and applying integration by parts yields

$$\begin{aligned} \dot{V} \le & -\frac{R(\pi^2-2)}{4}\|\mu(x,t)\|_{2,n}^2 + \mu^T(0,t)e(t) + \frac{d_k^2}{4c\varepsilon k_h} + \frac{\gamma\tilde{\theta}_s(t)\hat{\theta}_s(t)}{\beta} - e^T(t)e(t) \\ & -\frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx + \frac{\gamma}{2k_c}\int_0^1\int_0^x w^T(\eta,t)w_{\eta\eta}(\eta,t)d\eta dx + \frac{Rd_l^2}{2\varepsilon^2} \end{aligned}, \tag{115}$$

$$\begin{aligned}\Rightarrow \dot{V} \le & -\frac{R(\pi^2-2)}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx - \frac{e^T(t)e(t)}{2} \\ & + \frac{\gamma}{4k_c}\left[w^T(1,t)w(1,t) - w^T 0,t)w(0,t)\right] - \frac{\gamma}{2k_c}\int_0^1\int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta dx \cdot \\ & + \frac{\mu^T(0,t)\mu(0,t)}{2} + \frac{\gamma\theta_{a\max}^2}{2\beta} + \frac{d_k^2}{4c\varepsilon k_h} + \frac{Rd_l^2}{2\varepsilon^2} - \frac{\gamma\tilde{\theta}_s^T(t)\tilde{\theta}_s(t)}{2\beta}\end{aligned} \tag{116}$$

From (66) and (106) it can be seen that $w^T(1,t)w(1,t) \le \bar{k}\|\tilde{v}\|_{2,n}^2 \le k_h\|\tilde{z}\|_{2,n}^2 \le k_s[\tilde{\theta}_s^T\tilde{\theta}_s + \|\mu\|_{2,n}^2]$, where $k_s = max\{2k_h, 2k_h\bar{f}\}$ and $\bar{f} = \max_{0\le x\le 1}\|F(x,t)\|_2$. Therefore, it leads to

$$\begin{aligned}\dot{V} \le & -\frac{R(\pi^2-2)-\gamma-2}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{1}{2}e^T(t)e(t) - \frac{\gamma}{2\beta}\tilde{\theta}_s^T(t)\tilde{\theta}_s(t) \\ & -\frac{\gamma}{2k_c}[\int_0^1\int_0^x w_\eta^T(\eta,t)w_\eta(\eta,t)d\eta dx + \frac{1}{2}w^T 0,t)w(0,t)] \qquad . \\ & -\frac{\gamma c}{4\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx + \frac{\gamma}{2\beta}\theta_{a\max}^2 + \frac{Rd_l^2}{2\varepsilon^2} + \frac{\gamma}{4}\tilde{\theta}_s^T(t)\tilde{\theta}_s(t) + \frac{d_k^2}{4c\varepsilon k_h}\end{aligned} \tag{117}$$

Then, apply Poincare inequality [40] to arrive at

$$\begin{aligned}\dot{V} \le & -\frac{R(\pi^2-2)-\gamma-2}{4}\|\mu(x,t)\|_{2,n}^2 - \frac{\gamma(2-\beta)}{4\beta}\tilde{\theta}_s^T(t)\tilde{\theta}_s(t) \qquad . \\ & -\frac{\gamma(\varepsilon+2c)}{8\varepsilon k_c}\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx + \frac{\gamma\theta_{s\max}^2}{2\beta} + \frac{Rd_l^2}{2\varepsilon^2} + \frac{d_k^2}{4c\varepsilon k_c}.\end{aligned} \tag{118}$$

Therefore, $\dot{V} < 0$ if $R > \frac{\gamma+2}{\pi^2-2}$ and

$$\begin{aligned}& \|\mu(x,t)\|_{2,n} > \sqrt{\frac{2\gamma\varepsilon^2 ck_c\theta_{s\max}^2 + 2R\beta ck_c d_l^2 + \beta\varepsilon}{\beta\varepsilon^2 ck_c[R(\pi^2-2)-\gamma-2]}} \\ & or\ \|\tilde{\theta}_s(t)\| > \sqrt{\frac{2\theta_{s\max}^2}{2-\beta} + \frac{2R\beta d_l^2}{\gamma\varepsilon^2(2-\beta)} + \frac{\beta d_k^2}{\gamma\varepsilon ck_c(2-\beta)}} \\ & or\ \int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta dx > \frac{4\varepsilon k_c\theta_{s\max}^2}{\beta(\varepsilon+2c)} + \frac{4Rk_c d_l^2}{\gamma\varepsilon^2(\varepsilon+2c)} + \frac{2d_k^2}{4\gamma(\varepsilon+2c)}.\end{aligned} \tag{119}$$

Thus, μ and $\tilde{\theta}_s$ are ultimately bounded. Next $\tilde{z}$ is also bounded since $\tilde{z}(x,t) = \mu(x,t) - F(x,t)\tilde{\theta}_s(t)$ and $F(x,t)$ is bounded. It has been shown that $\int_0^1\int_0^x w^T(\eta,t)w(\eta,t)d\eta\, dx$ and $w(x,t)$ is continuous in $x \in [0,1]$, so $w(x,t)$ is bounded. Then we know that $r(0,t)$ is also bounded because $r(0,t) = w(0,t)$ from the transformation (41).

2.2.2.4. Time to Accommodation (TTA)

Unlike a fault-tolerant method, a fault accommodation scheme is activated after the detection of fault and by that time the system performance or output has deviated from the desired performance or output. In this case, it is important to find an estimate of the amount of time needed to bring the system back to its desired operation. For this purpose, time to accommodation (TTA) is introduced. The estimated TTA is defined as the time available before the tracking error decreases below a predefined limit with the fault accommodation scheme. TTA estimation using full states was first proposed in [41].

Given an initial value of the output tracking and estimated state tracking errors, and the tracking error limit, upon detecting and activating the fault accommodation scheme, the TTA can be estimated as $TTA(t) = \max_{1\leq i\leq n} t_{a(i)}(t)$ where

$$t_{a(i)} = \frac{1}{\lambda_i} Ln\left(\frac{\lambda_i \bar{\vartheta}_i + \varepsilon p_i \dfrac{\hat{r}(2h,t) - 2\hat{r}(h,t) + r(0,t)}{h^2}}{\lambda_i p_i r(0,t) + \varepsilon p_i \dfrac{\hat{r}(2h,t) - 2\hat{r}(h,t) + r(0,t)}{h^2}} \right). \tag{120}$$

The above formula is derived based on the tracking error dynamics (39). The transformation $\bar{r}(x,t) = Pr(x,t)$ is utilized when Λ is not diagonal to convert the dynamics of the tracking error (39) to $\bar{r}_t(x,t) = \varepsilon \bar{r}_{xx}(x,t) + \bar{\Lambda}\bar{r}(x,t)$ where $\bar{\Lambda} = P\Lambda P^{-1} = diag(\lambda_1, \lambda_2, \ldots \lambda_n)$ with $\lambda_i (i = 1,2, \ldots n)$ being the eigenvalue of Λ and $P = [p_1^T, p_2^T, \ldots, p_n^T]^T \in \Re^{n\times n}$.

By using finite difference method, $\bar{r}_{xx}(x,t)$ can be derived as

$$\bar{r}_{xx}(x,t) = Pr_{xx}(x,t) = P \lim_{h\to 0^+} [r(x+2h,t) - 2r(x+h,t) + r(x,t)]/h^2 \,, \tag{121}$$

thus $\bar{r}_t(0,t)$ can be obtained as

$$\bar{r}_t(0,t) = \varepsilon P[r(2h,t) - 2r(h,t) + r(0,t)]/h^2 + \bar{\Lambda}\bar{r}(0,t) \,, \tag{122}$$

where $h > 0$ is a sampling interval. The solution $\bar{r}(0,t) = [\bar{r}_1(0,t), \ldots, \bar{r}_n(0,t))]^T$ to (122) in the interval $[t, t_{a(i)}]$ is given by

$$\begin{aligned} \bar{r}_i(0,t_{a(i)}) = & \int_t^{t_{a(i)}} e^{\bar{\Lambda}(t_{a(i)}-\tau)} \varepsilon p_i^T [r(2h,\tau) - 2r(h,\tau) + r(0,\tau)]/h^2 d\tau \\ & + e^{\bar{\Lambda}(t_{a(i)}-t)} p_i^T r(0,t), \forall t_{a(i)} > t \; and \; i = 1,2,\ldots,n \end{aligned} \,, \tag{123}$$

where t is the current time instant and $t_{a(i)}$ is the future time when the value of $\bar{r}_i(0,t)$ decrease to its corresponding limit $\bar{\vartheta}_i = p_i[\vartheta_1, \vartheta_2, \dots, \vartheta_n]^T$ for the first time where $\vartheta_i(i = 1,2,\dots,n)$ is the limiting value of output tracking error $r_i(0,t)$. Assume that the term $r(2h,\tau) - 2r(h,\tau) + r(0,\tau)$ is held in the interval $\left[t, t_{a(i)}\right]$. Then it can be show that

$$t_{a(i)} = \frac{1}{\lambda_i} Ln\left(\frac{\lambda_i \bar{\vartheta}_i + \varepsilon p_i^T [r(2h,t) - 2r(h,t) + r(0,t)]/h^2}{\lambda_i p_i^T r(0,t) + \varepsilon p_i^T [r(2h,t) - 2r(h,t) + r(0,t)]/h^2} \right). \tag{124}$$

In (124), since $r(2h,t)$ and $r(h,t)$ are unknown, $\hat{r}(2h,t) = \hat{v}(2h,t) - v_d(2h,t)$ and $\hat{r}(h,t) = \hat{v}(h,t) - v_d(h,t)$ will be used instead, as given in (120). Because the output tracking error for all the states must be less than their limits, the TTA is obtained as the maximum among all the individual TTA given by (120).

The filter-based detection observer for distributed parameter systems presented in this chapter uses output measurement alone. Compared to ODE representation of DPS, the PDE-based observers provide a more accurate estimation of the state, which is beneficial to both fault detection and accommodation. Furthermore with the filter based observer, both actuator and sensor faults are accommodated provided they occur one at a time. Upon detection, the adaptive estimator incorporated in the observer provides valuable information about the fault function in order to estimate the time-to-accommodation. The filter based approach is critical when dealing with the implementation on practical systems.

References

[1] Eva Wu, N., Zhang, Y. and Zhou, K. (2000). Detection, estimation, and accommodation of loss of control effectiveness. *International Journal of adaptive control and signal processing,* 14, 7, pp. 775-795.

[2] Huang, S. N. and Kok Kiang, T. (2008). Fault Detection, Isolation, and Accommodation Control in Robotic Systems, *Automation Science and Engineering, IEEE Transactions on,* vol. 5, pp. 480-489.

[3] Zhang, K., Jiang, B., and Staroswiecki, M. (2010). Dynamic Output Feedback-Fault Tolerant Controller Design for Takagi Sugeno Fuzzy Systems With Actuator Faults, *IEEE Transactions on Fuzzy Systems,* vol. 18, pp. 194-201.

[4] Shen, Q., Jiang, B., Shi, P. and Lim, C. C. (2014). Novel Neural Networks-Based Fault Tolerant Control Scheme with Fault Alarm, *IEEE Transactions on Cybernetics,* vol. 44, pp. 2190-2201.

[5] Fan, L. L. and Song, Y. D. (2012). Neuro-adaptive model-reference fault-tolerant control with application to wind turbines, *IET Control Theory & Applications,* vol. 6, pp. 475-486.

[6] Skaf, Z., Wang, H. and Guo, L. (2011). Fault tolerant control based on stochastic distribution via RBF neural networks," *Journal of Systems Engineering and Electronics,* vol. 22, pp. 63-69.

[7] Jain, S. and Khorrami, F. (1997). Decentralized adaptive control of a class of large-scale interconnected nonlinear systems, *Automatic Control, IEEE Transactions on,* vol. 42, pp. 136-154.

[8] Huang, S. N., Tan, K. K. and Lee, T. H. (2002). A decentralized control of interconnected systems using neural networks, *Neural Networks, IEEE Transactions on,* vol. 13, pp. 1554-1557.

[9] Koo, G., Park, J. and Joo, Y. (2011). Robust fuzzy controller for large-scale nonlinear systems using decentralized static output-feedback, *International Journal of Control, Automation and Systems,* vol. 9, pp. 649-658.

[10] Qureshi, A. and Abido, M. (2014). Decentralized discrete-time quasi-sliding mode control of uncertain linear interconnected systems," *International Journal of Control, Automation and Systems,* vol. 12, pp. 349-357.

[11] Demetriou, M. A., Ito, K. and Smith, R. C. (2007). Adaptive Monitoring and Accommodation of Nonlinear Actuator Faults in Positive Real Infinite Dimensional Systems, *Automatic Control, IEEE Transactions on,* vol. 52, pp. 2332-2338.

[12] Boskovic, J. D. and Mehra, R. K. (2002). A decentralized scheme for accommodation of multiple simultaneous actuator failures, in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, pp. 5098-5103.

[13] Panagi, P. and Polycarpou, M. M. (2011). Decentralized Fault Tolerant Control of a Class of Interconnected Nonlinear Systems, *Automatic Control, IEEE Transactions on,* vol. 56, pp. 178-184.

[14] Jin, X. Z. and Yang, G. H. (2009). Distributed adaptive robust tracking and model matching control with actuator faults and interconnection failures, *International Journal of Control, Automation and Systems,* vol. 7, pp. 702-710.

[15] Hao, L. Y. and Yang, G. H. (2015). Robust adaptive fault-tolerant control of uncertain linear systems via sliding-mode output feedback, *International Journal of Robust and Nonlinear Control*, vol. 25, pp. 2461-2480.

[16] Ferdowsi, H. and Sarangapani, J. (2017). Decentralized Fault Diagnosis and Prognosis Scheme for Interconnected Nonlinear Discrete-Time Systems, *International Journal of Prognostics and Health Management (IJPHM),* vol. 8, no. 9, pp. 1-15.

[17] Ferdowsi, H. and Jagannathan, S. (2013). A unified model-based fault diagnosis scheme for non-linear discrete-time systems with additive and multiplicative faults, *Transactions of the Institute of Measurement and Control,* vol. 35, pp. 742-752.

[18] Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica* 26, 3, pp. 459-474.

[19] Gertler, J. J. (1988). Survey of model-based failure detection and isolation in complex plants. *Control Systems Magazine, IEEE* 8, 6, pp. 3-11.

[20] Gertler, J. (1998). Fault detection and diagnosis in engineering systems. CRC press.

[21] Dochain, D. (2001). State observation and adaptive linearizing control for distributed parameter (bio) chemical reactors. *International Journal of Adaptive Control and Signal Processing*, 15, 6, pp. 633-653.

[22] Beniich, N., El Bouhtouri, A. and Dochain, D. (2010). Input constrained adaptive tracking for a nonlinear distributed parameter tubular reactor. *International Journal of Adaptive Control and Signal Processing*, 24, 4, pp. 249-260.

[23] Sallberg, S., Maybeck, P. S. and Oxley, M. E. (2010). Infinite-dimensional sampled-data Kalman filtering and the stochastic heat equation. *Proc. of the Decision and Control*, December, pp. 5062-5067.

[24] Yu, D. and Chakravotry, S. (2012). A randomly perturbed iterative proper orthogonal decomposition technique for filtering distributed parameter systems. *American Control Conference,* Montreal, Canada.

[25] Patan, M. and Ucinski, D. (2005). Optimal activation strategy of discrete scanning sensors for fault detection in distributed-parameter systems. *Proc of the 16th IFAC world congress*, Prague, Czech Republic, pp. 4-8.

[26] Baruh, H. (1986). Actuator failure detection in the control of distributed systems. *Journal of Guidance, Control and Dynamics*, 2, pp. 181-189.

[27] El-Farra, N. H. and Ghantasala, S. (2007). Actuator fault isolation and reconfiguration in transport-reaction processes. *AIChE Journal* 53, 6, pp. 1518-1537.

[28] Ghantasala, S. and El-Farra, N. H. (2009). Robust diagnosis and fault-tolerant control of distributed processes over communication networks. *International Journal of Adaptive Control and Signal Processing*, 23, 8, pp. 699-721.

[29] Mhaskar, P., Gani, A., El-Farra, N. H., McFall, C., Christofides, P. D. and Davis, J. F. (2006). Integrated fault-detection and fault-tolerant control of process systems. *AIChE Journal* 52, 6, pp. 2129-2148.

[30] Demetriou, M., Ito, K. and Smith, R. C. (2004). On-line monitoring and accommodation of nonlinear actuator faults in positive real infinite dimensional systems. *Proc of the IEEE Conference on Decision and Control*, 3, pp. 2871-2875.

[31] Jagannathan, S. (2006). *Neural Network Control of Nonlinear Discrete –time Systems*. NY: CRC publications.

[32] Ioannou, P. A. (1986). Decentralized adaptive control of interconnected systems, *Automatic Control, IEEE Transactions on,* vol. 31, pp. 291-298.

[33] Sunan, H., Kok Kiong, T., and Tong Heng, L. (2003). Decentralized control design for large-scale systems with strong interconnections using neural networks, *Automatic Control, IEEE Transactions on*, vol. 48, pp. 805-810.

[34] Baccoli, A., Orlov, Y. and Pisano, A. (2014). On the boundary control of coupled reaction-diffusion equations having the same diffusivity parameters. *Proc. of the IEEE Conference on Decision and Control*, Los Angeles, US, pp. 5222-5228.

[35] Yao, Z., and El-Farra, N. H. (2011). Robust fault detection and reconfiguration in sampled-data uncertain distributed processes. *Proc. CDC*, Orlando, FL, pp. 4925-4930.

[36] Baniamerian, A. and Khorasani, K. (2012). Fault detection and isolation of dissipative parabolic PDEs: Finite-dimensional geometric approach. *Proc. of the American Control Conference*, Montreal, QC, pp. 5894-5899.

[37] Krstic, M. and Smyshlyaev, A. (2008). *Boundary control of PDEs: A course on backstepping designs*. SIAM.

[38] Smyshlyaev, A. and Krstic, M. (2005). Backstepping observers for a class of parabolic PDEs. *Systems & Control Letters*, 54, 7, pp. 613-625.

[39] Ferdowsi, H., and Jagannathan, S. (2018). Fault Diagnosis of Distributed Parameter Systems Modeled by Linear Parabolic Partial Differential Equations With State Faults. *Journal of Dynamic Systems, Measurement, and Control*, 140, no. 1, pp. 011010.

[40] Hardy, G. H., Littlewood, J. E. and Pólya, G. (1952). *Inequalities.* Cambridge University Press.

[41] Cai, J., Ferdowsi, H. and Jagannathan, S. (2015). Model-based actuator fault accommodation for distributed parameter systems represented by coupled linear PDEs. *IEEE Conference on Control Applications (CCA)*, pp. 978-983.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8
© 2020 Nova Science Publishers, Inc.

Chapter 3

Fault-Tolerant Systems for Unmanned Multirotor Aerial Vehicles

***Juan I. Giribet*[1,2,3,*]*, Claudio D. Pose*[1] *and Ignacio A. Mas*[1,3,4]**
[1]LAR-GPSIC - Facultad de Ingeniería, Universidad de Buenos Aires,
CABA, Argentina
[2]Instituto Argentino de Matemática "Alberto Calderón" (IAM),
CABA, Argentina
[3]Consejo Nacional de Investigaciones
Científicas y Técnicas (CONICET), Argentina
[4]Instituto Tecnológico de Buenos Aires, CABA, Argentina

Abstract

This chapter presents some recent results on fault-tolerant control systems for unmanned aerial systems, in particular for multirotor-type vehicles, commonly known as drones. Over the last years, these vehicles have become widely popular. Simplicity and cost-effectiveness have turned out to be very appealing and, as a consequence, an increasing number of applications have risen in many fields such as agriculture, surveillance, and photography, among others. As mission requirements become more demanding, the matter of fault tolerance emerges as a key challenge, especially if system certification is sought.

Here, the focus is placed particularly on rotor failures in multirotor vehicles, and a specific definition for fault tolerance is considered based on the maneuverability capabilities in case of a failure. A geometric analysis is presented to evaluate the fault tolerant capabilities of a given vehicle, together with an experimental validation. Then, the limitations of this concept are analyzed. Finally, a novel reconfigurable structure is proposed for a fault-tolerant hexarotor, that presents good flight performance in failure cases, together with experimental results.

Keywords: fault tolerance, fault detection and isolation, multirotor aerial vehicles, unmanned aerial systems

[*] Corresponding Author's Email: jgiribet@fi.uba.ar.

1. Introduction

Since the early 2000s, when the first commercial radio-controlled small quadrotor entered the market (the Draganflyer quadcopter in 1999, despite there was also a less-known small japanese quadcopter in 1991, the Keyence Gyrosaucer II E-570), the popularity and availability of multirotor aerial vehicles have grown exponentially. At first mainly used by hobbyists, driven by curiosity over the new technology and willing to test its limits, this novelty soon proved to present practical advantages over other flying systems. For example, the vertical take-off and landing (VTOL) capability of these vehicles, allowed them to be easily operated indoors or in reduced spaces. On the other hand, the ability to hover at any point during flight allows for a higher degree of safety for inexperienced users, as the aircraft can remain almost motionless with no commanded actions.

Around 2006, the Federal Aviation Administration (FAA) of the U.S.A. issued the first authorization to use unmanned aerial systems (UAS) and unmanned aerial vehicles (UAV) for commercial applications, and manufacturers of multirotors started to develop specific commercial products aimed at different sectors, such as the movie industry, where the use of these vehicles began to replace manned helicopters, or inspection of difficult-access civil structures (bridges, power lines, buildings) [1]. This also encouraged the production of multirotor vehicles for civil applications, mainly used for personal entertainment, but with an increasing focus on the field of aerial photography and filming. Several companies started designing ready-to-fly products that came with the full package: optimized multirotor design (mainly quadrotors, but also hexarotors), flight controllers with inertial sensors, digital compass and GPS, remote controller and, frequently, first-person view (FPV) video systems. These systems provided a friendly interface to configure the vehicle, as well as a step-by-step set of simple calibration instructions, which allowed even inexperienced users to have a fully functional system in just a few hours, and a professional user to have it ready in a matter of minutes. Also, different companies started producing autopilot controllers, small boards that contained a microcontroller and several sensors that could be configured to control different unmanned systems, were they aerial, terrestrial or aquatic. For UAS, features could range from a simple remote-controlled manual flight to fully autonomous solutions where a preset path could be given for the vehicle to follow. As the use of these products became massive, concerns about individuals' safety and collateral damage in case of accidents began to arise.

Before 2013, most of the (reported) drone related accidents in the U.S. territory were caused by military experimental drones in isolated areas near military bases. At the beginning of that year, a few companies released to the market quadrotor vehicles aimed at amateurs and hobbyists, that allowed to mount small high-definition digital video recorders (e.g., a GoPro camera) for aerial film-making and photography. At a reasonable cost, these products were quite successful, as they were also quickly adopted for professional uses. During that year, the number of reported drone accidents related to civilian-owned systems went up and started to be comparable to the number of military drones accidents, only to exceed them a couple of years later.

These accidents called attention on the need for stricter regulations for UAS that flew over crowds or near restricted zones. In 2014, some of the commercial UAV flight controllers manufacturers added a *no-fly* feature, that prevented the vehicle from entering pre-established restricted zones, even if the pilot wanted to fly manually into them.

It was only recently, in October 2017, when the first waiver to fly a UAV over people was granted by the FAA to the news network CNN [**?**] to be used for news coverage. The aircraft was the Snap, from Vantage Robotics, a lightweight vehicle, with shrouded blades, held together using magnets which, on impact, should come apart and minimize possible damages. While this is not a fault tolerant system but rather a safety measure to prevent harm to third parties, it presents a practical solution to one of the main dangers of flying in public spaces.

Currently, the manufacturer DJI is in the process of implementing ADS-B receivers on all its UAS above 250g, that will allow to detect nearby aircraft, thus creating dynamically no-fly zones and therefore increasing airspace safety [3].

There exist several lines of work regarding fault tolerance in multirotors that deal with many different failures in this type of systems. However, there are no clear requirements in any country regarding this issue, neither for personal or professional use of such vehicles.

The next section introduces the working principles of multi-rotors necessary to develop the main ideas of the chapter.

1.1. Working Principles of Multirotors

Generally, a multirotor is an aircraft with three or more rotors, where the flight control is based on the speed variation of each rotor. In standard commercial vehicles, the structure (called frame) is commonly composed of a center where the electronic components and power source are mounted, and several arms that extend radially from that center, at which end the rotor is placed. These arms have all the same length and are uniformly distributed in a circle. Generally, a four-rotor aircraft (quadrotor) is enough for most basic personal and commercial applications, including good quality aerial photography and filmmaking with small cameras, but when heavier payloads are required or more stability is needed, vehicles with six and eight rotors are a preferred choice. Other common configurations include those where each arm of the frame has two coaxial rotors, both generating thrust in the same direction but with propellers rotating in opposite directions. In this way, the weight of the arms is reduced, but the combined thrust is not twice that of one single motor due to aerodynamic effects. These alternative multirotor configurations are shown in Figure 1.

Multirotors do not rely on complex mechanics to maneuver, but instead are based on speed (and thus force) variation of the set of motors. Suppose a standard hexarotor, with six arms of the same length, uniformly distributed in a circle. Then, the individual actions of the rotors to perform basic maneuvers are described in Figure 2. Consider a reference frame fixed to the vehicle (body frame), where the origin is at the center of the vehicle coincident with the center of mass, the x axis points to the front of the vehicle, the y axis to the right, and the z axis downwards. If the rotors are generating all the same thrust, and the total force is equal to the weight of the vehicle, then no torque is exerted in any of the three axes, and the vehicle remains still in the air. If all the rotors' forces are increased or decreased equally as in Figure 2.a, then the vehicle will ascend or descend. To tilt the nose of vehicle downwards (over the y axis, defined as the *pitch* angle), the rear rotors should increase their force while the front ones should decrease them, as shown in Figure

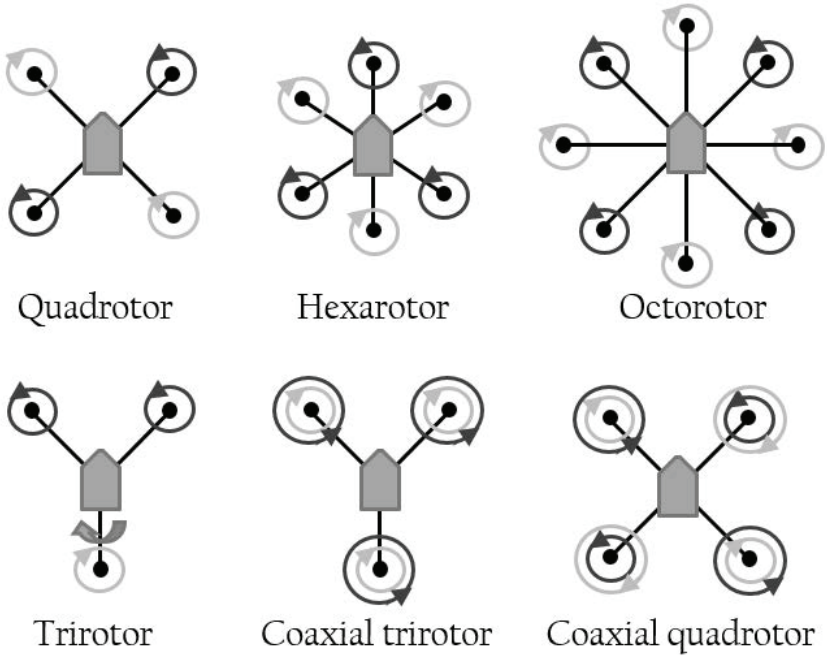


Figure 1. Common types of multirotors.

2.b. Analogously, to tilt the vehicle sideways (over the x axis, defined as the *roll* angle), the right (or left) rotors should increase their force while the left (or right) ones should decrease them, as shown in Figure 2.c. A rotation over the z axis, called a maneuver in yaw, is also exerted by rotor speed variation. As there is a rotor with a propeller rotating at the end of each arm, by conservation of angular momentum there appears an opposite torque in the frame over each rotor's axis, coincident in this case with the z axis of the vehicle. If all propellers rotated in the same direction, that would result in an uncompensated torque in the z axis that could not be controlled. The solution is to use two types of propellers, called CW and CCW (clockwise and counter-clockwise rotating propellers), which are identical in their construction, but generate thrust in the same direction while rotating in opposite directions. Then, CW and CCW propellers are used (generally) alternately in the vehicle's rotors, and a yaw maneuver can be performed by increasing the speed of the CW (or CCW) rotors, while decreasing the speed of the CCW (or CW) ones.

The yaw axis in standard multirotors is the one in which it is, in some sense, most difficult to exert torque. This is because the torque produced in the z axis due to the lightweight spinning propeller is significantly lower than the torque in the other two axes, between 10 and 50 times, according to experimental tests [4, 5]. This will present an important limitation when dealing with failures in rotors, as, usually, it is the yaw axis the one that restricts the most the maneuverability of the system.

To provide a better understanding of how the maneuvers described in Figure 2 are performed, a brief description of multirotors mechanics and dynamics is presented next. Modern multirotors consist of three important groups of components: the frame, the flight computer, and the power system. In general, the frame is the mechanical structure over which all

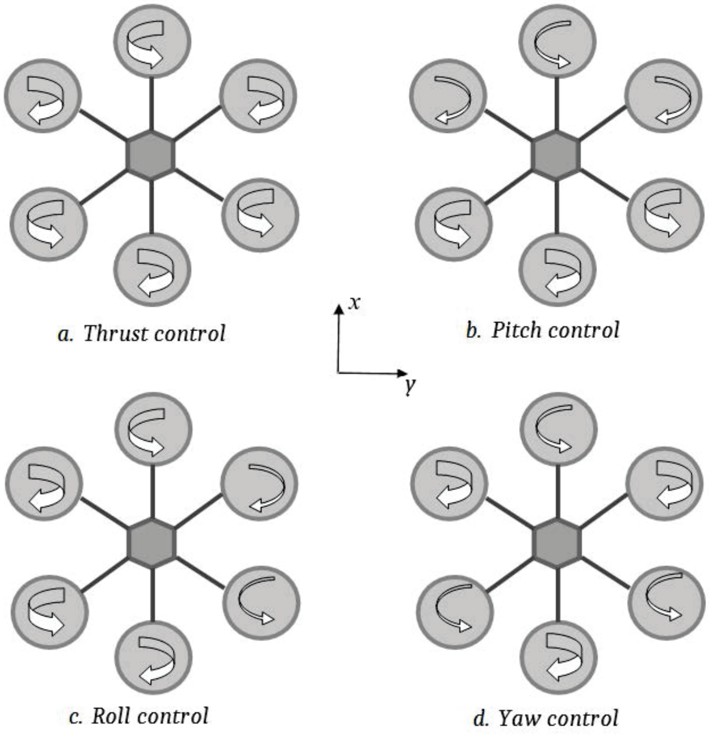


Figure 2. Multirotor control mechanics.

the rest of the components are mounted, and it is generally fixed, but may also have mobile parts (e.g.,, in trirotors, servos to move the rotors are needed to control 4 DOF [6, 7]). The flight computer comprises all the electronics (except for the actuators) that are mounted on the vehicle: the necessary sensors (Inertial Measurement Unit (IMU), compass, barometer, GPS, among others), a microcontroller that runs the algorithms for position and attitude estimation, the control algorithm, and additional tasks, and other components such as voltage regulators, visual and sound indicators, and radio-control receivers. Finally, the power system is composed of a main power source (LiPo and LiIon batteries most frequently used), and the actuators. Each actuator set includes a motor, usually of the BLDC type (brushless, DC powered, synchronous), a propeller, commonly of the fixed or foldable type (in rare cases with variable pitch), and an electronic speed controller (ESC) which allows to convert DC voltage to an AC signal, in order to control the speed of the motor. An example of the components for a standard quadrotor is shown in Figure 3.

In Figure 4, a simplified diagram of the data acquisition and control processes in a typical multirotor system is shown. The flight computer uses the information acquired by the different sensors into a fusion algorithm that outputs the best estimation of the orientation and position of the vehicle, in order to be used by the control system. Generally, two different control modes are used to operate a vehicle: commanding directly its orientation and vertical thrust, or generating a position reference that the vehicle has to follow.

In the first case, the orientation reference is compared with the estimation, and, if a difference exists, the orientation controller (commonly a PID) outputs a torque $q \in \mathbb{R}^3$ in

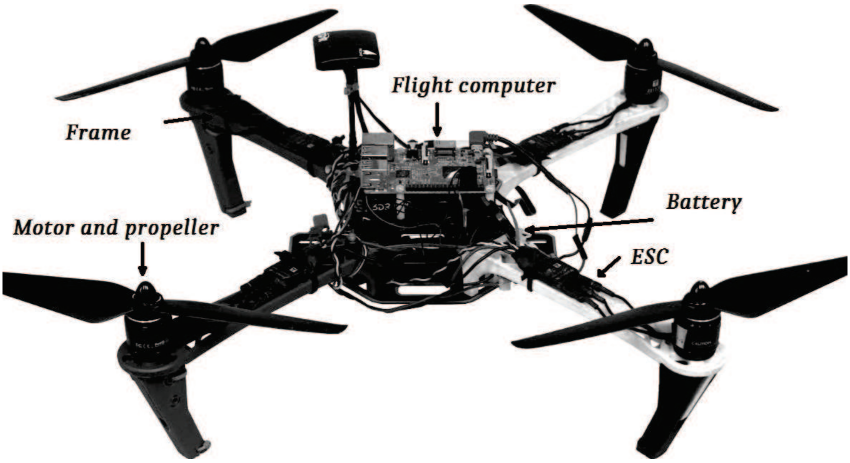


Figure 3. Main parts of a multirotor aircraft.

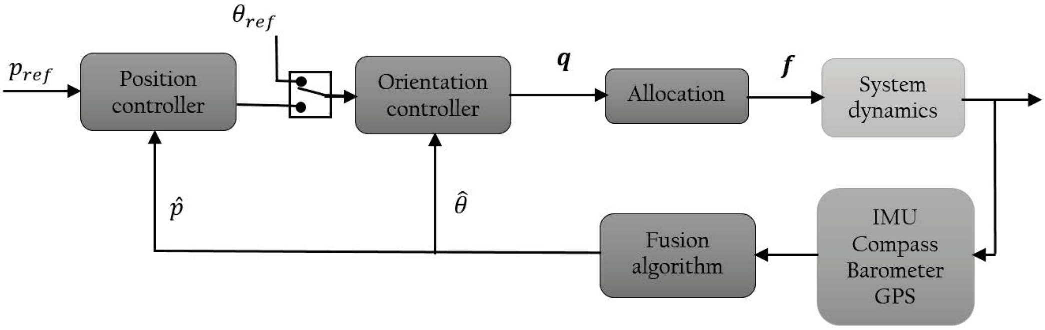


Figure 4. Simplified typical data acquisition and control diagram of a multirotor vehicle.

order to correct the deviation. This torque is converted into a set of forces to be commanded to each of the rotors of the aircraft by the allocation algorithm.

For the position control case, an outer control loop is added, where the position reference is compared with the estimation, and another controller computes the necessary maneuver to correct deviations, which is then commanded to the inner loop controller.

1.2. Notions of Fault Tolerance and Safety Actions

Fault tolerance is the ability of a system to continue operating properly in the event of one or more failures in some of its components. The key to define fault tolerance in multirotor aerial systems is how *proper operation* is defined for a given situation, and how is the system's degradation measured in the event of a failure. Based on this, different strategies can be adopted, being the following some of the most common.

In case of a failure, the vehicle is able to:

a) Fall to the ground in a safe manner to prevent harm to third parties (e.g., ultralight vehicle, or by means of a sound alarm that alerts of the falling vehicle).

b) Fall to the ground in a safe way to prevent harm to others and to itself (e.g., deploying a parachute or similar solution).

c) Descend and land in a controlled way when losing part of the sensor information, or of the control of the vehicle (e.g., failures of non-critical sensors, rotor failures in over-actuated systems).

d) Continue flying, and modify the flight plan according to the degraded capabilities of the system (e.g., shorten the flight plan, avoid certain objectives).

e) Continue flying normally (only for cases when the failure does not affect the performance of the vehicle).

Item **(a)** is the minimum allowed degree of safety for responsible users to operate this kind of aircraft, in order to minimize possible damage. Generally, these aircraft are used either away from populated areas where a falling vehicle would not cause damage to other people (hobbyist flights, aerial photography of landscapes, etc.), or in zones where the attendees are aware of the risks involved and adequate security measures are taken (drone racing, professional aerial filming, etc.). Still, collateral damages to wildlife or wildfire ignitions are within the possibilities. In some cases, this is the only possible solution to deal with some kind of failures. For example, a total failure in the power system (i.e., the battery, due to a burnt wire, short circuit, or other) would leave the vehicle completely unresponsive, falling immediately to the ground. Then, the only option to minimize damages is if the vehicle is light enough not to cause damage on impact, and/or try to warn bystanders using a self-powered sound alarm.

Similar cases to **(a)** are considered in the safety action taken in **(b)**. That is, the vehicle is still going to fall to the ground due to a failure, but further measures can be taken in order to guarantee the integrity of the vehicle and its payload, while also decreasing the risks of collateral damage. Parachute systems are a reasonable way to provide additional safety, at the cost of an increased weight, proportional to that of the vehicle [8]. This may, in turn, limit the allowed weight of the payload, or decrease the overall flight time. On the other hand, this provides a fast-deploying safety measure that can handle any type of failure of the vehicle, as long as the parachute activation system is independent from the main battery, to cover a possible failure in it. The vehicle should still fall to the ground without any direction control, but at a greatly reduced speed, giving more time for bystanders to move away, and decreasing the destruction on impact.

Item **(c)** considers some cases where, while the failure may be severe, there is a safe way to land the vehicle without major difficulties. For example, a vehicle that loses all position information (frequently GPS systems) or heading information (compass) while following a preset path, cannot continue the mission, and manual control has to be taken by the pilot to return the vehicle and land it safely. Another case, frequent in over-actuated systems, is a failure of a rotor that causes to lose control over some, but not all, degrees of freedom (DOF) of the vehicle. This is a case similar to a loss of the tail rotor of an helicopter, but, in the case of multirotors, while the vehicle starts to spin over the vertical axis uncontrollably, still allows position control while remaining stabilized, letting a pilot land it safely.

In **(d)**, it is considered that the vehicle suffers a failure in a component, and somehow its capabilities are degraded, but is still able to fly. For example, a vehicle that uses as a power source two batteries in parallel and suffers a failure in one of them, may still continue with its preset path or mission, but sees its flight time severely reduced, which may prevent it to complete the mission. This is also the case for vehicles for which rotor disposition allows the loss of a rotor while not losing degrees of freedom, but at a cost of a reduced maneuverability. Ideally, the vehicle should, as soon as a failure occurs, return to a safe point and land, but there are nowadays applications where the completion of the mission may be critical, as happens in medical and urgent aid related missions.

Finally, fault tolerance as defined in **(e)** considers the cases where a failure in a component is transparent from the point of view of the mission. For example, as IMU, digital compass, and other relevant sensors have a negligible weight in the vehicle, redundant sensors may be mounted within the flight computer, using one or another depending on the health of each [9, 10]. Thus, in case of a failure in one of them, the system could switch to another sensor, and the failure would pass as unnoticed from the mission point of view.

2. Advances in Multirotor Design and Fault Tolerance

Through the years, there have been many contributions that dealt with varied multirotor designs to approach several issues regarding both nominal operation and fault situations. As mentioned before, in a nominal condition, it is desired to control the aircraft in 4DOF, to be able to exert torques around the vehicle's three axes, and force in the vertical axis; then, the movement in the horizontal plane is accomplished by pitch and roll maneuvers, which is the working principle of standard multirotors with four or more rotors. Moreover, when six or more rotors are used, it is possible to control independently position and attitude (i.e., 6DOF).

In [11], the authors present a modification to the standard hexarotor design, by tilting the rotors' axis inwards (towards the center of the vehicle) to achieve a fully actuated vehicle, while in [12] it is also considered tilting the rotors sideways (along the arms' axis), where an optimal disposition of the rotors is obtained for a case where a desired trajectory is to be followed with the minimum control effort. The work done in [13] proves that almost any non-planar hexarotor structures (those where the rotors are not distributed in a co-planar way) can be approximated by a planar arrangement, by an adequate orientation of each rotor, thus reducing the problem of hexarotor design to the orientation of each rotor-propeller set. Also, it is shown that the inwards and sideways tilting angles may be optimized to obtain the best behaviour in translational and/or rotational dynamics.

Different overactuated hexarotor designs were proposed to achieve 6DOF control, such as those presented in [14, 15], where a servomotor is added to a hexarotor to tilt all the rotors sideways simultaneously, and the tilting angle can be optimized from the point of view of energy efficiency to follow a given trajectory.

In the works mentioned above, all the rotor-propeller sets are considered to be unidirectional, that is, the actuators can only exert force in one direction. Then, stating that such vehicles are capable of 6DOF control is not accurate, as the vehicles cannot exert force downwards in the vertical axis, but rather rely on the existence of the gravity force. This means that a vehicle is not controllable if it is turned upside down. To achieve true

6DOF control, it is proved in [16] that seven is the minimum amount of unidirectional rotor-propeller sets needed, and an example configuration of the rotors to achieve it is given. Other approaches to this issue are, for example, an octorotor with bidirectional actuators in a cube-like disposition [17] and in a rod-like disposition [18], and a twelve-rotor, six-arm vehicle with coaxial unidirectional actuators, that can be dynamically rotated over the arms' axes using servomotors [19].

Another advantage of over-actuated vehicles is that they could be able to compensate for the failure of one of the rotors with the action of the remaining ones, making possible to fly with a vehicle that does not lose maneuverability, in particular maintains its 4DOF (attitude and altitude), even when one of its rotors fails. Standard multirotors, even some over-actuated aircraft, are not always capable to maintain control in 4DOF with a rotor failing, leading to different approaches to design vehicles that consider this issue.

2.1. Rotor Fault Tolerance Design

Back to the topic of fault tolerant control in case of rotor failures, one may ask if any of the structures presented above represent a more convenient choice when dealing with rotor failures, whichever the nature of the failure may be.

For convenience in notation, when referring to the spinning direction of a rotor, a CW rotating rotor will be defined as a **P** rotor and a CCW one as an **N** rotor. Then, a PNPNPN hexarotor will represent a vehicle in which consecutive rotors have alternate spinning directions.

In [20], a thorough analysis is made over multirotor vehicles with six or eight unidirectional actuators, distributed in a co-planar way over a regular polygon, with all their axes perpendicular to this plane, considering the attainable torque set for a hovering state. It is shown that PNPNPN hexarotors cannot maintain 4DOF control in case of a total failure (impossibility to exert thrust) in any of its rotors, but a PPNNPN design allows for 4DOF control only in the cases of a fault of this kind in the first four rotors, and even in a particular case of two rotors in failure. Also, it is proved that an octorotor vehicle, both for a PNPNPNPN and a PPNNPPNN configuration, does not lose control over any DOF in case of a single failure in a rotor, and even most cases of two failures. A more thorough analysis of the PPNNPPNN octorotor planar vehicle can be found in [21], where the control allocation problem is analyzed for particular cases of simultaneous failures in 1, 2, 3 and 4 rotors.

Other complex mechanical solutions are proposed in the literature to address total rotor failures, as the one proposed in [22], where a fault tolerant quadrotor structure is proposed by adding servomotors to reconfigure the position and orientations of the rotors in-flight, which rendered it capable of rejecting perturbations even in the event of a failure in two of its rotors.

Another kind of more relaxed solutions for cases of rotor failures in multirotors are those in which the vehicle relinquishes control over one DOF, but still being able to fly in a predictable way. One example of this solution in the one presented in [23], where a standard PNPNPN hexarotor gives up control of the yaw angle in order to maintain control over pitch, roll, and vertical force, allowing it to hover in a static position while spinning uncontrollably in the vertical axis. As mentioned before, the torque in the z axis is more limited with respect to the other axes due to the nature of the maneuver (also, the moment of inertia in this axis tends to be higher). Moreover, maintaining pitch, roll and vertical force control allows to control the position of the vehicle without restrictions, thus permitting to land it safely. Another example is shown in [24], where a quadrotor is considered tolerant to a failure in 1, 2 or 3 rotors, if it is able to fly in the vicinity of a position reference point. This same principle is applied to the design of a single-rotor vehicle in [25]. However, as typical commercial systems are designed such that the weight of the vehicle is half of the maximum thrust provided by the set of actuators (for maximum maneuverability), this solution is not always feasible when there exist failures in at least half of the rotors.

There are different works that deal with rotor failures different to that of an incapability to exert thrust. In [26], analysis of a blockage in a rotor (locked at a given speed) yields that a quadrotor is not able to control 4DOF, but both a PNPNPN and a PPNNPN planar hexarotor are. In [27], an incremental backstepping controller is implemented to deal with modelling errors of the vehicle, as well as degradation in the efficiency of the rotors.

In [28] a hexarotor structure was proposed, consisting of a standard PNPNPN planar distribution where the rotors are tilted all at the same (fixed) angle towards or away from the center of the vehicle. Experimental validations of this result can be found in [29], where an inwards-tilted hexarotor is capable of hovering with limited maneuvering, and in [30], where instead of a zero pitch and roll hovering state, an optimal orientation of the vehicle is found in order to achieve a static flight position. Results presented in [28], were extended in [31] to obtain a vehicle independently controllable in 6DOF, even in case of any rotor failure.

The remainder of the chapter deals exclusively with rotor failures, particularly when a rotor loses all capability of exerting thrust and torque. Moreover, a multirotor aerial system will be considered fault tolerant only if, in case of a failure of a rotor, is still able to control independently attitude and altitude (4DOF).

In the following section we present a simple model for the unmanned multirotor system, which is useful to illustrate how it is possible to study the fault-tolerant control problem as a simple algebraic problem.

2.2. Vehicle Model

In a normal state of operation, each unidirectional rotor-propeller set produces a force $f_i \in [0, F_M]$, being F_M the maximum force at top speed. In practice, each motor's speed is commanded through a Pulse Width Modulated (PWM) signal u_i, which takes a value between 0 and 100%. Near the nominal operating point, a linear relation between the PWM signal and the exerted force is assumed, with $f_i = k_f u_i$. It is also considered that each motor exerts a torque on its spinning axis, $m_i = \pm k_t u_i$, where the sign depends on the spinning direction (CW or CCW), which may also be expressed as $m_i = \pm(k_t/k_f) f_i$. The k_f and k_t constants are usually established experimentally.

The total vehicle force $f \in \mathbb{R}^3$ and torque $q \in \mathbb{R}^3$ in the body frame coordinates (see Figure 5) satisfies the following equations:

$$f = k_f E u, \qquad q = (k_t E J + k_f H)\, u \tag{1}$$

$$E = [e_i^c]_{i=1,n}, \qquad H = [d_i^c \times e_i^c]_{i=1,n} = [h_i^c]_{i=1,n}, \tag{2}$$

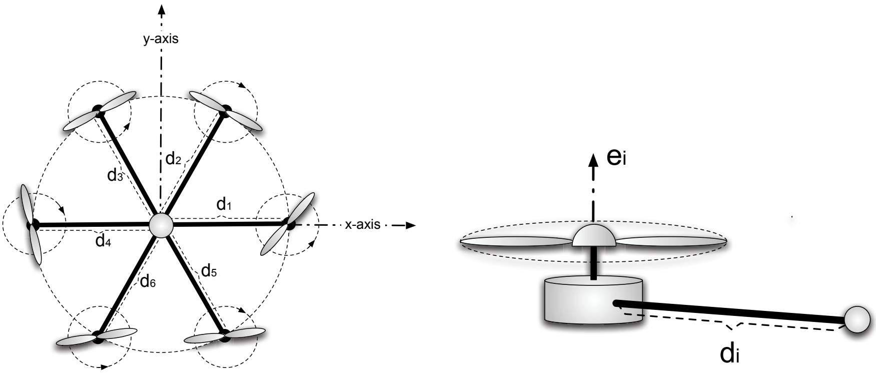


Figure 5. Hexacopter axes in standard configuration.

Here, the location of the center of mass of the i–th motor is given by the column vector $d_i^c \in \mathbb{R}^3$, and the direction of the corresponding force is given by the column vector $e_i^c \in \mathbb{R}^3$. Both vectors are represented in body frame coordinates. J, is a diagonal matrix with diagonal entries $j_{ii} = (-1)^{i+1}$, for $i = 1, ..., n$, for a PNPNPN configuration.

Let $A = (k_t EJ + k_f H)$, given a desired torque q, the control allocation algorithm finds (if there exists) $u \geq 0$ (with non-negative components) such that $q = Au$. The following theorem gives necessary and sufficient conditions for the existence of solutions of this problem [32].

Theorem 1. *Let $A \in \mathbb{R}^{3\times n}$. The following conditions are equivalent:*

1. *For each $q \in \mathbb{R}^3$ there exists $u \geq 0$ such that $q = Au$.*
2. *Matrix A has full rank and there exists $w \in \mathbb{R}^n$ with strictly positive components, i.e.,, $w > 0$ such that $Aw = 0$ i.e.,, w belongs to the kernel of A (denoted $ker(A)$).*

The proof this theorem gives an idea of how to find the solutions. First, suppose that Item 1. holds, and let $v > 0$ be an arbitrary vector with strictly positive components. Since for every $q \in \mathbb{R}^3$, there exists a vector $u \geq 0$ such that $Au = q$, in particular it is possible to find $u \geq 0$ such that $-Av = Au$, then $w = v + u \in \mathbb{R}^n$ is a vector with strictly positive components such that $Aw = 0$. On the other hand, if A has full rank, then for every $q \in \mathbb{R}^3$, there exists $v \in \mathbb{R}^n$ such that $Av = q$. If $w > 0$ satisfies $Aw = 0$, it is possible to find a number $0 < \alpha \in \mathbb{R}$ such that $\alpha w + v$ is a strictly positive vector, and $A(v + \alpha w) = q$.

The positive vectors $w > 0$ in the kernel of A, allow to construct the PWM signals $u \geq 0$ such that achieve the desired torque. Analyzing the rank and kernel of A, it is possible to see if with the given actuators' disposition, it is possible to achieve any desired torque.

2.3. Standard Hexacopter Configuration

Consider the vehicle in Figure 5, where all rotors are identical and their thrust and torque are exerted in the direction of the vehicle's z axis. In this case we have:

$$d_i^c = \ell \begin{bmatrix} c\alpha_i \\ s\alpha_i \\ 0 \end{bmatrix}, \qquad e_i^c = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad h_i^c = \begin{bmatrix} \pm\ell s\alpha_i \\ \mp\ell c\alpha_i \\ 0 \end{bmatrix} \tag{3}$$

where $\alpha_i = (i-1)\frac{\pi}{3}$ rad, $i = 1, ..., 6$ and $\ell > 0$ is the distance to each vertex of the hexagon. As a consequence,

$$A = \begin{bmatrix} k_f\ell & 0 & 0 \\ 0 & k_f\ell & 0 \\ 0 & 0 & k_t \end{bmatrix} \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ -1 & -0.5 & 0.5 & 1 & 0.5 & -0.5 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{4}$$

Notice that, $A \in \mathbb{R}^{3\times 6}$ has full rank and the vectors w in kernel of A can be written as $w = \begin{bmatrix} \alpha & \beta & \gamma & \alpha & \beta & \gamma \end{bmatrix}^T$, with $\alpha, \beta, \gamma \in \mathbb{R}$. Taking $\alpha, \beta, \gamma > 0$, from Theorem 1 it follows that a standard hexarotor aircraft can reach any desired torque.

If we assume that one of the rotor fails, for instance rotor 2, the force excerted by this rotor is zero. In this condition, if we want to study if the aircraft can reach any torque, we need to study if there is a vector in $ker(A)$ of the form $w = \begin{bmatrix} w_1 & 0 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$ with components $w_j > 0$, for $j = 1, 3, 4, 5, 6$.

With this idea, it is easy to prove that we need at least six rotors to have a fault-tolerant vehicle, because if $x, y, z \in \mathbb{R}^n$ are vectors such that $x_i = 0$ and $x_j > 0$ (for every $j \neq i$), $y_k = 0$ (for some $k \neq i$ and $y_j > 0$, for every $j \neq k$) and $z_l = 0$ (for some $l \neq i, l \neq k$ and $z_j > 0$, for every $j \neq l$), then x, y, z are linearly independent vectors. In fact, let $N = \begin{bmatrix} x & y & z \end{bmatrix}$ be the matrix containing vectors $x, y, z \in \mathbb{R}^n$ as columns. Observe that, there exists a row-permutation matrix $U \in \mathbb{R}^{nxn}$ such that

$$M = UN = \begin{bmatrix} 0 & m_{12} & m_{13} \\ m_{21} & 0 & m_{23} \\ m_{31} & m_{32} & 0 \\ m_{41} & m_{42} & m_{43} \\ \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} \end{bmatrix},$$

with $m_{ij} > 0$. Since

$$det \begin{bmatrix} 0 & m_{12} & m_{13} \\ m_{21} & 0 & m_{23} \\ m_{31} & m_{32} & 0 \end{bmatrix} > 0,$$

it follows that the columns of M are linearly independent vectors and then also are columns of N because U is invertible. So, if we want a matrix $A^{3\times n}$ such that A is full rank and vectors $x, y, z \in ker(A)$, then $n \geq 6$. The question is if with six rotors is enough.

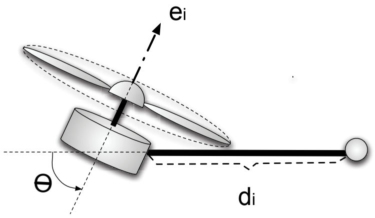


Figure 6. Tilted motor configuration.

2.4. Fault-Tolerant Hexacopter Design

Observe that, if one motor fails, matrix A loses one of its columns, the one corresponding to the failed rotor. Suppose that $A_j \in \mathbb{R}^{n\times 5}$ is matrix A after removing column j. For instance,

$$A_2 = \begin{bmatrix} k_f\ell & 0 & 0 \\ 0 & k_f\ell & 0 \\ 0 & 0 & k_t \end{bmatrix} \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ -1 & 0.5 & 1 & 0.5 & -0.5 \\ 1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{5}$$

So, to verify if the aircraft can be fully controllable in attitude if motor 2 fails, it is necessary to study if there exists a strictly positive vector in $ker(A_2)$, or analogously, if there is a vector in $ker(A)$ of the form $w = \begin{bmatrix} w_1 & 0 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$ with components $w_j > 0$, for $j = 1, 3, 4, 5, 6$.

Observe that matrix A_2 is a full rank matrix, but vectors $s \in ker(A_2)$ are of the form $s = \begin{bmatrix} \alpha & \beta & \alpha & 0 & \beta \end{bmatrix}^T$, where $\alpha, \beta \in \mathbb{R}$; since there is no vector in $ker(A_2)$, with strictly positive components, then the standard hexacopter is not fault tolerant if motor 2 fails. In fact, it is easy to see that it is not fully controllable if any of its motors fails. The vehicle can not reach torques in a particular direction. To overcome this limitation, a tilted-rotor hexacopter design proved to be useful.

Tilt-rotor aircraft have been widely used for different reasons, for improving the maneuverability [12, 14], power efficiency [33] or fault tolerance [28, 30, 31, 34]. In the next section, based on Theorem 1, it is shown that by tilting the rotors (or arms) a fixed-angle, it is possible to design a hexarotor vehicle capable of preserving 4DOF if any of its motors fails. We start with the simplest design and then we summarize the most relevant generalizations of this design. We also present some limitations and appropriate solutions to overcome these limitations.

2.4.1. Tilted Configuration

Since the probability of failure is the same for each motor, it is reasonable to propose a symmetric configuration as shown in Figure 5. Consider a design where the rotors are tilted a fixed angle θ as shown in Figure 6, with the tilt angle being the same for all rotors ($i = 1, ..., 6$). The standard configuration is a particular case, where $\theta = \pi/2$. As a consequence, the matrix that relates the PWM signals u with the torque q is given by $A = A(\theta)$,

$$A = k_t \begin{bmatrix} -c\theta & \frac{1}{2}\left(c\theta + \frac{\sqrt{3}\ell k_f s\theta}{k_t}\right) & \frac{1}{2}\left(c\theta + \frac{\sqrt{3}\ell k_f s\theta}{k_t}\right) & -c\theta & \frac{1}{2}\left(c\theta - \frac{\sqrt{3}\ell k_f s\theta}{k_t}\right) & \frac{1}{2}\left(c\theta - \frac{\sqrt{3}\ell k_f s\theta}{k_t}\right) \\ -\ell\frac{k_f}{k_t}s\theta & \frac{1}{2}\left(\sqrt{3}c\theta - \frac{\ell k_f s\theta}{k_t}\right) & \frac{1}{2}\left(\frac{\ell k_f s\theta}{k_t} - \sqrt{3}c\theta\right) & \ell\frac{k_f}{k_t}s\theta & \frac{1}{2}\left(\sqrt{3}c\theta + \frac{\ell k_f s\theta}{k_t}\right) & \frac{1}{2}\left(-\sqrt{3}c\theta - \frac{\ell k_f s\theta}{k_t}\right) \\ s\theta & -s\theta & s\theta & -s\theta & s\theta & -s\theta \end{bmatrix} \tag{6}$$

where $c\theta = cos(\theta)$ and $s\theta = sin(\theta)$. It is not hard to see that A has full rank and $w = \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T \in ker\,(A)$, for any $0 \leq |\theta - \pi/2| < \pi/2$. So, according to Theorem 1, the tilted-rotor hexacopter is fully controllable if every rotor is working properly. But, in this case, it is also fully controllable after a failure in any of its rotors. Suppose that rotor 2 fails, it is easy to prove that A_2 has full rank. In fact,

$$\det\left(A_2 A_2^T\right) = \frac{27}{4} k_t^2 sin^2(\theta)\left[cos(2\theta)\left(k_t^2 - \ell^2 k_f^2\right) + \ell^2 k_f^2 + k_t^2\right]^2 \neq 0, \tag{7}$$

for every $0 \leq |\theta - \pi/2| < \pi/2$. Furthermore, let $w = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix}^T$, with:

$$\begin{aligned} w_1 &= \frac{1-\alpha}{\alpha+1} - w_4\frac{1-\alpha}{2\alpha} && (8)\\ w_2 &= 1 && (9)\\ w_3 &= 1 - \frac{1-\alpha}{2\alpha}w_4 && (10)\\ w_5 &= w_4 + \frac{1-\alpha}{\alpha+1}, && (11) \end{aligned}$$

where $\alpha = \frac{k_t}{\sqrt{3}\ell k_f \tan\theta}$ and $|\alpha| < 1$. This vector $w \in ker(A_2)$ and, if $0 < w_4 < |2\alpha/(1+\alpha)|$ it follows that $w > 0$. Then, for a symmetric and tilted configuration of rotors with angle $\theta \neq \pi/2$ and $|\tan\theta| \neq \frac{k_t}{\sqrt{3}\ell k_f}$, the hexacopter can reject perturbation torques in any direction in $\mathbb{R}^3$ in order to maintain its attitude, even with the failure of one of its rotors. Also, observe that $|\alpha| \to 0$, *i.e.,*, $\theta \to \pi/2$, is a desired condition since it maximizes thrust. In order to compute a practical value for the tilt angle θ, also the vehicle's thrust should be considered.

2.5. Thrust Equations

Let $v > 0$ be the thrust of the hexacopter, it can be computed as the sum of the forces in the z axis in body frame. Then, thrust v depends on the PWM signals $u \geq 0$, in the following way:

$$v = k_f \sin(\theta)\mathbf{1}^T u \qquad \text{with} \qquad \mathbf{1} = \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^T. \tag{12}$$

The mapping $u \to (q, v)$ is given by

$$\begin{bmatrix} q \\ v \end{bmatrix} = B(\theta)u = \begin{bmatrix} A(\theta) \\ k_f \sin(\theta)\mathbf{1}^T \end{bmatrix} u. \tag{13}$$

With the standard hexarotor configuration, i.e., with $\theta = \pi/2$, the vertical thrust is maximized. On one hand, as shown previously, fault tolerance cannot be achieved in this case. On the other hand, for every $0 < |\pi/2 - \theta| < \pi/2$ and $|\tan\theta| \neq \frac{k_t}{\sqrt{3}\ell k_f}$ it is possible to achieve any desired torque, with tolerance to one faulty rotor. It is expected to have a trade-off in the selection of θ, between the capability to reject torque disturbances and the ability to exert vertical thrust on the vehicle.

In order to address this issue suppose rotor number 2 is faulty, as before. The usual approach for the allocation of torque control commands q, if it exists, is to compute the actuator signal u of minimum norm. In case of failure in one rotor, there will be among all possible torque commands, a torque in a given direction that will more difficult to achieve. This particular *worst case* torque command (q_{wc}), whose direction induces a maximum over all minimum norm u actuator signals, will depend on the θ angle. In the case of $\theta = \pi/2$, as the torque gets closer to the worst case direction, the norm of vector u needed to allocate such a torque goes to infinity.

Suppose that based upon practical considerations a given bound $q_{max} > 0$ is set on the torque commands whose allocation is sought. Within all torques $q \in \mathbb{R}^3$ with $||q|| < q_{max}$, the following θ dependent function is proposed:

$$f(\theta) = \max_{\substack{q \in \mathbb{R}^3 \\ ||q|| < q_{max}}} \; \min_{\substack{B_j u = q \\ u \geq 0}} ||u|| \tag{14}$$

where $B_j = B_j(\theta)$ corresponds to matrix $B(\theta)$ with a failure in the j-th rotor, i.e., matrix $B(\theta)$ with column j removed.

For typical multirotors, the torques about the x and y directions are more important than those about the z direction. This is because angular accelerations about x and y change the vehicle's thrust direction, and therefore jeopardize position control [20, 23]. Thus, a weighted norm could be considered for $||q||$ in order to prioritize the x and y directions.

The objective is to compute the curve $f(\theta)$ as the one indicated in Figure 7 that plots the minimal motor forces $||u||$ needed to reject the worst case perturbation torques under motor failure.

The general idea is to determine a practical way to design the motor slant angle based on the worst case perturbation torque to be rejected and the minimum vertical thrust that maintains the hexacopter flying. From Figure 7 it can be observed that as θ approaches $\pi/2$ then the minimal force $u \geq 0$ needed to reach the worst case torque rapidly increases. On the other hand, as θ moves above or below $\pi/2$, the thrust is reduced according to $1/\sin(\theta)$. This establishes a compromise between the thrust reduction that can be afforded by tilting the rotors and the maximum perturbation torque that can be rejected after a failure of one rotor.

Although previous results provide a criteria to design the geometry of the vehicle analyzing the torque exerted, the fact that $u \geq 0$ does not consider the vehicle vertical thrust. In practice, this force $u \geq 0$ is chosen in such a way that it guarantees certain torque $q \in \mathbb{R}^3$ and vertical thrust $v > 0$.

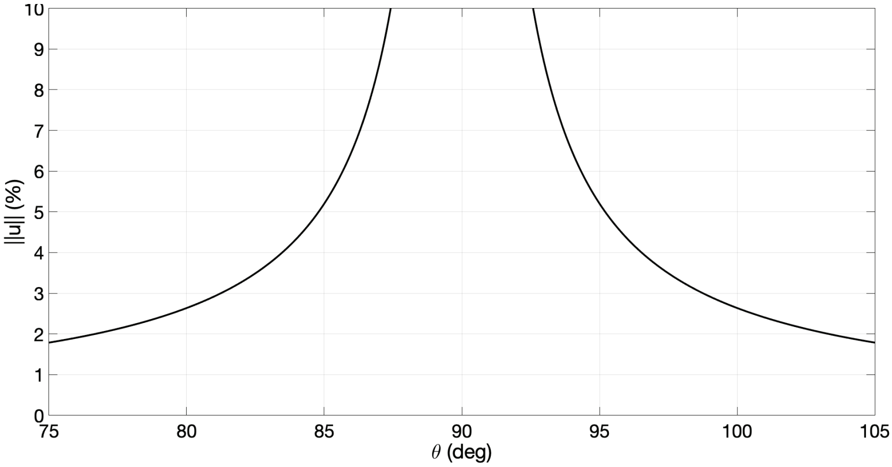


Figure 7. Minimum force for worst case torques.

2.5.1. Actuator Allocation

Assuming no rotor failures, in order to allocate a given pair torque/thrust $(q, v) \in \mathbb{R}^3 \times \mathbb{R}_+$, the actuators' signal $u \in \mathbb{R}^6$ is usually chosen as:

$$u_0 = B(\theta)^\dagger \begin{bmatrix} q \\ v \end{bmatrix} = \begin{bmatrix} A(\theta) \\ k_f s\theta \mathbf{1}^T \end{bmatrix}^\dagger \begin{bmatrix} q \\ v \end{bmatrix} \tag{15}$$

The reason to allocate u using the Moore-Penrose pseudoinverse $B^\dagger$, is that it renders the minimum norm solution. Other solutions based on generalized pseudo-inverses can improve the control allocation at the expense of a higher real time computational cost [35, 36].

Although it is possible to prove that for a given pair (q, v) there exists a positive solution $u \geq 0$ of Eq. (13) due to the existence of positive vectors in $ker\ (A(\theta))$, the positiveness of u_0 is not guaranteed. Let $C = A^\dagger$ and observe that, since $A = A(\theta)$ is full rank,

$$B(\theta)^\dagger = \begin{bmatrix} A \\ k_f s\theta \mathbf{1}^T \end{bmatrix}^T \left(\begin{bmatrix} A \\ k_f s\theta \mathbf{1}^T \end{bmatrix} \begin{bmatrix} A^T & k_f s\theta \mathbf{1} \end{bmatrix} \right)^{-1}. \tag{16}$$

Since $A\mathbf{1} = 0$ it follows that,

$$u_0 = \begin{bmatrix} A^T(AA^T)^{-1} & \frac{1}{6k_f s\theta}\mathbf{1} \end{bmatrix} \begin{bmatrix} q \\ v \end{bmatrix} = \begin{bmatrix} C & \frac{1}{6k_f s\theta}\mathbf{1} \end{bmatrix} \begin{bmatrix} q \\ v \end{bmatrix} \tag{17}$$

Then, a necessary and sufficient condition for the positiveness of u_0 is

$$\left| \min_{i=1,\dots,6} (c_i^r)^T q \right| \leq \frac{v}{6k_f s\theta}, \tag{18}$$

where c_i^r is the $i - th$ row of matrix C. A consequence of Eq. (18), is that for each torque q, it gives a lower bound on the total thrust $v = v(\theta)$ in order to guarantee $u_0 \geq 0$. A more restrictive condition that simplifies the calculations would consist in a thrust that doesn't depend on q, *i.e.,*, a thrust that assures $u_0 \geq 0$ in the worst case.

In [28] the following bound for the thrust is given. Let $0 < |\theta - \pi/2| < \pi/2$ and $q_{max} > 0$. Given a pair torque/thrust (q, v), if u_0 as given in Eq. (15), then $v > 0$ guarantees $u_0 \geq 0$ for every torque $\|q\| \leq q_{max}$ if and only if

$$v \geq k_f\, q_{max} \sqrt{\frac{(s\theta)^2(\ell^2 k_f^2 + 4k_t^2) + k_t^2(c\theta)^2}{\ell^2 k_f^2 k_t^2 (s\theta)^2 + k_t^4 (c\theta)^2}}. \tag{19}$$

This bound follows from a direct calculation of $\|c_i^r\|$. A similar bound can be obtained in the failure case.

When rotor j fails it is not as simple to find a condition as the one in Eq. (18) because $\mathbf{1} \notin ker\,(A_j(\theta))$. However, with additional calculations we can find similar conditions.

Suppose that $0 < |\theta - \pi/2| < \pi/2$, and define the following,

$$B_j(\theta) \ = \ \begin{bmatrix} A_j(\theta) \\ k_f sin(\theta)\mathbf{1}^T \end{bmatrix}, \qquad B_j^\dagger(\theta) = \begin{bmatrix} M & N \end{bmatrix} \tag{20}$$

for any $j = 1, ..., 6$, $M \in \mathbb{R}^{5\times3}$, $N \in \mathbb{R}^5$ and $\mathbf{1} \in \mathbb{R}^5$. Given the pair torque/thrust (q, v), let

$$u_0 = B_j^\dagger(\theta) \begin{bmatrix} q \\ v \end{bmatrix}. \tag{21}$$

As a consequence, $u_0 \geq 0$ if and only if

$$M\,q + N\,v \geq 0. \tag{22}$$

As in the case without faulty rotors, a lower bound on $v > 0$ is sought, such that the existence of $u_0 \geq 0$ can be guaranteed for every $\|q\| < q_{max}$ with $q_{max} > 0$. The bound can be obtained, if the inequality

$$n_i\, v \geq q_{max}\, \|m_i^r\| \tag{23}$$

is satisfied for every $i = 1, ...5$, where m_i^r is the i-th row of M, and n_i the i-th element of vector N. Then, $v > 0$ guarantees $u_0 \geq 0$ for every torque $\|q\| \leq q_{max}$ if and only if

$$v \geq q_{max} \max_{i=1,\ldots,5} \frac{\|m_i^r\|}{n_i}. \tag{24}$$

Equation (24) provides a very practical design tool in order to determine the tilt angle θ based on minimum vertical thrust v and maximum perturbation torque q_{max}.

In [28] it was proved that $n_i > 0$ if and only if $0 < |\theta - \pi/2| < \pi/2$ and $|\tan\theta| \neq \frac{k_t}{\sqrt{3}\ell k_f}$. Furthermore, in [29] the numbers n_i and $\|m_i^r\|$ are computed as a function of θ. However, a numerical calculation of these values are enough in order to determine the optimal value of θ.

2.6. Experimental Validation

To provide experimental proof of the fault tolerant capabilities of an inward-tilted vehicle, a standard hexarotor model was used, weighing $3\,\mathrm{kg}$, with a rotor to rotor distance of $0.55\,\mathrm{m}$, and with actuator sets capable of exerting $1\,\mathrm{kg}$ of thrust, with a constant $k_t/k_f = 0.014$. The rotors were tilted inwards at an angle $\theta = 73°$. The experiment consisted in the vehicle taking off the ground, performing a number of maneuvers, and landing, with rotor 2 turned off during the entirety of the flight, which lasted around $100\,\mathrm{s}$.

The orientation of the vehicle during the flight is presented in Figure 8, where it is shown that the vehicle is able to follow the commanded references, which include impulses of high magnitude but short duration. However, it shows a mild oscillating behaviour, specially during aggressive maneuvers. In Figure 9, the PWM commands applied to the motors during the flight are shown, where it can be noted that rotor 5, the opposite to rotor 2, is working very close to its lower saturation limit. The PWM command for rotor 5 is saturated at 16%, as a lower value would turn it off (if the rotor is constantly turning on and off, it is working in a highly non-linear zone, which is desirable to avoid).

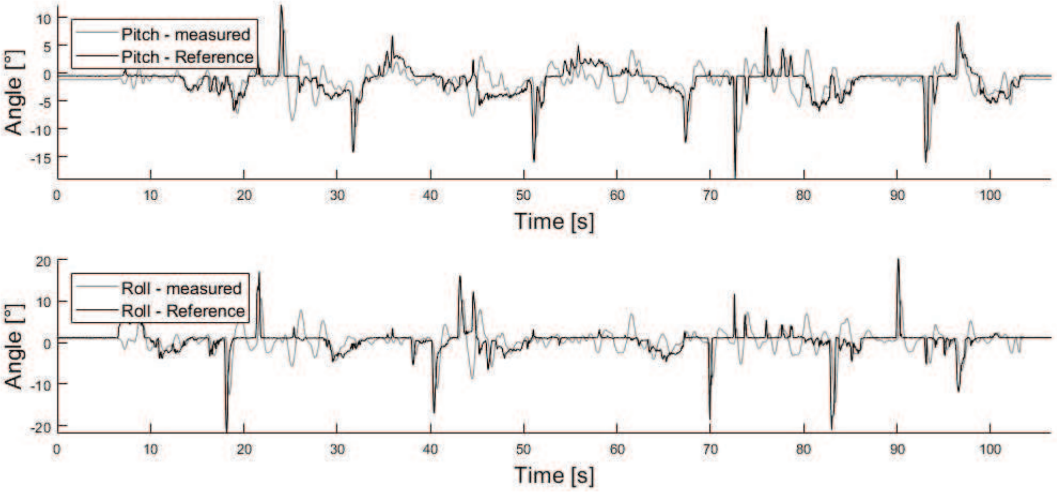


Figure 8. Orientation of the inwards-tilted hexarotor during a flight in which rotor 2 is turned off.

As the yaw angle torque is the most affected in case of a rotor failure, another experiment was carried out to analyze the performance in the z axis. In Figure 10, two opposite maneuvers in yaw are shown for the same vehicle, with rotor 2 turned off. The left one is performed by rotating the vehicle in the negative direction, maneuver which would require, in the nominal case, a reduction in the speed of rotor 2 (now turned off). In the failure case, it requires an increment in the speed of rotor 5, driving it away from saturation. The right plot of Figure 10 shows the opposite maneuver, rotating the vehicle in the positive direction, which would require an increase in the speed of rotor 2 in the nominal case, and in the failure case now pushes rotor 5 closer to saturation, presenting a worse behavior in the response, but still able to follow it.

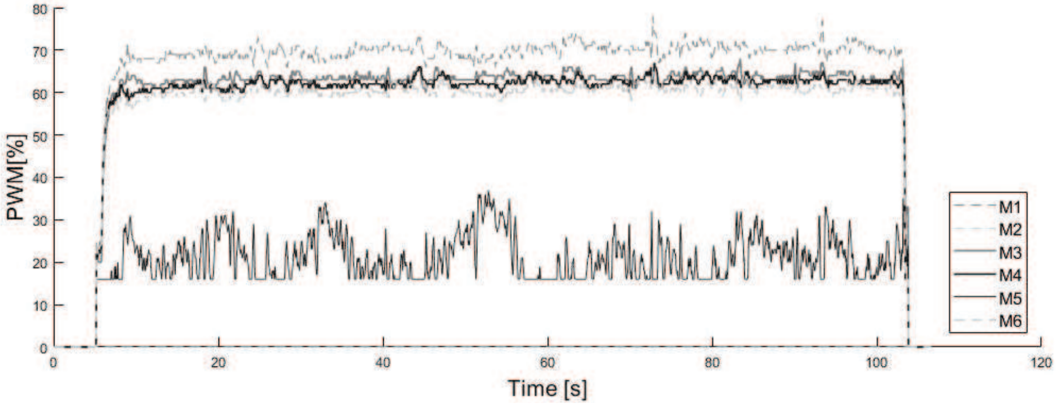


Figure 9. PWM commands of the inwards-tilted hexarotor during a flight in which rotor 2 is turned off.

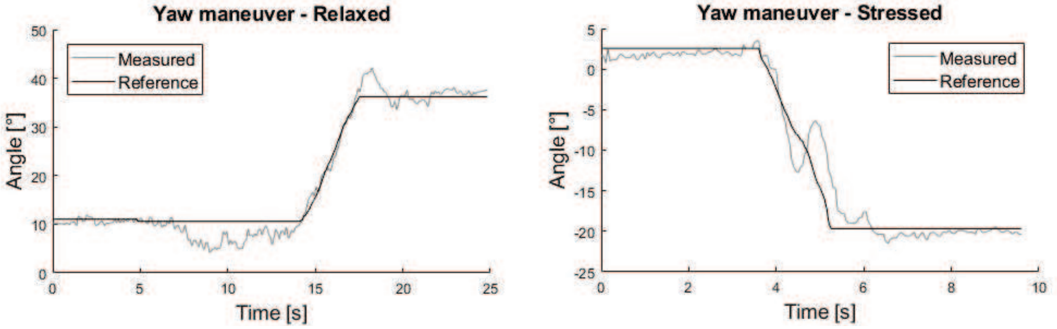


Figure 10. Yaw angle of the inwards-tilted hexarotor with rotor 2 turned off, during a yaw maneuver in the most relaxed direction (left) and in the most stressed direction (right).

2.7. Limitations of the Geometric Analysis

While the solution presented above is able to tell whether the vehicle is fault tolerant in case of a failure, it does not quantify how much its maneuverability is degraded.

Consider a standard hexarotor model, with the same physical characteristics as described above regarding size, weight and actuators, and suppose the motors are tilted inwards in order to have a fault tolerant vehicle. This time, the tilting angle is selected as $\theta = 65°$, allowing a 10% vertical thrust loss in the nominal case, a reasonable amount considering that the vehicle may carry an additional payload.

Considering that the vehicle is in hovering state (i.e., exerting a vertical force equal to its weight), Figure 11 shows the space of achievable torques both for the nominal and for a failure case in rotor 3. While the nominal vehicle shows a symmetric behaviour in its three axes (but with a ten times smaller torque in the z axis, as it is the axis in which it is most difficult to exert torque in), the vehicle with a failure shows a severe degradation in most directions, those in which the failing rotor has an appreciable contribution. As for the magnitude, the maximum torque that can be exerted in any direction, designated q_{max}, is $33.4 \times 10^{-3}\,\mathrm{kg\,m}$ for the nominal case, and $21 \times 10^{-6}\,\mathrm{kg\,m}$ for the failure case,

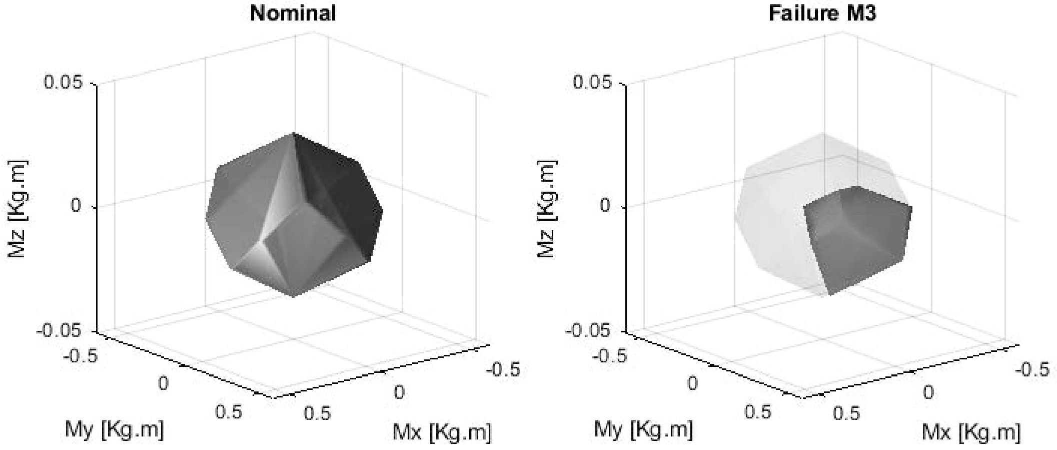


Figure 11. Achievable torque space for an inwards-tilted hexarotor in the nominal case (left), and for a case of a failure in rotor 3 (right).

three orders of magnitude lower. For the other cases of failure, the shape of the resulting achievable torque space is similar, but oriented in a different direction according to the rotor in failure.

The magnitude of q_{max} is given by a particular direction in the torque space, in which it is most difficult to exert torque in, and, both for the nominal and the failure case, is pointed almost parallel to the z axis. This suggests that q_{max} is highly influenced by the lower torque achievable in this axis, which is true for standard multirotor designs. A minimal value for this magnitude, in the considered vehicle, in order to achieve a reasonable flight performance under adverse weather conditions, is about $10 \times 10^{-3}\,\mathrm{kg\,m}$. Then, it is reasonable to assume that, in case of a failure, it will only be able to present an acceptable flight performance in controlled environments, and is not suitable for standard outdoor flight missions.

3. Improving the Maneuverability in Case of a Failure

At this point is notorious, for the proposed vehicles, the influence of the lower torque achievable in the z axis over the maneuverability of the vehicle when a failure occurs, and that the inwards-tilting solution, while theoretically valid, presents an unsuitable performance in real applications.

Regarding the experiments shown in Section 2.6, as rotor 2 (CW) is turned off, rotor number 5, which rotates in opposite direction (CCW), is forced to work near its lower saturation point. Increasing its speed would increment the torque it exerts in the z axis, but only two CW rotating rotors remain to compensate that increment. Then, if there existed a way in which the remaining CW rotors could increase the torque produced in the z axis, or in which all the CCW rotors decreased said torque, then it may be possible to drive the working point of rotor 2 away from its lower saturation.

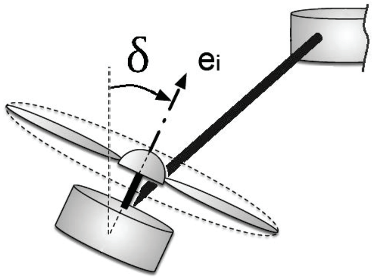


Figure 12. Side tilt angle δ. $\delta = 0$ is defined as the rotor pointing upwards, parallel to the vehicle's z axis, and $\delta > 0$ represents a tilt angle such that the torque the rotor exerts in the z axis is increased w.r.t. $\delta = 0$

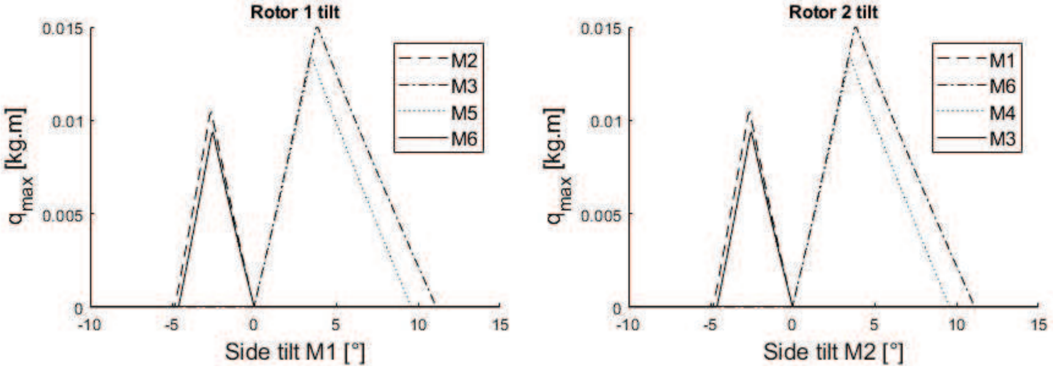


Figure 13. Maximum torque achievable in any direction in function of the tilt angle δ of rotor 1 (left) and rotor 2 (right), for each of the salvageable failures.

Consider now that, besides the fixed inwards tilt angle, rotors may be tilted around the arms' axis at an angle δ, as shown in Figure 12. It will be considered that $\delta > 0$ corresponds to a tilt angle such that the torque exerted in the z axis of the vehicle is increased, with respect to $\delta = 0$. This also means that the sense of rotation around the arm is different for CW and CCW rotors.

Going back to the proposed inwards-tilted hexarotor with $\theta = 65°$, suppose that one of the rotors is tilted at a fixed angle δ to analyze the behaviour of the vehicle in case of a failure. In Figure 13, it is shown how the q_{max} changes when tilting at a fixed angle only one of the rotors sideways, a CCW (rotor 1, left) or a CW (rotor 2, right), for each of the rotors in failure.

For rotor 1, a tilt angle $\delta_1 > 0$ improves q_{max} in cases of failure of rotors 3 and 5, which are all CCW rotating. In case one of the latter rotors fails, as rotor 1 can exert an increased torque in the z axis, it is able to compensate for the missing rotor by increasing slightly its speed. The opposite case occurs for a failure in rotors 2 and 6, which are CW rotating, as rotor 1 needs to lower the torque it exerts in the z axis, thus being better to tilt it at an angle $\delta_1 < 0$. As the vehicle is symmetric, a similar analysis is valid for a tilt in rotor 2, for a tilt angle $\delta_2 > 0$ for failures in rotors 4 or 6, and $\delta_2 < 0$ for failures in rotors 1 and 3. On the other hand, when a failure occurs in one of the rotors, the opposite tends to almost shut down, and is driven to a working point near the lower saturation limit. Hence, when rotor 4 fails, there is no noticeable improvement in q_{max} by tilting rotor 1, therefore it is not shown on the figure. An analogous situation occurs for a failure in rotor 5, as there is no improvement when tilting rotor 2.

The previous analysis shows that there does not exist an adequate fixed hexarotor structure to deal with all possible failures; but instead, if the system is able to actively change the sideways tilt of the rotors, it will be possible to improve its performance depending on which rotor fails. To accomplish this, at least two rotors should have tilting capabilities, and it should be implemented in a way such that one of them is CCW and the other CW, to achieve a better overall performance in case of failure. The tilting rotors cannot be placed opposite each other, as it is the case described above for a failure in rotor 4, unable to be compensated by tilting rotor 1. Therefore, in a PNPNPN hexarotor, the tilting rotors have to be placed in contiguous positions.

Consider that rotors 1 and 2 are selected to be actively tilted in-flight, then, for each of the possible failures, there is an optimal point to tilt either rotor 1 or 2 in order to obtain the highest q_{max}, being better to tilt rotor 1 for failures in rotors 3 and 5, and to tilt rotor 2 for failures in rotors 4 and 6 using $\delta > 0$. If rotor 1 or 2 fails, the only option is to tilt the remaining reconfigurable rotor at an angle $\delta < 0$.

Taking the best and worst cases presented above, in Figure 14, the achievable torque space is shown for the described hexarotor in case of a failure either in rotor 2 or 3, compensated by tilting rotor 1 adequately in order to obtain the best q_{max}. While the reconfigured fault tolerant solutions shows directions of preference when exerting torque, as the vehicle is still asymmetric, the volume of the achievable torque set is increased with respect to the non-reconfigurable case, as well as the magnitude of the torque achievable in all directions. The reconfigured system for a failure in rotor 3 shows a better performance than that for rotor 2, in accordance with Figure 13, obtaining a q_{max} of $14.7 \times 10^{-3}\,\mathrm{kg\,m}$ for the first case, and $9.3 \times 10^{-3}\,\mathrm{kg\,m}$ for the second. This represents a substantial increase in the maneuvering capabilities, rendering now the vehicle suitable for average outdoor flight conditions.

At this point, only the maneuverability of the vehicle in case of a failure has been analyzed, but in a real application, the system has to be able to detect a failure in a reasonable amount of time, and transition adequately between the nominal and the failure case. While the fault detection system is not analyzed here, previous results [37, 38] state that an adequate design for a similar system results in detection times under $400\,\mathrm{s}$. To analyze the real performance of the reconfigurable hexarotor, an experiment was carried out for an inward-tilted vehicle, with a distance of $0.55\,\mathrm{m}$ between rotors, with a weight of $3\,\mathrm{kg}$, carrying actuator sets of $1\,\mathrm{kg}$ of maximum thrust force. A micro servomotor was placed in rotor 1 in order to tilt it sideways in-flight, as shown in Figure 15.

The experiment consisted in the vehicle taking off in nominal conditions, with all rotors working, going into a hovering state and, while the reference pitch, roll and yaw commands remained at zero, injecting a failure in rotor 3 (turning it off). After $400\,\mathrm{s}$, rotor 1 is tilted to compensate the failure, the vehicle regains the hovering condition, and lands safely. Figure 16 shows the pitch, roll and yaw response of the vehicle for two identical experiments in an indoor environment, where the time axis is adjusted so that the failure is injected at t=$1\,\mathrm{s}$ and the system reconfigured at t=$1.4\,\mathrm{s}$. The vehicle is able to quickly recover, with both

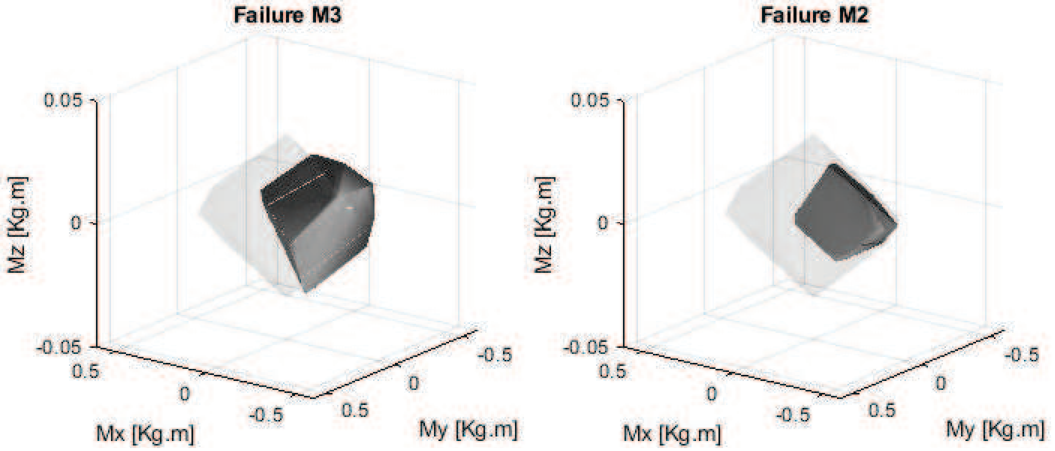


Figure 14. Achievable torque space for a reconfigurable hexarotor in case of a failure in rotor 3 (left) and for a case of a failure in rotor 2 (right). Rotor 1 is tilted at 3.8° and $-2.5°$ respectively, representing the best and worst cases for degraded maneuverability.

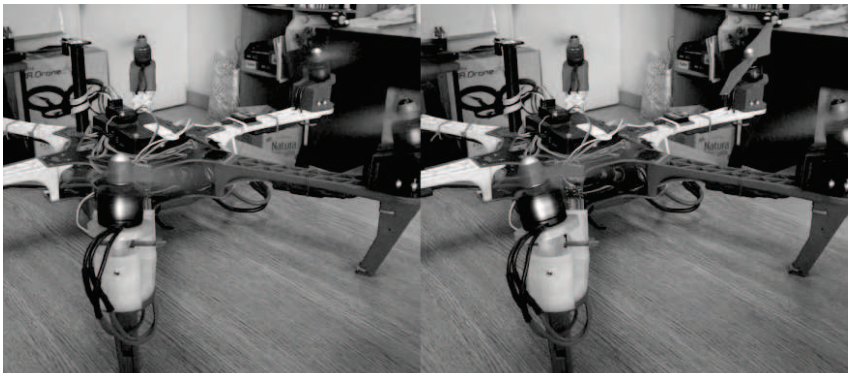


Figure 15. Inwards-tilted hexarotor with reconfigurable rotor 1.

trials presenting an almost identical performance.

The PWM commands of the rotors are presented in Figure 17, for one of the trials, where before the failure all the rotors are operating in a similar working point, and after reconfiguration are still well away from both upper and lower saturation limits, giving more room for speed variations to perform maneuvers.

3.1. Rotor Fault Detection and Isolation

When dealing with failures in any kind of system, an important step prior to the adaptation of the system to the fault, is the detection of the existence of a failure, and the isolation of its cause. While direct methods can be applied, such as direct condition monitoring of the actuators through rotor power and/or speed sensing [39, 40, 41, 42], these are often not the preferred methods, as they require the use of specific rotors or speed controllers, or the addition of specific sensors for health monitoring.

Figure 16. Angular response of an inwards-tilted hexarotor vehicle during two different experiments, where the fail is injected at t=1 s, and the vehicle is reconfigured after 400 s, maintaining all references at zero.

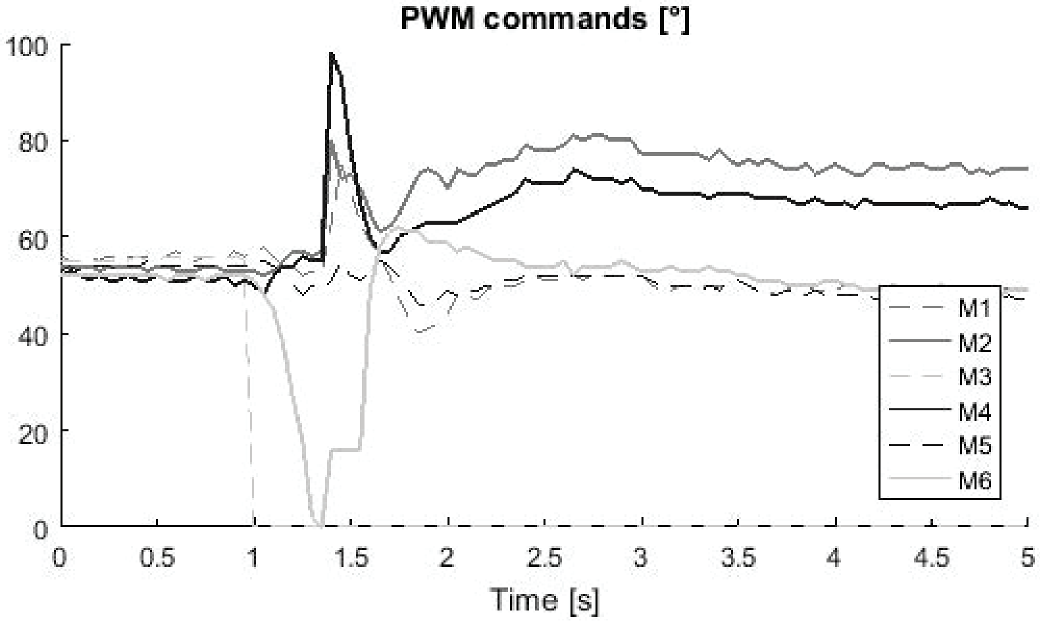


Figure 17. PWM command response of an inwards-tilted hexarotor vehicle corresponding to one of the reconfiguration delay experiments, where the fail is injected at t=1 s, and the vehicle is reconfigured after 400 s.

A more common method to approach rotor fault detection is through a dynamical model of the vehicle, where it is represented by a body with a given mass and inertia tensor, affected by the forces and torques generated by the actuators. While this still requires an accurate modelling of the vehicle, as well as of the actuator set, it is more versatile and does not require additional electronics. This allows to predict accurately the vehicle's behaviour when a set of forces is commanded to the rotors, and inconsistencies in the predictions with respect to the real behaviour may allow to detect rotor failures.

This approach was studied in the literature. In [43, 44], a sliding mode observer is proposed to deal with partial or total failures in an octorotor, combined with an LPV or dynamic control allocation for a 4DOF, fault tolerant octorotor.

In [45], a nonlinear observer is proposed for a coaxial octorotor, where the fault detection is achieved by a deviation of the expected behaviour through residues analysis, and the isolation of the fault is obtained by observing the direction of rotation in the three axes after a failure occurs. An improvement of this work is shown in [46], where a nonlinear sliding mode observer is used for detection with the same isolation technique, and experimental results for up to four specific rotor failures are presented.

The work in [37, 38] proposes a bank of Luemberger observers, one for the nominal plant, and one for each of the possible failures considered, where the detection and isolation of a total rotor fault (impossibility to exert force) is achieved by analyzing the residues. It also establishes the conditions needed to find a common virtual actuator that enables the system to recover from failures that are non-isolable. Simulated and experimental results of the detection and isolation algorithm are shown for a coaxial octorotor and a PPNNPN hexarotor, respectively.

There are also works that approach the issue of fault detection and isolation through the dynamical model by a statistical approximation instead of an analytical model, such as the works in [47], where an algorithm is presented based in supervised learning using a random forest classifier and in [48], where a statistical time series is used for the same purpose. This kind of modelling requires a great amount of training data of the behaviour of the system, either from the real vehicle analyzed, or from an accurate simulated model.

CONCLUSION

Nowadays, given the proliferation of unmanned aerial vehicles that has been possible due to a reduction in production costs and an increasingly simple operation, fault tolerance has become a critical issue to ensure safety, both of the system and of third parties.

While covering several issues that have been researched in the last years, this chapter has focused mainly in a particular type of failure in multirotor vehicles, that where one of the rotors fails completely and is incapable of exerting thrust or torque. This is one of the most critical failures in this kind of systems, as it affects its maneuverability and flight time. Additionally, a definition of fault tolerance was proposed, stating that is of interest to maintain independent control in attitude and altitude (4DOF) in case of a failure.

It was stated that to achieve fault tolerance in a standard multirotor, a minimum of six rotors is needed. Considering this, a geometrical analysis over the force-torque matrix of an hexarotor is presented; by analyzing the matrix rank and its null space, it is possible to assess whether the vehicle is fault tolerant or not. However, this analysis by itself cannot evaluate the performance of the vehicle in case of a failure. This fact was proved for a symmetric, inwards-tilted hexarotor that, while practically capable of 4DOF control in case of a failure, has its maneuverability extremely limited, and therefore is unsuitable for real outdoor missions.

To conclude, a reconfigurable inward-tilted hexarotor design was presented, which relied on actively tilting two of its rotors in case of a failure in order to improve maneuverability. This vehicle proved to be capable of transitioning adequately between its nominal and failure states, given that the fault is detected in a reasonable time. Nevertheless, it has to be taken into account that the need of electromechanical devices to tilt the rotors (servomotors in the presented case) is an additional source of possible failures, therefore designs of this kind should be approached carefully to minimize the impact on the reliability of the system.

As small multirotor vehicles become ubiquitous, taking concrete actions to ensure safety is paramount. Complementing the positive steps being already taken by the industry –for example with the definition of no-fly zones– with other approaches such as the ones proposed in this chapter will enable a diverse variety of applications from which society can greatly benefit from.

References

[1] Atkins, Ella, Aníbal Ollero, and Antonios Tsourdos. *Unmanned Aircraft Systems*. Wiley, New York, NY, USA, 1 edition, 2016.

[2] *CNN gets first FAA fly-over-people waiver*.

[3] *DJI Adds Airplane And Helicopter Detectors To New Consumer Drones*.

[4] Kharsansky, Alan. *Diseño e implementación de un sistema embebido de control de actitud para aeronaves no tripuladas* [*Design and implementation of an attitude control embedded system for unmanned aerial vehicles*]. Electronic Engineering Bachelor Thesis, Facultad de Ingeniería, Universidad de Buenos Aires, 2013.

[5] Saedan, M. and P. Puangmali. Characterization of motor and propeller sets for a small radio controlled aircraft. In *2015 10th Asian Control Conference (ASCC)*, pages 1–6, May 2015.

[6] Gonçalves, F. S., J. P. Bodanese, R. Donadel, G. V. Raffo, J. E. Normey-Rico, and L. B. Becker. Small scale uav with birotor configuration. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 761–768, May 2013.

[7] Salazar-Cruz S. and R. Lozano. Stabilization and nonlinear control for a novel trirotor mini-aircraft. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2612–2617, April 2005.

[8] Shao, P., C. Wu, and S. Ma. Research on key problems in assigned-point recovery of uav using parachute. In *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, pages 1–4, Oct 2013.

[9] Al Younes, Y., H. Noura, A. Rabhi, A. El Hajjaji, and N. Al Hussien. Sensor fault detection and isolation in the quadrotor vehicle using nonlinear identity observer approach. In *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 486–491, Oct 2013.

[10] López-Estrada, F. R., Ponsart, and D. et al. JC., Theilliol. Lpv model-based tracking control and robust sensor fault diagnosis for a quadrotor uav. *Journal of Intelligent & Robotic Systems*, 84(1-4):163–177, 2016.

[11] Shimizu, T., S. Suzuki, T. Kawamura, H. Ueno, and H. Murakami. Proposal of 6DOF multi-copter and verification of its controllability. In *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 810–815, July 2015.

[12] Rajappa, S., M. Ryll, H. H. Blthoff, and A. Franchi. Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4006–4013, May 2015.

[13] Tadokoro, Y., T. Ibuki, and M. Sampei. Maneuverability analysis of a fully-actuated hexrotor uav considering tilt angles and arrangement of rotors. In *IFAC-PapersOnLine*, volume 50, 07 2017.

[14] Ryll, M., D. Bicego, and A. Franchi. Modeling and control of fast-hex: A fully-actuated by synchronized-tilting hexarotor. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1689–1694, Oct 2016.

[15] Morbidi, F., D. Bicego, M. Ryll, and A. Franchi. Energy-efficient trajectory generation for a hexarotor with dual- tilting propellers. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6226–6232, Oct 2018.

[16] Tognon M. and A. Franchi. Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control. *IEEE Robotics and Automation Letters*, 3(3):2277–2282, July 2018.

[17] Brescianini D. and R. D'Andrea. Design, modeling and control of an omni-directional aerial vehicle. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3261–3266, May 2016.

[18] Park, S., J. Lee, J. Ahn, M. Kim, J. Her, G. Yang, and D. Lee. Odar: Aerial manipulation platform enabling omnidirectional wrench generation. *IEEE/ASME Transactions on Mechatronics*, 23(4):1907–1918, Aug 2018.

[19] Bodie, K., Z. Taylor, M. Kamel, and Siegwart R. Towards Efficient Full Pose Omnidirectionality with Overactuated MAVs. *International Symposium of Experimental Robotics (ISER)*, 2018.

[20] Schneider, T. *Fault-tolerant multirotor systems*. Master Thesis, Swiss Federal Institute of Technology (ETH), 2011.

[21] Marks, A., J. F. Whidborne, and I. Yamamoto. Control allocation for fault tolerant control of a vtol octorotor. In *Proceedings of 2012 UKACC International Conference on Control*, pages 357–362, Sep. 2012.

[22] Segui-Gasco, P., Y. Al-Rihani, H. Shin, and A. Savvaris. A novel actuation concept for a multi rotor uav. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 373–382, May 2013.

[23] Guang-Xun Du, Quan Quan, and Kai-Yuan Cai. Controllability analysis and degraded control for a class of hexacopters subject to rotor failures. *Journal of Intelligent Robotic Systems*, 2014.

[24] Mueller, M. W., and R. D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 45–52, May 2014.

[25] Zhang, Weixuan, M. W. Mueller, and R. D'Andrea. A controllable flying vehicle with a single moving part. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3275–3281, May 2016.

[26] Vey D. and J. Lunze. Structural reconfigurability analysis of multirotor UAVs after actuator failures. *54th Conference on Decision and Control*, pages 5097–5104, 2015.

[27] Falconi, G. P., V. A. Marvakov, and F. Holzapfel. Fault tolerant control for a hexarotor system using incremental backstepping. *IEEE Conference on Control Applications (CCA)*, pages 237–242, 2016.

[28] Giribet, J. I., R. S. Sanchez-Peña, and A. S. Ghersin. Analysis and design of a tilted rotor hexacopter for fault tolerance. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1555–1567, 2016.

[29] Giribet, J. I., C. D. Pose, A. S. Ghersin, and I. Mas. Experimental validation of a fault tolerant hexacopter with tilted rotors. *International Journal of Electrical and Electronic Engineering and Telecommunications*, 7(2):1203–1218, 2018.

[30] Michieletto, G., M. Ryll, and A. Franchi. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2747–2752, May 2017.

[31] J. I. Giribet, C. D. Pose, and I. Mas. Fault tolerance analysis of a multirotor with 6dof. In *4th Conference on Control and Fault Tolerant Systems (SysTol)*, Casablanca, Morocco, Sep 2019.

[32] Sanchez-Peña, R. S., R. Alonso, and P. A. Anigstein. Robust optimal solution to the attitude/force control problem. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3):784–792, July 2000.

[33] Holda, C., B. Ghalamchi, and M. W. Mueller. Tilting multicopter rotors for increased power efficiency and yaw authority. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 143–148, 2018.

[34] Michieletto, G., M. Ryll, and A. Franchi. Fundamental actuation properties of multirotors: Force-moment decoupling and fail-safe robustness. *IEEE Transactions on Robotics*, 34(3):702–715, June 2018.

[35] Ducard G. and M. D. Hua. Discussion and practical aspects on control allocation for a multi-rotor helicopter. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 38, pages 95–100, 11 2011.

[36] Pose, C. D., J. I. Giribet, and A. S. Ghersin. Hexacopter fault tolerant actuator allocation analysis for optimal thrust. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 663–671, June 2017.

[37] Vey, D., K. Schenk, and J. Lunze. Simultaneous control reconfiguration of systems with non-isolable actuator failures. In *2016 American Control Conference (ACC)*, pages 7541–7548, July 2016.

[38] Vey D. and J. Lunze. Experimental evaluation of an active fault-tolerant control scheme for multirotor UAVs. *3rd International Conference on Control and Fault-Tolerant Systems*, pages 119–126, 2016.

[39] Wolfram, D., F. Vogel, and D. Stauder. Condition monitoring for flight performance estimation of small multirotor unmanned aerial vehicles. In *2018 IEEE Aerospace Conference*, pages 1–17, March 2018.

[40] Dobra, P., M. Dobra, D. Moga, V. I. Sita, P. Dobra, M. Trusca, D. Moga, and R. A. Munteanu. Model based fault detection for electrical drives with BLDC motor. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 1–5, May 2014.

[41] Eissa, M. A., M. S. Ahmed, R. R. Darwish, and A. M. Bassiuny. Model-based sensor fault detection to brushless dc motor using luenberger observer. In *2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pages 1–6, Dec 2015.

[42] Zandi, O. and J. Poshtan. Fault diagnosis of brushless dc motors using built-in hall sensors. *IEEE Sensors Journal*, 19(18):8183–8190, Sep. 2019.

[43] Alwi H. and C. Edwards. LPV sliding mode fault tolerant control of an octorotor using fixed control allocation. In *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 772–777, Oct 2013.

[44] Alwi, H., M. T. Hamayun, and C. Edwards. An integral sliding mode fault tolerant control scheme for an octorotor using fixed control allocation. In *2014 13th International Workshop on Variable Structure Systems (VSS)*, pages 1–6, June 2014.

[45] Saied, M., B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja. Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor. *IEEE International Conference on Robotics and Automation*, pages 5266–5271, 2015.

[46] Saied, M., B. Lussier, I. Fantoni, C. Francis, and H. Shraim. Fault tolerant control for multiple successive failures in an octorotor: Architecture and experiments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 40–45, Sep. 2015.

[47] Pose, C. D., A. Giusti, and J. I. Giribet. Actuator fault detection in a hexacopter using machine learning. In *2018 Argentine Conference on Automatic Control (AADECA)*, pages 1–6, Nov 2018.

[48] Dutta, A., M. McKay, F. Kopsaftopoulos, and M. Gandhi. Rotor fault detection and identification on a hexacopter based on statistical time series methods. In *Proceedings of the 75th Vertical Flight Society Annual Forum*, May 2019.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8

Chapter 4

CONCEPTS AND METHODS IN FAULT TOLERANT CONTROL WITH APPLICATION TO A WIND TURBINE SIMULATED SYSTEM

Silvio Simani[1,*] ***and Paolo Castaldi***[2]
[1]Department of Engineering, University of Ferrara, Ferrara, Italy
[2]Department of Electrical, Electronic, and Information Engineering
University of Bologna, Bologna, Italy

Abstract

Faults in automated processes will often cause undesired reactions and shutdown of a controlled plant, and the consequences could be damage to technical parts of the plant, to personnel or the environment. Fault tolerant control combines diagnosis with control methods to handle faults in an intelligent way. The aim is to prevent that simple faults develop into serious failure and hence increase plant availability and reduce the risk of safety hazards. Fault-tolerant control merges several disciplines into a common framework to achieve these goals. The desired features are obtained through online fault diagnosis, automatic condition assessment and calculation of appropriate remedial actions to avoid certain consequences of a fault. The envelope of the possible remedial actions is very wide. Sometimes, simple re–tuning can suffice. In other cases, accommodation of the fault could be achieved by replacing a measurement from a faulty sensor by an estimate. In yet other situations, complex reconfiguration or online controller redesign is required. This chapter gives an overview of well–established and more recent tools to analyse and explore structure and other fundamental properties of an automated system such that any inherent redundancy in the controlled process can be fully utilised to maintain availability, even though faults may occur. On the other hand, the effectiveness of the analysed solutions has been verified when applied to a wind turbine system. In fact, wind turbine plants are complex dynamic and uncertain processes driven by stochastic inputs and disturbances, as well as different loads represented by gyroscopic, centrifugal, and gravitational forces. Moreover, as their aerodynamic models are nonlinear, both modelling and control become challenging

[*]E-mail address: silvio.simani@unife.it (Corresponding author).

problems. On one hand, high–fidelity simulators should contain different parameters and variables in order to accurately describe the main dynamic system behaviour. Therefore, the development of fault tolerant control solutions for wind turbine systems should consider these complexity aspects. On the other hand, these solutions have to include the main wind turbine dynamic characteristics without becoming too complicated. The second point of this chapter is thus to provide practical examples of the development of robust fault tolerant control strategies when applied to a simulated wind turbine plant. Experiments with the wind turbine simulator represent the instruments for assessing the main aspects of the developed control methodologies.

Keywords: fault diagnosis, fault tolerant control, advanced control, sustainability, reliability and robustness, wind turbine

1. Introduction

Model–based and data–driven Fault Detection and Isolation (FDI) strategies have been studied for over 40 years, however they still represent an open research domain when considered to be applied to wind turbine systems, and many problems are waiting to be solved. The material presented in this monograph has inevitably had to end before all the interesting topics for future FDI research could be fully explored. In the following sections the authors describe some important topics that should help the reader to understand how to move from fault diagnosis to fault tolerance. Moreover, this chapter presents the fault tolerant control algorithms applied to wind turbine systems. In general, they are based on the signal correction principle, which means that the control system is not modified since the inputs and outputs of the baseline controller are compensated according to the estimated faults. Passive and active fault tolerant control systems are also discussed and compared, in order to highlight the achievable performances and the complexity of their design procedures. Controller reconfiguration mechanisms are also considered, which are able to guarantee the system stability and satisfactory performance.

As it will be described in the following, there are different approaches for eliminating or minimising disturbance and modelling error effects on residuals and hence for achieving robustness in fault diagnosis. However, these techniques were developed for ideal systems or with a special uncertainty structure, and then efforts have been made to include non–ideal or more general uncertainty. In contrast, frequency domain design methods are designed to possess robustness properties. In particular, H_∞ optimisation has been developed from the very beginning with the understanding that no design goal of a system can be perfectly achieved without being compromised by an optimisation in the presence of uncertainty, hence this technique is very suitable for tackling uncertainty issues.

It is worth noting that the work [1] first discussed the possibility of using frequency domain information to design fault diagnosis algorithms. The design of a residual generator in the frequency domain was firstly based on a frequency domain optimal observer and then by using the factorisation of the transfer function matrix of the monitored system. These methods were developed and later extended in [1]. Some important modifications in robust FDI design were made in [2] by using the factorisation–based H_∞ optimisation technique. The more elegant and advanced H_∞ optimisation methods are based mainly on the use of the Algebraic Riccati Equations (ARE). In particular, the robust fault estimation problem was solved by using Riccati equation approach through the use of H_∞ and μ robust estimator synthesis methods [1]. These approaches were further extended to time–variant and nonlinear systems.

The majority of studies considered in the related literature involved the use of a slightly modified H_∞ filter for residual generation. That is to say the design objective is to minimise the effect of disturbances and modelling errors on the estimation error and subsequently on the residual. The residual has to be remain sensitive to faults whilst the effect of disturbance has to be minimised. Hence, the essential idea is to reach an acceptable compromise between disturbance robustness and fault sensitivity. The final goal is to find an observer design which provides the maximum ratio between fault sensitivity and disturbance sensitivity. Solutions for this optimisation problem were given and revised, in order to obtain robust FDI technique [1]. Unfortunately, it was shown that this solution was quite conservative for certain frequency range. It should be pointed out that the disturbance transfer function matrix can only be defined for disturbances, hence the technique presented can only deal with robustness against disturbance. The robust problem with respect to modelling errors has still not been solved. The only solution suggested is to calculate the residual bound and set and adaptive threshold.

Few progresses were made solving the robust fault diagnosis problem against modelling errors when μ synthesis with H_∞ optimisation is incorporated. Robust FDI design based on H_∞ optimisation and μ synthesis is still in its early development, even if some research is still needed. This could be a direction for future research which has great potential. In connection with frequency domain, fault diagnosis techniques can exploit different data–driven approaches. For example, an identification method based on the frequency domain approach for Errors–In–Variable (EIV) models and its application to the dynamic Frisch scheme estimation technique was presented in [3]. Such a procedure can provide an accurate estimation of the transfer matrices from input–output measurements affected by white, mutually uncorrelated and correlated noises. This general method, using the frequency domain approach, facilitates a unique determination of both the characteristics of the noise affecting the data as well as the transfer matrices of the process under investigation. A comparison between time–domain and frequency-domain approaches can be found in [3].

It is clear that the system dynamics and parameters may vary or may be perturbed during the system operation. A fault diagnosis system designed for a system model corresponding to nominal system operation may not perform well when applied to the system with perturbed conditions. To overcome this problem, instead of using complex nonlinear models, a residual generator scheme using adaptive observers were proposed. The idea is to estimate and compensate system parameter variations. Figure 1 illustrates the basic principle of this approach. It can be applied to linear systems with parametric variations if stability and convergence conditions are satisfied.

Adaptive residual generation schemes for both linear and nonlinear uncertain dynamic systems using adaptive observers were proposed in the literature [1]. Unfortunately, the disadvantage of this approach is the complexity. In [1] it was presented an alternative way to generate adaptive symptoms using a method to estimate the bias term in the residuals due to modelling errors, then compensate it adaptively. This technique decreases the effects of uncertainties on residuals. The approach to estimate such a bias term in residuals rather than computing modelling errors themselves avoids complicated estimation algorithms.

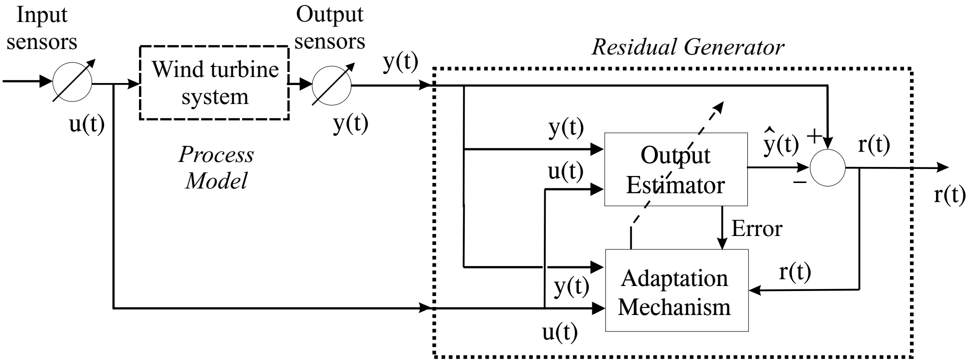


Figure 1. Residual generator with adaptive observer.

The state and the estimated parameters provided by the estimation mechanism provided by recursive estimation algorithms presented in [4, 5] can be also used to generate adaptive residuals. With reference to Figure 1, the output estimator exploited in the adaptive residual generation algorithm for fault diagnosis normally involves both the state and the process model parameter estimation, which can be thus considered as a combination of observer and identification based FDI approaches. Hence, complementary advantages in both approaches can be gained.

For all adaptive methods, the main problem to be tackled is that fault effects may be compensated as well as modelling errors and parameter variations. This makes the detection for incipient faults almost impossible whilst for abrupt faults this can be acceptable. To overcome this problem, the effect of faults can be considered as a slow varying parameter which can be estimated along with parameters. Under the assumption that parameters and faults varying at different rates, two filters with different gains can be used. However, much research effort is still needed in the theory and application of adaptive residual generation methods.

2. From Fault Diagnosis to Fault Tolerant Control

A conventional feedback control design for complex systems may result in unsatisfactory performance in the event of malfunction in input–output sensors, actuators and system components. A fault tolerant closed–loop control system is very attractive because it can tolerate faults whilst also maintaining desirable performance.

The conventional approach to the design of a fault-tolerant control includes different steps and separate modules: modelling or identification of the controlled system, design of the controller, FDI scheme and a method for reconfiguring the control system. Identification and design of the controller can be performed separately or using combined methods. Hence, the FDI and controller are linked through the reconfiguration module. The fundamental problem with such a system lies in the identification stage, in the independent design of the control and FDI modules. Significant interactions occurring among these modules can be neglected. There is therefore a need for a research study into the interactions between system identification, control design, the FDI stage and the fault–tolerant control design strategy.

Fault identification is the most important of all the fault diagnosis tasks. When a fault is estimated, detection and isolation can be easily achieved since the fault nature can improve the diagnosis process. However, the fault identification problem itself has not gained enough research attention.

Most fault diagnosis techniques, such as parameter identification, parity space and observer–based methods cannot be directly used to identify faults in sensors and actuators. Very little research has been done to overcome the fault identification problem. The Kalman filter for statistical testing and fault identification was proposed in [1]. However, the statistical testing methods can impose a high computational demand. A fault identification scheme solving a system inversion problem was proposed in [1, 6, 5].

In the scheme depicted in Figure 2 fault identification is performed by estimating the nonlinear relationship between residuals and fault magnitudes. This is possible because robust residuals should only contain fault information.

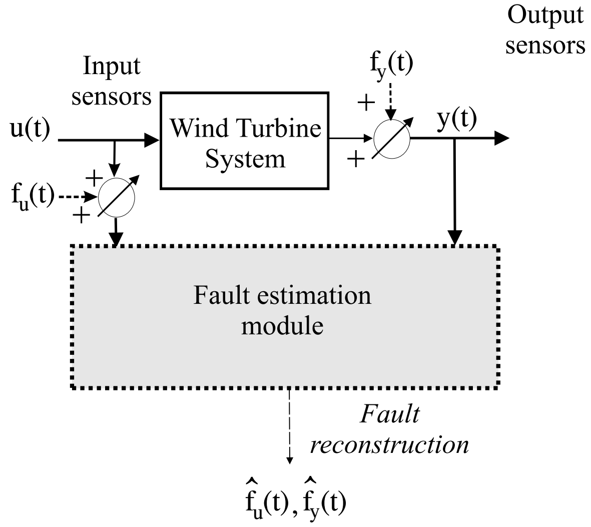


Figure 2. Fault estimation scheme for fault tolerant control.

Such a nonlinear function approximation and estimation can be performed by using fuzzy systems, neural networks or an inversion of the transfer matrix between residuals and faults [6, 5]. The central task in model–based fault detection is the residual generation. Most residual generation techniques are based on linear system models. For nonlinear systems, the traditional approach is to linearise the model around the system operating point. However, for systems with high nonlinearity and a wide dynamic operating range, the linearised approach fails to give satisfactory results.

One solution is to use a large number of linearised models corresponding to a range of operating points. This means that a large number of FDI schemes corresponding to each operating points is needed. Hence, it is important to study residual generation techniques which tackle nonlinear dynamic systems directly. There are some research studies on the residual generation of non–linear dynamic systems, for example using nonlinear observers [7, 1]. There have been some attempts to use nonlinear observers to solve nonlinear system FDI problem [7, 1], *e.g.* nonlinear unknown input observers, including adaptive observers and sliding mode observers. If the class of nonlinearities can be restricted, observers for bilinear systems were also proposed [1].

On the other hand, the analytical models, which the nonlinear observer approaches are based on, are not easy to obtain in practice. Sometimes, it is impossible to model the system using an explicit mathematical model. To overcome this problem, it is desirable to find a universal approximate model which can be used to represent the real system with an arbitrary degree of accuracy. Different approaches were proposed and they are currently under investigation: neural networks, fuzzy models and hybrid models.

As shown in [6, 5], fuzzy systems and neural networks are a powerful tool of handling nonlinear problems. One of the most important advantages of neural networks is their ability to implement nonlinear transformations for functional approximation problems. Therefore, neural networks can be used in a number of ways to tackle fault diagnosis problems for nonlinear dynamic systems. In early publications, they were mainly exploited as fault classifier with steady state processes, whereas, neural networks have been used as residual generators and for modelling nonlinear dynamic systems for FDI purposes [1].

Fuzzy models can be used both as residual classifier and as nonlinear system parametric model [1]. In the second case, the main idea is to build an FDI scheme based on fuzzy observers. Estimated outputs and residuals are computed as fuzzy fusion of local observer output and residuals. The main problem of this approach concerns the stability of the global observer. A linear matrix inequality method was proposed in [1] using Lyapunov theorem, but this solution can be quite conservative.

Hybrid models can describe the behaviour of any nonlinear dynamic process if they are described as a composition of several local affine models selected according to the process operating conditions [1, 6]. Instead of exploiting complicated nonlinear models obtained by modelling techniques, it is possible to describe the plant by a collection of affine models. Such a compound system requires the identification of the local models from data. Several works [1, 6] addressed a method for the identification and the optimal selection of the local affine models from a sequence of noisy measurements acquired from the process. Application of these results to model–based fault diagnosis for wind turbine systems is another research area worth of mention.

3. Wind Turbine Control Schemes

Wind turbines are complex dynamic systems forced by gravity, and stochastic wind disturbance, which are affected by gravitational, centrifugal, and gyroscopic loads. Their aerodynamics are nonlinear, whilst their rotors are subject to complicated turbulent wind inflow fields driving fatigue loading. Therefore, wind turbine modelling and control is a challenging task [8]. Accurate models have to contain many degrees of freedom in order to capture the most important dynamic effects. Moreover, the rotation of the turbine adds further complexity to the dynamics modelling. In general, off–the–shelf commercial software usually is not adequate for wind turbine dynamics modelling, but special dynamic simulation codes are required. It is clear that the design of control algorithms for wind turbines has to take into account these complexities. The main goal of the controller can consist of maintaining safe turbine operation, achieving prescribed control performances, and managing possible fault conditions, as shown *e.g.* in [9, 5].

Today's wind turbines employ different control actuation and strategies to achieve the required goals and performances. Some turbines perform the regulation action through passive control methods, such as in fixed–pitch, stall control machines. In these machines, the blades are designed so that the power is limited above rated wind speed through the blade stall. Thus, no pitch mechanism is needed [10]. In this case, the rotational speed control is proposed thus avoiding the inaccuracy of measuring the wind speed. Rotors with adjustable pitch are often used in constant–speed machines, in order to provide turbine power control better than the one achievable with blade stall [11]. In order to maximise the power output below the wind speed, the rotational speed of the turbine must vary with wind speed. Blade pitch control is used above rated wind speed in order to limit power [12]. Another control strategy for large commercial wind turbines can employ yaw regulation to orient the machine into the wind. A yaw error signal from a nacelle–mounted wind direction sensor is used to calculate a control error. In this situation the yaw motor is used when the yaw error exceeds a certain amount [13].

Other data–driven approaches can be based on schemes relying on the direct fuzzy identification of the controller model. As the wind turbine mathematical model is partially known and nonlinear, fuzzy identification represents an alternative for developing experimental models from input–output data. In contrast to pure nonlinear identification methods, fuzzy identification is capable of deriving nonlinear models without detailed system assumptions. Therefore, this approach derives the controller models using the data acquired from the plant under investigation. These fuzzy controllers are described by a collection of local affine systems of the type of Takagi–Sugeno (TS) fuzzy prototypes [14], whose parameters are obtained by identification procedures. In this way, the fuzzy controllers adjust both the wind turbine blade pitch angle and the generator torque of the wind turbine benchmark. Note that, with respect to [15], an off–line identification approach for fuzzy prototypes is exploited here, without the need of any further optimisation procedure, thus enhancing the real–time application. Moreover, in [5] it was presented a different solution to the design of the fuzzy regulators. In particular, the papers [5] presented the development of a fuzzy regulator, which is mainly based on the fuzzy identification of the wind turbine system; after this step, the fuzzy regulator is derived without any further identification procedure, but the PI fuzzy controller parameters are analytically computed from the identified process; the PI fuzzy controller parameters are thus derived using a suboptimal design procedure, i.e., using a fuzzy combination of the local PI controller parameters; the fuzzy membership functions of the PI fuzzy regulators are the same ones of the identified fuzzy model of the wind turbine system. On the other hand, in the controller design relies on the direct identification of the fuzzy controllers, whose parameters are directly identified, and not derived from the fuzzy models of the wind turbine process. Also the fuzzy membership functions of these controllers are directly identified, and they are different from the membership functions of the identified fuzzy models of the wind turbine system.

Other controller design methods can be also used that exploit adaptive schemes, which were not addressed in [5]. With reference to this adaptive control method, it considers the application of model on–line identification mechanisms in connection with model–based adaptive control design. This control method belongs to the field of adaptive control, see e.g., [16]. On–line parametric model identification schemes represent an alternative for developing experimental prototypes. Therefore, this approach exploits the implementation of controllers based on adaptive identification schemes, used for the on–line estimation of the controlled process. Recursive identification approaches extended to the adaptive case can make use of exponential forgetting algorithms, as proposed e.g., in [17]. In this way, a system identification scheme is exploited for the on–line estimate of the parameters of time–varying systems [16].

4. Sustainable Control

In general, wind turbines in the megawatt size are expensive, and hence their availability and reliability must be high in order to maximise the energy production. This issue could be particularly important for offshore installations, where Operation and Maintenance (O & M) services have to be minimised, since they represent one of the main factors of the energy cost. The capital cost, as well as the wind turbine foundation and installation determine the basic term in the cost of the produced energy, which constitute the energy 'fixed cost'. The O & M represent a 'variable cost' that can increase the energy cost up to about the 30%. At the same time, industrial systems have become more complex and expensive, with less tolerance for performance degradation, productivity decrease and safety hazards.

This leads also to an ever increasing requirement on reliability and safety of control systems subjected to process abnormalities and component faults. As a result, it is extremely important the Fault Detection and Diagnosis (FDD) or the FDI tasks, as well as the achievement of fault–tolerant features for minimising possible performance degradation and avoiding dangerous situations. With the advent of computerised control, communication networks and information techniques, it makes possible to develop novel real–time monitoring and fault–tolerant design techniques for industrial processes, but brings challenges. In the last years, many works have been proposed on wind turbine FDI/FDD, and the most relevant are in [9]. On the other hand, regarding the FTC problem for wind turbines, it was recently analysed with reference to an offshore wind turbine benchmark e.g., in [18].

In general, FTC methods are classified into two types, i.e., Passive Fault Tolerant Control (PFTC) scheme and Active Fault Tolerant Control (AFTC) scheme [19]. In PFTC, controllers are fixed and are designed to be robust against a class of presumed faults. In contrast to PFTC, AFTC reacts to the system component failures actively by reconfiguring control actions so that the stability and acceptable performance of the entire system can be maintained. In particular for wind turbines, FTC designs were considered and compared in [18]. These processes are nonlinear dynamic systems, whose aerodynamics are nonlinear and unsteady, whilst their rotors are subject to complicated turbulent wind inflow fields driving fatigue loading. Therefore, the so–called wind turbine 'sustainable' control represents a complex and challenging task [20].

Therefore, this chapter is motivated by the need of outlining the main solutions to sustainable control design, which are able of handling faults affecting the controlled wind turbine. For example, changing dynamics of the pitch system due a fault cannot be accommodated by signal correction. Therefore, it should be considered in the controller design, to guarantee stability and a satisfactory performance. Among the possible causes for changed dynamics of the pitch system, they can due to a change in the air content of the hydraulic system oil. This fault is considered since it is the most likely to occur, and since the reference controller may becomes unstable, as highlighted in [18]. Another issue raises when the generator speed measurement is unavailable, and the controller should rely on the measurement of the rotor speed, which is contaminated with much more noise than the generator speed measurement. This makes it necessary to reconfigure the controller to obtain a reasonable performance of the control system.

Section 4.1 outlines the main differences between active and passive fault–tolerant control systems and suggests how they are applied to the considered wind turbine system.

4.1. Fault Tolerant Control Architectures

In order to outline and compare the controllers developed using active and passive fault–tolerant design approaches, they should be derived using the same procedures in the fault–free case. In this way, any differences in their performance or design complexity would be caused only by the fault tolerance approach, rather than the underlying controller solutions. Furthermore, the controllers should manage the parameter–varying nature of the wind turbine along its nominal operating trajectory caused by the aerodynamic nonlinearities. Usually, in order to comply with these requirements, the controllers are usually designed for example using LPV modelling or fuzzy descriptions [21]. The two fault–tolerant control solutions have different structures as shown in Figures 3 and 4. Note that only the AFTC relies on a fault diagnosis algorithm. This represents the main difference between the two control schemes.

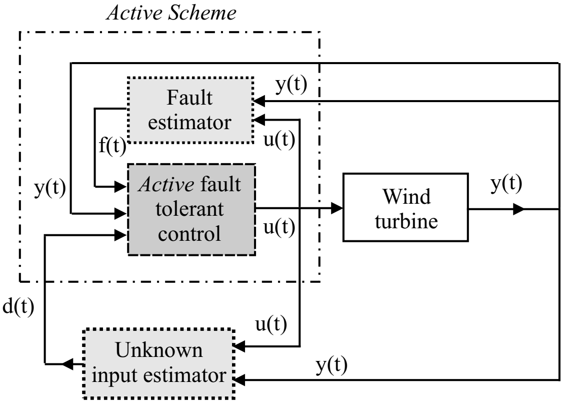


Figure 3. General scheme of the active fault tolerant control strategy.

The main point between AFTC and PFTC schemes is that an active fault–tolerant controller relies on a fault diagnosis system, which provides information about the faults f to the controller. In the considered case the fault diagnosis system FDD contains the estimation of the unknown input (fault) affecting the system under control. The knowledge of the fault f allows the AFTC to reconfigure the current state of the system.

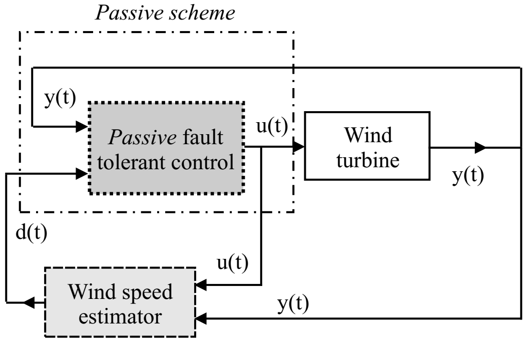


Figure 4. General layout of the passive fault tolerant control scheme.

On the other hand, the FDD is able to improve the controller performance in fault–free conditions, since it can compensate e.g., the modelling errors, uncertainty and disturbances. On the other hand, the PFTC scheme does not rely on a fault diagnosis algorithm, but is designed to be robust towards any possible faults. This is accomplished by designing a controller that is optimised for the fault–free situation, while satisfying some graceful degradation requirements in the faulty cases. However, with respect to the robust control design, the PFTC strategy provides reliable controllers that guarantee the same performance with no risk of false FDI or reconfigurations.

In general, the methods used in the fault–tolerant controller designs should rely on output feedback, since only part of the state vector is measured. Additionally, they should take the measurement noise into account. Moreover, the design methods should be suited for nonlinear systems or linear systems with varying parameters. The latest proposed solutions for the derivation of both active and the passive fault–tolerant controllers rely on LPV and fuzzy or neural network descriptions, to which the fault–tolerance properties are added, since these frameworks methods are able to provide stability and guaranteed performance with respect to parameter variations, uncertainty and disturbance. Additionally, LPV and fuzzy or neural controller design methods are well–established in multiple applications including wind turbines [21]. To add fault–tolerance to the common LPV and fuzzy or neural controller formulation, different approaches can be exploited. For example, the AFTC scheme can use the parameters of both the LPV and fuzzy structures estimated by the FDD module for scheduling the controllers [22].

On the other hand, different approaches can be used to obtain fault–tolerance in the PFTC methods. For this purpose, the design methods described in [23] can be modified to cope with parametric uncertainties, as addressed e.g., in [22]. Alternatively, other methods could have been used such as [21], which preserves the nominal performance. Generally, these approaches rely on solving some optimisation problems where a controller is calculated subjected to maximising the disturbance attenuation. These problems are formulated as LMI [24].

4.2. Controller Compensation and Active Fault Tolerance

The key point of the controller compensation and active fault tolerance issues regards the fault estimation task, which is also the most important of all the fault diagnosis phases. In fact, when a fault is estimated, both the detection and the isolation phases can be easily achieved, since the fault nature can improve the diagnosis process. However, the fault identification problem itself has not gained enough research attention.

Most fault diagnosis techniques, such as parameter identification, parity space and observer–based methods addressed in [6, 5] in general cannot be directly used to identify faults in sensors and actuators. Very little research has been done to overcome the fault identification problem. The Kalman filter for statistical testing and fault identification was proposed e.g., in [5]. However, the statistical testing methods can impose a high computational demand. A fault identification scheme solving a system inversion problem was proposed e.g., in [6, 5] applied to power systems.

In the scheme represented in Fig. 5, the fault identification task is performed by estimating the nonlinear relationship between residuals and fault magnitudes. This is possible because the generated robust residuals must contain only fault information, and do not depend on the system under diagnosis.

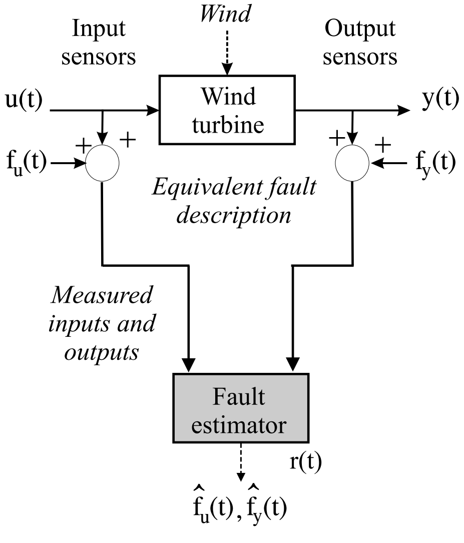


Figure 5. The fault estimation scheme for fault tolerant control.

The fault estimation module depicted in Figure 5 requires a nonlinear function approximation that can be performed by using data–driven or model–based approaches as highlighted in [1, 6].

Another important fault identification strategy can be achieved via a purely nonlinear scheme, which provides the fault detection, the isolation and the fault size estimation. As already remarked this FDD method is based on the NonLinear Geometric Approach (NLGA) principle developed in [25] and described in [6, 5]. By means of this methodology, disturbance decoupled adaptive nonlinear filters providing the fault reconstruction are developed.

It is worth observing that the original NLGA FDD scheme based on residual signals cannot provide fault size estimation. The achieved results in fault–free and faulty conditions that will be illustrated in Section 5 highlight the enhancement of the control requirements, the asymptotic fault accommodation, and the control objective recovery.

Finally, regarding the AFTC strategies for fault compensation that are considered in this chapter, the possible logic scheme of the integrated fault tolerant approach is represented in Figure 6.

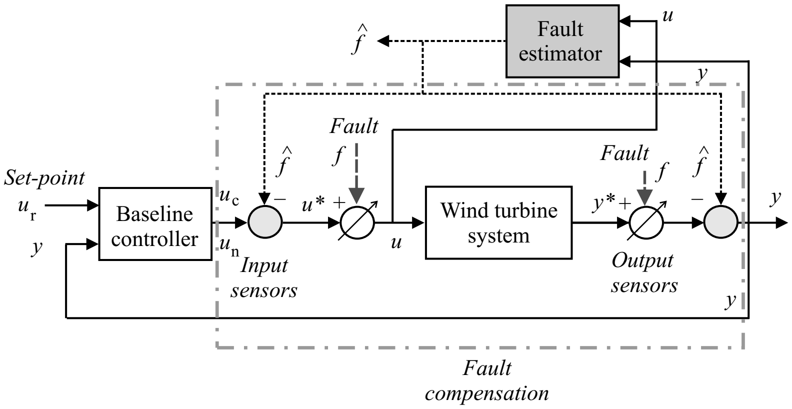


Figure 6. Active fault tolerant control strategy relying on the fault compensation mechanism.

With reference to Fig. 6, the following nomenclature and symbols are used. u_r represent the reference inputs (e.g., the reference set–point), whilst u are the actuated inputs. u^* are the unmeasurable inputs, u_c the controlled inputs, and u_n the feedback signals from the baseline controller. y are the controlled outputs (e.g., the wind turbine monitored outputs), and y^* the unmeasurable outputs. f are generic equivalent faults, and $\hat{f}$ the estimated faults.

Therefore, the logic scheme depicted in Figure 6 shows how the AFTC strategy has been implemented by integrating the fault diagnosis module (FDD) with the existing control system. From the controlled input and output signals, the FDD module provides the correct estimation $\hat{f}$ of the f actuator fault, which is injected to the control loop, for compensating the effect of the actuator fault. After this correction, the current controller provides the exact tracking of the reference signal u_r. Note that this signal can be generated by a further compensation block, as suggested in [6]. It can be shown that the feedback of the estimated fault $\hat{f}$ improves the identification of the fault signal f itself, by reducing also the estimation error and possible bias due to the model–system mismatch. Further results recalled in the following will highlight the achieved performance of this integrated FDD and AFTC strategy. However, the enhancement of the control requirements, the asymptotic fault accommodation, and the control objective recovery, that in this book are verified in simulation, can require further studies and investigations when applied to a wind turbine system, as remarked in Section 5.

Finally, these sections suggested the possible development of advanced fault tolerant control schemes. The methodologies were based on fault detection and diagnosis procedures relying on adaptive filters designed via model–based and data–driven approaches. The controller reconfiguration can exploit a further control loop, depending on the on–line estimate of the fault signal. One of the advantages of this strategy is that, for example, a structure of logic–based switching controller is not modified [18]. The adaptive fault tolerant control schemes are applied to different wind turbine systems in different working conditions, in the presence of faults, disturbance, measurement noise, and modelling errors.

4.3. Fault Tolerant Control for Fault Diagnosis

In the following the discrete–time monitored system, i.e., the wind turbine plant, is assumed to be affected by equivalent and additive faults on the input and output sensor measurements, which are able to properly describe actuator, system and sensor faults affecting the considered systems, as represented in Fig. 7, in forms of Eqs. (1):

$$\left\{\begin{array}{lcl}\mathbf{u}(k) & = & \mathbf{u}^*(k)+\mathbf{f}_u(k)\\ \mathbf{y}(k) & = & \mathbf{y}^*(k)+\mathbf{f}_y(k)\end{array}\right. \tag{1}$$

where $\mathbf{u}^*(k), \mathbf{y}^*(k)$ are the actual unmeasurable variables, $\mathbf{u}(k), \mathbf{y}(k)$ represent the sensor acquisitions, affected by both the measurement noise and the faults. As remarked in [6], $\mathbf{f}_u(k) and \mathbf{f}_y(k)$ are additive signals, that assume values different from zero only in presence of faults.

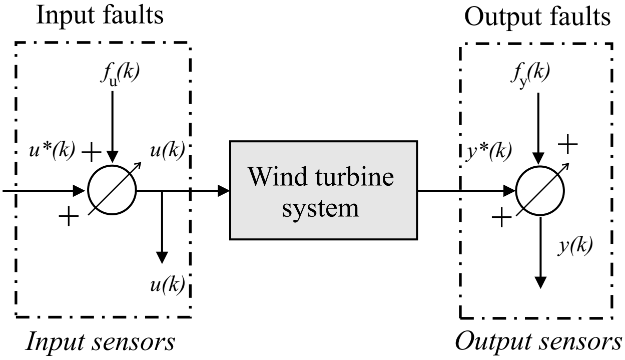


Figure 7. Equivalent additive input and output sensor faults affecting the wind turbine system.

Figure 7 shows the general scheme with the faults affecting the system under diagnosis, i.e., the wind turbine or the wind farm, as additive signals on the input (actuator) and output measurements.

Among the different approaches to generate the residual signals, available in [6, 5], the solutions recalled in this chapter can exploit data–driven or model–based prototypes, which are able provide an on–line estimation of the faulty signals. Hence, as shown in Fig. 8 the residual signals $\mathbf{r}$ are by means these filters by using the measured inputs $\mathbf{u}(k)$ and outputs $\mathbf{y}$ from the systems under diagnosis:

$$\mathbf{r}(k) = \hat{\mathbf{f}}(k) \tag{2}$$

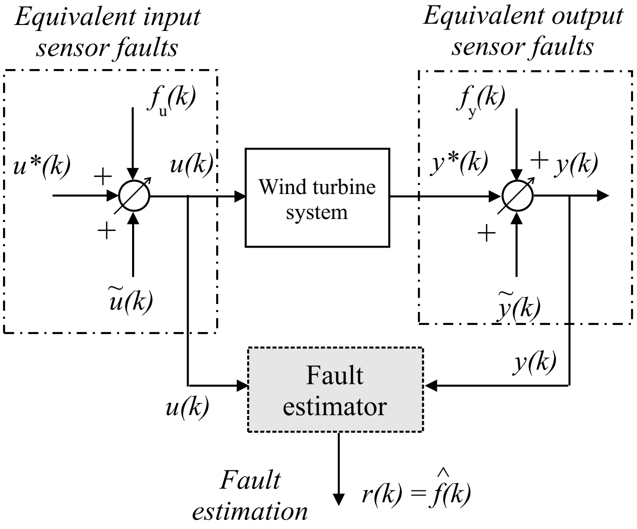


Figure 8. The general residual generation scheme for fault estimation.

which are the estimated faults $\hat{\mathbf{f}}(k)$ representing the equivalent faults $\mathbf{f}_u(k) and \mathbf{f}_y(k)$ of Eqs. (1).

Figure 8 highlights the residual generation scheme that is achieved by the proper processing of the acquired measurements, as described in [1, 6]. As already remarked, the fault diagnosis process involves, as first step, the fault detection task. It is performed here by using a proper thresholding logic operating on the residuals after their elaboration into a proper evaluation function:

$$\mathbf{r}_e(k) = F\left(\mathbf{r}(k)\right) \tag{3}$$

Where the proposed function $F(\cdot)$ can be the identity function, in case of the fault diagnosis solutions considered in this monograph, or suitable moving average or statistical operations on the residual signals, as explained in [1]. Then, the occurrence of the i–th fault can be detected according to a simple thresholding logic described in Eqs. (4):

$$\begin{cases} \bar{r}_{e_i} - \delta\sigma_{r_i} \leq r_{e_i} \leq \bar{r}_{e_i} + \delta\sigma_{r_i} & \text{in fault–free situations} \\ r_{e_i} < \bar{r}_{e_i} - \delta\sigma_{r_i} \text{ or } r_{e_i} > \bar{r}_{e_i} + \delta\sigma_{r_i} & \text{in faulty cases} \end{cases} \tag{4}$$

where the i-th item r_{e_i} of the residual vector $\mathbf{r}_e$ is considered a random variable, whose unknown mean $\bar{r}_{e_i}$ and variance $\sigma^2_{r_i}$ can be estimated in fault–free condition, after the acquisition of N samples, as described by the relations of Eqs. (5):

$$\begin{cases} \bar{r}_{e_i} & = & \frac{1}{N}\sum_{k=1}^{N} r_{e_i}(k) \\ \sigma^2_{r_i} & = & \frac{1}{N}\sum_{k=1}^{N}(r_{e_i}(k) - \bar{r}_{e_i})^2 \end{cases} \tag{5}$$

Note that he tolerance parameter $\delta \geq 2$ has to be properly tuned in order to separate the fault–free from the faulty condition. The δ value determines the trade–off between the false alarm rate and the fault detection probability. A common choice of δ relies on the three–sigma rule, otherwise extensive simulations can be performed to optimise the δ value.

Consequently to the fault detection, the fault isolation task is easily achieved by means of a bank of estimators. As described by Eqs. (1), the faults are considered as equivalent signals that affect the input measurements, i.e., $\mathbf{f}_u$, or the output measurements, i.e., $\mathbf{f}_y$.

Under this assumption, by following the scheme of the generalised estimator configuration of Fig. 9, in order to uniquely isolate one of the input or output faults, by considering that multiple faults cannot occur, a bank of Multi–Input Single–Output (MISO) fault estimators is used. In general, the number of this estimators is equal to the number of faults that have to be diagnosed, i.e., equal to the number of input and output measurements, $r + m$. Therefore, in general the i–th fault estimator that reconstructs the fault $\hat{f}(k) = r_i(k)$ is driven by the components of the input and output signals $\mathbf{u}(k)$ and $\mathbf{y}(k)$ that are sensitive to the specific fault $f_i(t)$. Therefore, it should be clear that the design of these fault estimators is enhanced by the so–called fault sensitivity analysis tool, described in [6, 5]. For each fault case, the failure modes and their resulting effects on the rest of the system are analysed, and in particular the most sensitive input and output measurements to that specific fault situation are identified. In this way, it is possible to derive the dynamic relationships between the input–output measurements and the faults, as represented by the estimator bank of Figure 9.

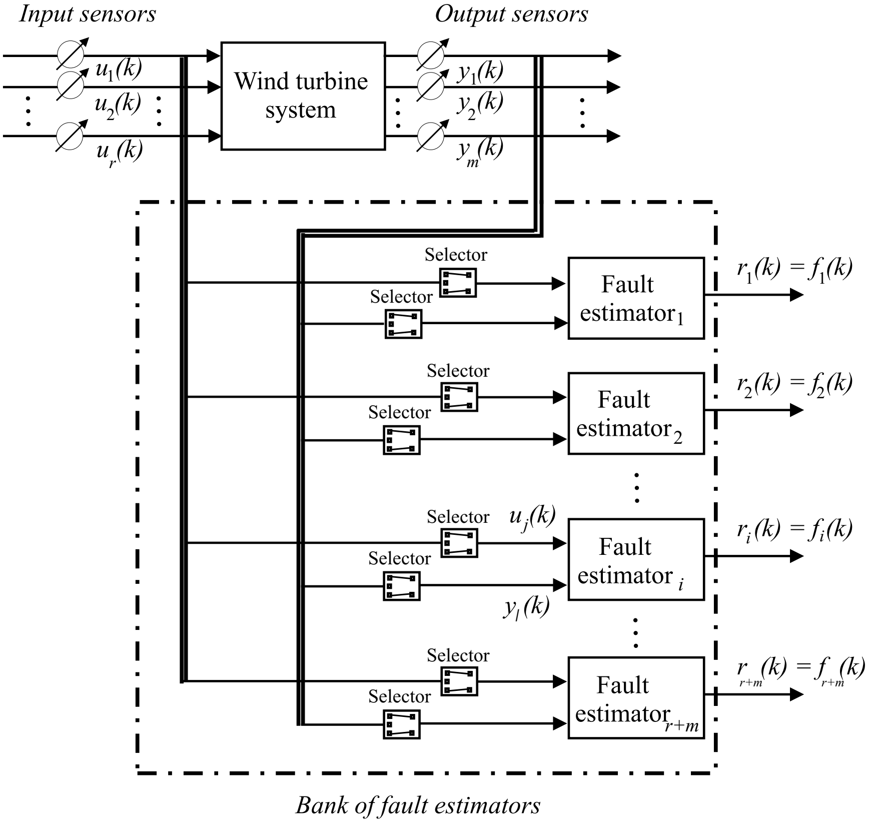


Figure 9. Bank of fault estimators for fault isolation.

Figure 9 shows this generalised fault estimator scheme, where the fault estimators are driven only by the input–output signals selected via this fault sensitivity tool and analysed for the wind turbine system considered in this chapter, so that the relative residual $r_i(k) = \hat{f}_i(k)$ is insensitive only to the fault affecting those inputs and outputs defined by the selector blocks. It is worth noting that multiple faults occurring at the same time cannot be correctly isolated, using this configuration.

The capabilities of the adopted fault diagnosis module can be summarised by means of the so–called *fault signature* matrix, depicted in Table 1, where each entry that is characterised by a value equal to '1' means that the considered residual (i.e., the equivalent fault) is sensitive to the actual fault effect ('0' otherwise), under the hypothesis above mentioned.

Table 1. Fault signatures for FDI

	u_1	u_2	$\ldots$	u_r	y_1	y_2	$\ldots$	y_r
r_1	0	1	$\ldots$	0	0	0	$\ldots$	0
r_2	1	0	$\ldots$	0	0	0	$\ldots$	0
$\vdots$			$\ddots$				$\ldots$	$\vdots$
r_i	0	0	$\ldots$	1	0	1	$\ldots$	0
$\vdots$			$\ldots$				$\ddots$	$\vdots$
r_{r+m}	0	0	$\ldots$	0	0	0	$\ldots$	1

As already remarked, the fault sensitivity tool, which has to be executed before the design of the fault estimators, suggests how to select the input–output configuration for the fault estimator blocks. Then, the design of the fault diagnosis block can be performed. Finally, the threshold test logic of Eq. (4) allows the achievement of the fault diagnosis tasks.

5. Results and Discussion

The structure of the FTC systems considered in this chapter, and applied to the wind turbine benchmark described in [5], is mainly based on fault diagnosis modules that provide the on–line fault estimations. The result of a proper fault identification allows for the compensation of the faulty measurement signals, before their access to the controller, so that the proper reference signal can be send to the turbine system, without the modification of the pre-existent controller. Fig. 10 shows the overall FTC strategy.

Figure 10 shows the general schemes of the fault tolerant control strategy proposed in this chapter: the fault diagnosis module provides the on–line estimation of the faults, which are used to compensate the faulty input–output signals. In this way, the controller can force the system to track the desired reference. On the other hand, the same fault diagnosis module can be used also for controller reconfiguration.

Therefore, the fault estimations $\hat{\mathbf{f}}$ in Fig. 10, i.e., the signals $\hat{\mathbf{f}}_u$, $\hat{\mathbf{f}}_y$ of Eqs. (1) are exploited for the compensation of both the input and the output measurements used by the system controller. In particular, the actuator signal coming from the controller is compensated by $\hat{\mathbf{f}}_u$, while $\hat{\mathbf{f}}_y$ corrects the output measurement acquired from the monitored system. After the fault compensation, the controller can track the nominal power reference signals.

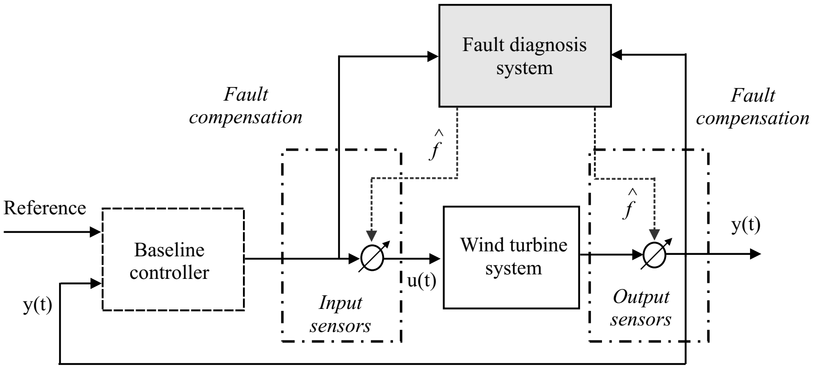


Figure 10. The overall FTC strategy applied to the wind turbine system.

It is worth noting that, thanks to this fault estimation feedback, the controller could be easily designed considering the fault-free system condition.

Further investigations regarding the stability analysis of the overall FTC module are addressed in [5], where it is shown that the variables of the models remain bounded in a set, which assure control performance, even in presence of faults. Moreover these faults do not modify the system structure, hence the global stability is guaranteed. However, whilst the fault effect is eliminated in steady–state condition, during the transient the compensation can be not properly handled, and the stability properties should be considered.

5.1. Simulation Results

In order to show the capabilities of the proposed fault tolerant control strategy, the system has been simulated as described in [5]. Extended simulations have been also performed by the authors according to a more realistic wind turbine test–rig [5]. The designed nonlinear filters provide the estimate the magnitude of the different faults acting on the the wind turbine model, as considered in [5].

As an example, the fault estimator provided the reconstruction $\hat{f}_u$, which is decoupled from the effect of both the wind speed and the model–reality mismatch. In order to compute the simulation results described below, the FTC scheme has been completed by means of the standard wind turbine controller proposed in [18]. The following results refer to the simulation of the accommodated controller with an input sensor fault case. Hence, after the derivation of the fault estimator, this filter provides an accurate estimate of the fault size, with minimal detection delay. The tests refer to the simulation of the input fault modelled as a sequence of rectangular pulses with variable amplitude and length. Fig. 11 (a) shows the estimate of the intermittent fault $\hat{f}_u$ (dotted grey line), compared with the actual fault (dashed black line). It is shown that the fault estimation module provides a quite good reconstruction of the fault signal. Under this condition, Fig. 11 (b)shows the reference signal compared with its desired value. The estimate feedback used by the AFTC scheme is applied at $t = 260s$. without any delay.

Fig. 11 (b) highlights the effectiveness of the presented integrated FDD and FTC strategy, which is able to improve the control objective recovery, and the reference tracking in the presence of control fault. However, the transient and the asymptotic stability of the

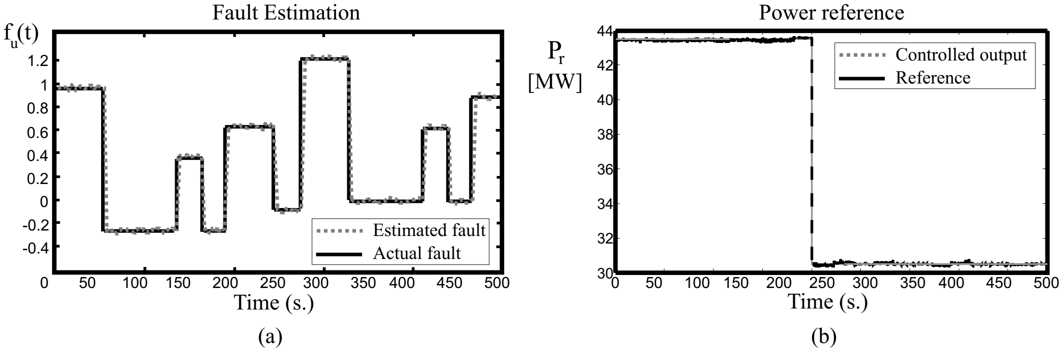


Figure 11. (a) Real–time estimate of the fault and (b) reference with the control signals.

controlled system, which in this paper are assessed in simulation, may require further investigations.

In order to summarise the advantages of the proposed strategy, the performance of the fault tolerant control applied to the wind turbine simulator with and without the fault compensation scheme has been evaluated in terms of per–cent Normalised Sum of Squared tracking Error ($NSSE$) values defined in (6):

$$NSSE\% = 100\sqrt{\frac{\sum_{k=1}^{N}\left(r(k)-y(k)\right)^2}{\sum_{k=1}^{N} r^2(k)}} \tag{6}$$

The simulation of different test data sequences has been performed by exploiting the offshore wind turbine simulator, followed by a Matlab Monte–Carlo analysis.

In particular, the nonlinear wind turbine simulator originally developed in the Simulink environment [18] was modified by the authors in order to vary the statistical properties of the signals used for modelling process parameter uncertainty, and measurement errors. Therefore, for performance evaluation of the control schemes, the best, average, and worst values of the $NSSE\%$ index were computed, and experimentally evaluated with 500 Monte–Carlo runs. Under these conditions, Table 2 summarises the results obtained by considering the proposed fault tolerant control integrating the original wind turbine controller for the different fault cases.

Table 2. Monte–Carlo analysis with faults

Fault case	Best case	Average case	Worst case
f_u	8.04%	11.23%	15.05%
f_y	9.01%	12.23%	14.74%

In particular, Table 2 summarises the values of the considered performance index according to the best, worst and average cases, with reference to possible uncertainty, disturbance and model–reality mismatch of the wind turbine. The results demonstrate also that

Monte–Carlo simulation is an effective tool for experimentally testing the design robustness of the proposed methods with respect to modelling uncertainty.

Finally, the remainder of this section compares the results of the considered fault tolerant control solutions with respect to the the passive approach developed in [26] and the adaptive scheme presented in [27]. On the other hand, the FTC strategy relying on the analytic disturbance decoupling approach addressed in [28] is also considered.

Table 3. Comparison of different FTC solutions in terms of $NSSE\%$ values

FTC Scheme	Fault f_u	Fault f_y
Data–driven FDD + FTC	14.94%	14.37%
NLGA FTC [28]	14.99%	14.86%
Passive FTC [26]	40.04%	39.93%
Adaptive FTC [27]	31.83%	30.94%

Regarding the FTC method proposed in this study, Table 3 illustrates that there are some deviations between the achieved results, but consistent with the ones from the Monte–Carlo analysis. Although there are some deviations between the simulation and the experimental results, these deviations are not critical and the results obtained are accurate enough for future wind turbine real applications. Moreover, the comparison of Table 3 highlights that this scheme seems to achieve better performances in terms of tracking error.

5.2. Discussion

Some concluding remarks can be finally be drawn here. This chapter provided some theoretical results for the diagnosis and the compensation of faults in the actuators, systems and sensors of wind turbine plants, through the use of different fault diagnosis and control accommodation schemes. These strategies were designed from the linear and nonlinear input–output descriptions of the system under diagnosis, and the disturbance decoupling was obtained. Procedures for optimising the fault sensitivity and dynamic response were also suggested.

An important aspect of the strategies based on linear residual generators is the simplicity of the technique used to generate these residuals when compared with different schemes. The algorithmic simplicity is a very important aspect when considering the need for verification and validation of demonstrable schemes for the viable application of these strategies to ream systems. The more complex the computations required to implement the scheme, the higher the cost and complexity in terms of verification and validation.

On the other hand, nonlinear methodologies can rely on a design scheme based on the structural decoupling of the disturbance obtained by means of proper coordinate transformations in the state space and in the output space. To apply the nonlinear theory, simplified models of the system under diagnosis can be required. The mixed $\mathcal{H}_-/\mathcal{H}_\infty$ optimisation of the trade–off between fault sensitivity, disturbances and modelling errors is now well understood in the theoretical work and is a promising area for application study to energy conversion systems.

The nonlinear fault diagnosis and fault compensation strategies have been based also on adaptive filters scheme. In addition to a proper detection and isolation, these methods provided also a fault size estimation. This feature is not usual for a fault detection and isolation method and can be very useful during on–line automatic control system reconfiguration, in order to recover a faulty operating condition. Compared with similar methods proposed in the literature, the nonlinear adaptive fault diagnosis and accommodation techniques described here have the advantage of being applicable to more general classes of nonlinear systems and less sensitive to measurement noise, since it does not use input/output signal derivatives.

Suitable filtering algorithms for stochastic systems have been also recalled. The knowledge regarding the noise process acting on the system under diagnosis can be exploited by the fault diagnosis and accommodation designs, hence the proposed schemes provided possible solutions to nonlinear system diagnosis with non–Gaussian noise and disturbance. The main advantage of nonlinear–based fault diagnosis and compensation techniques with disturbance decoupling features is represented by the fact that they take into account directly the model nonlinearity and the system reality–model mismatch.

The fault compensation techniques that have been outlined in this chapter have been applied to an high fidelity simulator of a wind turbine, which is able to take into account disturbances and measurement errors acting on the system under investigation. Moreover, the robustness characteristics and the achievable performances of the fault tolerant control approaches described have been carefully considered and investigated.

The effectiveness and the reliability of the proposed fault tolerant control schemes can be verified in simulation, whilst a more detailed comparison and discussion with widely used data–driven and model–based schemes with disturbance decoupling is available in [5]. The robustness properties of the designed fault estimators to model uncertainty, disturbances and measurements noise have been also analysed via extensive simulations, including the use of Monte–Carlo simulation experiments to tune the fault diagnosis and compensation module parameters.

Finally, the need to bridge the design gap between fault diagnosis and recovery mechanisms, i.e., the sustainable control schemes is obvious and properly analysed in this chapter. Moreover, fault diagnosis and fault tolerant control strategies can be properly designed and combined as shown in this chapter.

CONCLUSION

This chapter recalled the main fault diagnosis and the fault tolerant schemes that can be applied to wind turbine systems. Firstly, in the light of the design of a fault diagnosis module already proposed and oriented to the design of the control reconfiguration and accommodation system, the effective design of the fault estimators for fault tolerant control is addressed. Then, by exploiting the fault diagnosis scheme, interesting by–products consisting of the fault detection and isolation tasks are achieved. Finally, possible implementations of fault tolerant controller schemes are shown, which represent the key issue of the sustainable control design for safety–critical systems, such as the offshore wind turbine installations. In general, it was shown that the most effective fault tolerant control schemes rely on the on–line estimation and compensation of the system faults, modelled as equivalent input and output sensor faults, which are able to effectively describe any fault conditions affecting the considered wind turbine systems.

REFERENCES

[1] Chen, J. and Patton, R. J. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, (2012).

[2] Gertler, J. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, (1998).

[3] Van Huffel, S. and Lemmerling, P., editors. *Total Least Squares and Errors–in–Variables Modeling: Analysis, Algorithms and Applications*. Springer–Verlag, London, UK, 1st edition, February (2002). ISBN: 1402004761.

[4] Ljung, L. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, N.J., second edition, (1999).

[5] Simani, S. and Farsoni, S. *Fault Diagnosis and Sustainable Control of Wind Turbines: Robust data–driven and model–based strategies*. Mechanical Engineering. Butterworth–Heinemann – Elsevier, Oxford (UK), 1st edition, Jan. 4th (2018). ISBN: 9780128129845.

[6] Simani, S., Fantuzzi, C., and Patton, R. J. *Model-based fault diagnosis in dynamic systems using identification techniques*. Advances in Industrial Control. Springer–Verlag, London, UK, first edition, November (2003). ISBN: 1852336854.

[7] De Persis, C. and Isidori, A. *IEEE Transactions on Automatic Control* **45**(6), 853–865 June (2001).

[8] Johnson, K. E., Pao, L. Y., Balas, M. J., and Fingersh, L. J. *IEEE Control Systems Magazine* **26**(3), 70–81 (2006). DOI: 10.1109/MCS.2006.1636311.

[9] Hossain, M. L., Abu-Siada, A., and Muyeen, S. M. *Energies* **11**(5), 1–14 May (2018). DOI: 10.3390/en11051309.

[10] Honrubia-Escribano, A., Gomez-Lazaro, E., Fortmann, J., Sorensen, P., and Martin-Martinez, S. *Renewable and Sustainable Energy Reviews* **81**(2), 1939–1952 January (2018). DOI: 10.1016/j.rser.2017.06.005.

[11] Luo, N., Vidal, Y., and Acho, L., editors. *Wind turbine control and monitoring*. Advances in Industrial Control. Springer, Berlin, Germany, 2014 edition, Sept. (2014). ISBN: 9783319084121.

[12] Habibi, H., Nohooji, H. R., and Howard, I. *IEEE Access* **6**, 37464–37479 July (2018). DOI: 10.1109/ACCESS.2018.2853090.

[13] Odgaard, P. F. and Stoustrup, J. *IEEE Transactions on Control Systems Technology* **23**(3), 1221–1228 April (2015).

[14] Babuška, R. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, USA, (1998).

[15] Galdi, V., Piccolo, A., and Siano, P. *IEEE Transactions on Energy Conversion* **23**(2) June (2008).

[16] Bobál, V., Böhm, J., Fessl, J., and Machácek, J. *Digital Self–Tuning Controllers: Algorithms, Implementation and Applications*. Advanced Textbooks in Control and Signal Processing. Springer, 1st edition, (2005).

[17] Jens G. Linden, J. G., Larkowski, T., and Burnham, K. J. *International Journal of Control* **85**(11), 1625–1643 June (2012). DOI:10.1080/00207179.2012.696145.

[18] Odgaard, P. F., Stoustrup, J., and Kinnaert, M. *IEEE Transactions on Control Systems Technology* **21**(4), 1168–1182 July (2013). ISSN: 1063–6536. DOI: 10.1109/TCST.2013.2259235.

[19] Mahmoud, M., Jiang, J., and Zhang, Y. *Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis*. Lecture Notes in Control and Information Sciences. Springer–Verlag, Berlin, Germany, February (2003). ISBN: 3540003185.

[20] Odgaard, P. F. and Stoustrup, J. In *Proceedings of the IEEE Multiconference on Systems and Control – MSC2013*, 1–6 (, Hyderabad, India, 2013).

[21] Habibi, H., Koma, A. Y., and Howard, I. *Control Engineering and Applied Informatics* **19**(2), 31–42. DOI:.

[22] Harrabi, N., Kharrat, M., Aitouche, A., and Souissi, M. *International Journal of Applied Mathematics and Computer Science* **28**(2), 323–333 June 28 (2018).

[23] van Engelen, T., Schuurmans, J., Kanev, S., Dong, J., Verhaegen, M., and Hayashi, Y. In *Proceedings of EWEA 2011* (, Brussels, Belgium, 2011).

[24] Adegas, F. D. and Stoustrup, J. In *Proc. of the 18th IFAC World Congress*, 7933–7938 (IFACIFAC, Milan, Italy, 2011).

[25] De Persis, C. and Isidori, A. *IEEE Transactions on Automatic Control* **46**(6), 853–865 June (2001). DOI: 10.1109/9.928586.

[26] Simani, S. and Castaldi, P. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes – SAFEPROCESS 2012,* Astorga-Zaragoza, C. M. and Molina, A., editors, volume 8, 108–113 (Instituto de Ingeniería, Circuito escolar, Ciudad Universitaria, CP 04510, México D.F.IFAC, Mexico City, Mexico, 2012). Invited session paper. ISBN: 978–3–902823–09–0. ISSN: 1474–6670. DOI: 10.3182/20120829-3-MX-2028.00036.

[27] Simani, S. and Castaldi, P. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes – SAFEPROCESS 2012,* Astorga-Zaragoza, C. M. and Molina, A., editors, volume 8, 319–324 (Instituto de Ingeniería, Circuito escolar, Ciudad Universitaria, CP 04510, México D.F.IFAC, Mexico City, Mexico, 2012). Invited session paper. ISBN: 978–3–902823–09–0. ISSN: 1474–6670. DOI: 10.3182/20120829-3-MX-2028.00066.

[28] Simani, S. and Castaldi, P. *International Journal of Robust and Nonlinear Control* **24**(8–9), 1283–1303 May–June (2014). John Wiley. DOI: 10.1002/rnc.2993.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8
© 2020 Nova Science Publishers, Inc.

Chapter 5

RECONFIGURABLE FAULT TOLERANT CONTROL AGAINST SENSOR/ACTUATOR FAULTS APPLIED TO AUTONOMOUS UNDERWATER VEHICLE DYNAMICS

Chingiz Hajiyev*[*] *and Sitki Yenal Vural
Aeronautics and Astronautics Faculty,
Istanbul Technical University, Istanbul, Turkey

ABSTRACT

In this study, a reconfigurable fault-tolerant flight control system against sensor/actuator faults for autonomous underwater vehicles (AUVs) is proposed. First, an approach for detecting and isolating AUV sensor/actuator faults affecting the mean of the Kalman filter (KF) innovation sequence is proposed. Second, an augmented Kalman filter is used to isolate the sensor and actuator faults and estimate the control derivatives corresponding to the faulty actuator. In the case of a sensor fault, the robust Kalman filter algorithm with the filter gain correction is used. With the use of defined variables named as measurement noise scale factor, the faulty measurements are taken into consideration with a small weight and the estimations are corrected without affecting the characteristic of the accurate ones. In case of an actuator fault, fault isolation and identification are performed using the augmented KF. The control reconfiguration procedure is executed by utilizing the identified control distribution matrix. The parameters of the feedback controller are tuned by the control reconfiguration procedure. In the simulations, the steering subsystem dynamics of the AUV model is considered, and the sensor/actuator fault detection and isolation are examined. Some simulation results for the reconfigurable active fault tolerant control against actuator faults are given.

[*] Corresponding Author's Email: cingiz@itu.edu.tr.

Keywords: autonomous underwater vehicle, system identification, Kalman filter, fault tolerant control, LQR controller, steering subsystem, control reconfiguration

1. Introduction

The fault-tolerant control is an important problem in aerospace and naval applications and attracts interest of many researchers [1-5]. Their proposed methods fall into two categories: passive and active. In the passive category, the impaired vehicle continues to operate with the same controller; the effectiveness of the scheme depends on the original control law's degree of robustness. The active category involves either an on-line re-design of the control law after failure has occurred and has been detected, or the selection of a new pre-computed control law. Each version may use different actuators, or may use the same actuators in a manner different from the one used before the failure [1].

In this chapter, the active methods are considered. The active fault-tolerant control systems consist of two basic subsystems [2]:

1. Fault detection and isolation (FDI) or system identification, and
2. Control reconfiguration or restructure.

In an active fault-tolerant control system, the faults are detected and identified by a FDI scheme, and the controllers are reconfigured accordingly on-line in real-time. An effective FDI procedure is critical for designing high performance active fault tolerant control systems. Many model-based FDI techniques have been developed to detect and identify the sensor and actuator faults by using the analytical redundancy, state estimation and parameter identification approaches [3, 6, 7, 8].

AUVs require a precise navigation system for localization, positioning, path tracking, guidance, and control during a long period of duty cycle. In order to develop an accurate and robust navigation and control system for an AUV, it is needed to derive the fault tolerant filtration algorithms for estimation of AUV dynamics. Since its inception, Kalman filter has been widely used as the AUV motion dynamics parameters estimation technique [9] and different Kalman filter (KF) types have been developed for that purpose. By using KF, it is possible to estimate motion dynamics parameters of an AUV, which has a typical navigation sensor outfit such as compass, pressure depth sensor, and some class of inertial navigation system (INS) [10].

In the normal operating conditions of an AUV, conventional Kalman filter gives sufficiently good estimation results. However, if the measurements are not reliable because of any kind of malfunction in the estimation system, KF gives inaccurate results and diverges from correct values by time. The conventional KF has no capability to adapt itself to the changing conditions of the measurement system. Malfunctions such as abnormal measurements, changes in the background noise and the like affect instantaneous filter outputs and the process may result in the failure of the filter. In order to avoid from such condition, the filter must be operated robustly.

One of the methods for constructing the robust Kalman filter (RKF) algorithm is to use a single adaptive factor as a multiplier to the process or measurement noise covariance matrices [11,12]. This algorithm, which may be named as adaptive fading Kalman filter (AFKF), can be both used when the information about the dynamic process or the priori measurements is absent. However, when the point at issue is the recent measurements, another technique to scale measurement noise covariance matrix and make filter robust (insensitive to recent measurement faults) should be proposed. Therefore, if there is a malfunction in the measurement system, RKF algorithm can be utilized and insensitiveness of the filter to the current measurement faults can be satisfied by the use of a measurement noise scale factor (MNSF) as a multiplier on the measurement noise covariance matrix. As a consequence, via a correction applied to the filter gain, good estimation behaviour of the filter will be secured without being affected from faulty current measurements [13].

In this chapter, RKF algorithm with single measurement noise scale factor is introduced and applied for the motion dynamics parameters estimation process of an AUV. The proposed RKF for measurement noise scaling are considerably simpler than the existing and may be preferred, especially for the AUV motion dynamics estimation.

Under the assumption that system faults can be detected, isolated and identified via fault detection and isolation (FDI) techniques, several reconfigurable control methods have been developed in the literature. Existing reconfigurable controller design methods are based on one of the following approaches: linear quadratic regulator (LQR) [14, 15]; adaptive control [16]; pseudo-inverse [3, 17, 18]; multiple model [19, 20]; eigenstructure assignment [21, 22, 23, 24]; model predictive control [25] and neural networks [26].

In Refs. [14, 22], for estimation of the actuator fault parameter and the system state variables, the adaptive two-stage Kalman filter (TSKF) is employed. The proposed methods are based on the estimation of effectiveness factor of the faulty actuator. The actuators are 100% effective (in executing the control commands), if they operate exactly as the controller directs them during normal operation. When faults occur in the actuators, such as partial loss of a control surface, or pressure reduction in the hydraulic lines (in case of an aircraft; partial blockage of a control valve in process control, or voltage reduction/amplifier saturations in electrical servo systems) the actuators would not be able to fulfill the control commands completely. In such cases, it is said that the effectiveness of the actuators decreases [22].

In Refs. [14, 22, 27], a parameter known as the control effectiveness reduction factor is used to quantify the faults that enter to the control systems through actuators and it represents the loss of the direct relationship between the control command and the true actuator actions. In these studies, the control effectiveness factor is employed as the actuator fault parameter and estimated via the TSKF but in this, the control effectiveness factor of the faulty actuator is assumed to be the same for all the elements of the corresponding control distribution vector (or the appropriate column of the control distribution matrix).

A potential problem arises when the actuator is fault free but the corresponding control surface is damaged. The associated loss of effectiveness of the actuation/control surface system can not be detected in this way [28]. In practice, it can meet certain surface faults, for instance partial loss of a control surface (when a part of the control surface breaks off), deformation of the control surface, control surface icing etc., and these correspond to different control effectiveness factors for the actuator.

In this study, a reconfigurable fault-tolerant flight control system against sensor/actuator faults for AUV is proposed. The proposed method is based on two types of KF: a) a conventional linear KF, which estimates the states of a AUV, and b) an augmented KF, which estimates the AUV states and the control distribution matrix elements that correspond to the faulty actuator. An approach to detecting and isolating AUV sensor/actuator faults affecting the mean of the Kalman filter innovation sequence is proposed. The actuator faults were isolated and identified using the augmented KF. In the proposed approach, if the sensor fault is detected, then the RKF, which is robust to the sensor faults, is used. In the case of an actuator fault, the reconfigurable fault tolerant control against actuator failures is performed. Under the faulty conditions, a control reconfiguration action is taken in order to keep the performance of the impaired AUV the same as that of the unimpaired AUV.

2. Mathematical Model of AUV Steering Dynamics

AUV modeling is fairly complicated, and an exact analysis is only possible by including the underlying infinite dimensional dynamics of the surrounding fluid (sea water). While this can be done using partial differential equations in Computational Fluid Dynamics (CFD) computer tools, it still involves a formidable computational burden, infeasible for most practical applications.

AUVs move in 6 degrees of freedom (6DOF) since six independent coordinates are necessary to determine the position and orientation of a rigid body (See Figure 1). The first three coordinates and their time derivatives are of translational motion along the x, y and z-axes, while the last three coordinates (ϕ,θ,ψ) and time derivatives are used to describe orientation and rotational motion.

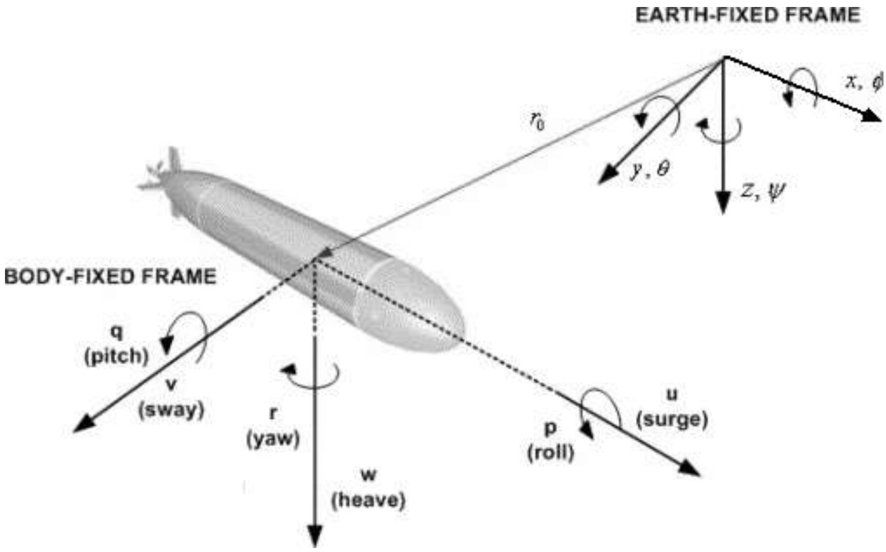


Figure 1. 6-DOF AUV angular and translational motions.

The linearized model of torpedo will be used instead of sample AUV in calculations. 6 different motion variables help to determine the position and orientation. First three coordinates (x, y, z) are used to determine the position. Time derivatives of three coordinates (u, v, w) define transitions along x, y and z. Euler angles show the orientation. Time derivatives of Euler angles (p, q, r) express the rotational motion.

In the following sections, we will describe dynamic model for the steering subsystem of AUVs, and design the reconfigurable active controller against actuator faults.

2.1. Steering Subsystem of Sample AUV

Steering subsystem equations are shown below [29];

$$\begin{bmatrix} m - Y_{\dot{V}_r} & -Y_r & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v}_r \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_{\dot{V}_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t), \tag{1}$$

$$M = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_r & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where m is the mass of the body, v_r is the sway velocity, r is the yaw rate, ψ is the heading angle, δ_r is the rudder deflection. The authors recommend [30,31] for details about the rest of the parameters above and their derivation.

If the inverse of M matrix in (2) is calculated and both sides are multiplied with M^{-1} in (1), equation transforms to;

$$\begin{bmatrix} \dot{v}_r \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_{\dot{V}_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t), \tag{3}$$

2.2. Discretization of Steering Subsystem

The transition matrix of the system A_S and control distribution matrix B_S in equation (3) are defined as below:

$$A_S = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_{\dot{V}_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$B_S = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix}, \tag{5}$$

If A_S and B_S matrices are defined as (4) and (5), A_S^* and B_S^* matrices are also defined for discretization as below;

$$A_S^* = I + \Delta t \times A_S\,;\; B_S^* = \Delta t \times B_S \tag{6}$$

Let us define the state vector as $X_S = \begin{bmatrix} v_r & r & \psi \end{bmatrix}^T$. Then the mathematical model of the steering subsystem can be written in the discrete form as:

$$X_S(k+1) = A_S^* \times X_S(k) + B_S^* \times U_S(k) \tag{7}$$

Here, $U_S(k)$ is control input by rudders. Discretized model (7) will be used for Kalman applications.

3. KF for Estimation and Identification of AUV Dynamics

Three types of Kalman filter are developed in this Section: optimum linear KF for estimation of AUV dynamics, robust KF for estimation of AUV dynamics in the presence of sensor faults and augmented KF for simultaneous estimation of AUV dynamics and identification of the control distribution matrix elements that correspond to the faulty actuator.

3.1. Optimum Linear KF for Estimation of AUV Dynamics

Consider the following linear discrete dynamic system:

$$X_S(k+1) = A_S^* \times X_S(k) + B_S^* \times U_S(k) + G \times W(k) \tag{8}$$

$$z(k) = H\left(k\right) X_S(k) + v\left(k\right), \tag{9}$$

where $X_S(k)$ is the 3-dimensional state vector of the system at time t_k, A_S^* is the 3×3 transition matrix of the system, B_S^* is the 3×1 control distribution matrix, $U_S(k) = \delta_r(k)$ is the one-dimensional control input, $W(k)$ is the random 3-dimensional Gaussian noise vector (system noise) with zero mean and known covariance structure, G is the 3×3 transition matrix of the system noise, $z(k)$ is the 3-dimensional measurement vector at time t_k, $H(k)$ is the 3×3 measurement matrix of the system, and $v(k)$ is the 3-dimensional measurement noise vector with zero mean and known covariance structure. The random vectors $W(k)$ and $v(k)$ are both assumed to represent Gaussian white noise. Their mean values and covariances are:

$$\begin{aligned}
&E\left[W(k)\right] = 0; E\left[W(k)W^T(j)\right] = Q(k)\delta(kj) \\
&E\left[v(k)\right] = 0; E\left[v(k)v^T(j)\right] = R(k)\delta(kj) \\
&E\left[W(k)v^T(j)\right] = 0
\end{aligned}$$

where E is the statistical averaging operator, and $\delta(kj)$ is the Kronecker delta symbol.

Apparently, the optimum Kalman filter (OKF) [32], that estimates the state vector of the system (8)-(9) is expressed with the following recursive equations system:

Equation of the estimation value is,

$$\hat{X}_S\left(k/k\right) = \hat{X}_S(k/k\text{-}1) + K(k)\left[z(k)\text{-}H(k)\hat{X}_S(k/k\text{-}1)\right], \tag{10}$$

where:

$$\hat{X}_S(k/k-1) = A_S^* \times \hat{X}_S(k-1/k-1) + B_S^* \times U_S(k-1) \tag{11}$$

is the extrapolation value and $K(k)$ is the gain matrix of the optimum linear Kalman filter:

$$K(k) = P(k/k-1)H^T(k)\left[H(k)P(k/k-1)H^T(k) + R(k)\right]^{-1} \tag{12}$$

where $R(k)$ is the covariance matrix of measurement noise.

The covariance matrix of the filtering error is,

$$P(k/k) = \left[I - K(k)H(k)\right]P(k/k-1), \tag{13}$$

where I is the identity matrix.

The covariance matrix of the extrapolation error is,

$$P(k/k-1) = A_S^* P(k-1/k-1)A_S^{*T} + GQ(k-1)G^T, \tag{14}$$

where $Q(k-1)$ is the covariance matrix of system noise.

The innovation sequence $\Delta(k)$ and innovation covariance $P_\Delta(k)$ of the OKF respectively are

$$\Delta(k) = z(k)\text{-}H(k)\hat{X}_S(k/k-1) \tag{15}$$

$$P_\Delta(k) = H(k)P(k/k-1)H^T(k) + S(k)R(k) \tag{16}$$

The normalized innovation can be written in the form

$$\tilde{\Delta}(k) = \left[H(k)P(k/k-1)H^T(k) + R(k) \right]^{-1/2} \Delta(k) \tag{17}$$

3.2. Robust Kalman Filter with the Filter Gain Correction

Kalman filter is definitely sensitive to measurement noise (abnormal measurements, instantaneous shifts in measurement channel and decrease in device accuracy, background noise etc.). If the state of the process of measurement system does not correspond to mathematical model used in filter, the changes caused by normal malfunctions in measurement channel, decrease the accuracy of estimation significiantly. In this case, RKF can be used to prevent noise [33].

The state space model of system is explained by (8)-(9). In case of normal operation of the measurement system, the filter works according to the conventional algorithms. But if the condition of operation of the measurement system does not correspond to the models used in the synthesis of filter, then the gain matrix of Kalman filter automatically changes due to a change in the covariance matrix of the innovation sequence according to the following rule [13];

$$P_{\Delta}(k) = H(k)P(k/k-1)H^T(k) + S(k)R(k) \tag{18}$$

in which adaptive factor (measurement noise scale factor) $S(k)$ is calculated from the innovation sequence (15) analysis results. The filter gain matrix in this case can be written in the following form:

$$K(k) = P(k/k-1)H^T(k)\left[H(k)P(k/k-1)H^T(k) + S(k)R(k) \right]^{-1} \tag{19}$$

According to the proposed approach the gain matrix is changed when the following condition is valid

$$tr\left\{ \Delta(k)\Delta^T(k) \right\} \geq tr\left\{ E\left[\Delta(k)\Delta^T(k) \right] \right\} \tag{20}$$

where *tr(.)* is the trace of matrix. The right side of the expression (20) can be written in the following form:

$$\begin{aligned} & tr\left\{ E\left[\Delta(k)\Delta^T(k) \right] \right\} = tr\{E\{\left[H(k)\left(X_S(k) - \hat{X}_S(k/k-1) \right) + v(k) \right] \\ & \times\left[H(k)\left(X_S(k) - \hat{X}_S(k/k-1) \right) + v(k) \right]^T \}\} = \\ & tr\left\{ H(k)P(k/k-1)H^T(k) + R(k) \right\} = tr\left[P_{\Delta}(k) \right] \end{aligned} \tag{21}$$

Taking (21) into account, the inequality (20) can be expressed as

$$tr\left\{\Delta(k)\Delta^{T}(k)\right\} \geq tr\left[P_{\Delta}(k)\right] \tag{22}$$

When a significant change in the operation conditions of the measurement system occurs, the prediction of observations $H(k)\hat{X}_{S}(k/k-1)$ will considerably differ from the observation results $z(k)$. Consequently, the sum of the discrepancy squares on the left side of (22) will characterize the real filtration error, while the right side determines the theoretical accuracy of the innovation sequence, obtained on the basis of a priori information. If condition (22) is met, then the real filtration error exceeds the theoretical error. Therefore, it is necessary to correct the filter gain matrix beginning from this moment. In order to calculate the measurement noise scale factor $S(k)$, the equality given by

$$tr\left\{\Delta(k)\Delta^{T}(k)\right\} = tr\left[P_{\Delta}(k)\right] \tag{23}$$

can be used. In this case, by plugging (18) into (23) the following expression is obtained;

$$tr\left\{\Delta(k)\Delta^{T}(k)\right\} = tr\left\{H(k)P(k/k-1)H^{T}(k)\right\} + S(k)tr\left\{R(k)\right\} \tag{24}$$

Hence taking the expression $tr\left\{\Delta(k)\Delta^{T}(k)\right\} = \Delta^{T}(k)\Delta(k)$ into consideration, the following formula for the adaptive factor $S(k)$ is obtained:

$$S(k) = \frac{\Delta(k)\Delta^{T}(k) - tr\left\{H(k)P(k/k-1)H^{T}(k)\right\}}{tr\left\{R(k)\right\}} \tag{25}$$

Using (18), (19) and (25) in the optimal estimation algorithm gives the possibility to accomplish an adaptation of filter to the change of measurement system operation conditions. If the left side of the expression (22) is greater than the right side, the adaptive factor value $S(k)$ will increase. This corresponds to the beginning of adaptation of filter. Consequently the covariance matrix of innovation sequence $P_{\Delta}(k)$ (18) increases, and the filter gain matrix $K(k)$ (19) decreases, which will cause strengthening of the corrective influence of innovation sequence in the estimation algorithm and decrease the difference between the estimation value $\hat{X}_{S}\left(k/k\right)$ and the actual value $X_{S}(k)$. This will lead to the decrease of innovation sequence $\Delta(k)$ and measurement noise scale factor $S(k)$, weakening of the corrective influence of innovation sequence, etc.

In contrast to the standard optimal filtration algorithm, in which the filter gain $K(k)$ is changed by program, current measurements in the proposed algorithm have larger weight, since the coefficients of matrix $K(k)$ are corrected by the results of each observation. This algorithm is adapted to the measurement system operation conditions by the approximation of theoretical covariance matrix $P_\Delta(k)$ to the real covariance matrix of innovation sequence, by applying the changing measurement noise scale factor $S(k)$. The mentioned change can be accomplished using the matrix $\Delta(k)\Delta^T(k)$, which characterizes the real filtration error. The presented RKF will ensure the guaranteed adaptation of the filter to the change of the measurement system operation conditions.

3.3. KF for Estimation and Identification of AUV Dynamics

The linear control system for the steering subsystem of AUV can be given by the discretized state model as in (8)-(9). $b_i, i = \overline{1,3}$ are the values in the control distribution matrix B_S^* that is defined to determine the actuator faults in the system. For this purpose the extended state vector is redefined as below;

$$U = \begin{bmatrix} x_1 & x_2 & x_3 & b_1 & b_2 & b_3 \end{bmatrix}^T \tag{26}$$

The augmented dynamic system can now be written as:

$$U(k+1) = \tilde{F}U(k) + \tilde{\Gamma}\tilde{W}(k) \tag{27}$$

and the measurement equation becomes

$$\tilde{z}(k) = \tilde{H}(k)U(k) + \tilde{v}(k) \tag{28}$$

where $U(k)$ is the 6-dimensional augmented system state vector, $\tilde{F}$ is the 6×6 augmented system matrix, $\tilde{\Gamma}$ is the 6×6 augmented system noise transition matrix, $\tilde{W}(k)$ is the random 6-dimensional vector, $\tilde{z}(k)$ is the 3-dimensional measurement vector, $\tilde{H}(k)$ is the 3×6 system measurement matrix, and $\tilde{v}(k)$ is the 3-dimensional measurement noise vector.

The augmented system matrix $\tilde{F}$ can be presented in the following form:

$$\tilde{F} = \begin{bmatrix} A^{*}_{S(3\times3)} & U_S \times I_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{29}$$

The augmented KF for the system given in (27)-(28) is;

Estimation equation:

$$\begin{gathered} \hat{U}(k+1/k+1) = \hat{U}(k+1/k) + \\ K(k+1)\left[\tilde{z}(k+1) - \tilde{H}(k+1)\hat{U}(k+1/k)\right] \end{gathered} \tag{30}$$

Extrapolation equation:

$$\hat{U}(k+1/k) = \tilde{F}\hat{U}(k/k) \tag{31}$$

Filter-gain matrix:

$$\begin{gathered} K(k+1) = P(k+1/k)\tilde{H}(k+1)^T \times \\ \left[\tilde{H}(k+1)P(k+1/k)\tilde{H}(k+1)^T + R(k+1)\right]^{-1} \end{gathered} \tag{32}$$

Covariance matrix of extrapolation error:

$$P(k+1/k) = \tilde{F}P(k/k)\tilde{F}^T + \tilde{\Gamma}\tilde{Q}(k)\tilde{\Gamma}^T \tag{33}$$

Covariance matrix of estimation error:

$$P(k+1/k+1) = \left[I - K(k+1)\tilde{H}(k+1)\right]P(k+1/k) \tag{34}$$

where $\tilde{Q}(k)$ is the covariance matrix of the random noise $\tilde{W}(k)$.

The innovation $\Delta(k+1)$ and normalized innovation $\tilde{\Delta}(k+1)$ of the augmented KF respectively are

$$\Delta(k+1) = \tilde{z}(k+1) - \tilde{H}(k+1)\hat{U}(k+1/k) \tag{35}$$

$$\tilde{\Delta}(k+1) = \left[\tilde{H}(k+1)P(k+1/k)\tilde{H}(k+1)^T + R(k+1)\right]^{-1/2}\Delta(k+1) \tag{36}$$

4. SENSOR/ACTUATOR FAULT DETECTION AND ISOLATION

Detection and isolation of both sensor and actuator faults is considered for the AUV steering subsystem mathematical model. The sensor fault is represented by the difference between real and estimated values of measured output [34]. The proposed fault detection and isolation method works with the assumption that only one sensor or one actuator is faulty at a time, which is a reasonable assumption in practice. To detect sensor/actuator failures affecting the innovation sequence, a squared normalized innovation of the optimum linear KF can be used [3, 34],

$$\beta(k+1)=\tilde{\Delta}(k+1)^T\,\tilde{\Delta}(k+1) \tag{37}$$

where $\tilde{\Delta}(k+1)$ is the normalized innovation sequence of optimum Kalman filter. This statistical function has χ^2 distribution with s degree of freedom, where s is the dimension of the innovation vector. Now consider the following two hypotheses:

H_0 : Sensor fault occurs
H_1 : No sensor fault.

If the level of significance, α , is selected as,

$$P\left\{\chi^2 > \chi^2_{\alpha,s}\right\}=\alpha;\ \ 0<\alpha<1 \tag{38}$$

the threshold value, $\chi^2_{\alpha,s}$ can be found. Hence, when the hypothesis H_1 is true, the statistical value of the function $\beta(k+1)$ will be greater than the threshold value $\chi^2_{\alpha,s}$ i.e.,

$$H_0\ :\ \beta(k+1)\le\chi^2_{\alpha,s}\ \ \forall k \tag{39}$$

$$H_1\ :\ \beta(k+1)>\ \chi^2_{\alpha,s}\ \ \exists k\,. \tag{40}$$

To determine if the fault is a sensor fault or an actuator fault,a squared residual $SR(k+1)$ of the augmented KF is introduced

$$SR(k+1)=\left[\tilde{z}(k+1)-\tilde{H}(k+1)\hat{U}(k+1/k+1)\right]^{T} \times\left[\tilde{z}(k+1)-\tilde{H}(k+1)\hat{U}(k+1/k+1)\right] \tag{41}$$

where $\tilde{z}(k+1)$ is the measurement vector and $\hat{U}(k+1/k+1)$ is the estimated state of the augmented KF (30)-(36).

If an actuator fault occurs, the mean value of the squared residual $SR(k+1)$ should be small (limited between zero and a threshold value). Otherwise, the mean value of $SR(k+1)$ will exceed the threshold and the sensor fault will be determined.

In the proposed approach, if the sensor fault is detected, then the RKF, which is robust to the sensor faults, is used. The actuator faults are isolated and identified through the augmented KF. In the case of an actuator fault, the reconfigurable control against actuator failures is performed. The general block diagram of the proposed fault tolerant estimation and control system is shown in Figure 2.

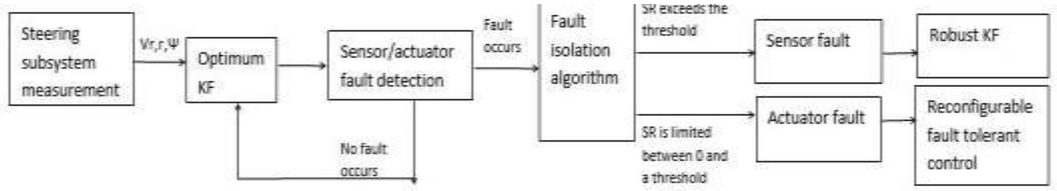

Steering subsystem measurement
Vr,r,Ψ
Optimum KF
Sensor/actuator fault detection
No fault occurs
Fault occurs
Fault isolation algorithm
SR exceeds the threshold
SR is limited between 0 and a threshold
Sensor fault
Actuator fault
Robust KF
Reconfigurable fault tolerant control

Figure 2. General block diagram of the proposed fault tolerant estimation and control system

5. Reconfigurable Control Against Actuator Failures

Using the proposed KF, the estimation values of the states and the control distribution matrix elements related to actuator/surface failure effects are found and a new control matrix is built. In this study as an optimal technique linear quadratic regulator is used. The performance index to be minimized is:

$$J=\sum_{k=0}^{\infty}\left[\hat{X}_S^T(k)Q_c\hat{X}_S(k)+U_S^T(k)R_cU_S(k)\right] \tag{42}$$

where, $\hat{X}_S(k)$ is the estimation of the steering states of AUV, Q_c is a semi-positive definite symmetrical matrix and R_c is a positive definite symmetric marix. The control input can be calculated as:

$$U_S\left(k\right)=-K_c(k)\hat{X}_S\left(k\right) \tag{43}$$

Here,

$$K_c(k)=\left(R_c+\hat{B}_S^{*T}P_c(k)\hat{B}_S^*\right)^{-1}\hat{B}_S^{*T}P_c(k)A_S^* \tag{44}$$

$\hat{B}_S^*$ is the estimation of the steering subsystem control distribution matrix B_S^*. The matrix $P_c(k)$ can be found iteratively using the Riccati equation [15]

$$\begin{gathered}P_c(k-1)=Q_c+A_S^{*T}\times\\ \left[P_c(k)-P_c(k)\hat{B}_S^*\left(R_c+\hat{B}_S^{*T}P_c(k)\hat{B}_S^*\right)^{-1}\hat{B}_S^{*T}P_c(k)\right]A_S^*\end{gathered} \tag{45}$$

As can be seen in (43)-(45) reconfigurable control procedure is applied using redefined control distribution matrix.

6. Simulation Results And Comments

6.1. Simulation Results for Sensor/Actuator Fault Detection

A yaw rate gyro fault is assumed to occur at $t=6s$. The fault in the yaw rate gyro is simulated by multiplying the standard deviation of the sensor noise with 3. Behavior of the fault detection statistic $\beta(k)$ in the presence of yaw rate gyro fault is presented in Figure 3.

As seen from presented graphs, till t = 6s, the values of fault detection statistic $\beta(k)$ lay between the admissible limits. On the other hand, the statistic $\beta(k)$ increases abruptly after 6th s (when the sensor fault occurs) and exceeds the threshold bound.

Behavior of the statistic $\beta(k)$ in the presence of actuator (rudder) fault is given in Figure 4. The simulation results show that the values of statistic $\beta(k)$ change abruptly after the 6th s (when the rudder fault occurs) and exceed the threshold bound.

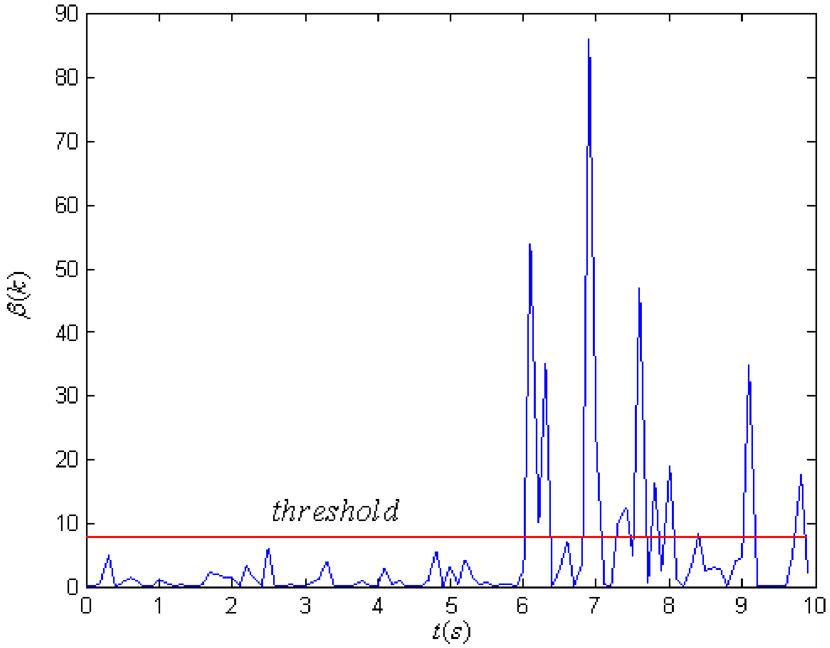


Figure 3. Behavior of the statistic $\beta(k)$ in the precsence of yaw rate gyro fault.

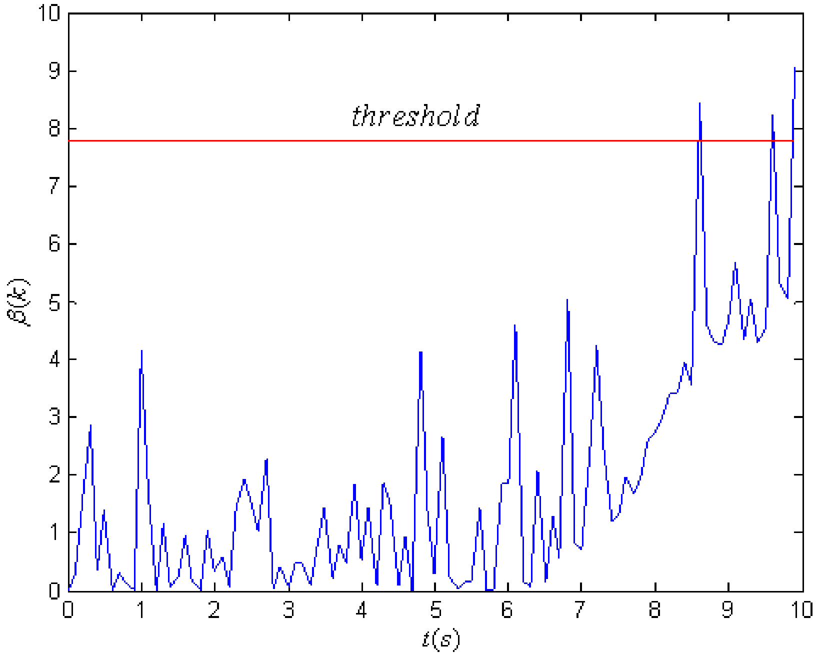


Figure 4. Behavior of the statistic $\beta(k)$ in the precsence of actuator fault.

6.2. Simulation Results for Sensor/Actuator Fault Isolation

The fault in the actuator is simulated by changing the control derivatives corresponding to rudder (by multiplying the control derivaties with 0.1 at the iteration 1000). The fault in the yaw rate gyro is simulated beginning at the 2000th iteration. For the sensor/actuator fault isolation purpose the squared error (41) between the measurements and the estimated states of the augmented KF are used. Behaviors of the squared residual $SR(k)$ and squared residual mean $SRM(k)$ are plotted in Figure 5 and Figure 6 respectively.

The presented graphs in Figures 5 and 6 show that, squared residual of the augmented KF is insensitive to the actuator fault, but sensitive to the sensor fault. In the presence of actuator fault the squared residual mean $SRM(k)$ does not exceed the threshold value. But after sensor fault occurs at the iteration 2000, the values of statistic $SRM(k)$ increase abruptly and exceed the admissible bound. As a result sensor fault is isolated.

As the augmented KF is not sensitive to actuator faults but sensitive to sensor faults, the sudden increase in the plotted values after 2000 iterations (20 sec) reveals a sensor fault (the residual exceeds the selected threshold value 1).

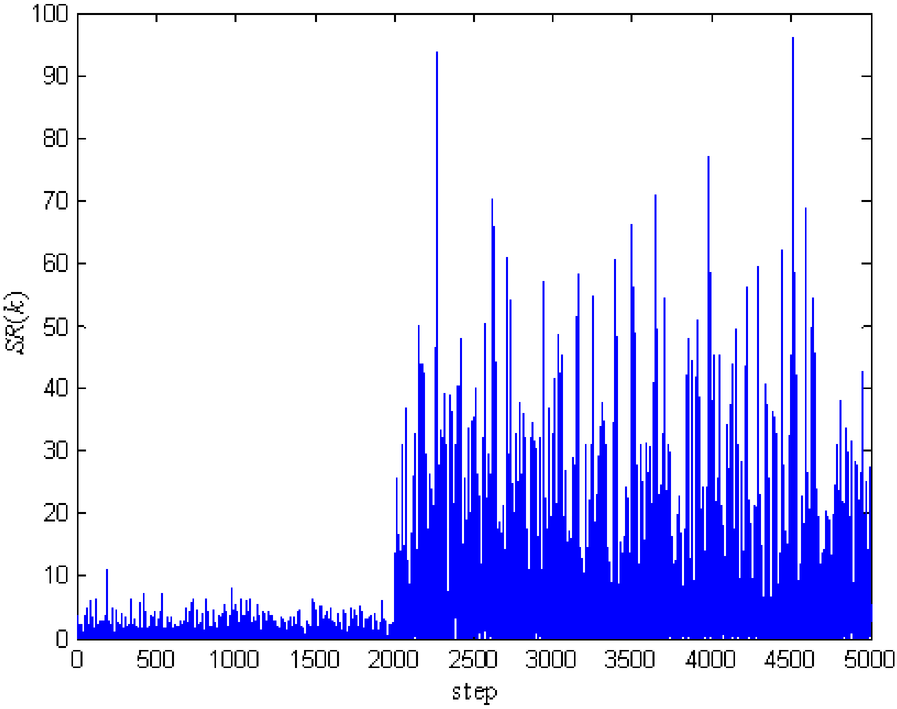


Figure 5. Behavior of the squared residual $SR(k)$.

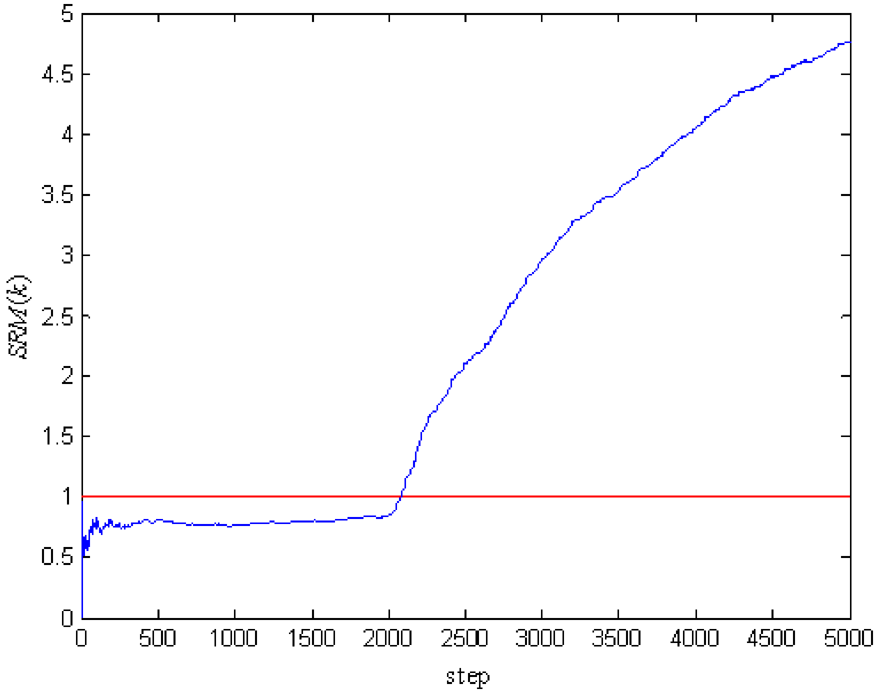


Figure 6. Behavior of the squared residual mean $SRM(k)$.

6.3. OKF Simulation Results

Simulation results of the OKF in the sensor/actuator fault free case for the steering subsystem are plotted in Figure 7. In the Figure, green line refers to actual value, red line refers to measurement value and blue line refers to Kalman value. It can be seen from the graphs, that the Kalman values converge to the actual values.

In measurement channel of r parameter, continuous bias is simulated after the 6th second via the help of the formula distributed:

$$z_r(k) = r(k) + 0.3 + \sigma_r randn \tag{46}$$

In case of malfunctions, the estimation of state variables of steering subsystem autonomous underwater vehicle by conventional KF is shown in Figure 8. As seen from Figure 8, the estimation values of parameter r diverge from the actual values after the 6th second. Consequently, it can be said that the regular KF provides bad results in case of malfunctions in measurement channel.

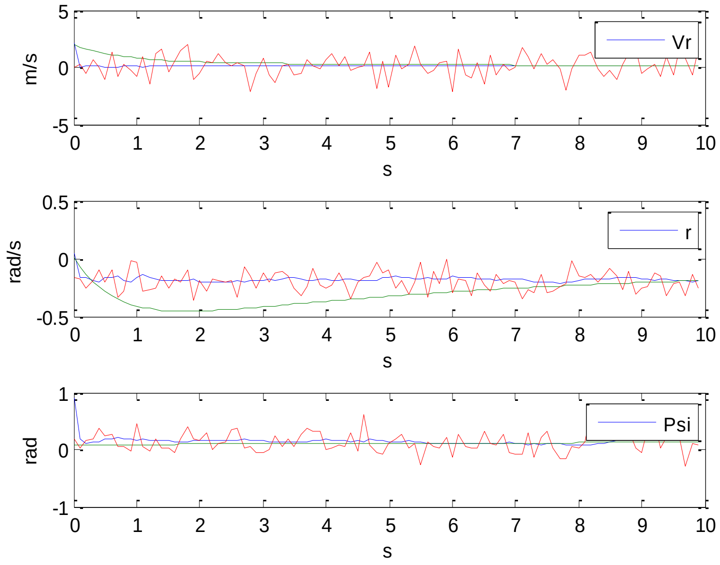


Figure 7. Kalman filter results for v_r, r, ψ parameters in case of normal conditions.

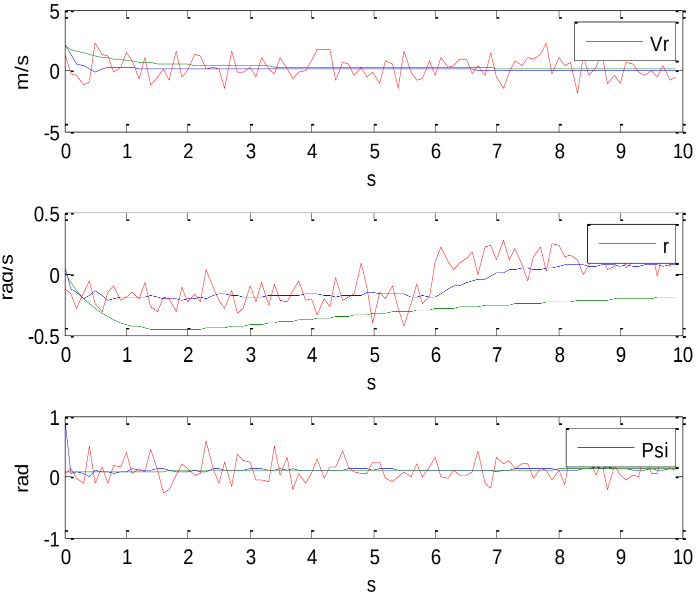


Figure 8. Conventional KF results for v_r, r, ψ parameters in case of continuous bias in yaw rate (r) measurement channel (after 6th second).

6.4. RKF Simulation Results

If sensor fault is detected and isolated, then the robust Kalman filter insensitive to the sensor faults should be used. RKF estimation results showing constant bias in r

parameter's measurement channel (after 6th second) are presented in Figure 9. As seen from Figure 9, although there is a malfunction in measurement channel, the estimation values provided by RKF converge to the actual values. In this case, filter works robustly against measurement malfunction.

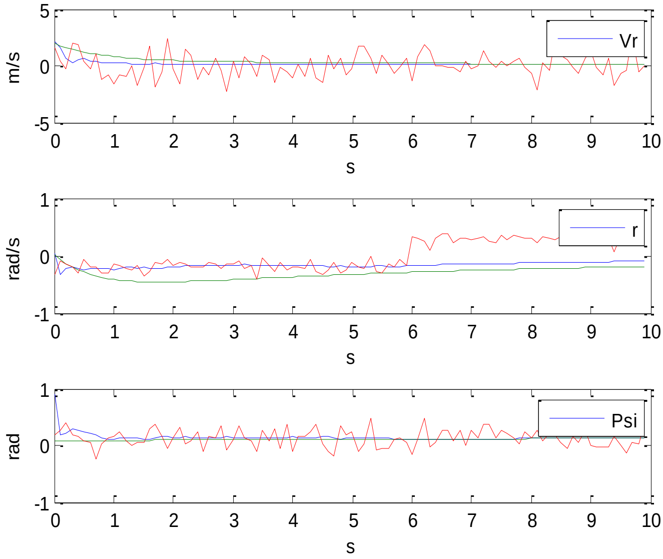


Figure 9. RKF estimation results for v_r, r, ψ parameters in case of continuous bias in yaw rate (r) measurement channel (after 6th second).

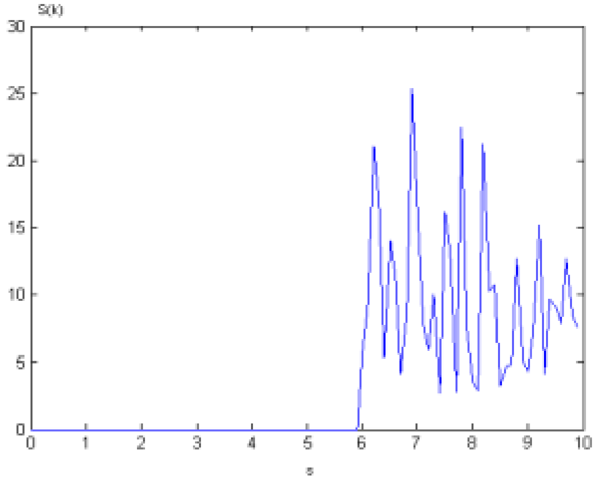


Figure 10. The change of measurement noise scale factor $S(k)$.

The change of measurement noise scale factor $S(k)$ is shown in Figure 10. Until the 6th second (when the malfunction appears), $S(k)=1$. After malfunction appears, adaptive factor increases and changes the filter gain matrix to prevent from the effect of malfunction. In conclusion, the effect of faulty measurements to estimation values is decreased and RKF provides estimation values converging to actual values.

6.5. Reconfigurable Control Simulation Results

Two different simulations are performed. In the first scenario the old control rule (conventional LQR) is used after the actuator fault. In the second scenario the actuator faults are present in the system and the optimal control rule is changed using the identified control distribution matrix.

6.5.1. Conventional LQR Control Results in the Presence of Actuator Faults

The conventional LQR control is applied to the AUV steering subsystem. The measurements are built using optimum KF (10)-(16), which estimates the state vector of the AUV steering subsystem. In simulations, the control derivatives related to rudder are changed in accordance with faulty system condition.

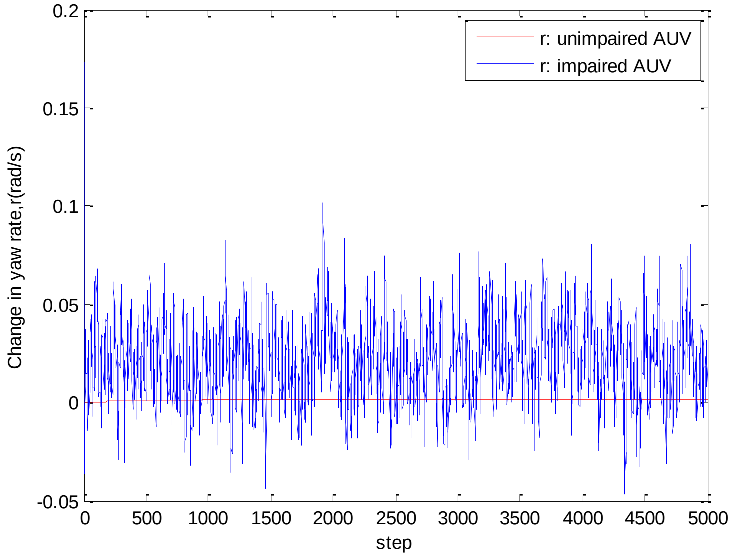


Figure 11. Yaw rate for unimpaired (red line) and impaired (blue line; rudder fault) AUV without reconfiguration.

The control rule can be found using conventional LQR control approach [35]. The actuator fault is simulated by changing the rudder control derivatives. The change in yaw rate is investigated when conventional LQR control in the presence of actuator faults is applied. The reference value is taken as $r_{ref} = 0$ rad/s. The simulation results are shown in Figure 11.

In the Figure the blue line shows the conventional LQR control plus OKF (10)-(16) simulation results for the noisy and impaired AUV (rudder fault) and the red line shows the conventional LQR control results for the deterministic and unimpaired AUV. As can be observed from Figure 11, for the noisy system the conventional LQR control plus KF simulation results diverge for the yaw rate. The similar simulation results can be obtained for the other state parameters of AUV.

The change in B_S^* matrix is affecting the system and causing the controlled values to diverge. It can be expected as the control distribution matrix is changed in this case. In both cases, when faults are present, the AUV control can't be achieved using the previously designed LQR controller.

6.5.2. Reconfigurable LQR Control Results in the Presence of Actuator Faults

The active reconfigurable control is applied to the AUV steering subsystem. The measurements are built using augmented KF (30)-(36), which estimates the extended state vector of the AUV steering subsystem and the control distribution matrix related to faulty actuator configuration (28), (29) and (31). In simulations, the control derivatives related to rudder are changed according to faulty system condition. The change in yaw rate and yaw angle are investigated when conventional LQR control in the presence of actuator faults is applied.

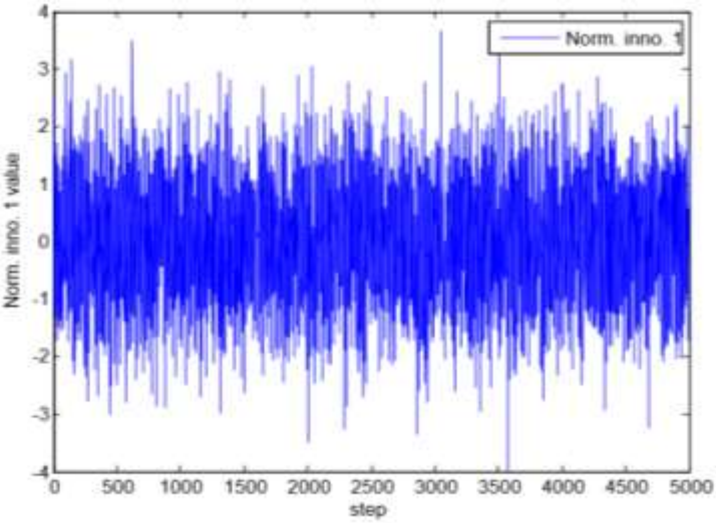


Figure 12. The side velocity (v_r) channel normalized innovation values for impaired (rudder fault) AUV with reconfiguration.

The actuator fault is simulated by changing the rudder control derivatives. To overcome this situation the control reconfiguration procedure is executed by considering the identified control distribution matrix. The changed B_S^* control distribution matrix values are estimated using the augmented KF (30)-(36) and the estimation values are used to update the LQR control rule at each step. The reference values are taken as the same with the first scenario. The results when the values of B_S^* are estimated using augmented KF are presented in Figures 12-16. The normalized innovation values of the augmented KF are shown in Figures 12-14. As is known [36], if the system operates normally, the normalized innovation sequence (36) in an augmented KF (30)-(36) is a Gaussian white noise with a zero mean and unit covariance matrix. Since the normalized innovation (36) obeys N(0, 1) distribution, for normal operation of augmented KF (30)-(36) the values will lie on the limits of the interval [–3, +3] with a probability of 0.9986. As we can see from normalized innovation graphs, there is no fault present in the system and the augmented KF is working properly.

The change in the yaw rate (r) and yaw angle (ψ) when KF based reconfigurable LQR control is applied is shown in Figures 15 and 16 respectively. The same reference values in the preceding scenarios are also used here. In the figures the blue line shows the results of the reconfigurable LQR control for the noisy and impaired AUV (rudder fault) and the red line shows the results of the LQR controller for the deterministic and unimpaired AUV. As can be observed, the results of the reconfigurable controller are close to that of the LQR controller used for the unimpaired AUV.

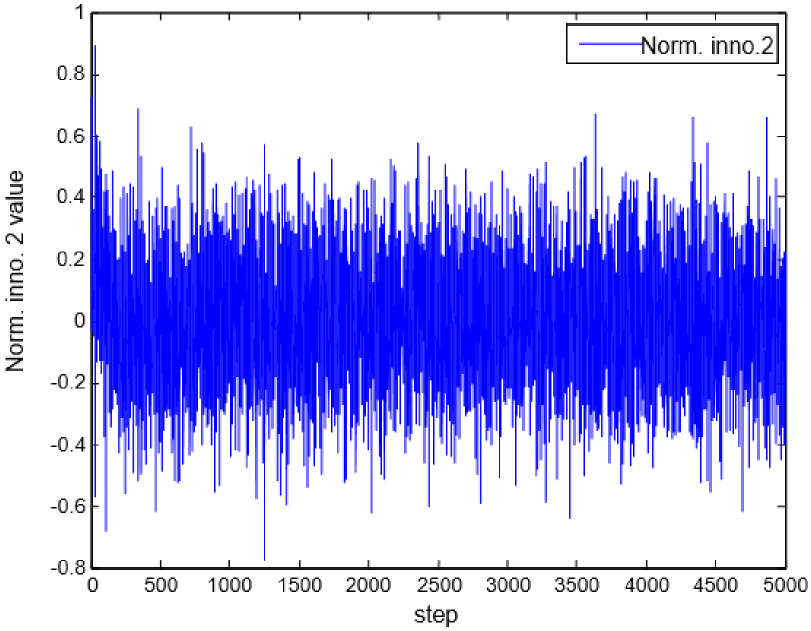


Figure 13. The yaw rate (r) channel normalized innovation values for impaired (rudder fault) AUV with reconfiguration.

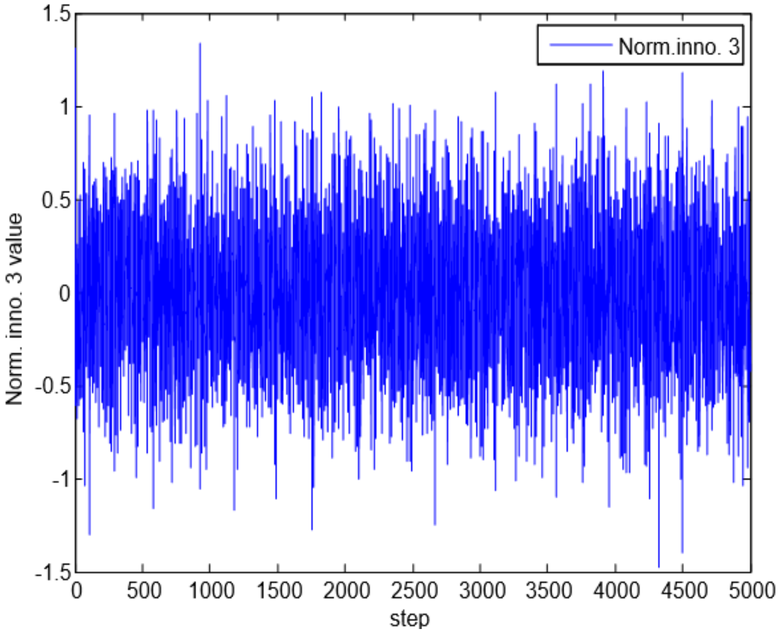


Figure 14. The yaw angle (ψ) channel normalized innovation values for impaired (rudder fault) AUV with reconfiguration.

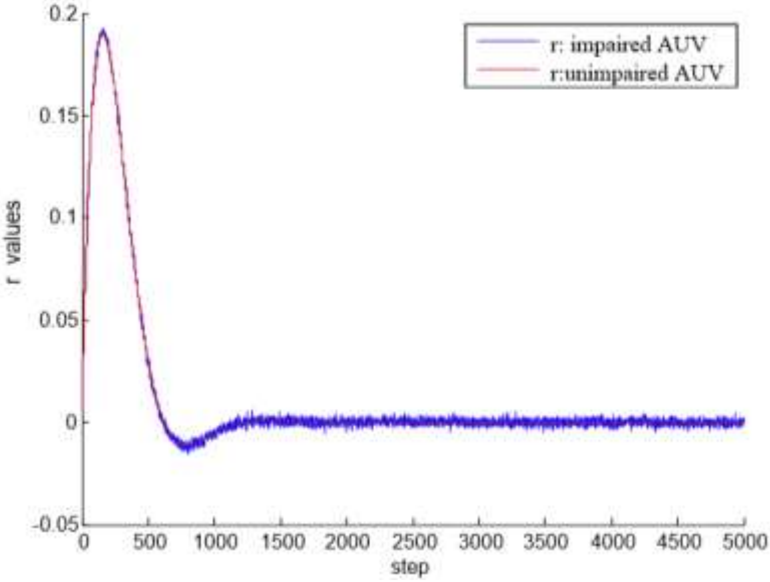


Figure 15.Yaw rate for unimpaired (red line) and impaired (blue line; rudder fault) AUV with reconfiguration.

The changed B_S^* matrix values are estimated using the augmented KF (30)-(36). LQR controller is working with the actuator fault identificator. As the heading angle goes to 0.52 radians the yaw rate stays at zero. Thus it is shown that the KF based reconfigurable LQR controller is working well.

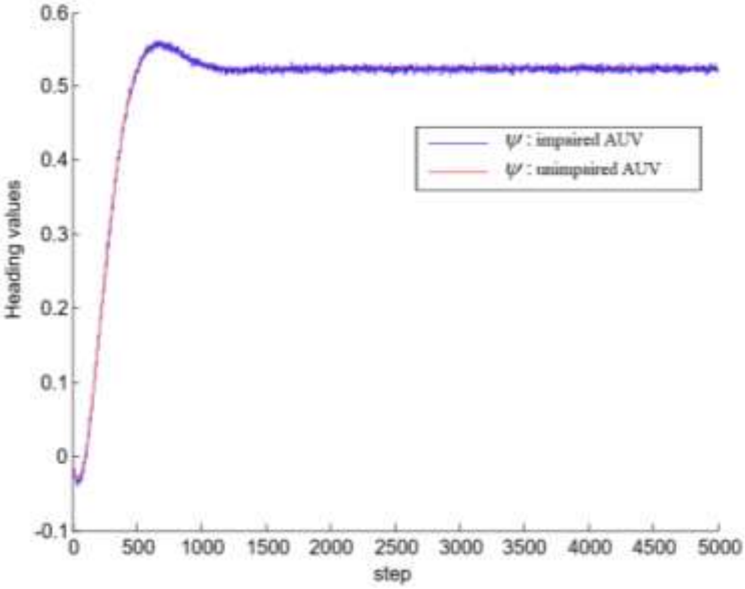


Figure 16. Yaw angle for unimpaired (red line) and impaired (blue line; rudder fault) AUV with reconfiguration.

The estimated values are used to calculate the gain value of the LQR controller at each step. Thus the controller achieves the same efficiency with the non faulty case and takes the heading value to the reference one. As it can be seen in the Figures, although there is a fault in the system the heading value goes to the reference one and the change in yaw ratio stays close to zero. This also means that the estimation algorithm (augmented KF) and the reconfigurable LQR controller are working properly for the AUV steering subsystem. The simulation shows that reconfigurable controller results for the impaired AUV dynamics are close to the unimpaired AUV dynamics.

CONCLUSION

In this study, a reconfigurable fault-tolerant flight control system against sensor/actuator faults for autonomous underwater vehicles is proposed. The proposed method is based on two types of KF: a conventional linear KF, which estimates the states of an AUV, and an augmented KF, which estimates the AUV states and the control distribution matrix elements that correspond to the faulty actuator. An approach to detecting and isolating AUV sensor/actuator faults affecting the mean of the optimum Kalman filter innovation sequence is proposed. If an actuator fault occurs, then the mean value of the residual—a squared error between the measurements and the corresponding augmented KF estimates—should be small (limited between zero and a threshold value).

Otherwise, the mean value of the residual will exceed the threshold, and the fault will be identified as a sensor fault.

The actuator faults were isolated and identified using the augmented KF. In the proposed approach, if the sensor fault is detected, then the RKF, which is robust to the sensor faults, is used. The actuator faults are isolated and identified through the augmented KF. In the case of actuator fault, the reconfigurable fault tolerant control against actuator faults is performed. Under the faulty conditions a control reconfiguration action is taken in order to keep the performance of the impaired AUV the same as that of the unimpaired AUV.

In simulations, the linearized model of dynamics of steering subsystem of AUV is considered, and the performance of the proposed sensor/actuator fault detection and identification and reconfigurable control techniques are examined. The simulation results show that the investigated AUV dynamics controlled with an augmented KF based reconfigurable controller converge to the unimpaired AUV dynamics. In other words, the proposed fault tolerant control approach becomes robust to sensor/actuator faults.

ACKNOWLEDGEMENT

This work was supported in part by TUBITAK (The Scientific and Technological Research Council of Turkey) under Grant 109M702.

REFERENCES

[1] McLean, D. and Aslam-Mir, S. 1991. “Reconfigurable Flight Control Systems”. *Proc. Int. Conference on Control* ’91, Edinburgh, pp.234-242.

[2] Patton, R. J. 1997. Fault tolerant control: The 1997 situation, in: *Proc. IFAC Symposium on Fault Detection, Supervision, and Safety for Technical Processes,* SAFEPROCESS’97, Hull, UK, pp. 1033-1055.

[3] Hajiyev, C. and Caliskan, F. 2003. *Fault Diagnosis and Reconfiguration in Flight Control Systems,* Kluwer Academic Publishers, Boston, USA.

[4] Zhang, Y. and Jiang, J. 2008. “Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems”. *Annual Reviews in Control,* 32, 229–252.

[5] Ferreira, B., Matos, A. and Cruz, N. 2011. “Fault Tolerant Dep Control of the MARES AUV”. In: A. Bartoszewicz (Ed.), Challenges and Paradigms in Applied Robust Control, *InTech,* 49-72. Available: http://www.intechopen.com/books/challenges-and-paradigms-in-applied-robustcontrol/fault-tolerant-depth-control-of-the-mares-auv

[6] Zhang, Y. and Li, X. R. 1997. "Detection and Diagnosis of Sensor and Actuator Failures using Interacting Multiple-Model estimator". *Proc. IEEE Conference on Decision and Control,* vol 5, pp. 4475-4480.

[7] Hajiyev, C. and Caliskan, F. 2005. "Sensor and Control Surface/Actuator Failure Detection and Isolation Applied to F-16 Flight Dynamics". *Aircraft Engineering and Aerospace Technology: An International Journal,* 77, 152-160.

[8] Tao, G., Qi, R. and Tan, Ch. 2011. "A Parameter Estimation Based Adaptive Actuator Failure Compensation Control Scheme". *Journal of Systems Engineering and Electronics,* 22, 1–11.

[9] Lammas, A., K.Sammut and F. He. 2010. "6-DoF Navigation Systems for Autonomous Underwater Vehicles". In book: *Mobile Robots Navigation* (Ed. Alejandra Barrera). InTech Publishing, pp.457-483.

[10] Stutters, L., H. Liu, C. Tiltman, D. J. Brown, 2008. "NavigatiTechnologies for Autonomous Underwater Vehicles". *IEEE Transactions on Systems, Man, and Cybernetics: Applications and Reviews,* vol. 38, no. 4.

[11] Hide, C., T. Moore, and M. Smith. 2004. "Adaptive Kalman Filtering Algorithms for Integrating GPS and Low Cost INS". *Proceedings of Position Location and Navigation Symposium,* Monterey, USA, pp. 227-233.

[12] Hu, C., W. Chen, Y. Chen, and D. Liu. 2003. "Adaptive Kalman Filtering for Vehicle Navigation". *Journal of Global Positioning Systems,* 2, pp. 42-47.

[13] Hajiyev, C. 2007. "Adaptive filtration algorithm with the filter gain correction applied to integrated INS/Radar altimeter". Proceedings of the IMechE, Part G: *Journal of Aerospace Engineering,* 221, pp. 847-885.

[14] Wu, N. E., Zhang, Y. and Zhou, K. 2000. "Detection, Estimation, and Accommodation of Loss of Control Effectiveness". *International Journal of Adaptive Control and Signal Processing,* 14, 775-795.

[15] Hajiyev, C. 2013. "Two-Stage Kalman Filter-Based Actuator/Surface Fault İdentification and Reconfigurable Control Applied To F-16 Fighter Dynamics". *Int. J. Adapt. Control Signal Process,* 27, 755–770.

[16] Maybeck, P. S. and Stevens, R. D. 1991. Reconfigurable Flight Control via Multiple Model Adaptive Control Methods". *IEEE Transactions on Aerospace and Electronic Systems,* 27, 470-479.

[17] Gao, Z. and Antsaklis, P. J. 1991. "Stability of the Pseudo-Inverse Method for Reconfigurable Control System". *Int. J. Control,* 53, 717-729.

[18] Mammadov, H., Hajiyev, C. 2018."Reconfigurable Fault Tolerant Flight Control for UAV with Pseudo-Inverse Technique". *IFAC-PapersOnLine,* 51(30), 83-88.

[19] Miyazawa, Y. 1992. "Robust Flight Control System Design with Multiple Model Approach". *J. Guidance,* 151, 785-788.

[20] Zhang, Y. M. and Jiang, J. 2001. "Integrated Active Fault-Tolerant Control Using IMM Approach". *IEEE Transactions on Aerospace and Electronic Systems,* 37, 1221-1235.

[21] Jiang, J. 1994. "Design of Reconfigurable Control Systems Using Eigenstructure Assignment". *Int. J. Control*, 59, 395-410.

[22] Zhang, Y. M. and Jiang, J. 2002. "Active Fault Tolerant Control System Against Partial Actuator Failures". *IEE Proc. Control Theory Application,* 149, 95-104.

[23] Nabil, E., Sobaih, A. and Abou-Zalam., B. 2013. "Active Fault-Tolerant Control of Unmanned Underwater Vehicles". *International Journal of Automation and Power Engineering (IJAPE),* 2(3), 41-51.

[24] Talange, D. and Joshi, S. 2016. "Fractıonal Order Fault Tolerant Controller for AUV". *Computational Science and Systems Engineering*, 287-292.

[25] Sun, S.-Q. Dong, L. Li, L. and Gu, S.-S. 2008. "Fault-Tolerant Control for Constrained Linear Systems Based on MPC and FDI". *International Journal of Information and Systems Sciences*, 4, 512-523.

[26] Wang, H. and Wang, Y. 1999. "Neural-Network Based Fault Tolerant Control of Unknown Nonlinear Systems". *IEE Proc., Control Theory Appl.*, 146, 389-398.

[27] Amoozgar, M. H., Chamseddine, A. and Zhang, Y. 2013. "Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter". *J. Intelligent Robot Systems*, 70, 107–117.

[28] Varga, A. 2010. "Detection and Isolation of Actuator/Surface Faults for a Large Transport Aircraft". In: *Fault Tolerant Flight Control*, C. Edwards et al. Eds., LNCIS 399, Springer-Verlag Berlin Heidelberg, pp. 423–448.

[29] Jalving, B. 1994. "The NDRE – AUV Flight Control System". *IEEE Journal of Oceanic Engineering*, 19, 497-501.

[30] Fossen, T. I. 1994. *Guidance and Control of Ocean Vehicles,* John Wiley & Sons, New York.

[31] Lingli, N. I. 2001. *Fault-Tolerant Control of Unmanned Underwater Vehicles,* PhD Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[32] Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediciton Problems". *Transactions of the ASME Journal of Basic Engineering,* 82, 35-45.

[33] Hajiyev C., Vural S. Y., Shumsky A., Zhirabok, A. 2018. "Robust Kalman Filter Based Estimation of AUV Dynamics". *IFAC-PapersOnLine,* 51(30), 424-429.

[34] Hajiyev, C. and Caliskan, F. 2001. "Integrated Sensor/Actuator FDI and Reconfigurable Control for Fault-Tolerant Flight Control System Design". *The Aeronautical Journal*, 105, 525-533.

[35] Vural, Y., and Hajiyev, C. 2014. "A Comparison of Longitudınal Controllers for Autonomous UAV". *International Journal of Sustainable Aviation (IJSA),* 1(1), 58-71.

[36] Hajiyev, C., Vural, S. Y. 2019. "Active Fault Tolerant Lateral Control Against Actuator Faults Applied to AUV Dynamics". *Proc. 4th Conference on Control and Fault Tolerant Systems (SysTol),* Casablanca, Morocco, September 18-20, 2019, pp. 153-158.

Biographical Sketches

Chingiz Hajiyev, PhD, DSc (Eng.)

Affiliation: Istanbul Technical University, Istanbul, Turkey

Education: Moscow Aviation Institute

Business Address: Aeronautics and Astronautics Faculty, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey

Research and Professional Experience:

Chingiz Hajiyev graduated in 1981 from the Faculty of Automatic Control Systems of Flying Apparatus, Moscow Aviation University (Moscow, Russia) with honour diploma. He received the PhD and DSc (Eng.) degrees in Process Control from Superior Certifying Commission at the Council of Ministers of the USSR from Azerbaijanian Scientific and Production Association (ASPA) "Neftgazavtomat" (Sumgait, Azerbaijan), in 1987 and 1993, respectively.

Professional Appointments:

From 1987 to 1994 Dr. Hajiyev worked as a Scientific Worker, Senior Scientific Worker, Chief of the Information-Measurement Systems Dept. at the ASPA "Neftgazavtomat". Since 1994 to 1996 he was a Leading - Scientific Worker at the Institute of Cybernetics of the Academy of Sciences of Azerbaijan Republic. He was also a Professor in the Department of Electronically-Calculated System Design, Azerbaijan Technical University, where he had been teaching 1995-1996.

Since 1996 he has been with Department of Aeronautics and Astronautics, Istanbul Technical University (Istanbul, Turkey), where he is currently a Professor. From December 2016 he is head of the Aeronautical Department of ITU.

Honors:

1981: Honors Certificate at Moscow Aviation Institute (Russia).
1994: Award from the International Science Foundation (USA).

2012: Best Paper Award, 13th International Carpathian Control Conference 2012, High Tatras, Slovak Republic.

2016: Sustainable Aviation Research Society (SARES) Science Award.

2016: Award for the Best Paper Presented of the 3rd IEEE International Workshop on Metrology for Aerospace, Florence, Italy, 2016.

2016: Recognized Reviewer Award - Ocean Engineering Journal, (UK).

2016: Recognized Reviewer Award - Aerospace Science and Technology Journal, (France).

2017: Outstanding Reviewer Award - Aerospace Science and Technology Journal, (France).

2019: Project Performance Award - The Scientific and Technological Research Council of Turkey (TUBITAK), (Turkey).

Publications from the Last 3 Years:

1. Guler, D.C. and C. Hajiyev (2017). *Gyroless Attitude Determi-nation of Nanosatellites Single-Frame Methods and Kalman Filtering Approach*. LAP LAMBERT Academic Publishing, Saarbrücken Deutschland/Germany, 175 p. ISBN 978-3-330-04781-5.
2. Cilden, D., H.E. Soken, C. Hajiyev. (2017). Nanosatellite Attitude Estimation From Vector Measurements Using SVD-Aided UKF Algorithm. *Metrology and Measurement Systems,* Vol. 24, No. 1, pp. 113-125. DOI: 10.1515/mms-2017-0011.
3. Hajiyev, C., D.C. Guler. (2017). Review on Gyroless Attitude Determination Methods for Small Satellites. *Progress in Aerospace Sciences,* Vol. 90, pp. 54-66.
4. Guler, D.C., E.S. Conguroglu, C. Hajiyev (2017). Single-Frame Attitude Determination Methods for Nanosatellites. *Metrology and Measurement Systems,* Vol. 24, No. 2, pp. 313-324.
5. Hajiyev, C. (2017). Fault tolerant estimation of electro-mechanical actuator parameters via robust EKF. *International Journal of Sustainable Aviation,* Vol. 3, No. 2, pp. 100-114.
6. Vural, S.Y. and C. Hajiyev (2017). Two-stage Kalman filter for estimating the actuator control effectiveness factors of UAV. *International Journal of Sustainable Aviation,* Vol. 3, No. 3, pp. 233-251.
7. Guler, D.C., C. Hajiyev (2017). Singular value decomposition based satellite attitude determination using different sensor configurations, *International Journal of Metrology and Quality Engineering,* Vol.8, No. 15, pp.1-6. DOI: 10.1051/ijmqe/2017010.
8. Guler, D.C., C.Hajiyev (2017). Integrated SVD/EKF Attitude Estimation with UD Factorization of the Measurement Noise Covariance. In *Proc. 8th International*

Conference on Recent Advances in Space Technologies (RAST-2017), 19-22 June 2017, Istanbul, Turkey.

9. Guler, D.C., C. Hajiyev, J. Dasdemir (2017). Fault Tolerant Control of Attitude Dynamics of Nanosatellites. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017)*, 19-22 June 2017, Istanbul, Turkey.
10. Hajiyev, C., D.C. Guler, Ye. Somov (2017). Attitude Determi-nation of Nanosatellites in the Sun-Eclipse Phases. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017),* 19-22 June 2017, Istanbul, Turkey.
11. Bağci, M., C.Hajiyev (2017). Comparison of Linear and Nonlinear Measurements Based Orbit Estimation EKFs. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017),* 19-22 June 2017, Istanbul, Turkey.
12. Hajiyev, C., D.C. Guler, H.E. Soken (2017). Nontraditional UKF Based Nanosatellite Attitude Estimation with the Process and Measurement Noise Covariances Adaptation. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017),* 19-22 June 2017, Istanbul, Turkey.
13. Somov, Ye., S. Butyrin, S. Somov, C. Hajiyev (2017). Precise Astroinertial Attitude Determination of a Maneuvering Land-survey Satellite. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017),* 19-22 June 2017, Istanbul, Turkey.
14. Guler, D.C., Z. Kaymaz, C.Hajiyev (2017). Geomagnetic Models at Low Earth Orbit and Their Use in Attitude Determination. In *Proc. 8th International Conference on Recent Advances in Space Technologies (RAST-2017),* 19-22 June 2017, Istanbul, Turkey.
15. Guler, D.C., H.E. Soken, C. Hajiyev (2017). Non-Traditional Robust UKF against Attitude Sensors Faults. In *Proc. 31st International Symposium on Space Technology and Science (ISTS-2017),* June 3-9, 2017, Himegin Hall, Matsuyama-Ehime, Japan, paper ISTS-2017-d-077/ISSFD-2017-077.
16. Hajiyev, C., H.E. Soken, D.C. Guler, (2017). Q-Adaptation of SVD-aided UKF Algorithm for Nanosatellite Attitude Estimation. *Preprints of the 20th World Congress The International Federation of Automatic Control*, Toulouse, France, July 9-14, 2017, pp. 8603-8608.
17. Guler, D.C., C. Hajiyev (2017). Integrated SVD/EKF for Nano-satellite Attitude Determination in case of Magnetometer Faults. *7th European Conference for Aeronautics and Space Sciences (EUCASS),* Milano, Italy, 3-6 July, 2017, 9 P.
18. Contreras, A.M., C. Hajiyev (2017). Fault Tolerant Integrated Barometric-Inertial-GPS Altimeter. *7th European Conference for Aeronautics and Space Sciences (EUCASS*), Milano, Italy, 3-6 July, 2017, 9 P.
19. Hajiyev, C., D.B. Kamiloglu (2017). Kalman Filter Based Loosely-Coupled GPS/INS Integration for Boeing-747 Aircraft Model. *Proc. of the 9th Ankara*

International Aerospace Confe-rence 20-22 September 2017 -METU, Ankara Turkey, Paper AIAC-2017-091, 11P.

20. Contreras, A.M., Hajiyev C. (2018). Integration of Baro-Inertial-GPS Altimeter via Complementary Kalman Filter. In: Karakoç T., Colpan C., Şöhret Y. (eds) *Advances in Sustainable Aviation.* Springer, Cham, pp. 251-268. DOI: https://doi.org/10.1007/978-3-319-67134-5_18.
21. Hajiyev, C., Hacizade U. (2018). Testing the Determinant of the Innovation Covariance Matrix Applied to Aircraft Sensor and Actuator/Surface Fault Detection. In: Karakoç T., Colpan C., Şöhret Y. (eds) *Advances in Sustainable Aviation*. Springer, Cham, pp. 269-284. DOI: https://doi.org/10.1007/978-3-319-67134-5_19.
22. Cilden-Guler, D., Kaymaz, Z., Hajiyev, C. (2018). Evaluation of geomagnetic field models using magnetometer measurements for satellite attitude determination system at low earth orbits: Case studies. *Advances in Space Research,* 61(1), pp. 513-529.
23. Cilden-Guler, D., Hajiyev, C., Dasdemir, J. (2018). Fault tolerant control of attitude dynamics of nanosatellites. *International Journal of Sustainable Aviation*, Vol. 4, Iss. 2, pp.99-113.
24. Hajiyev, Ch. and Sofyali, A. (2018). Spacecraft localization by indirect linear measurements from a single antenna. *Aircraft Engineering and Aerospace Technology*, Vol. 90, No. 5, pp. 734-742. DOI: 10.1108/AEAT-12-2015-0245.
25. Hajiyev, C. and Kamiloğlu, D.B. (2018). Design of loosely-coupled INS/GPS integration system applied to Boeing-747 aircraft model, *Int. J. Sustainable Aviation*, Vol. 4, Nos. 3/4, pp.163-177.
26. Cilden-Guler, D., Soken, H. E., Hajiyev, C. (2018). SVD-Aided UKF for Estimation of Nanosatellite Rotational Motion Parameters. *WSEAS Transactions on Signal Processing,* Vol. 14, pp. 27-35.
27. Hajiyev, C., Cilden-Guler D. (2018). Gyroless Nanosatellite Attitude Estimation in Loss of Sun Sensor Measurements. *IFAC-PapersOnLine*, 51(30), 89-94.
28. Cilden-Guler, D., Hajiyev C. (2018). UD Factorization Based Non-Traditional Attitude Estimation of Nanosatellite with Rate Gyros FDI. *IFAC-PapersOnLine*, 51(30), 95-100.
29. Hajiyev C. (2018). An Innovation Approach Based Model Change Detection Applied to UAV Actuator/Surface FDI. *IFAC-PapersOnLine*, 51(30), 77-82.
30. Hajiyev, C., Vural, S.Y., Shumsky A., Zhirabok, A. (2018). Robust Kalman Filter Based Estimation of AUV Dynamics. *IFAC-PapersOnLine*, 51(30), 424-429.
31. Mammadov, H., Hajiyev C. (2018). Reconfigurable Fault Tolerant Flight Control for UAV with Pseudo-Inverse Technique. *IFAC-PapersOnLine*, 51(30), 83-88.
32. Vural, S.Y., Janset Dasdemir J., Hajiyev C. (2018). Passive Fault Tolerant Lateral Controller Design for an UAV. *IFAC-PapersOnLine*, 51(30), 446-451.

33. Somov, Ye., Hajiyev, Ch. (2018). Satellite Guidance and Control During Operative Optoelectronic Imagery for Disaster Management. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Volume XLII-3/W4, 2018 GeoInformation For Disaster Management (Gi4DM), 18-21 March 2018, Istanbul, Turkey, pp. 475-482. https://doi.org/ 10.5194/isprs-archives-XLII-3-W4-475-2018.
34. Hajiyev, Ch., Soken, H.E. (2018). Q-Adaptation of UKF Algorithm for Estimation of the Autonomous Underwater Vehicles Dynamics. *Proceedings of the 5th International Conference of Control, Dynamic Systems, and Robotics (CDSR'18),* Niagara Falls, Canada - June 7 - 9, 2018, Paper No. 103, DOI: 10.11159/ cdsr18.103.
35. Hajıyev, C., Cılden-Guler, D., Hacizade, U. (2019).Two-Stage Kalman Filter for Estimation of Wind Speed and UAV States by using GPS, IMU and Air Data System. *WSEAS Transactions On Electronics*, Vol. 10, pp. 60-65.
36. Hajiyev, C.,Soken, H.E., Cilden-Guler, D. (2019). Nontraditional Attitude Filtering with Simultaneous Process and Measurement Covariance Adaptation. *Journal of Aerospace Engineering (ASCE),* 32(5): 04019054.
37. Cilden-Guler, D., Raitoharju, M., Piche, R., Hajiyev, C. (2019). Nanosatellite attitude estimation using Kalman-type filters with non-Gaussian noise. *Aerospace Science and Technology*, Vol. 92, pp.66-76.
38. Contreras A.M., Hajiyev C. (2019). Comparison of Conventional and Robust Adaptive Kalman Filters Based Integrated Altimeters. *Proc. of the 20th International Carpathian Control conference* (ICCC-2019), Krakow, Wieliczka, Poland, 26-29 May, 2019, IEEE, 6P.
39. Contreras, A.M., Hajiev, C. (2019). Comparison of Linear and Nonlinear Kalman Filters Based Integrated Inertial, Baro and GPS Altimeters. Chapter in the book "*Advances in Engineering Research*", Editor V.M Petrova, Vol. 31. Nova Science Publishers, Inc., NY, USA, pp. 117-155.
40. Cilden-Guler, D., Kaymaz, Z., Hajiyev, C. (2019). Assessment of Magnetic Storm Effects under Various Magnetometer Noise Levels for Satellite Attitude Estimation. Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 769-773.
41. Soken, H.E., Hacizade, C. (2019). Tuning the Attitude Filter: A Comparison of Intuitive and Adaptive Approaches. Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp.747-752.
42. Bagci, M., C. Hajiyev,C. (2019). Measurement Conversion Based RKF for Satellite Localization via GPS. Proc. 9th International Conference on Recent

Advances in Space Technologies “Space for the Sustainable Development Goals” (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 861-868.

43. Turan, E., Hajiyev, C. (2019). Performance Comparison of Guidance Algorithms for Planetary Landing, Asteroid Intercept and Rendezvous. Proc. 9th International Conference on Recent Advances in Space Technologies “Space for the Sustainable Development Goals” (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 919-925.
44. Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). EKF for Wind Speed Estimation and Sensor Fault Detection Using Pitot Tube Measurements. Proc. 9th International Conference on Recent Advances in Space Technologies “Space for the Sustainable Development Goals” (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 887-893.
45. Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). Two-Stage Kalman Filter for Estimation of Wind Speed and UAV Flight Parameters Based on GPS/INS and Pitot Tube Measurements. Proc. 9th International Conference on Recent Advances in Space Technologies “Space for the Sustainable Development Goals” (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 875-880.
46. Haciyev, C., Hacizade, U., Cilden-Guler, U. (2019). Data Fusion for Integrated Baro/GPS Altimeter. Proc. 9th International Conference on Recent Advances in Space Technologies “Space for the Sustainable Development Goals” (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 881-885.
47. Hajiyev, C., Vural, S.Y. (2019). Active Fault Tolerant Lateral Control Against Actuator Faults Applied to AUV Dynamics. *Proc. 4th Conference on Control and Fault Tolerant Systems (SysTol),* Casablanca, Morocco, September 18-20, 2019, pp. 153-158.
48. Cilden-Guler, D., Hajiyev, C. (2019). SVD-aided EKF attitude estimation with UD factorized measurement noise covariance. *Asian Journal of Control,* Vol. 21, pp.1423–1432.

Sitki Yenal Vural

Affiliation: Istanbul Technical University, Istanbul, Turkey

Education: Istanbul Technical University, Istanbul, Turkey

Business Address: Aeronautics and Astronautics Faculty, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey

Research and Professional Experience:

Sitki Yenal Vural is a PhD student at the Aeronautical Engineering Department at Istanbul Technical University. He graduated from the Faculty of Mechanical Engineering, Istanbul Technical University in 2002. He received his MSc degree in Aerospace Engineering from Istanbul Technical University in 2008. His research interests include unmanned aerial vehicles (UAV), flight control techniques, Kalman filtering applications and fault-tolerant flight control.

Professional Appointments:

Sitki Yenal Vural is a PhD student at the Aeronautical Engineering Department at Istanbul Technical University. He worked in Turkish Aerospace Industries (TAI) as an Aeronautics-Avionics Systems Design Engineer between 2010 and 2012. He has been working in Turkish Airlines since then.

Honors:

2012: Tubitak Young Researcher Scholarship, 2012 (Tubitak project under grant 109M702).

Publications from the Last 3 Years:

1. Vural, S.Y. and C. Hajiyev (2017). Two-stage Kalman filter for estimating the actuator control effectiveness factors of UAV. *International Journal of Sustainable Aviation,* Vol. 3, No. 3, pp. 233-251.
2. Hajiyev, C., Vural, S.Y., Shumsky A., Zhirabok, A. (2018). Robust Kalman Filter Based Estimation of AUV Dynamics. *IFAC-PapersOnLine*, 51(30), 424-429.
3. Vural, S.Y., Janset Dasdemir J., Hajiyev C. (2018). Passive Fault Tolerant Lateral Controller Design for an UAV. *IFAC-PapersOnLine*, 51(30), 446-451.
4. Hajiyev, C., Vural, S.Y. (2019). Active Fault Tolerant Lateral Control Against Actuator Faults Applied to AUV Dynamics. *Proc. 4th Conference on Control and Fault Tolerant Systems* (SysTol), Casablanca, Morocco, September 18-20, 2019, pp. 153-158.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8

Chapter 6

SELF-ORGANIZATION AND CONTROL RECONFIGURATION OF UNMANNED AUTONOMOUS SYSTEMS FOR IMPROVED RESILIENCE

George Vachtsevanos*[*]*, Sehwan Oh and Benjamin Lee
The Georgia Institute of Technology, Atlanta, Georgia US

ABSTRACT

This contribution addresses the reconfigurable design and operation of complex systems, with emphasis on autonomous systems, building upon concepts of autonomy, incipient failure diagnosis and prognosis algorithms, while introducing a novel methodology for reconfigurable design, control and/or operation formulated as an optimization problem where new or reconfigured designs and their operational characteristics are optimized to perform as designed/desired. The innovative feature of the adverse event mitigation architecture is the utility of real-time prognostic information in the design of the control algorithms. Given accurate on-line prognostic information in terms of estimates of the Remaining Useful Life (RUL) or Time to Failure (TTF) of a failing component/subsystem, the proactive fault accommodation system manages the accumulation of further damage through control actions until major flight/mission objectives are achieved although the system is in an impaired state. This approach constitutes a major paradigm shift in the way fault-tolerant systems are designed and operated. The implications to system survivability, safety and availability to complete a critical flight/mission are significant. Existing/published research focuses either on single component (i.e., navigation controller), or specific system (i.e., single type of Unmanned Aerial Vehicle UAV) reconfiguration, or reconfigurable control [without providing the fundamentals of a general and justifiable methodology for overall (hardware/software components) system reconfiguration. Reconfiguration is achieved based on metrics related to measures of effectiveness and performance. Once metrics are defined, graph-

[*] Corresponding Author’s Email: gjv@ece.gatech.edu.

based (dependency, directed graphs) and non-homogeneous Markov-based modeling approaches are followed to arrive at different system configurations and choose the best alternative according to mission requirements.

We introduce two complementary approaches to fault tolerance or reconfigurable control of complex unmanned systems. Both assume that an incipient failure or fault is detected and the failing component's remaining useful life is estimated. The fault to failure evolution allows a sufficient period for the application of the reconfiguration strategy. A self-organization method is introduced as a compensatory measure to maintain system functionality under the presence of failure modes. It is noted that resilience requirements refer to severe disturbances, i.e., failure modes compared to usual disturbances compensated by conventional technologies such as robust or PID control. A typical unmanned autonomous ground vehicle – the hexapod – is employed as the testbed for the development and validation of the self-organizing strategy. Methods to understand system behavior include data acquisition, system modeling, and proper construction of performance metrics; the strategy includes a policy to address the changing system conditions and success criteria to evaluate the optimal action. The physical, functional, nonlinear dynamic, and graph theoretic models will be considered to examine system behaviors under both normal and faulty conditions. Then, the self-organization strategy is introduced in the form of a Markov Decision Process (MDP) with dynamic programming for optimal performance. Finally, the success criteria for the control method are constructed with Lyapunov stability conditions so that the self-organization strategy can be modified throughout the system operation for system resilience regarding stability and resource limitations. Simulation results are presented to demonstrate the efficacy of the approach. The second approach introduces a design methodology for resilient-based control reconfiguration of Unmanned Autonomous Systems (UAS) when extreme disturbances, such as a largely growing fault or a component failure mode occur. An optimal control approach with Differential Dynamic Programming (DDP) and Model Predictive Control (MPC) is deployed as a means for control authority redistribution and reconfiguration; the system continues performing its mission while compensating for the impact of the extreme disturbances. Prognostic knowledge is considered in a quadratic cost function of the optimal control problem as a soft constraint. A trade-off parameter is introduced between the prognostic constraint and the terminal cost. An autonomous ground operable under-actuated hovercraft is employed to demonstrate the efficacy of the proposed reconfiguration strategy.

Keywords: resilience, self-organization, control reconfiguration, health management, reasoning, autonomous systems

1. Introduction/Motivation

The emergence of complex and autonomous systems, such as modern aircraft, Unmanned Aerial Vehicles (UAVs), automated industrial processes, among many others, is driving the development and implementation of new control technologies that are aimed to accommodate incipient failures and maintain a stable system operation for the duration of the emergency. The motivation for the research in this broad area began in the area of avionics and flight control systems for the purpose to improve the reliability and safety of aircraft. In the scope of this work, reliability is defined as,

Definition 1.1 (Reliability) *The probability that a system will perform within specified constraints for a given period of time.*

Fatal accidents in the worldwide commercial jet fleet during the years 1987-2005 were due primarily to (i) controlled flight into terrain, (ii) loss-of-control in flight and (iii) system/component failure or malfunction [1]. In a coordinated effort to improve aviation safety, industry and government worked together to reduce the number of fatal commercial aircraft accidents, which dropped by 65% during the period of 1996-2007 [2]. As a result of this effort, accidents due to controlled flight into terrain have been virtually eliminated through the addition of various safeguards, but the same cannot be said for accidents due to loss-of-control in flight and system/component failure or malfunctions. System/component failure and malfunctions are recognized as contributing factors to aircraft loss-of-control in flight.

Military fixed wing aircraft programs drove much of the reconfigurable controls work in the 1980s and 1990s, but applications to passenger and general aviation began to receive substantial interest under the NASA Aviation Safety Program (ASP) founded in 1997 [3]. NASA ASP research focuses on vehicle design, manufacturing, operation, and maintenance. More recently, two major NASA ASP initiatives, Integrated Vehicle Health Management (IVHM) and the Integrated Resilient Aircraft Control (IRAC) project, are addressing these needs by funding the private sector, academia and government sponsored laboratories to develop innovative technologies, tools and methods to protect against hardware system/component failure or malfunctions.

The current state of the art in aircraft IVHM is exemplified by the Joint Strike Fighter (JSF) program. This program has incorporated Prognostics Health Management (PHM) into its design using sensors, advanced processing and reasoning, and a fully integrated system of information and supplies management. The on-board JSF PHM system is hierarchical, dividing the aircraft into areas such as propulsion and mission systems. Area data is generated by a mixture of dedicated, purpose-built sensors and analysis on existing control sensors to identify degradation and failures, which are compiled and correlated by area reasoners and then correlated by system-level model- based reasoners. Maintenance data-links transmit vehicle health data to ground-based information systems focused on maintenance and management of the supply chain. Prognostic events are detected by prognostic built-in-tests, automated post-flight trending, and reasoning with an emphasis on disambiguating sources of degradation rather than failure. An autonomic logistics information system provides logistic support to the end-user and also provides off-board trending across the entire JSF fleet [4]. Although these represent significant achievements, it is widely acknowledged that more work is required to build reliable, effective health management systems that rely upon fundamental break-through technologies in health management and resilient design and operation of complex engineered systems to enable safe and efficient implementation of mitigation strategies [5].

2. Technical Approach

A systematic methodology to resilience design is introduced, enabled by self-organization and reconfigurable control concepts; situational awareness assists to understand and model extreme disturbances and their impact on the integrity of the selected system (the autonomous system) while compensatory behaviors are derived and evaluated and, finally, the “best” system behavior is estimated using optimization tools that will decide on the recovery process from extreme disturbances relying on self-organization and control reconfiguration techniques. For experimentation and testing, the baseline consists of two autonomous unmanned systems, a hexapod and a hovercraft specifically designed to exhibit sensing, control and resilience capabilities.

2.1. Situational Awareness

The first level of protection against disturbances involves monitoring the system state, i.e., correlating internal and external events to temporal, spatial and functional constraints. Then, the design for resilience of cyber physical systems imposes a crucial requirement: Faults/incipient failure modes, caused from severe disturbances that may endanger the operational integrity of the system at hand must be detected reliably. Our proposed design methodology is augmented with novel, rigorous and verifiable methods/tools for fault diagnosis and failure prognosis accompanied by appropriate performance metrics, such as quantifiable metrics of confidence, risk and trust [8, 9, and 10]. This capability needs to be evaluated at the design stage as a critical element of resilience. The approaches considered use multi-objective optimization to assess the best responses (with respect to resilience) which will flow into the design as a constraint. Rigorous and verifiable situational awareness couples directly into self-organization and reconfigurable or fault tolerant control as a fundamental enabler for immunity and self-healing of complex cyber physical systems. We augment these methodologies with new and reliable uncertainty representation and management algorithms that will enhance basic properties of design for resilience.

The basic elements of the scheme are shown in Figure 1.

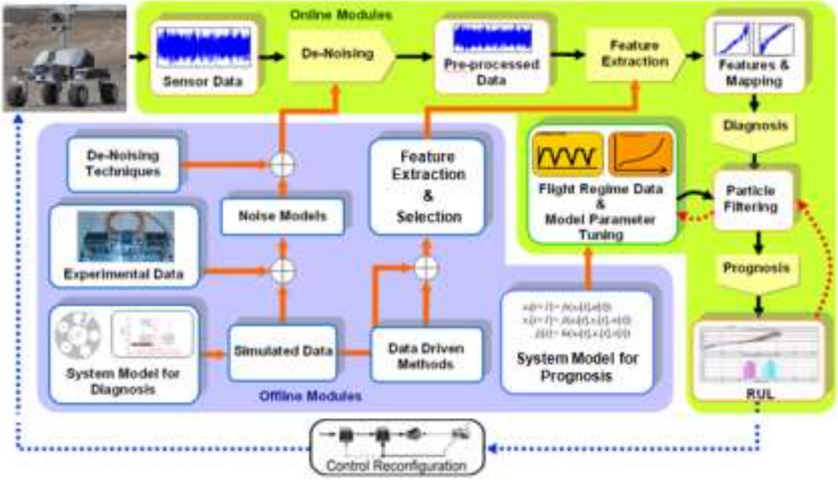


Figure 1. Overall architecture for implementation of fault diagnosis and failure prognosis algorithms.

2.2. Fault Diagnosis and Failure Prognosis

Adverse events can lead to potentially serious consequences if left undetected. Researchers in such diverse disciplines as medicine, engineering, the sciences, business and finance have been developing methodologies to detect fault (failure) or anomaly conditions, pinpoint or isolate which component/object in a system/process is faulty, and decide on the potential impact of a failing or failed component of the health of the system [8]. Fault Detection and Diagnosis (FDD) consists of two elements, fault detection and fault diagnosis. The goal of the fault detection element is to apply validated technologies to detect anomalies from adverse events throughout the system [5].

Definition 2.1. (Fault Detection [9]) *an abnormal operating condition that is detected and reported.*

Whereas, the fault diagnosis element is developed to integrate and validate technologies to determine the causal factors, the nature and severity of an adverse event (or fault identification) and to distinguish that event within a family of potential adverse events (fault isolation) [5].

Definition 2.2. (Fault Diagnosis [9]) *Detection, isolation and identification of an impending or incipient failure condition—the affected component (subsystem, system) is still operational even though at a degraded mode."*

Over the past three decades, the growing demand for reliability, maintainability, and survivability in dynamic systems has drawn significant research in FDD [10, 11, 12, 13, and 14]. Historically, FDD has been used in FTC to retrieve fault information from the system for use in a control recovery strategy and procedure, which is commonly referred to as reconfiguration. Research by [15, 16] demonstrated that state estimation-based schemes are most suitable for fault detection since they are inherently fast and cause a very short time delay in real-time decision making. However, the information from state estimation-based algorithms may not be detailed enough for subsequent control system reconfiguration. Work presented by [17, 18] recommends that parameter estimation schemes be used for control reconfiguration while using state estimation-based schemes for FDD. A unified approach to state estimation/prediction and parameter estimation/identification for FDD using particle filtering was thoroughly studied by M. Orchard [19].

2.3. Failure Prognosis and Long-Term Prediction

The term prognosis has been used widely in medical practice to imply the foretelling of the probable course of a disease. In the industrial and manufacturing fields, prognosis is interpreted to answer the question, “What is the remaining useful life (RUL) of a machine or component once an impending failure condition is detected, isolated, and identified?”

Definition 2.3. (Prognosis [9]) *the ability to predict accurately and precisely the remaining useful life (RUL) of a failing component or subsystem.*

Prognosis in engineering systems is a relatively new area of study with many unanswered questions. Current research interests have been initiated by government agencies over the past decade to reduce the life- cycle costs and improve overall reliability of ground equipment/vehicles, aircraft and aerospace systems.

2.4. Particle Filtering - A Novel System Estimation Method

Particle filtering is an emerging and powerful methodology for sequential signal processing with a wide range of applications in science and technology. Founded on the concept of sequential importance sampling and the use of Bayesian theory, particle filtering is particularly useful in dealing with difficult nonlinear and/or non-Gaussian problems. The underlying principle of the methodology is the approximation of relevant distributions with particles (samples from the space and unknowns) and their associated weights. Compared to classical Monte-Carlo methods, sequential importance sampling enables Particle Filtering to reduce the number of samples required to approximate the distributions with necessary precision, and makes it a faster and more computationally efficient approach than Monte-Carlo simulation. This is of particular benefit in diagnosis and prognosis of complex dynamic systems, such as actuators, engines, etc., because of the non-linear nature and ambiguity of the rotating machinery world when operating under fault conditions. Moreover, particle filtering allows information from multiple measurement sources to be fused in a principled manner [20].

Prognosis or long-term prediction of the failure evolution is based on both an accurate estimation of the current state and a model describing the fault progression. If the incipient failure is detected and isolated at the early stages of the fault initiation, it is reasonable to assume that sensor data will be available for a certain time window allowing for corrective measures to be taken. At the end of the observation window, the prediction outcome is passed on to the user (operator, maintainer) and additional adjustments are not feasible since corrective action must be taken to avoid a catastrophic event.

Prognosis is the Achilles' heel of fault diagnosis and failure prognosis systems. It can be understood as the generation of long-term predictions describing the evolution in time of a particular signal of interest or fault indicator [73]. Since prognosis projects the current condition of the indicator in the absence of future measurements, it necessarily entails large-grain uncertainty. This suggests a prognosis scheme based on recursive Bayesian estimation techniques, combining both the information from fault growth models and on-line data obtained from sensors monitoring key fault parameters (observations or features). A prognostic framework is proposed that takes advantage of a non-linear process (fault/degradation) model, a Bayesian estimation method using particle filtering and real-time measurements [21, 22].

Prognosis is achieved by performing two sequential steps, prediction and filtering. Prediction uses both the knowledge of the previous state estimate and the process model to generate the a-priori state pdf estimate for the next time instant,

$$p(x_{0:t}|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})dx_{0:t-1}. \tag{1}$$

The filtering step generates the posterior state pdf by using Bayes' formula,

$$p(x_{0:t}|y_{1:t-1}) = p(y_t|x_t)p(x_t|x_{0:t-1})\, p(x_{0:t-1}|y_{1:t-1}). \tag{2}$$

Equation (1) does not have an analytical solution in most cases. Instead, Sequential Monty Carlo (SMC) algorithms, or particle filters, are used to numerically solve (1) in real-time through the use of efficient sampling strategies. Particle filtering approximates the state pdf using samples of "particles" having associated discrete probability masses ("weights") as,

$$p(x_t|y_{1:t}) \approx \textstyle\sum_{i=1}^{N} \widetilde{w}_t\left(x_{0:t}^i\right) \cdot \delta\left(x_{0:t} - x_{0:t}^i\right), \tag{3}$$

where $x_{0:t}^i$ is the state trajectory and $y_{1:t}$ are the measurements up to time t. The simplest implementation of this algorithm, the Sequential Importance Resampling (SIR) particle filter, updates the weights with the likelihood of y_t,

$$w_t = w_{t-1} \cdot p(y_t : x_t). \tag{4}$$

By using the state equation to represent the evolution of the fault dimension in time, it is possible to generate a long-term prediction for the state pdf, in the absence of new measurements, in a recursive manner using the current pdf estimate for the state,

$$\tilde{p}\left(x_{t+p}|y_{1:t}\right) \approx \textstyle\sum_{i=1}^{N} \tilde{p}(x_t|y_{1:t}) \prod_{j=t+1}^{t+p} p\left(x_j|x_{j-1}\right) dx_{1:t+p-1}, \tag{5}$$

which can be approximated as,

$$\tilde{p}\left(x_{t+p}|y_{1:t}\right) \approx \textstyle\sum_{i=1}^{N} w_t^{(i)} \int \cdots \int p\left(x_{t+1:t+p-1}\right) \cdots \prod_{j=t+2}^{t+p} p\left(x_j|x_{j-1}\right) dx_{t+1:t+p-1} \tag{6}$$

The probability of failure at any future time instant is estimated by combining both the weights $w_{t+k}^{(i)}$ of predicted trajectories and specifications for the hazard zone through the application of the Law of Total Probabilities. The resulting RUL pdf, where t_{RUL} refers to RUL, provides the basis for the generation of confidence intervals and expectations for prognosis,

$$\hat{p}_{t_{RUL}} = \textstyle\sum_{i=1}^{N} p\left(Failure|X = \hat{x}_{t_{RUL}}^{(i)}, H_{lb}, H_{up}\right) \cdot w_{RUL}^{(i)}. \tag{7}$$

The boundaries of the hazard zone are design parameters related to the false-alarm rate (type I error). The upper and lower-boundaries of the hazard zone are denoted as H_{ub} and H_{lb}, accordingly. The probability of failure occurring at some time t is defined as,

$$p_{failure}(t) = p(H_{lb} \leq L(t) \leq H_{ub}). \tag{8}$$

Finally, it is often convenient to describe the minimum time-horizon corresponding to a failure with a particular level of certainty. This is formally referred to as the RUL, represented by the symbol t_{RUL}, and is defined as,

$$t_{RUL}(t_0) \triangleq \min(t^\star)\ \ s.t.\, p_{failure}(t^\star|t_0) \geq \beta, \tag{9}$$

where $t^{\star} \in (t_0, \infty)$ and $0 < \beta < 1$. The symbols t_0 and β refer to the initial prediction time and the confidence level (or type II error) associated with the prediction accordingly. Figure 3 illustrates the predicted fault growth of a system where a fault is detected at time t_{detect} and a prediction of the RUL is made at time $t_{prognosis}$. The probability of failure occurs outside the hazard-zone boundaries and is defined as the false-alarm rate, α. The time corresponding to each predicted fault trajectory in the hazard-zone is represented as a distribution on the time-axis. The maximum and minimum RUL values that encompass a confidence interval of value β are represented as t_{RUL}^{+} and t_{RUL}^{-}. Figure 2 shows a typical example of long-term prognosis.

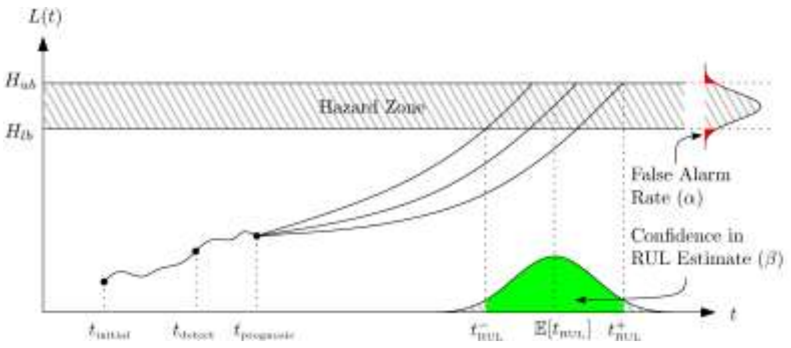


Figure 2. Illustration of long-term prediction.

A unified approach to state estimation/prediction and parameter estimation/identification for FDD using particle filtering was thoroughly studied in [22].

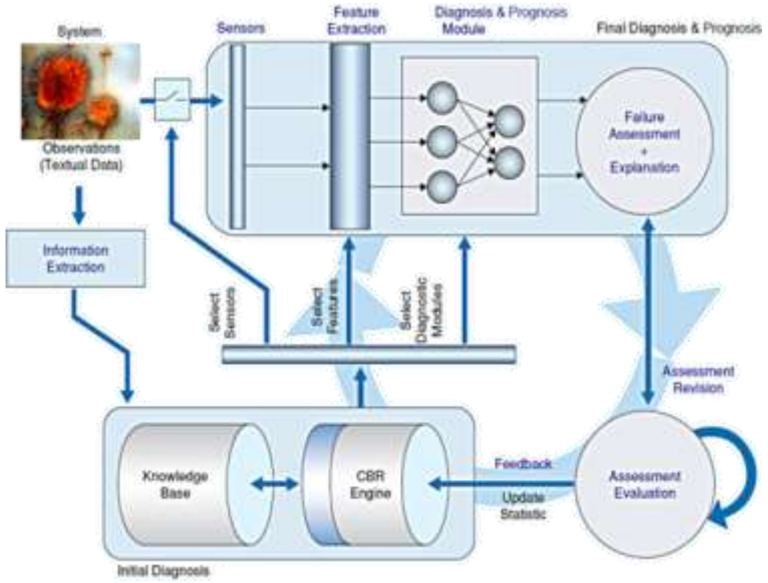


Figure 3. DCBR Architecture for integrated crack diagnosis, prognosis and maintenance.

3. The "Smart" Knowledge Base-A Paradigm in Reasoning

A reasoning paradigm called Dynamic Case based Reasoning (DCBR) is adapted as the "smart" knowledge base residing off-board and functioning as the central repository of cases and algorithms to ascertain that the reconfiguration strategies are executed effectively, robustly and efficiently.

We exploit this novel construct as the knowledge base where cases from historical evidence are stored and new cases are compared with stored ones to determine the current system state. Case-Base Reasoning (CBR) constitutes the main system level reasoning paradigm of the reconfiguration architecture and incorporates essential elements of a learning strategy. Cognitive aspects of learning and adaptation are integral and essential elements of the DCBR architecture. CBR was founded on the belief that human memory is episodic in nature. This episodic memory, which comprises human knowledge, is accumulated from past experience. Each memory episode is contributed by a single past situation or event. Faced with a new problem, a human often relates the problem to one or more memory episodes and composes a solution from these episodes. CBR is a computer program to simulate this human recognition process and has been applied to a variety of process operation support systems. The CBR application domain usually involves problem solving, i.e., identify similar cases for better understanding, assessing and/or comparing with the current situation. The proposed Dynamic Case-Based Reasoning (DCBR), module is shown in Figure 5.

The step of the selection of the cases in the case base on which reasoning will be based is essential, because it is here that the notion of similarity between the cases in memory and the new episodic evidence is applied. Let *Ent_e* be a new case presented to the system. A similarity by proximity notion may be calculated by the following scoring function:

$$sim(Ent_e, Ent_j) = \frac{\sum_{k=1}^{n} \alpha \times sim\left(El_{i,k}, El_{l,k}\right) + \sum_{k=1}^{n} n_{k_i,pred} \times n_{i,pert} \times sim\left(El_{i,k}, El_{l,k}\right)}{\alpha \times n + \sum_{k=1}^{n} n_{k_i,pred} \times n_{i,pert}}$$

where Ent_j are cases previously presented, El_i is a feature or an (attribute, value) pair, $n_{i,pert}$ is a pertinence weighted variable associated with the description element El_i, $n_{i,pred}$ is a predictive weighted variable associated with each case in memory; this weight is increased as the corresponding element (feature) is favorably selecting a case, and is decreased as this selection leads to a failure.

Incremental learning occurs whenever a new case is processed and its results are identified. Thus, the memory keeps track of each of its experiences, whether success or failure, in a declarative way; it is then ready to take advantage of future experiences.

Incremental learning is pursued using Q-Learning, a popular reinforcement learning scheme for agents learning to behave in a game-like environment. Q-Learning is highly adaptive for on-line learning since it can easily incorporate new data as part of its stored database. An attractive feature in a game-like situation is that the player is learning to choose the best action for each particular game setting. In this framework, the expected reward or “cost-to-go” is stated as[1]:

$$\overset{\substack{\text{Expected reward:}\\\text{"cost-to-go" function}}}{Q(s,a)} \leftarrow Q(\underset{\text{Current state}}{s},\underset{\text{Current action}}{a}) + \overset{\text{Learning}}{\alpha}[\overset{\text{Immediate}}{r} + \overset{\text{Discount}}{\gamma}\max_{a'} Q(\underset{\text{Next state}}{s'},\underset{\text{Next action}}{a'}) - Q(s,a)]$$

The potential impact of reconfiguration strategies on the asset’s integrity and performance may be assessed now exploiting the capabilities provided by DCBR and accompanied learning and adaptation algorithms.

4. Resilient Design of Unmanned Autonomous Systems

The term “resilience” has slightly different definitions in many areas such as engineering, ecology, social sciences, etc., but the key idea of resilience is the ability to maintain an acceptable level of functionality in the face of failures and challenges.

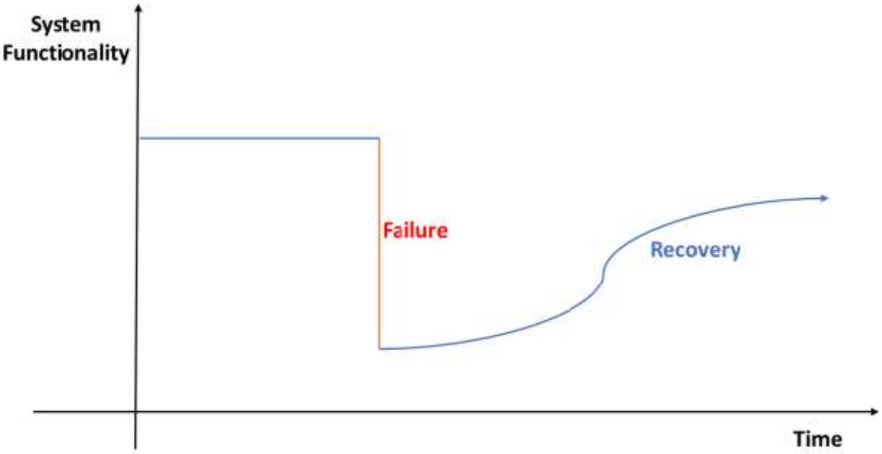


Figure 4. Resilience as in recovery of system functionality.

[1] Watkins, C and Dayan, P., “Q-Learning,” Machine Learning Vol 8(3/4) 1992.

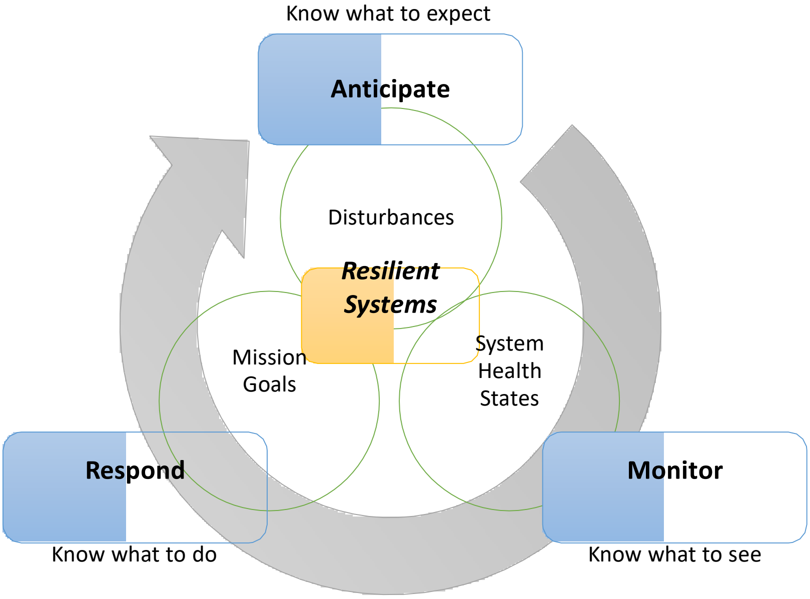


Figure 5. Fundamental functionality of resilient systems [13, 36].

A visualization of maintaining functionality and recovering is shown in Figure 4. Tradeoff is inevitable in control system operation under failure modes, where all the limited resources or functionalities need to be considered. For example, if the main limitation of a system is to operate with certain amount of fuel, the system may maintain its movement even under severe disturbances using a new control method, but it may not be possible to reach the goal position. Therefore, it can be deduced that the knowledge of the respective importance of different aspects in system operation is paramount. Figure 5 depicts the fundamental functionality of resilient systems.

4.1. Definition: Resilient Systems

1) A system that can adjust its functions prior to or following changes and disturbances so that it can go on working even after a major mishap or in the presence of continuous stress, mainly by being able to be proactive on safety [24]
2) A system that is capable of deploying tactical changes, while supported by its built-in robustness, in order to avoid a given set of threats, or restore its mission capability and health levels, if degraded

Based on the definitions, mission capability is significant for resilient systems. Mission capability was defined as [25]:

Definition: Mission Capability

– A measure of the results of the mission; given the condition of the system during the mission.

System resilience is one of the system characteristics, which contributes to system safety, robustness, and survivability [24]. In this notion of system resilience, failure means the failure of appropriate adaptation to operational disturbances and unexpected events with finite resources and reaction time [36]. Success, likewise, means successful adaptation to the risks to avoid possible dangerous outcomes.

The concept of resilience engineering can be considered as an advanced version of traditional safety and survivability engineering disciplines. Particularly, Hollnagel described challenges that resilience engineering should be able to tackle [24]:

1) Performance conditions are always underspecified.
2) Adverse events can be attributed to an unexpected combination of normal performance variability.
3) Safety management cannot be based on hindsight nor solely rely on error tabulation and failure probability calculations.

A significant difference between traditional safety management approaches and resilience engineering is whether a system can proactively reduce system susceptibility or not [24]. The term, proactive, is an essential feature because it means a system can actively remove dangerous environment and prevent unwanted events during the operations. This feature is a new vision of survivability engineering in that traditional safety management approaches have been reactive or passive, and they also focused on vulnerability reduction. With the consideration of proactive characteristics, Hollnagel proposed basic functionality that a resilient system has to be able to: anticipate, monitor, and respond [24]. Figure 5 describes the three basic functions of a resilient system and their relationships.

4.2. Resilient Design

Modern systems, such as unmanned autonomous systems, contain an ever-increasing number of components and subsystems that interact with each other in often unpredictable ways. Unintended interactions lead to unexpected behaviors and consequences, some of which have proven to be catastrophic. A key technical challenge in developing such complex systems is to predict catastrophic interactions earlier during development, and to also dynamically perform this assessment, as these systems evolve over time. *Resilience* is a key driver in the design of systems that must operate in uncertain and unstructured environments, and forms a key metric in assessing the capacity for systems to perform within the specified performance envelope despite disturbances to their operating environment. Modern autonomous systems typically operate in closed loop feedback control configurations, which are hierarchical and are designed to integrate hardware and software architectures. *Design for Resilience* involves the system itself as well as the control routines and software support building upon concepts of *self-organization* and *control reconfiguration*, or *fault tolerance* accompanied by appropriate performance metrics. The ultimate goal is to establish a "resilient-by-construction" design principle which requires quantifying resilience properties prior to prototyping and subsequent testing. Software resilience and reliability has been also attracting attention, as well as coupling effects between software and hardware system components. We are addressing in this paper the hardware issue only in order to deliver a rigorous and integrated framework to resilience of critical autonomous systems.

Resilient indices are defined exploiting graph notions and computing the *percentage* of node combinations that guarantees connectivity. *Confidence (%)* is a measure of resilience, i.e., how reliable statistical results are to malicious attacks or extreme disturbances, and indicates the probability of a decision system outcome (fault/malicious event declaration, control effectiveness, prognostic horizon, etc.) being correct. Entropic concepts (Shannon information entropy), error metrics, control theoretic notions of observability, controllability and stability, could also constitute source performance data for resilience analysis. Performance data are derived from system operations recordings, or from simulation models. The foundational concepts for a self-organization strategy introduced in this paper borrow from the emerging field of Complex Adaptive Systems (CAS). Self-organization is viewed as the CAS enabler. We introduce at this stage the CAS theory constituting the foundation of self-organization.

4.3. The Modeling Framework

4.3.1. Graph-Based Approaches

We take advantage of a rich array of modeling tools and methods representing the physical connections and dependencies of such complex autonomous systems. We pursue in parallel Markov modeling-a probabilistic approach to represent complex systems, their states and state transitions. The modeling toolset includes also structural and functional representations and dynamical system models that integrate disturbance factors into their structure.

4.3.2. Structural and Functional Modeling

The structural and functional models, in combination with analytic approaches and taking into account physical constraints, are employed to derive the reconfiguration states. Markov chain modeling approach is adopted to model these effects of reconfiguration on the components of the system. Concepts of “Relevant connections, Non-relevant connections and Escalated states” are used in the framework of Markov chain modeling to capture the effects of reconfiguration.

4.4. Disturbance Factor Analysis

In the context of a cyber physical system’s life degradation, we introduce a generalized heuristic modeling approach with consideration of critical disturbance/stress factors. We establish appropriate life-cycle models that encapsulate cost/benefit of the envisioned design solution, i.e., the models need to assess behavior and consequences during degraded (or faulted) state; models should also be able to assess mitigation actions that are part of a health management approach.

Disturbance factor analysis considers the impact of varied operational and environmental disturbance factors on system end-of-life (EOL). Generally speaking, a degrading system can be expressed as

$$L(t+1) = f(L(t), U,) + \omega_1(t+1)$$
$$U = g(\sigma, t) + \omega_2$$
$$y(t+1) = h\big(L(t)\big) + \omega_3(t+1),$$

where $L(t)$ denotes a time-varied life state, for example, a life degradation or loss condition; $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ is a series of environmental and operational disturbance factors that affect the time evolution of system degradation, such as temperature, shocks, vibrations, load profiles, and fault/failure modes, among others; U is the disturbance factor severity function; y represents system output; $\omega_{1,2,3}$ are noises. In real applications, the degradation model f(·) usually assumes a nonlinear function, sometimes has non-monotonic attributes (for example, recovery effects), corrupted by non-Gaussian noise, and typically is appropriately represented by probability density functions (pdfs) to indicate the underlying uncertainty. Appropriate life-cycle models need to assess behavior and consequences during degraded (or faulted) state and should also be able to assess mitigation actions that are part of a health management approach.

4.4.1. Dynamical System Models

We take advantage of extensive coverage for dynamical complex system modeling presented over the past years by several investigators. We will modify and extend these approaches as necessary.

5. Complex Adaptive Systems: A Rigorous Framework for Self-Organization and Control Reconfiguration of Complex Systems for Improved Resilience, Safety and Reliability

"I think that the next century (21st) will be the century of complexity"
- Stephen Hawking

Much of the work in Complexity Theory has developed around the natural and social sciences, as a method for understanding complex systems in the physical and human realm. Typical examples include the human brain, the heart, a flock of birds, an ant colony, among many others. In the engineering domain, examples of unmanned autonomous systems, swarms of autonomous systems, "smart" manufacturing processes, abound and are characterized as "systems of systems" exhibiting characteristic behaviors of complexity. We seek the theoretical underpinnings for the resilient design and operation of unmanned systems and explore methods/tools based on Complex Adaptive Systems to address military organizational problems. Complex Adaptive Systems, as a school of thought, took hold in the mid-1980's with the formation of the Santa Fe Institute, a New Mexico think tank formed in part by former members of the nearby Los Alamos National Laboratory. Examples of CAS considered by the think tank are widespread in both the natural and human world like brains, immune systems, ecologies, cells, developing embryos, and ant colonies. In the human world, political parties, scientific communities and the economy are examples. A fundamental aspect of complex adaptive systems is that they allow local behavior to generate global characteristics that then alter the way agents interact (Burkhart, 1996). Actions not only proceed along feedback loops; they can also change these loops. The focus of the Santa Fe approach has been upon Complex Adaptive Systems, in which complex and patterned output arises from simple, fundamental principles, but requires many actors/agents and multiple interactions over time to produce the emergent complexity. In view of our autonomous system, it is dynamic and nonlinear with system states evolving from time t to t+1 in mostly unknown ways. The research objective is to determine a policy to maintain acceptable performance (mission completion, stability) even in the presence of extreme disturbances (component failures). CAS models require us to specify how the behavior of an agent/component at time t influences the behavior at time t (or time t + 1 if there is a lag) of others with whom the agent has ties. Complexity Theory and Complex Adaptive Systems has begun to develop an understanding of physical and social systems that is an alternative to a more linear and reductionist mode of thinking. Much (if not most) of this advancement has been due to the extensive use of computer-based modeling, such as agent-based simulations, which has expanded the set of tools used to explore complex system behavior.

Modeling of CAS captures the critical aspects of the system. Figure 6 depicts a conceptual schematic of a CAS configuration [25]. Note the local interaction of components/agents generating emergent global structure and behaviors. Engineering applications of CAS concepts are mostly absent from the research literature. They refer to broad CAS notions suggesting how they may be applied to complex systems. Without specific reference to CAS, a flurry of activity on control reconfiguration technologies, a CAS enabler, has been reported over the past years. There is a rich literature describing proper adjustments to control actions that assure resilient behaviors. Fault Tolerant Control Systems (FTCS), motivated by commercial aircraft accidents [26] have been researched extensively. Clements [34] developed a hierarchical control architecture showing the interconnections among fault detection & identification, set-point controller, control redistribution, control gain adaptation, and component restructuring. Ge, Kacprzynski, Roemer, and Vachtsevanos [27] introduced a higher-level adaptive system framework using an Automated Contingency Management (ACM) concept. Drozeski, Saha, and Vachtsevanos [28] proposed a three-tier hierarchical control scheme as Active FTCS. Tang, Kacprzynski, Goebel, Saxena, Saha, and Vachtsevanos [29] extended the ACM framework by integrating it with a prognostics module. Brown, Georgoulas, Bole, Pei, Orchard, Tang, Saha, Saxena, Goebel, and Vachtsevanos [30] proposed prognostics enhanced low-level reconfigurable control for an avionics component. Bole, Tang, Goebel, and Vachtsevanos [31] described a fault adaptive control architecture, and Bole [32, 33] addressed uncertainties in prognostics and reconfigurable control allocation strategies. It is hoped that the proposed project will motivate further research in this emerging field.

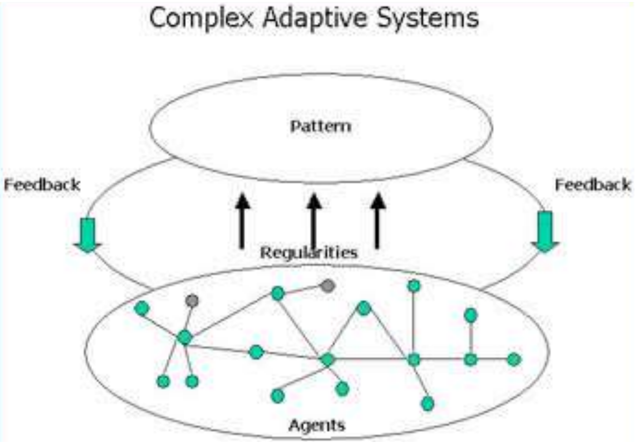


Figure 6. Conceptual Representation of CAS Structure (From Jon Wade1 and Babak Heydari1, Complexity: Definition and Reduction Techniques Some Simple Thoughts on Complex Systems Jon Wade1 and Babak Heydari1.

Pursuing CAS principles, we define a hierarchical and decomposable system structure and partition it so that that resilience of the system as a whole is not reliant on any one sub-element. We view complex systems as forming a core-periphery network exhibiting attributes of robustness to random disturbances at a systemic level. Given a complex system (our unmanned autonomous system or a group of soldiers), the design task is to self-organize its components/agents, reconfigure its controls, to meet stated operational requirements or mission profiles even in the presence of extreme disturbances (a broken leg, for example, in our hexapod testbed, a severe fault in the hovercraft case, presence of adversaries in the battlefield). The enablers include granular data to capture the interactions and conditional distributions within and between complex sub-systems, modeling tools, and policy design methods- self-organization or control reconfiguration strategies, testing/validation for decomposable systems.

We introduce a thorough and proactive methodology to resilient design and operation of unmanned autonomous systems for improved safety and risk assurance/management that entails the following major steps:

1) Threat/hazard characterization - detection, identification, prediction of hazard/fault/failure evolution; performance metrics are specified and used in the development and validation process.
2) Self-organization strategy for systems subjected to severe disturbances (internal or external) with performance metrics/stability conditions.
3) Reconfigurable control strategy for systems subjected to severe fault modes.

5.1. Disturbance/Hazard/Threat Analysis

Disturbance/hazard analysis considers the impact of varied operational and environmental disturbance factors on system (force, military organization, unmanned autonomous systems) safety and survivability. The CAS-based organizational system design, as well as the resilient design of hardware/software assets, may not be effective without a thorough threat analysis in a real-life application or mission. The system design must imply that it contains the additional capacity of absorbing the effects of a threat, through either safety margins, system survivability, resilience or robustness. It is a fundamental objective of the proposed research is the design and operation of complex military human or hardware assets to ascertain that they respond and accommodate severe threats/disturbances/disruptions in the battlefield environment.

Hardware, software, the environment, and human factors are major sources of hazards. For the systems (organizational change, unmanned autonomous systems) under consideration, we seek historical hazard data and categorize them as to their severity, frequency of occurrence, and testability. It is, of course, true that "you can only manage what you can measure" and data/information regarding hazards and their potential impact on system safety are absolute requirements to modeling, representation and control of hazards and safety margins. We pursue multiple modeling methods to represent threats/hazards. Dynamic system models, behavioral representations, neuro-fuzzy constructs, expert systems, among others, will be considered not only to represent them but also to detect, identify them and predict their future evolution. We build upon our pioneering work on integrity management/Prognostics and Health Management (PHM) to derive a suite of algorithms seeking to achieve the threat analysis objectives [35].

5.2. CAS in Unmanned Autonomous Systems

Complex Adaptive Systems theory presents major challenges but also significant opportunities that go beyond the human and socio-economic domains. Engineering systems are becoming more complex taxing the designer's abilities to endow them with attributes of resilience, robustness and reliability. As a protopypical example, unmanned autonomous systems are proliferating at an exponential rate while a large percentage of Class A air mishaps are attributed to UAVs. UAVs place significant demands to human (sensor, pilot) operators who are required to make informed decisions in (almost) hard real-time, and require significant computational resources for data/information processing. It was suggested by an Autonomous Vehicle Operator (AVO) that, at times, "he's been more overcome by the torrent of information pouring in during a drone flight than he was in the cockpit." Currently, limitations in autonomy lead to operator work time exceeding the time of unmanned system deployment and gains in the field of autonomy are required to reverse the current trend. The final report of the Defense Science Board Summer Study on Autonomy, June 2016, provides recommendations for "accelerating DoD's adoption of autonomous capabilities" [36]. Unmanned robotic platforms (UAVs, UUVs, UGVs) operating as a single vehicle or in swarm formation, are complex Cyber Physical Ssystems(CPSs). System resilience and improved autonomy are bound to benefit substantially from. the utility of CAS concepts in the design and operation of complex engineering systems. The application of similar principles to various CPSs – "smart manufacturing processes, transportation systems, aerospace and space assets, among many others – promises to bring about a new design philosophy for complex systems viewing complexity and its attributes as the driving mechanism.

5.3. Innovative Features of the Research and Development

The theoretical underpinnings of the R&D effort introduced in this chapter build on a developing body of knowledge in complex systems called Complex Adaptive Systems

(CAS). CAS provides insight regarding the system’s dynamics and structural characteristics. Current thinking is that CAS concepts range from those derived through chaos theory and agent-based modeling to self-organization and control reconfiguration of system components (agents) that interact with each other in unpredictable ways, co-adapt and exhibit “emergent” behaviors. They are adaptive in that the individual and collective behavior self-organizes and learns in the presence of extreme disturbances increasing their survivability. The emerging CAS field has sought to apply "thought" processes to human and socio-economic processes/systems. The engineering disciplines have seen minimum exploitation of the fundamental CAS enablers. Specifically, in deference to the core CAS and associated complexity topics, researchers in science and engineering have reported extensively results on fault tolerant or reconfigurable control technologies applied to a variety of complex systems subjected to incipient failure modes (not complete failure events). Recent publications explored self-organization strategies using search techniques or trial and error methods incurring significant computational burden thus rendering such approaches unable to address in real time dynamic IPS. The proposed framework for resilient design and operation of IPS builds on CAS concepts and its enabling technologies avoiding these difficulties through the introduction of rigorous self-organization and control reconfiguration strategies that may be implemented on-platform in almost real time. We begin by analyzing threats/disturbances to IPS that go beyond the incipient failure variety and may entail severe structural anomalies like component/agent failure modes. Self-organization relies on graph spectral and epidemic spreading models to represent the system behaviors under normal and faulty conditions, and uses a Markov Decision Process as the basic self-organization enabler. The connectivity of a graph is an important measure of its resilience, as it indicates how much more node/edge disconnections can be tolerated until the graph is fully disconnected. The goal for a robust, reliable and resilient system under fault/failure effects would be to reorganize the graph and maximize the connectivity while observing the system constraints. Spectral graph theoretic notions and a Markov Decision Process (MDP) are designed to observe the overall system behavior. The solution of the MDP is in the form of a “policy,” and the optimal policy for a resilient autonomous system operation is obtained by a dynamic programming method, namely the Bellman equation. The optimal policy from MDP and resilience metrics are combined to ensure that the system maintains its mission profile under severe disturbances. Control reconfiguration, another CAS enabler, is configured as a three-tiered architecture. The first one is a Dynamic Case Based Reasoning (DCBR) module that detects and identifies a system faulty condition during its operation, and proposes a reconfigurable control strategy to the second, a Reinforcement Learning (RL) module. The latter determines the level of control reconfiguration and updates it online considering the fault consequences. Finally, the Model Predictive Control-Differential Dynamic Programming (MPC-DDP) module redistributes the control authority in order to maintain acceptable stability bounds while performing a short-term performance trade-off. Experimental testing using a novel hovercraft platform has shown promising results. Both self-organization and control reconfiguration technologies are accompanied by appropriate performance metrics.

5.4. Spontaneous Order and Self-Organization

When emergence describes the appearance of unplanned order, it is spontaneous order (in the social sciences) or self-organization (in physical sciences). In the unmanned autonomous systems case, self-organization is viewed as a *resilience* enabler where a complex UAS or a swarm of UASs are subjected to severe disturbances (external or internal) and must self-organize its components/parts to accommodate the disturbance.

6. Self-Organization: An Overview

Figure 7 depicts a schematic of the framework for self-organization of a complex system subjected to severe disturbances, namely failure modes. Initially, a system-level mission profile is suggested, where the system is commanded to travel a prescribed path in 2D or 3D spaces. For traditional disturbance rejection, the system will operate under a control law, which employs local controllers for feedback compensation. In the presence of a failure mode, the mission is unachievable, and the proposed self-organizational method takes over as the higher-level control architecture.

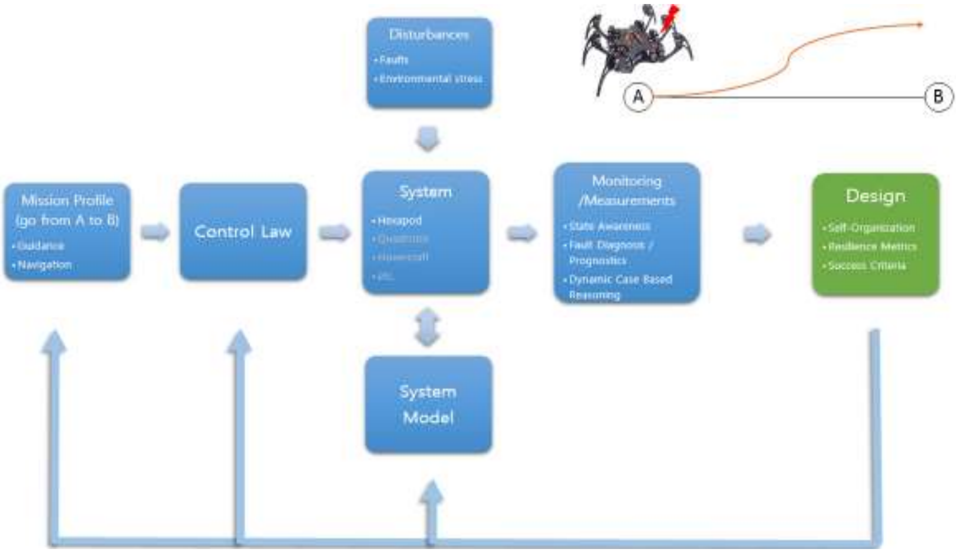


Figure 7. Overall self-organization methodology.

The self-organization strategy begins with two types of modeling techniques aimed to describe the system behaviors in data format: a graph theoretic model to examine the current state of the system, and a dynamic/kinematic model to predict the optimal next step. Along with the data obtained from the models, monitoring processes such as situation awareness and fault diagnosis/prognosis will examine the current health state of the system, and the optimal action will be determined based on a fully implemented MDP. The MDP output will be the optimal action for the system to proceed to the next step of the mission profile while maintaining an acceptable level of stability and observing physical/functional constraints. The system model and control law will be updated based on the evaluation of the outcome of the process through appropriate success criteria.

6.1. The Modeling Framework

We pursue a Markov modeling approach - a probabilistic approach to represent complex systems, their states and state transitions. We formulate an Epidemic Spreading Model to estimate a probabilistic measure of system immunity and recovery time (i.e., self-healing). In the epidemic spreading model, disturbances are cascaded within the system model, and system components take on one of three states: susceptible, failed, or fixed (SFF model). The model is probabilistic due to the uncertainties (e.g., model uncertainty, state transitions), providing probabilistic measures of system immunity as well as recovery time (i.e., self-healing). The modeling toolset includes structural and functional representations and dynamical system models that integrate disturbance factors into their structure.

Within the epidemic spreading framework, we introduce a graph-based self-organization method by considering two graphs: the physical graph and the behavioral/spatial graph. The physical graph depicts the actual structure and interconnectivity of the system, while the spatial graph contains grids of positional states in the Cartesian space. Assuming discrete time, the underlying assumption in using the spatial graph is that state transitions can only take place between adjacent (neighboring) grid points, where the system either moves by one or zero state at each time step. The optimal path to the goal point can be obtained by defining a value function so that each state contains a “value” that will be accumulated along the way. Similar to the state transition probabilities, the values will be assigned so that a higher value is given to the state that leads to optimal system behavior. The goal is to find the path that maximizes the cumulative value. The solution to the maximum value problem is pursued via the Bellman equation.

The self-organization strategy introduced in this document follows the CAS framework in terms of the modeling and control activities. *Connectivity* describes the ways in which the agents/components in a system connect and relate to one another is critical to the survival of the system. It is from these connections that the patterns are formed and the feedback disseminated.

Generic aspects of the approach may find numerous applications within the Cyber Physical Systems (CPS) domain of engineering processes.

We introduce a novel behavior-based approach to self-organization of unmanned systems. The approach proceeds along the following steps: defining relevant system parameters, the vector and covariance matrix leading to a determination of the principal components affecting the system behavior. Behaviors are defined with appropriate kernels of the parameters and the self-organization strategy aims to reorganize the parameters via constructing goal parameters and minimizing the mean square error between the goal and current parameters.

The modeling framework and self-organization strategies are tested on a typical ground unmanned autonomous system – a hexapod – instrumented appropriately and exhibiting structural attributes amenable to the self-organization methodology introduced in this document. Simulation results obtained through the injection of failure modes (broken leg) and the application of the self-organization strategy demonstrate the efficacy of the approach. It is clearly shown that the scheme is applicable on-platform and performs in real time compared with trial and error or search techniques.

6.2. A Self-Organization Strategy for Unmanned Autonomous Systems

The field of self-organization seeks general rules about the growth and evolution of systemic structure, the forms it might take, and finally methods that predict the future organization that will result from changes made to the underlying components under the influence of severe disturbances. Self-organization implies that a system aligns itself to a problem and is self-sustaining, even when the environment changes under severe disturbance conditions. The self-organization strategy begins with two types of modeling techniques aimed to describe the system behaviors in data format: a graph theoretic model to examine the current state of the system, and a dynamic/kinematic model to predict the optimal next step. Along with the data obtained from the models, monitoring processes such as situation awareness and fault diagnosis/prognosis will examine the current health state of the system, and the optimal action will be determined based on a fully implemented MDP. The MDP output will be the optimal action for the system to proceed to the next step of the mission profile while maintaining an acceptable level of stability and observing physical/functional constraints. The system model and control law will be updated based on the evaluation of the outcome of the process through appropriate success criteria.

Complex systems are constructed from multiple subsystems and components with each serving incremental tasks, where the “emergent” system behavior cannot be deduced from the behaviors of the individual parts. The key requirement of complex systems is the ability to compensate for unforeseen and extreme disturbances, so it is important to design a control method that ensures acceptable level of system resilience throughout its operation. Therefore, detailed and accurate knowledge of system behaviors is paramount for the design of complex system control strategies. This paper presents a self-organizing control strategy that incorporates both situational awareness and failure impact compensation for a resilient unmanned autonomous system. Furthermore, complex systems are “systems of systems” comprised of hierarchical sets of subsystems or components, where the combined simultaneous operation of many components can lead to unforeseen “emergent” behaviors. Vinerbi et al. [37] suggest that even though good knowledge of system behaviors is significant, “full” knowledge of complex system behavior may not be achievable. Also, complex systems are vulnerable to multiple failures at once, while the effect of individual failures may not be evident. Complexity Theory has shown to improve understanding of system behavioral modes and provide a viable means for modeling of such complex systems. This paper postpones discussion of Complexity Theory attributes to a future document and focuses on the resilient design via self-organization.

Unexpected change in complex system behavior occurs in case of extreme disturbances, such as a complete failure of a subsystem. The most commonly used robust control technique for disturbance rejection is PID (feedback) control. A major limitation of traditional control systems using PID is the lack of online adjustments to changing system properties. Manual computation and adjustment to new system behaviors are impractical for complex system applications such as unmanned autonomous systems (UAS).

A self-organizational method could be an alternative to the traditional robust control avoiding a heavy computational burden. A system is considered *organized* if it has certain structure and functionality, and self-organization implies that the organization of the system occurs internally, without any external or centralized control unit [38]. In the simplest case, a self-organization strategy consists of two components: response and adaptation, responding to the system’s functionality. Therefore, systems with the same structure may require a different adaptation strategy depending on the system’s operational objectives. Along with a reduced computational burden due to the targeted operation, a self-organization method provides the benefit of random noise adaptation, since the process is spontaneous with intrinsic update rules [39].

In this contribution, a novel self-organizational method is introduced as a compensatory measure to maintain system functionality under the presence of failure modes. It is noted that resilience requirements refer to severe disturbances, i.e., failure modes compared to usual disturbances compensated by conventional technologies such as robust or PID control. A typical unmanned autonomous ground vehicle – the hexapod – is employed as the testbed for the development and validation of the self-organizing strategy. Methods to understand system behavior include data acquisition, system modeling, and proper construction of performance metrics; the strategy includes a policy to address the changing system conditions and success criteria to evaluate the optimal action. The physical, functional, nonlinear dynamic, and graph theoretic models will be considered to examine system behaviors under both normal and faulty conditions. Then, the self-organization strategy will be introduced in the form of a Markov Decision Process (MDP) with dynamic programming for optimal performance. Finally, the success criteria for the control method will be constructed with Lyapunov stability conditions so that the self-organization strategy can be modified throughout the system operation for system resilience regarding stability and resource limitations. Simulation results will be presented at the end to demonstrate the efficacy of the approach. The first one relies on graph spectral and epidemic spreading models to represent the system behaviors under normal and faulty/threat conditions, and uses a Markov Decision Process as the basic self-organization module. The second builds on a behavioral model of the system and self-organizes exploiting a reactive control scheme. For the first approach, decision-making is based on the current state only. The connectivity of a graph is an important measure of its resilience, as it indicates how much more node/edge disconnections can be tolerated until the graph is fully disconnected. Hence, the goal for a robust, reliable and resilient system under fault impacts would be to reorganize the graph and maximize the connectivity while observing the system constraints. A graph can be mathematically represented with the Laplacian matrix, defined as L = D – A, where L is the Laplacian matrix, D is the degree matrix (diagonal matrix showing the number of edges at each node), and A is the adjacency matrix (square matrix indicating the connection between nodes with 1s and 0s). The connectivity of a graph can be algebraically obtained by taking the second-smallest eigenvalue of the Laplacian matrix. The occurrence of severe disturbances is represented with disconnections of corresponding nodes/edges, resulting in greatly reduced algebraic connectivity of the graph. To begin looking into the behavior of how a fault epidemic affects and spreads through a system, the transition matrix is defined as a square matrix with elements indicating the probability of traveling from node i to node j. The probabilities can be obtained through the derivation of a random walk normalized Laplacian, which can be written as $D^{-1}A$. This matrix serves as the transition matrix of a random walker on the graph, containing the likelihood of the epidemic spreading direction. A Markov Decision Process (MDP) can be constructed to observe the overall system behavior. MDP is a tuple consisting of {S, A, T, R, γ}, where:

- S = set of system states
- A = set of state-transitioning actions
- T = state transition matrix

- R(s,a) = reward for taking action a at state s
- γ = discount factor (to be further explained below)

In the MDP construction, the reward function gives scalar values for each state transition, with greater value awarded to state transitions that result in moving toward the ideal behavior. A good definition of the reward function is the key to designing a resilient system, as the reward values can be constantly updated to optimize system resilience. The solution of the MDP will be in a form of a "policy," denoted by π and is the mapping from S to A, such that the system operation will proceed by repeating two steps: determine current state and execute action $\pi(s) = a.$ Note here that the action is determined by only the current state and not the history of previous states, since the process is Markovian. To determine the policy, define a value function V(s) that accumulates the immediate rewards from each state along a series of state transitions. The optimal policy for a resilient autonomous system operation can be obtained by a dynamic programming method, namely the Bellman equation, written as $V(s) = \max(R(s,a) + \gamma \sum_{s'} T(s'|s,a)V(s'))$. Here, the discount factor $\gamma \in [0,1]$ suppresses the effect of future iterations to ensure convergence to a solution. The resulting system behavior following the obtained policy can be evaluated with proper resilience metrics. Due to limited resources (time, energy, etc.) or functional capabilities (joint angles, motor speed, etc.), tradeoff among the resources is inevitable in establishing system resilience. Some conditions for resilience metrics can be considered so that the metrics must be useful for decision making; should result in values so that the performance can be quantitatively assessed and compared; reflect uncertainty of the result; and consider failure recovery time. Overall, the optimal policy from MDP and the resilience metrics can be combined to ensure that the system maintains its mission profile under severe disturbances. These algorithmic developments were tested successfully using a hexapod as the test case.

We introduce a novel behavior-based approach to self-organization of unmanned systems. The approach proceeds along the following steps: defining relevant system parameters, the vector and covariance matrix leading to a determination of the principal components affecting the system behavior. Behaviors are defined with appropriate kernels of the parameters and the self-organization strategy aims to reorganize the parameters via constructing goal parameters and minimizing the mean square error between the goal and current parameters.

The modeling framework and self-organization strategies are tested on a typical ground unmanned autonomous system – a hexapod – instrumented appropriately and exhibiting structural attributes amenable to the self-organization methodology introduced in this document. Simulation results obtained through the injection of failure modes (broken leg) and the application of the self-organization strategy demonstrate the efficacy of the approach. It is clearly shown that the scheme is applicable on-platform and performs in real time compared with trial and error or search techniques.

6.3. Spectral Graph Theory

A graph G is a set $G(V, E)$ that consists of vertices and edges (connections between vertices). In systems engineering, the structure of a complex system can be represented as a graph with the list of system nodes and their respective interconnectivity.

Wilson [40] (1996) lists several useful matrix representations in graph theory. The adjacency matrix A is a $n \times n$ matrix where the element a_{ij} indicates connectivity from node i to node j with 1, and 0, otherwise. The degree matrix D is a $n \times n$ diagonal matrix, where the diagonal elements d_i are the degree (number of edges) of node i.

The Laplacian matrix L is defined as in Equation (1) and is also called the system matrix.

$$L = D - A \tag{10}$$

The second smallest eigenvalue (denoted by λ_2) of the Laplacian matrix is called the algebraic connectivity of the graph and represents how well the graph is connected. The algebraic connectivity is also an important indication of a network's resilience, and the quantification of the importance of a node or a link with the effect of node removal on the algebraic connectivity is studied in the work of Liu et al. [41].

Another useful matrix in spectral graph theory is the transition matrix T, in which the element p_{ij} is the probability of transitioning from node i to node j. The aforementioned adjacency and degree matrices can be combined as in Equation (11) to determine the transition matrix within a system graph [42} (Butler, 2008):

$$T = D^{-1}A \tag{11}$$

The transition matrix is a versatile tool to represent probabilistic processes. In addition to storing the probabilities within nodes of a system, the transition matrix can also represent state-transition probabilities.

6.4. Markov Decision Process

Proper definition of system states and the state transition probabilities is the first step for constructing a Markov Decision Process (MDP), formulated by Bellman [43], and applying the Markovian property to move from the current state to the next optimal state based on a predefined policy. MDP is a dynamic programming method, where the control problem for a complex system is divided into simpler sub problems in a memory-based structure so that the solution for the next occurring sub problem can be looked up immediately instead of re-composing the solution, thereby reducing significantly the computational burden compared to traditional control methods.

MDP is a set of decision-making rules consisting of {S,A,T,R}, where S is the finite set of achievable states, A is the finite set of actions that connect a state to other states, T is the transition matrix that stores the likelihoods of the state transitions, and R is the reward matrix that indicates the immediate effect of an action applied to a state. In other words, the transition matrix can be used to describe the system's behavior, and the reward matrix can be used to guide the control action towards mission completion. Yukalov and Sornette [44] suggest that any complex system, under given conditions, is more inclined to occupy the most stable state. From the work of Gabbai [45], self-organization is an evolving process towards a state of equilibrium, commonly called an attractor. An attractor could provide a lower dimensional representation of complex system dynamics, and an example attractor could be a desired path for the system to follow. Therefore, the transition and reward matrices should be constructed with higher probability and reward assigned for approaching the desired path.

6.5. Dynamic Programming

The main goal of MDP is to construct a "policy" π(s) that provides the optimal available action for each state. The algorithm to obtain the policy is represented by Equation (12), known as the Bellman equation. Value functions, V(s), are defined for each state to accumulate the immediate reward from an action at each time step, until the overall reward converges to steady values.

$$V(s) = R(s,a) + \gamma \sum_{s'} T(s'|s,a)V(s') \tag{12}$$

The discount factor $\gamma \in [0,1]$ is used to suppress future rewards and ensure convergence of the overall reward. Based on the solution of the Bellman equation, the policy will return the optimal action for a state with the maximum overall reward.

The policy obtained as the output of the MDP is constantly updated through success criteria for desirable system performance, while the system is supporting a self-organization control method that spontaneously and internally compensates for severe disturbances.

6.6. Self-Organization Method for a Hexapod

To illustrate the proposed self-organization method, a hexapod robot is selected as the test system. The mission profile is set for the hexapod to travel from a current point A to a goal point B in a straight-line path. Cully et al. [46] suggest an improved trial and error method to determine the optimal action for a walking hexapod with a broken leg, but the large original search space and minutes of lengthy adaptation time to the next step hamper their development. Instead, a self-organization method that spontaneously generates the optimal action can provide an alternative to decrease significantly the computational burden.

6.7. Hexapod Dynamic/Kinematic Model

The hexapod used in this case study is composed of a body and six legs. Each leg is a three DOF (degrees of freedom) subsystem with three servo joints. From the work of Sorin et al. [47] (2011), the names and functions of each joint and link are shown in Figure 8.

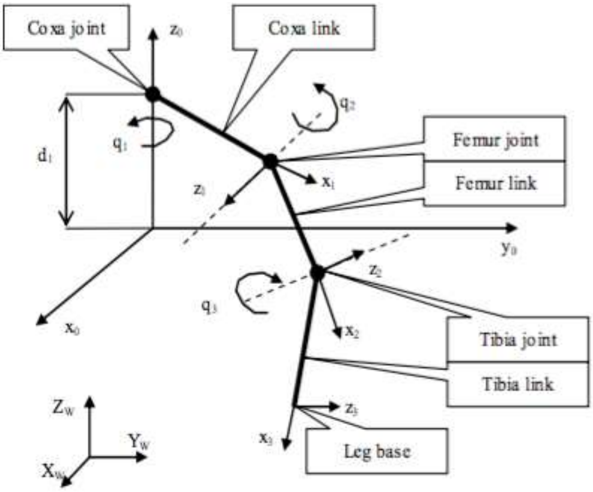
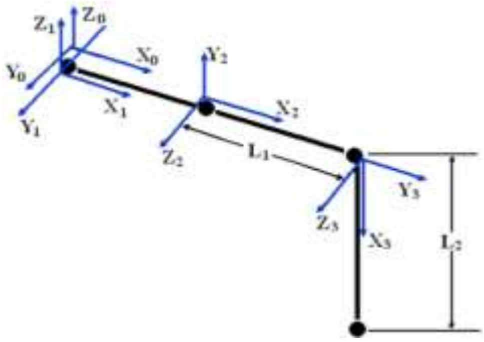


Figure 8. Hexapod leg structure.

It can be seen that the Coxa joint rotates horizontally, and the other joints rotate vertically. The hexapod leg's kinematic movement is governed by the joint motor rotations, and the position of the end point of a leg contacting the ground with respect to the hexapod body can be obtained by applying consecutive coordinate transformation matrices, assuming the origin as the center of the body, as shown in Equations (4,5,6,7), Barai et al. [48]. In the equations, ${}^{i}_{j}T$ is the transformation matrix from joint i to joint j, s_i and c_i are the sine and cosine functions of ith joint angle, L_i is the link length, and P_f is the end position of the leg. The Coxa joint operates horizontally for the leg's swinging action, and the other two joints operate vertically for the leg's stance (vertical lift/put) action. Considering a leg as a combination of three different joint coordinates, the visualization can be described with:



Scheme 1: Visualizing the leg joint coordinates

Each joint has its own coordinates as seen in the figure above, and the endpoint location of the foot can be obtained by applying a series of coordinate transformation matrices: Each joint has its own coordinates as seen in the figure above, and the endpoint location of the foot can be obtained by applying a series of coordinate transformation matrices (M1):

$$ {}^{0}_{3}T = {}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T $$

$$ {}^{0}_{3}T = \begin{bmatrix} c_1(c_2c_3 - s_2s_3) & c_1(-c_2s_3 - s_2c_3) & -s_1 & c_1c_2L_1 \\ s_1(c_2c_3 - s_2s_3) & s_1(-c_2s_3 - s_2c_3) & c_1 & s_1c_2L_1 \\ -c_2s_3 - s_2c_3 & s_2s_3 - c_2c_3 & 0 & -s_2L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(M1)} $$

The left and right indexes indicate joint numbers, while s denotes sine, c denotes cosine, and L denotes the link length. The end position of the foot from the origin (index 0 in the figure above) can be obtained by (M2):

$$ {}^{o}P_f = {}^{0}_{3}T\,{}^{3}P_f = {}^{0}_{3}T \begin{bmatrix} L_2 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_{fx} \\ P_{fy} \\ P_{fz} \\ 1 \end{bmatrix}, \quad \begin{aligned} P_{fx} &= L_2c_1(c_2c_3 - s_2s_3) + c_1c_2L_1, \\ P_{fy} &= L_2s_1(c_2c_3 - s_2s_3) + s_1c_2L_1, \\ P_{fz} &= -L_2(c_2s_3 + s_2c_3) - s_2L_1. \end{aligned} \qquad \text{(M2)} $$

The coordinate transformation in the left equation is from the origin to the foot. As mentioned above, each joint angle has its own reachable range. Therefore, as the joint angles vary during motion, the end point can be obtained by plugging in the corresponding joint angle values in the equations above.

However, in case of a locked joint failure, a joint angle is fixed at a certain state and the controllability is lost. Therefore, the attainable end-point location (workspace) of the faulty leg is reduced due to the limited joint angle. Different joint failures will have different effect on the system behavior. For example, if the Femur node of a leg is under locked joint failure, θ_2 in the transformation matrix equations will be a constant, while other joint angles can operate in their ranges. The workspace under locked joint failure will also be affected by what angle the joint is locked in, since different angle values result in different foot positions, as mentioned in the coordinate transformation equations above. Visualization of the impact of locked joint failure on hexapod leg's workspace is shown in Figure 9 below.

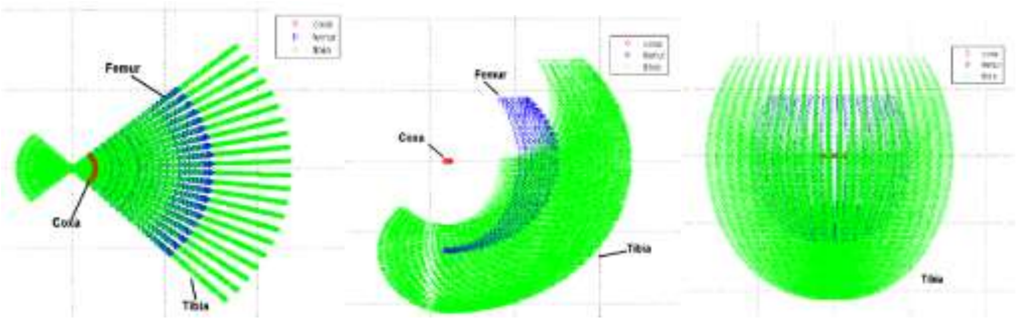
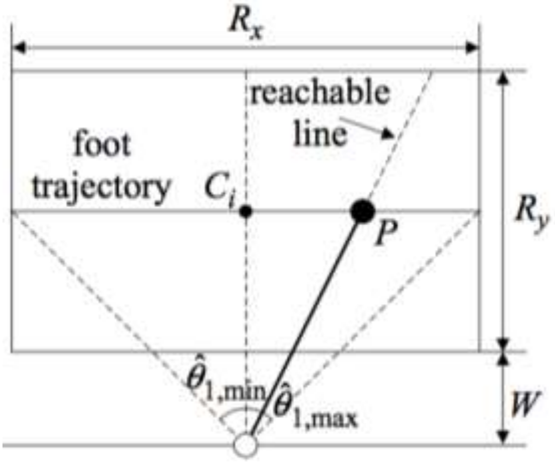
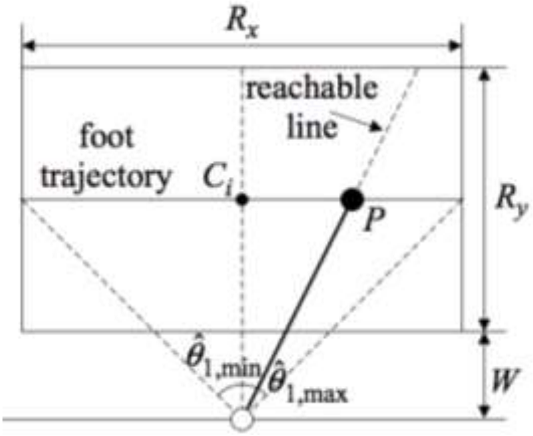


Figure 9. Locked joint failure impacts on the leg workspace (failure of Coxa, Femur, and Tibia joints).



Scheme 3: The foot trajectory

Analytically, the locked joint failure of the Coxa node causes the faulty leg to only place its foot on a single foot position, since there is no communication (controllability) available between the core and the other joints. A visual representation of the impact of Coxa failure from a vertical viewpoint is shown below.

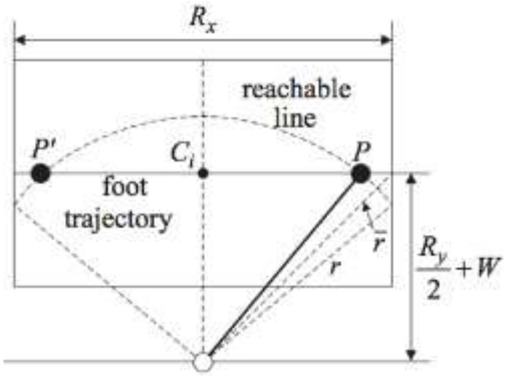


Scheme 4: Representation of the impact of Coxa failure

R_x and R_y are the reachable range in x and y directions, C_i is the center of gravity of the leg, P is the foot position, and W is the distance of the leg workspace from the hexapod body. Here, since the locked joint angle should lie within the joint's original reachable range, the constraint of the Coxa angle can be written as (J1):

$$\hat{\theta}_{1,\min} = -\arctan\left(\frac{R_x}{R_y + 2W}\right)$$
$$\hat{\theta}_{1,\max} = \arctan\left(\frac{R_x}{R_y + 2W}\right). \tag{J1}$$

On the other hand, locked joint failure of the Femur or Tibia joint causes the faulty leg to only move its lower joint, which is either Tibia or none. This results in an arc-like workspace as shown in the visualization above. As for the Femur joint failure, the vertical visualization of the failure can be shown by:



Scheme 5: The Femur joint failure

r indicates the radius of the arc and has the range of (R1):

$$\frac{R_y}{2} + W \leq r \leq \bar{r} \tag{R1}$$

Here, $\bar{r}$ and r can be rewritten as (R2):

$$\begin{gathered} \bar{r} = \frac{1}{2}\sqrt{R_x^2 + (R_y + 2W)^2} \\ r = l_1 \cos\hat{\theta}_2 + l_2 \cos\theta_3 \end{gathered} \tag{R2}$$

Then the constraint on r results in (R3):

$$\frac{R_y}{2} + W \leq l_1 \cos\hat{\theta}_2 + l_2 \cos\theta_3 \leq \frac{1}{2}\sqrt{R_x^2 + (R_y + 2W)^2} \tag{R3}$$

This constraint also applies to the case of Tibia failure.

All hexapod joints have a range of reachable angles ($\in [\theta_{min}, \theta_{max}]$), and the combination of the horizontal rotation of the Coxa joint and the vertical rotations of the Femur and Tibia joints create a set of reachable workspaces for the legs.

The hexapod locomotion to move to the goal position will be assumed a conventional tripod gait. At each time step, three legs are swinging in space, and the other three are supporting the robot on the ground. The duty factor of the gait between the left and right side of the hexapod is 0.5, implying an equal duration in alternating sides, where the swinging legs at each step are the middle leg on one side and the front and rear legs on the other side. The walking direction of the hexapod can be determined by combining the direction vectors of the three swinging legs at each step. Assuming normal condition with no disturbance, the legs will swing forward in the direction of the desired path, and the three direction vectors from the swinging legs can be combined, as shown in Figure 10, where the arcs indicate the reachable workspaces of the swinging legs.

To explore the impact of locked joint failure on system behavior, system modeling through spectral graph theory and epidemic spreading can be considered. In spectral graph theory, a complex, interconnected system can be represented by a graph G (V, E), which is a set where V contains the nodes, and E contains the edges (connections between two nodes). Adjacency matrix (A_{ij}) is defined as a square matrix where an element a_{ij} is 1 if there is connection between node i and node j, and 0 otherwise. Also, degree matrix (D) is a diagonal matrix with diagonal elements representing the node degrees (edges). Finally, the Laplacian matrix, which is also known as the system matrix, is defined as D – A_{ij}. It can be seen from the expression that the diagonal elements of the

Laplacian matrix are the node degrees, and the non-diagonal elements are -1 if there is connection from node i to j, and 0 otherwise. The Laplacian matrix can be converted into the linear system state matrix (A) by A = I – L, where I is simply the identity matrix.

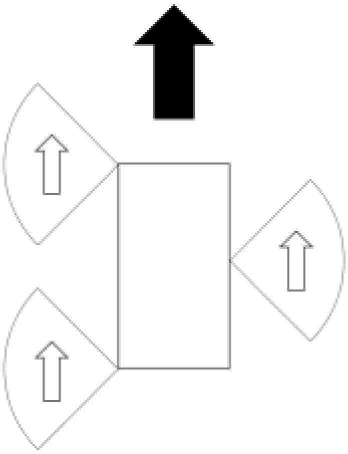


Figure 10. Upper view of the walking direction of the hexapod from the swinging motion.

6.8. Failure Mode (Locked Joint Failure)

The three joints on each leg can be combined for the hexapod to have 18 DOF. Such high degree of manipulation allows versatile motion for the hexapod, but the complexity also induces vulnerability to severe failure modes. An example of a possible failure mode in a hexapod is the locked joint failure, where a joint angle is fixed at a certain state and cannot be controlled. In the work of Yang [49]), locked joint failures of different joints are shown to result in different effects on the leg workspace. Since the Coxa joint is in charge of the horizontal swinging movement of the leg, locked joint failure at the Coxa joint completely disables the leg's swinging motion and the leg can only lift and plant itself vertically. On the other hand, locked joint failure at the vertically operating Femur and Tibia joints will have no effect on the swinging motion, but will diminish the leg's stretchable length, so the upper view of the leg's workspace will have the same arc shape but with reduced size. The impact of locked joint failures on the leg's workspace will cause the hexapod to derail from its original path in an unexpected manner.

6.9. Hexapod Graph Model

Figure 11 shows a graph representation of a hexapod with each node numbered appropriately.

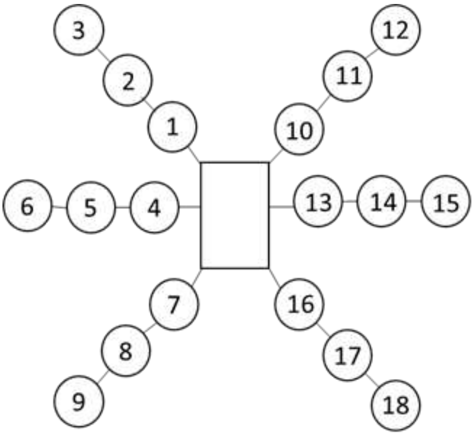


Figure 11. Graph representation of a hexapod with node numbers.

The impact of a locked joint failure in a hexapod can be represented simply by a node removal. More specifically, a failed node loses controllability and converts into an edge, so the adjacency and degree of the corresponding node become zero. The aforementioned algebraic connectivity can be used to indicate the presence or severity of locked joint failure(s). Demonstration of the effect of locked joint failure on algebraic connectivity is shown in Table 1, where the algebraic connectivity of the hexapod system graph is evaluated in normal condition first followed by the failure mode at node 8, and finally another failure mode at node 15.

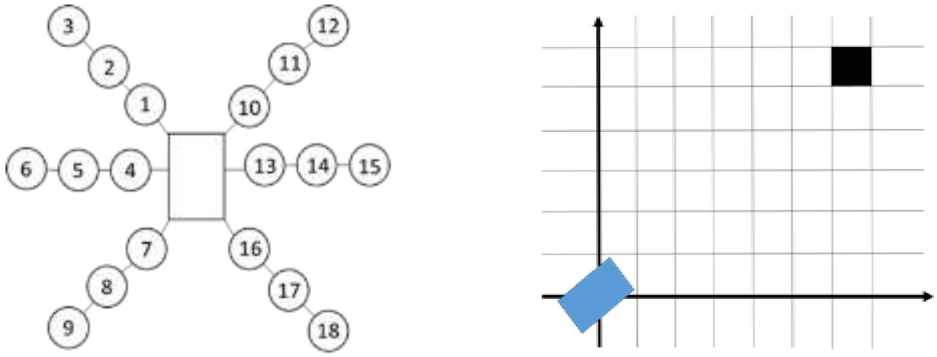
Table 1. Effect of failure mode on algebraic connectivity

# of Failure Mode	Algebraic Connectivity
0	0.3384
1	0.04874
2	-3.0199e-16 ≈ 0

It is shown that the incremental addition of failure modes decreases the algebraic connectivity, and when there are two failure modes present, the algebraic connectivity becomes zero, meaning the graph is disconnected.

6.10. Hexapod Epidemic Spreading Model

The graph-based self-organization method will consider two graphs: the physical graph and the behavioral/spatial graph, as shown in Figure 12. The physical graph will depict the actual structure and interconnectivity of the system, while the spatial graph will contain grids of positional states in the Cartesian space. For example, when operating a hexapod robot, the two graphs are as follow:



(a) Physical graph

(b) Behavioral/Spatial graph

Figure 12. Illustration of the physical and the spatial graphs of a hexapod.

In the spatial graph, the individual boxes in the grid represent the positional states the system can occupy, where the position of the state can be considered as the center point of the box. The grid spacing will be chosen carefully. Assuming discrete time, the underlying assumption in using the spatial graph is that state transitions can only take place between adjacent (neighboring) boxes, where the system moves either one or zero state at each time step. In other words, the system's operational speed will be regulated so that it does not jump multiple states at a time, which could lead to high instability.

When the goal point is located to the upper right side of the system in the Cartesian space, the state transition probabilities will be assigned so that higher probability is given to the direction to the goal point, with the sum of the neighboring transition probabilities equal to one. Therefore, the system is definitely more likely to move towards the upper right direction.

0.05	0.25	0.379
0.01	System	0.25
0.001	0.01	0.05

Scheme 6: State transition probabilities

The optimal path to the goal point can be obtained by defining a value function so that each state contains a "value" that will be accumulated along the way. Similar to the state transition probabilities, the values will be assigned so that higher value is given to the state that leads to optimal system behavior. However, the values do not necessarily have to be probabilities, but rather they can be any real valued numbers ($< \infty$). For this problem, we are assuming flat ground with no obstacles, so the value assignment simply focuses on the proximity to the goal point, but in case of different terrains or obstacles, the values can be assigned accordingly. The goal is to find the path that maximizes the cumulative value.

In the case of a failure, the system will digress from the optimal straight-lined path, so the state values will need to be modified. The rate of change of the value will be controlled in each time step so that the effect of failure makes sense in the spatial graph.

$$V_{new} = \varepsilon(V_{target} - V_{original}) \tag{V1}$$

Where the constant ε is a small scalar value ($\ll 1$) that regulates the changing value, V_{new} is the actual resulting value in the next time step, and V_{target} is the target value to which it must increase/drop due to the change in system environment.

6.11. Hexapod MDP

The MDP solution in this problem will be a policy that maps the optimal action for each state for the hexapod to move along the desired path. Chades et al. [50] provide a Markov Decision Process Toolbox for MATLAB to compute the MDP solution. By initializing the S×S×A transition and reward matrices, where S is the number of states, and A is the number of actions, the finite horizon solution of the MDP is concluded in N number of steps. In addition to the policy, the toolbox also outputs the used CPU time to solve the problem, which can be used to compare the computational burden to that of traditional robust control methods.

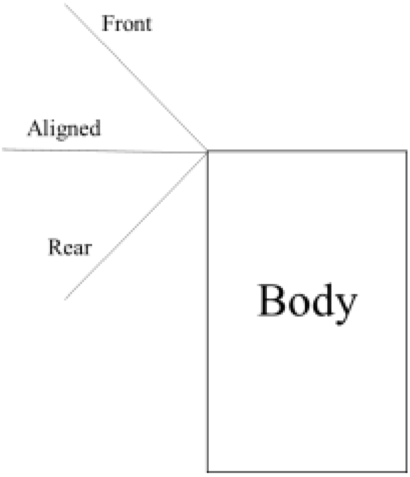


Figure 13. Example valid state space of a hexapod leg.

The MDP formulation is applied, as the self-organization strategy, to the hexapod under a locked joint failure. As suggested by Cuaya-Simbro and Munoz-Melendez [51], the state space S can be defined as the valid positions of the legs, and the action space A as the transitions that enable the robot to move from one valid state to another. The finite deterministic case of the MDP algorithm is used for this problem, and the state space is represented by three available leg positions at each step (front, aligned, and rear with respect to the leg's connection to the body, as shown in Figure 13).

Since three legs (two from one side and one from the other) either swing or support the system identically in a tripod gait, there are 9 total available states. Assuming the states can move to any other valid state, the action space A will have the same dimension as the state space.

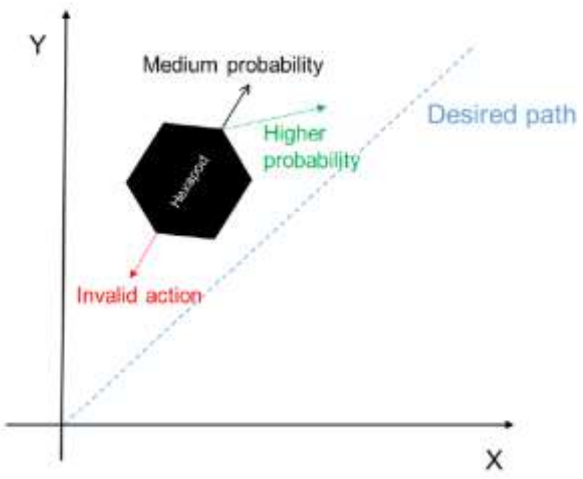


Figure 14. Visualization of the hexapod MDP.

The self-organization strategy for a hexapod is tested in VREP (Virtual Robot Experimentation Platform), which is an open source robot simulator with an integrated development environment. The hexapod starts at the origin and the diagonal path of y = x on the XY plane is assumed to be the desired path for the hexapod to travel. The XY plot of the normal case without failure mode can be obtained in the simulator and is shown in Figure 14, where the hexapod successfully follows the desired path.

6.12. Success Criteria (Lyapunov Stability)

The hexapod's moving path following the MPD policy solution is evaluated via Lyapunov stability conditions to verify the stability and effectiveness of the self-organization method. Path-based Lyapunov stability analysis of a hexapod is detailed in the work of Jeong et al. [52], where the positional error vector, e, is defined as in Equation (8) so that the Lyapunov function V and the Lyapunov equation are stated in Equation (9).

$$e = \begin{pmatrix} x_e \\ y_e \end{pmatrix} = \begin{pmatrix} x_d - x_c \\ y_d - y_c \end{pmatrix} \tag{13}$$

$$\begin{cases} V = e^T K e \\ A^T K A - K + Q = 0 \end{cases} \tag{14}$$

x_d denotes the desired position, x_c denotes the current position, A is the state transition matrix where $\overrightarrow{x_{t+1}} = A\overrightarrow{x_t}$, and the matrices K and Q are symmetric positive definite matrices.

6.13. Results

Using the MDP Toolbox, the 9×9×9 transition and reward matrices can be defined following the hexapod structure and tripod gait behavior. Assuming 10 steps between current and goal points, the finite horizon MDP problem can be solved to give the value functions for each state, policy of optimal action for the states at each time step, and the used CPU time to compute the policy. The value functions of the first five states (S1 ~ S5) are shown in Figure 15, where the value (immediate reward) converges to zero over the time steps for each state so the overall reward converges to a certain value.

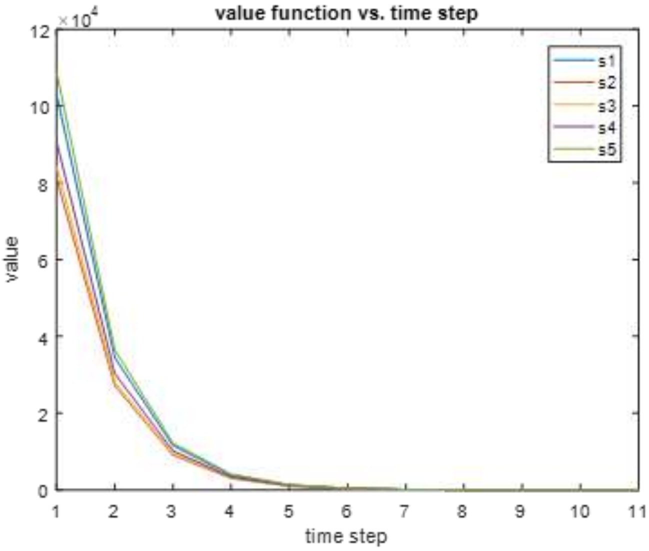


Figure 15. Value function computed over 10-time steps.

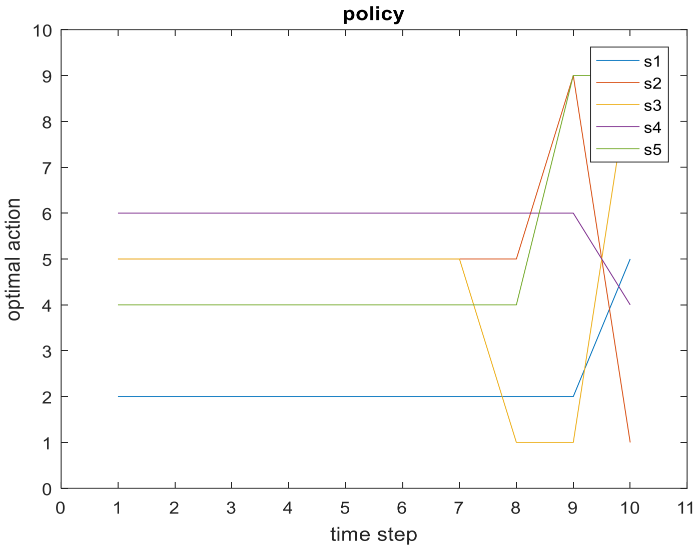


Figure 16. Graph Representation of the Policy.

The policy is generated as a 9×10 matrix where each row corresponds to a state and the columns store the optimal action for the state at each time step. A graph representation of the policy (first five states, for simplicity) is shown in Figure 16.

The notable part of the MDP Toolbox demonstration is the CPU time, which results in 0.0156s for mission completion. System parameters need to be adjusted in case of a failure mode, as the change in dynamics can cause unexpected system behaviors. Since commonly used robust control methods with PID controllers are designed for systems with predefined dynamics and properties, spontaneous and reliable adjustment to new system dynamics with PID controllers is impractical in real life.

Considering the hexapod for example, the lengthy settling time required for the PID controllers to reach steady state will cause the hexapod to take several seconds to adjust to a different configuration while tuning the gains for all 18 joint motors. Even assuming perfectly synchronous joint rotations, using PID control for a real-time hexapod gait is impractical. Moreover, compared to the minutes of computation needed for the trial and error method mentioned in Section 2.3, the proposed self-organization method offers a holistic guide of system behavior for disturbance accommodation with dramatically less computational burden.

Assuming the hexapod starts at the origin, a diagonal path in the XY plane can be assumed to be the desired path for the hexapod to travel. A locked joint failure is added to the left-middle leg of the hexapod to test the self-organization behavior. For both conditions (with or without failure mode), the hexapod behavior is governed by the MDP policy. The resulting paths are shown in Figure 17, where the hexapod travels along the desired diagonal path. With a failure mode present, the hexapod abruptly moves to the left due to the failure mode and then gradually converges back to the desired path.

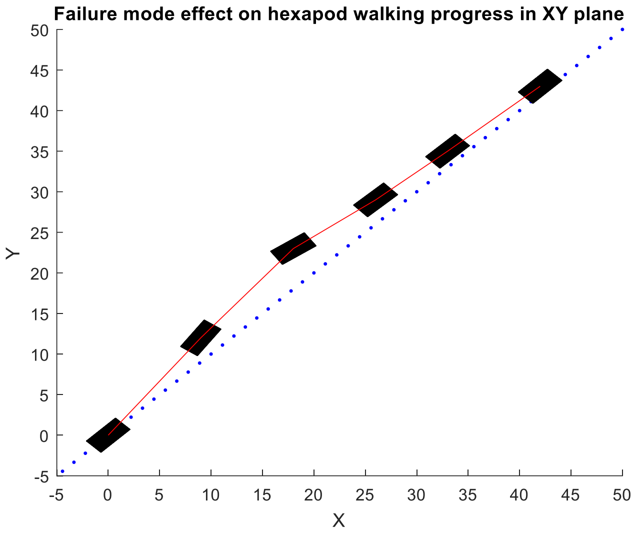


Figure 17. Hexapod walking path in nominal condition and with locked joint failure.

To verify that the hexapod with locked joint failure returns to a stable mode, the distance from the desired path (= difference between x and y coordinates) can be used as the error value and apply the Lyapunov stability conditions. The stability condition is constructed as in Equation (10), assuming K = 1 or the identity matrix (for dimensions larger than 2).

$$\begin{cases} e_t = |y_t - x_t| \\ Q = 1 - A^2 = 1 - \left(\frac{e_{t+1}}{e_t}\right)^2 > 0 \end{cases} \tag{15}$$

In other words, the positional error of the vehicle must decrease in magnitude at each time step for the process to be stable. The resulting Q value evaluation is shown in Figure 18.

It can be seen from Figure 18 that the process becomes unstable when the failure mode is initiated, then returns to a stable state between the second- and third-time steps. Upon detection of noticeable digression from the desired path, the predetermined MDP policy guides the hexapod to return to traveling in the direction of the desired path, which shows in the stability evaluation as the Q value drops to negative upon occurrence of a failure mode then turns positive as self-organization process is activated.

The self-organization method proposed in this paper combines the Markov Decision Process with Lyapunov stability conditions for a complex system to maintain stability under a severe failure mode. The proposed method demonstrated its usefulness with highly reduced computational burden in the test case applied to a hexapod under locked joint failure compared to traditional disturbance rejection methods, while the system maintains stability conditions. Future work will be toward improving the self-organization method through deeper analysis in resilience, focusing on the vulnerability and recoverability of systems under failure modes.

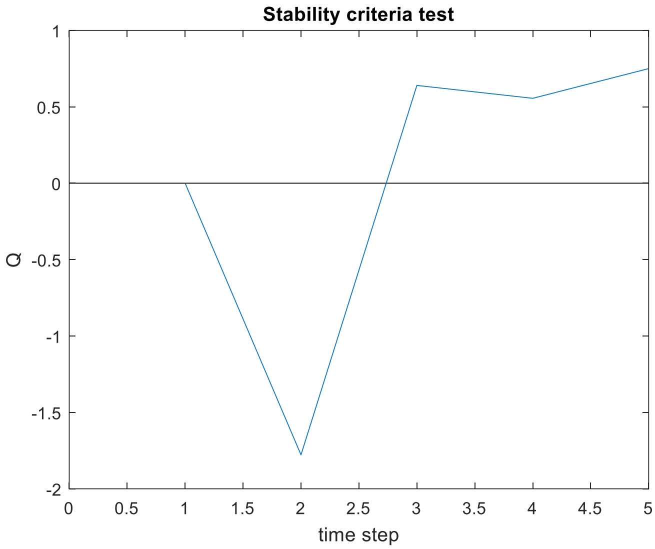


Figure 18. Q value of the hexapod with locked joint failure.

7. Fault–Tolerant Control (FTC) Strategies

Modern technological systems rely on sophisticated control systems to meet increased performance and safety requirements.

Definition 7.1. (Fault Tolerant Control [26])

Control systems that possess the ability to accommodate system component failures automatically [while] maintaining overall system stability and acceptable performance.

Traditionally, FTC systems are classified into two categories: passive and active [53]. Passive Fault Tolerant Control Systems (PFTCS), are designed to make the closed loop system robust against system uncertainties and anticipated faults [54]. For this reason, PFTCS have a limited fault-tolerant capability. Alternatively, Active FTC Systems (AFTCS) react to the system component failures by reconfiguring control actions to maintain stability and acceptable system performance. In such control systems, the controller compensates for the effects of faults by selecting a pre-computed control law or synthesizing a new control scheme on-line.

7.1. Model Predictive Control (MPC)

Model predictive control, or receding horizon optimal control (RHOC), is a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. This is its main difference from conventional control which uses a pre-computed control law [55, 56]. MPC generates a discrete-time controller which takes action at regularly spaced, discrete time instances. The interval separating successive sampling instants is the sampling period, Δt. where the latest measured output, $\mathbf{y_k}$, and previous measurements, $\mathbf{y_{k-1}}$**,** $\mathbf{y_{k-2}}$, . . ., are known.

To calculate the next control input the controller operates in two phases, *estimation* and *optimization*,

1) **Estimation.** The controller updates the true value of the controlled variable, $\mathbf{y_k}$ and any internal variables that influence the future trend, (i.e., $\mathbf{y_{k+1}}$, ..., $\mathbf{yk_{+P}}$).
2) **Optimization.** Values of set points, measured disturbances, and constraints are specified over a finite horizon of future sampling instants, $k + 1, k + 2, ..., k+P$ where $P \in Z^{+}$. The controller computes M modes u_k, u_{k+1},…, u_{k+M-1}, where 1 ≤ M ≤ P is referred to as the control horizon.

The MPC is obtained by solving the optimization problem,

$$J = \int_{t_k}^{t_{k+1}} [(r-y)^T Q(r-y) + \Delta u^T R \Delta u]\, dt + \rho_\varepsilon \varepsilon^2 \qquad (16)$$

where the variables **r**, **y** and Δ**u** correspond to the input reference, plant output and control correction. The weight matrices **Q** and **R** are defined a-priori as the inverse of the maximum allowable tracking error and control correction, respectively. An illustration of the non-linear system with MPC is provided in Figure 19.

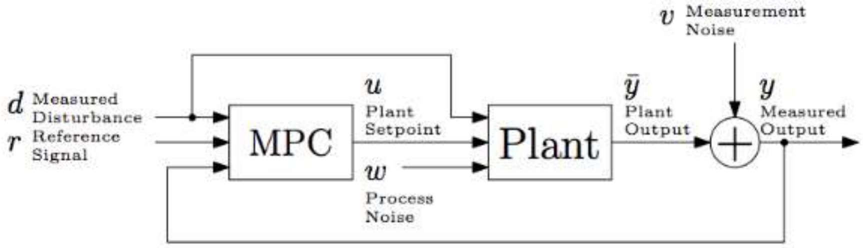


Figure 19. Block diagram of MPC with plant and signals.

Monaco et al. [57] demonstrated an MPC based framework used to retrofit the F/A-18 Fleet Support Flight Control Computer (FSFCC) with an adaptive flight controller. The authors utilize a con- strained parameter identification algorithm to provide on-line model corrections to account for uncertainties or changes in the current aircraft dynamics. The updated estimates are utilized in a MPC to provide increments to pilot commands. The increments from the adaptive control law reduce tracking error given the closed-loop dynamics of the aircraft. A key benefit of the approach is the adaptation is only significant if the aircraft behavior differs appreciably from the intended closed-loop flying qualities.

8. Control Reconfiguration of Unmanned Autonomous Systems

8.1. Control Reconfiguration Fundamentals

Reconfigurable design of systems centers on incorporating autonomy and resilience, sustainment and reliability under changing operational requirements, severe disturbances (internal and external) and uncertain/dynamic environments or mission profiles, without major changes to the system's initial design. We address such challenging questions as i.) How does reconfiguration of one component affect the operation of other, neighboring, components? ii.) What is an appropriate strategy to maintain desired system behavior? We present a methodology for reconfigurable design and performance evaluation of complex systems paving the way for the design and construction of resilient, high-confidence autonomous systems. We are introducing a novel approach to fault-tolerance by considering the impacts of severe fault modes on system performances as inputs to Reinforcement Learning (RL) strategy that trades off system performances with control activity in order to extend the Remaining Useful Life (RUL) of the unmanned system so that a detrimental event does not occur in the presence of severe fault modes. The proposed approach employs a software solution and does not require a hardware complement to be integrated in the system design. The proposed architecture performs one of three actions, low-level control reconfiguration at the component level, mid-level control redistribution at the sub-system level and high-level mission adaptation at the highest echelon of the architecture. The fault-tolerant control begins with reconfiguration at the low-level since the impact of reconfiguration is localized to the individual component. If component reconfiguration is not sufficient to meet the mission objectives, control redistribution is performed at the middle level. The impact of control redistribution affects all components within the subsystem. This action provides more flexibility over component reconfiguration at the expense of increased computational complexity. Finally, if the previous actions are insufficient in achieving the desired objectives, mission adaptation is performed. During this action, lower-priority mission objectives are compromised or traded-off to achieve higher priority objectives. To handle severe disturbances, the middle-level reconfiguration plays critical roles for resilience in that trade-offs of system performances are actively considered. The theoretical underpinnings for the proposed resilience-based reconfiguration rely on concepts of DCBR, RL, and MPC-DDP. The integrated framework of three modules is a decision-making process, which optimizes control actions and system behaviors in order to extend RUL under severe fault modes. What distinguishes this framework from traditional reconfigurable control methods is to consider not only the current degraded states, but also consequences of them after all by RL.

RL is a supervised learning algorithm, seeking actions in environments (or mappings from states to actions) to maximize given rewards [58]. In RL, an agent takes an action in environments and observe changes of states accordingly. The observed states are translated into rewards, then an agent takes another action to collect maximum rewards over a given mission. What distinguishes it from typical optimal control methods is that the environment is not known to an agent; thus, the agent learns dynamics of environments by interacting with environments, so that it can choose optimal actions, which can produce the maximum rewards after all. In this sense, RL explicitly considers the whole picture of a mission.

RL utilizes the concepts of DP and Bellman's principle based on the MDP formulation of an environment, without the knowledge of system dynamics, but measurement data coming from interactions with environments. Since the environment is unknown, learning is realized by exploiting learnt policies and exploring an unknown state-action space [59]. There are two approaches: off-policy and on-policy methods. Off-policy methods, also called "an estimation policy," use a greedy search to determine control actions. A behavior policy makes decisions about control actions among all possible actions having a finite probability of being selected. On-policy methods, on the other hand, evaluate and improve control policies at the same time with a ε-greedy method, which chooses control actions by the probability, ε, to determine whether it takes a greedy action or random move. Both approaches include random moves, and it may cause unstable system behaviors during the exploring phase. To address this issue, RL learns the level of adaptation by adjusting an adaptation parameter of a cost function in MPC-DDP, in the proposed reconfigurable control framework. A general formulation of MPC in this framework addresses the solution of:

$$V(\boldsymbol{x}(t_0), t_0) = \min_{\boldsymbol{u}} \left[\int_{t_0}^{t_f} l(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau)\, d\tau + \Phi\big(\boldsymbol{x}(t_f), t_f\big) \right] \tag{17}$$

subject to:

$$\frac{dx}{dt} = F\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \tag{18}$$

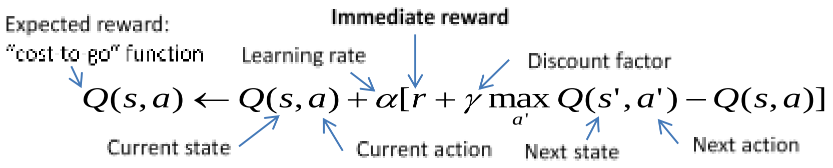
$$g\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \leq 0 \tag{19}$$

where t_0 is an initial time and t_f is a terminal time. $l(\cdot)$ is a scalar running cost, $\Phi(\cdot)$ is a scalar terminal cost, $F\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big)$ is a generic nonlinear system dynamics as an equality constraint of the optimization problem, and $g\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big)$ is a general function for inequality constraints.

The running cost function, $l(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau)$, typically formulated as a quadratic function, as shown in Equation 20. It is a linear combination of two terms: system performances and control efforts.

$$l(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \frac{1}{2}\big(\boldsymbol{x}(t) - \boldsymbol{r}(t)\big)^T K \,\big(\boldsymbol{x}(t) - \boldsymbol{r}(t)\big) + \boldsymbol{\alpha} \cdot \frac{1}{2}\boldsymbol{u}(t)^T R \,\boldsymbol{u}(t) \tag{20}$$

where $\boldsymbol{r}(t)$ is a reference, K and R are coefficient matrices, and $\boldsymbol{\alpha}$ is an adaptation parameter. The right hand-side of Equation (3) refers to the energy of the states and control inputs at each time instance. The adaptation parameter determines the trade-offs between system performances and control efforts. In this way, model-free and model-based optimal control techniques complement each other; MPC-DDP utilizes system dynamics to produce control actions stabilizing system behaviors for a finite time, and RL finds optimal weightings for the entire mission while it does not incur unstable behaviors during the exploration. Among RL methods, Q-learning is one of the most popular methods using TD learning. A general Q-learning formulation is:



$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{21}$$

The action a to be learned in the Q-learning process is $\boldsymbol{\alpha}$ in the MPC formulation.

8.2. Control Reconfiguration: The Design Process

The fault-tolerant control framework, as applied to a generic autonomous system model, may be summarized as follows: We propose that the vehicle has a nominal controller (a pilot or autopilot) that controls the vehicle adequately when no fault is present. When a fault is detected, the fault-tolerant control scheme will reconfigure the nominal control commands to maximize safety and minimize performance degradation.

The general development tasks are identified as:

1) Define an objective function to quantify the tradeoff between minimizing performance degradation and risk of failure
2) Define a probabilistic prognostic model for all fault conditions
3) Understand the ability to redistribute component loads in the system
   - Constraints are derived from: stability bounds, performance bounds, prognostic bounds and effector limits
4) Define the optimization algorithm
5) Understand how the pilot will react to control reconfiguration commands and explore means to inform the pilot/autopilot if imminent fault-tolerant mitigation actions.

The control architecture is comprised of the closed-loop system (plant & production controller), reconfigurable controller and an external prognostic module. Initially, the production controller is utilized with no modification while the prognostic module continuously monitors the system for one, or more, fault modes. Once a fault is detected, the RUL is evaluated by the prognostic module. If the estimated RUL is greater than the desired (or mission) RUL, no action is taken. During this period the RUL is re-evaluated periodically. However, if the estimated RUL is less than the desired RUL a reconfiguration action is triggered. The reconfigurable controller relaxes constraints on the error boundaries by adjusting the weight matrices in the MPC controller. This continues until either the RUL is satisfied or the weight matrices can no longer be adapted. The remainder of this section presents a detailed description of each module in the reconfigurable control architecture.

8.3. The Reconfiguration Strategy

We propose a novel fault-tolerant (fault-mitigation, reconfigurable control and fault accommodation) methodology that builds upon a central theme starting with low-level reconfiguration but promoting “intelligent” concepts, such as game theory, “smart” search engines, etc., as we migrate to the higher echelons. Our primary objective is two-fold: To design an adverse event mitigation strategy that is mathematically rigorous and generic, while incorporating prognostics, and is applicable to critical aerospace and autonomous systems.

The innovative feature of the adverse event mitigation architecture is the utility of real-time prognostic information in the design of the control algorithms. Given accurate on-line prognostic information in terms of estimates of the Remaining Useful Life (RUL) or Time to Failure (TTF) of a failing component/subsystem, the proactive fault accommodation system manages the accumulation of further damage through control actions until major flight/mission objectives are achieved although the system is in an impaired state [60] (Tang et al., 2008). This approach constitutes a major paradigm shift in the way fault-tolerant systems are designed and operated. The implications to system survivability, safety and availability to complete a critical flight/mission are significant [61, 62].

We leverage work performed in data mining, diagnostics/prognostics and uncertainty representation and management. Our approach initially will address the enhancements from the deterministic case through consideration of higher order moments for the pdfs representing the system states and the prognostic profile. Assessing the uncertainty-complexity tradeoffs, we will investigate further improvements moving towards more complete stochastic representations for system variables/parameters. Our approach to fault-tolerance utilizes a three-tiered architecture. Figure 20 depicts the major components of the architecture.

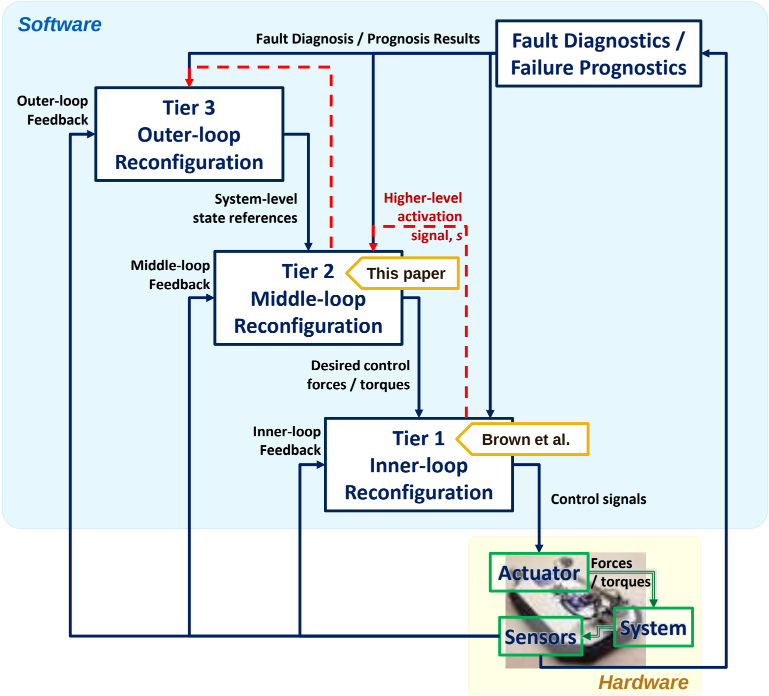

Software
Fault Diagnosis / Prognosis Results
Fault Diagnostics / Failure Prognostics
Outer-loop Feedback
Tier 3 Outer-loop Reconfiguration
System-level state references
Higher-level activation signal, s
Middle-loop Feedback
Tier 2 Middle-loop Reconfiguration
This paper
Desired control forces / torques
Inner-loop Feedback
Tier 1 Inner-loop Reconfiguration
Brown et al.
Control signals
Actuator
Forces / torques
Sensors
System
Hardware

Figure 20. The three-tiered reconfiguration architecture.

The highest tier passes down trajectories and performance requirements for all of the subsystems in accordance with the current mission/flight. The middle level manages the subsystems, i.e., each subsystem must satisfy the requirements passed from the high level controller by distributing command signals and performance requirements to all of the components within that subsystem. The lowest level manages the individual components; it predicts the components' Remaining Useful Life and modifies the control of components in order to extend their RUL. The fault-tolerant control scheme performs control reconfiguration, redistribution and mission/flight adaptation as necessary to meet specified objectives. A flowchart of the proposed fault-tolerant control hierarchy is provided in Figure 21.

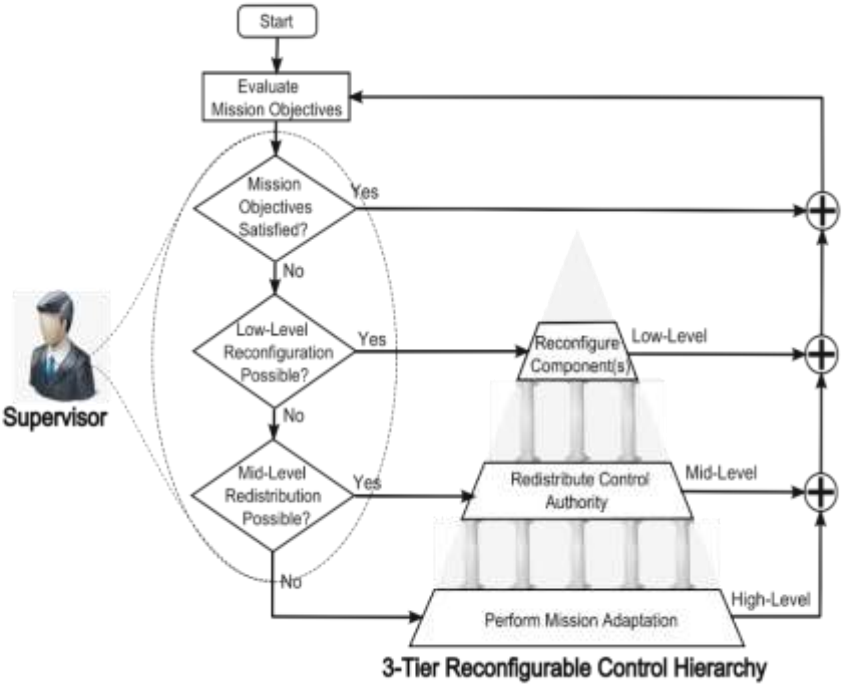

Start
Evaluate Mission Objectives
Mission Objectives Satisfied?
Yes
No
Low-Level Reconfiguration Possible?
Yes
No
Mid-Level Redistribution Possible?
Yes
No
Supervisor
Reconfigure Component(s)
Low-Level
Redistribute Control Authority
Mid-Level
Perform Mission Adaptation
High-Level
3-Tier Reconfigurable Control Hierarchy

Figure 21. The Control Reconfiguration Architecture.

8.4. Low-Level Reconfiguration

The goal is to introduce a reconfigurable controller to trade-off RUL for performance. In this design philosophy, we will use testbeds to define, develop, test, and illustrate the control concepts but we will also strive to extract and highlight those attributes that will eventually form the underpinnings of a general approach to automated contingency management for complex aerospace systems. First, consider an Electro-Mechanical Actuator with a Model Predictive Control used for low-level reconfiguration. In the case of the EMA, the RUL can be increased by lowering the applied motor current, im (or motor voltage Um). Although, the motor current cannot be adjusted directly, it can be controlled indirectly by making adjustments to the reference input, θref, as shown in **Error! Reference source not found.**. The purpose of the model predictive controller is to find the optimal for a given RUL and performance requirements. The method for estimating fault growth for the system will utilize the fault diagnosis and prognosis routines that we have already detailed at the component level [63, 64, 65].

Model Predictive Control (MPC)-The MPC formulation maximizes the system performance by minimizing the tracking error between the set points r and the measured plant output y. A general non-linear state equation can be expressed as,

$$\begin{cases} x(t+1) = f_m(x,u,v) \\ y(t) = h_m(x,u,v,w) \end{cases} \tag{MP1}$$

where x, u, d, v and w represent the model states, control input, measured disturbance, measured noise and process noise, respectively. An illustration of the non-linear system with MPC is given in Figure 19. The optimal control adjustments Δu are found by minimizing the cost function,

$$J(\Delta u, r, y) = \int_{t_k}^{t_{k+1}} \left[(r-y)^T Q(r-y) + \Delta u^T R \Delta u \right] dt + \rho_\varepsilon \varepsilon^2 \tag{MP2}$$

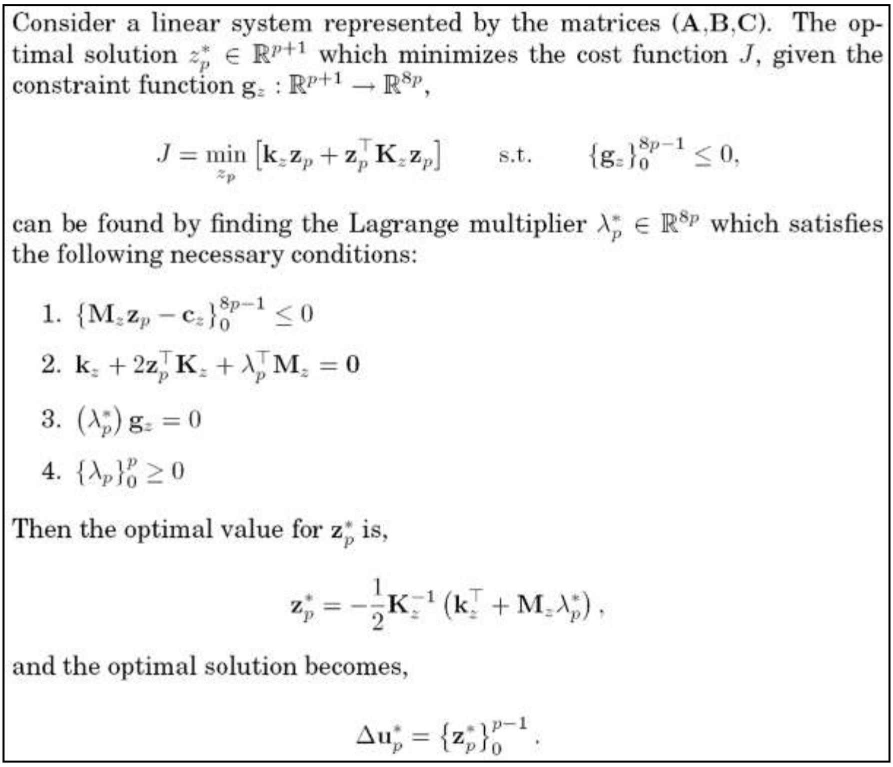
where ρε > 0 and ε = max(|u(t)| – |uRUL|, 0) s.t. t ∈ (tt, tk+1). The solution to the constrained MPC problem with prognosis is summarized in Figure 20.

8.5. Mid-Level Redistribution

The middle level of the fault mitigation architecture constitutes its most essential module since it enables the transition from the component-level reconfiguration to the subsystem and system fault tolerance, thus expanding significantly the practical utility of these emerging technologies. The Redistribution Controller at the middle level is tasked with the rerouting of the remaining available control authority between the subsystems when one or more are experiencing a fault mode. When the kth subsystem fails, the output of that subsystem, yk¸ will no longer be the same as the corresponding nominal output. The effect of this change will be felt by other subsystems via the interconnection variables. However, it may be possible to alter the flow of information through the interconnection structure in such a way that the impact on the other subsystems will be minimized or even eliminated. This task falls under the purview of the Redistribution Controller.

We propose the use of game theory as a generic means of finding solutions to this type of problem. We set up the problem as a two-level hierarchy, as shown in Figure 21, where the upper tier is occupied by a supervisor (or manager) who controls a set of variables that affect the behavior of the interacting, goal-oriented players or agents. The players manipulate the control of individual components according to some bounded rational strategy for minimizing their own cost functions, based on limited information about system states and the strategies of the other players. The middle level supervisor will modify the strategies of the players in such a way that the collection converges to a Nash Equilibrium that satisfies both RUL and performance objectives. The utility functions of the players may be maximized via a reinforcement learning scheme or an outer correction loop feedback arrangement.

For illustration purposes, we will adopt initially in simulation an EMA providing torque to a vehicle control surface (the load). When the motor is subjected to a fault condition (shorting windings, bearing spalling, etc.) it is desired to reconfigure the motor configuration so that its RUL is extended beyond the allowable (flight or mission specified) limits by trading off motor performance (speed, acceleration, etc.). The joint behavior strategy of the players (motor or controller) is defined in probabilistic term and a transition equilibrium is sought to optimize set objectives. The players are viewed as single-step MPC controllers. We will begin by considering a set of simplifying assumptions to address complexity issues before moving to an optimal management problem formulation.



Consider a linear system represented by the matrices $(\mathbf{A},\mathbf{B},\mathbf{C})$. The optimal solution $z_p^* \in \mathbb{R}^{p+1}$ which minimizes the cost function J, given the constraint function $\mathbf{g}_z : \mathbb{R}^{p+1} \rightarrow \mathbb{R}^{8p}$,

$$J = \min_{z_p} \left[\mathbf{k}_z \mathbf{z}_p + \mathbf{z}_p^\top \mathbf{K}_z \mathbf{z}_p\right] \qquad \text{s.t.} \qquad \{\mathbf{g}_z\}_0^{8p-1} \leq 0,$$

can be found by finding the Lagrange multiplier $\lambda_p^* \in \mathbb{R}^{8p}$ which satisfies the following necessary conditions:

1. $\{\mathbf{M}_z \mathbf{z}_p - \mathbf{c}_z\}_0^{8p-1} \leq 0$
2. $\mathbf{k}_z + 2\mathbf{z}_p^\top \mathbf{K}_z + \lambda_p^\top \mathbf{M}_z = \mathbf{0}$
3. $(\lambda_p^*)\, \mathbf{g}_z = 0$
4. $\{\lambda_p\}_0^p \geq 0$

Then the optimal value for $\mathbf{z}_p^*$ is,

$$\mathbf{z}_p^* = -\frac{1}{2}\mathbf{K}_z^{-1}\left(\mathbf{k}_z^\top + \mathbf{M}_z \lambda_p^*\right),$$

and the optimal solution becomes,

$$\Delta \mathbf{u}_p^* = \{\mathbf{z}_p^*\}_0^{p-1}.$$

8.6. High-Level Flight/Mission Adaptation

The final and highest level of the Adverse Event Mitigation hierarchy is intended to safeguard strict flight/mission objectives through the deployment of flight adaptation mechanisms when the middle and low-level of the fault-tolerant control scheme are not capable of achieving such objectives due to the severity of the contingency. Flight adaptation allows the control architecture to pursue relaxed flight objectives that do not belong to the strict or hard class, to achieve greater vehicle/system usefulness and absolutely necessary flight goals. The assigned flight objectives are expressed as system performance variables or a sequence of waypoints in the vehicle case. Mission adaptation alters parameters of the system states or of the individual waypoints, such as velocities and accelerations used by the planners to generate flight paths. By adapting state or waypoint parameters, the mission adaptation component enables the system/aircraft to accomplish an altered mission with increased usefulness. Changes to the parameters can be implemented within the receding window employed by the middle level control redistribution or path re-planning stage.

8.7. On-Line Reconfiguration: Mission Re-Planning

In path planning applications that require field exploration, incremental search algorithms, such as D* and its variants are widely used [66, 67, 68, 69]. Field D* is an interpolation-based algorithm that is able to generate direct, low-cost, and smooth paths in non-uniform environments [68].

In most applications, the vehicle has a large observation range. In most applications, the robot (Hexapod) has a large observation range. It is desirable for planning purposes to include all grids in the observation area. With this consideration, a receding horizon planning (RHP) strategy [70, 71] is proposed using all available map information and eliminating interpolation, which is not valid for a nonlinear cost function. To avoid deadlocking (a situation that the robot can no longer make progress toward its goal), a recursive searching is superimposed to RHP to make the planning algorithm more robust. Since field D* is fast in re-planning, the proposed recursive RHP (RRHP) outperforms other planning approaches when the system is reconfigured and goal is changed.

Figure 22 illustrates the RRHP scheme in three-levels. In the first level, an RHP is carried on in the observation area. In the second level, an RRHP is activated when the original RHP cannot find a feasible path in the observation area. In the third level, the algorithm re-plans the trajectory when the goal is changed. Since the map is explored in real-time, the RRHP algorithm is able to avoid collisions with moving objects/obstacles.

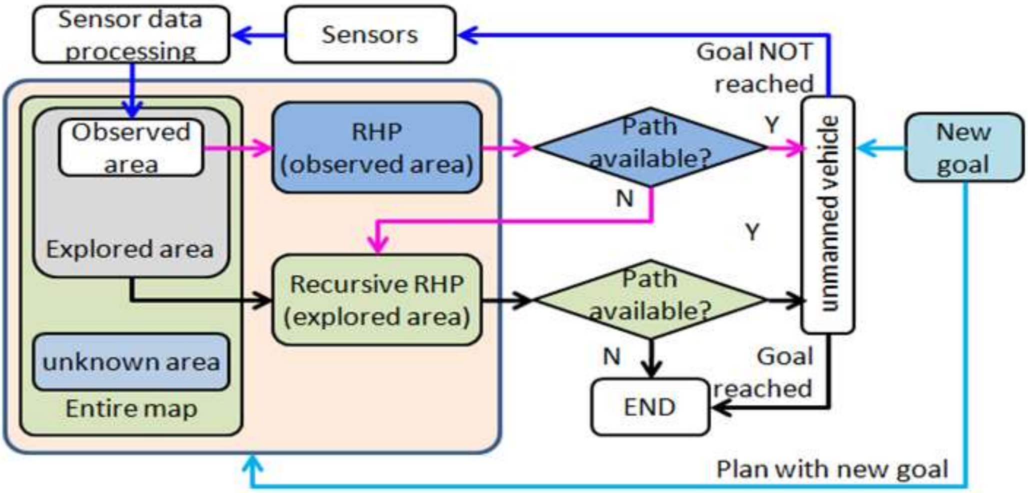


Figure 22. Recursive receding horizon planning algorithm for mission reconfiguration.

8.8. Receding Horizon Planning

In the proposed RHP scheme, the planning is carried out iteratively in the following steps:

1) At waypoint p_i, a fixed horizon optimization problem over $[p_{i+1}, p_{i+L}]$ is solved.
2) Execute the first waypoint p_{i+1} generated by step 1.
3) Update the map at waypoint p_{i+1}.
4) Repeat the step 1 to step 3 at waypoint p_{i+1} over $[p_{i+2}\ p_{i+L+1}]$.

The implementation of the RHP is elaborated in Figure 23(a). In this figure, the inner box (green) is the Implementation area, the outer box (magenta) is the Observation area, and all other areas beyond this one are grouped into the Unknown area. Each node indicates a possible path defined by two straight line segments from the robot current location *C* to the next node and to goal *G*. For node *s'*, the cost of the path is $\boldsymbol{c}(C, s')+\boldsymbol{g}(s')$, where $\boldsymbol{c}(C,s')$ is the cost on path segment $C \rightarrow s'$ in the observed area while $\boldsymbol{g}(s')$ is the cost of a path segment on $s' \rightarrow G$ in the unknown area. The planning algorithm attempts to find the node that has the minimal cost:

$$\min_{s'\in O}\left(c(C,s')+g(s')\right)$$

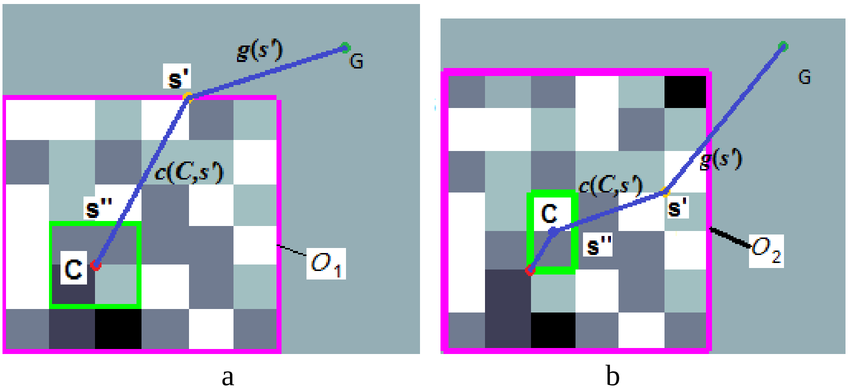


Figure 23. Illustration of RHP; (a) Planning at step 1 from initial position *C*. (b) Planning at step 2 from new *C* [*s''* in subfigure (a)].

Note that a waypoint *s''* is generated on the edge of a neighboring grid. According to RHP, when the path is implemented, the robot moves to the next waypoint *s''*. Then the map is updated and the path is planned again if necessary. An example of path re-planning is illustrated in Figure 23(b). This example shows that RHP is able to locate waypoints anywhere on the edge of a grid without interpolation. The proposed algorithm can generate smooth paths with flexible heading directions that are able to reduce unnecessary turns and fit planning needs better.

8.9. Recursive RHP

In case the RHP cannot plan a path in the observation area, a recursive searching is activated in the explored area, which is the union of all previously observed areas. The searching horizons are illustrated in Figure 24. To implement RRHP, the nodes are divided into available nodes (where robots can move to) and unavailable nodes (where robots cannot move to).

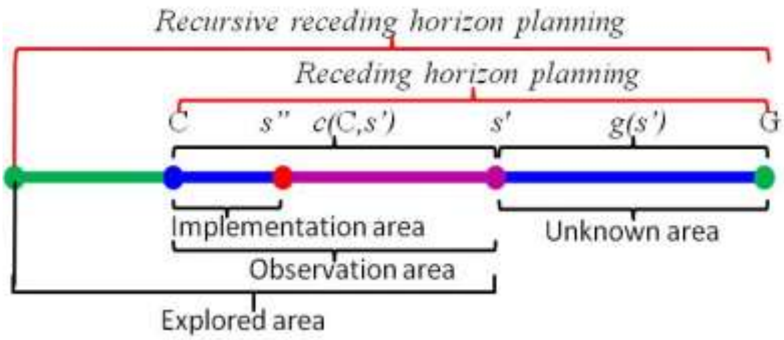


Figure 24. Illustration of horizons for RHP and RRHP.

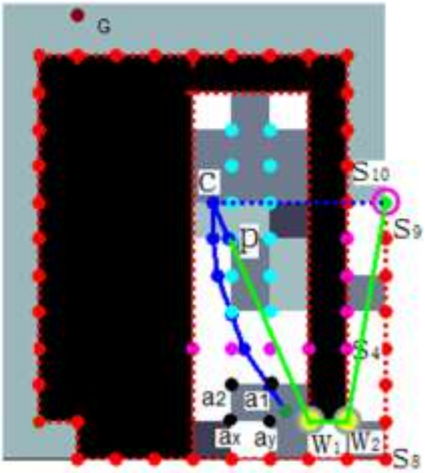


Figure 25. Illustration.

An example of the RRHP is shown in Figure 25. In this figure, *G* is the goal and the current location is *C*. It is clear that no path can be planned directly. To plan a path in the explored area, RRHP starts from an available node on the edge of the explored area (s_9 in this figure) and finds an available node in the explored area (such as W_2) that can lead to s_9 without collision. Then starting from W_2, RRHP finds another available node W_1 that is able to lead to W_2 without collision. Finally, RRHP finds robot can move to W_1 directly without collision. With these two available nodes W_1 and W_2, the path in the explored area can be planned as $C \rightarrow W_1 \rightarrow W_2 \rightarrow S_9$. For the path segment in the unknown area, it is the same as RHP.

8.10. Mission Reconfiguration with Goal Changes

When the system is reconfigured, the goal or destination of the system may change as well. If the new goal is located in the explored area, all planning will be carried out in the known area. If the new goal is located in an unknown area, the RRHP algorithm will be employed to search the area and plan the path from the robot's current location to the new goal.

9. Candidate Platforms

We adopt a number of experimental autonomous vehicles available in our laboratories, as the testbeds. They are designed to support appropriate sensing and control strategies as well as the ability to reconfigure on-line hardware and software modules of the system architecture. The three autonomous vehicles are shown in Figure 26. The first one is a hovercraft designed and built under NASA sponsorship. The second is biologically inspired micro aerial vehicle sponsored by AFOSR (Ratti, J., Jones, E., Vachtsevanos, G., Hovering and Gliding Multi-Wing Flapping Micro Aerial Vehicle, U.S. Patent No. US9, 290,268 B2, March 22, 2016). The third one is a hexapod instrumented appropriately to support the testing and algorithm development and validation. An autonomously operable underactuated hovercraft was used as a testbed [72, 73]. The hovercraft dynamics model was derived based on a ground-fixed coordinate system, as depicted in the right side of Figure 26. The hovercraft operates with two differential thrust fans with electrical motors and a LIDAR sensor for simultaneous localization and mapping.

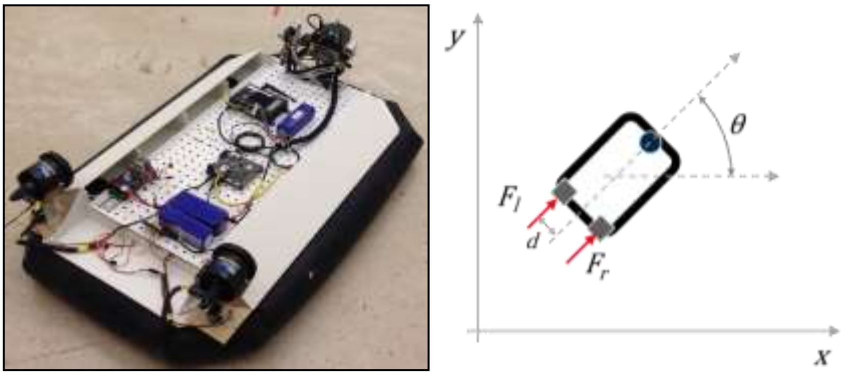


Figure 26. The autonomously operable hovercraft with two differential thrusts (left), and 2D hovercraft dynamics and kinematics representation (right).

9.1. Hovercraft Dynamics Model

The hovercraft is assumed to move in two-dimensional planar motion; thus, it is an under-actuated system given two input controls. Equations 19 are the system dynamics model; x and y are absolute positions on the ground fixed coordinate, θ is a heading angle, $\dot{X}$ is a velocity, $\ddot{X}$ is an acceleration, m is the mass, J is the moment of inertia of the hovercraft, d is the distance between a thruster and an imaginary longitudinal line crossing the mass center while assuming that the mass center coincides with the geometric center, and F_l & F_r are left and right thrust forces, respectively. Based on the dynamics derived, the state is $\boldsymbol{x} = \{x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}\}^T$, and the input is $\boldsymbol{u} = \{F_l, F_r\}^T$. Han, and Zhao [74] evaluated the underactuated hovercraft controllability. The analysis showed that the existance of the yaw torque can guarntee the system controllability. It implies that one thrust motor failure does not affect the controllability as long as the other motor can produce proper torque values. Table 1 shows the system properties used for the following experiments.

$$\begin{aligned}\ddot{x} &= -\frac{d_t}{m}\dot{x} + F_l \cdot \cos\theta + F_r \cdot \cos\theta \\ \ddot{y} &= -\frac{d_t}{m}\dot{y} + F_l \cdot \sin\theta + F_r \cdot \sin\theta \\ \ddot{\theta} &= -\frac{d_r}{J}\dot{\theta} + d(F_r - F_l)\end{aligned} \tag{22}$$

Table 1. System properties

Parameters	**Values**	**Description**
m (kg)	11.8	Vehicle mass
J $(kg \cdot m^2)$	1	Moment of Inertia
d (m)	0.25	Moment arm
d_t (-)	0.05	Frictional damping (translation)
d_r (-)	0.005	Frictional damping (rotation)
F_{max} (N)	2	Control input constraint (max.)
F_{min} (N)	-2	Control input constraint (min.)

9.2. Fault Growth Model

A fault growth dynamics model is given as a function of time and actuator control inputs as:

$$\dot{\sigma}(t) = \rho_\sigma \cdot u_{?}^2 + \sigma_0 + \omega_\sigma(t) \tag{22}$$

where σ is the state of a fault on the right thrust motor, $\omega_\sigma(t)$ is noise, ρ_σ is a coefficient representing how fast a fault grows with respect to an actuator control input, and σ_0 is a control-independent parameter. As a dimensionless representation, the fault severity is ranked from 1 to 10 with one as a healthy condition and 10 as an indication of a component failure. At the fault severity 10, the motor control thrust force is no longer active. For simplicity, the impact of the fault mode on the effective thrust force is assumed to be inversely proportional to the severity of the fault, as shown in Equation (13).

$$F_{\text{actual}} = {}^{F_{\text{desired}}}\!/_{\sigma} \tag{13}$$

Table 2 shows the fault growth model parameters that were used in the following tests.

Table 2. Parameters for the fault growth model

Parameters	Values	Description
ρ_σ	2.5	Control input effect coefficient.
σ_0	0.02	Operational time effect coefficient.

9.3. Energy Consumption Model

As a system resource, an electric energy consumption rate is modeled as a function of control inputs.

$$\dot{e}(t) = \rho_e \cdot \boldsymbol{u}(t)^T \boldsymbol{u}(t) + \omega_e(t) \tag{24}$$

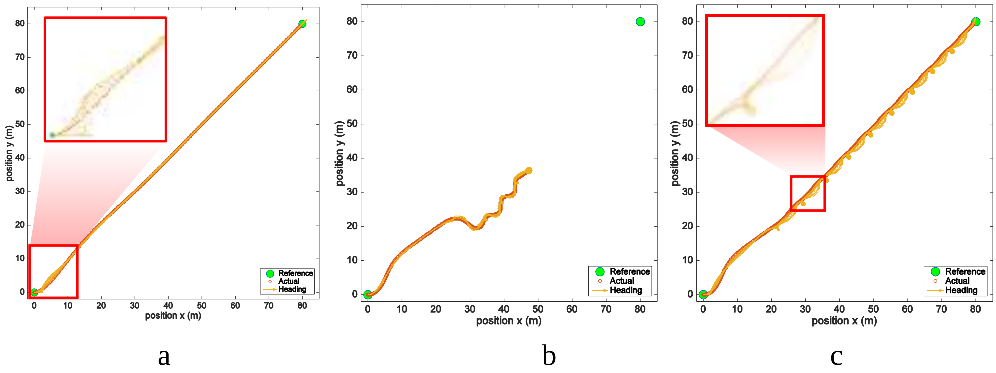


Figure 27. Hovercraft position trajectory result comparisons: (a) healthy condition; (b) nominal controller under faulty condition; and (c) reconfigurable controller under faulty condition.

Notionally, the maximum energy available is set to 300 (dimensionless). If the total consumption reaches its maximum value, it is impossible to move the hovercraft any longer; thus, is becomes uncontrollable. For the experiment, ρ_e was set to 10.

9.4. Simulation Results and Discussion

The hovercraft test mission is to move from a starting point, (0, 0), to a target point, (80, 80). A fault occurs in the right thrust motor during the operation initiated at 50 sec., and its severity monotonically increases as modeled in Equation (23). Figure 27 depicts the hovercraft position and heading trajectories (a) under healthy condition, (b) with the nominal controller under faulty condition, and (c) with the proposed reconfigurable controller under faulty condition. In the reconfigurable controller, ρ_f and ρ_R were set to 100. As expected, the nominal controller could not handle such an extreme fault and could not get to the target point at the end. With the reconfigurable controller, on the contrary, the hovercraft reached the target, but it exhibited an oscillatory behavior in the middle of the operation. This behavior is attributed to the redistributed control authority. As illustrated in Figure 28, the reconfigurable controller endowed agility characteristics to the healthy thrust motor while suppressing the usage of the faulty one. At $t_{k+\delta}$, the actual force exerted from the right faulty thrustor was less than the left healthy thrustor. Instead of exerting more effort on the faulty thrustor, the controller forced to turn the vehicle right until the heading angle pointed backward, and then produced a reverse thrust on the left motor to turn the vehicle heading back to the forward direction as well as to proceed in the direction of the target point. This control strategy repeated until the hovercraft reached the given target. Figure 29 depicts the arsenal of unmanned autonomous systems available in our laboratory.

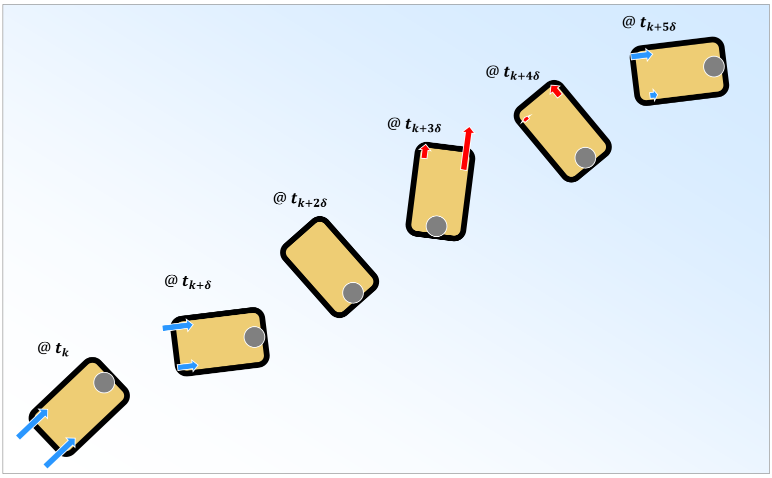


Figure 28. Pictorial representation of the redistributed optimal control sequence.

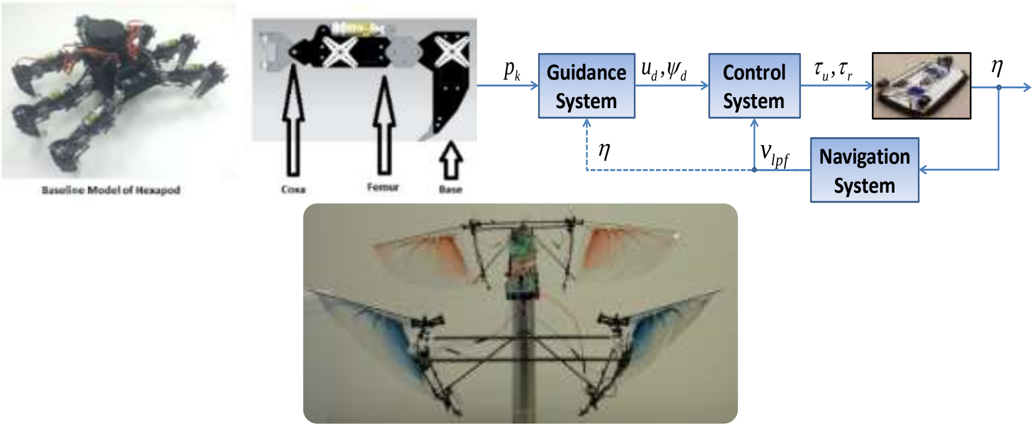


Figure 29. Laboratory Testbeds.

10. Simulation Results

The time-evolution of the turn-to-turn winding faults for different operating conditions were simulated in Simulink using (21) with specified modeling parameters and is shown in Figure 30. The RUL estimates were generated for different motor currents. The initial fault condition was set to $L_{plr} = 0.05$ for each instance. The expected RUL is computed for each operating condition. Notice, as the operating current decreases, the estimated RUL increases.

The expected RUL is inversely proportional to the magnitude of the operating current. Thus, the RUL can be extended by reducing the operating current. The MPC controller discussed earlier takes advantage of this relationship by reducing the operating current magnitude based on the RUL requirement. The degree of relaxation is dependent on the weight matrices chosen during the controller design phase. To demonstrate the feasibility of the approach the MPC toolbox in MATLAB was used to expedite the design process. In addition, each constraint has an associated cost defined appropriately. Also, the motor was simulated using ta 5th order actuator model. Results for three different fault scenarios were generated using the MPC with given control parameters and their corresponding boundaries and weights were defined. The results are provided in Figure 30. Notice, as the RUL reduces (left-to-right), the MPC places more emphasis on reducing the magnitude of the motor current. As a consequence, the rise time of the actuator position increases and the magnitude of the winding temperature decreases thereby increasing the estimated RUL.

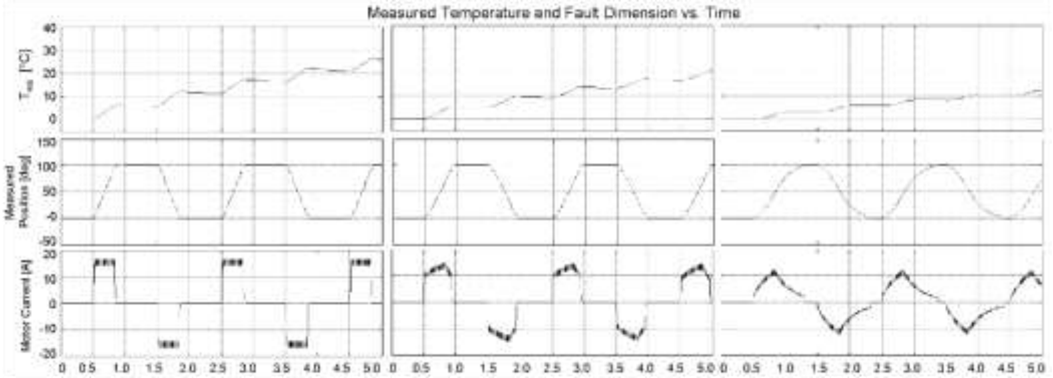


Figure 30. Simulation results for the reconfigurable control with (a) *Lplr* = 10% (b) *Lplr* = 5% and (c) *Lplr* = 1%.

Figure 31 shows the result of a vehicle moving from location [16.5 1.3] to goal [8.5 7.5]. At the waypoint A, a new goal is obtained and the robot needs to move.

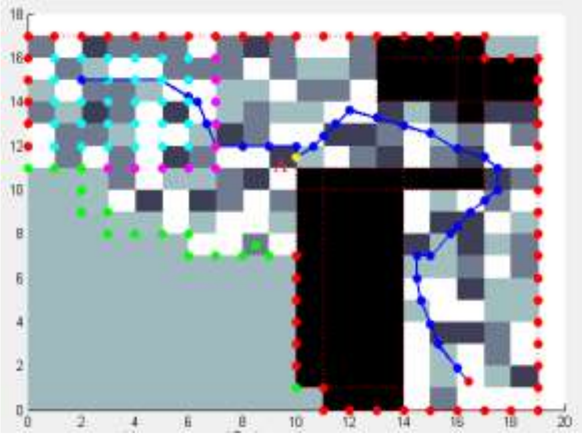


Figure 31. Mission adaptation results.

The cost function in the optimization routine is a multi-objective one with travel time, terrain, and fault constraints. From this figure, it is clear that the vehicle is able to accomplish tasks in an unexplored non-uniform area with mission reconfiguration. The map is divided into square grids with unit length. The terrain is indicated by colors: the darker the color, the more difficult to traverse (grids with black color indicate obstacles). The robot is able to observe four grids.

CONCLUSION

This chapter introduces a novel methodology for the resilient design and operation of complex Cyber Physical Systems (CPS) (unmanned autonomous systems, aerospace and space systems, manufacturing processes, etc.) with emphasis on unmanned systems for development, testing and validation purposes. There is a need to design and operate unmanned systems capable of withstanding severe disturbances that may endanger the integrity of the vehicle. The proposed framework for autonomous systems is founded on rigorous and verifiable technologies for endowing such critical assets with capabilities that go beyond the “normal” operating regime and posses the ability to perform missions in the presense of extreme hazardous environments. Achieving these gains will require developing new and innovative methods and tools to establish *assured and trusted autonomy* through integrated system health management, resilient design and operation of UAVs and swarms of vehicles, adaptive vehicle control, enabling complex systems to operate across a range of functional capabilities. The foundations of the proposed design for resilience build upon lessons learned from early successes/failures of the interplay between life sciences and complex engineered systems, and rely upon characteristic attributes of the biological world such as *immunity* and *self-healing* to withstand and absorb severe disturbances. The proposed self-organization method combines the Markov

Decision Process with resilience for a complex system to maintain functionality under a severe failure mode. The proposed method demonstrated its usefulness with the ability to fulfill system purpose in the test case of locked joint failure applied to a hexapod. Future work will be toward improving the self-organization method through deeper analysis in resilience metrics, focusing on the vulnerability and recoverability of systems under failure modes.

The proposed method demonstrated its usefulness with highly reduced computational burden in the test case applied to a hexapod under locked joint failure compared to traditional disturbance rejection methods, while the system maintains stability conditions. Future work will be toward improving the self-organization method through deeper analysis in resilience, focusing on the vulnerability and recoverability of systems under failure modes.

The proposed self-organization method combines the Markov Decision Process with resilience for a complex system to maintain functionality under a severe failure mode. The proposed method demonstrated its usefulness with the ability to fulfill system purpose in the test case of locked joint failure applied to a hexapod. Future work will be toward improving the self-organization method through deeper analysis in resilience metrics, focusing on the vulnerability and recoverability of systems under failure modes.

The self-organization method proposed in this paper combines the Markov Decision Process with Lyapunov stability conditions for a complex system to maintain stability under a severe failure mode. The proposed method demonstrated its usefulness with highly reduced computational burden in the test case applied to a hexapod under locked joint failure compared to traditional disturbance rejection methods, while the system maintains stability conditions. Future work will be toward improving the self-organization method through deeper analysis in resilience, focusing on the vulnerability and recoverability of systems under failure modes.

Fault-tolerant and reconfigurable control strategies for improved critical system reliability and survivability under fault/failure conditions has attracted the attention of the controls community in recent years. To apply these technologies, it is essential the system health status be monitored continuously and incipient failures be tracked so that remedial action can be taken as soon as possible to assure its safety. Control reconfiguration at the component level, constitutes the first level of the hierarchical framework for fault-tolerance. The reconfigurable control strategy is implemented at the second level and the mission adaptation strategy occupies the highest level of the control hierarchy. Finally, complexity issues must be addressed for specific application domains. Other modules of the integrated fault-tolerant control hierarchy, such as the control re- distribution, mission adaptation, etc., are not addressed in this paper but they contribute significantly towards the development of high-confidence systems. Future work must also focus on the development and implementation of rigorous uncertainty representation and management tools/methods and techniques for Verification and Validation (V&V) for system qualification.

References

[1] Darby, R., "Commercial jet hull losses, fatalities rose sharply in 2005," *Datalink*, pp. 51–53, August 2006.

[2] Wald, M., "Fatal airplane crashes drop 65%," *The New York Times*, October 1 2007.

[3] Kanki, B. G., "Aircraft maintenance research: The NASA program," in *Proceedings of the Human Factors and Ergonomics Society*, 2000, pp. (3)771–774.

[4] Hess, A., (2005, October 7-10,) Prognostics and health management lead. Keynote Speaker at the *Integrated System Health Engineering and Management Conference*. Joint Strike Fighter Program Office. Napa Valley, CA.

[5] Srivastava, A. N., Mah, R. W., and Meyer, C., "Integrated vehicle health management – automated detection, diagnosis, prognosis to enable mitigation of adverse events during flight," *National Aeronautics and Space Administration, Technical Plan Version 2.02*, December 2008.

[6] Vachtsevanos, G., Tang, L., Drozeski, G., and Gutierrez, L., "From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools," *Annual Reviews in Control*, vol. 29, no. 1, pp. 101–115, April 2005.

[7] Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., and Wu, B., *Intelligent Fault Diagnosis and Prognosis for Engineering Systems.* Hoboken, NJ, USA: John Wiley & Sons, 2006, ISBN 987-0-0471-72999-0.

[8] Filippetti, F., Franceschini, G., Tassoni, C., and Vas, P., "Recent developments of induction motor drives fault diagnosis using ai techniques," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 994–1004, October 2000.

[9] Kleer J. D., and Williams, B. C., "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, 1987.

[10] Skormin, V. A., Apone, J., and Dunphy, J.J., "On-line diagnostics of a self-contained flight actuator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 186–196, January 1994.

[11] Willsky, A. S., "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[12] Wu, B., Saxena, A., Khawaja, T., and Sparis, P., "An approach to fault diagnosis of helicopter planetary gears," in *IEEE Autotestcon*, San Antonio, TX, USA, September 20-23, 2004.

[13] Jiang J., and Zhao, Q., "Should we use parameter estimation or state estimation bated methods for fdi," in *Proceedings of the international federation of automatic control on SAFEPROCESS,* Hull, UK, August 1997, pp. 474–479.

[14] Patterson, R. J., "Fault-tolerant control: the 1997 situation," in *Proceedings of the International Federation of Automatic Control symposium on SAFEPROCESS,* Hull, UK, August 1997, pp. 1033–1055.

[15] Wu, N. E., Zhang, Y. M., and Zhou, K., "Detection, estimation and accommodation of loss of control effectiveness," *International Journal of Adaptive Control and Signal Processing,* vol. 14, no. 7, pp. 948–956, 2000.

[16] Zhang Y. M., and Jiang, J., "An active fault-tolerant control system against partial actuator failures," in *IEE Proceedings - Control Theory and Applications,* 2002, pp. 95–104.

[17] Orchard, M., *A particle filtering-based framework for on-line fault diagnosis and failure prognosis*, Ph.D. dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA, November 2007.

[18] Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., and Vachtsevanos, G., "Advances in uncertainty representation and management for particle filtering applied to prognostics," in *1st International Conference on Prognostics and Health Management (PHM),* Denver, CO, USA, October 6-9, 2008.

[19] Orchard, M., Wu, B., & Vachtsevanos, G. (2005). A particle filtering framework for failure prognosis. *Proceedings of World Tribology Congress III.* Washington DC.

[20] Orchard, M., & Vachtsevanos, G. (2007). A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine. *26th American Control Conference*. New York.

[21] Saha, B., & Vachtsevanos, G. (2006). A novel model-based reasoning approach to system-level diagnostics of a helicopter intermediate gearbox. *Proceedings of the 60th Meeting of the Society for Machinery Failure Prevention Technology*, (pp. 281-290).

[22] Hollnagel, E., Woods, D. D., & Leveson, N., (2007). *Resilience engineering: Concepts and precepts*: Ashgate Publishing, Ltd.

[23] Patron, P., Miguelanez, E., Petillot, Y. R., Lane, D. M., & Salvi, J., (2008). Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles. *OCEANS 2008, IEEE*, pp. 1–9. doi:10.1109/OCEANS.2008.5151975.

[24] Lewin, R. (1992, 1999), *Complexity: Life at The Edge of Chaos,* (2nd edition), Chicago: The University of Chicago Press.

[25] Ge, J., Kacprzynski, G. J., Roemer, M. J., & Vachtsevanos, G., (2004). Automated contingency management design for UAVs. *AIAA 1st Intelligent Systems Technical Conference*, pp. 20–22. doi:10.2514/6.2004-6464.

[26] Clements, N. S., (2003). *Fault tolerant control of complex dynamical systems*. Doctoral dissertation, Georgia Institute of Technology, Atlanta, USA.

[27] Ge, J., Kacprzynski, G. J., Roemer, M. J., & Vachtsevanos, G., (2004). Automated contingency management design for UAVs. *AIAA 1st Intelligent Systems Technical Conference*, pp. 20–22. doi:10.2514/6.2004-6464.

[28] Drozeski, G. R., Saha, B., & Vachtsevanos, G., (2005). A fault detection and reconfigurable control architecture for unmanned aerial vehicles. *Aerospace Conference, IEEE*. doi:10.1109/AERO.2005.1559597.

[29] Tang, L., Kacprzynski, G. J., Goebel, K., Saxena, A., Saha, B., & Vachtsevanos, G., (2008). Prognostics-enhanced automated contingency management for advanced autonomous systems. *Prognostics and Health Management, International Conference, IEEE*, pp. 1–9, IEEE. doi:10.1109/PHM.2008.4711448.

[30] Brown, D. W., Georgoulas, G., Bole, B., Pei, H. L., Orchard, M., Tang, L., Saha, B., Saxena, A., Goebel, K., & Vachtsevanos, G., (2009). Prognostics enhanced reconfigurable control of electro-mechanical actuators. *Annual conference of the prognostics and health management society*.

[31] Bole, B., Tang, L., Goebel, K., & Vachtsevanos, G., (2011). Adaptive load allocation for prognosis-based risk management. *Annual conference of the prognostics and health management society*, pp. 1–10.

[32] Clements, N. S., (2003). *Fault tolerant control of complex dynamical systems*. Doctoral dissertation, Georgia Institute of Technology, Atlanta, USA.

[33] Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A. and Wu, B., *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, John Wiley & Sons, Inc. 2006.

[34] David, R. A., and Nielsen, P., *Defense science board summer study on autonomy*. Defense Science Board Washington United States, 2016.

[35] Vinerbi L., Bondavalli A. & Lollini P. (2010). Emergence: A new Source of Failures in Complex Systems. *IEEE Third International Conference on Dependability* (133-138), July 18-25, DOI: 10.1109/DEPEND.2010.28.

[36] Prehofer C. & Bettstetter C. (2005), Self-organization in communication networks: principles and design paradigms, *IEEE Communications Magazine,* July 25, DOI: 10.1109/MCOM.2005.
1470824.

[37] Heylighen F. (1999), The Science of Self-Organization and Adaptivity, *The Encyclopedia of Life Support Systems,* Vol. 5 (No. 3), 253-280.

[38] Wilson R. (1996). *Introduction to Graph Theory*. Edinburgh Gate, Harlow, Essex CM20 2JE, England: Longman.

[39] Liu W., Sirisena H., Pawlikowski K. & McInnes A. (2009), Utility of algebraic connectivity metric in topology design of survivable networks, *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop,* 25-28 Oct. 2009, DOI: 10.1109/DRCN.2009.5340016.

[40] Butler S. K., (2008). Eigenvalues and Structures of Graphs. Doctoral dissertation. University of California, San Diego, http://orion.math.iastate.edu/butler/PDF/dissertation.pdf.

[41] Bellman R. (1957), A Markovian Decision Process, *Indiana University Mathematics Journal,* Vol. 6 (No. 4), 679-684.

[42] Yukalov V. I. & Sornette D. (2014), Self-organization in complex systems as decision making, *Adv. Complex Syst.*, 17 (2014) 1450016.

[43] Gabbai J., (2005). *Complexity and the Aerospace Industry: Understanding Emergence by Relating Structure to Performance using Multi-Agent Systems*. Doctoral dissertation. University of Manchester, http://gabbai.com/files/ J%20M%20E%20Gabbai%20EngD%20,Thesis.pdf.

[44] Cully A., Clune J., Tarapore D. & Mouret J. (2015), Robots that can adapt like animals, *Nature,* Vol. 521 (No. 7553), 503-507, doi: 10.1038/nature14422.

[45] Sorin M., Mircea N. & Viorel S. (2011), Hexapod robot: Mathematical support for modeling and control, System Theory, *Control, and Computing (ICSTCC), 2011 15th International Conference,* Oct 14-16.

[46] Barai R., Saha P. & Mandal A. (2013), SMART-HexBot: a Simulation, Modeling, Analysis and Research Tool for Hexapod Robot in Virtual Reality and Simulink, *AIR '13 Proceedings of Conference on Advances In Robotics*, doi>10.1145/2506095.
2506126.

[47] Yang J. (2003), Fault-tolerant gait generation for locked joint failures, Systems, Man and Cybernetics, 2003. *IEEE International Conference*, Oct. 8, DOI: 10.1109/ICSMC.2003.1244216.

[48] Cuaya-Simbro G. & Munoz-Melendez A. (2008), Adaptive Locomotion for a Hexagonal Hexapod Robot Based on a Hierarchical Markov Decision Process, *WSPC – Proceedings*, Vol. 0 (No. 12), June 2008.

[49] Chades I., Chapron G., Cros M. J., Garcia F., Sabbadin R. (2014). MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecograph*y 37:916-920.

[50] Jeong W., Kim H., Kim S. & Jun B. (2013), Path Tracking Controller Design of Hexapod Robot for Omni-directional Gaits, *Control Conference (ASCC),* 2013 9th Asian, 23-26 June 2013, DOI: 10.1109/ASCC.2013.6606206.

[51] "Bibliographical review on reconfigurable fault tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, March 2008.

[52] Eterno, J. S., Weiss, J., L., Looze, D. O., and Willsky, A. S., “Design issues for fault-restructurable aircraft control,” survey,” *Automatica,* vol. 25, no. 3, pp. 335–348, May 1989.

[53] Garcià, C. E., Prett, D. M., and M. Morari, M., “Model predictive control: Theory and practice – a survey,” *Automatica,* vol. 25, no. 3, pp. 335–348, May 1989.

[54] Monaco, J. F., and Bateman, A. J. D., “A retrofit architecture for model-based adaptive flight control,” in *AIAA 1st Intelligent Systems Technical Conference,* Chicago, IL, USA, September 20-22 2004.

[55] Yu W., and Harris, T. A., “A new stress-based fatigue life model for ball bearings,” *Tribology Transactions*, vol. 44, no. 1, pp. 11–18, 2001.

[56] Paris P., and. Erodogan, F., “A critical analysis of crack propagation laws,” *ASME Journal of Basic Engineering Transactions,* vol. 85, pp. 528–534, 1963.

[57] Paris, P. C., Gomez, M. P., and Anderson, W. E., “A rational analytic theory of fatigue,” *The Trend in Engineering,* vol. 13, no. 1, pp. 9–14, 1961.

[58] Tang, L., Kacprzynski, G., Goebel, K., Saxena, A., Saha, B., and Vachtsevanos, G., Prognostics-enhanced Automated Contingency Management for Advanced Autonomous Systems, *Prognostics and Health Management Conference,* 2008.

[59] Vachtsevanos, G., Tang, L., & Reimann, J. (2004). An intelligent approach to coordinated control of multiple unmanned aerial vehicles. *The American Helicopter Society 60th Annual Forum.* Baltimore, MD.

[60] Vachtsevanos, G., Tang, L., Drozeski, G., & Gutierrez, L. (2005, April). From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools. *Annual Reviews in Controls, 29*(1), 101-115.

[61] Orchard, M., Wu, B., & Vachtsevanos, G. (2005). A particle filtering framework for failure prognosis. *Proceedings of World Tribology Congress III.* Washington DC.

[62] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, and G. Vachtsevanos, *Advances in Uncertainty Representation and Management for Particle Filtering Applied to Prognosis*, K. P. Valavanis, Ed. Springer Netherlands, 2009, vol. 39, no. 1, iSBN 978-90-481-3017-7.

[63] Roemer, M., Byington, C., Kacprzynski, G., & Vachtsevanos, G. (2005). An overview of selected prognostic technologies with reference to an integrated PHM architecture. *Proceedings of NASA Integrated Vehicle Health Management Workshop.* Nappa, CA.

[64] Stentz, A., Optimal and efficient path planning for partially-known environments, *Proc. Intl Conf. Robotics and Automation:* 3310–3317, 1994.

[65] D. Ferguson, D., Stentz, A., Field D*: an interpolation-based path planner and replanner, *Proc. Intl. Symp. Robotics Research,* 2005.

[66] Ferguson, D., Stentz, A., The field D* algorithm for improved path planning and replanning in uniform and non-uniform cost environments, *CMU-RI-TR-05-19*, June, 2005.

[67] Koenig, S., Likhachev, M., Fast re-planning for navigation in unknown Terrain, *IEEE Transactions on Robotics and Automation,* 21(3), 354-363, 2005.

[68] J. Bellingham, Richards, A., How, J., Receding horizon control of autonomous aerial vehicles, *Proc. 2002 American Control Conference,* Anchorage, AK, May 2002.

[69] T. Schouwenaars, T., How, J., and Feron, E., receding horizon path planning with implicate safety guarantees, *Proc. 2004 American Control conference,* Boston MA June 2004.

[70] Kim, K., Lee, Y., Oh, S., Moroniti, D., Mavris, D., Vachtsevanos, G. J., Papamarkos, N., & Georgoulas, G., (2013). Guidance, navigation, and control of an unmanned hovercraft. *Control & Automation (MED), 2013 21st Mediterranean Conference, IEEE,* pp. 380–387. doi:10.1109/MED.2013.6608750.

[71] Sconyers, C., Lee, Y., Kim, K., Oh, S., Mavris, D., Oza, N., Mah, R., Martin, R., Raptis, I. A., & Vachtsevanos, G. J., (2013). Diagnosis of fault modes masked by control loops with an application to autonomous hovercraft systems. *International Journal of Prognostics and Health Management*.

[72] Han, B., & Zhao, G. L., (2004). Course-keeping control of underactuated hovercraft. *Journal of Marine Science and Application*, 3(1), 24-27. doi:10.1007/BF02918642.

[73] Roemer, M., Byington, C., Kacprzynski, G., Vachtsevanos, G., Goebel, K., Prognostics, Chapter 17, in *System Health Management: With Aerospace Applications*, Wiley, 31 May 2011

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8

Chapter 7

FAULT DETECTION OF NONLINEAR NETWORKED CONTROL SYSTEM BASED ON MULTIMODAL APPROACH SUBJECT TO INDUCED DELAY

***Ben Mabrouk Zaineb*[*], *Ben Hamed Mouna*,**
Abid Aicha and Lassad Sbita
National Engineering School of Gabes, Gabes, Tunisia

ABSTRACT

In this chapter, we present a fault detection filter for the induction motor speed as a class of nonlinear system in networked control systems (NCSs) subject to induced time delays. We used the multi-model approach for modeling of induction motor described by a set of linear models. Recent research shows that the multi-model approach is a powerful tool to deal with nonlinear system. Thus, we were interested particularly in electric machine, especially in induction motor as a strongly nonlinear system. The necessity to assure the induction motor safety operation implicates protective supervision process based on fault diagnosis techniques. The first focus of this chapter is to describe the induction motor via an interpolation of a set of linear local models. This representation require a strategy of four steps that are database acquisition, cluster estimation, structural and parametric identification and local models combination. Then, an adaptive state filter is presented which can provide the information of faults and states of induction motor. In reality, certain observations may be missing possibly due to network-induced delay, random packet dropout; access constraints, etc. Therefore, in this work, an approach is proposed to perform estimation in network-induced delay. The induced time delays are from the controller to the plant and from the sensor to the controller. An example is included to show the efficiency of the proposed method.

[*] Corresponding Author's Email: benmabroukza@gmail.com.

Keywords: networked control system, multimodal approach, time delay, induction motor

1. INTRODUCTION

The significant increase of technological developments of industrial systems is certainly accompanied by a growing complexity of these processes. Thus, it is interesting to maximize performance, increase skills, and guarantee the security and safety of personnel and equipment. These objectives have contributed to the development of new procedures and monitoring algorithms for the Faults Detection, Isolation and Identification (FDI). Among the most commonly used FDI techniques are those based on the generation of residues from a healthy model. These procedures include a step of failure symptoms generation or residues generation. The production of these residues requires the use of information from a suitable model to compare them with those provided by the measuring instruments. In addition, an initial and preliminary analysis phase is the modeling of these processes. Modeling objectives is to take into account both the complexity and nonlinearity of the system, to get a true representation of their behavior, on one hand and it has better to have a simple and easy handling model so to make the easiest possible diagnosis task, on the other hand. However, the linearity of the studies is a strong assumption that limits the pertinence of the results that can be obtained. Direct extension of the control methods and estimation developed in the context of linear models to the case of nonlinear models is delicate. Thus, a modeling approach as an alternative that overcomes these difficulties is the multi-model approach that many recent researches have a great interest in it (Didier et al., 2013; Men et al., 2014; Shankar et al., 2012).

The power of such an approach is that it would allow the exploitation of simple algorithms applied to linear models to complex real systems. The multi-model approach eases the complexity of nonlinear systems by representing it via a combination of a set of local linear models. Each local model contributes to the total representation through weighting functions.

In literature, there exist various multi-model categories. Particularly, the coupled state multi-model called Takagi-Sugeno models or decoupled state multi-model. Concerning the first structure, local models share the same state vector whereas the second structure is a decoupled local models representation with different state vectors. The Takagi-Sugeno multi-model structure is the most prevalent in the analysis and the synthesis of the multimodels (Latrach, et al., 2015; Bello, et al., 2014). Whereas,on the other hand, the works which concern the decoupled multimodel are less frequent. However, it is advisable to underline that works on the identification (Abid, et al., 2012; Venkat, et al., 2003; Domla, et al., 2011) and the control (Garcia-Nieto, et al., 2008) of nonlinear systems have already taken advantage of this structure and confirmed its efficiency. As for the state estimation, an approach to make the synthesis of a PI multi-observer, by using a decoupled multi-model was proposed in (Orjuela et al., 2008; Orjuela et al., 2010)

As the workhorse of industry, Induction motors occupy currently an important place in industrials drive especially based on variable speed. Thus, a growing interest is given to the implementation of a new and effective diagnosis process to ensure the safety of these engines (Khalid, et al., 2014; Samko, et al., 2013; Yang, et al., 2010: Hou et al., 2005). In this context, we propose in this work to detect and isolate the IM faults via multi-model approach. This approach is applied in many applications, however it is applied here for the first time on electric machine and it is as well experimentally validated.

The Networked Control Sytems (NCSs) are control systems in which the information is transmitted through a communication network. The networked control architecture has many advantages over a traditional point-to-point design, such as low cost of installation, ease of maintenance, lower cost and greater suppleness. Due to these reasons, the Networked Control architecture is already used in many applications, particularly where weight and volume are about consideration, for example in automobiles, robotic and tele-operation. However, since the information is diffused over the communication channel of limited bandwidth, there exist same constraints such as networked-induced delay (Gao and al. 2008), (MAO Ze-Hui and JIANG Bin 2007), (Yue and Han 2006), (X. Zhu, C. Hua and S. Wang 2008), (Zhang and al. 2007), quantization problems (Goodwin and al. 2004) and (Gao and al. 2007) and packet dropout (Jin and al. 2006), (Wu and T. W. Chen 2007), (Mao and Jiang 2007), (Wang and al. 2006) and (WANG Yong-Qiang and al. 2009). The different constraints may degrade the performance and stability of the closed–loop control system. Networked control architecture has also some problems due to the presence of a communication network. In this chapter, the main constraint is the presence of the random induced delays.

The objective in this chapter is to propose an algorithm of faults detection FDI for a class of nonlinear systems which is the induction motor in the networked control system that take into account the induced delay constraint.

Recently, the IM has been developed and applied in several science and engineering domains (Elfelly and al. 2008) and (Piguet 1999). The induction motor IM is presented by multi-model system. Both, the flux and the speed affect the IM torque. The variation of its parameters generally with temperature and the nonlinear behavior of the IM represent a great problem. To solve and cope with these problems, the multimodel approach has used. It consists in replacing the unique system by a set of simpler models.

The main contributions of this chapter are in three aspects:

- An adaptive state detection filter is proposed to estimate fault for a system in NCS subject to induced delay.
- The adaptative FDI is applied in order to exploit the improved IM model. The IM represented by a multiple model in presence of load disturbances and parameter variations. We use diagonal detection filter and a GLR test bank (Willsky and Jones 1976) to estimate a state and faults. The GLR test bank allows decoupling perfect detection signals (Keller 1999).
- The tracking ability of the Diagonal detection filter (Keller 2007) is improved from the updating strategy of Willsky and Jones (Willsky and Jones 1976). A simulation example is presented for validation of the methods studied.

In this chapter, four parts are studied. The first part presents an overview of multi-model approach. The second part treats the multi-model modeling strategy. The third concern the design of the Diagonal Detection Filter and the final part concern the diagnosis of IM faults.

2. General Information on Multi-Model Approach

The principle of this approach, inspired by the fuzzy representation, using the principle that divides to reign, is to describe the dynamic behavior of a nonlinear system by a set of local models, often linear characterizing the behavior of the system in different operating areas while decomposing its operating space into a finite number of clusters. The motivation of this decomposition based on the concept that it is often difficult to develop a comprehensive model that could account for all the features and the complexity of a system (Abid, et al., (2011); Mazinan et al., (2011); Orjuela et al., (2013); Yu, C. H et al., (2014)).

Generally, a multi-model is a combination of N local models; each one is weighted by a validity function. Each local model contributes to this global representation via a weighting function that has value in the interval [0, 1].

$$y_m = \sum_{i=1}^{N} V_i(k) y_i(k) \tag{1}$$

With yi is the output of ith local model usually with simple and linear structure. Vi is the weighting function related to the contribution of each local-model to the approximate representation of the overall system. Depending on the area where the system evolves, this function shows the contribution, more or less, the local model corresponding to the representation of the global model (multi-model). It ensures a gradual transition from model to neighboring local models. These functions can be constructed from several approaches such as with geometric approach, probability or residue. In general, they can be triangular, sigmoidal or Gaussian and must satisfy the conditions (2).

$$\begin{cases} \sum V_i = 1 \\ 0 \le V_i \le 1 \end{cases} \tag{2}$$

A judicious choice of the structure of local models and validity functions Vi allows, in theory, to approach, with an imposed precisely, any nonlinear behavior in a wide range of operation.

In the case of switching models, the weighting functions are Boolean functions that can only take 0 or 1; 1, in the case where the local model is valid, and 0, otherwise, so that at each moment, only one model is valid. This case is not the subject of this work.

2.1. Structure of the Multi-Model

Depending on the nature of coupling between local models, there are two main structures of multi-model: coupled states structure and decoupled states structure.

2.1.1. Coupled Structure

For this type of structure often called “multi-model Takagi-Sugeno”, the general representation of the system is obtained by interpolation of a set of local models which depend on each other and share a single state vector (Shu-Ping, et al., (2014); Angelov, et al., (2004); Chadli, et al., (2013); Ichalal, et al., (2014)). The output of the overall system is a mixture of the local model parameters:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{N} V_i (A_i x(t) + B_i u(t)) \\ y(t) = \sum_{i=1}^{N} \left(V_i C_i \right) x(t) \end{cases} \tag{3}$$

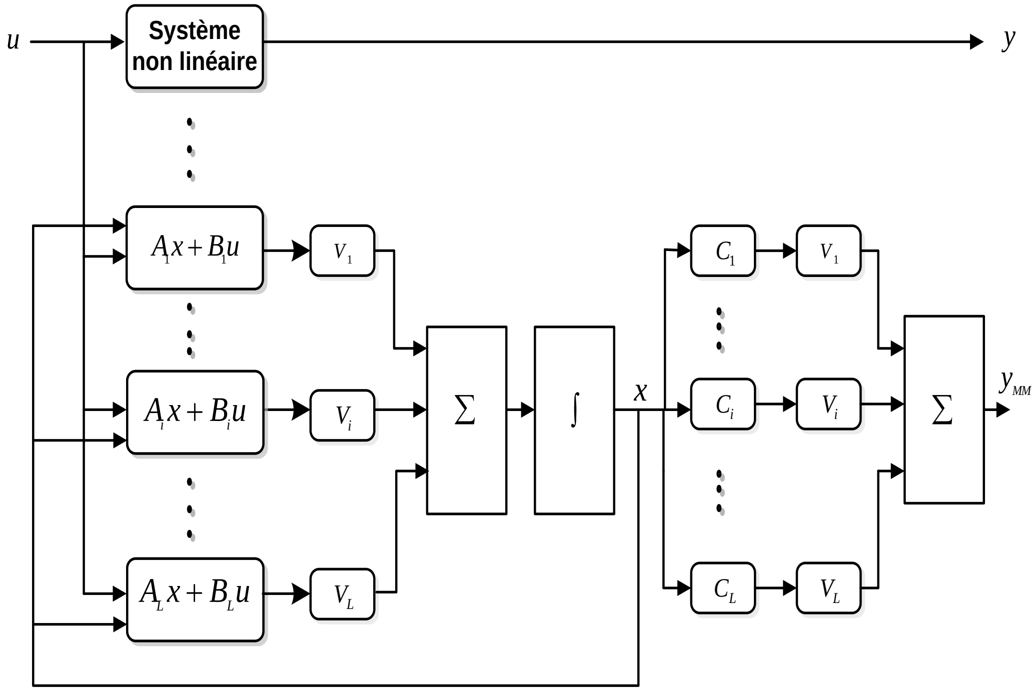


Figure 1. Coupled states multimodel.

2.1.2. Decoupled Structure

The difference between this structure and the coupled one lies in the fact that for this structure, each local model is independent on all others, and the final output of the multimodel is the weighted sum of the local models outputs. Indeed, the state space of each local model is independent, and its order may be different. The output of the overall model is a combination of the local models (Abid, et al., (2015)).

$$\begin{cases} \dot{x}_i(t) = \sum_{i=1}^{N} V_i(A_i x_i(t) + B_i u(t)) \\ y(t) = \sum_{i=1}^{N} (V_i C_i) x_i(t) \end{cases} \tag{4}$$

Vi functions, in this case, are the contributions of the outputs of each local-model for the formation of the global model, without mixing the models parameters Ai, Bi, Di and Ci.

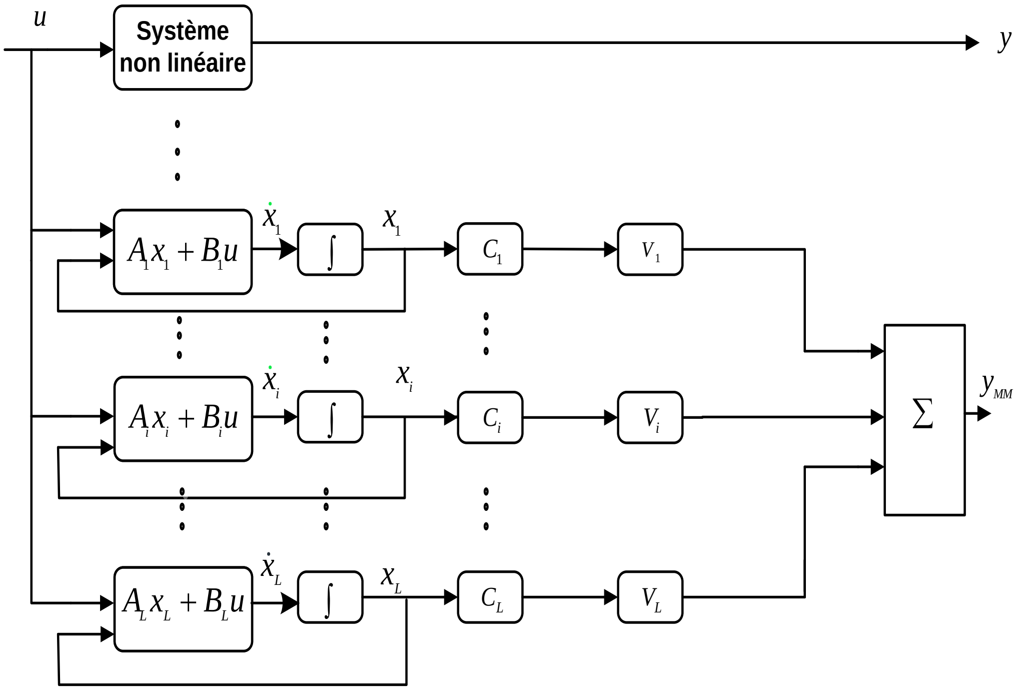


Figure 2. Decoupled states multimodel.

2.2. Principle of Multi-Model Modeling

2.2.1. Building Strategy of Local Models

The general principle of the multi-model approach is to represent the non-linear system in the form of a combination of contributions relating to a set of local models, by means of weighting functions. Each local model is a valid dynamic system in an operating area. Modeling task by multi-model approach requires the research of local models as well as the weighting functions.

Several methods allow us to obtain the local models. If the measures of inputs and outputs of the system are available, one can proceed by identifying, seeking or imposing multi-model structure. Then each local model is identified by parametric and conventional structural identification algorithms. However, if one has a non-linear model explicitly, that one wishes to “simplify” or make it more manipulable, we can proceed by linearization around the various points of operation. For high accuracy in the approximation of the global model by its various local models, it is recommended to simply adjust their numbers and the expressions of the various weighting functions (Elfelly et al., (2012)).

By having a rich input/output database collected on the real system, and then it is essential to choose the identification method for the determination of the local models, used to construct the multi-model.

3. Multimodel Modeling of Induction Motor

In this context, we will develop a systematic process for the representation of a nonlinear system with a multi-model.

Determining a base of simple models is a fundamental step in building a multimodel.

Currently, the conception of a multimodel requires a strategy of four problems to solve as shown in Figure 1, that are; the acquisition of a rich database, data classification, structural and parametric identification and fusion of local models.

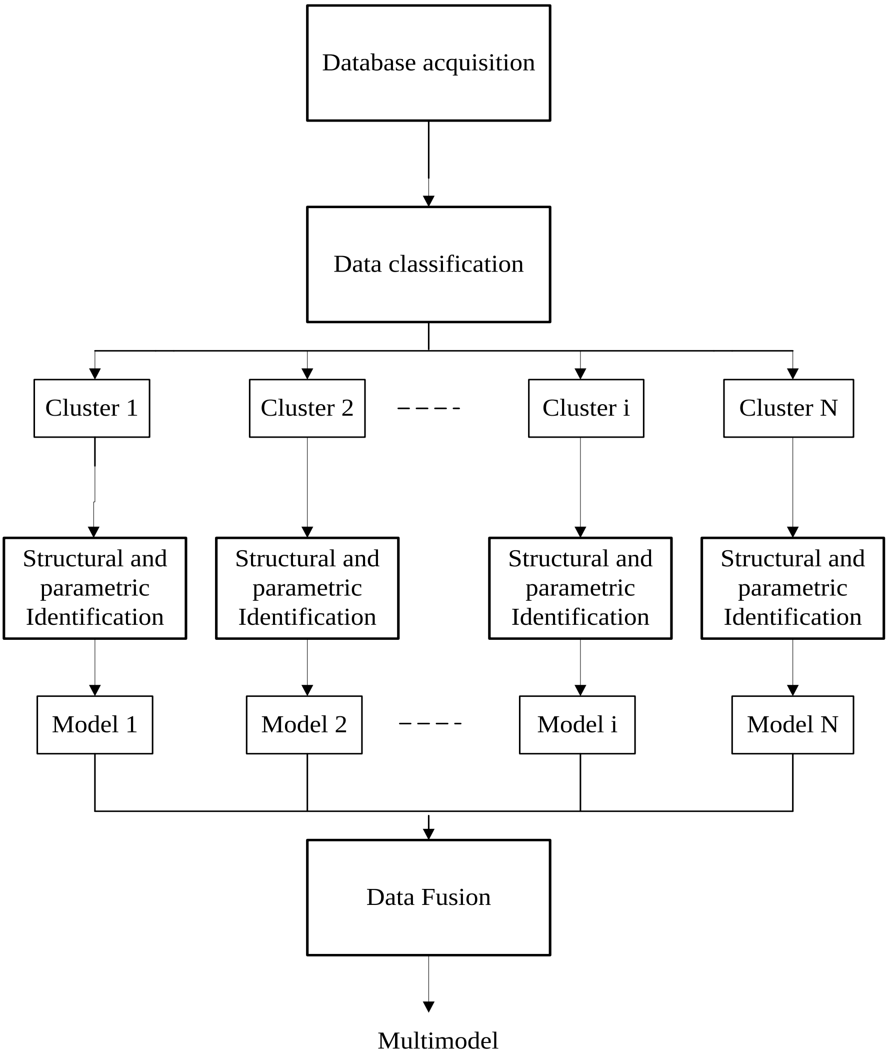

Database acquisition
Data classification
Cluster 1
Cluster 2
- - - -
Cluster i
Cluster N
Structural and parametric Identification
Structural and parametric Identification
Structural and parametric Identification
Structural and parametric Identification
Model 1
Model 2
- - - -
Model i
Model N
Data Fusion
Multimodel

Figure 3. Multimodel modeling strategy.

3.1. Database Acquisition

Having a rich inputs/outputs database collected from the real system is an initial step for the construction of a multimodel. Thus, a series of measurements of the input/output signals of the process are prepared using a data acquisition card.

In the following, we will develop the steps of adopted modeling strategy in a sequential manner.

3.2. Data Classification

The decomposition of the collected database of the nonlinear system into N groups is performed by segmentation or clustering.

The clustering is to assembly in the same group (or cluster), the data considered similar. Typically, the similarity between the data is estimated using a function calculating the distance between these data. Once this function defined distance, the database clustering is to minimize the distance between the elements of the same group.

For clustering a set of n data points yi, {i=1,…,N}, the used clustering algorithm is the "subtractive clustering". It consists in considering each data point as a potential center. The potential of data point pi is calculated as a function of the Euclidean distances to all the other data points (5).

$$p_i = \sum_{j=1}^{n} e^{-4\frac{\|y_i - y_j\|^2}{r_a^2}} \tag{5}$$

With ra is a positive factor that is in charge of controlling the decreasing ratio of the potential. Once every data potential has been calculated, the first cluster center is designated as the data point that has the highest potential. Let y_1^* be the first cluster center and p_1^* be its potential value. Next, the data point's potentials are recalculated as the expression (6).

$$p_i \leftarrow p_i - p_1^* e^{-4\frac{\|y_i - y_1\|^2}{r_b^2}} \tag{6}$$

The factor rb >0 is a radius describing the neighborhood that will have assessable diminutions in potential. It has to be rather greater then ra to evade attaining closely spaced cluster centers; thus rb = 1.5, ra is a decent choice (Chiu (1994); Casalino et al., (2014)).

Furthermore, the second cluster center is designated as the data point that has the highest remaining potential. In a similar way, the kith cluster center y_k^* has been computed. The potential of each data point is modified as follow:

$$p_i \leftarrow p_i - p_k^* e^{-4\frac{\left\| y_{pi} - y_{pk} \right\|^2}{r_b^{\ 2}}} \tag{7}$$

This process is repeated until the condition $p_k^* < \varepsilon p_1^*$ is attained. With ε is a positive amount.

Finally, the data points are sited in the cluster that has the nearest center. Thus, the collected database is split into N data sets.

3.3. Identification of Local Models

Having a set of clusters, our objective is to identify the associated models. To achieve this it is essential to go through a structural identification, for the determination of levels of local models, as well as a parametric identification (Abid et al., (2013)).

3.3.1. Structural Identification

The objective is to identify the order of each local model. Two structural identification approaches may be used. The first approach is the general procedure for the order estimation, which is to check for each model of order m (m = 1, ..., mmax), a criterion based on the resemblance between the real system output and the model output.

The second approach is based on using the instrumental determinants ratio (IDR) that consists on calculating both information matrices Qd and Qd + 1. By varying d, the exact order of the system is identified in the order d, associated with the first rapid increase of the report (8).

$$Q_d = \frac{1}{n_h} * \sum_{k=1}^{n_h} \left[u(k)\ u(k+1)\ u(\text{k-1}) \cdots u(\text{k-d+1})\ u(\text{k+d}) \right]^T \left[y(k+1)\ \text{u(k+1) ... y(k+d) u(k+d)} \right] \tag{8}$$

$$RDI = \left| \frac{\det(Q_d)}{\det(Q_{d+1})} \right| \tag{9}$$

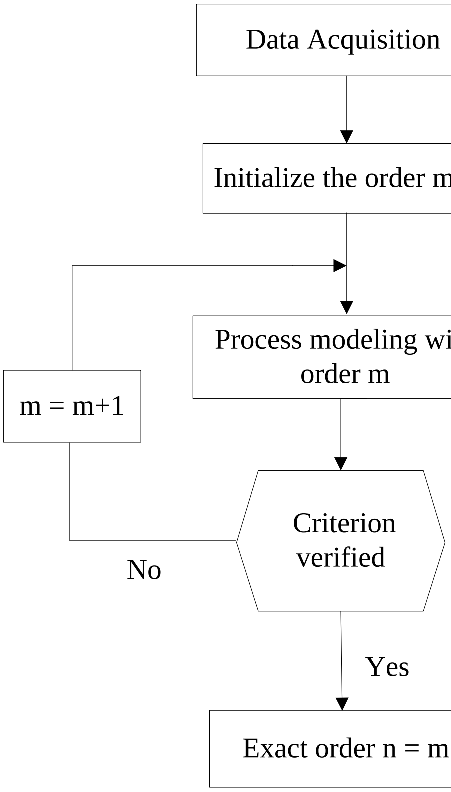


Figure 4. General procedure for order estimation.

3.3.2. Parametric Identification

For the estimation of model parameters, several numerical optimization methods can be used, according to information available a priori, such as the gradient method, Newton, Gauss-Newton, etc. These methods are generally based on the minimization of a function that computes the difference between the estimated output and the system output y (t). The criterion is that of the classic mean square that is based on the minimization of the squared difference between the two specified outputs. Furthermore, the recursive least square method will be used, whose objective is to estimate the parameters vector θ, from Inputs/outputs measurements.

A linear model is written in such form:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})} \text{ u(k-1)} \tag{10}$$

It can be written in the recurrent form as described in (11).

$$y(k) = -a_1\, y(k-1) - \cdots - a_n\, y(k-n) + b_1\, u(k-1) + \cdots + b_m\, u(k-m) \tag{11}$$

N: System order.

After having N observations measurements, we can rewrite the output y in matrix form

$$y = \theta^T * \Psi \tag{12}$$

where

$$\begin{cases} \theta^{\mathrm{T}} = [\, \mathrm{a}_1 \ \mathrm{a}_2 \ \cdots \ \mathrm{a}_{\mathrm{n}} \ \ \mathrm{b}_1 \ \mathrm{b}_1 \cdots \ \mathrm{b}_{\mathrm{m}} \ ] \\ \Psi^{\mathrm{T}} = [-y(k-1) \cdots - y(k-n) \ \mathrm{u(k\text{-}1)} \cdots \mathrm{u(k\text{-}m)}] \end{cases}$$

and

θ: parameters vector to be identified,
Ψ: data vector,
y: system output,

The prediction error is defined as the difference between the system output and the model output (13).

$$\varepsilon(k) = y(k) - \hat{y}(k) \ \ (13) \ \ \hat{y}(k) = \theta^T * \Psi(k) \tag{14}$$

where $\hat{\theta}^T$ is the parameters vector estimated via generalized recursive least square (15).

$$\begin{cases} \hat{\theta}_k = \hat{\theta}_{k-1} + K_k \varepsilon(k) \\ \varepsilon(k) = y(k) - \hat{\theta}_{k-1}{}^T \varphi(k) \\ K_k = \dfrac{P_{k-1}\varphi_k}{\lambda + \varphi_k^T P_{k-1}\varphi_k} \\ P_k = P_{k-1} - \dfrac{P_{k-1}\varphi_k \varphi_k{}^T P_{k-1}}{\lambda + \varphi_k^T P_{k-1}\varphi_k} \end{cases} \tag{15}$$

3.4. Fusion

Having built the local-models, the strategy of fusion consists in the combination of the latter each weighted via an activity function or validity as expressed in (16).

$$y_m = \sum_i V_i y_i \tag{16}$$

where,

ym: multi-model output.
Vi: validity of the ith local model.
yi: output of the ith local model.

3.4.1. Validity Computation

The calculation of the validity can be obtained by means of several approaches, such as the geometric, probability or residue approach. The first two approaches require knowledge of the properties of the system to be modeled, as well the local models for calculating the validity offline; on the other side, the residue computation is an interesting approach to determine the validity in line, since it requires only the knowledge of the system output and the locals models outputs.

3.4.2. Residual Approach

The residue is a function that calculates the error between the two values. In the case of the multi-model approach, the residue ri can be expressed by the error between the system output y and the local model output yi.

$$r_i = \left\| y - y_i \right\| \tag{17}$$

The normalized residue rn is expressed by (18).

$$r_{ni} = \frac{r_i}{\sum_{i=1}^{N} r_i} \tag{18}$$

ri: the residue at the ith local model,
N: number of local models.

Then we have:

$$\begin{cases} r_{ni} \in [0\ 1] \\ \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{r_{ni}} \ = 1 \end{cases} \tag{19}$$

where,

rni: normalized residue at the ith local model.

It is notable that, the more the value of the residue is big, the more the validity is low. Then, the validity can be expressed by the complementary value of the residue

$$v_i = 1 - r_{ni} \tag{20}$$

The vni is the normalized validity that is expressed by (21).

$$v_{ni} = \frac{1 - r_{ni}}{N - 1} \tag{21}$$

To benefit from the most valid model all the time, the validity vrni, defined in (23), assures, at the same time the reinforcement as well as the normalization.

$$v_{ri} = v_i \prod_{j=1,\, j \neq i}^{N} (1 - v_j) \tag{22}$$

$$v_{rni} = \frac{v_{ri}}{\sum_{i=1}^{N} v_{ri}} \tag{23}$$

Finally, the multi-model output can be expressed by (24).

$$y_m = \sum_{i=1}^{N} v_{rni} y_i \tag{24}$$

For improved performance, reduces the disturbance phenomenon due to inconceivability models. Reinforced validity can be calculated as in (25).

$$v_{ri} = v_i \prod_{j=1, j\neq i}^{N} (1 - \mathrm{e}^{-\left(\frac{r_j}{\sigma}\right)^2}) \tag{25}$$

σ is a positive constant,

Next, these validities are normalized using equation (23).

4. Case Study: Application of Multi-Model Approach to Modeling Induction Motor

We envisage in this part to test and to validate experimentally the modeling via multi-model approach. To do it, an experimental study is realized on an induction motor of 1kw.

In real time, the problem of modeling the real system is more delicate than that handled (treated) in simulation. Indeed, in addition to the coupling between its variables, which can be more particularly expressed in the dependent relationship between the torque simultaneously with the speed and flow, the engine is subject to various disturbances, such as the variation of its parameters, as well as the effect of the loads insertion.

In fact, the electrical machine, integrated in its environment, may undergo variations of its parameters, under the effect of multiple physical phenomena. An increase in temperature, for example, may cause changes in motor resistance.

The load driving can also be disruptive and detrimental factor that can cause degradation of engine operation. In fact, the variation of the load torque can generate a variation of the speed and flow of the machine.

Thus, we have been led to study the multi-model modeling of the speed and statoric currents of the (IM) in a point of functioning belonging to the low and medium speed around 600tr/min.

The experiment was carried out using Matlab/Simulink and dSpace control card containing analog/digital and digital/analog converters, allowing the sending and the acquisition of the input and output signals exchanged between the computer and the motor. An adequate experimental bench exposed in Figure 5, to validate the modeling approach is finely prepared.

To take into account the parameter variation, we propose to vary the stator resistances via three rheostats and the insertion of load through a resistive bank as more explained in the next section.

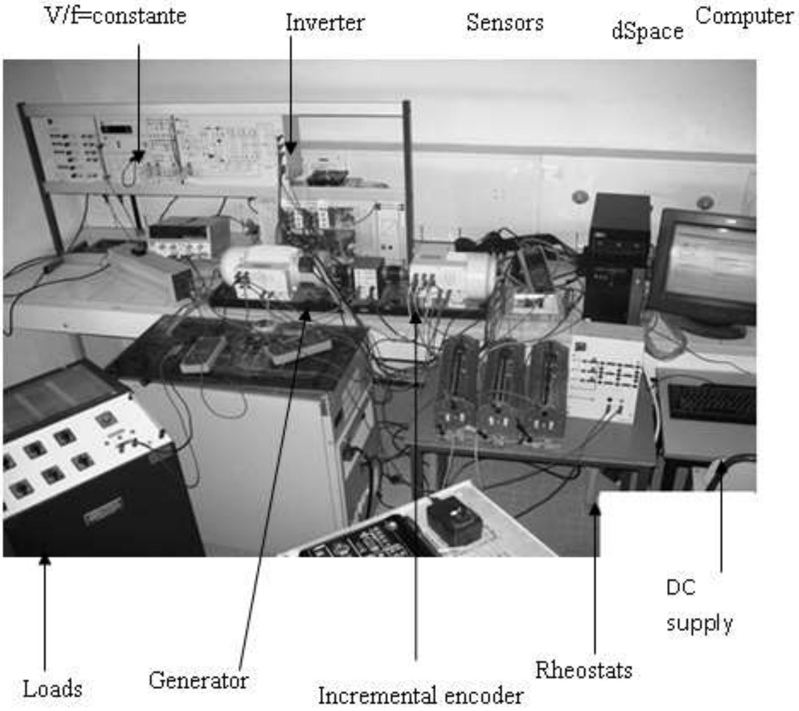


Figure 5. Experimental set-up.

Hence, we propose to implement the modeling strategy previously developed in a sequential manner.

4.1. Database Acquisition

Successive measurements of the two stator current, through Hall type sensors Hall type sensors LM LA 100 – P through 16bits analogical – digital converter to determine the average current is carried out. The speed measurement is performed by using an incremental encoder that generates 1024 pulses per revolution. The insertion of the load is produced using a DC generator mechanically coupled to the IM, feeding a resistive bank. A set of three rheostats, connected in series with each phase of the motor in order to cause the variation of the stator resistances. The performance of the cluster estimation depends on the quality of data base which must be rich in information.

Thanks to dSpace system with DS1104 controller board based on digital signal processors (DSP) a large database is obtained after a process of dispatching and acquisition of the input/outputs signals of the IM around the operating point around 600 rpm while performing parametric variations and successive insertions load at separate moments. This rich database is exploited in the classification phase and structural and parametric modeling of the different local models.

4.2. Database Clustering

The database classification by the proposed Chiu clustering algorithms engendered a set of eight clusters, shown in Figure 6 and Figure 7.

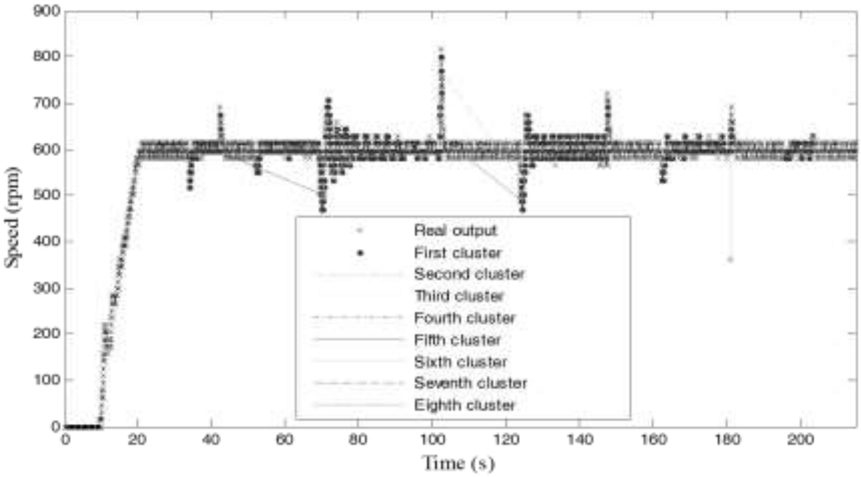


Figure 6. Speed database clustering.

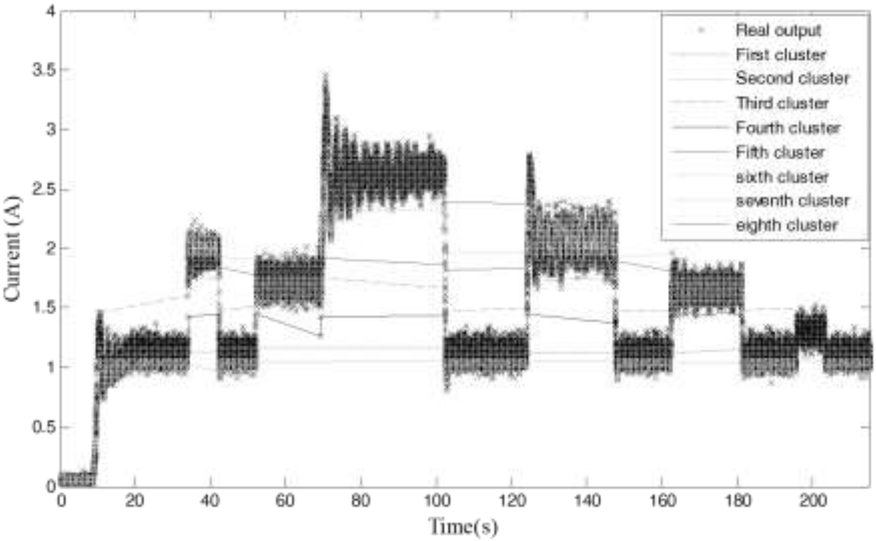


Figure 7. Current database clustering.

4.3. Models Identification

The obtained clusters will be identified to create the local models. Both the Instrumental Determinants Ration (IDR) and the general procedure for order estimation are exploited on each cluster to determine the cluster's order. Concerning the local models parameters identification, the application of the generalized recursive least square (RLS) method helps to generate eight local models that are defined via the following systems of recurrent equations (26)-(33).

First local model

$$\begin{cases} y_{11}(k) = q^{-1}\dfrac{0.107}{1+0.892q^{-1}}u(k) \\ y_{21}(k) = q^{-1}\dfrac{0.013}{1+0.930q^{-1}}u(k) \end{cases} \tag{26}$$

Second local model

$$\begin{cases} y_{12}(k) = q^{-1}\dfrac{0.084}{1+0.914q^{-1}}u(k) \\ y_{22}(k) = q^{-1}\dfrac{0.003}{1+0.978q^{-1}}u(k) \end{cases} \tag{27}$$

Third local model

$$\begin{cases} y_{13}(k) = q^{-1}\dfrac{0.1109}{1+0.889q^{-1}}u(k) \\ y_{23}(k) = q^{-1}\dfrac{0.223}{1+0.037q^{-1}+0.155q^{-2}}u(k) \end{cases} \tag{28}$$

Forth local model

$$\begin{cases} y_{14}(k) = q^{-1}\dfrac{0.0115}{1+0.7056q^{-1}+0.284q^{-2}}u(k) \\ y_{24}(k) = q^{-1}\dfrac{0.0127}{1+0.9387q^{-1}}u(k) \end{cases} \tag{29}$$

Fifth local model

$$\begin{cases} y_{15}(k) = q^{-1}\dfrac{0.0084}{1+0.9904q^{-1}}u(k) \\ y_{25}(k) = q^{-1}\dfrac{0.0811}{1+0.8153q^{-1}}u(k) \end{cases} \tag{30}$$

Sixth local model

$$\begin{cases} y_{16}(k) = q^{-1}\dfrac{0.1297}{1+0.6029q^{-1}+0.266q^{-2}}u(k) \\ y_{26}(k) = q^{-1}\dfrac{0.1558}{1+0.4268q^{-1}+0.1215q^{-2}}u(k) \end{cases} \tag{31}$$

Seventh local model

$$\begin{cases} y_{17}(k) = q^{-1}\dfrac{0.0901}{1+0.6524q^{-1}+0.2575q^{-2}}u(k) \\ y_{27}(k) = q^{-1}\dfrac{0.0328}{1+0.9285q^{-1}}u(k) \end{cases} \tag{32}$$

Eighth local model

$$\begin{cases} y_{18}(k) = q^{-1}\dfrac{0.0686}{1+0.6508q^{-1}+0.2808q^{-2}}u(k) \\ y_{28}(k) = q^{-1}\dfrac{0.1619}{1+0.4693q^{-1}}u(k) \end{cases} \tag{33}$$

4.4. Models Fusion

The obtained local models are combined based on fusion approach. The combination is gotten via the eights validities functions that are illustrated in Figure 8.

The obtained multimodel that is a result of the multi-model modeling approach is represented in Figure 9 and Figure 10. It is observable that the multimodel outputs follow the real ones.

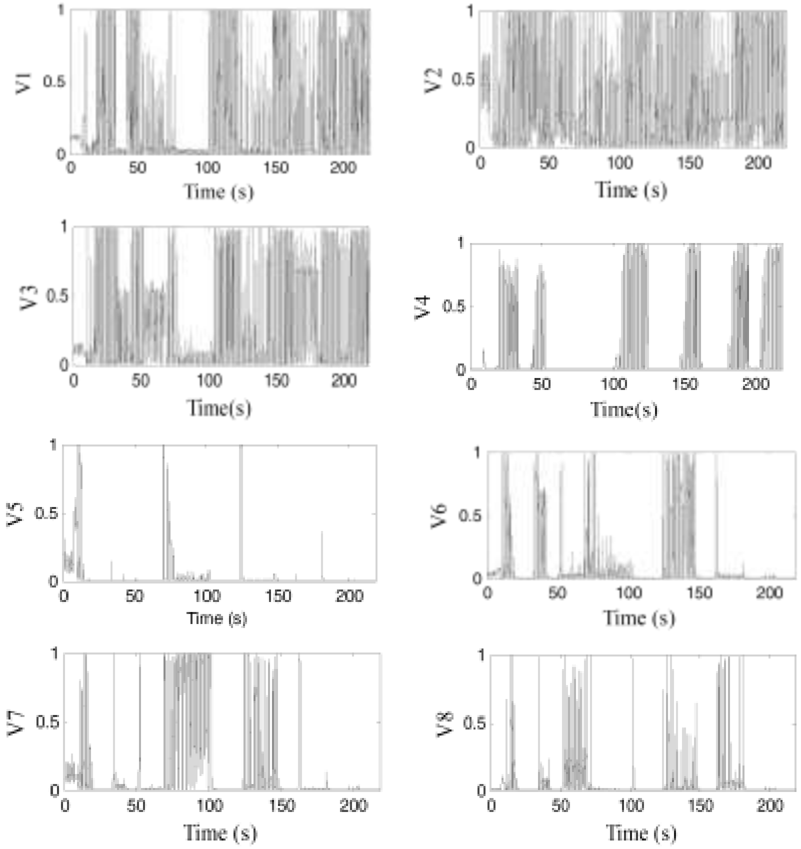


Figure 8. Validities functions.

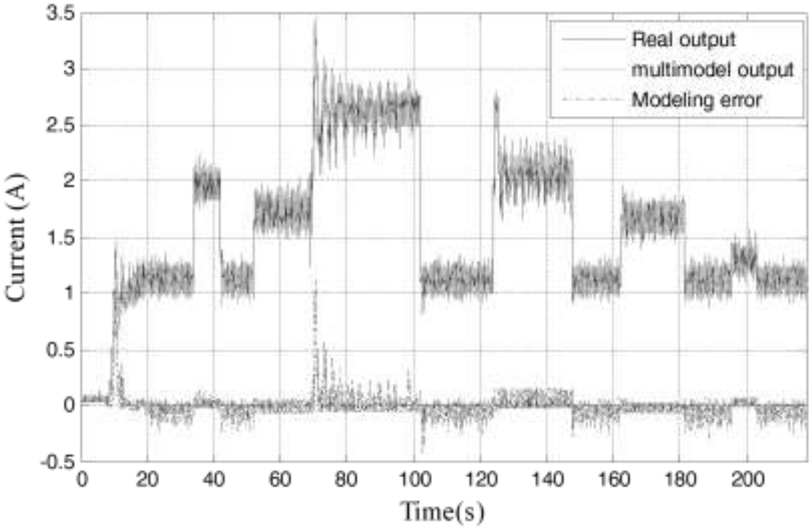


Figure 9. Real and multimodel output evolutions.

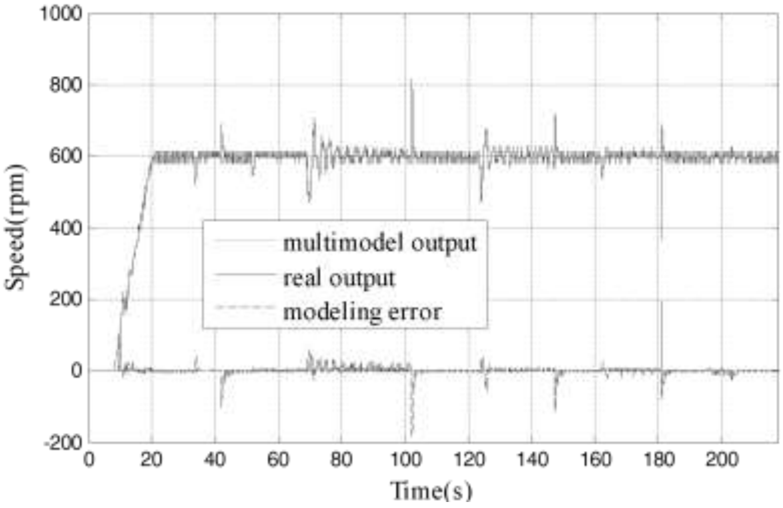


Figure 10. Real and multimodel output evolutions.

5. Networked Control System

The system is controlled by controller via communication channel. We have to take into consideration controller to actuator delays and the sensor to controller delays as present in Figure 11. In this chapter, one constraint due to the network has been incorporated: the induced time delays. It can be a source of degradation and instability in control performance (Xiong and Lam 2006) and (Besancon and al. 2007) in networked. In this paper, an NCS is assumed to be composed of plant, an actuator, a sensor and controller. The controller is closed to the actuator and sensor via networks (Chabir and al. 2007).

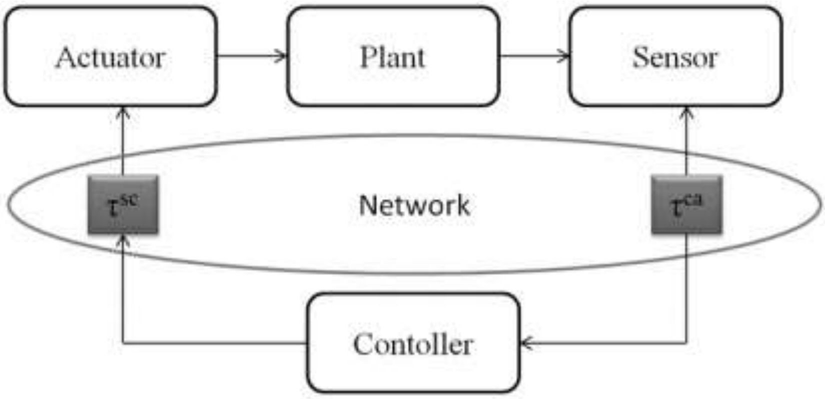


Figure 11. Schema of networked control systems with network induced delay.

In general, the delays can not be considered as known and constant. Network induced delays is related to the medium access protocol, network traffic and the hardware. Fault detection of Networked Control System (NCS) with random and unknown network induced delay that might be less than on sampling period is studied in this chapter.

$\tau sc + \tau ca < T$.

In this section, our focus is to derive a discrete time model of the system, considering the impact of the network induced delay. Consider the system depicted by the continuous time process (34):

$$\begin{cases} \dot{x}(t) = \Phi x(t) + \Gamma u(t) + \Psi d(t) + w_t \\ y(t) = Cx(t) + v_t \end{cases} \tag{34}$$

where $\boldsymbol{x}$ (t) ϵ R^n, $\boldsymbol{u}$ (t) ϵ R^m $\cdot$ $\boldsymbol{y}$ (t) ϵ R^r are respectively the state vector, the input vector, and the outputs vector. Φ, Γ, Ψ and C are the matrices with appropriate dimensions describing the system dynamics. $\boldsymbol{w_t}$ and $\boldsymbol{v_t}$ are white noise process. $d(t)$ ϵ R^q represents the vector of fault magnitudes and $\Psi = (f_1\ f_2 \dots f_q)$ ϵ $R^{n,q}$ the distribution matrix of component faults.

We can express the system (34) as the form (35) in the situation of NCS (Chabir and al. 2007) and (Ben Mabrouk and al. 2013).

$$\begin{cases} x_{k+1} = Ax_k + \Gamma_0 u_k + \Gamma_1 u_{k-1} + \Psi d_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{35}$$

With

$$A = e^{\Phi T}, F = \int_0^T e^{\Phi s}\Psi ds, B = \int_0^{T-\tau_k} \Gamma e^{\Phi s} ds, \Gamma_1 = \int_{T-\tau_k}^{T} e^{\Phi s}\Gamma ds \tag{36}$$

Equation set systems (35) leads to (37):

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + \Phi\Gamma\Delta u_k \tau_k + Fd_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{37}$$

With considering the following condition ((38) and (39)):

$$\Delta u_k = u_{k-1} - u_k, B = \int_0^T e^{\Phi s}\Gamma ds \tag{38}$$

$$\Gamma_1(\tau_k) = \int_{T-\tau_k}^{T} e^{\Phi s}\Gamma ds = \Phi^{-1}\left[1 - e^{\Phi \tau_k}\right]e^{\Phi T}\Gamma \approx \Phi\Gamma\tau_k \tag{39}$$

d_k is the fault vector $d_k = \left[d_k^1 \;\; .. \;\; d_k^i \;\; .. \;\; d_k^q\right]^T \in \Re^q$ with distribution matrix $F \in \Re^{n,q}$. The term $\Phi\Gamma\Delta u_k \tau_k$, deeds like an unknown input through the presence of the network. The term Fd_k can be considered as an unknown input. The next hypotheses can be written by equation (40)

$$H_0^i : d_k^i = 0 \quad H_1^i : d_k^i = \delta_{k,r_i}\nu^i \quad i = 1,..,q \tag{40}$$

With H_0^i represent the null hypothesis, H_1^i its alternative hypothesis and $d_k^i = \delta_{k,r_i}\nu^i$ the hypothesized fault of ν^i is unknown size and r_i is occurrence time. We adopt $rank(F) = q$ and $rank(CF) = q$. The $w_k \in \Re^n$ and, $v_k \in \Re^m$ are process and sensor noises with zero mean uncorrelated random sequences.

$$E\left\{\begin{bmatrix} w_k \\ v_k \end{bmatrix}\begin{bmatrix} w_j \\ v_j \end{bmatrix}^T\right\} = \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix}\delta_{k,j} \text{ with} W \geq 0 \tag{41}$$

The initial state x_0, supposed to be uncorrelated with w_k and v_k. It is a Gaussian random variable with $E\{x_0\} = \bar{x}_0$ and $P_0 = E\left\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\right\} > 0$. We use the transformed state $z_k = Tx_k$ with $T \in \Re^{n,n}$ non singular, where $\hat{A} = TAT^{-1}$, $\hat{C} = CT^{-1}$, $\hat{B} = TB$, $\hat{W} = TWT^T$ and $\hat{F} = TF$. The design of the adaptive filter detection is developed in the next section.

6. Diagnostic of Induction Motor in Network

In this section, we present the design of adaptive diagonal filter detection beneficial in the networked control system subject to induced delay. Diagonal filter detection will be proposed for fault diagnosis, which can provide the information of faults and states.

The diagonal detection filter must generate a residual. It must be also sensitive to simultaneous faults and able to differentiate faults from the other unknown disturbance inputs such as network effects. We look up, in this paper, on time delays induced by network. The residuals must be near to zero in faults free condition while in the presence of faults, the residuals deviated from zero.

We us the following state observer

$$\hat{z}_{k+1/k} = \hat{A}\hat{z}_{k/k-1} + \hat{B}u_k + \hat{D}_k(y_k - \hat{C}\hat{z}_{k/k-1}) \tag{42}$$

$$\hat{y}_k = \hat{C}\hat{x}_k \tag{43}$$

The residual generator is introduced by:

$$\gamma_k = H_k(y_k - \hat{C}\hat{z}_{k/k-1}) \tag{44}$$

where H_k and $\hat{D}_k$ are gains that are designed in order to satisfy isolation requirements and fault detection.

From (42), (43) and (44), the output of the filter γ_k and the state estimation error $e_k = z_k - \hat{z}_k$ propagate as:

$$\begin{cases} e_{k+1} = (\hat{A} - \hat{D}_k\hat{C})e_k - \Phi\Gamma\Delta u_k\tau_k + \hat{F}d_k + w_k - \hat{D}_k v_k \\ \gamma_k = H_k(\hat{C}e_k + v_k) \end{cases} \tag{45}$$

The next theorem is presented to design $\hat{D}_k$ and H_k, we used the following filter, which confirms the isolation of multiples faults.

$$\hat{z}_{k+1/k} = \hat{A}\hat{z}_{k/k-1} + \hat{B}u_k + \hat{D}_k(y_k - \hat{C}\hat{z}_{k/k-1}) \tag{46}$$

$$\hat{P}_{k+1/k} = (\hat{A} - \hat{D}_k\hat{C})\hat{P}_{k/k-1}(\hat{A} - \hat{D}_k\hat{C})^T + \hat{D}_k\hat{D}_k^T + \hat{W} \tag{47}$$

with $\hat{z}_{k+1/k}$ is the prediction of z_k and $\hat{P}_{k+1/k} = E\left\{(z_{k+1} - \hat{z}_{k+1/k})(z_{k+1} - \hat{z}_{k+1/k})^T\right\}$ is the state prediction error covariance matrix. The diagonal detection filter will be found by solving the constrained optimization problem taking into account the effect of the time delay, similarly to the problem of the optimization methods (Keller 1999),

$$\min_{\hat{D}_k,\Lambda_k} \hat{\Phi}_k = tr(\hat{P}_{k+1/k}) \tag{48}$$

under $$(\hat{A}-\hat{D}_k\hat{C})\hat{F} = \hat{F}\Lambda_k \tag{49}$$

with $\hat{F} = \begin{bmatrix} \hat{f}_1 & .. & \hat{f}_i & .. & \hat{f}_q \end{bmatrix}$ and

$$\Lambda_k = diag(\begin{bmatrix} \lambda_k^1 & .. & \lambda_k^i & .. & \lambda_k^q \end{bmatrix}) \tag{50}$$

The result of (46) will products a set $S = \left\{\tilde{\gamma}_k^1,..,\tilde{\gamma}_k^i,..,\tilde{\gamma}_k^q\right\}$ of scalar detection signals satisfying the diagonal detection filter and whiteness properties:

$\hat{f}_i$ is the eigenvector associated to the eigenvalue λ_k^i .

- $\tilde{\gamma}_k^i$ sensitive to d_k^i and decoupled from

$$\left\{d_k^1,..,d_k^{i-1},d_k^{i-1},..,d_k^q\right\} \tag{51}$$

$$\text{-} E\left\{\tilde{\gamma}_j^i\tilde{\gamma}_k^i\right\} = \hat{V}_k^i\delta_{k,j} \tag{52}$$

geometrical (52) and the statistical properties (51) will be exploited to solve the hypotheses testing problem (40) in parallel for $i = 1,..,q$ from a bank of GLR detectors, the ith GLR detector of the bank designed on the ith detection signal $\tilde{\gamma}_k^i$. Under $z_k = Tx_k$ determined so that the optimized value $\hat{\lambda}_k^i$ of λ_k^i be decoupled from $\left\{\hat{\lambda}_k^1 \;\; .. \;\; \hat{\lambda}_k^{i-1}\hat{\lambda}_k^{i+1} \;\; .. \;\; \hat{\lambda}_k^q\right\}$, GLR detector of the bank will work individually from each other as explained in the next algorithm:

1. Detect one fault on $\tilde{\gamma}_k^i$,
2. Update the diagonal detection Filter with $\hat{\lambda}_k^i = 0$, When a fault is detected on $\tilde{\gamma}_k^i$,
3. At time $k+1$, go to step 1 for the treatment of another fault.

From an augmented state model of the system, the above algorithm is used to derive an adaptive state filter for discrete-time systems subject to multiple abrupt changes.

The gain $\hat{D}_k$ result of $(\hat{A}-\hat{D}_k\hat{C})\hat{F}=\hat{F}\Lambda_k$ can be be parameterized from the reduced gain $\tilde{D}_k \in \Re^{n,m-q}$ as

$$\hat{D}_k = (\hat{A}\hat{F} - \hat{F}\Lambda_k)H + \tilde{D}_k\Sigma \tag{53}$$

The structure of the gain (53) implicitly generates two reduced output sequences $\tilde{\bar{\gamma}}_k \in \Re^q$ and $\bar{\gamma}_k \in \Re^{m-q}$ resulting from the output transformation where $\Sigma = \beta(I - \hat{C}\hat{F}H)$ $H = (\hat{C}\hat{F})^+$, and $\beta \in \Re^{m-q,m}$ so that $rank(\Sigma) = m - q$.

$$\begin{bmatrix} \tilde{\bar{\gamma}}_k \\ \bar{\gamma}_k \end{bmatrix} = \begin{bmatrix} H \\ \Sigma \end{bmatrix} (y_k - \hat{C}\hat{z}_{k/k-1}) \tag{54}$$

$\hat{\bar{P}}_{k+1/k}$ the state prediction errors covariance matrix obtained after having substituting (53) in (47) and $\tilde{\bar{\gamma}}_k^i$ the i^th^ component of $\tilde{\bar{\gamma}}_k$. We solved the unconstrained optimization problem

$$\min_{\tilde{D}_k,\Lambda_k} \hat{\bar{\Phi}}_k = tr(\hat{\bar{P}}_{k+1/k}) \tag{55}$$

the set $S = \left\{\tilde{\gamma}_k^1,..,\tilde{\gamma}_k^i,..,\tilde{\gamma}_k^q\right\}$ of scalar detection signals will be produced from the reference signal $\bar{\gamma}_k$ and the primary signal $\tilde{\bar{\gamma}}_k$ produced by (49) as presented in (56)

$$\tilde{\gamma}_k^i = \tilde{\bar{\gamma}}_k^i - \hat{J}_k^i\bar{\gamma}_k \tag{56}$$

With $\bar{\gamma}_k$ offers information about the noise interference acting on $\tilde{\bar{\gamma}}_k$ in order to reduce its effect via the noise canceller gain $\hat{J}_k^i \in \Re^{1,m-q}$ (see (Goodwin and Sin 1984) for a description of the adaptive noise cancelling problem in the area of signal processing).

$\hat{\bar{A}} = \hat{A} - \hat{A}\hat{F}\hat{\tilde{C}}$, $\hat{\tilde{C}} = H\hat{C}$ and $\hat{\bar{C}} = \Sigma\hat{C}$ with

($E \in \Re^{n-q,n}$ satisfies $EF = 0$) is so that $\hat{F}^+ = \hat{\tilde{C}}$ with

$T = \begin{bmatrix} (CF)^{+}C \\ E \end{bmatrix}$ ($E \in \Re^{n-q,n}$ satisfies $EF = 0$) is so that $\hat{F}^{+} = \hat{\tilde{C}}$ with $\hat{F} = \begin{bmatrix} I \\ 0 \end{bmatrix}$. $\tilde{\gamma}_k^i$ given by (56) with The set $S = \left\{\tilde{\gamma}_k^1,..,\tilde{\gamma}_k^i,..,\tilde{\gamma}_k^q\right\}$ of scalar detection signal is generated by the next diagonal detection filter.

$$\hat{z}_{k+1/k} = \hat{A}\hat{z}_{k/k-1} + \hat{B}u_k + (\hat{A}\hat{F} - \hat{F}\hat{\Lambda}_k)\tilde{\bar{\gamma}}_k + \hat{\tilde{D}}_k\bar{\gamma}_k \tag{57}$$

$$\tilde{\bar{\gamma}}_k = H(y_k - \hat{C}\hat{z}_{k/k-1}) \;\; \bar{\gamma}_k = \Sigma(y_k - \hat{C}\hat{z}_{k/k-1}) \tag{58}$$

$$\begin{aligned}\hat{\bar{P}}_{k+1/k} &= (\hat{\bar{A}} + \hat{F}\hat{\Lambda}_k\hat{\tilde{C}} - \hat{\tilde{D}}_k\hat{\bar{C}})\hat{\bar{P}}_{k/k-1}(\hat{\bar{A}} + \hat{F}\hat{\Lambda}_k\hat{\tilde{C}} - \hat{\tilde{D}}_k\hat{\bar{C}})^T + \\ &\hat{\tilde{D}}_k\Sigma\Sigma^T\hat{\tilde{D}}_k^T + (\hat{A}\hat{F} - \hat{F}\hat{\Lambda}_k)HH^T(\hat{A}\hat{F} - \hat{F}\hat{\Lambda}_k)^T + \hat{W}\end{aligned} \tag{59}$$

$$\hat{\bar{D}}_k = \hat{\bar{A}}\hat{\bar{P}}_{k/k-1}\hat{\bar{C}}^T\hat{\bar{H}}_k^{-1} ; \hat{\bar{H}}_k = \hat{\bar{C}}\hat{\bar{P}}_{k/k-1}\hat{\bar{C}}^T + \Sigma\Sigma^T \tag{60}$$

$$\hat{\tilde{D}}_k = \hat{\bar{D}}_k + \hat{F}\hat{\Lambda}_k\hat{J}_k \;\; \hat{J}_k = \hat{\tilde{C}}\hat{\bar{P}}_{k/k-1}\hat{\bar{C}}^T\hat{\bar{H}}_k^{-1} \tag{61}$$

$$\hat{\Lambda}_k = diag(\begin{bmatrix} \hat{\lambda}_k^1 & .. & \hat{\lambda}_k^i & .. & \hat{\lambda}_k^q \end{bmatrix}) \tag{62}$$

$$\hat{\lambda}_k^i = -\hat{\tilde{C}}_i(\hat{\bar{A}}S_{k/k}\hat{\tilde{C}}_i^T - \hat{A}\hat{F}HH_i^T)(\hat{V}_k^i)^{-1} \tag{63}$$

$$\hat{V}_k^i = \hat{\tilde{C}}_i S_{k/k}\hat{\tilde{C}}_i^T + H_i H_i^T \tag{64}$$

$$S_{k/k} = \hat{\bar{P}}_{k/k-1} - \hat{\bar{P}}_{k/k-1}\hat{\bar{C}}^T\hat{\bar{H}}_k^{-1}\hat{\bar{C}}\hat{\bar{P}}_{k/k-1} \tag{65}$$

with $\hat{\tilde{C}}_i$, H_i and $\hat{J}_k^i$ are the ith rows of $\hat{\tilde{C}}$, H and $\hat{J}_k$.

$\hat{\rho}_{k,r_i}^i = \hat{\lambda}_{k-1}^i\hat{\lambda}_{k-2}^i..\hat{\lambda}_{r_i}^i$ the signature of the hypothesized fault $d_k^i = \delta_{k,r_i}\nu^i$ on the detection signal $\tilde{\gamma}_k^i$. On the series of detection signals $S = \left\{\tilde{\gamma}_k^1,..,\tilde{\gamma}_k^i,..,\tilde{\gamma}_k^q\right\}$, the hypotheses analysis problem (40) can be rewritten

$$H_0^i : E\left\{\tilde{\gamma}_k^i\right\} = 0 \;\; H_1^i : E\left\{\tilde{\gamma}_k^i\right\} = \hat{\rho}_{k,r_i}^i\nu^i \;\; i = 1,..,q \tag{66}$$

Between a large spectrum of current solution for detecting changes in white scalar signals, one possible solution comprises of designing a bank of q GLR tests running in parallel on $\left\{\tilde{\gamma}_k^1,..,\tilde{\gamma}_k^i,..,\tilde{\gamma}_k^q\right\}$ where the ith GLR test of the bank is calculated on $\tilde{\gamma}_k^i$ as shown in (67)

$$T_k^i = \max_{r_i \in [k-(N_i-1) \quad k]} \left\{ \frac{(\nu_{k,r_i}^i)^2}{P_{k,r_i}^i} \right\} \overset{H_1^i}{\underset{H_0^i}{\gtrless}} \mu_i \tag{67}$$

where

$$\nu_{k,r_i}^i = [\sum_{j=r_i}^{k} \hat{\rho}_{j,r_i}^i (\hat{V}_j^i)^{-1} \hat{\rho}_{j,r_i}^i]^{-1} [\sum_{j=r_i}^{k} \hat{\rho}_{j,r_i}^i (\hat{V}_j^i)^{-1} \tilde{\gamma}_j^i] \tag{68}$$

And $$P_{k,r_i}^i = [\sum_{j=r_i}^{k} \hat{\rho}_{j,r_i}^i (\hat{V}_j^i)^{-1} \hat{\rho}_{j,r_i}^i]^{-1} \tag{69}$$

With r_i is supposed inside a sliding window of size N_i for a real-time application. If $T_k^i \geq \mu_i$, then

$$\hat{r}_i = \arg(\max_{r_i \in [k-(N_i-1) \quad k]} \left\{ \frac{(\nu_{k,r_i}^i)^2}{P_{k,r_i}^i} \right\}) \tag{70}$$

is the fault appearance time estimate and $\nu_{k,\hat{r}_i}^i$ the pulse magnitude estimate of covariance $P_{k,\hat{r}_i}^i$. When $T_k^i \geq \mu_i$, the detection filter's pursuit ability should be improved by generating the Willsky and Jones's updating strategy

$$\hat{z}_{k+1/k}^+ = \hat{z}_{k+1/k}^- + \hat{\eta}_{k+1,\hat{r}_i}^i \nu_{k,\hat{r}_i}^i \tag{71}$$

$$\hat{\bar{P}}_{k+1/k}^+ = \hat{\bar{P}}_{k+1/k}^- + \hat{\eta}_{k+1,\hat{r}_i}^i \hat{P}_{k,\hat{r}_i}^i \hat{\eta}_{k+1,\hat{r}_i}^{iT} \tag{72}$$

With $\hat{\eta}_{k+1,\hat{r}_i}^i = \hat{f}_i \hat{\lambda}_k^i \hat{\lambda}_{k-1}^i .. \hat{\lambda}_{\hat{r}_i}^i$ is the pulse signature on the state prediction error. $(\hat{z}_{k+1/k}^-, \hat{\bar{P}}_{k+1/k}^-)$ and $(\hat{z}_{k+1/k}^+, \hat{\bar{P}}_{k+1/k}^+)$ denote $(\hat{z}_{k+1/k}, \hat{\bar{P}}_{k+1/k})$ just before and just after the detection time k.

But, when $d_k^i = \delta_{k,r_i} v^i$ is rapidly detected, $v_{k,\hat{r}_i}^i$ do not give an accurate pulse magnitude estimate and functional several times for the similar pulse can lead to the detection filter's instability. To resolve this problem, the FDI scheme is proposed as shown in Figure 12:

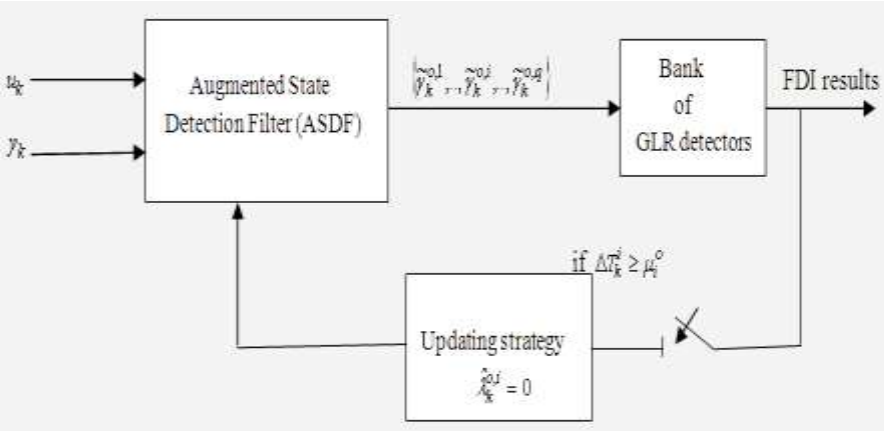

u_k
y_k
Augmented State Detection Filter (ASDF)
$\left\{\tilde{\gamma}_k^{o,1},..,\tilde{\gamma}_k^{o,i},..,\tilde{\gamma}_k^{o,q}\right\}$
Bank of GLR detectors
FDI results
if $\Delta T_k^i \geq \mu_i^o$
Updating strategy
$\hat{\lambda}_k^{o,i} = 0$

Figure 12. Simultaneous pulses detection and isolation scheme.

For growing the Detection filter's tracking ability when $T_k^i \geq \mu_i$, the pulse deadbeat attribution $\hat{\lambda}_k^i = 0$ is triggered. We have $\hat{\lambda}_k^i$ decoupled from $\left\{\hat{\lambda}_k^1 \;\; .. \;\; \hat{\lambda}_k^{i-1} \hat{\lambda}_k^{i+1} \;\; .. \;\; \hat{\lambda}_k^q\right\}$ and the lost of optimally due to $\hat{\lambda}_k^i = 0$ is then limited in the detection space of d_k^i (spanned by the image space of $\hat{f}_i$). So, GLR detector of the bank can work independently from each other as explained in the following algorithm:

1. Detection, isolation and estimation of one fault on $\tilde{\gamma}_k^i$ with GLR detector,
2. Updating of the adaptive detection filter with $\hat{\lambda}_k^i = 0$ When $T_k^i \geq \mu_i$, to improve its tracking,
3. Go to step 1 for the treatment of another fault at time $k+1$.

Let us consider the following augmented state system

$$\begin{cases} X_{k+1} = A^o X_k + B^o u_k + \Phi\Gamma\Delta u_k \tau_k + F^o d_k + w_k^o \\ y_k = C^o X_k + v_k \end{cases} \tag{73}$$

With $X_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$, $A^o = \begin{bmatrix} A & F \\ 0 & I \end{bmatrix}$, $B^o = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $F^o = \begin{bmatrix} F \\ I \end{bmatrix}$ $C^o = \begin{bmatrix} C & 0 \end{bmatrix}$,

$$w_k^o = \begin{bmatrix} w_k \\ \varepsilon_k \end{bmatrix},\ E\left\{ \begin{bmatrix} w_k^o \\ \nu_k \end{bmatrix} \begin{bmatrix} w_j^o \\ \nu_j \end{bmatrix}^T \right\} = \begin{bmatrix} W^o & 0 \\ 0 & I \end{bmatrix} \delta_{k,j},\ W^o = \begin{bmatrix} W & 0 \\ 0 & \Omega \end{bmatrix} \geq 0,$$

$$\nu_k = \begin{bmatrix} \nu_k^1 & .. & \nu_k^i & .. & \nu_k^q \end{bmatrix}^T \in \Re^q,\ \Omega = diag(\begin{bmatrix} \sigma_1 & .. & \sigma_i & .. & \sigma_q \end{bmatrix}) > 0,$$

$$d_k = \begin{bmatrix} d_k^1 & .. & d_k^i & .. & d_k^q \end{bmatrix}^T \in \Re^q$$

and the following hypotheses

$$\Delta H_0^i : d_k^i = 0 \ \ \Delta H_1^i : d_k^i = \delta_{k,r_i} \Delta \nu^i \quad i = 1,..,q \tag{74}$$

If the arbitrary variance $\Omega = E\{\varepsilon_k \varepsilon_k^T\} = diag(\begin{bmatrix} \sigma_1 & .. & \sigma_i & .. & \sigma_q \end{bmatrix})$ of the random process $\nu_{k+1} = \nu_k + \varepsilon_k$ is chosen small, the occurrence of a fault represents, for the original system (37), an abrupt change of type step (Basseville and Benveniste 1986). Successive faults of same magnitude represent abrupt changes of type slope. Similarly, an abrupt change of type "intermittent unknown input" is represented by a sequential collection of fault with growing or decreasing magnitude. On the transformed system $\hat{A}^o = T^o A^o (T^o)^{-1}$, $\hat{B}^o = T^o B$,

$$\hat{C}^o = C^o (T^o)^{-1} \ \hat{F}^o = T^o F^o,\ \hat{W}^o = T^o W^o T^{oT}$$

$$\text{with } T^o = \begin{bmatrix} (C^o F^o)^+ C^o \\ E^o \end{bmatrix}$$

($E^o \in \Re^{n,n+q}$ so that $E^o F^o = 0$), the Augmented State Detection Filter (ASDF) of state $Z_k = T^o X_k$ is presented by the equations (75)-(83).

$$\hat{Z}_{k+1/k} = \hat{A}^o \hat{Z}_{k/k-1} + \hat{B}^o u_k + (\hat{A}^o \hat{F}^o - \hat{F}^o \hat{\Lambda}_k^o) \tilde{\bar{\gamma}}_k^o + \hat{\tilde{D}}_k^o \bar{\gamma}_k^o \tag{75}$$

$$\tilde{\bar{\gamma}}_k^o = H^o (y_k - \hat{C}^o \hat{Z}_{k/k-1}) \ \ \bar{\gamma}_k^o = \Sigma^o (y_k - \hat{C}^o \hat{Z}_{k/k-1}) \tag{76}$$

$$\begin{aligned} \hat{\bar{P}}_{k+1/k} = {} & (\hat{\bar{A}}^o + \hat{F}^o \hat{\Lambda}_k^o \hat{\tilde{C}}^o - \hat{\tilde{D}}_k^o \hat{\bar{C}}^o) \hat{\bar{P}}_{k/k-1} (\hat{\bar{A}}^o + \hat{F}^o \hat{\Lambda}_k^o \hat{\tilde{C}}^o - \hat{\tilde{D}}_k^o \hat{\bar{C}}^o)^T \\ & + \hat{\tilde{D}}_k^o \Sigma^o \Sigma^{oT} \hat{\tilde{D}}_k^{oT} + (\hat{A}^o \hat{F}^o - \hat{F}^o \hat{\Lambda}_k^o) H^o H^{oT} (\hat{A}^o \hat{F}^o - \hat{F}^o \hat{\Lambda}_k^o)^T + \hat{W}^o \end{aligned} \tag{77}$$

$$\hat{\bar{D}}_k^o = \hat{\bar{A}}^o \hat{\bar{P}}_{k/k-1} \hat{\bar{C}}^{oT} (\hat{\bar{H}}_k^o)^{-1} \ \ \hat{\bar{H}}_k^o = \hat{\bar{C}}^o \hat{\bar{P}}_{k/k-1} \hat{\bar{C}}^{oT} + \Sigma^o \Sigma^{oT} \tag{78}$$

$$\hat{\tilde{D}}_k^o = \hat{\bar{D}}_k^o + \hat{F}^o \hat{\Lambda}_k^o \hat{J}_k^o \ \ \ \hat{J}_k^o = \hat{\tilde{C}}^o \hat{\bar{P}}_{k/k-1} \hat{\bar{C}}^{oT} (\hat{\bar{H}}_k^o)^{-1} \tag{79}$$

$$\hat{\Lambda}_k^o = diag(\begin{bmatrix} \hat{\lambda}_k^{o,1} & .. & \hat{\lambda}_k^{o,i} & .. & \hat{\lambda}_k^{o,q} \end{bmatrix}) \tag{80}$$

$$\hat{\lambda}_k^{o,i} = -\hat{\tilde{C}}_i^o(\hat{\bar{A}}^o S_{k/k}\hat{\tilde{C}}_i^{oT} - \hat{A}^o\hat{F}^o H^o H_i^{oT})(\hat{V}_k^{o,i})^{-1} \tag{81}$$

$$\hat{V}_k^{o,i} = \hat{\tilde{C}}_i^o S_{k/k}\hat{\tilde{C}}_i^{oT} + H_i^o H_i^{oT} \tag{82}$$

$$S_{k/k} = \hat{\bar{P}}_{k/k-1} - \hat{\bar{P}}_{k/k-1}\hat{\bar{C}}^{oT}(\hat{\bar{H}}_k^o)^{-1}\hat{\bar{C}}^o\hat{\bar{P}}_{k/k-1} \tag{83}$$

with $H^o = (\hat{C}^o\hat{F}^o)^+$, $\Sigma^o = \beta^o(I - \hat{C}^o\hat{F}^o H^o)$ ($\beta^o \in \Re^{m-q,m}$ so that $rank(\Sigma^o) = m - q$), $\hat{\bar{A}}^o = \hat{A}^o - \hat{A}^o\hat{F}^o\hat{\tilde{C}}^o$, $\hat{\tilde{C}}^o = H^o\hat{C}^o$ and $\hat{\bar{C}}^o = \Sigma^o\hat{C}^o$ where $\hat{\tilde{C}}_i^o$, H_i^o, $\hat{J}_k^{o,i}$ are the ith rows of $\hat{\tilde{C}}^o$, H^o, $\hat{J}_k^o$. The ith detection signal $\tilde{\gamma}_k^{o,i} = \tilde{\bar{\gamma}}_k^{o,i} - \hat{J}_k^{o,i}\bar{\gamma}_k^o$ is sensitive to d_k^i and perfectly decoupled from $\left\{d_k^1,..,d_k^{i-1},d_k^{i-1},..,d_k^q\right\}$ under $E\left\{\tilde{\gamma}_j^i\tilde{\gamma}_k^i\right\} = \hat{V}_k^{o,i}\delta_{k,j}$ and the hypotheses testing problem (74) is rewritten

$$\Delta H_0^i : E\left\{\tilde{\gamma}_k^{o,i}\right\} = 0 \quad \Delta H_1^i : E\left\{\tilde{\gamma}_k^{o,i}\right\} = \hat{\rho}_{k,r_i}^{o,i}\Delta\nu^i \quad i = 1,..,q \tag{84}$$

from the pulse signature $\hat{\rho}_{k,r_i}^{o,i} = \hat{\lambda}_{k-1}^{o,i}\hat{\lambda}_{k-2}^{o,i}..\hat{\lambda}_{r_i}^{o,i}$.

The bank of GLR detectors on $S = \left\{\tilde{\gamma}_k^{o,1},..,\tilde{\gamma}_k^{o,i},..,\tilde{\gamma}_k^{o,q}\right\}$ is described as follows

$$\Delta T_k^i = \max_{r_i \in \left[k-(N_i^o-1) \quad k\right]} \left\{\frac{(\Delta\nu_{k,r_i}^i)^2}{\Delta P_{k,r_i}^i}\right\} \begin{matrix} \Delta H_1^i \\ \geq \\ < \\ \Delta H_0^i \end{matrix} \mu_i^o \tag{85}$$

With

$$\Delta\nu_{k,r_i}^i = [\sum_{j=r_i}^{k}\hat{\rho}_{j,r_i}^{o,i}(\hat{V}_j^{o,i})^{-1}\hat{\rho}_{j,r_i}^{o,i}]^{-1}[\sum_{j=r_i}^{k}\hat{\rho}_{j,r_i}^{o,i}(\hat{V}_j^{o,i})^{-1}\tilde{\gamma}_j^{o,i}] \text{and} \, \Delta P_{k,r_i}^i = [\sum_{j=r_i}^{k}\hat{\rho}_{j,r_i}^{o,i}(\hat{V}_j^{o,i})^{-1}\hat{\rho}_{j,r_i}^{o,i}]^{-1} \tag{86}$$

The adaptive state prediction is $\hat{X}_{k+1/k} = \begin{bmatrix} \hat{x}_{k+1/k} \\ \hat{\nu}_{k+1/k} \end{bmatrix} = (T^o)^{-1}\hat{Z}_{k+1/k}$ of the augmented state $X_{k+1} = \begin{bmatrix} x_{k+1} \\ \nu_{k+1} \end{bmatrix}$.

At the detection time of $d_k^i = \delta_{k,r_i} \Delta v^i$, the pulse deadbeat assignment $\hat{\lambda}_k^{o,i} = 0$ increases the tracking ability of $\hat{v}_{k+1/k}^i$ without any interference on $\left\{\hat{v}_{k+1/k}^1, .., \hat{v}_{k+1/k}^{i-1}, , \hat{v}_{k+1/k}^{i+1}, .., \hat{v}_{k+1/k}^j\right\}$ during the tracking. The tuning parameters $\left\{\mu_1^o, .., \mu_i^o, .., \mu_q^o\right\}$ and $\left\{N_1^o, .., N_i^o, .., N_q^o\right\}$ must solve the traditional tradeoffs: missed detections rate P_D^i vs. false alarms rate P_F^i, minimum pulse magnitude $\Delta v_{\min}^i$ that we wish to detect vs. the mean detection delay. The choice of μ_i is highly dependent to N_i^o but the determination of (μ_i^o, N_i^o) is decoupled to the choice of (μ_j^o, N_j^o), $\forall j \neq i$.

7. Simulation Results

In this section, we present a numerical example that illustrates the results of the chapter.

We employ the previous approach on the Networked Control System described by Figure 11 with considering the form of the model of IM that is described in section 4.

The System is Affected by a variable faults occurring at time 4000 and disappearing at time 4300. The variance $\Omega = E\left\{\varepsilon_k \varepsilon_k^T\right\} = diag(\left[\sigma_1 \quad \sigma_2\right])$ is chosen small to ensure a small tracking ability on $\hat{v}_{k+1/k}$. The threshold levels $\mu_1^o = 2$ have been fixed in accordance with the sizes $N_1^o = N_2^o = 5$ |($N^o = 5$) of the sliding windows.

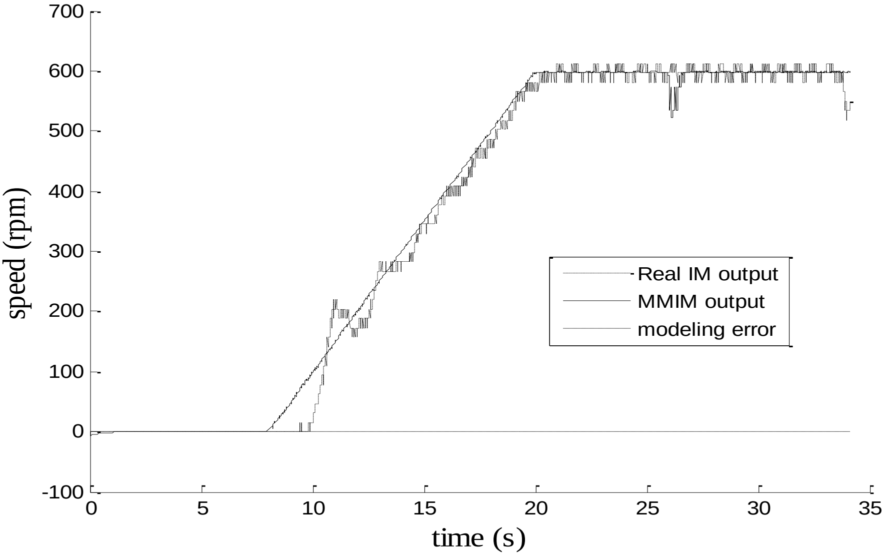


Figure 13. The real speed output of multi-model induction motor, real speed output and the error.

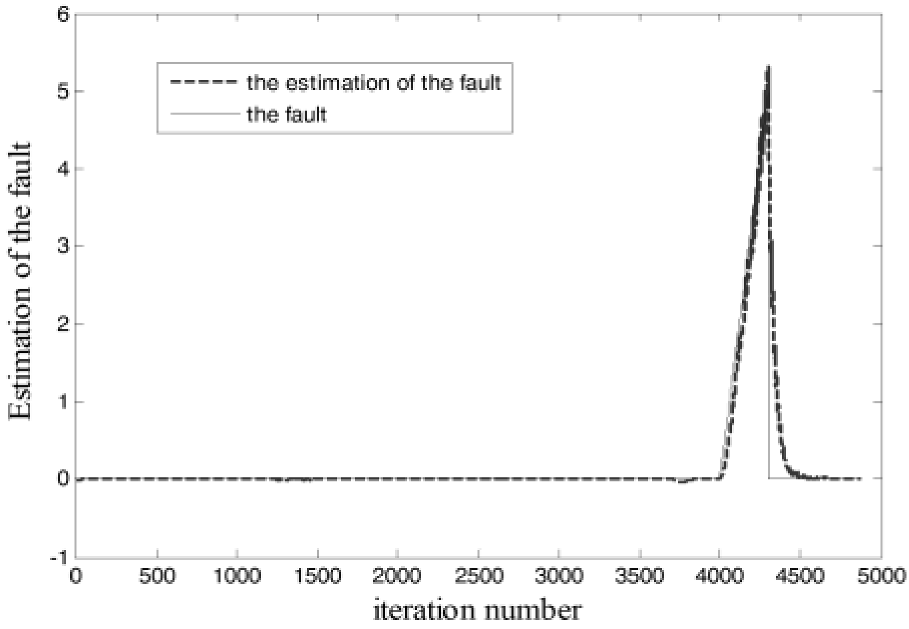


Figure 14. Estimation of the fault.

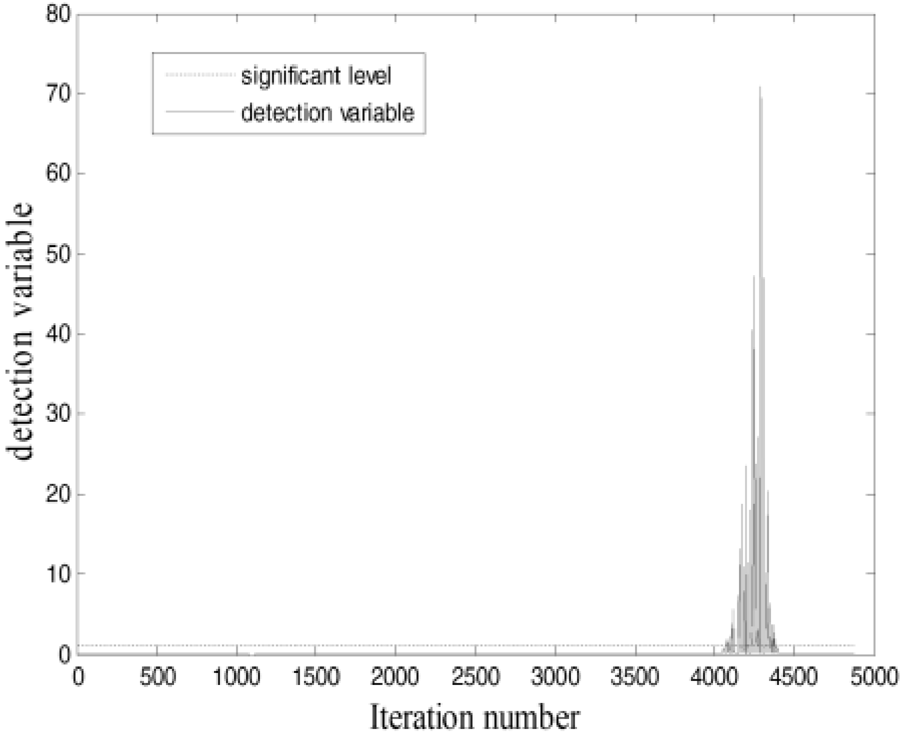


Figure 15. Detection variable ΔT_k^1 and its significant level μ_1^o .

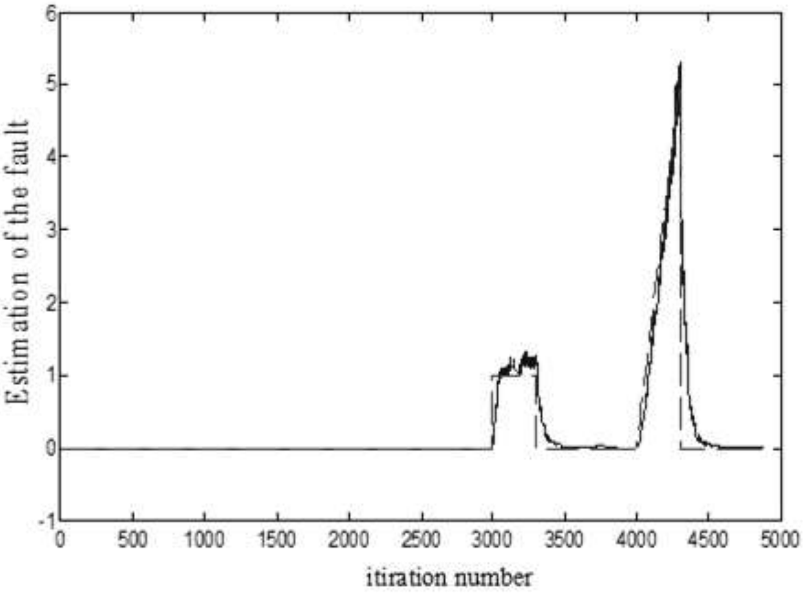


Figure 16. Estimation of the fault.

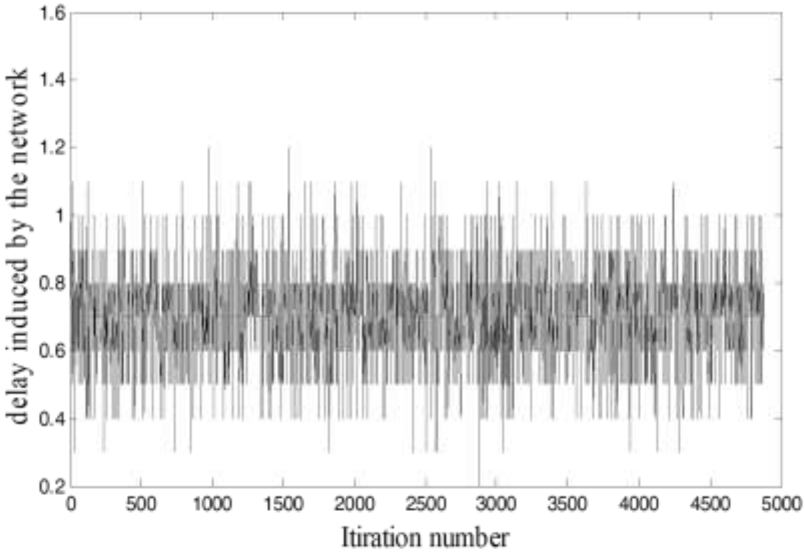


Figure 17. Bounded iduced Delay by the network.

Figures 14-16 show the simulation results acquired by the ASDF presented in Figure 12. The bonded induced delay by the network is given in Figure 17. The real and the estimated faults obtained by the Adaptive State Filter are shown in Figures 14 and 16. The detection variable and the threshold in the fault case is ullistrated in Figure 15. Figures 14 and 16 clearly show the efficiency of the ASDF to estimate the faults. The speed output of multi-model induction motor, the reel speed output of IM and the error of modeling are illustrated in Figure 13.

Conclusion

In this chapter, the problem for estimating sequential faults in Networked Control System is addressed using an adaptive state filter. In this context, an adaptive augmented state filtering is implemented. We estimate the states and faults of the IM. The IM is a nonlinear system modeled by the multi-model approach. We used in, this chapter, the adaptive detection filter (Keller 1999) and the extended of Generalised Likelihood Ratio (GLR) test of (Willsky and Jones 1976) to estimate the faults in Networked Control Systems subject to the induced delay. The detection filter has been optimized with generating a set of white detection signals, so that each GLR detector works autonomously from each other and so that its tracking ability be increased at the detection time of one fault without any consequence on the detection of the others faults.

A multi-model approach has been performed for the modeling of the IM. The developed approach is based on a decoupled multimodel. To do this four steps have been developed. Firstly a rich database has been collected from the inputs /outputs measurements. Then, the obtained data base is clustered into N clusters which are identified structurally and parametrically to obtain the local models that are combined through validities functions to generate the multimodel.

The different obtained experimental results have affirmed that the obtained multimodel outputs follow not only the healthy real system outputs.

The obtained simulation results show the effectiveness of the proposed method for NCS.

References

Abid, A., A. Adouni, M. Ben Hamed and L. Sbita, “A New Pv Cell Model Based On Multi Model Approach,” *The Fourth International Renewable Energy Congress*, Sousse, Tunisia, pp, 20-22, December 2012.

Abid, A., Ben Hamed M., Sbita L., (2011), Induction Motor Real Time Application of Multimodel Modeling Approach, *International Review of Electrical Engineering* (IREE), 6: 655-660.

Abid, A., Ben Mabrouk Z., Ben Hamed M., Sbita L., (2013). Multiple Lunberger Observer for an Induction Motor represented by decoupled multiple model, *2013*

10th International Multi-Conference on Systems, Signals & Devices (SSD) Hammamet, Tunisia, March 18-21.

Angelov P. P. and Filev, D. P., (2004). An Approach to Online Identification of Takagi-Sugeno Fuzzy Models. *IEEE transactions on systems, Man, and cybernetics-part B: Cybernetics*, 34: 484-489.

Bello, O., Hamam, Y., Djouani K., (2014). Fuzzy dynamic modelling and predictive control of a coagulation chemical dosing unit for water treatment plants. *Journal of Electrical Systems and Information Technology*, 1: 129–143.

Ben Mabrouk, Z., A. Abid, M. Ben Hamed and S. Lassaad 2013. "Diagonal detection filter for simulation faults in Networkes Control Systems." *International Review on modeling and Simulation* (IREMOS) 6:1355-1362.

Besancon, G., D. Georges and Z. Benayache 2007. "Asymptotic state prediction for continuoustime systems with delayed input and application to control." *Proceedings of the European Control Conference*, Kos, Greece, July.

Casalino, G., Del Buono, N., Mencar, C., (2014), Subtractive clustering for seeding non-negative matrixfactorizations, *Information Sciences*, 257: 369–387.

Chabir, K., M. Koni Ben Gayed, M. Naceur Abdelkrim and D. Sauter 2007. "Diagnostic des Systèmes Contrôlés par un Réseau." *Sciences et Techniques de l'Automatique conference* STA'2007-SSI-275, Tunisie.

Chadli, M., Abdo, A., X Ding, S., (2013), H−/H∞ fault detection filter design for discrete-time Takagi–Sugeno fuzzy system, *Automatica*, 49: 1996–2005.

Chiu, S. L. (1994), Fuzzy Model Identification Based On Cluster Estimation, *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 267 – 278.

Didier, B., Capron, O., Odloak, D., (2013). LMI-Based Multi-model Predictive Control of an Industrial C3/C4 Splitter, *Journal of Control*, *Automation and Electrical Systems*, 24: 420-429.

Domlan, E., Huang, B., Xu, F., Espejo, A., (2011), A decoupled multiple model approach for soft sensors design, *Control Engineering Practice* 19: 126–134.

Elfelly, N., Iieulot, J. Y., Benrejeb, M., Borne, P., (2012), A Multimodel Approach for Complex Systems Modeling based on Classification Algorithms, *Int J Comput Commun*, 7: 644-659.

Elfelly, N., Jean-Yves Dieulot and P. Borne 2008. "A Neural Approach of Multimodel Representation of Complex Processes." International Journal of Computers, *Communications & Control Computers* 3:149–160.

Gao, H. J., T. Chen, and T. Chai 2007. "Passivity and passification for networked control systems." *SIAM J. Contr. Optimiz* 46:1299–1322.

Gao, H. J., T. W. Chen and J. Lam 2008. "A new delay system approach to network-based control." *Automatica* 44:39−52.

Garcia-Nieto, S., Martinez, M., Blasco, X., and Sanchis, J., (2008), Nonlinear predictive control based on local model networks for air management in diesel engines, *Control Engineering Practice*, 16(12): 1399–1413.

Goodwin, G. C., H. Haimovich, E. Quevedo, and S. Welsh 2004. “A moving horizon approach to networked control system design.” *IEEE Trans. Autom. Contr*ol 49:1427–1445. Accessed February 28, 2010. doi:10.1086/599247.

Hamdi, H., Rodrigues, M., Mechmeche, C., Theilliol, D. and BenHadj Braiek, N., (2012). Fault detection and isolation in linear parameter-varying descriptor systems via proportional integral observer. *International journal of adaptive control and signal processing*, 26: 224-240.

Hou Hai-liang, Yang Tong-guang, “A Novel Method of Induction Motor Stator Fault Diagnosis,” Electrical Machines and Systems, 2005. ICEMS 2005. *Proceedings of the Eighth International*, 3: 2254–2258.

Ichalal, D., Marx, B., Maquin, D., Ragot, J., (2014), Fault detection, isolation and estimation for Takagi-Sugeno nonlinear systems, *Journal of the Franklin Institute*, 351: 3651-3676.

Jari M. Boling, Dale E. Seborg, Joao P. Hespanha (2007), Multi-model adaptive control of a simulated pH neutralization process, *Control Engineering Practice*, 15: 663–672.

Jin, Z. P., V. Gupta and R. M. Murray 2006. “State estimation over packet dropping networks using multiple description coding.” *Automatica* 42: 1441−1452.

Keller J. Y. 1999. “Fault isolation filter for linear systems.” *Automatica* 35:1701-1706.

Keller J. Y. 2007. “A fault detection filter including an adaptive noise cancellation strategy.” *European Journal of Control* 6:627-638.

Khalid Dahi, Soumia Elhani,Said Cuedira and Nabil Ngote (2014), Fault Diagnosis in Induction Motor Using Motor’s Residual Stator Current Signature Anaysis,” *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, pp. 631-643,.

Latrach, C., Kchaou, M., Rabhi, A., and El Hajjaji, A. (2015). Decentralized Networked Control System Design Using Takagi-Sugeno (TS) Fuzzy Approach. *International Journal of Automation and Computing*, 12:125-133.

Mao Z. and B. Jiang 2007. “Fault identification and fault-tolerant control for a class of networked control systems.” *Int. J. Innovat. Comput., Inf.* Contr. 3:1121–1130.

Mao, Ze-Hui and Jiang Bin 2007. “Fault Estimation and Accommodation for Networked Control Systems with Transfer Delay.” *Acta Automatica Sinica* 33, no. 7:738-743.

Mazinan, A. H., Sadati, N., (2011), An intelligent multiple models based predictive control scheme with its application to industrial tubular heat exchanger system, *Applied Intelligence*, 34: 127-140.

Men, Z., Yee E., Lien F. S., Ji, H., Liu, Y. (2014). Bootstrapped Multi-Model Neural-Network Super-Ensembles for Wind Speed and Power Forecasting. *Energy and Power Engineering*, 6, 340-348.

Orjuela, R., Marx, B., Ragot, J., and D. Maquin,(2010), Fault diagnosis for nonlinear systems represented by heterogeneous multiple models, 2010 Conference on Control and Fault Tolerant Systems, Nice, France, 600–605.

Orjuela, R., Marx, B., Ragot, J., and Maquin, D., (2008), "State estimation for nonlinear systems using a decoupled multiple mode." International *Journal of Modelling Identification and Control*, (1): 59–67.

Orjuela, R., Marx, B., Ragot, J., and Maquin, D., (2013), Nonlinear system identification using heterogeneous multiple models, *Int. J. Appl. Math. Comput. Sci*., 23: 103–115.

Piguet, Y., U. Holmberg and R. Longchamp1999. "A minmax approach for multi-objective controller design multiple models." *IEEE International Journal of control* 72:716 – 726.

Rodrigues, M., Theilliol D., and S., Dominique, (2005). Design of an Active Fault Tolerant Control for Nonlinear Systems described by a Multi-Model Representation, Proceedings of the 13th Mediterranean Conference on Control and Automation Limassol, Cyprus, June 27-29, 2005, 1579-1584.

Samko Golabi, Ebrahim Babaei, Mohammad Bagher Bannae Sharifian, Zahed Golabi, "Application of Speed, Rotor Flux, Electromagnetic, Load Torque Observers and Diagnostic System in a Vector-Controlled High-Power Traction Motor Drive," *Arabian Journal for Science and Engineering*, vol. 39, no. 4, pp. 2979–2996, November 2013.

Shankar, H., P. L. N. Raju, K. Ram Mohan Rao (2012). Multi Model Criteria for the Estimation of Road Traffic Con-gestion from Traffic Flow Information Based on Fuzzy Logic. *Journal of Transportation Technologies*, 2: 50-62.

Shu-Ping, H., (2014), Fault Estimation for T-S Fuzzy Markovian Jumping Systems based on the Adaptive Observer, *International Journal of Control, Automation, and Systems*, 12: 977-985.

Venkat, A. N., Vijaysai P., and Gudi, R. D, (2003), Identification of complex nonlinear processes based on fuzzy decomposition of the steady state space. *Journal of Process Control*, 13(6): 473–488.

Wang, Z., F. Yang, D. Ho, and X. Liu 2006. "Robust H∞7 filtering for stochastic time-delay systems with missing measurements." *IEEE Trans. Signal Process* 54:2579–2587.

Willsky A. S. and H. L. Jones 1976. "A generalized likelihood ratio approach to detection and estimation of jumps in linear systems." *IEEE Transactions on Automatic Control* 21:108-112.

Wu, J., T. W. Chen 2007. "Design of networked control systems with packet dropouts." *IEEE Transactions on Automatic Control* 52:1314−1319.

Xiong J. and J. Lam 2006. "Stabilization of linear systems over networks with bounded packet loss". *Automatica* 43:80–87.

Yang, H., Jiang B, Cocquempot V, “Fault tolerant control design for hybrid system,*”* *Lecture notes in control and information sciences,* Springer, Berlin, 2010.

Yong-Qiang, W., YE Hao1 DING Steven, WANG Gui-Zeng 2009. “Fault Detection of Networked Control Systems Subject to Access Constraints and Random Packet Dropout.” *Acta Automatica Sinica* 35:1230-1234.

Yu, C. H., and Choi, J. W.,(2014), Interacting Multiple Model Filter-based Distributed Target Tracking Algorithm in Underwater Wireless Sensor Networks, *International Journal of Control, Automation, and Systems,* 12: 618-627.

Yue D. and Q. L. Han 2006. “Network-based robust H∞ filtering for uncertain linear systems.” *IEEE Trans. Signal Process*. 54:4293–4301.

Zhang, H. S., L. H. Xie and G. R. Duan 2007. “H∞ control of discrete-time systems with multiple input delays.” *IEEE Transactions on Automatic Control* 52:271−283.

Zhu, X., C. Hua and S. Wang 2008. “State feedback controller design of networked control systems with time delay in the plant.” *Int. J. Innovat. Comput., Inf. Contr*. 4:283–290.

In: A Closer Look at Fault-Tolerant Control
Editor: Jeremy M. Hutton

ISBN: 978-1-53617-528-8
© 2020 Nova Science Publishers, Inc.

Chapter 8

DIAGNOSIS OF SENSORES FAILURE IN INDUCTION MOTOR

***Ben Mabrouk Zaineb*[*], *Abid Aicha*,**
Ben Hamed Mouna and Lassad Sbita
National Engineering School of Gabes, Gabes, Tunisia

ABSTRACT

The chapter deals with a diagnosis of an induction motor followed by sensor failure modes. First, construction of induction motor has been presented. Then a review of induction motor failures has been discussed. The third part studies the problem of diagnosis strategy for an induction motor sensor faults. This strategy is based on unknown input proportional integral (PI) multiobserver. The need of a sensorless drive requires soft sensors such as estimators or observers. The convergence of the estimation error is guaranteed by using the Lyapunov's based theory. The proposed diagnosis approach is experimentally validated on a 1 kW Induction motor. Obtained simulation results confirm that the adaptive PI multi-observer consent to accomplish the detection, isolation and fault identification tasks with high dynamic performances.

Keywords: induction motor, defaults, failures, diagnosis, multimodal approach, adaptive PI multi-observer

[*] Corresponding Author's Email: benmabroukza@gmail.com.

INTRODUCTION

As the workhorse of industry, Induction motors occupy currently an important place in industrials drive especially based on variable speed. Thus, a growing interest is given to the implementation of a new and effective diagnosis process to ensure the safety of these engines (Khalid, et al., (2014); Samko, et al., (2013); Yang, et al., (2010): Hou et al., (2005)). So, diagnosis of induction motor has been a lot research in the literature and different induction motor failure modes are studied. In this context, we are interested in sensor faults of induction motor. Two types of sensor faults are considered, they affect respectively the speed **Ω** and the stator current. In this chapter, the detection and isolation of the induction motor's faults is designed based on the synthesized PI multi-observer.

New procedures and monitoring algorithms for the Faults Detection, Isolation and Identification (FDI) are developed. Among, the most commonly used FDI techniques are those based on the generation of residues from a healthy model. These procedures include a step of failure symptoms generation or residues generation. The production of these residues requires the use of information from a suitable model to compare them with those provided by the measuring instruments. Direct extension of the control methods and estimation developed in the context of linear models to the case of nonlinear models is delicate. Thus, a modeling approach as an alternative that overcomes these difficulties is the multi-model approach that many recent researches have a great interest in it (Didier et al., (2013); Men et al., (2014); Shankar et al., (2012)).

The power of such an approach is that it would allow the exploitation of simple algorithms applied to linear models to complex real systems. The multi-model approach eases the complexity of nonlinear system by representing it via combination of a set of local linear models. Each local model contributes to the total representation through weighting functions.

In literature, there exist various multi-model categories. Particularly, the coupled state multi-model called Takagi-Sugeno models or decoupled state multi-model. Concerning the first structure, local models share the same state vector whereas the second structure is a decoupled local models representation with different state vectors. The Takagi-Sugeno multi-model structure is the most prevalent in the analysis and the synthesis of the multimodels (Latrach, et al., (2015); Bello, et al., (2014)). Whereas, On the other hand, the works which concern the decoupled multimodel are less frequent. However, it is advisable to underline that works on the identification (Abid, et al., (2012); Venkat, et al., (2003); Domla, et al., (2011)) and the control (Garcia-Nieto, et al., (2008)) of nonlinear systems have already taken advantage of this structure and confirmed its efficiency. As for the state estimation, an approach to make the synthesis of a PI multiobserver, by using a decoupled multimodel was proposed in (Orjuela et al., (2008); Orjuela et al., (2010)).

In this chapter, three parts are studied. The second part presents an overview of failure modes of induction motor. The third concern the design of the multiobserver and the final part concern the diagnosis of IM faults.

1. Failure Modes of Induction Motor

An induction motor is an AC electric motor. This machine comprises a magnetic circuit interlinking two electric circuits which are placed on the two main parts of the machine: the stationary part called the stator and the rotating part called the rotor. Figure 1 shows the construction of the induction Motor.

For this induction machine is referred as an electromechanical energy conversion device which converts electrical energy into mechanical energy. The electric current in the rotor needed to produce torque is obtained by electromagnetic induction from the magnetic field of the stator winding.

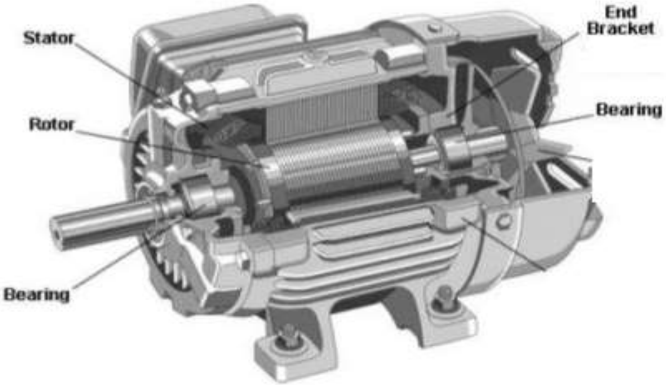


Figure 1. Construction of the Induction Motor.

Therefore, an induction motor can be made without electrical connections to the rotor. An induction motor’s rotor can be either wound type or Three-phase squirrel-cage induction motors. An induction motor is widely used as industrial drives because they are rugged, reliable and economical. The stator of an induction motor consists of poles carrying supply current to induce a magnetic field that penetrates the rotor. To optimize the distribution of the magnetic field, windings are distributed in slots around the stator. The magnetic field having the same number of north and south poles. Induction motors are most commonly run on three-phase power.

Generally, both the stator and rotor consist of two circuits: an electric circuit to carry a current and normally made of insulated copper or insulated aluminum and a magnetic circuit, to carry the magnetic flux made of laminated magnetic material normally steel.

Stator: The stator is the outer stationary part of the motor. It consists of the outer cylindrical frame, the magnetic path, and a set of insulated electrical windings.

Rotor: It is the rotating part of the motor. It is placed inside the stator bore and rotates coaxially with the stator. Like the stator, rotor is also made of a set of slotted thin sheets, called laminations, of electromagnetic substance (special core steel) pressed together in the form of a cylinder. Slots consist of the electrical circuit and the cylindrical electromagnetic substance acts as magnetic path. Rotor winding of an induction motor may be of two types: squirrel-cage type and wound type. Depending on the rotor winding induction motors are classified into two groups: squirrel-cage type induction motor and wound-rotor type induction motor.

The Induction Motors Failure Modes

The induction motors failure modes are in large sense. Different types of faults may occur of different kinds in rotor and in stator. It can be listed as follows:

Electrical-related faults: Faults under this classification are unbalance supply voltage or current, single phasing, under or over voltage of current, reverse phase sequence, earth fault, overload, inter-turn short-circuit fault, and crawling.

Mechanical-related faults: Faults under this classification are broken rotor bar, mass unbalance, air gap eccentricity, bearing damage, rotor winding failure, and stator winding failure.

Environmental-related faults: Ambient temperature as well as external humidity will affect the performance of an induction motor. Vibrations of the machine, due to any reason, such as installation defect, foundation defect, etc., also will affect the performance.

Faults in induction motors can be also categorized as follows:

Broken bar fault: In general, this fault stated as rotor fault.

Rotor mass unbalance fault.

In an Induction Motor, multiple faults may occur simultaneously and in that case determination of the initial problem is quite difficult. The Effects of such faults in induction motor result in unbalanced stator currents and voltages, oscillations in torque, reduction in efficiency of torque, overheating, and excessive vibration. The induction motor performance may be affected by any of the faults. Figure 2 shows the classification of Induction Motor. In the next section, diagnosis of induction motor sensors faults is discussed based on the synthesized PI multi-observer.

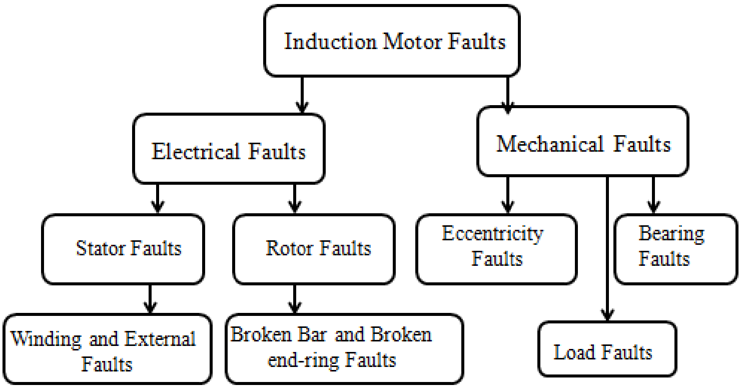


Figure 2. Classification of Induction Moto Faults.

2. Proportionnal-Integral Multiobserver Design

The decoupled multi-model structure is adapted such as to consider the unknown input vector that may be the fault or disturbance, and then exploited in developing the based observer diagnosis's strategy. The PI multi-observer is considered as a robust observer able to identify not only the state but also the faults.

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u(k) + D_i + E_i f(k) \\ y_i(k) = C_i x_i(k) + Mf(k) \\ y(k) = \sum_{i=1}^{N} V_i(k) y_i(k) \end{cases} \tag{1}$$

Ei and f(t) are respectively the influence caused by the unknown input on the state's system and the unknown input vector.

The main objective of the PI multiobserver is the simultaneously estimation of both state and unknown inputs that are considered varying feebly. Its structure is developed as (2).

$$\begin{cases} \hat{x}_i(k+1) = A_i \hat{x}_i(k) + B_i u(k) + D_i + E_i \hat{f}(k) + K_{pi}(y(k) - \hat{y}(k)) \\ \hat{f}(k+1) = \hat{f}(k) + \sum_{i=1}^{N} \hat{V}_i(k) K_I (y(k) - \hat{y}(k)) \\ \hat{y}(k) = \sum_{i=1}^{N} \hat{V}_i C_i \hat{x}_i(k) + M\hat{f}(k) \end{cases} \tag{2}$$

$\hat{x}_i$ and $\hat{y}$ are respectively the observed state vector and output vector.

In (Orjuela et al., (2010)) the validity functions Vi used to create the PI multiobserver output are the same as computed in system modeling however here the validity function are updating and recalculated.

$$\hat{V}_i = \frac{v_i \prod_{j=1,\, j\neq i}^{N} \left(1 - \mathbf{e}^{-\left(\frac{r_j}{\sigma}\right)^2}\right)}{\sum_{i=1}^{N} \left(v_i \prod_{j=1,\, j\neq i}^{N} \left(1 - \mathbf{e}^{-\left(\frac{r_j}{\sigma}\right)^2}\right)\right)} \tag{3}$$

with

$$v_i = 1 - \frac{r_i}{\sum_{i=1}^{N} r_i} \tag{4}$$

$$r_i = \left\| y - \hat{y}_i \right\| \tag{5}$$

To estimate the system state, The PI observer synthesis consist in benefit from the rebuilding output error with a proportional effect, where the integral effect is used to estimate the sensor or actuator’s faults. Thus, the main role of the observer design is to obtain the proportional and integral gain matrices KPi and KI.

The local models’ outputs yi(k), that are modeling’s artificial signals generated to represent the real system’s behavior, are not exploitable to control an observer as they are not measurable, indeed the multi-model output y(k) as it is the only one that is accessible to measurement, can be evaluated with a system’s physical amount. Thus, to deal with this problem, an augmented system is defined as

$$\begin{cases} x(k+1) = Ax(k) + Bu(K) + D + Ef(k) \\ \qquad y(k) = C(k)x(k) + Mf(k) \end{cases} \tag{6}$$

where

$$x(t) = \left[x_1^T(t) \ldots x_i^T(t) \ldots x_N^T(t)\right]^T \in R^n n = \sum_{i=1}^{N} n_i \tag{7}$$

$$\begin{cases} A = diag\{A_1, \dots, A_N\} \in R^{n\times n} \\ B = [B_1^T, \dots, B_N^T]^T \in R^{n\times m} \\ C(k) = \sum_{i=1}^{N} V_i\,(k)C_i \in R^{p\times n} \\ C_i = [0 \dots C_i \dots 0] \\ D = [D_1^T, \dots, D_N^T]^T \in R^{n\times 1} \\ E = [E_1^T, \dots, E_N^T]^T \in R^{n\times 1} \end{cases} \tag{8}$$

The multiobserver associated to this augmented multi-model has the following structure (9).

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(K) + D + Ef(k) + K_p(y(k) - \hat{y}(k)) \\ \hat{f}(k+1) = \hat{f}(k) + K_I C(x(k) - \hat{x}(k)) \\ \hat{y}(k) = C(k)\hat{x}(k) + M\hat{f}(k) \end{cases} \tag{9}$$

So considering the definition of augmented system, the augmented error ea is constructed of a state estimated error e and a fault estimated error ε.

$$e_a(k) = \begin{bmatrix} e(k) \\ \varepsilon(k) \end{bmatrix} = \begin{bmatrix} x(k) - \hat{x}(k) \\ f(k) - \hat{f}(k) \end{bmatrix} \tag{10}$$

The augmented error can be expressed as

$$e_a(k+1) = (A_a - K_a C_a(k))e_a(k), \tag{11}$$

where

$$K_a = \begin{bmatrix} K_P \\ K_I \end{bmatrix} and\ K_P = [K_{P1}^T, \dots, K_{PN}^T]^T \tag{12}$$

and

$$A_a = \begin{bmatrix} A\ E \\ 0\ I \end{bmatrix}, \bar{C}_i(t) = [C_i(t)\ M] \tag{13}$$

With regards to the fact that the unknown inputs are supposed constant or having very slow dynamics,

$$f(k+1) - f(k) \approx 0 \tag{14}$$

Then we can obtain

$$\begin{bmatrix} e(k+1) \\ \varepsilon(k+1) \end{bmatrix} = \begin{bmatrix} A - K_P C(k) \; E - K_P M \\ -K_I C(k) \; I - K_I M \end{bmatrix} \begin{bmatrix} e(k) \\ \varepsilon(k) \end{bmatrix} \tag{15}$$

In order to ensure the multiobserver stability, and to guarantee the convergence of the augmented estimation error, we propose to exploit the Lyapunov's approach.

The Lyapunov approach consist in the definition of a candidate function

$$V(e_a(k)) = e_a^T(k) P e_a(k), P > 0 \text{ P=P}^{\text{T}} \tag{16}$$

The exponential convergence is guaranteed if there exist a symmetric definite positive matrix P and a positive scalar α verifying the following condition:

$$\Delta V(e_a(k)) + 2\alpha V(e_a(k)) < 0 \tag{17}$$

where

$$\Delta V(e_a(k)) = V(e_a(k+1)) - V(e_a(k)) \tag{18}$$

Based on the Lyapunov's approach defined above, a set of conditions in forms of linear matrix inequalities (LMI), ensuring the exponential convergence of the estimation's error, are generated as approved by the theorem (1) (Orjuela et al., (2010)).

Theorem: The estimation error between the decoupled multi-model and the PI observer converges exponentially towards zero if there exist a symmetric definite positive matrix P and a matrix G verifying the LMI following:

$$\begin{bmatrix} (2\alpha - 1)P & \bar{C}_i^T G^T - \text{A}_\text{a} P \\ G\bar{C}_i - PA_a & \text{-P} \end{bmatrix} < 0, \text{ i=1,...,N} \tag{19}$$

The resolution of these inequality permits to compute the observer's gains.

α is the attenuation rate which serves to quantify the convergence speed of the estimation error. Having 0 < α < 0.5 let to obtain the KI and Kp gains as:

$$K_a = P^{-1} G \tag{20}$$

3. Induction Motor Sensor Fault Detection and Isolation

In this section, the detection and isolation of the induction motor‘s faults is designed based on the synthesized PI multi-observer.

The impact of the sensors faults on the system state is expressed with E_i, i=$\{1,\dots,N\}$ while M represents the faults impact on the IM outputs.

Two types of sensor faults are considered, they affect respectively the speed Ω and the stator current.

The IM is a multivariable system with an input u that is the IM control and two outputs $y=\left[w \ \ \mathrm{i}_s\right]$.

Therefore the PI multi-observer structure is equivalent to a bank of two multi-observers since $\hat{y}=\left[\hat{w} \ \ \hat{\mathrm{i}}_s\right]$.

The model of the induction motor (21) can be written as state form

$$\begin{cases}\hat{x}_i(k+1) = A_i\hat{x}_i(k) + B_iu(k)\\ \hat{y}(k) = \sum\limits_{i=1}^{N} V_iy_i(k)\end{cases} \tag{21}$$

where

$$\begin{cases}x_i(k) = [w_i(k) \ \ \mathrm{i}_s(\mathrm{k})]',\\ y_i(k) = C_ix_i(k)\ , \ \forall\ \mathrm{i}=1,\dots\mathrm{N}\\ \mathrm{N}=8\end{cases}$$

$$A_1 = \begin{bmatrix}0{,}89 & 0\\ 0 & 0.93\end{bmatrix},\quad A_2 = \begin{bmatrix}0{,}914 & 0\\ 0 & 0.978\end{bmatrix},$$
$$A_3 = \begin{bmatrix}0{,}889 & 0 & 0\\ 0 & 0 & 1\\ 0 & 0.155 & 0.037\end{bmatrix}, A_4 = \begin{bmatrix}0 & 1 & 0\\ 0 & 0.705 & 1\\ 0 & 0 & 0.938\end{bmatrix},$$
$$A_5 = \begin{bmatrix}0{,}99 & 0\\ 0 & 0.815\end{bmatrix}, A_6 = \begin{bmatrix}0 & 1 & 0 & 0\\ 0{,}266 & 0{,}602 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},$$
$$A_7 = \begin{bmatrix}0 & 1 & 0\\ 0{,}257 & 0{,}652 & 0\\ 0 & 0 & 0{,}928\end{bmatrix}, A_8 = \begin{bmatrix}0 & 1 & 0\\ 0{,}280 & 0{,}65 & 0\\ 0 & 0 & 0{,}469\end{bmatrix} \tag{22}$$
$$B1 = \begin{bmatrix}0{,}107\\ 0{,}013\end{bmatrix}, B2 = \begin{bmatrix}0{,}084\\ 0{,}003\end{bmatrix}, B3 = \begin{bmatrix}0{,}110\\ 0\\ 0{,}223\end{bmatrix}, B4 = \begin{bmatrix}0\\ 0{,}011\\ 0{,}012\end{bmatrix}$$
$$B5 = \begin{bmatrix}0{,}008\\ 0{,}081\end{bmatrix}, B6 = \begin{bmatrix}0\\ 0{,}129\\ 0\\ 0{,}155\end{bmatrix}, B7 = \begin{bmatrix}0\\ 0{,}090\\ 0{,}032\end{bmatrix}, B8 = \begin{bmatrix}0\\ 0{,}068\\ 0{,}161\end{bmatrix}$$

$$C1=\begin{bmatrix}0 & 1\end{bmatrix}, C2=\begin{bmatrix}0 & 1\end{bmatrix}, C3=\begin{bmatrix}0 & 0 & 1\end{bmatrix}, C4=\begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

$$C5=\begin{bmatrix}0 & 1\end{bmatrix}, C6=\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}, C7=\begin{bmatrix}0 & 0 & 1\end{bmatrix}, C8=\begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

The resolution of the different LMI conditions helps to calculate the matrixes gains of the PI multiobserver.

The KPi and KI gains are given by (23).

$$K_{p1}=\begin{bmatrix}0{,}113 & 0.0348\\ 0{,}048 & 0.1184\end{bmatrix}, K_{p2}=\begin{bmatrix}0.130 & 0.0376\\ 0.046 & 0.1183\end{bmatrix}, K_{p3}=\begin{bmatrix}0.121 & 0.0340\\ 0.025 & 0.0200\\ 0.026 & 0.0208\end{bmatrix},$$
$$K_{p4}=\begin{bmatrix}0.025 & 0.0200\\ 0.085 & 0.0679\\ 0.047 & 0.1123\end{bmatrix}, K_{p5}=\begin{bmatrix}0.148 & 0.0354\\ 0.057 & 0.0813\end{bmatrix}, K_{p6}=\begin{bmatrix}0.025 & 0.0200\\ 0.063 & 0.0503\\ 0.025 & 0.0200\\ 0.043 & 0.0349\end{bmatrix},$$
$$K_{p7}=\begin{bmatrix}0.025 & 0.0200\\ 0.072 & 0.0575\\ 0.047 & 0.1096\end{bmatrix}, K_{p8}=\begin{bmatrix}0.025 & 0.0200\\ 0.071 & 0.0572\\ 0.047 & 0.0376\end{bmatrix}$$
$$K_I=\begin{bmatrix}0.025 & 0\\ 0 & 0.020\end{bmatrix} \tag{23}$$

The implementation of the PI multiobserver helps to generate two residual equations that are given in (24).

$$\begin{cases}R_w=\hat{y}_1-y_1\\ R_{is}=\hat{y}_2-y_2\end{cases} \tag{24}$$

where Rw, Ris are respectively the speed and stator current residual signals. In this sensor FDI case, the residuals signal are the same as the estimated fault signal $\hat{f}_1$ and $\hat{f}_2$ as expressed in (25).

$$\begin{cases}R_w=\hat{f}_1\\ R_{is}=\hat{f}_2\end{cases} \tag{25}$$

yi is the ith multimodel output in no faulty case, while ŷi is the ith observed output.

At 109s a constant amplitude signal Figure 3 (11) is added as a fault signal f1 to the speed sensor output, and then this fault is eliminated at 130,4s.

At the time of 14s a constant amplitude signal represented in Figure 6 is added as a fault signal f2 to the current sensors output, then this fault is eliminated at the 21s.

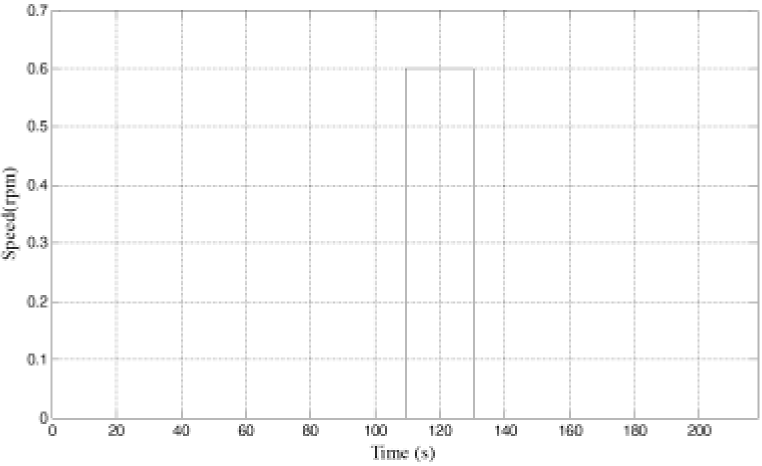


Figure 3. Evolution of speed sensor fault f1.

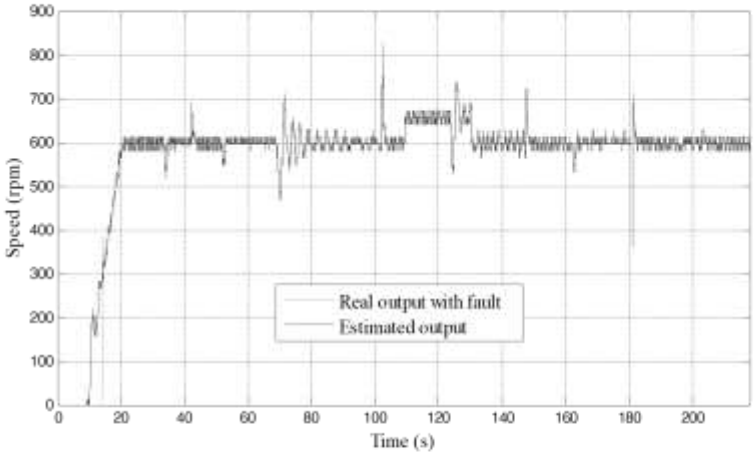


Figure 4. Evolution of real IM speed under fault and observed speed.

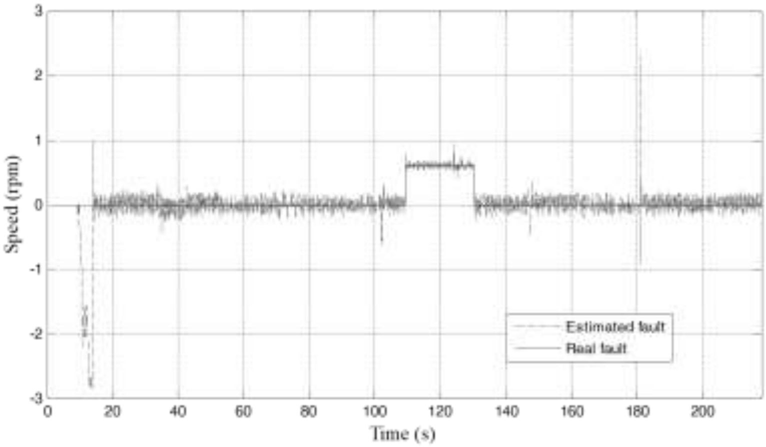


Figure 5. Speed sensor fault identification: Observed speed sensor fault and real speed sensor fault evolutions.

The figure shows that the observed speed signal follows with acceptable error the real speed affected with the fault signal depicted in Figure 4. The proposed multiobserver help not only to detect the IM sensors fault that happens between t = 109s and 130,4s (Figure 4) but also to identify these faults as shown in Figure 5 where the speed sensor fault is identified.

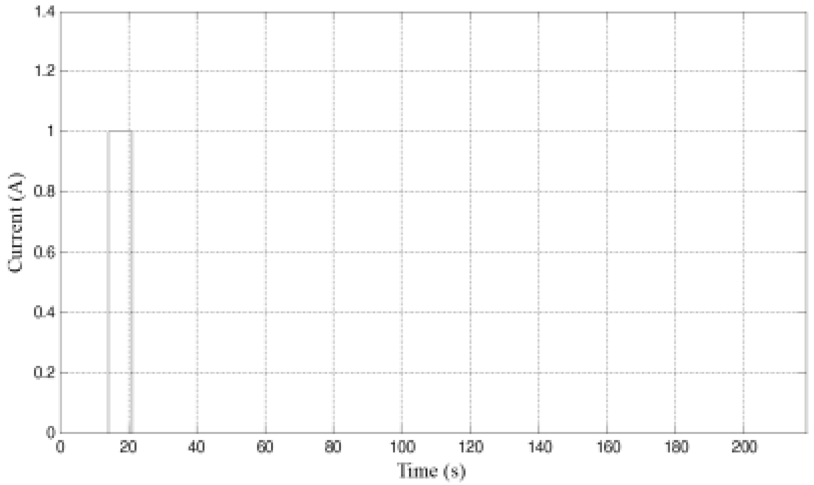


Figure 6. Evolution of current sensor fault.

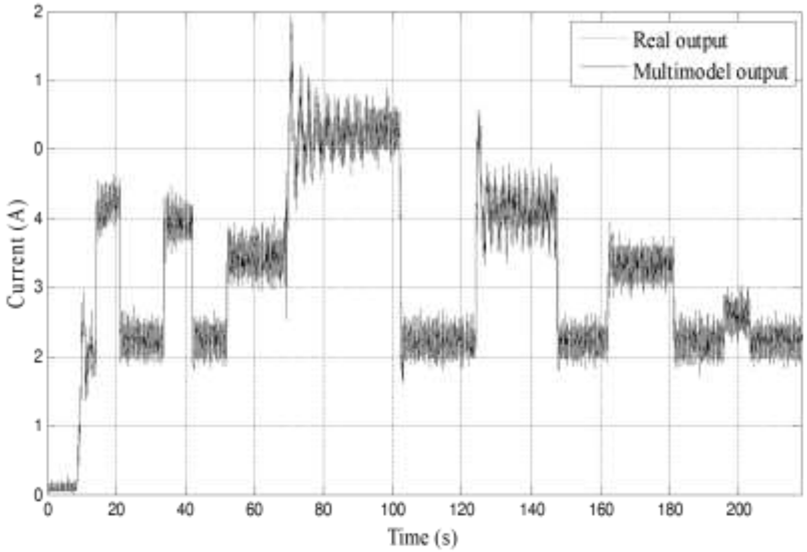


Figure 7. Evolution of real IM stator current with fault and observed one.

Similar results are reached with the current signal diagnosis. Figure 7 shows that the observed current tracks the real faulty one affected by the fault signal depicted in Figure 6. Figure 8 affirms the Stator current sensor fault identification.

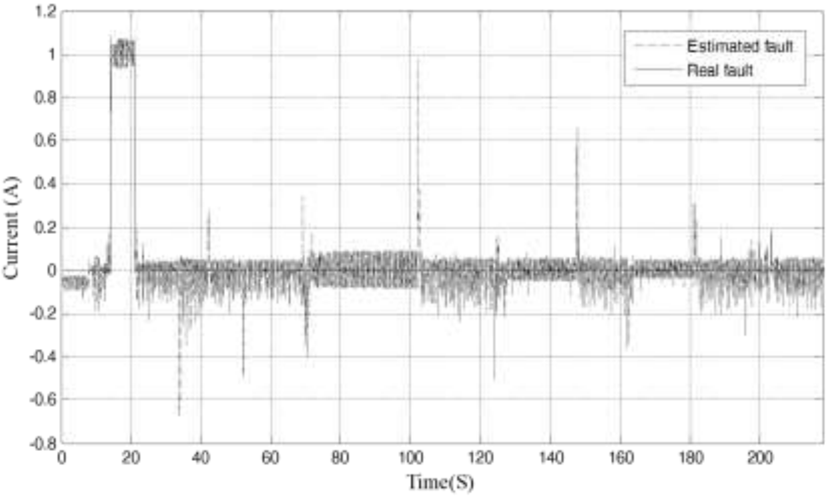


Figure 8. Observed current sensor fault and real current sensor fault evolution.

Obtained experimental results approve the performance of the fault estimation method; the two observed outputs of speed and current follow respectively the two real faulty ones with a contented error. Thus multiobserver can detect and identify the two types of IM sensors faults. The use of two independent residual that are in the same time the estimated faults guarantee the localization of the faults as each one is associated with an only sensor output.

CONCLUSION

In this chapter, we present the different failures of IM. Then, we are interested on sensors faults. A PI multiobserver has been applied for the detection and isolation of IM sensors faults. The developed diagnosis approach is based on a multi-model diagnosis. Firstly, the IM's modeling is investigated through the multi-model approach. Then considering the system's decoupled multi-model structure an adaptive PI multi-observer is synthesized. The multiobserver is synthesized exploiting the classic PI multiobserver that is modified to obtain the adaptive one. The modification consists in the multiobserver validities calculation. This multiobserver is used in the fault detection and isolation of the different sensor faults that can affect the system's outputs. The different experimental results have affirmed that the obtained multiobserver outputs follow not only the healthy real system outputs but also the faulty ones. The objectives are reached since the different computed residuals signals affirm that the detection, identification and isolation of the sensors faults are well achieved.

References

Abid, A., A. Adouni, M. Ben Hamed and L. Sbita, "A New Pv Cell Model Based On Multi Model Approach," *The Fourth International Renewable Energy Congress*, Sousse, Tunisia, pp. 20-22, December 2012.

Bello, O., Hamam, Y., Djouani K., (2014). Fuzzy dynamic modelling and predictive control of a coagulation chemical dosing unit for water treatment plants. *Journal of Electrical Systems and Information Technology*, 1: 129-143.

Didier, B., Capron, O., Odloak, D., (2013). LMI-Based Multi-model Predictive Control of an Industrial C3/C4 Splitter, *Journal of Control, Automation and Electrical Systems*, 24: 420-429.

Domlan, E., Huang, B., Xu, F., Espejo, A., (2011), A decoupled multiple model approach for soft sensors design, *Control Engineering Practice* 19: 126-134.

Garcia-Nieto, S., Martinez, M., Blasco, X., and Sanchis, J., (2008), Nonlinear predictive control based on local model networks for air management in diesel engines, *Control Engineering Practice*, 16(12): 1399-1413.

Hai-liang, H., Yang Tong-guang, "A Novel Method of Induction Motor Stator Fault Diagnosis," Electrical Machines and Systems, 2005. ICEMS 2005. *Proceedings of the Eighth International*, 3: 2254-2258.

Hamdi, H., Rodrigues, M., Mechmeche, C., Theilliol, D. and BenHadj Braiek, N., (2012). Fault detection and isolation in linear parameter-varying descriptor systems via proportional integral observer. *International Journal of Adaptive Control and Signal Processing*, 26: 224-240.

Khalid, D., Soumia Elhani, Said Cuedira and Nabil Ngote (2014), "Fault Diagnosis in Induction Motor Using Motor's Residual Stator Current Signature Anaysis," *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, pp. 631-643.

Latrach, C., Kchaou, M., Rabhi, A., and El Hajjaji, A., (2015). Decentralized Networked Control System Design Using Takagi-Sugeno (TS) Fuzzy Approach. *International Journal of Automation and Computing*, 12:125-133.

Men, Z., Yee E., Lien F. S., Ji, H., Liu, Y., (2014). Bootstrapped Multi-Model Neural-Network Super-Ensembles for Wind Speed and Power Forecasting. *Energy and Power Engineering*, 6, 340-348.

Orjuela, R., Marx, B., Ragot, J., and Maquin, D., (2008), "State estimation for nonlinear systems using a decoupled multiple mode." *International Journal of Modelling Identification and Control*, (1): 59-67.

Orjuela, R., Marx, B., Ragot, J., and D. Maquin (2010), Fault diagnosis for nonlinear systems represented by heterogeneous multiple models, 2010 *Conference on Control and Fault Tolerant Systems*, Nice, France, 600-605.

Samko, G., Ebrahim Babaei, Mohammad Bagher Bannae Sharifian, Zahed Golabi, “Application of Speed, Rotor Flux, Electromagnetic, Load Torque Observers and Diagnostic System in a Vector-Controlled High-Power Traction Motor Drive,” *Arabian Journal for Science and Engineering*, vol. 39, no. 4, pp. 2979-2996, November 2013.

Shankar, H., P. L. N. Raju, K. Ram Mohan Rao (2012). Multi Model Criteria for the Estimation of Road Traffic Con-gestion from Traffic Flow Information Based on Fuzzy Logic. *Journal of Transportation Technologies*, 2: 50-62.

Venkat, A. N., Vijaysai P., and Gudi, R. D., (2003), Identification of complex nonlinear processes based on fuzzy decomposition of the steady state space. *Journal of Process Control*, 13(6): 473-488.

Yang, H., Jiang B., Cocquempot V., “Fault tolerant control design for hybrid system,” *Lecture Notes in Control and Information Sciences*, Springer, Berlin, 2010.

INDEX