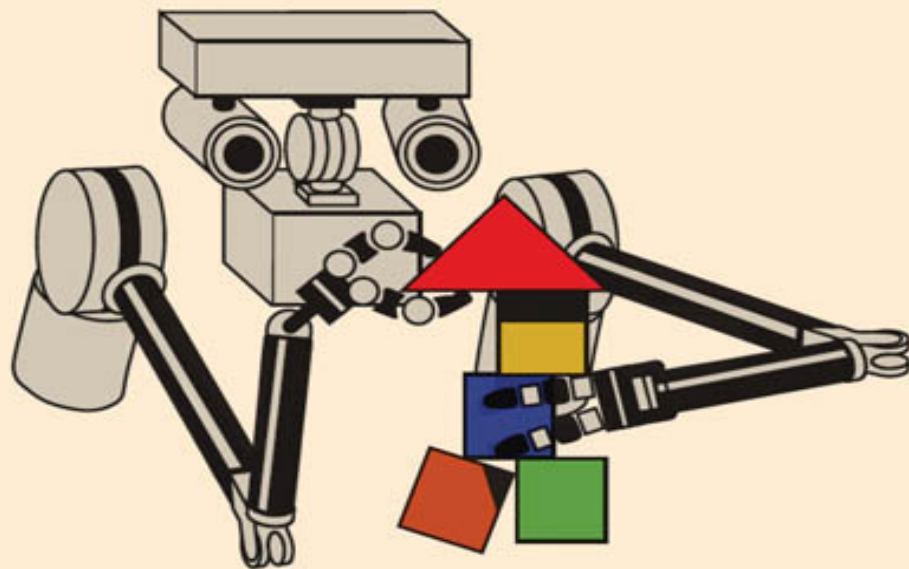


THE
DEVELOPMENTAL
ORGANIZATION OF
ROBOT BEHAVIOR



RODERIC A. GRUPEN

Intelligent Robotics and Autonomous Agents

Edited by Ronald C. Arkin

A complete list of the books in the Intelligent Robotics and Autonomous Agents series appears at the back of this book.

The Developmental Organization of Robot Behavior

Roderic Grupen

**The MIT Press
Cambridge, Massachusetts
London, England**

© 2023 Roderic Grupen

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

Library of Congress Cataloging-in-Publication Data

Names: Grupen, Roderic A., author.

Title: The developmental organization of robot behavior / Roderic A. Grupen, Laboratory for Perceptual Robotics Department of Computer Science, University of Massachusetts Amherst.

Description: Cambridge: The MIT Press, [2023] | Series: Intelligent robotics and autonomous agents series | Includes bibliographical references and index.

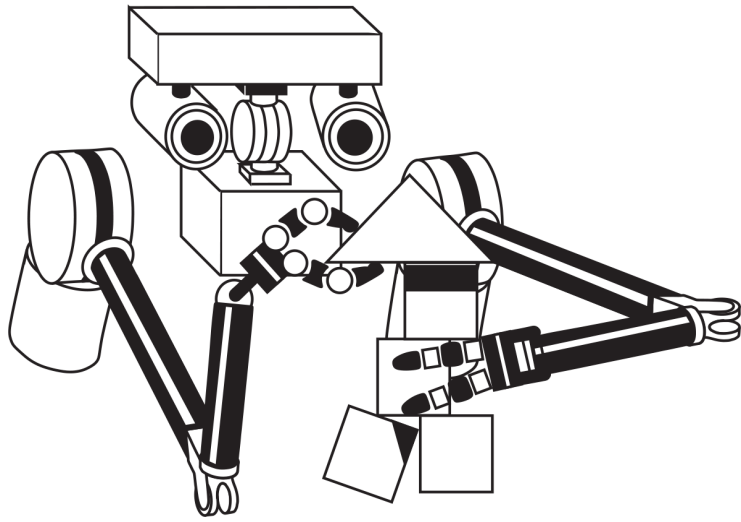
Identifiers: LCCN 2020033973 | ISBN 9780262073004 (hardcover)

Subjects: LCSH: Autonomous robots. | Cooperating objects (Computer systems) | Machine learning.

Classification: LCC TJ211.495.G77 2021 | DDC 629.8/92—dc23

LC record available at <https://lcn.loc.gov/2020033973>

d_r0



Contents

Preface

Acknowledgments

1 Introduction

1.1 Knowledge and Representation

1.2 Embodied Cognitive Systems

1.3 Developmental Robotics

Example: Learning to Walk: A Developmental Conspiracy

1.4 Frontiers in Robotics

1.5 Organization of the Book

1.6 Exercises

I MOTOR UNITS

2 Actuation

2.1 Muscle

2.1.1 The Contractile Proteins

2.1.2 The Sliding Filament Model

2.1.3 Active and Passive Muscle Dynamics

2.2 Robot Actuators

2.2.1 Permanent Magnet DC Electric Motors

Example: Torque-Speed Calculation

2.2.2 Hydraulic Actuators

- 2.2.3 Pneumatic Actuators
- 2.2.4 Emerging Actuator Technologies
- 2.3 Exercises

3 Closed-Loop Control

- 3.1 The Closed-Loop Spinal Stretch Reflex
 - 3.1.1 Spinal Processing
 - 3.1.2 Motor Nuclei
- 3.2 The Canonical Spring-Mass-Damper
 - 3.2.1 Equation of Motion: The Harmonic Oscillator
 - 3.2.2 Stability: Lyapunov's Direct Method
 - Example: Stability Analysis for the Spring-Mass-Damper*
- 3.3 Proportional-Derivative Feedback Control
 - 3.3.1 A Primer for Laplace Transforms
 - 3.3.2 Stability in the Time-Domain
 - 3.3.3 The Transfer Function, SISO Filters, and the Time-Domain Response
 - Example: Closed-Loop Oculomotor Transfer Function*
 - 3.3.4 The Performance of Proportional-Derivative Controllers
 - Example: Controlling Eye Movements*
- 3.4 Exercises

Part I Summary: Muscles, Motors, and Control

II STRUCTURE IN KINODYNAMIC SYSTEMS

4 Kinematic Systems

- 4.1 Terminology
- 4.2 Spatial Tasks
 - Example: Kinematic Description of Roger-the-Crab*
- 4.3 Homogeneous Transforms
 - 4.3.1 Translational Components

- 4.3.2 Rotational Components
- 4.3.3 Inverting the Homogeneous Transform
- 4.4 Manipulator Kinematics
 - 4.4.1 Forward Kinematics
 - Example: Forward Kinematics of the Planar 2R Manipulator*
 - 4.4.2 Inverse Kinematics
 - Example: Geometric Inverse Kinematic Solution for the Planar 2R Manipulator*
- 4.5 Kinematics of Stereo Reconstruction
 - 4.5.1 Pinhole Camera: Projective Geometry
 - 4.5.2 Binocular Localization: Forward Kinematics
 - Example: Stereo Localization in the Plane*
- 4.6 Hand-Eye Kinematic Transformations
- 4.7 Kinematic Conditioning
 - 4.7.1 Jacobian
 - 4.7.2 The Manipulator Jacobian
 - Example: First-Order Velocity Control for the Planar 2R Manipulator*
 - Example: Velocity and Force Ellipsoids for Roger*
 - 4.7.3 Stereo Localizability
 - Example: Roger's Oculomotor Jacobian and Stereo Localizability*
- 4.8 Kinematic Redundancy
 - Example: Self-Motion Manifold*
- 4.9 Exercises

5 Hands and Kinematic Grasp Analysis

- 5.1 The Human Hand
- 5.2 Kinematic Innovations in Robot Hands
- 5.3 Mathematical Description of Multiple Contact Systems
 - 5.3.1 Screw Systems

Example: Twist Constraints on Object Mobility in a Planar Grasp

5.3.2 The Grasp Jacobian

5.3.3 Contact Types

5.3.4 The Generalized Grasp Jacobian

Example: The Two-Contact Grasp Jacobian

5.3.5 Grasp Performance: Form and Force Closure

Example: Solving for Forces in Force Closure Grasps

5.4 Exercises

6 Dynamics of Articulated Systems

6.1 Newton's Laws

6.2 The Inertia Tensor

Example: Rotational Moment of Inertia

6.2.1 The Parallel Axis Theorem

Example: Translating the Center of Rotation

6.2.2 Rotating the Inertia Tensor

6.3 The Computed Torque Equation

Example: Dynamic Model of Roger's Eye

Example: Dynamic Model of Roger's Arm

6.3.1 Simulation

6.3.2 Feedforward Control

6.3.3 Analysis: The Dynamic Manipulability Ellipsoid

Example: Gravity and Roger

6.4 Exercises

Part II Summary: The Kinodynamic Affordances of Embodied Systems

III STRUCTURE IN SENSORY FEEDBACK

7 Stimuli and Sensation: Organs of Visual and Tactile Perception

7.1 Light

7.1.1 Image Formation

- 7.1.2 The Evolution of the Human Eye
- 7.1.3 Photosensitive Image Planes
- 7.2 Touch
 - 7.2.1 Cutaneous Mechanoreceptors
 - 7.2.2 Robotic Tactile Sensing
- 7.3 Exercises

8 Signals, Signal Processing, and Information

- 8.1 Sampling Continuous Signals
 - Example: Spectral Properties of the Human Voice*
 - 8.1.1 The Sampling Theorem
- 8.2 Discrete Convolution Operators
 - 8.2.1 Spectral Filtering
 - 8.2.2 Frei and Chen Signal Decomposition Operators
 - 8.2.3 Noise, Differentiation, and Differential Geometry
 - Example: Edge Sharpening*
- 8.3 Structure and Causality in Signals
 - 8.3.1 Gaussian Operators
 - 8.3.2 The Gaussian Pyramid: Blobs
 - 8.3.3 Multi-Scale Edges, Ridges, and Corners
- 8.4 Exercises

Part III Summary: Transducers, Signals, and Perceptual Structure

IV SENSORIMOTOR DEVELOPMENT

9 Infant Neurodevelopmental Organization

- 9.1 The Evolution of the Brain
- 9.2 Hierarchy in the Neocortex
- 9.3 Neurodevelopmental Organization
 - 9.3.1 Limbic Reflexes: Visceral, Vegetative, and Behavioral
 - 9.3.2 Spinal- and Brainstem-Mediated Reflexes
 - 9.3.3 Bridge Reflexes

- 9.3.4 Postural Reflexes
- 9.3.5 Maturational Processes
- 9.4 Developmental and Functional Chronology in the First Year
- 9.5 Sensory and Cognitive Milestones
 - 9.5.1 Sensory Performance
 - 9.5.2 Cognitive Development in the Sensorimotor Stage
- 9.6 Exercises

10 A Computational Framework for Experiments in Developmental Learning

- 10.1 Parametric Closed-Loop Reflexes
 - 10.1.1 Potential Functions: ϕ
 - 10.1.2 Closed-Loop Actions: $\phi|_{\tau}^{\sigma}$
 - 10.1.3 A Taxonomy of Parametric Actions
 - Example: Manipulability Reflex*
 - 10.1.4 Co-Articulation: Multi-Objective Control
 - Example: The Mechanics of Human Finger Movement*
 - 10.1.5 States: $\gamma(\phi, \dot{\phi})$
 - Example: Representing Grasp Dynamics*
- 10.2 A Multimodal Landscape of Attractors
 - Example: Multi-Objective Visual Inspection Task*
 - 10.2.1 Reinforcement Learning in a Landscape of Attractors
 - 10.2.2 Skills
- 10.3 Exercises

11 Case Study: Learning to Walk

- 11.1 *Thing*: A Quadruped
- 11.2 Controllers and Control Combinatorics
- 11.3 Locomotion Controllers
 - 11.3.1 Aggregate State Representation
 - Example: Logical Organization of Locomotor Skills*
- 11.4 Learning the ROTATE Skill
- 11.5 The STEP Skill

- 11.6 Hierarchical WALK and NAVIGATE Skills
- 11.7 Developmental Performance: Hierarchical Gross Motor Skills

Part IV Summary: Foundations for Hierarchical Skills

Appendix A Tools for Linear Analysis

- A.1 Linear Algebra
- A.2 Matrix Inverse
- A.3 Definiteness
- A.4 Hessian
- A.5 Matrix Norms
- A.6 Quadratic Forms
 - Example: Plotting the Quadratic Form*
- A.7 Singular Value Decomposition
- A.8 Scalar Condition Metrics for Linear Transforms
 - A.8.1 Minimum Singular Value
 - A.8.2 Condition Number
 - A.8.3 Volume
 - A.8.4 Radius
 - Example: Scalar Conditioning Metrics Applied to Roger's Arm*
- A.9 The Pseudoinverse
- A.10 Linear Integral Transforms
 - A.10.1 Complex Numbers
 - A.10.2 Fourier Transform
 - A.10.3 Laplace Transform
 - Example: Laplace Transform of an Exponential Function $f(t)=e^t$*
 - Example: Laplace Transform of the Unit Step Function $f(t)=1, t \geq 0$*
- A.11 Time-Domain Responses for the Harmonic Oscillator
 - Example: Time-Domain Response of the Spring-Mass-Damper*

Example: The Root Locus Diagram for the PD Control System

- A.11.1 Frequency-Dependent Amplitude and Phase Response
- A.11.2 Stiffness and Impedance

Appendix B The Dynamics of Kinematic Chains

- B.1 Deriving the Inertia Tensor
- B.2 Inertial Coordinate Frames
- B.3 Rotating Coordinate Systems
- B.4 Newton-Euler Iterations
 - B.4.1 Propagating Velocities in Open Kinematic Chains
 - B.4.2 Propagating Force in Open Kinematic Chains
 - B.4.3 The Outward-Inward Iteration

Example: The Computed Torque Equation for the Planar 2R Manipulator

- B.5 Lagrangian Mechanics

Appendix C Numerical Methods for Solving Laplace's Equation

Example: A Collision-Free Arm Controller for Roger

Bibliography

Index

List of Figures

Roger-the-Crab

A mobile manipulator concept for exploring robotics, control, signal processing, learning, and behavior.

Figure 1.1. Using a circular network of definitions from *Webster's Dictionary*, Deb Roy [246] explains that to use such symbolic networks, at least one of the references must be grounded (adapted from [246]).

Figure 1.2. Thelen's ontogenetic landscape. Time in the right column is in units of months after birth and denotes approximately when the behavior begins to be observed (estimated from figure 9.18). (Adapted from [274].)

Figure 2.1. Myosin molecules consist of multiple molecular subchains that participate in specialized roles. (Source: Adapted from [45, 192])

Figure 2.2. Structure of striated muscle. (Source: Adapted from [192])

Figure 2.3. The Huxley model of a single actin-myosin interaction undergoing a constant force contraction. (Source: Originally due to [124], adapted from [192])

Figure 2.4. A qualitative description of the isometric twitch, unfused, and fused tetanic responses in larger mammalian muscles subject to periodic activation at 5, 20, 40, and 60 [Hz]. (Source: Adapted from [192])

Figure 2.5. Qualitative force capacity of muscle as a function of length and velocity. Tension T is normalized by the maximum developed tension (T_{max}), power P by maximum power output (P_{max}), length l by muscle free length (l_0), and velocity v by the maximum unloaded contractile velocity (v_{max}). (Source: Adapted from [192])

Figure 2.6. *The Huxley model combining active and passive muscle dynamics*

Figure 2.7. A charge q moving along a conductor at velocity \mathbf{v} through a magnetic field, \mathbf{B} , from north to south magnetic poles

Figure 2.8. Moving electric charges create a torque on the coil due to the Lorentz force

Figure 2.9. Faraday's law describes how electrical current is generated in response to a mechanical input energy

Figure 2.10. The iron core DC motor

Figure 2.11. Electromechanical model of the DC motor

Figure 2.12. Plot colors and the corresponding vertical axes illustrate a family of performance curves relating motor speed, current, power, and efficiency to the torque generated by a DC motor

Figure 2.13. The compound load of the motor-gearhead combination

Figure 2.14. The hydraulic servo valve directs high pressure hydraulic fluid from a reservoir to drive a piston: (*left*) flow for extending the linear actuator; (*right*) flow for retracting the actuator.

Figure 2.15. The Sarcos GRLA (General Large Robot Arm)

Figure 2.16. The jet-pipe servo valve for controlling pneumatic actuators

Figure 2.17. The Utah/MIT dextrous hand with polymeric tendons and analog control box

Figure 2.18. The operation of an air muscle

Figure 2.19. The 2001-era “Hand B” Integrated Shadow Hand-Arm System (www.shadow.org.uk/)

Figure 3.1. The segmental structure of the spinal cord, the corresponding sensory and motor regions of the body, and the cross-sectional anatomy of the cord applied to the withdrawal reflex. (Source: Adapted from [28])

Figure 3.2. The α - and γ -motor feedback loops that regulate muscle tone and stiffness. (Source: Adapted from [192])

Figure 3.3. The mechanisms that contribute to the quadriceps femoris stretch reflex (L4) in response to the tapping of the patellar tendon

Figure 3.4. The spring-mass-damper (SMD) system

Figure 3.5. The free body diagram of the SMD system

Figure 3.6. A trajectory of an asymptotically stable harmonic oscillator on a bowl-shaped Lyapunov function (*left*) and the top-view projection on the phase plane (*right*)

Figure 3.7. Qualitative state space trajectories of the harmonic oscillator

Figure 3.8. The 1 DOF revolute (1R) mechanism is the inertial object in a PD control process with reference θ_{ref}

Figure 3.9. The transfer function for the generic harmonic oscillator in the form of a SISO filter

Figure 3.10. Series and parallel composition of transfer functions and the closed-loop transfer function (CLTF)

Figure 3.11. The musculature for the eye’s lateral (pan) degree of freedom

Figure 3.12. A control model for the 1-DOF oculomotor system consists of active and passive elements

Figure 3.13. A sequence of compositions yields the closed-loop transfer function for the PD controlled, oculomotor system

Figure 3.14. The unit step reference input

Figure 3.15. The response of the second-order PD position controller in figure 3.8 as a function of ζ ($K = 1.0$ [Nm/rad], $I = 2.0$ [kgm²]) given boundary conditions $\theta_0 = \dot{\theta}_0 = 0$ and $\theta_\infty = 1.0$ [rad]

Figure 3.16. The response of the second-order PD position controller to $\cos(\omega t)$ input function, where ω is the driving frequency for various values of ζ

Figure 4.1. Coordinate frames commonly used to specify robot tasks

Figure 4.2. The kinematic definitions and intermediate coordinate frames used to define Roger

Figure 4.3. Two coordinate frames related through a pure translation

Figure 4.4. Two coordinate frames related through a pure rotation

Figure 4.5. Projecting \hat{x}_C onto the x - y plane of frame **B**

Figure 4.6. Two coordinate frames related through a rotation and a translation

Figure 4.7. The planar 2R manipulator

Figure 4.8. The reachable workspace for the planar 2R robot

Figure 4.9. The Unimate PUMA 560

Figure 4.10. The human shoulder, arm, and wrist. Three intersecting revolute axes in the wrist create an approximately spherical wrist joint important for brachiation in human ancestors

Figure 4.11. A geometric construction for simplifying the inverse kinematic solution

Figure 4.12. The complete inverse kinematic mapping for the planar 2R manipulator

Figure 4.13. The pinhole camera geometry

Figure 4.14. Top view of a 2D stereo geometry where depth is encoded exclusively in stereo disparity

Figure 4.15. The binocular imaging geometry with independently panning cameras. On the right, the stereo geometry is defined in terms of parameter

$\gamma = \theta + \phi$, the sum of the eye configuration, θ , and the angular offset from the fovea, ϕ

Figure 4.16. Multi-sensor models of space and hand-eye coordination. This illustration uses a different coordinate system than other Roger examples, and arms/hands have been removed from images for clarity.

Figure 4.17. The Jacobian of the 2R manipulator transforms sets of executable velocities in configuration space from posture (θ_1, θ_2) into sets of achievable velocities in Cartesian space from endpoint position (x, y)

Figure 4.18. Singularities in the 2R manipulator

Figure 4.19. The 2D planar manipulator and a specific endpoint velocity command

Figure 4.20. The velocity ellipsoid in terms of the eigenvalues and eigenvectors of $\mathbf{J}\mathbf{J}^T$ for the planar 2R manipulator

Figure 4.21. The manipulator conditioning ellipsoids $\mathbf{J}\mathbf{J}^T$ for Roger's 2R manipulator

Figure 4.22. Triangulation parameters for Roger's stereo system

Figure 4.23. Scaled localizability ellipsoids for Roger's stereo geometry

Figure 4.24. Redundant solutions for the planar 3R manipulator at $x=-1$, $y=\sqrt{2}$

Figure 4.25. Internal motions along the self-motion manifold of the planar 3R manipulator with $l_1 = l_2 = l_3 = 1$ and fixed endpoint position at $x=-1.0$, $y=\sqrt{2}$

Figure 5.1. A palm view of the right hand shows the finger's intrinsic interossei muscles containing as many as 120 muscle spindles per gram of tissue

Figure 5.2. Cutkosky's taxonomy of manufacturing grasps (Source: [68]. Reprinted with permission.)

Figure 5.3. A newspaper photo of the computer-controlled mechanical hand built by Heinrich Ernst at MIT under the direction of Claude Shannon

Figure 5.4. Freddy was designed and built by a team of researchers at the University of Edinburgh: Pat Ambler, Harry Barrow, Chris Brown, Rod Burstall, Gregan Crawford, Robin Popplestone (pictured), and Stephen Salter

Figure 5.5. The Hanafusa manipulator. (Source: Adapted from [102])

Figure 5.6. The Okada manipulator (Source: [217]. ©IEEE. Reprinted with permission.)

Figure 5.7. The Lian, Peterson, Donath manipulator. (Source: [173])

Figure 5.8. The nine degree of freedom Stanford/JPL hand [250] equipped with fingertip load cells on the end of a five degree of freedom GE P50 robot arm

Figure 5.9. The DLR Hand II. (Source: Courtesy of DLR, Institute of Robotics and Mechatronics)

Figure 5.10. A 2D Cartesian hand-object system. (Source: Adapted from Kobayashi [148]. © MIT Press)

Figure 5.11. The object frame \mathbf{O} and the local contact frames \mathbf{C}_i , $i=1, 3$ oriented with respect to the local surface normal

Figure 5.12. Two contacts form a grasp on an ellipsoidal object. (*Left*) The geometry of a pair of contact frames relative to the object frame. (*Right*) The magnitude of forces λ aligned with the contact frames.

Figure 5.13. The planar projection of the three-dimensional grasp in figure 5.12

Figure 6.1. A single element of a set of point masses comprising a planar lamina. The \hat{z} axis of frame \mathbf{O} is directed out of the plane of the figure.

Figure 6.2. A rectangular body with total mass m rotates about the \hat{x} axis of coordinate frame \mathbf{A}

Figure 6.3. An eccentric mass distribution that rotates about an axis that passes through the origin of frame \mathbf{A}

Figure 6.4. Moving the center of rotation to the center of mass

Figure 6.5. The simple dynamic model describing Roger's eye

Figure 6.6. The dynamic parameters of Roger's arm relative to inertial shoulder frame $\mathbf{0}$

Figure 6.7. The feedforward compensator linearizes and decouples multiple degree of freedom controllers

Figure 6.8. A smaller and dramatically underpowered pair (\mathbf{L} and \mathbf{R}) of Roger's 2R manipulators yields dynamic manipulability ellipsoid ($\mathbf{J}\mathbf{M}^{-2}\mathbf{J}^T$)

for $l_1 = l_2 = 0.25$ [m], $m_1 = m_2 = 0.2$ [kg], and $\tau^T \tau \leq 0.005$ [$N^2 m^2$]. Gravity acts in the negative \hat{x} direction.

[Figure 7.1](#). Projective geometry and image formation

[Figure 7.2](#). Refraction at an optical interface. An incident light ray traveling in air and an optical interface with $n > n_{air}$.

[Figure 7.3](#). Geometric parameters for the biconvex lens

[Figure 7.4](#). Milestones in the evolution of the human eye: (a) the eyespot consists of a layer of photosensitive cells; (b) a pit forms that makes the primitive retina directionally sensitive; (c) the pinhole eye filled with water to create sharper images; (d) transparent aqueous humor fills cavity with transparent corneal covering; (e) the lens forms; and (f) the human eye developed a separate iris, focusing musculature, and spherical eyeball enclosed in a bony socket. (Source: Adapted from [100] and [28])

[Figure 7.5](#). The extrinsic muscles for the oculomotor system. The oculomotor nerve innervates the inferior oblique; the inferior rectus; the medial rectus; and the superior rectus. The trochlear nerve innervates the superior oblique, and the abducens nerve innervates the lateral rectus. The inset at the bottom shows the displacement of the pupil due to the exclusive action of the muscles.

[Figure 7.6](#). Some of the cutaneous sensory neurons that provide feedback signals to the somatosensory cortex

[Figure 7.7](#). Raibert's VLSI tapered tactile cell. (Source: Adapted from [238])

[Figure 7.8](#). Hillis's elastic contact resistance sensor. (Source: Adapted from [114]. © MIT Press)

[Figure 7.9](#). Boie's capacitive tactile sensor [19]

[Figure 7.10](#). Begej's optical tactile sensor. (Source: [35] © IEEE)

[Figure 7.11](#). The Dario et al. artificial skin. (Source: Adapted from [69])

[Figure 8.1](#). The open-closed quarter-wave resonance cavity

[Figure 8.2](#). An analysis of the sound pressure level as a function of frequency using the Computerized Speech Lab. The signal is the phrase "the rainbow passage." (Source: Courtesy Dr. M. Andrianopoulos)

[Figure 8.3](#). An analysis of sound pressure level as a function of frequency for a prolonged [a] vowel sound using the Computerized Speech Lab (Source: Courtesy Dr. M. Andrianopoulos)

[Figure 8.4](#). The F_1 vs. F_2 chart for 10 vowel sounds sampled from men, women, and children (Source: Adapted from [277])

[Figure 8.5](#). The continuous spatial function $g(x)$ is sampled by a sequence of regularly spaced Dirac delta functions $f(x)$ to create the discrete approximation $h(x)$ of signal $g(x)$.

[Figure 8.6](#). The effects of sampling on reconstruction

[Figure 8.7](#). Local computation of image features

[Figure 8.8](#). The performance of a simple low-pass filter

[Figure 8.9](#). The Fourier transform of the $\text{rect}(x)$ function shows why it behaves as a low-pass filter

[Figure 8.10](#). The distinguishing appearance of an edge can depend on the overall shape of the intensity surface rather than the magnitude of its first derivative alone

[Figure 8.11](#). Second-order N -jets can be used to sharpen edges in signals by identifying the inflection point in the intensity function.

[Figure 8.12](#). Fourier transform pairs for one-dimensional, derivative of Gaussian operators at three scales $\sigma = 1, 2, 4$: (a) the Gaussian low-pass operator, (b) noise reduction and differentiation in the gradient operator, and (c) the Laplacian (second-derivative) of Gaussian (LoG) operator in one-dimension

[Figure 8.13](#). An image of a field of sunflowers (reprinted by permission from [175])

[Figure 8.14](#). A part of the Gaussian Pyramid for the image of a field of sunflowers

[Figure 8.15](#). The top 200 scale-space blob responses in the scale-normalized Laplacian of Gaussian (LoG) pyramid (reprinted by permission from [175])

[Figure 8.16](#). Edges and ridges reported at $\sigma = 1.0, 16.0,$ and $256.0,$ respectively (reprinted by permission from [175])

[Figure 8.17](#). The result of automatic scale selection to detect edge (*left*) and ridge (*right*) invariants (reprinted by permission from [175])

Figure 8.18. The result of automatic scale selection to detect corner invariants (reprinted by permission from [175])

Figure 9.1. The afferents and efferents of the spinal cord

Figure 9.2. The evolution of the human brain

Figure 9.3. The sensory homunculus

Figure 9.4. The visual cortex

Figure 9.5. The motor cortex

Figure 9.6. The flow of information and control from sensory to motor areas of the brain

Figure 9.7. Hierarchies in the sensory and motor organization of human behavior

Figure 9.8. The Moro reflex

Figure 9.9. The palmar grasp reflex

Figure 9.10. The plantar reflex

Figure 9.11. The asymmetric tonic neck reflex

Figure 9.12. The Galant reflex

Figure 9.13. The Landau response

Figure 9.14. The symmetric tonic neck reflex triggered by neck extension

Figure 9.15. The lateral propping reflex

Figure 9.16. The parachute reflex is a bilateral extension to brace for a fall

Figure 9.17. The emergence and subsequent inhibition of several developmental reflexes

Figure 9.18. A developmental sequence—a chronicle of some of the milestones involving postural stability and the use of the hand during the first 12 months. (Right) Gross motor development. (Left) Some of the early fine motor developmental milestones.

Figure 10.1. Critical points in potential function ϕ , distinguished by the value of the *curvature* of ϕ at the critical point

Figure 10.2. The Utah/MIT dexterous hand

Figure 10.3. Maximum manipulability model for the Utah/MIT finger. (Left) Level curves. (Right) Scalar field $\phi(x,y) = \max_{\theta} \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ and streamlines

inside the reachable workspace generated by [equation 10.7](#).

[Figure 10.4](#). A geometrical interpretation of the gradient and nullspace of artificial potential ϕ_1

[Figure 10.5](#). Cascaded nullspace projections create the $c_3 \triangleleft c_2 \triangleleft c_1$ multi-objective controller

[Figure 10.6](#). Three distinct control responses for the same initial and final configurations: (a) the PD control (second-order) arm response; (b) the steepest descent in configuration space; and (c) the steepest descent in Cartesian space

[Figure 10.7](#). A subordinate POSTURAL action based on the manipulability index is projected into the nullspace of the superior Cartesian endpoint controller

[Figure 10.8](#). The evolution of a two-contact grasp using the moment residual control gradient. (*Left*) Three intermediate two-contact geometries en route to the final grasp configuration and their respective $(\phi, \dot{\phi})$ coordinates on the phase portrait for this pair of faces. (*Right*) Multiple distinct basins of attraction when different pairs of faces are considered (reprinted by permission [59]).

[Figure 10.9](#). A simulated planar grasp of a square object. States are indicators of membership in models m_i , $i=0, 4$. X denotes “unknown,” “T” is the thumb, “1” is the index finger, and “2” is the middle finger.

[Figure 10.10](#). The region of attraction R_1 and the equilibrium set $E_1 \subset R_1$ for navigation function ϕ_1 corresponding to controller c_1

[Figure 10.11](#). The region of attraction surrounding E_1 can be extended by sequential control. A sequential combination of c_1 and c_2 can be configured to approach minimum E_1 from all states in the interior of $R_1 \cup R_2$.

[Figure 10.12](#). A visual inspection sequence traverses an aspect transition graph defined by $[\gamma_g \ \gamma_l \ \gamma_m \ \gamma_r]$ states and POSTURAL, TRACK, and SEARCH actions in \mathcal{A} to read the barcode on packages

[Figure 11.1](#). *Thing*, a 12-DOF quadruped

[Figure 11.2](#). A single leg in contact with an external, horizontal surface creates a moment about the platform’s center of mass

Figure 11.3. Two different three-legged stances that share legs 1 and 3. Only one of the two stances formed by legs $(l_1l_2l_3)$ and $(l_0l_1l_3)$ can be stable at a time.

Figure 11.4. Learning performance for the ROTATE gait. (Left) A learning curve. (Right) Greedy performance.

Figure 11.5. The ROTATE policy with contingencies for a variety of run-time contexts. The central cycle has transition probabilities of greater than 95% in the training context.

Figure 11.6. The 0-1-2-3 leg control sequence in the ROTATE policy accumulates approximately 45 [deg] per cycle

Figure 11.7. Performance of the STEP skill during learning (left) and the greedy performance (right) with exploration turned off

Figure 11.8. (Left) Performance of WALK skill learning in the three cases considered. (Right) Percentage of actions that were derived from ROTATE and STEP as opposed to primitives in each of the three problem designs.

Figure 11.9. Trajectories toward $x=1.4$ [m] and $y=1.6$ [m] from a variety of initial positions. The final heading of the platform is not specified.

Figure 11.10. Behavior generated by the integrated, hierarchical mobility controller for *Thing*

Figure 11.11. A summary of the developmental staging of an integrated navigation behavior. In stage 1, the acquired ROTATE policy achieves 0.35 [rad/action] in 11 [min]. In stage 2, STEP achieves 1.3 [mm/action] in 180 [min], and in stage 3, WALK achieves 1.53 [mm/action] in 480 [min].

Figure 11.12. A demonstration of walking over irregular terrain. Steps (white) are 1 [cm] high.

Figure A.1. The geometrical interpretation of the vector product

Figure A.2. A set of inputs on the left, $\mathbf{x} \in \mathcal{U}$, maps to the set of outputs on the right, $\mathbf{y} \in \mathcal{E}$, via the linear mapping $\mathbf{y} = \mathbf{A}\mathbf{x}$

Figure A.3. Scalar kinematic conditioning metrics for Roger's planar 2R arm configuration. Each panel shows the range of the respective metric from black (minimum) to white (maximum).

Figure A.4. The complex plane

Figure A.5. The unit step input function

Figure A.6. The response of the second-order PD position controller as a function of ζ ($K=1.0$ [N/m], $m=2.0$ [kg]) given boundary conditions $x_0=\dot{x}_0=0$ and $x_\infty=1.0$

Figure A.7. The root locus for the continuum of responses underlying figure 3.15 when $K=1.0$ [Nm/rad], $I=2.0$ [kgm²], and $-3.5 \leq B \leq 3.5$ (corresponding to $-1.24 \leq \zeta \leq 1.24$)

Figure A.8. The magnitude (a) and the phase response (b) of the second-order SMD position controller evaluated at $s=\pm i\omega$ for various values of ζ

Figure A.9. The transfer functions describing the (a) stiffness and (b) compliance of the SMD controller

Figure A.10. The transfer functions describing the (a) impedance and (b) admittance relations for the SMD controller

Figure B.1. (Left, A stack of planar lamina approximates the distribution of mass in a three-dimensional object. (Right) The contribution of a single lamina to the inertia tensor describing rotation about \hat{a} .

Figure B.2. Velocity in an inertial frame due to a time varying rotation matrix

Figure B.3. Free body diagram of link i

Figure B.4. Propagating velocity and acceleration into a noninertial coordinate frame

Figure B.5. The propagation of forces in a kinematic chain

Figure B.6. Stages in the Newton-Euler iterations for a 3R planar manipulator ($n=3$). The equations defining the iteration denoted by the blue and red block arrows are summarized in table B.1.

Figure B.7. The planar 2R manipulator with discrete mass distribution

Figure B.8. The equivalence principle applied to the planar 2R manipulator

Figure C.1. The mapping of the Cartesian occupancy grid to the configuration space occupancy grid

Figure C.2. Gradient descent on a harmonic function yields a collision-free motor plan for Roger's right arm. (Left) The Cartesian space. (Right) The c-space.

List of Tables

[Table 5.1](#). Contact types and the degree of freedom remaining at the contact interface

[Table 5.2](#). Point contacts with and without friction and the soft finger contact

[Table 7.1](#). Sensory signal performance on the fingertip

[Table 8.1](#). Gradient (First Derivative) Operators

[Table A.1](#). Fourier Transform Pairs

[Table A.2](#). Laplace transform pairs

[Table B.1](#). Newton-Euler iterations.

Preface

The technician in charge, guided by the directions for testing in the *Handbook of Robotics*, said, “How are you?” The indicated answer was to have been, “I am well and ready to begin my functions. I trust you are well, too,” or some trivial modification thereof. ... He was instantly jolted by the nature of the LNE-Prototype’s voice. It had a quality like no robotic voice he had ever heard.... So surprising was this that it was only after several moments that the technician heard, in retrospect, the syllables that had been formed ... They were, *Da, da, da, goo*.

—“Lenny,” by Isaac Asimov [9]

Robots have been used to entertain, to teach, and to manufacture goods. Examples of such systems extend back thousands of years and have sparked discussions about autonomy, free will, and the difference between thoughts and actions. Despite this long, multifaceted history, the vast majority of successful robotics applications today rely on very little sensor feedback and the modern-day equivalent of a mechanism used in eighteenth-century music boxes.

Now, sophisticated sensors can be used to provide many types of continuous feedback. Nevertheless, the average robot today still accumulates very little *experiential knowledge* over time regarding interactions with the world and engages in little or no reflection about whether things are going well and are likely to continue doing so in the future. A manufacturing robot, for example, may get a signal that a pallet of parts has arrived that triggers preprogrammed strategies for picking up each part and placing it in the required position. There is a lot of knowledge encoded in this response to the signal, but none of it resides in the robot itself—it is almost exclusively the contribution of the human programmer.

Consequently, manufacturing robots excel in repetitive tasks and highly structured environments where uncertainty is assiduously avoided.

New applications are being considered, however, in situations where *more* autonomous robots are required to interact with *less* well-structured (natural) environments. These applications will challenge the ability of external programmers to anticipate all the situations that might arise—the robots themselves will have to own the knowledge behind their actions. In these contexts, robots must participate in the discovery, organization, and reuse of control knowledge acquired over time in the course of direct experimental interactions with the world.

One of the principle challenges facing this new robot is computational complexity—the complexity of the robot itself and of the unstructured environment. In this setting, when faced with a new task, there exists a huge variety of sensory and motor alternatives with different expected outcomes and rewards relative to a task. Moreover, some of the events on which expected rewards depend may not be in evidence or even directly observed.

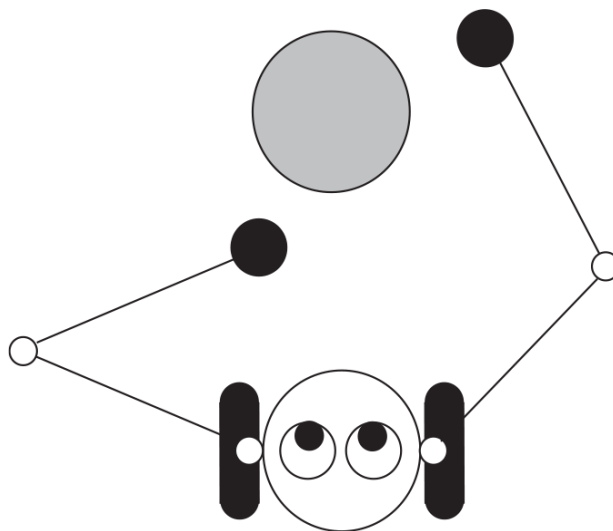
How do animals cope so successfully with this overwhelming complexity? The collection of processes underlying development are certainly part (or most) of the answer. We focus on three principal observations of the sensorimotor developmental stage: developmental mechanisms exploit morphological properties of the infant that support basic skills in low-dimensional actions, they improve as additional resources are engaged, and development translates a huge amount of tacit information encoded in the genome into implicit, actionable forms.

Embodiment presents a kinodynamic and perceptual setting that encodes important innate structure. Properly activated by stimuli in the environment, this combination of opportunities for effective action built into a body can be composed to create hierarchies of acquired behavior (or skills) that exploit these aptitudes, avoid kinodynamic limitations, and serve as *roadmaps* through an otherwise intractable state space. In the process, patterns of stimuli that activate roadmaps are discovered and used to carve up the complicated world into recognizable affordances for behavior. This process is clearly observable in animals of several species in the sensorimotor developmental stage when young animals explore their aptitudes and represent the world in terms of opportunities to express them.

We can test these insights from developmental psychology and neurology by creating hypotheses that run on robots. In this book, we focus on the first stage of human postnatal development, the sensorimotor stage, wherein policies for postural stability, locomotion, and object manipulation begin to emerge. This text explores what robotics says about the aptitudes of embodiment and what psychology and neurology can tell us about the sensorimotor stage to create a principled computational account of development that can be applied to all embodied systems. Such a framework provides a synthetic, controllable experimental basis for evaluating theories in developmental psychology and, at the same time, will lead to better, smarter robots.

Introducing Roger

At several places in the text, we make use of a simple robot named Roger to reinforce the discussion. Roger is the descendant of another, simpler visual and motor system also named Roger-the-Crab by Paul Churchland [53] to study the kinematics of hand-eye coordination. Churchland's Roger consisted of a stereo vision system and a simple planar manipulator with two joints that he used to show how independent spatial observers can be related through a simple neural map relating Roger's hands and eyes.



Roger-the-Crab

A mobile manipulator concept for exploring robotics, control, signal processing, learning, and behavior.

We use a more elaborate Roger as a pedagogical device for acquiring a practical understanding of the concepts discussed in the book. Roger moves about in flatland (the x - y plane) by actuating two drive wheels in a differential steering arrangement. He has two planar arms, each incorporating two rotational degrees of freedom—one in the shoulder and one in the elbow. A control program can apply a torque to each of these joints. There are tactile sensors at the endpoint of each arm and in the body that measure the magnitude and direction of the net contact force acting on these bodies. Roger has two eyes (pinhole cameras) that can pan independently. The eyes produce a stereo RGB (red-green-blue) image pair.

In total, Roger has eight degrees of freedom (two wheels, two eyes, and two arms) and each DOF has sensors that report the current position and velocity of the joint. The relationship between applied torque and accelerations in the joints is governed by the body posture, the state of motion, and equations describing the dynamics of the robot—Roger must contend with significant inertial forces during controlled movement.

Intended Audience

Robotics is a unique discipline in that it affords us the ability to study sophisticated machines and computation simultaneously in a platform where we can precisely control experimental studies of intelligence. The discussion in the book begins by studying the embodied systems on which development depends—in particular, how embodied structure can be used to compensate for the lack of structure in the open environment. The discussion attempts, wherever possible, to compare and contrast human bodies with their robotic counterparts. We explore the mathematical foundations of sensory and motor systems, that might normally be covered in a first course in robotics (kinematics, dynamics, actuators, sensors, control, and signal processing). Much of the supporting mathematics is introduced in the appendices so as not to interfere with the flow of the book. The more advanced (and specialized) material presented (knowledge representations, biological sensors, muscles and mechanisms, developmental neurology, potential fields, learning, and redundant systems) is critical to the overall thesis of the book. It provides much of the

representational framework that can be used to model development in robot systems.

In addition to roboticists, this book might be of interest to

- developmental psychologists and neurologists who wish to acquire a mathematical understanding of kinematics, dynamics, control, and stability—essential characteristics of developing human infants;
- biomechanists and kinesiologists who typically have a strong background in kinematics and dynamics and can benefit from the discussions of peripheral sensors, control, neurology, redundancy, and learning;
- behavioral neurologists who wish to get an introduction to computational models for describing control, learning, and development;
- computer scientists engaged in knowledge representations and learning; and
- cognitive scientists who wish to explore the role of embodiment.

The text has been used in an undergraduate course entitled Introduction to Robotics, which caters to computer scientists and engineers as an introduction to the mathematics underlying robot systems, perception, and control. The content presented in chapters 2, 3, 4, and 6 represent the core material for many undergraduate introductions to robotics. A recent offering at the University of Massachusetts used chapters 1 (introduction), 2 (actuation), 3 (control), 4 (kinematics), 6 (dynamics), and 8 (signal Processing), with selected subjects from other chapters and eight written and related programming assignments.

The complete text (with liberal use of the written and programming assignments) has been used successfully in a graduate course (primarily computer science and engineering students) that focused on the special properties of embodied machine learning systems. For this audience, it was possible to cover all the material (albeit, at a slightly aggressive pace) in 32 hours of lecture and approximately 6–8 hours per week of independent work outside of the classroom. The text has also been used for graduate seminars engaging computer scientists, engineers, psychologists, and students interested in cognitive science and philosophy. In this case, the classes typically focused on the sections of the text that covered the physiology, biomechanics, and cognitive attributes of human infants

(chapters 1, 2.1, and 3.1) as well as the proposed computational architecture and the case study regarding hierarchical locomotion skills (chapter 10). These classes also included supplemental material from other texts in developmental psychology.

Several appendices are included to provide more complete introductions to the mathematical concepts used in the body of the text. Appendix A provides a selective primer to linear algebra and integral transforms. Appendix B introduces common methods for deriving the dynamic equation of motion for articulated systems. Appendix C reviews the basics of numerical relaxation.

Our focus excludes a number of popular topics such as path planning, geometrical reasoning, and SLAM (simultaneous localization and mapping) as well as many new and specialized topics in robotics. We have done so to emphasize the analog to biological systems and because excellent texts already exist for these subjects. Moreover, a number of related topics will not be pursued in any depth because to do so would lengthen the discussion significantly. Human behavior depends ultimately on a hierarchy of concerns from segmental reflexes to cortical sensory and motor processes, social behavior, and cultural institutions. A complete account of development must consider the influence of all these factors and is beyond the scope of this text; these include the relationship between development and language, emotion, social aspects of development (imitation and adult scaffolding). We will also omit a discussion of modeling at the neural level, despite the close relationship between computational learning methods and neuroscience. However, we expect that the focus of the book will have an increasingly important role in all of these issues as research proceeds—it is our position that cooperation, communication, and social behavior can emerge from exactly the same developmental processes that eventually lead to mastery of the physics of movement and manipulation.

Acknowledgments

In 1996, a collaboration among several faculty at UMass Amherst led to a research theme focused on understanding infant human development. I attended this influential seminar with Andy Barto, Paul Cohen, David Jensen, and Rich Sutton in the computer science department and Carole Beal, Neil Berthier, Rachael Keen, and Gary Marcus in the psychology department. Over time our common interests transformed into a particularly interesting series of seminars and, finally, focused research projects to explore the question more deeply. In my lab (the Laboratory for Perceptual Robotics), this research concerned a computational framework for sensory and motor processes that could support autonomous learning and world modeling using what we understood about important developmental processes—our goal was to create a developmentally organized, embodied computational system. This effort remains active today and much of it has found its way into the contents of this book. I sincerely thank these friends and colleagues for their expertise, patience, and hard work as we negotiated a common language for these compelling questions and for creating such a rich and fertile environment at UMass for scientific research.

I would also like to acknowledge colleagues from Johnson Space Center (Robert Ambrose, William Bluethmann, Myron Diftler, Bob Savely) and Vanderbilt University (Richard Alan Peters) who were important collaborators on numerous breakthroughs in grasping, manipulation, and humanoid robots that inform the ideas in this text. I am indebted to these friends for decades of interesting (and fruitful) research.

The book is informed by the work performed in the Laboratory for Perceptual Robotics over the past 20 years. Herein are the result of thousands of enjoyable discussions with student collaborators in the LPR and the UMass Computer Vision Laboratory (Ed Riseman and Al Hanson). Collaborators who were students in the LPR over this period deserve special thanks for sharing their curiosity and creativity on these projects for a time. They include Elizeth Araujo, Ravindran Balaraman, Rohan Bandaru, Sanuj Bhatia, Don Berkich, Alyx Burns, Jeff Cleveland, Chris Connolly, Jefferson Coelho, Patrick Deegan, Ashish Deshpande, Khoshrav Doctor, Andy Fagg, Miles Gepner, Debasmita Ghose, Hia Ghosh, Vijay Gullapalli, Junzhu (Janet) Guo, Ed Hannigan, Steven Hart, Kjeldy Haugsjaa, Mitchell Hebert, Gary Holness, Emily Horrell, Manfred Huber, Haoyu Ji, Deepak Karrupiah, George Konidakis, Li Yang Ku, Scott Kuindersma, Rakesh Kumar, Michael Lanighan, Tiffany Liu, Ziyang Liu Will MacDonald, Sarah Osentoski, Shichao Ou, Devdhar Patel, Justus Piater, Rob Platt, Kelly Porpiglia, Srinivas Ravela, Michael Rosenstein, Dirk Ruiken, Shiraj Sen, Prakhar Sharma, Grant Sherrick, Soumitra Sitole, Kamal Souccar, Alenna Spiro, Andrew Stout, Anmol Suag, John Sweeney, Takeshi Takahashi, Bryan Thibodeau, Tulsi Vembu, David Wheeler, Jay Ming Wong, Dan Xie, and Oscar Youngquist. I very much enjoyed all those animated arguments over the years. Several extended visitors to the lab contributed important ideas and experiments. among them Young-Jo Cho, Antonio Morales, Cosimo Distanto, and Luiz Marcos Garcia Gonçalves.

Numerous collaborators from federal agencies and corporations should be acknowledged whose engagement and support over more than 20 years has contributed significantly to this text. In particular, this text owes a great deal to support from DARPA (Mark Swinson, Douglas Gage) in the MARS and SDR programs and (Robert Mandelbaum) in the ARM-S program, NASA (Robert Savely) in GCT program, NSF (Howard Moraff for early career support, mentoring, and encouragement), and the Microsoft Corporation and Tandy Trower in particular for inspiration and project support.

Finally, I owe a profound and heartfelt thank-you to my family—Mary, Yianni, and Niko—for inspiring me and for their support, and to Doris and Bill Grupen (Mom and Dad) for encouragement (and feedback on early drafts).

1

Introduction

Thousands of years ago, early roboticists may have been engineers engaged in state-of-the-art public works or priests and philosophers inspired by mechanical automata to consider the relationship between mind and body. Over many centuries, an increasingly diverse set of scientists and engineers have remained attracted to this experimental platform as a means of advancing science and technology in many different fields. Rapid recent advances in our ability to build better and more sophisticated robots have made them influential in increasingly important economic and social roles as well.

Case in point: Robots are now an established tool in modern manufacturing. However, most successful applications of manufacturing robots still depend on programmer-specified states, actions, and precisely controlled run-time contexts. In the manufacturing application, even small deviations from the design context usually lead to expensive retooling and reprogramming. The robotics revolution in manufacturing has been realized by imposing a great deal of structure—essentially, restricting the run-time environment until it meets constraints imposed by the automation. The credit for progress in robotics, thus far, is due largely to new materials, sensors, mechanisms, actuators, batteries, computing power, open-source code repositories, and brilliant engineers. Very little credit can be attributed to cognitive breakthroughs in the robots themselves. In fact, one could say that robots suffer from a severe form of savantism—a developmental disorder in which their impressive physical ability is in stark contrast to their overall cognitive disability.

Today, roboticists are anticipating applications where robots will become more commonplace in close proximity to human beings and provide services that support daily human activities in unstructured human environments. Unlike the manufacturing setting, robots in unstructured environments (*in the wild*, so to speak) must have multiple kinds of sensors to differentiate run-time situations and must, themselves, construct control contingencies for dealing with many of these situations. These robots will grow increasingly skilled at solving common problems in familiar situations.

Since the 1950s, a growing community of scientists, combining insights from psychology, artificial intelligence, linguistics, neuroscience, computer science, anthropology, and philosophy, have focused specifically on theories of intelligence. In 1973, Christopher Longuet-Higgins coined the term *cognitive science* to describe the study of the processes of *thought* by supporting testable theory and experiment as prescribed in the scientific method like other natural sciences. However, cognitive science often considers disembodied processes and disregards the significant contribution of the *body* to knowledge and representation. In addition to creating more flexible robots, studying knowledge representation in robotics advances a platform for a *synthetic* approach to cognitive science that may well lead to advances in that field as well. *Cognitive robots*—those that use stored representations of the robot itself and the interactions it has with a complex environment—must function adequately, as humans do, with incomplete information. They must learn and transfer what they know to new situations, substituting experience and cognitive structure for the spatial and temporal structure of the manufacturing line.

A computational perspective on cognitive robotics quickly reveals major theoretical challenges related to the scalability of machines that learn. Nikolai Bernstein [23], credited with the invention of a field of study he called “biomechanics,” described “dexterity” as a quality of intelligent behavior denoting skill, flexibility, and creativity in sensory and motor problem-solving. In his book, Bernstein noted that

mammals have relatively many more *singular, aimed movements* of attack, hunting, and so on. All these actions are not stereotyped, differing from one occasion to another and displaying very accurate and quick *adaptability*. They manifest progressive increase in an ability to quickly create *new, unlearned motor combinations* exactly fitting an emerging situation.

Bernstein endorsed the idea that modular skills—recombined in many ways with little effort to represent many contingencies—may have an advantage over comprehensive, high-dimensional, and fully integrated behavior. He elaborated the argument by identifying multiple independently useful nonprehensile skills underlying the prehensile skills of primates:

accurate, precise, and strong *targeted movements*, such as aiming, touching, grasping, accurate and strong *striking*, far and precise *throwing*, accurate and carefully planned *pressing*. These simple movements ... gradually developed into numerous *meaningful chain actions*, such as the handling of *objects* [and] the using of *tools*.

To Bernstein, dexterity depends on the ability to create new motor combinations, rapidly and efficiently, in new situations. It is supported by unique idiosyncratic skills and strategies that can be different in every individual and it reveals a deep understanding of problems and solutions. The concept of a rich and flexible sensory and motor substrate giving rise to cognitive representation of the external world is an essential part of human intelligence—it's likely to be an important key to creating new, multifunctional dexterous robots as well. Bernstein's concept of dexterity quickly confronts what he called the “degrees of freedom” problem, wherein perceptual and mechanical flexibility combined with environments that are uncertain and unstructured leads to too many choices for control. A computational perspective on dexterity suggests that the complexity of deciding how to act in this context will be formidable, and yet the quality of dexterity admired by Bernstein and others (quick-wittedness, adroitness) is synonymous with computational efficiency.

Infants of many species, however, demonstrate the acquired ability to act dexterously by relying on the structure imposed by embodiment together with memories of past experiences. How can we understand the structure introduced by embodiment, and how should memory be organized to support dexterity? Mathematical tools commonly used in robotics to describe the kinematics and dynamics of multiple degree of freedom, actuated systems provide a solid basis for analyzing the aptitudes and limitations of embodied developmental systems. In addition, computational models of animal learning provide powerful machine learning tools for robots that learn by exploration—interacting directly with the domain. These tools provide some input into the content and organization of memory necessary to support efficient responses to unexpected run-time

conditions. Still, a developmental curriculum is required that guides exploration toward skills on the developmental frontier. For example, an aspect of growth related to neurological maturation manages sensory and motor resources over time to control the complexity of developmental learning and control. Computational approaches to resource scheduling and model checking may, likewise, present opportunities to learn incrementally and in situ.

1.1 Knowledge and Representation

Nowhere is the difference between animals and robots more conspicuous than in their comparative abilities to acquire, represent, and reuse knowledge. Knowledge can be represented in different ways to support different kinds of informative exchanges with the environment and to facilitate different kinds of reasoning [70, 184].

Explicit knowledge is knowledge that has been codified in a compact, symbolic, and easily articulated form. It supports communication between individuals and transfer to new tasks. Encyclopedias, dictionaries, and user manuals are all good examples of explicit knowledge representations—they presume a “common ground” between the source and the consumer of information. For example, in [figure 1.1](#) the meaning of *push* is defined by reference to *force*, which in turn is defined by circular reference to *energy* and *strength* [246]. This is how dictionaries work—language is grounded by association. Consequently, the meaning of *push* is determined only after one of these references bottoms out in something the agent *really* knows. Only then can this grounded reference to meaning be propagated throughout the semantic network in [figure 1.1](#) and to related concepts. In this sense, language is purposefully vague. When we are asked to consider the phrase “fresh-baked bread,” for example, these three syllables are instantly associated with sights, sounds, aromas, tastes, and memories of people and perhaps a kitchen from long ago or your favorite bakery. Linguistic tokens are *labels* for concepts that require the listener to provide proprietary, grounded associations to color and nuance them according to individual experience.

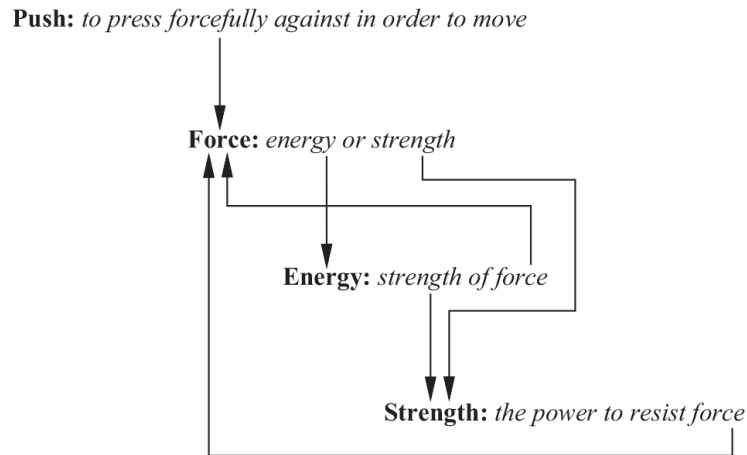


Figure 1.1

Using a circular network of definitions from *Webster's Dictionary*, Deb Roy [246] explains that to use such symbolic networks, at least one of the references must be grounded (adapted from [246]).

Skills, like riding a bike or playing the piano, are idiosyncratic and generally cannot be articulated or communicated completely between individuals. In taxonomies of knowledge representations, this is referred to as *implicit knowledge*—acquired as the agent explores direct interactions with the environment. This kind of knowledge encodes policies for transforming sensor feedback into control inputs with respect to an objective function. Implicit knowledge, represented as control circuits, for example, represents a coordinated strategy for perception and action that solves a familiar problem in the context of a range of run-time environments. Implicit knowledge cannot, in general, be completely engineered by a third party, nor can it be created in a manner that is completely independent of how it must be used. It describes patterns of co-occurring stimulation that actually occur during interaction *now* with the real world and so it avoids problems of circular semantics. Moreover, implicit knowledge represents sensory and motor interactions that unfold in reliable sequences and, thus, reflect both the agent and the environment.

Tacit knowledge is encoded in the agent's physical and precognitive configuration. It includes such things as the performance of sensors and actuators as well as the kinematic characteristics of a controllable sensor geometry. In animals, tacit knowledge is selected by objective measures of fitness in the course of genetic variation and natural selection. It is compiled into neuroanatomical structures including peripheral sensors, body size and

mass distribution, and programs for growth and maturation. Animals are born with considerable knowledge built in.

Built into the body by the genetic code are structures (hard head, strong muscles, good eyes) and functions (being able to duck without thinking, for example). Structure, function, special adaptations, instincts—all of these presets in the organism provided for by the genes—are “knowledge.” (Wilson [293])

Thus, embodiment determines the range of interactions an agent can have with the environment. It determines which aspects of this interaction can be observed and what subset of the interaction state it can control. The knowledge hierarchy rests on the foundation that tacit knowledge provides—it informs higher-level decision-making by biasing interactions toward those that exploit the agent’s kinodynamic and perceptual aptitudes, and it provides the foundations for acquired implicit and explicit conceptual structure.

1.2 Embodied Cognitive Systems

Since the middle of the twentieth century, there has been a growing interest among roboticists and cognitive scientists regarding the role of embodiment in cognitive systems [282, 162, 41, 278, 224, 275]. The emerging philosophy challenges almost 500 years of Cartesian dualist philosophy that considers acting and thinking to be separate and independent concerns requiring different principles and representations. Instead, the integrated embodied system paradigm acknowledges the active nature of cognition where movement and perception support direct experimental interactions with the world and, hence, the accumulation of *situated* knowledge. Still, it remains to be determined how the skill and knowledge that reside in control circuits like these can be learned autonomously, organized hierarchically, and acquired in an order that manages the incremental complexity of learning—this is the objective of active research in developmental programming.

1.3 Developmental Robotics

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s? If this were then subjected to an appropriate course of education one

would obtain the adult brain.
—A. M. Turing, 1950 [280]

The “curse of dimensionality” is often cited as a basic challenge in machine learning. The “curse” requires that to achieve a given coverage in sampled training data, the number of samples grows exponentially as the dimensionality of the data increases. If 10 samples are sufficient to approximate a function of one parameter, then on the order of 10^n samples will be necessary for functions that depend on n independent parameters. This gets out of hand very quickly. There are, conservatively, 200 independent degrees of freedom (DOF) in the human body. Ten unique postures for each of these degrees of freedom yields on the order of 10^{200} postural states of the body.¹ Moreover, precise movements between postures can require activating different combinations of thousands of motor units, and humans inhabit a large variety of different habitats on Earth (and occasionally, elsewhere). Searching for rewards in this enormous number of states and actions is challenging, but animals of many species routinely solve problems that are comparable in scale.

A community of *developmental roboticists* are studying principles of animal development that can explain how embodied systems can defeat the enormous complexity of cumulative learning in open, unstructured environments [290]. Observations of such systems reveal that developmental learning leverages structure tacitly encoded in the embodied system to learn from direct exploratory interactions with the world [253, 254, 193, 179] and can, therefore, address tasks that cannot be fully anticipated in advance by human engineers [117, 32, 212, 160, 109].

Computational accounts of development propose a tightly coupled relationship between learning and embodiment mediated by elective actions, perception, and cognitive representation [198, 199]. These approaches have been incorporated into experimental frameworks for cumulative learning in order to address many tasks and run-time environments [253, 75, 228, 58, 33]. Several studies (e.g., [8, 179]) state the explicit desire to influence theories of development and intelligence in biological systems embedded in natural settings as well. Some research in this area specifies a developmental scaffolding provided by a human programmer [212, 160, 108]. Others seek stronger methods and autonomous progression through developmental stages, often using

information theoretic measures to identify developmental frontiers and to determine when those frontiers should advance [267, 257].

An influential hypothesis at the boundary between robotics and developmental psychology formulates development in terms of dynamical systems theory [274, 273, 302, 141]. The developmental chronology leading to adult human walking behavior is a striking example of how the intrinsic dynamics and neurological structure of the infant team up to solve difficult learning problems during the first year of life.

Example: Learning to Walk: A Developmental Conspiracy

The earliest indications of a mature walking gait can be elicited from the newborn child. At birth, if an infant is held so that the soles of its feet touch a supporting surface and the experimenter moves the infant forward, it will raise one leg, move it forward, and lower it again—first one leg and then the other—using a motor circuit called the bipedal gait reflex [274]. To accomplish this, the infant’s nervous system triggers a pattern of muscle innervations that couple the pair of limbs through the pelvic girdle in response to proprioceptive signals from the periphery.

The gait reflex illustrates the genetic structuring of a neuromuscular and kinodynamic *synergy* in service to important locomotion tasks that lie in the infant’s future. To become an independent biped, several other developmental dominoes must fall into place. The newborn’s leg has been flexed in gestation and needs exercise to increase the range of motion and improve the strength-to-weight ratio of the extensor musculature. While this is happening, the infant grows, lengthening and hardening bones, changing the distribution of mass, and improving muscle tone as the infant acquires policies for postural stability and balancing.

The influence of growth and maturation on the performance of the developing infant is obvious in the “ontogenetic landscapes” created in the seminal work of Thelen [274]. For example, [figure 1.2](#) illustrates the emergence and differentiation of stable locomotive behavior over the course of the first year. At three months, the same experiment that elicited the bipedal gait reflex at birth will show a reflexive extension of the leg when the soles of the feet contact the supporting surface. This response is called the *positive support* reflex, and its emergence prepares the limb to support

the mass of the body. At about five months, infant growth begins to disrupt the dynamics of the bipedal gait reflex. The support and forward motion stimuli still presumably trigger the gait reflex, just as before, but the amplitude of the response is significantly attenuated in part by dynamic influence of the body itself. The positive support reflex and a related distal stiffening² dominate in this stage and produce the stiff-legged appearance characteristic of a toddler. The infant is now focused on the problem of balancing while still supporting itself using its arms and contacts with walls and furniture. As the infant ventures out on its legs, away from other forms of support, it has a characteristically wide stance and stiff knees. With a little practice, the infant begins to adapt to variation in the slope of the terrain or obstacles where their foot must fall. With additional growth, strength, and skill the gait narrows, generating greater forward velocity [39] and eventually presenting the bipedal swing gait, buttressed by the reactivated gait reflex along with an array of supporting skills.

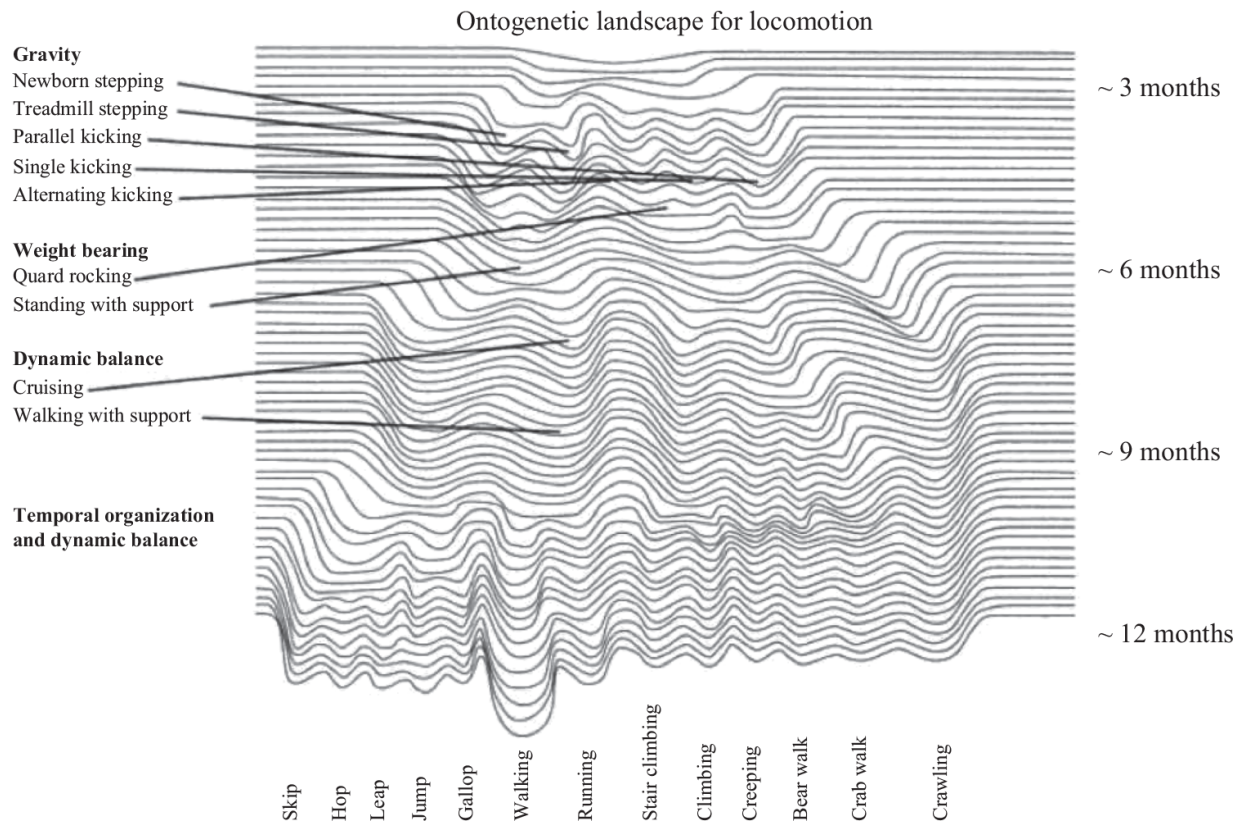


Figure 1.2

Thelen’s ontogenetic landscape. Time in the right column is in units of months after birth and denotes approximately when the behavior begins to be observed (estimated from [figure 9.18](#)). (Adapted from [\[274\]](#).)

□

In the spectrum of the research on developmental robotics described above, this book focuses on the beginning—the infant’s first year—where so much “robotics” is accomplished in the maturing infant’s brain (see [figure 9.18](#)). The infant is exploring what its body can do and where the environment affords rewarding opportunities to exercise these aptitudes. From the moment of birth, the processes that guide the sensorimotor stage of development are on full display providing the opportunity to study this foundational layer of postnatal development and focus on the nexus between developmental learning and robotics.

1.4 Frontiers in Robotics

The impact of a theory describing sensorimotor development that can be applied in synthetic (robot) systems will yield behavioral robustness and, thus, support applications of autonomous robotics that are currently beyond the state of the art. For example, the service robotics industry is projected to grow quickly with new products entering the market at an accelerating rate. In this respect, the robotics industry shares important characteristics with the automobile industry in the early twentieth century and the home computer market in the 1980s [256]. New technologies for actuation and sensing promise to make robots more responsive to unexpected events in their immediate surroundings. New system architectures are improving responsiveness and robustness, and new technologies for grasping and manipulation provide the basis for mobile manipulators that can interact manually with their surroundings to set the world into motion and do work. Coupled with maturing technologies for mapping and navigation, mobile manipulators may well evolve to serve new roles in less well-structured *human* environments. Principles of development may prove to be indispensable tool as robots acquire cognitive representations of this richer class of interactions.

One emerging industry that will benefit is health care, in particular, residential eldercare. As a whole, health care is the largest segment of the US economy and is becoming too expensive to deliver. The mainframe approach to medical care is limited in capacity and some experts are calling for new technologies that support distributed and more personalized health care. Asia and Europe are already experiencing demographic pressure that demands more efficiency and productivity in eldercare now [247]. Technology for “aging in place” includes information technology and assistive devices that can coexist with humans in home health care settings. Appropriate technology will provide affordable cognitive and physical assistance and improve the relationship of clients to caregivers, family, and community support services. Some of the tools for personalized medicine will incorporate sophisticated knowledge specific to particular clients and their homes that adapts over time to lifestyle and disability. Clearly, it would be difficult, if not impossible, to engineer this knowledge in the traditional sense. Here again, developmental principles may prove to be indispensable.

The frontier of human exploration in the twenty-first century requires long-term exposure to extreme environments on Earth, in space, and on other planets. The logistics of maintenance and supply at the threshold of these environments exposes people to significant risks. Servicing satellites in orbit and refueling interplanetary expeditions are examples of possible future roles for autonomous robots. Likewise, robots that build human habitat on the Earth's Moon and Mars without requiring constant, low-level human supervision may pave the way for permanent colonies on other planets. Autonomous machines might assist human pioneers—digging trenches, mining mineral deposits, repairing other machines, transporting equipment over large distances on challenging terrain, or moving gingerly over the surface of gossamer orbiting structures. It is difficult to imagine that human engineers can anticipate the situational knowledge that these robots will need. Once again, principles of cognitive development may support robots that adapt to their peculiar environments autonomously.

Finally, successful computational approaches to development in robots may provide new ways of understanding developmental phenomena in humans. Neuroscientists, neurologists, and developmental psychologists study the organs and processes of knowledge acquisition in humans—each discipline contributes critical pieces of the puzzle, but they are hampered by serious methodological constraints. At one extreme, neuroscientists study structure without reference to the environment or the ecological niche that it serves. At the other end of the spectrum, neurologists and developmental psychologists study integrated cognitive agents using relatively short-term empirical studies that are difficult to completely control. In either case, the resulting description of the developing human infant, for instance, is procedural rather than declarative and sheds little light on the *mechanisms* of development. A synthetic computational approach to development and knowledge acquisition in sensory and motor systems is likely to become an important new tool in these communities as well because it must rest on theories of mechanism and can be used to conduct controlled experimentation into the interactions between a cognitive agent and its environment. As a consequence, the biggest impact of research on next generation robots may well be its influence on our basic understanding of behavior in many animals and in ourselves.

1.5 Organization of the Book

Early in his book on synthetic psychology, Braitenberg [38] states the law of uphill analysis and downhill invention, which observes that it is difficult to infer the processes responsible for the behavior of an organism by observing its behavior alone—that *analysis* is harder than *invention*. A consequence of this asymmetry holds that when we analyze behavior to infer mechanism, we tend to overstate its complexity and, as in many other domains, the simplest explanations are often the best.

What I cannot create, I do not understand.

—Richard Feynman, California Institute of Technology

The epigraph above was taken from Nobel laureate Richard Feynman’s blackboard at the time of his death. When Feynman said “create,” he almost certainly did not mean it literally. Instead, he meant that, starting from scratch, a complete understanding of a concept required a firm understanding of each step of the reasoning involved and that the exercise of explaining it simply and concisely contributes a deeper appreciation for the problem. The wisdom of this insight is especially relevant to the understanding of development and its relationship to robotics, where “create” in Feynman’s quote may take a different meaning than it would to a theoretical physicist. In the context of developmental theory, “create” may literally mean building a developing robot system. This book approaches developmental robotics with this objective in mind.

If robots and computational systems are to be effective platforms for the study of human development, we should pause to consider what the successful outcome of such a study is likely to be. We should not necessarily expect the developing robot to behave qualitatively as the human infant behaves. The central hypothesis of this text holds that development is inseparable from the embodiment of the agent—the performance of motor systems, the mechanics of movement, and the performance and spatial distribution of sensors. Robots (even humanoid robots) are more like an alien species than they are a bifurcation from some existing branch of the evolutionary tree of life on Earth. The qualities of development in this strange new agent should be recognized in terms of its impact on the complexity of learning, its positive impact on the

transformation of tacit knowledge into actionable forms, and its improved ability to transfer acquired knowledge into new contexts and tasks. This book takes a strong architectural view of development as a design guide for implementing developing robot systems and, thus, achieving robust and intelligent sensorimotor machines—as Feynman might have counseled. We set modest goals that focus on the role of development in the sensorimotor stage first because it is required in embodied development, and, as a result, we discuss what unique forms of structure the robot system brings to the table. To this end, the book is organized in four parts to introduce a variety of topics from many disciplines.

Part I: Motor Units—The book begins at the end of the cognitive spectrum most responsible for *movement*. Mechanisms for actuation have a huge influence on the performance of movements. The viscoelastic properties of muscle tissue, for example, contribute significantly to movement and control, and it is clear that the dynamics of robot actuators have a similarly profound impact on the quality of movement in robot systems. In chapter 2, the active and passive properties of actuation are discussed and their influence on the quality of movement is presented.

All motor behavior, whether reflexive or highly trained, ultimately causes movement by submitting references to motor units. In vertebrates, these primitive motor units use closed-loop spinal circuitry that accelerates the animal’s limbs in order to eliminate postural errors. Chapter 3 introduces techniques from linear, closed-loop control to provide an analog of such equilibrium setpoint controllers for use in robot systems. A classical mathematical treatment of the *spring-mass-damper* system is used to derive the time- and frequency-domain response of such systems and to introduce analytical descriptions of system stability.

Part II: Structure in Kinodynamic Systems—The geometry of the sensors incorporated into the embodied agent determine their relationship to information distributed in the three-dimensional world. Consequently, the kinematic and dynamic properties of an embodied system influence access to information that is otherwise hidden in an unstructured environment. The most durable knowledge available to a robot is the reliable kinodynamics of its own body. Chapters 4 and 6 review tools for describing geometric and inertial properties of mechanisms, especially linear techniques for

identifying the kinematic and dynamic aptitudes of articulated mechanisms. These mechanisms create controlled movements and, thus, change the viewpoint and acuity of sensors. Redundant kinematic systems are discussed that solve a given problem in many different ways *and* afford the opportunity to optimize behavior over multiple performance criteria at the same time.

The distinctive human brand of cognitive ability derives largely from the dexterity of our hands. Hands impose a powerful inductive bias on learning, development, and cognitive organization. They are special in the spectrum of perceptual organs because, in addition to being an important means of deploying our sense of touch, they are used to manipulate the world in ways that expose new, hidden information. They are an important cognitive organ as well and much of this value stems from the kinematic relationship between hands and objects. Chapter 5 applies the new tools developed in chapter 4 to examine the kinematic qualities of a candidate grasp.

Part III: Structure in Sensory Feedback—New sensor devices are invented every year that measure an increasingly diverse range of signals, some of which have no biological analog. Part III starts by focusing on two important sensors that do have a biological counterpart. Chapter 7 introduces natural and artificial sensors for detecting light reflecting from environmental surfaces, introduces some of the mechanoreceptors of the human hand, and surveys some of the robot sensors designed to serve these roles in engineered systems.

Signal processing is a rich field that has a mature computational foundation. We concentrate on low-level signal processing that can produce feedback for use in closed-loop control processes and that can be applied generally to many kinds of signals. Chapter 8 discusses how information is encoded in these heterogeneous signals. Spectral analysis using the Fourier transform is introduced to consider how information is affected when continuous signals are sampled to produce feedback, and scale-space filters are discussed as a means of recovering the *deep structure* in a signal.

Part IV: Sensorimotor Development—Humans (and many other animals) enjoy a neurological organization that guides the developing infant through a sequence of sensorimotor milestones. Chapter 9 discusses developmental reflexes as well as mechanisms of growth and maturation in human infants

that serve this role while controlling the incremental complexity of sensorimotor learning. We identify several principles that support developmental algorithms for use in robot systems.

In chapter 10, we discuss how *admissible* control laws can be used to implement hierarchies of closed-loop movement primitives that submit references to embedded motor units. The *control basis* framework is employed to provide a discrete set of closed-loop movement primitives parameterized by sensory and motor resources that can co-articulate multiple objectives in a strictly prioritized way. Every combination of these resources creates a (unique) closed-loop action that submits new references to motor units and returns its internal state to supervisory programs.

Finally, this framework is demonstrated in a case study concerning a reinforcement acquiring mechanism for acquiring hierarchical skills. We sketch a framework that can incorporate policies for several other locomotion skills in a gross motor hierarchy that can *walk* and *navigate*. The examples of synthetic sensorimotor development presented in this chapter illustrate a staged developmental process in which implicit knowledge about the domain is acquired and reused to implement solutions to more complex tasks.

1.6 Exercises

1. **Knowledge types.** Are the following most appropriately considered tacit, implicit, or explicit sources of knowledge? Please support your answer.

- (a) “apples are red”
- (b) an expert golf swing
- (c) the distance between your eyes
- (d) your hair/skin color
- (e) the difference between keys and loose change in your pocket
- (f) shivering/perspiring
- (g) the ratio of the circumference of a circle to its diameter
- (h) laughing when tickled

2. **Transmitting knowledge.** Describe how tacit, implicit, and explicit knowledge is transmitted effectively between two agents. Support your

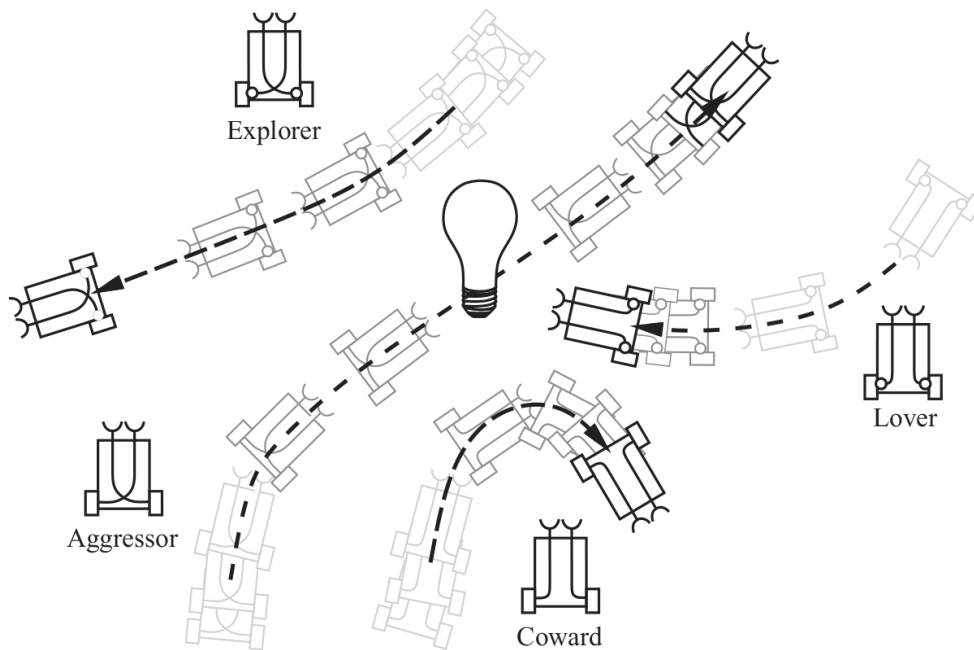
answer with examples.

3. Closed-loop processes.

(a) The field of cybernetics draws a relationship between open- and closed-loop control and analogous biological processes. Give one example each of open- and closed-loop mechanisms in the human body.

(b) Closed-loop stability is an important aspect of other environmental processes as well. Give some examples of stable closed-loop systems at the planetary scale that are critical for life on Earth.

4. **Tacit knowledge: Braitenberg vehicles.** In 1984, Valentino Braitenberg published a book entitled *Vehicles: Experiments in Synthetic Psychology* that demonstrated how tacit neurological structures can transform stimuli from complex environments into rich and complex behavior [38]. The four simplest Braitenberg “bugs” are illustrated here.



The lover incorporates ipsilateral (same side) connections between the vehicle’s sensors in the front and its wheel motors. The connection is inhibitory or excitatory (the former designated by the small circles near the wheels). A lover’s motors are inhibitory—full-on in no light, slowing as stimulation increases, and capable of reversing direction in very bright light. When more light is detected on the right photocell it slows the right motor more than the left, orienting the vehicle toward the light and causing

the trajectory in the picture. When these bugs are themselves sources of stimulation to one another, they exhibit very complex behavior that is strikingly natural in appearance.

(a) Explain the trajectories of the other three types of vehicles (the coward, explorer, and aggressor) created when other combinations of inhibitory/excitatory and ipsilateral/contralateral control logic is used.

(b) Suppose that the vehicles each carry a light source.

i. How will a pair of lovers behave?

ii. How will a lover and a coward interact? Is there a more appropriate name for these two agents in this multi-agent context?

iii. How will a lover and an explorer interact? Is there a name for the role of these two agents in this multi-agent context?

iv. Predict how a random population of hundreds of these bug types might behave in a closed environment.

5. Invent your own homework. Make a homework problem out of your favorite content in chapter 1. Write a question and a solution for it from the material in the reading. The problem can call for a short discussion, analysis, or calculation—it should not take more than 30 [*min*] to solve open-book.

1. By way of comparison, the number of atoms in the known universe is estimated to be between 10^{78} and 10^{82} .

2. A distal co-contraction reflex appears to be a common motor policy during the acquisition of new motor skills throughout the life of a human being. One can see the effect by observing a grown-up novice learning to ice skate.

I

MOTOR UNITS

2

Actuation

Definition 2.1: Actuator. *A physical device that transforms electrical, chemical, or thermal energy into mechanical energy.*

Actuators determine how quickly and smoothly a visual sensor can track stimuli; how precisely a contact force can be controlled; and how much mechanical work can be applied to the environment. Actuation is, thus, an appropriate place to begin the study of the origins of knowledge in active perceptual and motor systems.

This chapter begins by reviewing the properties of muscle that collectively lead to grace and power in biological systems. In addition, the physics and electrodynamics of permanent magnet DC motors are presented in detail followed by a brief description of other familiar and commonly used actuators. Finally, some emerging technologies that may lead to new actuation concepts in future robot systems are reviewed.

2.1 Muscle

Quite a long time ago, a specialized soft tissue evolved in a common ancestor to many animals that creates contractile forces from specialized biomolecular interactions in response to electrical stimulation. Muscle has important active and passive properties that influence the acceleration and stiffness of an actuated limb. From the perspective of a molecular biologist, an organism is a complex assemblage of many such molecular machines that move and interact with each other through malleable noncovalent

bonds. The force source in muscle is one such molecular machine based on the interaction of two important proteins, actin and myosin. This combination is ubiquitous in the animal kingdom as a means of transforming the chemical energy in cells into mechanical work. So common is this actuator in biology that working hybrid muscles can be fabricated by combining actin and myosin from different animals. Slime molds (and soil amoeba) employ a pure form of actin that when combined with meromyosin extracted from rabbit muscle will produce a kind of hybrid slime mold–rabbit muscle. Human muscle is made of the same stuff—the descendent of molecular structures inherited from some of the oldest organisms on earth.

2.1.1 The Contractile Proteins

Figure 2.1 shows the main structural characteristics of the myosin molecule at the heart of muscular contraction. Myosin consists of several molecular chains that researchers have successfully extracted from processed muscle tissue. The distinctive myosin molecule has a long tail with a double-lobed bulbous head on one end where interactions with actin take place to create contractile forces. The heavy molecular chains (heavy meromyosin or HMM) form the bulbous working end and constitute roughly two-thirds of the total molecular weight. Separate light chains have also been identified, including the alkali light chains (A-1 and A-2) and the DTNB light chains—named after the treatments that dissociate them from the rest of the molecule. The light meromyosin (LMM) tail is about 1000 [Å]¹ in length and is primarily an elastic structural element that connects the molecule into the thick filament structure (figure 2.2).

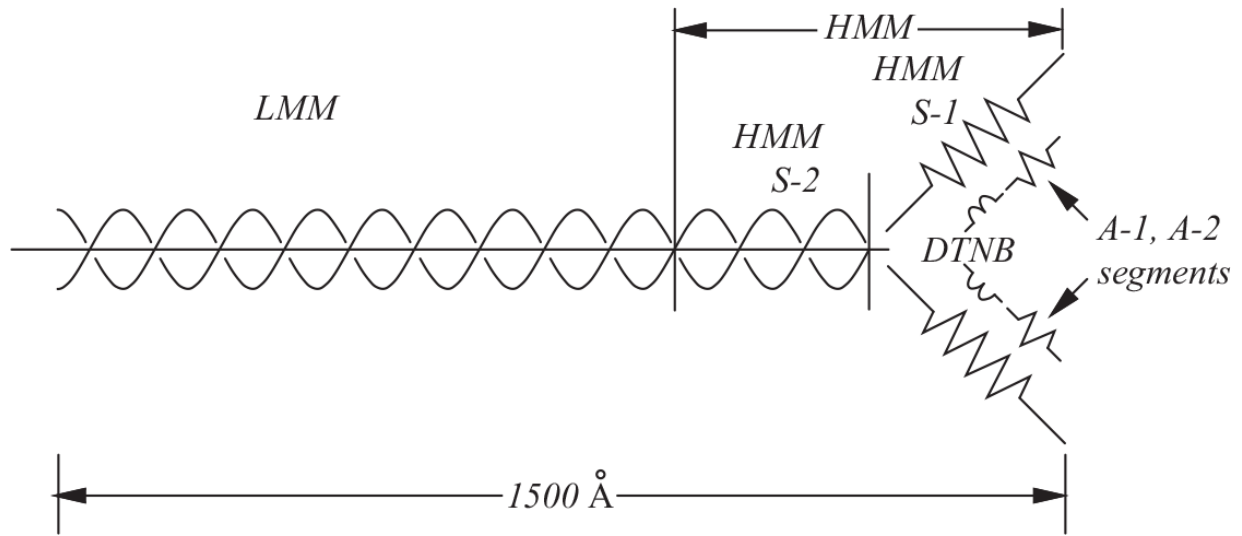


Figure 2.1

Myosin molecules consist of multiple molecular subchains that participate in specialized roles.
 (Source: Adapted from [45, 192])

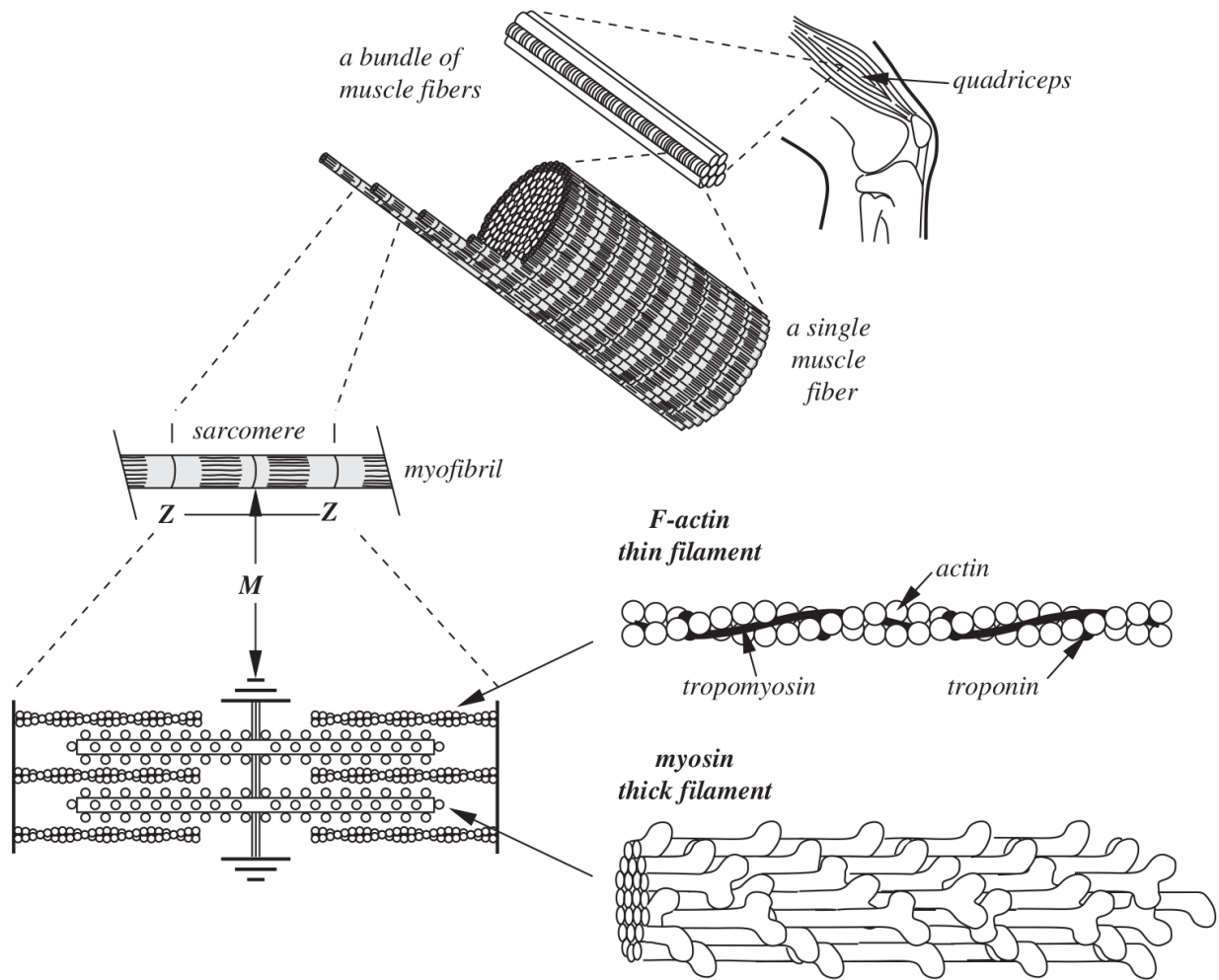


Figure 2.2
Structure of striated muscle. (Source: Adapted from [192])

The counterpart of the myosin molecule in the actin-myosin interaction is the relatively small G-actin molecule. In its polymerized form, it forms a fibrous double helix called F-actin—part of the thin filament structure (figure 2.2). Together, the myosin and actin filament structures form the central mechanism for active muscular contractile forces.

2.1.2 The Sliding Filament Model

Striated muscle tissue is a hierarchical structure that is organized to support as many actin-myosin interactions as possible per unit volume of muscle tissue. Figure 2.2 illustrates the structural hierarchy of striated muscle at several levels of detail from bulk tissue down to the molecular interactions that generate contractile forces in the quadriceps muscle. This large muscle

is made up of bundles of muscle fibers (also known as myocytes or muscle cells), each of which is composed of several myofibrils. The myofibrils are constructed from linear chains of sarcomeres—the smallest unit of actuation.

A sarcomere combines a latticework of thin (actin) and thick (myosin) filaments into a linear actuator between two disks of connective tissue called Z-disks (Z sections in [figure 2.2](#)). Neural stimuli propagate over the surface of the muscle tissue and penetrate into the sarcomeres through the Z-disk, causing the thin actin fibers to bind free calcium ions in troponin. These Ca^{++} ions prepare attachment sites on the thin filament for the S-1 subchain ([figure 2.1](#)) in the head of myosin molecules in the thick filaments. After it attaches, the head of the myosin molecule changes shape and its angle of attachment to the thin filament, producing a shear force between the thick and thin filaments [[192](#)]. If the bulk force is sufficient to cause a contraction against the load, the thin filament will slide 50–100 [\AA] relative to the thick filament, changing the relationship between actin and myosin and causing them to detach. This process is repeated asynchronously over many sites in the sarcomere to generate forces and to amplify displacements over the length of the myofibril.

[Figure 2.3](#) illustrates a model of the actin-myosin attachment involved in the contraction of striated muscle under a uniform load. The model is originally due to Sir Andrew Fielding Huxley, a Nobel prize–winning physiologist and biophysicist [[124](#), [192](#)]. The model assumes that the thick (myosin) filament is fixed at section M in [figures 2.2](#) and [2.3](#). When the thin filament moves to the right, the muscle is stretching, and when it moves to the left, the muscle is contracting. The relative position of the active head of the myosin molecule is subject to elastic forces connecting it to the thick filament with an equilibrium position defined at $x=0$ ([figure 2.3](#)) and a maximum deflection $x=h$ of the active site. When the head of the myosin molecule links to the active site on the thin (actin) filament, a *crossbridge* is formed that applies these elastic forces to the thin filament.

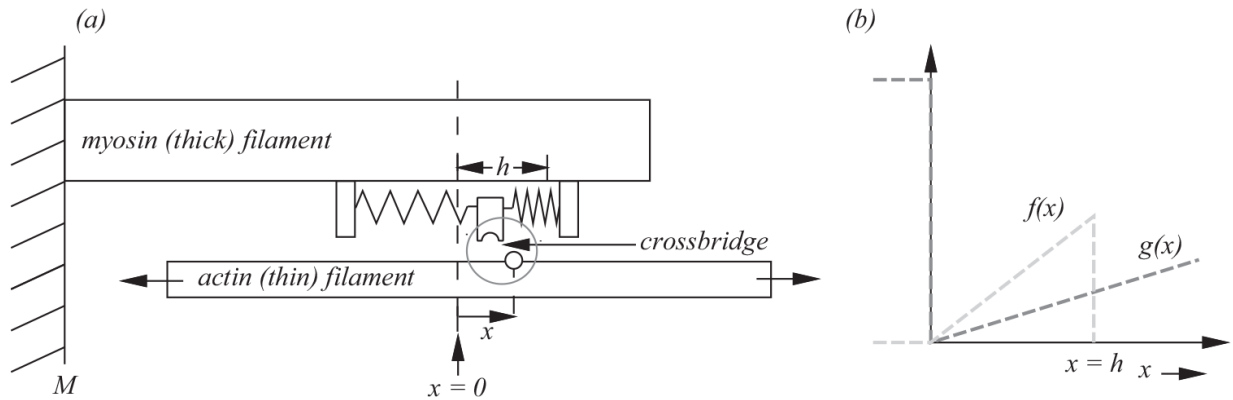


Figure 2.3

The Huxley model of a single actin-myosin interaction undergoing a constant force contraction. (Source: Originally due to [124], adapted from [192])

Let $0 \leq n(x) \leq 1$ be the probability that the crossbridge in figure 2.3 has been formed at displacement x . Huxley estimated the rate of change in the probability of attachment as the weighted sum of two kinds of events in the population of active crossbridges:

$$\frac{dn(x)}{dt} = [1 - n(x)]f(x) - n(x)g(x), \quad (2.1)$$

where $f(x)$ describes the probability that a new attachment will form where a crossbridge doesn't currently exist, and $g(x)$ describes the probability that an existing crossbridge will detach [125]. The shapes of functions $f(x)$ and $g(x)$ that Huxley chose are illustrated in figure 2.3 (b). To continue the probabilistic formulation, these functions should be normalized on the interval $[-h, h]$ so that they constitute probability distributions. By doing so, these same models can be used to describe the bulk properties of muscle as well as the behavior of a single actin-myosin interaction.

Huxley demonstrated that the shape of these functions is less important than the asymmetry about $x=0$. If $x < 0$, muscular forces would be negative, requiring muscles to *push* on tendons rather than pull. Therefore, for $x < 0$, Huxley chose the rate of attachment $f(x)=0$ and the rate of detachment $g(x) \gg 0$. Conversely, when $0 < x < h$, $f(x)$ dominates $g(x)$. These properties lead to behavior where muscles tend to shorten in response to neural excitation. While the model is very simple and says nothing about the biomolecular processes that lead to this behavior, it provides a descriptive account that predicts reasonable muscle behavior. There is a large literature

that extends the model and a more complete introduction can be found in the seminal text by McMahon [192].

So far, the discussion has focused on the *developed tension*—the *active* generation of force in muscle tissue. However, the net output of a bundle of muscle fibers (as in the quadriceps muscle in figure 2.2) depends on passive viscoelastic properties of muscle tissue as well.

2.1.3 Active and Passive Muscle Dynamics

After receiving neural activation and before any contraction is evident, muscle tissue responds by becoming resistant to stretch. After a short delay, the activated muscle begins to contract. Under isometric (constant length) conditions and subject to periodic activation, the developed tension combined with the passive properties of the tissue produces forces like those in figure 2.4.

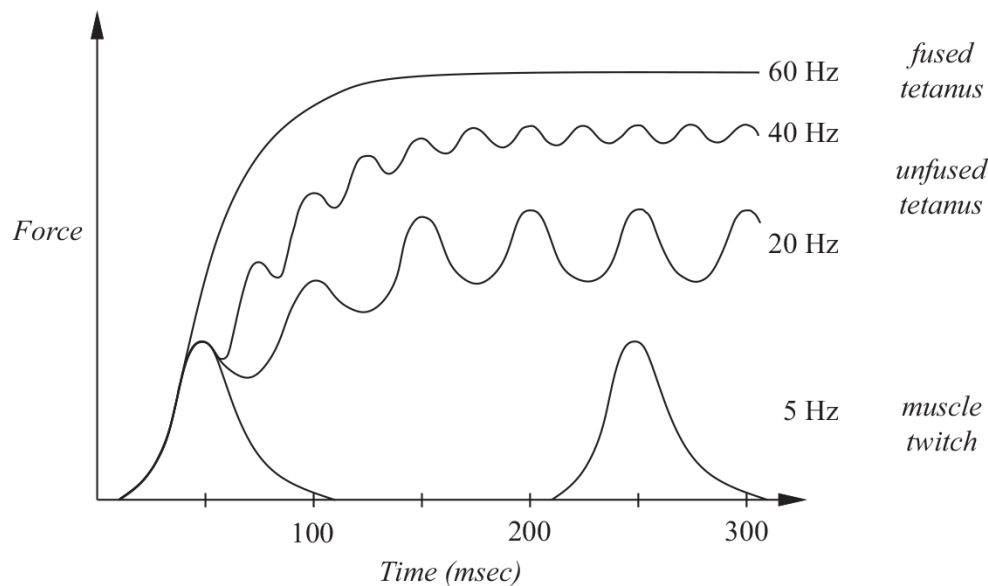


Figure 2.4

A qualitative description of the isometric twitch, unfused, and fused tetanic responses in larger mammalian muscles subject to periodic activation at 5, 20, 40, and 60 [Hz]. (Source: Adapted from [192])

For low frequency activation, a sequence of independent muscle *twitch* contractions is observed that can last anywhere between 7.5 and 100 [msec]. If the activation pulses are applied at higher frequencies, individual twitch responses overlap and add—the average steady state contractile force

increases and the amplitude of the output ripple decreases. This phenomenon is called *unfused tetanus*. In mammalian muscle, the steady state ripple in the output is insignificant at an input activation frequency of about 60 [Hz]. In this situation, the output is a constant contractile force called *fused tetanus* and it yields a maximum output in a muscle like the quadriceps of about 20.0 [N/cm²].

These phenomena all relate to the isometric test conditions in which they are observed. Even so, to account for these observations, the molecular mechanisms of developed tension discussed in section 2.1.2 must be extended to include the passive viscoelastic character of the muscle tissue itself. These passive properties are even more influential when the system is in motion. Figure 2.5 shows how muscle dynamics influences overall performance [192]. It illustrates the capacity of “muscle in motion” to generate forces as a function of its position and velocity. Figure 2.5 (a) illustrates normalized muscle tension as a function of the length of the muscle relative to its free length l_0 . The diagram shows both flexed ($l/l_0 < 1$) and extended ($l/l_0 > 1$) configurations. The passive tension for flexed configurations is assumed to be zero. The diagram provides a qualitative illustration of what the total muscle tension—the sum of the developed tension and passive viscoelastic tension—looks like for some larger muscles as well.

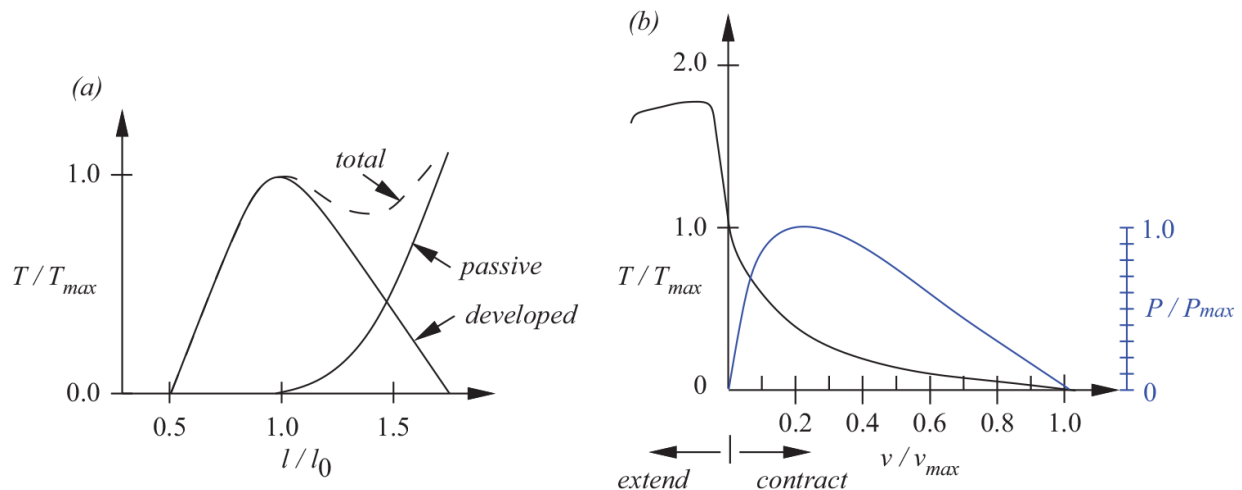


Figure 2.5

Qualitative force capacity of muscle as a function of length and velocity. Tension T is normalized by the maximum developed tension (T_{max}), power P by maximum power output (P_{max}), length l by muscle free length (l_0), and velocity v by the maximum unloaded contractile velocity (v_{max}). (Source: Adapted from [192])

The influence of muscle (and limb) velocity of muscle force is illustrated in [figure 2.5 \(b\)](#). It plots normalized tension against normalized velocity. For positive v/v_{max} , the net force generated decreases as velocity increases. For $v/v_{max} < 0$, the muscle is capable of greater tension to a point but quickly gives up and yields when the muscle begins to lengthen. The developed power (the product of developed tension and velocity) generated by the muscle is illustrated in the blue curve in [figure 2.5 \(b\)](#). The most striking feature of these plots is the nonlinear and asymmetric behavior of muscle as an actuator. It is unilaterally contractile, has limited range of motion, and behaves very differently when it is extended versus flexed and when it is contracting versus lengthening.

Despite the clearly nonlinear character of muscle, a linear model of its behavior can provide insight into some of the important aspects of bulk muscle dynamics. [Figure 2.6](#) presents such a model that is often cited by researchers [192]. Function $F(x, t)$ is used to model the active contractile behavior of muscle (the developed tension). [Figure 2.6](#) extends Huxley's elastic molecular account of the actin-myosin interaction with additional viscoelastic components to represent the passive properties of muscle tissue. For example, it has been observed that at a given stimulation level, muscle tension decreases with velocity. To model this phenomenon, a linear

dashpot (or damper)² is introduced that produces an opposing force proportional to velocity $F_B = -B\dot{x}$. This may also serve as a linear approximation of the pre-stiffening behavior prior to contraction. To model the passive elasticity of the muscle system, two linear springs are included. The spring K_{muscle} is arranged in parallel with the force source and the dashpot to represent the elasticity of the muscle tissue. Another spring K_{tendon} is arranged in series to model the elastic tendons that attach the muscle to the skeletal load. We will analyze systems composed of such linear elements in more detail in chapter 3.

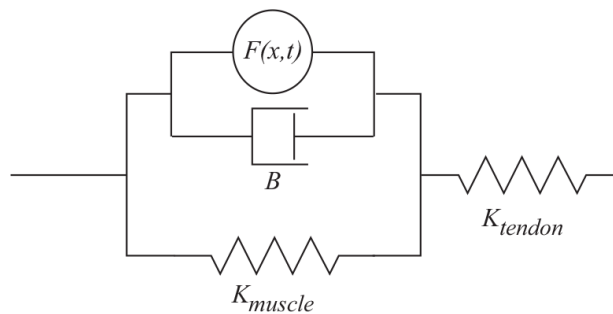


Figure 2.6
The Huxley model combining active and passive muscle dynamics

Muscle tissue is remarkable stuff, capable of storing a small amount of energy for direct use and making more from using both aerobic (fast) and anaerobic (slower) chemical pathways. The raw materials for these pathways ultimately come from glucose and fatty acids in the bloodstream and stored in fat. It can function efficiently even in oxygen deficit and for long periods of time before fatigue. It incorporates specialized cells and tunes bulk viscoelastic properties for each degree of freedom in the skeleton to deliver a variety of motor responses. Moreover, it grows and changes in response to use.

2.2 Robot Actuators

Although commonly used robotic actuators have little in common with muscle tissue, they too influence the active and passive behavior of the limb, and each choice presents different challenges with respect to power output, dynamic range, and packaging. Like muscle, actuators for robotic

applications take stored energy and transform it into mechanical work. There are a wide variety of technologies available, and the research community is actively developing new concepts. In this section, a survey of commonly used actuators is presented. The most common robot actuator is the DC electric motor, so this device will be covered in detail. Other technologies in common use, like hydraulic and pneumatic actuators, as well as more specialized and/or emerging technologies, will be introduced at the end of the chapter.

2.2.1 Permanent Magnet DC Electric Motors

Electric motors are used often in consumer products and produced in such volume that they are relatively inexpensive. These actuators employ electromagnetic fields to transform electrical current into forces or torques. The behavior of the permanent magnet DC motor is governed by the Lorentz force,

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}, \quad (2.2)$$

where q is the electrical charge of a particle moving at velocity \mathbf{v} through a conductor that is positioned in a magnetic field \mathbf{B} (figure 2.7). The “particle” is actually the net behavior of many electrons that produce a current in the conductor. By convention, the direction of positive current flow is the direction of flow of positive charges in the conductor—opposite to the direction that electrons flow—as in figure 2.8. In equation 2.2, the current $q\mathbf{v}$ and the magnetic field, \mathbf{B} , are vector quantities with magnitude and direction. The Lorentz force is the vector product (or cross product) of $q\mathbf{v}$ and \mathbf{B} . As a result, we find that the conductor is subject to a force that is mutually perpendicular to both the current and the magnetic field. In figure 2.7, this force is directed into the plane of the paper.

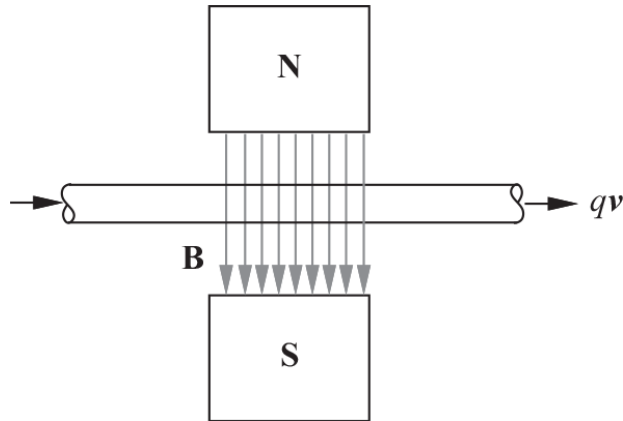


Figure 2.7

A charge q moving along a conductor at velocity v through a magnetic field, B , from north to south magnetic poles

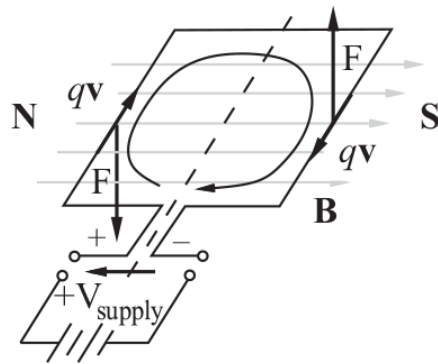


Figure 2.8

Moving electric charges create a torque on the coil due to the Lorentz force

When a current *loop* is fashioned out of the conductor and positioned in the magnetic field as in [figure 2.8](#), the result is an upward force on the right side of the loop and a downward force on the left—generating a net torque and causing the loop to accelerate as it rotates about the dotted line in the figure.

The torque on the rotor in [figure 2.8](#) cannot act continuously through an entire revolution. One can verify this fact by rotating the current loop 90 [deg] until the plane of the loop becomes perpendicular to the magnetic field. In this configuration, the vector product indicates that the Lorentz force is now directed radially away from the axis of rotation and, therefore, creates no more torque. In fact, this rotor-field geometry is a stable equilibrium. If the rotor continues past 90 [deg], the Lorentz force will

create a torque back toward the 90 [deg] rotor posture. To construct a continuously rotating DC motor, the current in the loop must be shut off as the coil approaches 90 [deg] and reversed after it passes this point—a process called *commutation*. The commutator can be implemented using a solid state switching circuit or mechanically using brushes and conducting pads. The collective behavior of many current loops, each commutated appropriately and separated by small angular offsets about the axis of rotation produces an approximately constant torque at a given speed even though each loop is switching polarity twice every rotation.

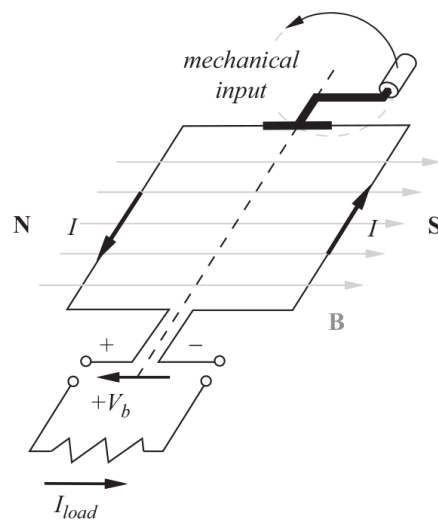


Figure 2.9

Faraday’s law describes how electrical current is generated in response to a mechanical input energy

If we remove the supply voltage (thus, opening the circuit) and rotate the commutated rotor by some other means (a wind/water-driven turbine or the hand crank in [figure 2.9](#)), then the mechanical rotation of the conductor loop in the magnetic field will generate a voltage on the open terminals of the rotor. This phenomenon is described by Faraday’s law,

$$\mathcal{E} = -\frac{d\phi_b}{dt} \quad (\text{or, } -N\frac{d\phi_b}{dt} \text{ for } N \text{ turns}),$$

which states that the electromotive force (\mathcal{E}) that pushes electrons around the circuit is proportional to the time rate of change of the magnetic flux $\phi_B = \int \int B \cdot dA$, surrounded by the loop in [figure 2.9](#). Informally, magnetic flux can be thought of as the number of magnetic field lines in [figure 2.9](#) that

pass through the interior plane of the current loop (zero in the geometry shown in the figure). Because the electromotive force is related to the time rate of change of the magnetic *flux*, the faster the crank is turned, the more potential is generated. This is the principle underlying a *generator*. If a load is connected to the terminals of the commutated rotor to close this circuit, then this electromotive potential creates a generated current in the connected load. Comparing figures 2.8 and 2.9, we find that if the hand crank rotates the generator in the same direction that the motor in figure 2.8 turned, then the current induced by the generator is in the opposite direction of the current supplied by the battery in the DC motor.

In addition to the angular velocity of the rotor $\dot{\theta}_r$, the generated potential V_b depends on the number of active motor windings and the magnetic field strength. Motor manufacturers publish a motor constant K_m to describe the net relationship between motor velocity and the generated potential,

$$V_b = K_m \dot{\theta}. \quad (2.3)$$

If the motor is perfectly efficient, then it can be shown that this K_m is the same K_m constant that relates motor current and torque $\tau_m = K_m I$. To see how, imagine that torque production depends on a different constant K_τ . Then by equating the mechanical power output ($\tau \dot{\theta}$) to the sum of electrical power in (VI) and resistive losses (I^2R) at steady state (i.e., $dI/dt=0$), we find

$$\begin{aligned} \tau \dot{\theta} &= VI - I^2R, \text{ so by expanding, we find} \\ (K_\tau I) \dot{\theta} &= (IR + K_m \dot{\theta})I - I^2R = K_m I \dot{\theta}, \text{ and} \\ \therefore K_\tau &= K_m. \end{aligned}$$

Therefore, without other losses like bearing friction, the motor constant performs double duty, turning velocity into backward electromotive force (back emf) and current into torque.³

Both effects are present simultaneously in the commutated DC motor. The rotating motor produces a back emf in each current loop—the net potential created by the moving rotor V_b throttles the net current in the rotor as a function of the rotational velocity. As the rotor speed increases, the sum of the supply voltage V_s and the generated potential $-V_b$ approaches zero. At this point, no current is flowing in the rotor and, therefore, no additional

torque is produced by the motor. If the rotor velocity increases further, then the back emf will decelerate the rotor until it once again balances the supply voltage and the back emf. We draw two important conclusions regarding the dynamics of DC motors:

1. the torque produced in the motor is proportional to the rotor current; and
2. the steady state velocity of the motor is determined by the applied voltage—determined by the velocity at which the back emf equals the supply voltage.

Figure 2.10 illustrates how modern motors package many commutated current loops into a functioning DC motor. The figure is a view down the axis of rotation, showing how insulated motor wire is wound in many loops within the magnetic field. In this example, the rotor is a piece of iron with radial fingers that serve to columnate the magnetic field and provide a channel for several motor windings. A loop is formed when a wire comes out of the plane of this figure through a pair of these fingers and returns down through the plane of the paper between another pair of rotor fingers 180 [deg] away from the first pair.

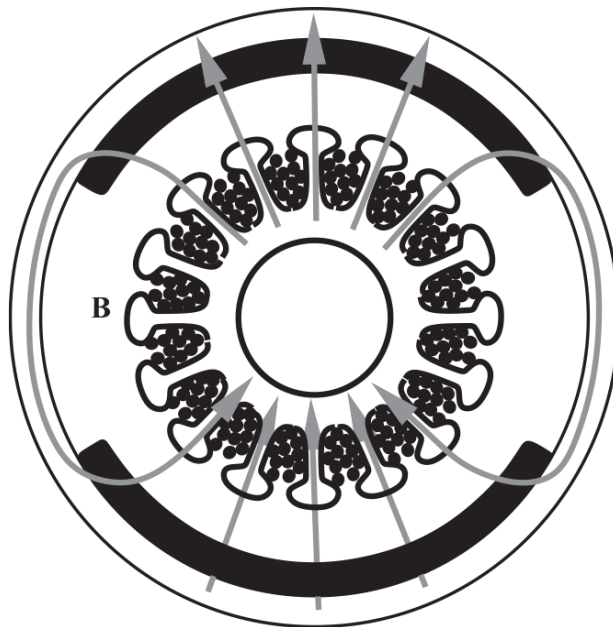


Figure 2.10
The iron core DC motor

The permanent magnet DC motor is very popular because it is reliable, it has a good power-to-weight ratio, and it can produce relatively large torques. However, the massive iron core rotor compromises motor performance. Moreover, it can lead to a measurable ripple in the magnetic field strength as the motor rotates. This is known as *cogging* and can be observed as a preference for rotor positions that maximize the amount of iron between the poles of the permanent magnet. One can feel the cogging effect by turning the rotor of an inexpensive motor without a gearhead—it makes it difficult to control the position of the rotor in these motors precisely.

Variations on this basic design can increase motor efficiency and performance. Surface wound motors employ more expensive rare earth magnets to produce stronger magnetic fields that can eliminate some or all of the iron in the rotor. As result, rotor inertia can be reduced significantly and cogging can be virtually eliminated. This approach is taken to the extreme in the *moving coil* DC motor, where the coil *is* the rotor. As a result, the rotor inertia is very low compared to the iron core rotor. Moving coil motors can be manufactured in a large range of geometries including the “printed circuit motor” with thicknesses down to 0.02 [in] and diameters up to 12 [in] and frameless motors that save weight by incorporating the motor directly into the structure of the actuated device.

Manufacturers measure DC motor performance and publish it in motor catalogs to make it easier to select the right motor for an application. The catalogs report electromechanical properties like electrical resistance R and inductance L in the motor windings, the moment of inertia J of the rotor, and the mass and geometry of the motor package. They also tabulate empirical parameters that characterize the overall performance of the integrated motor system in terms of the DC motor electro-dynamics.

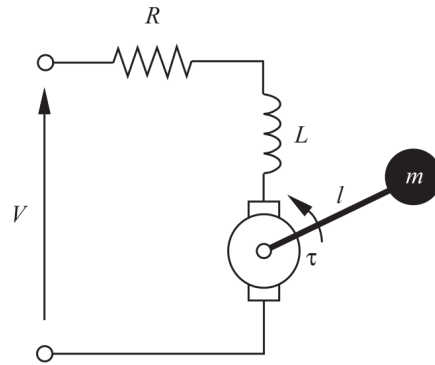


Figure 2.11

Electromechanical model of the DC motor

DC motor electrodynamics—Figure 2.11 illustrates a simple circuit model for the DC motor. The motor windings are modeled in terms of the electrical resistance R and inductance L of the rotor. Resistance R defines a linear relationship between supply voltage and the rotor current,

$$V = IR. \quad (2.4)$$

The inductance L measures how efficiently the geometry of the current flow generates a magnetic field. The strength of a magnetic field is measured in terms of magnetic flux $\phi = LI$. A change in the current through an inductor corresponds to a changing magnetic field and, thus, by Faraday's law, creates an induced voltage

$$V = \frac{d\phi}{dt} = \frac{dLI}{dt} = L \frac{dI}{dt}. \quad (2.5)$$

Equation 2.5 defines the current-voltage relation for an inductor. Inductance is measured in *Henries* [H], where—an inductance of 1 [H] produces 1 [V] when the current through the inductor changes at the rate of 1 [A/s].⁴ The third circuit element in figure 2.11 that must be considered is the net potential V_b generated by the motor velocity (equation 2.3).

Kirchhoff's voltage law states that the voltage V applied by the power source across the two motor terminals in figure 2.11 must be the same as the sum of the voltage changes across the resistor, the inductor, and the generator in series.

Performing this sum using results in equations 2.4, 2.5, and 2.3 yields

$$V = IR + L \frac{dI}{dt} + K_m \dot{\theta}. \quad (2.6)$$

Often, the inductance in the motor windings is insignificant (relative to the resistance in the windings) and can be neglected. Under these conditions, Kirchhoff's voltage law takes a simpler (approximate) form,

$$V \approx IR + K_m \dot{\theta}. \quad (2.7)$$

The equation describing the electrodynamics of the motor is derived by using Euler's equation to equate motor torque and acceleration:⁵

$$\begin{aligned} \sum \tau &= J\ddot{\theta} = K_m I \\ &= K_m \left[\frac{V - K_m \dot{\theta}}{R} \right], \end{aligned}$$

where, J [$kg \cdot m^2$] is the rotational moment of inertia of the rotor and we have substituted an expression for current I derived from [equation 2.7](#). Rearranging terms, we get

$$\ddot{\theta} + \frac{K_m^2}{JR} \dot{\theta} - \frac{K_m V}{JR} = 0. \quad (2.8)$$

[Equation 2.8](#) is a second-order differential equation of motion that approximates the motor's dynamics. Tools for solving equations like this for $\theta(t)$ will be introduced in chapter 3.

Performance—DC motor performance is quite different from that of muscle tissue ([figure 2.5](#)). Muscle performance is limited by the finite range of motion of the contractile elements, and forces depend directly on the length of the muscle relative to its free length and the stretch velocity. In contrast, DC motors can produce a continuous, bidirectional torque, and performance is independent of rotor position.

For a fixed supply voltage V_s and no external load, the steady state velocity of a motor is that speed where the back emf is equivalent to the supply voltage. This free-running condition rotates the motor at the *no-load speed* ω_0 , which can be measured in the laboratory. In this condition, there is a small measurable current, called the *no-load current* I_0 , required to

overcome the friction in the motor bearings and thermodynamic losses. Figure 2.12 illustrates these conditions using a family of relations called torque-speed curves that are commonly used to characterize the performance of the motor.

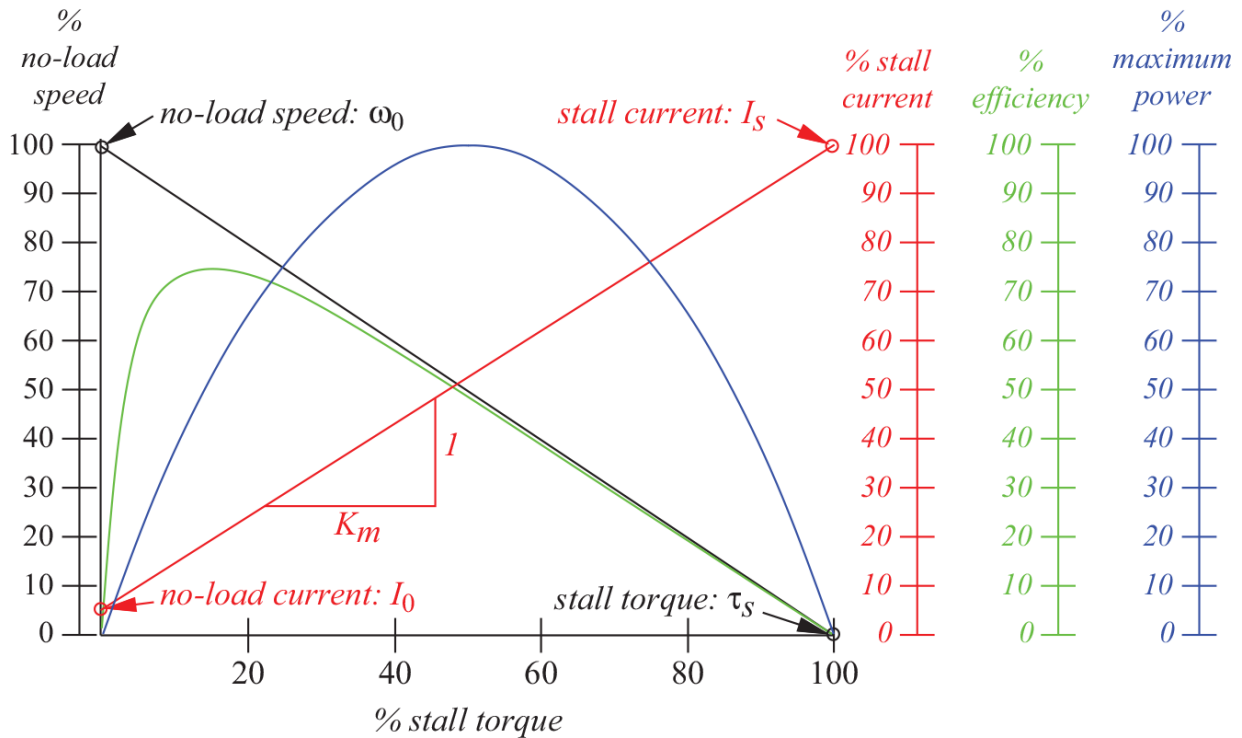


Figure 2.12

Plot colors and the corresponding vertical axes illustrate a family of performance curves relating motor speed, current, power, and efficiency to the torque generated by a DC motor

Other conditions within the spectrum of operating conditions can also be plotted on this family of curves. For instance, by holding the supply voltage V_s fixed, a resistance torque on the motor shaft can be increased in the laboratory until it is just sufficient to bring the motor to a stop.⁶ The load τ_s that causes this to happen is called the *stall torque* and it corresponds to a measured *stall current* I_s . These quantities can be found in figure 2.12 as well.

The line from the no-load ($\tau = 0, \omega = \omega_0$) coordinate in the upper left of the diagram to the stall torque condition ($\tau = \tau_s, \omega = 0$) in the lower right describes the motor's torque-speed relation. We know it is linear because of the linear relationship between back emf and velocity expressed in equation

2.3 and the slope of this relation is the negative ratio of no-load speed to stall torque $-\omega_0/\tau_s$.

Another linear torque-current relationship is constructed to estimate the motor constant K_m that describes how well the motor transforms current into torque $\tau = K_m I$. Figure 2.12 illustrates this relation by drawing a line from zero torque and no-load current in the lower left of the diagram to the stall torque and stall current in the upper right.

The power output of the motor is determined by computing the product of torque and speed for several loads between zero and the stall torque.

$$P_{out} = \tau_{load} \omega_{\tau} = \tau_{load} \left[\omega_0 - \frac{\Delta\omega}{\Delta\tau} \tau_{load} \right] = - \left(\frac{\Delta\omega}{\Delta\tau} \right) \tau_{load}^2 + (\omega_0) \tau_{load}, \quad (2.9)$$

which results in the parabolic power curve in figure 2.12. It indicates that maximum power production occurs at 1/2 the stall torque and 1/2 the no-load speed. The blue torque-power function in the figure indicates that maximum power production occurs at 1/2 the stall torque. Examining the corresponding black torque-speed curve at 1/2 τ_s , we see that maximum power production also occurs when the motor speed $\omega \approx 1/2 \omega_0$.

Motor efficiency follows directly as the ratio of output power P_{out} to the input electrical power VI , where V is the constant supply voltage and I is the motor current at load torque τ_{load} .

$$\eta_{\tau} = \frac{- \left(\frac{\Delta\omega}{\Delta\tau} \right) \tau_{load}^2 + (\omega_0) \tau_{load}}{V (I_0 + \tau_{load}/K_m)} \quad (2.10)$$

Example: Torque-Speed Calculation

The Faulhaber company provides a detailed specification sheet for its DC motors, one of which is summarized in the table. These four parameters are enough to construct the torque-speed curves for the motor.

Selected parameters from the Faulhaber 3257-024-CD
DC micromotor datasheet

No-load speed	$\omega_0 = 5900$ [rpm]
No-load current	$I_0 = 0.129$ [A]
Stall torque	$\tau_s = 539$ [mN · m]

From the no-load and stall parameters, the slope of the torque-speed function can be determined,

$$\frac{\Delta\omega}{\Delta\tau} = \frac{5900 [rpm]}{539 [mN \cdot m]} = 10.946 \frac{[rpm]}{[mN \cdot m]},$$

and an estimate for the stall current is derived:

$$I_s = I_0 + \frac{\tau_s}{K_m} = 0.129 + \frac{539}{37.7} = 14.426 [A].$$

This is a relatively high stall current. It is likely that the motor would not survive at stall condition for 24 [V] input, so stall must be avoided and/or the current that the motor draws under these conditions must be limited. This is reflected in the manufacturer's recommendation that this motor operate in the interval [0, 70] [mN · m] (far short of the stall torque).

Suppose that the motor is subjected to a 24 [V] input and a (midrange) load torque of 35 [mN · m]. In this case, motor current is significantly smaller,

$$I_\tau = I_0 + \frac{\tau_{load}}{K_m} = 0.129 + \frac{35}{37.7} = 1.06 [A],$$

and the electrical power input at this load can be computed:

$$P_{in} = V_{in}I_\tau = (24)(1.06) \approx 25.4 [W].$$

The manufacturer also recommends that the motor be operated at speeds in the interval [0, 5000] [rpm]. At the hypothetical 24 [V] input and 35 [mN · m] load, the predicted motor speed is

$$\omega_\tau = \omega_0 - \frac{\Delta\omega}{\Delta\tau} \tau_{load} = 5900 - (10.95)(35) \approx 5517 [rpm],$$

(greater than the manufacturer's recommended motor speed), and the mechanical power generated for this loading condition is

$$P_{out} = \tau_{load} \omega_{\tau} \approx 20.2 [W],$$

so that the overall efficiency at this load-speed combination is

$$\eta_{\tau} = \frac{P_{out}}{P_{in}} = \frac{20.2 [W]}{25.4 [W]} \approx 0.79.$$

If this analysis is repeated for several loads in the interval of recommended torques for this motor, the complete torque-speed curves for this motor (as in [figure 2.12](#)) can be generated.

□

Gearheads—Often, motors are connected through a transmission to external loads like the limb of a robot. Sequences of gears can be used to amplify motor torques and reduce output velocities for applications like this. The dynamics of the integrated system (motor, limb mass, and external loads) can be altered dramatically by the gear ratio chosen and is, thus, an important aspect of the overall design.

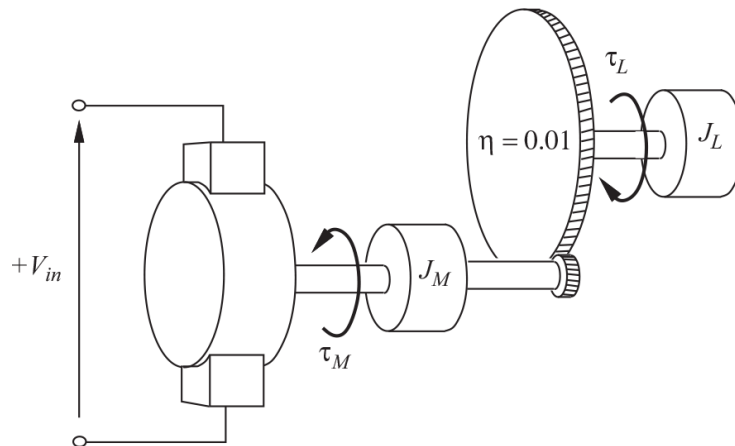


Figure 2.13

The compound load of the motor-gearhead combination

Consider the compound load in [figure 2.13](#). The motor produces a torque, $K_m I$, that accelerates the combined load mass moment of inertia of the motor J_M and the load J_L . [Figure 2.13](#) shows a pair of gears between the motor and the load that scales the angular displacement, velocity, and acceleration of the load relative to the motor.

$$\begin{aligned}\theta_L &= \eta\theta_M, \\ \dot{\theta}_L &= \eta\dot{\theta}_M, \text{ and} \\ \ddot{\theta}_L &= \eta\ddot{\theta}_M.\end{aligned}$$

Normally, the gear ratio, η , is less than 1.0—a 100:1 reduction has $\eta = 0.01$. If we consider the transmission to be perfectly efficient⁷ and linear, then the power output is equal to the power input, or

$$\begin{aligned}\tau_{out}\omega_{out} &= \tau_{in}\omega_{in} \\ \tau_{out}(\eta\omega_{in}) &= \tau_{in}\omega_{in} \\ \tau_{out} &= \tau_{in}\frac{1}{\eta}.\end{aligned}\tag{2.11}$$

Therefore, a gearbox reduction ($\eta < 1$) decreases velocity and amplifies output torque; if $\eta = 0.01$, then the output shaft carries 100 times the torque at 0.01 times the velocity of the input shaft.

The gearhead has no direct analog in biological muscle⁸—it changes the dynamics of the actuator in important ways that change its ability to sense forces. To see how, we follow a discussion by Craig [65] in which the dynamics of the compound load in [figure 2.13](#) is derived. Ignoring frictional losses, Newton’s second law for the compound load is written

$$\begin{aligned}\sum \tau &= \tau_M = J_M\ddot{\theta}_M + \eta J_L\ddot{\theta}_L, \text{ or since } \ddot{\theta}_L = \eta\ddot{\theta}_M \\ \tau_M &= [J_M + \eta^2 J_L]\ddot{\theta}_M.\end{aligned}$$

Therefore, from the perspective of the motor, the net inertia of the compound load is $J_{net} = J_M + \eta^2 J_L$. For a 100:1 reduction ($\eta = 0.01$), the load inertia is attenuated by a factor of 10, 000 relative to the motor inertia. Conversely, from the perspective of the external load, the motor behaves like a massive flywheel. Such a drive configuration will look very stiff from the outside and, as a consequence, DC motors with large gear reductions are generally not easily *backdrivable*—such an actuator is passively stiff and insensitive to external loads from the limb or from contacts between the limb and the environment. For robots designed for precision free-space movements (as is often the case in manufacturing operations) large

reductions are appropriate. However, when unexpected contacts may occur or energetic disturbances must be dissipated, backdrivable drive trains can comply and avoid damage in the robot and in the environment.

2.2.2 Hydraulic Actuators

Hydraulic actuators transmit power via hydraulic fluids. A hydraulic fluid is nearly incompressible and transmits power at the sonic velocity of the fluid. An example of such a system is the brake in an automobile. In older cars, the driver applies a force to the brake pedal to actuate a master cylinder and create an elevated pressure in the hydraulic brake line. The pressure wave propagates very quickly via the hydraulic fluid to the brake cylinder located near the wheel, where the pressure in the brake line is transformed back into the linear displacement of another piston that actuates the brake mechanism.

The ratio of the cross-sectional area of the master cylinder A_1 to that of the brake cylinder A_2 defines the gain in the transformation. If s_1 is the displacement of the master cylinder with area A_1 and if s_2 is the displacement of the brake cylinder with area A_2 , then the reduction is

$$\eta = \frac{A_1}{A_2} = \frac{s_2}{s_1}, \text{ and the force applied to the brake is amplified } f_2 = \frac{1}{\eta} f_1$$

relative to the force applied to the brake pedal. A displacement of a couple of centimeters at the brake pedal can correspond to a displacement on the order of millimeters in the brake caliper. This implies that $\eta < 1$ and the mechanical advantage created can apply much greater clamping force on the brake caliper than is applied at the brake pedal. Just as in DC motors with gearheads, a significant mechanical advantage means that it is difficult to backdrive the system. In the automobile braking system, and in other applications discussed in this section, this is a virtue—to backdrive the brake peddle, we would have to apply huge forces to the brake caliper.

Power brakes use the car's engine to turn a compressor and create a high pressure hydraulic source. Displacements of the brake pedal actuate a hydraulic servo valve and control how much hydraulic pressure is applied to the brake line. As result, much lighter pressure on the pedal is required to create appropriate braking forces. Power steering systems operate in much the same way using servo valves like that illustrated in [figure 2.14](#). This

bidirectional configuration uses high pressure power assist to actuate the steering gear in response to user inputs (left/right) on the steering wheel. The gain in power steering systems can be inversely proportional to vehicle velocity to provide more help at low speeds and to improve road feel (backdrivability) at higher speeds. However, when high pressure assist is used to drive small forces, the servo valve must eliminate a lot of energy as heat and, therefore, compromises efficiency.

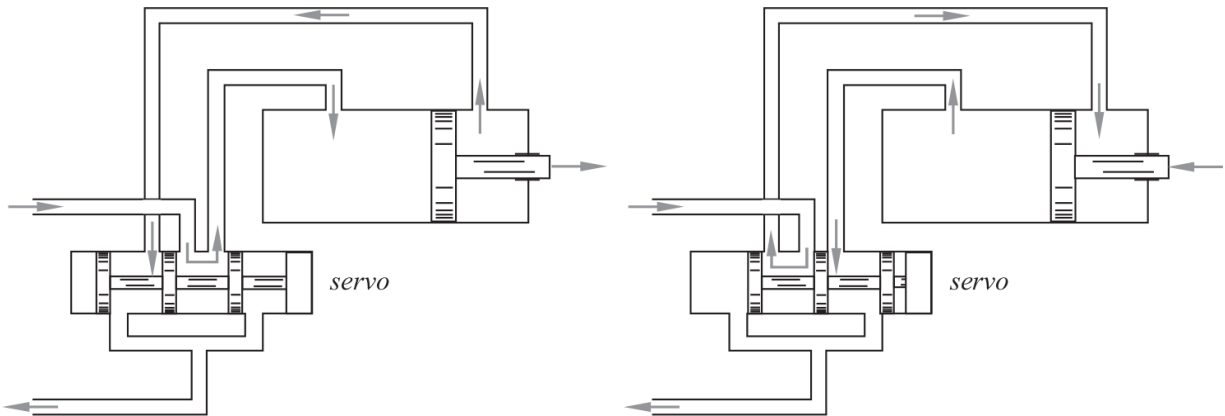


Figure 2.14

The hydraulic servo valve directs high pressure hydraulic fluid from a reservoir to drive a piston: (*left*) flow for extending the linear actuator; (*right*) flow for retracting the actuator.

Still, when a high pressure hydraulic source is controlled by servo valves and when there exists an appropriate mechanical advantage, hydraulic systems have excellent power-to-weight ratios. The common backhoe, for example, uses high pressure hydraulic actuators to great advantage. These machines are powerful and relatively fast and are easily packaged in articulated arm structures.

The Sarcos GRLA (for General Large Robot Arm) in [figure 2.15](#) runs hydraulic actuators at up to 3000 [*psi*] to drive a robot arm that is 1.75 [*m*] long from shoulder to wrist. The robot can be driven by an exoskeletal master worn by a human teleoperator. As in the backhoe, the geometrical scale and the work output of the user are amplified dramatically by managing the high pressure hydraulic fluid flow rather than relying exclusively on human muscle power.

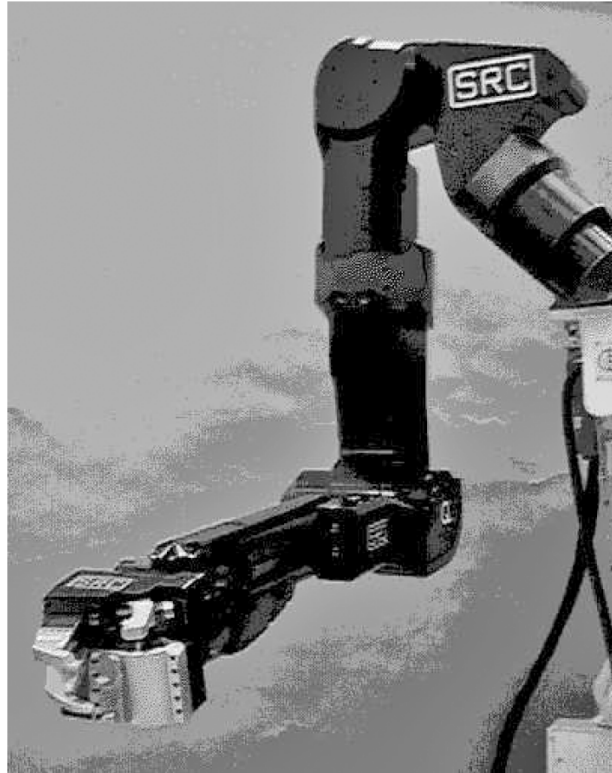


Figure 2.15

The Sarcos GRLA (General Large Robot Arm)

Hydraulic actuators can be quite useful in environments where electric motors may not be recommended, like explosive or wet environments. But they can be messy, and the servo valves can be quite expensive and difficult to maintain. Hydraulic fluid transmits relatively high frequencies (up to about 5 [KHz]), and servo valves can reverse the direction of flow in as little as 5 [msec]. This bandwidth makes it possible to implement active controllers on top of the passively stiff actuator. For example, in the power assisted braking scenario discussed earlier, a small embedded controller can enhance the pedal input by superimposing a high frequency signal to keep the wheel from skidding, as in ABS braking systems.

Nature has devised ways for animals to use hydraulic actuators as well. Some spiders, for example, cannot extend their legs by activating muscles alone—they have no extensor musculature that is adequate for the task. To solve this problem, these spiders use their blood as hydraulic fluid. Consequently, blood pressure in spiders can be very high compared to other animals. Special valves and muscles compress their forebodies and act as

actuators for their legs. Some jumping spiders create huge leaps (of up to 10 times their body size) using hydraulic forces created in a specialized fourth pair of legs.

2.2.3 Pneumatic Actuators

Pneumatic actuators use compressed air as the working fluid. Pressurized air from a reservoir or a compressor provides a relatively large energy density that can be routed to the place where mechanical work is required. Air is inherently elastic at low velocities, and as a consequence, these actuators are passively backdrivable and naturally compliant. Two principal configurations of pneumatic actuators are in use today: piston-cylinder arrangements and air muscles.

Pneumatic cylinders—Mechanical work can be derived from compressed air by using it to drive a piston. A pneumatic valve is required to control how air is introduced into the drive piston and, thus, controls the tension in attached tendons. Valves can be simple and support “bang-bang” control, or they can be relatively sophisticated (and more expensive) and, thus, support continuously controllable air flow. One such continuous pneumatic control valve (figure 2.16) was used in the Utah/MIT dexterous hand in figure 2.17 [131, 129, 133, 132]. The actuators in the hand combine glass cylinders and graphite pistons to create a fast, low-friction, and passively compliant actuator.

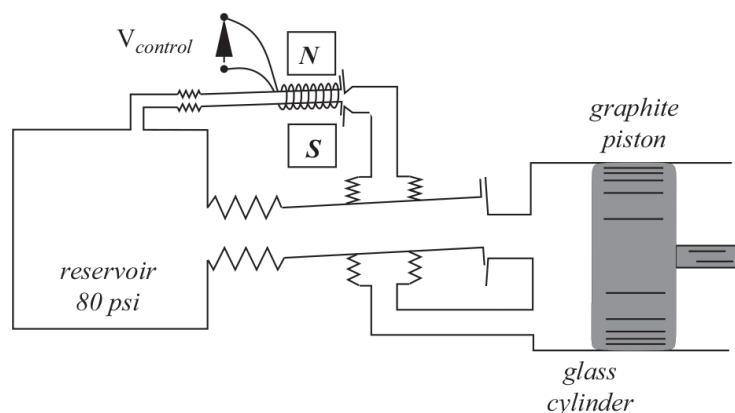


Figure 2.16
The jet-pipe servo valve for controlling pneumatic actuators

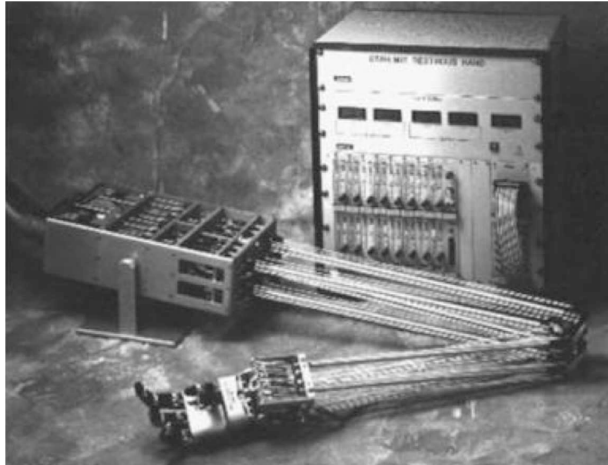


Figure 2.17

The Utah/MIT dextrous hand with polymeric tendons and analog control box

Intricate tendon routing geometry in the Utah/MIT hand supported fully antagonistic actuation for 16 degrees of freedom using a remote actuator pack containing 32 actuators driving pairs of tendons. A pair of jet-pipe servo valves was used to actively control the effective compliance of the fingers—increasing or decreasing it relative to the passive compliance of the air. The result was a force source with very little mass and damping.

The servo valve in [figure 2.16](#) combines a relatively large power capacity with speed by controlling a large air flow using a small, light jet-pipe. An input voltage deflects the jet-pipe within a magnetic field to adjust the differential pressure in a bellows that deflects to regulate the pressure in the piston. The result is a high performance, continuous force source. These actuators are still relatively cheap and light, with a power-to-weight ratio in the range of 16:1 and a relatively high bandwidth (40 [Hz]). They can, however, have issues with stiction, they may need additional damping for stability, and they can be relatively delicate and sensitive to shock. The Utah/MIT hand also employed high-speed analog controllers at the joint level to stabilize the high-performance actuators.

McKibben air muscles—One of the oldest theories of muscular action, dating to the third century BC, holds that animal spirits, or *pneuma*, flow down nerves to fill muscles and cause contraction [192]. Although this intuition has proven to be without merit, contemporary pneumatic actuators use principles like this to great effect.

Air muscles are popular for their strength at the expense of the speed of pneumatic cylinders. These actuators use the energy stored in compressed air just as the piston-cylinder arrangement does. Air muscles consist of a cylindrical rubber bladder surrounded by a tough plastic netting as illustrated in [figure 2.18](#). When the bladder expands radially, the plastic mesh “scissors” and shortens in length. As a result, the transformation from pressure to force is nonlinear. The greatest forces are possible when the muscle is fully elongated and force capacity decreases as the muscle contracts.

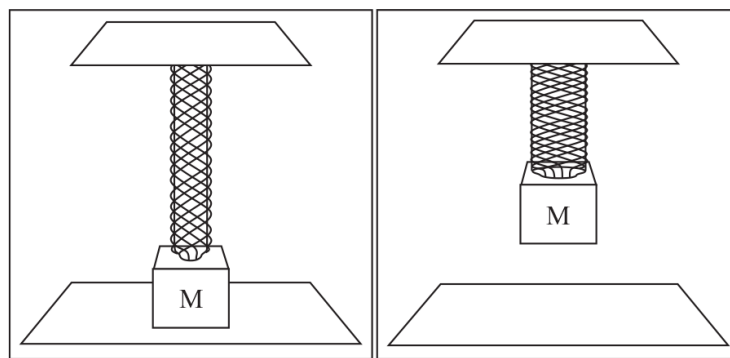


Figure 2.18

The operation of an air muscle

Simple construction leads to very lightweight actuators that are robust with respect to misalignment, passively compliant, and backdrivable, with a maximum power-to-weight ratio on the order of 100:1. The range of motion in the actuator is proportional to the length of the bladder, leading to a contraction of up to 40 percent of the original length and strength proportional to the bladder diameter.

The Shadow Robot Company manufactures a 20 degree of freedom hand that can be mounted on a 2 degree of freedom wrist and 4 degree of freedom arm entirely actuated by this type of air muscle in a variety of different sizes ([figure 2.19](#)). The joints are driven antagonistically by opposing pairs of muscles. Air muscles lead to flexibility in packaging and an overall size comparable to human hands and a total weight of between 5 and 10 [kg]. The hand uses compressed air at 60 [psi] for the main power supply and about 64 [W] of electrical power to run a communication bus and the servo valves. The family of air muscles produce between 0.5 [Nm]

and 1.5 [Nm] that yield a control bandwidth of approximately 3 [Hz], which is comparable to human performance.

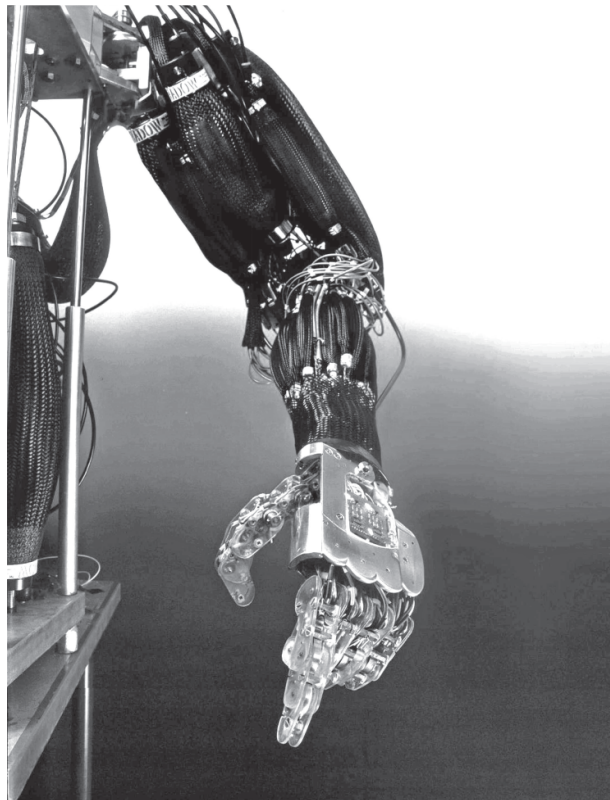


Figure 2.19

The 2001-era “Hand B” Integrated Shadow Hand-Arm System (www.shadow.org.uk/)

2.2.4 Emerging Actuator Technologies

Research in innovative new actuator technologies is very active. As a result, there are a number of promising new technologies that still await breakthroughs related to strength and bandwidth, mechanical properties (elastic modulus, tensile strength, fatigue life, thermal and electrical conductivity), thermodynamic issues (efficiency, power and force densities, power limits), power requirements, device fabrication, integration and packaging, and control.

Shape memory alloys (muscle wire)—Nickel titanium (also known as nitinol) is in a unique class of materials known as shape memory alloys. A crystallographic phase transformation in the material is responsible for its shape memory effect. When heated by an electric current, the alloy

undergoes a transition from martensite to austenite and the resulting strain in the new crystal structure causes the filament to deform into a preset shape. Thus, nitinol is a candidate for transformations from thermal energy to mechanical work.

Flexinol© is a trade name for small diameter shape memory alloy actuator wires. Individual wires contract by 5%–7% of their length when heated electrically and can then be easily stretched out again as they cool back to room temperature. The contraction is opposite to the ordinary thermal expansion and is 100 times larger. Even though the wires are small diameter, they can exert comparatively high forces and are well suited to applications like smart endoscopes that can be steered by the surgeon and stents for use in structural support for veins and arteries that are inserted and positioned in one shape and then left behind in another. Nitinol wire can take a lot of current to heat electrically and responds rather quickly, although its heating-cooling cycle time is relatively slow (on the order of 1 [Hz]).

Polymeric actuators—The mechanical properties of polymers can be sensitive to thermal, electrical, and chemical stimulation, which makes these materials candidates for new kinds of actuators. For example, polymer gels can exhibit abrupt volume changes (up to 1000-fold) in response to temperature, pH, and electric fields. These changes are relatively large and reversible. A number of devices have already been constructed that demonstrate the conversion of chemical or electrical energy into mechanical work in this manner with forces up to 100 [N/cm^2] and contraction rates on the order of a second [270].

Chemical gels consist of a polymer lattice with an interstitial fluid—sharing properties of both liquid and solid. There are many examples of natural gels (i.e., Jell-O gelatin, the vitreous humor of the eye (section 7.1.2), and artificial gels like polystyrene). The shape and dynamics of a gel are defined by the interaction between the polymer and the fluid. Chemically activated gels are compliant elements that undergo reversible length changes in response to chemical stimuli. Gel dynamics are limited by the diffusion of molecules in the fluid through the polymer lattice. Consequently, the larger the distance that a molecule must migrate, the longer it will take for the material to cycle through full range. Therefore,

polyacrylamide gel fibers with a diameter of 25×10^{-6} [m] contract on the order of a second, while 0.01 [m] fibers can take 2.5 [days] to contract [48]. In addition, some gels can support substantial loads. Polyacrylonitrile-polypyrrole (PAN-PPY) and polyvinyl alcohol (PVA) gel fibers generate up to 100 [N/cm²] [48], approximately equal to that of a human muscle.

In one published study, researchers explored using an antagonistic arrangement of PVA muscles to actuate a parallel jaw gripper; an apparatus is presented that innervates the artificial muscle by controlling the concentration of an acetone bath [40]. However, in general, these devices are still very slow and weak compared to other actuators. Recent research focuses on thin films and groups of small fibers that lead to faster contraction rates and larger forces. A number of fundamental issues remain to be addressed. Foremost among these is the design of an actuator package that can bathe the gel in a succession of chemical solutions (acid and base, for example) in a manner that is feasible given the robot mechanism.

Electroactive polymers are plastics that expand or contract in the presence of an electric field. These materials store electrons inside large molecules in the polymer and allow charge to migrate between molecules. As a result, electroactive polymers have a significant capacitance and can be used as batteries. Moreover, as charge is stored, the length of chemical bonds in the polymer is changed, which leads to its potential as an artificial muscle. However, electroactive polymers require a considerable amount of voltage to push electrons into the polymer and thus change shape. Most electroactive polymers capable of large shape changes deform to a degree proportional to the square of voltage. Movements on the order of 10 percent of the original length are attainable.

The dielectric constant is a relative measure of a material's ability to store electric charge. The greater the dielectric constant, the larger the capacity to store charge. Practical electroactive polymers like those used in batteries have relative dielectric constants of around 5. However, new composite electroactive polymers exhibit dielectric constants as high as 1000 [305] without sacrificing the elasticity of the polymer substrate. These new hybrids may reduce the voltage necessary to induce a movement, which would make these actuators more friendly to other electronic devices.

In one example, researchers made an actuator by using two polypyrrole films to sandwich an insulating plastic tape [218]. When the sandwich is polarized by applying a positive voltage to one side and a negative voltage to the other, the composite bends. Polypyrrole is interesting as an actuator for dexterous robots since its electrical conductivity changes proportionately with pressure. Since these sandwiches draw a small amount of current, it is possible that such devices may serve as tactile sensors as well.

The main problem with the polymer technologies is their relatively slow response (on the order of seconds). An electrorheological fluid (ERF) is a material that changes from a liquid to a solid upon application of an electric field. These fluids have a response on the order of a millisecond. A novel approach to artificial muscle utilizes the fast time response of the ERF and the elasticity of polymeric gels to create a faster artificial muscle [34]. The polymer gel with ERF stiffens to a solid within 100 [msec] after the application of an electric field. When subjected to a strong electric field (3000 [V/cm]), the material flexes. Although the actuator response is approaching the speed of muscle, its strength is limited (~ 0.001 [N]) and the voltages required are very high.

Synthetic muscle—Artificial muscle has been fabricated from naturally occurring long chain proteins. In muscle tissue, actin and myosin proteins interact to form a kind of biochemical ratchet (section 2.1). Energy-rich ATP molecules power the attachment, bending, and straightening of the myosin molecules.

Researchers have extracted actin and myosin protein from scallops and used chemical reactions to link the molecules together into polymeric gels. When a microscopic piece of actin gel is placed against the myosin gel and immersed in an ATP solution, the actin gel springs into motion at about one thousandth of a millimeter per second [135]. Obviously, this approach has a long way to go before it leads to practical actuators, if it ever does at all, but the technology may hold promise for implantable assistive technology for human muscle since the body's immune system might accept implants made from human muscle proteins.

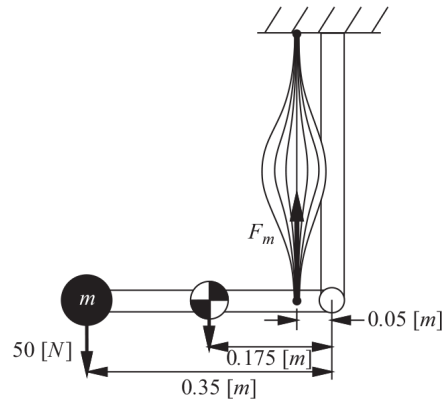
Buckytubes—Fullerenes (buckyballs) and nanotubes (buckytubes) are specialized crystalline configurations of graphitic carbon. Carbon nanotubes

are very thin and long tubes. Their diameter is only a few nanometers, about the diameter of typical molecules, and they can be up to millimeters in length. Carbon nanotubes increase their length when electrons are pushed into the carbon structure. This effect can be used for electromechanical actuators. The relatively large length change and the high elastic modulus lead to very large forces for carbon nanotube actuators.

A laminar composite consisting of a film of parallel buckytubes bonded to each side of a tape substrate that is rigid in tension but can bend was used to demonstrate the use of carbon nanotubes as an actuator [17]. The polarization of the sandwich can be relatively small—on the order of ± 1 [V]. The buckytubes elongate when positive voltages are applied and shorten when negative voltages are applied, flexing the tape substrate between them. Carbon nanotubes can, in principle, be used to create macro-, micro-, and nano-scale actuators that are extremely robust to harsh thermal and chemical conditions. Moreover, these new actuators can potentially achieve a greater mechanical stress than any other technology—much greater than muscle tissue.

2.3 Exercises

1. **Static force capacity of muscle.** The impressive performance of human muscle is demonstrated by estimating the muscular force required for a familiar motor task using the human morphology. A planar mechanism is illustrated that simulates the human elbow joint, biceps muscle, and a forearm that is 0.35 [m] in length. The biceps (possibly a McKibben air muscle) is attached on one end to ground and on the other end to the forearm 0.05 [m] from the elbow. Assume that the total forearm weight is 17 [N] and acts at the forearm's center of mass located 0.175 [m] from the elbow.



The robot arm is stationary in the posture shown, where it supports a 50 [N] load attached to the endpoint of the manipulator.

How much force F [N] must the biceps apply at the forearm insertion point to hold the arm and load in the posture illustrated?

2. DC motor physics.

(a) What electromagnetic phenomenon is responsible for transforming electrical current into torque in the permanent magnet DC motor?

(b) What physical phenomenon accounts for the backward electromotive force (back emf)? How does it contribute to the behavior of the DC motor?

(c) What role does *commutation* play in a DC motor?

(d) What is *cogging*?

3. **DC motor: Current-speed relation.** A DC motor with an armature resistance $R = 20$ [Ω] is connected to a supply voltage $V_s = 20$ [V]. At full speed the voltage generated by the back emf is $V_b = 18$ [V].

(a) Calculate the motor current at the instant that the motor starts from rest ($\dot{\theta} = 0$).

(b) Calculate the current when the motor is at full speed.

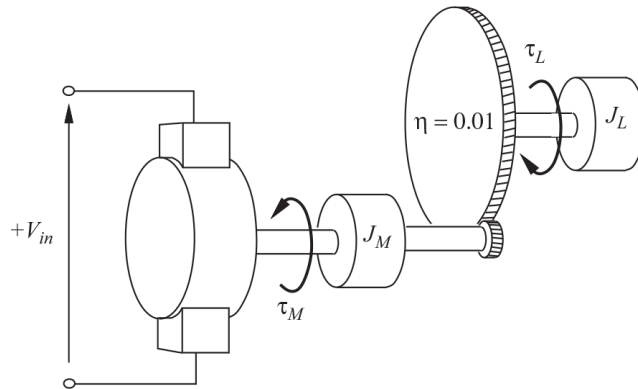
4. **DC motor: Steady-state velocity.** Because of its state of motion, a motor has a voltage derived from the back emf of $V_b = 14$ [V] at 2100 [rpm]. What is V_b at 3500 [rpm]?

5. **DC motor: Current/power.** A 24 [V] DC motor has an armature resistance of $R = 10$ [Ω] and draws 2 [A] at 300 [rpm].

(a) Calculate the voltage due to back emf V_b at 300 [rpm].

- (b) What is V_b at 1200 [rpm]?
- (c) What current will the motor draw at 1200 [rpm]?
- (d) Compute the power dissipated by the motor at 1200 [rpm].

6. **DC motor: Gearhead.** A motor-gearhead combination is shown in the figure below.



$$V_b = K_b \dot{\theta} \text{ [V]}$$

$$\tau = K_t I \text{ [Nm]}$$

$$K_t = 0.02 \text{ [Nm/A]}$$

$$K_b = 0.02 \text{ [Vsec/rad]}$$

$$R = 10 \text{ [\Omega]}$$

$$V = IR$$

- (a) What is the steady-state angular velocity of the *load* for 10 [V] in?
- (b) When the rotor is locked (i.e., $\dot{\theta} = 0$) and 10 [V] is applied, how much torque is generated on the *load*?
- (c) If $J_M = 0.005 \text{ [kg} \cdot \text{m}^2]$ and $J_L = 1.0 \text{ [kg} \cdot \text{m}^2]$, which is most significant in the output inertia?

7. **DC motor: Torque-speed curves.** The motor/gearbox parameters for Roger's wheel motors, shoulder motors, elbow motors, and eye motors are recorded in the table. These are really powerful motors, but not extremely fast (you may assume that they have a gearbox included that is built into the motor constants). All motors run on 24 [V] supply voltage.

Parameter	Wheel	Shoulder	Elbow	Eye
$\omega_0 \text{ [rad/sec]}$ no-load speed	175.0	30.1	50.3	122.2
$I_0 \text{ [A]}$ no-load current	0.38	0.26	0.17	0.12
$\tau_s \text{ [N} \cdot \text{m]}$ stall torque	475.0	205.3	120.7	2.7
$K_m \left[\frac{\text{N} \cdot \text{m}}{\text{A}} \right] \left[\frac{\text{V} \cdot \text{sec}}{\text{rad}} \right]$ torque constant	0.105	0.623	0.364	0.163

Assume that the motor/gearbox is 100 percent efficient, and plot the speed, current, efficiency, and power curves for the eye motor. Create separate

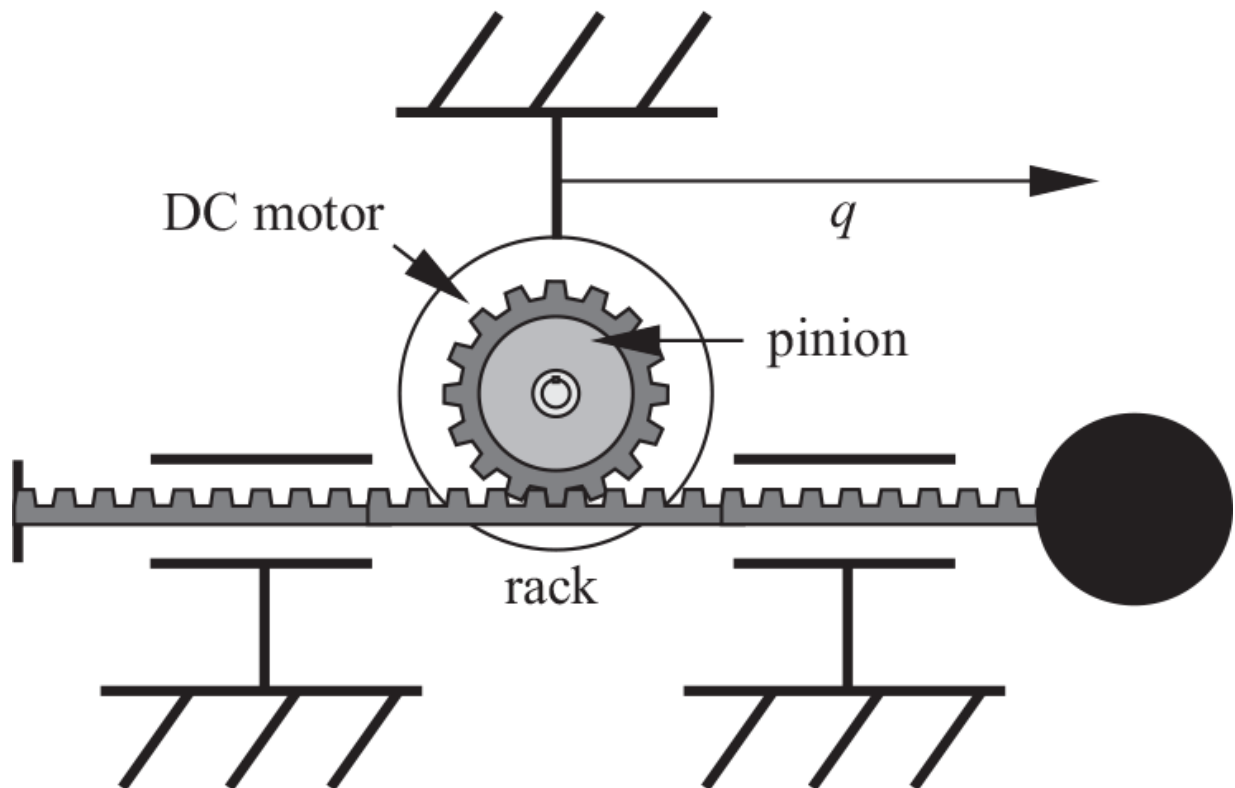
plots and use engineering units.

8. **Prismatic actuators.** One way to implement a prismatic actuator combines a DC motor and the rack-and-pinion gear system in this schematic that transforms motor torque and angular velocity into linear force and translational velocity respectively, for the prismatic joint.

$$\dot{x} = \omega r \left[\frac{m}{sec} \right] \quad f = \tau / r [N]$$

where $r = 0.01 [m]$ is the radius of the pinion gear.

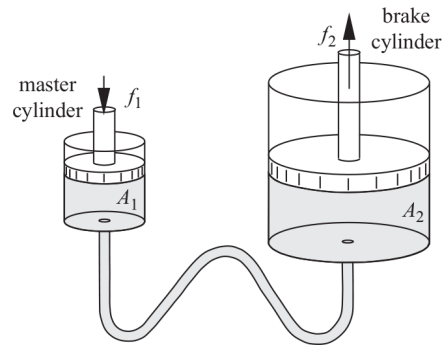
Suppose that the DC motor specifies a $24 [V]$ input, has a motor torque constant $K_m = 0.105 [Nm/A]$, rotor resistance $R = 1.7 [\Omega]$.



- (a) Compute the no-load speed of the motor and the upper bound on maximum velocity \dot{x} of the linear actuator.
- (b) Compute the locked-rotor torque and use this to find the upper bound on the force output f of the linear actuator.

9. **Hydraulic transmission.** A red 1965 Volkswagen Beetle weighs only 1609 [lb], which explains why the VW engineers didn't think it required

power assisted brakes.



For the hydraulic transmission between a brake peddle (the master cylinder) and a brake cylinder in this schematic, suppose that it requires 600 [lb] of force on the drum brake to cause the wheel to lock up.

The brake cylinder diameter is 1.5 [in] and the master cylinder diameter is 0.5 [in].

- (a) How much brake peddle force is necessary to skid to a stop?
- (b) It is important that air bubbles in a hydraulic brake line be eliminated. Why is this? How will the brake system behave if air is in the hydraulic line?

10. Invent your own homework. Make a homework problem out of your favorite content in chapter 2. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here, with extra credit to those that draw a comparison between muscles and motors. The question should call for a short discussion, analysis, and calculation—it should not take more than 30 [min] to solve open-book.

1. The symbol Å stands for Angstrom, which is equivalent to 10^{-10} [m].
2. In reality, muscle damping is nonlinear and depends on position as well as velocity. It was originally thought to be due to the viscous flow of water within the muscle fiber itself, but later this was found to be insufficient to account for experimental observations.

3. DC motors are, in fact, typically only 75%–90% efficient—brushless motors with solid state switching for commutation are normally higher efficiency, and brushed motors usually fall at the lower end of the range.
4. Joseph Henry (1797–1878) was an American who discovered electromagnetic induction at about the same time as Michael Faraday (1791–1867). The standardized unit of inductance [H] was chosen in recognition of Henry.
5. We derive Euler’s equation in section 6.2.
6. Not all motors will survive this procedure; the motor catalogs should be checked for limits on continuous current.
7. Typically, efficiency (power out / power in) for planetary gearboxes is between about 60% and 90%. It decreases sharply for very low loads. A useful rule of thumb is that transmission losses are approximately 10% per engaged gear pair. Overall motor system efficiency is the product of the motor and gearbox efficiencies.
8. One could argue that the human patella (kneecap) significantly increases the leverage applied by the quadriceps to the femur (lower leg bone) through the patellar tendon. This is a torque advantage implemented in the tendon routing between muscle and skeleton.

3

Closed-Loop Control

When an organism (or a robot) interacts purposively with the environment around it, it coordinates patterns of actuation over multiple limbs using control inputs. Control inputs are used to modify the passive behavior of the embodied system. An *open-loop* controller can produce an extended pattern of motor inputs in response to a single trigger event. In this case, the control logic tacitly encodes an effective response to the trigger. For example, human beings blink in response to loud noises and when nearby objects loom threateningly on the retina (section 9.3.1), and specialized peripheral receptors on the tip of the finger can cause a response in spinal neurons that *withdraw* the arm and hand from painful stimuli (section 3.1). Open-loop responses are, therefore, often described as *feed forward* control laws.

Closed-loop controllers use continuous feedback to measure progress toward reference states and modify motor inputs to eliminate errors. Under some conditions, closed-loop systems can guarantee asymptotic properties of controlled processes and, therefore, serve as an attractive model for biological motor control. Moreover, deciding which measurable errors to eliminate over time provides the basis for models of motor learning in animals and robots.

Both open- and closed-loop controllers can be implemented in the lowest levels of the neural control circuitry—generating responses before the subject is conscious of the triggering stimuli. Closed-loop actions, however, can also reside at higher levels in the central nervous system and can recruit multimodal feedback and many mechanical degrees of freedom to, for example, navigate an automobile through a four way stop using a full array

of visual, auditory, tactile, proprioceptive, GPS, and inertial signals along with multimodal memories of similar circumstances.

In this chapter, we begin by discussing how biological circuits consisting of muscle spindles, tendon organs, and α - γ motor neurons manage groups of muscle fibers using a closed-loop construction called a motor unit. This simple computational element relies on *negative feedback*—a fundamental principle of stable closed-loop control. The spring-mass-damper (SMD) system is examined in some detail as a simple mathematical model of the biological motor unit, wherein errors in the position of a limb—the mass—are attenuated over time by the combination of forces generated by an elastic element—the spring—and a viscous element—the damper. We will use an energy-based analysis to demonstrate that the position error in a SMD goes to zero over time, a useful property of the biological motor unit. The SMD is also the basis for a proportional and derivative (PD) feedback control law that can be easily implemented for use in robot controllers.

The chapter closes by deriving the differential equation of motion for the SMD and solving for its response as a function of time. To get a sense of the performance of PD controllers in robot systems, the sensitivity of the SMD response to control parameters and to the frequency of the reference input is evaluated.¹

3.1 The Closed-Loop Spinal Stretch Reflex

The lowest level in the human motor control hierarchy is also the most ancient (section 9.1). It resides at the interface between the central nervous system (CNS) and the peripheral nervous system (PNS), and it is responsible for the stable control of the distal musculature. This is the foundation for a full spectrum of stimulus-response motor behavior.

3.1.1 Spinal Processing

The spinal cord is part of the central nervous system and consists of five regions, each of which is made up of one or more vertebral segments. From top to bottom they are the cervical (C1–C8), thoracic (T1–T12), lumbar (L1–L5), sacral (S1–S5), and coccygeal segments, for a total of 31 spinal segments. Thirty-one sets of spinal nerves emerge from the spinal cord—

one from each segment. Neural projections in the cord form a bidirectional pathway connecting the rest of the central nervous system with the peripheral nervous system and the musculoskeletal system.

The cord itself is about 1 [cm] in diameter and is protected inside the bony vertebral column. However, the cross section of the cord varies along its length. The vast number of mechanoreceptors (section 7.2.1) and muscles in the hand and forearm (chapter 5), for example, project to the largest cross section at the cervical eight vertebra. Two types of neural tissue reside in the spinal cord: gray matter, containing spinal sensory and motor cells, and white matter, containing ascending and descending tracts. White matter gets its name from the appearance of the neuroglial cells that support and insulate these pathways. Three regions (or *fasciculi*) of white matter can be identified (dorsal, ventral, and lateral). The dorsal fasciculus is primarily ascending sensory fibers, while ventral and lateral fasciculi contain both sensory, and motor fibers. Ascending tracts in the cord communicate sensory information such as discriminative touch, proprioception, pain, and temperature for different regions of the body's surface to the brain. Descending tracts conduct information from the brain back out to the periphery to innervate muscles, organs, and glands.

Figure 3.1 shows five segments of the spinal cord (C5 through T1) that project to sensors and muscles in the arm. Each set of spinal nerves consists of dorsal and ventral roots that correspond to sensory and motor fibers, respectively. Sensory projections gather peripheral sensory signals, and motor fibers project activation signals back to the muscles. Each dorsal root receives sensory input from a contiguous region of the body's surface called a *dermatome*, and the set of muscles innervated by a single ventral root is called a *myotome*. Dorsal and ventral roots from the same side of the cord join to form a peripheral nerve that serves that side of the body. Peripheral nerves contain mixed sensory and motor fibers from several adjacent spinal roots.

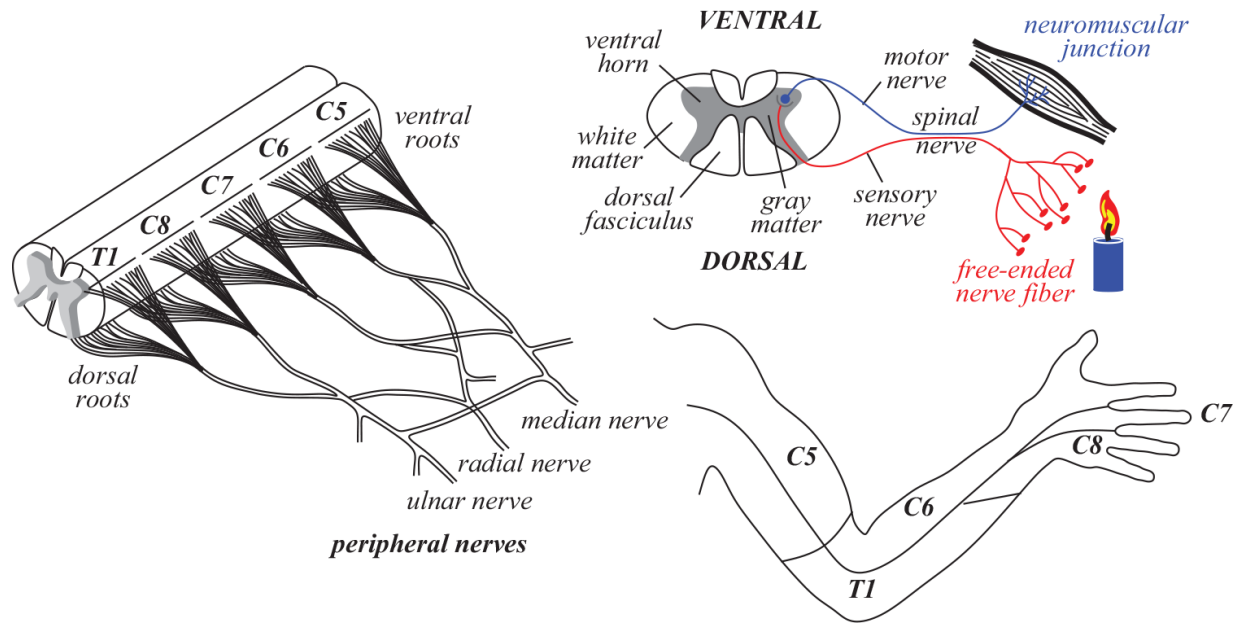


Figure 3.1

The segmental structure of the spinal cord, the corresponding sensory and motor regions of the body, and the cross-sectional anatomy of the cord applied to the withdrawal reflex. (Source: Adapted from [28])

Figure 3.1 also shows how this section of the cord participates in a low-level motor response called the *withdrawal reflex*. It is initiated by stimuli from free-ended nerve fibers that respond to pain and, depending on the position of the stimuli on the dermatome, activate a corresponding set of myotomes through spinal nerves. The number of motor elements enlisted by the reflex is proportional to the strength of the stimulus. The response is generated by synapses in the cord and can be inhibited by higher levels in the CNS. There are many variations on this basic theme. For example, the extensor reflex is a withdrawal reflex in one arm accompanied by an extension about 0.5 [sec] later in the other arm. The same kind of coupling is responsible for reciprocal inhibition to manage antagonistic muscles.

3.1.2 Motor Nuclei

Upper motor neurons (UMNs) in the motor cortex project commands down through a motor hierarchy to muscles. High-level motor commands fan out through subcortical regions, midbrain, brain stem, and spinal interneurons to lower motor neurons (LMNs) in spinal gray matter. Ultimately, these signals terminate at neuromuscular junctions. The LMNs include the α - and

γ -motor neurons in [figure 3.2](#) that connect upward to the rest of the CNS via spinal interneurons and outward to the periphery via the dorsal and ventral roots of spinal nerves. The α -motor neurons are responsible for initiating contractions in extrafusal muscle. A single motor neuron is the exclusive source of activation for a bundle of individual muscle fibers comprising a motor unit that may engage as few as 30 muscle fibers in the small intrinsic muscles of the hand and as many as 3000 muscle fibers in large muscles like the quadriceps.

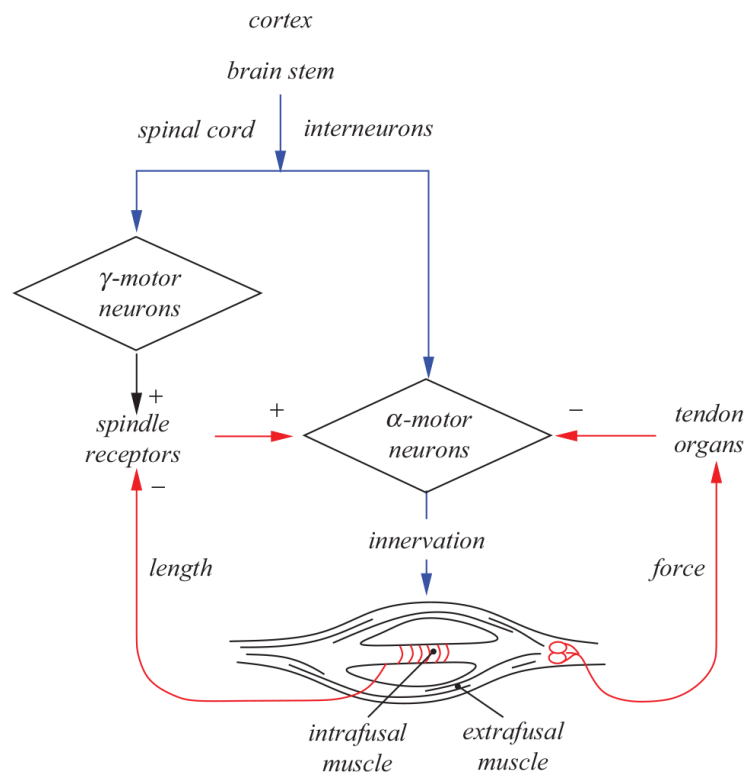


Figure 3.2

The α - and γ -motor feedback loops that regulate muscle tone and stiffness. (Source: Adapted from [192])

The relatively fast α -motor neurons create closed-loop behavior in the motor unit that depends directly on two important proprioceptive mechanoreceptors: the neuromuscular spindle and the Golgi tendon organ (section 7.2.1). Spindles measure the degree of stretch in the muscle fiber and contribute to precision position control. The relatively slow γ -motor neuron regulates muscle tone by setting the reference length of the spindle receptor. If the measured length is greater than the reference, the α -motor

neuron activates a muscle contraction; otherwise, it is inhibited, allowing activity in an antagonist to extend the muscle. The Golgi tendon organ measures the degree of stretch at the tendon-muscle interface and inhibits the α -motor neuron if stretch approaches maximum levels. Thus, muscle and tendon stretch feedback supports stable equilibrium postures in the limbs and provides built-in safety overrides.

The action of the motor unit can be elicited by the familiar clinical practice of striking the patellar tendon just under the kneecap with a rubber mallet. [Figure 3.3](#) illustrates the musculature involved and the motor unit responsible for this response. Striking the tendon induces a slight elongation in muscle spindles in the quadriceps muscle, which causes a reflexive contraction of the muscle and produces a characteristic kick in the leg. The reflexive response is implemented in spinal synapses and cannot be inhibited by higher-level motor processes. The responsible α -motor neurons in this case can be found in spinal segments L2–L4. Similar muscle stretch reflexes can be elicited in the biceps (C5–C6), the triceps (C6–C7), and the ankle (S1). The muscle stretch reflex is also used to compensate for muscle fatigue by increasing α -motor activation and recruiting additional motor units when the original motor units produce diminished responses to changes in length.

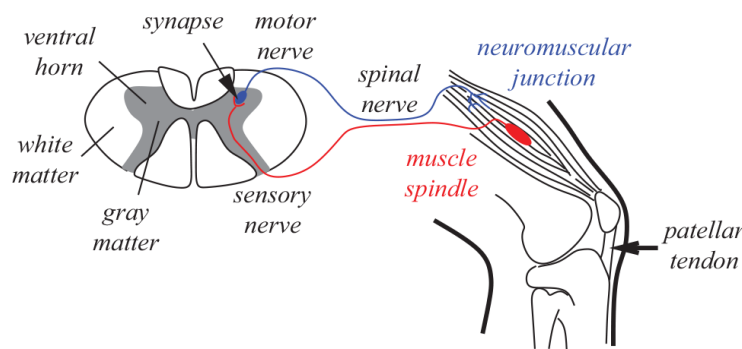


Figure 3.3

The mechanisms that contribute to the quadriceps femoris stretch reflex (L4) in response to the tapping of the patellar tendon

The motor unit is an example of a negative feedback stabilized system. *The position circuit acts to cancel some of the (signed) error in the spindle length.* Under the correct conditions, such a system will track the reference input—in this case, the reference spindle length supplied by the γ -motor

neuron. Negative feedback was first submitted for a patent in 1928 by Harold S. Black. Black's patent application was initially met with skepticism, reportedly compared to a perpetual motion machine. However, it is now recognized as a fundamental principle of stability in compensated dynamical systems. We will examine the conditions that govern the behavior of stable systems in section 3.2.2 after a brief introduction to the dynamics of the spring-mass-damper.

3.2 The Canonical Spring-Mass-Damper

The prototype for many closed-loop robot control applications is the spring-mass-damper (SMD) (figure 3.4). Mass m moves exclusively in the \hat{x} direction and behaves in accordance with Newton's second law $\sum f_x = m\ddot{x}$ (chapter 6). The energy necessary to change the momentum of the mass is stored as kinetic energy,

$$KE = \int (mv)dv = \frac{1}{2}mv^2.$$

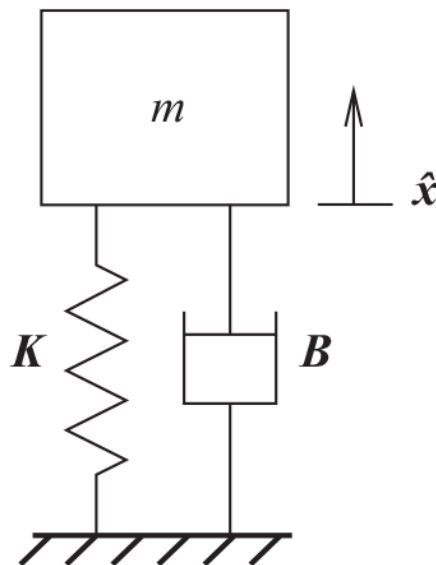


Figure 3.4

The spring-mass-damper (SMD) system

If the mass is displaced from its equilibrium position, spring K produces a force proportional to the deflection and in the opposite direction $f_k = -(Kx)$

\hat{x} . This force tends to restore the spring to its undeformed shape (where $x=0$). The work required to produce this deformation is stored as potential energy in the spring. It is computed by integrating the product of the spring force and the differential displacement of the mass over the trajectory

$$PE = \int (Kx)dx = (1/2)Kx^2.$$

The energy stored in the deformed spring is transformed into kinetic energy when it changes the momentum (mv) of the mass. An ideal spring stores and releases energy reversibly—with no losses.

The third component of the SMD system is the damper B illustrated as a piston-cylinder arrangement (or dashpot) in [figure 3.4](#). To change length, a viscous fluid inside the dashpot must flow through small holes in the piston. The force in the dashpot is due to the resistance to this fluid flow—proportional to and in the opposite direction of the piston velocity $f_B = -(B\dot{x})\hat{x}$. Consequently, whenever mass m is moving, the damper produces a force that dissipates its kinetic energy and tends to bring it back to the rest state over time.

A common application of the SMD is as a mechanism that returns your screen door to the closed position. The door itself is the inertial object and the device that connects the door to the doorjamb contains a spring and a damper. In this case, the damper is a piston and cylinder arrangement with holes in the piston so that air moves into and out of the cylinder subject to viscous losses. Opening the door requires the user to overcome the inertia of the door as well as the forces in the spring and damper. When released, the spring and the damper, properly balanced, cause the door to close quickly and to shut gently. The same principle is used in automobile suspensions to isolate the passenger compartment from bumps and potholes and to keep the tires on the road. Typically, the damper in a screen door will fail as the mechanism ages, allowing unwanted air to leak past the piston by other means than the holes designed for that purpose. As a result, the damping coefficient decreases, the relationship between spring and damper becomes unbalanced, and the door begins to slam shut.

3.2.1 Equation of Motion: The Harmonic Oscillator

If the external disturbance force $f_d(t)$ (illustrated in [figure 3.5](#)), is applied to mass m in the $+\hat{x}$ direction while it is at rest at $x=0$, it will cause the mass to accelerate in the same direction as the applied force and, over time, it will accumulate a positive velocity and displacement. As a consequence, the mass is subject to forces in the $-\hat{x}$ direction due to the spring and the damper as illustrated. In general, all forces and positions in the system are functions of time; however, in the following we will drop the explicit reference to time to simplify the notation.

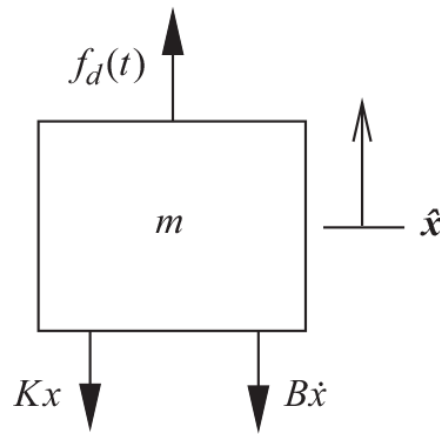


Figure 3.5

The free body diagram of the SMD system

We treat mass m as a free inertial body subject to multiple external forces. For this system, Newton's second law (section 6.1), which states that the sum of all forces causes the mass to accelerate $\sum f=ma$, is written

$$\sum f_x = m\ddot{x} = f_d - B\dot{x} - Kx.$$

By rearranging terms, this equality defines the second-order equation of motion for the one-dimensional SMD.

$$f_d = m\ddot{x} + B\dot{x} + Kx \quad (3.1)$$

The homogeneous (unforced) version of [equation 3.1](#) is the *characteristic differential equation* defining the dynamics of the one-dimensional linear SMD.

$$\ddot{x} + (B/m)\dot{x} + (K/m)x = 0 \quad (3.2)$$

The second-order behavior of the SMD belongs to a class of systems known as *harmonic oscillators*. This form of the equation makes it clear that the dynamic behavior of the SMD depends directly on the mass m and the choice of spring constant K and damping constant B . Equation 3.2 is often written in a manner that highlights physically meaningful aspects of the system,

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0, \quad (3.3)$$

where

$$\zeta = \frac{B}{2\sqrt{Km}}, \quad \text{and} \quad \omega_n = \sqrt{K/m}. \quad (3.4)$$

Parameter ω_n [rad/sec] is the *natural frequency* of the viscoelastic system. It describes the frequency at which the undamped spring and mass exchange potential and kinetic energy if the system is set in motion. The dimensionless parameter, $0 \leq \zeta \leq \infty$, is called the *damping ratio* and represents the relative rates at which energy is dissipated in the damper and released by the spring.

We will see (figure 3.15) that this system is capable of producing qualitatively different behavior in response to the same input depending on the values of ω_n and ζ (and, thus, on the values of K and B). To design control systems, we would like to determine, at a minimum, under what conditions the controlled system is *stable*. In the most basic terms, this dynamical system is stable if the position of the mass stays within bounds in response to inputs. More generally, the stability of any controlled system requires that bounded inputs produce bounded outputs (BIBO) for all time [225]. This criterion and other, stronger conditions of stability are captured for linear and nonlinear systems alike in the elegant methods of Lyapunov.

3.2.2 Stability: Lyapunov's Direct Method

Aleksandr Lyapunov (1857–1918) worked on potential theory and probability theory, and invented methods for characterizing the behavior of dynamical systems such as those we have been considering. Lyapunov's direct method requires a dynamic model of the system and its current *state* to determine future states. The state \mathbf{q} of an n th-order linear differential

equation governing function $f(t)$ is a coordinate in a multidimensional state space at time τ ,

$$\mathbf{q}_\tau = [q_0 \dots q_{n-1}]^T, \text{ where } q_i = \left. \frac{d^i f(t)}{dt^i} \right|_{t=\tau}$$

is the value of the i -th time derivative of $f(t)$ evaluated at time τ .²

Perhaps the simplest condition that guarantees stability requires that the state of the system never leaves the interior of a bounded region in the *phase space* defined by \mathbf{q} . This condition is stated in definition 3.1.

Definition 3.1: Stability. *The origin of the state space is stable if there exists a region $S(\mathbf{q})$ containing the origin, such that system trajectories that begin at states within $S(\mathbf{q})$ remain within $S(\mathbf{q})$.*

For the second-order differential equation describing the SMD, the phase space is defined by two independent state variables (x, \dot{x}) . Figure 3.6 illustrates the *phase plane* describing the state of the second-order SMD system and identifies a region $S(\mathbf{q})$ that contains the origin. The state of the SMD (the gray dot in the diagram) is a coordinate in the (x, \dot{x}) plane and evolves according to the second-order dynamics (equations 3.2 and 3.3) of the system. Definition 3.1 requires that states that start inside region $S(\mathbf{q})$ remain there always.

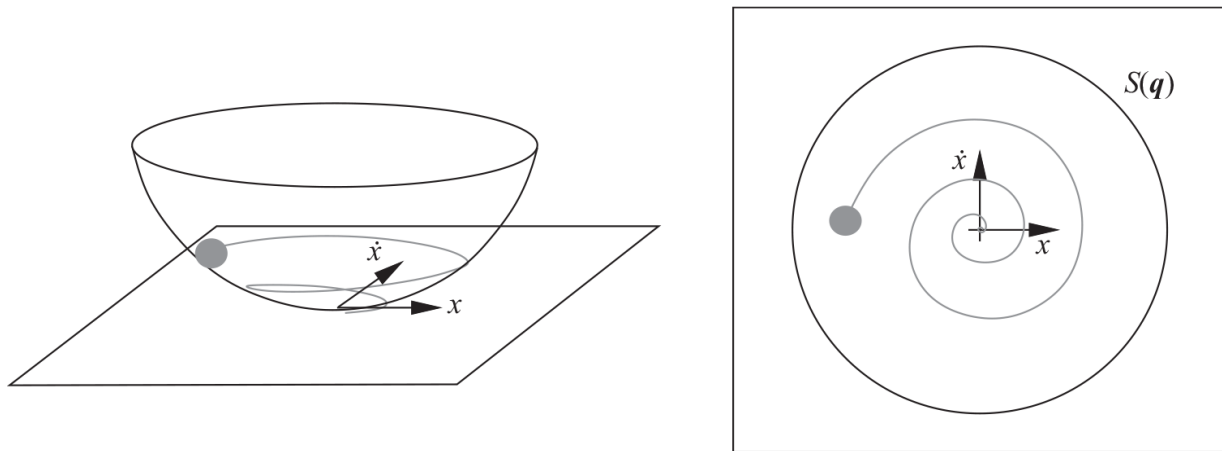


Figure 3.6

A trajectory of an asymptotically stable harmonic oscillator on a bowl-shaped Lyapunov function (*left*) and the top-view projection on the phase plane (*right*)

A subset of Lyapunov stable systems—called asymptotically stable systems—satisfies a stronger condition where the system comes to rest at the origin of the state space as $t \rightarrow \infty$. For the one-dimensional SMD in [figure 3.6](#), state variables (x, \dot{x}) both approach zero as time goes to infinity.

Definition 3.2: Asymptotic stability. *Systems that satisfy definition 3.1 are asymptotically stable if as $t \rightarrow \infty$, the systems state approaches the origin of the state space.*

Qualitative specifications such as these are formalized by way of the *Lyapunov function*, $V(\mathbf{q}, t)$ —a scalar function written in terms of state variables \mathbf{q} and time that is continuous in all first derivatives.

Theorem 3.2.1: (Lyapunov’s direct method (1892)). *If function, $V(\mathbf{q}, t)$, exists such that*

$$V(\mathbf{0}, t) = 0$$

$$V(\mathbf{q}, t) > 0, \text{ for } \mathbf{q} \neq \mathbf{0} \text{ (} V(\mathbf{q}, t) \text{ is positive definite), and}$$

$$\frac{dV(\mathbf{q}, t)}{dt} < 0 \text{ (} dV/dt \text{ is negative definite),}$$

then the dynamical system described by $V(\mathbf{q}, t)$ is asymptotically stable.

The first two conditions in Lyapunov’s direct method establish that the origin is a unique minimum over the entire phase space. The third condition requiring a negative definite dV/dt guarantees that the closed-loop dynamics of the system lead inexorably downhill on $V(\mathbf{q}, t)$ to the unique minimum. Lyapunov established that if the system is stable, then a Lyapunov function that satisfies these criteria is guaranteed to exist—a system that complies with the conditions outlined in theorem 3.2.1 is said to be *stable in the sense of Lyapunov* or *Lyapunov stable*.

Lyapunov’s direct method is of great consequence because it establishes a strong stability guarantee without ever explicitly solving for a time-domain solution to the dynamic equations of motion and can be applied to linear and nonlinear systems alike. However, if a candidate Lyapunov function does not satisfy the conditions of theorem 3.2.1, it does not

constitute proof that the system is unstable. In this situation, one must continue to search for a suitable Lyapunov function.

Example: Stability Analysis for the Spring-Mass-Damper

For the one-dimensional spring-mass-damper (figure 3.4) the state is defined in the position-velocity space (x, \dot{x}) . Consider a Lyapunov function that is the sum of kinetic energy and potential energy.

$$E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}Kx^2 \quad (3.5)$$

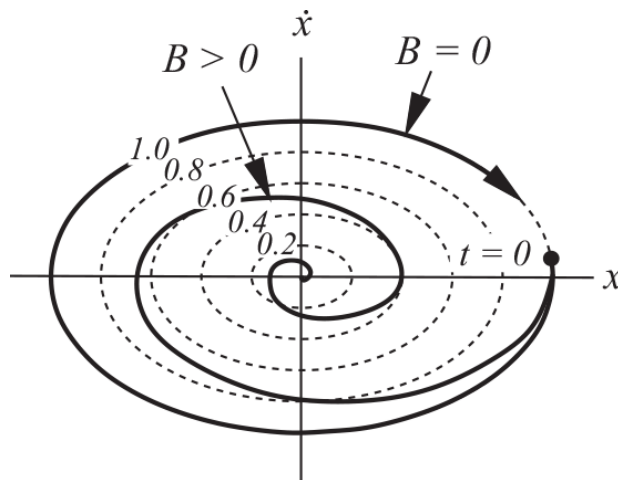


Figure 3.7
Qualitative state space trajectories of the harmonic oscillator

Level curves in the energy function are elliptical (figure 3.7) with shapes determined by the relative values of m and K . Energy is zero at the origin of the state space and is greater than zero everywhere else, thus equation 3.5 serves as a reasonable candidate Lyapunov function. Differentiating equation 3.5 with respect to time yields

$$\frac{dE}{dt} = m\dot{x}\ddot{x} + Kx\dot{x}. \quad (3.6)$$

This expression can be written exclusively in terms of the state variables by solving equation 3.2 for \ddot{x} in terms of x and \dot{x} . Rewriting equation 3.6 accordingly yields

$$\begin{aligned}\frac{dE}{dt} &= m\dot{x} [-(B/m)\dot{x} - (K/m)x] + Kx\dot{x} \\ &= -B\dot{x}^2.\end{aligned}\tag{3.7}$$

Since $\dot{x}^2 \geq 0$, whenever $B > 0$ the rate of change in the system energy $dE/dt \leq 0$. Every cycle, however, $dE/dt=0$ when \dot{x} is zero at the crest and trough of each oscillation. This observation precludes a global assertion of asymptotic stability by the direct method. We can repair the analysis by noticing that the only equilibrium position for this system is at $x=\dot{x}=0$. Therefore, other states for which $x \neq 0$ and $dE/dt=0$ are still subject to nonzero spring forces directed toward $x=0$. As the mass accumulates velocity, once again [equation 3.7](#) guarantees that energy is dissipated. This happens every cycle in the harmonic oscillator until the state converges to the origin of the state space. The qualitative result is a trajectory like the clockwise spiral labeled $B > 0$ in [figure 3.7](#). Moreover, if $B = 0$, energy is conserved and the system orbits along the $E = 1.0$ isolevel curve continuously—reversibly exchanging potential energy in the spring for kinetic energy in an unending cycle.

Note that, even though we believe the SMD to be asymptotically stable for $B > 0$, we have not yet proved that result using Lyapunov’s direct method alone—to do that, we must find a new candidate for the Lyapunov function that satisfies all the conditions required in [theorem 3.2.1](#). A fruitful approach is to add a *skew* term of the form $(\epsilon m x \dot{x})$ into [equation 3.5](#), where ϵ is an appropriately chosen positive constant. Under this Lyapunov function, the derivative with respect to time can be shown to be negative definite (see [exercise 3.4.8.b](#)). As a result, we conclude that the linear spring-mass-damper system is asymptotically stable over the entire state space as we suspected.

□

Global stability assertions for more complicated systems are quite difficult to establish in general, but it is sometimes significantly easier to establish local asymptotic stability guarantees.

Theorem 3.2.2: (Local asymptotic stability). *Beginning from the usual conditions on the Lyapunov function, $V(\mathbf{0}, t)=0$ and $V(\mathbf{q} \neq \mathbf{0}, t) \geq 0$, if it is possible to establish a bound V_C , such that*

- the sublevel set $C = \{\mathbf{q} \mid (V(\mathbf{q}, t) \leq V_c)\}$ is bounded, and
- $dV(\mathbf{q}, t)/dt \leq -\lambda V(\mathbf{q}, t)$ for all $\mathbf{q} \in C$,

then every trajectory that starts in C converges to $\mathbf{q}=\mathbf{0}$ as $t \rightarrow \infty$.

3.3 Proportional-Derivative Feedback Control

Harmonic oscillators like the SMD are very common in nature and they provide a useful model for describing important features of biological motor systems. In particular, the viscoelastic parameters of the SMD can be used to emulate the bulk elasticity and damping of muscle tissue (section 2.1.3) and to implement negative feedback like that produced by the α - γ motor neurons (section 3.1.2) to stabilize the system over a wide range of these parameters. The passive properties of the SMD make it an attractive candidate for a closed-loop, parametric motor unit in robotic actuation systems where limb position, velocity, force, and stiffness can be controlled.

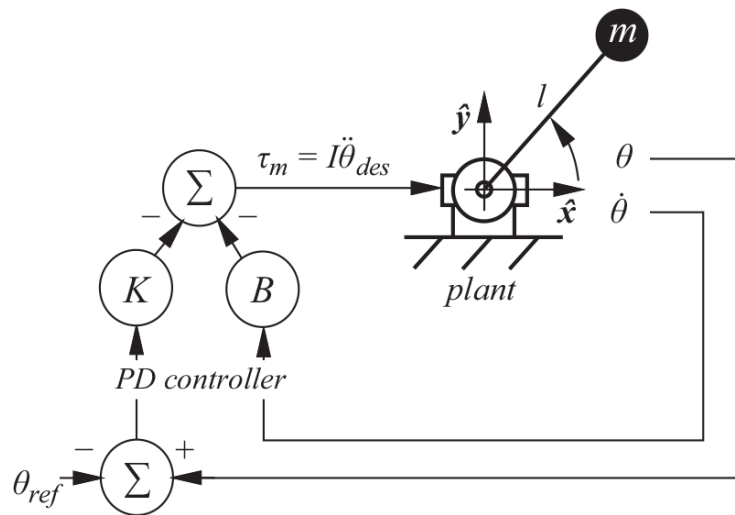


Figure 3.8

The 1 DOF revolute (1R) mechanism is the inertial object in a PD control process with reference θ_{ref}

Figure 3.8 illustrates a hybrid implementation of the SMD that combines sensory, computational, and motor elements to build an integrated, one degree of freedom (1-DOF), closed-loop motor unit. The closed-loop design includes feedback sensors that measure the angular position and velocity ($\theta, \dot{\theta}$) of the limb and a proportional-derivative (PD) control process (named

after the feedback it requires) that simulates the sum of the torques from a torsional spring K [$N \cdot m/rad$] and damper B [$N \cdot m/(rad/s)$], so that

$$\tau_m = -B\dot{\theta} - K(\theta - \theta_{ref}).$$

The resulting motor command torque τ_m is submitted to the motor driver circuitry to accelerate the limb consisting of mass m located at a distance l from the axis of rotation and, thus, approximate the behavior of a SMD.

Euler's equation $\sum \tau = I\ddot{\theta}$ expresses Newton's second law for such a system, where τ [$N \cdot m$] is torque applied to the load at the origin and $I = ml^2$ [$kg \cdot m^2$] is the combined moment of inertia of the motor and the load about the \hat{z} axis. The derivation of the differential equation governing the response of the PD controlled motor unit follows that of the analog SMD on which it is based (section 3.2.1). The homogeneous (unforced) equation of motion for the 1-DOF revolute (1R) motor unit is

$$\begin{aligned} \sum \tau = I\ddot{\theta} &= -B\dot{\theta} - K(\theta - \theta_{ref}), \\ I\ddot{\theta} + B\dot{\theta} + K(\theta - \theta_{ref}) &= 0, \quad \text{or} \\ \ddot{\theta} + \frac{B}{I}\dot{\theta} + \frac{K}{I}(\theta - \theta_{ref}) &= 0. \end{aligned} \quad (3.8)$$

As before, [equation 3.8](#) can be written in terms of the damping ratio ζ and natural frequency ω_n of the harmonic oscillator,

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2(\theta - \theta_{ref}) = 0, \quad \text{where, in this case,} \quad (3.9)$$

$$\zeta = \frac{B}{2\sqrt{KI}}, \quad \text{and} \quad \omega_n = \sqrt{K/I}. \quad (3.10)$$

The PD feedback implementation of the SMD provides flexibility. In particular, it makes it easy to change the reference position of the controller in a manner analogous to the way the γ -motor neuron is used to set the reference length of a muscle spindle, and it makes it convenient to change the values of the control parameters K and B .

The added flexibility of the hybrid PD controller comes with a cost. The implementation employs discrete-time inputs to control a continuous-time plant by sampling the state feedback, storing it in memory, computing the

command torque, and writing the result in another memory location for use by the motor driver circuitry. This sample-and-hold process introduces a small time lag between the perceived state and the motor response. If this lag is comparable to or greater than the natural period of the harmonic oscillator ($1/\omega_n$ [sec]), then continuous-time analysis is inaccurate and the resulting control design may produce an unstable system response.³ If, however, the time lag is small relative to $1/\omega_n$, then the error associated with continuous-time analysis is small as well, and solutions for $\theta(t)$ given input references $\theta_{ref}(t)$ can be good approximations of the behavior of real systems.

In the following, we focus on this case and introduce tools for continuous-time analysis using the Laplace transform. The Laplace transform makes it easy to manipulate and solve linear differential equations by transforming them into an algebraic form. In this form, solutions for system behavior can be determined relatively easily for a variety of inputs to provide insight into the performance of controlled movement that applies equally to animal and robot motor systems.

The rest of this chapter provides a brief introduction to the properties of the Laplace transform that are most relevant to this discussion and summarizes the results of applying this analytical tool to the analysis of the 1R motor unit in [figure 3.8](#). The interested reader can find the complete derivation for these results in appendix A.10—the more casual reader can skip the appendix without compromising the background required by subsequent chapters.

3.3.1 A Primer for Laplace Transforms

The Laplace transform is a bijection defined by a pair of reciprocal mappings,

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st} dt \quad \text{and} \quad f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st} ds.$$

[Table A.2](#) in appendix A.10.3 summarizes several Laplace transform pairs that relate time-domain functions to their counterparts in the complex frequency-domain. As we will discuss further, they provide a uniform mechanism for analyzing the behavior of control systems subject to a large

variety of reference inputs $\theta_{ref}(t)$ and, thus, reveal a great deal about a particular control design.

The function $F(s)$ is the *image* of $f(t)$ in the complex s -domain. Ignoring boundary conditions,⁴ this exponential basis leads to an important consequence. If $\mathcal{L}[f(t)]=F(s)$, then

$$\mathcal{L}\left[\frac{df(t)}{dt}\right]=sF(s), \quad \text{or, more generally,} \quad \mathcal{L}\left[\frac{d^n f(t)}{dt^n}\right]=s^n F(s). \quad (3.11)$$

Therefore, the Laplace transform maps an n th-order linear differential equation with constant coefficients describing the behavior of function $f(t)$ into an n th-order polynomial $F(s)$. The inverse Laplace transform $f(t)=\mathcal{L}^{-1}[F(s)]$ represents $f(t)$ as a weighted sum of exponential functions of the form e^{st} parameterized by complex frequencies $s \in \mathbb{C}$.

Applied to the PD controller (equation 3.9), the Laplace transform yields a dual representation for the second-order system.

$$\begin{array}{ccc} \ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = \omega_n^2\theta_{ref} & \begin{array}{c} \xrightarrow{\mathcal{L}[\cdot]} \\ \xleftarrow{\mathcal{L}^{-1}[\cdot]} \end{array} & [s^2 + 2\zeta\omega_n s + \omega_n^2] \Theta(s) = \omega_n^2 \Theta_{ref}(s) \quad (3.12) \\ \text{time-domain} & & \text{complex frequency-domain} \end{array}$$

3.3.2 Stability in the Time-Domain

The behavior of the hybrid system illustrated in figure 3.8 is completely determined once the parameters (ζ, ω_n) , or equivalently (K, B) , are specified in the equation of motion. For example, in the homogeneous form of the right side of equation 3.12, the polynomial

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (3.13)$$

defines the *characteristic equation* of the harmonic oscillator. The pair of roots to equation 3.13,

$$s_{1,2} = \frac{-2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - 4\omega_n^2}}{2} = \frac{2\omega_n[-\zeta \pm \sqrt{\zeta^2 - 1}]}{2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}, \quad (3.14)$$

may be complex conjugates $\sigma \pm i\omega$ with real $\sigma = \text{Re}(s)$ and imaginary $\omega = \text{Im}(s)$ parts, repeated real roots, or distinct real roots and *determine the*

values of s in basis functions of the form Ae^{st} that comply with constraints imposed by the original homogeneous second-order differential equation $\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = 0$.

The roots of the characteristic equation depend solely on the parameters of the hybrid SMD, but they govern the PD response more globally—in particular, they lead to straightforward conclusions regarding closed-loop stability. Clearly, if any roots of the characteristic equation have positive real parts $Re(s) > 0$, then, as $t \rightarrow \infty$, terms of the form Ae^{st} will also grow to infinity, and the state of the system $(\theta, \dot{\theta})$ will leave region $S(\mathbf{q})$ (figure 3.6). Such a system is unstable by definition 3.1. Conversely, when $Re(s) \leq 0$, the system is stable, and if $Re(s) < 0$, the system is asymptotically stable so that the error goes to zero exponentially as $t \rightarrow \infty$.

3.3.3 The Transfer Function, SISO Filters, and the Time-Domain Response

Equation 3.12 can be rewritten to describe the 1R motor unit (figure 3.8) in the complex frequency-domain.

$$\begin{aligned} [s^2 + 2\zeta\omega_n s + \omega_n^2] \Theta(s) &= \omega_n^2 \Theta_{ref}(s), \quad \text{or} \\ \frac{\Theta(s)}{\Theta_{ref}(s)} &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned} \quad (3.15)$$

Among the advantages of describing the system in the s -domain is the ability to express the PD control law as a single-input, single-output (SISO) filter as in figure 3.9. This perspective has its roots in signal processing, where these filters are called *transfer functions*.

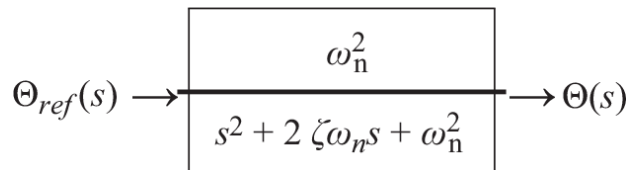


Figure 3.9

The transfer function for the generic harmonic oscillator in the form of a SISO filter

Transfer functions provide a convenient mechanism for composing filters and, thus, analyzing complicated, interacting dynamical systems. As a result, control systems can be composed using modular transfer functions with independent dynamics, and the composite transfer function can be derived by manipulating the circuit into the equivalent closed-loop transfer function.

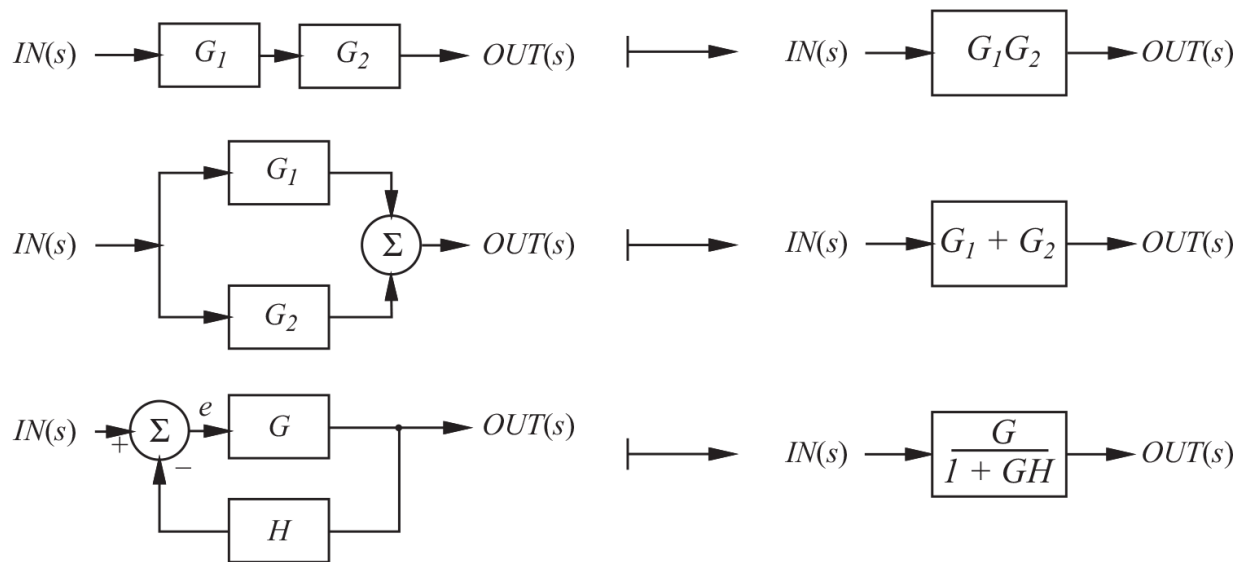


Figure 3.10

Series and parallel composition of transfer functions and the closed-loop transfer function (CLTF)

For example, [figure 3.10](#) illustrates how linear combinations of pairs of transfer functions combine to form an equivalent SISO filter. Of particular importance to us is the combination of transfer functions in closed-loop feedback systems. The loop structure on the bottom of [figure 3.10](#) is a generic prototype for such systems. It incorporates a *feedback* transform H to prepare sensor data for comparison to reference inputs, a summation junction where feedback errors are computed, and a *feedforward* transform G (which includes the plant and the feedforward controller) to compute system outputs. Writing the ratio of output $OUT(s)$ to input $IN(s)$ for the loop structure

$$\frac{OUT(s)}{IN(s)} = \frac{Ge}{IN(s)} = \frac{G[IN(s) - H(OUT(s))]}{IN(s)}, \quad \text{so that}$$

$$\frac{OUT(s)(1 + GH)}{IN(s)} = G, \quad \text{and therefore} \quad \frac{OUT(s)}{IN(s)} = \frac{G}{1 + GH}. \quad (3.16)$$

Equation 3.16 defines the *closed-loop transfer function* (CLTF) of a feedback control system. It captures the dynamics of the feedback control structure in the form of a single equivalent SISO filter.

Together, identities like those in figure 3.10 provide a means of transforming more complex closed-loop systems into equivalent closed-loop transfer functions.

Example: Closed-Loop Oculomotor Transfer Function

Figure 3.11 illustrates the geometry of a pair of antagonistic muscles in the human *oculomotor system* that are primarily responsible for *panning* movements of the eyeball (section 7.1.2). These movements are used to track moving stimuli on the retina or to quickly saccade to targets inferred from other sensor data—to multiple elements of a complex visual scene or to the direction of a sound, for example. In the latter case, the history of stimuli is said to orient the closed-loop visual response.

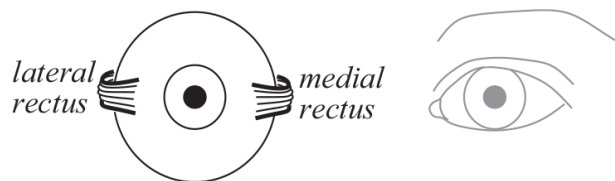


Figure 3.11

The musculature for the eye's lateral (pan) degree of freedom

Figure 3.12 shows a simple, linear control model for this part of the oculomotor control system, in which the eye is the inertial object ($1/I$) subject to passive damping (B) in the musculature. This part of the control is bold in the figure.

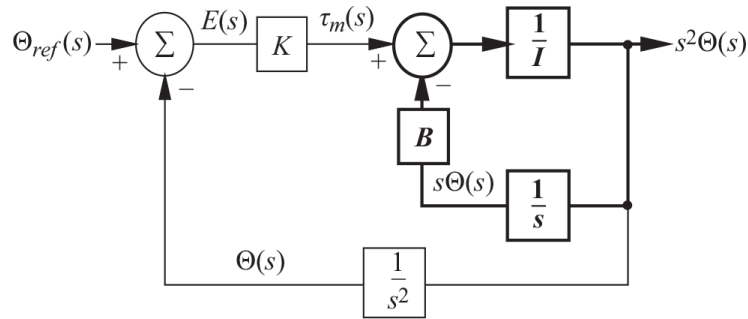


Figure 3.12

A control model for the 1-DOF oculomotor system consists of active and passive elements

The torque generated by the antagonistic muscles is modeled by an elastic element that produces a motor torque proportional to the error,

$$\mathcal{T}_m(s) = K(\Theta_{ref}(s) - \Theta(s)).$$

The sum of torques generated actively by the motor and passively by the damper is applied to the inertial load. Integrators, $1/s$ and $1/s^2$, incorporated into the nested feedback loops are required to transform the output acceleration $s^2\Theta(s)$ into velocity and position feedback, respectively. A sequence of pairwise compositions from [figure 3.10](#) is used to simplify this relatively complex control circuit in order to determine the net, closed-loop transfer function for the PD controlled, oculomotor system.

In [figure 3.13\(a\)](#) the inner loop is simplified by performing a serial combination of the damper and the velocity integrator.

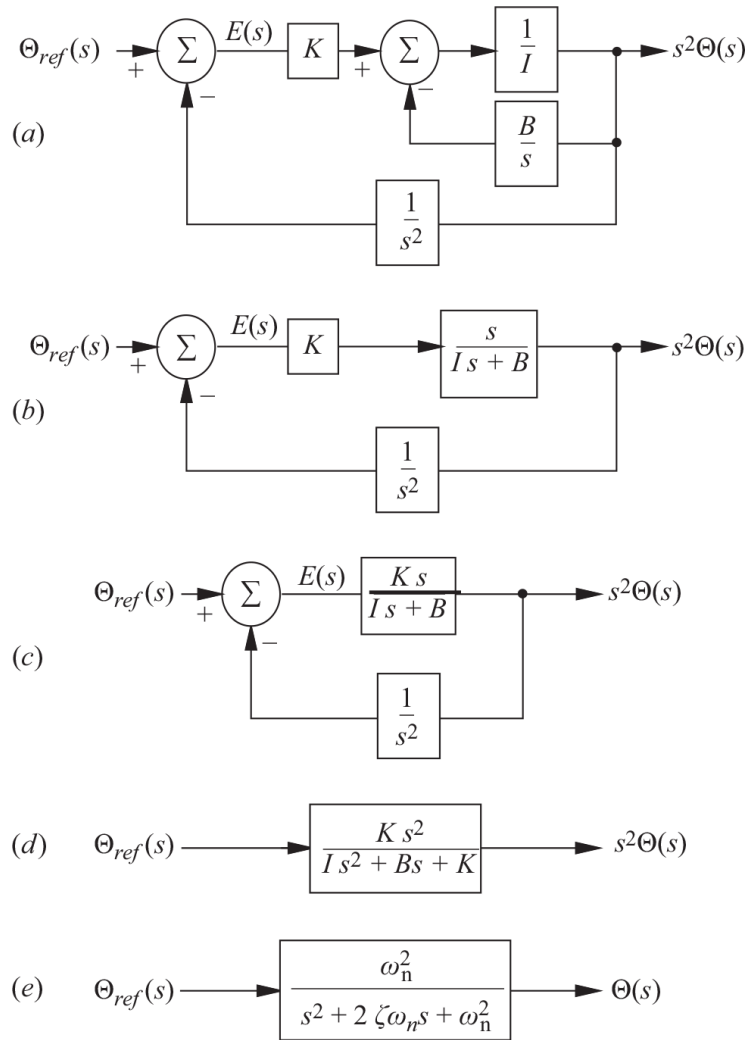


Figure 3.13

A sequence of compositions yields the closed-loop transfer function for the PD controlled, oculomotor system

From [figure 3.13\(a\) to \(b\)](#), the inner loop is replaced by the local closed-loop transfer function with

$$G = \frac{1}{I} \quad \text{and} \quad H = \frac{B}{s}, \quad \text{so that}$$

$$\frac{G}{1 + GH} = \frac{1/I}{1 + B/Is} = \frac{s}{Is + B} .$$

The series combination of feedforward transforms in (b) is evaluated yielding (c).

Once again, an equivalent closed-loop transfer function is derived that results in (d), where

$$G = \frac{1}{I} \quad \text{and} \quad H = \frac{B}{s}, \quad \text{so that}$$

$$\frac{G}{1 + GH} = \frac{1/I}{1 + B/Is} = \frac{s}{Is + B} .$$

Finally, by converting the output to $\Theta(s)$ and transforming into standard form, we arrive at [figure 3.13\(e\)](#).

Compare this result with the Laplace transform of the equation of motion for the 1R motor unit proposed in [equation 3.15](#).

□

The closed-loop transfer function in [figure 3.13\(e\)](#) represents the PD controlled plant in a form that supports analysis for any Laplace-transformable input function, $\theta_{ref}(t)$.

3.3.4 The Performance of Proportional-Derivative Controllers

An exponential basis function for a complex root of [equation 3.13](#) can be written as the product of two exponential functions $e^{(\sigma + i\omega)t} = e^{\sigma t} e^{i\omega t}$. For all stable parameterizations of [equation 3.13](#) (i.e., when $\sigma = \text{Re}(s) \leq 0$), the first term $e^{\sigma t}$ is an exponentially decreasing function of time. By Euler's formula ([appendix A.10.1](#)), the second term can be written $e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$. This term, therefore, introduces oscillatory responses for roots with imaginary parts. PD control responses are, thus, the product of an exponentially decreasing response and an oscillatory response—the relative importance of these two responses depends on parameters K and B .

The unit step response—The unit step response is a natural choice for characterizing the behavior of the PD control design in [figure 3.8](#). It represents a discrete change in the reference input in [figure 3.14](#), where the reference input changes instantaneously at $t=0$ from 0 to 1 [rad] where it remains for $0 \leq t < \infty$.

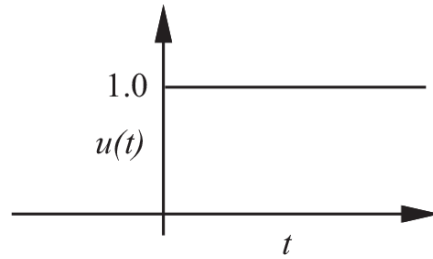


Figure 3.14

The unit step reference input

Appendix A.10.3 treats this case and derives the control responses in [figure 3.15](#) for a PD controller with fixed values of spring constant K and rotational moment of inertia I and different values for the damping coefficient B that produce damping ratios $0 \leq \zeta \leq 2$. Recall that ζ represents the ratio of the rate at which kinetic energy is dissipated by the damper to the rate at which potential energy is released from the spring. Low values of ζ are, thus, dominated by the energetics of the spring and are oscillatory, while larger values of ζ are increasingly dissipative.

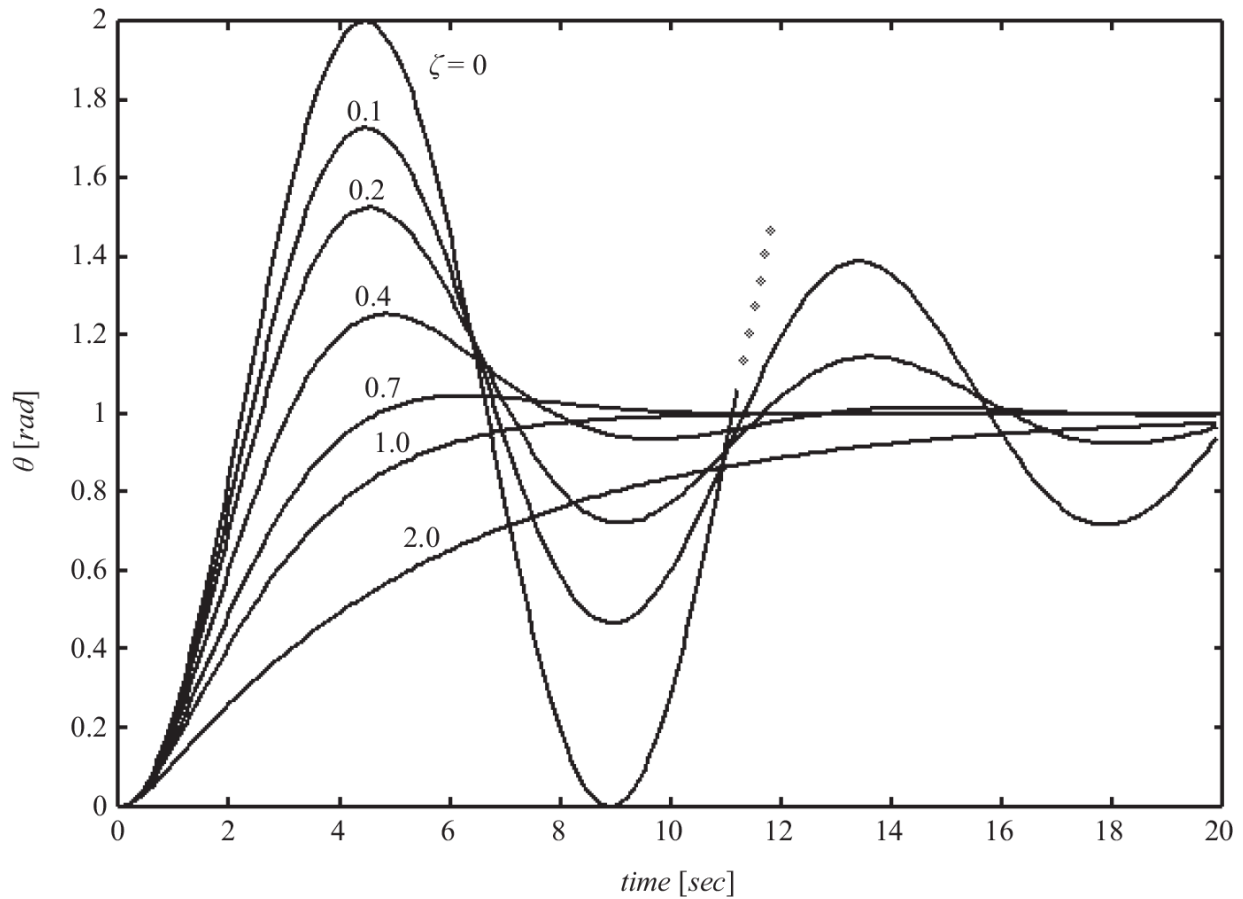


Figure 3.15

The response of the second-order PD position controller in [figure 3.8](#) as a function of ζ ($K=1.0$ [Nm/rad], $I=2.0$ [kgm²]) given boundary conditions $\theta_0 = \dot{\theta}_0 = 0$ and $\theta_\infty = 1.0$ [rad]

When the controlled system is stable, different configurations of the PD control law produce four qualitatively different responses.

Undamped—When B , and therefore ζ , are zero, the roots of [equation 3.13](#) are purely imaginary complex conjugates ($s_{1,2} = \pm i\omega$). There is no dissipative element in the system and as result, the control response conserves energy by storing and releasing energy in the spring at the natural frequency $\omega_n = \sqrt{K/I}$ of the system. This response is illustrated in [figure 3.15](#) by the curve labeled $\zeta = 0$. For the idealized mathematical formulation where, for instance, springs are perfectly reversible, the system produces a constant amplitude oscillation for $t > 0$. Such a system is *marginally* or *orbitally* stable.

Underdamped—For $0 < \zeta < 1$, the roots of [equation 3.13](#) are complex conjugates with real $\sigma = \text{Re}(s) < 0$ and imaginary $\omega = \text{Im}(s)$ parts. The imaginary parts lead, once again, to oscillation, but the (negative) real parts are dissipative and introduce an exponentially decaying envelope that causes the amplitude of oscillation to decrease exponentially over time. For small positive convergence thresholds, a slightly underdamped response will converge more quickly than other PD configurations. Underdamped behavior—illustrated in curves labeled $\zeta = 0.1, 0.2, 0.4,$ and 0.7 in [figure 3.15](#)—is *asymptotically* stable by definition 3.2.

Critically damped—When $\zeta = 1$, there exist repeated, negative real roots to [equation 3.13](#). For a given moment of inertia I , this corresponds to a unique combination of values for K and B that separate under- and over-damped configurations. This configuration of the PD controller, illustrated using the $\zeta = 1$ curve in [figure 3.15](#), is non-oscillatory, asymptotically stable, and often selected as an optimal control design point.

Overdamped— $\zeta > 1$ corresponds to two distinct, negative, real roots so that the time-domain solution is the weighted sum of two negative exponential terms. The overdamped PD response is dominated by the damper's viscous behavior and is a consequence of the behavior generated by the least negative real root. It is non-oscillatory and takes more time to converge as ζ increases. An example of the overdamped response in [figure 3.15](#) is the curve labeled $\zeta = 2.0$.

Example: Controlling Eye Movements

Oculomotor responses include saccadic movements of up to 900 [*deg/sec*]—the fastest movements in the human body. A saccade can be triggered by anticipatory goals (e.g., when your eyes scan ahead while reading text). In contrast, smooth-pursuit oculomotor controllers track features as the visual scene evolves over time.

All oculomotor responses depend on closed-loop motor units ([figure 3.12](#)) that combine passive damping, active contractile forces, and α - γ motor neurons to actuate the eye. Starting from the closed-loop transfer function for the generic motor unit ([figure 3.13\(e\)](#)) and selecting a critically damped response ($\zeta = 1.0$), the closed-loop transfer function is

$$\frac{\Theta(s)}{\Theta_{ref}(s)} = \frac{\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2}.$$

From [Table A.2](#), the Laplace transform of the unit step is

$$\mathcal{L}[u(t)] = \frac{1}{s}.$$

Therefore, the system response to the unit step input of magnitude θ_{ref} is

$$\Theta(s) = \left[\frac{\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \right] \left[\frac{\theta_{ref}}{s} \right] = \frac{\omega_n^2 \theta_{ref}}{s(s + \omega_n)^2}. \quad (3.17)$$

The mapping back to the time-domain response is defined by the inverse Laplace transform ([equation A.20](#)); however, we can accomplish the same result by rewriting [equation 3.17](#) in terms of a linear combination of Laplace transform pairs in [table A.2](#). A partial-fraction expansion of this quotient yields

$$\begin{aligned} \Theta(s) &= \theta_{ref} \left[\frac{\omega_n^2}{s(s + \omega_n)^2} \right] = \theta_{ref} \left[\frac{a}{s} + \frac{b}{s + \omega_n} + \frac{c}{(s + \omega_n)^2} \right] \\ &= \theta_{ref} \left[\frac{1}{s} + \frac{-1}{s + \omega_n} + \frac{-\omega_n}{(s + \omega_n)^2} \right], \end{aligned} \quad (3.18)$$

so that [table A.2](#) can be used to determine the inverse Laplace transform term by term.

$$\theta(t) = \theta_{ref} \left[1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t} \right] \quad (3.19)$$

Compare this result to the solution to the equivalent boundary value problem ([equation A.24](#)).

□

This example illustrates how the closed-loop response of the PD controller can be derived for any Laplace-transformable driving function (e.g., [table A.2](#)). These inputs provide useful diagnostic performance feedback. Of particular importance is the response of the harmonic oscillator to oscillatory driving functions, which we examine next.

Frequency response: Amplitude and phase—Other input reference functions can be used to identify different aspects of the behavior of the classical harmonic oscillator. Consider, for example, the time varying input reference function $\theta_{ref}(t)=A\cos(\omega t)$.⁵ Figure 3.16 plots the amplitude and phase of the harmonic oscillator relative to the amplitude and phase of the $A\cos(\omega t)$ reference input for various values of the driving frequency ω and damping ratios ζ . The PD control model exhibits significant frequency-dependent behavior at different driving frequencies. An obvious feature of the response is the sharp peak in the amplitude response. This peak is a *resonance* phenomenon that is located at driving frequencies where $\omega/\omega_n=1$, that is, when the reference input frequency is equal to the natural frequency of the harmonic oscillator. This effect is more pronounced as the damping ratio decreases. In fact, for $\zeta=0$, the gain becomes theoretically infinite—the undamped ($\zeta=0$) system can be destabilized by pumping energy in at the resonant frequency much like a child pumps to increase the amplitude of a playground swing.

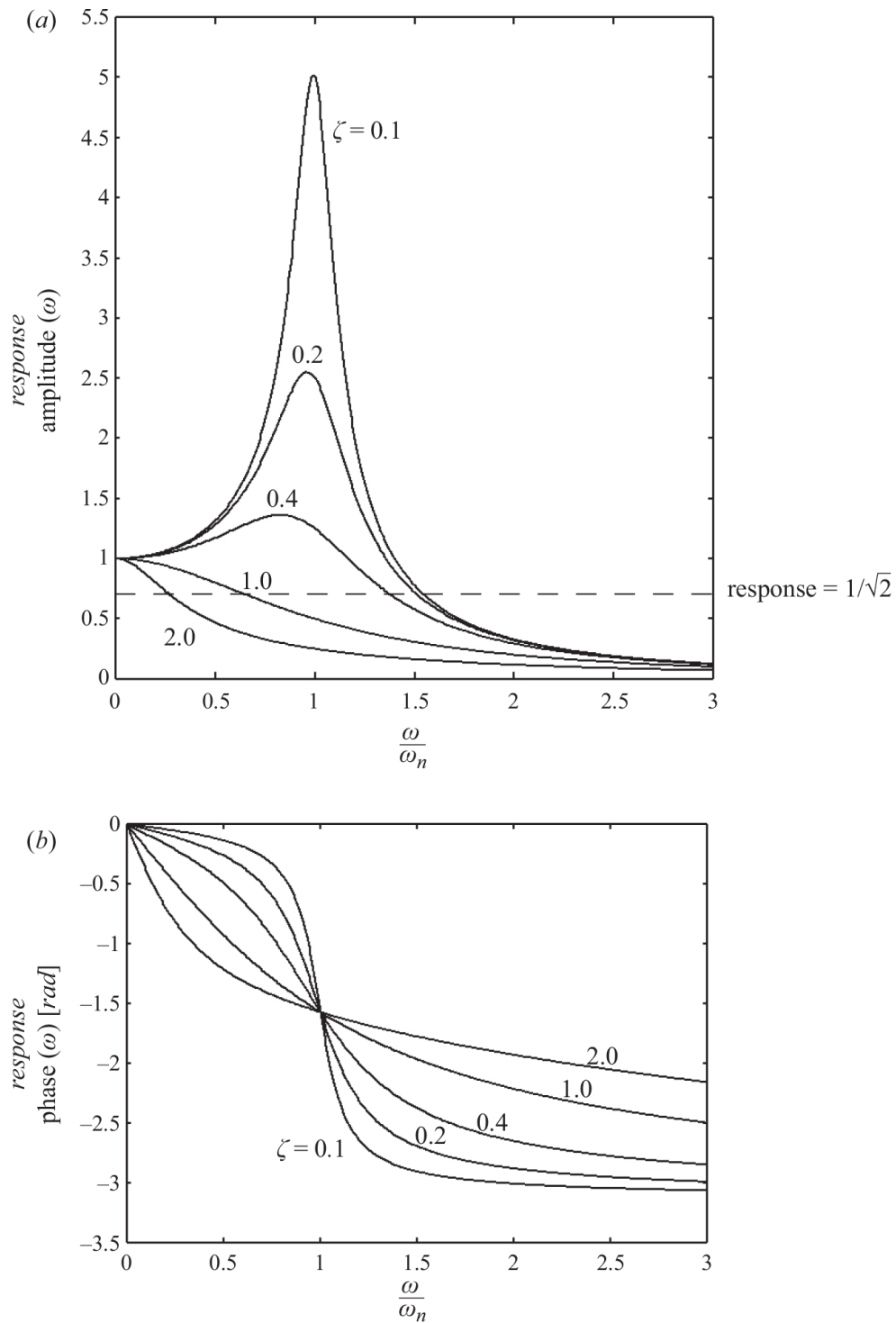


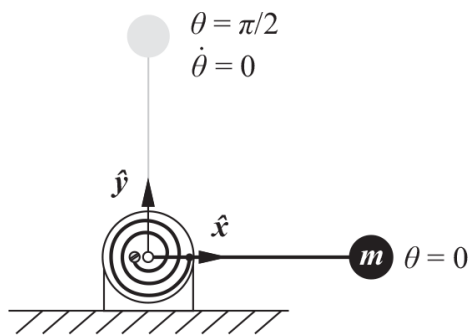
Figure 3.16

The response of the second-order PD position controller to $\cos(\omega t)$ input function, where ω is the driving frequency for various values of ζ

Figure 3.16(a) shows that if the driving frequency is relatively large, the amplitude response goes to zero asymptotically and the ability of the PD controller to track the reference input is compromised. The *bandwidth* of the system is that frequency where the amplitude response drops $1/\sqrt{2}$ of the reference amplitude. The input frequency corresponds to the *half power* or -3 dB point of the PD controller. The over- and critically damped configurations in figure 3.16(a) have a relatively small bandwidth using this criteria; the situation improves for underdamped configurations at the expense of overshoot and ringing in the resulting oscillatory output. In figure 3.16(b), we see that the natural frequency also identifies the point at which the response lags 90 [deg] behind the reference input. If the driving frequency increases beyond the natural frequency, the response goes asymptotically toward a response that is 180 [deg] out of phase with the forcing function.

3.4 Exercises

1. **The spring-mass-damper (SMD).** A torsional SMD system is illustrated in its equilibrium position at $\theta = 0$ (there is no gravity). The moment of inertia I of the system about the \hat{z} axis at the base is 1 [kg m²]. The response of the system to angular displacements from equilibrium is governed by the spring K and the damper B (not shown).



$$m = 0.5 \text{ [kg]}$$

$$l = \sqrt{2} \text{ [m]}$$

$$I = ml^2 = 1 \text{ [kg m}^2\text{]} \quad I\ddot{\theta} + B\dot{\theta} + K\theta = 0$$

$$K = 10 \text{ [Nm/rad]}$$

$$B = 11 \text{ [Nm sec/rad]}$$

- Write the characteristic equation of the SMD.
- Compute the natural frequency ω_n and the damping ratio ζ .
- What are the roots of the characteristic equation?
- You push on the mass to rotate the system to $\pi/2$, as shown in the figure. How much force must you apply to the mass to hold it in this

position?

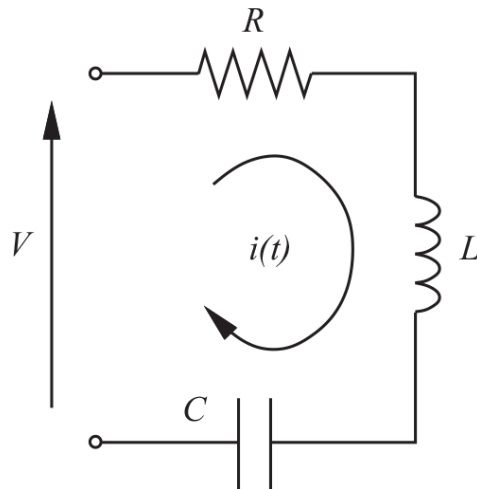
(e) At time zero, the SMD is released from rest $\dot{\theta} = 0$ at position $\theta = \pi/2$.

i. Solve for the time response $\theta(t)$.

ii. Identify the term in the solution that dominates the behavior of the system for large t .

iii. Plot the phase portrait for this system—that is, θ versus $\dot{\theta}$ for $t=0, \infty$.

2. Electrical analog for the SMD. The series combination of a resistor, inductor, and capacitor creates the electrical analog of the mechanical SMD system. Kirchoff's voltage law states that the supply voltage V is equal to the sum of the voltage drops across these three circuit elements.



$$V = V_R + V_L + V_C$$

$$V_R = iR$$

$$V_L = L \frac{di}{dt}$$

$$V_C = \frac{1}{C} \int i dt$$

(a) Derive the differential equation that governs the behavior of the RLC circuit.

(b) Compare the result to [equations 3.1](#) and [3.3](#) and show how RLC parameters correspond to SMD parameters in this pair of harmonic oscillators.

(c) Derive the natural frequency ω_n and the damping ratio ζ in the RLC system.

3. Critical damping.

(a) Find the damping coefficient, B , for critical damping in the second-order system $\ddot{x} + B\dot{x} + 16x = 0$.

(b) Show that when $\zeta = 1$ and the roots of the characteristic equation $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$ are $s_1, s_2 = -\omega_n$, terms in the time-domain solution like $Ae^{-\omega_n t}$ and $Ate^{-\omega_n t}$ both satisfy the original differential equation $\ddot{x} + 2\zeta\omega_n \dot{x} + \omega_n^2 x = 0$.

4. **Natural frequency.** Design a second-order control law, $m\ddot{x} + B\dot{x} + Kx$ with natural frequency, $\omega_n = 50$ [rad/sec]:

(a) For $m=1$ [kg], find K and B for critical damping.

(b) Comment on why it might be useful to design the natural frequency of a controlled system.

5. **Characteristic equation.** Consider the second-order characteristic equation $3s^2 + 24s + 21 = 0$.

(a) What is the natural frequency of the system?

(b) What is the damping ratio?

(c) What is the bandwidth of the system?

6. **Time-domain solutions.** Suppose that the plant described by the following characteristic equations are released from state $x(0) = 1, \dot{x}(0) = 0, x(\infty) = 0$. Derive the time-domain response $x(t)$ of the system.

(a) $\ddot{x} + 5\dot{x} + 6x = 0$

(b) $\ddot{x} + 2\dot{x} + 10x = 0$

7. **Stability: Complex frequency domain.**

(a) **Root locus.** The governing characteristic equation for the SMD system is $s^2 + (B/m)s + (K/m) = 0$. Let $m=1$ [kg] and damping coefficient $B=2$ [N sec/m]. Plot the roots of the characteristic equation on the real-imaginary plane as K varies between $-\infty$ and $+\infty$. Identify the parts of the resulting *root locus* that are unstable, overdamped, critically damped, and overdamped.

(b) **Characteristic equation.** For each characteristic equation, determine if the system is stable and explain your result.

i. $s^2 + s - 2$

ii. $s^2 - s - 6$

iii. $s^2 + 8s + 16$

iv. $s^2 + 3s + 9$

v. $s^2 - 4s + 1$

vi. $s^2 + 9s - 1$

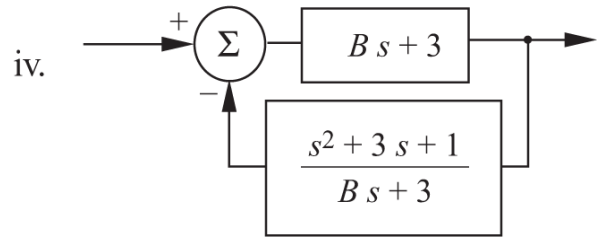
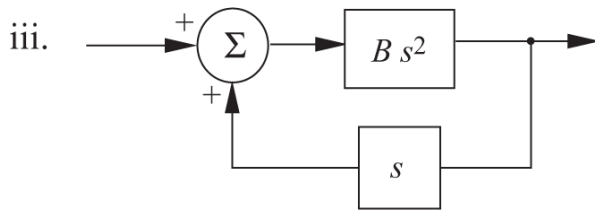
vii. $s^3 + 4s^2 + s - 6$

viii. $s^3 + 3s^2 + 3s + 1$

(c) **Closed-loop transfer function.** For each closed-loop transfer function, determine if the system is stable and explain your result.

i. $\frac{G}{1+GH} = \frac{s^2}{s^2+s-2}$

ii. $\frac{G}{1+GH} = \frac{s^2}{s^2-s-6}$



8. Stability: Lyapunov.

(a) **Comparing frequency- and time-domain analysis.** Consider the system governed by $\ddot{x} = -x - 2\dot{x}$.

i. Determine the roots of the characteristic equation. Is this system stable?

ii. Given a Lyapunov function of the form $V(x, t) = \frac{1}{2}(x^2 + \dot{x}^2)$, what can be concluded about the asymptotic stability of this dynamic system?

(b) **Skew symmetric terms.**

In section 3.2.2, Lyapunov's direct method was applied to the SMD system,

$$\ddot{x} = -(B/m)\dot{x} - (K/m)x,$$

with ambiguous results. The problem arises because the candidate Lyapunov function was defined to be the total energy of the system,

$$V(x, t) = E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}Kx^2.$$

It has been demonstrated that adding a skew symmetric term of the form $\epsilon x\dot{x}$ to the total energy function can solve this problem.

Consider the modified function

$$V(x, t) = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}Kx^2 + \epsilon m\dot{x}x$$

for small positive *epsilon*. Evaluate the Lyapunov conditions for this function to define two constraints on the value of ϵ (in terms of m , K , and B) that guarantee that the SMD system is asymptotically stable in the sense of Lyapunov.

(c) **Nonlinear systems.** Consider the SMD system with nonlinear viscoelastic components defined by

$$F_k = Kx^3$$

$$F_b = \alpha(1 - x^2)\dot{x}.$$

The dissipative force in the damper is a function of the x coordinate of the mass,

$$B(x) = \alpha(1 - x^2), \alpha > 0.$$

i. The equation of motion for this system becomes

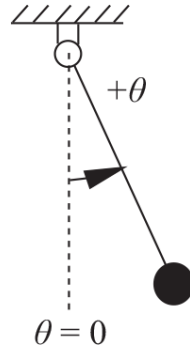
$$M\ddot{x} + Kx^3 + \alpha(1 - x^2)\dot{x} = 0.$$

Write the expression for the total energy in this system.

ii. Using total energy as a candidate Lyapunov function, evaluate the three criteria necessary for establishing asymptotic stability using Lyapunov's second method.

iii. In the phase portrait, plot the region for which this candidate Lyapunov function has properties required of an asymptotically stable dynamical system.

(d) **Pendulum.** An idealized, frictionless pendulum is illustrated in the figure.



The equation of motion for the *ideal* pendulum below is

$$I\ddot{\theta} = -(mg)l\sin(\theta).$$

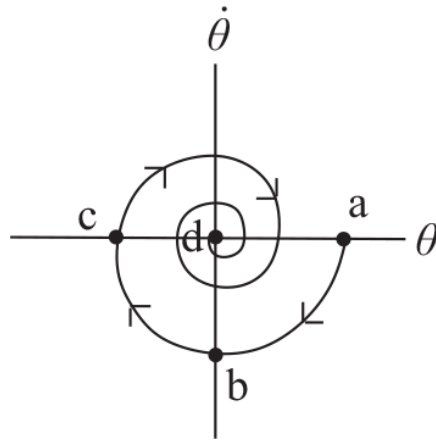
The kinetic energy for this system is

$$KE = \frac{1}{2}I\dot{\theta}^2$$

and the potential energy in the pendulum is

$$PE = mgl(1 - \cos(\theta)).$$

- i. Use the total energy function $KE + PE$ as a candidate Lyapunov function. Is the system stable? Is it asymptotically stable?
- ii. Suppose that the pendulum is released from rest at $\theta = \pi/4$.
 - (A) Label this starting state on the phase plane and sketch the trajectory that this system should execute after it is released.
 - (B) What is the magnitude of the velocity when $\pi = 0$?
- iii. Suppose that this experiment actually performed a trajectory like so:



What term was missing in the dynamic model of this system?

9. **Invent your own homework.** Make a homework problem out of your favorite content in chapter 3. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here. Ideally, they should call for a short discussion and quantitative analysis—it should not take more than 30 [min] to solve open-book.

Part I Summary: Muscles, Motors, and Control

At the most basic level, sensorimotor agents interact with the world by moving parts of their body, and these interactions reflect the character (both strengths and limitations) of the system that transforms energy into movement. This is where our discussion begins. Part I of the book introduces the two most frequently used actuators—muscle and the DC motor. The former dates back to the dawn of multi-celled organisms and is recognizable in a wide variety of contemporary animals, and the latter is the most widely used actuator over the past century for automation and robotics.

Muscle tissue itself is a massive army of biomolecular “robots” with viscoelastic properties derived from the interaction of two proteins—actin and myosin. From this close-up perspective, the mathematics for modeling and analyzing the behavior of muscle is very similar to descriptions of actuators employed by engineers. Huxley’s description of the sliding filament model proposes a means for producing energetic, whole-body movements that depend on the probabilistic activation of trillions of simultaneous actin-myosin interactions.

In the discussion of robotic actuators, our focus centered on the ubiquitous permanent magnet DC motor. We introduced the physics responsible for turning electrical current into force with this device and wrote the equations for the motor dynamics. Motor performance curves for the DC motor are very different from those of muscle tissue and lead, and thus to differences in the quality of movement as well. The conclusion of the discussion about actuation provides a brief survey of other actuator technologies in use today and speculates about emerging actuator technologies that may be available in the future.

Coordinating the behavior of muscle tissue or DC motors depends on closed-loop control to produce movement in response to perceptible errors. An example in the nervous system is the negative feedback response in spinal nerves responsible for the stretch reflex. The discussion of control in part I focuses on spinal processing and the motor nuclei in animal systems. It also provides a mathematical abstraction for motor units based on the

spring-mass-damper (SMD) and a classical treatment of time-domain solutions and stability for such systems.

Part I begs the question—how are integrated movements in animals and in robots organized around the behavior of multiple, stable, closed-loop systems? This structure simplifies the enormously complex task of learning and/or planning such movements. It also begins an analysis of how physical properties of an agent determine what it can and cannot do and, therefore, what it can and cannot experience. This theme carries through part II, where we examine how the embodied control system influences the quality and representation of environmental interactions.

1. The presentation in this chapter emphasizes the intuition underlying the analysis. Interested readers can find complete derivation of these results in appendix A.10.3.

2. For the sake of this definition, we define the zeroth-order derivative $q_0 = \left. \frac{d^0 f(t)}{dt^0} \right|_{\tau} = f(\tau)$.

3. In this case, a related technique for discrete-time system analysis using the z-transform can be used.

4. The more general definition of the transform for derivatives is presented in appendix A.10.3,

$$\mathcal{L}\left[\frac{d^n}{dt^n}f(t)\right] = s^n F(s) - \sum_{k=1}^n s^{k-1} f^{(n-k)}(0),$$

where $f^{(n-k)}(0)$ is a boundary condition representing the value of the $(n-k)$ th derivative of function $f(t)$ at $t = 0$.

5. Appendix A.11.1 derives the equations that describe the system behavior as a function of time for this reference input.

II

STRUCTURE IN KINODYNAMIC SYSTEMS

4

Kinematic Systems

Kinematics is a subfield of dynamics in which the geometrical aspects of motion are considered—apart from issues of mass and acceleration. The kinematic influence of the skeleton on the movement of an animal is fundamental and inescapable. Far from being a curse, however, exploiting the kinematic aptitudes of a biomechanical system leads to higher performance: it can produce an economy and grace of movement, enhance sensory and motor acuity, and guide the search for skillful motor strategies. The same kinematic principles can be used to guide robot programming and to exploit the aptitudes of robot devices.

This chapter introduces tools for analyzing movement in kinematic systems. All machines introduce sweet spots where attractive kinematic properties can be exploited. We begin with a discussion of mechanisms [220], after which the kinematic performance of articulated machines is examined in some detail, including forward and inverse mappings between body configurations and Cartesian task descriptions. The governing kinematic equations for hand-eye coordination are presented, and the chapter concludes by reviewing measures of *kinematic conditioning*—descriptions of the ability of mechanisms to project velocity and force efficiently into Cartesian space.

4.1 Terminology

Engineers have been studying principles of machine design for a very long time. At the most basic level, animals and robots are mechanisms whose

behavior is governed by the same principles.

A *mechanism* is a collection of bearings, gears, cams, links, and belts that convert motions from one form to another and transmit power. It is an assemblage of individual rigid bodies known as *links* that are connected in pairs through kinematic constraints called *joints*. A joint permits a specific relative motion between adjacent links—*prismatic* joints permit translations, and *revolute* joints support rotations. Any parameter (length or angle) of a mechanism that specifies the position of an independent joint is called a *configuration variable*. A *kinematic chain* is an assemblage of interconnected links whose kinematic properties depend on how links and joints are combined. The posture of devices with multiple configuration variables is a coordinate in the *configuration space*. The minimum number of such variables necessary to fully define the configuration of a mechanism is called the *degree of freedom* (DOF) of the system.

A mechanism is formed when one of the links is held fixed and the others move relative to the fixed link. The fixed link is referred to as the *ground link* or *ground frame*. Most mechanisms considered in the machine design literature consist of *closed chains*—that is, kinematic chains with every link connected through joints to two adjacent links. However, many robotic devices are *open chains*, wherein one or more so-called *unitary links* are connected to only one joint.

4.2 Spatial Tasks

Tasks specify (directly or indirectly) the spatial configuration of limbs for locomotion and/or manipulation, the positioning of sensory devices, and the relative geometry of external objects in the environment. To meet task specifications, control inputs are used to modify the spatial configuration of the mechanisms comprising the robot.

In general, several frames of reference may be necessary to fully specify a robot control task. Kinematic relations such as those in [figure 4.1](#) are naturally arranged in tree structures rooted in a fixed *world* frame. The distribution of objects is often described in terms of their position and orientation in the world frame or task frames fixed in the world to avoid accumulating uncertainty. For example, in [figure 4.1](#), the *station* coordinate system is such a frame. It describes aspects of the task related to the world

through fixed spatial offsets. From the station frame, several other features of the task (*obj1*, *obj2*, *goal*) may be specified as fixed offsets or determined using sensor systems.

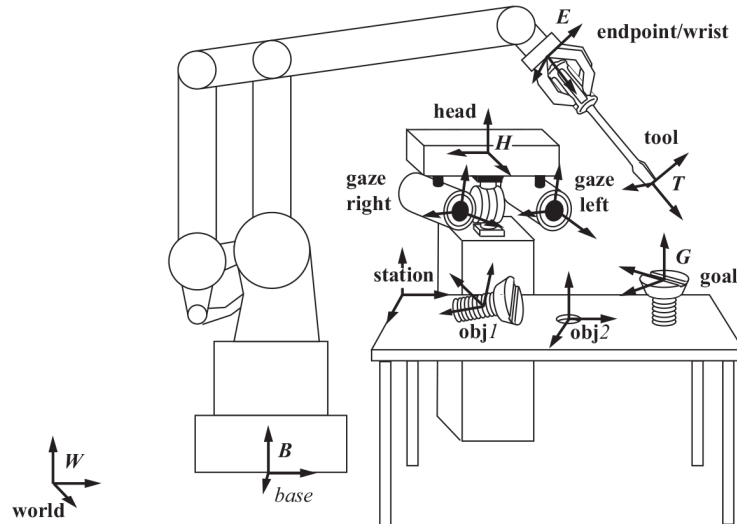


Figure 4.1

Coordinate frames commonly used to specify robot tasks

The sensory and motor resources of the robot can also be located in the world frame using spatial transformations. For example, the transform from world to *base* locates a coordinate frame attached to the robot. It could be a fixed position relative to the world frame, as in [figure 4.1](#), or it could be time varying and controllable, as in the case of a mobile manipulator. In the latter case, it is established by integrating displacements¹ in the mobile base in a process called *odometry* or by relation to other visual or tactile features in the room whose positions are known.

From the base frame, two important frames establish the spatial distribution of sensory and motor resources that can be applied to the task. The first locates the *wrist* frame, and the second identifies the pose of the sensor *head* relative to the base. These frames reflect both fixed offsets in the body and the posture of controllable degrees of freedom measured using sensors in every joint. Together with knowledge of the hand posture, they constitute multiple *forward kinematic* relations in the robot that depend of configuration variables and, therefore, support reasoning about spatial errors and control. The *tool* frame in [figure 4.1](#) identifies the working end of

the grasped screwdriver. It is estimated by considering visual and tactile feedback in the context, perhaps, of prior information about the geometry of screwdrivers.

Given a complete specification of the kinematic tree described above, control tasks can be expressed in terms of the coordinate frames in [figure 4.1](#). In principle, then, movements in the actuated degrees of freedom can be computed in order to bring the tool and goal frames together and turn the screw. However, spatial estimates are not all equally precise. The geometry of the arm, fingers, and head depends on the dimensions of the mechanisms and measurements of joint angles, which are relatively precise. In contrast, fingertip contact locations, visual feedback, and estimates of the grasped pose of the screwdriver can be significantly less precise.

Example: Kinematic Description of Roger-the-Crab

Roger-the-Crab was originally developed by Paul Churchland in 1988 to demonstrate how the Cartesian space around a simple organism can be encoded in neural projections between sensory and motor maps [53]. The original Roger was a single, planar, two degree of freedom arm and a stereovision system. [Figure 4.2](#) defines the kinematic parameters and intermediate coordinate frames of a more elaborate mobile, bimanual version of Roger. The robot lives in the x - y plane and has a nine-dimensional configuration space consisting of two pan degrees of freedom in the eyes (θ_1, θ_2), two degrees of freedom for the left arm (θ_3, θ_4) and the right arm (θ_5, θ_6), and three degrees of freedom in the mobile base (x, y, θ_0). Each revolute joint (except for the wheels) rotates about the world frame \hat{z} axis. Roger's wheel and arm joints rotate continuously, and his eyes have a range of motion limited to $-\pi/2 \leq \theta_1, \theta_2 \leq \pi/2$.

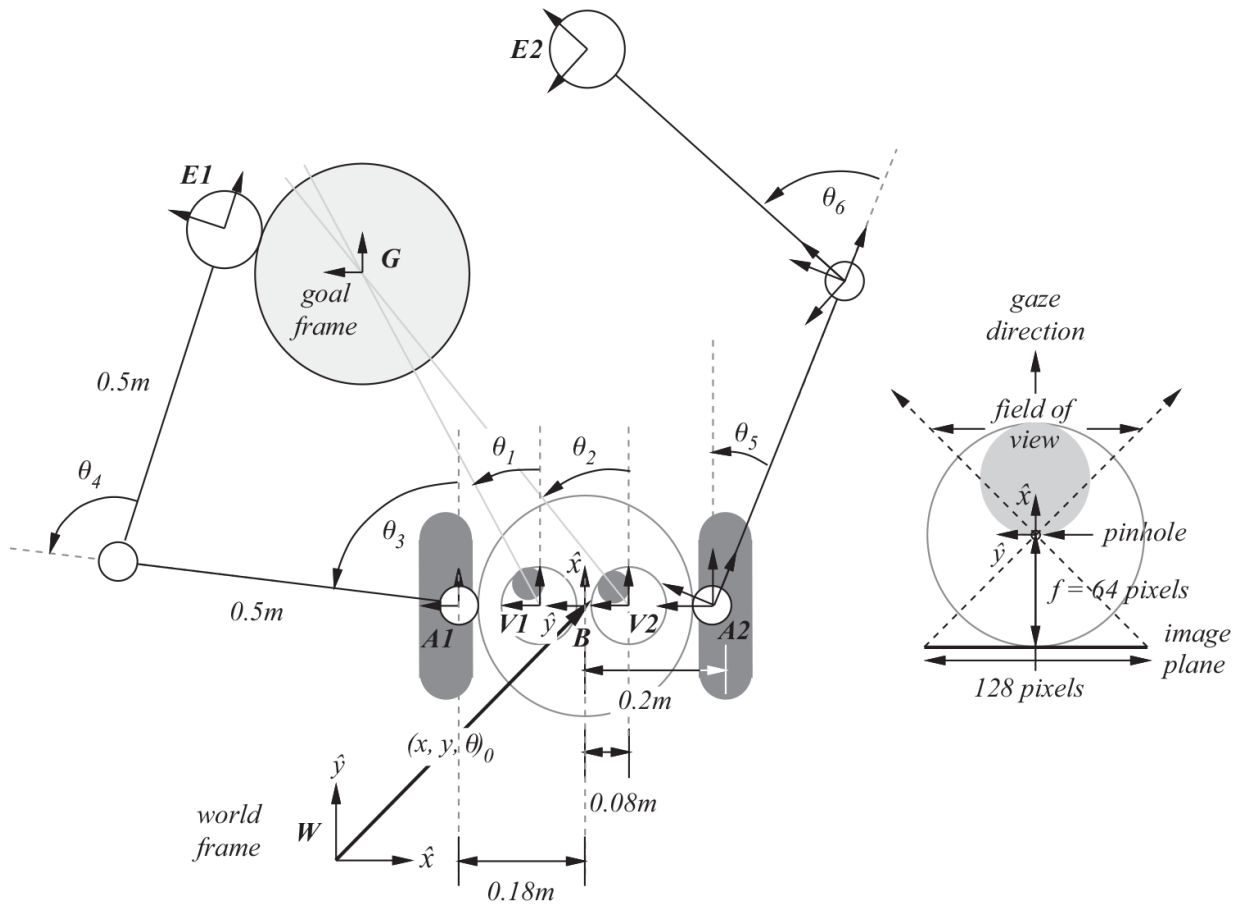


Figure 4.2

The kinematic definitions and intermediate coordinate frames used to define Roger

Roger consists of several kinematic chains that establish spatial quantities marked by intermediate coordinate frames. Four important coordinate frames are rooted at fixed locations relative to the base frame \mathbf{B} . Frames $\mathbf{V1}$ and $\mathbf{V2}$ establish the positions of Roger's eyes relative to \mathbf{B} , and frames $\mathbf{A1}$ and $\mathbf{A2}$ establish the positions of the shoulders of Roger's arms relative to \mathbf{B} . In Roger, these frames are rigid translations from frame \mathbf{B} in the \hat{y}_B direction and are used to describe and control the geometry of sensors and effectors deployed by the robot.

Information about the goal frame \mathbf{G} is derived from visual and tactile sensors. Cameras at frames $\mathbf{V1}$ and $\mathbf{V2}$ can pan independently using actuated joints θ_1 and θ_2 . The transform from frames $\mathbf{V1}$ and $\mathbf{V2}$ to frames aligned with the gaze (right inset) is a pure rotation parameterized by the configuration variables. The eyes are pinhole RGB cameras (section 4.5.1) with a focal length of 64 [pixel]² that produce a one-dimensional image 128

pixels wide. Each eye, therefore, has a field of view of ± 45 [deg] directed into the world by motor actions that pan the eyes and/or the body. Signals from the camera pair can be used to estimate the goal location by triangulation.

Tactile sensors are located at endpoint frames **E1** and **E2**. The transform from arm frames to the *endpoint* frames depends on the geometry of links and actuated joints ($\theta_3, \theta_4, \theta_5, \theta_6$) in the arms. Tactile observations take the form of force vectors in the x - y plane written in the end effector coordinate frame. Like vision, signals from the tactile sensor together with knowledge of the configuration of the robot can be used to determine the Cartesian location of the goal frame **G** via forward kinematic relations (section 4.4.1).

The task of positioning the endpoint frames and directing the gaze of the eyes is completely controllable using commands to motors that actuate each of the degrees of freedom in these kinematic chains. Such completely controllable kinematic systems are *holonomic*.³ The pose of the mobile base is established by defining the location and orientation of the base coordinate frame using odometry to estimate $(x, y, \theta)_0$ in *world* coordinates. Navigation tasks for the wheeled base are expressed in this three-dimensional pose space as well. However, unlike the other kinematic systems comprising Roger, the mobile base is *nonholonomic*—the wheels do not permit translational velocities along \hat{y}_B . In *world* coordinates, from any given pose $\mathbf{q} = [x \ y \ \theta]^T$, the wheeled system can generate rotational velocities $\dot{\mathbf{q}}_\theta = [0 \ 0 \ 1]^T$ and translational velocities along the current heading $\dot{\mathbf{q}}_{xy} = [\cos(\theta) \ \sin(\theta) \ 0]^T$. Therefore, the controllable velocity from any given pose is a two-dimensional subset of the pose velocity space and is expressed as a linear combination of a rotational velocity and a pose-dependent translational velocity

$$\dot{\mathbf{q}} = v_\theta \dot{\mathbf{q}}_\theta + v_{xy} \dot{\mathbf{q}}_{xy},$$

where v_θ and v_{xy} are scalar magnitudes for rotational and translational velocities, respectively. The difference between the number of degrees of freedom in the task and those in the control is stated succinctly in the form of a single nonholonomic kinematic constraint that restricts translational velocities

$$f(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{l}^T \dot{\mathbf{q}} = [\sin(\theta) \ -\cos(\theta) \ 0] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0,$$

where \mathbf{l} is the vector in the x - y plane that is orthogonal to the current vehicle heading (the lateral direction). Now, the navigation task becomes more challenging—the absence of a “whole kinematic law” introduces interesting control and path planning issues. Other examples exist of important nonholonomic systems—for instance, when fingertips roll over the surface of a grasped object during manipulation.

□

To construct control tasks, we require a means of transforming (sets of) Cartesian quantities expressed in one coordinate frame into another frame appropriate for control. In the next section, we introduce one such representation called the homogeneous transform.

4.3 Homogeneous Transforms

A *group* is a set of elements that is closed under a binary operation—that is, applying the operator to pairs of elements in the set produces a result that is also an element of the set. In addition, to constitute a group the operator must be associative—that is, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$. The group must include an identity element I , such that $a \cdot I = I \cdot a = a$, and every element a in the set must have an inverse a^{-1} , such that $a \cdot a^{-1} = I$, that is also in the set. The classical example of a group is the set of integers and the addition operator.

Rigid body motions describe Euclidean groups. In particular, a three-dimensional translation group T consists of elements that are displacements $\mathbf{t} \in \mathbb{R}^3$ ($[x \ y \ z]^T$, for example). The translation group is closed under addition. Moreover, the elements of the *special orthogonal* group in three dimensions $SO(3)$ designates three orthogonal rotations (roll, pitch, yaw, for example). The rotation group is closed under multiplication, and compositions of elements preserve Euclidean distances. Groups T and $SO(3)$ are independent subgroups of the *special Euclidean* group $SE(3)$, elements of which describe how rigid bodies move in a six-dimensional Cartesian space. As a consequence, to completely define the pose of a rigid body, one

needs to specify a minimum of six independent variables that specify the translation and rotation of the body.

4.3.1 Translational Components

Figure 4.3a shows two coordinate frames, A and B , that are related through a pure translation. A position vector r_B written in frame B is expressed in coordinate frame A using the linear transformation

$$r_A = r_B + {}^A t_B,$$

where ${}^A t_B$ represents the translation from frame A to frame B written in frame A coordinates. During a pure translation, the relative orientation of the two frames remains fixed and the \hat{x} , \hat{y} , and \hat{z} coordinate axes remain parallel.

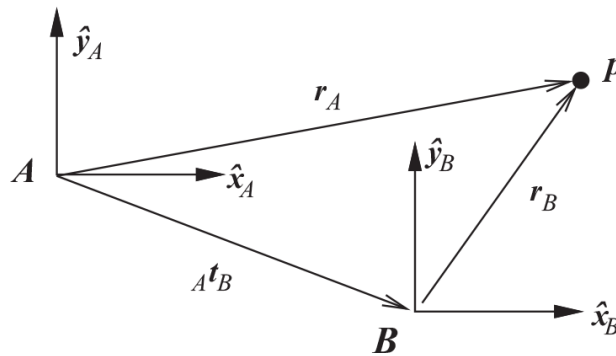


Figure 4.3

Two coordinate frames related through a pure translation

4.3.2 Rotational Components

There are several alternatives for representing rotations [286], including exponential coordinates [202], Euler angles, roll-pitch-yaw notations, quaternions, and direction cosines. Of these choices, the direction cosines are the least compact—requiring nine numbers to specify three rotation variables—but, as we will see, direction cosines contribute to a simple, invertible, and homogeneous representation for $SE(3)$.

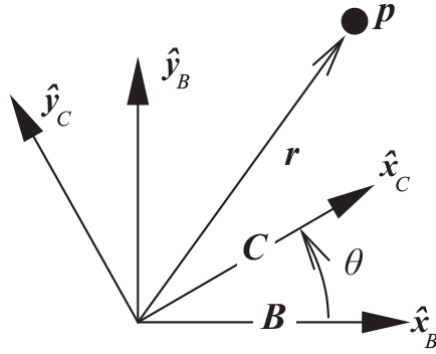


Figure 4.4
Two coordinate frames related through a pure rotation

To derive the direction cosine matrix, consider frames B and C in [figure 4.4](#), related through a pure rotation. Position vector r written in frame C coordinates can be expressed in frame B coordinates by employing the 3×3 direction cosine matrix ${}_B\mathbf{R}_C$, such that

$$r_B = {}_B\mathbf{R}_C r_C, \text{ and}$$

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}_B = \begin{bmatrix} \hat{x}_B \cdot \hat{x}_C & \hat{x}_B \cdot \hat{y}_C & \hat{x}_B \cdot \hat{z}_C \\ \hat{y}_B \cdot \hat{x}_C & \hat{y}_B \cdot \hat{y}_C & \hat{y}_B \cdot \hat{z}_C \\ \hat{z}_B \cdot \hat{x}_C & \hat{z}_B \cdot \hat{y}_C & \hat{z}_B \cdot \hat{z}_C \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}_C, \quad (4.1)$$

where \hat{x} , \hat{y} , and \hat{z} represent the orthonormal basis for a coordinate frame.

The dot operator in [equation 4.1](#) is the dot (or scalar) product ([appendix A.1](#)). Columnwise, the direction cosine matrix maps the orthonormal (orthogonal and unit length) basis vectors in frame C into a new orthonormal basis written in frame B . The first column, for example, projects basis vector \hat{x}_C onto the \hat{x} , \hat{y} , and \hat{z} axes of frame B . The matrix multiply in [equation 4.1](#) sums the independent projections of the x , y , and z components of r_C onto the basis vectors for frame B .

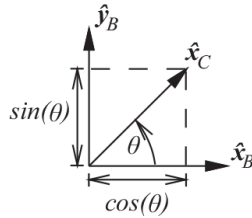


Figure 4.5
Projecting \hat{x}_C onto the x - y plane of frame B

The rotation matrix ${}_B\mathbf{R}_C$ can be derived geometrically by writing basis vectors $(\hat{x}, \hat{y}, \hat{z})_C$ in frame \mathbf{B} coordinates and then entering the result into the appropriate columns of ${}_B\mathbf{R}_C$. Consider the rotation illustrated in [figure 4.5](#) and let \hat{x}_C^B be the projection of \hat{x}_C onto coordinate frame \mathbf{B} .

$$\hat{x}_C^B = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix}$$

Repeating this geometric construction for \hat{y}_C^B and \hat{z}_C^B yields

$${}_B\mathbf{R}_C = \begin{bmatrix} \hat{x}_C^B & \hat{y}_C^B & \hat{z}_C^B \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.2)$$

The columns of ${}_B\mathbf{R}_C$ represent the basis vectors of frame \mathbf{C} projected onto the basis vectors of frame \mathbf{B} . Conversely, the rows of ${}_B\mathbf{R}_C$ represent the projection of the basis vectors for frame \mathbf{B} onto the basis vectors of frame \mathbf{C} . This suggests, and it is clear by inspection of [equation 4.1](#), that the inverse of the direction cosine matrix is its transpose, ${}_B\mathbf{R}_C^{-1} = {}_B\mathbf{R}_C^T = {}_C\mathbf{R}_B$.

Finally, note that the rotation in [figure 4.3b](#) is about the \hat{z} axis and, therefore, preserves the direction of the \hat{z} axis. The positive sense of the rotation indicates the rotation from frame \mathbf{B} to frame \mathbf{C} .

$${}_B\mathbf{R}_C = \text{rot}(\hat{z}, \theta)$$

Rotations about \hat{x} and \hat{y} are constructed in the same way. For the sake of completeness, the direction cosine matrix for pure rotations about the \hat{x} , \hat{y} , and \hat{z} axes are summarized here,

$$\begin{aligned} \text{rot}(\hat{x}, \theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} & \text{rot}(\hat{y}, \theta) &= \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \\ \text{rot}(\hat{z}, \theta) &= \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (4.3)$$

where $c\theta$ and $s\theta$ are shorthand notation for $\cos(\theta)$ and $\sin(\theta)$, respectively.

The *homogeneous transform* incorporates translation and rotation in a single linear transformation. It applies a translation from frame **A** to frame **B** followed by a rotation from frame **B** to frame **C**, as in [figure 4.6](#). The order of application of the transforms is important—spatial transforms such as these are not commutative.

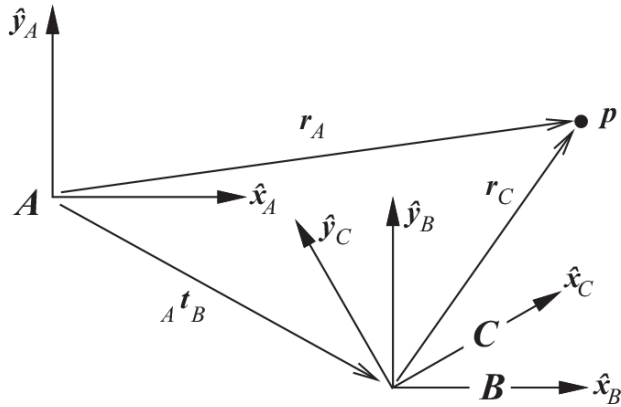


Figure 4.6
Two coordinate frames related through a rotation and a translation

The homogeneous transform ${}^A\mathbf{T}_C$ and the homogeneous position vector are defined as follows:

$${}^A\mathbf{T}_C = \left[\begin{array}{cc|c} & & \\ & {}^B\mathbf{R}_C & {}^A\mathbf{t}_B \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad \mathbf{r}_C = \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix}_C .$$

The fourth row in each of these definitions is there to provide the bookkeeping required to sum the affects of the translation and rotation correctly. It is left to the reader to verify that the product $\mathbf{r}_A = {}^A\mathbf{T}_C \mathbf{r}_C$ yields the intended homogeneous position vector for the construction in [figure 4.6](#).

4.3.3 Inverting the Homogeneous Transform

The inverse of the homogeneous transform is simple to derive. Here, we simply present the solution and leave it to the reader to verify.

$${}^A\mathbf{T}_B = \left[\begin{array}{ccc|c} \hat{\mathbf{x}}_B^A & \hat{\mathbf{y}}_B^A & \hat{\mathbf{z}}_B^A & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{array} \right] \quad {}^B\mathbf{T}_A = [{}^A\mathbf{T}_B]^{-1} = \left[\begin{array}{ccc|c} (\hat{\mathbf{x}}_B^A)^T & -(\hat{\mathbf{x}}_B^A)^T \mathbf{t} & & \\ (\hat{\mathbf{y}}_B^A)^T & -(\hat{\mathbf{y}}_B^A)^T \mathbf{t} & & \\ (\hat{\mathbf{z}}_B^A)^T & -(\hat{\mathbf{z}}_B^A)^T \mathbf{t} & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

The homogeneous transform is commonly used in robotics and computer graphics and is one way of representing the kinematic transformations between the coordinate frames indicated in task descriptions like [figures 4.1](#) and [4.2](#). It provides a closed-form method for describing spatial transformations associated with articulated mechanisms and is easily composed and inverted. Moreover, the homogeneous transform is the basis for the standardized *Denavit-Hartenberg* (D-H) notation that makes it possible to represent a complete kinematic specification for a robot using tables of standardized D-H parameters [221].

4.4 Manipulator Kinematics

Forward and inverse kinematic relations describe reciprocal mappings between an n -dimensional configuration space ($\mathbf{q} \in \mathbb{R}^n$) describing the degrees of freedom in the kinematic system and the Euclidean space ($\mathbf{r} \in SE(3)$) consisting of the complete pose of a Cartesian frame (typically the endpoint of the manipulator). The dimensionality of vector \mathbf{r} depends on the representation used for orientations. Section 4.3 introduced the 4×4 homogeneous transform to represent $SE(3)$ so, in this case, $\mathbf{r} \in \mathbb{R}^{16}$ —not the most compact representation. In the following, we will often consider only positions in Cartesian space—that is, $\mathbf{r} \in \mathbb{R}^3$.

Forward kinematic transformations map subsets of configuration space to subsets of Cartesian space, $\mathbf{q} \mapsto \mathbf{r}$, and inverse kinematic transformations map them back again, $\mathbf{r} \mapsto \mathbf{q}$. The character of these mapping functions reveals a great deal about the aptitudes and limitations of robotic mechanisms.

4.4.1 Forward Kinematics

Closure and associativity properties of $SE(3)$ imply that homogeneous transforms can be used to compose other homogeneous transforms. This is

of great consequence when describing tasks, especially when this description incorporates the configuration variables of a robot.

Consider the forward kinematics of Roger-the-Crab's two degree of freedom robot arm (figure 4.2). This manipulator geometry consists of a series of two revolute joints with parallel axes of rotation so that the manipulator remains in the plane normal to this axis of rotation. For this reason, it is often called the planar 2R manipulator.

Example: Forward Kinematics of the Planar 2R Manipulator

Figure 4.7 illustrates Roger's arm. It includes several intermediate coordinate frames distributed along the length of the arm. Frame 0 is the arm frame that locates the shoulder joint with respect to the body frame in figure 4.2. Frames 1, 2, and 3 represent cumulative displacements from frame 0. Frame 1 is rotated by θ_1 so that \hat{x}_1 remains directed along link 1 as the shoulder joint moves.

$${}^0\mathbf{T}_1 = \text{rot}(\hat{z}_0, \theta_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

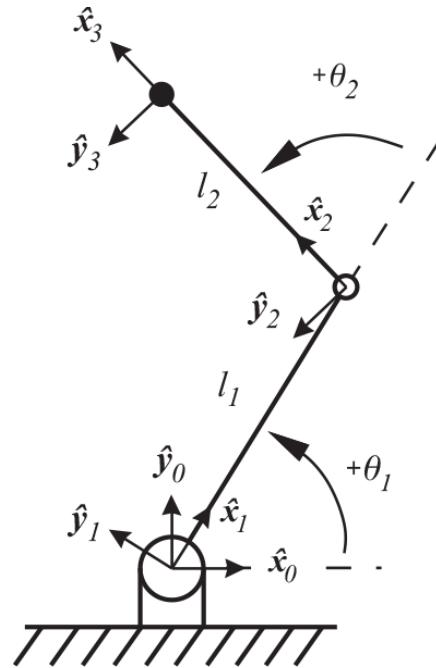


Figure 4.7
The planar 2R manipulator

Relative to frame 1, a translation along \hat{x}_1 for a distance l_1 followed by a rotation of θ_2 about \hat{z}_1 results in frame 2, with \hat{x}_2 directed along link 2. Finally, relative to frame 2, a translation along \hat{x}_2 of distance l_2 locates the endpoint frame 3.

$${}^1\mathbf{T}_2 = \text{trans}(\hat{x}_1, l_1) \text{rot}(\hat{z}_1, \theta_2) = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2\mathbf{T}_3 = \text{trans}(\hat{x}_2, l_2) = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The net transformation from frame 0 to frame 3 is the product of this sequence of homogeneous transformations.

$$\begin{aligned}
{}^0\mathbf{T}_3 &= {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.4}
\end{aligned}$$

where $c_i = \cos(\theta_i)$; $s_i = \sin(\theta_i)$; and $c_{12} = \cos(\theta_1 + \theta_2)$. Identities $\sin(\alpha \pm \beta) = \sin\alpha \cos\beta \pm \cos\alpha \sin\beta$ and $\cos(\alpha \pm \beta) = \cos\alpha \cos\beta \mp \sin\alpha \sin\beta$ are used to simplify the result. □

The forward kinematic relation in the form of the homogeneous transform ${}^0\mathbf{T}_3$ provides insight into the kinematic character of the planar 2R manipulator. For example, inspecting the rotational part of the transform, we determine that the orientation of the end effector relative to frame 0 is a rotation about the \hat{z} axis of $(\theta_1 + \theta_2)$. Moreover, the position of the end effector is determined in the fourth column of the transform,

$$\begin{aligned}
x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\
y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\
z &= 0.
\end{aligned} \tag{4.5}$$

These equations can be evaluated over all reachable joint angles to yield the *reachable workspace* for the manipulator. [Figure 4.8](#) illustrates the result for the planar 2R manipulator assuming that each joint is free to rotate continuously (light gray). Also illustrated in darker gray is the reachable workspace for the human shoulder-elbow analog of the 2R manipulator with realistic joint range limits ($-\pi/4 \leq \theta_1 \leq 5\pi/4$, $0 \leq \theta_2 \leq \pi$). Three kinematic variations of Roger's arm are sketched in the figure. The leftmost panel shows Roger's arm where $l_1 = l_2$. Under these conditions, Roger can reach the entire interior (including the perimeter) of the disk with radius $l_1 + l_2$ centered on the shoulder joint. Absent joint range limitations, each position in the interior of the reachable workspace can be achieved using

two configurations of the arm—one with the elbow flexed and one with the elbow (hyper)extended. The adult human arm conforms fairly closely to the $l_1 = l_2$ case.

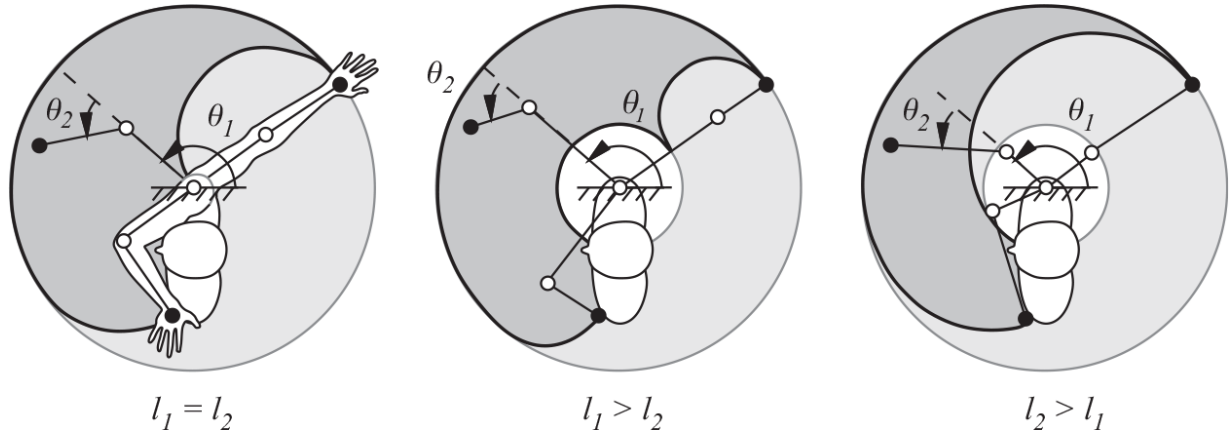


Figure 4.8

The reachable workspace for the planar 2R robot

The *dexterous workspace* is the subset of the reachable workspace where all orientations of the end effector can be achieved. This subset is non-empty only when $l_1 = l_2$ (the left panel). It includes only a single Cartesian location (the origin), corresponding to a one-dimensional subset of configurations where $\theta_2 = \pm \pi$ and $-\pi \leq \theta_1 \leq \pi$. Given this elbow posture, the endpoint orientation can assume any value in the plane by controlling the position of shoulder in the range $-\pi$ to π .

Realistic joint range limits interact with relative link lengths to significantly influence the reachable workspace and, as the center and right panels of [figure 4.8](#) illustrate, there are significant advantages to keeping l_1 and l_2 approximately equal in length.

4.4.2 Inverse Kinematics

Tasks are efficiently expressed in terms of Cartesian goals $r \in SE(3)$ that are transformed into configuration variables $q \in \mathbb{R}^n$ for use in motor controllers. The mapping required is the inverse of the kinematic mapping discussed in section 4.4.1. However, forward kinematic functions are nonlinear—many Cartesian goals are unreachable, and others can be realized by many

possible joint angle configurations. As result, inverting forward kinematic mappings is nontrivial.

Complete inverse kinematic solutions yield all joint angle configurations for any reachable Cartesian goal. In general, complete inverse kinematic solutions are very difficult (or impossible) to compute in closed form; however, they do exist for some important special cases. For example, Pieper [229] formulated a general inverse kinematic solution for six degree of freedom manipulators consisting of three revolute or prismatic joints followed by three consecutive joints with rotational axes that intersect at a point. The first three degrees of freedom generate positions in \mathbb{R}^3 and the spherical wrist generates $SO(3)$. Several robot designs incorporate this kinematic simplification, a notable example of which was the Unimate PUMA 560 (figure 4.9a), the first commercially available arm. The human arm-wrist geometry in figure 4.9b embodies these kinematic properties as well. Three proximal degrees of freedom in the shoulder and elbow position the wrist in \mathbb{R}^3 . Three orthogonal rotational axes (humeral rotation, wrist flexion/extension, and wrist adduction/abduction) intersect at a point in the wrist near the base of the palm and, thus, generate $SO(3)$. Therefore, the ability to position and orient the hand are decoupled to a large degree.

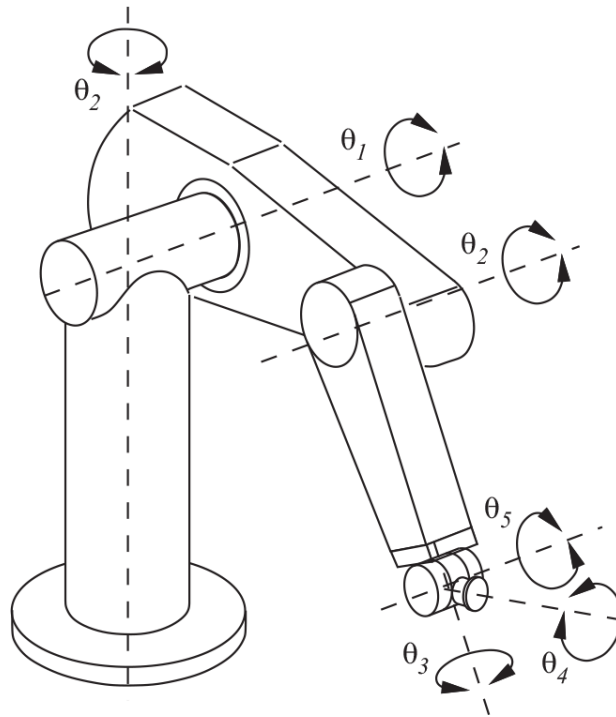


Figure 4.9
The Unimate PUMA 560

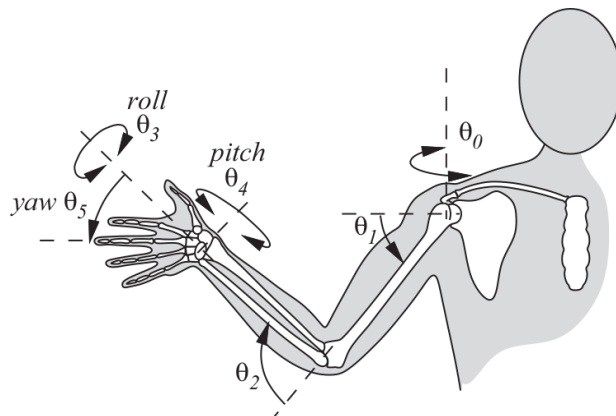


Figure 4.10
The human shoulder, arm, and wrist. Three intersecting revolute axes in the wrist create an approximately spherical wrist joint important for brachiation in human ancestors

Most closed-form solutions exploit structure in the kinematic design and employ algebraic or geometric methods [285]. Of these approaches, geometric methods attempt to identify opportunities to decompose the inverse kinematic problem for a particular manipulator into several lower-dimensional problems. The properties underlying Pieper's conditions are an

example of such an approach, whereby the problem is decomposed into inverse position and inverse orientation components. Instead of a comprehensive treatment of these techniques, we will pose a particularly simple example in the context of inverse kinematic solutions applicable to Roger’s two degree of freedom planar arm that first appeared in [65].

Example: Geometric Inverse Kinematic Solution for the Planar 2R Manipulator

Kinematic parameters for Roger’s planar, 2R arm are defined in figures 4.2 and 4.7. We note that for Roger, $l_1 = l_2$. In this example, we construct a complete inverse kinematic solution for this limb. Recall that by “complete” in this case, we mean that every point in the interior of the reachable workspace should yield two discrete solutions, one each for positive and negative values of the elbow joint. The exception is the origin $(0, 0)$ —the dexterous subset of the reachable workspace—where there exists an infinite number of inverse kinematic solutions.

Figure 4.11 defines a set of auxiliary variables that will simplify the derivation of the inverse kinematics—the mapping from a Cartesian target (x, y) to the corresponding values for the joint angles (θ_1, θ_2) .

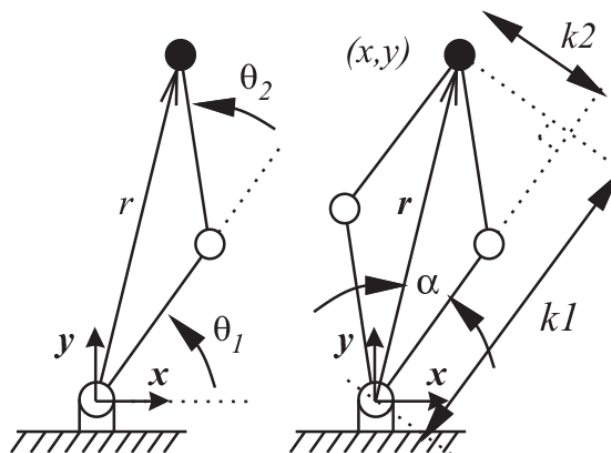


Figure 4.11

A geometric construction for simplifying the inverse kinematic solution

We can write the square of the magnitude of vector r in terms of the (x, y) coordinate of the endpoint for the 2R manipulator $r^2 = x^2 + y^2$. Introducing

the forward kinematic relations for this manipulator (equation 4.5) yields

$$\begin{aligned} r^2 &= x^2 + y^2 \\ &= l_1^2 c_1^2 + 2l_1 l_2 c_1 c_{12} + l_2^2 c_{12}^2 + l_1^2 s_1^2 + 2l_1 l_2 s_1 s_{12} + l_2^2 s_{12}^2, \end{aligned}$$

and after simplifying,

$$r^2 = l_1^2 + 2l_1 l_2 c_2 + l_2^2.$$

Rearranging terms and solving for c_2 yields

$$c_2 = \frac{r^2 - l_1^2 - l_2^2}{2l_1 l_2}. \quad (4.6)$$

The quotient on the right side of equation 4.6 determines the geometric feasibility of target \mathbf{r} for the 2R mechanism with link lengths l_1 and l_2 . For an inverse kinematic solution to exist, equation 4.6 must produce a value for c_2 in the interval $[-1, +1]$ —when this is true, \mathbf{r} is reachable.

To solve for both θ_2 solutions, we note that $s_2^2 + c_2^2 = 1$, so that $s_2^{+/-} = +/- (1 - c_2^2)^{1/2}$, and

$$\theta_2^{+/-} = \tan^{-1} \frac{s_2^{+/-}}{c_2}. \quad (4.7)$$

The pair of θ_1 values that correspond to these θ_2 solutions are determined by using auxiliary variables k_1 and k_2 (figure 4.11),

$$k_1 = r c_\alpha = l_1 + l_2 c_2, \text{ and} \quad (4.8)$$

$$k_2^{+/-} = r s_\alpha = l_2 s_2^{+/-}, \quad (4.9)$$

so that

$$\alpha^{+/-} = \tan^{-1} \frac{k_2^{+/-}}{k_1}. \quad (4.10)$$

Finally,

$$x = k_1 c_1 - k_2 s_1 = (rc_\alpha)c_1 - (rs_\alpha)s_1 = r\cos(\alpha + \theta_1)$$

$$y = k_1 s_1 + k_2 c_1 = (rc_\alpha)s_1 + (rs_\alpha)c_1 = r\sin(\alpha + \theta_1)$$

and

$$\tan(\alpha + \theta_1) = \frac{r\sin(\alpha + \theta_1)}{r\cos(\alpha + \theta_1)} = \frac{y}{x}$$

so that

$$\theta_1^{+/-} = \tan^{-1} \frac{y}{x} - \alpha^{+/-}. \quad (4.11)$$

These kinematic relations are summarized in the pseudocode presented in [figure 4.12](#) [65].

GIVEN (x,y) endpoint position goal:

$$\begin{aligned} r^2 &= x^2 + y^2 \\ c_2 &= (r^2 - l_1^2 - l_2^2) / (2l_1 l_2) \\ \text{if } (-1 \leq c_2 \leq +1) \\ & \quad s_2^{+/-} = +/ - (1 - c_2^2)^{1/2} \\ & \quad \theta_2^{+/-} = \tan^{-1} (s_2^{+/-} / c_2) \\ & \quad k_1 = l_1 + l_2 c_2 \\ & \quad k_2^{+/-} = l_2 s_2^{+/-} \\ & \quad \alpha^{+/-} = \tan^{-1} (k_2^{+/-} / k_1) \\ & \quad \theta_1^{+/-} = \tan^{-1} (y/x) - \alpha^{+/-} \\ \text{else "out of reach"} \end{aligned}$$

Figure 4.12

The complete inverse kinematic mapping for the planar 2R manipulator

□

Forward and inverse kinematic transformations contribute to a *proprioceptive* sense of space. Proprioception refers to the sense of the position and movement of one's body—including the orientation and

viewpoint of externally directed sensors. Combined with other sensor modalities, like tactile sensing and vision, proprioceptive information can contribute to the acquisition of *exteroceptive* information regarding the space outside the robot's body. One of the most important sources of this kind of information is derived from stereoscopic vision.

4.5 Kinematics of Stereo Reconstruction

Stereo vision is largely a kinematic issue. In section 4.4, forward kinematic functions were used to map the configuration variables of a robot manipulator into Cartesian coordinates. This mapping function can be used to map tactile events on the surface of the manipulator to the Cartesian places where contacts occur. Forward kinematic relations can also be used to localize other sensory events. For example, oculomotor configuration variables corresponding to binocular visual events can be mapped to the Cartesian location of a light source. In this section, we will introduce a simple model of the imaging geometry and the kinematic equations underlying stereoscopic reconstruction.

4.5.1 Pinhole Camera: Projective Geometry

Figure 4.13 illustrates a pinhole camera, constructed by making a single small hole in an opaque enclosure. The pinhole is positioned at the origin of the camera's coordinate frame in the diagram and provides the only pathway for electromagnetic energy to enter the enclosure. Rays of light travel in straight line paths through the opening to strike the image plane at $x=-f$, where f is called the focal length. The process creates an image of environmental light sources. Such a *projective geometry* preserves the topology of the light sources from the *perspective* of the camera (the $+\hat{x}$ axis in figure 4.13).

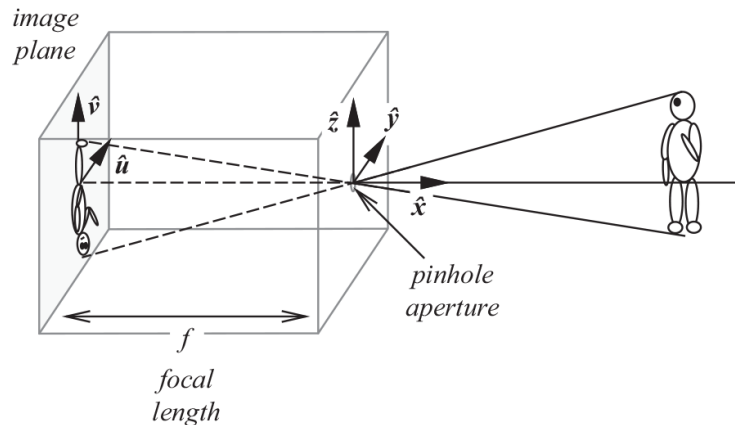


Figure 4.13

The pinhole camera geometry

Pinhole cameras can be found in the writing of philosophers, astronomers, and scientists going back thousands of years in references from China (Mo-Ti, fifth century BC), ancient Greece (Aristotle, ca. 350 BC), and Persia (Alhazen of Basra, tenth century). It was described in the notebooks of Leonardo da Vinci in the fifteenth century, and the sixteenth-century Dutch artist Johannes Vermeer, one of the early realists, used the camera obscura to create his remarkably accurate interior still lifes. With the advent of optical lenses and good quality light sensitive materials early in the nineteenth century, the camera obscura evolved into the modern camera.

The *perspective projection* performed by the pinhole camera is defined by the geometric construction in [figure 4.13](#). By similar triangles, world coordinate (x, y, z) projects to image plane coordinates $u = -fy/x$ and $v = -fz/x$. The positive directions for \hat{u} and \hat{v} are defined by the positive directions of \hat{y} and \hat{z} , respectively, and all valid world coordinates have positive \hat{x} components. Consequently, the image of these sources is inverted on the focal plane. Moreover, scale on the image plane (the ratio of the image plane distance to the Cartesian distance) is inversely proportional to range x . The resulting depth-dependent projective distortion explains why train tracks appear to come together on the horizon at a position called the *vanishing point*. Projective scale distortion diminishes as a subject with finite depth approaches $x = \infty$. Under these circumstances, the projection is approximately *orthographic*. In practical applications, perspective distortion is generally insignificant if the depth of the object is small

relative to its range. Under these circumstances, the subject is said to have *shallow structure* and the projection can be assumed to be essentially orthographic.

4.5.2 Binocular Localization: Forward Kinematics

The perspective projection equations map light rays reflected from three-dimensional objects onto a two-dimensional image plane. Information about depth in the scene is lost in this projection and cannot be recovered without additional information. A stereo pair of cameras with two different perspectives on the scene can reconstruct some of the three-dimensional structure lost in the projection.

Example: Stereo Localization in the Plane

Consider Roger's planar binocular imaging geometry in [figure 4.14](#). At the bottom of the figure, a pair of pinhole cameras separated by $2d$ in the \hat{y} direction are opaque enclosures with pinhole apertures aimed along parallel gaze directions, \hat{x}_L and \hat{x}_R . The reference frame midway between the pinholes is called the stereo (or cycloptic) coordinate frame. The goal is to recover the (x, y) coordinate of point \mathbf{p} in the stereo frame. This is a special case where the two cameras have fixed gazes mutually parallel to the cycloptic \hat{x} axis; depth is encoded exclusively in the "disparity" between the image coordinates of point \mathbf{p} on the left and right image planes. Under these conditions:

$$u_L = \frac{-f(y-d)}{x}, \quad u_R = \frac{-f(y+d)}{x},$$

so that

$$xu_L = -f(y-d), \quad xu_R = -f(y+d).$$

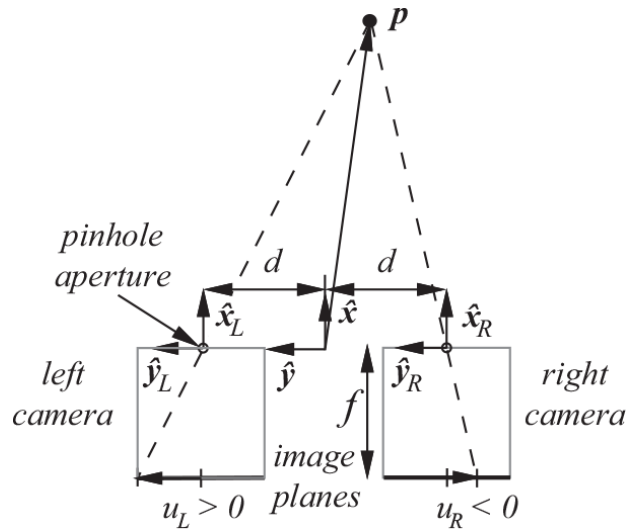


Figure 4.14

Top view of a 2D stereo geometry where depth is encoded exclusively in stereo disparity

The difference of these functions yields $x(u_L - u_R) = 2df$, and therefore

$$x = \frac{2df}{(u_L - u_R)} . \quad (4.12)$$

The range estimate is inversely proportional to the stereo *disparity*, $(u_L - u_R)$, which is positive for any feature p in front of the camera pair. The numerator of [equation 4.12](#) is a positive constant. As a result, the range goes to positive infinity as the disparity goes to zero and regions with equal disparity are equally distant.

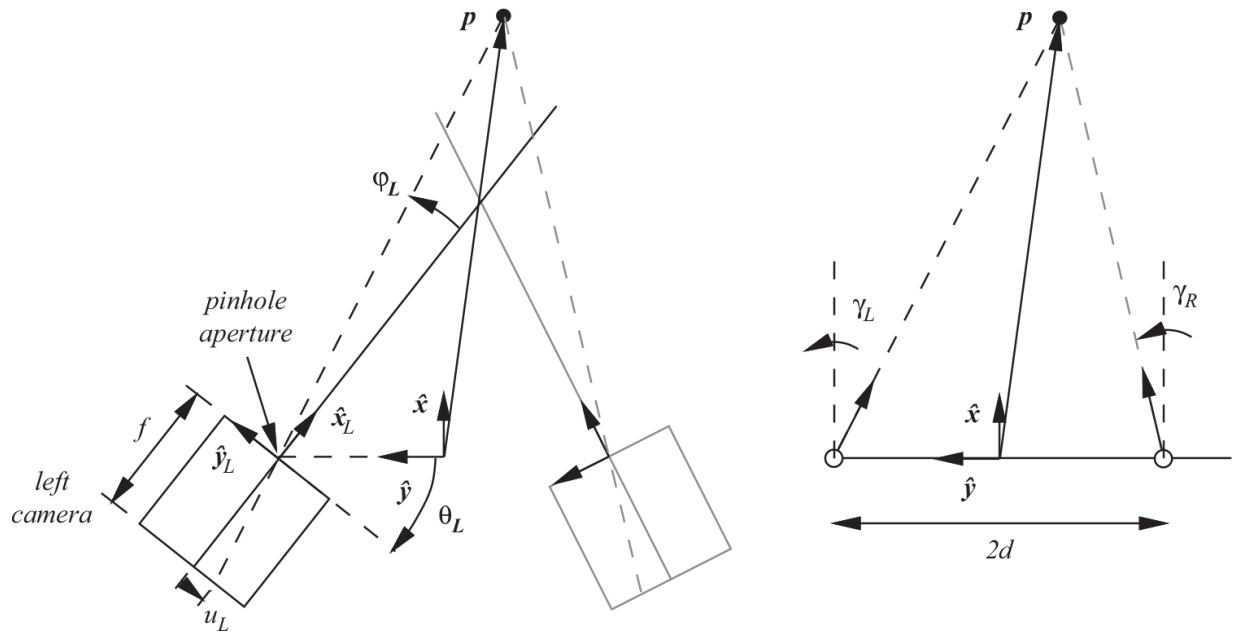


Figure 4.15

The binocular imaging geometry with independently panning cameras. On the right, the stereo geometry is defined in terms of parameter $\gamma = \theta + \phi$, the sum of the eye configuration, θ , and the angular offset from the fovea, ϕ

It is relatively straightforward to generalize this result to cases where the cameras can pan independently (i.e., that can “verge” toward feature p). [Figure 4.15](#) shows the new geometry for this situation and defines variables θ and ϕ for the left camera. We assume that the verge degrees of freedom are located at the pinhole apertures and rotate about axes parallel to the stereo frame \hat{z} axis. The new variable, γ , is the sum of the eye angle, θ , and the angular offset of the feature p from the image center. The illustration on the right of [figure 4.15](#) identifies the governing geometry of this stereo system. It is relatively straightforward to solve for the spatial coordinates of feature p under these conditions.

$$x = 2d \frac{\cos(\gamma_R) \cos(\gamma_L)}{\sin(\gamma_R - \gamma_L)} \quad (4.13)$$

$$y = d + 2d \frac{\cos(\gamma_R) \sin(\gamma_L)}{\sin(\gamma_R - \gamma_L)}$$

It is left as an exercise for the reader to verify these triangulation equations for this geometry.

□

4.6 Hand-Eye Kinematic Transformations

In addition to contact interactions with the world, kinematic structures are a means of delivering sensors to places where useful information is accessible. Supported by joint angle feedback, degrees of freedom in the arms, hands, and head can be used to orient tactile and visual sensors to the world geometry. Visual signals are subject to line-of-sight constraints, and tactile signals are subject to reachability constraints, but together these independent sensors provide richer and more complete information than is possible with either one alone.

Figure 4.16 summarizes the kinematic mapping functions for the arm and eye subsystems of Roger-the-Crab developed in this chapter. It illustrates how they are related to each other and to the Cartesian task space.⁴ The middle panel shows a regular Cartesian grid describing the space around the robot. The reachable subset of the world is green and can be observed using both visual and tactile sensors. The region of space that is not reachable by either arm is colored red—environmental stimuli in this region are accessible only by vision.

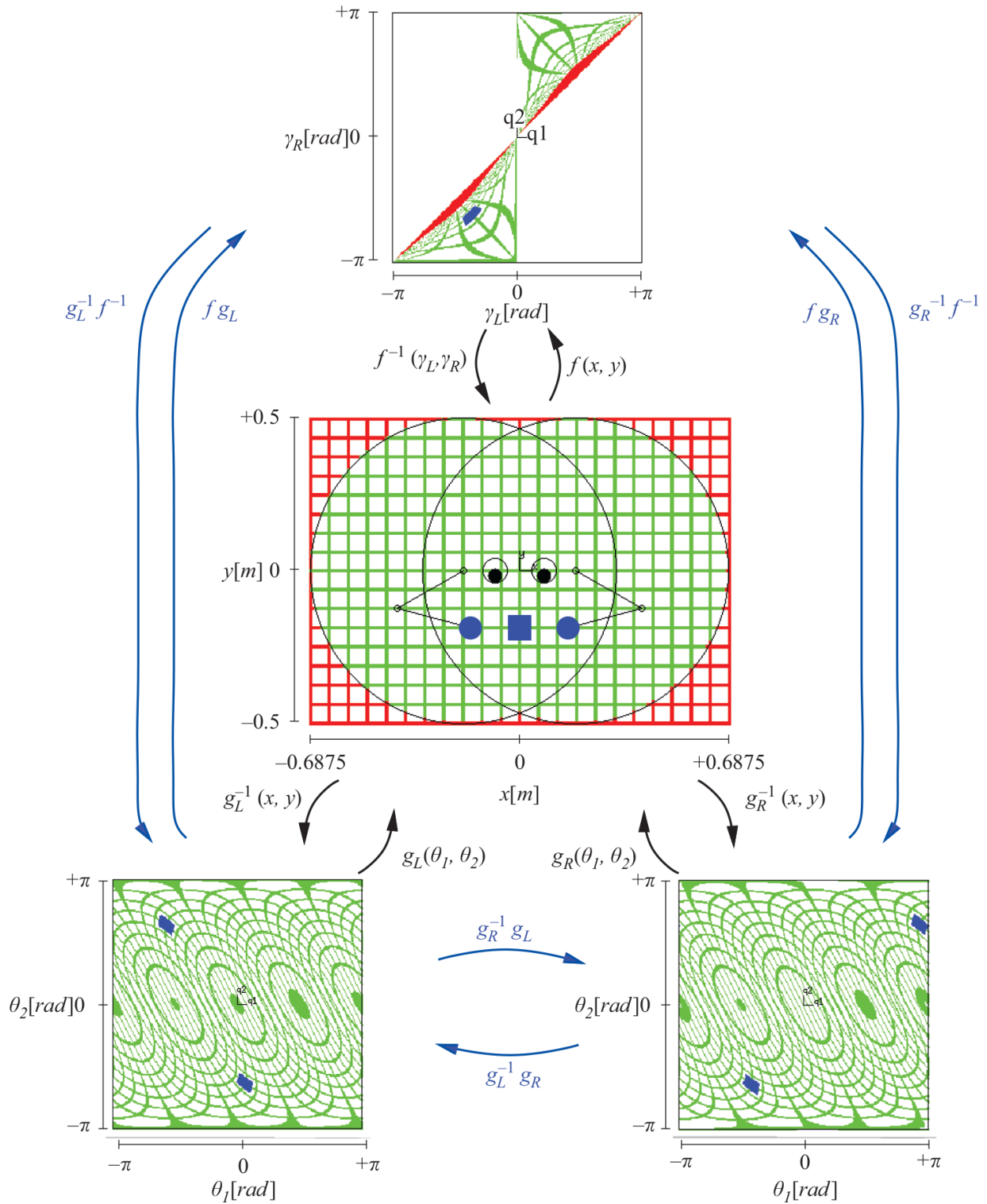


Figure 4.16

Multi-sensor models of space and hand-eye coordination. This illustration uses a different coordinate system than other Roger examples, and arms/hands have been removed from images for clarity.

The bottom panels of [figure 4.16](#) remap the reachable subset of Cartesian space onto the configuration variables of Roger’s arms. Joint angles in the arm can rotate continuously, so these maps are *toroidal*—the top/bottom and left/right edges of the configuration space wrap around at $\pm \pi$. [Figure 4.16](#) shows how the reachable subset of Cartesian space is deformed when parameterized in terms of the configuration variables of Roger’s arms.

The top panel in the figure shows the results of projecting Cartesian space onto the configuration parameters of Roger’s stereo oculomotor system. The world beyond arm’s length is red and maps to a narrow diagonal where the gaze of the two eyes is convergent but very nearly parallel—that is, $\gamma_L \approx \gamma_R$. We observe a kind of fish eye stereo effect that emphasizes a region of space directly in front of the pair of eyes. Lateral and distant regions of space are compressed into a relatively small range of stereo configuration parameters.

The regular Cartesian grid lines in [figure 4.16](#) are deformed when they are projected into these configuration spaces. Both the visual and proprioceptive maps preserve the topology of the Cartesian space, but as a result of this deformation, configuration space observers vary in precision and sensitivity depending on where they are “looking.”

The kinematic mapping functions derived in earlier sections are illustrated in [figure 4.16](#). Function $f(x, y)$ is the pinhole camera projections (section 4.5.1) for the stereo pair that map a coordinate in Cartesian space to visual parameters γ_L and γ_R . Its inverse, $f^{-1}()$, represents the stereo triangulation equations (section 4.5.2). Functions $g(\theta_1, \theta_2)$ and $g^{-1}(x, y)$ are the forward and inverse kinematic transformations, respectively, for Roger’s arms (section 4.4). These mappings are written for the left arm, $g_L()$, and the right arm, $g_R()$. Compositions of these transformations can be used to map directly from oculomotor to arm configurations and vice versa.

Goals in one motor map are strongly associated with goals in the other maps—a visual stimulus, for example, is associated with arm configurations where spatially colocated tactile events are likely to also occur. All the blue features in the configuration space maps refer to the single blue square in the Cartesian map. A unique observation in the oculomotor space is associated with up to two configurations of each arm. This is the basis for a multimodal and fused interpretation of space.

Immediately after birth, human infants use a relatively mature binaural auditory sense of space to bootstrap reaching and visual orientation skills [54, 24, 25]. During the infant’s first year (and beyond), cells from layers of tissue in the visual, auditory, and somatosensory cortices project via neural processes to associated cells in the motor cortex where colocated stimuli are expected. During this period, developmental reflexes orient the neck and hand so as to place the hand in the field of view. Among other things, this reflex generates training data and bootstraps the acquisition of hand-eye projections like this (section 9.3.2). All of these precursor skills and cumulative perceptual abilities support more complex skills in a developmental progression—from spatial mapping, to objects, to grasping and manipulation, and to words (section 9.4). The first complete sense of the space outside the infant is supported by correlations between sight, sound, and touch that capture mappings like those in [figure 4.16](#).

4.7 Kinematic Conditioning

Techniques for evaluating the *kinematic condition* of a robot rely on methods for analyzing the ability of the mechanism to deliver velocities as well as positions. Tools from linear analysis provide useful insight; however, forward kinematic transformations are generally nonlinear. To make use of these tools, the forward kinematic equations must be linearized around an operating configuration using the *Jacobian* of the nonlinear kinematic equations.

4.7.1 Jacobian

Nonlinear forward kinematic functions $\mathbf{r}(\mathbf{q})$, like those in [equations 4.5](#) and [4.13](#), map coordinates in configuration space, \mathbf{q} , to Cartesian space. Assume that these functions are *analytic* so that they can be written in the neighborhood of $\mathbf{q} = \mathbf{a}$ in the form of a Taylor series:

$$\mathbf{r}(\mathbf{a} + d\mathbf{q}) = \mathbf{r}(\mathbf{a}) + \frac{d\mathbf{q}}{1!} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} + \frac{d\mathbf{q}^2}{2!} \frac{\partial^2 \mathbf{r}}{\partial \mathbf{q}^2} + \dots \quad (4.14)$$

The first-order linear approximation of the function $\mathbf{r}(\mathbf{q})$ near $\mathbf{q} = \mathbf{a}$ is extracted from the Taylor series by ignoring higher-order terms,

$$dr|_{q=a} \approx \left. \frac{\partial r}{\partial q} \right|_{q=a} dq = \mathbf{J}|_{q=a} dq. \quad (4.15)$$

Equation 4.15 maps differential displacements in configuration space, $d\mathbf{q}$, into differential displacements in Cartesian space, $d\mathbf{r}$. Equivalently, it transforms input velocities to output velocities. The matrix of partial derivatives, $\partial\mathbf{r}/\partial\mathbf{q}$, is the *Jacobian*, \mathbf{J} , named after the mathematician Carl Gustav Jacobi. It represents the tangent hyperplane of the function $\mathbf{r}(\mathbf{q})$ in the neighborhood $d\mathbf{q}$ surrounding $\mathbf{q} = \mathbf{a}$. The Jacobian is an accurate local approximation of the original nonlinear system in the limit as $d\mathbf{q} \rightarrow \mathbf{0}$ and it can be analyzed using powerful tools applicable to linear systems.

4.7.2 The Manipulator Jacobian

The Cartesian velocity of the manipulator is derived by differentiating the forward kinematic mapping and, in general, depends on both the pose and the velocity of the joints in the mechanism. The analysis can be applied at any point on an articulated structure; however, without loss of generality, we consider the velocity of the endpoint of the kinematic chain and illustrate the idea using the planar 2R manipulator whose forward kinematic equations were derived in section 4.4.1.

Example: First-Order Velocity Control for the Planar 2R Manipulator

The x - y components of the forward kinematic function for the planar 2R manipulator (**equation 4.5**) are repeated here:

$$r_x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2),$$

$$r_y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2).$$

The first derivative of these equations is written

$$dr_x = -l_1 \sin(\theta_1) d\theta_1 - l_2 \sin(\theta_1 + \theta_2) d\theta_1 - l_2 \sin(\theta_1 + \theta_2) d\theta_2, \text{ and} \quad (4.16)$$

$$dr_y = l_1 \cos(\theta_1) d\theta_1 + l_2 \cos(\theta_1 + \theta_2) d\theta_1 + l_2 \cos(\theta_1 + \theta_2) d\theta_2,$$

which can be rewritten to emphasize the locally linear relationship:

$$dr = \begin{bmatrix} dr_x \\ dr_y \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} = \mathbf{J}d\boldsymbol{\theta}. \quad (4.17)$$

The manipulator Jacobian \mathbf{J} is a 2×2 matrix that defines the locally linear mapping from a two-dimensional displacement/velocity in configuration space $[d\theta_1 \ d\theta_2]^T$ to a two-dimensional displacement/velocity in Cartesian space $[dx \ dy]^T$ at the manipulator endpoint.

With no joint range limits, this manipulator can execute continuous rotations in both θ_1 and θ_2 —that is, it can execute arbitrary velocities $[\dot{\theta}_1 \ \dot{\theta}_2]$ from any configuration $\boldsymbol{\theta}$ in configuration space. Suppose that the set of executable velocities forms the unit disk in configuration space centered on configuration $\boldsymbol{\theta}$, as illustrated on the left side of [figure 4.17](#). In general, the projection into Cartesian space represented by the Jacobian deforms this set, as illustrated on the right side of the diagram.

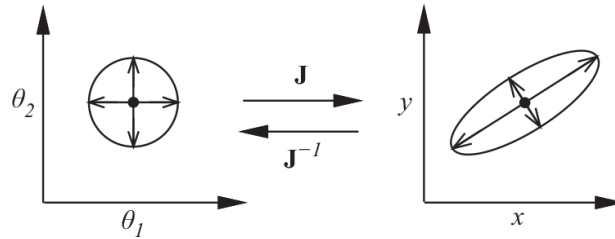


Figure 4.17

The Jacobian of the 2R manipulator transforms sets of executable velocities in configuration space from posture (θ_1, θ_2) into sets of achievable velocities in Cartesian space from endpoint position (x, y)

The determinant (appendix A.2) of the Jacobian in [equation 4.17](#) is a scalar value with units $[m^2/rad^2]$ proportional to the “scale” of the transformation. The determinant for the 2R manipulator Jacobian is written

$$\begin{aligned} \begin{vmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{vmatrix} &= -l_1 l_2 s_1 c_{12} - l_2^2 s_{12} c_{12} + l_1 l_2 c_1 s_{12} + l_2^2 c_{12} s_{12}, \\ &= l_1 l_2 (c_1 s_{12} - s_1 c_{12}) = l_1 l_2 s_2. \end{aligned} \quad (4.18)$$

It describes the degree to which velocities in configuration space are amplified or attenuated as they are projected into velocities in the Cartesian plane. A determinant of zero indicates that the Jacobian is *singular* and has lost rank. In general, this means that an n -dimensional input maps (locally)

to a lower-dimensional manifold of the output space. From [equation 4.18](#), the 2R manipulator Jacobian is singular when $\sin(\theta_2)=0$, or when $\theta_2 = 0, \pi$ ([figure 4.18](#)). These configurations correspond to endpoint positions on the outer and inner boundary of the reachable workspace. In these postures, the manipulator can not generate arbitrary velocities in the x - y plane. [Figure 4.18](#) shows that the achievable endpoint velocities for these singular configurations are limited to a one-dimensional subset of the Cartesian plane.

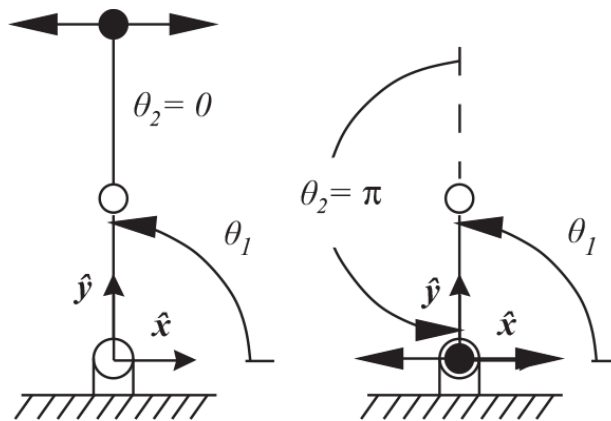


Figure 4.18
Singularities in the 2R manipulator

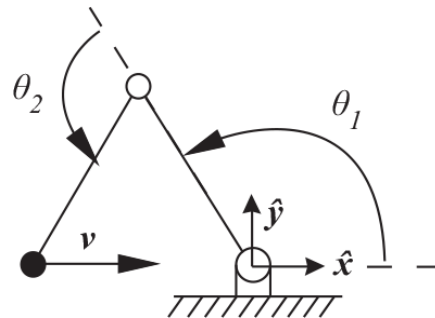


Figure 4.19
The 2D planar manipulator and a specific endpoint velocity command

In nonsingular configurations, however, the manipulator Jacobian is full rank and is, therefore, invertible. Suppose that Roger’s arm is required to execute an endpoint velocity of $v=1$ [m/sec] in the \hat{x} direction as in [figure 4.19](#). Inverting [equation 4.17](#) (appendix A.2) yields

$$\mathbf{J}^{-1} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}$$

so that

$$\begin{aligned} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} &= \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} \\ -l_1 c_1 - l_2 c_{12} \end{bmatrix} \begin{bmatrix} \text{rad} \\ \text{sec} \end{bmatrix}. \end{aligned} \quad (4.19)$$

This closed-form relation defines the joint angle velocities that produce the target endpoint velocity for every pose (θ_1, θ_2) of the manipulator. As the arm approaches singular configurations like those in [figure 4.18](#), the inverse of the Jacobian becomes ill-conditioned because $1/l_1 l_2 s_2$ goes to infinity. As a consequence, [equation 4.19](#) predicts that the fully extended 2R manipulator requires infinite joint space velocities to withdraw the endpoint radially (along the $-\hat{y}$ direction in [figure 4.18](#)) toward the origin. Readers can easily verify that this is not the correct conclusion using their own arms. Omitting the higher-order terms in [equation 4.14](#) and relying solely on a first-order approximation can introduce significant inaccuracies, and the predicted behavior near singularities can be misleading in important ways.

□

Manipulability Ellipsoid—The scalar determinant is only a coarse measure of manipulator conditioning—the manipulator Jacobian also provides an opportunity to examine the spatial character of the velocity transform. If we assume that joint angle velocities in the manipulator are limited to the unit (hyper)sphere, $\dot{\theta}^T \dot{\theta} \leq 1$, then we can derive the quadratic form ([appendix A.6](#)) describing the corresponding envelope of Cartesian endpoint velocities.

$$\dot{\theta}^T \dot{\theta} = (\mathbf{J}^{-1} \dot{r})^T (\mathbf{J}^{-1} \dot{r}) = \dot{r}^T [(\mathbf{J}^{-1})^T \mathbf{J}^{-1}] \dot{r} = \dot{r}^T (\mathbf{J}\mathbf{J}^T)^{-1} \dot{r} \leq 1 \quad (4.20)$$

The rightmost inequality in [equation 4.20](#) defines the quadratic form we are interested in. It describes the Cartesian *manipulability ellipsoid*—a set of Cartesian velocities generated by mapping the unit (hyper)sphere of joint

space velocities through the manipulator Jacobian. Appendix A.6 shows how the ellipsoid is determined from the eigenvalues and eigenvectors of $\mathbf{J}\mathbf{J}^T$.

The velocity ellipsoid is pictured for the 2R planar manipulator in [figure 4.20](#). The diagram illustrates the relative ability of the arm to generate endpoint velocities from this configuration in different directions. Unit vectors, \hat{e}_1 and \hat{e}_2 , are the eigenvectors of $\mathbf{J}\mathbf{J}^T$. They define the principal axes of the transformation. The relative amplification of velocity in these directions is proportional to the square root of the corresponding eigenvalues λ_1 and λ_2 of $\mathbf{J}\mathbf{J}^T$. In this light, the directions in which the manipulator is best at transforming small joint angle velocities $\dot{\theta}$ into large Cartesian velocities \dot{r} is identified by the eigenvector with the largest eigenvalue and the scale is proportional to the square root of that eigenvalue.

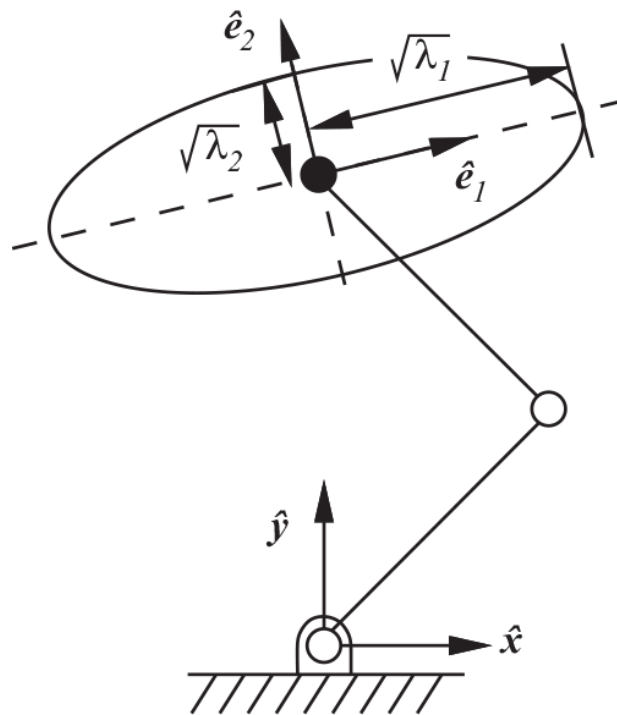


Figure 4.20

The velocity ellipsoid in terms of the eigenvalues and eigenvectors of $\mathbf{J}\mathbf{J}^T$ for the planar 2R manipulator

Conversely, imagine an encoder that measures joint angle positions over the interval $[0, 2\pi)$ with 8 bits of precision. In this case, the encoder

resolves $2\pi/2^8 = [\text{rad}/\text{tick}]$. Projecting this resolution through the Jacobian yields

$$\mathbf{J} \begin{bmatrix} m \\ \text{rad} \end{bmatrix} \frac{2\pi}{256} \begin{bmatrix} \text{rad} \\ \text{tick} \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix} \frac{2\pi}{256} \begin{bmatrix} m \\ \text{tick} \end{bmatrix}.$$

Thus, the eigenvector with the smallest eigenvalue identifies the direction in Cartesian space where precision is optimized—the direction from the current pose where the sensor resolves the smallest Cartesian displacements per encoder tick.

Force Ellipsoid—The work output produced at the endpoint of a mechanism that is in contact with the world is the product of endpoint force \mathbf{f} and distance $d\mathbf{r}$ traveled. If we assume no losses, then the work input in configuration space must be equal to the work output in Cartesian space [65],

$$\boldsymbol{\tau}^T d\boldsymbol{\theta} = \mathbf{f}^T d\mathbf{r}.$$

However, since $d\mathbf{r} = \mathbf{J}d\boldsymbol{\theta}$, we can write

$$\boldsymbol{\tau}^T d\boldsymbol{\theta} = \mathbf{f}^T [\mathbf{J}d\boldsymbol{\theta}],$$

so that

$$\boldsymbol{\tau}^T = \mathbf{f}^T \mathbf{J}, \quad \text{or} \quad \boldsymbol{\tau} = \mathbf{J}^T \mathbf{f}. \quad (4.21)$$

The manipulator Jacobian, therefore, describes the transformation from joint velocity to Cartesian velocity *and* the mapping from Cartesian endpoint forces to joint torques.

As a consequence, the velocity ellipsoid (equation 4.20) has a direct analog in the quadratic form representing the *force ellipsoid*

$$\boldsymbol{\tau}^T \boldsymbol{\tau} = (\mathbf{J}^T \mathbf{f})^T (\mathbf{J}^T \mathbf{f}) = \mathbf{f}^T (\mathbf{J}\mathbf{J}^T) \mathbf{f} \leq 1. \quad (4.22)$$

Thus, unit hypersphere $\boldsymbol{\tau}^T \boldsymbol{\tau}$ maps through the manipulator Jacobian to the Cartesian force ellipsoid defined by the eigenvalues and eigenvectors of $(\mathbf{J}\mathbf{J}^T)^{-1}$. The eigenvectors of $\mathbf{J}\mathbf{J}^T$ and $(\mathbf{J}\mathbf{J}^T)^{-1}$ are identical, and the eigenvalues of $\mathbf{J}\mathbf{J}^T$ (velocity amplifier) are reciprocals of the eigenvalues of

$(\mathbf{J}\mathbf{J}^T)^{-1}$ (force amplifier). Therefore, once the eigenvectors and eigenvalues of $\mathbf{J}\mathbf{J}^T$ are established, both the velocity and force capabilities of the manipulator can be determined.

Example: Velocity and Force Ellipsoids for Roger

Velocity and force conditioning ellipsoids for Roger’s arms are illustrated in [figure 4.21](#). Close inspection verifies that the eigenvectors are the same and that eigenvalues are reciprocal. For velocity (left), a large eigenvalue of $\mathbf{J}\mathbf{J}^T$ amplifies relatively small $\dot{\theta}$ inputs into large \dot{r} outputs along the corresponding eigenvector. This is evident near $x=0.5$ [m], where the left arm is nearly fully extended and joint velocities are mapped efficiently through the Jacobian to Cartesian velocities in the $\pm\hat{y}$ direction. Conversely, the arm is quite good at detecting small Cartesian displacements of the endpoint in the \hat{x} direction as relatively large displacements of the joints in the arm.

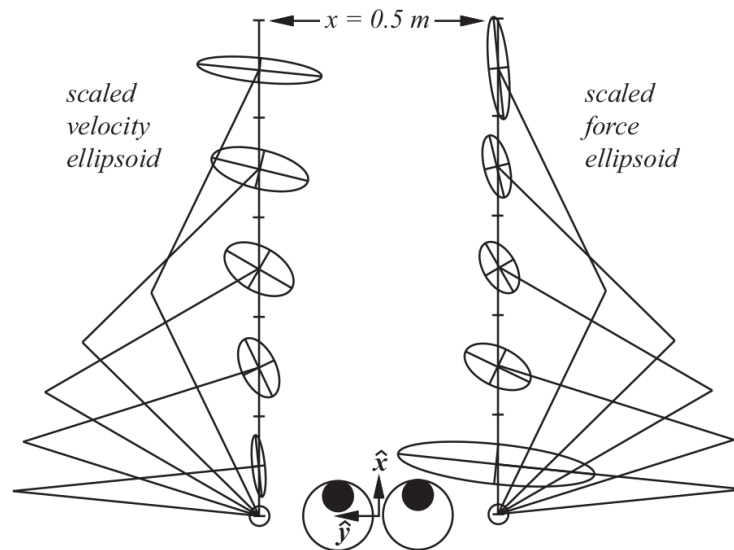


Figure 4.21
The manipulator conditioning ellipsoids $\mathbf{J}\mathbf{J}^T$ for Roger’s 2R manipulator

The nearly extended right arm demonstrates the same kind of kinematic efficiency in the force generating capacity of the arm. Unlike the capacity for Cartesian velocity, when the robot arm is extended, it is quite good at converting relatively small joint torques into relatively large endpoint

forces. Loads in the \hat{x} direction on the extended arm are carried primarily in the links (skeleton) and not in the actuators (musculature). Conversely, in directions with small eigenvalues, small external loads produce relatively large joint torques. As result, these robot configurations are capable of detecting and precisely controlling relatively small interaction forces at the expense of strength.

□

The tools introduced in this section characterize (to first order) how kinematic devices produce Cartesian velocities and forces. The Jacobian is the key to this analysis—it can be viewed as a configuration-dependent velocity amplifier that describes kinematic mappings. In the next section, the same analysis will be applied to stereo triangulation. In this analysis, however, the emphasis is not force and velocity; instead, we focus on how visual acuity depends on the kinematic configuration of the oculomotor system.

4.7.3 Stereo Localizability

In section 4.5.2, stereo reconstruction is posed as a kinematics problem. Stereo triangulation equations are used to transform the configuration of a pair of eyes and the signal on the image plane (collectively, an oculomotor configuration) into an estimate of the Cartesian coordinate of the light source.

Errors in the result of stereo triangulation arise from errors in the mechanism, imprecision in joint angle sensors, optical aberrations in lenses, and finite resolution on the image plane. In this section, we consider the last of these sources of uncertainty, the finite resolution of the image plane, and consider how bounded input errors map to expected triangulation errors.

The precision of the stereo system can be analyzed using the techniques developed in the previous section for the manipulator—namely, by mapping a bounded differential error in the oculomotor configuration through a quadratic form based on the oculomotor Jacobian to produce a distribution over the differential errors expected in the Cartesian location of the light source.

Example: Roger's Oculomotor Jacobian and Stereo Localizability

Figure 4.22 reproduces the important geometry from the binocular system treated in section 4.5 in the context of Roger's stereo visual system. The stereo triangulation results for this geometry are repeated here in equation 4.23.

$$x = 2d \frac{\cos(\gamma_R) \cos(\gamma_L)}{\sin(\gamma_R - \gamma_L)} \quad (4.23)$$

$$y = d + 2d \frac{\cos(\gamma_R) \sin(\gamma_L)}{\sin(\gamma_R - \gamma_L)}$$

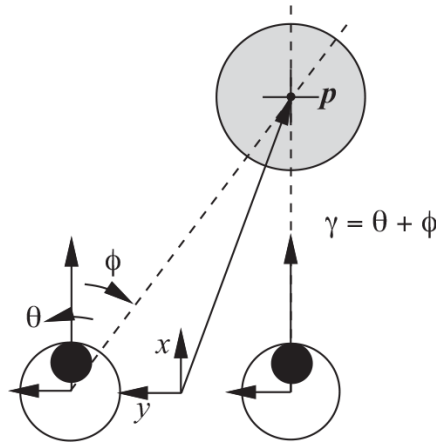


Figure 4.22

Triangulation parameters for Roger's stereo system

These equations localize point p . To estimate the sensitivity of this result to small errors in the imaging geometry, we compute the oculomotor Jacobian.

The triangulation equations are differentiated to create the local, linear approximation $dp = Jdy$. The partial derivative of equation 4.23 with respect to configuration variables, γ_L and γ_R , yields

$$\begin{aligned} \begin{bmatrix} dx \\ dy \end{bmatrix} &= \begin{bmatrix} \frac{\partial x(\gamma_L, \gamma_R)}{\partial \gamma_L} & \frac{\partial x(\gamma_L, \gamma_R)}{\partial \gamma_R} \\ \frac{\partial y(\gamma_L, \gamma_R)}{\partial \gamma_L} & \frac{\partial y(\gamma_L, \gamma_R)}{\partial \gamma_R} \end{bmatrix} \begin{bmatrix} d\gamma_L \\ d\gamma_R \end{bmatrix}, \\ &= \frac{2d}{\sin^2(\gamma_R - \gamma_L)} \begin{bmatrix} \cos^2(\gamma_R) & -\cos^2(\gamma_L) \\ \sin(\gamma_R)\cos(\gamma_R) & -\sin(\gamma_L)\cos(\gamma_L) \end{bmatrix} \begin{bmatrix} d\gamma_L \\ d\gamma_R \end{bmatrix}. \quad (4.24) \end{aligned}$$

Equation 4.24 maps differential changes in a feature coordinate on the image plane, $d\boldsymbol{\gamma}^T = [d\gamma_L \ d\gamma_R]$, to differential changes in the estimated position of the source $d\mathbf{p}^T = [dx \ dy]$.

If the net error from all sources is bounded to a disk of radius k on the image plane, then

$$d\boldsymbol{\gamma}^T d\boldsymbol{\gamma} = d\mathbf{r}^T (\mathbf{J}\mathbf{J}^T)^{-1} d\mathbf{r} \leq k^2 \quad (4.25)$$

defines the stereo localizability ellipsoid. Roger's focal length is 64 pixels. If features on the image plane are accurate to 1 pixel, then the maximum angular error in $\boldsymbol{\gamma}$ is bounded by $k = \tan^{-1}(1/64) = 0.01562379 \text{ rad}$. Therefore, the localizability ellipsoid has principal axes of length $k\sqrt{\lambda_1}$ and $k\sqrt{\lambda_2}$, where λ_1 and λ_2 are the eigenvalues of $\mathbf{J}\mathbf{J}^T$.

Figure 4.23 illustrates localizability ellipsoids at several positions (marked by crosshairs) in Roger's field of view. The ellipsoids illustrate the shape and relative magnitude of the Cartesian error covariance. Generally, the lateral error is relatively small and the radial error can be significant depending on the position of the subject. The circular ellipsoid closest to the eyes is the position where stereo acuity is maximized.

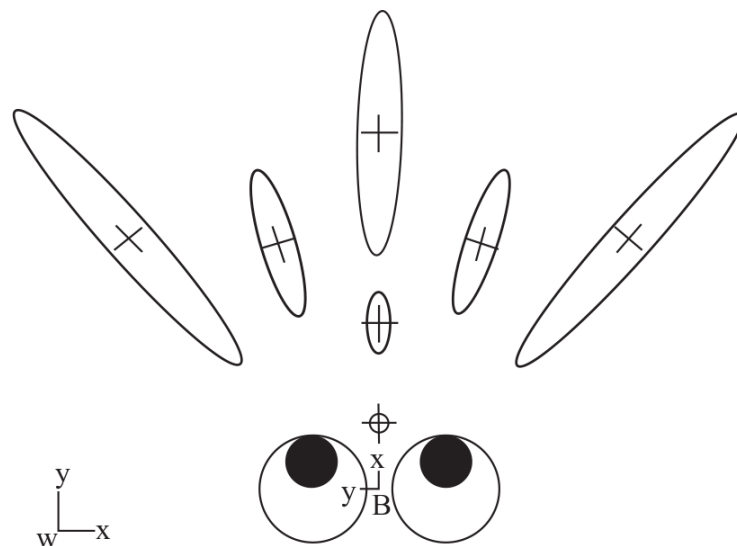


Figure 4.23

Scaled localizability ellipsoids for Roger's stereo geometry

□

4.8 Kinematic Redundancy

All the Jacobians introduced to this point have been square—the dimension of the input space has been equal to the dimension of the output space. Thus, the nonsingular 2R manipulator in section 4.7.2 generates output velocities in the two-dimensional Cartesian plane. However, there is no requirement that input and output dimensions should be the same. In general, a forward velocity transform can be written in terms of the nonsquare Jacobian $\mathbf{J} \in \mathbb{R}^{m \times n}$, where $m \neq n$.

The velocity transformation $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$ with $\dot{\mathbf{q}} \in \mathbb{R}^n$ and $\dot{\mathbf{r}} \in \mathbb{R}^m$ is *redundant* if $\mathbf{J} \in \mathbb{R}^{m \times n}$ has more columns than rows ($n > m$) and \mathbf{J} is full rank—that is, the determinant of $\mathbf{J}\mathbf{J}^T \in \mathbb{R}^{m \times m}$ is nonzero. Under these conditions, the manipulator has redundant degrees of freedom, and an infinite number of inverse kinematic solutions exist. The set of such solutions is defined on a continuous $(n - m)$ dimensional manifold, where $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$, known as the *nullspace* of the redundant Jacobian. A sequence of movements in the nullspace trace out a path on the multidimensional *self-motion* manifold.

Example: Self-Motion Manifold

With respect to endpoint positions in the x - y plane, the planar 3R manipulator in [figure 4.24](#) is redundant. The forward kinematic equations for this manipulator are:

$$\begin{aligned}x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)\end{aligned}$$

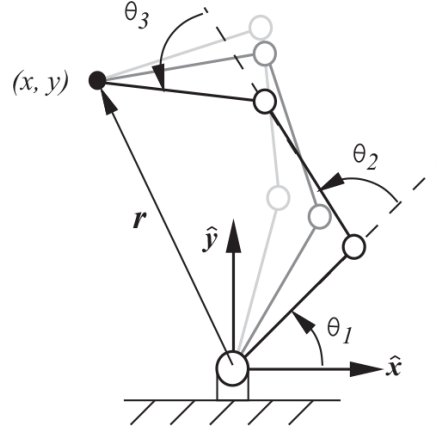


Figure 4.24

Redundant solutions for the planar 3R manipulator at $x=-1$, $y=\sqrt{2}$

and the Jacobian is written

$$\mathbf{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix}.$$

The Jacobian defines the velocity relation, $\dot{\mathbf{r}} = \mathbf{J}\dot{\boldsymbol{\theta}}$, and the self-motion manifold is determined by solving for the configuration space velocity, $\dot{\boldsymbol{\theta}}_{null}$, that causes no displacement in the endpoint—that is, $\dot{\mathbf{r}} = 0$.

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \\ &= \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (4.26) \end{aligned}$$

Equation 4.26 represents a system of two equations in three unknowns that can be solved for internal motions by introducing one more constraint—namely, that $\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 = 1$. Under these circumstances, one can verify that

$$\dot{\boldsymbol{\theta}}_{null} = \begin{bmatrix} l_2 l_3 \sin(\theta_3) \\ -l_2 l_3 \sin(\theta_3) - l_1 l_3 \sin(\theta_2 + \theta_3) \\ l_1 l_2 \sin(\theta_2) + l_1 l_3 \sin(\theta_2 + \theta_3) \end{bmatrix}. \quad (4.27)$$

A trajectory on the self-motion manifold for a particular endpoint position is generated by following $\dot{\theta}_{null}$ through a sequence of manipulator postures (figure 4.25).

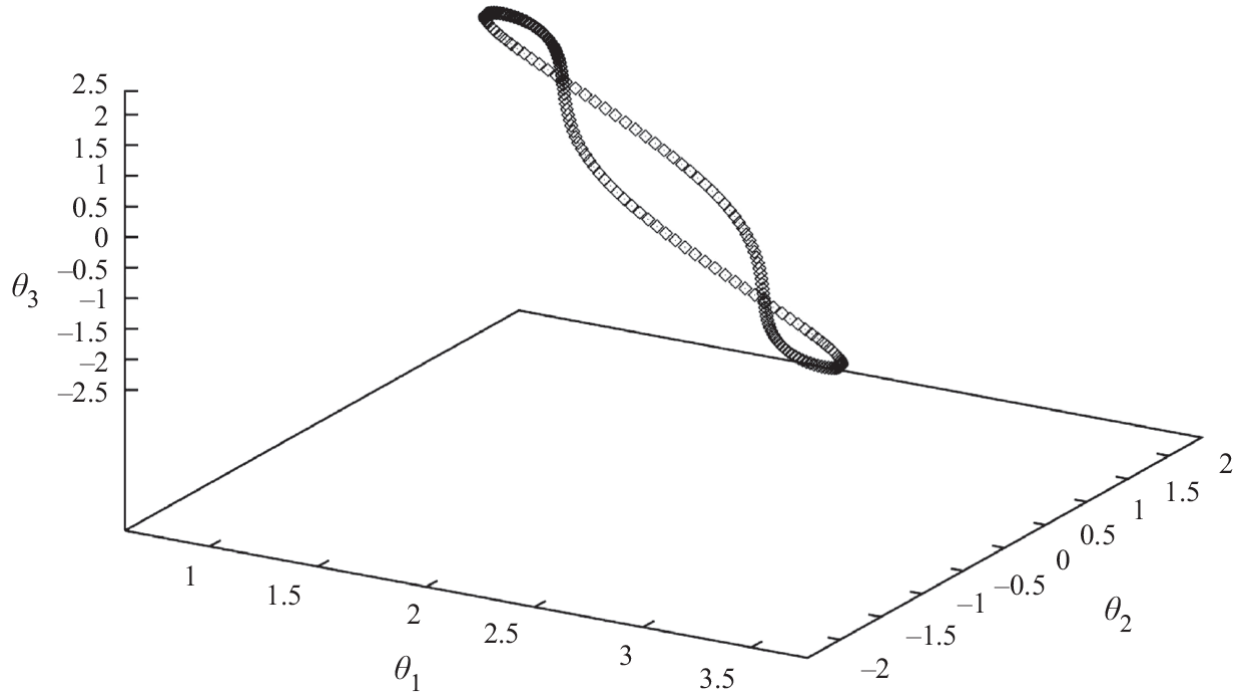


Figure 4.25

Internal motions along the self-motion manifold of the planar 3R manipulator with $l_1 = l_2 = l_3 = 1$ and fixed endpoint position at $x = -1.0$, $y = \sqrt{2}$

□

Internal motions introduce postural flexibility that supports multiple objective control designs—to avoid kinematic singularities or meet force and velocity requirements, for example—to optimize subordinate tasks in the nullspace of the Cartesian endpoint velocity task.

The general solution for redundant systems (like equation 4.26) involves inverting nonsquare Jacobians $\mathbf{J} \in \mathbb{R}^{m \times n}$. When the Jacobian has more rows than columns ($m > n$), there are more constraint equations than unknowns and the system of equations is *overconstrained*. When there are fewer rows than columns ($m < n$) the system is underconstrained—there are fewer constraint equations than there are unknowns. Redundant Jacobians are examples of underconstrained systems and, consequently, there is no unique

inverse. In the previous example concerning the planar 3R manipulator, there are many velocities in configuration space that map through \mathbf{J} to the same velocity in Cartesian space.

In appendix A.9, the pseudoinverse is introduced to address both under- and overconstrained systems using least squares optimal solutions. In the underconstrained (redundant) case, $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$, where $\dot{\mathbf{r}} \in \mathbb{R}^m$, $\dot{\mathbf{q}} \in \mathbb{R}^n$, and $m < n$, the result is the *right pseudoinverse*

$$\mathbf{J}^\# = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \quad (4.28)$$

so that $\dot{\mathbf{q}} = \mathbf{J}^\# \dot{\mathbf{r}}$ yields a unique inverse kinematic solution that minimizes $\mathbf{q}^T \mathbf{q}$. Chapter 10 will return to the implications of redundancy in the control of complex robots by introducing tools that exploit redundancy in order to co-articulate multiple, simultaneous control objectives.

4.9 Exercises

1. **Inverting the homogeneous transform.** Given the general expression for the homogeneous transform and its inverse:

$${}_A\mathbf{T}_B = \left[\begin{array}{ccc|c} \hat{\mathbf{x}}_{3 \times 1} & \hat{\mathbf{y}}_{3 \times 1} & \hat{\mathbf{z}}_{3 \times 1} & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]_{4 \times 4}$$

$${}_B\mathbf{T}_A = [{}_A\mathbf{T}_B]^{-1} = \left[\begin{array}{cc|c} \hat{\mathbf{x}}_{1 \times 3}^T & -\hat{\mathbf{x}}^T \mathbf{t} \\ \hat{\mathbf{y}}_{1 \times 3}^T & -\hat{\mathbf{y}}^T \mathbf{t} \\ \hat{\mathbf{z}}_{1 \times 3}^T & -\hat{\mathbf{z}}^T \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]_{4 \times 4}$$

Prove that ${}_B\mathbf{T}_A$ is the inverse of ${}_A\mathbf{T}_B$.

2. **Homogeneous structure of operators in SE(3).** Homogeneous transforms are not the most compact representation for describing spatial relationships; in fact, much of the information in a homogeneous transform is redundant. Given the following transform,

$$\mathbf{T} = \begin{bmatrix} ? & 0 & -1 & 0 \\ ? & 0 & 0 & 1 \\ ? & -1 & 0 & 2 \\ ? & 0 & 0 & 1 \end{bmatrix},$$

use the structural properties of the homogeneous transform to solve for the unknown values.

3. Spatial algebra with the homogeneous transform.

(a) **Composing homogeneous transforms.** Consider three coordinate frames: W (world), A , and B , where

$${}^W\mathbf{T}_A = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad {}^W\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

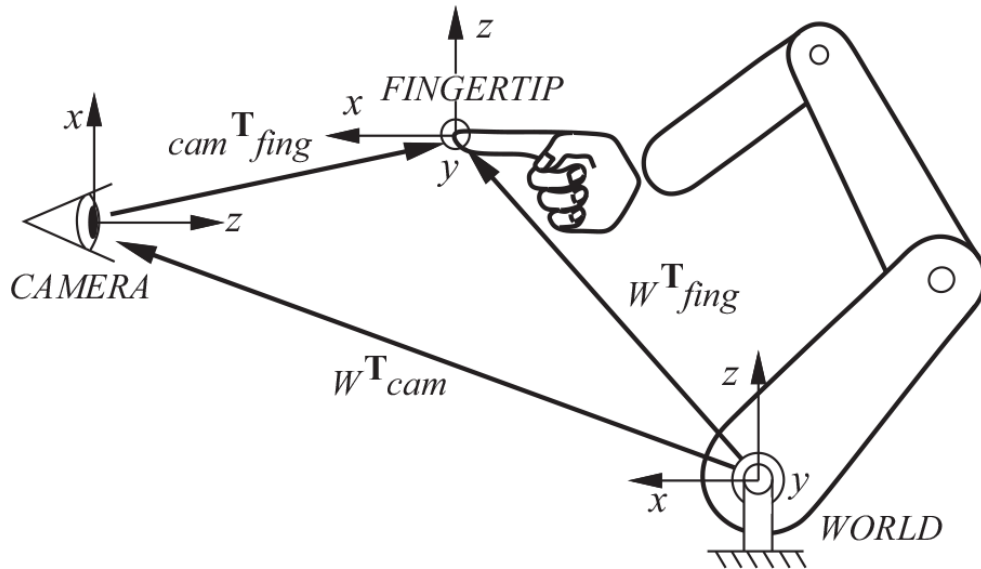
- i. Use the given transforms for ${}^W\mathbf{T}_A$ and ${}^W\mathbf{T}_B$ to calculate the transform from A to B , ${}^A\mathbf{T}_B$.
- ii. The position vector that locates point P with respect to coordinate frame B is given by

$$\mathbf{r}_B^T = [-1, 1.5, 0, 1].$$

Solve for the position vector that locates point p with respect to frame A , \mathbf{r}_A .

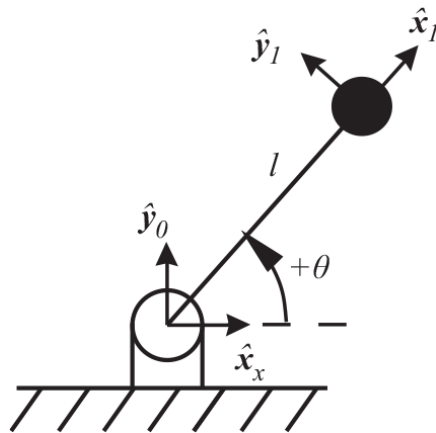
(b) **Hand-eye coordination.** Given homogeneous transforms from world to camera coordinates and from world to fingertip coordinates:

$${}^W\mathbf{T}_{cam} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 0 & -1 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^W\mathbf{T}_{fing} = \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- i. Solve for ${}_{cam}T_{fing}$.
- ii. Compute the position of the hand's fingertip in the camera's coordinate frame.

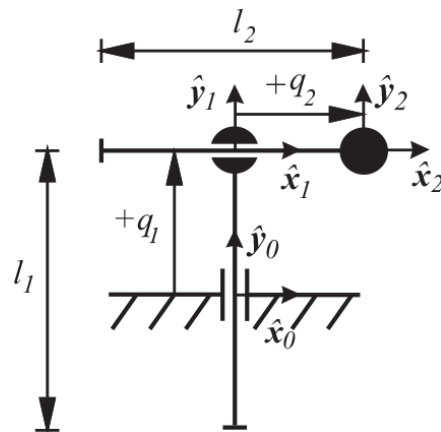
4. **Simple, open-chain machines: 1R mechanism.** The mechanism in this diagram is composed of a single revolute degree of freedom. The configuration $q = \theta$ maps the endpoint of the mechanism to outputs (x, y) . Derive the



- (a) *forward kinematics* by composing the set of homogeneous transforms ${}_{i-1}T_i; i=1, N$, through the intermediate coordinate frames illustrated, and compute the composite ${}_{0}T_N$ transform;
- (b) *reachable workspace*, the subset of Cartesian space that the endpoint (frame N) can reach; and

(c) *Jacobian*, describing how joint velocities map (instantaneously) into Cartesian velocities.

5. **Simple, open-chain machines: 2P mechanism.** The machine in this diagram is composed of two consecutive prismatic degrees of freedom (implemented as linear sliding elements) with input degrees of freedom (q_1 , q_2) and generates endpoint positions (x , y). Derive the

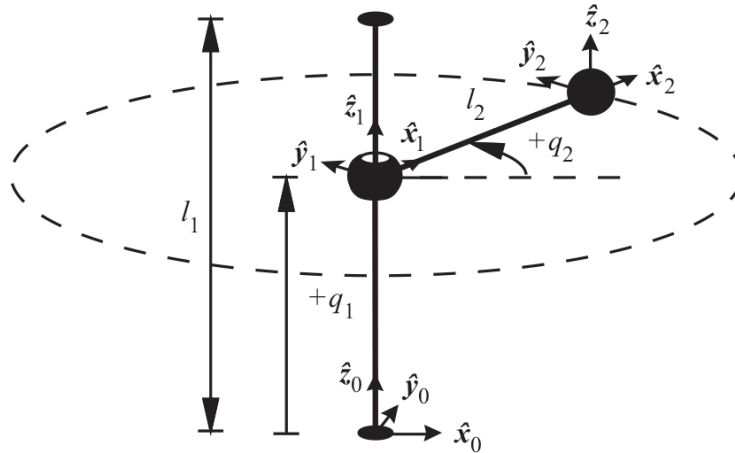


(a) *forward kinematics* by composing the set of homogeneous transforms ${}_{i-1}T_i; i=1, N$, through the intermediate coordinate frames illustrated, and compute the composite ${}_0T_N$ transform;

(b) *reachable workspace*, the subset of Cartesian space that the endpoint (frame N) can reach; and

(c) *Jacobian*, describing how joint velocities map (instantaneously) into Cartesian velocities.

6. **Simple, open-chain machines: PR mechanism.** In this diagram of an open-chain mechanism consisting of a sequence of prismatic and revolute degrees of freedom, the prismatic joint moves through range of motion $0 \leq q_1 \leq l_1$ and the revolute joint can rotate continuously $0 \leq q_2 < 2\pi$. The length of the link from the revolute joint to the endpoint is l_2 . Derive the



(a) *forward kinematics* by composing the set of homogeneous transforms ${}_{i-1}T_i$: $i=1, N$, through the intermediate coordinate frames illustrated, and compute the composite ${}_0T_N$ transform;

(b) *reachable workspace*, the subset of Cartesian space that the endpoint (frame N) can reach; and

(c) *Jacobian*, describing how joint velocities map (instantaneously) into Cartesian velocities.

7. **Kinematic analysis.** The planar 2R mechanism is used in several examples and serves as Roger's arm in the Roger-the-Crab simulator. It consists of a pair of adjacent revolute joints that rotate about the world frame \hat{z} axis.

(a) Write the homogeneous transforms for ${}_0\mathbf{T}_1$, ${}_1\mathbf{T}_2$, ${}_2\mathbf{T}_3$, and compute the net transform ${}_0\mathbf{T}_3 = {}_0\mathbf{T}_1 {}_1\mathbf{T}_2 {}_2\mathbf{T}_3$. Simplify the result.

(b) The Jacobian for the 2 DOF manipulator is derived in the text:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

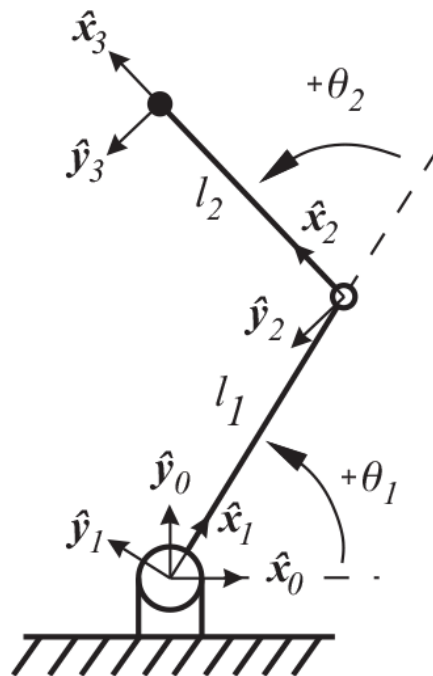
i. Compute the joint angle velocities required to execute an instantaneous velocity of 1 [m/s] in the \hat{x}_0 direction from any initial joint angle configuration—that is,

$$\dot{\theta} = \mathbf{J}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

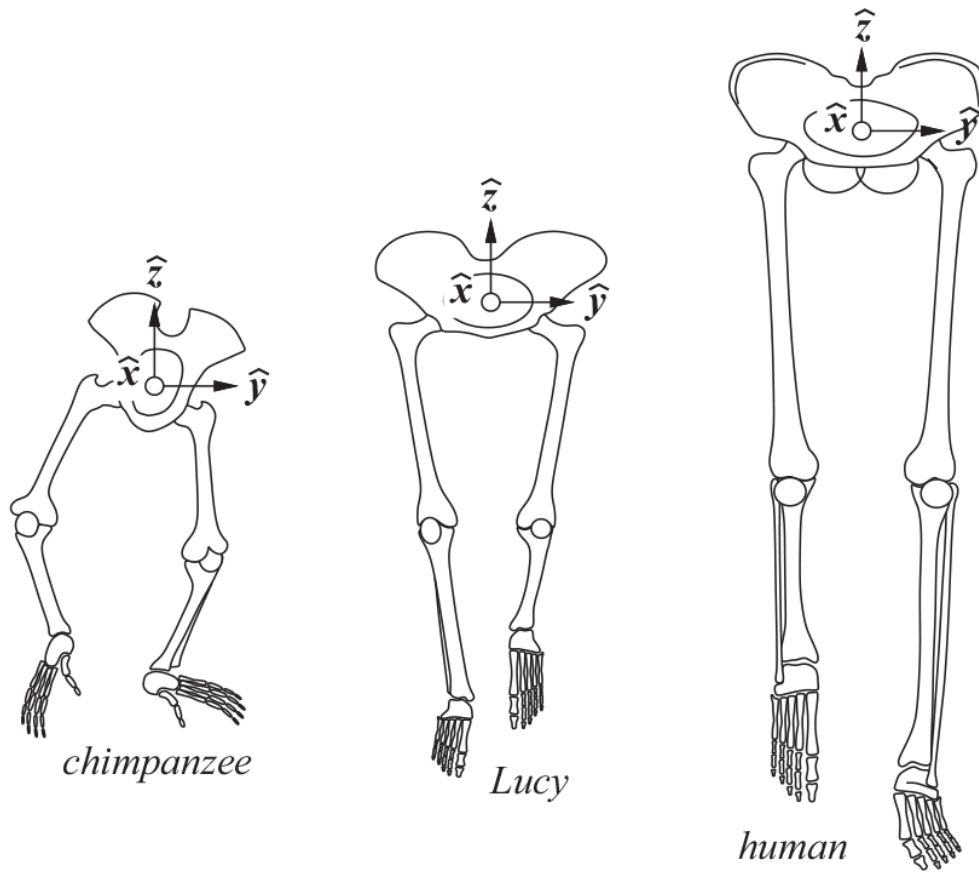
ii. Compute an expression for the torques on the joints of the manipulator necessary to apply 1 [N] endpoint force in the $-\hat{y}_0$ direction from any initial posture in the interior of the reachable workspace.

(c) Compute the eigenvalues and eigenvectors for $\mathbf{J}\mathbf{J}^T$ at $\theta_1 = \pi/4$, $\theta_2 = \pi/2$, assuming $l_1 = l_2 = 1$. Draw the velocity ellipsoid at this configuration.

8. **Forensic anthropology: Force and velocity ellipsoids.** “Lucy” (Australopithecus afarensis) was discovered in 1978 in Ethiopia by Mary Leakey. Lucy is a rare data point



concerning the emergence of contemporary humans in the fossil record. She is classified *Hominid*—an ape closely related to human beings—that lived around 3.75 million years ago. In terms of overall body and brain size, Lucy resembles a chimpanzee. However, her hip joint, knee, and pelvis indicate that she walked uprightlike a modern human. This suggests that human-like bipedalism may have preceded the development of the contemporary human hand and brain. Subsequently, the discovery of *Ardipithecus ramidus* (“Ardi”), dating back another 500,000 years before Lucy, supports this general hypothesis.



Although the skeletons of Lucy and the chimp are similar, the respective ranges of motion of hip and knee are quite different. The musculature, tendons, and ligaments of the chimp establish a relatively bent-leg and duck-footed posture that influences the quality of bipedal movements. This is clearly demonstrated in *Walking with Lucy* (<https://youtu.be/xT8Np0gI1dl>).

Based on the anatomical differences in the two animals, formulate representative configurations of the planar 2R manipulator to approximate a support leg midstride for the chimp and for Lucy in the sagittal (\hat{x} - \hat{z}) plane, and plot their respective force and velocity ellipsoids. Using this result, contrast the kinematic ability of Lucy's hip and knee with that of the chimpanzee and speculate about the kinds of habitat that could select for these skeletal variations.

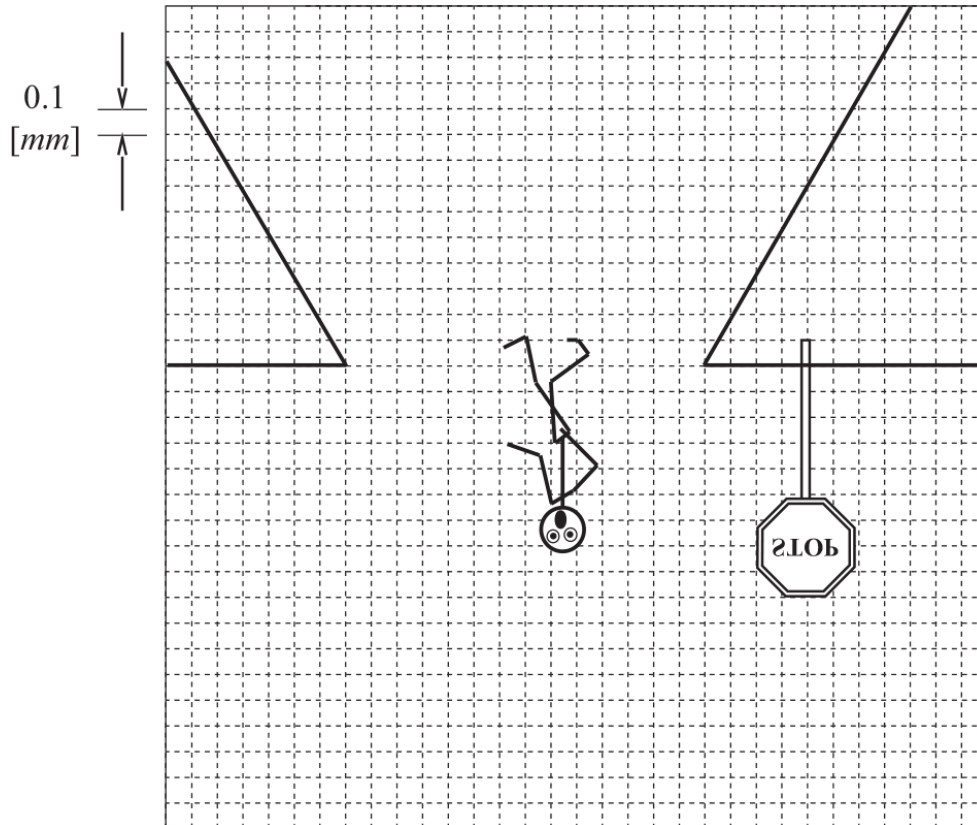
9. Pinhole camera.

This image is captured by a self-driving car going 20 [mph]. The roads are icy, and the vehicle estimates that it requires (conservatively) 35 [m] to

come to a controlled stop from this speed in these conditions.

Suppose that the stop sign is 2.5 [m] high and that the vehicle's forward-looking camera has a focal length $f=0.015$ [m].

- (a) Is it possible for the vehicle to stop in time?
- (b) Approximately how tall is the pedestrian?



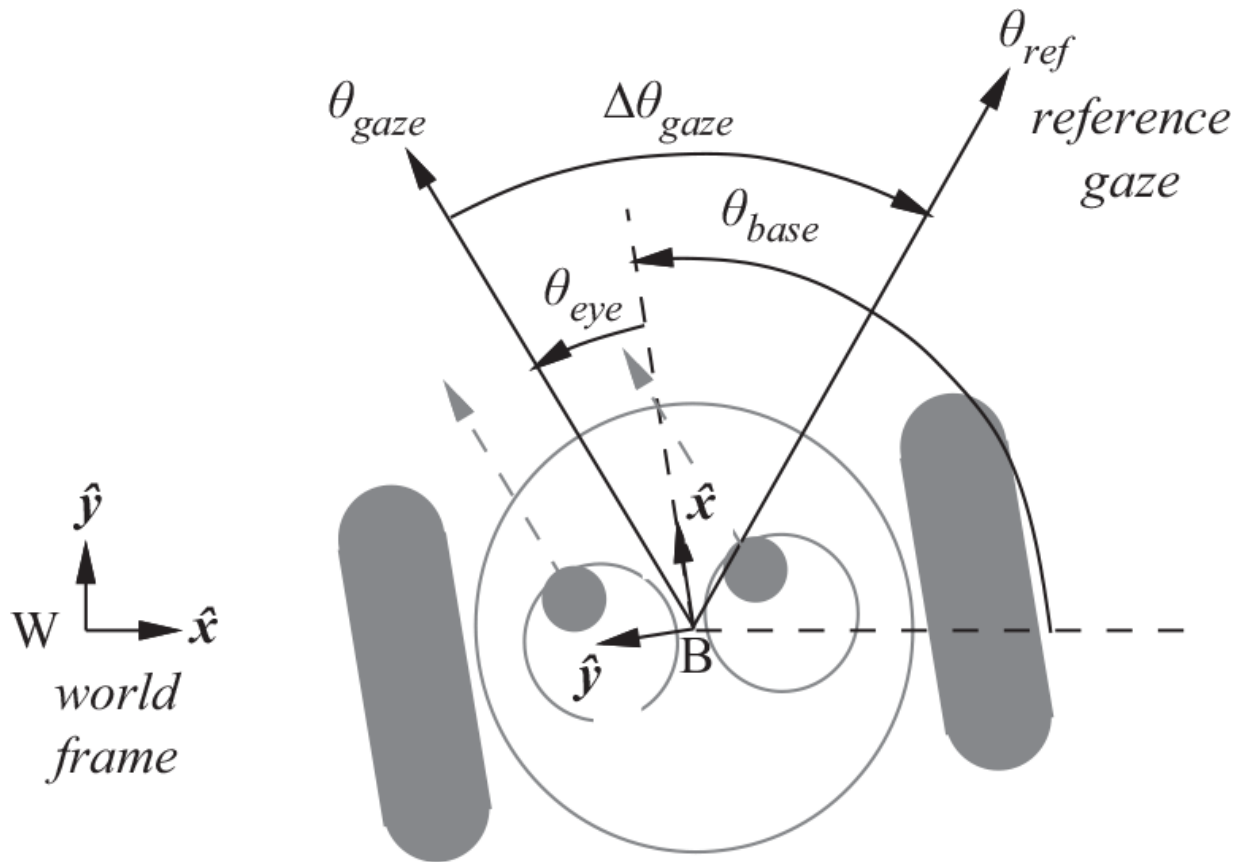
10. **Redundant gaze control for Roger-the-Crab.** Roger must direct its one-dimensional cycloptic gaze to a reference direction θ_{ref} by distributing the one-dimensional error $\Delta\theta_{gaze}$ over two controllable configuration variables

$$\mathbf{q} = [\theta_{base}, \theta_{eye}]^T, \text{ where } \theta_{gaze} = \theta_{base} + \theta_{eye}.$$

- (a) Use the pseudoinverse to find displacement

$$\Delta\mathbf{q} = \mathbf{J}^\# \Delta\theta_{gaze}$$

that minimizes the sum of squared displacements in the base and eye rotation.



(b) Describe how joint range limits in the eyes $-\pi/2 \leq \theta_{eye} \leq +\pi$ influence how you would implement such a solution.

(c) The base is more massive than the eyes. How would you modify this approach to minimize the total time or energy or the movement rather than the sum of squared displacements?

11. **Invent your own homework.** Write a question and a solution from the material in chapter 4. The problem should be different than the questions already here. Ideally, the solution should require a quantitative analysis and a conclusion. Good questions should not take more than 30 [min] to solve open-book.

1. Displacements are commonly determined by measuring wheel rotations, counting strides, or using inertial measurement units (IMUs).

2. A pixel (a picture element) is a single receptive field on the image plane. Pixels tile the image plane and, therefore, define a unit of length.
3. *Holo-* (in Greek ὅλος) means “whole” and *-nomic* (from νόμος) means “law.”
4. [Figure 4.16](#) was constructed by scanning a discretized configuration space for the limb in question and using forward kinematic functions to index into a Cartesian map. The color of the Cartesian map at that location is copied to the corresponding configuration space bin. Small asymmetries in the figure are artifacts of the discrete model and the projection process.

5

Hands and Kinematic Grasp Analysis

Dexterity, as defined by Bernstein [23], is the bridge between a physical and a cognitive relationship with the world. It is exemplified by the deft interaction between hands (specialized parallel kinematic chains) and the environment, where multiple kinds of contact interactions are supported and from which feedback conveys information about objects, materials, and forces.

5.1 The Human Hand

Hands are levers of influence on the world that made intelligence worth having. Precision hands and precision intelligence coevolved in the human lineage, and the fossil record shows that hands led the way.

—Steven Pinker, *How the Mind Works* [230]

Several books have been written on the human hand [21, 279, 210], underscoring its importance in human behavior. In a book entitled *The Hand* [293], Frank Wilson discusses the impact of the hand on the evolution of the human brain. Wilson's argument tracks human evolution through the oldest known primates (the Paleocene mammals) that were about the size of a squirrel and had developed binocular vision. Not long after that on an evolutionary time scale, there followed a modified forearm and collar bone that increased the range of motion and mobility of the wrist and arm; five ray (pentadactyl) paws with individually prehensile fingers and toes (but not yet an opposable thumb); and nails instead of claws. Most of these adaptations were in service to life in the trees. In addition, jaws, skull, and

teeth changed in response to a changing diet, and brain size increased. Napier [210] proposed that the four-footed climbing behavior of the smaller primates (like a squirrel) was supplanted by an inverted brachiation style of locomotion, where the body of the animal swings under a supporting grasp on a tree limb. Brachiation requires that the bones of the arm and wrist are in tension, and greater mobility is required in the shoulder to swing freely.

As these primates moved onto the ground, two important kinematic modifications ensued as part of a gradual co-evolution of the brain and musculoskeletal system toward *Homo sapiens*: (1) a bipedal gait and (2) modifications in the arm and hand toward tool use. Most primates have four fingers and one thumb that is somewhat shorter and more mobile than the fingers. Chimpanzees stand out as accomplished tool users even though their hands are less similar to the human hand than they are to those of the ape. What chimps and humans do have in common is extra mobility in the wrist where it meets the ulna (figure 5.1), which effectively increases the yaw range of motion of the hand relative to the forearm (figure 4.10). Lucy (*Australopithecus afarensis*) is one of the few examples of a relatively complete skeleton for a prehuman hominid in the fossil record. Anthropologists propose that Lucy's most valuable new functionality was her ability to hold, pound, and throw stones. In contrast to chimps, which can only lob objects underhand, her pelvis, gluteus maximus, shoulder (from brachiation), rotating forearm, and improved grip enhanced her ability to fight with her arms and throw projectiles overhand at high velocity and with precision. Lucy no longer had to depend on her jaws and teeth as the primary means of attack and defense [187].

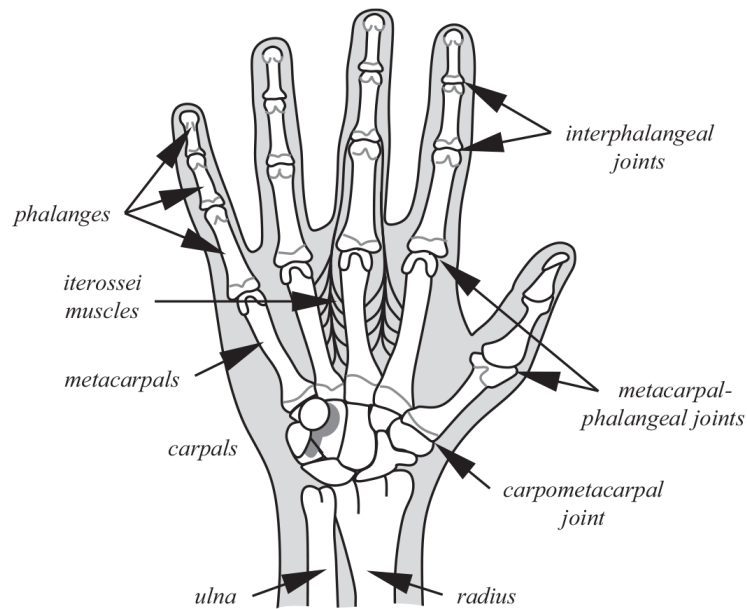


Figure 5.1

A palm view of the right hand shows the finger's intrinsic iterossei muscles containing as many as 120 muscle spindles per gram of tissue

Lucy had a remarkably humanlike lower body (see exercise 8 in section 4.9). However, her hand was more like an ape's than a modern human's, with a relatively short and immobile thumb unable to reach completely across the palm. [Figure 5.1](#) illustrates some of the special-purpose skeletal and intrinsic muscular adaptations of our hands since Lucy that make it such a useful device. Each of the four fingers of the human hand consists of three joints, the most proximal of which is called the metacarpal-phalangeal joint [173]. It has two degrees of freedom and is capable of adduction/abduction over a range of approximately 30 [deg], as well as flexion/extension of approximately 120 [deg]. The next two joints of the human finger are the interphalangeal joints. These are one degree of freedom hinge joints with a range of motion of approximately 90 [deg].

The human thumb is a more complex mechanism. It is relatively long and can reach fingertip oppositions with each of the four fingers. Altogether, nine muscles actuate the thumb: four extrinsic muscles that originate in the forearm and five more intrinsic muscles. The proximal joint is called the carpometacarpal joint and contains two degrees of freedom. The first is an adduction/abduction with a range of motion of approximately 90 [deg]. The axis of this rotation is skewed somewhat from the plane

defined by the fingers. The single most powerful muscle in the hand originates in the forearm and flexes the carpometacarpal joint of the thumb toward the other fingers. Some of the smaller intrinsic muscles configure the thumb in opposition to the fingers so that this strong flexor can apply the forces that squeeze objects between the fingers and the thumb. The next joint of the thumb is the metacarpal-phalangeal joint, which can be extended to place the thumb in the plane of the palm or flexed so that the tip of the thumb approaches the palm through a displacement of about 60 [deg]. The last (distal) joint of the thumb is a single degree of freedom hinge joint called the interphalangeal joint with a range of motion of about 90 [deg].

The cumulative evolutionary innovations that led to human dexterity—the collar bone, fingernails, shoulder and wrist mobility, pelvic adaptations and the bipedal gait, fingertip-thumb oppositions—were driven by competitive pressure and have produced the most dexterous primate hand in the fossil record. Lucy’s modern pelvis preceded a modern hand, and both emerged before significant modification to a chimp-like brain. It is, therefore, reasonable to suppose that all of these new interacting changes toward bipedalism and bimanual dexterity called for a new kind of brain that supported the ability to acquire and differentiate materials, to fabricate and use tools, and to apply these skills to solve problems related to survival and reproduction. All of this evolutionary development accommodates a specific range of environmental mass, geometry, and force—all determined by the arrangement of bones and muscles in the limbs—and supports strategies for grasping and manipulation that depend on the ability to control *internal* (isometric) loads in the skeleton.

The movement of hands has been analyzed by physiologists [139, 211] and engineers using time and motion studies [68, 67]. These studies often characterize the posture of the hand in terms of its capacity to deliver power and precision. Power grasps envelop the object between the fingers and the palm in a manner that firmly connects hand and object. Precision grasps hold the object between the tips of the fingers and the thumb to more fully control the mobility of the object relative to the palm. For example, one such taxonomy created by Mark Cutkosky [68] (figure 5.2) was produced from observations of grasps used by machinists in a manufacturing setting. At the root of the taxonomy is the distinction between precision and power,

but these are further resolved into seven types of precision grasps and nine types of power grasps. Power grasps are further delineated into prehensile grasps that use hand forces exclusively and nonprehensile grasps that depend on gravitational (or inertial) forces as well. Figure 5.2 reveals a strong preference to exploit the dominant spherical and cylindrical symmetries of the hand—a direct product of our evolutionary history and a preference that is also evident in many objects engineered for use by human hands.

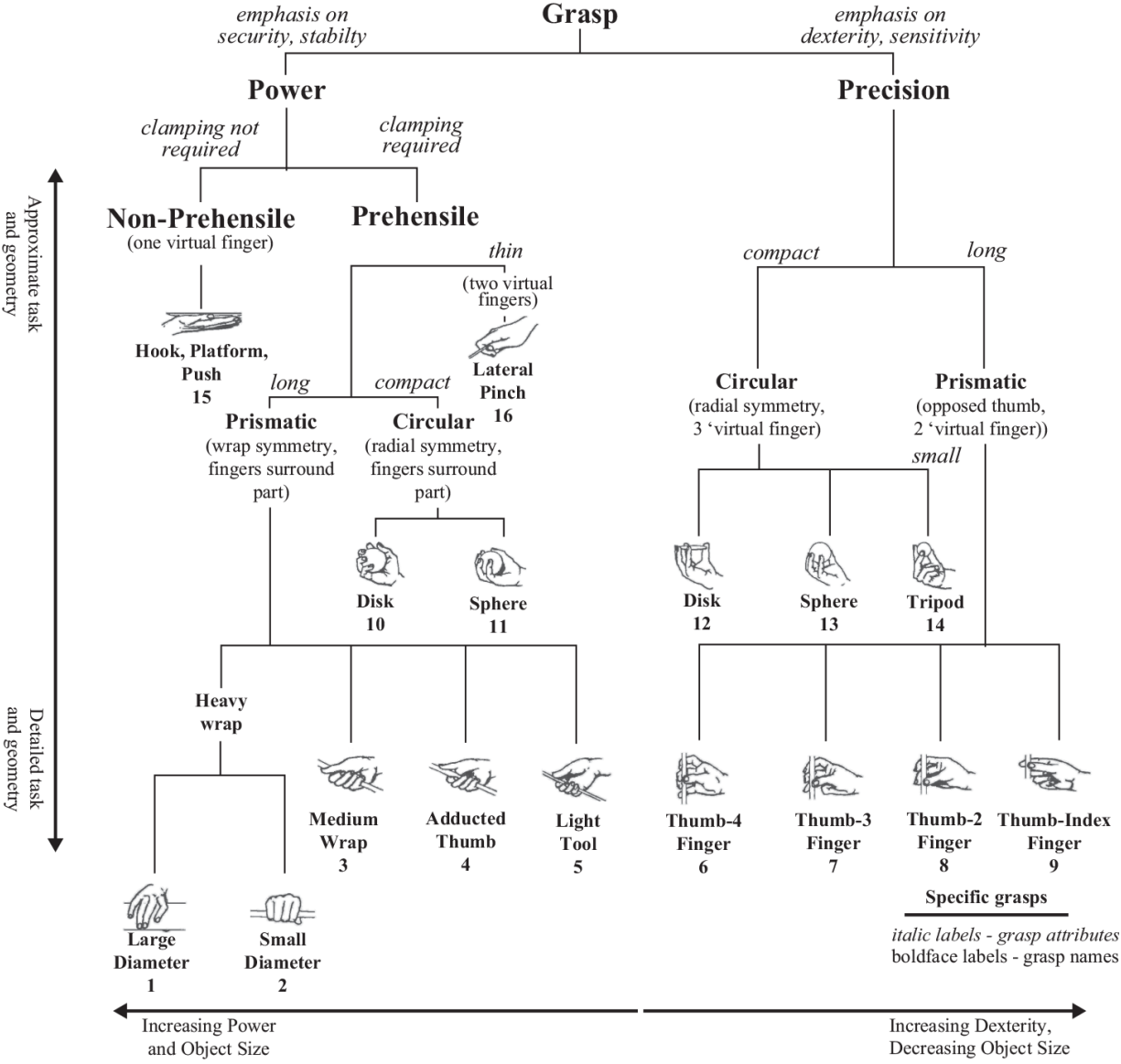


Figure 5.2 Cutkosky’s taxonomy of manufacturing grasps (Source: [68]. Reprinted with permission.)

5.2 Kinematic Innovations in Robot Hands

Subtle innovations in the geometry, actuation, and sensory abilities of the primate hand supported new ways of interacting with objects. Along with changes in the pelvis, oral-facial structure, and the larynx, a new brain evolved in *Homo sapiens* capable of organizing all of this new information. This extended period of coevolution is responsible for many of the traits we recognize as distinctively human and intelligent. It may be the case that robot systems are undergoing such a critical stage of evolution right now. If this is true, then the incremental refinements in robot hands over the last 50 years will likely have played an important role.

In 1961, Heinrich Ernst [79] demonstrated what may have been the first computer-controlled mechanical hand that found and manipulated wooden blocks using sensor feedback (figure 5.3). The hand had a total of 30 joint angle and tactile sensors that measured contact pressure. Ernst demonstrated a tactile probing process and a procedure by which blocks were grasped and stacked.

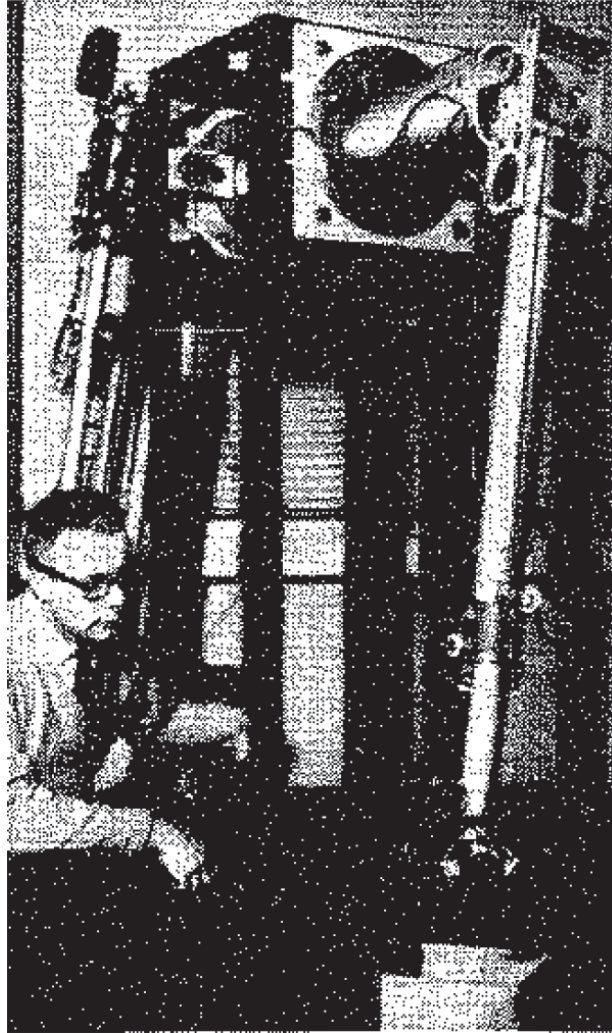


Figure 5.3

A newspaper photo of the computer-controlled mechanical hand built by Heinrich Ernst at MIT under the direction of Claude Shannon

In the early 1970s, a robot named Freddy ([figure 5.4](#)) was built at the University of Edinburgh. The research team demonstrated that automated assembly could be accomplished using an instrumented manipulator. A set of wooden parts for a toy car and ship were placed randomly in Freddy's workspace, and the robot localized and assembled them. Computer vision was used to separate light-colored parts from a darker background, and two views were used to recognize and localize the parts. Parts of the scene that were not recognized were assumed to be piles, and Freddy would grasp and move features of the pile to separate them. If grasps didn't work, Freddy would plow through the pile to disturb it and try again. After all the parts

required were detected and cataloged, force feedback was used to execute an assembly sequence for the toy.

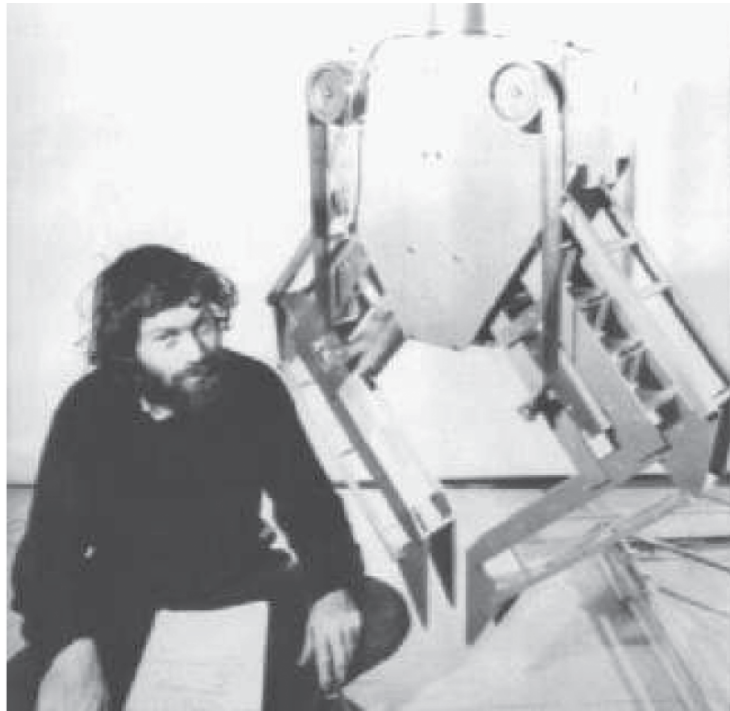


Figure 5.4

Freddy was designed and built by a team of researchers at the University of Edinburgh: Pat Ambler, Harry Barrow, Chris Brown, Rod Burstall, Gregan Crawford, Robin Popplestone (pictured), and Stephen Salter

These landmark experiments used a *parallel jaw gripper*—the oldest and most commonly used type of robot hand. It consists of a single degree of freedom mechanism that opens and closes a pair of parallel (or pivoting) jaws and produces a clamping motion. Because it is so simple and mechanically robust, many industrial applications still rely on this technology. Often, the presentation of the part, or the geometry of the part and/or the fingers, can be customized for the automated task. There are estimates that 80 percent of automated industrial assembly tasks can be accomplished using these relatively simple devices [105].

However, whenever the significant structure of the manufacturing line is not available or the manipulator must accommodate many objects and tasks, these simple hands can be less effective. There is little (or no) remaining mobility in the grasped object relative to the hand with the parallel jaw

gripper, so fine assembly motions must be accomplished using the larger, heavier degrees of freedom in the arm. As a consequence, in some task geometries, the performance of the system can be sensitive to limitations in motor precision and errors in sensor feedback that cannot be corrected using fine motor control in the hand.

In 1977, Hanafusa and Asada [101] introduced a concept for a planar three-fingered hand design to improve robustness to uncertainty and fine motor control for grasped objects. Figure 5.5 is a schematic of the *Hanafusa hand* gripping an object. Three single degree of freedom fingers are positioned symmetrically, 120 [deg] apart, around the object to form a planar grasp. Contacts are actuated radially by stepper motors through coil springs. The combination of a force source in series with a spring provides passive elasticity and force sensing. The deflection of the spring (and therefore the contact normal force) is measured by a potentiometer inside the coil spring, and a contact roller eliminates the tangential components of the grip force. To grasp a planar shape, a vision system was used to extract the object silhouette [36], and an equilibrium grasp was selected that minimized the potential energy in the springs $\int F \cdot ds$, where F is the force generated by spring as it deflects and ds is the differential deflection in the springs as they conform to the shape of the object. This approach is robust to sensory and motor imprecision—small errors are rejected by the passive elasticity of the hand. Moreover, the pose of the grasped object remains controllable to a limited degree by way of the stepper motor inputs.

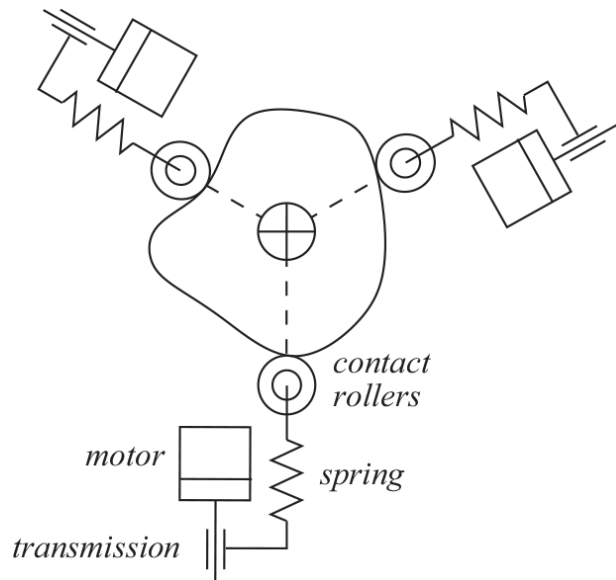


Figure 5.5

The Hanafusa manipulator. (Source: Adapted from [102])

To improve fine motor control, new designs emerged using multiple degree of freedom fingers to control contact position and elasticity in \mathbb{R}^3 . Many of these designs adopted anthropomorphic geometries with multiple fingers and an opposable thumb. For example, in 1979, Okada [216, 217] developed a five degree of freedom arm-wrist combination coupled with a three-fingered, eleven degree of freedom hand. The *Okada hand* (figure 5.6) was more anthropomorphic than its predecessors with four degrees of freedom in the fingers and three degrees of freedom in the thumb. It could, therefore, implement common human grasping primitives such as wrapping and pinching, and the extra fingertip mobility supported active tactile exploration. This more ambitious kinematic design required innovations in packaging to minimize weight and simplify the routing of electrical wires for power and communication. Okada placed the motors in the trunk of the robot and actuated the hand through wire-sheathed cables approximately 1.7 [m] long that introduced significant additional friction and elasticity. Hybrid position-torque controls were implemented in hardware for the fingers that combined position control and force control over opposing pairs of contacts to simultaneously control the position of a grasped object and the magnitude of internal grasp forces. The hand could grasp and hold objects up to 5 [N] in weight and demonstrated the ability to tighten nuts on

a threaded shaft, to move a grasped object from side to side, to twirl a baton, and to rotate a grasped ball.

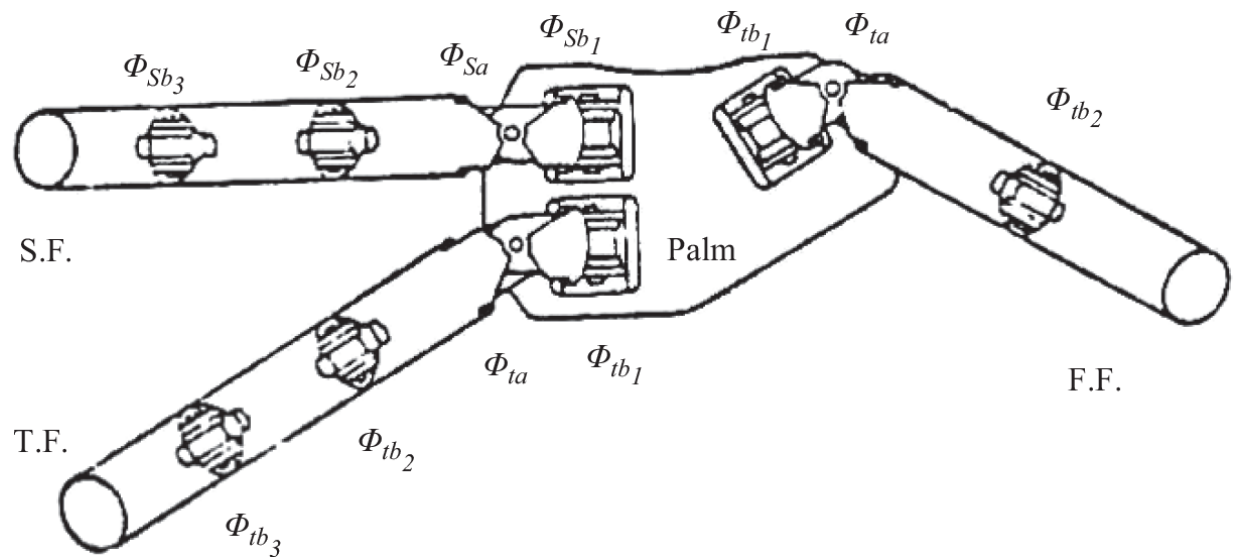


Figure 5.6

The Okada manipulator (Source: [217]. ©IEEE. Reprinted with permission.)

Advances in packaging, actuators, and control led to enhanced mobility of the grasped object, but this functional advantage came at the expense of mechanical complexity. These designs could not match the mechanical robustness of the simpler parallel jaw manipulator. The steady and aggressive development of new actuators, tendons, sensors, and control for more reliable and dexterous hands has been a focus ever since.

In the 1980s, many roboticists focused on dexterity and considered how optimization techniques could be applied directly to improve hand function independent of any particular task. For example, Lian et al. [173] developed an anthropomorphically motivated hand designed to reproduce several important human movements and postures including multiple types of opposition (fingertip, lateral, and digito-palmar) and palmar (cylindrical) and spherical power grasps. The design of the *Lian manipulator* (figure 5.7) employed graphical simulations of these grasps using a parametric kinematic model of the hand to explore combinations of parameters that supported these movements. The resulting three-fingered hand incorporated three degree of freedom fingers with two parallel distal interphalangeal joints and an orthogonal axis in the proximal (metacarpal-phalangeal) joint

—the former flexing and extending in the plane and the latter operating the finger in adduction/abduction. The thumb is similar except that the proximal joint is inclined at 60 [deg] to the palm, while the fingers are perpendicular to the palm. As in its predecessors, the Lian hand used DC electric motors and sheathed stainless steel cables, and grasping forces were determined by measuring cable tensions and joint positions.

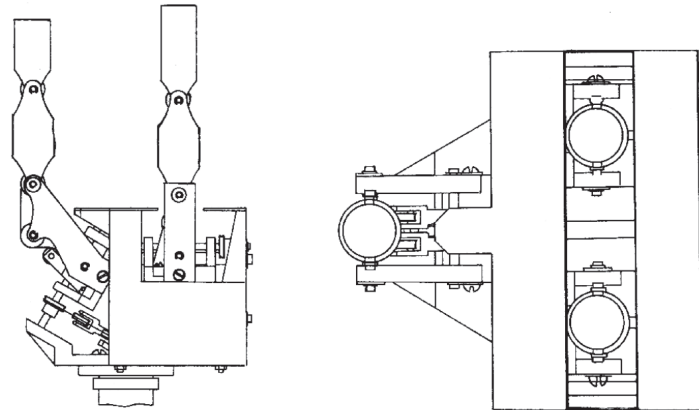


Figure 5.7

The Lian, Peterson, Donath manipulator. (Source: [173])

Rather than emulating human grasp postures, the design of the *Stanford/JPL hand* in figure 5.8 employed number synthesis [220] to analyze the mobility and connectivity of alternative hand designs and ranked them using quantitative performance criteria in relation to a reference object. This work systematically enumerated candidate designs with up to three fingers and up to three degrees of freedom per finger. In addition, a range of values for continuous parameters defining the position of fingers around the palm and link lengths were explored. Each candidate was scored with respect to the resulting range of motion and the anticipated fingertip contact geometry on a reference object to optimize the hand geometry [252, 249].

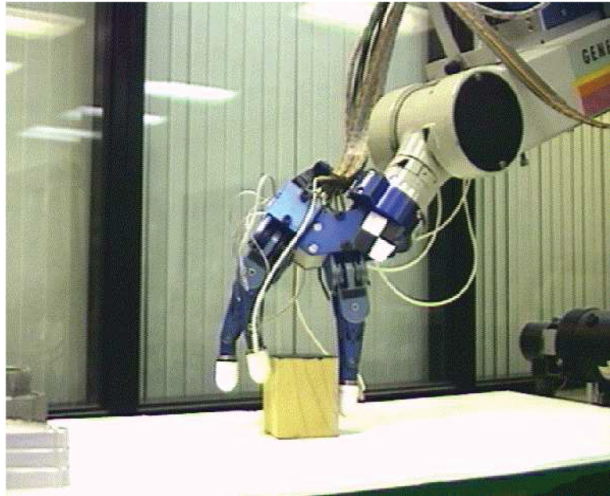


Figure 5.8

The nine degree of freedom Stanford/JPL hand [250] equipped with fingertip load cells on the end of a five degree of freedom GE P50 robot arm

Simulations were used to evaluate over 600 candidate designs. Kinematic parameters were chosen to minimize the condition number (appendix A.8) of the finger Jacobians on the surface of a 1 [in] sphere grasped over the palm. The design also considered features aimed at improving the ability to perform power grasps.

The resulting hand design incorporated three fingers, each with three degrees of freedom (figure 5.8). It was actuated using an $N + 1$ tendon scheme, where four antagonistic tendons control three joints in each finger. Teflon coated tendons were routed through flexible conduit and driven by 12 motors and gear trains mounted in the forearm. A position control loop used measurements of tendon tension near the fingertip, motor position, and velocity to estimate joint positions and velocities. The effects of friction and elasticity were mitigated by sensing tendon tension close to the point of application. The hand was fitted with hemispherical fingertip load cells to measure contact forces and positions accurately.

Kinematic innovation in robot hands has continued since these early concepts. In addition, new materials, sensor and actuator technologies, and powerful embedded computing platforms have succeeded in creating smaller, lighter, faster robots, and this integrated approach is rapidly advancing the state of the art in dexterous and anthropomorphic robot hands. For example, the Deutsches Zentrum für Luft- und Raumfahrt

(DLR) designed a series of hands for use in space applications. The *DLR II hand* in [figure 5.9](#) consists of four identical fingers, each with four joints and three independent degrees of freedom. The medial and distal joints are coupled as in human fingers due to cross-joint tendon coupling. An additional palm degree of freedom is incorporated for a total of 13 DOF. The hand is actuated by remotely located electric motors capable of forces at the fingertip of about 30 [N], and the hand itself (without actuators) weighs approximately 1.8 [kg].



Figure 5.9

The DLR Hand II. (Source: Courtesy of DLR, Institute of Robotics and Mechatronics)

5.3 Mathematical Description of Multiple Contact Systems

Thus far, this chapter has focused on kinematic innovations in hand design that supported important functions in primates (including hominids) and that led to better hands for robots. In this section, the kinematic basis of contact systems is introduced to describe how contacts transmit generalized forces and velocities through environmental surfaces. We will focus on multiple contact systems that can be used to stabilize an object that is held within a grasp. To begin, a brief introduction to *screw theory* is presented and used to examine how contacts restrict the free motion of an object. A variety of *contact types* is presented and incorporated into a systematic procedure for constructing the *grasp Jacobian*.¹ Following this introduction, we discuss *form-* and *force closure* conditions that support prehensile grasping. The chapter ends with an example demonstrating how grasp force

solutions are determined given a contact configuration and a manipulation *task*.

5.3.1 Screw Systems

Screw theory is a useful tool for kinematic and static analysis. As a consequence, it is often used in research concerning contact and grasping. The first concept we consider from screw theory is that of the *twist*.

Definition 5.1: Twist. *A generalized velocity $\mathbf{v}=[\mathbf{v} \ \boldsymbol{\omega}]^T$, where $\mathbf{v} \in \mathbb{R}^6$ represents translational velocity $\mathbf{v} \in \mathbb{R}^3$ concatenated with angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$.*

In the case of contacts on an object, a twist can be used to describe the velocities that the object can move unimpeded by a contact constraint.

Example: Twist Constraints on Object Mobility in a Planar Grasp

Following similar analyses [102, 251, 82, 80, 81, 148, 149], let \mathbf{V} describe the set of object twists that are not restricted by contact constraints. Let the complement $\bar{\mathbf{v}}$ describe the set of object twists restricted by the contact system. Given this framework, $\mathbf{v} \cup \bar{\mathbf{v}} = \mathbb{R}^6$ and $\mathbf{v} \cap \bar{\mathbf{v}} = \emptyset$. To completely immobilize an object using a system of n contacts, we require that

$$\{\mathbf{v}_1 \cap \mathbf{v}_2 \cap \dots \cap \mathbf{v}_n\} = \{\emptyset\}, \text{ and} \quad (5.1)$$

$$\text{span}\{\bar{\mathbf{v}}_1 \cup \bar{\mathbf{v}}_2 \cup \dots \cup \bar{\mathbf{v}}_n\} = \mathbb{R}^6, \quad (5.2)$$

where \mathbf{v}_i and $\bar{\mathbf{v}}_i$ denote elements of \mathbf{V} and $\bar{\mathbf{v}}$, respectively, derived from contact i .

Figure 5.10 illustrates a simple hand-object system that we will use to consider the influence of multiple contact twist constraints on the mobility of a grasped object [148, 149]. In this simple planar system, the object frame remains parallel to the world frame (no object rotation is permitted), and neither hand nor object can translate out of the plane (in the \hat{z} direction). Therefore, in this case we consider object twists $\mathbf{v}_O = [v_x \ v_y]^T \in \mathbb{R}^2$. The elements labeled 1, 2, and 3 are linear actuators whose length can be controlled. When assembled into this hand-object system the long axis of

the actuators define the active (controllable) subset of the \hat{x} - \hat{y} plane for each actuator. In the lateral direction, the actuators are passive—free to slide along unactuated and frictionless prismatic joints. For now, we assume that contacts on the square object are Velcro and can push and pull the object.

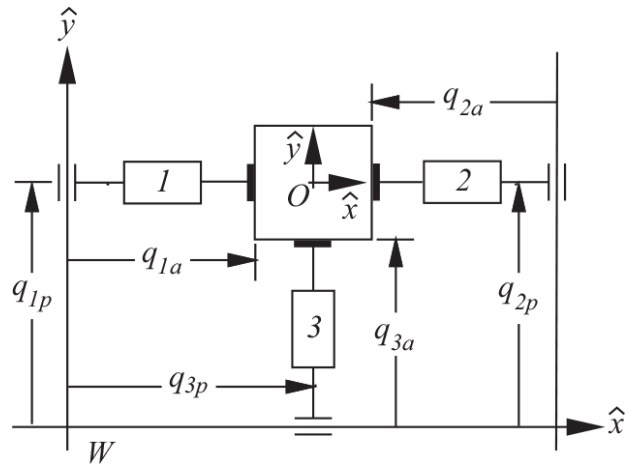


Figure 5.10

A 2D Cartesian hand-object system. (Source: Adapted from Kobayashi [148]. © MIT Press)

Considering the contact geometry of finger 1 alone, we find that an object twist $[v_x \ v_y]^T_O$ corresponds to a combination of active and passive velocities in the hand mechanism:

$$\begin{aligned} \text{finger 1: } \begin{bmatrix} v_x \\ v_y \end{bmatrix}_O &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}_O \dot{q}_{1a} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}_O \dot{q}_{1p} \\ &= \bar{\mathbf{v}}_1 \dot{q}_{1a} + \mathbf{v}_1 \dot{q}_{1p}. \end{aligned} \quad (5.3)$$

Active joints can be moved (and locked) so they contribute to set \bar{V} describing how the object's pose is restricted (and, thus, controllable) in the hand. Passive joints contribute to set V , which represents object degrees of freedom that are unrestricted by the contact. Writing this relation for fingers 2 and 3 yields

$$\begin{aligned} \text{finger 2: } \begin{bmatrix} v_x \\ v_y \end{bmatrix}_O &= \begin{bmatrix} -1 \\ 0 \end{bmatrix}_O \dot{q}_{2a} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}_O \dot{q}_{2p} \\ &= \bar{\mathbf{v}}_2 \dot{q}_{2a} + \mathbf{v}_2 \dot{q}_{2p} \end{aligned} \quad (5.4)$$

$$\begin{aligned} \text{finger 3: } \begin{bmatrix} v_x \\ v_y \end{bmatrix}_O &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}_O \dot{q}_{3a} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}_O \dot{q}_{3p} \\ &= \bar{\mathbf{v}}_3 \dot{q}_{3a} + \mathbf{v}_3 \dot{q}_{3p}. \end{aligned} \quad (5.5)$$

Equation 5.1 can be used to evaluate the impact of different combinations of the contacts on the resulting mobility of the object. For example, with just fingers 1 and 2,

$$V = \bigcap_{i=1}^2 \mathbf{v}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Therefore, fingers 1 and 2 alone do not fully immobilize the object—it can move uncontrollably in response to perturbations in the \hat{y} direction. However, for all three fingers,

$$V = \bigcap_{i=1}^3 \mathbf{v}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \emptyset,$$

from which we conclude that when the three contacts are fixed in place by the actuators and the object does not break contacts, the object is fully constrained. Furthermore, the space of controllable object velocities using this contact configuration is the union of the motion constraints defined by the active degrees of freedom.

$$\bar{V} = \bigcup_{i=1}^3 \bar{\mathbf{v}}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cup \begin{bmatrix} -1 \\ 0 \end{bmatrix} \cup \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbb{R}^2$$

We conclude that this planar contact geometry (with Velcro contacts) will fully immobilize the object when actuators are fixed in place and that the object position in the (x, y) plane is fully controllable. □

5.3.2 The Grasp Jacobian

The Velcro contacts in figure 5.10 do not permit any relative movement between the contact pad and the object. This analysis can be generalized to account for other hand mechanisms, object surface geometries, and contact

types by eliminating the Velcro pads and replacing them with independent contact coordinate frames aligned with the local surface normal as in [figure 5.11](#). By convention, we designate the \hat{z}_i axis of the contact frame to be aligned with the inward surface normal. The \hat{x}_i axis is chosen to be the surface tangent to the contact surface in the same direction as positive \dot{q}_{ip} in the planar mechanism. The \hat{y}_i axis is determined to produce a right-handed *contact frame* C_i , $i=1, 3$.

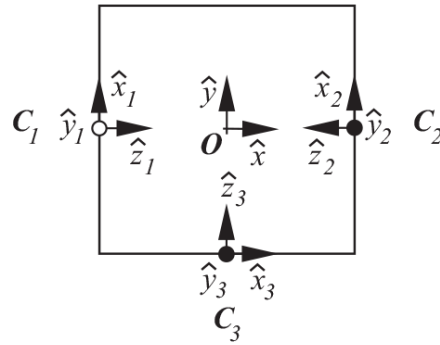


Figure 5.11

The object frame O and the local contact frames C_i , $i=1, 3$ oriented with respect to the local surface normal

In section 5.3.3, we will see that, in addition to Velcro, there are several other contact types that can transmit different combinations of forces and moments that are defined conveniently in the local contact frame. However, for the case of the Velcro contacts

proposed in [figure 5.10](#), using independent forces that act along the local \hat{x}_i and \hat{z}_i axes at each contact in [figure 5.11](#), we can write a linear expression that maps two-dimensional object twists $\mathbf{v}_O = [v_x \ v_y]^T$ into contact frame twists $\mathbf{v}_C = [v_{1x} \ v_{1z} \ v_{2x} \ v_{2z} \ v_{3x} \ v_{3z}]^T$. For the planar grasp in [figure 5.11](#),

$$\begin{bmatrix} v_{1x} \\ v_{1z} \\ v_{2x} \\ v_{2z} \\ v_{3x} \\ v_{3z} \end{bmatrix}_C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}_O \quad (5.6)$$

or

$$\mathbf{v}_C = \mathbf{G}^T \mathbf{v}_O, \quad (5.7)$$

where $\mathbf{G} = [\mathbf{v}_1 \bar{\mathbf{v}}_1 \mathbf{v}_2 \bar{\mathbf{v}}_2 \mathbf{v}_3 \bar{\mathbf{v}}_3]$. Matrix \mathbf{G} is called the grip transform or the grip matrix [188], the grasp Jacobian [251], or the grasp matrix [235]). Assuming that none of the contacts are broken when the object moves, \mathbf{G}^T maps object velocity \mathbf{v}_O to the corresponding velocity of each contact in the contact coordinate frame \mathbf{v}_C . In the following, we adopt the term *grasp Jacobian* for matrix \mathbf{G} to emphasize the relationship to other kinematic Jacobian mappings.

When a contact restricts the movement of the object, it does so by applying forces through the contacting surfaces.

Definition 5.2: Wrench. *A generalized force $\mathbf{w} = [\mathbf{f} \ \mathbf{m}]^T$, where $\mathbf{w} \in \mathbb{R}^6$ is a force $\mathbf{f} \in \mathbb{R}^3$ concatenated with a moment $\mathbf{m} \in \mathbb{R}^3$.*

A wrench can be used to restrict the free motion of the object. Therefore, twists and wrenches form a straightforward, complementary account of the quality and controllability of a contact interface. Moreover, the screw representation for these quantities leads to a concise expression for the **power** transmitted from the contact to the object.

Definition 5.3: Power. *The product of a wrench and a twist $\mathbf{w}^T \mathbf{v}$ [Nm/s] is the rate of work produced or power.*

If there is no dissipative loss, then the net power transmitted to the object is equivalent to the power produced by the contact,

$$\mathbf{w}_O^T \mathbf{v}_O = \mathbf{w}_C^T \mathbf{v}_C.$$

Since $\mathbf{v}_C = \mathbf{G}^T \mathbf{v}_O$ (equation 5.7), we can write

$$\mathbf{w}_O^T \mathbf{v}_O = \mathbf{w}_C^T [\mathbf{G}^T \mathbf{v}_O],$$

so that

$$\begin{aligned} \mathbf{w}_O^T &= \mathbf{w}_C^T \mathbf{G}^T, \text{ or} \\ \mathbf{w}_O &= \mathbf{G} \mathbf{w}_C. \end{aligned} \quad (5.8)$$

Like the manipulator Jacobian, grasp Jacobian \mathbf{G} captures instantaneous velocity and force mappings from contact coordinates to object coordinates, and it changes if contacts roll, slip, or are broken. Moreover, the performance of a grasp depends on factors (internal loads, contact friction) that are necessary to maintain the grasp geometry.

By [equation 5.8](#), the grasp Jacobian for the planar velocity mapping in [figure 5.11](#) also applies to the transformation from the contact loads $\boldsymbol{\omega}_C = [f_{1x} f_{1z} f_{2x} f_{2z} f_{3x} f_{3z}]^T$ to the net wrench applied to the object $\boldsymbol{\omega}_O = [f_x f_y]^T$.

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix}_O = \begin{bmatrix} 0 & 1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{1x} \\ f_{1z} \\ f_{2x} \\ f_{2z} \\ f_{3x} \\ f_{3z} \end{bmatrix}_C \quad (5.9)$$

In general, the number of rows in the grasp Jacobian is 6 to represent $\boldsymbol{\omega}_O \in \mathbb{R}^6$, and the number of columns can be as large as $6N$, where N is the number of contacts. In practice, the number of columns is determined by the number of contacts and the *contact type*.

5.3.3 Contact Types

The contact coordinate systems in [figure 5.11](#) can represent more general types of contact forces than the Velcro contacts we started with—even contacts that can push but cannot pull. [Table 5.1](#) lists a variety of contact types distinguished by the geometry and material properties of the contacting surfaces. The least restrictive of these is the frictionless point contact (bottom of the table), which constrains (compressive) translations along the contact normal but preserves five other free/unrestricted degrees of freedom between the pair of bodies.

Table 5.1
Contact types and the degree of freedom remaining at the contact interface

Remaining DOF	Type of contact
0	Planar contact with Velcro/glue
1	Line contact with friction
2	Soft finger

Remaining DOF	Type of contact
3	Point contact with friction; plane contact without friction
4	Line contact without friction
5	Point contact without friction

Source: Adapted from Salisbury [250]. © Stanford University Department of Mechanical Engineering.

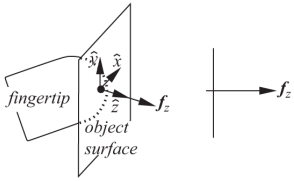
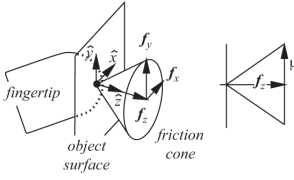
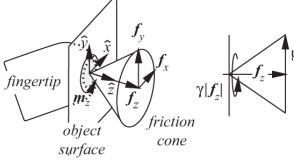
Moving from the bottom to the top of the table, each more elaborate contact type removes additional degrees of freedom until the planar Velcro/glue contact permits no relative motion between the contact and the object. Combinations of contacts like those in [table 5.1](#) form contact systems in which object motion is restricted by the union of constraints applied at each contact.

In order to reduce the complexity of grasp planning, Iberall [126] observed that the human hand often exploits oppositions between the thumb and the fingers and chose to represent a set of fingertip contacts as a single *virtual* finger with an enhanced contact type. For example, a pair of discrete fingertip point contacts without friction on a plane create a virtual line contact without friction on the plane. [Table 5.1](#) indicates that this pair of contacts reduces the relative mobility of the object to four degrees of freedom.

[Table 5.2](#) describes three common contact types (a subset of those in [table 5.1](#)) using the local contact coordinate frame. In keeping with the conventions used in [202, 138], the contact wrench space supported by each contact type is defined by a set of wrenches in the column space of the selection matrix \mathbf{H}^T . These wrenches are due to independent unit magnitude forces or moments acting along the axes of the contact coordinate frame and scaled by *effort* variables λ . This representation makes it straightforward to write contact maintenance constraint inequalities in the fourth column of [table 5.2](#). A frictionless point contact, for example, transmits a force along the surface normal. To maintain compression at the contact (and thus maintain contact), λ_{fz} must be greater than or equal to zero. Likewise, the point contact with friction requires the same positive (compressive) normal force along the \hat{z} direction but also supports frictional forces tangent to the contact surface (i.e., in the \hat{x} and \hat{y} directions). These forces may not exceed limits imposed by a Coulomb friction model that

requires that the magnitude of the net tangential force $[\lambda_{fx}^2 + \lambda_{fy}^2]^{1/2}$ does not exceed the product of the coefficient of friction and the magnitude of the normal force, $\mu\lambda_{fz}$.

Table 5.2
Point contacts with and without friction and the soft finger contact

Contact type	Geometry	Selection matrix \mathbf{H}^T $\mathbf{w}_C = \mathbf{H}^T \boldsymbol{\lambda}$	Constraints
Frictionless point contact		$\mathbf{w}_C = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [\lambda_{fz}]$	$\lambda_{fz} \geq 0$
Point contact with friction		$\mathbf{w}_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_{fx} \\ \lambda_{fy} \\ \lambda_{fz} \end{bmatrix}$	$\lambda_{fz} \geq 0$ $[\lambda_{fx}^2 + \lambda_{fy}^2]^{1/2} \leq \mu\lambda_{fz}$
Soft finger		$\mathbf{w}_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{fx} \\ \lambda_{fy} \\ \lambda_{fz} \\ \lambda_{mz} \end{bmatrix}$	$\lambda_{fz} \geq 0$ $[\lambda_{fx}^2 + \lambda_{fy}^2]^{1/2} \leq \mu\lambda_{fz}$ $\lambda_{mz} \leq \gamma\lambda_{fz}$

The bottom row of [table 5.2](#) describes the soft finger contact—similar, in terms of contact mechanics, to compliant tissue on human fingertips—that deforms to comply with surface patches in the environment. As a result, this contact type can also support moments about the local surface normal subject to constraints that depend on a finite torsional coefficient of friction γ and the normal force $\lambda_{mz} \leq \gamma\lambda_{fz}$.²

5.3.4 The Generalized Grasp Jacobian

Independent contact wrenches for \mathbf{C}_i are mapped to the object frame wrench using a modified version of [equation 5.8](#) written in terms of $\boldsymbol{\lambda}_{Ci}$.

$$(\mathbf{w}_O)_i = \mathbf{G}_i \mathbf{w}_{Ci} = \mathbf{G}_i \mathbf{H}_i^T \boldsymbol{\lambda}_{Ci} \quad (5.10)$$

Mapping \mathbf{G}_i is derived by constructing the linear transformations that (1) rotate wrenches in the contact frame into a frame aligned with the object frame and then (2) translate the wrench to the location of the object frame. The rotation matrix ${}^o\mathbf{R}_{C_i}$ that transforms the object frame to contact coordinates C_i is written in a block diagonal form,

$$\bar{\mathbf{R}}_i = \left[\begin{array}{ccc|ccc} {}^o\mathbf{R}_{C_i} & & & \mathbf{0} & & \\ & & & & & \\ & & & & & \\ \hline & & & & {}^o\mathbf{R}_{C_i} & \\ & & & & & \\ & & & & & \end{array} \right],$$

which applies the rotation to the force and moment components of a contact wrench independently.

To translate the result from the contact site to the object frame, we note that the force components of the wrench map to the same forces in the object frame, whereas contact frame moments sum with the couple $\boldsymbol{\rho} \times \mathbf{f}_{C_i}$, where $\boldsymbol{\rho} \in \mathbb{R}^3$ is the position vector locating frame C with respect to frame O ,

$$\mathbf{P}_i = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & -\rho_z & \rho_y & 1 & 0 & 0 \\ \rho_z & 0 & -\rho_x & 0 & 1 & 0 \\ -\rho_y & \rho_x & 0 & 0 & 0 & 1 \end{array} \right]. \quad (5.11)$$

The top half of this matrix maps contact forces to the object frame, and the bottom half sums the contact moments with moments caused by the contact forces at the position of the object frame. The bottom-left 3×3 portion of \mathbf{P}_i is the matrix form of the cross product used to compute $\boldsymbol{\rho} \times \mathbf{f}_{C_i}$. The net result is a geometrical means of computing the grasp Jacobian given knowledge of the contact configuration, the local contact frames, and the individual contact types:

$$(\mathbf{w}_O)_i = \mathbf{G}_i^* \lambda_{C_i}, \text{ where}$$

$$\mathbf{G}_i^* = \mathbf{P}_i \bar{\mathbf{R}}_i \mathbf{H}_i^T.$$

For an n contact grasp configuration, the grasp Jacobian and effort is written

$$\mathbf{G}^* = [\mathbf{G}_1^* \cdots \mathbf{G}_n^*]$$

and

$$\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{C_1}^T \cdots \boldsymbol{\lambda}_{C_n}^T]^T.$$

Example: The Two-Contact Grasp Jacobian

Following the analysis in [251], we derive the grasp Jacobian for the two-contact planar grasp geometry in figure 5.12 and use it to solve for grasp forces that meet frictional constraints. The grasp Jacobian maps n contact wrenches (written in the contact frames) into a net wrench in the object frame. The left contact is a point contact with friction $\boldsymbol{\lambda}_{C_1} = [\lambda_1 \lambda_2 \lambda_3]^T$, and the right contact is a soft finger $\boldsymbol{\lambda}_{C_2} = [\lambda_4 \lambda_5 \lambda_6 \lambda_7]^T$. Referring to table 5.2,

$$\mathbf{H}_1^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{\lambda}_1 = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad \mathbf{H}_2^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{\lambda}_2 = \begin{bmatrix} \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \end{bmatrix}.$$

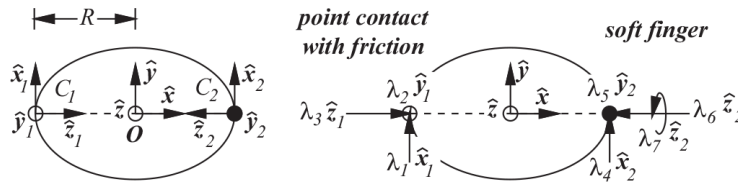


Figure 5.12

Two contacts form a grasp on an ellipsoidal object. (Left) The geometry of a pair of contact frames relative to the object frame. (Right) The magnitude of forces $\boldsymbol{\lambda}$ aligned with the contact frames.

The local contact frames must be rotated to align with the object frame. In this example, the rotation from object frame to C_1 is a rotation about \hat{x}_O of $\pi/2$ followed by a rotation about \hat{y}_O of $\pi/2$.

$${}^O\mathbf{R}_{C_1} = \text{rot}(\hat{x}_O, \pi/2) \text{rot}(\hat{y}_O, \pi/2) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Similarly, the rotation from the object frame to C_1 is a rotation about \hat{z}_0 of $\pi/2$ followed by a rotation about \hat{x}_0 of $-\pi/2$.

$${}^O\mathbf{R}_{C_2} = \text{rot}(\hat{z}_0, \pi/2) \text{rot}(\hat{x}_0, -\pi/2) = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

The rotation matrices are used to construct the block diagonal forms that apply to both force and moment components of the contact wrenches independently:

$${}^O\bar{\mathbf{R}}_{C_1} = \left[\begin{array}{ccc|ccc} 0 & 0 & 1 & & & \\ 1 & 0 & 0 & & \mathbf{0} & \\ 0 & 1 & 0 & & & \\ \hline & & & 0 & 0 & 1 \\ & \mathbf{0} & & 1 & 0 & 0 \\ & & & 0 & 1 & 0 \end{array} \right] \quad {}^O\bar{\mathbf{R}}_{C_2} = \left[\begin{array}{ccc|ccc} 0 & 0 & -1 & & & \\ 1 & 0 & 0 & & \mathbf{0} & \\ 0 & -1 & 0 & & & \\ \hline & & & 0 & 0 & -1 \\ & \mathbf{0} & & 1 & 0 & 0 \\ & & & 0 & -1 & 0 \end{array} \right].$$

To translate wrench from the contact site to the object frame, [equation 5.11](#) is evaluated for $\rho_{C_1}^O = [-R \ 0 \ 0]^T$ and $\rho_{C_2}^O = [R \ 0 \ 0]^T$:

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & R & 0 & 1 & 0 \\ 0 & -R & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -R & 0 & 1 & 0 \\ 0 & R & 0 & 0 & 0 & 1 \end{bmatrix}$$

As a consequence of these transformations, we can map each of the independent contact forces into object frame wrenches,

$$\mathbf{G}_1^* = \mathbf{P}_1 \bar{\mathbf{R}}_1 \mathbf{H}_1^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & R & 0 \\ -R & 0 & 0 \end{bmatrix} \quad \mathbf{G}_2^* = \mathbf{P}_2 \bar{\mathbf{R}}_2 \mathbf{H}_2^T = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & R & 0 & 0 \\ R & 0 & 0 & 0 \end{bmatrix},$$

so that

$$\mathbf{w}_O = [\mathbf{G}_1^* \ \mathbf{G}_2^*] \begin{bmatrix} \lambda_{C_1} \\ \lambda_{C_2} \end{bmatrix}$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix}_O = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & R & 0 & 0 & R & 0 & 0 \\ -R & 0 & 0 & R & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \end{bmatrix}.$$

In general, the λ_i create wrenches in the contact frames that map through the grasp Jacobian and sum to produce wrenches (\mathbf{w}_O) in the object frame. Column i of \mathbf{G} corresponds to the independent wrench that is applied to the object when the corresponding $\lambda_i = 1$. For example, the third column in \mathbf{G} is the object frame wrench $[1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ corresponding to $\lambda_3 = 1$ in contact C_1 . We assume that each component of λ is independently controllable. A reference wrench \mathbf{w}_{ref} can be applied to the object if a linear combination λ in the column space of \mathbf{G} can be found subject to contact type constraints.

□

5.3.5 Grasp Performance: Form and Force Closure

Scientists that study the mechanics of contact speak formally about the quality of contact systems in terms of mechanical *closure* properties that describe how well grasp configurations can resist object twist or wrench disturbances. In the literature, the case when a fixed contact geometry immobilizes an object without consideration of additional forces (called *form* closure) is distinguished from the case when the contact system implements a viscoelastic potential well in which changing contact forces resist disturbance forces (called *force* closure).

Reuleaux originally [241] defined *form closure* as “the condition under which a positive combination of contact wrenches derived from frictionless contacts can resist perturbation forces.” Because of the assumption that contacts are frictionless, the contact forces cited in Reuleaux’s definition arise solely from compression in the contacting materials and are directed

along the local surface normal. If we consider rigid material, we can restate this definition exclusively in terms of restrictions on object twists.

Definition 5.4: Form closure. *A condition of complete restraint in which any object twist $\in \mathbb{R}^6$ is geometrically incompatible with rigid body assumptions concerning the object and a set of fixed contacts.*

Under these conditions, form closure does not depend on internal forces or friction to immobilize the object and can, therefore, be attractive when such forces are problematic. For example, milling operations impose large forces to stock material that must be held rigidly to achieve precise part geometries. Form closure fixtures can provide high precision up to the yield point of the materials without requiring large internal forces as would be the case for clamps and vises. Theoretical results have established bounds on the number of fixturing elements needed to completely immobilize a workpiece. In 1875, Reuleaux used a first-order analysis to show that (non-exceptional) bodies require at least four frictionless contacts for form closure in the plane [241]. Exceptional surfaces are surfaces of revolution and cannot be rendered completely immobile by any number of frictionless point contacts. In 1897, Somov [264] used the same kinds of analysis to prove that at least 7 frictionless point contacts are necessary for form closure in three dimensions. Mishra et al. [196] have established an upper bound of 6 frictionless point contacts on planar objects with piecewise smooth contours and 12 for the spatial case (except in the case of Reuleaux's exceptional surfaces). Second-order analyses incorporating the curvature of contact surfaces can be used to verify form closure using fewer discrete contacts in special cases (see, for example, [242]).

A form closure contact geometry essentially builds a cage around the object that eliminates any movement of the object relative to the hand. Manipulation, however, often requires precise relative motions to, for instance, start a nut on a threaded bolt. There are better options available when the hand can control grasp forces and impart frictional forces. Several of the contact types listed in table 5.1 incorporate frictional forces that act in the local surface tangent at the contact location. These forces depend, in general, on the magnitude of the normal contact force and the materials and surface textures comprising the hand-object system. To exploit these properties, stable grasp solutions depend on the *force closure* property and

concern the resilience of the grasp to random, bounded wrench disturbances by modifying grip forces and engaging frictional forces.

Definition 5.5: Force closure. *A grasp configuration is force closure if a solution for contact frame wrenches λ exists that complies with contact type constraints such that $\mathbf{G}\lambda = -\mathbf{w}_{dist}$ for arbitrary disturbance wrenches \mathbf{w}_{dist} relative to the object frame.*

A grasp is force closure (and stabilizable) if and only if \mathbf{G} is surjective (onto) [202], requiring that there are no object frame wrenches, \mathbf{w}_o , that are inaccessible by mapping contact effort variables λ through the grasp Jacobian. The mapping is, in general, many-to-one, reflecting a continuum of homogeneous solutions in the nullspace of the grasp Jacobian $\mathcal{N}(\mathbf{G})$ whose $rank(\mathcal{N}(\mathbf{G})) \geq 1$. In general, force closure depends on the ability of the hand to squeeze the object within $\mathcal{N}(\mathbf{G})$ to scale frictional forces as required. A force closure grasp with frictionless point contacts is also form closed.

Stated another way, force closure requires that the contact configuration is capable of applying contact forces within a convex subset of object wrench space containing the origin where $\sum \mathbf{w}_i = \mathbf{0}$. If $rank(\mathcal{N}(\mathbf{G})) = k$, the envelope of this set is scalable in k independent ways to accommodate arbitrary \mathbf{w}_{dist} while satisfying frictional contact constraints.

Example: Solving for Forces in Force Closure Grasps

Suppose that the grasp forces in figure 5.12 must support an object mass of 0.1 [kg] against a gravitational load in the negative \hat{y} direction. This will require the sum of contact forces in the positive \hat{y} direction of the object frame of approximately 1.0 [N]. In this case, the three-dimensional analysis of the grasp in figure 5.12 can be reduced to the \hat{x} - \hat{y} plane of the object frame, and a contact force solution is generated for $\mathbf{w}_{dist} = [0 \ -1 \ 0 \ 0 \ 0 \ 0]^T$. The coefficients of friction for tangential forces are $\mu_1 = 0.2$ and $\mu_2 = 0.5$.

In general, solutions for contact wrenches \mathbf{w}_C such that $\mathbf{w}_O = \mathbf{G}\mathbf{w}_C$ have homogeneous and particular parts,

$$\lambda = \lambda_p + k\lambda_h. \tag{5.12}$$

The particular solution is a vector of contact force commands λ_p that produce the reference wrench $-\mathbf{w}_{dist}$. In this case, the task requires that $\lambda_1 + \lambda_4 = 1.0$. To apply a net zero moment about the \hat{z} axis $\lambda_1 = \lambda_4 = 0.5$. However, λ_1 and λ_4 correspond to tangential forces and depend on the magnitude of the normal forces, λ_3 and λ_6 , and the respective coefficients of friction. On the left side, $\lambda_1 \leq 0.2\lambda_3$ requires that $\lambda_3 \geq 2.5$, and on the right side $\lambda_4 \leq 0.5\lambda_6$ requires that $\lambda_6 \geq 1.0$. Static equilibrium requires that the sum of forces in the \hat{x} direction must be zero; therefore $\lambda_3 = \lambda_6 = 2.5$.

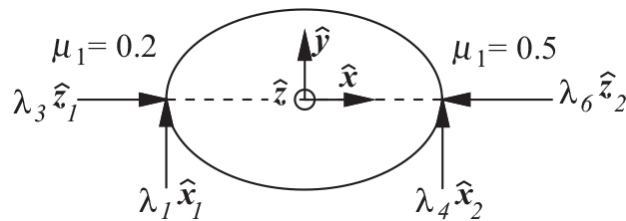


Figure 5.13

The planar projection of the three-dimensional grasp in [figure 5.12](#)

In [equation 5.12](#), λ_h is the homogeneous part of the solution and describes combinations of contact forces that impart a zero net wrench to the object.

$$\mathbf{G}\lambda_h = 0$$

Therefore, in [figure 5.12](#), the nullspace is the opposition between λ_3 and λ_6 ,

$$\lambda_h = [0 \ 0 \ k \ 0 \ 0 \ k \ 0]^T : k \geq 2.5,$$

to create compressive (internal) forces in the object capable of supporting the tangential forces required—as well as the soft finger moment λ_7 in the original version of the problem in [figure 5.12](#).

The grasp supports a convex six-dimensional hyper-rectangular volume in wrench space surrounding the equilibrium pose of the object that can be controlled to stabilize the grasp against disturbances. The volume can be scaled by squeezing the object (increasing k) and, thus, expanding the capacity of the wrench closure.

□

A natural choice for solving systems of interdependent contact forces is to use mathematical programming techniques to solve for contact forces subject to a system of inequality constraints [142, 99]. These techniques can be used in a closed-loop framework to solve for the minimum nullspace grasp forces required to stabilize a grasp against unexpected perturbations.

5.4 Exercises

1. **Screw space: Nonlinear vector space.** Explain why twists and wrenches do not constitute basis vectors for a linear screw space describing all velocities and forces, respectively.

2. **Object mobility.**

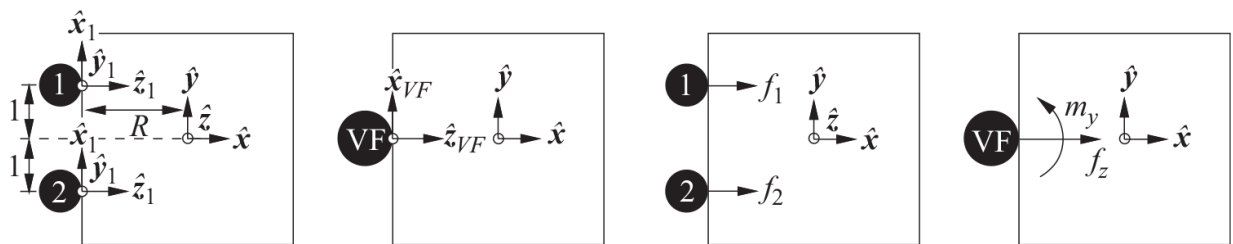
(a) What is the minimum number of Velcro contacts in figure 5.10 required to suppress perturbations and control position in \mathbb{R}^2 ?

(b) Analyze grasp stability for the more general system in figure 5.11 considering frictionless point contacts.

i. Is the three-contact grasp stable and controllable? Justify your answer.

ii. What is the minimum number and identity of contacts of this type that can suppress arbitrary perturbations $\in \mathbb{R}^2$? Provide the analysis to support your answer.

3. **Virtual fingers.** The left column of this figure depicts a pair of frictionless point contacts on the surface of an object. This contact geometry is sometimes called a *frictionless line contact*. The right column proposes a *virtual finger*, positioned midway between contacts 1 and 2. The bottom row of the figure shows the independently controllable contact forces for these two cases in contact coordinates.



(a) Contact force geometry

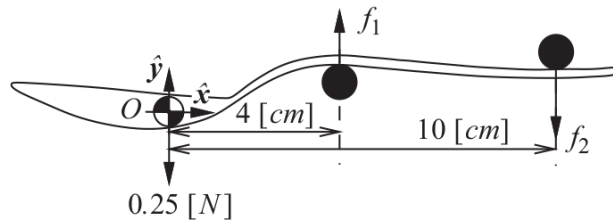
(b) Contact frame wrenches

The virtual finger is equivalent to the original contact system if for every contact wrench $[\lambda_{f_1} \ \lambda_{f_2}]^T$ in the two-contact system there exists a unique equivalent solution $[\lambda_{f_z} \ \lambda_{m_y}]^T$ and vice versa.

- Define the selection matrices \mathbf{H}_1^T , \mathbf{H}_2^T , and \mathbf{H}_{VF}^T using the conventions established in [table 5.1](#).
- Using the geometry specified in the figure, derive \mathbf{G}_{12}^* for the two-contact system and \mathbf{G}_{VF}^* for the proposed virtual finger.
- Use these relations to describe the reciprocal mapping between effort variables $\lambda_{1, 2}$ and λ_{VF} for these alternative representations of arbitrary contact wrenches.
- Does the virtual finger completely and correctly represent the original two-contact system?

4. Planar grasp force solution: Two contacts on the spoon.

In this planar two-contact grasp geometry, the contacts are frictionless point contacts. All contact forces are in the \hat{x} - \hat{y} plane and are parallel to the \hat{y}_O axis. Wrenches, in this case, can be written $\mathbf{w}=[f_x \ f_y \ m_z]^T$.

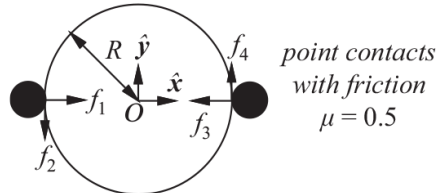


- Assuming that $\|f_i\|=1$, write the grasp Jacobian, $\mathbf{G} \in \mathbb{R}^{3 \times 2}$, such that $\mathbf{w} = \mathbf{G} \begin{bmatrix} \lambda_{f1} \\ \lambda_{f2} \end{bmatrix}$
- Write the constraints on λ imposed by the contact types.
- Write a solution for the λ necessary to apply the wrench $\mathbf{w}=[0 \ 0.25 \ 0]^T$ that will support the object in the posture shown.
- Is this grasp form closure? Explain.
- Is this grasp geometry force closure if we consider point contacts *with* friction? Explain.

5. Planar grasp force solution: Two contacts on a circle. A two-contact grasp is performed on this circular knob with radius $R = 0.025 [m]$. The

contacts are point contacts with friction with controllable contact forces. Assume only forces in the x - y plane, so that wrenches

$$\mathbf{w} = \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix}.$$



(a) Assuming that $\|\mathbf{f}_i\|=1$, write the grasp Jacobian, $\mathbf{G}^* \in \mathbb{R}^{3 \times 4}$, such that

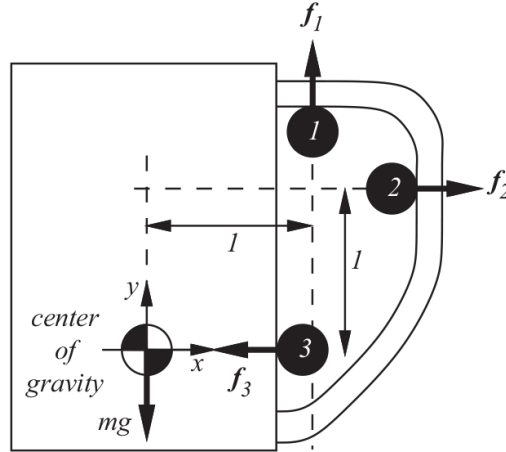
$$\mathbf{w} = \mathbf{G}^* \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}.$$

(b) Write the constraints on λ imposed by the point contact with friction model.

(c) Write a solution for the λ necessary to apply a pure torque of 0.5 [Nm] to the knob $\mathbf{w} = [0 \ 0 \ 0.5]^T$.

6. Planar grasp force solution: Three contacts on the coffee cup. Three *frictionless* point contacts are positioned on the planar shape to produce object frame wrenches about the object's center of gravity.

$$\mathbf{w}_i = \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix}_i \quad i = 1, 2, 3$$



(a) Assuming that $\|\mathbf{f}_i\|=1$, write the grasp Jacobian, $\mathbf{G}^* \in \mathbb{R}^{3 \times 3}$, such that

$$\mathbf{w} = \mathbf{G}^* \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}.$$

(b) Write the constraints on λ imposed by the frictionless point contact type at each contact.

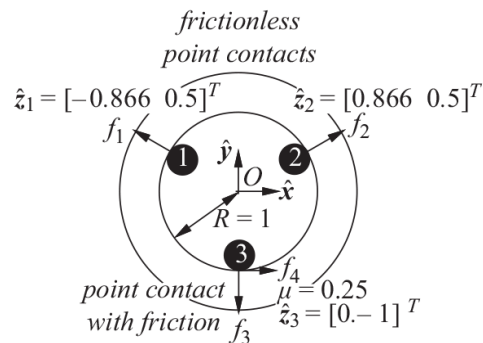
(c) Solve for a (force closure) solution λ such that

$$\begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} = \mathbf{G}^* \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}.$$

(d) Are there other grasp configurations involving fewer frictionless point contacts than this for which a force closure solution exists?

7. Planar grasp force solution: Three contacts on a ring. In this inside-out planar three-contact grasp geometry on the inside of a donut with hole radius $R = 1$, the top two contacts are frictionless point contacts. The bottommost contact is a point contact with friction ($\mu=0.25$). All forces are restricted to the x - y plane, so that wrenches

$$\mathbf{w} = \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix}.$$



- (a) Is this geometry form closure? Explain.
- (b) Assuming that $\|f_i\|=1$, write the grasp Jacobian, $\mathbf{G}^* \in \mathbb{R}^{3 \times 4}$, such that

$$\mathbf{w} = \mathbf{G}^* \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}.$$

- (c) Write the constraints on λ imposed by the contact types.
- (d) Write the solution for the minimal $\|\lambda\|$ necessary to apply a $[0 \ 0 \ 0.5]^T$ wrench to the object.
- (e) Is the grasp geometry force closure? Explain.

8. **Invent your own homework.** Make a homework problem out of your favorite content in chapter 5. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here. Ideally, they should call for a short discussion and quantitative analysis—it should not take more than 30 [min] to solve open-book.

1. Many of the mathematical tools employed to describe systems of multiple contacts and grasping are specialized versions of the more general methods introduced in chapter 4.
2. The constraint on the magnitude of the moment that can be supported by the soft finger contact also depends on the magnitude of the tangential force [137, 138]. Table 5.2 uses a simpler, less

accurate, but nonetheless useful model.

6

Dynamics of Articulated Systems

The dynamic analysis of an inertial system describes how forces act on bodies to create accelerations. In structures that are designed to be rigid like bridges and trusses, it can concern the time varying pattern of deflection in the elastic members and the distribution of forces resulting from external loads. In robot structures, most often we consider each link to be rigid and derive the relationship between forces and accelerations over the entire articulated structure subject to kinematic constraints. These expressions can be complex and, depending on the kinematics of the mechanism, they can couple the motion of several degrees of freedom.

Techniques for writing the governing equations of motion for more complicated mechanisms like those used in robotic devices is the subject of considerable research. The interested reader can find two important examples (the Newton-Euler iteration and the Lagrangian) in appendix B. In this chapter, our focus is on how knowledge of dynamics supports (1) simulation, (2) compensation for nonlinearities and inertial coupling in robot control, and (3) analytical tools for describing robot performance.

6.1 Newton's Laws

Newton's famous laws concern the motion of a point mass in \mathbb{R}^3 .

First law—A particle remains in a state of constant rectilinear motion unless acted on by an external force. Stated another way, a point mass experiencing no external forces will move in a constant velocity *along a*

straight line. The first law implicitly adds the observer to the system—it requires that the complete state of motion of the observer is known (or is nonaccelerating) in order for Newton’s laws to apply. Such a frame of reference for the observer is called an *inertial* coordinate frame.

Second law—The time-rate-of-change in the momentum ($m\mathbf{v}$) of the particle is proportional to the externally applied forces, $\mathbf{f} = \frac{d}{dt}(m\mathbf{v})$. When the mass of the body is constant, the second law leads to the most commonly used form of Newton’s equation, $\mathbf{f} = \frac{d}{dt}(m\mathbf{v}) = m\mathbf{a}$.

Third law—A force imposed on body A by body B is accompanied by a reciprocal reaction force—equal in magnitude and opposite in direction—on body B by body A . This law has important consequences when multiple bodies interact in articulated mechanical structures.

Newton’s three laws provide the starting point for the derivation by defining the dynamics of a point mass. Many kinodynamic systems employ revolute degrees of freedom. In these cases, Newton’s laws are recast in terms of torques rather than forces and, instead of mass, the critical inertial quantity is the *mass moment of inertia*. We begin this chapter by deriving the moment of inertia and describing how it depends on the geometry of motion. With this foundational physics, we can write the dynamic equation of motion in a form called the computed-torque equation for a simple mechanism consisting of a single revolute degree of freedom.

6.2 The Inertia Tensor

To describe how torques cause angular accelerations in a rigid body, the distribution of mass about the center of rotation must be considered. In [figure 6.1](#), point mass m_k is an element of a rigid planar lamina that rotates about the \hat{z} axis of frame \mathbf{O} . Thus, when $\omega \neq 0$, mass m_k describes a circular orbit of radius r_k . [Figure 6.1](#) defines two additional basis vectors, \hat{r} and \hat{t} , aligned with the radial and tangential directions of the orbit, respectively.

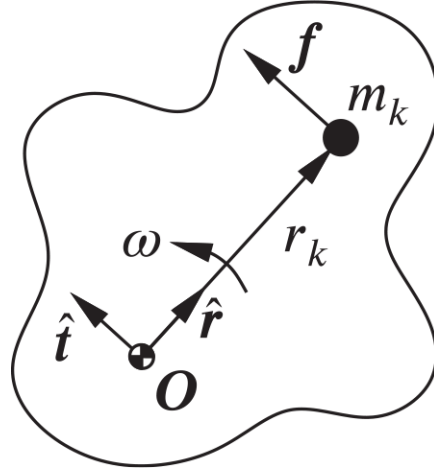


Figure 6.1

A single element of a set of point masses comprising a planar lamina. The \hat{z} axis of frame \mathbf{O} is directed out of the plane of the figure.

To generate an angular acceleration about \mathbf{O} , a torque is applied around the \hat{z} axis,

$$\boldsymbol{\tau}_k = \mathbf{r} \times \mathbf{f} = r_k \hat{\mathbf{r}} \times \frac{d}{dt}(m_k \mathbf{v}_k) = m_k r_k \left[\hat{\mathbf{r}} \times \frac{d}{dt}(\mathbf{v}_k) \right].$$

The velocity of m_k due to the rotation about \mathbf{O} is $\mathbf{v}_k = (\omega \hat{\mathbf{z}} \times r_k \hat{\mathbf{r}}) = (r_k \omega) \hat{\mathbf{t}}$, so that $\boldsymbol{\tau}_k = (m_k r_k^2) \dot{\omega} \hat{\mathbf{z}} = J_k \dot{\omega} \hat{\mathbf{z}}$. Here, $J_k = m_k r_k^2$ describes how much torque $\boldsymbol{\tau}_k$ is required to produce angular acceleration $\dot{\omega}$ for mass m_k as it orbits the origin of coordinate frame \mathbf{O} about the $\hat{\mathbf{z}}$ axis. The rotational inertia of the entire lamina is the sum over all m_k ,

$$\boldsymbol{\tau} = \left(\sum_k m_k r_k^2 \right) \dot{\omega} = J \dot{\omega}. \quad (6.1)$$

The inertial parameter J [$\text{kg} \cdot \text{m}^2$] is the scalar *moment of inertia* of the rotating system and depends on the distribution of mass relative to the center of rotation. Equation 6.1 relates torque to angular acceleration in the same way that Newton's equation $\mathbf{f} = \frac{d}{dt}(m\mathbf{v}) = m\mathbf{a}$ relates force to linear acceleration. The counterpart of linear momentum ($p=m\mathbf{v}$) is *angular momentum* $L=J\omega$ in the rotating system, so that Euler's equation can be written in the same form as Newton's second law,

$$\boldsymbol{\tau} = \frac{d}{dt} [J\boldsymbol{\omega}] = J\ddot{\boldsymbol{\theta}}. \quad (6.2)$$

Therefore, analogous to the Newtonian particle, rotating bodies conserve angular momentum (and remain in a constant state of angular velocity) unless acted upon by an external torque.

To extend this analysis to objects that rotate about three orthogonal axes requires the construction of the *inertia tensor*.¹ Appendix B.1 considers a discrete stack of planar laminae (figure B.1) and derives the inertia tensor of a three-dimensional body about an arbitrary axis $\hat{\mathbf{a}} \in \mathbb{R}^3$. The result is the well-known 3×3 positive definite and symmetric inertia tensor

$${}^A \mathbf{J} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix}, \quad (6.3)$$

where the diagonal elements J_{xx} , J_{yy} , and J_{zz} are called the *moments of inertia* and the off-diagonal elements J_{xy} , J_{xz} , and J_{yz} are the *products of inertia*.

$$\begin{aligned} {}^A J_{xx} &= \int \int \int (y^2 + z^2) \rho \, dx \, dy \, dz & {}^A J_{yx} &= {}^A J_{xy} = \int \int \int (xy) \rho \, dx \, dy \, dz \\ {}^A J_{yy} &= \int \int \int (x^2 + z^2) \rho \, dx \, dy \, dz & {}^A J_{zx} &= {}^A J_{xz} = \int \int \int (xz) \rho \, dx \, dy \, dz \\ {}^A J_{zz} &= \int \int \int (x^2 + y^2) \rho \, dx \, dy \, dz & {}^A J_{zy} &= {}^A J_{yz} = \int \int \int (yz) \rho \, dx \, dy \, dz \end{aligned} \quad (6.4)$$

Example: Rotational Moment of Inertia

The rectangular body illustrated in figure 6.2 rotates about the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ axes of frame \mathcal{A} . For rotation around the $\hat{\mathbf{x}}$ axis,

$$\begin{aligned} {}^A J_{xx} &= \int_0^h \int_0^l \int_0^w (y^2 + z^2) \rho \, dx \, dy \, dz = \int_0^h \int_0^l (y^2 + z^2) w \rho \, dy \, dz \\ &= \int_0^h \left[\left(\frac{y^3}{3} + z^2 y \right) \right]_0^l w \rho \, dz = \int_0^h \left(\frac{l^3}{3} + z^2 l \right) w \rho \, dz \\ &= \left(\frac{l^3 z}{3} + \frac{l z^3}{3} \right) \Big|_0^h (w \rho) = \left(\frac{l^3 h}{3} + \frac{l h^3}{3} \right) w \rho. \end{aligned}$$

The other elements of the inertia tensor can be computed in a similar fashion. Since the mass of the rectangular body is $m=(lwh)\rho$, we find that

$${}^A\mathbf{J} = \begin{bmatrix} \frac{m}{3}(l^2 + h^2) & -\frac{m}{4}wl & -\frac{m}{4}hw \\ -\frac{m}{4}wl & \frac{m}{3}(w^2 + h^2) & -\frac{m}{4}hl \\ -\frac{m}{4}hw & -\frac{m}{4}hl & \frac{m}{3}(l^2 + w^2) \end{bmatrix}. \quad (6.5)$$

□

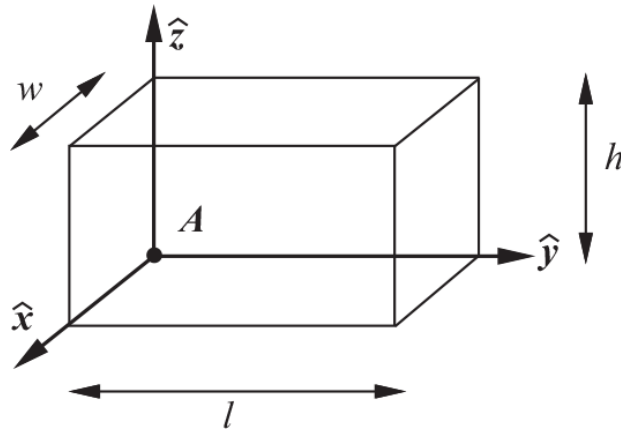


Figure 6.2

A rectangular body with total mass m rotates about the \hat{x} axis of coordinate frame A

In [figure 6.2](#), the mass is distributed on one side of the \hat{x} , \hat{y} , and \hat{z} axes of frame A . Eccentric mass distributions like this increase the magnitude of the inertia tensor ([appendix A.5](#)) so that it requires more torque to accelerate the body about the axes of frame A than it would about a frame with a more symmetric mass distribution. Moreover, an eccentric mass distribution leads to nonzero products of inertia in the tensor that inertially couple across the \hat{x} , \hat{y} , and \hat{z} axes. Therefore, given an arbitrary frame A , to generate an acceleration exclusively around the \hat{z} axis ($\dot{\omega} = [0 \ 0 \ 1]^T$) requires torques around all three axes. This is evident by combining [equations 6.1](#) and [B.4](#):

$$\boldsymbol{\tau} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -J_{xz} \\ -J_{yz} \\ J_{zz} \end{bmatrix}.$$

For symmetric bodies, when the axes of rotation are parallel to the axes of symmetry of the body and the center of rotation coincides with the *center of mass*, the inertia tensor is diagonalized and its magnitude is minimized, leading to an efficient and uncoupled transformation from torques to accelerations. Unfortunately, links in a robot mechanism generally do not rotate about their centers of mass, nor do they rotate exclusively about axes of symmetry. Often, for a link in a mechanism, it is convenient to calculate the inertia tensor in a frame that reflects its symmetry and then to transform the result into a different frame that expresses the kinematic constraints of the mechanism. To do so, we require a means of recomputing the inertia tensor in response to translations and rotations of frame A .

6.2.1 The Parallel Axis Theorem

The center of mass is the location about which the mass of the body is distributed symmetrically—it is the mean position of the mass distribution in three dimensions. Given an arbitrary choice for frame A in [figure 6.2](#), the *center of mass* of the body is defined

$$\mathbf{r}_{cm} = \frac{\sum m_i \mathbf{r}_i}{\sum m_i}.$$

For a body with a known inertia tensor at the center of mass ${}^{CM}\mathbf{J}$, the parallel axis theorem states that the inertia tensor at any other parallel frame (related to frame CM by pure translation) is computed as

$${}^A\mathbf{J} = {}^{CM}\mathbf{J} + \begin{bmatrix} m(r_y^2 + r_z^2) & -m(r_x r_y) & -m(r_x r_z) \\ -m(r_y r_x) & m(r_x^2 + r_z^2) & -m(r_y r_z) \\ -m(r_z r_x) & -m(r_z r_y) & m(r_x^2 + r_y^2) \end{bmatrix},$$

where $\mathbf{r}_{cm} = [r_x \ r_y \ r_z]$ and m is the total mass of the body. The correction to ${}^{CM}\mathbf{J}$ is precisely the contribution to the inertia tensor of a point mass equal in magnitude to the total mass of the body and concentrated at \mathbf{r}_{cm} .

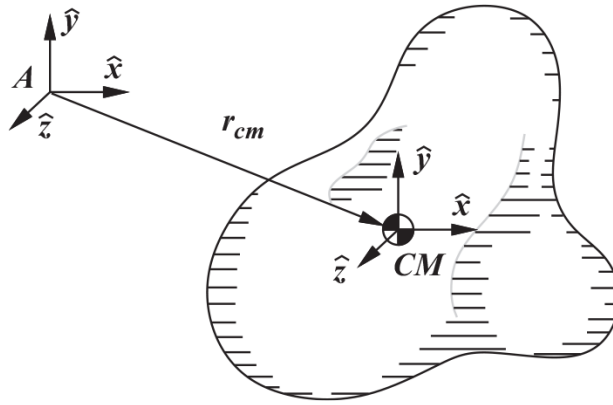


Figure 6.3
An eccentric mass distribution that rotates about an axis that passes through the origin of frame A

Example: Translating the Center of Rotation

In the previous example, the axes of frame A are parallel to the axes of symmetry of the body, but the center of mass of the rectangular body is offset from any of the axes of rotation.²

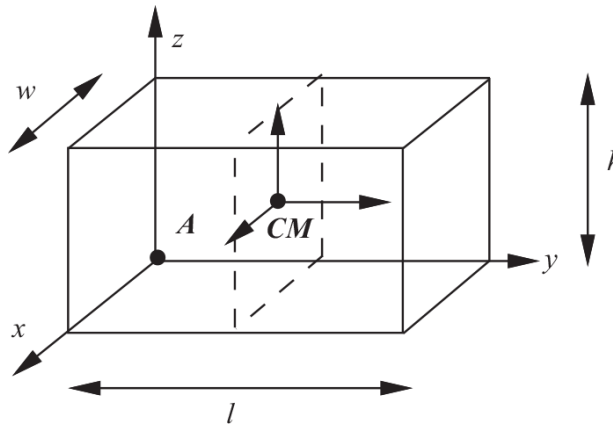


Figure 6.4
Moving the center of rotation to the center of mass

Suppose that the center of rotation is now translated to the object's center of mass at $r_x = w/2$, $r_y = l/2$, and $r_z = h/2$ as in [figure 6.4](#). In this situation, the parallel axis theorem holds that

$$\begin{aligned}
{}^{CM}J_{zz} &= {}^A J_{zz} - m(r_x^2 + r_y^2) \\
&= \frac{m}{3}(l^2 + w^2) - \frac{m}{4}(l^2 + w^2) \\
&= \frac{m}{12}(l^2 + w^2)
\end{aligned}$$

and

$$\begin{aligned}
{}^{CM}J_{xy} &= {}^A J_{xy} + m(r_x r_y) \\
&= -\frac{m}{4}(wl) + \frac{m}{4}(wl) = 0.
\end{aligned}$$

Completing similar computations for the remaining four unique elements of the inertia tensor reveals that the result of moving the center of rotation to the center of mass is the diagonalized inertia tensor

$${}^{CM}\mathbf{J} = \frac{m}{12} \begin{bmatrix} (l^2 + h^2) & 0 & 0 \\ 0 & (w^2 + h^2) & 0 \\ 0 & 0 & (l^2 + w^2) \end{bmatrix}.$$

The diagonalized inertia tensor is a consequence of moving frame A to the center of mass, *and* it reflects the alignment of the frame to the axes of symmetry of the rectangular body. Other choices for the orientation of frame CM produces an eccentric mass distribution as before and reintroduces nonzero products of inertia in the tensor.

□

6.2.2 Rotating the Inertia Tensor

The angular momentum vector $\mathbf{J}\boldsymbol{\omega}$ of a rotating body is conserved during the rotation of the coordinate frame; therefore,

$$\begin{aligned}
\mathbf{J}_1 \boldsymbol{\omega}_1 &= {}_1\mathbf{R}_0 (\mathbf{J}_0 \boldsymbol{\omega}_0) = {}_1\mathbf{R}_0 \mathbf{J}_0 ({}_1\mathbf{R}_0^T {}_1\mathbf{R}_0) \boldsymbol{\omega}_0 \\
&= ({}_1\mathbf{R}_0 \mathbf{J}_0 {}_1\mathbf{R}_0^T) \boldsymbol{\omega}_1,
\end{aligned}$$

and the new inertia tensor can be written

$$\mathbf{J}_1 = {}_1\mathbf{R}_0 \mathbf{J}_0 {}_1\mathbf{R}_0^T.$$

Exercise 6.4.1 provides an opportunity to solve an example problem that requires the rotation of the inertia tensor.

6.3 The Computed Torque Equation

The dynamics of articulated mechanisms stem directly from foundations in Newtonian mechanics, although for interesting robots, these relations can be quite complex. The complexity stems from the interaction between the multiple bodies involved that influence how velocities propagate through a kinematic chain and how forces present at one part of the mechanism propagate to cause changes in momentum elsewhere in the structure.

There are multiple techniques for deriving the dynamics of a robot from a description of the system. Two important examples (the Newton-Euler iterations and the Lagrangian) are introduced in appendix B. The result is a description of the relationship between forces and accelerations that depends on the state of motion of the inertial device.

The result is the equation of motion written in a form called the *computed torque equation*.

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F} \quad (6.6)$$

For an n degree of freedom robot, \mathbf{q} is the $n \times 1$ vector of configuration variables and $\boldsymbol{\tau}$ is the corresponding $n \times 1$ vector of forces or torques applied at these degrees of freedom. $\mathbf{M}(\mathbf{q})$ is the configuration-dependent generalized inertia matrix. It is an $n \times n$, positive definite and symmetric matrix (by virtue of its physical relationship to mass and/or mass moments) and is, therefore, always invertible. The $n \times 1$ vector $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$ depends on both positions \mathbf{q} and velocities $\dot{\mathbf{q}}$ and represents centripetal and Coriolis forces (terms that contain the products of velocities). Vector $\mathbf{G}(\mathbf{q})$ is an $n \times 1$ vector of gravitational forces and/or torques. In addition, real systems may be subject to friction and/or forces derived from contacts. The $n \times 1$ vector \mathbf{F} in [equation 6.6](#) accounts for the forces and torques in the mechanism from these types of forces. [Equation 6.6](#) is sometimes referred to as the state

space form of the dynamic equations since it is written in terms of state variables (q, \dot{q}) .

Example: Dynamic Model of Roger's Eye

Roger's eye is modeled as a single revolute joint with point mass m located at the end of a massless link of length l (figure 6.5). For this idealized system, the center of mass is located at the point mass, and since it is dimensionless, the moment of inertia $J_{cm} = 0$. Therefore, by the parallel axis theorem, the moment of inertia of the link about the origin is ml^2 . A pure control torque about the \hat{z} axis is applied by a motor at the origin and, for the purposes of this example, we assume that mass m is subject to a gravitational acceleration $(-g)\hat{x}$. This simple robot can be treated using the same *free body* analysis we used to write the dynamic equation of motion for the spring-mass-damper (section 3.2.1). In this case, however, the sum of all the torques acting on the eye is equal to the time-rate-of-change in its angular momentum,

$$\sum \tau = \frac{d}{dt}(J\dot{\theta}) = (ml^2)\ddot{\theta} = \tau_m + mgl\sin(\theta),$$

where τ_m is the torque applied by the motor and $mgl\sin(\theta)$ is the gravitational torque on the mass as a function of θ .

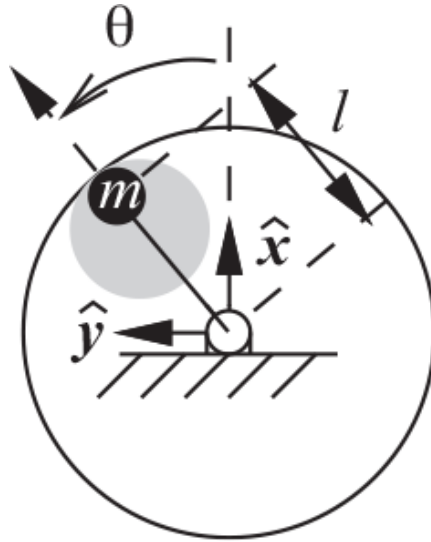


Figure 6.5

The simple dynamic model describing Roger's eye

Rearranging terms reveals the computed torque equation for the eye,

$$\tau_m = \mathbf{M}\ddot{\theta} + \mathbf{G}, \quad (6.7)$$

where: the generalized inertia is a scalar $\mathbf{M}=ml^2$; torques arising from Coriolis and centripetal forces $\mathbf{v}(\theta, \dot{\theta})$ do not exist; and $\mathbf{G}=-mgl\sin(\theta)$.

□

Appendix B introduces two methods for deriving the dynamic equations of motion for generic mechanisms with multiple degrees of freedom, including parallel chain mechanisms common in robotics. The first method is a systematic extension of the free body analysis used in the previous example, known as the Newton-Euler iteration. The second method employs Lagrange's equations, which yield the dynamics directly from the energetics of the system. Appendix B also includes examples of these techniques applied to the 2R planar manipulator to illustrate how inertial coupling works in multiple degree of freedom systems. The results of this derivation are summarized here in the form of the computed torque equation.

Example: Dynamic Model of Roger's Arm

Roger (figure 4.2) has two arms mounted at shoulder joints to a mobile platform. His arms are planar 2R mechanisms with point masses m_1 and m_2 at the ends of two massless links l_1 and l_2 as in figure 6.6. For the purposes of this example, we consider frame 0 in figure 6.6 (the shoulder joint) to be an inertial frame connected to ground. Moreover, in this example, we consider the case when the arm is not in contact with the environment. The dynamic equations of motion for this simplified version of Roger's arm is derived in appendix B.4.3 using the Newton-Euler iteration.

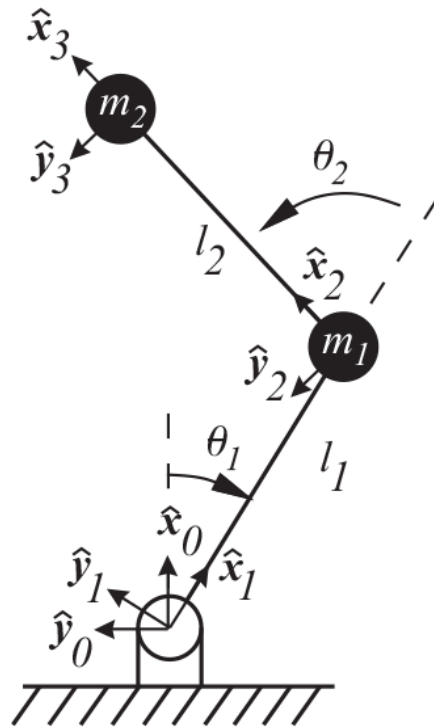


Figure 6.6
The dynamic parameters of Roger's arm relative to inertial shoulder frame 0

Under these circumstances, we can organize the governing dynamic equations derived in appendix B.4.3 into the state space form to arrive at the computed torque equation for Roger's arm,

$$\tau = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\theta, \dot{\theta}) + \mathbf{G}(\theta), \tag{6.8}$$

where

$$\mathbf{M}(\boldsymbol{\theta}) = \begin{bmatrix} m_2 l_2^2 + 2m_2 l_1 l_2 c_2 + (m_1 + m_2) l_1^2 & m_2 l_2^2 + m_2 l_1 l_2 c_2 \\ m_2 l_2^2 + m_2 l_1 l_2 c_2 & m_2 l_2^2 \end{bmatrix} \quad [\text{kg} \cdot \text{m}^2]$$

$$\mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} -m_2 l_1 l_2 s_2 (\dot{\theta}_2^2 + 2\dot{\theta}_1 \dot{\theta}_2) \\ m_2 l_1 l_2 s_2 \dot{\theta}_1^2 \end{bmatrix} \quad [N \cdot m]$$

$$\mathbf{G}(\boldsymbol{\theta}) = \begin{bmatrix} -(m_1 + m_2) l_1 s_1 g - m_2 l_2 s_{12} g \\ -m_2 l_2 s_{12} g \end{bmatrix} \quad [N \cdot m].$$

This is a slightly more complicated expression than we found in the previous example regarding the dynamics of the eye. All three terms in [equation 6.8](#) couple torques to multiple degrees of freedom. In general, arbitrary configurations of the manipulator introduce gravitational torques on both joints, as do the inertial loads in $\mathbf{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ due to the manipulator's state of motion. The generalized inertia is written in the form of a 2×2 matrix. In general, therefore, to achieve a specific acceleration in the arm requires a pattern of torques on both joints.

□

A consequence of the inertial coupling is that motor units addressing an error in a single configuration variable will cause associated accelerations in other configuration variables. The nonlinear and coupled dynamics illustrated by this example are the rule rather than the exception, especially for robots with revolute joints.

In principle, given a kinematic description of the robot and the distribution of mass in the device, the techniques introduced in appendix B can be used to describe the whole-body dynamics of a robot. However, closed-form derivations quickly become complicated for real robots, and numerical techniques are often employed. Regardless of the method employed, once the dynamic equations are derived, there are a number of ways that knowledge of this Newtonian structure can be exploited to enhance the understanding and control of complex robots.

6.3.1 Simulation

The computed torque equation can be inverted to solve for the net acceleration of the mechanism resulting from joint torques combined with inertial, gravitational, and external loads. Given the state of motion $((\mathbf{q}, \dot{\mathbf{q}}))$, we can write the net result of all of these forces and torques on accelerations as

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q}) [\boldsymbol{\tau} - \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q}) - \mathbf{F}]. \quad (6.9)$$

Equation 6.9 can be solved approximately using discrete-time, numerical integration. Given suitable initial values for the state variables, a simulation predicts future states by stepping through time and estimating the area under the acceleration function to determine the change in velocity and computing the area under the velocity curve to estimate the change in position. For example, using a simple Euler integrator, we find

$$\begin{aligned} \ddot{\mathbf{q}}(t) &= \mathbf{M}^{-1} [\boldsymbol{\tau} - \mathbf{V} - \mathbf{G} - \mathbf{F}] \\ \dot{\mathbf{q}}(t + \Delta t) &= \dot{\mathbf{q}}(t) + \ddot{\mathbf{q}}(t)\Delta t \\ \mathbf{q}(t + \Delta t) &= \mathbf{q}(t) + \dot{\mathbf{q}}\Delta t + \frac{1}{2}\ddot{\mathbf{q}}(t)\Delta t^2. \end{aligned} \quad (6.10)$$

Better integrators can be used that employ, for example, higher-order approximations and nonuniform time steps.

6.3.2 Feedforward Control

In section 3.2, a simple motor unit for robot control was proposed using the spring-mass-damper to produce an asymptotically stable system that approximates some of the viscoelastic properties of muscle and the lower motor neurons controlling limb movement in animals. Applied to the 2R manipulator, the coupled, nonlinear dynamics in equation 6.8 suggest that error suppression in one degree of freedom can cause errors in other degrees of freedom. Thus, the coupling can make the characteristic second-order response of the SMD quite different. To the extent that the equation of motion can precisely capture the dynamics of the robot, it can be used to compensate for these coupling terms.

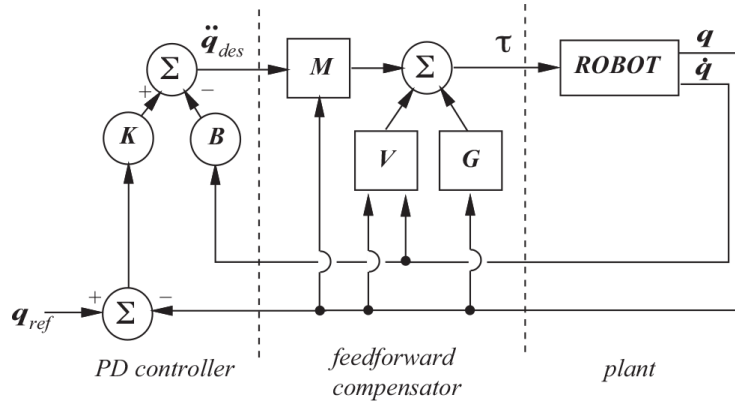


Figure 6.7

The feedforward compensator linearizes and decouples multiple degree of freedom controllers

The feedforward compensator in [figure 6.7](#) scales the output of the PD controller to linearize and decouple degrees of freedom in the plant. In this case, the PD controller is designed for uncoupled, unit masses (or mass moments, as appropriate). Therefore, for each degree of freedom, it produces a force command for this reference plant $\tilde{\tau}_{com} = \mathbf{I}_n \ddot{q}_{des}$. The feedforward compensator computes the command forces/torques for the real plant by correcting the vector of desired accelerations,

$$\boldsymbol{\tau}_{com} = \mathbf{M}\ddot{\mathbf{q}}_{des} + \mathbf{V} + \mathbf{G}. \quad (6.11)$$

The plant is an analog expression of the inverse dynamics

$$\ddot{\mathbf{q}}_{act} = \mathbf{M}^{-1} [\boldsymbol{\tau}_{com} - \mathbf{V} - \mathbf{G}] \quad (6.12)$$

so that plugging [equation 6.11](#) into [equation 6.12](#) yields the identity relation

$$\ddot{\mathbf{q}}_{act} = \mathbf{M}^{-1} [(\mathbf{M}\ddot{\mathbf{q}}_{des} + \mathbf{V} + \mathbf{G}) - \mathbf{V} - \mathbf{G}] = \ddot{\mathbf{q}}_{des}.$$

Therefore, the combination of the feedforward compensator with the plant creates the identity system. As result, motor units for each degree of freedom suppress errors precisely like a canonical second-order system by computing the pattern of actuation required to suppress the coupling influences of adjacent degrees of freedom.

6.3.3 Analysis: The Dynamic Manipulability Ellipsoid

When $\dot{\mathbf{q}} \approx 0$ and ignoring torques due to gravity, the computed torque equation can be simplified to $\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}$. Under these circumstances, the tools for linear analysis developed in section 4.7 and appendix A.6 provide insight into the mapping between applied forces and accelerations in multi-degree of freedom articulated systems. For example, the generalized inertia ellipsoid describes the relationship between applied torques $\boldsymbol{\tau}$ and accelerations $\ddot{\mathbf{q}}$ in terms of the eigenvalues and eigenvectors of $\mathbf{M}\mathbf{M}^T$. This is the basis for a linear analysis describing, in relative terms, how the acceleration in the manipulator depends on the manipulator's configuration [208].

To get a measure of how the manipulator dynamics, gravity, the current state of motion, and even limits on actuator performance influence the ability to generate endpoint accelerations in Cartesian space,³ we start from the velocity relation $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$, where \mathbf{J} is the manipulator Jacobian. Differentiating with respect to time yields

$$\ddot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}.$$

Substituting the expression for $\ddot{\mathbf{q}}$ (equation 6.9, with external loads $\mathbf{F}=0$), we find

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{J} \left[\mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{V} - \mathbf{G}) \right] + \dot{\mathbf{J}}\dot{\mathbf{q}} \\ &= \mathbf{J}\mathbf{M}^{-1}\boldsymbol{\tau} + \dot{\mathbf{v}}_{vel} + \dot{\mathbf{v}}_{grav}, \end{aligned} \tag{6.13}$$

where we have dropped the explicit dependencies on \mathbf{q} and $\dot{\mathbf{q}}$ and introduced two new acceleration terms:

$$\dot{\mathbf{v}}_{vel} = -\mathbf{J}\mathbf{M}^{-1}\mathbf{V} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \text{ and} \tag{6.14}$$

$$\dot{\mathbf{v}}_{grav} = -\mathbf{J}\mathbf{M}^{-1}\mathbf{G}. \tag{6.15}$$

The $\dot{\mathbf{v}}_{vel}$ term includes centrifugal and Coriolis forces, and $\dot{\mathbf{v}}_{grav}$ captures forces due to gravity. The first term on the right-hand side of equation 6.13 depends on torque $\boldsymbol{\tau}$, and therefore the ability of the limb to generate accelerations depends directly on the physical limits of the actuators. If we follow the formulation in [47, 245] and assume symmetric torque limits

$$-\tau_i^{limit} \leq \tau_i \leq +\tau_i^{limit}, \quad i = 1, n,$$

then a normalized actuator torque, $-1 \leq \tilde{\tau} \leq +1$, can be written as

$$\tilde{\tau} = \mathbf{L}^{-1} \boldsymbol{\tau}, \quad (6.16)$$

where $\mathbf{L} = \text{diag}(\tau_1^{limit}, \dots, \tau_n^{limit})$. The set of admissible torques is, thus, a unit hypercube defined by $\|\tilde{\tau}\|_\infty \leq 1$.

Substituting $\mathbf{L}\tilde{\tau}$ for $\boldsymbol{\tau}$ in [equation 6.13](#) yields

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{JM}^{-1} \mathbf{L}\tilde{\tau} + \dot{\mathbf{v}}_{vel} + \dot{\mathbf{v}}_{grav} \\ &= \mathbf{JM}^{-1} \mathbf{L}\tilde{\tau} + \dot{\mathbf{v}}_{bias}. \end{aligned} \quad (6.17)$$

In this form, the term $\dot{\mathbf{v}}_{bias}$ accounts for all the velocity and gravitational effects—that is, all accelerations in the system that are present even when the command torque is zero.

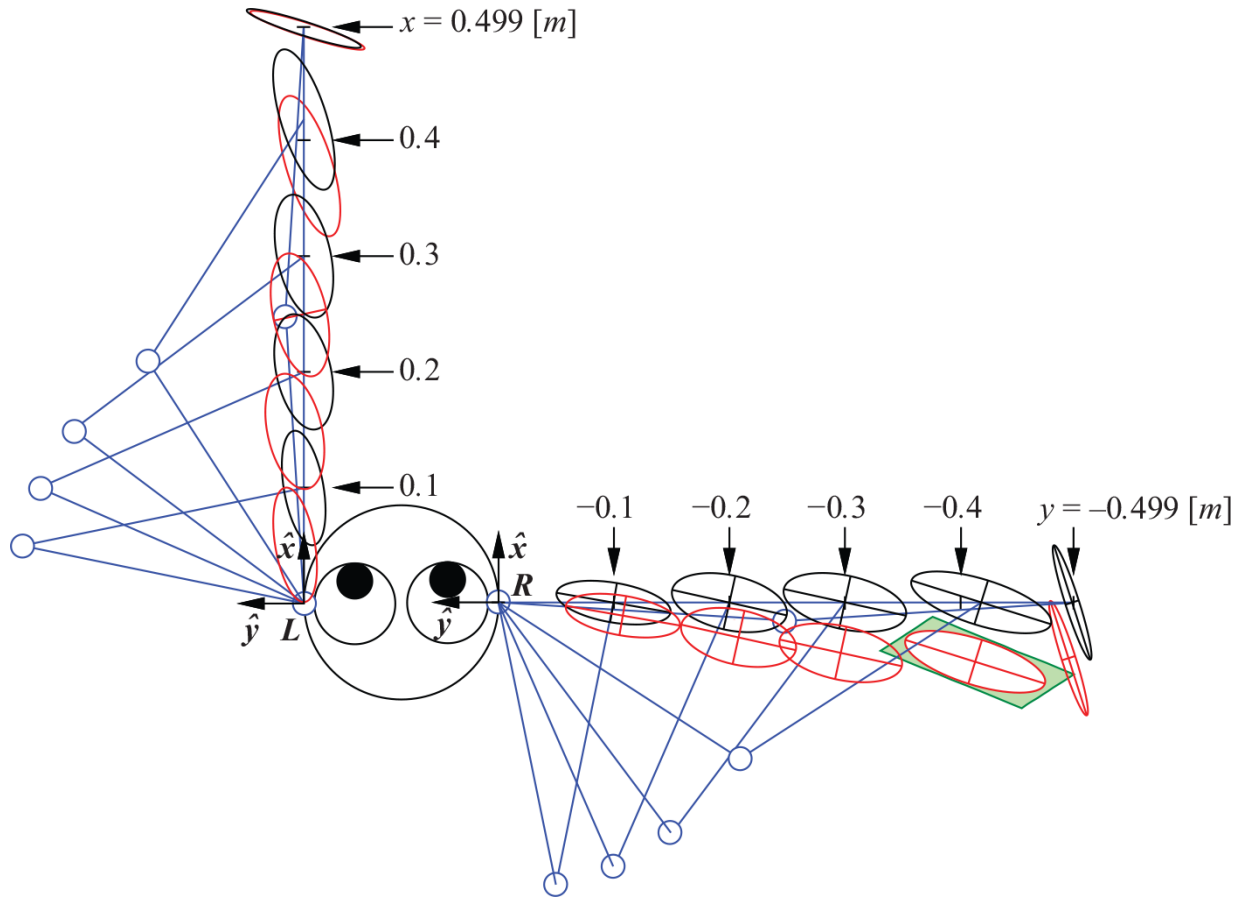


Figure 6.8

A smaller and dramatically underpowered pair (L and R) of Roger's 2R manipulators yields dynamic manipulability ellipsoid ($\mathbf{J}\mathbf{M}^{-2}\mathbf{J}^T$) for $l_1=l_2=0.25$ [m], $m_1=m_2=0.2$ [kg], and $\tau^T\tau \leq 0.005$ [N^2m^2]. Gravity acts in the negative \hat{x} direction.

Equation 6.17 maps the n -dimensional hypercube defined by $\|\tilde{\tau}\|_\infty \leq 1$ to an m -dimensional *acceleration polytope* [47] that defines the set of feasible end-effector accelerations. An illustration of an acceleration polytope for Roger is presented in figure 6.8. It describes the directional capacity for acceleration at the endpoint that includes the full dynamics of the limb and a symmetric model of torque limits in the actuators. Equation 6.17 can also be used to perform the now familiar elliptic deformation of an n -dimensional hypersphere, $\tilde{\tau}^T\tilde{\tau} \leq 1$, to an m -dimensional dynamic manipulability ellipsoid. In this case, the expression for the hypersphere is written in terms of the inertia matrix, \mathbf{M} , the Jacobian, \mathbf{J} , and the bias accelerations:

$$\tilde{\boldsymbol{\tau}}^T \tilde{\boldsymbol{\tau}} = (\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias})^T \left([\mathbf{J}\mathbf{M}^{-1}\mathbf{L}]^{-1} \right)^T \left([\mathbf{J}\mathbf{M}^{-1}\mathbf{L}]^{-1} \right) (\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias}) \leq 1.$$

Rosenstein [245] observed that since \mathbf{M} and \mathbf{L} are symmetric, this expression can be simplified,⁴

$$(\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias})^T \left[\mathbf{J}^{-T} \mathbf{M} \mathbf{L}^{-2} \mathbf{M} \mathbf{J}^{-1} \right] (\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias}) \leq 1, \quad (6.18)$$

so that

$$(\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias})(\ddot{\mathbf{r}} - \dot{\mathbf{v}}_{bias})^T \in \left[\mathbf{J} \mathbf{M}^{-T} \mathbf{L}^2 \mathbf{M}^{-1} \mathbf{J}^T \right]. \quad (6.19)$$

Equation 6.19 defines the elliptic deformation that characterizes the mapping from torques to end effector accelerations through the limb's dynamics. Yoshikawa [298] presented essentially the same formulation (without normalized torques) and called the elliptic mapping the *dynamic manipulability ellipsoid*. He also proposed the same kind of scalar conditioning metric used to measure kinematic conditioning,

$$\kappa_d(\mathbf{q}, \dot{\mathbf{q}}) = \sqrt{\det [\mathbf{J}(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{J}^T]},$$

and called it the *dynamic manipulability measure* [297, 298].

The dynamic manipulability metric is proportional to the volume of the dynamic manipulability ellipsoid and to the volume of the corresponding acceleration polytope in Cartesian space derived from the $\|\boldsymbol{\tau}\|_\infty \leq 1$ constraint.

Example: Gravity and Roger

Figure 6.8 illustrates the dynamic manipulability ellipsoid for several configurations of the arms of a smaller and underpowered Roger (without wheels) as it extends its arms to full length in the \hat{x} and \hat{y} directions. The lateral extension postures also show the principal axes of the dynamic ellipsoids. Black ellipsoids represent the unbiased Cartesian performance of the robot defined by equation 6.19 and red ellipsoids are gravity-biased dynamic manipulability ellipsoids scaled appropriately to reflect torque

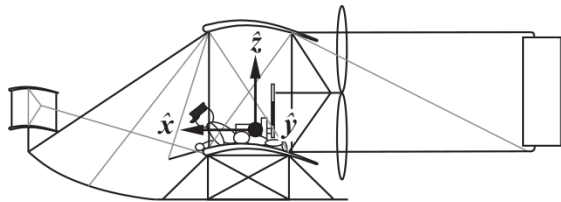
limits. An acceleration polytope is also illustrated in green for one of the candidate postures. This set contains all the accelerations possible from a given posture that are consistent with torque limitations in the manipulator.

The manipulator can support a candidate posture when the red ellipse contains the endpoint position for that posture—that is, when actuator torques can generate a net endpoint acceleration of zero. When released at rest from some of the postures shown, the manipulator will not be able to support the gravitational load. In particular, vertical extensions of the arm less than $x=0.3$ [m] and horizontal extensions greater than about $y = -0.175$ [m] cannot statically support the weight of the arm.

□

6.4 Exercises

1. **Rotating the inertia tensor.** The inertia matrix of the Wright Flyer relative to the x - y - z coordinate system located at its center of mass is shown in the following figure, where the \hat{x} axis is longitudinal and the \hat{y} axis is lateral.

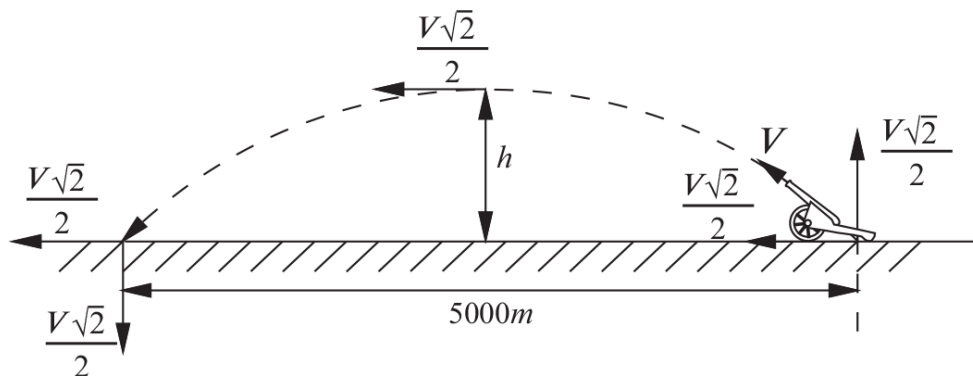


$$I = \begin{bmatrix} 11,755 & 0 & 4,821 \\ 0 & 15,000 & 0 \\ 4,821 & 0 & 23,245 \end{bmatrix}$$

- What could a—perhaps unintended—consequence be of designing control surfaces to produce torques for roll-pitch-yaw about the x - y - z axes given inertia tensor I ?
 - About which axis must you rotate I to produce I_{diag} ?
 - What angle (in radians or degrees) must the frame illustrated be rotated to diagonalize the inertia tensor?
 - Compute the principal moments of inertia.
2. **Coriolis forces on Earth.** A locomotive with a mass of 2×10^4 [kg] is traveling due north along a straight track at 40 [m/s] at $43^\circ N$ latitude. Starting with [equation B.7](#), calculate the magnitude and direction of the lateral force on the track.

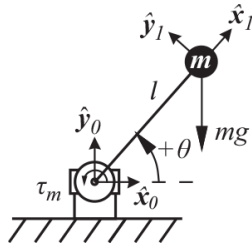
3. **Forensic dynamics.** Some historians believe that Gustave Coriolis (1792–1843) discovered the Coriolis effect early in the nineteenth century using observations of cannon accuracy during the Napoleonic Wars (ca. 1803–1815). The theory goes that the longer-range cannons were observed to miss reliably to the right. Could this theory be correct?

Assume that Coriolis fired a cannon due north in France (45 [deg] N latitude) and that the ball traveled at some unknown average velocity V throughout its flight. Further, assume an un-rifled cannon fired at a loft angle of 45 [deg] produced a shot with an average range of 5 [km] when fired on flat terrain and that lateral deviations due to turbulent airflow over the round projectile at this range could deflect the ball ± 100 [m].



- Determine the muzzle velocity for this firing geometry that produces the 5 [km] range.
- Estimate the duration/time of flight for the parabolic arc of the projectile.
- Compute the maximum altitude for the cannonball.
- Compute the Coriolis acceleration.
- How would a projectile appear to be deflected due to Coriolis accelerations?
- How plausible is the Napoleonic cannon theory as an empirical basis for the discovery of the Coriolis force?

4. **Simple, open-chain machines: 1R mechanism.** The mechanism illustrated is attached to inertial frame 0 and includes infinitesimal point masses at the end of each link that are subject to gravitational (mg) loads—each link is, otherwise, massless.

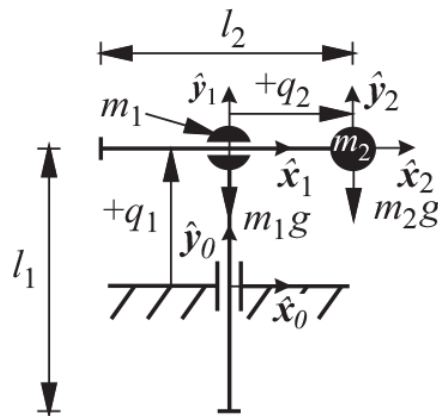


(a) Derive the equation of motion in the state space form for the simple mechanism (i) *by inspection*; (ii) using *the Newton-Euler equations*; and (iii) using *the Lagrangian*.

(b) Draw the *dynamic manipulability ellipsoid* ($\mathbf{JM}^{-2}\mathbf{J}^T$) for this configuration. On the same diagram draw a second ellipsoid illustrating the influence of the *gravitational bias*.

(c) Draw the *feedforward, dynamically compensated PD controller* (as in [figure 6.7](#)). Identify the required terms of the computed torque equation and their dependence on state feedback, and specify PD gains to produce a critically damped response. Explain why the feedforward compensation linearizes and decouples the system.

5. Simple, open-chain machines: 2P (Cartesian) mechanism.

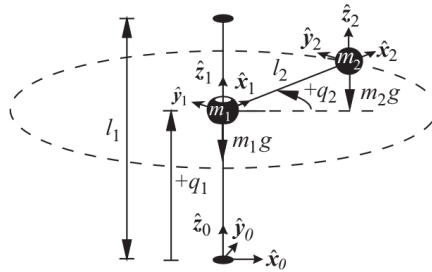


(a) Derive the equation of motion in the state space form for this simple mechanism (i) *by inspection*; (ii) using *the Newton-Euler equations*; and (iii) using *the Lagrangian*.

(b) Draw the *dynamic manipulability ellipsoid* ($\mathbf{JM}^{-2}\mathbf{J}^T$) for this configuration. On the same diagram draw a second ellipsoid illustrating the influence of the *gravitational bias*.

(c) Draw the *feedforward, dynamically compensated PD controller* (as in figure 6.7). Identify the required terms of the computed torque equation and their dependence on state feedback, and specify PD gains to produce a critically damped response. Explain why the feedforward compensation linearizes and decouples the system.

6. Simple, open-chain machines: PR mechanism.



(a) Derive the equation of motion in the state space form for this simple mechanism (i) *by inspection*; (ii) using *the Newton-Euler equations*; and (iii) using *the Lagrangian*.

(b) Draw the *dynamic manipulability ellipsoid* ($\mathbf{J}\mathbf{M}^{-2}\mathbf{J}^T$) for the configuration. On the same diagram draw a second ellipsoid illustrating the influence of the *gravitational bias*.

(c) Draw the *feedforward, dynamically compensated PD controller* (as in figure 6.7). Identify the required terms of the computed torque equation and their dependence on state feedback, and specify PD gains to produce a critically damped response. Explain why the feedforward compensation linearizes and decouples the system.

7. Invent your own homework. Make your own exercise out of your favorite content in chapter 6. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here. Ideally, it should be based on material in this chapter (and, perhaps, previous chapters); it should call for a quantitative analysis and a short discussion; and it should not take more than 30 [*minutes*] to solve open-book.

Part II Summary: The Kinodynamic Affordances of Embodied Systems

The sensorimotor stage in human development is a period when infants learn to manage their limbs and gross motor behavior. The knowledge acquired during this developmental stage is, to a large degree, focused on understanding the kinematics and dynamics of the body. In part II, we have reviewed some results from robotics that describe and characterize the kinodynamics of bodies in terms of their velocities, forces, and accelerations.

Mechanisms and Kinematic Relations—We reviewed some of the important terminology of mechanisms including the mathematical representation of spatial frames of reference. We used this framework to write forward and inverse kinematic relations for very simple devices including Roger's arms and eyes.

Jacobian and the Conditioning Ellipsoid—The degree of error amplification (or conversely, precision) in the kinematic mapping can profoundly influence the quality of sensory and motor behavior. We differentiated nonlinear forward models to reveal a locally linear velocity relationship called the Jacobian. The conditioning ellipsoid was derived as a method for characterizing the influence of the Jacobian mapping from configuration variables to spatial quantities. We noted that manipulators are not general purpose—they have a configuration-dependent ability to transform forces and velocities into the task space. Examples were presented that applied this analysis to Roger's manipulators and to his binocular vision systems.

Hands and Grasp Analysis—Seemingly minor variations in the kinodynamic configuration of arms and hands have made a significant evolutionary contribution to human intelligence. After a brief overview of the uniqueness of and importance of the human hand, chapter 5 provides a historical survey of influential robot hands that emerged as researchers learned more about the complementary nature of mechanism, control, and

representations for prehensile grasping. Following this survey, the grasp Jacobian and methods for solving for grasp forces given knowledge of the grasp geometry are presented.

Dynamics—We developed the governing dynamics for articulated mechanisms and the acceleration polytope and the dynamic manipulability ellipsoid. We reviewed extensions that explain how gravity and the state of motion can influence the ability of a robot to accelerate. These results describe the capabilities of the robot mechanism and its actuators.

These analytical tools describe how a robot’s body influences and sometimes dominates the control of motions and forces. There have been demonstrations of systems that learn to exploit dynamics in the literature. Robot weight lifters, for example, use nonlinear optimization to find motion plans for producing forces that significantly exceed the rated static payload of the robot [287, 247, 161]. They do so by storing kinetic energy and refocusing it via limb dynamics. Given knowledge of the robot, one can exploit such “kinodynamic lenses” using the techniques discussed in part II of the book.

In part III, we will examine where structure exists in the low-level feedback signals built into animals and robots. Then, in part in part IV, we will discuss supervisory control mechanisms capable of exploiting the kinodynamic aptitudes of robots.

1. A tensor is a geometric operator that describes relations between vectors and is independent of any particular choice for the coordinate system. Inner (dot) products and vector (cross) products are tensors, as is the inertia tensor that projects vectors of torques onto the vector space of accelerations.
2. For a rectangle with uniform mass density, the center of mass is the same as its geometric center.
3. In this discussion, we assume that the manipulator Jacobian is square and full rank. The general case is treated in [298, 208, 47] using the pseudoinverse (appendix A.9).
4. $\mathbf{A}^{-T} = (\mathbf{A}^{-1})^T$, $\mathbf{A}^{-2} = \mathbf{A}^{-1}\mathbf{A}^{-1}$, and for symmetric matrices, $\mathbf{A}^T = \mathbf{A}$.

III

STRUCTURE IN SENSOR FEEDBACK

7

Stimuli and Sensation: Organs of Visual and Tactile Perception

Animals (and autonomous robots) are embedded in ongoing sensorimotor interactions with a complex world. Stimuli originating in the environment can be used to activate purposeful movements that change access to these stimuli. Thus, animals and autonomous robots have the ability to actively control the type and quality of feedback as they reshape the environment to make it conform with the business of survival.

So far, our discussion has focused on how embodiment determines the relative quality of movement and perception, combining low-level issues related to actuation, control, and kinodynamics. The sensors we discuss in this chapter and the signal processing techniques we introduce in the next chapter complete the ground floor of an integrated cognitive architecture describing the possibilities for (extended) interaction with the world.

In this chapter, we introduce two principal sensing modalities—vision and touch—that contribute significantly to a human being’s *merkwelt* and that have direct analogs in robotics. So far, the robotic counterpart to these sensors is a poor approximation of the range of feedback available in animal systems, but they provide important, complementary contact and noncontact sensing of the external world. Together with proprioceptive feedback, they will contribute to the foundation of a growing cognitive agent.

The state of the art in these two sensors is also quite different. For example, parts of the human eye—those parts left after we exclude motor and behavioral systems that support vision—are similar to the now familiar cameras that are in our computers, cell phones, and doorbells. Therefore,

vision, in this chapter, will not focus on cameras per se, but will try to focus, instead, on how electromagnetic signals convey information that is influenced by interactions with the world and how one might go about building a practical, integrated, and interactive vision system for use by robots.

Tactile feedback is also critically important to human survival on Earth. However, the organs of touch are not as familiar and are less similar to their analogs in robot systems. Therefore, after a brief introduction to some of the mechanoreceptors in the human skin and muscle, we will spend some time surveying some of the common technologies for robot tactile sensing. In the following chapter, we will consider how signals from these devices can be processed to highlight information in the environment.

7.1 Light

Light is produced when an atom in an excited state *relaxes* into a lower energy state—the difference in energy is emitted in the form of a photon. *Black body* radiators are theoretical objects that absorb all incident radiation and emit electromagnetic energy over a spectrum of wavelengths determined by the body’s temperature. The net power radiated from such a body is the difference between the power emitted and absorbed and is determined by the Stefan-Boltzmann law,

$$P_{net} = A\sigma\epsilon(T^4 - T_0^4), \quad (7.1)$$

where A [m^2] is the surface area of the body; T [K] is the absolute temperature of the body in degrees Kelvin; T_0 [K] is the absolute temperature of the environment; $0 < \epsilon < 1$ is the emissivity of the body; and $\sigma = 5.670367 \times 10^{-8}$ [$W/(m^2K^4)$] is the Stefan-Boltzmann constant.

Our sun is essentially a black body radiator, and, therefore, the frequency spectrum of sunlight is a function of its surface temperature. In the absence of nearby matter, light energy radiated by the sun spreads uniformly across the surface of a sphere whose radius grows at the speed of light. Therefore, the energy density of light emitted from a point source at a distance R from the source is proportional to $1/R^2$. The sun delivers about 1200 [W/m^2] to the Earth’s surface when it is directly overhead.

The environment is bathed in electromagnetic energy with varying frequency content and intensity that originates in many phenomena: other black body radiators (like the glowing products of combustion and incandescent light bulbs); bioluminescence in organisms like fireflies and photo-luminescent plankton (a byproduct of chemical processes); and other man-made devices (e.g., LEDs and lasers).

7.1.1 Image Formation

The spatial distribution of electromagnetic energy is measured by projecting it onto a discrete array of receptive regions where it is transformed into electrical signals that constitute an *image*. The process of image formation in [figure 7.1](#) portrays a very brief history of a few rays of light as they leave multiple sources and make their way onto the image plane. Every illuminated object in the environment acts as a secondary light source by absorbing, transmitting, and reflecting incident radiation. On its way from the source to the image plane, light acquires geometric and chromatic artifacts from each of the objects it encounters en route.

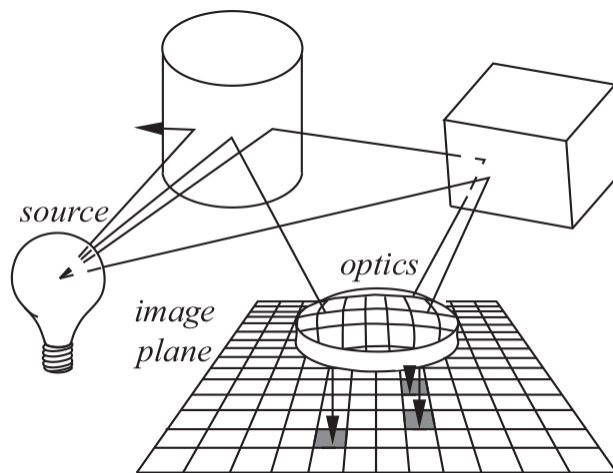


Figure 7.1

Projective geometry and image formation

To manipulate the path of a light ray, reflective and/or refractive optical elements are commonly used—biological systems make use of both. The result is a spatial distribution of pixel responses that represent a projection

of the *luminosity* function in the direction of the camera's gaze at time t that is refreshed with new data at (typically) 30 image frames per second.

In section 4.5.1, the kinematic properties of the pinhole camera were introduced to provide a basis for analyzing the relative acuity of stereo vision. The pinhole projective imaging geometry is ideal in many ways. Pinhole cameras create a focused image over an infinite depth of field. That is, light sources are in focus regardless of their distance from the camera. However, the pinhole only permits a very small amount of electromagnetic energy onto the image plane. Receptors, therefore, must be extremely sensitive. Optical elements can be used to gather light and bring it into focus over a *finite* depth of field. Together with mechanisms for actively controlling focus, optics can increase the size of the pinhole aperture significantly—gathering more of the environmental signal—while still providing all the information in the pinhole image. In this section, we will discuss refractive optical elements (lenses) that collect and focus light onto the imaging plane for this purpose.

Snell's law—The path of light is modified as it passes through the interface between optical materials with dissimilar optical properties. For example, when we examine an object submerged in water, the difference in the optical properties at the air-water interface causes light rays to be refracted. Most of us are aware of the fact that, under these circumstances, objects under the water are not precisely where they appear to be. The effect is diminished when the gaze is close to the normal of the water surface. In fact, we observe that the change in direction is proportional to the angle between the surface normal and the incident light. To understand why, we examine the physics of light.

A light ray is electromagnetic energy that consists of coupled, orthogonal electric and magnetic waves and propagates in the direction of the *Poynting* vector. The index of refraction for an optical material is the ratio of the speed of light in a vacuum to that in the optical material.

$$n = \frac{c}{v} = \sqrt{\frac{\epsilon\mu}{\epsilon_0\mu_0}}, \quad (7.2)$$

where μ is the magnetic permeability of the material (μ_0 is the permeability of free space) and ϵ is known as the electric permittivity (ϵ_0 is the

permittivity of free space).

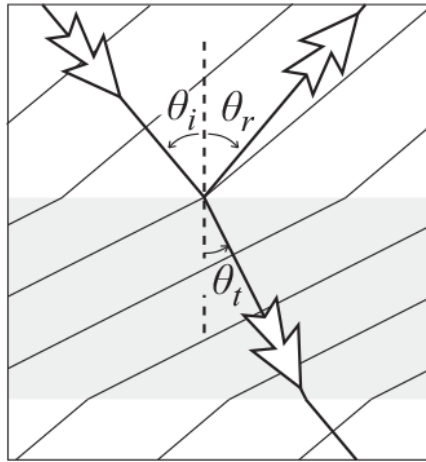


Figure 7.2

Refraction at an optical interface. An incident light ray traveling in air and an optical interface with $n > n_{air}$.

As illustrated in [figure 7.2](#), when an electromagnetic wavefront crosses an optical boundary from air into a material whose index of refraction is greater than that of air, the light rays are refracted, bending the wavefront toward the normal of the optical interface. The phenomenon is described by Snell's law,

$$\frac{\sin(\theta_i)}{\sin(\theta_t)} = \frac{n_t}{n_i}, \quad (7.3)$$

where n_i is the index of refraction for the medium through which the incident light ray travels, and n_t is the index of refraction for the medium through which the transmitted light ray travels. The geometry and optical properties of lenses are designed to manipulate the geometry of light rays to focus this energy into coherent images.

The Gaussian Lens Formula—Most lenses combine pairs of convex and/or concave spherical surfaces whose centers of curvature are located on the optical axis. By convention, the surfaces of the lens have a positive radius if the center of curvature is to the right of the lens and negative if it is to the left. Therefore, R_1 is positive and R_2 is negative for the biconvex lens shown in [figure 7.3](#).

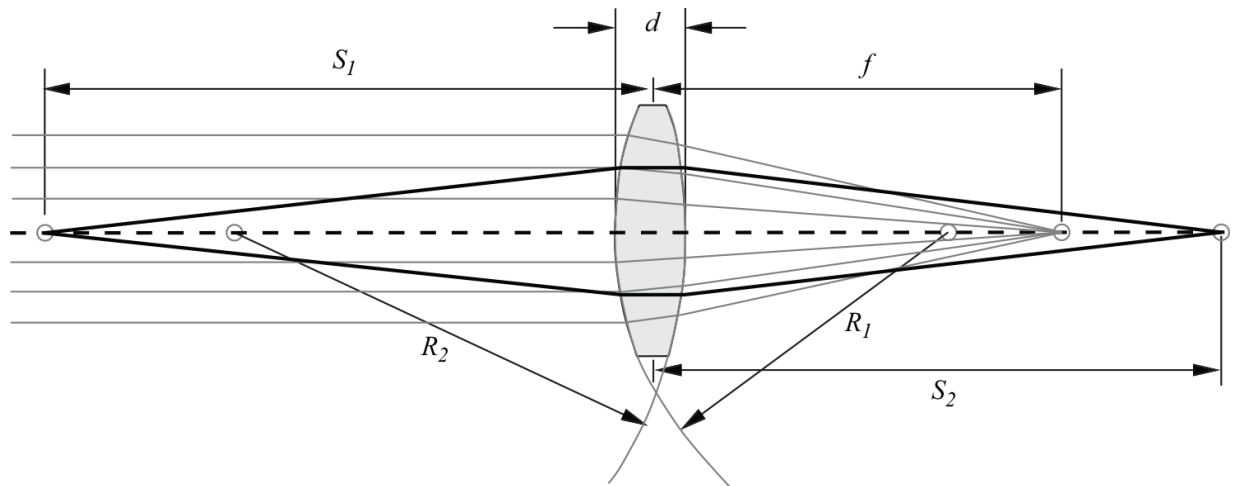


Figure 7.3

Geometric parameters for the biconvex lens

Incoming parallel light rays on the left of the figure originate from an infinitely distant source and are focused at a distance f (the focal length) to the right of the lens. By combining Snell's law and the geometry of the lens, the focal length of a lens like this in air ($n_{air} = 1.0$) can be written

$$\frac{1}{f} = (n - 1) \left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n - 1)d}{nR_1R_2} \right], \quad (7.4)$$

where f is the focal length, $n > 1$ is the refractive index of the lens material, and d is the thickness of the lens at the optical axis. A thin lens approximation can be used when d is small compared to R_1 and R_2 . Under these conditions, [equation 7.4](#) can be simplified to yield

$$\frac{1}{f} \approx (n - 1) \left[\frac{1}{R_1} - \frac{1}{R_2} \right]. \quad (7.5)$$

As the infinitely distant source approaches from the left, light rays from the source are increasingly divergent as they reach the lens, causing the point-of-focus to move further to the right. This case is also depicted in [figure 7.3](#).

Mechanisms for focusing cameras change the distance between the lens and the focal plane in order to bring objects at different depths into focus. The *thin lens equation* ([equation 7.6](#)) describes where objects at different ranges will be focused.

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f}, \quad (7.6)$$

where S_1 is the distance from the lens to the object; S_2 is the distance from the lens to the place where the image is formed; and f is the focal length of the lens.

If an object is placed at $f < S_1 \leq \infty$, then a focused image will be produced on an imaging surface at a distance S_2 accordingly. Small displacements from this position will cause the image to defocus. Objects at $S_1 = f$ will focus at $S_2 = \infty$, and $S_1 < f$ will create divergent rays that cannot be focused at a positive S_2 .

Modern high-performance cameras make use of optics based on these principles, together with a mechanical means of changing the distance of the lens from the image plane and algorithms for focusing on the subject that minimize image distortion and work well at low ambient light levels. Natural selection has produced a variety of species with eyes that exploit the same principles supported by an expanded array of reflexes for optimizing signal quality and protection.

7.1.2 The Evolution of the Human Eye

The human eye is a complex organ whose basic design is shared by several species. Many variations of the eye across species today relate to a common basic design that dates back perhaps 540 million years. [Figure 7.4](#) illustrates a sequence of five milestones in the evolution of a contemporary eye common to many animals. The first organisms that developed light sensitive nerve cells probably lived in the shallow seas where life on Earth is believed to have begun. These *eyespots* resembled the structure in [figure 7.4\(a\)](#), and provided ancient animals with the ability to differentiate light from dark or perhaps to detect when light was occluded by looming predators. Animals with these sensors couldn't resolve directions well, but they could distinguish night from day, set circadian rhythms, and, over time, became selectively responsive to the band of the electromagnetic spectrum that penetrated the shallow, salt water seas in which they lived.

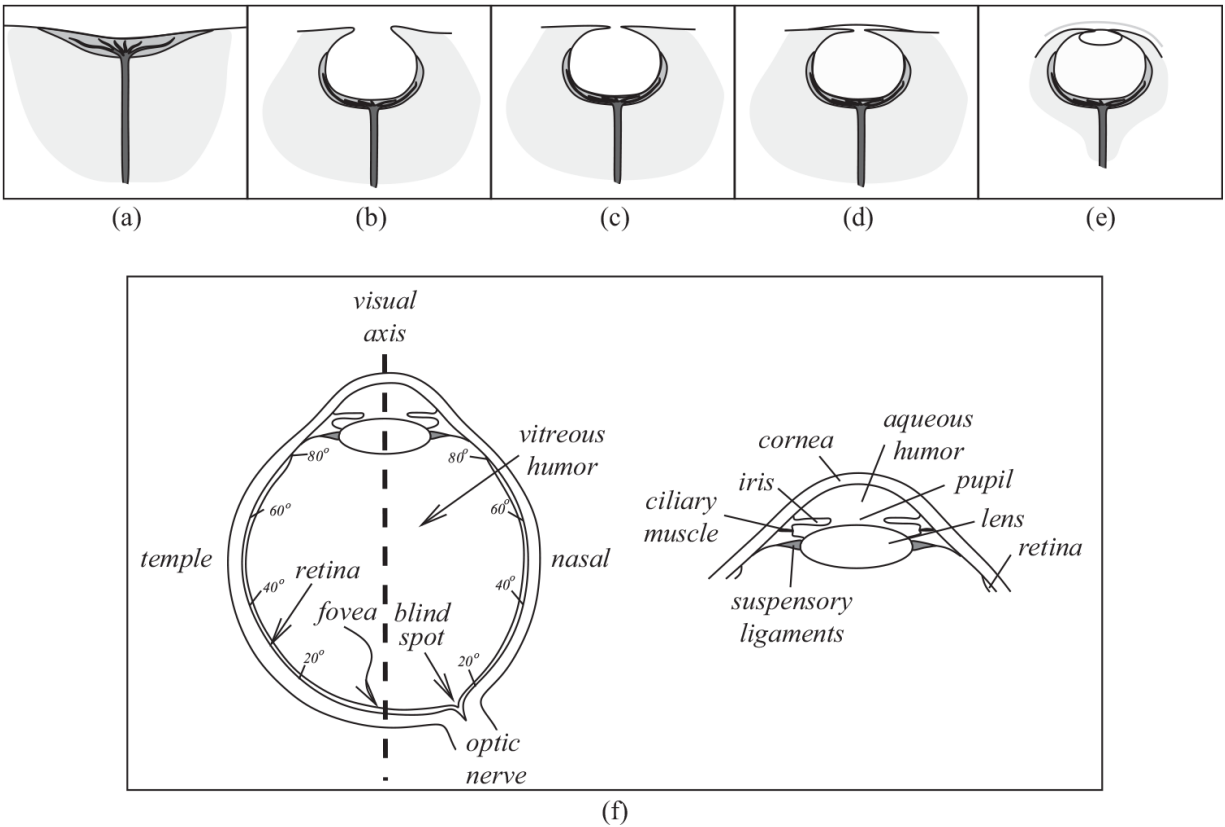


Figure 7.4

Milestones in the evolution of the human eye: (a) the eyespot consists of a layer of photosensitive cells; (b) a pit forms that makes the primitive retina directionally sensitive; (c) the pinhole eye filled with water to create sharper images; (d) transparent aqueous humor fills cavity with transparent corneal covering; (e) the lens forms; and (f) the human eye developed a separate iris, focusing musculature, and spherical eyeball enclosed in a bony socket. (Source: Adapted from [100] and [28])

The eyespot eventually evolved into a pit organ ([figure 7.4\(b\)](#)) and further into the pinhole eye ([figure 7.4\(c\)](#)) with much greater ability to focus and improved directional sensitivity. This eye can *image* the environment and discriminate between different shapes and brightnesses. The contemporary nautilus, a small mollusk with tentacles like a squid, uses one of these ancient eyes. Without a lens to gather light, these eyes are still relatively poor, but animals with such an eye can now hunt visually as well as flee when being hunted. Specialized transparent epithelial cells evolved to cover the pit organ, as in [figure 7.4\(d\)](#), to exclude contaminants from the imaging chamber and to contain the special vitreous humor. This fluid creates a greater index of refraction and supports vision in animals that move out of the oceans and onto the land. The transparent epithelial cells

evolved to form two layers with a fluid (the aqueous humor) between them. Optical properties were again enhanced, but the mechanical toughness of the outer structures were also improved, as well as the metabolic support of the tissues involved. Eventually, the lens (figure 7.4(e)) and the musculature to focus the eye actively and to control the amount of light entering the eye evolved (figure 7.4(f)) to create an imaging device whose basic form and function would be replicated in many species from this common lineage.

The photosensitive surface of the eye is called the retina and in some animals, including humans, consists of rod and cone receptors. The rods are more sensitive to overall light intensity and are about twice as numerous as the cones. The cones, however, are specialized receptors responding to red, green, and blue wavelengths. The greatest concentration of receptors is near the fovea. The entire array functions as a parallel image capture and signal processing architecture. There are about 100×10^6 receptors on the retina. When stimulated, they produce impulses in adjacent retinal cells. This early processing system computes properties of the image in compact retinal neighborhoods and compresses the signal significantly—only about 0.8×10^6 nerve fibers exit the eyeball via the optic nerve.

Human oculomotor anatomy—Figure 7.4(f) is an anatomical sketch of the human eye. It is roughly spherical in shape, about 2.4 [cm] in diameter and weighs about 7.5 [g] [28]. The eyeball itself is housed in a bony orbital cavity and consists of a larger posterior cavity filled with vitreous humor, which is a jellylike substance formed early in fetal development and never replaced. The vitreous humor is both structural and optical. Light enters the eye through the structures of the anterior cavity of the eyeball, which is filled with aqueous humor produced in the ciliary processes and flows through the pupil, where it is absorbed by the venous system. It maintains intraocular pressure and links the lens and the cornea to the circulatory system. Common disorders, like glaucoma, can result if the intraocular pressure is not regulated adequately.

The white of the eye (sclera, Greek *skleros*, “hard”) and choroid structures are both structural and optical. In this case, opacity is the important optical property and this is accomplished in part by the pigments in the choroid that block stray light from striking the light-sensitive retina. The pigmentation also includes the colors in the iris that have evolved in

response to ambient lighting conditions at various places on earth. The iris contains radial and circumferential fibers that surround the pupil. They behave as a diaphragm controlled by the autonomic nervous system to shutter incoming light.

The lens of the eye would normally assume a roughly spherical shape but is elongated slightly by tension in the suspensory ligaments. The ciliary muscle forms a ring between the iris and the lens that, when innervated, reduces tension in the ligaments and thereby improves the focus on nearby objects by decreasing focal length (equation 7.5). The ciliary muscle and the iris control intrinsic parameters of the stereo imaging system. The *pupillary light reflex* is a kind of closed-loop regulator that constricts or dilates the iris in response to ambient light levels. When an eye is exposed to changing stimuli, it responds and there are sympathetic responses in the other eye even if it is not similarly stimulated. These reflexive responses are often used to test for intact neural pathways.

In addition to the intrinsic ciliary muscle and iris, there are several extrinsic muscles that participate in the control of the binocular field of view. Figure 7.5 illustrates some of this musculature. *Squinting* can influence the shape of the lens and affect the intensity of incident light under voluntary control. Moreover, a parasympathetic response referred to as the *accommodation reflex* engages a feedback loop via the primary visual cortex to condition the visual system's near-field acuity. In response to an approaching object, the ciliary muscle relaxes appropriately to plump both lenses by virtue of their passive elasticity, the binocular configuration verges on the subject to reduce double vision and to keep the stimulus near the respective fovea, and pupillary constriction (miosis) improves visual acuity by using the iris to narrow the pupil toward a pinhole aperture (see section 4.5).

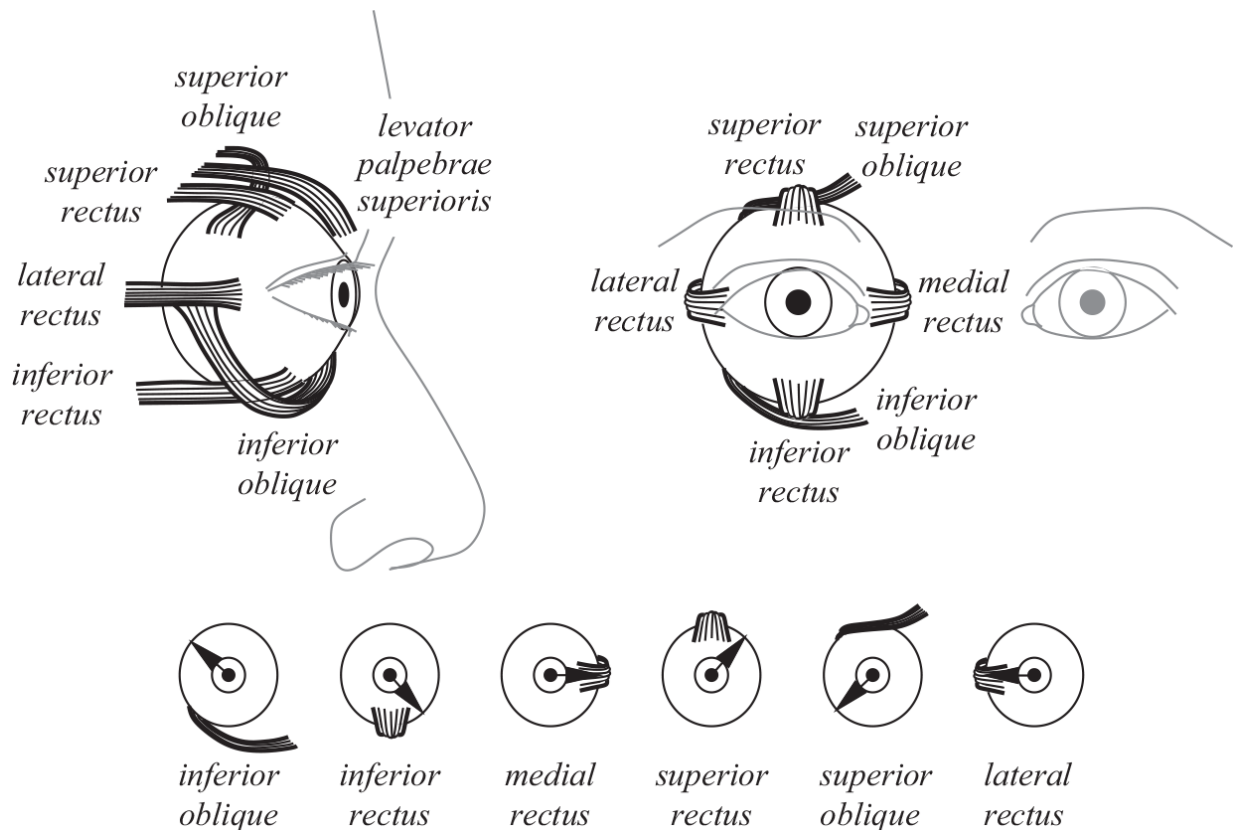


Figure 7.5

The extrinsic muscles for the oculomotor system. The oculomotor nerve innervates the inferior oblique; the inferior rectus; the medial rectus; and the superior rectus. The trochlear nerve innervates the superior oblique, and the abducens nerve innervates the lateral rectus. The inset at the bottom shows the displacement of the pupil due to the exclusive action of the muscles.

Variations on the Theme—In terms of structure, all vertebrates have eyes similar to ours. Invertebrates generally have poorly developed or compound eyes, except for the octopus [84]. The octopus has a well-developed brain and has evolved an eye much more like ours, with a cornea, eyelids, iris, and lens. This is an example of evolutionary lightning striking twice and creating essentially the same design via a dramatically different evolutionary thread. Other species have tuned the basic design of their eye to optimize its performance in particular environments and ecological niches. For example, bees, fish, butterflies, birds, and reptiles are all capable of seeing color, but most mammals do not (with the significant exception of the primates). Much of this commitment to color seems to be in service to reproductive success or to help distinguish flowers and fruits for food.

More examples of variations on the basic design of the vertebrate eye are found all over the animal kingdom. Herbivores often have side-facing (monocular) vision systems with almost no overlap in the field of view. They rely on a larger panoramic field of view more than stereo vision. The chameleon has eyes on independently controllable turrets capable of both side- and forward-looking configurations. However, predators are distinguished by their forward-facing (stereo) system that provides precision near-field depth perception with a narrow field of view (<180 [deg]). Rattlesnakes (a predator) have their eyes oriented to the side but also have forward-looking stereo pit organs, an ancient eye with no lens that responds selectively to infrared radiation (heat). They use the pit organs to orient their strike and the visual spectrum to detect motion over a wide field of view.

Cheetahs have a wide, eccentric foveal region spanning a broad horizontal band, presumably for locating prey on the African plains. Other forms of retinal variation serve nocturnal animals by employing a reflective back surface of the receptive cells to catch stimulus that might otherwise slip through unnoticed and bounce it back through the receptive tissue a second time. This is the reason why a deer's eyes will glow in a car's headlights and why photographs of people taken with a flash will often show red eye. Many species of small birds sacrifice muscles for size since space in their small skulls comes at a premium. Small owls, for instance, need large eyes to hunt at night and some species have evolved without the ability to pan their eyes within their skull in order to gather more light. These species direct their gaze using movements in the neck. The largest eye in nature belongs to the giant squid and may be up to 15 [in] in diameter.

Finally, fishing birds exploit the fact that light reflected from flat surfaces like roads or smooth water is polarized horizontally. The kingfisher, for example, uses a polarizing lens to filter out reflected glare in order to see its prey more clearly.

7.1.3 Photosensitive Image Planes

During the twentieth century, several devices were developed to emulate the eye's ability to create images—signals with both spatial and temporal

content that can be broadcast for viewing elsewhere and can be inputs to computational vision systems. Researchers have designed an analog of the human retina using integrated CMOS image capture chips [183]. These devices are fast, low power, and compact by virtue of decreasing pixel size and has become inexpensive due to its ubiquitous use in modern cell phones and web cams.

The solid state, charge-coupled device (CCD) has become synonymous with modern video signals. However, it is not a method for turning light into electrical charge; it is a high-speed analog shift register. It allows analog signals to be transferred along a chain of solid state capacitive elements under the control of a clock signal. The CCD implements a form of solid state “bucket brigade” for sampled analog signals that is relatively power efficient, fast, and free of the distortions common to previous methods.

The photoreceptors in modern solid state cameras detect a wide range of frequencies from infrared to ultraviolet and are quite sensitive, which makes them extremely valuable in astronomy. The Hubble Space Telescope uses a CCD and techniques for eliminating random thermal noise to create sharp images in the extended visible spectrum. Color images are created using CCD arrays by tiling the image plane with pixels that respond selectively to red, green, and blue wavelengths. Every 2×2 square of pixels uses filter masks to create one red, one blue, and two green receptive fields. More expensive color imaging devices use three CCD arrays (3CCD) and a beam splitting prism to separate white light into red, green, and blue components.

7.2 Touch

Haptics concerns the sense of touch—in particular, the perception of movement and contact using tactile and proprioceptive feedback. In human beings, it engages a variety of central and peripheral neural systems that measure forces, heat flux, pain, accelerations, and the degree of stretch in muscle fibers and tendons. Many of these haptic signals originate in a variety of mechanoreceptors with overlapping receptive fields that could number in the hundreds per cubic centimeter for tissue near the skin surface. They contribute to our sense of shape and material (thermal conductivity, hardness, texture), provide feedback to help stabilize a grasp,

and perceive a variety of other sensations associated with contact phenomena. Without these signals, as happens under anesthesia or when our hands get very cold, human beings experience profound difficulty with the simplest and most mundane activities of daily life. The quality of our haptic experience is due to the collective behavior of peripheral nerve cells.

7.2.1 Cutaneous Mechanoreceptors

A variety of specialized cells have evolved to measure the state of the musculoskeletal system (proprioceptive sensors) and external stimuli (exteroceptive sensors). Among these, there is a special class of mechanoreceptor that converts mechanical and thermal energy into electrical signals in the nervous system that contribute to a sense of touch. Some of these specialized cells are presented in [figure 7.6](#). Mechanoreceptors are specialized peripheral nerve cells that detect stresses of various forms related to contact and movement. The probability that the mechanoreceptor's cell membrane produces a spike (the action potential) is proportional to the magnitude of the stressor.

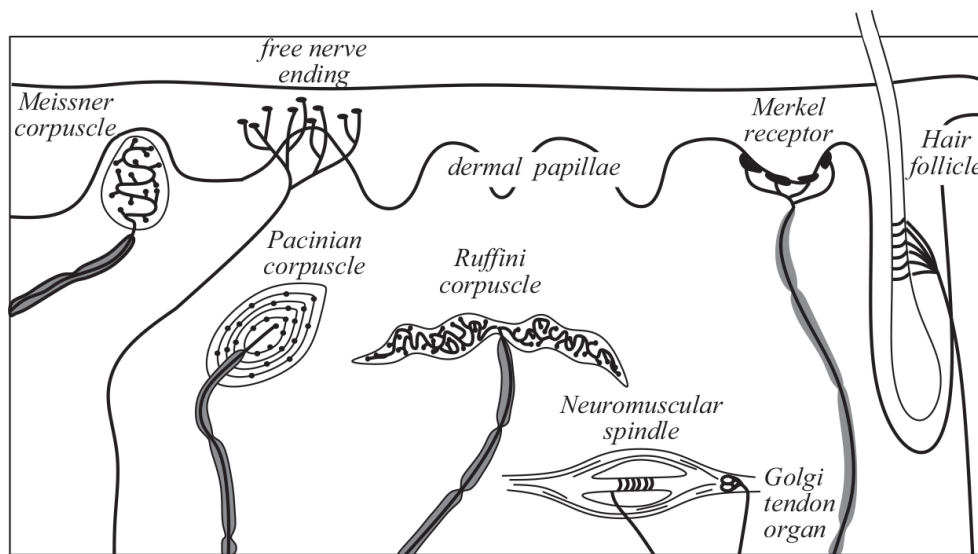


Figure 7.6

Some of the cutaneous sensory neurons that provide feedback signals to the somatosensory cortex

The collection of mechanoreceptors in [figure 7.6](#), responsible for the sense of touch, are organized conceptually in terms of their typical depth in the (sub)cutaneous tissue.

Internal receptors—In general, sensorimotor control requires both position and force feedback. Internal receptors measure the relative position of body parts—a sensor modality known as proprioception. Four receptors of this type are described below [60, 163].

- *Neuromuscular spindles* sense the degree of stretch in the muscle fiber. These receptors contribute feedback support for the reflexive cohesion of the skeleton and provide precise movement control. They come in two varieties: one responds to high frequencies and another responds to low frequencies and DC traction.
- The *Golgi tendon organ* exhibits a very slow response and contributes to the control of muscular tension by measuring the degree of stretch of the muscle fiber at the tendon-muscle interface.
- The *articular surfaces* in the joints of the body produce signals proportional to extreme position, velocity, or ligament tension. These receptors provide feedback for extreme movement of the joint.
- *Ruffini corpuscles* function, in part, as thermal receptors and contribute to the sensation of kinematic forces and accelerations.

Dermal receptors—Beneath the epidermal layer of the skin, the dermis incorporates two types of receptors [60]:

- *Meissner corpuscles* respond to light touch; they are located beneath the convoluted dermal papillary layer. These structures are ovoid, with the major axis perpendicular to the skin surface, and contain neural fibers running normal to and tangential to the skin surface. The Meissner corpuscles are specialized high-frequency transducers.
- *Pacinian corpuscles* (as distinguished from an intramuscular Pacinian-type receptor) respond best to accelerating mechanical displacement rather than constant velocity deformation. These sensors are quite sensitive to pressure but are not sensitive to direction, and they respond primarily to vibration stimuli.

Epidermal receptors—Epidermal receptors provide the most direct evidence of interactions between the skin surface and the external world. [Figure 7.6](#) illustrates three different types of receptors:

- Hair is the mechanical extension of the *hair follicle*. It measures deflections in the follicle caused by contact. The hair on the surface of our skin is the site of the greatest rate of on-going cell division in the body. Hair follicles create new cells and extrude a hair shaft that extends beyond the surface of the epidermis. Nerve cells in the follicle detect deflections in the hair shaft as, for instance, when brushed by environmental contacts, or subjected to air or fluid flow. In some animals, stiff hairs (whiskers) act as proximity sensors. Attached to each epidermal hair follicle is a tiny bundle of muscle fiber called the arrector pili that make the hair stand up and cause the follicle to protrude slightly above the surrounding skin (goose bumps). This is an evolutionary adaptation to make the animal seem larger in a defensive posture or to warm the animal by insulating the skin with a thin blanket of stationary air.

- *Merkel disks* are sensory receptors with a large bandwidth that are sensitive to compression and shear stimuli. They are located in relation to the dermal papillae in a position that localizes the strain associated with small contact forces on the surface of the skin.

- *Free-ended nerve fibers* respond to a variety of stimuli, including temperature and pain. These receptors are classified by their diameter, conduction speed, sensitivity or threshold level, and the presence or absence of a myelin sheath. The presence of a myelin sheath increases the diameter and induces a different conduction path in the nerve fiber. As the diameter increases, the threshold stimulation decreases, the amplitude and duration of the output signal increase, and the velocity of the signal increases. A- and C-type free-ended receptors are prevalent in the epidermis. Type A has a relatively large diameter and is myelinated while type C is smaller, more numerous, and unmyelinated.

The data in [table 7.1](#) was compiled from a variety of sources [[60](#), [97](#), [112](#), [136](#), [163](#)] to summarize the collective performance of the sensory apparatus in the human hand. The relative contribution of the various sensor signals in the human hand produces a very rich and diverse array of tactile sensation. The so-called *touch blend* [[255](#)] of temperature, pressure, and vibration signals is capable of distinguishing wet, slimy, greasy, syrupy, mushy, doughy, gummy, spongy, or dry, and properties such as hardness, texture, compliance, size, shape, and curvature can be estimated [[76](#), [266](#)].

Movement is critical to the formation of haptic images, and active exploratory strategies that make efficient use of tactile exploration have been reported [4, 69, 77, 97, 104, 114, 255].

Table 7.1

Sensory signal performance on the fingertip

Frequency response	0 to 400 [Hz] (+ very high frequency)
Response range	0 to 100 [g/mm] ²
Sensitivity	Approx. 0.2 [g/mm] ²
Spatial resolution	1.8 [mm]
Signal propagation	Motor neurons 100 [m/sec] Sensory neurons 2 to 80 [m/sec] Autonomic neurons 0.5 to 15 [m/sec]

7.2.2 Robotic Tactile Sensing

Technologies for robotic tactile sensing are diverse, ranging from simple binary contact sensors to artificial skins that provide complex, wide spectrum stress/strain and thermal conductivity information at the interface between a robot and the environment. However, tactile systems for use with robots are quite primitive relative to their biological counterparts, and there is a huge gap between the haptic abilities of animals and those of robots. The difficulty lies, in part, in our limited ability to fashion small sensors that can be distributed in tough, elastomeric robot “skin” that is easily repaired; that requires few electrical connections and is addressable; and whose signals convey complementary forms of information efficiently.

This section provides a historical overview of tactile technologies by considering a few innovative concepts for deriving information from contact. This is a very active area of research, and this review will not attempt to be comprehensive or cutting-edge. Instead, we focus on commonly used transducers and design concepts. For a more thorough review, the interested reader is directed to [66, 169, 170].

Several characteristics of viable contact sensor technologies are important when specifying a tactile sensor for general use in a dexterous robot [103]. The maximal *spatial resolution* of a particular sensor is determined by the maximum spatial density permitted in its fabrication. The *spatial frequency* observable in tactile signals is defined by the resolution of

the sensor. A typical specification calls for spatial resolution on the order of 1–2 [mm]—roughly, the minimum two-point tactile separation observable on the fingertip of the human hand. The *temporal bandwidth* is determined by the rate of data acquisition and bandwidth requirements depend on the task. For example, slip detection or active texture analysis requires relatively high-frequency signal content. The sensor *sensitivity* measures the ratio of the change in the sensor output to a change in the input, and the *dynamic range* of a sensor is the logarithm of the difference between the smallest and largest signal values that are detectable. *Hysteresis* in a sensor causes its output to depend on the recent history of inputs. In response to periodic stimulation, the output of the mechanoreceptors in [figure 7.6](#) show a fair degree of hysteresis; however, hysteresis in robotic tactile sensors proves to be very challenging for roboticists. Finally, roboticists must consider the means of *addressing* tactile patches and making room for the electrical connection from sensor to controller. The *number of wires* needed to access an array of sensors varies with the technology used and the scheme employed to address the sensor patch [[130](#)].

Binary contact switch—The most straightforward tactile sensor is perhaps the binary contact switch. It has been used in a wide variety of applications: homing robots by providing a reference-zero position for each degree of freedom; providing bumpers for mobile robots; or, arranged in spatial arrays, improving the precision of contact position and force observations. For example, Raibert [[238](#)] developed a tactile sensor by arranging binary contact sensors inside tapered separating spaces and used local VLSI processing to produce a smart sensor. A sketch of a separator cell is shown in [figure 7.7](#). The tapered cell yields an increasing contact force threshold for each successive switch, creating, in this case, an eight bit binary word encoding the contact pressure over the cell. A prototype chip using 48 tactile cells in a 6×8 array was constructed to produce 4 bits of pressure output per cell. Using serialized input and output data, a full-scale prototype incorporated 200 tactile cells with a 1 [mm] spacing driven by only five wires: power, ground, clock, data-in, and data-out.

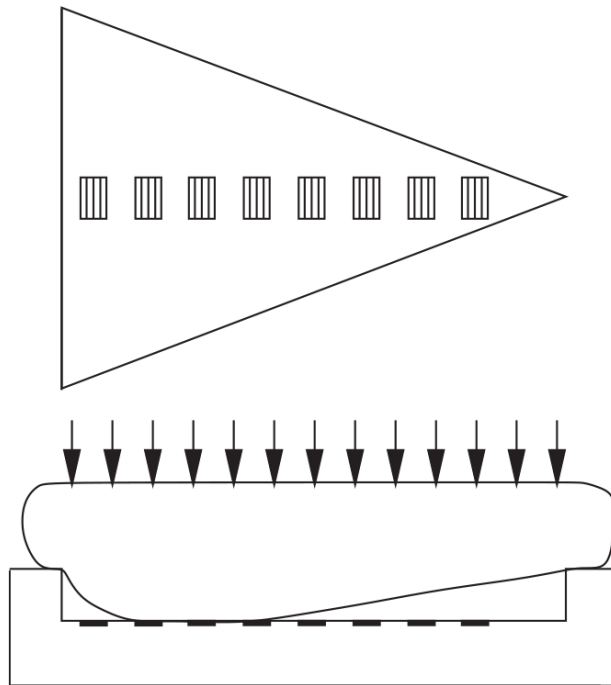


Figure 7.7

Raibert's VLSI tapered tactile cell. (Source: Adapted from [238])

Load cells—Strain gauges are simple electromechanical devices that change resistance in response to mechanical strain (deformation). They are electrical conductors that are bonded to mechanical members such that deflections in the underlying structural element also cause a small deflection in the sensor element. As the strain gauge flexes, small changes in its cross-sectional area create measurable changes in the electrical resistivity of the strain gauge. Given knowledge of the elastic modulus and geometry of the structural element, these changes in resistance can be used to compute the forces that cause these minute deformations. Strain gauges have been used for quite a long time to estimate loads in large structures such as bridges and buildings.

This idea has also been used to measure forces in robot hands [301]. Multiple strain gauges (six or more) can be configured to measure the full six-dimensional state of stress in a body—three components each of force and moment. Very small versions of six-axis load cells have been fabricated as contact sensors ([73], for example) and incorporated into hemispherical fingertips to determine contact loads [251, 29, 66] and extended to estimate contact normals. This technique yields high-precision contact position and

force measurements, with good sensitivity and dynamic range. However, integrated load cells can be relatively expensive and can undergo plastic deformations if they are subject to excessive loads.

Three-axis load cells using the same strain gauge technology are being developed for use in tactile arrays. MEMS fabrication techniques have been used to build silicon load cells with three (and fewer) axes that are small enough to embed in elastomeric skin [281, 18]. Emerging fabrication techniques make it conceivable that many such devices can be embedded in moldable elastomers to sense the distribution of strains in the robot's structure, actuators, transmissions, and coverings.

Conductive elastomers—Conductive elastomers have been used extensively in the design of tactile sensors [16, 69, 103, 105, 114, 238]. Since it is desirable to cover the tactile system with a compliant layer [83], it might be advantageous to use a conductive rubber, a doped rubber, or a conductive foam to act both as protective skin and pressure transducer. These materials produce predictable changes in resistivity when compressed. However, currently available materials are often hysteretic, and doped materials are generally not very rugged. Moreover, low sensitivity, noise, drift, long time constants, and low fatigue life pose problems in these transducers.

Hillis [114] addressed these limitations by proposing the sensor illustrated in figure 7.8. The sensor employs a conductive silicone rubber that deforms around the separator under load to come into contact with an underlying conductor. Increased contact pressure increases the size of the contact patch and, thus, decreases the contact resistance. Since the signal is proportional to the contact resistance rather than the point-to-point resistance in the silicone material, different separators can be designed to produce the correct sensitivity, resolution, and dynamic range for different applications. Hillis constructed an array of 256 tactile sensors with a spatial resolution of about 1 [mm] using this technology. A 16×16 array of 256 tactile cells was addressed by row and column using 32 wires in a cable less than 3 [mm] in diameter.

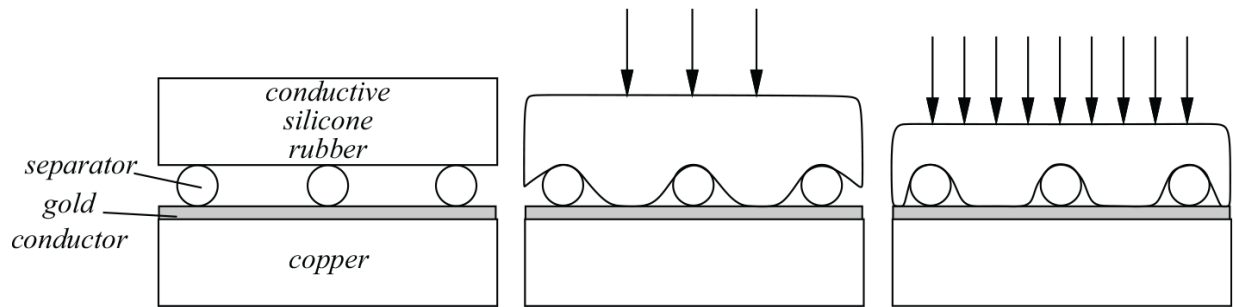


Figure 7.8

Hillis's elastic contact resistance sensor. (Source: Adapted from [114]. © MIT Press)

Optical technologies—Tactile sensors have been designed that change the intensity of an optical signal in response to contact loads. For example, a deformation caused by contact forces can be used to shutter a beam of light [60, 240]. The sensitivity and dynamic range of such a sensor can be changed by altering the mechanical properties of the sensor without altering the optical transduction. Commercial systems of this type are available [179] and have been used in tactile image research [206].

An interesting approach in the literature employs a phenomenon known as “frustrated total internal reflection” [19]. The effect is demonstrated by looking straight down into a cold glass of water on which condensation has formed. The silvery condensate reflects light—hiding objects outside the glass unless they come into contact with the glass’s surface. The more forceful the contact, the more prominent the image of the contact becomes on the otherwise silvery glass surface. A tactile sensor based on this effect is in figure 7.9.

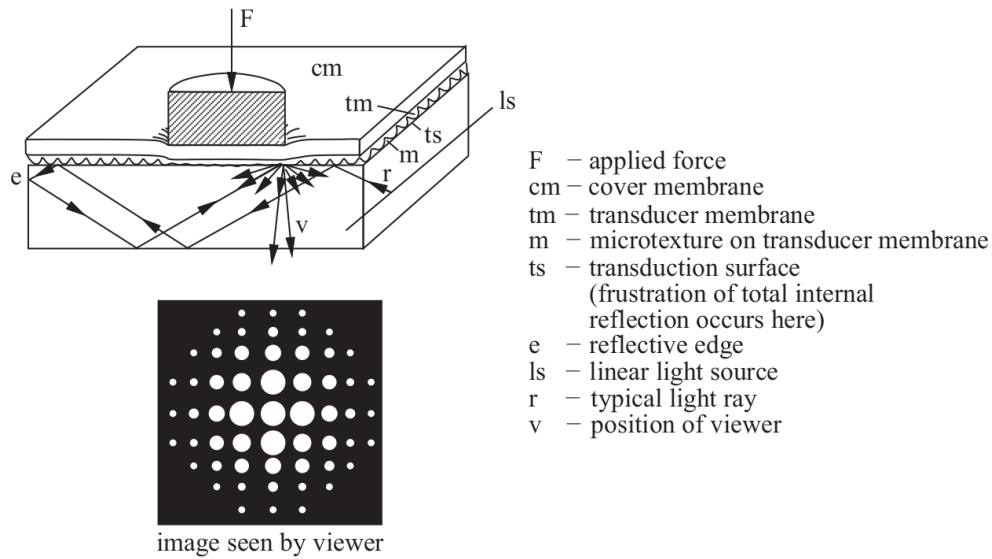


Figure 7.9

Boie's capacitive tactile sensor [19]

Light rays directed laterally in the figure remain in this light guide by the geometry of the optical system. However, the textured material on the top surface of the light guide reflects light diffusely, causing some reflected light to escape through the bottom surface. Pressure applied to the transduction membrane causes more of the textured surface to contact the transparent medium and creates an intensity image of the state of compression in the material. This tactile *image* can be conducted away from the contact site by way of optical fibers [20].

Tactile sensors can be constructed directly from optical fibers by observing that when a fiber is bent a few microns, the light intensity at the center of the fiber is attenuated. The sensitivity of the effect depends on optical parameters as well as the curvature of the bend. This kind of transduction can be used in a glove to measure the posture of a hand or fashioned into a tactile sensor. In related designs, a transmit-receive pair is separated by a polymeric light guide so that the received light intensity is proportional to the contact force.

Another optical sensor has been demonstrated in the laboratory based on the birefringence effect of photoelastic materials. This effect is used in engineering analysis to locate stress concentrations in planar parts subject to transverse loads. When subject to a mechanical stress, the strain in these materials causes a change in the polarization of transmitted light. Therefore,

the internal strain in these materials can be encoded in a varying light intensity.

Capacitive sensors—The capacitance between two parallel plates separated by a dielectric material is Ae/d , where A is the plate area, d the distance between the plates, and e the permittivity of the dielectric medium. Capacitive touch sensors are designed to change capacitance in response to applied forces. To measure the capacitance, and thus the contact load, one can use a very precise current source and measure the time-rate-of-change in the voltage, $C = Ae/d = (1/I)(dV/dt)$, or an RC circuit can be implemented using the load sensitive capacitor to transform contact load into signal frequency.

For example, [figure 7.10](#) illustrates a concept proposed by Boie [35], wherein the dielectric medium also serves as an elastic element that compresses (changing d) under load.

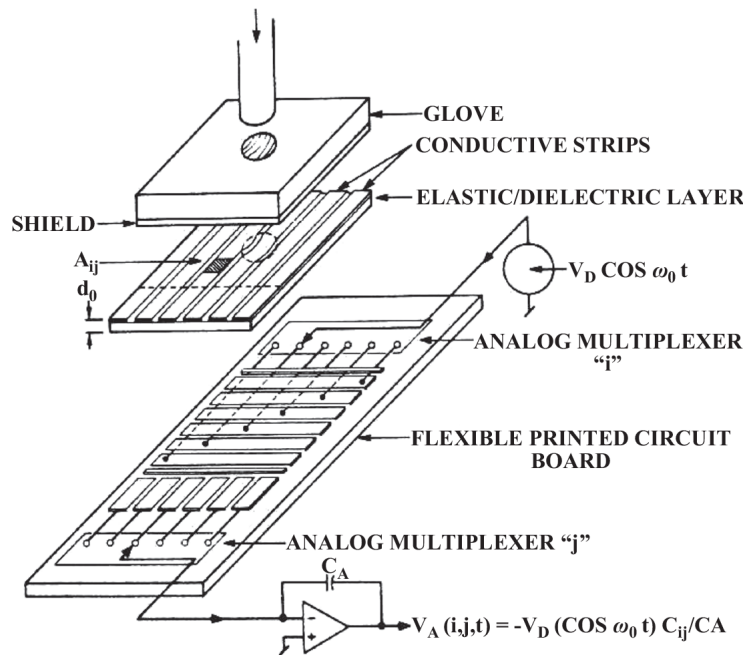


Figure 7.10
Begej's optical tactile sensor. (Source: [35] © IEEE)

The concept shows advantages over conductive elastomers in speed and immunity to noise. Investigators have shown promising results, and arrays of high spatial density appear to be possible [260]. In another concept, the

electrical and mechanical components of the sensor can be designed independently.

Magnetic sensors—The (relative) strength of a magnetic field can be measured in a number of ways. For example, if contact loads cause changes in the relative position of a pair of induction coils, the forces at the contact will show up as a change in the output voltage from a moving coil transformer. Luo et al. [180] and Vranish [283] have demonstrated touch sensors based on these principles. Luo et al. fabricated an array of 256 tactile cells with a spacing of 2.5 [mm] that demonstrated good linearity, sensitivity, and dynamic range, that can be mechanically robust, with low hysteresis and small thermal drift effects.

Magnetic properties of materials can also be used directly as in magnetoresistive and magnetoelastic materials that change magnetic properties when subjected to loads, and the physics of electromagnetic fields can be used as well in phenomena like the Hall effect.

Integrated, multimodal contact sensing—Few robotic systems have attempted to emulate the range of sensory feedback provided by the cutaneous mechanoreceptors presented in figure 7.6. One early exception was the research reported in [13, 69] and illustrated in figure 7.11, where several components of somatosensory signal content are integrated into an ambitious concept for robotic tactile sensing.

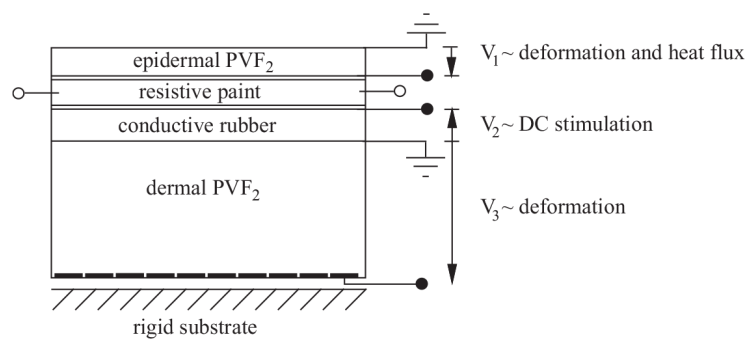


Figure 7.11

The Dario et al. artificial skin. (Source: Adapted from [69])

The figure depicts a composite structure that is intended to provide a large bandwidth response as well as to measure the relative thermal conductivity of the object. Three interdependent signals are produced by the

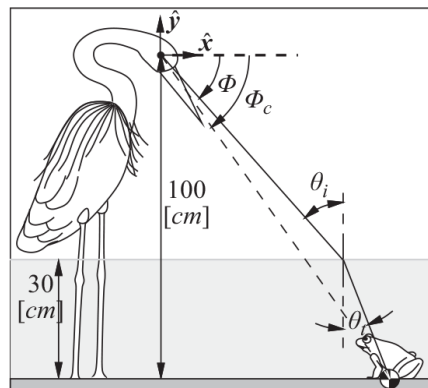
sensor—two of which derive from polyvinylidene difluoride (PVF_2), a material that possesses both piezo- and pyroelectric characteristics. Piezo- and pyro-electric materials respond directly to mechanical and thermal stresses, respectively, by producing output voltages. The dermal layer of PVF_2 in figure 7.11 is used to detect contact forces. The structure responds to frequencies up to 500 [Hz], limited by the resonant frequency of the composite structure. It does not, however, respond well to DC forces. The layer of conductive rubber on top of the dermal PVF_2 is included to add this signal component.

A layer of resistive paint in the structure is used to provide a reference temperature of 98.6 [F] to the epidermal PVF_2 layer. The signal output of this layer is due to a combination of contact deformation and the heat flux conducted through the epidermis and into the environment. The signal from the thermally insulated dermal PVF_2 helps to decompose the coupled piezo- and pyro-electric responses in the epidermal signal. The complexity of the integrated sensor together with the complex coupled signals it provides pushes the envelope in multimodal robotic tactile sensing.

7.3 Exercises

1. **Snell's law.** A great blue heron finds a frog sitting on the bottom of a shallow pond. Use Snell's law to determine the effects of refraction on the surface of a pond.

Assume that the index of refraction of air $n_{air} = 1.0$ and of water $n_{water} = 1.33$. Further, assume that the heron is able to estimate the direction of its foveal gaze $\phi = -\pi/4$ [rad] and senses that the pond is 30 [cm] deep.



(a) Use Snell's law and planar geometry to compute the value of the corrected Cartesian heading ϕ_c that incorporates refraction at the air-water interface and, thus, serves as a better reference for a motor action.

(b) Compute the error vector $\epsilon = [\Delta x \ \Delta y \ \Delta \theta]^T$ from the true Cartesian position of the frog to its apparent position due to refraction.

(c) Describe the appearance of the heron from the frog's perspective.

2. Optics: Thin lens equation. A biconvex thin lens with $R_1 = 30$ [mm] and $R_2 = -60$ [mm] is made out of an optical glass with an index of refraction $n = 1.5$.

The subject of a photograph is brought into focus on an image plane located at a distance of 70 [mm] from the center of the lens. Compute the range of this subject from the camera.

3. Compound lenses: The human eye. The cornea is a relatively fixed lens that does about two-thirds of the necessary work, after which the lens actively controls, focal length to bring distant subjects to focus on the retina. The lens in the human eye is relatively spherical in shape under no load and is stretched by muscular forces applied through the suspensory ligaments in order to control focus. In this exercise, use the thin lens equation to understand how elastic deformation of the lens influences focus.

Consider a biconvex thin lens, with positive R_1 , negative R_2 , and $|R_1| = |R_2|$. The lens material has an index of refraction $n = 1.5$. We will use this model to estimate the range of lens radii necessary for focusing over an infinite depth of field to get a sense of the deformation required in the elastic lens of the human eye.

Plot the R parameter against the range S_1 at which the eye produces a focused image 2.4 [cm]—roughly, the fixed distance between the lens and the retinal fovea in the human eye.

4. Black body radiation. Consider the radiant heat transfer between a human being and the environment. Assume that the skin of an adult human has surface area $A \approx 2$ [m²]. Furthermore, assume that absolute skin temperature $T = 306.15$ [K] in an environment with $T_0 = 293.15$ [K] and that the emissivity of skin $\epsilon = 1$.

(a) Calculate the amount of energy lost from an exposed human body under these circumstances over a 24-hour period by radiative heat loss,

and convert the result to [*kcal*] (food calories) to estimate how much caloric intake is required per day to maintain body temperature against this radiative heat loss.

(b) Do the same exercise for an ambient temperature of $T_0 = 310.15$ [*K*] (~ 100 [*F*]). How much radiant heat load on the human body must be dissipated by respiration/perspiration in a 24-hour period (in [*kcal*])?

5. Invent your own homework. Make your own exercise out of your favorite content in chapter 7. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here. Ideally, it should be based on material in this chapter (and, perhaps, previous chapters) and focused on sensors; it should call for a quantitative analysis and a short discussion; it should not take more than 30 [*min*] to solve open-book.

8

Signals, Signal Processing, and Information

Human beings and robots can use time varying signals generated by internal and external phenomena to construct richly annotated representations of the dynamic world. Proprioceptive information describes the internal state of the agent—things like actuator temperatures, battery levels, vestibular feedback, and the state of motion of the robot body. Exteroceptive signals derive from external sources—for humans, this includes visual, tactile, auditory, and olfactory stimuli in the environment. We focus on interpreting signals originating in the external environment that provide inputs to an active information processing system—the first step toward integrating perception, control, and decision-making.

In this chapter, we begin by examining the process of sampling continuous signals to see how it might distort information for humans and robots alike and analyze how signal processing operators influence the frequency content of constructed representations. A set of signal processing tools is presented that can be used to make useful information stand out and to diminish the effect of noise. The chapter concludes by introducing scale-invariant signal processing as a perceptual front end to support multimodal information gathering and domain-general problem-solving.

8.1 Sampling Continuous Signals

The first step in extracting information from the environment involves sampling data from continuous external signals. One of the greatest contributions to signal processing in the last century concerned the impact

of sampling on the information content of signals. To provide a basis for the following discussion, we begin with a short primer for the Fourier transform and its use in determining the frequency content of signals. A more complete introduction can be found in appendix A.10.2.

The Fourier transform defines a reversible mapping—a bijection—from a spatial or temporal function to a weighted sum of sinusoidal basis functions. For example, a video is a *temporally*-indexed stream of image frames, each of which captures *spatially*-indexed data on the two-dimensional image plane. The Fourier transform can describe both the temporal frequencies ($[rad/sec]$) in the video stream at each pixel and the spatial frequencies ($[rad/m]$) of intensities in the image plane.

The Fourier mapping of the one-dimensional spatial signal $g(x)$ (e.g., a row of gray-level values in an image plane) to an equivalent representation of the signal in the spatial frequency-domain $G(\omega_x)$ is defined by [equation 8.1](#).

$$\mathcal{F}[g(x)] = \mathbf{G}(\omega_x) = \int_{-\infty}^{\infty} g(x)e^{-i\omega_x x} dx, \quad (8.1)$$

where $i = \sqrt{-1}$, x $[m]$ is the spatial variable, and ω_x $[rad/m]$ is the corresponding spatial frequency variable so that $\omega_x x$ is in radians. The exponential term in the expression represents the sine and cosine series via Euler's formula $e^{-i\omega_x x} = \cos(\omega_x x) - i\sin(\omega_x x)$. One may view the Fourier transform as the projection of function $g(x)$ onto basis functions $e^{-i\omega_x x}$ for spatial frequencies $\omega_x \in [-\infty, \infty]$. The spectral coefficients $\mathbf{G}(\omega_x)$ determine how much sinusoidal basis functions account for the global shape of function $g(x)$.

The inverse Fourier transform is written

$$\mathcal{F}^{-1}[\mathbf{G}(\omega_x)] = g(x) = \int_{-\infty}^{\infty} \mathbf{G}(\omega_x)e^{i\omega_x x} d\omega_x. \quad (8.2)$$

Example: Spectral Properties of the Human Voice

Part of the distinctive character of human vocalization is captured using the influential source-filter theory. This presentation follows a more complete analysis by Titze [277], in which the input to the vocal tract is a glottal airflow shaped by the oscillation of the vocal folds much like the buzzing embouchure of the trumpeter. Vibrating vocal folds introduce acoustic energy distributed over a broad frequency spectrum through the narrow glottis. This is the auditory source. The filter is the plumbing between the input and the point at which sound is projected into the environment. In a trumpet, this is a conduit of varying diameter and length controlled by three valves. In the human vocal tract, the shape of the resonance cavity is articulated by the upper throat, mouth, tongue, and lips [277]. The filter leaves a recognizable spectral signature on all acoustic signals produced due to the resonant properties of the vocal tract—properties that are apparent in the way that energy is distributed over the frequency spectrum.

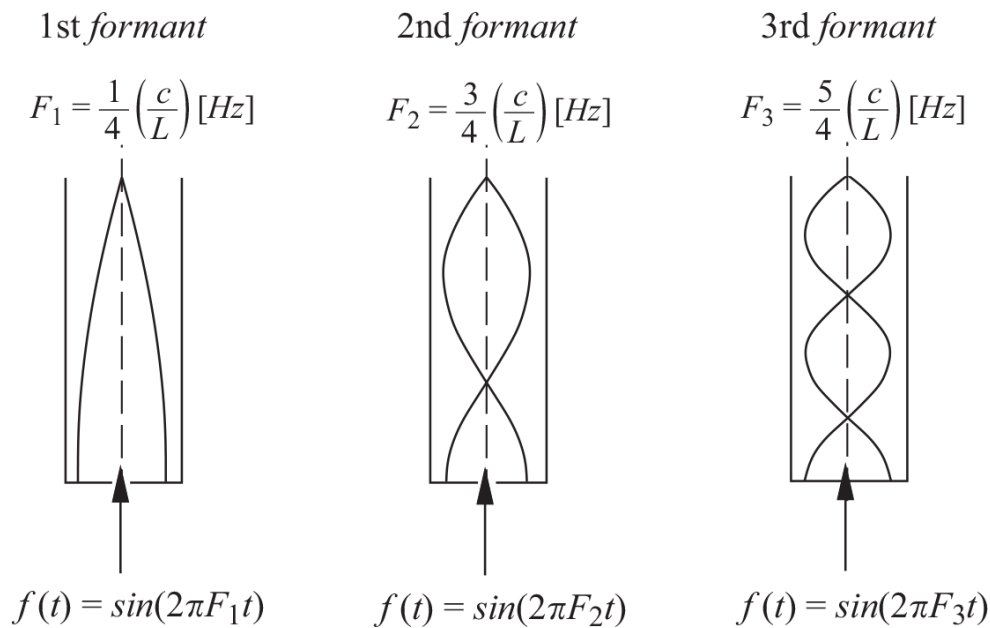


Figure 8.1
The open-closed quarter-wave resonance cavity

The acoustic signature for the human vocal tract can be predicted from the physics of sound production. For resonance, there must exist a constructive pattern of interference among pressure waves subject to multiple reflections in the vocal tract. To a first approximation, the vocal tract is modeled by the *open-closed resonance cavity*—a constant diameter

tube, open at one end, that radiates sound pressure into an infinitely larger environment (figure 8.1). The closed end of the tube reflects an incident pressure wave upward in the diagram, and the open end of the cavity reflects an *inverted* pressure wave that travels back down the tube. Resonance in the open-closed cavity implies the existence of standing waves as pictured. At resonance, the reflected pressure waves combine constructively to create a standing wave at frequencies

$$F_n = \frac{(2n - 1)}{4} \left(\frac{c}{L} \right) [Hz],$$

where c [m/sec] is the speed of sound in the gas filling the cavity and L [m] is the length of the open-closed resonator. Properly stimulated by an input signal, several resonance *modes* or *formants* will exist simultaneously, one for each F_n .¹

A standing wave is reinforced when the net phase change for a round trip (closed end to open end and back again) is an integer multiple of 2π . When $n=1$, $F_1 = c/4L$, a crest in the pressure wave at the closed end of the tube completes a quarter cycle by the time it reaches the open end. The open end reflects *and inverts* the pressure wave, and it reaches the input end of the tube a quarter cycle later in phase, once again, with the source. Thus, the open-closed resonator is also called a *quarter-wave* resonator. The resonance phenomenon amplifies tones at the resonance frequency and attenuates other frequencies and, thus, stamps the trademark sound of the trumpet, for example, on all signals broadcast into the environment.

Consider a speaker whose vocal tract—from vocal folds to the opening at the mouth—has length $L=17.5$ [cm] and use $c=35000$ [cm/s] for the speed of sound in air. Under these conditions, Equation 8.3 predicts the first four resonant frequencies (formants) to be $F_1 = 500$ [Hz], $F_2 = 1500$ [Hz], $F_3 = 2500$ [Hz], and $F_4 = 3500$ [Hz]. Figure 8.2 is the result of a computer analysis using the Fourier transform of an unstructured speech sample from a human subject voicing the phrase “the rainbow passage.”

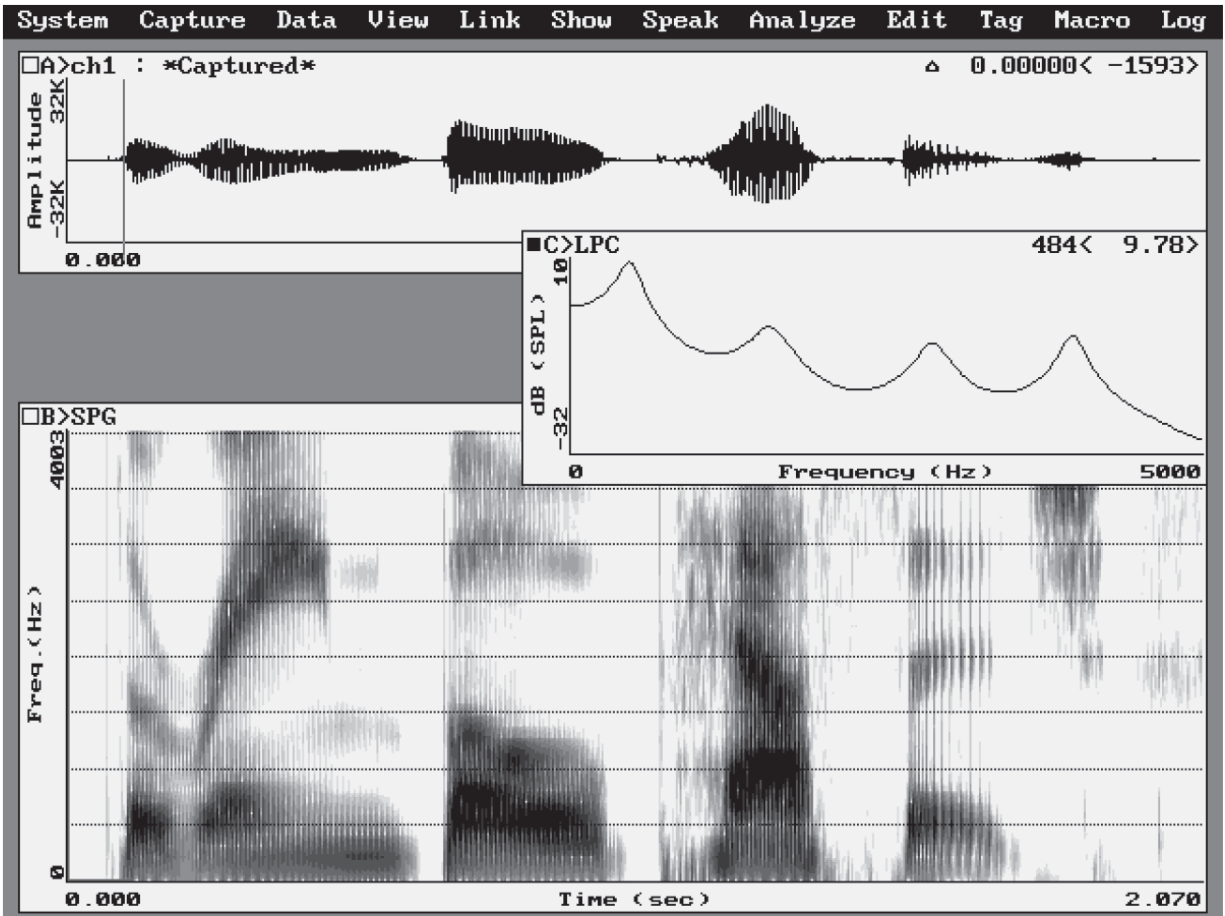


Figure 8.2

An analysis of the sound pressure level as a function of frequency using the Computerized Speech Lab. The signal is the phrase “the rainbow passage.” (Source: Courtesy Dr. M. Andrianopoulos)

The top panel of [figure 8.2](#) illustrates signal amplitude versus time recorded for “the rainbow passage.” The bottom panel shows a signal spectrogram depicting the distribution of signal energy over frequency and time.

The power in the signal at each frequency is proportional to $|F(\omega)|^2$. However, in practice, the sound pressure level (SPL) at a location is a related quantity and can be measured directly by using a microphone,

$$SPL = 20 \log_{10}(P/P_0) [dB],$$

where P_0 is a standard reference pressure of $0.00002 [Pa]$ and P is the measured sound pressure at the location of the listener. The smaller inset in [figure 8.2](#) contains a plot of the SPL observed for the speech sample as a

function of signal frequency. This speech sample contains many different sounds and, as result, the peaks in the SPL plot are roughly consistent with the first four resonant frequencies predicted using the quarter-wave model.

Figure 8.3 shows what happens when the source stimulus is a prolonged [a] vowel sound; to produce this sound, the subject opens their mouth slightly and flattens the tongue. This causes the first four formants in the inset SPL plot to be shifted significantly from the four regularly spaced peaks predicted by the open-closed resonator model (figure 8.2).

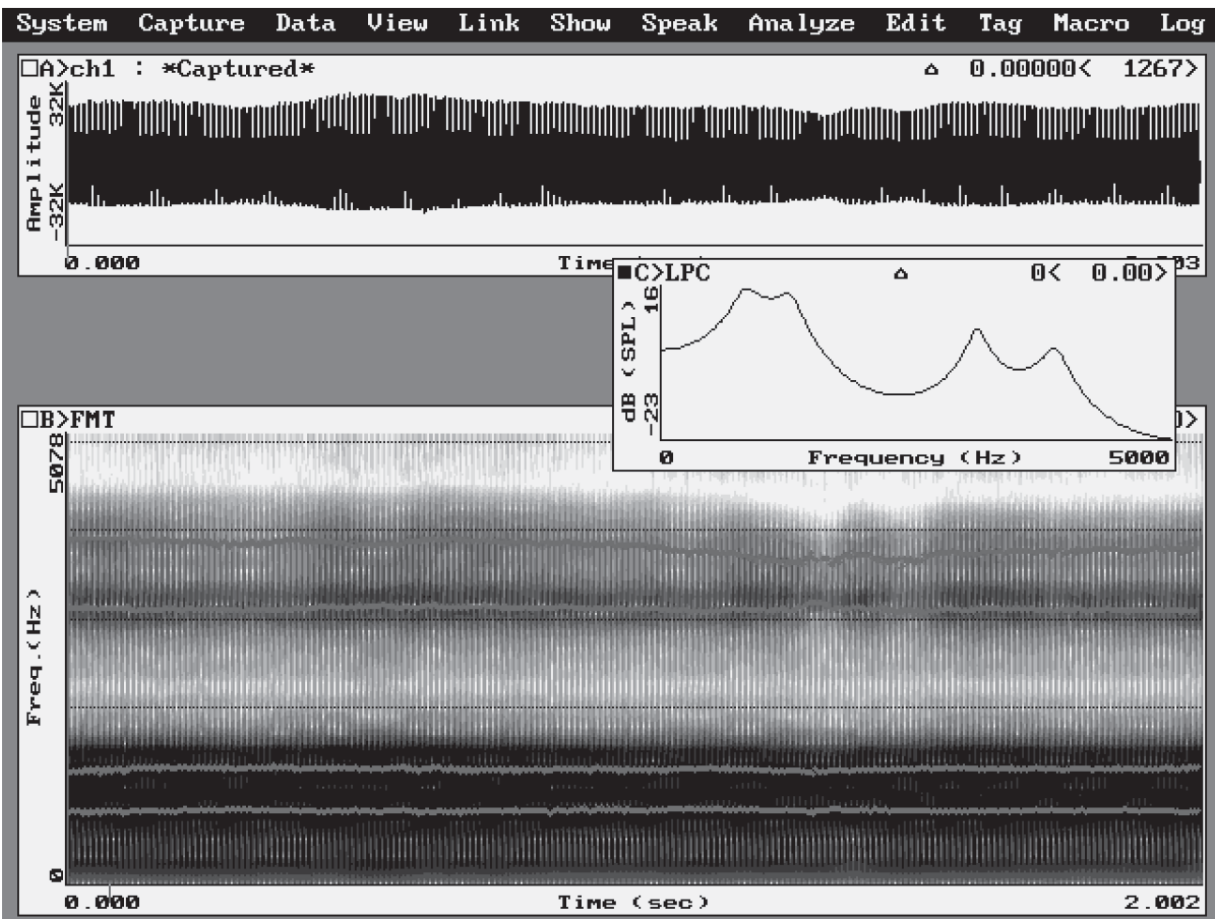


Figure 8.3

An analysis of sound pressure level as a function of frequency for a prolonged [a] vowel sound using the Computerized Speech Lab (Source: Courtesy Dr. M. Andrianopoulos)

This idea can be used to segment continuous speech—for instance, plotting the frequency of the first two formants on the F_1 - F_2 plane [277] can be used to recognize vowel sounds in unstructured speech (figure 8.4).

The discriminating power of this vocal *texture* increases as higher-order formants are considered.

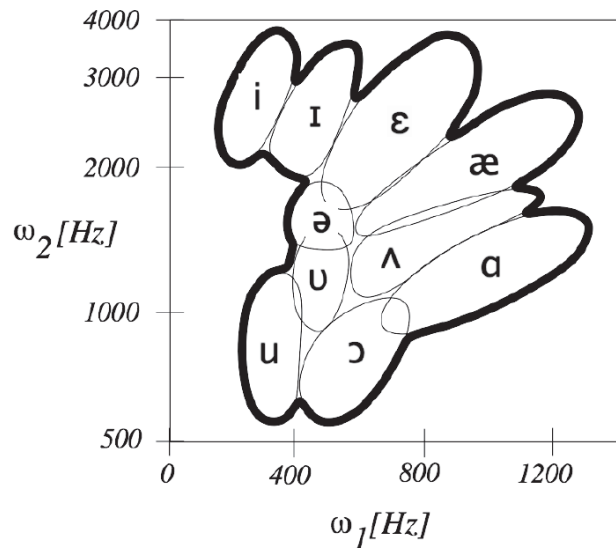


Figure 8.4

The F_1 vs. F_2 chart for 10 vowel sounds sampled from men, women, and children (Source: Adapted from [277])

□

Many other signal processing applications make use of the Fourier transform. An important instance of this class of problems concerns predicting the impact of the spatial and temporal frequency of observation on the precision of sampled data.

8.1.1 The Sampling Theorem

In both animals and robots, signal processing begins when sensor signals are sampled—a process by which the continuous signal is represented approximately in the form of discrete observations. The process of sampling has a direct impact on the fidelity of information represented in sampled feedback. Clearly, sampling discards some information. What may not be initially obvious is that sampling may also introduce information that is not faithful to the original signal.

The Dirac delta function is a mathematical abstraction of the sampling process. This operator is referred to as a singularity operator for reasons that

are obvious by its definition:

$$\delta(x - \xi) = \begin{cases} \infty & x = \xi \\ 0 & \text{otherwise.} \end{cases} \quad (8.3)$$

In addition, this function has the following two important properties:

$$\int_{-\epsilon}^{\epsilon} \delta(x) dx = 1 \text{ for } \epsilon > 0, \text{ and}$$
$$\int_{-\infty}^{\infty} g(\xi) \delta(x - \xi) d\xi = g(x).$$

The first of these properties is a form of normalization so that the sampling operator does not alter the scale of the original function, and the second is referred to as the *sifting* property because it samples the value of function $g(x)$ precisely at the location $x=\xi$. Real sensors always employ some integral over small neighborhoods in space and/or time. However, important characteristics of sampled data can be analyzed using this simple model.

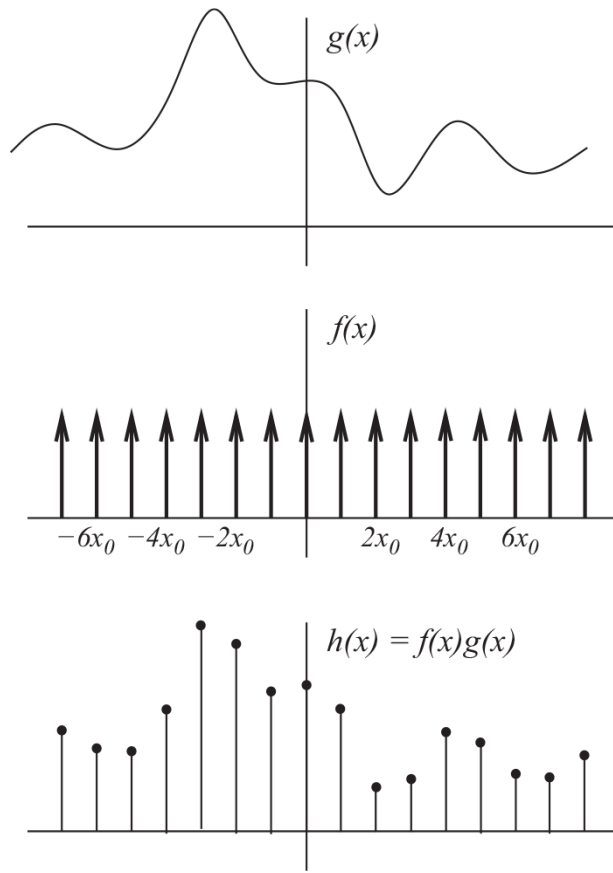


Figure 8.5

The continuous spatial function $g(x)$ is sampled by a sequence of regularly spaced Dirac delta functions $f(x)$ to create the discrete approximation $h(x)$ of signal $g(x)$.

Figure 8.5 illustrates two spatial functions, $g(x)$ and $f(x)$. The first, $g(x)$, is a continuous spatial function and the second, $f(x)$, is an infinite sequence of Dirac delta operators separated by x_0 . The product of these two functions is a sampled approximation of the original $g(x)$.

$$h(x) = g(x) \sum_n \delta(x - nx_0) = \sum_n g(nx_0) \delta(x - nx_0)$$

By the convolution theorem (appendix A.10.2), we know that the product of these two spatial functions is equivalent to the convolution of their Fourier transform pairs, so if

$$g(x) \xrightarrow{\mathcal{F}} G(\omega), \text{ and}$$

$$f(x) = \sum_n \delta(x - nx_0) \xrightarrow{\mathcal{F}} \frac{1}{x_0} \sum_n \delta(\omega - \frac{n}{x_0}) = F(\omega), \text{ then}$$

$$H(\omega) = F(\omega) * G(\omega).$$

By the definition of continuous convolution ([equation A.18](#)),

$$\begin{aligned}
H(\omega) &= \int_{-\infty}^{\infty} G(\alpha)F(\omega - \alpha)d\alpha \\
&= \int_{-\infty}^{\infty} G(\alpha) \left[\frac{1}{x_0} \sum_n \delta(\omega - \frac{2\pi n}{x_0} - \omega) d\alpha \right] \\
&= \frac{1}{x_0} \int_{-\infty}^{\infty} \sum_n G(\omega - \frac{2\pi n}{x_0}) \delta(\omega - \alpha - \frac{2\pi n}{x_0}) d\alpha \\
&= \frac{1}{x_0} \sum_n G(\omega - \frac{2\pi n}{x_0}) \int_{-\infty}^{\infty} \delta(\omega - \alpha - \frac{2\pi n}{x_0}) d\alpha
\end{aligned} \tag{8.4}$$

so that, by the properties of the Dirac delta sampling operator,

$$H(\omega) = \frac{1}{x_0} \sum_n G(\omega - \frac{2\pi n}{x_0}). \tag{8.5}$$

[Equation 8.5](#) states that the frequency spectrum of the sampled image consists of duplicates of the spectrum of the original signal distributed at $2\pi/x_0$ [rad/m] frequency intervals ([figure 8.6](#))—this quantity is inversely proportional to the spatial distance between samples ([figure 8.5](#)) and is known as the *sampling frequency*. Spectral energy in $G(\omega)$ at frequencies greater than half of the sampling frequency (or π/x_0 [rad/m]) will overlap in the sampled approximation $H(\omega)$. The resulting signal distortion is called *aliasing*—a phenomenon where high frequency energy in $g(x)$ is added to the reconstructed image in frequency bands where it does not belong.

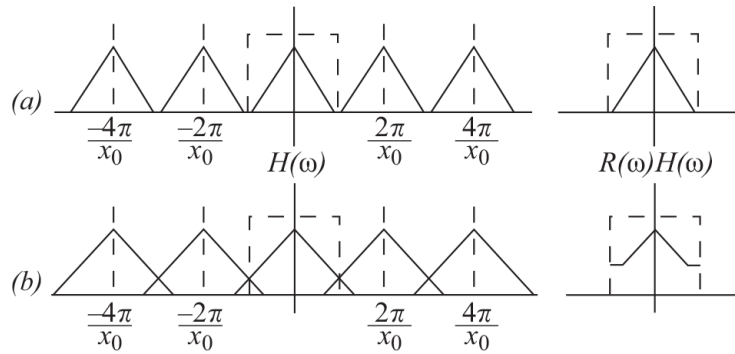


Figure 8.6
The effects of sampling on reconstruction

Figure 8.6 shows how this happens. Suppose we use a perfect rectangular band-pass filter $R(\omega)$ to extract a single copy of the repeated spectrum in $H(\omega)$, where

$$R(\omega) = 1 \text{ if } |\omega| < \pi/x_0$$

$$0 \text{ otherwise.}$$

The figure shows the repeated spectra due to sampling and the clipped-out copy, $R(\omega)H(\omega)$. For the band limited signal (figure 8.6(a)), this yields an approximation of the original $G(\omega)$ so that the inverse Fourier transform, $\mathcal{F}^{-1}[R(\omega)H(\omega)]$, is a faithful reproduction of the original spatial signal $g(x)$.

Figure 8.6(b) shows the qualitative effect of an inadequate sampling frequency. In this case, the sampling frequency is too small to separate a single copy of the repeated spectra in $H(\omega)$ and $R(\omega)H(\omega)$ is corrupted by information in adjacent spectra. Therefore, some of the information in $R(\omega)H(\omega)$ is an artifact of the sampling process itself. The sampling theorem follows directly from these observations.

Definition 8.1: Nyquist sampling theorem. *A band limited signal, $g(x)$, can be faithfully reconstructed from discrete samples at regular intervals, $x_0[m]$, if the sampling frequency, $2\pi/x_0$ [rad/m], is at least twice the maximum frequency in $g(x)$.*

The theorem is generally attributed to early work by Harry Nyquist (1928); however, other names are also associated with the result. It was first stated in the form presented here by Claude Shannon in 1949. As a consequence of the sampling theorem, we can now consider how the

performance of signal processing depends on the *bandwidth* of the signal and design filters for separating the informative signal from the noise. Integrated solutions that consider all of these issues relative to the requirements of a control task can avoid introducing spurious information in the sampled feedback.

8.2 Discrete Convolution Operators

Useful information in signals can be extracted from properties concerning the local *shape* of the signal as a function of independent spatial and temporal variables. Visual pathways in the human nervous system include special circuits for identifying oriented edges, motion, texture, and other features that depend only on local properties of the intensity function. The mathematical framework for extracting local shape information in signals is convolution.

In the following, the function $g(\cdot)$ represents an adequately sampled (faithful) approximation of a continuous environmental signal in space and time. This could be a sequence of image frames, sound pressure levels, or a history of tactile force observations. Without loss of generality, we will introduce techniques for processing these signals in the context of low-dimensional spatial signals.

For two-dimensional continuous images, the convolution operation (appendix A.10.2) is written,

$$\begin{aligned} h(x, y) &= f(x, y) * g(x, y) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) g(x - u, y - v) du dv, \end{aligned}$$

where $*$ denotes the convolution operator. The equivalent operation for discrete operators and signals is

$$h(x, y) = \sum_{i=-\alpha}^{+\alpha} \sum_{j=-\alpha}^{+\alpha} f(i + \alpha, j + \alpha) g(x + i, y + j), \quad (8.6)$$

where we assume that the number of rows and columns in operator $f(x, y)$ are equal and odd so that its center coincides with $g(x, y)$.

Figure 8.7 illustrates how the convolution process aligns operator f over a subset of image g near (x, y) , sums the pixel-wise products of f and g for this subset, and stores the result in output position $h(x, y)$. Conceptually, this computation happens in parallel over all (x, y) where complete f -shaped neighborhoods are defined in g , leaving a region of unprocessed image data around the image border.

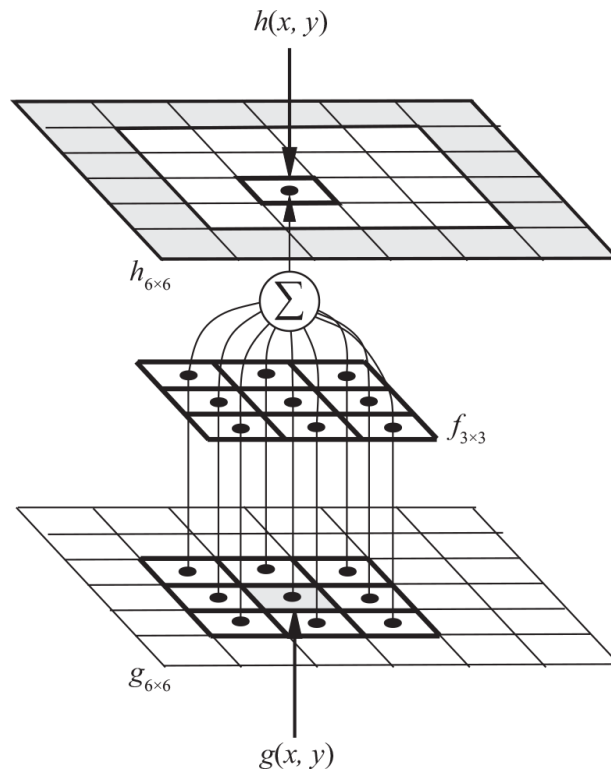


Figure 8.7
Local computation of image features

Specifying parameter α in equation 8.6 corresponds to specifying a $(2\alpha + 1) \times (2\alpha + 1)$ kernel for operator f . For example, when $\alpha = 1$, operator $f(x, y)$ is a 3×3 matrix, so that $h(x, y)$ depends on the weighted sum of only 9 pixels in the neighborhood of $g(x, y)$. Larger kernels gather support from larger neighborhoods of g . As a consequence, the *kernel size* of f has a direct influence on how sensitive the response $h = f * g$ is to spatial information embedded in image g .

8.2.1 Spectral Filtering

Informally, function $h=f * g$ represents the degree to which the shape of function f is correlated with the local shape of g . More formally, the Fourier transform allows us to view convolution operator f as a spectral filter.

Consider a one-dimensional signal $g(x)$ and operator $f(x)$. The convolution theorem (appendix A.10.2) states that convolution in the spatial domain is equivalent to multiplication in the frequency domain,² that is

$$\mathcal{F}[f(x) * g(x)] = F(\omega_x)G(\omega_x) .$$

Therefore, in the frequency domain, $F(\omega_x)$ can be viewed as a multiplicative weighted *mask* applied to $G(\omega_x)$. This perspective provides some insight into the frequency dependent influence of operator $f(x)$ on signal $g(x)$.

Low-Pass Filtering—If the one-dimensional rectangular convolution filter $f(x)$ in figure 8.8 is used to filter a signal $g(x)$ defined on the interval $x \in [0, 7]$. The convolution computes $h(x) = f(x) * g(x) = g(x-1) + g(x) + g(x+1)$ for values of $x \in [1, 6]$. The figure shows that $h(x)$ shares the general shape of $g(x)$, but suppresses rapid variations in the signal; it is a *smoothed* version of $g(x)$.

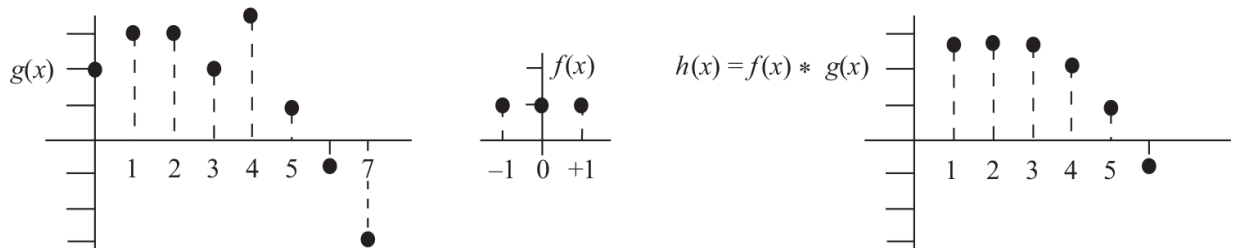


Figure 8.8

The performance of a simple low-pass filter

To see why this happens, we consider a continuous approximation of $f(x)$ using the rectangular function $rect(x)$ illustrated on the left side of figure 8.9. As a convolution operator, this function computes a response proportional to the local mean of a signal over the domain $x \in [-1/2, +1/2]$. The Fourier transform of $rect(x)$ is the $sinc(\omega/2\pi)$ function (see table A.1). Figure 8.9 shows that the spectral coefficients $F(\omega_x)$ for low frequencies are near unity, but they rapidly fall off (and can be inverted) for higher frequencies. This convolution operator will therefore attenuate high

frequencies while allowing lower frequencies to pass relatively unscathed. Because of its spectral sensitivity, operator $f(x)$ is an example of a *low-pass filter*. High frequency content in the signal will be eliminated by such operators—sharp edges in the original image will be smoothed in the filtered image—emphasizing information in lower frequency bands.

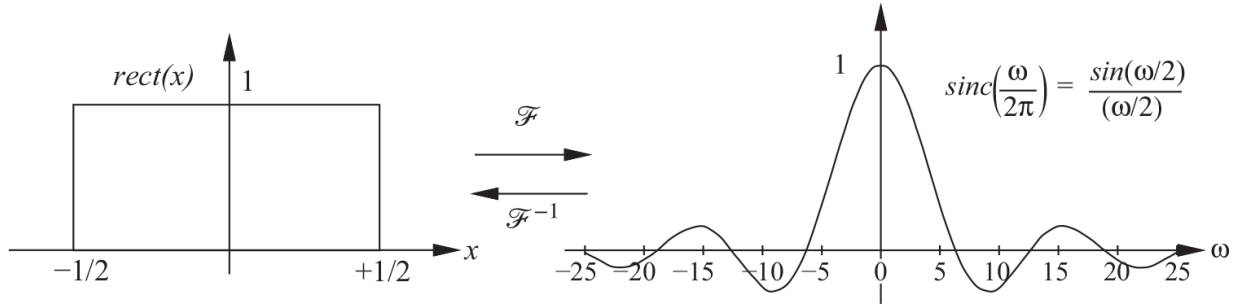


Figure 8.9

The Fourier transform of the $\text{rect}(x)$ function shows why it behaves as a low-pass filter

High-Pass Filtering—Rapid changes in environmental surface orientation or reflectivity, or spatial discontinuities between foreground objects and the background can cause *edges* in feedback signals. The steepness of image function $g(x, y)$ can be measured directly by computing its gradient. For example, the gradient at position (x, y) in a two-dimensional image function is $\nabla g(x, y) = (\partial g / \partial x)\hat{x} + (\partial g / \partial y)\hat{y}$ can be approximated using finite differences, where

$$\frac{\partial g(x, y)}{\partial x} \approx \frac{g(x+1, y) - g(x-1, y)}{2}, \quad \text{and} \quad \frac{\partial g(x, y)}{\partial y} \approx \frac{g(x, y+1) - g(x, y-1)}{2}.$$

If $\nabla g = [h_x \ h_y]^T$, where $h_x = f_x * g$ and $h_y = f_y * g$, the finite difference expressions above suggest one choice for the directional gradient operators

$$f_x = \left[-\frac{1}{2} \quad 0 \quad \frac{1}{2} \right], \quad \text{and} \quad f_y = \begin{bmatrix} \frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{bmatrix}.$$

The magnitude and orientation of ∇g everywhere in the two-dimensional signal can also be computed

$$|\nabla g(x, y)| = [h_x(x, y)^2 + h_y(x, y)^2]^{1/2} \tag{8.7}$$

$$\phi(x,y) = \tan^{-1} \left(\frac{h_y(x,y)}{h_x(x,y)} \right) \quad (8.8)$$

Places where the gradient magnitude (equation 8.7) is greater than some appropriate threshold are strongly correlated with edges in the signal. Several examples of convolution operators for detecting edges based on finite difference gradient approximation can be found in the literature. Some of these are presented in table 8.1. These operators constitute *high-pass* filters that emphasize high-frequency content in signals at the expense of low-frequency information.

Table 8.1
Gradient (First Derivative) Operators

Operator	f_1	f_2
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewit	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

8.2.2 Frei and Chen Signal Decomposition Operators

In 1977, Frei and Chen [88, 12] considered a set of nine independent convolution operators defined using a 3×3 kernel.

$$\begin{array}{cccc}
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \\
f_0 & f_1 & f_3 & f_5 & f_7 \\
\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \\
f_2 & f_4 & f_6 & f_8
\end{array} \tag{8.9}$$

The Frobenius inner product³ of any pair of these operators is zero, verifying that the Frei-Chen operators form an independent basis for signals in 3×3 neighborhoods of the image plane.

At each (x, y) location in the image, a nine-dimensional vector of responses images, $\mathbf{h}(x, y)$, whose elements are defined $h_k = f_k * g$, $k \in [0, 8]$, represents the local shape of the signal in the neighborhood around $g(x, y)$.

The *energy* of the response is proportional to the magnitude of $\mathbf{h}(x, y)$. The response energy for the edge-like operators f_1 and f_2 is $(h^2(x, y) + h^2(x, y))^{1/2}$. To decide whether pixel (x, y) is likely to be an edge, the Frei-Chen edge detector [88] compares the edge response energy to the total response energy over all nine Frei-Chen basis operators.

Figure 8.10 plots two hypothetical response vectors, $\mathbf{h}(x_1, y_1)$ and $\mathbf{h}(x_2, y_2)$. Response $\mathbf{h}(x_1, y_1)$ has significantly more total energy than $\mathbf{h}(x_2, y_2)$ and more edge-like energy as well. However, the distribution of energy in $\mathbf{h}(x_2, y_2)$ is heavily concentrated in the h_1 and h_2 elements of the response vector—the energy in response $\mathbf{h}(x_1, y_1)$ is not. If we label edges solely on the strength of the response to operators f_1 and f_2 (as in the left panel of figure 8.10), we erroneously conclude that $\mathbf{h}(x_1, y_1)$ is a better example of an edge than $\mathbf{h}(x_2, y_2)$.

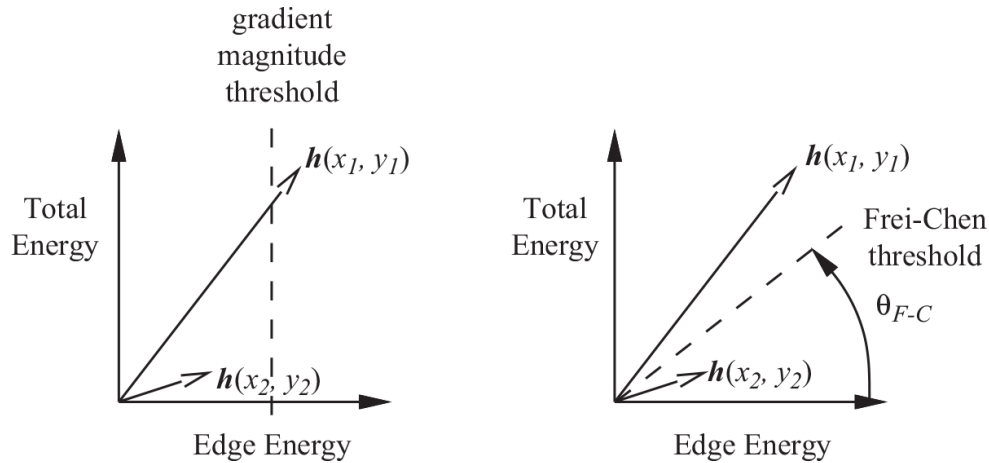


Figure 8.10

The distinguishing appearance of an edge can depend on the overall shape of the intensity surface rather than the magnitude of its first derivative alone

Instead, Frei and Chen proposed an angular threshold θ_{F-C} (right panel of figure 8.10) so that if

$$\tan^{-1} \left(\frac{\text{total energy}}{\text{edge energy}} \right) < \theta_{F-C}$$

then the information in $h(x, y)$ supports labeling the pixel located at (x, y) as an edge.

Appearance-based measurements such as this are more robust than those based solely on gradient magnitude. They demonstrate how *patterns of response* to multiple signal processing filters can be more informative than single operators. However, the Frei-Chen edge detector is limited to the expressive power of the 3×3 Frei-Chen kernel space. In section 8.3, these ideas are extended to include a more comprehensive family of signal operators.

8.2.3 Noise, Differentiation, and Differential Geometry

As a practical concern, all signals are influenced to some degree by noise—random processes that add spurious, generally high frequency disturbances to signals. The output of differential operators, like edge detectors, is particularly sensitive to noise. For this reason, most edge operators combine low-pass characteristics to suppress noise with high-pass filtering to detect derivatives. For example, the Prewit f_1 operator in table 8.1 combines finite

difference gradient computations in the horizontal direction with low-pass filtering (using the $rect(x)$ function) in the vertical direction. Useful information, however, is *band limited*, that is, confined to lower spatial or temporal frequency bands below a cutoff ω_{cutoff} . Parts of the signal above ω_{cutoff} may be subject to aliasing or noise.

Consider a differentiable one-dimensional low-pass filter $f(x)$ designed to suppress signal frequencies above an appropriate cutoff in a one-dimensional signal $g(x)$. It is relatively simple to show that

$$\frac{d(f * g)}{dx} = \frac{df}{dx} * g = f_x * g \quad (8.10)$$

(see [equation A.18](#)). Therefore, rather than smoothing signal g with the low-pass filter and then performing another convolution to differentiate the result, precisely the same result can be obtained if signal g is convolved once with f_x . As a consequence, the differential structure of a band-limited, one-dimensional signal can be represented as a vector of signal derivatives estimated using finite difference operators

$$[g \ g_x \ g_{xx} \ \dots] \approx [g \ (f_x * g) \ (f_{xx} * g) \ \dots]$$

to arbitrary order N called the N -jet. The result is a family of independent responses that capture the differential geometry (or shape) of the signal. This idea can be extended to address signals in higher dimensions. For example, in computer vision, the N -jet of order 2 for a two-dimensional image function $g(x, y)$ is the vector

$$[g \ g_x \ g_y \ g_{xx} \ g_{xy} \ g_{yy}] .$$

Differential structure has significant implications in the construction of domain-general representations of signal structure, as we shall see in [section 8.3](#).

Example: Edge Sharpening

There are many circumstances in which sharp geometric features in the world can appear as gradual intensity variations on an image plane. Optical aberrations in the lenses, thermal effects, and diffusive particles (fog) in the

air may defocus edges. Gradient information alone can be used to detect edges, but, in some cases, at the expense of precision. One way to improve precision over a wide range of edge strength is to use second derivative information to estimate the precise *center* of edges.

The top plot in [figure 8.11](#) represents the image of a diffuse edge in the discrete signal $g(x)$. Suppose that classifying edges in $g(x)$ depends exclusively on the gradient $g_x(x) \approx f_x * g(x)$ and, in particular, on the case when the gradient magnitude exceeds a threshold $|g_x| > \tau$. The middle plot in the figure shows the result of convolving $g(x)$ with the finite-difference gradient operator $f_x = [-1/2 \ 0 \ 1/2]$. Choosing a relatively small value for τ has the virtue of detecting weak as well of strong edges; however, strong edges will bloom over several pixels. In the middle plot, a relatively low threshold creates seven adjacent edge labels.

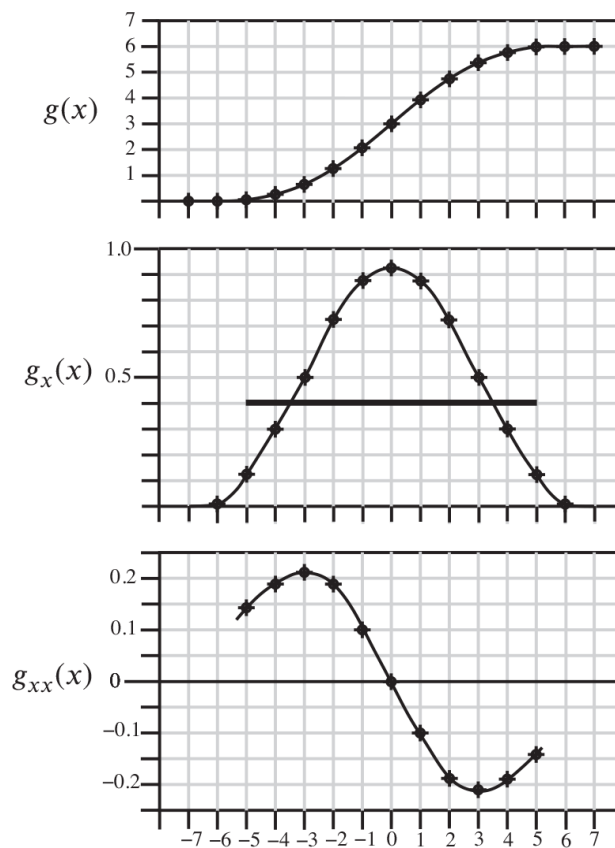


Figure 8.11

Second-order N -jets can be used to sharpen edges in signals by identifying the inflection point in the intensity function.

We can localize strong edges precisely without sacrificing the ability to detect weak edges by estimating the 2-jet for the one-dimensional signal $g(x)$ [g g_x g_{xx}] and looking for gradients that exceed a moderate threshold in places where the second derivative crosses zero,

$$|g_x| > \tau, \quad \text{and}$$

$$g_{xx} \approx f_x * g_x = 0.$$

The bottom panel in the figure shows the second derivative of function $g(x)$. The zero crossings in $g_{xx}(x)$ provide a real-valued estimate of the apparent center of the edge to sub-pixel accuracy.

Using both first- and second-order derivative information, one can precisely localize weak and strong edges alike. Often, this technique is used in two-dimensional computer vision applications using the a 3×3 Laplacian operator to approximate the sum of the diagonal elements of the image Hessian. In this case,

$$\nabla^2 g(x, y) = \frac{d^2 g}{dx^2} + \frac{d^2 g}{dy^2} \approx L_{xy} * g(x, y)$$

where the Laplacian operator is defined

$$L_{xy} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (8.11)$$

Exercise 8.4.3b asks the reader to derive this operator using finite differences on the image function.

□

The next section explores a parametric family of signal processing operators that support noise reduction and reveal differential structure over a large range of spatial and temporal scales.

8.3 Structure and Causality in Signals

Cartographers create maps (continents, countries, cities, towns, streets, buildings) for each scale of a navigation task. Reasoning at small scales focuses on fine details and requires a narrow receptive field, whereas other tasks require large scales, involve coarse details, and require large receptive fields—perhaps extended over time. In general, the spectrum of environmental scale (geometrical and inertial) to which an agent attends is determined by the scale (kinematic and dynamic) of the embodied⁴ agent. Scale remains important throughout the full spectrum of abstraction employed by human decision makers. Linguists, for example, describe a *basic level of discourse* with which to efficiently communicate categorical prototypes over scale—trees and forests. Most often, basic level categories in speech whose members are easily recognizable, share physical and functional attributes, and elicit a consistent response [71]. Basic level categories are maximally distinctive between classes and maximally informative within classes. They enhance the capacity for inference and provide prototypical associations that support new problem domains [302].

These observations support a hierarchical cognitive framework where scale is a factor in determining appropriate levels of granularity and abstraction. Computer vision researchers have contributed important concepts regarding the influence of scale on information in visual imagery. The shape of the sampled luminosity function shares common structure when viewed at long and short range although the scale of this structure is quite different—nearby objects looming in the foreground are still generally recognizable for some time as they recede toward the horizon. Up to limits on sample resolution, features that distinguish objects in the projected luminosity function are preserved across scales.

In 1983, Witkin [294] introduced the term *scale space* and developed a systematic way to relate signal representations and structures at different scales. Koenderink [151] proposed adding a scale parameter to the image function $g(x, y, \sigma)$ and used this representation to detect differential structure in the multiscale image function. An important consequence of expanding the description of image functions this way is that topological events associated with critical points in the luminosity function (extrema, ridge points, corners, blobs) can be computed robustly and form a convenient set of primitives for a *structural information theory*. Koenderink also proposed combining Gaussian derivative operators at different scales

into more explicit descriptors of image geometry [151]. Lindeberg and ter Haar Romeny [177] describe scale as a fundamental parameter of sampled data and argue that multi-scale representations are necessary for an agent to understand and control interactions across the full spectrum of viewpoints afforded by the environment.

In the following, we summarize the major findings of this research including an introduction to the Gaussian kernel, a means of determining the local *intrinsic scale* and differential structure of generic signals. We will review the underlying theory and implementation of this framework using the results and images reported in [175].

8.3.1 Gaussian Operators

Gaussian convolution operators are discrete approximations of the continuous Gaussian function

$$f_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}, \quad (8.12)$$

where σ is the standard deviation of the Gaussian kernel. Increasing σ dilates the function, so that the filter response draws support from a larger neighborhood in the signal. The left side of figure 8.12 (a) shows the continuous Gaussian operator at three scales. Smaller σ values in $f_{\sigma}(x)$ correspond to higher bandwidth (broader) spectra $F_{\sigma}(\omega)$ on the right side of the figure. Therefore, the Gaussian convolution operator behaves as a low-pass filter—a smoothing filter that suppresses noise—with a cutoff frequency that is inversely proportional to σ .

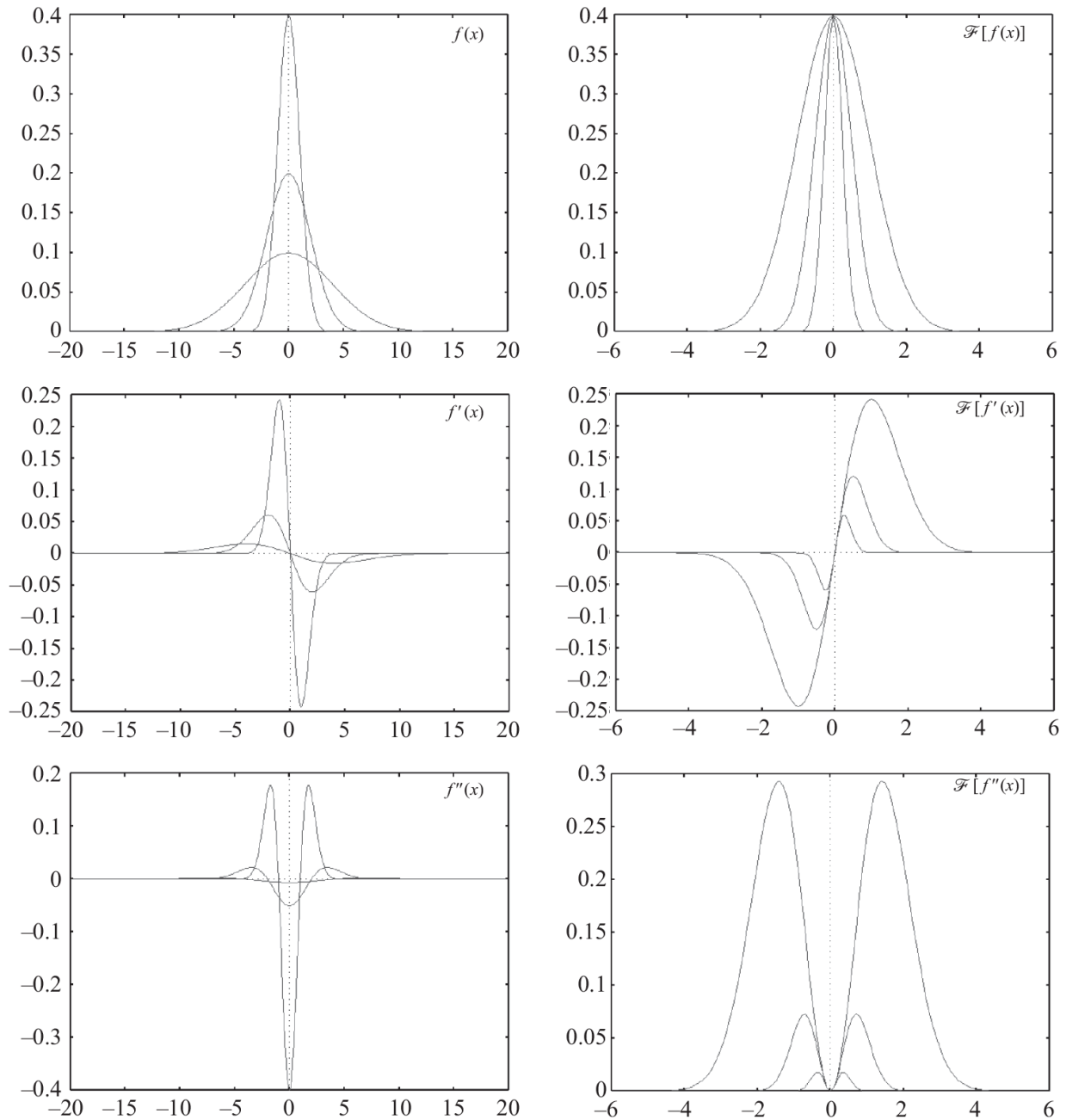


Figure 8.12

Fourier transform pairs for one-dimensional, derivative of Gaussian operators at three scales $\sigma=1, 2, 4$: (a) the Gaussian low-pass operator, (b) noise reduction and differentiation in the gradient operator, and (c) the Laplacian (second-derivative) of Gaussian (LoG) operator in one-dimension

The infinitely differentiable Gaussian operator $f_\sigma(x)$ supports smoothing and noise suppression and provides a convenient means of computing the N -jet to arbitrary order. For example, the first- and second-derivative

Gaussian operators that combine differentiation and smoothing at the scale specified by σ are written

$$f'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-x^2/2\sigma^2}, \text{ and} \quad (8.13)$$

$$f''_\sigma(x) = \frac{1}{\sqrt{2\pi}} \left[\frac{x^2}{\sigma^5} - \frac{1}{\sigma^3} \right] e^{-x^2/2\sigma^2}. \quad (8.14)$$

Figure 8.12 rows (b) and (c) show the first and second derivatives of the Gaussian operator at the same three scales illustrated in row (a). These functions are generalizations of the finite difference operators we saw in section 8.2.1 over the scale parameter. The first derivative of the Gaussian in row (b) of the figure is symmetric, odd, and behaves like an edge detector; the second derivative in row (c) behaves like a one-dimensional version of the Laplacian operator introduced in equation 8.11. However, implementing the Laplacian using Gaussian scale space operators creates band-pass characteristics that are dependent on σ .

Deep structure in signals consists of information that is persistent across scale. To estimate the differential geometry of a signal, a family of convolution operators must *correctly* suppress fine-grained detail without introducing spurious information. Done correctly, the result is a family of derived signals in the N -jet that describe the topological structure of the signal across scales. Koenderink captures this property in his *causality* requirement [151]. Causality requires that small scale structure is supported by structure at larger scales; for example, strong responses to large scale edge operators supports expectations of elevated responses to smaller scale edge operators that are co-linear with the larger feature in the signal $g(x, y, \sigma)$. Lindeberg and ter Haar Romeny use these arguments to assert that the Gaussian kernel is unique among all candidates as a means of generating scale space [177].

8.3.2 The Gaussian Pyramid: Blobs

Spatially isotropic Gaussian derivative filters treat spatial and scale variables uniformly—a property referred to as *homogeneity*. They support feature detection in visual signals that are robust to translations and

rotations of the sensor, viewing distance (scale), and intensity transformations. In general terms, this family of signal operators can be used to detect important *keypoints* in signals.

Equation 8.10 (re-written slightly here),

$$\frac{d^n g_\sigma}{dx^n} \approx \frac{d^n f_\sigma}{dx^n} * g = \frac{d^n}{dx^n} (f_\sigma * g) ,$$

states that signals can be smoothed and differentiated by applying a fixed-scale differentiation operator (d^n/dx^n) to a *blurred* version of the original signal ($f_\sigma * g$)—effectively down sampling g in a manner that depends on scale σ . When f_σ denotes the scale dependent Gaussian kernel, this idea leads to a useful conceptualization of scale-space in terms of the *Gaussian pyramid*.



Figure 8.13

An image of a field of sunflowers (reprinted by permission from [175])

Consider the image of a field of sunflowers (**figure 8.13**), for which extrema in the second derivative (the curvature) of $g(x, y, \sigma)$ are an important source of signal structure. In two dimensions, the Gaussian low-pass filter is written

$$f(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \quad (8.15)$$

Operator $f(x, y, \sigma)$ dilutes the overall contrast in the image by a factor proportional to $1/\sigma^2$. Therefore, to compare responses across scales, each response must be normalized by multiplying it by a factor proportional to σ^2 .

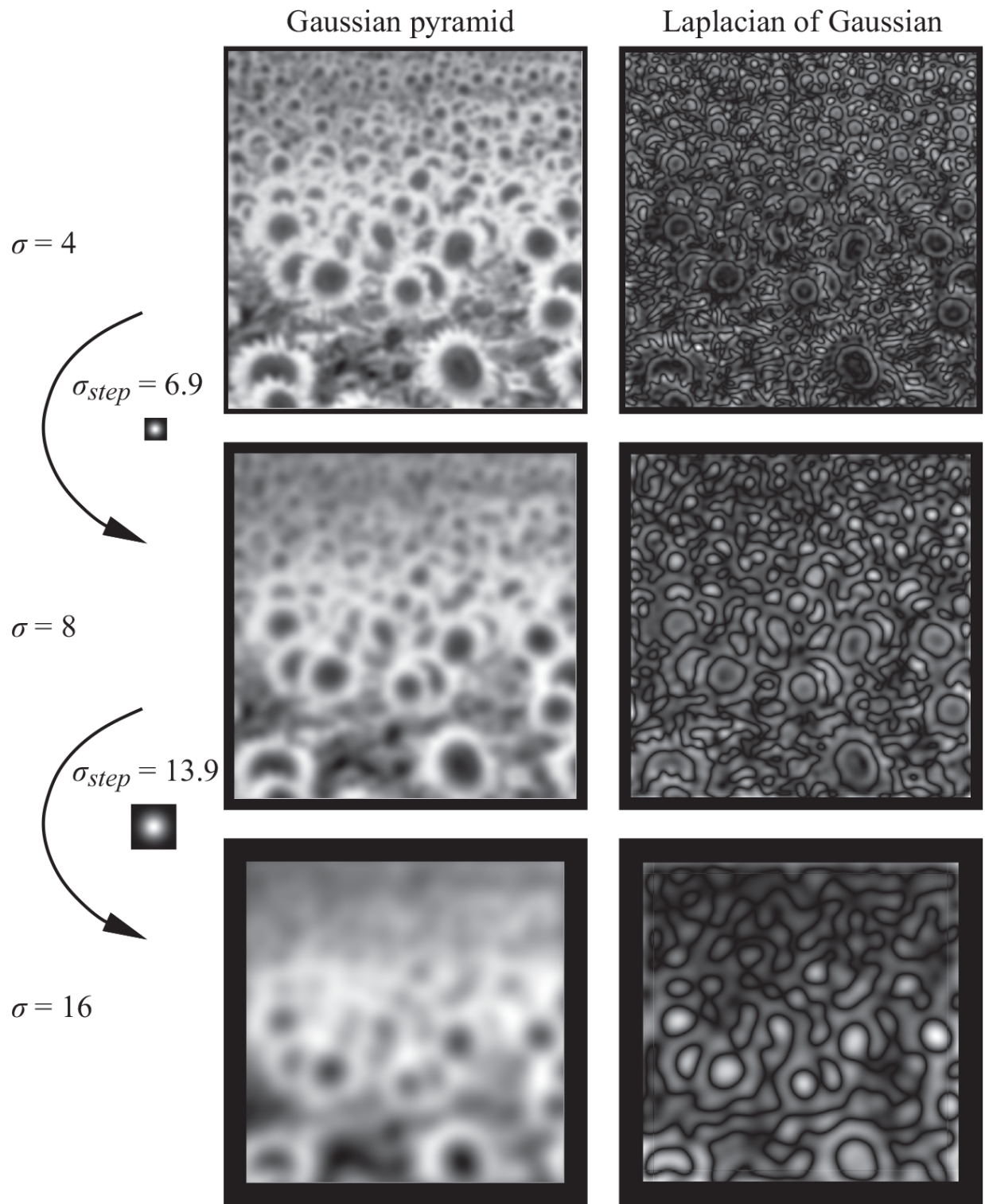


Figure 8.14

A part of the Gaussian Pyramid for the image of a field of sunflowers

An N -level pyramid in scale-space is configured using a sequence of exponential values $\sigma = 2^t$ in the Gaussian blurring operator (equation 8.15), for $t=0, N-1$. Figure 8.14 shows three levels in the scale-normalized Laplacian of Gaussian (LoG) pyramid computed

$$\nabla^2 g(x,y) = \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \approx L_{xy} * \left[\sigma^2 f(x,y, \sigma) * g(x,y) \right]$$

for the field of sunflowers, where L_{xy} is the Laplacian operator (equation 8.11).

The scale that maximizes the normalized LoG response for each pixel is called the *intrinsic scale* σ_{int} of that pixel. It describes the shape of the signal in a circular neighborhood around $g(x, y)$ with radius $r = \sqrt{2}\sigma_{int}$. The scale-normalized LoG responses over the entire Gaussian pyramid are sorted by magnitude and rendered in descending order. The result reported in [175] for the field of sunflowers is rendered, next to the original image, in figure 8.15. It captures a form of structure common to all sunflowers in the image from foreground to background.

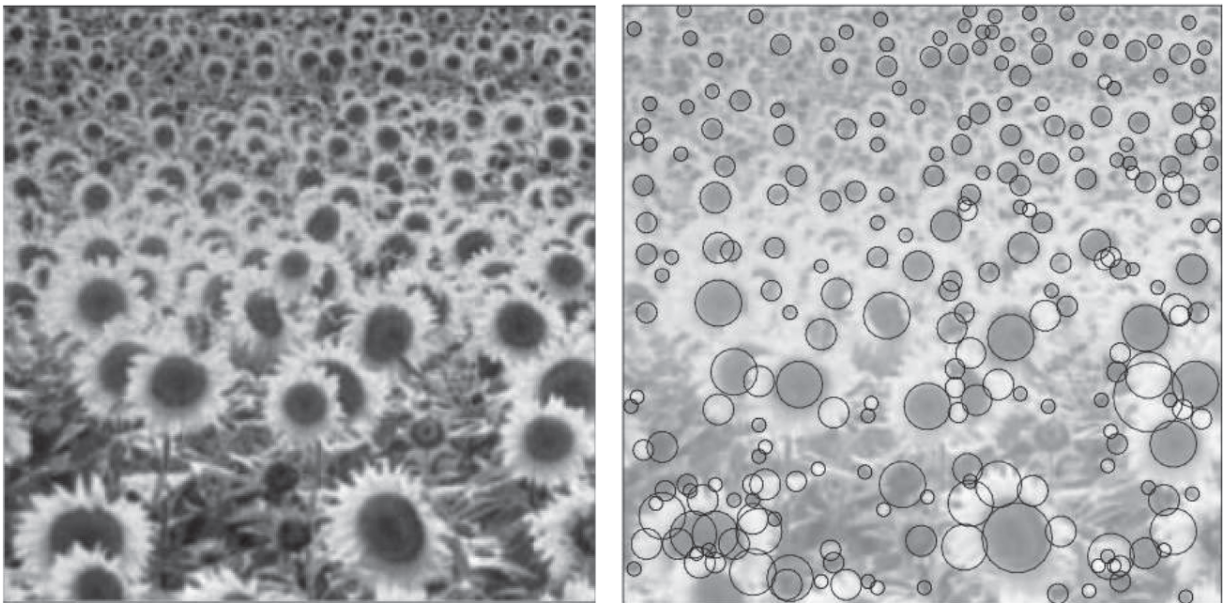


Figure 8.15

The top 200 scale-space blob responses in the scale-normalized Laplacian of Gaussian (LoG) pyramid (reprinted by permission from [175])

8.3.3 Multi-Scale Edges, Ridges, and Corners

In addition to blobs, other multi-scale differential invariants can also be defined using combinations of Gaussian derivatives at multiple scales. Once again, this presentation follows [175], in which differential invariants are introduced to capture edges, ridges, and corners in visual imagery.

Oriented edges—In section 8.2.3, the algorithm for detecting edges involved finding zero crossings in the second derivative response that are co-located with significant gradient (first derivative) support,

$$\begin{aligned}\nabla^2 g(x, y) = 0, \quad \text{and} \\ |\nabla g| > \tau.\end{aligned}$$

Lindeberg offers a variant of this approach that is generalized over scales. The proposed multi-scale differential invariant establishes properties in the second and third derivative using a pair of oriented Gaussian derivative operators. He introduces a local orthonormal coordinate frame (u, v) called *gauge coordinates* oriented with the v -axis parallel to the gradient direction at position (x, y) . To do so, the multi-scale Gaussian gradient operators are rotated by an amount $\phi(x, y) = \tan^{-1}(g_y/g_x)$. By this definition, g_v is always positive and the lateral first derivative, g_u , is zero. Steering the Gaussian operators in this fashion simplifies the definition of the edge feature in terms of its differential structure.

In the approach proposed in [175], edges are localized to sub-sample accuracy in places that comply with a pattern of differential responses,

$$\begin{aligned}g_{vv} = 0, \quad \text{and} \\ g_{vvv} < 0.\end{aligned}\tag{8.16}$$

Most sharp edge structures corresponding to object boundaries exist simultaneously at many scales. The left column in [figure 8.16](#) shows three images in the Gaussian pyramid and the corresponding oriented edge responses consistent with [equation 8.16](#) at three different scales.

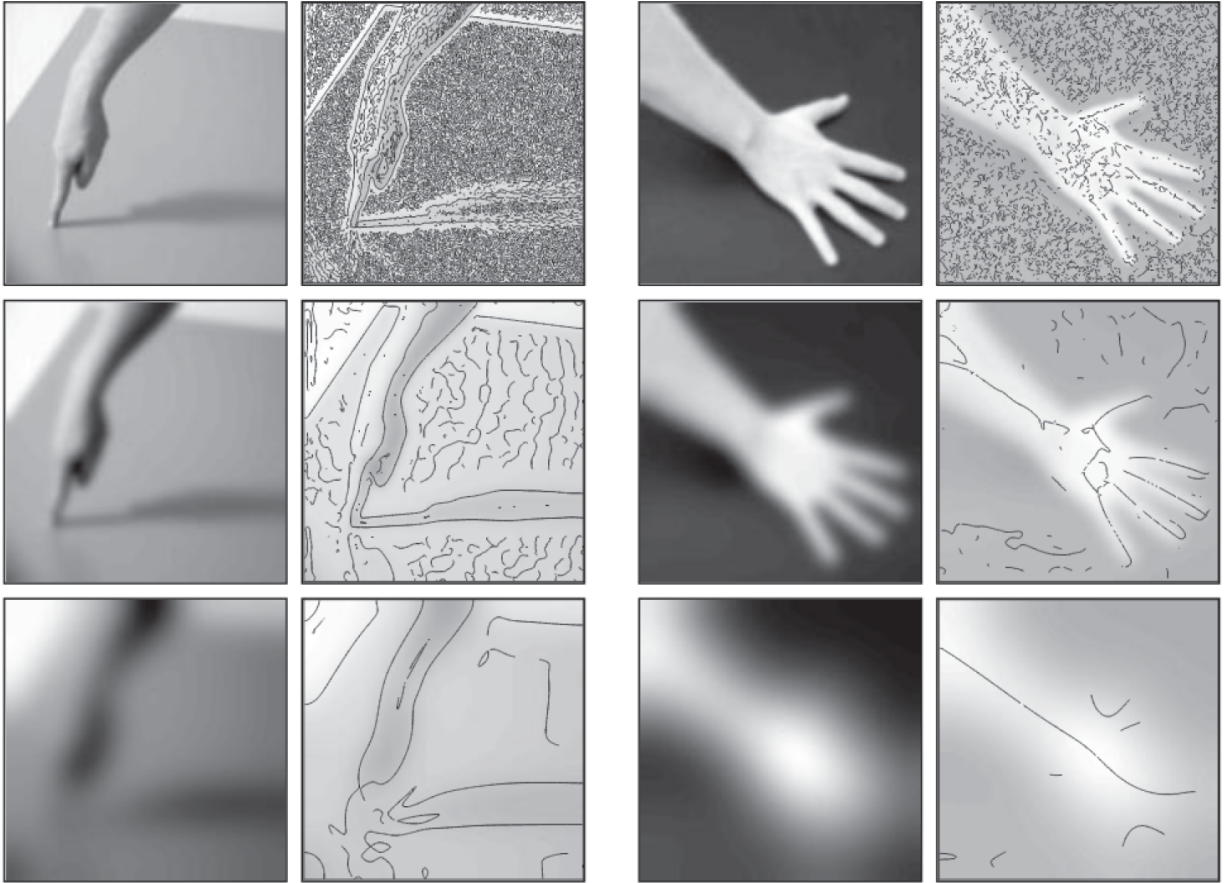


Figure 8.16

Edges and ridges reported at $\sigma=1.0$, 16.0 , and 256.0 , respectively (reprinted by permission from [175])

Ridges—Gauge coordinates (p, q) for ridge detection orient a frame with the local principal curvatures of the luminosity function such that off-diagonal terms in the image Hessian, $g_{pq} = g_{qp} = 0$. A ridge is an extremum in the same direction as the maximum principal curvature, which we align with direction p , such that

$$g_p = 0,$$

$$g_{pp} < 0, \quad \text{and} \quad |g_{pp}| \geq |g_{qq}|.$$

The pair of images in the right column of [figure 8.16](#) shows the response to this kind of ridge detector for three different scales.

[Figure 8.17](#) show the result of computing edge and ridge features at the intrinsic scale recommended by these metrics. The results illustrate that

these features extend over large spatial dimensions, across large and small scales, and link (across scales) into continuous, multi-scale entities.

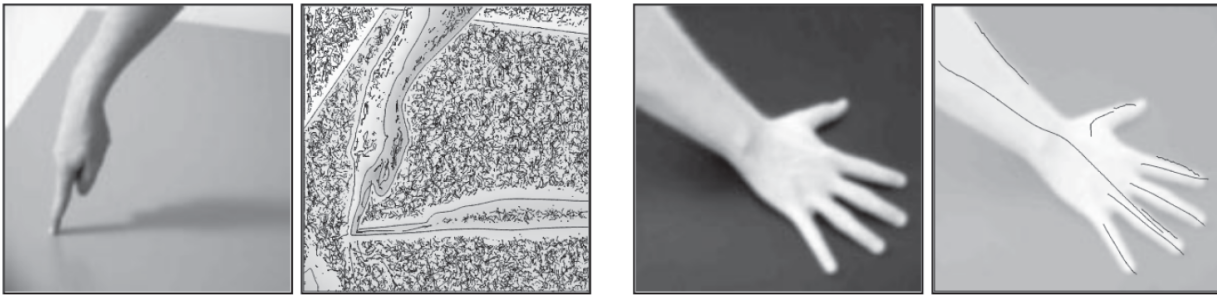


Figure 8.17

The result of automatic scale selection to detect edge (*left*) and ridge (*right*) invariants (reprinted by permission from [175])

Corners—Lindeberg also proposes a feature detector for corners using a differential geometry that maximizes

$$g_v g_{uu} = g_x^2 g_{yy} + g_y^2 g_{xx} - 2g_x g_y g_{xy}.$$

Figure 8.18 show the results of multi-scale corner features. In this rendering, one can see several examples of a corner feature where strong responses are detected at multiple scales simultaneously, thus enhancing the confidence that the environment affords a corner at that location.

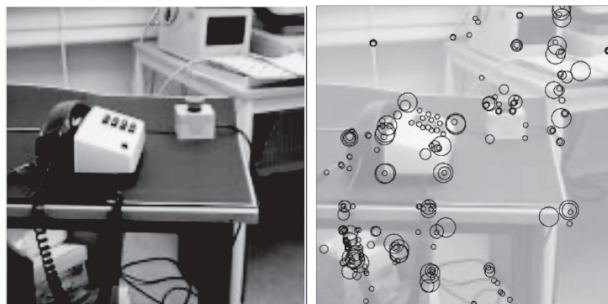


Figure 8.18

The result of automatic scale selection to detect corner invariants (reprinted by permission from [175])

Blobs, edges, ridges, and corners in scale space are highly correlated with interest points in the image. Moreover, blobs and edges have direct corollaries in spatio-temporal signals such as tactile force sensations. These

features can be used as landmarks for establishing correspondances in multiple signals, and for recovering reliable environmental geometry, and therefore, for recognizing and interacting with familiar scenes.

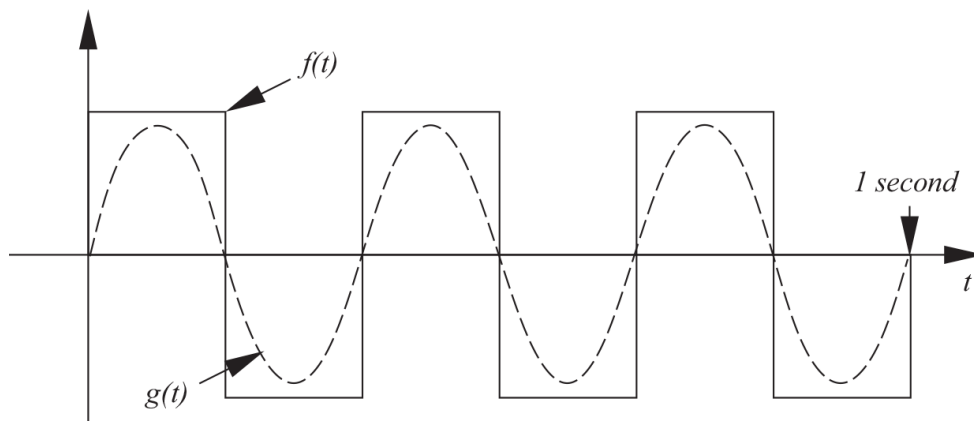
8.4 Exercises

1. **Spectral properties of the human voice.** The example in section 8.1.1 concerning the human capacity to vocalize employs the open-closed or quarter-wave resonator to approximate the human vocal tract. This model establishes the frequency-selective nature of the human organ of speech (or, in this case, song).

Luciano Pavarotti was a famous operatic tenor who was recorded hitting F5 (the second F above middle C) at a fundamental frequency ~ 698.456 [Hz]. What fundamental frequency might he have attained if he filled his lungs with helium before performing?

2. **Aliasing.** A square wave signal, $f(t)$, is depicted; $g(t)$ represents its fundamental frequency, shown as the dotted line.

- What is the bandwidth of the square wave, $f(t)$?
- Consider the sinusoidal function illustrated by the dotted line. What must the minimum sampling frequency be to reconstruct this function faithfully? Give your answer in samples per second.
- If the square wave, $f(t)$, is sampled at this frequency, will there be aliasing effects in the Fourier coefficient of $f(t)$ at 3 [Hz]?
- If the sampling rate is insufficient, what can you say about the expected effects of aliasing on the reconstructed square wave?



3. Convolution.

(a) Show that for any two functions, f and g , $f * g' = (f * g)'$.

(b) The Laplacian operator approximates the second derivative of the image function. Derive the Laplacian operator as it appears in section 8.2.3 using finite differences.

4. **Frei-Chen: Appearance-based primitives.** Show that the Frei-Chen operators are orthogonal representations of 3×3 image regions.

5. **Invent your own homework.** Make your own exercise out of your favorite content in chapter 8. Write a question and a solution for it from the material in the reading. The problem should be different from the questions already here. Ideally, it should be based on material in this chapter (and, perhaps, previous chapters) and focused on sensors; it should call for a quantitative analysis and a short discussion; it should not take more than 30 [*min*] to solve open-book.

Part III Summary: Transducers, Signals, and Perceptual Structure

Part III begins in chapter 7, where important organs of visual and tactile sensation are discussed. The physics of visual image formation, that illustrate how light is captured, focused onto the image plane, and sampled in both space and time, are introduced. The sense of touch is also highlighted in human beings in terms of rich array of mechanoreceptors and a remarkable (and less well understood) ability in humans to discern the “touch blend” of haptic categories present in multi-modal contact sensation.

In chapter 8, techniques for frequency analysis in generic signals are introduced. We derived the sampling theorem by considering how a continuous signal is altered when it is approximated as a set of discrete samples and the conditions that lead to spurious information (called aliasing) in the sampled feedback. Several tools for linear signal processing using convolution were presented, including spectral filtering; a variety of finite difference operators used to detect edges; and filters that detect the differential structure of signals. The appearance-based decomposition of luminance functions is presented and the Gaussian kernel is introduced as the correct means of generalizing these approaches to multi-scale differential filters used to detect the deep structure of signals. We have taken an information-theoretic perspective on signals that is uncommitted to any particular task, sensor, or environment. This view comes from the computer vision and machine learning communities over the last twenty years but is still an emerging paradigm in the robotics world. Part III concludes by presenting several famous examples from scale-space vision processing that is widely used and influential in computer vision.

With a large library of analytical tools and techniques at our disposal, part IV ties all of the insight presented up to this point into a discussion of sensorimotor development in human infants and a computational architecture that supports experimental work on developmental programming in an embodied (robot) system.

1. Here, $F_i = \omega_i/(2\pi)$ is the i -th resonance frequency in the quarter-wave resonator in [Hz]. This should not be confused with $F(\omega)$ that denotes the spectral coefficients in [equations 8.1](#) and [8.2](#).
2. The opposite is also true; convolution in the frequency domain is equivalent to multiplication in the spatial domain.
3. The Frobenius inner product is defined as $\mathbf{A} : \mathbf{B} = \text{trace}(\mathbf{A}\mathbf{B}^T) = \text{trace}(\mathbf{A}^T\mathbf{B})$.
4. Here, we generalized the term embodied to include extensions of the literal body, like those afforded by microscopes, telescopes, wheels, levers, and airplanes.

IV

SENSORIMOTOR DEVELOPMENT

9

Infant Neurodevelopmental Organization

The human genome specifies everything that defines *Homo sapiens* in terms of body shape, organs of perception and movement, and the configuration of a hierarchy of embedded processing from peripheral sensors to spinal cord, brainstem, cerebellum, and midbrain into the newest component of the central nervous system—the cerebral cortex. All of these innovations are the common genetic legacy of the species, as are important processes of growth and maturation that transform the newborn into an adult. Growth produces biochemical and neurological changes over time that have a profound impact on morphological and computational abilities and change a human being's relationship to the environment. Collectively, all of these developmental mechanisms lead to individuals that acquire physical and cognitive abilities well suited to life on Earth. This organization represents an embodied cognitive developmental system in which genetically mediated processes of growth and maturation play critical computational roles to provide a pathway to adulthood—including important physical and perceptual skills as well as the knowledge that supports critical decision-making and control.

In this chapter, we review the evolution of the contemporary human brain and focus on the role played by cortical control hierarchies to guide behavior and model a lifetime of integrated sensory and motor experience. The chapter continues by reviewing the neurological organization of the newborn human infant that leads reliably through milestones in sensorimotor development during the first year of life.

9.1 The Evolution of the Brain

Over the course of millions of years, our ancestors evolved mechanisms for connecting stimuli to response. In its simplest form, a signal that originates in a peripheral receptor passes through a synapse to innervate a group of muscle cells called a motor unit. As many such mappings evolved, prominent ganglia emerged consisting of clusters of highly interconnected nerve cells. Several such ganglia, organized in an elongated neural tube (figure 9.1), gave birth to highly specialized organs in the central nervous system. Animals with such a dorsal neural tube—fishes, reptiles, birds, and mammals—are called chordates and belong to the phylum chordata. These animals use the neural tube to connect perception to action at many different scales in space and time. Vertebrates evolved a bony spine about the cord to provide structure and to protect the neural tube. The elongated central ganglion receives inputs (afferents) from dorsal fibers and produces outputs (efferents) via ventral motor processes (figure 9.1).

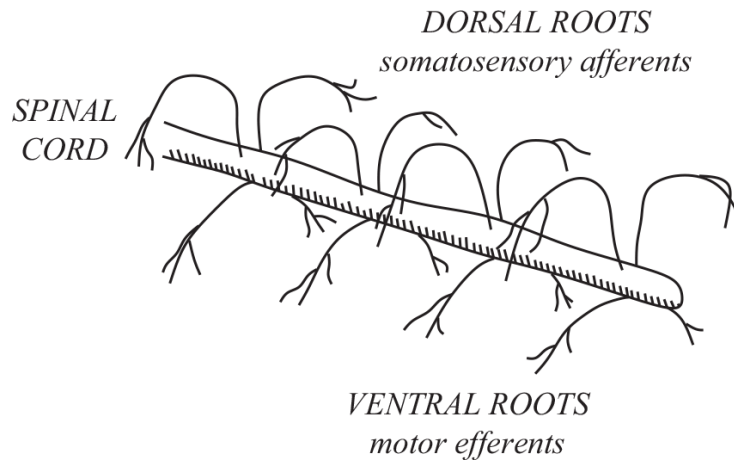


Figure 9.1

The afferents and efferents of the spinal cord

The dorsal-ventral organization was an early form of spatial specialization in the nervous system. Somatosensory fibers bring information about external stimuli, muscle traction, and tension in tendons and ligaments to the spinal cord, and motor fibers conduct reflexive actions back out to the muscles. In time, a dense brainstem consisting of specialized cells formed at one end (thus, defining the anterior aspect) of the ganglion

(figure 9.2). This organ, sometimes called the *animal* brain, is organized into the hindbrain, the midbrain, and the forebrain. The forebrain specialized originally to process olfactory information. However, in higher animals, it performs the bulk of the early sensory processing (see the reptilian brain in figure 9.2). The midbrain participates in the processing of sensory information and the hindbrain specialized to manage complex motor systems.

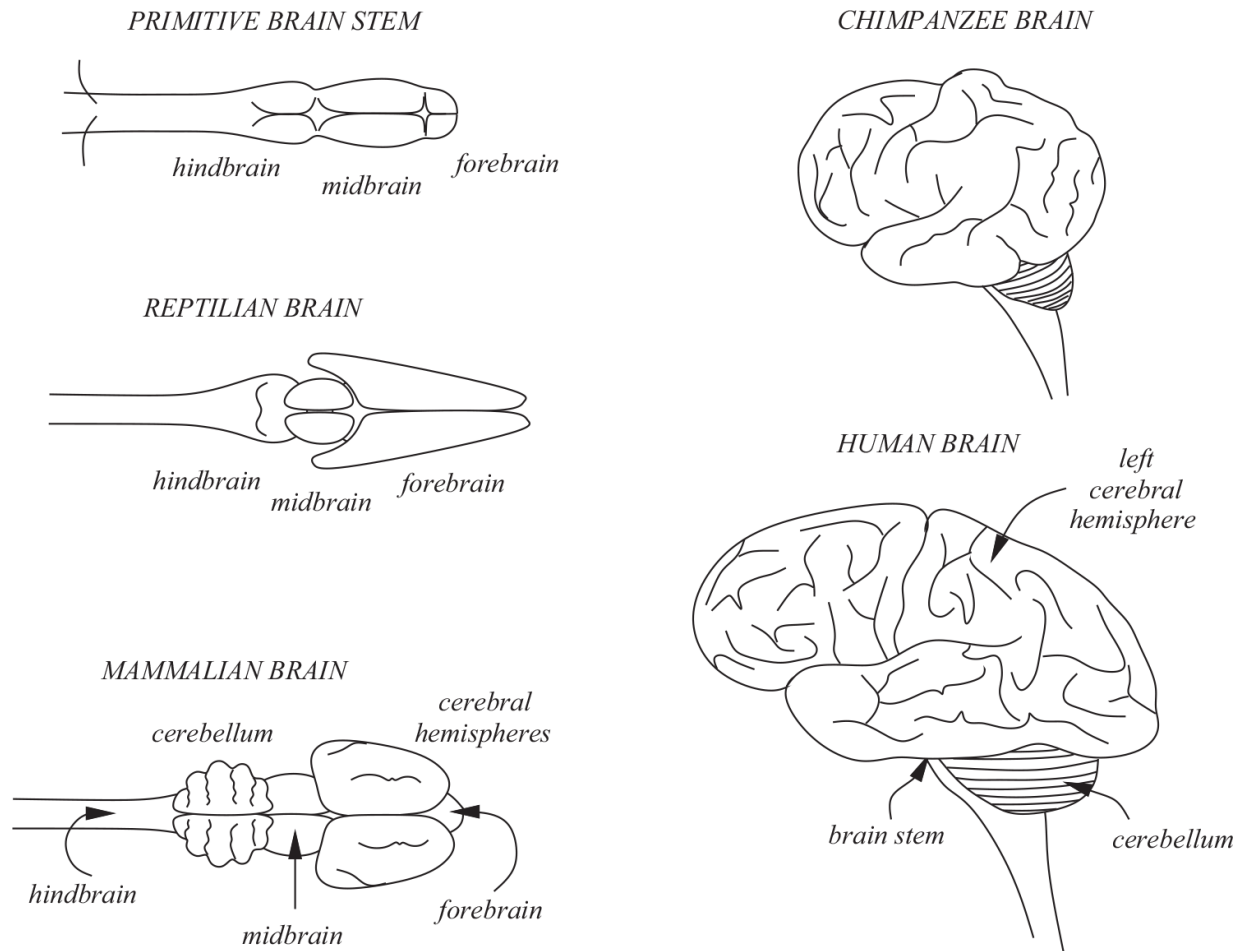


Figure 9.2
The evolution of the human brain

The cerebellum developed to smooth and coordinate the motor systems in the increasingly agile mammalian body. The most recent innovation in the central nervous system of mammals is the cerebrum, which incorporates the cerebral cortex (or just cortex), an organ specialized for interpreting and integrating a rich variety of sensory signals and for initiating sequential

motor behavior. The evolutionary process by which learned (cerebellar and cerebral) behavior is overlaid on top of autonomic control persisted through the early mammals and primates and into the human being (figure 9.2). In the human brain, these new structures dominate the older parts of the central nervous system. The hierarchy in the cortex extends the temporal scope of the neural control architecture and creates increasingly stable situational feedback from noisy sensors and unstructured environments. In the economy of cognitive organization, the value of increasing perceptual sophistication and motor acuity as well as an enormous capacity to remember and differentiate subtle environmental contexts justifies the expenditure of up to 20 percent of a human's resting metabolism in the neocortex.

9.2 Hierarchy in the Neocortex

The cortex (a word derived from the Latin word for the bark of a tree) exhibits a particularly dense organization of highly interconnected neurons. This structure forms the outer layer of the cerebral hemispheres, often called the gray matter. It is highly convoluted to maximize the surface area of the layer within the confined volume of the cranium. Unfolded, the 2 [mm] thick sheet of cortical tissue is about 0.3 [m^2] in area [110].

Signals stream into and out of the cortex from all over the periphery. About 1 million neural fibers project via the optic nerve through the subcortical structures¹ to the primary visual cortex. Another million ascend the spinal cord carrying proprioceptive and tactile stimuli from the periphery to the somatosensory cortex, and about 30,000 neural fibers carry auditory information to the primary auditory cortex.

There are about 100, 000 neurons per square millimeter in the cortex; this corresponds, conservatively, to a total of 30 billion neurons. Assuming a modest 1000 synapses per cell, a back-of-the-envelope calculation reveals that there are roughly 30 trillion synapses forming the cortical associative memory. The capacity of such a system is enormous. In a book entitled *On Intelligence*, Jeff Hawkins [110] observed that the cortical tissue computes a forward model that stores sequences of patterns and predicts what will happen next. The text contends that the brain's immense capacity as an associative memory derives from a simple replicated structure that is

important to understanding cortical function. Across the 2 [mm] thickness of the cortical “bark,” there exists a common six-layer columnar structure that includes ascending (encoding) and descending (decoding) projections. This relatively simple structure forms the basic unit of cortical computation [200].

Underneath the convoluted cortical layer in the brain resides a larger volume of white matter that provides metabolic support and contains relatively long axonal projections from one cortical area to another. A broad class of pyramidal neurons—named for the shape of the cell body—resides throughout the central nervous system and, importantly, in the top layer of the columnar cortical structure, where they project miles of axons to other cortical areas.

One can get a sense of the data flow in the cortex by tracing neural projections between functional areas in the pyramidal system. The cortex is organized into several functional areas that preserve the spatial qualities of the signal—spatial qualities that reflect the geometry of the agent’s body as well as that of the outside world. For example, mechanoreceptors in the skin project neural fibers that enter the spinal cord via dorsal roots (figure 9.1), and this information is conducted up the cord, through midbrain structures, and to the thalamus. From here, peripheral sensory information is projected onto the cerebral hemispheres in a region known as the somatosensory cortex (figure 9.3). The primary somatosensory cortex (region S1) is a map of the peripheral sensor distribution where the amount of cortical tissue reflects the density of peripheral sensors—the hands, feet, and digestive tract receiving a great deal of attention in the somatosensory cortex. The mapping on the cortex preserves the topology of the body, as illustrated in figure 9.3. Sensor processes project contra laterally to the cortex so that the left hemisphere receives stimuli from the right side of the body and vice versa.

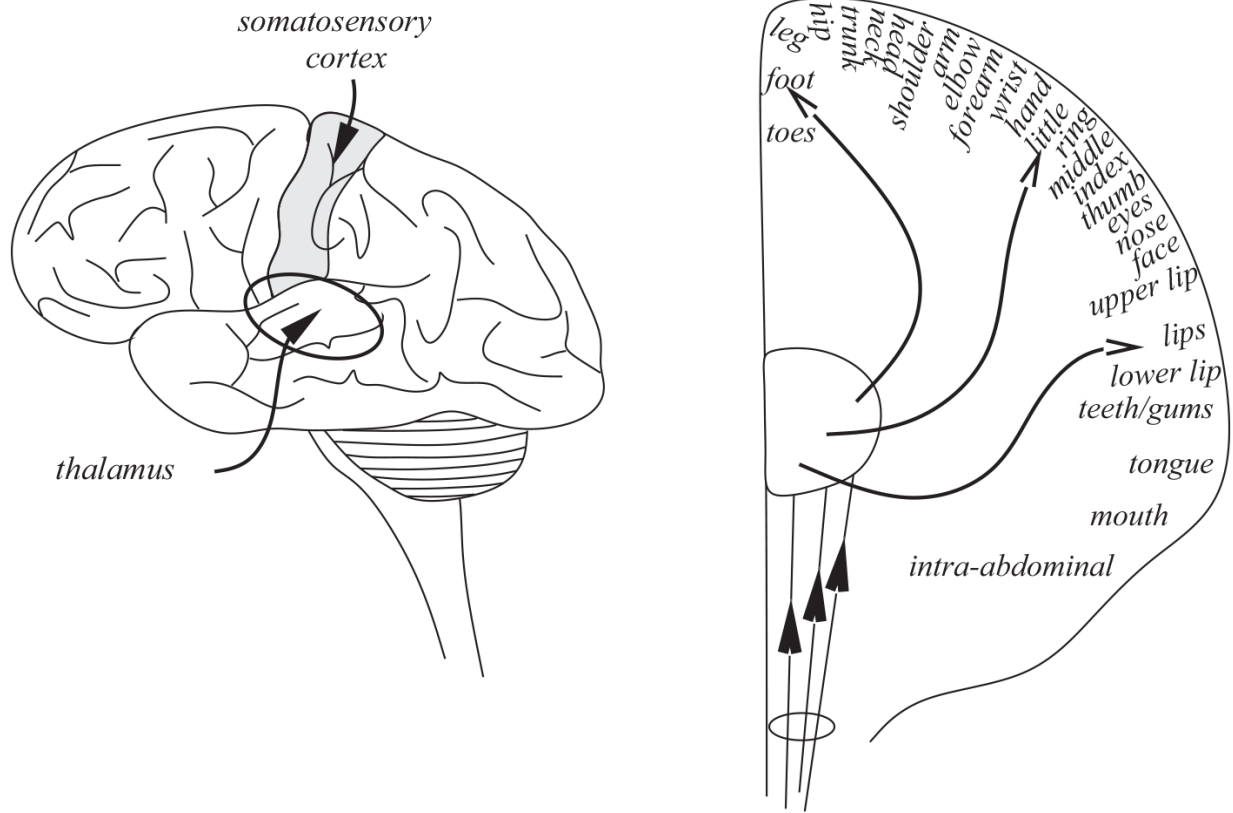


Figure 9.3
The sensory homunculus

The same topographic organization is evident in the visual pathways. The rods and cones of the retina project immediately onto an interconnected layer of neurons that perform local computations on signals from small, compact neighborhoods of retinal receptors. After this preprocessing, visual information is conducted along an elongated ganglion and exits the retina as the optic nerve. The optic nerve projects to a cortical nucleus (a region of interconnected cell bodies) at the rear of the thalamus called the lateral geniculate nucleus (LGN). The cells here also represent a retinotopic map of the visual field that is distorted in a manner that reflects salience. In this case, the fovea or optical center of the retina dominates this map. From here, visual information is projected onto the *primary visual cortex* (region V1), illustrated in [figure 9.4](#). Populations of cortical and precortical neurons are highly specialized experts at extracting important kinds of information from a visual scene. Some are proficient at isolating edges (or sharp intensity variations) oriented in certain directions. Others detect motion in

the receptive field or are specialized to interpret the disparities between left and right images as depth in the visual scene. There exists a hierarchical abstraction of the visual information, up to the level of objects and more global contexts in the scene.

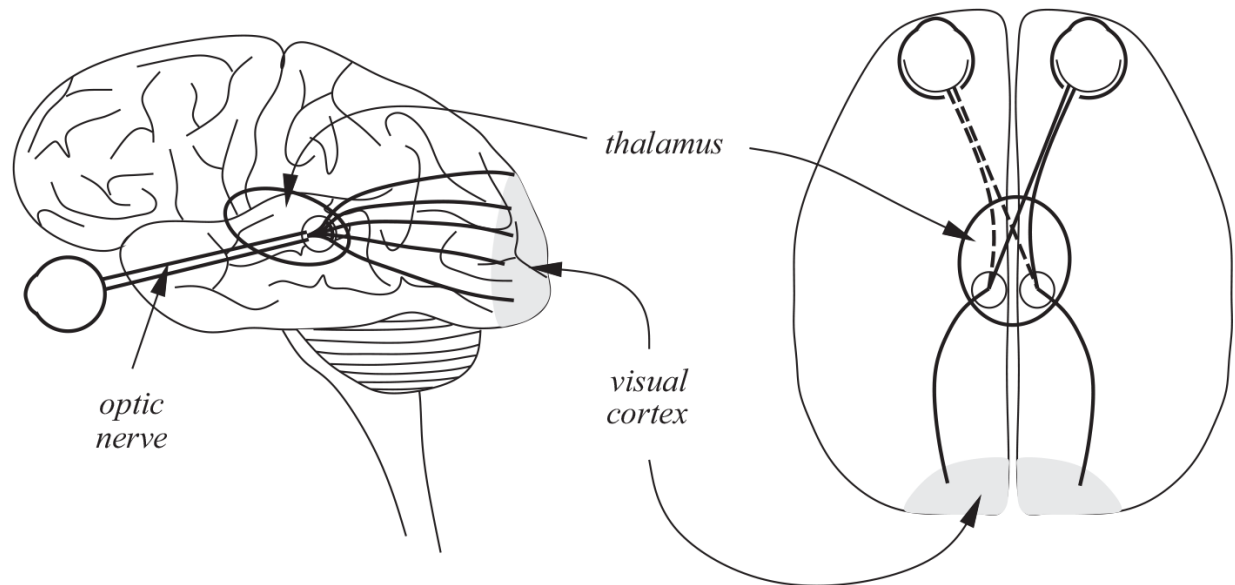


Figure 9.4
The visual cortex

In [192], there is an account attributed to Robert Boyle, a seventeenth-century Irish physicist, of a knight that suffered a depressed fracture of the skull and paralysis of one side of the body until the local barber elevated the depressed bone. Boyle suggested that this implies a specialized area of the cortex devoted to movement of the limbs. The *motor cortex* is a topographic map of the musculoskeletal system (figure 9.5). Neural processes from the motor cortex project through the thalamus, influenced by the basal ganglia, into the spinal cord, and eventually to the motor units that activate muscle cells. This is, of course, a major oversimplification. A single voluntary movement actually involves large synergies of muscle activity that are sensitive to dynamic and geometric constraints, as well as the context of the task. The motor cortex has reciprocal connections to the cerebellum and the basal ganglia. Some of the output generated in the motor cortex passes almost directly to the spinal cord. Precise individual finger movements are an example of such cortical muscle control. However, the primary role of

the motor cortex is to innervate lower-level motor subsystems. It is one element of a hierarchical motor organization that extends from low-level reflex arcs to long-term, deliberate motor plans.

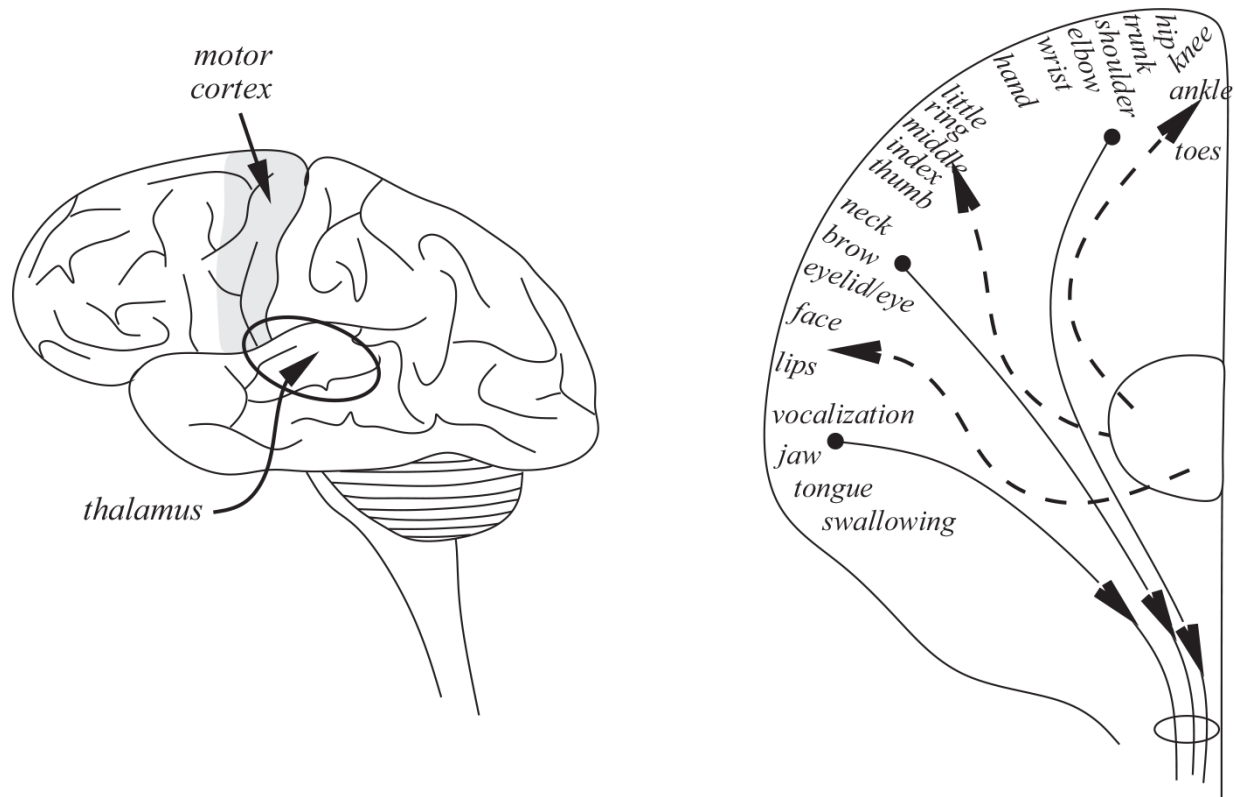


Figure 9.5
The motor cortex

Figure 9.6 is a schematic description of the flow of information in the cortex. Signals from the sensory organs project into the primary cortical areas where they are recoded and passed to secondary cortical regions. Information from other sensory and motor sources is combined in the association regions of the cortex. These association areas are less well understood. It may be here that the cumulative experience of the individual—acquired sensory and motor expertise and decisions regarding risk and reward—influences motor processes that interact with the environment.

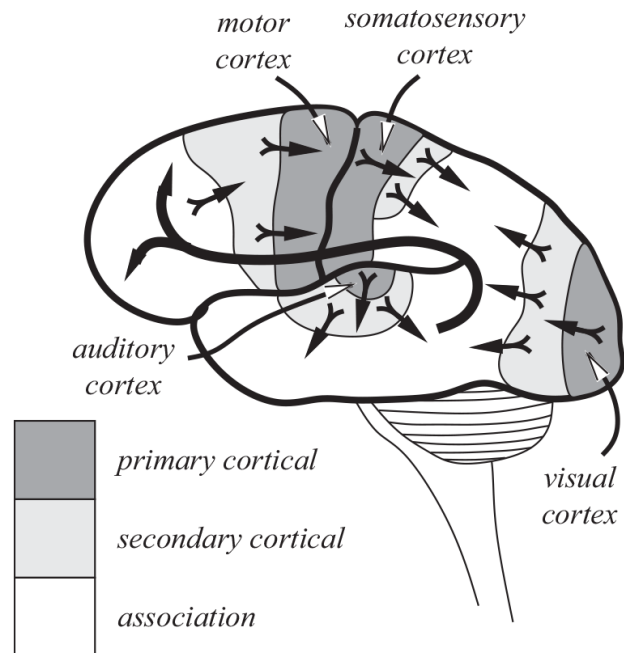


Figure 9.6

The flow of information and control from sensory to motor areas of the brain

Rather than a view of brain in which a world model insulates and decouples motor functions from perceptual functions, our understanding of neuroanatomy suggests an architecture in which information—from sensors and/or memory—are coupled hierarchically to motor activities. This view is summarized in [figure 9.7](#). At one extreme, sensor-motor mappings like the withdrawal response produce the fastest response times possible in the nervous system. This end of the motor control hierarchy is compiled into human neuroanatomy and resides in the spinal cord, the brainstem, and the cerebellum. Higher-level cortical pathways for control respond to training and continually reassess feedback. Each new layer in the information processing hierarchy *recodes* input stimuli to detect features that inform and stabilize decision-making.

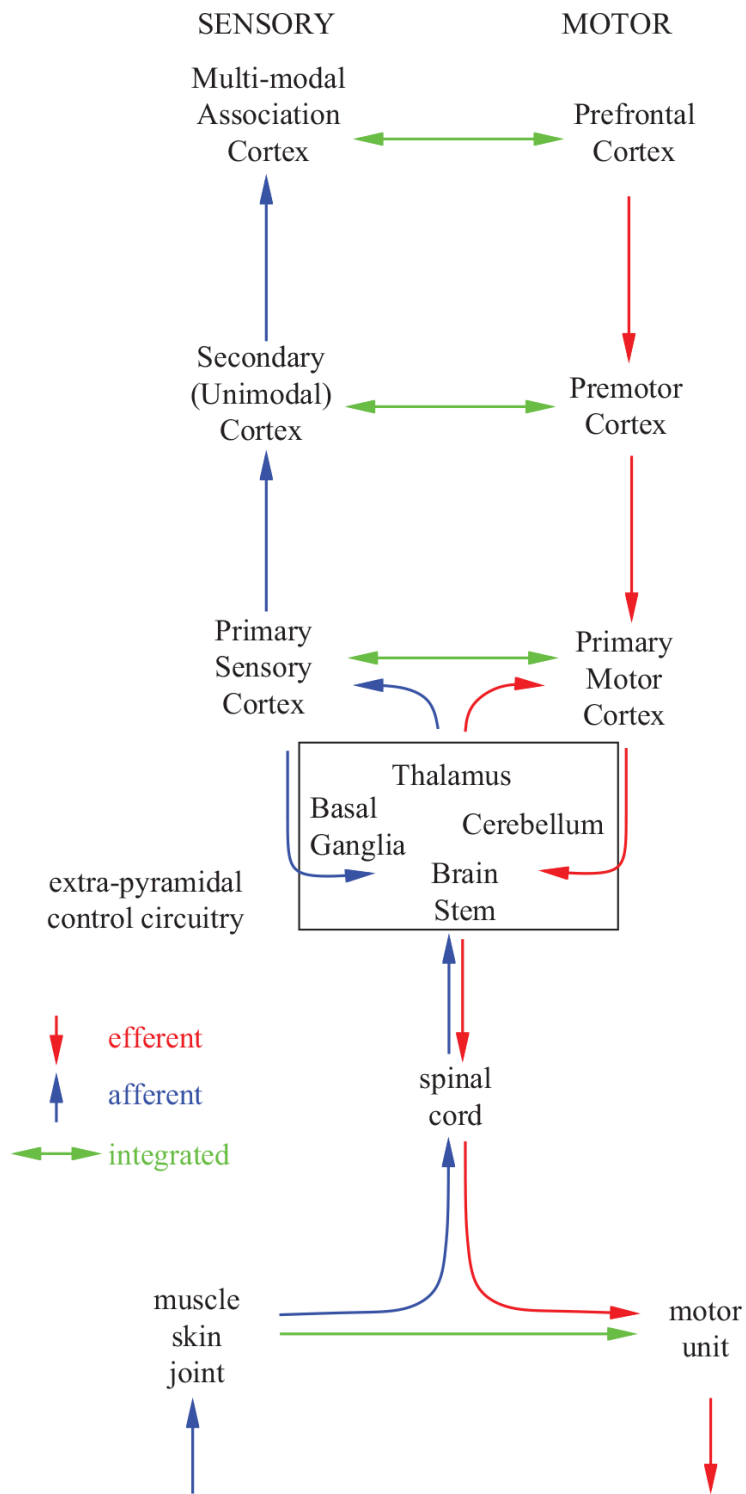


Figure 9.7
Hierarchies in the sensory and motor organization of human behavior

9.3 Neurodevelopmental Organization

Pathfinder was launched on December 4, 1996. After 28 weeks en route and a traumatic entry into the Martian atmosphere, it bounced to a stable position on the planet's surface on July 4, 1997. It had been dormant in transit to Mars, and now, after having come to rest, it underwent a metamorphosis into its working configuration—unfurling its body from the cramped confines of the spacecraft and assuming a configuration for mobility and exploration that it would have for the rest of its life. *Pathfinder* sent its final data transmission on September 27, 1997.

When a human infant is born, it has been in utero for 37–42 weeks. Its muscles are weak and the range of motion is severely limited. The human infant, unlike *Pathfinder*, has been very busy in the interim. Neural, sensory, and motor systems differentiated at the cellular level, developmental reflexes are prepared to exercise stiff joints and strengthen muscles after birth, to coordinate limbs, to protect the newborn, and to provide the kind of experiences that bootstrap cognitive models of the world it has entered. These reflexes serve developmental roles before, during, and long after the child is born. The vast majority of physical, neurological, and cognitive development occurs after birth, when growth and interaction with the environment express the potential encoded in the human genome.

The neonate² was prepared over the course of millions of years of evolution for a developmental trajectory that simultaneously discovers the abilities of its body, important properties of the environment, and the latent potential for controlled interactions between the two. This period of growth extends through significant changes in size and body mass, the hardness of bones and skeletal geometry, strength, and sensory acuity driven by an innate desire to interact with the world and to explore. The prohibitive complexity of this process is incomprehensible to the neonate and yet, a similar period of growth will reliably shepherd generations of newborns through infancy, adolescence, and toward adulthood within a decade or two.

Many reflexive stimulus-response arcs exist throughout the nervous system that get this growth started. These circuits have evolved with the body of the organism and are encoded in terms of neural pathways representing common patterns of interaction between the organism and the environment over the evolutionary time scale. The limbic reflexes contribute to survival and the normal metabolic functions of the organism.

They address visceral and vegetative functions, protect our sensory and motor organs, and control swallowing and breathing, often without any conscious attention [28]. Developmental reflexes, however, can also bootstrap hierarchical compositions of stimulus-response reactions by preparing the organism to *learn*. They serve as building blocks for more complex acquired behavior, and they support training contexts that make learning data efficient in situ. For example, the primary stepping reflex (section 9.3.2) is present at birth and yet does not participate in integrated walking behavior until sometime around the end of the first year. Before that, maturational mechanisms inhibit its expression and focus instead on vestibular responses and postural stability (section 9.3.4) to stiffen the leg, support the toddler’s weight, and provide a developmental context for learning to balance. Piaget [227] called such patterns—capable of prolonging certain types of interactions with the world—“primary circular reactions.” Thus, developmental mechanisms manage the learning process to acquire knowledge interactively in stages, in a manner that avoids the worst-case complexity of flat learning problems.

Developmental reflexes may persist for a lifetime or, after having served their roles, may disappear by inhibition or cell death. Reflexes that are inhibited once they have served their developmental role can reemerge as a result of disease or trauma. Clinical neurologists use reemergence as a diagnostic tool to help localize lesions in the central nervous system. Other developmental reflexes transform into mature forms of the original stimulus-response behavior. For example, the palmar grasp reflex (section 9.3.2) is elaborated over time and morphs into mature forms of the original reflex (e.g., the pincer grasp in figure 9.18). Both of these mechanisms of development, composition and adaptation, rely on primitive sensorimotor behavior (or primitive “preadaptation” behavior [42]) from which other motor skills are constructed.

There are well-documented clinical accounts of developmental reflexes, their onset and persistence, and the techniques required to elicit them, including videos demonstrating their characteristic presentation. The summary presented in this chapter is drawn from many such sources. Principle among these are the excellent accounts found in [165, 74, 87]. The remainder of this chapter will catalog many of the developmental reflexes cited in the literature in terms of their onset and persistence—the conditions

that cause them to emerge and those that, subsequently, cause them to be inhibited, respectively. We consider the roles that each of these reflexes play in helping the infant learn gross and fine motor skills in a developmental trajectory. The chapter concludes by describing several cognitive milestones that go along with these developmental artifacts.

9.3.1 Limbic Reflexes: Visceral, Vegetative, and Behavioral

A number of reflexive responses contribute to the normal function of an organism's body. They are the firmware of the biological plant.³ Protection is a central concern. For example, the **flexor-withdrawal reflex** in human beings (the neuroanatomy of which is discussed in chapter 3) consists of the flexion and withdrawal of an extremity in response to a painful stimulus. It is illustrated by the rapid withdrawal of one's hand and arm when the fingers come into contact with a hot object or the flexion of the ankle, knee, and hip as a result of a painful stimulus on the sole of the foot. This reflexive tactile response has counterparts that are triggered by auditory and visual stimuli. The **acoustic stapedius reflex** innervates the stapedius muscle (the smallest muscle in the human body) attached to the tiny stapes bone in the middle ear in response to loud noises. Noises of sufficient amplitude (80–90 [dB]) cause the muscle to contract and protect the middle ear from excessive excitation. This reflex is also active just prior to speaking to attenuate the response to one's own voice. **Oculomotor reflexes** can cause pupillary responses that dilate or contract the iris to control the intensity of light striking the retina. A pain in the neck can dilate the ipsilateral (same side) pupil, a response called **ciliospinal dilation**, the **orbicularis reflex** causes a forceful closing of the eye in response to pain, and a defensive eye **blink** is triggered by visual looming or corneal contact.

Other reflexes concern the vegetative needs of the organism. For example, many visceral reflexes are concerned with the control of the movement of food and waste.

Pharyngeal reflex—Also known as the swallowing reflex, this response is part of a complex orchestrated behavior to propel food into the stomach. Food is pushed to the back of the mouth where pharyngeal swallow is triggered. It consists of a peristaltic wave, moving food downward while protecting the airway.

Gag reflex—This esophageal reflex contracts the back of the throat and thus prevents objects from entering the throat except as part of normal swallowing to help prevent choking.

Elimination reflexes—An anal sphincter contraction reflex and the micturition reflex participate in the elimination of solid and liquid wastes, respectively.

As soon as a newborn emerges from the birth canal, it must begin the lifelong task of respiration. Several reflexes aid the autonomic processes of breathing:

Laryngeal reflex—The laryngeal reflex is one of the protective respiratory reflexes. It closes the vocal folds, and thus the airway, in response to irritating stimuli in the larynx, trachea, and bronchi.

Sneeze reflex—Receptors located in the nose or pharynx instigate a deep inhale, followed by an explosive expiration. This is a response to irritation, including photic sneeze reflexes (in response to bright light) present in about one in four people.

Attentional primitives are also expressed in a suite of reflexive behavior. A family of orienting reflexes exists that both direct attention to stimuli and configure sensors for attending in certain ways. Sensitivity can be enhanced toward novel stimuli—a loud noise or a flash of light. For example, the **audito-oculogyric reflex** turns the eyes toward a sharp sound. Orienting responses direct attention to the location of the new stimulus, including altering the initial response to become more sensitive to the stimulation (e.g., using dilation of the pupil to improve visual sensitivity to dim light).

There are many, many more examples of reflexes that contribute to normal function, but these reflexes do not always contribute to development per se.

9.3.2 Spinal- and Brainstem-Mediated Reflexes

Primitive reflexes are generally considered to foster cognitive development by helping an organism exercise useful neuronal pathways and by providing a compositional basis for higher-level behavior. In this section, we will

review a number of the primitive reflexes often cited in the clinical neurology literature.

Sucking/rooting/tongue retrusion reflexes—These reflexes are part of an array of related reflexes that coordinate finding objects/food and impelling (or expelling) them into (or out of) the mouth. Rooting and sucking emerge 24–28 weeks in utero and are fully present at birth; at this stage, they can be inhibited by satiation. They are strongest in the first 2 months and are inhibited normally at 3–4 months. They are triggered by tactile stimulation on the cheek (rooting), stimulation on the surface of the lip (sucking), and pressure from a hard object on the tongue (retrusion). Sucking develops the oral muscles, interacts with tongue control and swallowing, and inhibits the gag reflex. If retained too long, these feeding reflexes can affect normal swallowing, speech, and manual dexterity. Primitive reflexes like these can resurface in adult patients with senility or traumatic brain injury.

Moro reflex—Illustrated in [figure 9.8](#), this reflex is sometimes called a clasp reflex—a panic reflex that causes the newborn to hang on and cry in alarm: the arms go out and as the response grows, they come back in again in a clasp action with a loud cry.

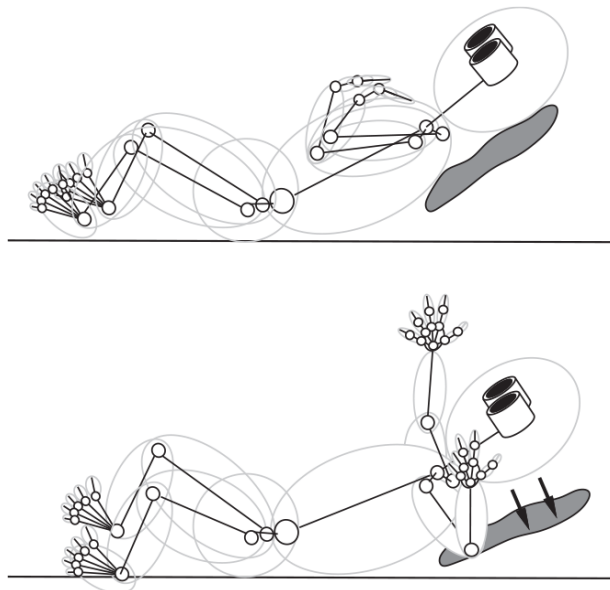


Figure 9.8
The Moro reflex

To elicit a Moro response, the baby's head and shoulders are supported, with the arms in flexion across the chest. From this posture, the head and shoulders are suddenly dropped a few inches while releasing the arms. The arms should fully abduct and extend and then return to the midline. This reflex emerges 8–9 weeks in utero, is fully present at birth, and is masked/inhibited at about 16 weeks.

The uterine presentation involves opening and then closing the arms and legs in response to a sudden loud sound. The uninhibited Moro reflex is attributed to brainstem (subcortical) activity, which is transformed into the adult *startle* response—the **Strauss reflex**—at about 16 weeks. The Strauss response engages more perceptual areas of the cortex to sort through the range of possible triggering stimuli.

The **palmar grasp reflex** (figure 9.9) is localized to spinal segments C6 to C8 just below the brainstem. A tactile stimulation on the palm or the sole of the foot can cause simultaneous flexion of the fingers or toes. In the hands, this motion often causes force closure (section 5.3) about the source of the tactile stimulation. It emerges at about 11 weeks in utero and persists for 2–3 months after birth; however, its *fractionated* form (the reflex for a single finger) can persist until 10 months of age. These movements participate in single-finger oppositions with the thumb. At its peak, the infant can support their complete body weight suspended from palmar grasps and traction response reflexes.

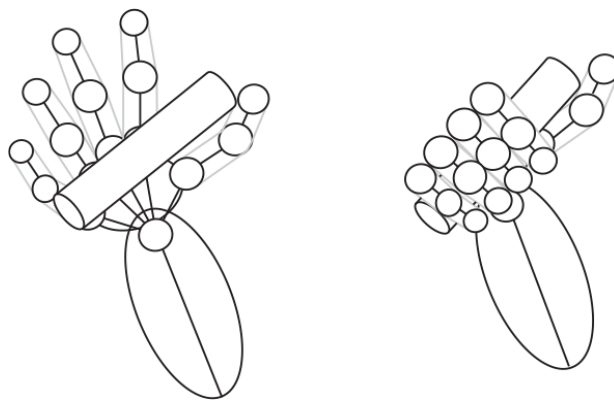


Figure 9.9

The palmar grasp reflex

The palmar reflex can be activated by sucking movements as well because the hands and mouth are related through the **Babkin response**. This type of innervation is gradually obscured by voluntary activity in roughly 4–6 months. If the palmar grasp reflex is retained beyond its normal developmental period, it may delay the development of individuated finger movements and hand-to-mouth coordination, may lead to swallowing problems and delayed speech. The pincer grip is the mature successor of the palmar reflex. It normally takes over at about 36 weeks after birth.

Stroking the lateral aspect of the foot near the plantar area (heel) will evoke the **plantar reflex** (figure 9.10), which causes the toes to fan out in extension. If the stimulus is directed across the ball of the foot, then a grasp reflex can be elicited. The plantar reflex emerges at 18 weeks in utero, is fully present at birth, and is normally inhibited at about 6 months. It produces kicking movements, enhances muscle tone, provides vestibular stimulation, and assists the birthing process. In addition, it begins a pattern of innervation of neural pathways that are used later for creeping, lateralized crawling movements, and hand-eye coordination and helps to integrate vestibular feedback into motor control.

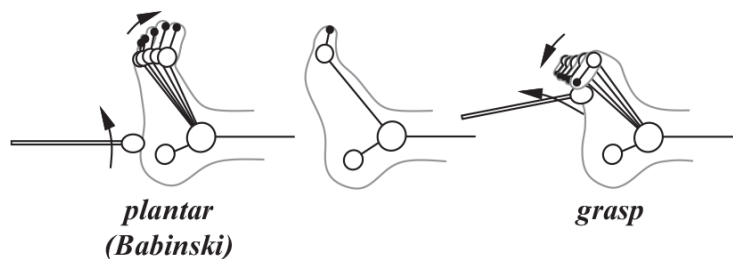


Figure 9.10

The plantar reflex

The **asymmetric tonic neck reflex (ATNR)** in figure 9.11 is evoked in response to a head turn and causes extension of the ipsilateral (same side) extremities and flexion in the contralateral limbs. This reflex emerges around 18 weeks in utero and is normally inhibited around 6 months after birth. It is believed that ATNR facilitates movement down the birth canal; helps to maintain unobstructed airways when lying prone; and focuses the eyes on the outstretched hand, thus extending the infant's ability to focus from about 17 [cm] (at birth) to arm's length. If the ATNR persists beyond

the normal period, it may frustrate the ability to focus at larger ranges and lead to preferences for homolateral movements of the arms and legs instead of important contralateral coordination patterns that are required for crawling and walking. Further, it may inhibit bimanual manipulation, cause delays in the development of a preferred hand/eye/ear/leg, and lead to poor ocular pursuit across midline.

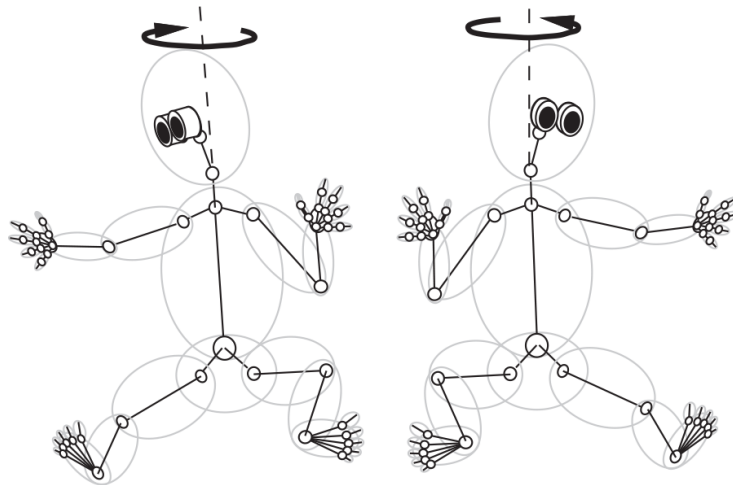


Figure 9.11
The asymmetric tonic neck reflex

The **Galant reflex** in [figure 9.12](#) is believed to be a spinal reflex. It emerges at 20 weeks in utero, is present at birth, and is inhibited by 3–9 months. It presents as an ipsilateral curving of the trunk in response to tactile stimulation on the flank and plays a role in learning to rotate and use torso movements in crawling, creeping, and walking. If it is retained too long, it can induce an asymmetric hip rotation during walking as well as posture and gait anomalies, and it can also interfere with the development of the **amphibian reflex** and the **segmental rolling reflex**.

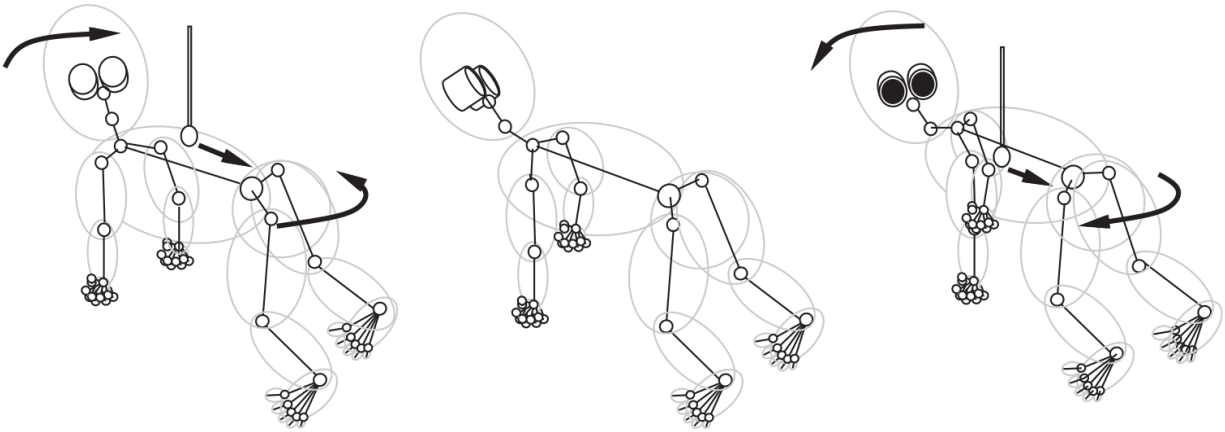


Figure 9.12
The Galant reflex

The **placing reaction** is mediated in subcortical nuclei (basal ganglia) and leads to a retraction, raising, and replacement of a limb (arm or leg) in response to contact on the shin or forearm. Imagine bumping your shin on a curb; the placing reaction responds during its active period by raising your foot up and stepping back down on the top of the curb. It emerges at about 9 months in utero and disappears by inhibition or integration at about 6 months after birth.

The **primary stepping reflex** is a spinal reflex triggered when the sole of the foot touches a supporting surface. A leg extension at the back of a stride (near toe off) causes a reflexive flexion, forward hip rotation, and then a leg extension again in anticipation of heel strike at the front of the stride. The result resembles a lateralized stepping action. It can be elicited at birth well before the newborn is ready to walk unattended, and it will disappear 3–4 weeks after birth. It reappears as a component of the mature, integrated gait at about 12–24 months in concert with a suite of supporting postural reflexes. This is about the point in time when the characteristically wide toddler's gait narrows and forward progress improves. The reflex temporarily disappears due to inappropriate gait dynamics and reemerges when leg length, mass, and strength are once again favorable during this period of growth [272]. Presumably, the reflex is intact, but its ability to create large amplitude movements in the limb is compromised. In such a situation, other postural reflexes dominate. There is some controversy about whether exercising the reflex helps or hinders the development of unassisted walking. In one study, it was concluded that exercise preserves

and strengthens the areas of the cortex associated with movement and, thus, integrates these areas more strongly with the spinal stepping reflex [303].

The corticospinal system matures during the first 5–7 years. Before maturation, primitive reflexes are pronounced and can be elicited clinically. However, as the corticospinal system matures, cortical systems influence control to a greater degree, and many primitive reflexes become difficult or impossible to elicit. For instance, the innervation of the plantar reflex yields a normal positive Babinski sign in the 3-month-old infant. It should not persist beyond 6 months of age. In the mature adult, however, corticospinal lesions due to trauma or dementia may remove inhibitory pathways and permit the primitive Babinski sign to reappear [28]. Clinicians can use such evaluations to help localize damage to the brain and plan therapies. If, on the other hand, primitive reflexes are retained beyond the normal developmental periods, they can be associated with spatial problems, motion sickness, problems with visual perceptual, poor sequencing skills, and a poor sense of time.

9.3.3 Bridge Reflexes

Bridge reflexes are a special class of reflexes that have a significant impact on sensorimotor development. Unlike the primitive reflexes (section 9.3.2), bridge reflexes are not present at birth, and in contrast to postural reflexes, which occur next in the chronology (section 9.3.4), they do not persist beyond the developmental period.

The **ulnar release** and **radial release responses** refer to reflexive support for the beginning of independent finger movements. From the palmar grasp configuration, the ulnar release begins by releasing the index finger, causing a subsequent cascade release in adjacent fingers until they are all extended. The radial release is similar, beginning on the little finger side of the fingers. These reflexes appear roughly at birth and persist until about 4 months, at which point they are integrated into other coordinated movements of the hand.

The **traction response** can be observed in two primary contexts—the proprioceptive and contactual phases. The proprioceptive phase can be elicited between 0 and 2 months of age and is characterized by a simultaneous flexion of the shoulder, elbow, wrist, and fingers

(withdrawing the whole limb) in response to pulling the shoulder into extension. At about 2 months, the infant will swipe at visual (and auditory) targets, and the shoulder extension will trigger the proprioceptive traction response. As a result, the infant will fail to make contact with the object. Between 3 and 5 months the fully integrated traction response participates in a more mature swiping motion with an open hand. Its abnormal persistence as a primitive action will interfere with voluntary reaching and grasping, object manipulation, and transport for visual exploration.

The contactual phase emerges at about 1 month of age and can be observed until about 3–5 months. A tactile stimulus on the palm between the index finger and the thumb produces an immediate flexion of the fingers around the stimulus followed by a flexion of the joints in the arm (via the traction response). The flexion of the fingers is the earliest form of the grasp reflex.

The **avoiding response** can appear about 1 month after birth and will be lost or integrated into mature manipulative behavior by about 6 months. In contrast to the deep pressure on the palm required to elicit the palmar grasp reflex, a light contact on the back of the distal phalange can cause the hand to open by extending and abducting the fingers, with a coordinated arm and wrist movement that places the hand next to the face with the palm facing forward. The palmar grasp and the avoid response overlap during the first year and can conflict, causing the infant to alternately open and close the hand next to their face, especially when excited. The avoid reflex also participates in the release of objects from the hand. A remnant of the avoid reflex can persist and be elicited in adult subjects under conditions of severe stress.

The **oculo-cephalic reflex** is sometimes called the “doll’s eye” response. It consists of a conjugate gaze stabilization in response to head movement. The vestibulo-ocular reflexes allow the eyes to remain fixed on a target even as the head moves vertically and horizontally. It emerges at roughly 2 weeks after birth and persists through 3 months.

When supported in the Landau position (prone, with examiner’s hand supporting from beneath the trunk), the infant will strike the “superman” pose (see [figure 9.13](#)). The head will extend above the plane of the trunk, with trunk and legs extended isometrically against gravity. When the infant’s head is pushed downward (into flexion), the legs reflexively drop

down into flexion as well in a response known as the **Landau reflex**. If the head is released, then the infant will resume the superman pose.

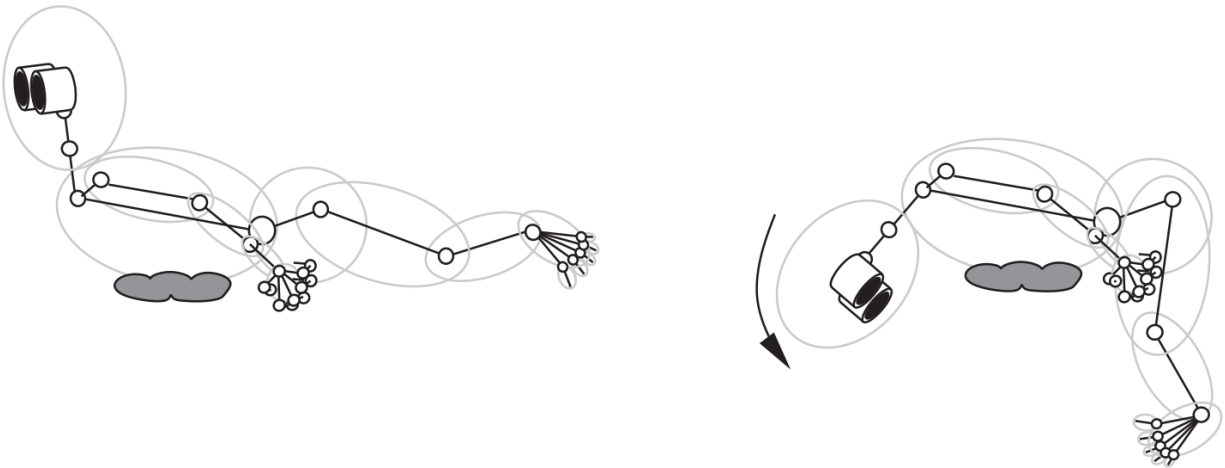


Figure 9.13
The Landau response

The Landau reflex emerges between 3 and 10 months and persists until about 36 months. It is related to labyrinthine and spinal reflexes, and it contributes to good trunk and hip muscular extension for straight sitting. It exerts an important inhibitory influence on certain postural reflexes (section 9.3.4), serves to strengthen muscles, and aids in the development of vestibulo-ocular motor skills.

The **instinctive grasp response** is a three-step motor response that accomplishes involuntary grasps. The *orienting* stage of this response emerges between 4 and 5 months. It causes the palm of the hand to turn toward a stimulation on the hand. A stimulation on the radial side of the hand supinates the hand; ulnar stimulation pronates the hand. At about 6 to 7 months of age, the *groping* stage of this response orients and then advances the palm toward the stimulus and between 8 and 10 months, the *trapping* stage in the sequence envelops the object in a grasp. Remnants of this involuntary grasp can persist into adulthood, but generally, the instinctive grasp is fully integrated by approximately 10 months of age. It is the most mature grasping reflex; subsequent manipulative control is increasingly cortical, visual, and tactile.

The **symmetric tonic neck reflex (STNR)** in [figure 9.14](#) creates a body posture with extension in the back and arms so that the upper limbs move

forward to bear weight. If the head is in flexion, the hands are brought to midline across the chest in front of the gaze. It emerges around 6–9 months after birth and is inhibited at about 9–11 months. During this period, it participates in the early stages of the development of upright postures and bipedal gaits. If retained, it can delay the development of hand-eye and swimming skills and can prevent forward progress in crawling motions by inhibiting independent limb movement.

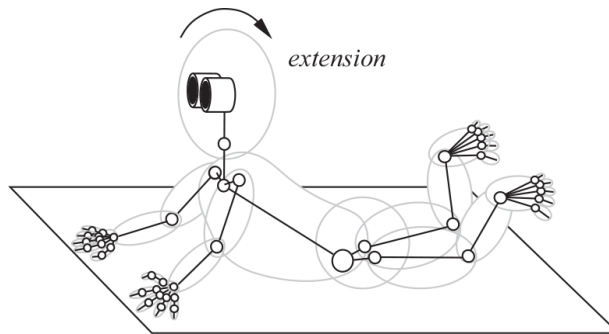


Figure 9.14

The symmetric tonic neck reflex triggered by neck extension

9.3.4 Postural Reflexes

Some primitive reflexes that use vestibular feedback contribute to the development of postural control associated with balance and stability. The vestibular systems involve the inner ear, vision, proprioception (like the muscle spindles in sections 3.1.2 and 7.1.2), and, cutaneous mechanoreceptors to provide kinesthetic feedback—a sense of spatial body posture and the relationship to the Earth through visual, inertial, and gravitational feedback. In the middle ear, for example, very specialized inertial structures act to provide a kind of accelerometry. Small crystals called otoliths (*stones*) located on cilia in the inner ear are part of specialized cells that measure the orientation of the head in the gravitational field. Much of the sensorimotor period in human infants is devoted to integrating a sense of gravity into balance and postural stability.

The **labyrinthine reflexes** are a family of primitive responses to vestibular information mediated in the brainstem and associated with the postural control of the head. They emerge a cascade of related postural reflexes. The **tonic labyrinthine reflex forward** (TLRf) emerges in utero,

is present at birth, and disappears by about 4 months. From a prone position, it causes the head to rise off the floor and is the first step toward stabilizing the head in a vertical orientation against gravity. Its cousin, the TLR backward (TLRb) emerges at birth and is gradually inhibited over the period from 6 weeks to 3 years. TLRb is involved in the development of postural reflexes that persist throughout life.

When the infant is suspended in an upright vertical orientation, tactile stimuli on the bottom of the foot triggers a muscle stretch reflex that stiffens the leg by co-contracting antagonist muscles. The lower extremities become pillar-like to support the infant's weight in a maneuver called the **positive supporting reflex**. It emerges at 3–4 months and persists throughout life. Most infants can actually support their weight by 5–6 months and bounce at 7 months. There is an analogous response in the upper limbs that interacts strongly with visual and labyrinthine righting reflexes and the Landau extension.

Sometimes called the “body-on-head” reflex, the **righting reflex** turns the shoulders and the trunk in the direction that the head is pointing. It is important for moving the neck configuration back toward its nominal, forward-looking configuration and is instrumental in the transition from prone to supine. It is present at birth and normally observable through 4 months, at which time it is integrated into voluntary movements.

Another pair of important head righting reflexes, distinguished by the use of different sensor feedback, delivers the head to a posture with nose vertical and eyes horizontal from any body posture. The **optical-head righting reflex** (OHRR) is driven by visual input. It emerges at 2–3 months of age and remains for life. It is important for balance and eye movement control. If underdeveloped, it can lead to poor visual tracking, motion sickness, and disorientation. The related **labyrinthine-head righting reflex** (LHRR) is the vestibular counterpart of OHRR. Both are essential for balance and for integrated control of visual focus. LHRR also emerges 2–3 months after birth and remains for life.

The **amphibian reflex** is a trunk movement (thought to be quite old) that supports coordinated asymmetric limb movements between the arms and legs. It emerges 4–6 months after birth and remains for life. Its absence in a clinical evaluation sometimes indicates the persistence of uninhibited primitive reflexes, particularly ATNR, and therefore can be used to

diagnose developmental abnormalities. It is essential for crawling, walking, and running.

The **segmental rolling reflex** emerges at around 6 months of age and persists throughout the rest of life. It is essential for coordinated cross-lateral movements such as walking, running, jumping, and swimming. It also helps the infant to roll over and up into a sitting position.

The **lateral propping reflex** (figure 9.15) is an analog of the positive support reflex in the lower extremities. It constitutes a protective, weight-bearing extension of an arm, essential for protective bracing and the ability to sit up unaided. This postural reflex appears at 5–7 months after birth—first in the form of anterior propping and later as lateral propping. It remains intact for the rest of the individual's lifetime.

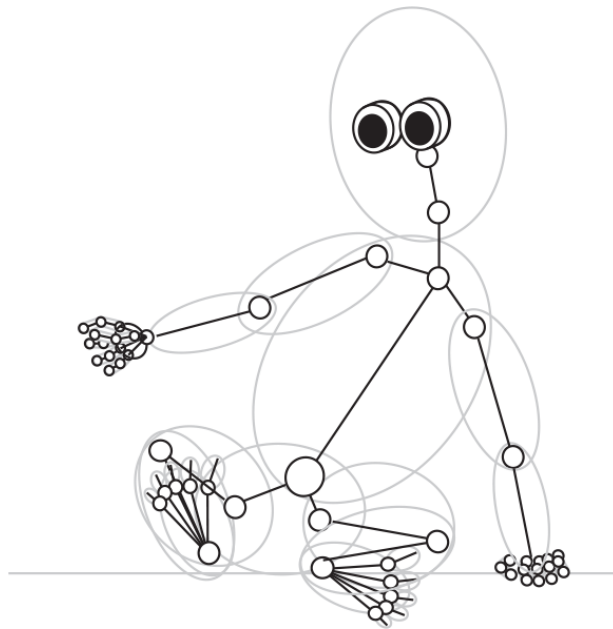


Figure 9.15

The lateral propping reflex

The **parachute reflex** (figure 9.16) emerges 8–9 months after birth and is retained for the rest of life. When the baby is turned posterior end up from the prone position, the arms extend symmetrically as if the baby is preparing to break its fall. It is a reaction to vestibular input and interacts with Moro and propping reactions to help support the sitting infant. It also aids in the transition to a seated position.

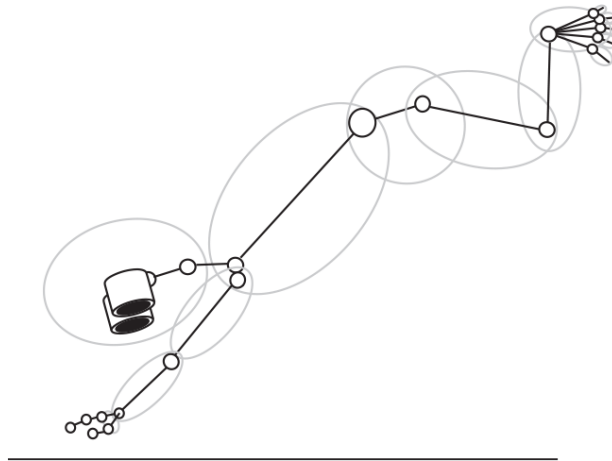


Figure 9.16

The parachute reflex is a bilateral extension to brace for a fall

9.3.5 Maturation Processes

In this section, we describe some of the early mechanisms of maturational growth that are mediated by genetic factors and hormonal influences and that determine growth and development in the nervous system. A very brief overview of the evolution and major structures of the brain is included in section 9.1.

Brain development and physical maturation—The early brain contains stem cells that replicate by neurogenesis and differentiate to form precursor cells (neuroblasts or glioblasts) for neurons or glial cells, respectively. These two types of cells constitute about half the volume of the central nervous system (CNS), even though glial cells may outnumber neurons by as much as 50:1. Still, there are approximately 1 billion neurons by 5 months gestation and about 100 billion at birth. The human brain is roughly 25 percent of adult weight at birth, and this ratio reaches about 50 percent by 6 months, 75 percent in 2 years, and roughly 90 percent by 6 years of age. The development of the central nervous system occurs in stages that overlap and interact. Neurons move from the neural tube ([figure 9.1](#)) to form different parts of the brain. This process is largely completed by birth. They differentiate to form specialized neurons and undergo synaptogenesis—a process by which neurons project fibers and synapses to neighbors and more distant neurons. The growth of new synapses creates a super connectivity and the potential for much greater intercellular communication

than will be present ultimately in the mature brain. There is a culling and rearrangement of these connections between neurons after birth. This pruning is guided by interaction and experience. Unused pathways between cells will atrophy and eventually disappear.

Glial cells are the glue in the nervous system, as the name implies, but they are more than that. They guide neuronal migration early in development, modify the growth of axons and dendrites, and play a role in synaptic plasticity and synaptogenesis. Glial cells are also active participants in synaptic transmission, regulating neurotransmitters in the synaptic cleft. They serve a supporting role for neurons—they protect and provide structural support, they supply nutrients and oxygen, they ingest dead neurons, and they form insulating myelin around neural processes. Myelin is a fatty sheath that surrounds many neural processes, a kind of insulation that changes the ability of axons to conduct signals. The effect is to change both the resistivity and capacitance of the fiber and therefore, the strength and timing of information propagated in the brain. Myelination begins in the third trimester in utero and continues rapidly through the first 2 years. It begins early for motor-sensory roots, the brainstem, and structures necessary for reflex and survival. The vestibular nerve (involved in balance) is the first cranial nerve to myelinate. This happens in utero, which suggests that balance and postural control begin to develop in utero as well. Facial nerves, related to breathing, sucking, swallowing, and coordinated movements of the lips, are the second cranial nerve to myelinate. Motor nerves myelinate before sensory nerves. By 3 months after birth, the vestibulo-ocular reflex is involved in producing conjugate eye movements in response to head motions to help track objects in the world visually and correct for egomotion. The corticospinal tract starts to myelinate at about 36 weeks after birth. It precedes from proximal to distal and from head to toe. The shortest axons myelinate first, followed by the upper extremities and trunk, and finally the lower extremities (with the longest axons) last.

Developmental mechanisms in animals are believed to contribute structure needed for overcoming worst case learning complexity. Developmental processes manage access to neural projections that engage sensory and motor resources. For example, motor learning for reaching movements proceeds from proximal-to-distal musculature—a consequence

of patterns of myelination in the motor cortex [23, 22, 24]—which has significant impact on the development of manipulation skills. The same processes are observed in the lower body musculature and bipedalism. Moreover, the sensory and motor resources, themselves, change characteristics over time. A newborn’s eyes are capable of focusing only to a range of about 21 *cm* [169]. Their immature visual organs have limited acuity as well as compromised sensitivity to brightness and color. As result, the infant’s attention is initially focused on large regions of movement and strongly contrasting features [114, 260]. Soft bones in the neonate’s inner ear respond preferentially to low frequencies in utero and during the first few weeks to avoid the full complexity and dynamic range of auditory signals. Several researchers have observed that limitations in the infant’s central and peripheral nervous system may also serve to bootstrap neonatal learning [213, 78, 288, 70]. Taken together, these characteristics of the immature infant serve to simplify initial interactions with the world so that important behavioral and cognitive substrates can develop.

Plasticity—Many aspects of brain development are determined genetically. However, other important aspects are plastic and can adapt to stimulation. Plasticity is demonstrated by the ability of some cortical regions to take on functions normally associated with other regions. The incentive to remap can arise from neural deficit or injury, as well as a lack of stimulation. For example, topographically aligned spatial mappings in the visual and auditory cortical regions will change and interact if the geometry of one or both of these sensor systems is altered [146]. The mechanisms for remapping combine maturational and experiential influences. There exists a greater capacity for plasticity early in life when neuro- and synapto genesis prepare the neonatal brain to accept information in many ways. Pathways that are stimulated repeatedly are wired up in the neural substrate, and those that are starved of stimulation atrophy and die or undergo synaptic rearrangement. This process can extend to a lesser degree throughout life as, for example, when expert skills are honed through practice and experience.

Hemispheric specialization—Brain function is distributed over both hemispheres in general, but there does appear to be some degree of hemispheric specialization. For example, handedness preference in young

children is usually established by 2 years of age and footedness by 5 years. Moreover, in 95 percent of right-handed individuals, speech is handled in the left hemisphere, whereas left hemisphere specialized speech occurs in only 70 percent of left-handed people. Much of this specialization is complete by birth, but in infants born with some form of hemispheric dysfunction a great deal of plastic compensation is possible. In dramatic seizure disorders where hemispherectomies (the removal of an entire hemisphere of the brain) are employed, young patients can achieve almost completely normal function by virtue of plasticity and neuronal rearrangement. Significant plastic rearrangement has also been demonstrated in apes [95].

Physical growth—Body growth creates changes in shape, proportion, and mass distribution, which has implications in the relative strength-to-weight ratio over time. Smaller limbs create smaller endpoint position errors in response to errors in motor inputs. Therefore, infants may be capable of reducing errors due to immature motor pathways by adapting morphologically. Genetic factors influence the rate at which growth occurs indirectly through neural and hormonal activity. Hormones produced in the hypothalamus regulate the production of human growth hormone, and chemical triggers for growth exist that can accelerate or decelerate growth in the brain and the body. During infancy and adolescence, humans can experience tremendous growth spurts. Different organs and body parts grow at different rates, preceding generally from head to foot (cephalo-caudal) and from trunk to periphery (proximo-distal).

9.4 Developmental and Functional Chronology in the First Year

There is a rich neurological organization in the infant that changes over time. [Figure 9.17](#) illustrates parts of this sequence for some of the reflexes discussed thus far in this chapter. Compositions of primitive reflexes can be used to create foundational stimulus-response arcs, including context sensitive policies for postural control, for balance and agility, and for hand-eye coordination. In general, this sequence proceeds from primitive

reflexes, through the bridge reflexes, and into postural reflexes that persist throughout life.

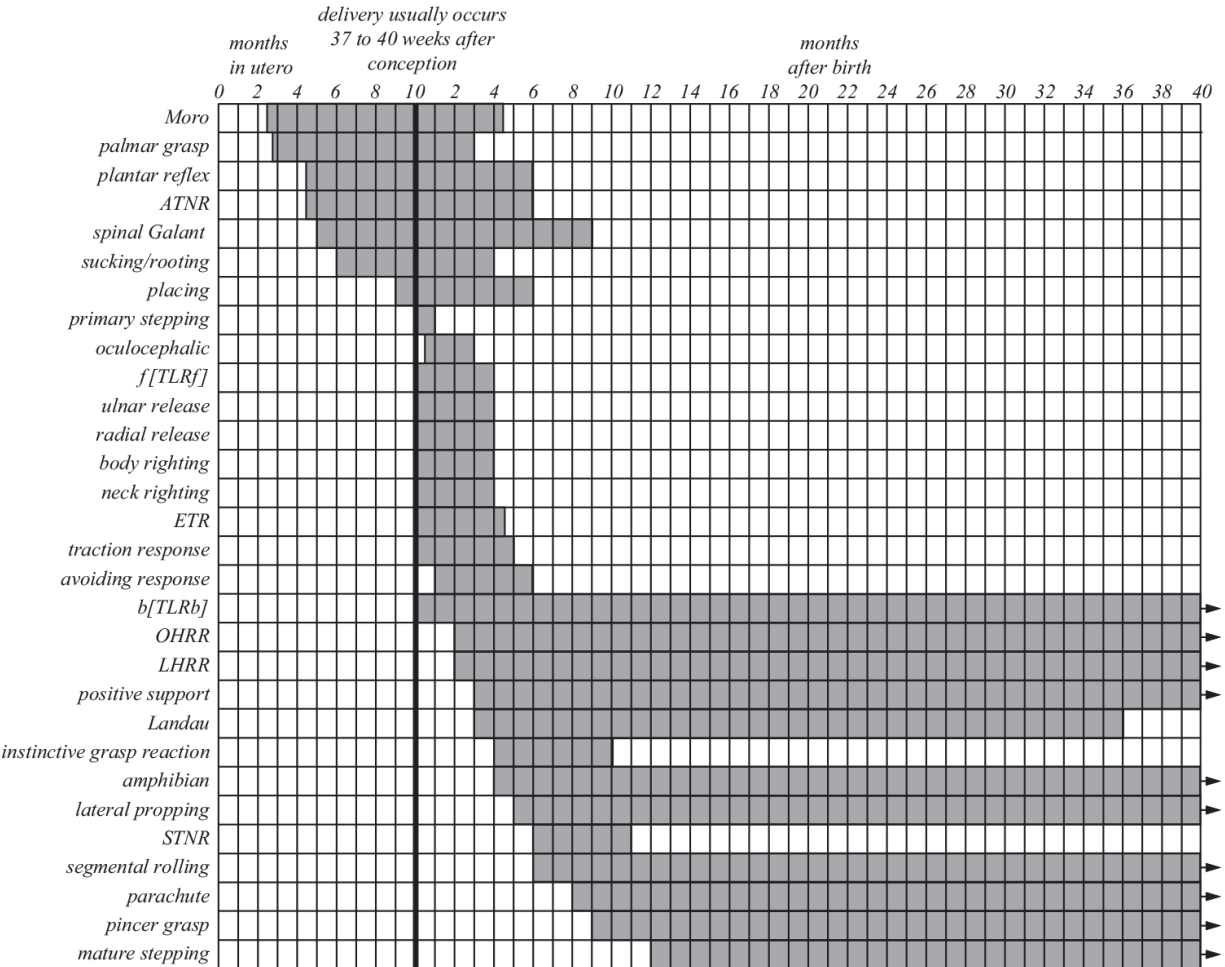


Figure 9.17
The emergence and subsequent inhibition of several developmental reflexes

The cognitive and behavioral outcome of developmental processes varies from individual to individual in a population to reflect unique experiences. Remarkably, the shared embodiment of the species—its morphology, strength, and perceptual acuity—links individuals with a diverse history of direct, situated experiences into transferable knowledge about the world.

As described in [165, 74, 87], the relationship between neurological structure and function provides a coarse model for the sensorimotor development of the infant during the first year of life. A summary of this developmental trajectory is provided in [figure 9.18](#). The right side of this

diagram shows the development of gross motor function in the context of postural stability and mobility, and the developmental subtree on the left represents fine motor skills for manual dexterity and prehension. These two domains interact in important ways during the first year of life.

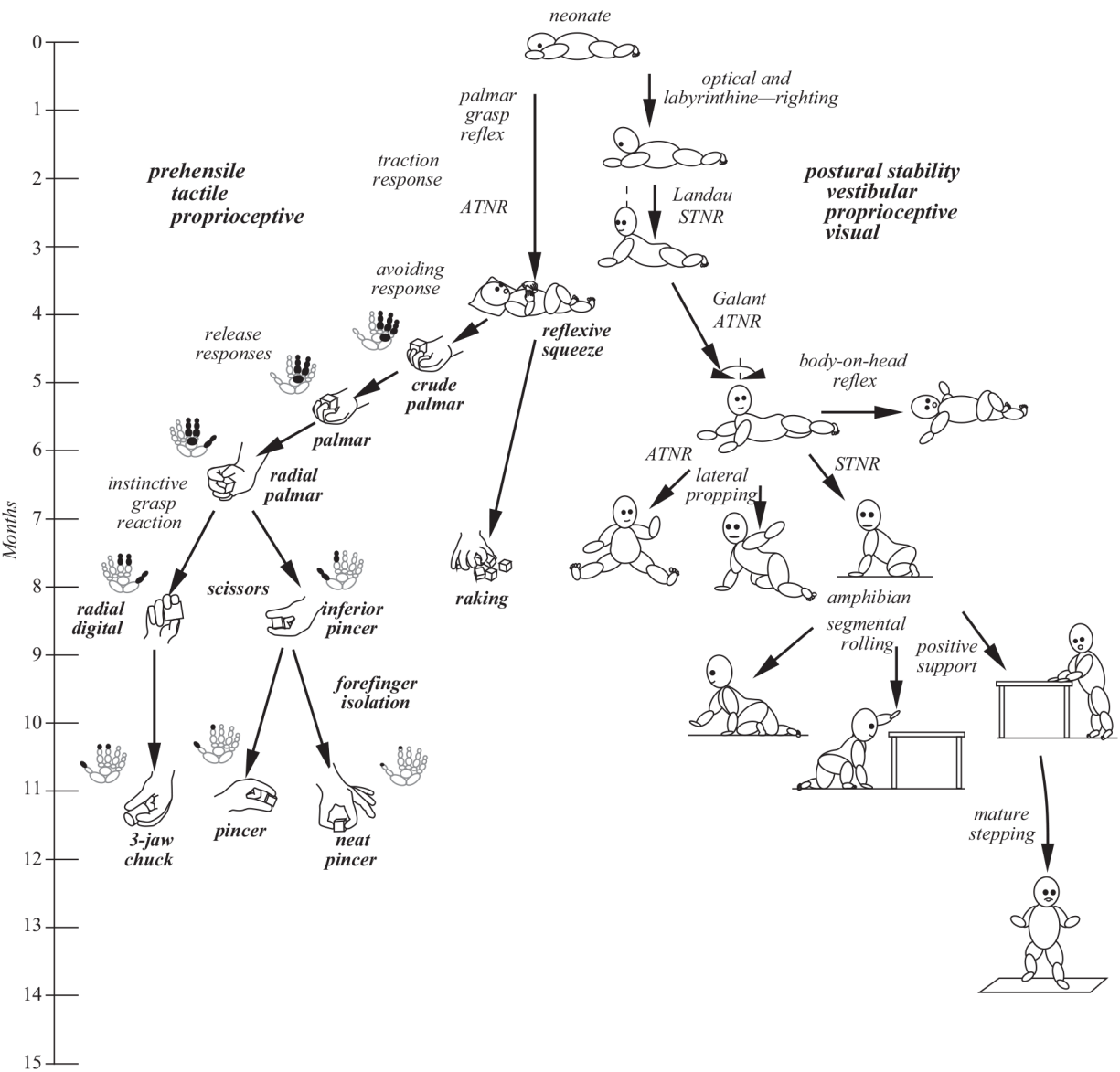


Figure 9.18
 A developmental sequence—a chronicle of some of the milestones involving postural stability and the use of the hand during the first 12 months. (Right) Gross motor development. (Left) Some of the early fine motor developmental milestones.

On the right side of figure 9.18, a sequence of postural stability tasks is illustrated. Feedback information for this subtree is primarily derived from

vestibular, proprioceptive, and (later) visual organs. Gross motor control starts with the infant acquiring the ability to control its head. At about 2 months, the child learns to raise its head off the floor (like a mast) into a configuration where the vertical axis of the head is aligned with gravity. This occurs during a period of learning about and strengthening muscles in the neck and the back that are as yet weak in extension. The infant uses optical and labyrinthine righting reflexes that develop over the first few weeks. From the prone position, these reflexes interact with the symmetric tonic neck reflex to develop a quadrupedal position by about 6 or 7 months of age. The purely proprioceptive body-on-head righting reflex helps to rotate the trunk in response to a head angle. The infant thus acquires policies for rotating the trunk and head about the body axis to pan the head and eyes. All of this behavior contributes to stabilizing the infant in sitting and later standing postures.

The developmental subtree on the left side of [figure 9.18](#) depicts milestones in prehensile behavior during the first year. The trajectory is depicted in terms of hand postures and grasp configurations, but it is more accurate to view these milestones as the consequence of acquired manual expertise along with well-coordinated hand, arm, and body controls. The palmar grasp and the traction responses, together with the avoid response, create a swiping motion with the arm and launches the palm of the open hand toward objects where a tactile stimulus will cause the palmar grasp reflex to close the fingers, pinning the object between the fingers and the palm. The asymmetric tonic neck reflex strengthens neural pathways that map visual signals to arm and hand postures. At about 6 months of age, when the infant first begins to explore voluntary postural actions that free up its hands, an explosion of developmental activity emerges related to the discovery of affordances for grasping and manipulation. The mechanisms of growth and maturation work with these reflexes to encourage cortical development—ulnar to radial, proximal to distal—that develops arm and hand controls, culminating at one extreme in grasps that engage the tips of the thumb and fingers to create a large degree of mobility in the grasped object relative to the palm.

Other developmental trajectories are taking place concurrently. Early on, when the infant succeeds in raising its head up from a prone posture, it begins to engage other people by capturing gaze and smiling. Shortly after

hands become free (around 6 to 9 months), deictic gestures (things like pointing, giving, showing, and waving) begin to develop. Near the end of the sequence in [figure 9.18](#), the child demonstrates skills for picking up objects within reach, transporting them to its mouth, putting them in cups, and stacking. Expertise in these pick-and-place tasks ignites another developmental explosion in language learning, wherein words are attached to experiences gathered using these newfound perceptual and motor skills.

9.5 Sensory and Cognitive Milestones

As infants proceed through developmental sequences, they exhibit a growing repertoire of sensory, motor, and cognitive abilities. These abilities are evaluated clinically and used to assess developmental progress. In contrast to most robot systems, sensory and motor performance in infants develops over the first year at the same time that the infant is learning how the environment *affords* information regarding actions, places, and things. Therefore, before we move onto an analytical treatment of sensorimotor systems, it is instructive to review when certain features of signals become detectable and when informative abstractions of these signals become available.

9.5.1 Sensory Performance

Touch is the fundamental means of interaction and is important for early growth. Rooting is a critical reflexive feeding response driven by tactile sensations on the cheek, and many of the reflexes discussed in section 9.3.2 are triggered in part by tactile stimuli. Mouthing is a form of tactile exploration—the mouth is heavily represented in the somatosensory cortex. Infants often put new objects into their mouths and then look at them. Mouthing behavior also supports subsequent verbal behavior. Exploratory mouthing usually stops at about 6 months of age.

Temperature is not sensed directly but is actually an observation of the heat flux into (sensation of hot) or out of (cold) the child's body and is observed by free-ended nerve fibers in the epidermal layers of the skin. Infants are generally more sensitive to cold than they are to hot.

Pain is transmitted to the brain from free-ended nerve fibers distributed around the body in the skin as well as from strain transducers such as Golgi tendon organs and articular surfaces that signal extreme positions, velocities, or ligament tension in joints. An infant responds to painful stimuli with a drop in heart rate and blood pressure.

Taste is relatively well developed at birth. Infants prefer sweet liquids to plain water and relax their faces in response to sweetness, purse their lips in response to sour sensations, grimace in response to bitter tastes, and show displeasure to salt.

Smell is related to taste. Neonates smile in response to the scent of a banana and frown to rotten eggs. They are capable of localizing smells and turn away from bad odors. Breastfed newborns can recognize their mother by smell, whereas bottle-fed babies prefer the scent of lactating to nonlactating women.

Hearing in infants is more sensitive and responsive to some sounds than to others. They prefer complex spectral content to pure tones. By 3 months, they respond reliably to the human voice and can differentiate [ba], [ga], [ma], and [na] sounds and can screen out distracting sounds, including human voices in other languages. Caregivers participate actively in auditory training by addressing infants in a manner that facilitates infant auditory development and receptive abilities for language. *Motherese* refers to the higher-pitched baby talk used by adults instinctively when interacting verbally with infants—it includes an increased average pitch, as well as exaggerated prosody, inflection, stress, and duration.

Vision is the least mature infantile sensory organ. The neonate's eye muscles, retina, and optic nerve are underdeveloped, resulting in poor overall visual acuity. The pupils are not able to dilate fully at birth, and the muscles that control the curvature of the lens are not yet fully developed, leading to a visual acuity of around 20/200 at birth. As a consequence, neonates attend preferentially to low spatial frequencies at first and respond to higher spatial frequencies as the ocular musculature matures. At 1 month, they are attentive to low bandwidth black and white imagery; between 3 and 5 months, color responses, visual attention, and attention to their own hands develop. By about 6 months of age, the normal developing infant's visual acuity reaches about 20/100, and by 2 years it is approaching adult visual performance.

Motion sensitivity and tracking develop soon after birth. For example, the blink reflex can be elicited by looming objects at about 3 weeks of age, and it is believed that infants detect kinetic depth cues and begin to exploit binocular depth cues at 2–3 months of age even though conjugate binocular eye movements are still relatively uncoordinated. Neonates track faces longer than other types of visual patterns, but they don't do as well at discerning static faces from a background of equal complexity. By 2–3 months, they respond selectively to facial pattern, recognize their mothers, and can differentiate two individuals in photographs. By 7 months, they often have acquired the ability to interpret the affective content of facial expressions and respond appropriately to happy and sad faces. Between 6 and 12 months, sensitivity to depth cues was demonstrated in the famous “visual cliff” study by Gibson and Walk [93]; during the development of crawling behavior, infants demonstrate reluctance to cross the apparent depth boundary even under the urging of their mothers.

Intermodal perception refers to the ability to associate multimodal sensory information with control decisions, places, and objects. Intermodal inference can introduce robustness when sensors have different spatial acuity or different viewpoint constraints or when the scene is only partially observable in context. The ability to form intermodal associations begins as early as 1 month of age, when infants begin to recognize the shape of something in their mouth—their pacifier, for instance. Infants will turn their head to orient vision to a sound by 4 months of age, when it is believed they also have visual expectations conditioned on what they have heard. About the same time, the infant can associate the sight and sound elicited from a wooden block dropping to strike a table and distinguish this experience from the intermodal perception of a rubber ball under the same circumstances.

9.5.2 Cognitive Development in the Sensorimotor Stage

More than simply an account of stimulus and response, a developmental sequence begins the process of acquiring a representation of experience that informs future control decisions. In developmental psychology, there are a variety of theories regarding the accumulation and use of background knowledge by the infant. For example, Piaget proposed that children begin

to associate cause and effect during interactions with the environment at about 7 or 8 years of age. By this account, before this age, a child will often mistake causation for animacy. However, more recent research concerning infants as young as 7 months old suggests that they understand how actions (cause) influence outcomes (effect) even while incorrectly attributing agency to inanimate objects.

For example, *object permanence* refers to the ability to infer the existence of a previously observed object even when it is not currently detectable in sensor feedback. It implies a representation for objects that is robust to changes in scale, shape, viewpoint and occlusion and the ability to predict how interactions with the scene change one's perceptions—make hidden objects observable, and vice versa. Spelke et al.'s [264] moving rod study suggested that infants as young as 3 months of age have acquired an implicit understanding of object permanence and a qualitative understanding of gravity. Baillargeon [10] reported similar results when 3-month-olds appeared to recognize objects over a wide variety of viewing contexts and to believe that objects would behave as rigid bodies. Moreover, young infants do seem to form prototypes for shapes and faces. Quinn et al. [237] found that 3-month-old children, who lived with cats, attended to dogs longer, implying that they found them novel and demonstrating situational awareness.

Many theories of adult social behavior incorporate a *theory of mind* that is used to infer the state of another person's mind. To describe the ability of a child to learn from parental demonstration, György Gergely [91] proposed that the capacity to imitate in young children is the product of a *teleological* process. The word teleological (Greek *teleos*, “goal; end state”) denotes a rational agent that, in Gergely's framework, extracts the *goals*, *means*, and *constraints* of a demonstrated movement or behavior. Partial support cited for this assertion include findings that infants as young as 6 months old can use a landmark to find nearby hidden objects and by 5–7 months of age, infants understand that movements of a parent's hands are often directed toward the goals of a demonstrated behavior. Related studies showed that infants start to respond to intermodal relational structure by 7 to 10 months of age [297] and form categories for birds, planes, animals, and vehicles [185, 186]. Poulin-Dubois et al. [234] reported that by 9 months of age, infants begin to demonstrate expectations about animate as well as

inanimate objects. Two-year-olds can use dead reckoning to some degree, but without straightforward landmarks, children as old as 7 years find it difficult. At 10 years of age, children can remember routes in terms of the actions needed to follow them—turn left, turn right—and they can integrate landmarks and (sub)routes, and discern spatial configurations.

In the next chapter, we explore a computational framework for describing a developmental agent that incorporates many of the motivations and mechanisms of human development introduced in this chapter. The goal is a developmental robot that explores, learns, and then exploits patterns of controllable interaction in the agent-environment system.

9.6 Exercises

1. **Primitive reflexes in human beings.** Indicate true or false and provide a brief explanation:

- (a) The pharyngeal reflex closes the vocal folds and the airway.
- (b) Sneezing due to bright light, called the photic sneeze reflex, is a somewhat rare response to very bright light.
- (c) The Moro reflex is present at birth and continues throughout a human's life.
- (d) At birth, the weight of the infant's brain is 75 percent of its adult weight and reaches full adult weight at about 5 years of age.

2. **Reflexive support for bipedalism.** Several developmental reflexes assist in the acquisition of skills related to bipedalism. Identify at least three such reflexes and explain their role in bipedal locomotion.

3. **Fine motor control.** Which reflexes can play a role in the inability of an infant to learn to hold a pen?

4. **Vestigial reflexes.** Can you identify any candidates for reflexes that were more important to our ancestors than they are, perhaps, to contemporary humans?

5. **Multi-functional reflexive behavior: ATNR.** It has been observed that the asymmetric tonic neck reflex may contribute to learning hand-eye behavior. Identify another possible role that the ATNR can support, and explain how.

6. **Population-normed developmental chronology.** Based on the following clinical observations, identify the reflexes elicited and give an estimate of the child's age.

(a) When you press your finger across the proximal finger joints (knuckles) on the palm, the infant grasps it tightly.

(b) When you turn the newborn's head to the left, the child's left arm and leg extend and the right arm and leg flex.

(c) When the infant is inverted (head down), he extends his arms and legs toward the ground.

(d) If the child's flank is stroked, the infant demonstrates no ipsilateral curving of the trunk.

(e) A newborn demonstrates body righting, but neither the primary stepping reflex nor the positive support reflex can be successfully elicited. Estimate the child's age.

(f) A child shows both ulnar release and radial release responses as well as body and neck righting. Estimate the child's age.

(g) The palmar grasp cannot be elicited any longer, but the child has not yet mastered the coordination of the index finger and thumb required for a pincer grasp. Estimate how long it will be before the child will be able to request its pacifier ("binky") verbally.

(h) A child is observed to reliably produce a parachute response and a pincer grasp, but, surprisingly, persists in producing a strong ATNR.

i. Estimate the age of the child.

ii. Identify a family of developmental reflexes (other than parachute and pincer) that may be disrupted by the unexpected persistence of ATNR.

iii. What behavioral consequences do we anticipate going forward in the rest of the unfolding gross motor developmental sequence?

(i) **Transfer to reflexive robot control.** Give at least two examples of reflexes that could be implemented for a robot (Roger) to serve the same useful/practical purposes attributed to developmental reflexes in infant humans.

(j) **Reflexive motor recruitment.** In the following scenarios, discuss the reflexive responses that could be elicited and observed, and explain the rationale behind them:

- i. There is a loud bang in your vicinity.
- ii. You accidentally look directly at the sun.
- iii. A child trips and falls.

7. **Invent your own homework.** Make a homework problem out of your favorite content in chapter 9. Write a question and a solution for it from the material in the reading. It should combine comparison, observation, short discussion, analysis, and/or calculation, and should not take more than 30 [*min*] to respond.

1. The subcortical structures (e.g., the thalamus, basal ganglia, amygdala, and hippocampus) play an important role by triaging incoming visual, auditory, and somatosensory signals and curating outgoing motor commands.
2. The term *neonate* means “newborn” and refers to infants up to 28 days after birth.
3. The term *plant* is often used like this in the context of control theory, where it refers to a controlled dynamical system.

10

A Computational Framework for Experiments in Developmental Learning

Clinical observations of newborn development demonstrate that infants use an array of prior reflexive and neurological structures to help them learn to interact with the world. Even so, for newborns (and interesting robots), exploring the breadth of possible interactions with the unstructured world around them is an enormously complex computational problem. However, we also know that processes related to biological growth (morphological, biochemical, and cognitive) collectively referred to as *maturation*, influence the newborn in profound ways during the transition from neonate to adult. Nowhere in the child's life is this more conspicuous than in the first twelve months after birth. Faced with unknowable complexity in the natural world, the infant copes using a pre-configured *developmental curriculum*—a sequence of tractable interaction contexts stored in the genome that helped this infant's ancestors adapt to their environments fast enough to survive.

The curriculum is implemented, in part, by scheduling sensory and motor resources that highlight critical skills. The developmental reflexes introduced in chapter 9 each focus on a relatively small fraction of the sensory and motor resources provided by the human body, but they train and/or serve as building blocks for more integrated skills, each of which, in turn, reveals more of the latent ability of the species and provides even more opportunity for higher-level control. In animal systems, developmental curricula change over time as new neural pathways emerge, limbs grow, muscles strengthen, sensing becomes more acute, and background control knowledge deepens.

In this chapter, we introduce a computational framework for experimental studies in developmental learning that can be implemented using robotic systems. Much of the motivation for such a framework comes from the discussion of the neurodevelopmental structure in human infants presented in chapter 9. In particular, we wish to consider how maturational schedules for allocating sensory and motor resources can be used to efficiently transform embodiment (the intent of the robot designer) into reusable hierarchical skills to form the foundation for higher-level cognitive structures. To this end, a combinatoric basis for control actions and feedback state is introduced using a parametric *landscape of attractors* [127] as a proxy for the native reflexes observed in infant behavior. A *taxonomy* of actions is proposed to provide structure for learning from exploration and to introduce a means of using multimodal associations in memory as probabilistic references for prospective control. Skills are constructed using sequential combinations of these enhanced *multi-objective* reflexes.

10.1 Parametric Closed-Loop Reflexes

The foundation of newborn behavior in many animals is a layer of segmental and developmental reflexes that combine sensory and motor resources into closed-loop interactions with the world. Our computational model of robot development also starts at the level of low-dimensional reflexes as well. The *control basis* framework is used to provide a discrete, combinatorial basis for an exhaustive set of primitive closed-loop controllers—every primitive closed-loop relation possible, in fact—given a description of the robot’s sensory and motor resources [117]. A **primitive** closed-loop linear control relationship with the world is specified, $\phi|_{\tau}^{\sigma} \in \{\Phi \times 2^{\Sigma} \times 2^{\mathcal{T}}\}$, where Φ is a set of potential functions, 2^{Σ} is the power set of feedback features, and $2^{\mathcal{T}}$ is the power set of independent motor units. Each of these primitives optimizes a particular relationship between the robot’s sensor geometry and internal and external sources of stimuli.

10.1.1 Potential Functions: ϕ

Scalar potential functions describe how energy is stored in the geometry of an elastic, electrical, or gravitational field. The value of a scalar potential field at the location of a particle in the field represents the energy that will be liberated if the particle is released from this configuration. The gradient of the potential field, therefore, defines a force acting on the particle.

The spring in the spring-mass-damper system in chapter 3 is an example of a system that stores energy in the deformed system geometry. In this case, the potential function is the cumulative energy stored in the elastic deformation of the spring,

$$\phi_K = \int \mathbf{F} ds = \int_0^x (Kx) dx = \frac{1}{2} Kx^2. \quad (10.1)$$

This energy is released when the spring is allowed to assume its original shape. [Equation 10.1](#) describes a simple bowl-shaped quadratic that is everywhere positive except at the origin—the only place where it is zero. Furthermore, the gradient of ϕ at any position in the field is a force acting on the mass that accelerates the system toward its undeformed state at $x=0$,

$$\mathbf{F}_K = -\nabla\phi = -(Kx) \hat{x}, \quad (10.2)$$

which is known as Hooke’s law. The particle in a quadratic potential field is asymptotically stable if there is a dissipative element in the system, as in the PD controller described in section 3.3. Because they are computationally efficient and relatively simple to implement they have inspired many researchers to propose a number of potential field-based motion controllers (e.g., [[154](#), [143](#), [144](#), [207](#), [181](#), [213](#), [150](#), [244](#), [15](#), [7](#), [166](#), [167](#)]).

Potential fields also have support in biological theories of adaptive motor control. For example, equilibrium point theory proposes that combining a discrete set of force fields—that independently generate equilibria—with the mechanical properties of muscle yield a means of generating continuous movements to continuous equilibrium postures [[30](#), [205](#)]. These ideas have been used to model the motor behavior in a frog’s leg as the superposition of force fields [[94](#)] and, in this chapter, serve as a modular basis for implementing purposeful movements [[203](#), [205](#)].

The conditions that preclude local minima in asymptotically stable systems have a useful geometrical interpretation—in particular, they depend

fundamentally on the *shape* of the potential function captured in its *differential geometry*. For a multivariate potential function $\phi(q_1, \dots, q_n)$ defined on domain $\mathbf{q} \in \mathbb{R}^n$, the *critical points* of the potential function are places where the gradient of $\phi(\mathbf{q})$ vanishes,

$$\nabla\phi = \left[\frac{\partial\phi}{\partial q_0} \quad \frac{\partial\phi}{\partial q_1} \quad \dots \quad \frac{\partial\phi}{\partial q_n} \right] = \mathbf{0}.$$

The image, under ϕ , of critical points are called the critical values of the field. The stability of a critical point depends on the second-order (curvature) information at the critical point as well. The general description of curvature for a multivariable function is the Hessian matrix that represents the second derivatives of the function, $d^2\phi/d\mathbf{q}^2$.

$$\frac{d^2\phi}{d\mathbf{q}^2} = \begin{bmatrix} \frac{\partial^2\phi}{\partial q_1^2} & \frac{\partial^2\phi}{\partial q_1\partial q_2} & \dots & \frac{\partial^2\phi}{\partial q_1\partial q_n} \\ \frac{\partial^2\phi}{\partial q_2\partial q_1} & \frac{\partial^2\phi}{\partial q_2^2} & \dots & \frac{\partial^2\phi}{\partial q_2\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\phi}{\partial q_n\partial q_1} & \frac{\partial^2\phi}{\partial q_n\partial q_2} & \dots & \frac{\partial^2\phi}{\partial q_n^2} \end{bmatrix} \quad (10.3)$$

Equation 10.3 captures the compound curvature of potential function $\phi(\mathbf{q})$. A critical point is *degenerate* if it also has zero curvature—that is, the Hessian is identically zero. In this case, the neighborhood around the critical point forms a local plateau where the state of the system is not controllable.

Figure 10.1 illustrates three types of controllable neighborhoods surrounding a critical point in a two-dimensional domain that can be distinguished by properties of the Hessian at the critical point. The type 0 critical point on the left is distinguished by a positive definite Hessian (appendix A.3). As a result, ϕ is *convex* in this neighborhood and gradient descent on $\phi(\mathbf{q})$ will converge exclusively to a unique type 0 minimum where it will suppress bounded perturbations. Type 0 critical points, therefore, constitute stable fixed points in the field.

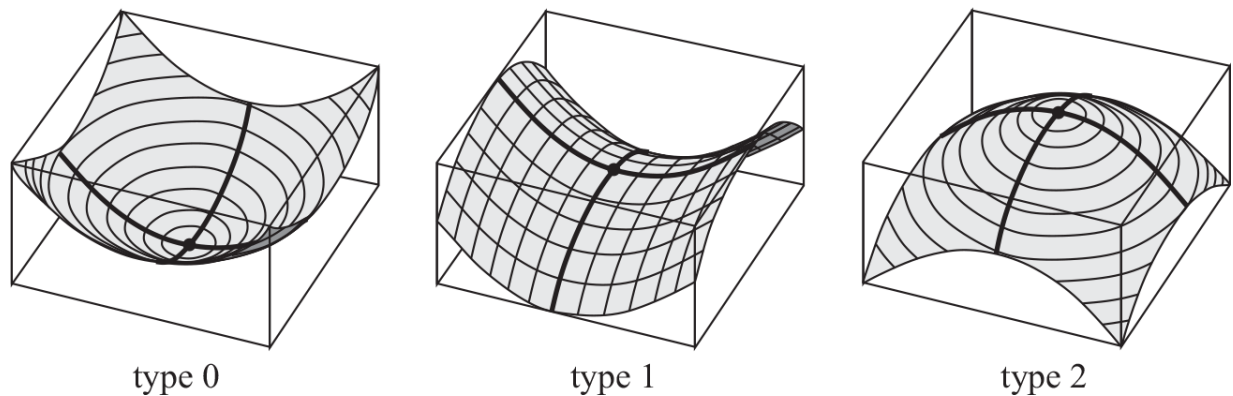


Figure 10.1
Critical points in potential function ϕ , distinguished by the value of the *curvature* of ϕ at the critical point

Type 1 critical points have positive curvature in some directions and negative curvature in others, corresponding to the *saddle point* in the middle of [figure 10.1](#). Finally, type 2 critical points have negative curvature in all directions (the Hessian is negative definite). Therefore, type 2 critical points are *maxima* in potential ϕ . In the context of gradient descent, both type 1 and type 2 critical points are unstable because, even when the gradient vanishes, minute perturbations to q_1 and/or q_2 result in nonzero gradients that cause the system configuration to depart from the critical point. In measurement theoretic terms, the region where the gradient vanishes is a set of measure zero.

Harmonic functions—The greedy character of the quadratic ϕ can be preserved in systems with constraint boundaries (obstacles) internal to domain q . Conceptually, the convex ϕ can be cleaved so that the gradient flows around obstacles instead of through them. A class of transformations called conformal deformations—functions that satisfy Laplace’s equation—can accomplish this goal without introducing local minima. In n dimensions, Laplace’s equation is written

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial q_1^2} + \frac{\partial^2\phi}{\partial q_2^2} + \dots + \frac{\partial^2\phi}{\partial q_n^2} = 0 \tag{10.4}$$

and requires that the trace of the Hessian (the sum of the diagonal elements in [equation 10.3](#)) is zero. These functions describe many natural physical processes that rely on minimum energy configurations such as the shape of

soap films, laminar fluid flow, and steady-state temperature and voltage distributions in conductive media. Appendix C introduces efficient numerical methods for solving [equation 10.4](#) for minimal surfaces in the interior of a closed domain subject to boundary constraints.

Informally, solutions for Laplace’s equation are smooth functions that distribute curvature evenly over the interior of the domain. The gradient flow produces nonintersecting *streamlines* that flow away from obstacles, exclude local minima (and maxima) in the interior of the domain, and terminate at goals. [Equation 10.4](#) demonstrates why this must be true: in the interior, where Laplace’s constraint is satisfied, if any of the terms on the right-hand side of [equation 10.4](#) is positive, then there must be other terms that are negative. Therefore, only type 1 critical points (saddle points) may exist on the interior of the domain and, except for sets of measure zero at critical points, for every coordinate \mathbf{q} in the interior there must always be a direction that is downhill on the potential function.

If the potential at obstacles is fixed to a value of 1.0 and the potential at goals is fixed at 0.0, then the harmonic potential in the interior of the domain is formally equivalent to the *hitting probability* of a random walk—the probability $p(\mathbf{q})$ that a random walk beginning at interior coordinate \mathbf{q} will encounter an obstacle before it reaches a goal [61]. Gradient descent on the scalar harmonic potential field, therefore, minimizes the chances of a collision with known obstacles. In cases when only partial knowledge of obstacles is available, this approach yields conservative strategies that can be refined efficiently as new information about obstacles (and goals) is acquired. Many examples of using harmonic functions for path control can be found in the robotics literature [62, 2, 112, 271, 145, 63, 64, 61, 128, 295].

Navigation functions—Rimon, Whitcomb, and Koditschek showed that, in certain domains, there always exist potential functions on which gradient descent achieves goals for almost all initial states (all but for a set of measure zero). In a series of papers, they described the properties of a potential function that support its use as a control function [244, 243, 292]. They call such functions *navigation functions*. Such a function should have four properties.

1. **Analyticity:** An analytic function, $\phi(\mathbf{q})$, is infinitely differentiable (C^∞ continuous) such that its Taylor series about \mathbf{q}_0 converges to $\phi(\mathbf{q})$ for \mathbf{q} close enough to \mathbf{q}_0 .
2. **Polar:** Polar functions produce gradients (streamlines) that terminate at a unique minimum.
3. **Morse:** Isolevel curves for Morse functions can only be single points, closed curves, or closed curves that join at critical points—that is, Morse functions cannot include degenerate critical points where the slope and the curvature are simultaneously zero leading to plateaus in ϕ (i.e., stationary points) where the system could come to rest short of the goal.
4. **Admissibility:** Potential fields for robot control require bounded torque at obstacle boundaries (and everywhere else in the interior subset of configuration space).

10.1.2 Closed-Loop Actions: $\phi|_\tau^\sigma$

Control primitives suppress errors by following the gradient of the scalar potential function $\phi(\sigma)$ to generate a series of references $\mathbf{u}_\tau \in \mathbb{R}^n$ to n motor units. The local sensitivity of the potential to changes in the motor variables is written in terms of the control Jacobian.

$$\mathbf{J}_c = \frac{d\phi(\sigma)}{d\mathbf{u}_\tau} = \left[\frac{\partial\phi(\sigma)}{\partial u_1} \quad \frac{\partial\phi(\sigma)}{\partial u_2} \quad \dots \quad \frac{\partial\phi(\sigma)}{\partial u_n} \right]_{1 \times n} \quad (10.5)$$

The subscript c is used in [equation 10.5](#) to designate a *control* Jacobian derived from the partial derivatives of a scalar potential function and to distinguish it from a manipulator Jacobian, which will be written \mathbf{J} (without a subscript). In the following, the subscript will sometimes be used to identify the potential from which the control Jacobian is derived.

In the neighborhood of the current kinematic configuration, \mathbf{J}_c defines a one-dimensional subset of \mathbb{R}^n (the gradient) and an $(n-1)$ -dimensional nullspace in which the potential remains approximately constant. The gradient of $\phi(\sigma)$ is used to optimize the objective function by generating new setpoints for the underlying motor units,

$$\begin{aligned}\Delta \mathbf{u}_\tau &= \kappa \mathbf{J}_c^\# (\phi_{ref} - \phi(\sigma)), \quad \text{and if } \phi_{ref} = 0, \\ &= -\kappa \mathbf{J}_c^\# \phi(\sigma),\end{aligned}\tag{10.6}$$

where κ is a small positive step size and $\mathbf{J}_c^\#$ is the Moore-Penrose right pseudoinverse of \mathbf{J}_c (appendix A.9). The right pseudoinverse yields an exact solution to [equation 10.6](#) that minimizes $\|\Delta \mathbf{u}_\tau\|_2$.

10.1.3 A Taxonomy of Parametric Actions

As proposed in the opening for section 10.1, a liberal upper bound on the number of closed-loop actions in the control basis is the size of set $A = \{\Phi \times 2^\Sigma \times 2^\mathcal{T}\}$, where 2^Σ and $2^\mathcal{T}$ are the power set of sensor and motor resources, respectively. It follows that even relatively simple robots and environments can produce an enormous space in which to search for a solution to a task. Without the benefit of additional structure, the control basis framework will overwhelm state-of-the-art planning and machine learning algorithms.

However, clinical observations suggest that neonates use a comparatively small set of primitive segmental, intersegmental, and developmental reflexes organized neurologically by maturational processes. In the sections that follow, we discuss how logical and developmental organization can be realized in the control basis for autonomous learning in robot systems as well. We begin by introducing a taxonomy of actions that partition the set of actions A into meaningful subsets based on the source of the feedback σ these actions employ.

POSTURAL Actions

Posture variation is a means through which motion and strength characteristics of the arm (are) made compatible with the task.

—S. L. Chiu [49]

POSTURAL actions $\phi|_\tau^\sigma$ respond to proprioceptive feedback $\sigma \in 2^{\Sigma_p} \subset 2^\Sigma$ in order to optimize the kinematic condition of the robot and avoid kinematic constraints. In this case, objective functions $\phi(\sigma)$ are functions of the configuration variables \mathbf{q} for kinematic (sub)chains, such that gradient descent $d\phi/d\mathbf{u}_\tau$ performs a step-wise greedy optimization of the objective function. Task independent POSTURAL primitives create movements that

improve perceptual acuity, enhance motor performance, and avoid range of motion constraints.

Several forms of POSTURAL actions have been cited in the literature as a means of exploiting the aptitudes of the robot body (see chapter 4 and appendix A.8). Scalar conditioning metrics have been used as design tools for relating manipulator geometry to a task domain [251] and as scalar objective functions for optimizing inverse kinematic solutions [50, 147, 171, 172, 209, 296, 299].

Example: Manipulability Reflex

Dexterous hands are sensory organs as well as effectors that adapt movements in response to tactile feedback. Often, there is significant uncertainty in the position and orientation of environmental surfaces with which the hand comes into contact. In these situations, the exploring hand is supported by configurations that produce forces and velocities equally well in all directions in response to contact feedback.

In [98], isotropic conditioning was proposed as an independent objective for motor control that can be integrated into a variety of tasks and situations—especially those that support haptic exploration and that depend on mobility of the grasped object. A postural reflex was described for the Utah/MIT dexterous hand in figure 10.2. Each finger of this hand has four degrees of freedom. The knuckle joint approximates adduction/abduction in the human finger about an axis that is perpendicular to the distal finger joints. The last three degrees of freedom (the phalanges and interphalangeal joints) are redundant in the plane they share, so there are a possibly infinite number of joint angle configurations for each reachable (x, y) coordinate in this plane (see figure 4.24).

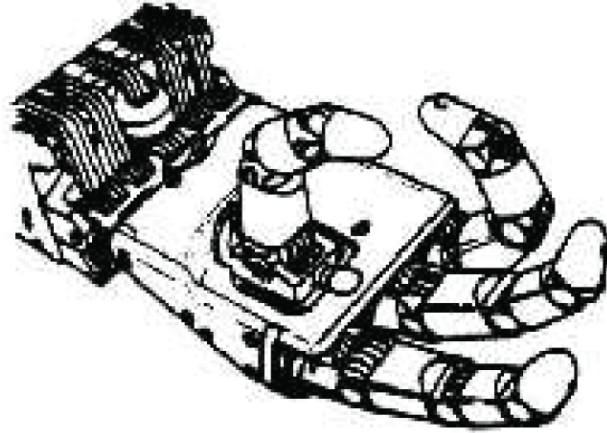


Figure 10.2

The Utah/MIT dexterous hand

A control Jacobian for use as a manipulability reflex is defined

$$\mathbf{J}_m = \frac{d\phi_m}{d\mathbf{u}_\tau} = \frac{d\left(-\sqrt{\det(\mathbf{J}\mathbf{J}^T)}\right)}{d\boldsymbol{\theta}},$$

where \mathbf{J} is the manipulator Jacobian for the last three joints of the finger. The control Jacobian \mathbf{J}_m represents the gradient of the manipulability field with respect to these joint angles and the corresponding POSTURAL reflex becomes

$$\Delta\boldsymbol{\theta} = -\kappa \mathbf{J}_m^\# \sqrt{\det(\mathbf{J}\mathbf{J}^T)}. \quad (10.7)$$

Reflexes like this map joint angle configurations in σ into differential joint angle movements in \mathbf{u}_τ .

Figure 10.3 shows the manipulability potential in the plane defined by the last three links of this finger at each location in the reachable workspace (ignoring joint range limits). Level curves for the function are in the left panel and the function itself, with an overlaid schematic of the finger and its reachable workspace, is in the right panel. Inside the reachable workspace of the finger, several fingertip trajectories resulting from equation 10.7 are illustrated. The trajectories begin at locally optimal manipulability configurations near the boundaries of the workspace and terminate at the finger posture indicated in the diagram.

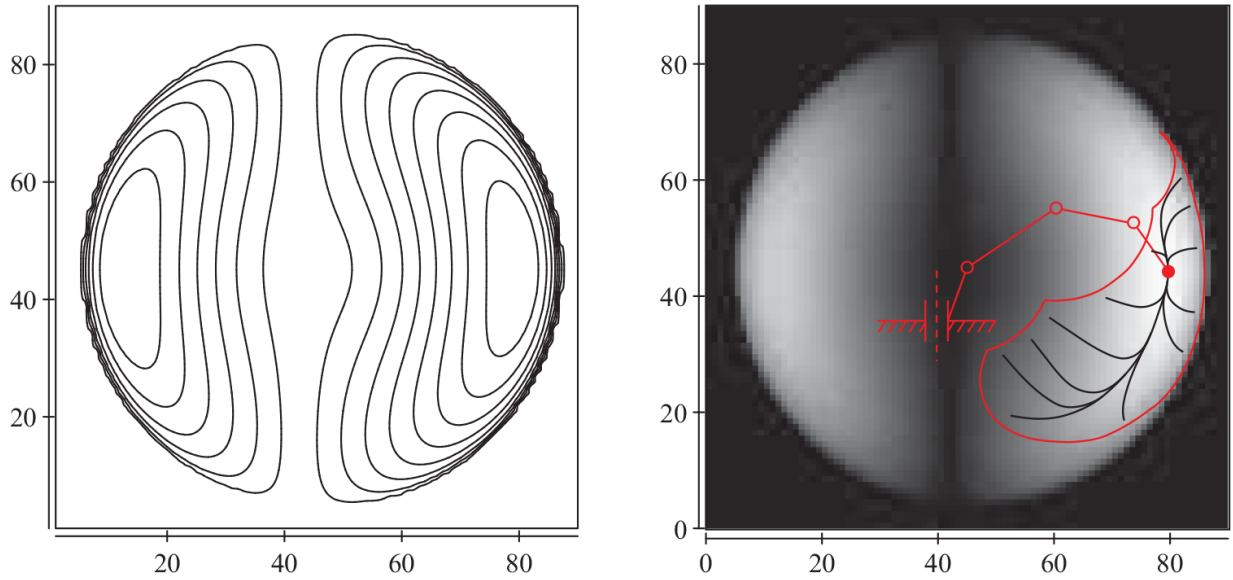


Figure 10.3

Maximum manipulability model for the Utah/MIT finger. (Left) Level curves. (Right) Scalar field $\phi(x,y) = \max_{\theta} \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ and streamlines inside the reachable workspace generated by equation 10.7.

□

New instances of related POSTURAL actions are created by configuring different combinations of proprioceptive sensor and effector resources from a common series or parallel kinematic chain. For example, in [98], a manipulability reflex was demonstrated to condition the kinematic configuration of the multifingered Utah/MIT dexterous hand in contact with a stationary workpiece by reposturing the corresponding multiple degree of freedom arm. Other candidates for POSTURAL reflexes have been used to optimize Cartesian precision or amplification (section 4.7.2), visual acuity (section 4.7.3), and acceleration (section 6.3.3). These metric fields are completely determined by the geometry and mass of the limb and capture the kinodynamic sweet spots of the embodied system [107].

TRACK Actions—TRACK actions are closed-loop controllers $\phi|_{\tau}^{\sigma}$ that employ feedback features $\sigma \in 2^{\Sigma_{ext}} \subset 2^{\Sigma}$ derived from external stimuli. These features are used to identify and localize goals and/or determine errors that are eliminated by following the gradient of the potential function $\mathbf{J}_c = \partial\phi/\partial\mathbf{u}_{\tau}$ downhill as in equations 10.5 and 10.6. Tethering the control state to features observed in the environment means that the feedback status of

TRACK actions establish concrete, observable *facts* about the run-time environment.

SEARCH Actions— SEARCH actions sample goals from probabilistic models stored in memory describing the empirical transition dynamics of actions in situ. If S is the set of all interaction states detectable by the sensors, then the control basis framework designates SEARCH actions $\phi|_{\tilde{\tau}}$ that sample references $\tilde{\sigma}$ from distributions of the form $Pr(\mathbf{u}_{\tau}|s, (s' \in P(s)))$, where $s \in S$ is the current multimodal state and \mathbf{u}_{τ} defines new setpoints for effectors τ that are likely to generate outcome states s' that belong to a (sub)goal partition $P(S)$ of S .

Initially, the distributions that guide SEARCH actions are uniformly distributed over values of the effector variables, and they are updated over the course of many interactions to reflect the cumulative, long-term statistics of the run-time environment.

SEARCH $\phi|_{\tilde{\tau}}$ and TRACK actions $\phi|_{\tau}^{\sigma}$ are natural counterparts. Together they support an enhanced SEARCHTRACK-er that uses background knowledge to find hidden signals in a complex, three-dimensional dataset and then regulates the value of those signals. For example, to find a coffee cup on a tabletop, SEARCH actions can be used to visually inspect places on the table where the cup has been found before. If this process is successful and the cup is successfully detected and localized, other SEARCH actions can predict where to move to and how to reach to produce signals from the cup on fingertip tactile sensors. The sequence of successful SEARCH actions have modified the sensor geometry to uncover new information in the world. The new information can guide additional SEARCH or be used to support a family of TRACK actions to, for example, actively regulate force closure while transporting the cup.

Integrated skills constructed from combinations of multiple POSTURAL, SEARCH, and TRACK actions can serve collectively as a SEARCH prefix that funnels the state of the system into an initiation set from which a new target skill can be learned. Such a framework relies on vertical transfer to integrate and reuse implicit knowledge embedded in skills.

10.1.4 Co-Articulation: Multi-Objective Control

In practice, multiple actions that use disjoint sets of effector variables \mathbf{u}_τ can superimpose commands to the motor units and rely on them to suppress disturbances due to unmodeled inertial or objective coupling. A more principled approach, however, even when sharing effector resources over multiple objectives, uses a null space composition to protect actions from destructive interactions.

In the linear transformation $\Delta\phi = \mathbf{J}_c \Delta\mathbf{u}_\tau$, the redundant control Jacobian $\mathbf{J}_c \in \mathbb{R}^{1 \times n}$ defines a one-dimensional subset of \mathbb{R}^n (the gradient of ϕ with respect to \mathbf{u}_τ) and an $(n-1)$ -dimensional subspace that is orthogonal to the gradient. In appendix A.9, the *annihilator* of the linear control Jacobian \mathbf{J}_c is presented in the form of a linear operator $[\mathbf{I}_n - \mathbf{J}_c^\# \mathbf{J}_c]$ that defines the *nullspace* of \mathbf{J}_c , where \mathbf{I}_n is the $n \times n$ identity matrix. Therefore, combinations of subordinate control actions can be projected into the nullspace of \mathbf{J}_c that produce no change in the value of the superior control potential ϕ .

We employ the notation proposed by Huber et al. [121, 118, 117]: if $c_1 = \phi_1|_\tau^{\sigma_1}$ and $c_2 = \phi_2|_\tau^{\sigma_2}$ are control basis primitives that employ the same effector resources, τ , then $c_2 \prec c_1$ designates that c_2 is subordinate to (left of) c_1 , and, therefore, changes to references \mathbf{u}_τ due to c_2 must be projected into the nullspace of c_1 :

$$\begin{aligned} \Delta\mathbf{u}_\tau &= \mathbf{J}_1^\# \Delta\phi_1(\sigma_1) + \mathcal{N}_1 \left(\mathbf{J}_2^\# \Delta\phi_2(\sigma_2) \right), \\ &= \mathbf{J}_1^\# \Delta\phi_1(\sigma_1) + [\mathbf{I} - \mathbf{J}_1^\# \mathbf{J}_1] \left(\mathbf{J}_2^\# \Delta\phi_2(\sigma_2) \right), \end{aligned} \quad (10.8)$$

where the quantity $\mathcal{N}_1 = [\mathbf{I} - \mathbf{J}_1^\# \mathbf{J}_1]$ represents the locally linear nullspace of controller c_1 . The operator \prec is called the *subject-to* operator, and the multi-objective controller $c_2 \prec c_1$ is read “ c_2 subject-to c_1 .”

Figure 10.4 shows how the potential function ϕ_1 defines a one-dimensional gradient and an orthogonal, $(n-1)$ -dimensional nullspace \mathcal{N}_1 , depicted here as the tangent to the local level set of function ϕ_1 . Displacements of the particle in \mathcal{N}_1 have no influence on the value of ϕ_1 .

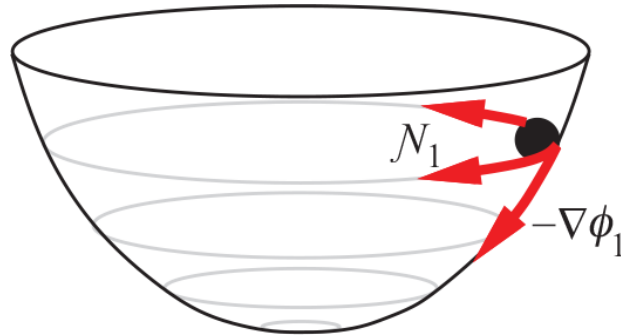


Figure 10.4

A geometrical interpretation of the gradient and nullspace of artificial potential ϕ_1

The projection of actions from c_2 into \mathcal{N}_1 guarantees that c_2 will not destructively interfere with c_1 (i.e., it preserves the asymptotic properties of c_1). Equation 10.8 is a conservative projection when all actions are derived from independent potential fields [174]. A version of this relation that optimizes progress on all subordinate objectives can be found in [208].

In sufficiently redundant systems, the nullspace projection technique can be applied to an arbitrary number of cascaded primitive control laws and can be generalized to controllers that engage different effector resources (appendix A.9). Figure 10.5 shows how such an expression involving three independent controllers, $c_3 \triangleleft c_2 \triangleleft c_1$, compiles into cascaded nullspace projections. This control expression is rewritten vertically with subordinate controllers above superior controllers. A system like this composes a multi-objective control reference and submits it to underlying motor units.

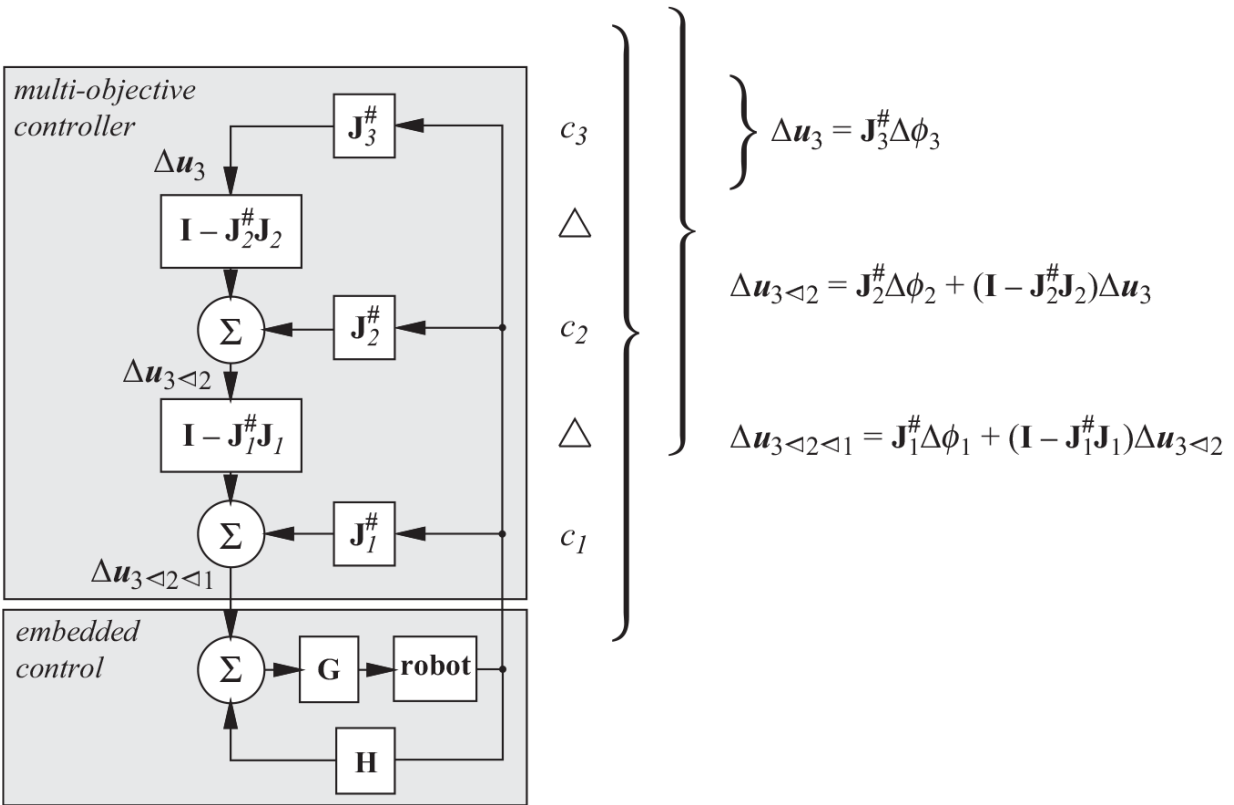


Figure 10.5
 Cascaded nullspace projections create the $c_3 \prec c_2 \prec c_1$ multi-objective controller

Example: The Mechanics of Human Finger Movement

The manipulator in [figure 10.6](#) is similar to a human finger absent the abduction (out-of-plane) movements of the knuckle joint. Flexing and extending the interphalangeal joints ($\theta_1, \theta_2, \theta_3$) causes movement of the fingertip within the reachable subset of the x - y plane. Each example in [figure 10.6](#) shows a trajectory between the initial configuration (light gray) and the final joint angle configuration (bold black), where the endpoint position of the manipulator is $(x, y)_{ref}$.

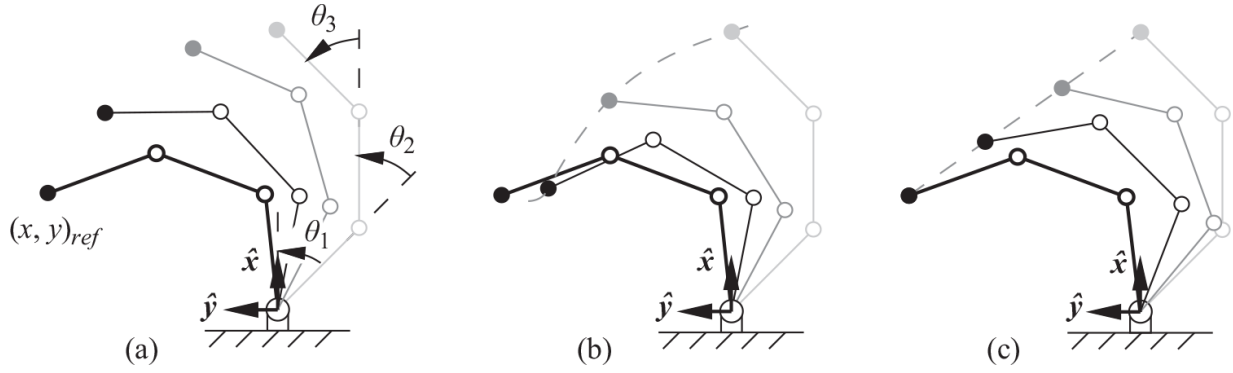


Figure 10.6

Three distinct control responses for the same initial and final configurations: (a) the PD control (second-order) arm response; (b) the steepest descent in configuration space; and (c) the steepest descent in Cartesian space

Panel (a) shows the response when the final reference posture is submitted to underlying motor units executing independent PD control laws for joints θ_1 , θ_2 , and θ_3 . The resulting finger trajectory is determined entirely by the second-order dynamics of the controlled limb.

Panels (b) and (c) show the response of two control basis primitives that each submit a sequence of θ reference configurations to underlying motor units and, thus, control the path of the manipulator as it descends the gradient of the Cartesian potential function

$$\phi_{xy} = \mathbf{r}^T \mathbf{r} \quad , \quad \text{where } \mathbf{r} = \begin{bmatrix} (x_{ref} - x_{act}) \\ (y_{ref} - y_{act}) \end{bmatrix}. \quad (10.9)$$

The case depicted in panel (b) uses the gradient

$$\mathbf{J}_{xy} = \frac{d\phi_{xy}}{d\mathbf{r}} \frac{d\mathbf{r}}{d\boldsymbol{\theta}} = \frac{d\phi_{xy}}{d\boldsymbol{\theta}}, \quad \text{so that } \Delta\boldsymbol{\theta} = -\kappa [\mathbf{J}_{xy}]^{\#} \phi_{xy}(\mathbf{r}). \quad (10.10)$$

In this underdetermined (redundant) case, the pseudoinverse of \mathbf{J}_{xy} produces an exact solution for the target endpoint position that minimizes $\|\Delta\boldsymbol{\theta}\|_2$ —it does not follow the direction of steepest descent of the Cartesian potential (equation 10.9). The difference between the actual trajectory and the straight line path in the task space can be significant.

To accurately descend the steepest gradient in the Cartesian potential, the control action is redesigned

$$\Delta\boldsymbol{\theta} = \left[\frac{d\mathbf{r}}{d\boldsymbol{\theta}} \right]^{\#} \left[-\kappa \left[\frac{d\phi_{xy}}{d\mathbf{r}} \right]^{\#} \phi_{xy}(\mathbf{r}) \right] \approx \left[-\kappa \mathbf{J}_{xy}^{\#} \phi_{xy}(\mathbf{r}) \right], \quad \text{so that } \mathbf{J}_{xy}^{\#} = \left[\frac{d\mathbf{r}}{d\boldsymbol{\theta}} \right]^{\#} \left[\frac{d\phi_{xy}}{d\mathbf{r}} \right]^{\#}. \quad (10.11)$$

This control configuration also produces an exact solution, but the pseudoinverse of the control Jacobian for this design minimizes $\|\Delta\mathbf{r}\|_2$, resulting in the trajectory in [figure 10.6](#), panel (c).

In the human hand, trajectories of the finger reflect the integrated biomechanical structure of the hand—the way that muscles in the forearm couple to produce movement in the hand and how tendons route past proximal joints to terminate on finger bones. These mechanical adaptations tacitly support the movements of the hand in its multiple roles as sensor and effector to respond to unexpected contacts and surface geometries and to enhance the mobility and stability of grasped objects.

In robot systems, the tacit support provided by the integrated biomechanical structure can be approximated in the multi-objective control framework. Yoshikawa [298] observed that when manipulating small objects, the posture of human fingers resembled those that optimized the scalar *manipulability* metric field $m = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ (section 10.1.3), where \mathbf{J} is the finger Jacobian. This metric is a measure of the volume of the conditioning ellipsoid (section 4.7.2), and maxima in the field correspond generally to configurations from which the manipulator can generate forces and velocities equally well in all directions.

For some manipulators, as in the example in section 10.1.3 that considered a finger of the Utah/MIT dexterous hand, the manipulability measure itself can be used as a navigation function. For these manipulators, a manipulability objective function is written in the form of the potential function¹

$$\phi_m = -\sqrt{\det(\mathbf{J}\mathbf{J}^T)}. \quad (10.12)$$

This potential leads to the closed-loop POSTURAL controller presented in [equation 10.7](#). The independent response of this controller applied to a finger in the Utah/MIT hand was presented in [figure 10.3](#).

[Figure 10.7](#) illustrates a case in which the manipulator starts in a suboptimal initial configuration with one joint fully extended and another

fully flexed. The distal joint remains strongly flexed throughout the entire Cartesian motion pictured in panel (a).

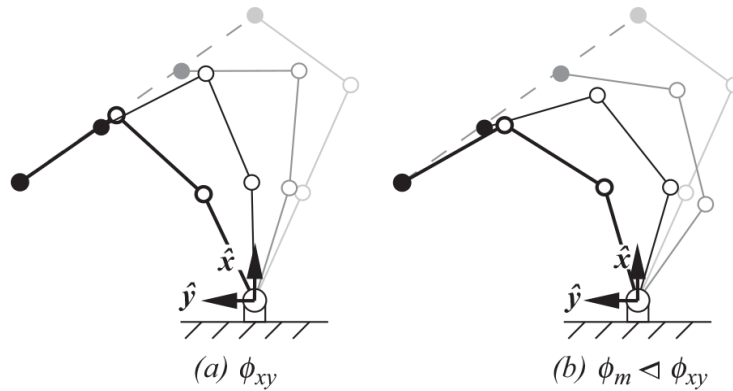


Figure 10.7

A subordinate POSTURAL action based on the manipulability index is projected into the nullspace of the superior Cartesian endpoint controller

To describe more humanlike finger movements, Yoshikawa [298] proposed a POSTURAL controller based on the manipulability potential (section 10.1.3) subject-to the Cartesian endpoint controller,

$$\Delta\theta = \mathbf{J}_{xy}^{\#} \Delta\phi_{xy} + \left[\mathbf{I} - \mathbf{J}_{xy}^{\#} \mathbf{J}_{xy} \right] \left(\mathbf{J}_m^{\#} \Delta\phi_m \right),$$

and results in the finger postures in figure 10.7(b) for these initial and final endpoint coordinates. In general, these postures of the finger are better at generating forces and velocities in all directions in the x - y plane and, so, approximate some of the tacit knowledge encoded in the biomechanical structures critical to the perceptual and motor functions of the human hand.

□

10.1.5 States: $\gamma(\phi, \dot{\phi})$

Many factors affecting the quality of a solution to a control task depend on decisions about sensory and motor resources. Matching informative features in the external environment with complementary sensor geometries in the embodied system is critical to solving everyday control problems safely and efficiently in unstructured environments. Often, the properties on which solution safety and quality depend are difficult or impossible to measure

directly. In these situations, extended periods of interaction and observation can be used to infer missing (or hidden) influences on control quality.

In the dynamical systems community, Taken's embedding theorem [269] states that a history of $(2n + 1)$ time-delayed observations of the same quantity (e.g., $[\phi(t), \phi(t-\tau), \phi(t-2\tau), \dots, \phi(t-2n\tau)]$) is sufficient to characterize the behavior of an n -dimensional dynamical system. There are equivalent results in control theory [1]. Time-delayed coordinates (such as these) arise in information theory as well, where the information in a signal is encoded in finite-temporal differences approximating time derivatives of the signal at time t (chapter 8),

$$\left[y(t), \frac{(y(t) - y(t - \tau))}{\tau}, \frac{(y(t) - 2y(t - \tau) + y(t - 2\tau))}{\tau^2}, \dots \right].$$

The time history of feedback in the *phase portrait* $(\phi, \dot{\phi})$ is, likewise, a faithful representation of system dynamics. The stream of coordinates comprising the phase portrait of controlled sensorimotor systems, however, characterize interactions—visual and tactile—between the robot and environmental stimuli.

State in the control basis is defined in terms of a *membership function* $\gamma(\phi, \dot{\phi})$ that, in its most general form, classifies the temporal response of the system to nonstationary, multimodal stimuli in the run-time environment and, thus, supports a uniform framework for active perception and information gathering. The approach is introduced using an example.

Example: Representing Grasp Dynamics

Coelho [56, 59, 57, 58, 55] used the three-fingered Stanford/JPL hand and contact force measurements in \mathbb{R}^3 from fingertip load cells to experiment with reinforcement learning algorithms for grasping. A pair of closed-loop control primitives derived from the control basis framework were used to minimize the sum of contact forces and moments, respectively, in a multiple contact grasp configuration. Learned compositions of these controllers can guarantee locally optimal contact configurations for a large class of object geometries. Coelho studied control decisions that could reallocate contact resources to find better grasps. One of the contributions of this work was a

method for using membership functions in the phase portrait to provide state feedback for reinforcement learning applications.

Figure 10.8 illustrates the phase portrait for a two-contact moment residual controller engaging an irregular planar triangle that is representative of the kinds of models that Coelho acquired from experiments using real robots and three-dimensional objects. The left panel shows the case when the two-contact moment residual controller derives tactile feedback from the particular pair of faces indicated. The initial contact configuration is illustrated in inset (a). The gradient of the moment residual potential drives the system to intermediate grasp configuration (b) and ultimately converges to configuration (c), where the rate of change in value of the potential $\dot{\phi}_m$ is approximately zero. The region defined by states that lead to the same attractor is called the *basin of attraction*—states in the neighborhood of this trajectory are funneled to the same equilibrium state. We say that the irregular triangle *activates* the controller in this fashion whenever the pair of contacts are placed on these two particular faces.

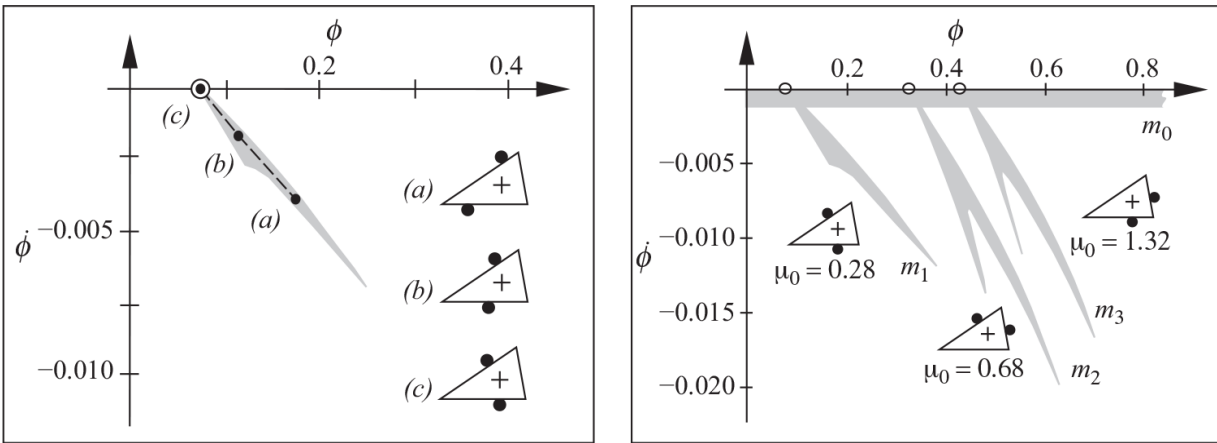


Figure 10.8

The evolution of a two-contact grasp using the moment residual control gradient. (Left) Three intermediate two-contact geometries en route to the final grasp configuration and their respective $(\phi, \dot{\phi})$ coordinates on the phase portrait for this pair of faces. (Right) Multiple distinct basins of attraction when different pairs of faces are considered (reprinted by permission [59]).

A single controller may have multiple basins of attraction to reflect all the ways that the irregular triangle can activate the moment residual controller, as in the right panel of figure 10.8. Three such basins of attraction, corresponding to the three unique combinations of two contacts

on three faces, are illustrated. Now, the state for the two-finger grasp on the irregular planar triangle is the distribution over membership in four independent phase portrait models $\{m_0, m_1, m_2, m_3\}$, where model m_0 denotes convergence ($\dot{\phi} \approx 0$) and is common to all asymptotically stable actions. Models $m_1, m_2,$ and m_3 summarize prior experience with the two-contact moment residual controller and this object.

Each basin of attraction in [figure 10.8](#) represents a prediction about the asymptotic quality of the evolving grasp. The probability distribution over models, therefore, can be used to compute the expected performance of future equilibrium grasp configurations as $t \rightarrow \infty$. For example, in [figure 10.8](#), the minimum coefficient of friction (μ_0) required (assuming point contacts with friction) to generate internal grasp forces in the equilibrium grasp configuration² is computed for each of the models and, weighted by the probability distribution over models, used to estimate the hidden frictional properties of objects.

In Coelho’s experiment, each γ_i for a given closed-loop action is a Boolean function asserting that $Pr(m_i|z_{1:k}) > \beta$, where β reflects the threshold of belief that run-time observations $z_{1:k}$ could have been generated by model m_i . Given this novel definition of state, Coelho demonstrated that the robot learns to allocate effectors $\tau \subseteq \{T, 1, 2\}$ and fingertip load cells $\sigma \subseteq \{T, 1, 2\}$ to measure contact positions and normals as needed to funnel the grasp state to the neighborhood of the optimal solution.

[Figure 10.9](#) illustrates the use of the control basis in a hypothetical regrasping sequence reported by Coelho [55]. Five models are used to explain observations. The grasp state for the two-contact grasp is a probability distribution over models $m_0, m_1,$ and m_2 ; for the three-contact grasp controller, the distribution is over models $m_0, m_3,$ and m_4 . The state of this grasping task is the union of these two control contexts. Observations from controller $\phi|_{T1}^{T1}$ generate state $[0 \ 0 \ 1 \ X \ X]$ (model m_2) and trigger a control transition to the three-contact grasp controller $\phi|_{T12}^{T12}$ that generates $[0 \ X \ X \ 1 \ 0]$ (model m_3). From this state, the grasp policy de-allocates contact #1 and runs $\phi|_{T2}^{T2}$ to generate state $[1 \ 1 \ 0 \ X \ X]$ (models m_0 and m_1). There is new information in the decision process every time the membership pattern changes for a given controller—it is not necessary to converge before a new controller is selected. This policy terminates when controller $\phi|_{T2}^{T2}$ recognizes

model m_0 and m_1 simultaneously, signifying that the system has reached the attractor with the smallest residual error.

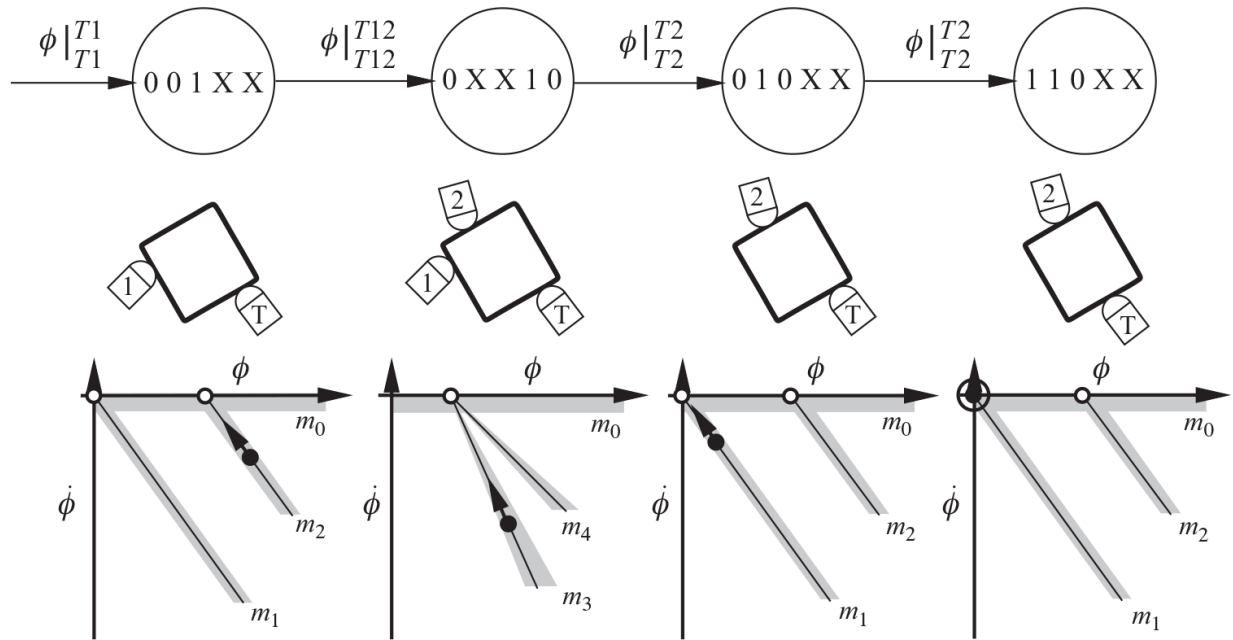


Figure 10.9

A simulated planar grasp of a square object. States are indicators of membership in models m_i , $i = 0, 4$. X denotes “unknown,” “T” is the thumb, “1” is the index finger, and “2” is the middle finger.

In one demonstration, Coelho reported 64 models describing the response of two- and three-contact moment residual controllers applied to three different objects. The result of enhancing the state space significantly reduced the number of tactile probes required to converge to a grasp and significantly improved the asymptotic quality of the result [59, 55]). The ensemble of control actions avoids challenging problems associated with the local perspective of each of them and views grasping with incomplete state as a sequential information gathering activity that optimizes the hand-object interaction.

□

Policies like that in figure 10.9 incorporate tacit properties of the hand into the sensory and motor subtasks involved in grasping objects. Information, like whether finger #1 is left or right of finger #2, is translated into implicit (actionable) forms that can be reused in different, but related,

circumstances. For example, Coelho also demonstrated that the state of the system can use the same transition structure to recognize the object based solely on the history of tactile feedback [55].

Membership functions can also manage complexity by effectively reducing the precision of state representation, particularly in the earliest stages of sensorimotor development. In keeping with observations of human neonates, the sensorimotor stage generally focuses on low-dimensional behavior guided by a maturational schedule. Likewise, skills on the developmental frontier of an “immature” robot can begin by regulating important control contexts in low-dimensional sensorimotor spaces, advancing the frontier as requisite skills are mastered. A developmental curriculum like this uses systems of constraints on perceptual and motor resources to throttle learning complexity and, therefore, to focus on foundational developmental milestones.

10.2 A Multimodal Landscape of Attractors

Figure 10.10 shows a potential function ϕ_1 defined on continuous coordinates $\mathbf{q} = [q_1 \ q_2]$ in a task space. Controller $c_1 = \phi_1|_{\tau_1}^{\sigma_1}$ defines a subset of the domain called the *region of attraction*³ R_1 , where sensors σ_1 detect the requisite multimodal goals and potential ϕ_1 is sensitive to movements in motor units τ_1 . In general, the shape of R_1 depends on the geometry of the environment and can change over time. The one-dimensional control gradient $-\nabla\phi_1$ is illustrated with the complementary one-dimensional nullspace \mathcal{N}_1 that is orthogonal to the gradient everywhere in the interior of R_1 .

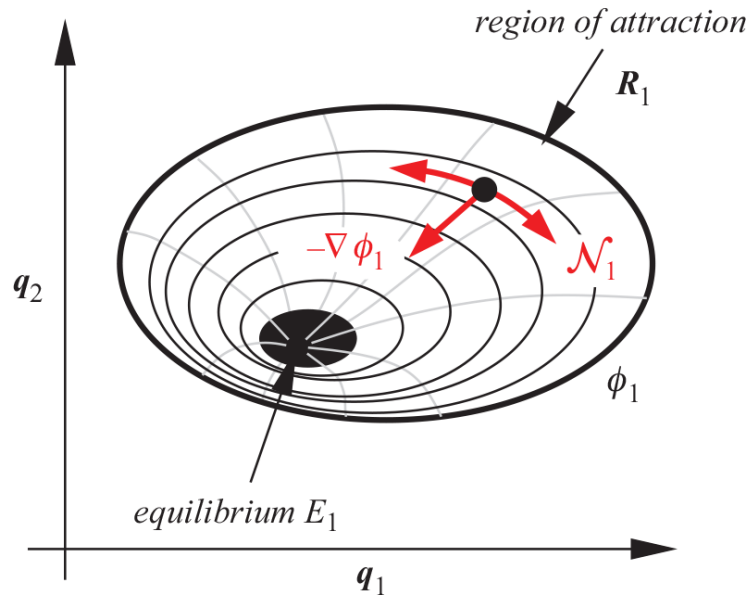


Figure 10.10

The region of attraction R_1 and the equilibrium set $E_1 \subset R_1$ for navigation function ϕ_1 corresponding to controller c_1

Whenever task space coordinates \mathbf{q} are in the interior of set R_1 , controller c_1 can be engaged to *funnel* [44] task space coordinates into set E_1 containing an equilibrium fixed point by gradient descent on ϕ_1 . The precise time-domain response depends on the underlying dynamics of the controlled process, and it may persist beyond the time at which the system first reaches the fixed point. In this case, the controller actively suppresses disturbances and, up to motor limits, confines task space coordinates to remain near set E_1 . In this light, the controller serves a role commonly found in other optimization frameworks. For example, the Lagrange multiplier method [27] uses state-dependent functions like ϕ_1 to represent both the objective to be optimized and the constraints that optimization must respect and the *objective functions* commonly used in optimization methods are essentially potential functions.

The response of the controller is classified using the membership function γ_1 introduced in section 10.1.5—where γ is a probability distribution over models in the phase portrait summarizing past experience. As before, the membership function is used to recognize different ways that the environment affords fixed points in potential ϕ_1 and, thus, different ways that it *activates* closed-loop reflexes. The precision of the membership

function can grow over extended periods of training and exploration but, perhaps more importantly, simple membership functions can provide critical structure in early developmental stages. For example, in a fully observable context, a simple Boolean membership function can be used to distinguish controllers that are activated by stimuli in the environment from those that are not.

$$\gamma_1(\phi, \dot{\phi}) = \begin{cases} \text{NO_REFERENCE} & : \sigma \text{ undetected} & \rightarrow 0 \\ \text{ACTIVATED} & : \|\mathbf{J}_1\| \geq 0 & \rightarrow 1 \end{cases}$$

Simple structures like this can be exploited to accelerate early stages of learning significantly.

When a second closed-loop action shares domain \mathbf{q} as in [figure 10.11](#), the state of the system is approximated by the multi-objective control status $\boldsymbol{\gamma} = [\gamma_1 \ \gamma_2]^T$. Panel (a) shows how the geometric arrangement of the two controllers can act as a proxy for task variables \mathbf{q} by carving up the domain into four disjoint regions in the q_1 - q_2 plane where combinations of c_1 and c_2 can influence the objective performance of the system in different ways. Panel (b) illustrates the *image* of the domain through closed-loop interactions c_1 and c_2 . The pair of control responses distinguish four different multimodal *aspects* of the environmental scene. The term *aspect* is a reference to a representation of objects invented in the computer vision community [[232](#), [231](#)] to model the influence of viewpoint on visual appearance. It is generalized here to model the influence of a controllable whole-body sensor geometry on multimodal control interactions [[258](#), [156](#), [157](#), [158](#), [155](#), [248](#), [164](#)]. This framework represents multimodal affordances in the environment using the common language of membership functions over combinations of controllers and is used to model the joint response of multiple, coupled dynamical systems.

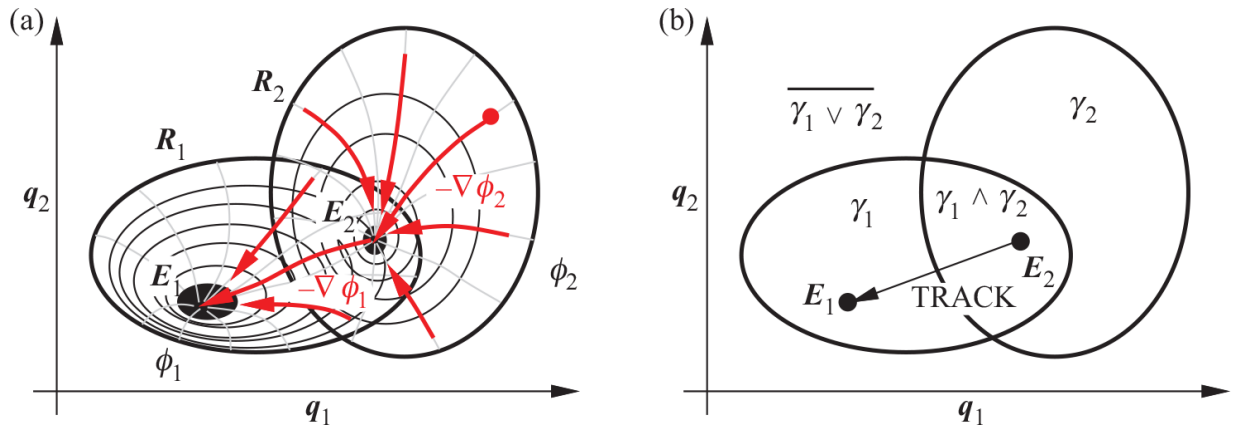


Figure 10.11

The region of attraction surrounding E_1 can be extended by sequential control. A sequential combination of c_1 and c_2 can be configured to approach minimum E_1 from all states in the interior of $R_1 \cup R_2$.

The relative geometry of the attractors induces a sequential structure called a probabilistic Aspect Transition Graph (ATG). It supports a mathematical framework (section 10.2.1) for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker.⁴ A trivial example of an ATG is illustrated in figure 10.11(b) by the single edge representing a TRACK action that transitions between vertices E_2 and E_1 .

In general, multimodal aspects are only partially observable, and a history of predictable transitions between aspects is required to recognize latent geometric structures completely as, for instance, when a visual geometry is altered so that previously unseen aspects of an object can be seen. In a similar fashion, the generalized aspect supports transitions from purely visual states to visual and tactile states to realize contact forces and grasps. In the remainder of the text, the term *aspect* is used as a synonym for the multimodal *state* of the control task.

As the number of attractors increases, so does the precision of the multi-objective control state. Moreover, the geometrical arrangement of attractors can reveal transition structures that form *roadmaps* through the state space. In the context of a particular room or object, for instance, such a roadmap can be used to optimize the current sensor geometry with respect to signals afforded by the environment and required by a task.

Example: Multi-Objective Visual Inspection Task

Hart and Grupen [107] presented a control basis program that uses POSTURAL, SEARCH, and TRACK control actions to construct a sequential skill for reading barcodes. A bimanual robot, named Dexter (figure 10.12), incorporates two 7-DOF whole arm manipulators (WAMs) from Barrett Technology; two 3-fingered, 4-DOF Barrett hands; and a 2-DOF head mechanism. Feedback from robot includes joint angle positions from each actuated degree of freedom, six fingertip contact load cells, and a pair of Sony cameras mounted on the head. The task was to inspect packages and read barcodes for the purpose of routing packages in a large distribution center.

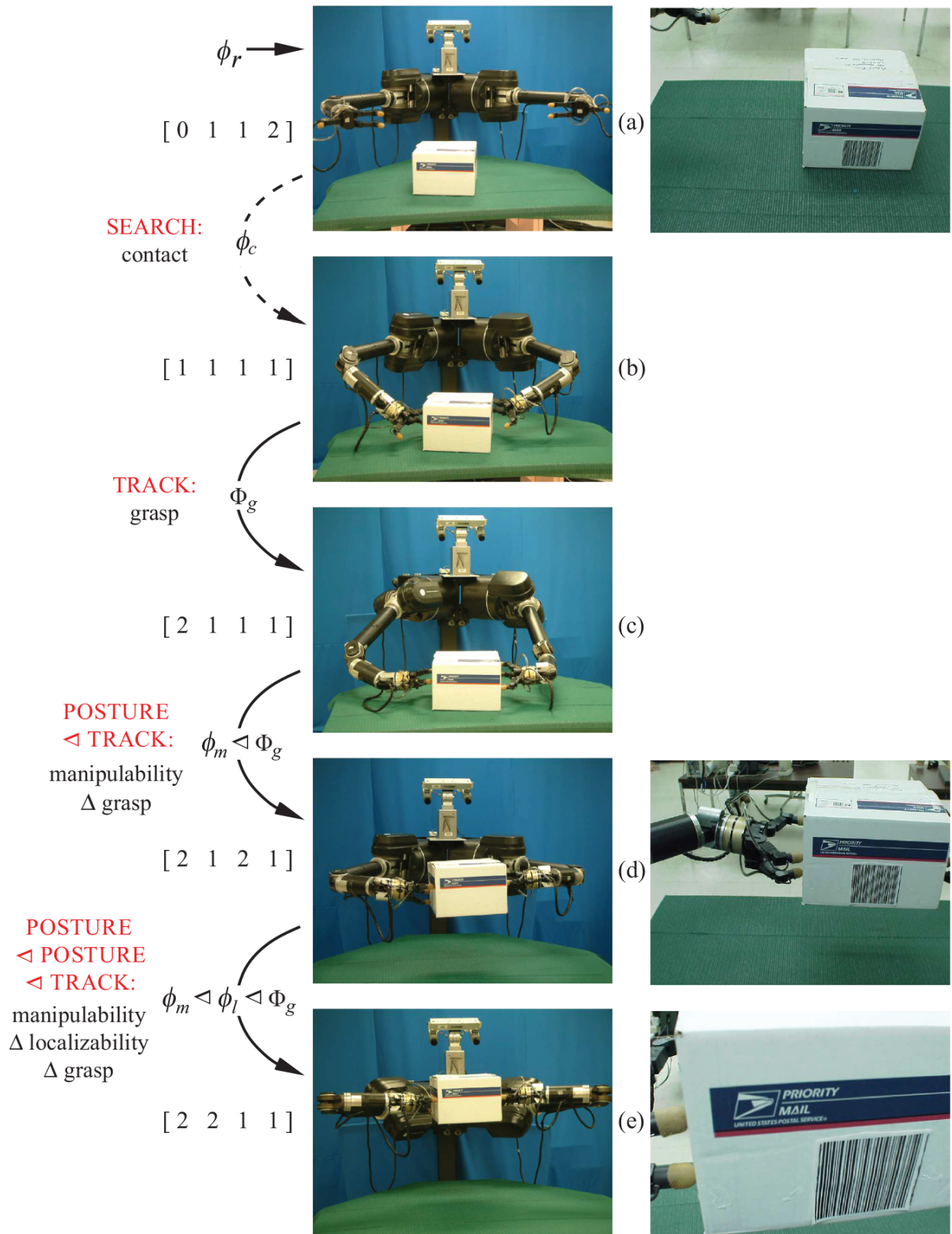


Figure 10.12

A visual inspection sequence traverses an aspect transition graph defined by $[\gamma_g \ \gamma_l \ \gamma_m \ \gamma_r]$ states and

POSTURAL, TRACK, and SEARCH actions in \mathcal{A} to read the barcode on packages

The focus of this example is a skill for reading the barcode on a randomly oriented package. In some cases, the robot requires a contingency for manipulating the package to control the viewing geometry. In this context, the skill is preprogrammed sequential policy for optimizing the viewing geometry of the object using control primitives and other preconfigured skills.

ϕ_r : the POSTURAL *range of motion* objective (exercise 10.3.4) for each of Dexter’s arms;

ϕ_c : the SEARCH action that samples places in \mathbb{R}^3 where *contact* signals are likely to be detected on fingertip loadcells given visual inputs;

Φ_g : a TRACK objective that engages sequences of contact placement and force TRACK-ing primitives to implement a policy⁵ for *grasp* control [55, 233];

ϕ_m : a POSTURAL objective that optimizes the bimanual *manipulability* of the arms (equation 10.12); and

ϕ_l : a POSTURAL *localizability* controller that optimizes stereo acuity using $\phi_l = -\sqrt{\det(\mathbf{J}\mathbf{J}^T)}$, where \mathbf{J} is the stereo reconstruction Jacobian (section 4.7.3).

The state of the barcode reading application is represented in terms of feedback status from TRACK and POSTURE controllers using membership functions that distinguish the following categories in the phase portrait.

$$\gamma_i(\phi, \dot{\phi}) = \begin{cases} \text{NO_REFERENCE} & : \sigma_i \text{ undetected} & \rightarrow 0 \\ \text{TRANSIENT} & : \|\mathbf{J}_i\| > \epsilon & \rightarrow 1 \\ \text{CONVERGED} & : \|\mathbf{J}_i\| \leq \epsilon & \rightarrow 2 \end{cases}$$

The program in the left column of figure 10.12 is designed to modify the sensor geometry in order to improve visual acuity as necessary until the barcode is readable. The middle column presents several snapshots of Dexter performing a visual inspection task [107]. In panel (a), Dexter is in his home posture—where all degrees of freedom in the robot (excluding the hands) are in the middle of their respective ranges of motion—the outcome of executing the closed-loop range of motion controller based on ϕ_r .

When Dexter is home, the visual field of view of the workspace is unoccluded, and image differencing techniques distinguish foreground and background pixels. The foreground blob is used as the reference for a TRACK controller to foveate each eye on this stimulus. This controller is not illustrated in [figure 10.12](#), but it is running throughout the task to preserve the conditions required for stereo triangulation and, thus, the conditions required to localize the object.

On the basis of the Cartesian blob feedback, control setpoints are submitted to arm controllers that position the fingertips near the surface of the package where contact forces are expected. This is accomplished using a SEARCH action based on potential ϕ_c that positions the hands at locations sampled from the distribution $Pr(\text{contact}|\text{foreground blob})$. The result is the sensor geometry in panel (b) that achieves contacts on some of the fingertip load cells. From this state, the grasp policy Φ_g is engaged to bring all six load cells into contact, distribute them so that the sum of measured contact forces and moments is minimized, and CONVERGE to a reference grasp force (panel (c)). Once the grasp is established, it continues to execute as the superior control task in the multi-objective control configuration for all subsequent actions to make sure that the grasp condition is preserved over the course of subsequent manipulation actions.

Panel (d) shows the outcome of a composite action $\phi_m \triangleleft \Phi_g$ that has locally optimized the manipulability of the bimanual hand-arm configuration *subject-to* grasp control constraints. Between panel (d) and panel (e), the robot executes $\phi_m \triangleleft \phi_l \triangleleft \Phi_g$, where a hand-arm manipulability controller is subordinate to a controller that descends the gradient of the binocular localizability field (section 4.7.3) to optimize visual acuity subject to Φ_g .

The rightmost column shows images from the robot's left camera (a) when Dexter is in the home posture; (d) after the box has been grasped and transported to a well-conditioned kinematic configuration; and (e) after the object is moved to the location where binocular localization is adequate for reading the barcode. Much of this strategy is aimed at exploiting kinematic properties of the robot—manipulability, range of motion, and visual localizability—to solve the task.

As demonstrated in this case study, programming control basis applications involves designing a sequence of control actions that exploit the tacit abilities of the sensorimotor system and transform them into actionable forms. Development is, in part, about accumulating policies like this with embedded control knowledge and using these temporally extended skills as actions in a hierarchical behavioral system.

The program Φ_{bcr} (barcode reader) in [figure 10.12](#) responds robustly to a large variety of instances in the class of barcode reading problems—it is robust to variation in run-time conditions like table height, object shape and initial location. Moreover, Φ_{bcr} handles these contingencies internally and summarizes all of this activity in status feedback $\gamma_{bcr} \in \{0, 1, 2\}$ for use in applications that invoke this skill.

In the next section, we introduce a framework for learning optimal control compositions for behavior like Φ_{bcr} . We discuss how implicit knowledge structures like Φ_{bcr} can be acquired using a semi-autonomous, exploration-based learning process and can be transferred to new instances of behavior in this class.

□

10.2.1 Reinforcement Learning in a Landscape of Attractors

The Markov Decision Process (MDP) is an appropriate mathematical framework for sequential decision processes like that shown in [figure 10.12](#). A system is Markovian if knowledge of its current state s_k provides a sufficient context for choosing the optimal control action a_k^* from that state. Formally, an MDP is a tuple,

$$M = \langle S, A, \Psi, P, R \rangle, \tag{10.13}$$

where S is the set of system states, A is the set of actions, $\Psi \subseteq S \times A$ defines the subset of actions that are allowed from a particular state, $P: S \times A \times S \rightarrow [0, 1]$ is the probability that state s_k and action a_k generates a transition to state s_{k+1} , and $R: S \times A \rightarrow \mathbb{R}$ is a function that maps state-action pairs to a real-valued reward [236].

A policy, $\pi(s, a)$, is a function that returns the probability of selecting action $a \in \Psi$ from state $s \in S$ in order to optimize future rewards R . One of the

virtues of reinforcement learning is that prior knowledge of the transition policy P in [equation 10.13](#) is not required to estimate an optimal policy. Instead, RL can compile these transition probabilities directly into implicit knowledge embedded in the policy. However, because P transfers efficiently to other tasks, reinforcement learning also contributes a means of accumulating explicit knowledge by estimating P explicitly.

In the rest of this section, we describe a particular means of solving for the optimal solution to an MDP using Q-learning [[288](#), [289](#)].

The Bellman Equation—Following Sutton and Barto [[268](#)], the value of state s under a policy π , denoted $V^\pi(s)$, is the expected sum of discounted future rewards when policy π is executed from state s ,

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right\}, \quad (10.14)$$

where $E\{\cdot\}$ denotes the expected value, $0.0 < \gamma \leq 1.0$ represents a discounting factor per decision, and scalar r_t is the reward received at time t .

If we extract the $k=0$ term from [equation 10.14](#),

$$\begin{aligned} V^\pi(s) &= E_\pi \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right\} \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \left[R_{ss'}^a + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s' \right\} \right], \end{aligned}$$

where $\pi(s, a)$ is the fixed policy defining the probability of selecting action a from state s ; $P_{ss'}^a$ is the transition probability of the underlying MDP—that is, the probability that $s \xrightarrow{a} s'$; and $R_{ss'}^a$ is the reward associated with this transition. The expected value in the last expression is just $V^\pi(s')$, so that

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')]. \quad (10.15)$$

Value functions derived from policy $\pi(s, a)$ interact with an underlying MDP to provide a partial ordering over all possible policies. If policy π achieves more reward than π' , then $V^\pi(s) \geq V^{\pi'}(s)$ for all $s \in S$. The optimal

policy π^* produces a value function, V^* , that dominates all other value functions possible,

$$V^*(s) = \max_{\pi} V^{\pi}(s).$$

Since the optimal policy produces the most value at every state, it can be written without reference to π^* ,

$$V^*(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^*(s')], \quad (10.16)$$

which is known as the *Bellman optimality condition*. This condition establishes a consistency relation between the value of state s , $V^{\pi}(s)$, and the value of possible successor states, $V^{\pi}(s')$. The Bellman condition requires that the value of a state must be the sum of the discounted value of the expected next state and a reward $R(s, a)$ expected en route. Once a suitable estimate of $V^*(s)$ is attained, any policy that is greedy with respect to $V^*(s)$ is an optimal policy.

Value Iteration—A collection of algorithms used to compute optimal policies from complete knowledge of the underlying MDP are called *Dynamic Programming* (DP) algorithms. Within this general class, Reinforcement Learning (RL) algorithms [268] are distinguished by properties that make them ideal for realizing developmental systems.

Equation 10.15 provides the basis for a numerical iteration with which to estimate $V^{\pi}(s)$.

$$V_{k+1}(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V_k(s')] \quad (10.17)$$

This relation is used in *value iteration* algorithms that converges to V^{π} as $k \rightarrow \infty$ [268]. Typically, dynamic programming employs an iterative *sweep* though the state-action space called a *full backup*. However, reinforcement learning techniques generally estimate $V^{\pi}(s)$ using *sampled* backups. If a robot starts from random initial states and follows a fixed policy π while accumulating statistics regarding the average discounted reward achieved from each state, then these averages will converge uniformly to the value function V^{π} as the number of visits to each state approaches infinity.

Although optimality guarantees are compromised for sampled exploration, this type of algorithm is important for developing robots because it permits samples to be drawn most often from regions of the state and action space that appear to be most fruitful with respect to the task. Regardless of the implementation, greedy ascent of the converged value function constitutes an optimal policy for accumulating reward.

To improve the policy, $\pi(s, a)$, the backup process explores actions other than those prescribed by the greedy ascent of $V(s)$ and uses the current estimate of $V^\pi(s)$ to approximate the subsequent value of adhering to π from that point on. To see this clearly, we can write the value recursion in state-action form, referred to as the *quality* function.

$$Q^\pi(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V_k(s')] \quad (10.18)$$

The policy adapts to exploration by evaluating whether there exists an action, a , other than $\pi(s)$, for which $Q(s, a) > Q(s, \pi(s))$. If so, then it is better to alter the policy to call for this new action every time state s is visited. The new policy, π' , is identical to π , except that in state s , it now recommends the new action a . The policy improvement theorem [268] guarantees that a procedure like this will lead monotonically toward optimal policies.

Another practical modification to value iteration examines the quality of adjacent states exclusively:

$$Q(s, a)_{k+1} \leftarrow Q(s, a)_k + \alpha \left[r(s') + \gamma \max_a Q(s', a)_{k+1} - Q(s, a)_k \right], \quad (10.19)$$

where γ is, once again, a discount factor that accounts for expected future rewards, and a new parameter $\alpha < 1$ is used to apply an appropriate weight to the most recent experiment relative to the existing experience codified in $Q(s, a)_k$. Watkins [288, 289] showed that this algorithm converges to optimal policies in many useful kinds of problems. Rewriting this expression in a slightly different form reveals the commonly used Q-learning backup equation

$$Q(s, a)_{k+1} \leftarrow (1 - \alpha)Q(s, a)_k + \alpha \left[r(s') + \gamma \max_a Q(s', a)_{k+1} \right]. \quad (10.20)$$

10.2.2 Skills

The control basis supports a Markovian description of controlled interactions between an embodied agent and a complex Newtonian environment. It provides all sensor-effector combinations in the embodied system and, with appropriate potential functions, all closed-loop actions and states afforded by the environment. For interesting robots and environments, this can yield an enormous variety of primitive actions and states.

Reinforcement learning [267] is a natural choice for solving an MDP derived from the control basis because it can compose optimal sequences of actions from direct exploratory interaction with the domain. However, machine learning algorithms that depend on randomized exploration are subject to the curse of dimensionality [22], which predicts that the probability of discovering rewarding state-action combinations decreases exponentially as the dimensionality of the search space grows. Without an effective developmental organization, infants (or interesting robots) might spend a large fraction of their lifetime exploring actions that do not produce reward.

Lessons learned from the overview of sensorimotor learning during the first year in a human infant's life (chapter 9) suggest that development is guided by a genetically determined curriculum that proceeds, for example, from midline to periphery and from head to tail. These patterns of maturation in the human infant can be conveniently expressed using $\Psi \subset A$ (equation 10.13)—a lower-dimensional subset of all actions A that are eligible for execution from each state. Ψ can be used to define a developmental frontier—a small niche in the state-action space that can be explored efficiently and that evolves as new skills accumulate. Skills that are acquired within a developmental frontier are added to A to support more skills as the frontier advances—acquiring hierarchical skills that quickly subsume more primitive actions when possible and help to throttle incremental learning complexity. Membership functions for $a \in \Psi$ can be used to satisfy logical conditions for safety, reachability, controllability, and learnability—properties that support otherwise autonomous learning agents.

Hierarchy reduces the computational expense of planning in AI systems by storing and reusing successful plans. A sequence of planned actions that

perform well in many related situations can be stored in memory as a *macro*—an operator that maps a set of initial conditions (or preconditions) into outcomes (or postconditions). Good macros are added to the set of actions A considered by a planner. Although the size of the state-action space increases as a result, the extra cost of searching for solutions in $S \times A$ is more than compensated by the value of the implicit knowledge stored in the macro—eliminating the need to reconstruct the sequence in every case in which it is applicable. Versions of this powerful idea are employed frequently in planners to accumulate knowledge and to amortize the cost of search [85, 197, 5, 195, 152, 161, 86, 37, 51, 52].

In developmental psychology, Piaget used the term *schema* to refer to a kind of macro that infants use to explore their environments [227, 226]. He proposed that infant development occurs in stages that acquire schemas using a process called *accommodation* that composes sequences of actions to achieve a goal. A schema serves both as an abstract action *and* as the basis for a cognitive representation. From the Piagetian perspective, new experiences are interpreted using a process of *assimilation*, wherein representations of the world are organized using these schematic units of knowledge. Thus, by this account, the infant interprets experiences based on what they already know.

Roboticians have also explored schemas as an action-based representation for common knowledge regarding objects and categories. Arbib [6] discussed schemas as hierarchical functional entities that are used to construct new schemas. Drescher [72] used schemas to represent empirical cause and effect relations in sensorimotor systems. Holmes and Isbell [116] provided a more general implementation of the schema proposed by Drescher. They created suitable rules for schema discovery and refinement and showed that they compare favorably to other predictive state representation.

Computational issues in robotic applications involving complex forms of state estimation and high-dimensional control have been used to motivate hierarchical architectures as well. Many examples exist in the literature ranging from architectures for process control, behavior-based systems, planners in partially observable systems, architectures that focus on the full range from gross motor (mobility) to fine motor (manipulation) skills, and architectures for next-generation autonomous robot systems [194, 3, 123,

215, 134, 11, 164]. This challenging problem remains an active and open area of research, and developmental systems offer a new theoretical and experimental perspective.

The final chapter illustrates the use of the control basis to construct an analog of the native reflexes with which an embodied agent discovers sequential skills—hierarchical, multimodal *roadmaps* in $S \times A$. We examine how knowledge is compiled into actionable forms and used to help solve problems that are out of reach without the necessary prerequisite knowledge. The goal is to motivate a cumulative hierarchy of skills and to evaluate the impact of the resulting implicit knowledge on the performance of future problem-solving.

10.3 Exercises

1. **Critical points.** Figure 10.1 distinguished three types of critical points on the basis of properties of the Hessian of the potential function in the neighborhood of the critical point.

(a) What property of the Hessian is associated with the neighborhood surrounding the type 0 critical point?

(b) What type of critical points can be present in a harmonic function, and how are these types of critical points reflected in the Hessian?

(c) What type of critical points correspond to unstable fixed points in a potential function, and what are the properties of the Hessian that determines that a critical point is unstable?

2. **Navigation functions: Hooke's law.** Argue for or against the proposition that the quadratic potential function (Hooke's law) is a navigation function.

3. **Navigation functions: Harmonic potential.** Discuss the qualifications of a harmonic function as a navigation function.

4. **Navigation functions: Range of motion potential.** Consider a 3R manipulator, where each joint angle θ_i is restricted to the interval $[-\pi/2, +\pi/2]$. The range of motion reflex helps to avoid joint range limits by distributing deviations from $\theta_i = 0$ evenly over all three degrees of freedom.

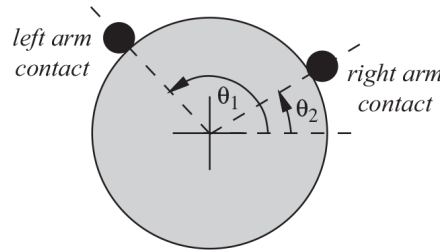
$$\phi_{rom} = - \sum_{i=1}^3 \cos(\theta_i) = -\cos(\theta_1) - \cos(\theta_2) - \cos(\theta_3)$$

- (a) Is this range of motion potential a navigation function?
 (b) Write the expression for the control Jacobian \mathbf{J}_c .

5. **Grasp contact control.** Consider a planar grasping problem where the generalized object force (or *wrench*) generated by contact i applied at position θ_i is

$$\mathbf{w}_i^T = [f_x \ f_y \ m_z] = [-\cos(\theta_i) \ -\sin(\theta_i) \ 0].$$

Define the wrench residual produced by N contacts, to be $\rho = \sum_{i=1}^N \mathbf{w}_i$. In the following, consider the two-contact system ($N=2$) on the circular object illustrated here.



- (a) Write a closed-form expression for potential function $\phi_w = \rho^T \rho$ that represents the (scalar) squared wrench residual.
 (b) Show that $\phi_w(\theta')$, where $\theta' = (\theta_1 - \theta_2)$ is a navigation function on $-\pi \leq \theta' < +\pi$.
 (c) Write the control Jacobian for the wrench residual

$$\mathbf{J}_c^w = \begin{bmatrix} \frac{\partial \phi_w}{\partial \theta_1} & \frac{\partial \phi_w}{\partial \theta_2} \end{bmatrix} \text{ so that } \Delta \phi_w = \mathbf{J}_c^w \Delta \theta.$$

- (d) Derive a closed-form solution for the right pseudoinverse

$$(\mathbf{J}_c^w)^\# = (\mathbf{J}_c^w)^T [\mathbf{J}_c^w (\mathbf{J}_c^w)^T]^{-1}.$$

- (e) Write the expression defining differential movements for θ_1 and θ_2 that descend the wrench residual potential

$$\Delta\theta = (\mathbf{J}_c^w)^\# \Delta\phi_w = -\kappa (\mathbf{J}_c^w)^\# \phi_w.$$

6. Invent your own homework. Make a homework problem out of your favorite content in chapter 10. Write a question and a solution for it from the material in the reading. They should call for a comparison, an observation, a short discussion, analysis, and calculation—it should not take more than 30 [*min*] to solve open-book.

1. In the form of a control Jacobian, the manipulability measure defined by Yoshikawa [299] is negated to make the extremum in the field a minimum instead of a maximum.
2. In [55], a mathematical programming problem subject to inequality constraints is used to compute μ_0 .
3. For our purposes, the interior of the region (or basin) of attraction is essentially equivalent to the gray areas in the phase portraits shown in figures 10.8 and 10.9—both sets represent the full range of interaction that a particular controller can elicit from the environment.
4. In general, this graph is equivalent to the probabilistic transition function required to compute optimal policies in discrete-time stochastic systems described using Markov Decision Processes (MDPs). This subject will be introduced in section 10.2.1.
5. The convention in the control basis framework is to distinguish control compositions (concurrent or sequential) from reflexive/primitive controllers by using a capital Φ .

11

Case Study: Learning to Walk

Gross motor skills involve whole-body movements that recruit large muscle groups and apply relatively large forces to the environment. They are a central focus of development in human children during the first two years of life and rely heavily on maturational structure to organize coordinated motor behavior. The pronounced influence of native reflexes and patterns of myelination in the brain during this period contribute to overcoming what would otherwise be crushing sensory and motor complexity. The result is layer upon layer of skills for balance and coordinated behavior in limbs that contribute to the mastery of one's body—focused on large-scale movement, strength and range of motion, vestibular and proprioceptive feedback, body dynamics, postural stability, and locomotion. The right side of [figure 9.18](#) illustrates some of the milestones that demonstrate these new skills for whole-body postural control. Gross motor skills are elaborated over time by processes of accommodation and assimilation to be robust to the demands of different circumstances—they are enhanced and more fully integrated into other tasks throughout childhood and into adulthood.

This book concludes with a discussion of a sequence of experiments conducted by Huber et al. [[121](#), [122](#), [120](#), [118](#), [119](#), [117](#)] in which hierarchical skills associated with four-legged postural stability and locomotion were learned using principles of developmental organization observed in human infants. This case study highlights an embodied system that uses computational models of operant conditioning and theories of reinforcement to learn to control, and subsequently to exploit, some of the events generated during its interaction with the world.

11.1 *Thing*: A Quadruped

The robot in [figure 11.1](#) (*Thing*) was developed to investigate techniques for developmental learning in multilegged locomotion. The robot coordinate frame is attached rigidly to the center of mass (COM) of the platform. The \hat{x} -axis of this frame defines the longitudinal axis of the robot, the \hat{y} -axis runs laterally, and the \hat{z} -axis is parallel to the world frame \hat{z} -axis, as illustrated in [figure 11.2](#).

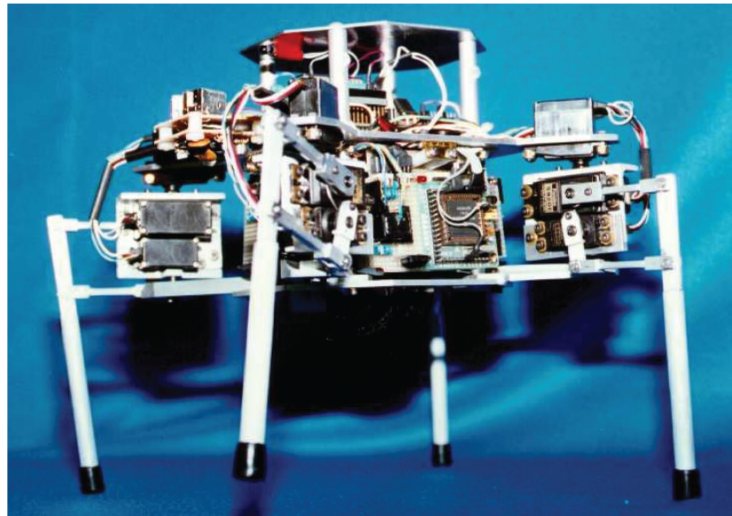


Figure 11.1
Thing, a 12-DOF quadruped

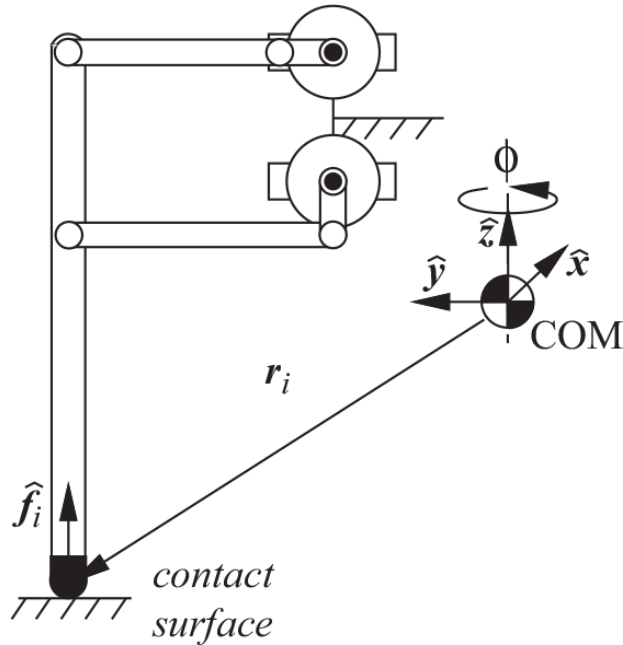


Figure 11.2

A single leg in contact with an external, horizontal surface creates a moment about the platform's center of mass

Thing has four legs $l_i; i=0, 3$ each with three independent degrees of freedom $[\theta_0 \theta_1 \theta_2]_i$ actuated by small servo motors. These angles are measured by sensors in the legs and used to compute the Cartesian endpoint position of each leg $r_i \in \mathbb{R}^3$ in the robot coordinate frame for $i=0, 3$ using forward kinematic relations. The contact load carried by each leg is estimated by measuring torques in the leg motors. Thresholds on motor torques are used to identify those legs currently supporting the platform. The robot continuously measures stride length over a sequence of supporting stances to estimate its pose $(r, \varphi)_{\text{COM}}$ in world coordinates, where φ is the yaw angle about the world frame \hat{z} -axis.

Finally, *Thing* uses a forward-looking infrared proximity detector that transmits IR light and observes reflected energy from environmental surfaces. Depending on the viewing geometry, environmental surfaces within about 20–25 [cm] of the robot typically reflect enough energy to be detected. *Thing* uses this information to accumulate a model of detected obstacles in a world frame occupancy grid for use in motion control.

11.2 Controllers and Control Combinatorics

Locomotor skills for *Thing* employed three classes of closed-loop actions drawn from the control basis: (1) moment residual controllers for platform stability; (2) kinematic conditioning controllers; and (3) a single, closed-loop path controller.

ϕ_m : Moment Residual Control—An important class of controllers for walking and grasping actively reconfigures the geometry of contact interactions in order to minimize the net forces and moments created by (sub)sets of contacts on environmental surfaces. All tasks performed by *Thing* required that the robot learn to place three or four contacts on the ground that define a support polygon and to shift its center of mass in the x - x plane into one of these polygons. A byproduct of transitioning between stable states is the accumulation of displacements (Δx , Δy , $\Delta \theta$) of the robot in the plane. In this case, this happens when the robot has its feet planted and moves its center of mass to achieve the zero moment point. Placing contacts is the responsibility of a family of *moment residual controllers* based on gradient descent of ϕ_m .

For *Thing*, the contact between the leg l_i and the ground is assumed to be a frictionless point contact (section 5.3.3, [table 5.2](#)) that produces unit magnitude force \mathbf{f}_i in the $+\hat{z}$ direction ([figure 11.2](#)) on the endpoint of the robot leg and, consequently, produces moment $\mathbf{r}_i \times \mathbf{f}_i$ about the COM.

The quadratic moment residual,

$$\phi_m = \boldsymbol{\rho}^T \boldsymbol{\rho}, \quad \text{where } \boldsymbol{\rho} = \sum_i (\mathbf{r}_i \times \mathbf{f}_i), \quad (11.1)$$

is a scalar potential function whose objective underlies many approaches to gait stability and prehensile control in the literature. For example, the zero moment point (ZMP) within a support polygon of a statically stable stance [[284](#)] coincides with the minimum in [equation 11.1](#). Minima in ϕ_m derived from unit forces such as these also distribute contact forces evenly.

The *sensory resource model* for moment control is the contact position \mathbf{r}_i between the endpoint of each leg and the ground plane $\Sigma_m = \{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. Legs that are not in contact yield $\mathbf{f}_i = \mathbf{0}$ and, thus, do not influence $\boldsymbol{\rho}$ or ϕ_m . The quadratic potential function in [equation 11.1](#) can be configured with one or more contacts, but to stabilize the walking platform, only $n \geq 3$ contacts are considered. Since *Thing* has four legs, there are $C_3^4 + C_4^4 = 5$

possible combinations¹ of contact states that can be optimized with respect to the moment residual control potential ϕ_m .

The *effector resource model* for moment control describes all the effector variables that can be used to descend the gradient of the squared moment residual $\mathcal{T}_m = \{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, (\mathbf{r}, \varphi)_{\text{COM}}\}$. However, we observe that for triples involving legs i , j , and k , the moment residual depends only on five independent effector variables, $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, (\mathbf{r}, \varphi)_{\text{COM}}\}$ that can be combined in

$$\sum_{k=1}^5 C_k^5$$

ways. For the (single) quadruped stance, the moment residual is sensitive to all six independent effector variables, so that the total number of unique parameterizations of moment residual controllers is

$$C_3^4 \left[\sum_{k=1}^5 C_k^5 \right] + C_4^4 \left[\sum_{k=1}^6 C_k^6 \right] = 187.$$

ϕ_k : Range of Motion Kinematic Conditioning Control—If other concerns (e.g., stability and locomotion) cause legs to approach kinematic limits, controllers derived from ϕ_k can be used to reposture multiple limbs so that each joint is closer to the center of its range of motion. When applied to legs in contact with the ground, this action produces byproduct displacements of the robot’s body and, therefore, can contribute to body translation and rotation as well as range of motion.

Each of *Thing*’s legs can be subject to a quadratic range of motion (ROM) potential²

$$\phi_k = \sum_{j=0}^2 (\bar{\theta}_j - \theta_j)^2, \quad (11.2)$$

where j enumerates the three degrees of freedom of leg \mathbf{l}_i and θ_j is the midrange setpoint for joint j . This objective function can be summed over any subset of the robot’s four legs and yields a potential with a positive definite Hessian that satisfies the essential criteria of a navigation function.

In this case, the *sensory resource model* for *Thing* consists of the joint angle configuration $[\theta_0 \theta_1 \theta_2]$ of independent legs in the set $\Sigma_k = \{\mathbf{l}_0, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}$, and, therefore, the power set of effector resources $\mathcal{P}(\Sigma_k) = \sum_{j=1}^4 C_j^4 = 15$ combinations of one, two, three, and four legs.

The kinematic condition of leg \mathbf{l}_i can be modified by changing its endpoint position or by leaving it planted on the ground and changing the value of *Thing*'s position \mathbf{r}_{COM} and/or orientation φ_{COM} . Therefore, the *effector resource model* for the ROM objective ϕ_k is $\mathcal{T}_k = \{\mathbf{l}_0, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, (\mathbf{r}, \varphi)_{\text{COM}}\}$, and the total number of kinematic conditioning controllers is

$$\sum_{j=1}^4 C_j^4 \left[\sum_{l=1}^j C_l^j + \sum_{l=1}^2 C_l^2 \right],$$

where $\sum_{j=1}^4 C_j^4$ denotes 15 unique σ_k whose kinematic condition is measured using proprioceptive sensors and the terms in the brackets count the unique combinations of effector resources used to optimize the ROM measures. The leftmost term in brackets is the number of combinations of leg effectors, and the rightmost term in brackets is the number of combinations of effector variables $(\mathbf{r}, \varphi)_{\text{COM}}$. The result is a family of

$$\sum_{k=1}^4 C_k^4 \left[\sum_{l=1}^k C_l^k + 3 \right] = 110$$

control configurations within the class of kinematic conditioning controllers.

ϕ_p : Path Control—The third primitive action is a closed-loop controller designed to execute collision-free trajectories to Cartesian goals. Potential field ϕ_p is a two-dimensional harmonic function (section 10.1.1) defined on a discrete (x, y) occupancy grid describing the ground plane. The potential at FREESPACE grid nodes is computed using numerical relaxation techniques introduced in appendix C to create the harmonic potential field. Observations of obstacles are added to the occupancy grid as they are discovered using a combination of odometry and the forward-looking infrared bumper. Goal configurations are provided by an external process or user. The gradient of the artificial potential defines collision-free motion

plans from every FREESPACE grid node to the GOAL states. If no goals are reachable, the interior of ϕ_p converges everywhere to the OBSTACLE potential and no gradient (nor, therefore, path) exists.

Only one (whole-body) ϕ_p primitive control configuration was provided in the control design described by Huber. However, during early stages of development, there is no control support for executing the paths generated—prerequisite skills for postural stability, walking, and changing direction do not, as yet, exist. The path control primitive is similar in this regard to the stepping reflex in human newborns; it exists at birth but is inhibited by a variety of mechanisms and becomes possible, both physically and computationally, only after supporting prerequisite skills are acquired and integrated.

Huber’s developmental approach involved the hierarchical programming of locomotor skills over a sequence of autonomous learning experiments that acquire implicit knowledge in the form of reusable skills. The goal was to evaluate how skill composition influences performance in exploration-based learning. Initially, the range of resources available is restricted and foundational skills are constructed from control basis primitives. After that, hybrid compositions of primitives and acquired skills extend the developmental frontier, structure continuing exploration, reuse acquired control knowledge, and address increasingly complex tasks.

11.3 Locomotion Controllers

The initial size of the landscape of attractors is determined by the number of resource parameterizations for potentials ϕ_m and ϕ_k , a total of $187+110=297$ possible choices from each state. If we assume binary control feedback $\in\{\text{!CONV}, \text{CONV}\}$, as proposed in [118, 117]), this control design could potentially yield 2^{297} states and, if k -way concurrency is permitted (section 10.1.4), then the number of actions is bounded by the sum of permutations,³

$$\sum_{i=1}^k P_i^{297} = P_1^{297} + P_2^{297} + \dots + P_k^{297}.$$

For $k=3$, this yields 26, 022, 249 possible concurrent actions accessible from each of the 2^{297} states! The resulting MDP is expressive enough to

represent a large variety of skills and tasks, but finding these solutions in such a large state-action space is extremely challenging by randomized exploration. Fortunately, many of these actions are unproductive; many states are unreachable or uncontrollable and should be eliminated to support safe exploration. Developmental constraints on state-action combinations can be used to create smaller search spaces on a skill-by-skill basis and, therefore, support autonomous and sample efficient reinforcement learning.

The Landscape of Stable Stances—*Thing* can execute four unique tripod stances. We assume that in each of these four cases, the robot can elect to apply ϕ_m -based moment residual control references to one of the three legs involved at a time. The effect is to improve the static stability of the tripod in question by placing a single leg optimally. This recruitment model yields 12 unique moment residual controllers—one for each of three controllable legs in each of the four unique tripods.

To manage periodic gaits involving four independent legs, a single kinematic conditioning controller based on the ϕ_k potential was incorporated into set Ψ at this stage. It addresses the condition of intermediate four-legged stances that occur during a gait by controlling the robot heading φ exclusively while foot placements remain fixed. The 12 moment residual controllers plus this single kinematic conditioning controller comprise a set of 13 unique primitive controllers for learning skills—restricting the range of sensor and motor resources biases the kinds of rewards accessible to the robot and, in this particular case, it emphasizes control sequences that cause rotation. Huber allowed up to three concurrent objectives to be addressed simultaneously in the experiments reported in [117], leading potentially to

$$P_1^{13} + P_2^{13} + P_3^{13} = 1885$$

permutations of the 13 primitive controllers.

11.3.1 Aggregate State Representation

To evaluate the functional condition of the four-legged robot, the status of the 13 control primitives is summarized using five Boolean membership functions γ (section 10.1.5), one signifying the convergence status of the moment residual controller for each of the four unique tripod stances—

regardless of the motor resources that were used to assert these conditions—and another indicating the status of the kinematic conditioning objective. Huber aggregated the feedback status of independent actions using * as a wildcard for the effector resource. If we write the control configurations in the form $\phi|_*^\sigma$ and define $\gamma(\phi_m|_*^\sigma) = (\gamma(\phi_m|_{\tau_i}^\sigma) \vee \gamma(\phi_m|_{\tau_j}^\sigma) \vee \gamma(\phi_m|_{\tau_k}^\sigma))$ —a disjunction over the three alternative τ resources for each tripod—then the state vector underlying this developmental context is written

$$[\gamma_0(\phi_M|_*^{012}) \ \gamma_1(\phi_M|_*^{023}) \ \gamma_2(\phi_M|_*^{123}) \ \gamma_3(\phi_M|_*^{013}) \ \gamma_4(\phi_K|_\varphi^{0123})], \quad (11.3)$$

resulting in 2^5 states, from which each of the 1885 concurrent primitive actions can be engaged.

Stability and Feasibility Constraints—Huber [118, 119] proposed an analog of native neurological structure in animals in the form of a Discrete Event Dynamic System (DEDS). In the DEDS formalism [219, 239, 263, 153], the state of the underlying system is assumed to evolve within a set of discrete events, some subset of which are controllable. In robotic systems, discrete event systems are used to regulate safety, kinodynamic conditioning, to guarantee real-time constraints and freedom from deadlock [121, 122, 120, 118, 119]. Constraints are represented as Boolean axioms that describe patterns of events that must either be preserved or excluded in the controlled system. This approach reduces the complexity of the state-action space significantly, eliminates catastrophic errors during exploration, and supports single trial learning algorithms.

Example: Logical Organization of Locomotor Skills

To guarantee that Thing will not fall over in the course of learning, Huber required that at least one moment residual controller was converged (yielding a stable stance) at all times. This specification was expressed as a logical disjunction over the four bits that describe the convergence status of the four distinct moment residual controllers. Given the state definition in [equation 11.3](#), this specification is written

$$\gamma_0 \vee \gamma_1 \vee \gamma_2 \vee \gamma_3. \quad (11.4)$$

Logical structure like this effectively reduces the number of legal actions available from each state. As Huber points out, concurrent moment residual controller $\phi_M|_0^{023} \triangleleft \phi_M|_2^{012}$, for example, can potentially modify all five predicates yielding the state $[0\ 0\ 0\ 0\ 0]$, where [equation 11.4](#) evaluates FALSE. Removing all such control compositions reduces the number of legal actions available per state from 1885 to 157—a great deal of exploration may be required to discover this kind of logical structure empirically. Moreover, observing constraints like [equation 11.4](#) supports autonomous exploration by avoiding conditions that might require costly external resets.

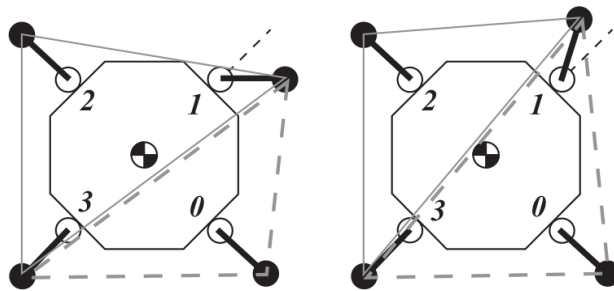


Figure 11.3

Two different three-legged stances that share legs 1 and 3. Only one of the two stances formed by legs $(l_1l_2l_3)$ and $(l_0l_1l_3)$ can be stable at a time.

Huber used a similar logical expression to capture kinematic structure specific to this legged platform and, thus, to make exploration and system identification more efficient. For Thing, it was observed that there are mutually exclusive stable stances ([figure 11.3](#)) when two different tripods share a pair of legs. This condition is expressed succinctly as a logical constraint on patterns of control equilibria.

$$\neg(\gamma_0 \wedge \gamma_2) \wedge \neg(\gamma_1 \wedge \gamma_3) \tag{11.5}$$

This kinematic constraint reduces the number of accessible states from 32 to 16 each with an average of about 50 control alternatives (down from 157) [[117](#)].

□

11.4 Learning the rotate Skill

Due in large part to the rotational symmetry of the robot, the first skill that *Thing* acquired was a policy for accumulating a heading change. The robot’s morphology and the structured control design were well suited to learning the ROTATE gait.⁴ The reward function is defined to be proportional to positive angular rotation of the robot’s heading.

$$R_k = \Delta\varphi = \varphi_k - \varphi_{k-1}$$

Index k designates consecutive values of the robot heading φ_k following convergence of action a_k . This provides the robot with immediate reward (that could be zero or negative) after each control decision.

Q-learning was used to approximate the optimal value function for rotate tasks subject to stability constraints. For each state-action combination, the value of that pair was approximated over the course of training (section 10.2.1) and stored in a lookup table. Learning performance was very repeatable—the same value function was computed over many separate experiments in a single, uninterrupted trial that lasted roughly 11 minutes. [Figure 11.4](#) shows a representative learning curve, and [figure 11.5](#) shows the transition dynamics of the newly acquired ROTATE skill. ROTATE incorporates 16 of the 32 observable states in the design. The four states in the central cycle constitute a four-legged rotation gait—the 12 other states occur stochastically during exploration and make the policy more robust to disturbances at run-time by restoring the system to the central cycle if conditions permit.

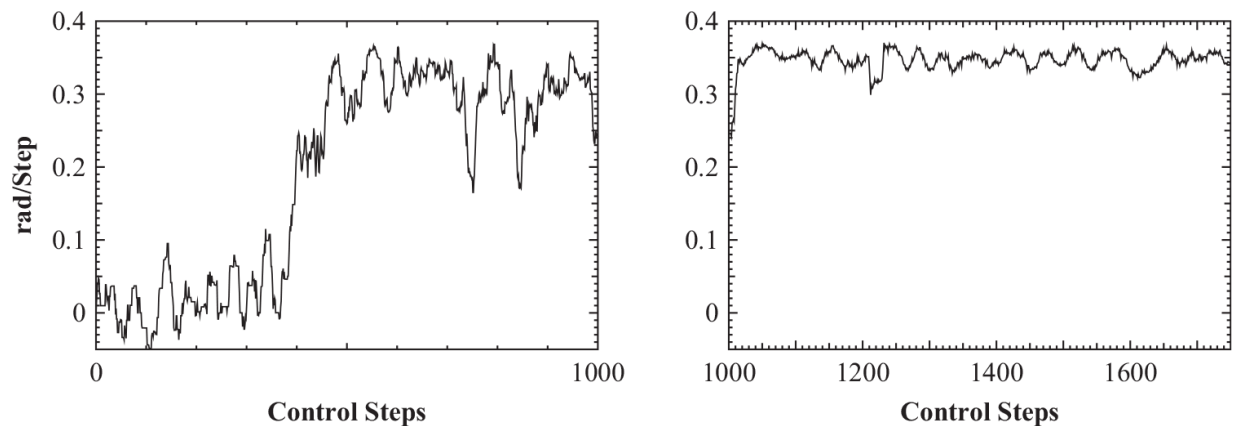


Figure 11.4

Learning performance for the ROTATE gait. (*Left*) A learning curve. (*Right*) Greedy performance.

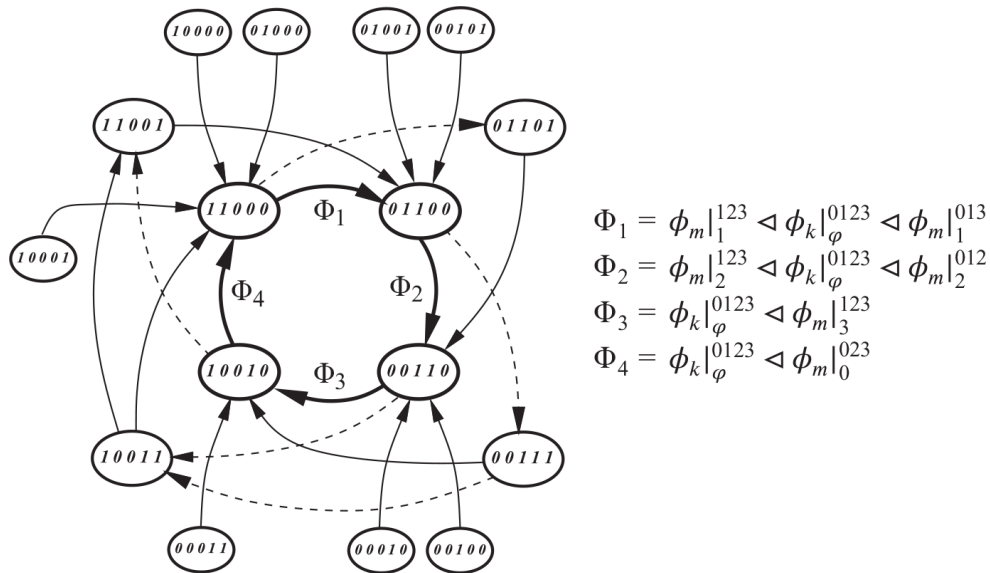


Figure 11.5

The ROTATE policy with contingencies for a variety of run-time contexts. The central cycle has transition probabilities of greater than 95% in the training context.

Figure 11.6 illustrates the policy for counterclockwise rotation showing the leg movements and net heading change accumulated by control action. It is the dual of figure 11.5 with actions at nodes and transitions triggered by events in the state representation. The figure omits the subordinate tertiary controllers included in figure 11.5 for actions Φ_1 and Φ_2 . Although they frequently show up in the learned policies, they are less consequential. Skills like ROTATE are, to some degree, error correcting toward a limit cycle, transferrable, and extensible (e.g., to other terrains), so they are attractive candidates for building blocks in a cumulative cognitive representation of the world.

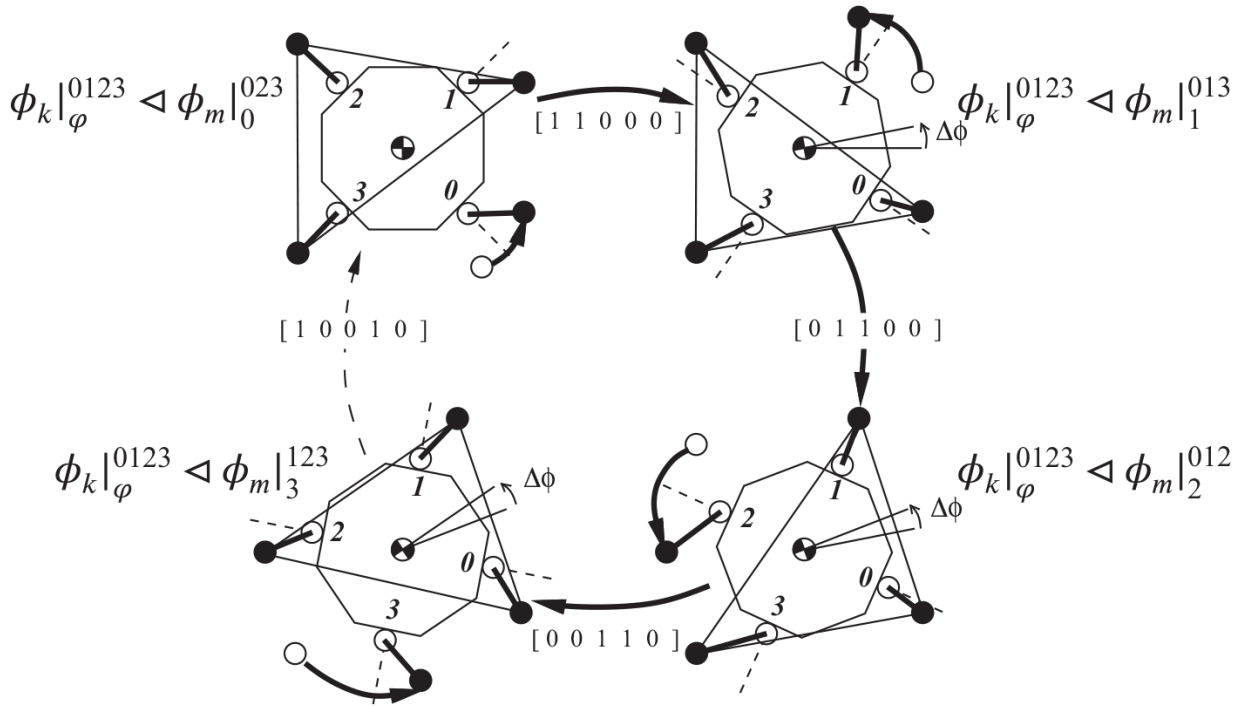


Figure 11.6

The 0-1-2-3 leg control sequence in the ROTATE policy accumulates approximately 45 [deg] per cycle

11.5 The STEP Skill

The STEP⁵ skill is used to coordinate a longitudinal stepping gait that accumulates translational movements of the platform along the initial heading while preserving at least one stable tripod at all times. The immediate reinforcement defined to achieve this goal rewards translational displacements along the current heading

$$R_k = \mathbf{h}^T \mathbf{d} - |\varphi_k - \varphi_{k-1}|,$$

where $\mathbf{h}^T = [\cos(\varphi_{k-1}) \sin(\varphi_{k-1})]$ is the current heading and $\mathbf{d}^T = [(x_k - x_{k-1}) (y_k - y_{k-1})]$ is the translational displacement in the ground plane. The second term in the reward is the same as the reward function for the ROTATE skill and thus encourages independent rotation and translation skills. Safety constraints and kinematic limitations remain as they were for the ROTATE skill.

Sensorimotor Recruitment—The resource model extended the model used in ROTATE by incorporating additional kinematic conditioning controllers

and longitudinal translation as an effector used to optimize range of motion in stable stances [117]. As a result, the upper bound on the number of states $|S|$ in the MDP is 2^8 (up from 2^5 for ROTATE) and the logical stability (equation 11.4) and kinematic feasibility (equation 11.5) constraints, in this context, restrict this design to 72 legal states and, with up to two-way concurrency, an average of 257 actions per state. This design has approximately four times the number of states and five times the average number of actions per state than the simpler ROTATE skill.

Two variants of the acquired STEP skill were learned in repeated experiments—a periodic strategy with approximately constant forward motion and a more complex, aperiodic strategy with time-varying forward velocity. The periodic policy was considered superior to the aperiodic variant because it produced more uniform translational progress with a significantly simpler policy. Figure 11.7 shows a typical learning curve for the periodic STEP skill. After approximately three hours of exploration (roughly 10, 000 control steps) in a single trial, the platform achieved roughly 0.0013 [m/action], where m is the projection of robot displacements in \mathbb{R}^2 along the current heading.

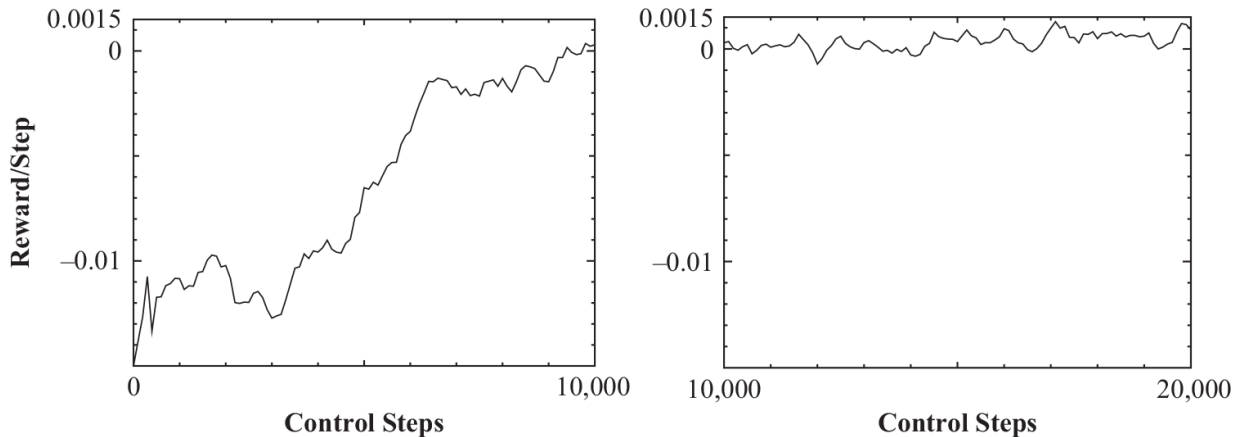


Figure 11.7

Performance of the STEP skill during learning (left) and the greedy performance (right) with exploration turned off

11.6 Hierarchical walk and navigate Skills

ROTATE and STEP represent coordinated motor patterns for creating displacements of the robot while preserving a stable stance. As with

parameterized primitive actions, these sequential skills are tied directly to the resources and morphology of the embodied system. They are also a functional abstraction that insulates subsequent learning and problem-solving from the details of low-level mobility and stability concerns.

The WALK⁶ skill is a hierarchical composition of control basis primitives together with the ROTATE and STEP skills. Goals in the neighborhood of the robot are defined as random offsets from the robot’s current position. To represent state, the run-time status of ROTATE and STEP are added to the state vector resulting in 2^{12} states and an average of 231 actions per state. The reward function for WALK favors sequences of actions that reduce the Cartesian distance to the goal,

$$r_k = d_{k-1} - d_k$$

where d_k is the x - y distance between the robot and the goal position following the convergence of action k .

The relative impact of the ROTATE and STEP skills on learning performance was evaluated by formulating three versions of the learning problem: (1) primitives alone (baseline); (2) primitives and the ROTATE skill; and (3) primitives with both ROTATE and STEP skills. Figure 11.8 compares the performance of the three WALK designs.

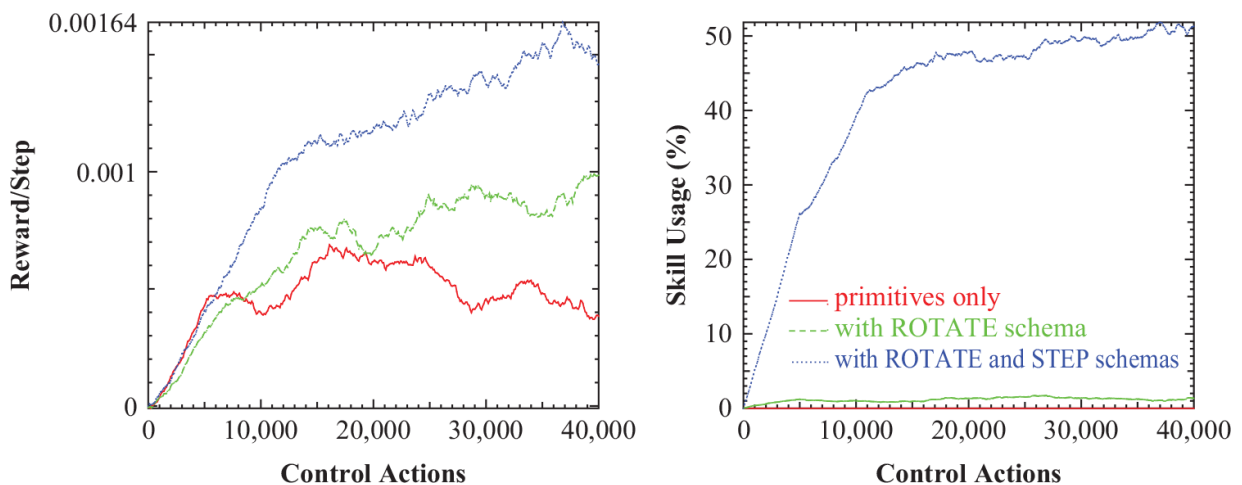


Figure 11.8

(Left) Performance of WALK skill learning in the three cases considered. (Right) Percentage of actions that were derived from ROTATE and STEP as opposed to primitives in each of the three problem designs.

The graph on the right of [figure 11.8](#) shows that when WALK engages primitives and ROTATE (case 2, green), the final skill only uses ROTATE approximately 2 percent of the time. Even this limited use, however, manages heading errors in the platform and improves performance significantly over case 1 (red) when primitive actions are used exclusively. When WALK uses primitives along with both ROTATE and STEP (case 3, blue), temporally extended skills are selected as optimal actions more than 50 percent of the time, indicating a significant reuse of the implicit knowledge in the ROTATE and STEP skills with primitives acting primarily to glue the skills together.

[Figure 11.9](#) illustrates several paths generated by the WALK skill from surrounding initial positions and randomized initial orientations. After roughly 10,000 actions (approximately 200 minutes of training), the performance of both larger problem designs exceeds the more economical base system that uses primitive actions exclusively. Moreover, although case 2 and case 3 designs yield approximately the same asymptotic performance after about 110000 actions (over 36 hours of training), the larger (case 3) design has significant advantages early in training. However, even after this extended period of training, the case 1 design (with access only to primitives) had significant difficulty achieving good performance on the WALK skill.

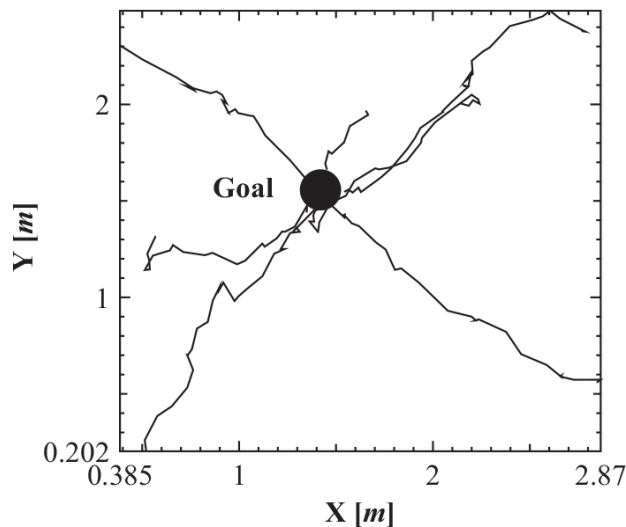


Figure 11.9

Trajectories toward $x=1.4$ [m] and $y=1.6$ [m] from a variety of initial positions. The final heading of the platform is not specified.

Huber concluded his experiments in hierarchical walking skills by constructing a hand-built NAVIGATE skill that could walk between start and goal states at some distance from one another given obstacles whose positions are unknown a priori. It incorporated versions of ROTATE, STEP, and WALK to implement rotation and translation gaits that also permitted a small range of crab angles. It also included the path control primitive introduced in section 11.2 to compute collision-free trajectories to the goal location. NAVIGATE detected environmental boundaries at run-time and integrated them into a spatially indexed occupancy grid. In this implementation, the numerical relaxation incorporates new boundaries and computes new harmonic potentials at about 5 [Hz].

Figure 11.10 is a demonstration of this integrated mobility controller [118]. The NAVIGATE skill employs a simple, three-state supervisory policy to traverse from the x in the right foreground to the x in the left background and avoids obstacles. Over time, the coarse geometry of the environment emerged in the occupancy grid in the left-hand column. The integrated mobility controller can solve many such mazes provided a path exists at the resolution of the occupancy grid [295]. The control knowledge encoded in this hierarchy of controllers is capable of creating motor plans that are not possible in flat control learning schemes. The contingencies built into each skill can be enhanced in every task that engages them and are activated as needed without the need to consider them directly while learning to compose the NAVIGATE skill.

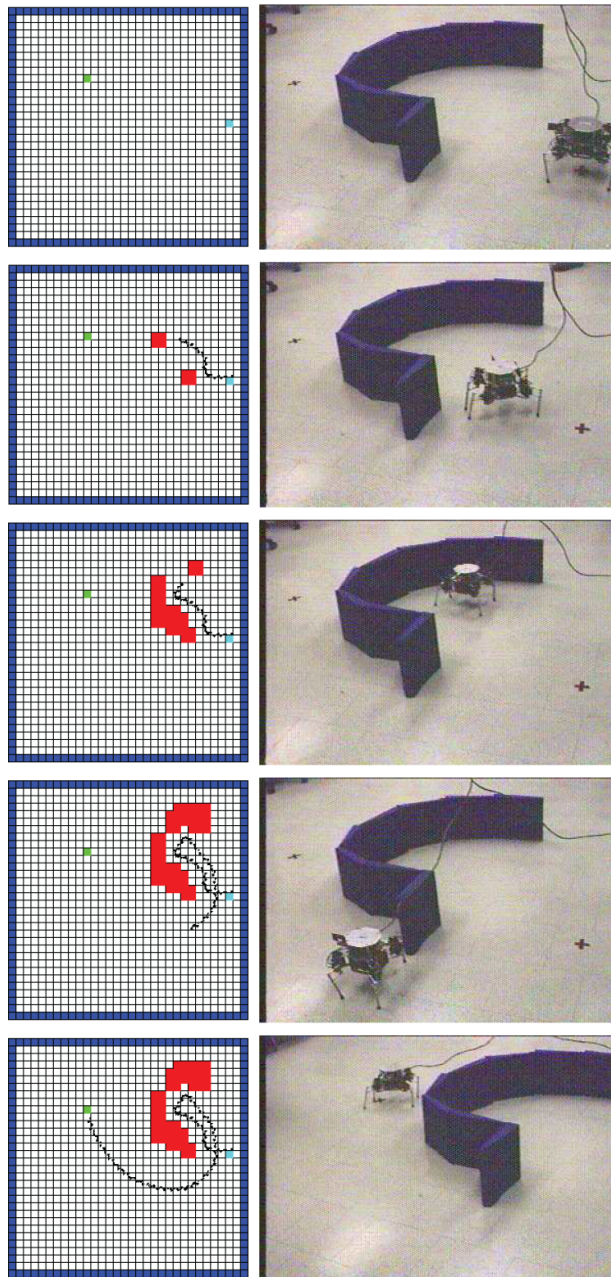


Figure 11.10
 Behavior generated by the integrated, hierarchical mobility controller for *Thing*

11.7 Developmental Performance: Hierarchical Gross Motor Skills

This case study examined a computational account of early development consisting of a native control stack (an analog of spinal reflexes), bootstrapped by tacit knowledge encoded in the sensory and motor

morphology and subject to a developmental curriculum for engaging resources in the maturing agent. The hypothesis, supported by experimental data, is that the newborn uses this structure to acquire hierarchical skills—implicit knowledge structures describing how environmental affordances align with kinodynamic aptitudes in the embodied agent—in an otherwise autonomous learning architecture.

A quadruped (*Thing*) was provided with a combinatorial basis for closed-loop interactions with the environment called the control basis and engaged in building a library of temporally-extended control skills. The data shows that native structure, a maturational schedule for engaging resources, and a recursive basis for control and abstraction is an effective means of controlling the incremental complexity of skill learning in high-dimensional systems.

The foundation of the hierarchical agent consists of a set native closed-loop actions A enumerated by combining sensory, motor, and computational resources using the control basis framework. This comprehensive set of potential actions is subject to morphological constraints to yield a set of exploratory actions A_0 . The developmental curriculum focuses exploration on $\Psi_{i+1} \subset A_i$ to discover sequential skills for controlling new aspects of the robot-world interaction. As these reusable skills are acquired, they are added to A_i and used to create a new developmental frontier Ψ to build new skills that extend the set of actions A_{i+1} over time.

Figure 11.11 summarizes the developmental schedule described in this section. The outcome of each stage is a new skill (underlined). In stage 1, the new ROTATE skill is learned using a constrained set of primitive actions and their associated states. This program is acquired reliably in about 11 [*min*] of training by virtue of the highly structured developmental context Ψ_1 . Much of the structure exploited in the design is used to guarantee that the robot can explore autonomously without making functional errors—in particular, *Thing* learns to walk without falling down. In stage 2, a slightly less restricted developmental context was used to learn the new STEP skill. STEP composes primitives to make forward progress along the current heading. Stage 3 was used to construct a skill called WALK that coordinates ROTATE and STEP skills along with primitive controllers to create quadrupedal gaits that move to arbitrary positions nearby. The performance

in the series of WALK designs suggests that the learning algorithm is successful at reusing prior control knowledge when it exists. Finally, a NAVIGATE skill integrates path controller ϕ_p with the WALK skill to execute a sequence of via points to circumvent obstacles and arrive at goal positions.

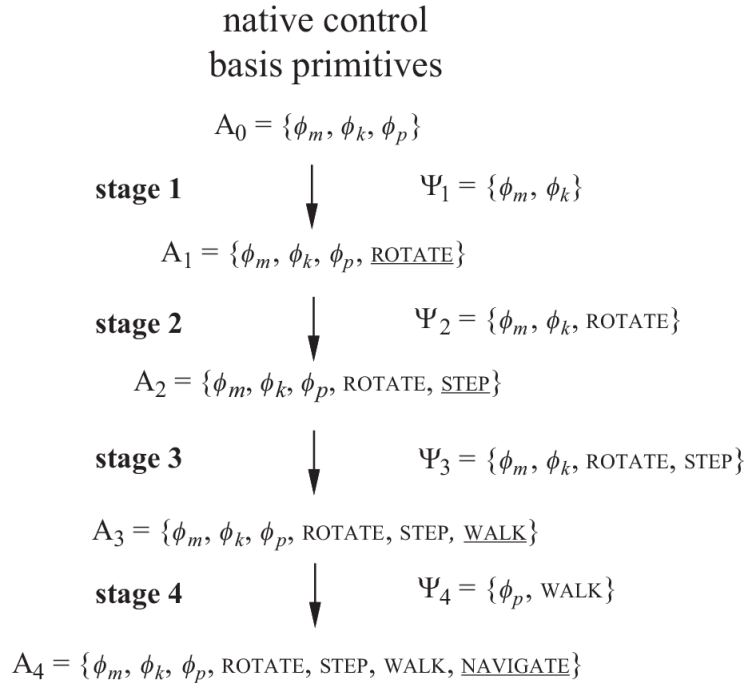


Figure 11.11

A summary of the developmental staging of an integrated navigation behavior. In stage 1, the acquired ROTATE policy achieves 0.35 [rad/action] in 11 [min]. In stage 2, STEP achieves 1.3 [mm/action] in 180 [min], and in stage 3, WALK achieves 1.53 [mm/action] in 480 [min].

Skills concerning flat terrain and constraints ensuring stability proved quite robust. Small changes in the environment occasionally generate unexpected transitions in the acquired skills, sometime visiting states that had been generated by stochastic exploration. If this was the case, then the learned policy may recover on its own. When environmental changes were more significant, these deviations may require additional learned contingencies. In subsequent work, MacDonald [122, 182] tested the proposition that skills represented in this form can be extended monotonically to address these contingencies without disrupting existing skills, as is the case with many behavior-based architectures. He found that incremental extensions of the skills presented in this chapter transferred

successfully to piecewise horizontal terrains. [Figure 11.12](#) shows a traversal that *Thing* executed using this extended WALK policy on such a terrain.

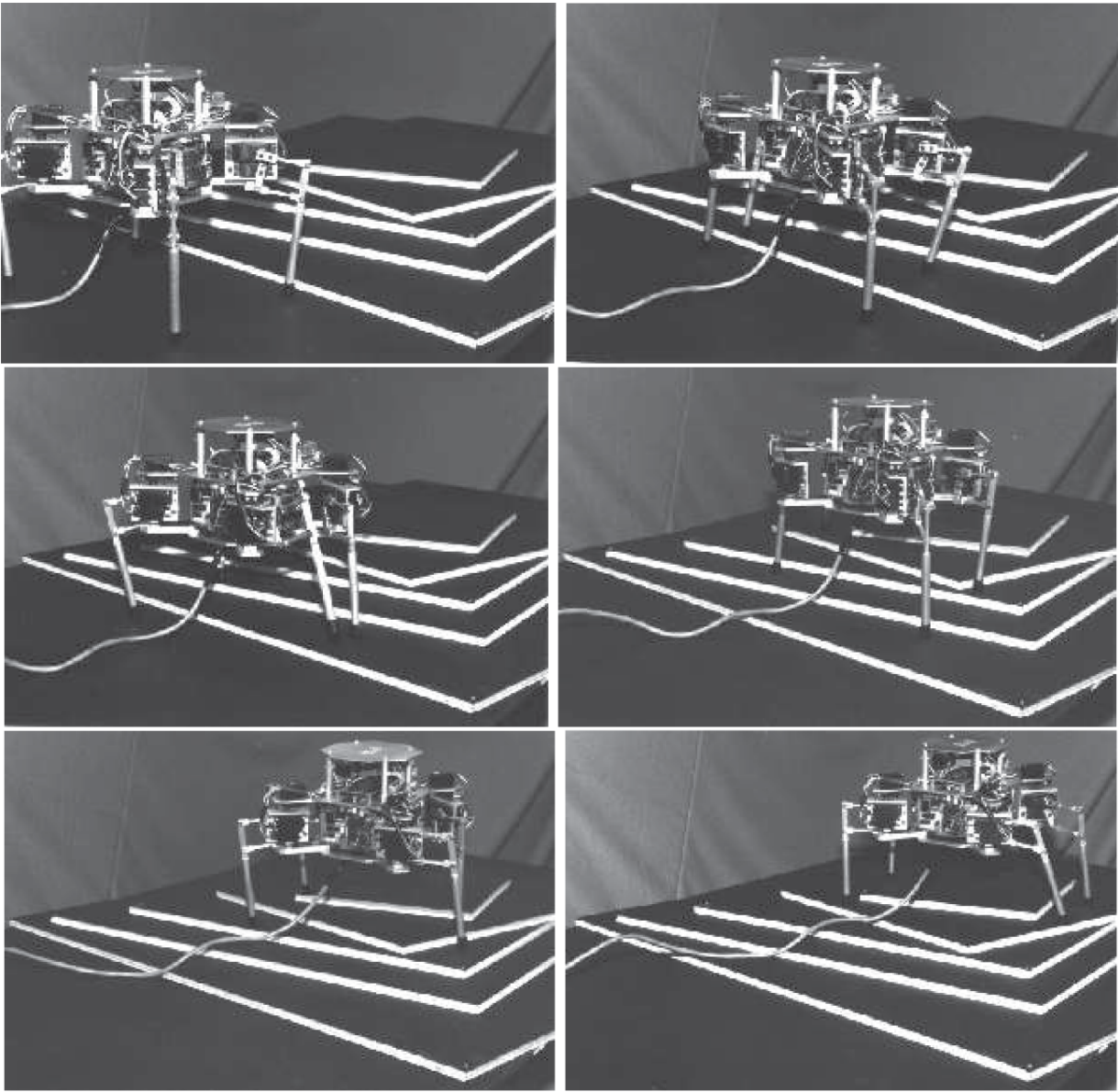


Figure 11.12
A demonstration of walking over irregular terrain. Steps (white) are 1 [cm] high.

Part IV Summary: Foundations for Hierarchical Skills

A great deal of the credit for patterns of growth and development in animals belongs to knowledge encoded tacitly in the infant's body. The infant human relies on reflexive structures to generate behavior and to shape the acquisition of higher-level (or more integrated) behavior. Thus, native reflexes help to defeat complexity as they help the infant learn what their bodies can do and how these skills are afforded by the world about them. Processes of discovery and growth are constantly extending sensory and motor skills and, with them, the complexity of the interactions that the growing child can explore.

Part IV considers how properties of sensorimotor machines introduced in earlier parts of the book lead efficiently to hierarchies of reusable skills. We begin with a description of hierarchy in the cerebral cortex and review a variety of low-level limbic, brainstem, bridge, and postural reflexes that structure and shape exploration during the first year in a human infant's life. Reflexes and the layers of integrated skills they support constitute a sensorimotor instruction set that *matures* over time via growth processes to control the latent complexity of interactions between the newborn and the world.

In chapter 9, important clinical observations regarding infant sensorimotor development are reviewed to inform the design of a computational framework that can be used in a robot. Rather than mimic the morphology of the infant or the sequence of milestones illustrated in [figure 9.18](#), the goal is a computational framework that can explain the milestones in the figure in the context of the infant human's morphology and developmental context.

The premise of chapter 10 is that the embodiment of an organism—its morphology (kinematics and dynamics), the type and geometrical distribution of sensors, aptitudes for precision, strength, and speed— influences the behavior acquired. The chapter introduces the foundations of a computational agent that can exploit opportunities encoded tacitly in the embodied robot system while controlling the incremental complexity of learning in unstructured environments.

The *control basis* framework was introduced to provide a computational analog for the native reflexes in animals that play an important role in cumulative learning and development. The architecture expresses control interactions in a landscape of attractors parameterized by a nonstationary set of sensor and motor resources. A developmental curriculum in this framework is used to support experimentation with the reflexes and maturational mechanisms presented in chapter 9. A taxonomy of closed-loop actions consisting of POSTURE, TRACK, and SEARCH actions distinguished by the source of stimuli. We illustrated a handcrafted example of a single skill in the “barcode reader” and conclude part IV with a longitudinal example of hierarchical skill development for quadruped walking and navigation (originally published by Huber [118]) that demonstrates the leverage that hierarchies of implicit knowledge (skills) lend to performance on subsequent learning tasks.

1. Combinations of n elements of a set chosen r at a time (aka “ n choose r ”) is computed as follows:

$$C_r^n = \frac{n!}{r!(n-r)!}.$$

2. Huber used a combination of the manipulability potential (equation 10.12) and joint range of motion constraints. Similar behavior is observed with the simpler quadratic formulation presented in equation 11.2, which we use here in order to introduce a useful ROM potential.
3. Permutations of n elements of a set chosen r at a time is computed $P_r^n = \frac{n!}{(n-r)!}$.
4. This presentation departs from the terminology in [117] by using ROTATE to describe the optimal, sequential policy that Huber called the “turning gait.”
5. Here, the STEP skill denotes the optimal, sequential policy that Huber called the *straight line gait* [117].
6. WALK is equivalent to Huber’s *goal directed walking gait* in [117].

A

Tools for Linear Analysis

A.1 Linear Algebra

Elements of the set of real numbers \mathbb{R} span a one-dimensional, linear subset of Euclidean space. Suppose that two such sets, S_1 and S_2 , describe independent one-dimensional spaces. Together, these sets span (or form a basis for) \mathbb{R}^2 so that every element of \mathbb{R}^2 can be represented by a unique pair of values in the set $S_1 \times S_2$. If we denote the *directions* of the lines in \mathbb{R}^2 derived from S_1 and S_2 by \hat{v}_1 and \hat{v}_2 , respectively, then elements of \mathbb{R}^2 can be expressed as $\mathbf{a} = a_1\hat{v}_1 + a_2\hat{v}_2$. Direction vectors \hat{v}_1 and \hat{v}_2 have unit length (or *magnitude*) (indicated by the “hat” over the variable) in \mathbb{R}^2 and vectors \mathbf{a} can have arbitrary magnitude. The magnitude of a vector is the Euclidean length of the vector and generalizes to \mathbb{R}^n in a straightforward manner.

Definition A.1: Vector magnitude. A vector $\mathbf{a} = [a_1 \ a_2 \ \cdots \ a_n] \in \mathbb{R}^n$ has magnitude $\|\mathbf{a}\|$. Vectors with unit magnitude are called unit vectors.

Definition A.2: Unit vector. Unit vector $\hat{\mathbf{a}} = [a_1/\|\mathbf{a}\| \ a_2/\|\mathbf{a}\| \ \cdots \ a_n/\|\mathbf{a}\|]$ is parallel to the vector \mathbf{a} and has unit length ($\|\hat{\mathbf{a}}\|=1$).

Basis vectors are *linearly independent* if it is not possible to generate \hat{v}_i from weighted combinations of \hat{v}_j , $j \neq i$. Definition A.3 states this requirement another way.

Definition A.3: Linear independence. $a_1\hat{v}_1 + a_2\hat{v}_2 + \cdots + a_n\hat{v}_n = \mathbf{0}$ if and only if $a_1 = a_2 = \cdots = a_n = 0$.

There can be many linearly independent bases for a vector space. However, as the definition implies, each will have the same number of independent basis vectors equal to the dimension of the vector space—to measure distance in Euclidean geometry, an *orthonormal* basis is required.

By convention, we denote all vectors \mathbf{v} as column vectors and their transpose \mathbf{v}^T as row vectors. Then, a mathematical projection operator called the scalar product is defined:

Definition A.4: Scalar (dot) product. For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, the scalar product is written

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n.$$

The scalar projection defined by definition A.4 has a geometrical interpretation in \mathbb{R}^n as well.

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\theta)$$

If \mathbf{a} and \mathbf{b} are both unit length, then the inner product yields the cosine of the angle between $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$.

The column vectors of $\mathbf{V} = [\hat{\mathbf{v}}_1 \hat{\mathbf{v}}_2 \dots \hat{\mathbf{v}}_n]$ constitute an *orthonormal* basis set for \mathbb{R}^n if the \mathbf{v}_i are unit length and mutually *orthogonal*.

Definition A.5: Orthogonality. For vectors $\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_j \in \mathbb{R}^n$, $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$ are orthogonal if and only if $\hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_j = 0$.

Definition A.5 is equivalent to requiring that $\hat{\mathbf{v}}_i$ does not *project* onto the linear subspace spanned by $\hat{\mathbf{v}}_j, \forall j \neq i$.

There are many uses for the scalar product. One that comes up often in chapters 3, 4, and 6 is the squared magnitude of a vector quantity, as for instance in the kinetic energy of a particle $T = \frac{1}{2} m (\mathbf{v}^T \mathbf{v})$, which is a scalar product of the particle's velocity vector weighted by the particle's mass. Thus, kinetic energy is a scalar quantity and has no direction per se, whereas momentum, $m\mathbf{v}$, is a vector quantity. Work, $W = \int \mathbf{f}^T d\mathbf{s}$, is an important scalar quantity computed as the integral of the scalar product of force and displacement vectors.

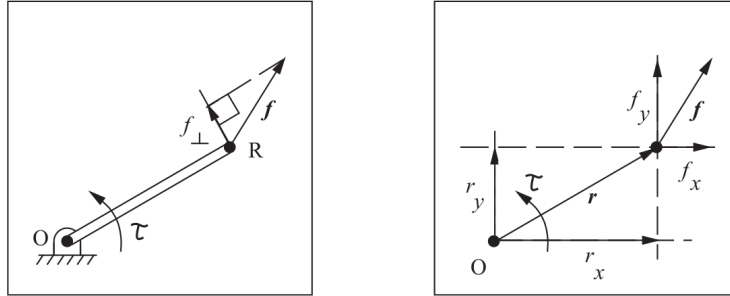


Figure A.1

The geometrical interpretation of the vector product

The torque due to an applied force acting through a lever is another form of vector transformation that we see in chapter 6. In [figure A.1](#), a point R , located at position \mathbf{r} in the x - y plane, experiences a force, \mathbf{f} , that creates a torque, $\boldsymbol{\tau}$, about an axis that is perpendicular to the plane and passes through the center of rotation, O . It is easy to see that the radial component of \mathbf{f} creates no torque around O , so that it is only the perpendicular component, \mathbf{f}_\perp , that contributes to the torque. Alternatively, we can see that the \hat{y} component of \mathbf{f} creates a torque in the same direction as $\boldsymbol{\tau}$ of magnitude $f_y r_x$, and the \hat{x} component of \mathbf{f} creates torque $f_x r_y$ in the opposite direction. Therefore, we find that the net torque is expressed $\tau_z = f_y r_x - f_x r_y$. The torque is subscripted z because it was derived from vectors in the x - y plane and rotates particle R about the \hat{z} -axis through O . This geometric intuition holds in three dimensions as well as the plane, and the result becomes

$$\tau_x = r_y f_z - r_z f_y$$

$$\tau_y = r_z f_x - r_x f_z$$

$$\tau_z = r_x f_y - r_y f_x.$$

This system of equations can be written in terms of the product of a matrix operator that defines the *cross* (or *vector*) product. Note that a vector crossed with a vector is another vector, unlike the scalar (dot) product.

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \mathbf{r} \times \mathbf{f}$$

Definition A.6: Vector (cross) product in \mathbb{R}^3 .

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \det \begin{bmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix},$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are the unit basis vectors for the first, second, and third components of vectors \mathbf{a} and \mathbf{b} , respectively. The cross product of two, non-parallel vectors creates a third vector that is mutually orthogonal to the two original vectors— $\mathbf{a} \times \mathbf{b} = \mathbf{0}$ is a sufficient condition to establish that \mathbf{a} and \mathbf{b} are parallel.

Alternatively, the cross product of vectors \mathbf{a} and \mathbf{b} can be written $\mathbf{a} \times \mathbf{b} = (\|\mathbf{a}\| \|\mathbf{b}\| \sin(\theta)) \hat{\mathbf{n}}$, where θ is the angle between \mathbf{a} and \mathbf{b} measured in the plane formed by the vectors, and $\hat{\mathbf{n}}$ is the unit normal to this plane. However, there are two choices for this normal, $\hat{\mathbf{n}}$ and $-\hat{\mathbf{n}}$. The sign is determined by employing a *right-handed* convention for coordinate systems. The right-handed coordinate frame obeys the commutative relations involving the vector product:

$$\hat{\mathbf{x}} \times \hat{\mathbf{y}} = \hat{\mathbf{z}}$$

$$\hat{\mathbf{y}} \times \hat{\mathbf{z}} = \hat{\mathbf{x}}$$

$$\hat{\mathbf{z}} \times \hat{\mathbf{x}} = \hat{\mathbf{y}}.$$

Finally, the rules for multiplication using vector and scalar products are not naive generalizations of rules for multiplication of scalars. Some important rules for vector operators include

$$\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$$

$$(\alpha \mathbf{a}) \times \mathbf{b} = \alpha (\mathbf{a} \times \mathbf{b})$$

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$$

$$\mathbf{a} \times \mathbf{a} = \mathbf{0}$$

$$\mathbf{a} \cdot (\mathbf{a} \times \mathbf{b}) = 0.$$

The outer product maps two vectors in \mathbb{R}^n into a $\mathbb{R}^{n \times n}$ matrix, sometimes known as the *covariance* of the two vectors.

Definition A.7: Outer product. Given $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$,

$$\mathbf{a}\mathbf{b}^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} [b_1 \ b_2 \ \cdots \ b_n] = \begin{bmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & \cdots & \vdots \\ a_n b_1 & a_n b_2 & \cdots & a_n b_n \end{bmatrix}. \quad (\text{A.1})$$

A.2 Matrix Inverse

The determinant of a square matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$, is an important scalar measure of the *condition* of a matrix. In chapter 4, we use the determinant as part of an analysis of the kinematic acuity of spatial transformations.

Definition A.8: Determinant.

$$\det(\mathbf{A}) = \sum_{j=1}^n a_{ij} \Delta_{ij},$$

where Δ_{ij} is the ij -cofactor of matrix \mathbf{A} . To compute Δ_{ij} , we construct matrix \mathbf{A}_{ij} by removing the i th row and j th column from \mathbf{A} , then

$$\Delta_{ij} = (-1)^{i+j} \det(\mathbf{A}_{ij}).$$

Definition A.9: Inverse. The inverse of square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is written \mathbf{A}^{-1} and is defined by the relation $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, where \mathbf{I} is the $n \times n$ identity matrix.

To compute the matrix inverse, an $n \times n$ matrix of the cofactors of \mathbf{A} is defined

$$\text{cof}(\mathbf{A}) = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \cdots & \Delta_{1n} \\ \Delta_{21} & \Delta_{22} & \cdots & \Delta_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta_{n1} & \Delta_{n2} & \cdots & \Delta_{nn} \end{bmatrix}.$$

The adjoint matrix of \mathbf{A} is defined to be the transpose of the matrix of cofactors

$$\text{adj}(\mathbf{A}) = (\text{cof}(\mathbf{A}))^T.$$

Now, if $\det \mathbf{A} \neq 0$, we can define a unique inverse for the square matrix

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \text{adj}(\mathbf{A}).$$

A.3 Definiteness

Throughout the text, the positive/negative definiteness of a matrix is used to establish other important properties in functions, mechanisms, and in the stability of dynamical systems—including those that employ feedback controllers. Definiteness is defined formally using the following conditions.

Definition A.10: Definiteness properties. *Definiteness properties for matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ are established by checking the validity of the following assertions:*

positive definite	<i>if and only if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0$</i>
positive semidefinite	<i>if and only if $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \quad \forall \mathbf{x} \neq 0$</i>
negative definite	<i>if and only if $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0 \quad \forall \mathbf{x} \neq 0$</i>
negative semidefinite	<i>if and only if $\mathbf{x}^T \mathbf{A} \mathbf{x} \leq 0 \quad \forall \mathbf{x} \neq 0$.</i>

The following three theorems can be used to establish the definiteness of a square matrix \mathbf{A} . Without loss of generality, we will only state the theorems in the form suitable for establishing positive definiteness.

Theorem A.3.1. *Symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if and only all the eigenvalues of \mathbf{A} are positive.*

Sylvester's criterion can also be used to establish positive definiteness. We define the i th leading principal minor of \mathbf{A} , \mathbf{A}_i , to be the first i rows and columns of \mathbf{A} .

Theorem A.3.2 (Sylvester's theorem). *Symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if and only if $\det \mathbf{A}_i > 0$, $i=1, \dots, n$.*

Square matrices that are not symmetric can still be positive definite.

Theorem A.3.3. *Square nonsymmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if and only if $(\mathbf{A} + \mathbf{A}^T)/2$ is positive definite.*

A.4 Hessian

The behavior of compensated dynamical systems is analyzed in several discussions throughout the book. A useful analogy of such systems considers the inertial system in the form of a particle that rolls over a potential surface. Properties of this surface can be used to establish important features of real systems, including their asymptotic stability.

Some aspects of the *shape* of a function can be characterized by examining the Hessian of the function. Consider a generic multivariable nonlinear function, $f(q_0, q_1, \dots, q_n)$, defined on some compact domain $\mathbf{Q} \in \mathbb{R}^n$.

Definition A.11 *The Hessian of function $f(\cdot)$, $\partial^2 f / \partial \mathbf{q}^2$, is defined by a matrix of partial second derivatives,*

$$\frac{d^2 f}{d\mathbf{q}^2} = \begin{bmatrix} \frac{\partial^2 f}{\partial q_1^2} & \frac{\partial^2 f}{\partial q_1 \partial q_2} & \cdots & \frac{\partial^2 f}{\partial q_1 \partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial q_n \partial q_1} & \frac{\partial^2 f}{\partial q_n \partial q_2} & \cdots & \frac{\partial^2 f}{\partial q_n^2} \end{bmatrix}.$$

Definition A.12: Convex. *If the Hessian is positive semidefinite over the domain \mathbf{Q} , then the function $f(\cdot)$ is convex over \mathbf{Q} .*

Definition A.13: Harmonic. *If the trace of the Hessian (Laplacian) is identically zero—that is,*

$$\nabla^2 f = \frac{\partial^2 f}{\partial q_0^2} + \frac{\partial^2 f}{\partial q_1^2} + \cdots + \frac{\partial^2 f}{\partial q_n^2} = 0,$$

then function $f(\cdot)$ is harmonic.

Definition A.14: Subharmonic. *If the trace of the Hessian is negative semidefinite—that is,*

$$\nabla^2 f = \frac{\partial^2 f}{\partial q_0^2} + \frac{\partial^2 f}{\partial q_1^2} + \cdots + \frac{\partial^2 f}{\partial q_n^2} \leq 0,$$

then function $f(\cdot)$ is subharmonic.

Each of these properties implies that a greedy descent (ascent) of the function $f(\cdot)$ will eventually reach an extremum in the field. In chapters 3 and 10, we see that constraints on the shape of functions like those discussed here can be used to see how the state of dynamical systems will evolve as $t \rightarrow \infty$.

A.5 Matrix Norms

It is often necessary to measure the distance between vectors of matrices or to characterize their magnitude. There are many choices for distance measures that all share the basic properties of a valid metric space.

Definition A.15: Distance metrics. *A valid distance metric can be any real-valued function such that*

1. $\|\mathbf{A}\| \geq 0$
2. $\|\mathbf{A}\| = 0$ iff $\mathbf{A} = \mathbf{0}$ identically
3. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$ $\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$
4. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
5. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$

For example, the magnitude of a vector $\mathbf{v} = [xy] \in \mathbb{R}^2$ defined using the l_2 norm $\|\mathbf{v}\|_2 = \sqrt{x^2 + y^2}$ satisfies these conditions as does the l_∞ norm of a matrix $\|\mathbf{A}\|$. Both of these choices can be used to measure distance in vectors and linear transforms.

A.6 Quadratic Forms

Linear transform $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be described by examining how sets of inputs map to sets of outputs. The mapping process reveals how matrix \mathbf{A} deforms one vector space into another and can be used to determine how constraints in \mathbf{x} map to constraints in \mathbf{y} . For example, in chapter 4, linearized kinematic mappings that depend on robot posture project force and precision from configuration space to Cartesian space and, thus, speak directly to the ability of the robot to address tasks in the world [208, 298].

To visualize the mapping, consider a set \mathcal{U} of input vectors $\mathbf{x} \in \mathbb{R}^m$ that define a (hyper)sphere of radius $\|\mathbf{x}\|^2 = x_1^2 + x_2^2 + \dots + x_m^2 \leq k^2$. Set \mathcal{U} is

deformed in the process of mapping through the $n \times m$ transform \mathbf{A} to produce an n -dimensional hyperellipsoid in the output space.

$$\mathbf{x}^T \mathbf{x} = (\mathbf{A}^{-1} \mathbf{y})^T (\mathbf{A}^{-1} \mathbf{y}) = \mathbf{y}^T [(\mathbf{A}^{-1})^T \mathbf{A}^{-1}] \mathbf{y} = \mathbf{y}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{y} \leq k^2 \quad (\text{A.2})$$

The *quadratic form* $\mathbf{y}^T \mathbf{M}^{-1} \mathbf{y}$ defines a (hyper)ellipsoidal set $\mathcal{E} = \{ \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y}^T \mathbf{M}^{-1} \mathbf{y} \leq k^2 \}$ centered on the origin for the positive definite and symmetric matrix $\mathbf{M}^{-1} \in \mathbb{R}^{n \times n}$. The shape of this ellipse reveals the anisotropic character of the linear transformation.

Example: Plotting the Quadratic Form

Consider $\mathbf{x}^T \mathbf{x} = \mathbf{y}^T \mathbf{M}^{-1} \mathbf{y} \leq k^2$, where $\mathbf{M}^{-1} = \begin{bmatrix} 3/2 & -1/2 \\ -1/2 & 3/2 \end{bmatrix}$. This quadratic yields the equation of the output ellipsoid:

$$[y_1 \ y_2] \begin{bmatrix} 3/2 & -1/2 \\ -1/2 & 3/2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{3}{2} y_1^2 - y_1 y_2 + \frac{3}{2} y_2^2.$$

The *principal axes* of the output set are computed by solving for the eigenvalues $\{\lambda_1, \lambda_2\}$ and eigenvectors $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\}$ of \mathbf{M}^{-1} , where unit length eigenvectors $\hat{\mathbf{e}}_i$ determine the orientation of the ellipse and eigenvalues λ_i determine the length of the corresponding axes. The eigenvalues of \mathbf{M}^{-1} are found by solving for the roots, λ , of the characteristic polynomial.

$$\det \begin{bmatrix} (3/2 - \lambda) & -1/2 \\ -1/2 & (3/2 - \lambda) \end{bmatrix} = 0 \quad (\text{A.3})$$

$$(3/2 - \lambda)(3/2 - \lambda) - 1/4 = 0$$

$$\lambda^2 - 3\lambda + 2 = 0$$

$$\lambda_{1,2} = 1, 2$$

The eigenvectors are determined by substituting the resulting eigenvalues back into [equation A.3](#) and solving for the corresponding direction in the output space.

for $\lambda_1 = 1$: $\begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, so that $\hat{e}_1 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$, and

for $\lambda_2 = 2$: $\begin{bmatrix} -1/2 & -1/2 \\ -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, so that $\hat{e}_2 = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}$.

The diagonalized \mathbf{M}^{-1} matrix represents the quadratic form in terms of the eigenvector basis where elements of the output set are written $\mathbf{y} = e_1 \hat{e}_1 + e_2 \hat{e}_2$ and $\mathbf{e} = [e_1 \ e_2]^T$.

$$\mathbf{x}^T \mathbf{x} = \mathbf{y}^T \begin{bmatrix} 3/2 & -1/2 \\ -1/2 & 3/2 \end{bmatrix} \mathbf{y} = \mathbf{e}^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{e} \leq k^2$$

The boundary of set $\mathcal{E} = \{ \mathbf{y} \mid \mathbf{y}^T \mathbf{M}^{-1} \mathbf{y} \leq k^2 \}$ is defined by the equality $\mathbf{y} = \lambda_1 e_1^2 + \lambda_2 e_2^2 = k^2$. Therefore,

when $e_2 = 0$, $\lambda_1 e_1^2 = k^2$, and $e_1 = k/\sqrt{\lambda_1}$, and

when $e_1 = 0$, $\lambda_2 e_2^2 = k^2$, and $e_2 = k/\sqrt{\lambda_2}$.

This ellipsoid is illustrated in [figure A.2](#) for $k = 1$.

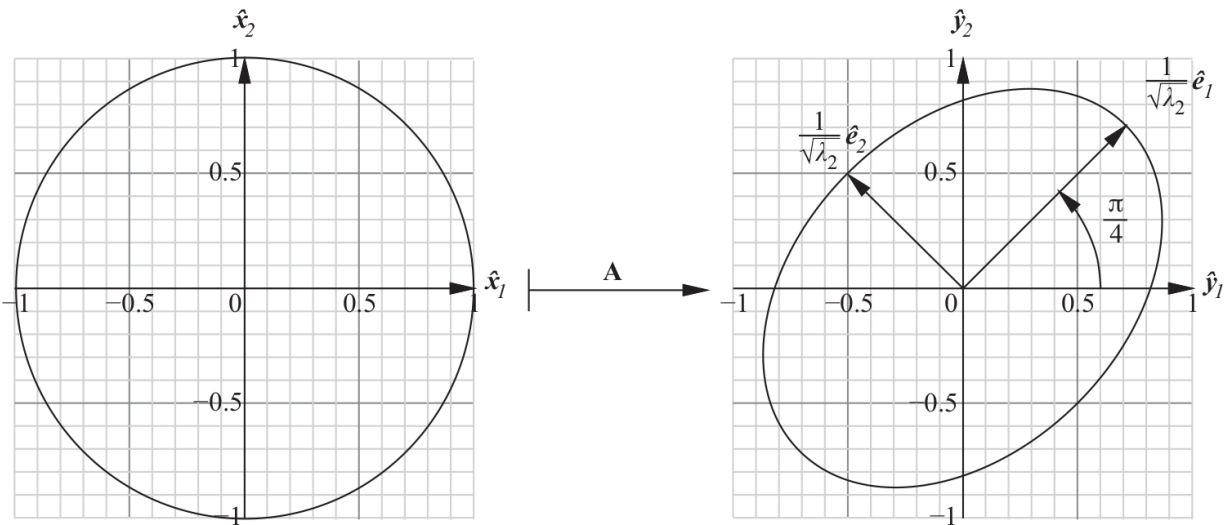


Figure A.2

A set of inputs on the left, $\mathbf{x} \in \mathcal{U}$, maps to the set of outputs on the right, $\mathbf{y} \in \mathcal{E}$, via the linear mapping $\mathbf{y} = \mathbf{A}\mathbf{x}$

The eigenvectors of $\mathbf{M}^{-1} = (\mathbf{A}\mathbf{A}^T)^{-1}$ are equivalent to those for \mathbf{M} . Moreover, if the eigenvalues of \mathbf{M}^{-1} are (λ_1, λ_2) , then the eigenvalues of \mathbf{M} are $((\lambda_1^* = 1/\lambda_1, \lambda_2^* = 1/\lambda_2))$. This observation has some practical significance because it eliminates the need to invert \mathbf{M} when plotting the quadratic form. The magnitude of the principal axes of the quadratic form representing the mapping $\mathbf{x} = \mathbf{A}\mathbf{y}$ (figure A.2) are, therefore, $k/\sqrt{\lambda_1} = k\sqrt{\lambda_1^*}$ in the \hat{e}_1 direction and $k/\sqrt{\lambda_2} = k\sqrt{\lambda_2^*}$ in the \hat{e}_2 direction.

□

A.7 Singular Value Decomposition

The singular value decomposition (SVD) provides the most significant decomposition of matrices [208] and yields the most direct insight into general (nonsquare) linear transformations. We consider the general linear mapping $\mathbf{y} = \mathbf{A}\mathbf{x}$, which transforms an input space, $\mathbf{x} \in \mathbb{R}^n$, to an output space, $\mathbf{y} \in \mathbb{R}^m$, using matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. The quadratic construction $\mathbf{A}^T\mathbf{A}$ is a positive semidefinite matrix whose eigenvalues (solutions to $\det(\lambda\mathbf{I}_n - \mathbf{A}^T\mathbf{A}) = 0$) are greater than or equal to zero. We define the *singular values* of \mathbf{A} , $\sigma_i = \sqrt{\lambda_i}$, which are sorted into descending order, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. Matrix \mathbf{A} has a rank $k \leq p = \min(m, n)$, which is defined by the number of nonzero singular values.

Definition A.16: SVD. For $\mathbf{A} \in \mathbb{R}^{m \times n}$, there exist orthogonal matrices

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathbb{R}^{m \times m}, \quad \mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n}, \quad \text{and} \quad \mathbf{\Sigma} \in \mathbb{R}^{m \times n}$$

such that

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T.$$

There are two possible cases to consider: (1) $m \geq n$, in which some outputs in \mathbb{R}^m are, in general, not achievable; and (2) $m < n$, in which there may be many inputs that map to the same output. These cases lead to two diagonal forms for $\mathbf{\Sigma}$:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & & \sigma_{p=n} \\ \hline & & & \\ & & & \mathbf{0} \\ & & & \end{bmatrix} \quad m \geq n$$

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \mathbf{0} \\ & \ddots & & \\ \mathbf{0} & & & \sigma_{p=m} \\ \hline & & & \\ & & & \mathbf{0} \\ & & & \end{bmatrix} \quad \begin{array}{l} \text{redundant} \\ m < n \end{array}$$

Methods exist for efficiently computing orthogonal matrices \mathbf{U} and \mathbf{V} given Σ [96]. Since \mathbf{U} and \mathbf{V} are orthogonal, they satisfy

$$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_m \quad \text{and} \quad \mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}_n.$$

Following the discussion in Nakamura [208] and Yoshikawa [298], if we define $\mathbf{y}^* = \mathbf{U}^T \mathbf{y}$ and $\mathbf{x}^* = \mathbf{V}^T \mathbf{x}$, then we can rewrite the original transformation $\mathbf{y}^* = \Sigma \mathbf{x}^*$. This form makes it clear that the transformation is composed of three steps; a rotation from \mathbf{x} to \mathbf{x}^* that preserves length, a directionally dependent scaling by the σ_i from \mathbf{x}^* to \mathbf{y}^* , and another length preserving rotation from \mathbf{y}^* to \mathbf{y} .

Definition A.17: Span and nullspace. *The singular value decomposition of matrix \mathbf{A} provides a convenient definition of the span and the nullspace of the transformation.*

1. $\mathcal{R}(\mathbf{A}) = \text{span}[\mathbf{u}_1, \dots, \mathbf{u}_k]$, where the $\mathbf{u}_i \in \mathbb{R}^m$, $i=1, k$ are the first k column vectors of the \mathbf{U} matrix and define the singular vectors (the principal axes) of the transform \mathbf{A} .
2. $\mathcal{N}(\mathbf{A}) = \text{span}[\mathbf{v}_{k+1}, \dots, \mathbf{v}_n]$, where the $\mathbf{v}_i \in \mathbb{R}^n$, $i = k+1, n$ are the last $(n-k)$ column vectors of the \mathbf{V} matrix and define an orthonormal basis for the nullspace of matrix \mathbf{A} —the input subspace that produces no net output (the homogeneous part of a solution for \mathbf{y}).

On the basis of the singular value decomposition, other important properties of matrix \mathbf{A} can be computed easily.

Definition A.18: Determinant. *The product of singular values is the determinant, $\det(\mathbf{A}) = \prod_{i=1}^p \sigma_i$.*

Definition A.19: Condition number. *The condition number of matrix \mathbf{A} is the ratio of the largest singular value to the smallest, $\kappa = \sigma_1 / \sigma_p$.*

A.8 Scalar Condition Metrics for Linear Transforms

Tools for linear analysis include a number of scalar metrics that provide insight into the quality of the transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$ [298, 208]. Often the measure of quality in linear transformations is described in terms of their *singular values* and *singular vectors*. The principal axes of the conditioning ellipsoid ($\mathbf{A}\mathbf{A}^T$) are the singular vectors of \mathbf{A} , and the degree of amplification in these directions are proportional to the corresponding singular values. Section A.7 presents a brief description of the singular value decomposition. For our purposes, however, it is sufficient to note that the singular vectors of \mathbf{A} are the eigenvectors of $\mathbf{A}\mathbf{A}^T$, and the singular values of \mathbf{A} are the square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^T$. Our summary here assumes that the ellipsoid is expressed in terms of its singular values $[\sigma_1 \ \sigma_2 \ \cdots \ \sigma_m]$ of \mathbf{A} (or the square root of the eigenvalues, $[\sqrt{e_1} \ \sqrt{e_2} \ \cdots \ \sqrt{e_m}]$ of $\mathbf{A}\mathbf{A}^T$) in descending order of magnitude.

A.8.1 Minimum Singular Value

Small singular values imply that relatively large differences in the inputs map to relatively small differences in the output space. This happens in directions parallel to the minor axis of the conditioning ellipsoid. The rank of \mathbf{A} is equivalent to the number, r , of nonzero singular values it has. If $r < m$, then the transformation is singular (has lost rank) and $\sigma_i = 0$, $i > r$. Outputs are insensitive to inputs in directions corresponding to zero singular values. In these directions, infinite changes in the inputs cause zero change in the outputs. One measure of the spatial performance of linear transformations is the magnitude of the minimum singular value, $\kappa_1(\mathbf{A}) = \sigma_{min}$, which can be thought of as the distance from a singularity.

A.8.2 Condition Number

The *condition number*, $1 \leq \kappa_2(\mathbf{A}) \leq \infty$, describes how errors in the input space can be amplified by the linear transformation. To derive the condition

number, define an estimate, \tilde{x} , that is an approximate solution to $\mathbf{x}=\mathbf{A}^{-1}\mathbf{y}$; then we may write $d\mathbf{y}=\mathbf{y}-\mathbf{A}\tilde{x}$ so that $\mathbf{x}-\tilde{x}=\mathbf{A}^{-1}d\mathbf{y}$. Therefore,

$$\|\mathbf{x}-\tilde{x}\|=\|\mathbf{A}^{-1}d\mathbf{y}\|\leq\|\mathbf{A}^{-1}\|\|d\mathbf{y}\|.$$

Since $\mathbf{y}=\mathbf{A}\mathbf{x}$, $\|\mathbf{y}\|\leq\|\mathbf{A}\|\|\mathbf{x}\|$ and $\|\mathbf{x}\|\geq\|\mathbf{y}\|/\|\mathbf{A}\|$, so that

$$\begin{aligned}\frac{\|\mathbf{x}-\tilde{x}\|}{\|\mathbf{x}\|}&\leq\frac{\|\mathbf{A}^{-1}\|\|d\mathbf{y}\|}{\|\mathbf{y}\|/\|\mathbf{A}\|} \\ &\leq\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|d\mathbf{y}\|}{\|\mathbf{y}\|}\end{aligned}$$

and

$$\frac{\|d\mathbf{x}\|}{\|\mathbf{x}\|}=\kappa(\mathbf{A})\frac{\|d\mathbf{y}\|}{\|\mathbf{y}\|}\quad\kappa(\mathbf{A})=\|\mathbf{A}\|\|\mathbf{A}^{-1}\|.$$

When $\kappa_2=1$, the linear transformation is isotropic and small errors in \mathbf{x} correspond uniformly to small errors in \mathbf{y} . As matrix \mathbf{A} approaches a singularity, $\kappa_2\rightarrow\infty$.

The condition number is equivalent to the ratio of the maximum and minimum singular values of \mathbf{A} and, therefore, describes the *eccentricity* of the conditioning ellipsoid. The inverse of the condition number

$$\frac{1}{\kappa}=\frac{\sigma_{min}(\mathbf{A})}{\sigma_{max}(\mathbf{A})}\tag{A.4}$$

varies continuously between zero (singular configurations) and unity (isotropic configurations).

A.8.3 Volume

The volume of the ellipsoid can be appreciable even when the matrix \mathbf{A} is near a singularity, but in general, volume increases as the conditioning ellipsoid becomes more spherical—that is, when the linear transformation is isotropic—and when \mathbf{A} is singular, its volume is zero.

The product of the singular values yields a measure proportional to the volume of the ellipsoid, which can also be derived using the determinant of $\mathbf{A}\mathbf{A}^T$,

$$\kappa_3(\mathbf{A}) = \prod_{i=1}^m \sigma_i = \sqrt{\det(\mathbf{A}\mathbf{A}^T)}.$$

A.8.4 Radius

The geometric mean is a related conditioning metric defined by

$$\kappa_4(\mathbf{A}) = (\sigma_1 \cdot \sigma_2 \cdots \sigma_m)^{1/m}.$$

The geometric mean describes the radius of the hypersphere with the same volume as the output ellipsoid.

This kind of analysis applied to linearized kinematic equations can be used to highlight kinematic aptitudes and limitations of embodied mechanical systems, including the transmission of forces and velocities; the precision and sensitivity of the device to errors; the ability to create accelerations; and the effects of inertia. We will examine these properties in the next several sections.

Example: Scalar Conditioning Metrics Applied to Roger's Arm

Figure A.3 illustrates the scalar conditioning metrics introduced in this section for Roger's arm using singular values computed as the square root of the eigenvalues of $\mathbf{J}\mathbf{J}^T$. In panels (b), (c), (e), and (f), regions inside the reachable workspace that map to black are singular configurations; maxima are endpoint positions that correspond to isotropic configurations. These metrics are qualitatively equivalent, distinguished primarily in their sensitivity near isotropic configurations and their computational expense. The information in panel (d), the condition number, is equivalent to panel (c).

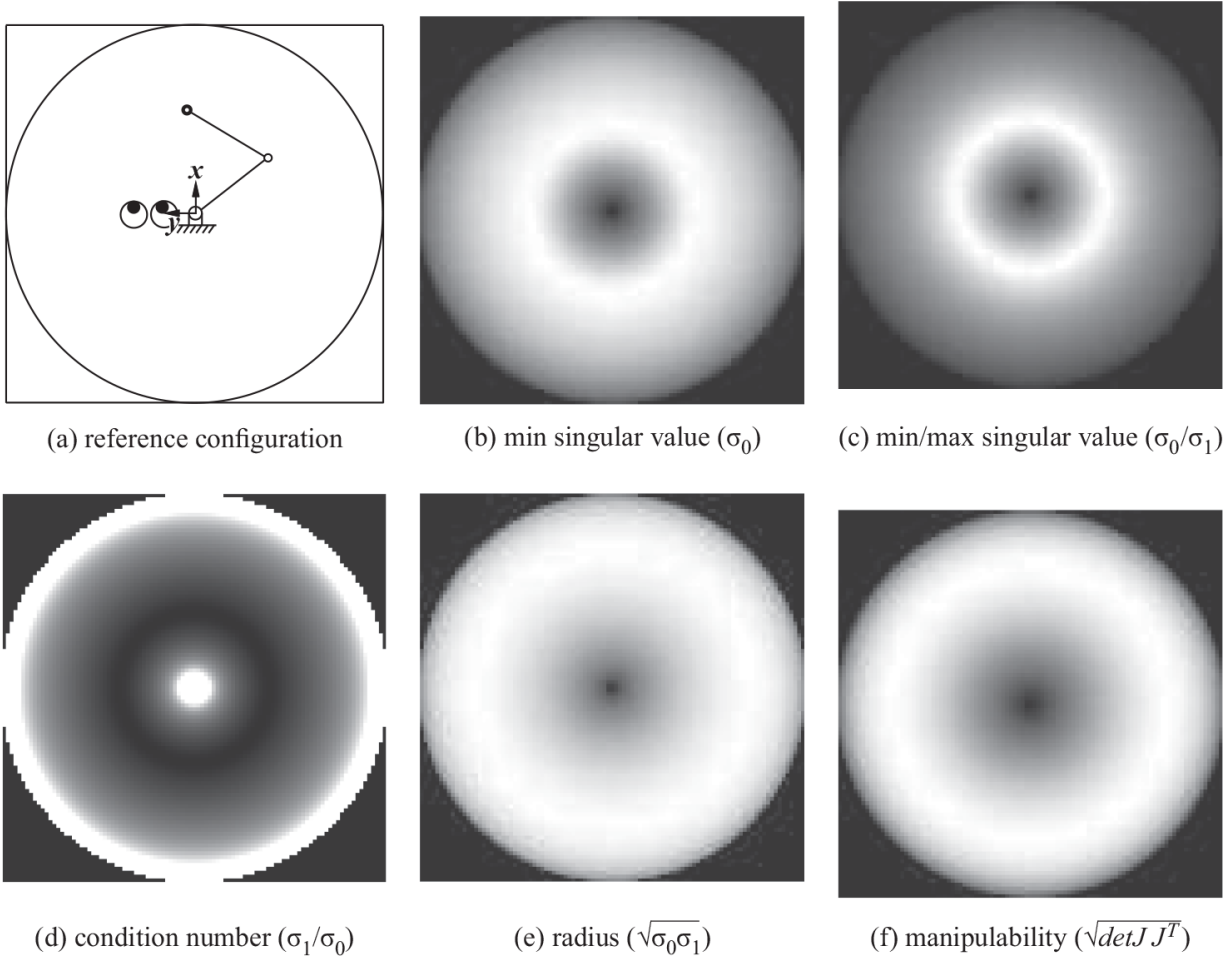


Figure A.3 Scalar kinematic conditioning metrics for Roger’s planar 2R arm configuration. Each panel shows the range of the respective metric from black (minimum) to white (maximum).

□

A.9 The Pseudoinverse

Given two matrices, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$, the Penrose conditions [222] are used to define the conditions under which \mathbf{B} can serve as an inverse of \mathbf{A} . These conditions distinguish increasingly strong inverse relations—the generalized inverse, the reflexive generalized inverse, and the pseudoinverse [208].

$$\begin{array}{l}
(1) \quad \mathbf{ABA} = \mathbf{A} \\
(2) \quad \mathbf{BAB} = \mathbf{B} \\
(3) \quad (\mathbf{AB})^T = \mathbf{AB} \\
(4) \quad (\mathbf{BA})^T = \mathbf{BA}
\end{array}
\left. \vphantom{\begin{array}{l} (1) \\ (2) \\ (3) \\ (4) \end{array}} \right\} \begin{array}{l} \\ \\ \\ \text{pseudoinverse} \end{array} \left. \vphantom{\begin{array}{l} (1) \\ (2) \\ (3) \\ (4) \end{array}} \right\} \begin{array}{l} \text{generalized inverse} \\ \text{reflexive generalized inverse} \end{array} \quad \begin{array}{l} \mathbf{A}^- = \mathbf{B} \in \mathbb{R}^{n \times m} \\ \mathbf{A}_R^- = \mathbf{B} \in \mathbb{R}^{n \times m} \\ \mathbf{A}^\# = \mathbf{B} \in \mathbb{R}^{n \times m} \end{array}$$

The generalized and reflexive generalized inverse are not unique. They represent a family of possible inverse relations. The pseudoinverse, however, selects a single, unique (least squares) element from the set of generalized inverses. If S represents a set of inverse mappings, then $S^\# \subset S_R^- \subset S^-$.

Consider $\mathbf{y} = \mathbf{Ax}$, where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\mathbf{x} \in \mathbb{R}^n$. For such a transformation, $\text{rank}(\mathbf{A})$ cannot exceed $p = \min(m, n)$, and the pseudoinverse (or Moore-Penrose generalized inverse) of \mathbf{A} is written for three cases [209, 140].

Case 1: $m > n$ and $\text{rank}(\mathbf{A}) = n$

In this case, $\mathbf{A}^T \mathbf{A} \in \mathbb{R}^{n \times n}$ is nonsingular, and arbitrary outputs \mathbf{y} are not achievable in general—the system represents a set of m equations in n unknowns (the x_i) and is therefore overdetermined. This is often the case when matrix \mathbf{A} is a linear regression for a scalar y in terms of n independent variables $x_i \in \mathbf{x}$. Matrix \mathbf{A} is estimated from m observations of the relation, each of which may be corrupted with noise, and m can be very large relative to n . In this case, the inverse for \mathbf{A} is selected that minimizes the quadratic output error

$$\begin{array}{ccc}
m \times 1 & m \times n & n \times 1 \\
\left[\mathbf{y} \right] & = & \left[\mathbf{A} \right] \left[\mathbf{x} \right] \\
E = \frac{1}{2} (\mathbf{y} - \mathbf{Ax})^T (\mathbf{y} - \mathbf{Ax}).
\end{array}$$

This quadratic form is positive semidefinite for any given \mathbf{A} , so gradient descent of the squared error defines a unique minimum where

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{x}} &= -\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}) = \mathbf{0} \\ \mathbf{A}^T \mathbf{y} &= \mathbf{A}^T \mathbf{A} \mathbf{x} \\ \mathbf{x} &= [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{y} = \mathbf{A}^\# \mathbf{y}, \text{ so that} \\ \mathbf{A}^\# &= [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \quad (m > n).\end{aligned}\tag{A.5}$$

Equation A.5 is the left pseudoinverse for $m \geq n$ and solves for the inverse of \mathbf{A} that minimizes the squared error $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ of the original overdetermined system.

Case 2: $m < n$ and $\text{rank}(\mathbf{A})=m$

For this case, $\mathbf{A}\mathbf{A}^T \in \mathbb{R}^{m \times m}$ is nonsingular and the transform represents too few constraint equations to uniquely determine the n unknowns. As a result, the transform is underdetermined and redundant. Consequently, there are many inputs \mathbf{x} that map to the same \mathbf{y} .

$$\begin{matrix} m \times 1 & m \times n & n \times 1 \\ \left[\mathbf{y} \right] & = \left[\mathbf{A} \right] & \left[\mathbf{x} \right] \end{matrix}$$

The pseudoinverse yields the single element of this set that satisfies a constrained optimization problem:

$$\begin{aligned}\text{minimize : } & 1/2(\mathbf{x}^T \mathbf{x}) \\ \text{subject to : } & \mathbf{y} - \mathbf{A}\mathbf{x} = 0.\end{aligned}$$

To solve such a problem, we formulate the Lagrangian equation for the constrained system in terms of Lagrange multiplier λ ,

$$\frac{1}{2}\mathbf{x}^T \mathbf{x} + \lambda^T (\mathbf{y} - \mathbf{A}\mathbf{x}) = 0.$$

Following [140], the solution is subject to two necessary conditions. The first of these establishes the optimality condition,

$$\frac{d}{dx} \left[\frac{1}{2} \mathbf{x}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{y} - \mathbf{A}\mathbf{x}) \right] = 0,$$

which yields

$$\begin{aligned} \mathbf{x}^T - \boldsymbol{\lambda}^T \mathbf{A} &= 0, \text{ so that} \\ \mathbf{x}^T &= \boldsymbol{\lambda}^T \mathbf{A}, \text{ or} \\ \mathbf{x} &= \mathbf{A}^T \boldsymbol{\lambda}. \end{aligned} \tag{A.6}$$

The second necessary condition requires

$$\frac{d}{d\boldsymbol{\lambda}} \left[\frac{1}{2} \mathbf{x}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{y} - \mathbf{A}\mathbf{x}) \right] (\mathbf{y} - \mathbf{A}\mathbf{x}) = 0.$$

Substituting the result from [equation A.6](#) into this expression yields

$$\begin{aligned} \mathbf{y} - \mathbf{A}\mathbf{A}^T \boldsymbol{\lambda} &= 0, \text{ and} \\ \boldsymbol{\lambda} &= (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{y}. \end{aligned} \tag{A.7}$$

Combining the results from [equations A.6](#) and [A.7](#) yields the least squares (optimal) solution

$$\mathbf{x} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{y},$$

from which the pseudoinverse of \mathbf{A} for the underdetermined case is written

$$\mathbf{A}^\# = \mathbf{A}^T [\mathbf{A}\mathbf{A}^T]^{-1} \quad (m < n). \tag{A.8}$$

[Equation A.8](#) is the right pseudoinverse for when there are excess degrees of freedom in an underdetermined system. It yields the exact solution $\mathbf{x} = \mathbf{A}^\# \mathbf{y}$ that minimizes $\|\mathbf{x}\|_2$.

Case 3: $m = n$ and $\text{rank}(\mathbf{A}) = m$

Both the left and right pseudoinverses simplify to the ordinary matrix inverse when $m = n$.

$$\mathbf{A}^\# = \mathbf{A}^{-1} \quad (m = n) \tag{A.9}$$

The pseudoinverse provides a general framework for treating problems that engage different resources in different contexts and that address many different tasks. Especially interesting is its application to robots with excess degrees of freedom. In this case, redundancy can be used to address multiple simultaneous goals in a concurrent control design.

The nonsingular Jacobian, $\mathbf{J} \in \mathbb{R}^{m \times n}$, where m is the number of rows and n is the number of columns, has rank $p = \min(m, n)$. When $m < n$, the pseudoinverse can also be used to define an $(n - m)$ dimensional orthogonal space defined by $\mathcal{N} = (\mathbf{I}_n - \mathbf{J}^\# \mathbf{J})$ —the annihilator of matrix \mathbf{J} —where \mathbf{I}_n is the $n \times n$ identity matrix. We can demonstrate that $\mathbf{J}^\#$ and $(\mathbf{I}_n - \mathbf{J}^\# \mathbf{J})$ are orthogonal by computing their inner product,

$$\begin{aligned} (\mathbf{J}^\#)^T (\mathbf{I}_n - \mathbf{J}^\# \mathbf{J}) &= \{(\mathbf{I}_n - \mathbf{J}^\# \mathbf{J}) \mathbf{J}^\#\}^T \\ &= \{\mathbf{J}^\# - \mathbf{J}^\# \mathbf{J} \mathbf{J}^\#\}^T \\ &= \mathbf{0}, \end{aligned}$$

given that $\mathbf{I} - \mathbf{J}^\# \mathbf{J}$ is symmetric (so that $[\mathbf{I} - \mathbf{J}^\# \mathbf{J}] = [\mathbf{I} - \mathbf{J}^\# \mathbf{J}]^T$) and that $\mathbf{J}^\# \mathbf{J} \mathbf{J}^\# = \mathbf{J}^\#$ by the Penrose conditions. Therefore, $(\mathbf{I}_n - \mathbf{J}^\# \mathbf{J})$ defines the orthogonal linear nullspace \mathcal{N} of pseudoinverse $\mathbf{J}^\#$. This important definition allows us to consider subordinate objectives that reside exclusively in the nullspace of superior objectives without disturbing the superior task.

A.10 Linear Integral Transforms

A *transform* is a function that performs a mapping between sets.¹ Transforms can take many forms. In section 4.3, the homogeneous transform is introduced to map Euclidean sets to other Euclidean sets. In this section, we introduce two more linear transforms that have had a huge impact on the solution and/or manipulation of equations that are central to the study of sensory and motor systems.

Linear integral transforms map equations from one *domain* to another in order to make certain forms of analysis and interpretation simpler. They change the representation of a function in a manner that makes important structure in the equations clear. The change in representation involves

mapping (or projecting) it into an orthogonal basis. In general, the integral transform takes the form

$$F(u) = \int_{t_0}^{t_1} \Phi(t, u) f(t) dt.$$

This mapping transforms functions $f(t)$ into image function $F(u)$ using the kernel function $\Phi(t, u)$. Important kinds of transforms yield well-defined inverse kernel functions that perform the inverse mapping

$$f(t) = \int_{u_0}^{u_1} \Phi^{-1}(u, t) F(u) du.$$

The integral transforms we will consider with invertible kernels constitute bijective mappings between sets $\{f(t)\}$ and its image $\{F(u)\}$. These mappings are *one-to-one*. This means that single elements of $\{f(t)\}$ map to single elements of $\{F(u)\}$. Moreover, bijections are *onto*—they “cover” the target set in the sense that they do not omit any element of either set and they do not leave any of the elements in the target domain inaccessible through the mapping. As a result, the integral transform we consider can be described by tabulating transform pairs $f(t) \leftrightarrow F(u)$ for (many) elements of these sets.

We will introduce two important bijective integral transforms. The Fourier transform maps time-domain functions (signals) into complex *spectral coefficients* that weight sinusoidal basis functions in a complex frequency space. This transformation makes finding and manipulating information in signals more efficient. The related Laplace transform maps differential functions (often found in dynamical/control systems) into polynomial functions of a complex frequency s whose roots can be used to solve the original differential equation.

A.10.1 Complex Numbers

The complex number s is used to denote a coordinate in the real-imaginary plane. It can be expressed in the Cartesian form,

$$s = \sigma + i\omega, \tag{A.10}$$

where $\sigma = \text{Re}(s)$ is the *real* part of s , $\omega = \text{Im}(s)$ is the *imaginary* part of s , and $i = \sqrt{-1}$. Parameters σ and ω are real numbers, and i is called imaginary because there is no real solution to $\sqrt{-1}$. Imaginary numbers are defined by their coordinates (σ, ω) on the two-dimensional complex plane, where the horizontal axis is the real part σ of s , and the vertical axis is the imaginary part ω of s as in [figure A.4](#).

This same concept can be expressed in polar form, where

$$s = \rho e^{i\phi}, \tag{A.11}$$

where $\rho = \sqrt{\sigma^2 + \omega^2}$ is the *magnitude* of s , and $\phi = \text{atan}(\omega/\sigma)$ is the *phase* of s .

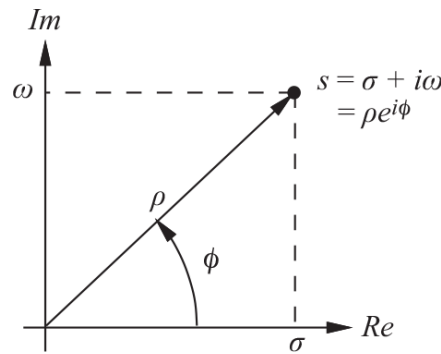


Figure A.4
The complex plane

Complex numbers extend the one-dimensional real number line to the two-dimensional complex plane. They are equal only if both their real and imaginary parts are equal. The complex conjugate of $s = \sigma + i\omega$, is $s^* = \sigma - i\omega$, both addition and subtraction work as they would in two-dimensional vectors in the plane. Other operators ($*$, $/$, $\sqrt{\quad}$) are defined to extend common algebraic manipulation into the complex plane in reasonable ways—relational operators don't naturally extend to the complex plane.

Exponentiation of purely imaginary numbers is defined using Euler's formula,

$$e^{i\theta} = \cos(\theta) + i\sin(\theta), \tag{A.12}$$

so that the exponentiation of an arbitrary complex number is defined:

$$\begin{aligned}
e^{\sigma+i\omega} &= e^{\sigma} e^{i\omega} \\
&= e^{\sigma} [\cos(\omega) + i\sin(\omega)],
\end{aligned}$$

A.10.2 Fourier Transform

The Fourier transform represents a signal by projecting it onto an infinite family of sinusoidal basis functions. It defines a mapping from a spatial or temporal function to a representation that makes its frequency content explicit. For example, a one-dimensional spatial signal $f(x)$ maps via the Fourier transform to its frequency-domain equivalent

$$\mathcal{F}[f(x)] = F(\omega_x) = \int_{-\infty}^{\infty} f(x) e^{-i(2\pi\omega_x x)} dx, \quad (\text{A.13})$$

where x [m] is the spatial variable and ω_x [cycles/m] is the corresponding spatial frequency so that $(2\pi\omega_x x)$ is in radians and $i = \sqrt{-1}$. One may view the Fourier transform as the projection of function $f(x)$ onto basis functions $e^{-i(2\pi\omega_x x)}$ for spatial frequencies $\omega_x \in [-\infty, \infty]$. The spectral coefficients $F(\omega_x)$ determine how much the exponential sinusoidal basis functions account for the global shape of function $f(x)$.

The inverse transform is written

$$\mathcal{F}^{-1}[F(\omega_x)] = f(x) = \int_{-\infty}^{\infty} F(\omega_x) e^{i(2\pi\omega_x x)} d\omega_x. \quad (\text{A.14})$$

Table A.1 lists several Fourier transform pairs that make evaluating the transform much more convenient for commonly used functions. The analysis can be applied to orthogonal spatial and temporal dimensions of a function independently. For example, in an acoustic signal, time varying acoustic waves can be represented as the weighted sum of pure tones with appropriate phase (determined by the mixture of sine and cosine at a particular frequency). In this case, $F(\omega_t) = \mathcal{F}[f(t)]$ defines the spectral coefficients for frequency components measured in *cycles/second*. In computer vision, the signal consists of a luminosity function that varies spatially over the image plane. When the sampling interval is in units of *pixels* (picture elements), the transform identifies the response of the

sinusoidal basis at spatial frequencies, $\boldsymbol{\omega} = (\omega_x, \omega_y)$, in *cycles/pixel*. In video applications, both spatial and temporal variation exists and $F(\boldsymbol{\omega})$ describes the amplitude and phase of the sinusoidal basis functions in terms of spatial and temporal frequencies.

Table A.1
Fourier Transform Pairs

$$F(\omega) = \mathcal{F}(f(x)) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x} dx \quad \omega[\text{rad}/m] = 2\pi u[\text{cycles}/m]$$

Name	$f(x)$	$F(\omega)$
Rectangular function	$rect(x) = 1 \quad -\frac{1}{2} < x < \frac{1}{2}$	$sinc(\omega/2\pi) = \frac{\sin(\omega/2)}{\omega/2}$
Triangular function	$tri(x) = 2(x + \frac{1}{2}) \quad -\frac{1}{2} < x < 0$ $1 - 2(x) \quad 0 < x < \frac{1}{2}$	$sinc^2(\omega/2\pi)$
Gaussian	$e^{-\alpha x }$ e^{-px^2}	$2\alpha/(\alpha^2 + \omega^2)$ $\frac{1}{\sqrt{2p}} e^{-\omega^2/4p}$
Unit impulse	$\delta(x)$	1
Comb function	$\sum_n \delta(x - nx_0)$	$\frac{1}{x_0} \sum_n \delta(\frac{\omega}{2\pi} - \frac{n}{x_0})$
Differentiation	$g^n(x)$	$(i\omega)^n G(\omega)$
Linear combination	$ag(x) + bh(x)$	$aG(\omega) + bH(\omega)$
Scale	$f(ax)$	$\frac{1}{ a } F(\frac{\omega}{a})$

Decomposed in this fashion, certain spatiotemporal frequencies can be isolated, they can be amplified or attenuated to influence the appearance of the signal, or they can be used to recognize distinctive patterns in periodic functions. The transform can be generalized to account for multiple orthogonal spatial dimensions (in image analysis) and spatiotemporal variables (in video analysis).

In computer vision, the transform is written to accommodate two spatial variables on the image plane. The corresponding definitions in 2D are

$$\mathcal{F}[f(x, y)] = F(u, v) = \iint_{-\infty}^{\infty} f(x, y) e^{-i(2\pi(ux+vy))} dx dy \text{ and} \quad (\text{A.15})$$

$$\mathcal{F}^{-1}[F(u, v)] = f(x, y) = \iint_{-\infty}^{\infty} F(u, v) e^{i(2\pi(ux+vy))} dudv. \quad (\text{A.16})$$

Transforming the spatial signal into the frequency-domain provides the basis for a very general spatial frequency filter. [Table A.1](#) summarizes some of the one-dimensional Fourier transform pairs that will be useful in this text.

We conclude this brief introduction to the Fourier transform with two important theorems—the shift theorem and the convolution theorem—that prove useful in section 8.1.1, where we analyze the effect of sampling on the information content of signals.

Shift Theorem. *The shift theorem follows directly from the definition of the Fourier transform. If*

$$\begin{aligned} \mathcal{F}[f(x)] &= \int_{-\infty}^{\infty} f(x) e^{-i(2\pi\omega_x x)} dx, \text{ then} \\ \mathcal{F}[f(x-a)] &= \int_{-\infty}^{\infty} f(x-a) e^{-i(2\pi\omega_x x)} dx, \text{ then} \\ &= \int_{-\infty}^{\infty} f(x') e^{-i(2\pi\omega_x(x'+a))} dx', \text{ and} \\ &= e^{-i(2\pi\omega_x a)} \int_{-\infty}^{\infty} f(x') e^{-i(2\pi\omega_x x')} dx', \text{ so that} \\ \mathcal{F}[f(x-a)] &= e^{-i(2\pi\omega_x a)} \mathcal{F}(f(x)). \end{aligned} \quad (\text{A.17})$$

We conclude that the Fourier transform of the shifted function is a weighted version of the Fourier transform of the original function. The new spectral coefficients are weighted by an exponential term that depends on the spatial frequency, ω_x .

Convolution Theorem. *The convolution of two functions $f(x)$ and $g(x)$, written $f(x)*g(x)$, is defined by the integral,*

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha, \quad (\text{A.18})$$

where α is the integration variable. The result, $h(x)$, is the spatial correlation of function $f(x)$ and function $g(x)$, whose origin has been shifted to $\alpha = x$. An important property of convolution lies in the way it maps through the Fourier transform.

$$\begin{aligned} \mathcal{F}[f(x) * g(x)] &= \mathcal{F}[h(x)] \\ &= \mathcal{F}\left[\int_{\alpha} f(\alpha)g(x - \alpha)d\alpha\right] \\ &= \int_x \left[\int_{\alpha} f(\alpha)g(x - \alpha)d\alpha\right] e^{-i2\pi\omega_x x} dx \\ &= \int_{\alpha} f(\alpha) \left[\int_x g(x - \alpha)e^{-i(2\pi\omega_x x)} dx\right] d\alpha, \quad \text{and by the shift theorem,} \\ &= \int_{\alpha} f(\alpha)e^{-i(2\pi\omega_x \alpha)} d\alpha \int_x g(x)e^{-i(2\pi\omega_x x)} dx, \quad \text{therefore,} \end{aligned}$$

$$\mathcal{F}[f(x) * g(x)] = F(\omega_x)G(\omega_x). \quad (\text{A.19})$$

This result is commonly known as the convolution theorem and it states that convolution in the spatial-domain is equivalent to multiplication in the frequency-domain. It is easy to show that the opposite is also true; convolution in the frequency-domain is equivalent to multiplication in the spatial-domain. As we will see, this result implies that convolution operators are essentially spectral filters whose bandpass characteristics are defined by their Fourier transforms.

A.10.3 Laplace Transform

The Laplace transform is a linear bijection between the time-domain (t) and the complex frequency-domain (s) defined by a reciprocal pair of integral transforms.

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st} dt \quad f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st} ds \quad (\text{A.20})$$

The Laplace transform $\mathcal{L}[f(t)]$ converges for functions $f(t)$ that are solutions to linear differential equations with constant coefficients—these functions are *Laplace-transformable*. The image $F(s)$ of $f(t)$ is a function the *complex frequency* variable s [rad/sec] so that the product st is in [rad]. The expression for the inverse Laplace transform $\mathcal{L}^{-1}[\cdot]$ makes it clear that $f(t)$ is assumed to be a weighted sum of exponential terms e^{st} . Recall that for $s=\sigma+i\omega$,

$$e^{st} = e^{(\sigma+i\omega)t} = e^{\sigma t} e^{i\omega t} = e^{\sigma t} [\cos(\omega t) + i\sin(\omega t)],$$

so that the function $f(t)$ is captured in the form of the sum of weighted, exponentially damped, odd and even sinusoids.

Example: Laplace Transform of an Exponential Function $f(t)=e^t$

In this case,

$$F(s) = \int_0^{\infty} e^t e^{-st} dt = \int_0^{\infty} e^{(1-s)t} dt = \frac{1}{1-s} e^{(1-s)t} \Big|_0^{\infty}.$$

If we assume that $Re(s)>1$ so that $e^{(1-s)t} \rightarrow 0$ as $t \rightarrow \infty$, then

$$F(s) = \frac{1}{1-s} [e^{(1-s)\infty} - e^{(1-s)0}] = \frac{1}{s-1},$$

and therefore,

$$\mathcal{L}[e^t] = \frac{1}{s-1}.$$

□

Example: Laplace Transform of the Unit Step Function $f(t)=1, t \geq 0$

The function in [figure A.5](#) is known as the *unit step* function $u(t)$. It is often used in the analysis of a differential equation whose inputs can change instantaneously (at $t=0$), such as when the reference input to a control system changes. The Laplace transform of the unit step is as follows:

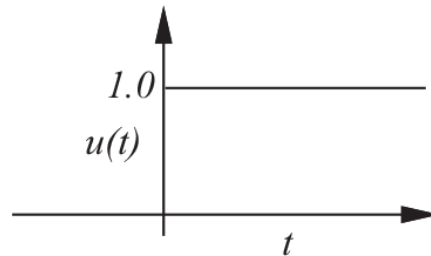


Figure A.5

The unit step input function

$$F(s) = \mathcal{L}[u(t)] = \int_0^{\infty} e^{-st} dt = -\frac{1}{s} e^{-st} \Big|_0^{\infty} = \frac{1}{s}$$

□

[Table A.2](#) includes these and several other functions that are commonly encountered in linear system analysis.

Table A.2

Laplace transform pairs

Name	$f(t)$	$F(s)$
Unit impulse	$\delta(t)$	1
Unit step	$u(t)$	$\frac{1}{s}$
Unit ramp	$tu(t)$	$\frac{1}{s^2}$
n th-order unit ramp	$t^n u(t)$	$\frac{n!}{s^{n+1}}$
Exponential	e^{-at}	$\frac{1}{s+a}$
Ramped exponential	$\frac{1}{(n-1)!} t^{n-1} e^{-at}$	$\frac{1}{(s+a)^n}$
Sine	$\sin at$	$\frac{a}{s^2 + a^2}$
Cosine	$\cos at$	$\frac{s}{s^2 + a^2}$
Damped sine	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2 + \omega^2}$
Damped cosine	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2 + \omega^2}$
Hyperbolic sine	$\sinh at$	$\frac{a}{s^2 + a^2}$
Hyperbolic cosine	$\cosh at$	$\frac{s}{s^2 + a^2}$

A.11 Time-Domain Responses for the Harmonic Oscillator

Consider only the class of asymptotically stable systems (those with roots that have negative real parts). In this case, [equation 3.14](#) yields either two distinct roots (when $\zeta \neq 1$) or a repeated real root (when $\zeta = 1$).

Two Distinct Roots ($\zeta \neq 1$)—Two distinct roots of [equation 3.13](#) include both distinct real roots ($\zeta > 1$) and complex conjugate roots ($\zeta < 1$)—the former leading to an overdamped response and the latter resulting in an underdamped response. In both of these situations, the time-domain solution is written in the form

$$x(t) = A_0 + A_1 e^{s_1 t} + A_2 e^{s_2 t},$$

and the complete solution in the time interval $[0, \infty)$ is derived by specifying three boundary conditions on $x(t)$ with which to solve for

constants A_0 , A_1 , and A_2 . For example, general boundary conditions can be used to describe a system that is released at time zero from position x_0 and velocity \dot{x}_0 that returns to equilibrium at x_∞ .² Under these conditions, three constraint equations can be written in the unknown coefficients,

$$\begin{aligned}x(0) &= x_0 = A_0 + A_1 + A_2 \\ \dot{x}(0) &= \dot{x}_0 = s_1 A_1 + s_2 A_2, \\ x(\infty) &= x_\infty = A_0\end{aligned}$$

so that a complete time-domain solution is determined:

$$x(t) = x_\infty + \frac{(x_0 - x_\infty)s_2 - \dot{x}_0}{s_2 - s_1} e^{s_1 t} + \frac{(x_0 - x_\infty)s_1 - \dot{x}_0}{s_1 - s_2} e^{s_2 t}. \quad (\text{A.21})$$

Repeated Real Roots—($\zeta=1$)—A special case exists when $\zeta=1$, in which case the quadratic characteristic equation produces repeated real roots, $s_1 = s_2 = -\omega_n$. We might presume that the solution will take the form $x(t) = A_0 + A_1 e^{-\omega_n t}$, and this solution does satisfy the original differential equation, but it does not provide the flexibility to address the general set of boundary conditions proposed. Instead, it can be shown that all solutions of the form

$$x(t) = A_0 + (A_1 + A_2 t) e^{-\omega_n t}$$

also satisfy the original differential equation (see section 3.4, exercise 3b).

Given generic boundary conditions where the system is released at time zero from position x_0 with velocity \dot{x}_0 and returns to equilibrium at x_∞ , three constraint equations can be written in the unknown coefficients,

$$\begin{aligned}x(0) &= x_0 = A_0 + A_1 \\ \dot{x}(0) &= \dot{x}_0 = -\omega_n A_1 + A_2, \\ x(\infty) &= x_\infty = A_0,\end{aligned}$$

so that a complete time-domain solution for the critically damped case is determined:

$$x(t) = x_\infty + [(x_0 - x_\infty) + (\dot{x}_0 + (x_0 - x_\infty)\omega_n) t] e^{-\omega_n t}. \quad (\text{A.22})$$

Example: Time-Domain Response of the Spring-Mass-Damper

The second-order response generates four qualitatively different responses that depend on the roots of the characteristic equation. To see how this happens, consider a harmonic oscillator with spring constant $K = 1.0$ [N/m], mass $m = 2.0$ [kg], and a variable damper $B = 2\zeta\sqrt{Km}$ [Nsec/m] chosen for ζ values in the interval [0.0, 2.0]. This example considers the time-domain performance of this system subject to a particular boundary condition,

$$x_0 = \dot{x}_0 = 0, \quad \text{and} \quad x_\infty = 1.$$

Case 1: Overdamped ($\zeta > 1$)

Under these boundary conditions, [equation A.21](#) simplifies to

$$x(t) = 1.0 - \frac{s_2}{s_2 - s_1} e^{s_1 t} - \frac{s_1}{s_1 - s_2} e^{s_2 t}. \quad (\text{A.23})$$

The overdamped condition leads to a pair of distinct real roots—for $\zeta = 2.0$, $B = 4\sqrt{2}$, and $s_{1,2} = -0.19, -2.64$. When these roots are inserted into [equation A.23](#), they determine the overdamped response in the curve labeled $\zeta = 2.0$ in [figure A.6](#). The real root with the smallest absolute value approaches its asymptote more slowly and will, therefore, dominate the system's asymptotic behavior. It produces a relatively sluggish, non-oscillatory response that reflects the excessively dissipative influence of the damper. As a result, overdamped control configurations take comparatively longer to converge to equilibrium setpoints than configurations with $\zeta = 1$.

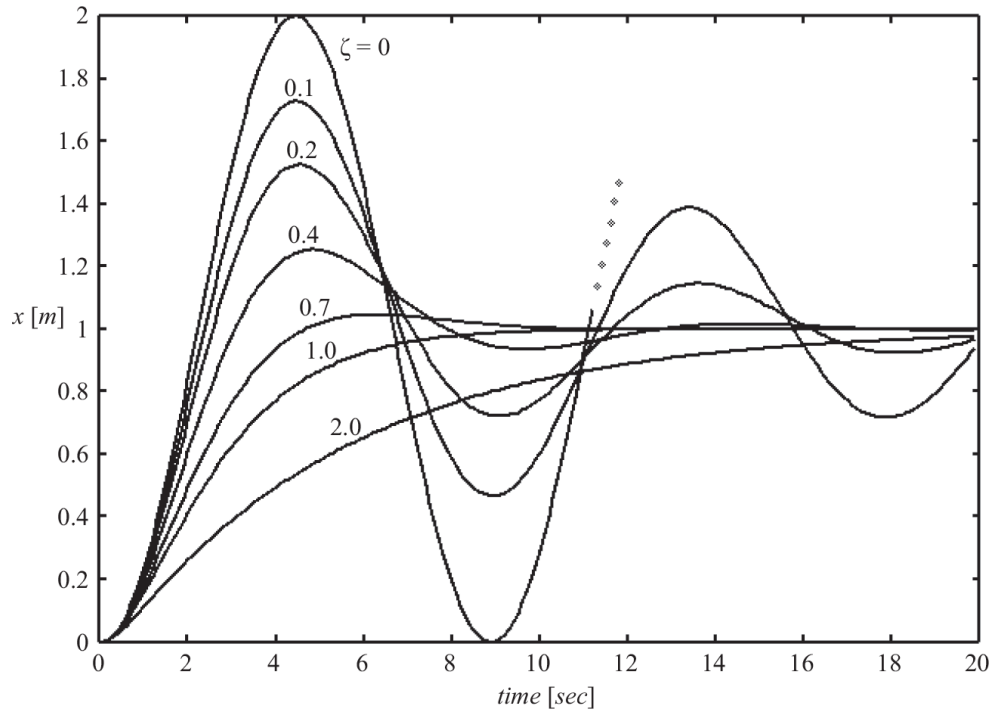


Figure A.6

The response of the second-order PD position controller as a function of ζ ($K=1.0$ [N/m], $m=2.0$ [kg]) given boundary conditions $x_0 = \dot{x}_0 = 0$ and $x_\infty = 1.0$

Case 2: Underdamped ($\zeta < 1$)

When $0 \leq \zeta < 1$, roots s_1 and s_2 are complex conjugates

$$s_{1,2} = \alpha \pm \beta i,$$

where $\alpha = -\zeta\omega_n$, $\beta = \omega_n\sqrt{1-\zeta^2}$, and $i = \sqrt{-1}$. The imaginary parts of these roots cause oscillations since $e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$, and the real parts cause the amplitude of the oscillations to decay exponentially over time. These kinds of systems are underdamped because the forces in the spring dominate dissipative forces in the damper.

Inserting complex conjugate roots into [equation A.23](#), we find

$$\begin{aligned}
x(t) &= 1 - \frac{s_2}{s_2 - s_1} e^{s_1 t} - \frac{s_1}{s_1 - s_2} e^{s_2 t} \\
&= 1 + \frac{(\alpha - \beta i)}{2\beta i} e^{(\alpha + \beta i)t} - \frac{(\alpha + \beta i)}{2\beta i} e^{(\alpha - \beta i)t}, \quad \text{and} \\
&= 1 + e^{\alpha t} \left[\frac{(\alpha - \beta i)}{2\beta i} e^{i\beta t} - \frac{(\alpha + \beta i)}{2\beta i} e^{-i\beta t} \right].
\end{aligned}$$

Substituting Euler's formulas ($e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$ and $e^{-i\omega t} = \cos(\omega t) - i\sin(\omega t)$), we find

$$x(t) = 1 + e^{\alpha t} \left[\frac{\alpha}{\beta} \sin(\beta t) - \cos(\beta t) \right].$$

This type of response is illustrated in [figure A.6](#) in all the curves where $\zeta < 1$. As $\zeta \rightarrow 0$, the behavior becomes less dissipative until at $\zeta = 0$ (when $B = 0$), the roots are purely imaginary and the system behaves like an ideal spring. In this configuration, $x(t)$ undergoes a constant amplitude oscillation at the natural frequency ω_n , converting potential energy into kinetic energy and back again, reversibly.

Case 3: Critically damped ($\zeta = 1$)

Evaluating this general solution ([equation A.22](#)) subject to these boundary conditions ($x_0 = 0$, $\dot{x}_0 = 0$, and $x_\infty = 1$) yields

$$x(t) = 1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}. \quad (\text{A.24})$$

Critically damped behavior is illustrated in [figure A.6](#) by the curve labeled $\zeta = 1$. It approaches zero error faster than the overdamped configuration but does not oscillate so it is often the design target for robot control applications.

□

The progression through unstable, undamped, underdamped, critically damped, and overdamped behavior in response to changes in the value of a PD control parameter can be visualized in another way using the *root locus* diagram that plots the pair of roots on the real-imaginary plane as a function of a control parameter (K or B) in the PD control design. Usually, the

parameter in question is K . For example, in exercise 3.4.7a, the value of mass m and damper B are held fixed, and the reader is asked to plot the roots of the resulting second-order system to inform the choice of a complementary spring constant K . However, the test parameter can be B as well, as is the case in the following example.

Example: The Root Locus Diagram for the PD Control System

Figure A.7 plots the two roots (s_1 in blue, and s_2 in red) of equation 3.13 under the same conditions considered in Figure 3.15.

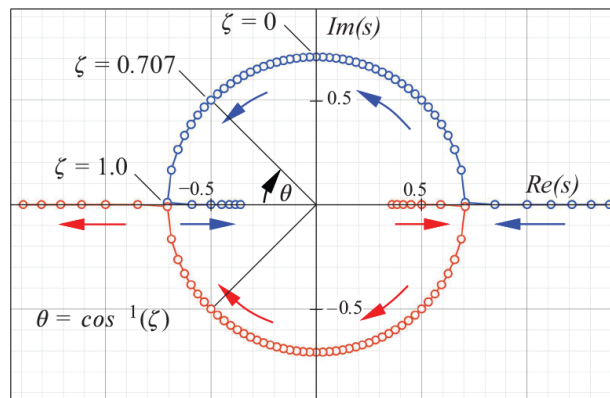


Figure A.7

The root locus for the continuum of responses underlying figure 3.15 when $K = 1.0$ [Nm/rad], $I = 2.0$ [kgm²], and $-3.5 \leq B \leq 3.5$ (corresponding to $-1.24 \leq \zeta \leq 1.24$)

All PD control configurations with roots on the right half plane ($\sigma = \text{Re}(s) > 0$), where both B and ζ are negative, are *unstable*, with damper B injecting rather than dissipating energy.

Starting on the right-hand side of the diagram, we observe a pair of distinct real (positive) roots s_1 and s_2 that approach one another along the real axis, converge into a repeated (positive) real root, and then depart to form complex conjugates.

When the conjugate roots lie on the imaginary axis (where $\text{Re}(s) = 0$), the system response is *undamped* and marginally (or orbitally) stable. This system configuration causes the system to oscillate about a setpoint while conserving energy. As B continues to increase from the marginally stable configuration, it generates the family of *underdamped* responses ($0.0 \leq \zeta$

<1.0) corresponding to $0 \leq B < 2.78$. When B increases to the unique *critically damped* configuration, where $\zeta = 1.0$ (and B is approximately 2.78), there is a pair of repeated (negative) real roots and the system is non-oscillatory. Increasing B further causes the pair of roots to, once again, depart from one another along the real axis. These system configurations are *overdamped* and the excess damping may be less sensitive to disturbances of greater magnitude at the cost of slower, more viscous behavior.

The root locus allows designers to customize control parameters or even include additional roots to change the locus and, thus, to shape the system behavior.

□

A.11.1 Frequency-Dependent Amplitude and Phase Response

Consider the transform pair for the sinusoidal input of frequency ω ([table A.2](#)).

$$\theta_{ref}(t) = A \cos \omega t \quad \begin{array}{c} \xrightarrow{\mathcal{L}[\cdot]} \\ \xleftarrow{\mathcal{L}^{-1}[\cdot]} \end{array} \quad \Theta_{ref}(s) = \frac{As}{s^2 + \omega^2}$$

As before, the output of the system is the product of the closed-loop transfer function and the input,

$$\Theta(s) = \left[\frac{\Theta(s)}{\Theta_{ref}(s)} \right] \left[\frac{As}{s^2 + \omega^2} \right] = \left[\frac{\Theta(s)}{\Theta_{ref}(s)} \right] \left[\frac{As}{(s - i\omega)(s + i\omega)} \right], \quad (\text{A.25})$$

and we see that the forcing function contributes a pair of purely imaginary conjugate roots $s = \pm i\omega$. Therefore, the partial-fraction expansion of the closed-loop transfer function subject to the sinusoidal forcing function will look like

$$\Theta(s) = \Theta_{cltf}(s) + \frac{k_1}{s - i\omega} + \frac{k_2}{s + i\omega} .$$

The inverse Laplace transform of these terms yields time-domain responses like $k_1 e^{i\omega t}$ and $k_2 e^{-i\omega t}$, respectively. Therefore, the response of the second-

order system to a sinusoidal input is also a sinusoid of the same frequency.³

Returning to [equation A.25](#), we see that the *magnitude* of the sinusoidal response is proportional to the amplitude of the forcing function, A , and the gain expressed in the closed-loop transfer function

$$\frac{\Theta(s)}{\Theta_{ref}(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{(s/\omega_n)^2 + 2\zeta(s/\omega_n) + 1} . \quad (\text{A.26})$$

The influence of the CLTF on the response can be determined by evaluating [equation A.26](#) at the roots introduced by the forcing function ($s = \pm i\omega$).

$$\left. \frac{\Theta(s)}{\Theta_{ref}(s)} \right|_{s=i\omega} = \frac{1}{(i\omega/\omega_n)^2 + 2\zeta(i\omega/\omega_n) + 1}$$

The result is a complex number with corresponding magnitude and phase,

$$\left| \frac{\Theta(s)}{\Theta_{ref}(s)} \right|_{s=i\omega} = \frac{1}{[(1 - (\omega/\omega_n)^2)^2 + (2\zeta(\omega/\omega_n))^2]^{1/2}} , \text{ and} \quad (\text{A.27})$$

$$\phi(\omega) = -\tan^{-1} \left(\frac{2\zeta(\omega/\omega_n)}{1 - (\omega/\omega_n)^2} \right) . \quad (\text{A.28})$$

[Figure A.8](#) plots the magnitude and phase of the harmonic oscillator for various values of the driving frequency ω and the damping ratio ζ . Note the striking resonance in the magnitude response near the natural frequency. In fact, for $\zeta = 0$, the gain becomes theoretically infinite—the undamped ($\zeta = 0$) system can be destabilized by pumping energy in at the resonant frequency much like a child pumps to increase the amplitude of a swing.

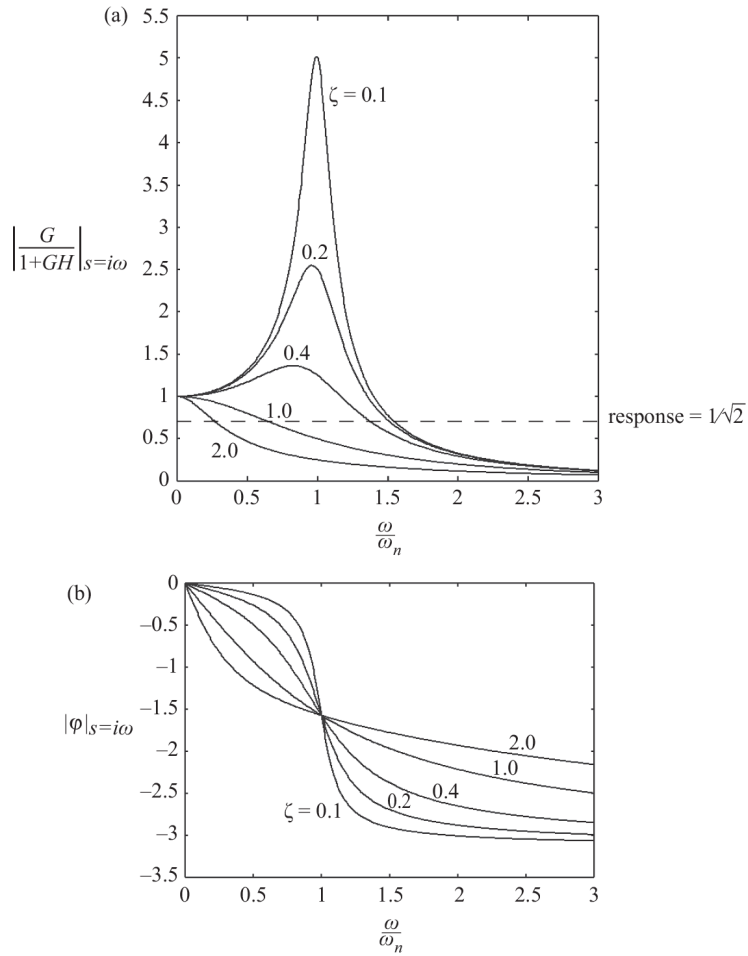


Figure A.8

The magnitude (a) and the phase response (b) of the second-order SMD position controller evaluated at $s = \pm i\omega$ for various values of ζ

Panel A.8 (a) shows that if the driving frequency is large, the gain in the CLTF goes to zero asymptotically, and the ability of the second-order system to track such reference functions is compromised. The *bandwidth* of the system is that frequency where the gain falls to $1/\sqrt{2}$, which corresponds to the *half power* or -3 [dB] point of the CLTF. The over- and critically damped configurations in panel (a) have a relatively small bandwidth using this criteria, and increases for underdamped configurations at the expense of overshoot and oscillation. In panel (b), we see that the natural frequency also identifies the point at which the response lags 90 [deg] behind the reference input. If the driving frequency increases beyond the natural frequency, the response goes asymptotically toward a phase lag of 180 [deg].

A.11.2 Stiffness and Impedance

Stiffness refers to the change in force due to a change in deformation. For a linear spring, the ratio of change in force to change deflection is the spring constant, $K = \Delta F / \Delta x$. The transfer function generalizes this concept to linear differential systems like the spring-mass-damper and to time-dependent displacement functions.

For example, the transform pair for the continuous-time SMD (equation 3.12) leads directly to the stiffness relation

$$\frac{\tilde{F}(s)}{X(s)} = s^2 + 2\zeta\omega_n s + \omega_n^2, \quad (\text{A.29})$$

which can be written in the form of an equivalent SISO filter as in figure A.9(a). The filter transforms an input displacement into externally applied force—it represents the inverse dynamics of the spring-mass-damper. The stiffness of the SMD is a time varying function that satisfies a differential relation. It is clear, in this context, that a time varying deformation will produce a second-order force response—the stiffness of the SMD is also, therefore, a second-order phenomenon.

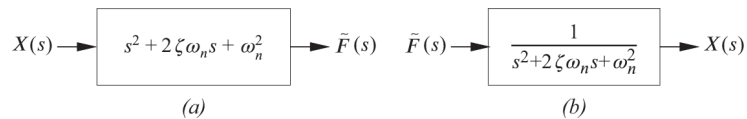


Figure A.9

The transfer functions describing the (a) stiffness and (b) compliance of the SMD controller

The reciprocal relation in figure A.9(b) is a transfer function describing the forward dynamic model of the SMD. It describes how an external input force causes a deflection. Therefore, the forward dynamic relation describes the *compliance* of the spring-mass-damper.

The transfer function can be used to describe how power is transmitted through the PD motor unit to the environment. *Impedance* describes how power is converted from one form to another in electrical, mechanical, hydraulic, thermal, and chemical systems that are described by linear differential equations. In all of these types of systems, power is described as the product of *effort* and *flow* variables. For example, in electrical circuits

the effort variable is voltage (an electromotive force), the flow variable is current, and the product of voltage and current is power.

The ratio of effort and flow describes a generalized resistance per unit of flow called *impedance*. For example, in the case of a current flow through a resistor ($V=IR$), the impedance of this circuit is $V/I=R$. Transfer functions generalize the concept of resistance to devices that have differential effort/flow relationships. For example, the capacitor is defined by $i(t)=C(dV(t)/dt)$. Applying the Laplace transform we get $I(s)=CsV(s)$, so that the impedance of the capacitor, $V(s)/I(s)$, is just $1/Cs$.

To describe the dynamics of mechanical systems, effort variables are forces and flow variables are velocities. Using the same techniques, we can describe the mechanical impedance of the controlled system as the ratio of force to velocity at the point at which the force is applied.

Starting from [equation A.29](#), and noting that $V(s)=sX(s)$, we write the impedance of the SMD as

$$\frac{\tilde{F}(s)}{V(s)} = \frac{\tilde{F}(s)}{sX(s)} = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{s}, \quad (\text{A.30})$$

which is written in the form of an equivalent SISO filter in [figure A.10\(a\)](#). The reciprocal of impedance is *admittance* ([figure A.10\(b\)](#)).

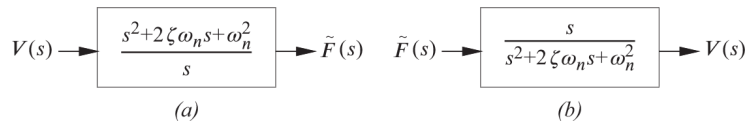


Figure A.10

The transfer functions describing the (a) impedance and (b) admittance relations for the SMD controller

Impedance matching techniques are often used at the interface between such systems to optimize the transfer of power between them. *Impedance control* techniques [115] are used to create complementary impedance couplings between a robot manipulator and the task in which it is engaged. For example, turning a crank mechanism that is stiff radially and compliant tangentially requires controls for the robot manipulator that are stiff tangentially and compliant radially in order to transmit power efficiently through the crank mechanism.

1. The mapping can transform elements of one set, X , into elements of another set, Y , or from set X back into itself.
2. The state at $t = \infty$ for the asymptotically stable system is the equilibrium state $(x_\infty, 0)$. We use the position constraint at $t = \infty$ as the third boundary condition for the time-domain solution.
3. Recall, $e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$ and $e^{-i\omega t} = \cos(\omega t) - i\sin(\omega t)$.

B

The Dynamics of Kinematic Chains

Newton's laws are sufficient to describe the dynamics of kinematic chains and whole-body mechanisms comprised of open, closed, serial, and parallel chain mechanisms. We begin by deriving the three-dimensional inertial counterpart of mass in rotating systems.

B.1 Deriving the Inertia Tensor

A tensor is a geometric operator that describes relations between vectors and is independent of any particular choice for the coordinate system. The dot product and cross product are tensors, as is the inertia tensor that projects vectors of torques onto the vector space of accelerations.

The analysis in section 6.2 yields an expression for the scalar moment of inertia about a single axis of rotation by examining the relationship between torque and angular acceleration in a planar lamina. In this section, we complete this derivation to construct the 3×3 inertia tensor that describes a rotational inertial quantity about three orthogonal—often dynamically coupled—axes by summing the contribution of a stack of the lamina introduced in section 6.2 to arrive at the inertia tensor describing the total moment of inertia of a three-dimensional body about an arbitrary axis $\hat{a} \in \mathbb{R}^3$.

The right side of [figure B.1](#) shows the j th laminar element alone and considers the contribution of all the particles m_k from this lamina to the total rotational inertia of the body about axis \hat{a} .

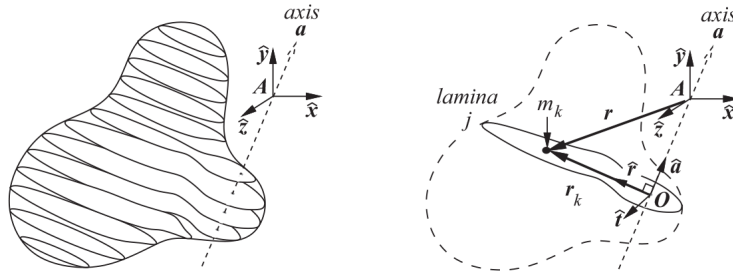


Figure B.1

(Left, A stack of planar lamina approximates the distribution of mass in a three-dimensional object. (Right) The contribution of a single lamina to the inertia tensor describing rotation about \hat{a} .

Equation 6.1 states that the mass moment of inertia $J = \sum_k m_k r_k^2$ is the inertial contribution of the entire lamina. This result applies to the j th lamina pictured in figure B.1 too; however, in this case, the moment arm is

$$\mathbf{r}_k = \mathbf{r} - (\mathbf{r}^T \hat{\mathbf{a}}) \hat{\mathbf{a}},$$

so that the squared magnitude r_k^2 required for equation 6.1 is

$$r_k^2 = \mathbf{r}^T \mathbf{r} - (\mathbf{r}^T \hat{\mathbf{a}})^T (\mathbf{r}^T \hat{\mathbf{a}}).$$

From figure B.1, the first term on the right is the squared magnitude of the position vector \mathbf{r} , and the second term is the squared magnitude of the projection of \mathbf{r} onto $\hat{\mathbf{a}}$. The difference is, therefore, the squared magnitude of \mathbf{r}_k .

The (total) inertia tensor sums terms like this over all masses m_k in a lamina and over all j laminae,

$${}^A J_{aa} = \sum_j \sum_k m_k [\mathbf{r}^T \mathbf{r} - (\mathbf{r}^T \hat{\mathbf{a}})^T (\mathbf{r}^T \hat{\mathbf{a}})], \quad (\text{B.1})$$

where the leading superscript A reminds us that the center of rotation is located at the origin of frame A .

The value of ${}^A J_{aa}$ depends on the distribution of mass and on the axis of rotation $\hat{\mathbf{a}}$. To separate these independent concerns, equation B.1 is written in a slightly different way.

$$\begin{aligned}
{}^A J_{aa} &= \sum_j \sum_k m_k [\mathbf{r}^T \mathbf{r} - (\mathbf{r}^T \hat{\mathbf{a}})^T (\mathbf{r}^T \hat{\mathbf{a}})] \\
&= \sum_j \sum_k m_k [(\hat{\mathbf{a}}^T \hat{\mathbf{a}})(\mathbf{r}^T \mathbf{r}) - (\hat{\mathbf{a}}^T \mathbf{r})(\mathbf{r}^T \hat{\mathbf{a}})] \\
&= \sum_j \sum_k m_k [\hat{\mathbf{a}}^T [(\mathbf{r}^T \mathbf{r}) \mathbf{I}_3] \hat{\mathbf{a}} - \hat{\mathbf{a}}^T (\mathbf{r} \mathbf{r}^T) \hat{\mathbf{a}}], \tag{B.2}
\end{aligned}$$

where \mathbf{I}_3 is the 3×3 identity matrix. In this form, the axis of rotation, $\hat{\mathbf{a}}$, can be moved outside the double summation,

$$\begin{aligned}
&= \hat{\mathbf{a}}^T [{}^A \mathbf{J}] \hat{\mathbf{a}}, \text{ where} \\
{}^A \mathbf{J} &= \sum_j \sum_k m_k [(\mathbf{r}^T \mathbf{r}) \mathbf{I}_3 - (\mathbf{r} \mathbf{r}^T)]. \tag{B.3}
\end{aligned}$$

Equation B.3 is the 3×3 positive definite and symmetric *inertia tensor*. It depends only on the mass distribution of the body and the choice of frame \mathbf{A} , but it is independent of the axis of rotation $\hat{\mathbf{a}}$. If $\mathbf{r} = [x \ y \ z]^T$, equation B.3 can be expanded into a commonly used form,

$$\begin{aligned}
{}^A \mathbf{J} &= \sum_j \sum_k m_k \left[\begin{bmatrix} (x^2 + y^2 + z^2) & 0 & 0 \\ 0 & (x^2 + y^2 + z^2) & 0 \\ 0 & 0 & (x^2 + y^2 + z^2) \end{bmatrix} \right. \\
&\quad \left. - \begin{bmatrix} x^2 & xy & xz \\ yx & y^2 & yz \\ zx & zy & z^2 \end{bmatrix} \right], \text{ or} \\
&= \sum_j \sum_k m_k \begin{bmatrix} (y^2 + z^2) & -xy & -xz \\ -yx & (x^2 + z^2) & -yz \\ -zx & -zy & (x^2 + y^2) \end{bmatrix} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix}, \tag{B.4}
\end{aligned}$$

where diagonal elements J_{xx} , J_{yy} , and J_{zz} are called the *moments of inertia* and the off-diagonal elements J_{xy} , J_{xz} , and J_{yz} are the *products of inertia*. In the limit, as the number of laminae increases, the stack approaches a continuous three-dimensional volume in which differential volume ($dx \ dy \ dz$) is associated with differential mass $dm = \rho(dx \ dy \ dz)$, where ρ is the mass

density of the body. Under these conditions, the elements of the inertia tensor are computed as continuous integrals over the object volume.

Mass moments of inertia	Mass products of inertia
${}^A J_{xx} = \int \int \int (y^2 + z^2) \rho \, dx \, dy \, dz$	${}^A J_{yx} = {}^A J_{xy} = \int \int \int (xy) \rho \, dx \, dy \, dz$
${}^A J_{yy} = \int \int \int (x^2 + z^2) \rho \, dx \, dy \, dz$	${}^A J_{zx} = {}^A J_{xz} = \int \int \int (xz) \rho \, dx \, dy \, dz$
${}^A J_{zz} = \int \int \int (x^2 + y^2) \rho \, dx \, dy \, dz$	${}^A J_{zy} = {}^A J_{yz} = \int \int \int (yz) \rho \, dx \, dy \, dz$

There are several procedures for deriving the computed torque equation for more complex mechanisms. This appendix presents two: (1) the Newton-Euler iterations provide the most direct analogy to a free body analysis implemented in a two-step iteration where the state of motion is propagated outward and the load in the mechanism is the subject of an inward iteration to define the complex dynamics of the articulated mechanisms comprising interesting robots. It's principle value is the direct application of Newton's Laws, but in practice it is rather cumbersome, and (2) a closed-form formulation based on Lagrange's equations.

B.2 Inertial Coordinate Frames

The *inertial* reference frame is that special frame. Inertial frames make Newton's laws work. If a particle has no external forces on it, then, by Newton's first law, it should remain in a state of constant rectilinear motion. This will be true for all candidate inertial frames when rotational velocity and acceleration, as well as translational acceleration, for all of these frames is zero (i.e., $\omega = \dot{\omega} = \dot{v} = 0$). To see why, say the particle was stationary in one instant relative to a particular inertial frame. From the perspective of an observer attached to that frame, it should remain stationary unless acted upon by an external force. From the perspective of any noninertial frame (one that might be rotating or accelerating), this will not be true. Even with no external forces on the particle, it will appear to start moving and some *fictitious* forces will have to be invented to account for the apparent (non-Newtonian) acceleration. Coriolis and centrifugal forces owe their existence

to the fact that human physicists make observations from the rotating surface of the Earth and are therefore accelerating with respect to an inertial frame of reference. By the same reasoning, the inertial frame can be moving at a constant absolute velocity (maybe $v = 0$) and still satisfy Newton's first law. The simplest expression of Newton's laws (section 6.1) predicts free body motion correctly only if care is taken to refer the analysis to an appropriately chosen inertial frame.

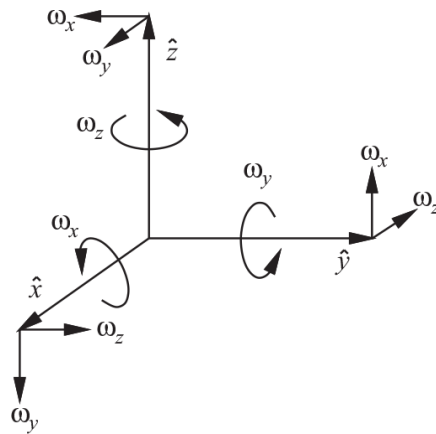
B.3 Rotating Coordinate Systems

When frame B undergoes a rotational velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ with respect to frame A , the rotation matrix ${}_A\mathbf{R}_B(t)$ is a function of time. Consider a rotation about the \hat{z} -axis. The component of $\boldsymbol{\omega}$ about the \hat{z} -axis, ω_z , causes a displacement of the \hat{x} -axis in the \hat{y} direction.

$$d\hat{x}_{\omega_z} = \omega_z \hat{y}$$

Similarly, the ω_y component of the net angular velocity displaces the \hat{x} -axis in the $-\hat{z}$ direction,

$$d\hat{x}_{\omega_y} = -\omega_y \hat{z},$$



so that, in general,

$$\begin{aligned} d\hat{x} &= \omega_z \hat{y} - \omega_y \hat{z} \\ d\hat{y} &= -\omega_z \hat{x} + \omega_x \hat{z} \\ d\hat{z} &= \omega_y \hat{x} - \omega_x \hat{y}. \end{aligned}$$

If frame 1 rotates with a constant angular velocity $\boldsymbol{\omega}$ relative to a fixed frame 0, then for arbitrary vectors $\mathbf{r}_1 = r_x \hat{x} + r_y \hat{y} + r_z \hat{z}$, where r_x, r_y, r_z are constants, the position vector with respect to the stationary frame changes as

$$d\mathbf{r}_0 = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \mathbf{r}_1 \\ = \boldsymbol{\omega} \times \mathbf{r}_1.$$

If we assume that coordinate frame B in [figure B.2](#) is moving with a constant angular velocity $\boldsymbol{\omega}$ with respect to an inertial frame A , then the velocity of particle Q in inertial coordinates ($\dot{\mathbf{r}}_A$) consists of its translational velocity in frame B ($\dot{\mathbf{r}}_B$) plus a term due to the angular velocity of frame B ($\boldsymbol{\omega} \times \mathbf{r}_B$). This second term is due directly to the angular velocity of frame B —an observer attached to frame B cannot correctly account for the forces on particle Q by observing its apparent motion.

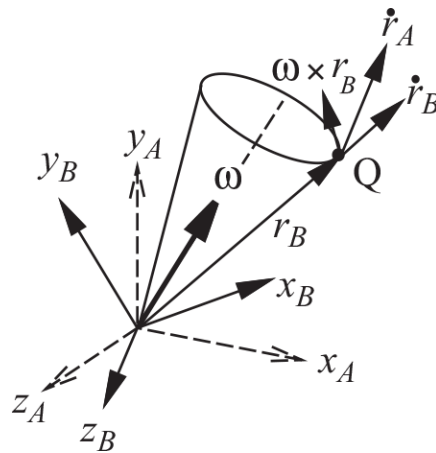


Figure B.2

Velocity in an inertial frame due to a time varying rotation matrix

From another perspective, we may consider coordinate frames A and B to be related by way of a time varying rotation matrix

$$\mathbf{r}_A(t) = {}_A\mathbf{R}_B(t) \mathbf{r}_B(t).$$

The velocity of Q with respect to frame A is then

$$\begin{aligned}
\dot{\mathbf{r}}_A &= \frac{d}{dt} [{}^A\mathbf{R}_B \mathbf{r}_B] \\
&= {}^A\mathbf{R}_B \dot{\mathbf{r}}_B + {}^A\dot{\mathbf{R}}_B \mathbf{r}_B \\
&= {}^A\mathbf{R}_B [\dot{\mathbf{r}}_B + (\boldsymbol{\omega}_B \times \mathbf{r}_B)].
\end{aligned} \tag{B.5}$$

In the final expression for $\dot{\mathbf{r}}_A$, we have expressed the absolute angular velocity of frame B in frame B coordinates for notational convenience.

Again we find that the velocity of particle Q relative to the inertial frame consists of the translational velocity of Q within the B coordinate system plus a term inherited from frame B 's state of motion. This result holds for any vector quantity that is expressed in a local coordinate system that is itself rotating with respect to an inertial frame of reference.

$$\frac{d}{dt} [{}^A\mathbf{R}_B(\cdot)_B] = {}^A\mathbf{R}_B \left[\frac{d}{dt} (\cdot)_B + (\boldsymbol{\omega}_B \times (\cdot)_B) \right] \tag{B.6}$$

Using this rule to differentiate [equation B.5](#), and assuming that ${}^A\mathbf{R}_B \dot{\boldsymbol{\omega}}_B = 0$,¹ we arrive at the equation of the total acceleration of particle Q in terms of local observations of its position, velocity, and acceleration in frame B .

$$\begin{aligned}
\ddot{\mathbf{r}}_A &= {}^A\dot{\mathbf{R}}_B [\dot{\mathbf{r}}_B + (\boldsymbol{\omega}_B \times \mathbf{r}_B)] + {}^A\mathbf{R}_B [\ddot{\mathbf{r}}_B + (\dot{\boldsymbol{\omega}}_B \times \mathbf{r}_B) + (\boldsymbol{\omega}_B \times \dot{\mathbf{r}}_B)] \\
&= {}^A\mathbf{R}_B [(\boldsymbol{\omega}_B \times \dot{\mathbf{r}}_B) + (\boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}_B)) + \ddot{\mathbf{r}}_B + (\boldsymbol{\omega}_B \times \dot{\mathbf{r}}_B)],
\end{aligned}$$

so, in terms of position \mathbf{r} , velocity $\mathbf{v} = \dot{\mathbf{r}}$, and acceleration $\mathbf{a} = \ddot{\mathbf{r}}$, we get

$$\mathbf{a}_A = {}^A\mathbf{R}_B [\mathbf{a}_B + 2(\boldsymbol{\omega}_B \times \mathbf{v}_B) + (\boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}_B))],$$

where the first term on the right side of the equation is the true acceleration of particle Q expressed in the noninertial (rotating) frame, the second term is the fictitious Coriolis acceleration, and the third term is the fictitious centrifugal acceleration.

Rearranging terms to solve for the acceleration visible to an observer attached to frame B , we find that

$$\mathbf{a}_B = {}^B\mathbf{R}_A \mathbf{a}_A - 2(\boldsymbol{\omega}_B \times \mathbf{v}_B) - (\boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}_B)). \tag{B.7}$$

B.4 Newton-Euler Iterations

Figure B.3 presents a generic link that is part of a multiple degree of freedom robot arm.² The link is assumed to be a rigid body that can incorporate an actuated degree of freedom θ_i . The local coordinate frame for link i is attached to the proximal end of the link, with the \hat{z}_i axis aligned with a joint axis and \hat{x}_i aligned with the link itself. The location of the center of mass and the velocities, accelerations, and forces describing the movement of the link are expressed in this local frame.

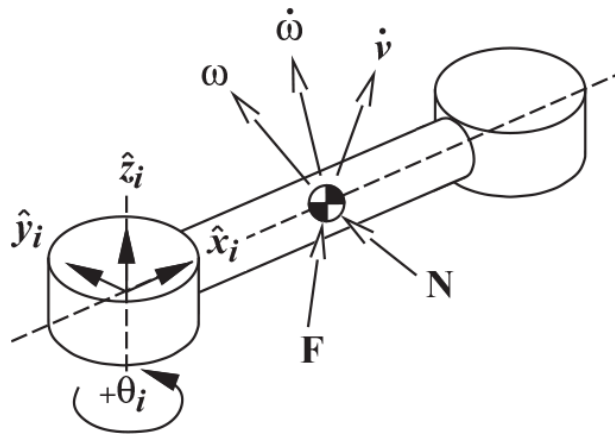


Figure B.3

Free body diagram of link i

Link i is part of a kinematic chain and, therefore, subject to interaction forces with neighboring links, but we will analyze the motion of the link as if it is a Newtonian free body whose center of mass is in a known absolute state of motion $(\omega, \dot{\omega}, \dot{v})$ and subject to a net force F and moment N . These hypothetical loads are used to *explain* the observed state of motion—if there are nonzero accelerations $(\dot{\omega}, \dot{v})$, then F and N are the net forces and moments required to produce these accelerations given the free body assumption and the link's inertial properties.

The net force and the time-rate-of-change in velocity are related through Newton's equation

$$F = \frac{d}{dt} (mv) = m\dot{v}, \quad (\text{B.8})$$

which states that the force acting on this free body is equivalent to the time-rate-of-change in the body's linear momentum. In the nomenclature of

figure B.3, the quantity $\dot{\mathbf{v}}$ is expressed in the local coordinate frame, which is itself rotating. In particular, if frame i in figure B.3 has angular velocity $\mathbf{R}_i(t)\boldsymbol{\omega}_i$ with respect to the inertial frame, then

$$\begin{aligned}\mathbf{F} &= \frac{d}{dt} (\mathbf{R}_i m_i \mathbf{v}_i) = \mathbf{R}_i m_i \dot{\mathbf{v}}_i + \dot{\mathbf{R}}_i m_i \mathbf{v}_i \\ &= \mathbf{R}_i [m_i \dot{\mathbf{v}}_i + (\boldsymbol{\omega}_i \times m_i \mathbf{v}_i)]\end{aligned}\tag{B.9}$$

is Newton's law for body i , where \mathbf{F} is the net force acting upon link i written in absolute/inertial coordinates, \mathbf{R}_i is the rotation matrix relating frame i to the inertial frame, and $\boldsymbol{\omega}_i$ is the absolute angular velocity of link i written in link i coordinates. The equivalent relation between angular acceleration and net torque incorporates the inertia tensor for the link.

$$\begin{aligned}\mathbf{N} &= \frac{d}{dt} (\mathbf{R}_i \mathbf{I}_i \boldsymbol{\omega}_i) \\ &= \mathbf{R}_i [\mathbf{I}_i \dot{\boldsymbol{\omega}}_i + (\boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i)]\end{aligned}\tag{B.10}$$

is Euler's equation for body i , where \mathbf{N} is the net torque acting upon link i written in inertial coordinates, and \mathbf{R}_i is the rotation matrix relating frame i to the inertial frame.

B.4.1 Propagating Velocities in Open Kinematic Chains

If we define the complete state of motion in a mechanism—that is, all positions, velocities, and accelerations in the inertial frame—then we can use Newton's laws to compute the forces in the structure resulting from this state of motion. The Newton-Euler equations provide an iterative means of propagating the absolute state of motion $(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \dot{\mathbf{v}})$ from link i to link $i+1$. Link $i+1$ inherits the absolute state of motion of the link i and adds velocities and accelerations from the i th actuator to arrive at the absolute state of motion for link $i+1$. The iteration begins at the inertial frame where $(\boldsymbol{\omega}_0 = \mathbf{0}, \dot{\boldsymbol{\omega}}_0 = \mathbf{0}, \dot{\mathbf{v}}_0 = \mathbf{0})$ and proceeds link by link until it reaches the distal end of the manipulator.

In the following derivations, we will assume that the absolute state of motion $(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \dot{\mathbf{v}})_i$ is known, and our goal is to write expressions for these quantities at frame $(i+1)$.

Angular velocity: ω

$$\begin{aligned} \text{REVOLUTE: } \quad {}^{i+1}\omega_{i+1} &= {}_{i+1}\mathbf{R}_i \dot{\omega}_i + \dot{\theta}_{i+1} \hat{z}_{i+1} \\ \text{PRISMATIC: } \quad {}^{i+1}\omega_{i+1} &= {}_{i+1}\mathbf{R}_i \dot{\omega}_i \end{aligned} \quad (\text{B.11})$$

Angular acceleration: $\dot{\omega}$

$$\begin{aligned} \text{REVOLUTE: } \quad {}^{i+1}\dot{\omega}_{i+1} &= {}_{i+1}\mathbf{R}_i \dot{\omega}_i + (-\dot{\theta}_{i+1} \hat{z}_{i+1} \times {}_{i+1}\mathbf{R}_i \dot{\omega}_i) + \ddot{\theta}_{i+1} \hat{z}_{i+1} \\ &= {}_{i+1}\mathbf{R}_i \dot{\omega}_i + ({}_{i+1}\mathbf{R}_i \dot{\omega}_i \times \dot{\theta}_{i+1} \hat{z}_{i+1}) + \ddot{\theta}_{i+1} \hat{z}_{i+1} \\ \text{PRISMATIC: } \quad {}^{i+1}\dot{\omega}_{i+1} &= {}_{i+1}\mathbf{R}_i \dot{\omega}_i \end{aligned} \quad (\text{B.12})$$

Linear acceleration: \ddot{p}

Consider the motion of particle q moving with respect to coordinate frame 1, which is moving in turn with respect to *inertial* frame 0 as in figure B.4.

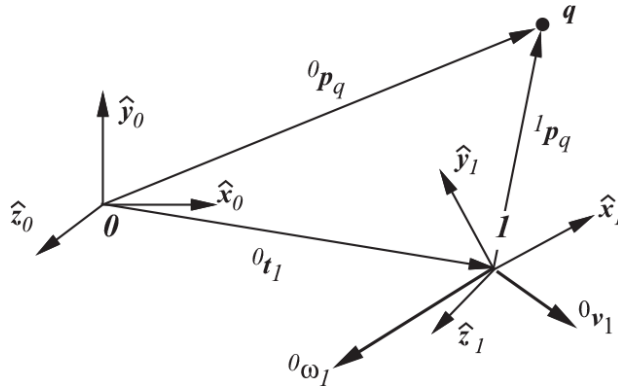


Figure B.4

Propagating velocity and acceleration into a noninertial coordinate frame

Here, ${}^0\omega_1$ and 0v_1 are the angular and translational velocity, respectively, of frame 1 expressed in frame 0 coordinates.

Given 1p_q , the position vector for point q in frame 1 coordinates, then

$${}^0p_q = {}^0\mathbf{R}_1 {}^1p_q + {}^0t_1,$$

where 0t_1 is just the position vector of frame 1 written in frame 0 coordinates. For general velocities in $SE(3)$, the velocity of particle q

expressed in frame 0 can be written

$${}^0\mathbf{v}_q = {}^0\mathbf{R}_1 {}^1\dot{\mathbf{p}}_q + ({}^0\dot{\boldsymbol{\omega}}_1 \times {}^0\mathbf{R}_1 {}^1\mathbf{p}_q) + {}^0\mathbf{v}_1.$$

The linear acceleration of particle q is then

$$\begin{aligned} {}^0\dot{\mathbf{v}}_q &= \frac{d}{dt} \left[{}^0\mathbf{R}_1 {}^1\dot{\mathbf{p}}_q \right] + ({}^0\dot{\boldsymbol{\omega}}_1 \times {}^0\mathbf{R}_1 {}^1\mathbf{p}_q) + ({}^0\boldsymbol{\omega}_1 \times \frac{d}{dt} \left[{}^0\mathbf{R}_1 {}^1\mathbf{p}_q \right]) + {}^0\dot{\mathbf{v}}_1 \\ &= {}^0\mathbf{R}_1 {}^1\ddot{\mathbf{p}}_q + ({}^0\boldsymbol{\omega}_1 \times {}^0\mathbf{R}_1 {}^1\dot{\mathbf{p}}_q) + ({}^0\dot{\boldsymbol{\omega}}_1 \times {}^0\mathbf{R}_1 {}^1\mathbf{p}_q) + ({}^0\boldsymbol{\omega}_1 \times {}^0\mathbf{R}_1 {}^1\dot{\mathbf{p}}_q) \\ &\quad + ({}^0\boldsymbol{\omega}_1 \times {}^0\boldsymbol{\omega}_1 \times {}^0\mathbf{R}_1 {}^1\mathbf{p}_q) + {}^0\dot{\mathbf{v}}_1. \end{aligned}$$

We can write this equation for each link, i , by substituting i for frame 1, $(i-1)$ for frame 0, and $(i+1)$ for particle q .

$$\begin{aligned} {}^{i+1}\dot{\mathbf{v}}_{i+1} &= {}^{i+1}\mathbf{R}_{i-1} \left[{}^{i-1}\mathbf{R}_i {}^i\ddot{\mathbf{p}}_{i+1} + 2({}^{i-1}\boldsymbol{\omega}_i \times {}^{i-1}\mathbf{R}_i {}^i\dot{\mathbf{p}}_{i+1}) + ({}^{i-1}\dot{\boldsymbol{\omega}}_i \times {}^{i-1}\mathbf{R}_i {}^i\mathbf{p}_{i+1}) \right. \\ &\quad \left. + ({}^{i-1}\boldsymbol{\omega}_i \times {}^{i-1}\boldsymbol{\omega}_i \times {}^{i-1}\mathbf{R}_i {}^i\mathbf{p}_{i+1}) + {}^{i-1}\dot{\mathbf{v}}_i \right] \\ &= {}^{i+1}\mathbf{R}_i \left[{}^i\ddot{\mathbf{p}}_{i+1} + 2({}^i\boldsymbol{\omega}_i \times {}^i\dot{\mathbf{p}}_{i+1}) + ({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{p}_{i+1}) \right. \\ &\quad \left. + ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{p}_{i+1}) + {}^i\dot{\mathbf{v}}_i \right] \end{aligned} \tag{B.13}$$

Equation B.13 can be simplified given the type (prismatic or revolute) of the $(i+1)$ st degree of freedom.

$$\begin{aligned} \text{REVOLUTE: } & {}^i\mathbf{p}_{i+1} = \text{const}, \quad {}^i\dot{\mathbf{p}}_{i+1} = {}^i\ddot{\mathbf{p}}_{i+1} = 0 \\ & {}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R}_i \left[{}^i\dot{\mathbf{v}}_i + ({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{p}_{i+1}) + ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{p}_{i+1}) \right] \\ \text{PRISMATIC: } & {}^i\mathbf{p}_{i+1} = d_i\hat{\mathbf{x}}_i, \quad {}^i\dot{\mathbf{p}}_{i+1} = \dot{d}_i\hat{\mathbf{x}}_i, \quad {}^i\ddot{\mathbf{p}}_{i+1} = \ddot{d}_i\hat{\mathbf{x}}_i \\ & {}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R}_i \left[{}^i\dot{\mathbf{v}}_i + \ddot{d}_i\hat{\mathbf{x}}_i + 2({}^i\boldsymbol{\omega}_i \times \dot{d}_i\hat{\mathbf{x}}_i) + ({}^i\dot{\boldsymbol{\omega}}_i \times d_i\hat{\mathbf{x}}_i) \right. \\ & \quad \left. + ({}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{\omega}_i \times d_i\hat{\mathbf{x}}_i) \right] \end{aligned} \tag{B.14}$$

Now we have established the complete state of motion $(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \dot{\mathbf{v}})$ for link i . Next, we project the accelerations at frame i to the link's center of mass where we can use the Newton-Euler equations to solve for the total acceleration, $(\dot{\boldsymbol{\omega}}, \dot{\mathbf{v}})$, and consequently the net forces and moments (\mathbf{F}, \mathbf{N}) that are consistent with these accelerations.

$${}^{i+1}\dot{\mathbf{v}}_{cm,(i+1)} = ({}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} \times {}^{i+1}\mathbf{p}_{cm}) + ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{p}_{cm}) + {}^{i+1}\dot{\mathbf{v}}_{i+1} \quad (\text{B.15})$$

$${}^{i+1}\mathbf{F}_{i+1} = m_{i+1} {}^{i+1}\dot{\mathbf{v}}_{cm,(i+1)} \quad (\text{B.16})$$

$${}^{i+1}\mathbf{N}_{i+1} = \mathbf{I}_{i+1} {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} + ({}^{i+1}\boldsymbol{\omega}_{i+1} \times \mathbf{I}_{i+1} {}^{i+1}\boldsymbol{\omega}_{i+1}) \quad (\text{B.17})$$

B.4.2 Propagating Force in Open Kinematic Chains

Section B.4.1 establishes that velocities are inherited from link to link starting from an inertial frame and culminating in the velocity at the end of the kinematic chain. The same is true of forces in the mechanism. If our mechanism is moving in free space, then the distal unitary link, link n , in the chain sees only those forces accounted for in its state of motion and applied to its proximal end by link $n-1$.

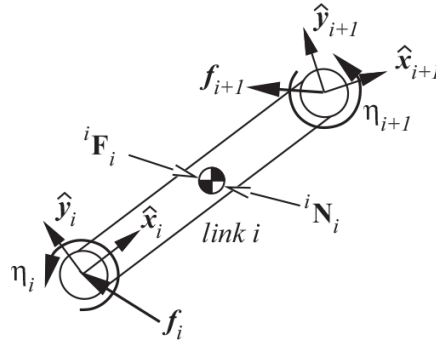


Figure B.5

The propagation of forces in a kinematic chain

As we move along the kinematic chain, distal to proximal, each successive link inherits a component of force applied to its distal end. Just as velocity propagates from the inertial frame (where velocities are zero) outward, forces propagate from the free end of the manipulator (where forces are zero) inward. [Figure B.5](#) shows link i in a kinematic chain. We define ${}^i\mathbf{f}_i$ and ${}^i\boldsymbol{\eta}_i$ as the force and torque, respectively, exerted on link i by link $i-1$. Furthermore, the superscript designates that these quantities are written in frame i coordinates. If we assume that the structure is in static equilibrium, then

$$\begin{aligned}\sum Forces &= {}^i\mathbf{F}_i = {}^i\mathbf{f}_i - {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1}, \text{ or} \\ {}^i\mathbf{f}_i &= {}^i\mathbf{F}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1}\end{aligned}\quad (\text{B.18})$$

and

$$\sum Torques = {}^i\mathbf{N}_i = {}^i\boldsymbol{\eta}_i - {}^i\boldsymbol{\eta}_{i+1} - ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{f}_i) - (({}^i\mathbf{p}_{i+1} - {}^i\mathbf{p}_{cm}) \times {}^i\mathbf{f}_{i+1}),$$

but ${}^i\mathbf{f}_i = {}^i\mathbf{F}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1}$ by equation B.18 so that

$$\begin{aligned}{}^i\mathbf{N}_i &= {}^i\boldsymbol{\eta}_i - {}^i\boldsymbol{\eta}_{i+1} - ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{F}_i) - ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{f}_{i+1}) - ({}^i\mathbf{p}_{i+1} \times {}^i\mathbf{f}_{i+1}) + ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{f}_{i+1}) \\ &= {}^i\boldsymbol{\eta}_i - {}^i\boldsymbol{\eta}_{i+1} - ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{F}_i) - ({}^i\mathbf{p}_{i+1} \times {}^i\mathbf{f}_{i+1})\end{aligned}$$

or

$${}^i\boldsymbol{\eta}_i = {}^i\mathbf{N}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\boldsymbol{\eta}_{i+1} + ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{F}_i) + ({}^i\mathbf{p}_{i+1} \times {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1}). \quad (\text{B.19})$$

To summarize, the Newton-Euler equations are solved iteratively starting from the inertial frame ($(\dot{\mathbf{v}} = \dot{\boldsymbol{\omega}} = 0)$) and proceeding from proximal to distal. This procedure propagates velocities and accelerations from link to link and solves for the net force (${}^i\mathbf{F}_i$) and moment (${}^i\mathbf{N}_i$) on the center of mass of each link in the kinematic chain. Having finished this *outward* iteration, we begin an *inward* iteration starting from the free (distal) end of the manipulator, where ${}^{i+1}\mathbf{f}_{i+1} = {}^{i+1}\boldsymbol{\eta}_{i+1} = 0$. Equations B.18 and B.19 are applied to resolve the force and torque on the proximal end of link i . The forces and moments at frame i result in mechanical strains in a link or a joint, or they produce translational or rotational accelerations in the system. In the case of revolute joint i with axis of rotation about the $\hat{\mathbf{z}}_i$ -axis, the component of the torque about the $\hat{\mathbf{z}}_i$ -axis,

$$\boldsymbol{\tau}_i = {}^i\boldsymbol{\eta}_i \cdot \hat{\mathbf{z}}_i, \quad (\text{B.20})$$

produces an angular acceleration of the joint, whereas other components of the torque at frame i are resisted in the mechanism and propagated into other portions of the structure.

B.4.3 The Outward-Inward Iteration

The Newton-Euler method begins with the designation of an inertial (or absolute) frame of reference where the complete state of motion is defined. Typically, it is adequate to use the coordinate frame where the robot is attached to the Earth.³ In [figure B.6](#), the complete state of motion of frame 0 ($\omega, \dot{\omega}, \dot{v}$)₀ must be known in inertial coordinates. It serves as the boundary condition for the Newton-Euler iteration and establishes the Newtonian framework for the rest of the analysis.

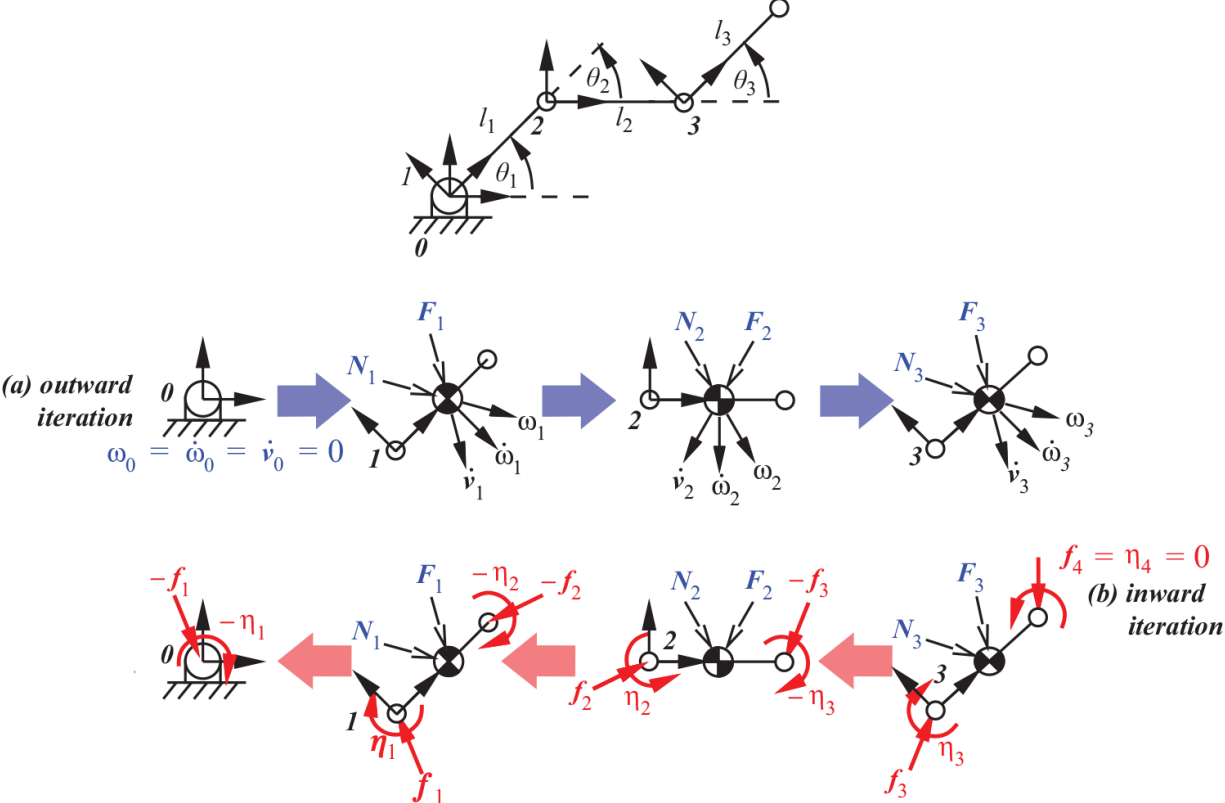


Figure B.6 Stages in the Newton-Euler iterations for a 3R planar manipulator ($n=3$). The equations defining the iteration denoted by the blue and red block arrows are summarized in [table B.1](#).

Table B.1 Newton-Euler iterations.



Outward iterations

Angular velocity: ω

Equation B.11

$$\text{REVOLUTE: } {}^{i+1}\omega_{i+1} = {}^{i+1}\mathbf{R}_i {}^i\omega_i + \dot{\theta}_{i+1}\hat{z}_{i+1}$$

$$\text{PRISMATIC: } {}^{i+1}\omega_{i+1} = {}^{i+1}\mathbf{R}_i {}^i\omega_i$$

Angular acceleration: $\dot{\omega}$

Equation B.12

$$\text{REVOLUTE: } {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}\mathbf{R}_i {}^i\dot{\omega}_i + ({}^{i+1}\mathbf{R}_i {}^i\omega_i \times \dot{\theta}_{i+1}\hat{z}_{i+1}) + \ddot{\theta}_{i+1}\hat{z}_{i+1}$$

$$\text{PRISMATIC: } {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}\mathbf{R}_i {}^i\dot{\omega}_i$$

Linear Acceleration: \dot{v}

Equation B.14

$$\text{REVOLUTE: } {}^{i+1}\dot{v}_{i+1} = {}^{i+1}\mathbf{R}_i [{}^i\dot{v}_i + ({}^i\dot{\omega}_i \times {}^i\mathbf{p}_{i+1}) + ({}^i\omega_i \times {}^i\omega_i \times {}^i\mathbf{p}_{i+1})]$$

$$\text{PRISMATIC: } {}^{i+1}\dot{v}_{i+1} = {}^{i+1}\mathbf{R}_i [{}^i\dot{v}_i + \ddot{d}_i\hat{x}_i + 2({}^i\omega_i \times \dot{d}_i\hat{x}_i) + ({}^i\dot{\omega}_i \times d_i\hat{x}_i) + ({}^i\omega_i \times {}^i\omega_i \times d_i\hat{x}_i)]$$

Linear acceleration (center of mass): \dot{v}_{cm}

Equation B.15

$${}^{i+1}\dot{v}_{cm,(i+1)} = ({}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}\mathbf{p}_{cm}) + ({}^{i+1}\omega_{i+1} \times {}^{i+1}\omega_{i+1} \times {}^{i+1}\mathbf{p}_{cm}) + {}^{i+1}\dot{v}_{i+1}$$

Net force: \mathbf{F}

Equation B.16

$${}^{i+1}\mathbf{F}_{i+1} = m_{i+1} {}^{i+1}\dot{v}_{cm,(i+1)}$$

Net moment: \mathbf{N}

Equation B.17

$${}^{i+1}\mathbf{N}_{i+1} = \mathbf{I}_{i+1} {}^{i+1}\dot{\omega}_{i+1} + ({}^{i+1}\omega_{i+1} \times \mathbf{I}_{i+1} {}^{i+1}\omega_{i+1})$$



Inward iterations

Interlink forces

Equation B.18

$${}^i\mathbf{f}_i = {}^i\mathbf{F}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1}$$

Interlink moments

Equation B.19

$${}^i\boldsymbol{\eta}_i = {}^i\mathbf{N}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\boldsymbol{\eta}_{i+1} + ({}^i\mathbf{p}_{cm} \times {}^i\mathbf{F}_i) + ({}^i\mathbf{p}_{i+1} \times {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1})$$

The *outward* iteration propagates the known state of motion at the inertial frame outward into successive links. At each step, the absolute state of motion for link i is the sum of the state of motion for link $(i-1)$ and the local state of motion of the i th degree of freedom. When the outward

iteration is completed, the complete state of motion $(\omega, \dot{\omega}, \dot{v})_i$ is established for every link in a manner that respects kinematic constraints in the mechanism—constraints associated with joint forces that preserve the connection between adjacent links in the kinematic chain. However, we do not have enough information, yet, to determine what these joint forces are. If we treat each link of the robot as a free body, however, then Newton’s second law can determine the net (hypothetical) force \mathbf{F} and moment \mathbf{N} that must be applied to each link at the center of mass to explain the state of motion determined by the outward iteration.

The *inward* iteration is used to propagate forces and torques from the endpoint of a manipulator back to the inertial frame. This iteration begins at the free end of the manipulator and proceeds from the $(n+1)$ st coordinate frame back to the inertial frame. It requires a boundary condition at frame $n+1$ to define contact forces and torques on the endpoint of the manipulator. Normally, for a manipulator in free space (no contact), this boundary condition specifies $\mathbf{f}_{n+1} = \boldsymbol{\eta}_{n+1} = 0$; however, it could be established using contact sensing. In the course of the inward iteration, the net force \mathbf{F}_i and torque \mathbf{N}_i computed in the outward iteration is equated to the sum of the interaction forces, $(\mathbf{f}, \boldsymbol{\eta})_i$ and $(\mathbf{f}, \boldsymbol{\eta})_{i+1}$, between adjacent links.

Example: The Computed Torque Equation for the Planar 2R Manipulator

A planar 2R manipulator is modeled as two point masses m_1 and m_2 at the end of two massless links l_1 and l_2 . This choice simplifies the derivation of the computed torque equation because for this distribution of mass, the scalar moment of inertia about the \hat{z} -axis at the center of mass is

$$I_1 = I_2 = 0.$$

In addition to a description of the mass distribution, the Newton-Euler equations require a kinematic specification of the manipulator. In this case, it requires the position vectors from frame i to frame $i+1$ and from frame i to the center of mass of link i ,

$${}^1\mathbf{p}_2 = {}^1\mathbf{p}_{cm} = l_1\hat{\mathbf{x}}_1$$

$${}^2\mathbf{p}_3 = {}^2\mathbf{p}_{cm} = l_2\hat{\mathbf{x}}_2,$$

where the vector quantity (in this case \mathbf{p}) uses the trailing subscript to indicate a spatial reference (i.e., the origin of a coordinate frame or the center of mass) and the leading superscript identifies the coordinate frame in which it is written.

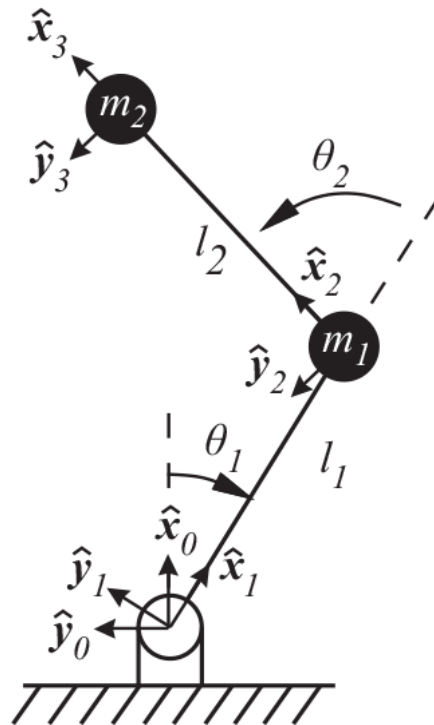


Figure B.7
The planar 2R manipulator with discrete mass distribution

Finally, the rotation matrices for use in the Newton-Euler iterations are pure rotations about the $\hat{\mathbf{z}}$ axes,

$${}^i\mathbf{R}_{i+1} = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0 \\ s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad {}^{i+1}\mathbf{R}_i = \begin{bmatrix} c_{i+1} & s_{i+1} & 0 \\ -s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

To establish a Newtonian framework, the analysis requires that the absolute state of motion of the inertial frame (frame 0) must be specified as a boundary condition for the outward iteration. Generally, this would be

considered a nonrotating ($\omega = 0$) and nonaccelerating inertial frame ($\dot{\omega} = \dot{v} = 0$). However, the affects of a gravitational acceleration can be incorporated directly into the choice of boundary conditions.

The *equivalence principle* states that the motion of a free body subject to gravitational acceleration $-g\hat{x}_0$ cannot be distinguished from the motion of the same free body subject to no external forces whose frame of reference is accelerating in the opposite direction $g\hat{x}_0$. These cases are illustrated in [figure B.8](#) (a) and (b), respectively. In this example, the equivalence principle is used to specify boundary conditions at the inertial frame ${}^0\omega_0 = {}^0\dot{\omega}_0 = \mathbf{0}$ and ${}^0\dot{v}_0 = g\hat{x}_0$ that will be propagated into link 1 and link 2 using the outward iteration equations summarized in [table B.1](#).

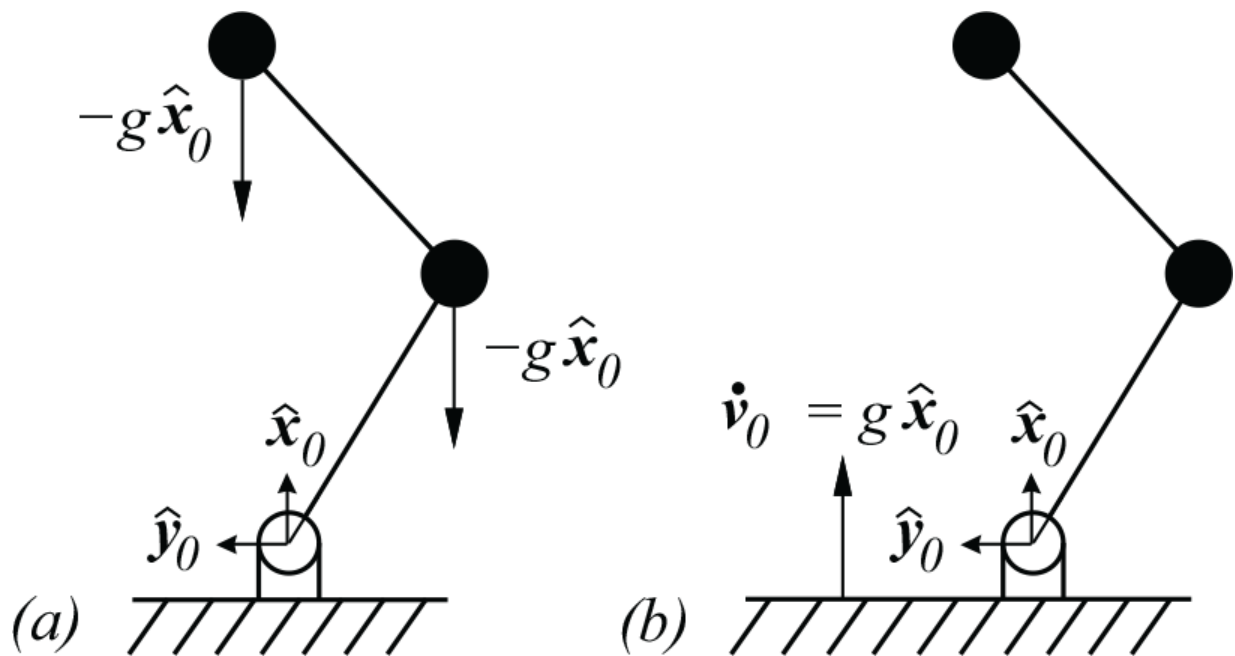


Figure B.8

The equivalence principle applied to the planar 2R manipulator

Outward iteration

Link 1: ${}^1\boldsymbol{\omega}_1 = {}_1\mathbf{R}_0 {}^0\boldsymbol{\omega}_0 + \dot{\theta}_1 \hat{\mathbf{z}}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}$

$${}^1\dot{\boldsymbol{\omega}}_1 = {}_1\mathbf{R}_0 {}^0\dot{\boldsymbol{\omega}}_0 + ({}_1\mathbf{R}_0 {}^0\boldsymbol{\omega}_0 \times \dot{\theta}_1 \hat{\mathbf{z}}_1) + \ddot{\theta}_1 \hat{\mathbf{z}}_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix}$$

$${}^1\dot{\mathbf{v}}_1 = {}_1\mathbf{R}_0 \left[{}^0\dot{\mathbf{v}}_0 + ({}^0\dot{\boldsymbol{\omega}}_0 \times {}^0\mathbf{p}_1) + ({}^0\boldsymbol{\omega}_0 \times {}^0\boldsymbol{\omega}_0 \times {}^0\mathbf{p}_1) \right] = {}_1\mathbf{R}_0 {}^0\dot{\mathbf{v}}_0$$

$$= \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} gc_1 \\ -gs_1 \\ 0 \end{bmatrix}$$

$${}^1\dot{\mathbf{v}}_{cm} = ({}^1\dot{\boldsymbol{\omega}}_1 \times {}^1\mathbf{p}_{cm}) + ({}^1\boldsymbol{\omega}_1 \times {}^1\boldsymbol{\omega}_1 \times {}^1\mathbf{p}_{cm}) + {}^1\dot{\mathbf{v}}_1$$

$$= \begin{vmatrix} i & j & k \\ 0 & 0 & \ddot{\theta}_1 \\ l_1 & 0 & 0 \end{vmatrix} + {}^1\boldsymbol{\omega}_1 \times \begin{vmatrix} i & j & k \\ 0 & 0 & \dot{\theta}_1 \\ l_1 & 0 & 0 \end{vmatrix} + \begin{bmatrix} gc_1 \\ -gs_1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ l_1 \ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{vmatrix} i & j & k \\ 0 & 0 & \dot{\theta}_1 \\ 0 & l_1 \dot{\theta}_1 & 0 \end{vmatrix} + \begin{bmatrix} gc_1 \\ -gs_1 \\ 0 \end{bmatrix} = \begin{bmatrix} gc_1 - l_1 \dot{\theta}_1^2 \\ -gs_1 + l_1 \ddot{\theta}_1 \\ 0 \end{bmatrix}$$

$${}^1\mathbf{F}_1 = m_1 {}^1\dot{\mathbf{v}}_{cm} = m_1 \begin{bmatrix} gc_1 - l_1 \dot{\theta}_1^2 \\ -gs_1 + l_1 \ddot{\theta}_1 \\ 0 \end{bmatrix}$$

$${}^1\mathbf{N}_1 = I_1 {}^1\dot{\boldsymbol{\omega}}_1 + ({}^1\boldsymbol{\omega}_1 \times I_1 {}^1\boldsymbol{\omega}_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Link 2: ${}^2\boldsymbol{\omega}_2 = {}_2\mathbf{R}_1^{-1}\boldsymbol{\omega}_1 + \dot{\theta}_2\hat{\mathbf{z}}_2$

$$= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$${}^2\dot{\boldsymbol{\omega}}_2 = {}_2\mathbf{R}_1^{-1}\dot{\boldsymbol{\omega}}_1 + ({}_2\mathbf{R}_1^{-1}\boldsymbol{\omega}_1 \times \dot{\theta}_2\hat{\mathbf{z}}_2) + \ddot{\theta}_2\hat{\mathbf{z}}_2$$

$$= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} + \left(\begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} + \begin{vmatrix} i & j & k \\ 0 & 0 & \dot{\theta}_1 \\ 0 & 0 & \dot{\theta}_2 \end{vmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$$\begin{aligned}
{}^2\dot{\mathbf{v}}_2 &= {}_2\mathbf{R}_1 \left[{}^1\dot{\mathbf{v}}_1 + ({}^1\dot{\boldsymbol{\omega}}_1 \times {}^1\mathbf{p}_2) + ({}^1\boldsymbol{\omega}_1 \times {}^1\boldsymbol{\omega}_1 \times {}^1\mathbf{p}_2) \right] = {}_2\mathbf{R}_1 {}^1\dot{\mathbf{v}}_1 \\
&= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} gc_1 \\ -gs_1 \\ 0 \end{bmatrix} + \begin{vmatrix} i & j & k \\ 0 & 0 & \ddot{\theta}_1 \\ l_1 & 0 & 0 \end{vmatrix} + {}^1\boldsymbol{\omega}_1 \times \begin{vmatrix} i & j & k \\ 0 & 0 & \dot{\theta}_1 \\ l_1 & 0 & 0 \end{vmatrix} \right) \\
&= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} gc_1 \\ -gs_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ l_1\ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{vmatrix} i & j & k \\ 0 & 0 & \dot{\theta}_1 \\ 0 & l_1\dot{\theta}_1 & 0 \end{vmatrix} \right) \\
&= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} gc_1 - l_1\dot{\theta}_1^2 \\ -gs_1 + l_1\ddot{\theta}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} gc_{12} - l_1\dot{\theta}_1^2 c_2 + l_1\ddot{\theta}_1 s_2 \\ -gs_{12} + l_1\dot{\theta}_1^2 s_2 + l_1\ddot{\theta}_1 c_2 \\ 0 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
{}^2\dot{\mathbf{v}}_{cm} &= ({}^2\dot{\boldsymbol{\omega}}_2 \times {}^2\mathbf{p}_{cm}) + ({}^2\boldsymbol{\omega}_2 \times {}^2\boldsymbol{\omega}_2 \times {}^2\mathbf{p}_{cm}) + {}^2\dot{\mathbf{v}}_2 \\
&= \begin{vmatrix} i & j & k \\ 0 & 0 & (\ddot{\theta}_1 + \ddot{\theta}_2) \\ l_2 & 0 & 0 \end{vmatrix} + \left({}^2\boldsymbol{\omega}_2 \times \begin{vmatrix} i & j & k \\ 0 & 0 & (\dot{\theta}_1 + \dot{\theta}_2) \\ l_2 & 0 & 0 \end{vmatrix} \right) + \\
&\quad \begin{bmatrix} gc_{12} - l_1\dot{\theta}_1^2 c_2 + l_1\ddot{\theta}_1 s_2 \\ -gs_{12} + l_1\dot{\theta}_1^2 s_2 + l_1\ddot{\theta}_1 c_2 \\ 0 \end{bmatrix} = \begin{bmatrix} gc_{12} - l_1\dot{\theta}_1^2 c_2 - l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1\ddot{\theta}_1 s_2 \\ -gs_{12} + l_1\dot{\theta}_1^2 s_2 + l_1\ddot{\theta}_1 c_2 + l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \\
{}^2\mathbf{F}_2 &= m_2 {}^2\dot{\mathbf{v}}_{cm} = m_2 \begin{bmatrix} gc_{12} - l_1\dot{\theta}_1^2 c_2 - l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1\ddot{\theta}_1 s_2 \\ -gs_{12} + l_1\dot{\theta}_1^2 s_2 + l_1\ddot{\theta}_1 c_2 + l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \\
{}^2\mathbf{N}_2 &= I_2 {}^2\dot{\boldsymbol{\omega}}_2 + ({}^2\boldsymbol{\omega}_2 \times I_2 {}^2\boldsymbol{\omega}_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}$$

To create boundary conditions for the inward iteration, the force and torque on the endpoint of link 2 (i.e., at frame 3) must be established. If the manipulator is not in contact with the environment at the endpoint, then this boundary condition is $\mathbf{f}_3 = \boldsymbol{\eta}_3 = \mathbf{0}$. However, we will consider the more general case where $\mathbf{f}_3 = [f_x \ f_y \ 0]^T$ and $\boldsymbol{\eta}_3 = \mathbf{0}$. This boundary condition

represents arbitrary frictionless point contacts in the plane on the endpoint of the 2R manipulator.

Inward iteration

Link 2:

$$\begin{aligned}
{}^2\mathbf{f}_2 &= {}^2\mathbf{F}_2 + {}_2\mathbf{R}_3 {}^3\mathbf{f}_3 \\
&= m_2 \begin{bmatrix} gc_{12} - l_1 \dot{\theta}_1^2 c_2 - l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1 \ddot{\theta}_1 s_2 \\ -gs_{12} + l_1 \dot{\theta}_1^2 s_2 + l_1 \ddot{\theta}_1 c_2 + l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} m_2 l_1 \ddot{\theta}_1 s_2 - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 - m_2 l_1 \dot{\theta}_1^2 c_2 + m_2 gc_{12} + f_x \\ m_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 \ddot{\theta}_1 c_2 + m_2 l_1 \dot{\theta}_1^2 s_2 - m_2 gs_{12} + f_y \\ 0 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
{}^2\boldsymbol{\eta}_2 &= {}^2\mathbf{N}_2 + {}_2\mathbf{R}_3 {}^3\boldsymbol{\eta}_3 + ({}^2\mathbf{p}_{cm} \times {}^2\mathbf{F}_2) + ({}^2\mathbf{p}_3 \times {}_2\mathbf{R}_3 {}^3\mathbf{f}_3) \\
&= \begin{vmatrix} i & j & k \\ l_2 & 0 & 0 \\ ({}^2\mathbf{F}_2)_x & ({}^2\mathbf{F}_2)_y & ({}^2\mathbf{F}_2)_z \end{vmatrix} + \begin{vmatrix} i & j & k \\ l_2 & 0 & 0 \\ f_x & f_y & 0 \end{vmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ l_2 ({}^2\mathbf{F}_2)_y + l_2 f_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau_2 \end{bmatrix}
\end{aligned}$$

where

$$\tau_2 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \ddot{\theta}_1 c_2 + m_2 l_1 l_2 \dot{\theta}_1^2 s_2 - m_2 l_2 gs_{12} + l_2 f_y \quad (\text{B.21})$$

Link 1:

$$\begin{aligned}
{}^1\mathbf{f}_1 &= {}^1\mathbf{F}_1 + {}_1\mathbf{R}_2 {}^2\mathbf{f}_2 \\
&= \begin{bmatrix} m_1 g c_1 - m_1 l_1 \dot{\theta}_1^2 \\ -m_1 g s_1 + m_1 l_1 \ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ({}^2\mathbf{f}_2)_x \\ ({}^2\mathbf{f}_2)_y \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -m_2 l_2 s_2 (\ddot{\theta}_1 + \ddot{\theta}_2) - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 - (m_1 + m_2) l_1 \dot{\theta}_1^2 + (m_1 + m_2) g c_1 + f_x c_2 \\ -f_y s_2 \quad m_2 l_2 c_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 \ddot{\theta}_1 - m_2 l_2 s_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 - m_2 g s_1 + f_x s_2 + f_y c_2 \\ 0 \end{bmatrix} \\
{}^1\boldsymbol{\eta}_1 &= {}^1\mathbf{N}_1 + {}_1\mathbf{R}_2 {}^2\boldsymbol{\eta}_2 + ({}^1\mathbf{p}_{cm} \times {}^1\mathbf{F}_1) + ({}^1\mathbf{p}_2 \times {}_1\mathbf{R}_2 {}^2\mathbf{f}_2) \\
&= \begin{bmatrix} 0 \\ 0 \\ ({}^2\boldsymbol{\eta}_2)_z \end{bmatrix} + \begin{vmatrix} i & j & k \\ l_1 & 0 & 0 \\ ({}^1\mathbf{F}_1)_x & ({}^1\mathbf{F}_1)_y & 0 \end{vmatrix} + \left({}^1\mathbf{p}_2 \times \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ({}^2\mathbf{f}_2)_x \\ ({}^2\mathbf{f}_2)_y \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} 0 \\ 0 \\ ({}^2\boldsymbol{\eta}_2)_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ l_1 ({}^1\mathbf{F}_1)_y \end{bmatrix} + \begin{vmatrix} i & j & k \\ l_1 & 0 & 0 \\ ({}^2\mathbf{f}_2)_x c_2 - ({}^2\mathbf{f}_2)_y s_2 & ({}^2\mathbf{f}_2)_x s_2 + ({}^2\mathbf{f}_2)_y c_2 & 0 \end{vmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ ({}^2\boldsymbol{\eta}_2)_z + l_1 ({}^1\mathbf{F}_1)_y + l_1 (({}^2\mathbf{f}_2)_x s_2 + ({}^2\mathbf{f}_2)_y c_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau_1 \end{bmatrix}
\end{aligned} \tag{B.22}$$

where

$$\begin{aligned}
\tau_1 &= m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) c_2 + (m_1 + m_2) l_1^2 \dot{\theta}_1^2 - m_2 l_1 l_2 \dot{\theta}_2^2 s_2 \\
&\quad - 2m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 s_2 - (m_1 + m_2) l_1 g s_1 - m_2 l_2 g s_{12} + (l_1 c_2 + l_2) f_y + l_1 f_x s_2 \tag{B.23}
\end{aligned}$$

□

B.5 Lagrangian Mechanics

The Lagrangian L is defined to be the difference between the kinetic energy T and the potential energy V of a dynamical system.

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q})$$

This simple expression, together with results from the calculus of variations, yields an elegant means for constructing the dynamic equations of motion.

Theorem B.5.1 (Lagrange's equations). *The equations of motion for a mechanical system with generalized coordinates $q \in \mathbb{R}^m$ and Lagrangian L are given by*

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i \quad i = 1, \dots, m,$$

where Υ_i is the vector of external forces acting on the i th generalized coordinate, q_i .

In vector notation,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \Upsilon.$$

The proof of [equation B.5.1](#) involves the calculus of variations and is beyond the scope of this text. However, we may gain some physical intuition into Lagrange's equation by examining [equation B.5.1](#) more closely. For example, suppose that kinetic energy terms in the Lagrangian look like $\frac{1}{2}m\dot{q}_i^2$ and that potential energy is of the form mgq_i . Substituting these kinds of energy terms into the Lagrangian, we find

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] = \frac{\partial L}{\partial q_i} + \Upsilon_i$$

$$\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_i} \left(\frac{1}{2}m\dot{q}_i^2 - mgq_i \right) \right] = \frac{\partial}{\partial q_i} \left(\frac{1}{2}m\dot{q}_i^2 - mgq_i \right) + \Upsilon_i$$

$$\frac{d}{dt} (m\dot{q}_i) = -mg + \Upsilon_i.$$

We see that the time-rate-of-change in the momentum (left-hand side) is equal to the net external force on the system (right-hand side). This is just Newton's equation as we would expect.

1. If $\dot{\omega}_A \neq 0$, this term will generate the fictitious Euler acceleration.
2. This link is illustrated with a revolute joint, but prismatic joints can be treated the same way.
3. This common assumption is not strictly true. Exercise 3 in chapter 6.4 considers estimating the magnitude of the error stemming from the rotation of the earth.

C

Numerical Methods for Solving Laplace's Equation

For solutions to Laplace's equation

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial q_1^2} + \frac{\partial^2 \phi}{\partial q_2^2} + \cdots + \frac{\partial^2 \phi}{\partial q_n^2} = 0,$$

the value at the center of a symmetric, interior subset of freespace Ω is the average of the values of its 2^n Manhattan neighbors—this is known as the *mean-value* property. We can write this in the form of an equality,

$$u(x_i, y_j) = \frac{1}{4} [u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1})].$$

The mean-value property leads to numerical methods for solving Laplace's equation using finite difference methods [43]. Let $\phi(x, y)$ be a function that satisfies Laplace's equation and let $u(x_i, y_j)$ represent a discrete regular sampling of ϕ on a grid indexed by integers i and j . A finite difference approximation to the second derivatives can then be used to derive the discrete form of Laplace's equation,

$$\nabla^2 \phi(x_i, y_j) \approx u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j) = 0. \quad (\text{C.1})$$

[Equation C.1](#) defines a linear system of equations on the grid and can be extended to higher dimensions in a straightforward fashion. Boundary conditions define the value of the potential on $\partial\Omega$, and we wish to determine the potential at all other grid points such that [equation C.1](#) is satisfied.

Relaxation methods are traditionally used to solve such systems. Techniques such as Jacobi iteration, Gauss-Seidel iteration, and Successive Over-Relaxation (SOR) can be used [43]. For Laplace's equation in n dimensions, these iterative methods repeatedly replace each grid element's value with a weighted average of its $2n$ neighbors. This approach computes discrete potential values on the nodes of a grid. However, multilinear interpolation functions that are smooth and harmonic can yield piecewise continuous potentials between discrete potential values for use in control.

For Laplace's equation, the Jacobi iteration replaces every nonboundary $u(x_i, y_j)$ with the average of its neighbors' values,

$$u^{k+1}(x_i, y_j) = \frac{1}{4} \left[u^k(x_{i+1}, y_j) + u^k(x_{i-1}, y_j) + u^k(x_i, y_{j+1}) + u^k(x_i, y_{j-1}) \right], \quad (\text{C.2})$$

where k is the iteration number. The iteration updates the entire configuration space simultaneously in the sense that old and new data are not combined, and the process is repeated until the residual at each node (equation C.1) drops below a convergence threshold everywhere in the grid.

The Gauss-Seidel iteration is similar to the Jacobi iteration, except that it mixes values from different iterations:

$$u^{(k+1)}(x_i, y_j) = \frac{1}{4} \left[u^{(k)}(x_{i+1}, y_j) + u^{(k+1)}(x_{i-1}, y_j) + u^{(k)}(x_i, y_{j+1}) + u^{(k+1)}(x_i, y_{j-1}) \right]. \quad (\text{C.3})$$

Gauss-Seidel is more space efficient since it iterates in a single solution array. Moreover, it iterates to a fixed precision faster than Jacobi iteration.

Successive Over Relaxation (SOR) adds an acceleration constant (ω) that modifies the value of ϕ at a node in the grid by an amount proportional to the current residual (equation C.1).

$$u^{(k+1)}(x_i, y_j) = u^{(k)}(x_i, y_j) + \frac{\omega}{4} \left[u^{(k)}(x_{i+1}, y_j) + u^{(k+1)}(x_{i-1}, y_j) + u^{(k)}(x_i, y_{j+1}) + u^{(k+1)}(x_i, y_{j-1}) - 4u^{(k)}(x_i, y_j) \right] \quad (\text{C.4})$$

SOR generally requires fewer iterations to converge than does Gauss-Seidel, which requires fewer iterations typically than Jacobi iteration [64, 43]. However, the transient response of the Jacobi iteration approaches the

harmonic function monotonically, which is very appealing when the environment is dynamic since gradient direction is valid even before the relaxation process has converged.

The numerical methods employed to compute harmonic functions lead to especially efficient implementations on SIMD machines,¹ which permit an iteration over the grid to be accomplished in one parallel step using only one solution variable per processing element. On single processor architectures the full multigrid method appears to be the fastest numerical technique for computing harmonic functions [191]. This method has an asymptotic running time of $O(N)$, where N is the number of nodes in the grid. This method compares favorably with other complete path planning techniques (e.g., distance propagation).

Example: A Collision-Free Arm Controller for Roger

To generate collision-free motions in the simple 2R robot in [figure C.1](#), we define a closed domain, $\Omega \in \mathbb{R}^2$, defined on the configuration variables, θ_1 and θ_2 . This representation is formulated in the discrete model in the rightmost configuration space (or c-space) occupancy grid in [figure C.1](#).

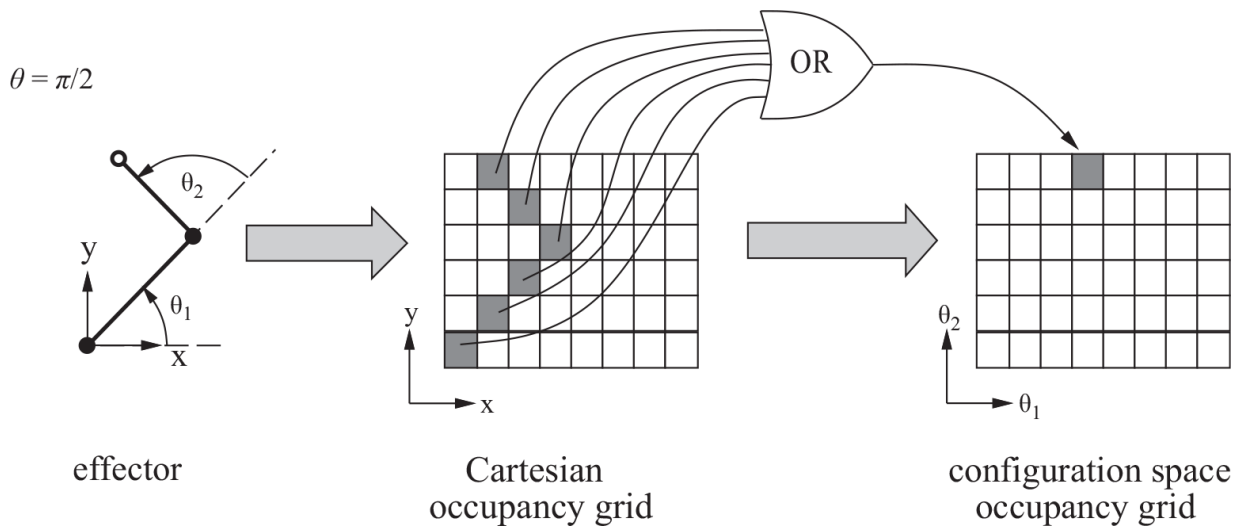


Figure C.1

The mapping of the Cartesian occupancy grid to the configuration space occupancy grid

Every grid element in the configuration space receives a unique label: OBSTACLE, GOAL, or FREESPACE. To perform this labeling, the Cartesian

occupancy grid is queried. A single candidate configuration, (θ_1, θ_2) , maps to a set, R , of Cartesian grid elements that describe the entire volume occupied by the manipulator in this configuration. If any element of R is occupied by an obstacle, the corresponding c-space grid element is labeled OBSTACLE. If R is obstacle-free, the c-space grid element is labeled FREESPACE. Finally, if the terminal element of R (the manipulator endpoint) is a goal, the c-space grid element is labeled GOAL. Note that it is possible, using this mapping, for GOALS to be surrounded by OBSTACLES and therefore inaccessible from FREESPACE.

The boundary of Ω , $\partial\Omega$, is the union of OBSTACLE and GOAL subsets. GOALS are assigned to the lowest potential in the field, $\phi_{goal}=0.0$, and paths are computed by descending the gradient of the potential field. The quality of a path can be influenced by constraints on the potential function at the obstacle boundaries. Neumann boundary conditions require that the gradient flow into obstacle boundaries is zero. Gradient flow subject to Neumann boundary conditions is tangential to the local obstacle boundary. Dirichlet boundaries fix the potential at OBSTACLE boundaries at the highest potential in the system $\phi_{obs}=1.0$. As a consequence, obstacles act as sources of flow and gradients near obstacles depart along the normal to local Dirichlet boundaries. In this example, all boundaries are modeled as Dirichlet boundaries.

Discrete configuration space maps can model arbitrary Cartesian shapes up to the spatial resolution of the discrete occupancy grid. In this example, θ_1 is toroidal, allowing it to rotate continuously, and θ_2 is subject to joint angle limits—namely, $-3.1 \text{ rad} < \theta_2 < +3.1 \text{ rad}$. Values outside of this range for θ_2 are labeled OBSTACLES in the c-space grid to eliminate self-collisions between the links.

The potential of FREESPACE grid elements is computed by the numerical relaxation techniques described earlier in this section to satisfy Laplace's equation—this example uses SOR. Gradients in the converged harmonic function create streamlines in c-space that originate at OBSTACLES (sources) and terminate at GOALS (sinks). [Figure C.2](#) shows some examples of collision-free motor solutions. A Cartesian goal position (green) is placed in the small room defined by obstacles (red) with a small doorway. The Cartesian goal maps to two GOAL regions in c-space, one for each of the two

arm configurations possible. One of these GOAL regions is embedded inside an OBSTACLES region and is, therefore, inaccessible. The streamline that the robot follows depends on the robot's initial configuration. The time history of two representative trajectories is shown (blue) in both Cartesian and configuration space. In the lower pair of diagrams, an additional rectangular obstacle is added, and its effect on the robot's trajectory is illustrated.

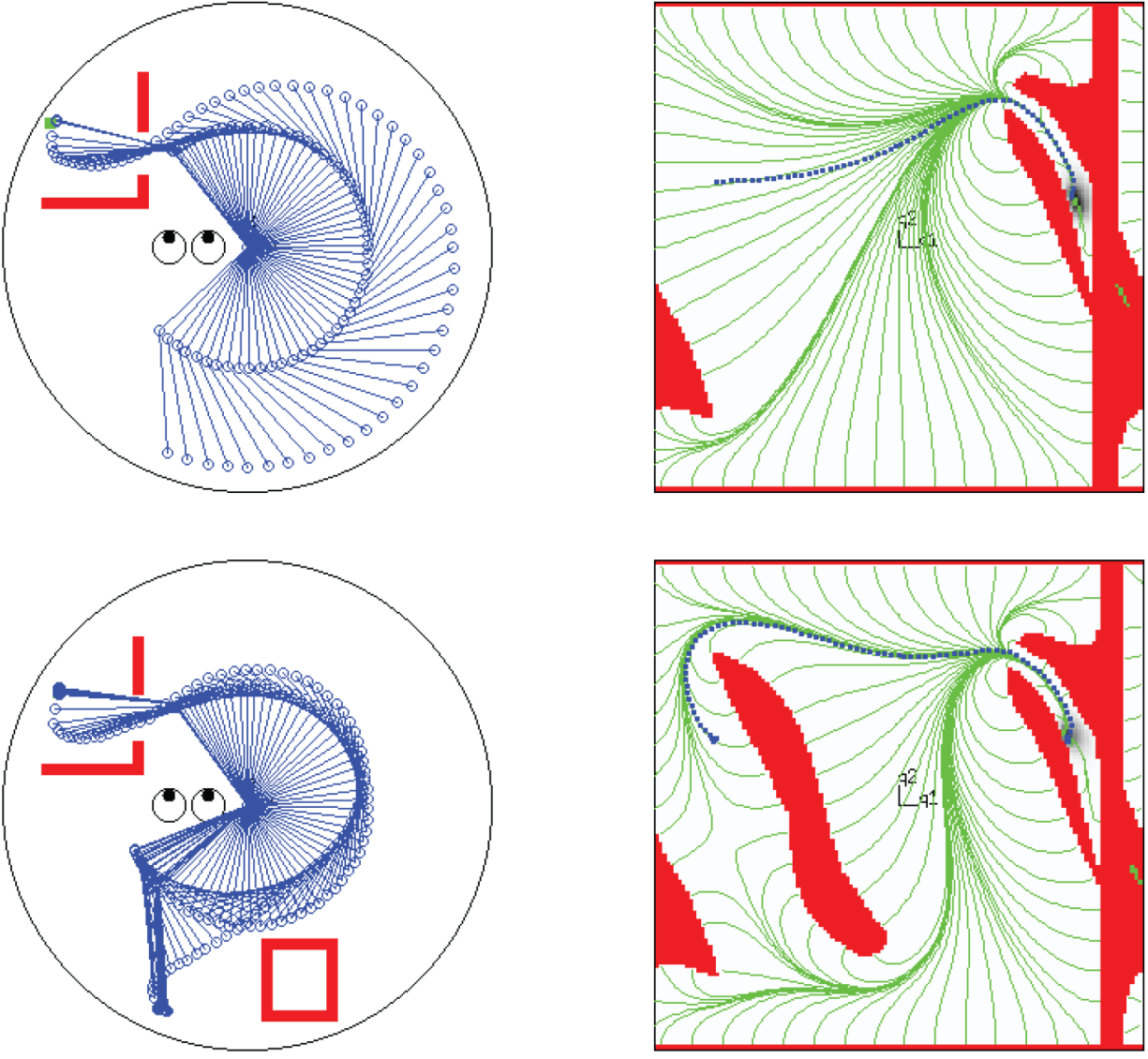


Figure C.2
 Gradient descent on a harmonic function yields a collision-free motor plan for Roger's right arm.
 (Left) The Cartesian space. (Right) The c-space.

□

The example introduces harmonic functions as a means of defining a sequence of postural goals to the underlying motor system. We assume that dynamic compensation and appropriate dissipative elements exist in the lower-level motor units and need not be considered here. Harmonic functions are *resolution complete* in the sense that if a path exists at the resolution of the occupancy grid, it will be found. To check whether a path exists, it is sufficient to verify the presence of a nonzero gradient in the converged harmonic function. This is accomplished in the same amount of time required to compute the next step in a path. This property is of some practical significance since many search techniques can find paths quickly on average but may take an indeterminate amount of time in some environments to conclude that no path exists. In the framework outlined, obstacles can be arbitrary in shape and can be used to represent other geometrical constraints. For example, if Roger wants to protect an unoccluded line of sight to a desired object while reaching to it, then he may map the ray from his eye to the target as an obstacle. This mechanism could be used to implement the visual cliff constraint (section 9.5.1) in a straightforward manner. One of the challenging features of harmonic function motion planners is the numerical precision required. The path is extracted from the harmonic potential function by computing the local gradient, but this gradient can be very shallow. In practice, numerical precision limitations may make it hard to differentiate shallow gradients from cases where the gradient vanishes and no path to the goal exists.

Voltage distributions in conductive media can be described as a diffusion process subject to Laplace's equation, which has caused some to speculate about the possibility of extremely fast path planners. Resistive networks were employed in the 1940s as analog computers to solve Laplace's equation [189]. Tarassenko and Blake [271] describe a VLSI version of this hardware scheme in which a programmable resistive grid consisting of an array of transistors whose drain-source resistances are governed by gate voltages. The transistor network can be used to set both Dirichlet and Neumann boundary conditions. The steady-state voltages at FREESPACE nodes represent a discrete sampling of a harmonic function. Given that the computation is performed by voltages settling in a resistive network, this appears to provide the fastest theoretical framework for robot path planning

—although this approach has not yet been successfully demonstrated in realistic applications.

Harmonic functions introduce some fundamental limitations, however. As the grid size is increased, the settling time will also increase, and voltage differences within the grid will decrease. Thus, there are physical limits to the size (and hence resolution) of a useful resistive grid. Nonetheless, these limits appear to be in the range of hundreds or perhaps even thousands of resistive elements on a side, using currently available VLSI technology.

In nature, fields with these properties describe minimal energy phenomena such as laminar fluid flow, the shape of soap films, and the steady-state temperature and voltage in conductive media. In robotics, navigation functions guarantee that a particle subject to the local field gradient will eventually reach a globally minimum potential configuration. The harmonic function is subject to Laplace's equation as the governing constraint. Like the soap film, Laplace's constraint requires that the scalar field minimizes the global strain energy of the scalar field with known boundary conditions. As a result, there are no local minima, and gradient descent on the field formally minimizes the *hitting probability* of the system. We developed an array of techniques for computing harmonic functions and using them in collision-avoiding path planning and continuous, quasi-static path control tasks.

1. SIMD is an acronym for Single Instruction, Multiple Data parallel architectures.

Bibliography

- [1] Aeyels, D. Generic observability of differentiable systems. *SIAM Journal of Control and Optimization*, 19(5):595–603, 1981.
- [2] Akishita, S., S. Kawamura, and K. Hayashi. Laplace potential for moving obstacle avoidance and approach of a mobile robot. In *1990 Japan-USA Symposium on Flexible Automation, A Pacific Rim Conference*, pages 139–142, 1990.
- [3] Albus, J. S. A control system architecture for intelligent machine systems. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Arlington, VA, October 1987. IEEE. doi: 10.1.1.14.1370.
- [4] Allen, P. K. Sensing and describing 3-d structure. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 1, pages 126–131, San Francisco, April 1986. IEEE.
- [5] Anderson, J. R. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, 1983.
- [6] Arbib, M. Schema theory. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, 2nd ed., pages 1427–1443. Wiley-Interscience, New York, 1992.
- [7] Arkin, R. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.
- [8] Asada, M., K. McDorman, H. Ishiguro, and Y. Kuniyoshi. Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*, 37:185–193, 2001.
- [9] Asimov, I. *Robot Visions*. Penguin, New York, 1991.
- [10] Baillargeon, R. Infant’s physical world. *Current Directions in Psychological Science*, 13(3):89–94, 2004.
- [11] Baldassare, G., and M. Mirolli. Computational and robotic models of the hierarchical organization of behavior: An overview. In G. Baldassare and M. Mirolli, editors, *Computational and Robotic Models of the Hierarchical Organization of Behavior*, pages 1–10. Springer-Verlag Berlin Heidelberg, 2013.
- [12] Ballard, D., and C. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [13] Bardelli, R., P. Dario, D. De Rossi, and P. C. Pinotti. Piezo- and pyroelectric polymers skin-like tactile sensors for robots and prostheses. In *Proceedings of Robotics International of SME*, pages 18-45–18-56, 1983.

- [14] Barr, M. *The Human Nervous System*. Harper and Row, New York, 1974.
- [15] Barraquand, J., and J.-L. Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649, December 1991.
- [16] Barry Wright Corporation Products for Flexible Automation Bulletin. Sensoflex tactile sensing system data sheet. Technical Report, Part No. TS 402-1, Watertown, MA, 1984.
- [17] Baughman, R., C. Cui, A. Zakhidov, Z. Iqbal, J. Barisci, G. Spinks, G. Wallace, A. Mazzoldi, D. De Rossi, A. Rinzler, O. Jaschinski, S. Roth, and M. Kertesz. Carbon nanotube actuators. *Science*, 284:1340–1344, May 1999.
- [18] Beccai, L., S. Roccella, A. Arena, F. Valvo, P. Valdastri, A. Menciassi, M. Carrozza, and P. Dario. Design and fabrication of a hybrid silicon three-axial force sensor for biomechanical applications. *Sensors and Actuators A*, 120(2):370–382, 2005.
- [19] Begej, S. An optical tactile array sensor. *SPIE Intelligent Robots and Computer Vision*, 521:271–280, 1984.
- [20] Begej Corporation. Product literature for the FTS-2 fingertip shaped tactile sensor. Technical Report, Technical Bulletin No. 2, Littleton, CO, 1986.
- [21] Bell, C. *The Hand: Its Mechanism and Vital Endowments, As Evincing Design*. Harper & Brothers, New York, 1840.
- [22] Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [23] Bernstein, N. On dexterity and its development. In M. Latash and M. Turvey, editors, *Dexterity and Its Development*, pages 1–244. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.
- [24] Berthier, N. E., R. E. Clifton, V. Gullapalli, D. McCall, and D. J. Robin. Visual information and the control of reaching. *Journal of Motor Behavior*, 28:187–197, 1996.
- [25] Berthier, N. E., R. E. Clifton, D. D. McCall, and D. J. Robin. Proximodistal structure of early reaching in human infants. *Experimental Brain Research*, 127:259–269, 1999.
- [26] Berthier, Neil. Learning to reach: A mathematical model. *Developmental Psychology*, 32:811–823, 1996.
- [27] Bertsekas, D. *Nonlinear Programming*. Athena Scientific, Cambridge, MA, 1999.
- [28] Bhatnagar, S. *Neuroscience for the Study of Communication Disorders*. 2nd ed. Lippincott, Williams, and Wilkins, Baltimore, MD, 2002.
- [29] Bicchi, A., K. Salisbury, and D. Brock. Contact sensing from force measurements. *International Journal of Robotics Research*, 12(3):249–262, 1993.
- [30] Bizzi, E., F. Mussa-Ivaldi, and S. Giszter. Mechanical properties of muscles: Implications for motor control. *Science*, 253:287–291, July 1991.
- [31] Bjorklund, D., V. Periss, and K. Causey. The benefits of youth. *European Journal of Developmental Psychology*, 1(6):120–137, 2009.
- [32] Blank, D., D. Kumar, and L. Meeden. Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. In *Proceedings of the Workshop Growing Up Artifacts That Live, Simulated Adaptive Behavior 2002, From Animals to Animats*, 2002.
- [33] Blank, D., D. Kumar, L. Meeden, and J. Marshall. Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems*, 2(36), 2005.

- [34] Bohon, K., and S. Krause. An electrorheological fluid and siloxane based electromechanical actuator: Working toward an artificial muscle. *Journal of Polymer Science Part B Polymer Physics*, 36:1091–1094, 1998.
- [35] Boie, R. A. Capacitive impedance readout tactile image sensor. In *Proceedings of the 1984 Conference on Robotics*, pages 370–378, Atlanta, March 1984. IEEE.
- [36] Boissonnant, J. D. Stable matching between a hand structure and an object silhouette. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6):603–612, November 1982.
- [37] Botea, A., M. Enzenberger, M. Müller, and J. Schaeffer. Macro-ff: Improving AI planning with automatically learned macro-operators. *Journal of Artificial Intelligence Research*, 24(1):581–621, October 2005.
- [38] Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA, 1984.
- [39] Bril, B., and Y. Brenière. Postural requirements and progression velocity in young walkers. *Journal of Motor Behavior*, 24(1):105–116, 1992.
- [40] Brock, D. L. Review of artificial muscle based on contractile polymers. Technical Report AI Memo No. 1330, MIT, Cambridge, MA, 1991.
- [41] Brooks, R., C. Breazeal, M. Marjanovic, B. Scassellati, and M. Williamson. The cog project: Building a humanoid robot. In C. Nehaniv, editor, *Computation for Metaphors, Analogy, and Agents*, Lecture Notes in Computer Science 1562, pages 52–87. Springer-Verlag, Heidelberg Berlin, 1999.
- [42] Bruner, J. S. Organization of early skilled action. *Child Development*, 44:1–11, 1973.
- [43] Burden, R., J. Faires, and A. Reynolds. *Numerical Analysis*. Prindle, Weber and Schmidt, Boston, 1978.
- [44] Burrige, R., A. Rizzi, and D. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–555, 1999.
- [45] Carlson, F., and D. R. Wilkie. *Muscle Physiology*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [46] Carlson, N. *Physiology of Behavior*. 2nd ed. Allyn and Bacon, Boston, 1981.
- [47] Chiacchio, P. A new dynamic manipulability ellipsoid for redundant manipulators. *Robotica*, 18(4):381–387, 2000.
- [48] Chiarelli, P., D. De Rossi, and K. Umezawa. Progress in the design of an artificial urethral sphincter. In *Proceedings of the 3rd Vienna International Workshop on Functional Electro stimulation*, Vienna, Austria, September 1989.
- [49] Chiu, S. L. Control of redundant manipulators for task compatibility. In *Proceedings of the 1987 Conference on Robotics and Automation*, volume 3, pages 1718–1724, Raleigh, NC, April 1987. IEEE.
- [50] Chiu, S. L. Task compatibility of manipulator postures. *Journal of Robotics Research*, 7(5), October 1988.
- [51] Chrupa, L., M. Vallati, T. McCluskey, and D. Kitchin. Generating macro-operators by exploiting inner entanglements. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA)*, Palo Alto, CA, 2013. AAAI Press.
- [52] Chrupa, L., M. Vallati, T. McCluskey, and D. Kitchin. Mum: A technique for maximising the utility of macro-operators by constrained generation and use. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 65–73, Portsmouth, RI, 2014. AAAI Press.

- [53] Churchland, P. M. *Matter and Consciousness: A Contemporary Introduction to the Philosophy of Mind*. Bradford/MIT Press, Cambridge, MA, 1988.
- [54] Clifton, R., P. Rochat, D. Robin, and N. Berthier. Multimodal perception in the control of infant reaching. *Journal of Experimental Psychology: Human Perception and Performance*, 20:876–886, 1997.
- [55] Coelho, J. *Multifingered Grasping: Haptic Reflexes and Control Context*. PhD thesis, University of Massachusetts Amherst, September 2001.
- [56] Coelho, J., and R. Grupen. Online grasp synthesis. In *Proceedings of the Conference on Robotics and Automation*, pages 2137–2142, Minneapolis, MN, April 1996. IEEE.
- [57] Coelho, J., and R. Grupen. Learning in non-stationary conditions: A control theoretic approach. In *Proceedings of the Seventeenth International Conference on Machine Learning*, Stanford University, July 2000. IEEE.
- [58] Coelho, J., J. Piater, and R. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. *International Journal on Robotics and Autonomous Systems*, Volume 37, no. 2, pages 195–218, 2001.
- [59] Coelho, J. A., and R. A. Grupen. A control basis for learning multifingered grasps. *Journal of Robotic Systems*, 14(7):545–557, 1997.
- [60] Coiffet, P. *Robot Technology*, volume 2. Prentice Hall, New York, 1981.
- [61] Connolly, C. Harmonic functions and collision probabilities. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 3015–3019, San Diego, CA, April 1994.
- [62] Connolly, C., J. Burns, and R. Weiss. Path planning using Laplace’s Equation. In *International Conference on Robotics and Automation*, pages 2102–2106, Cincinnati, OH, May 1990. IEEE.
- [63] Connolly, C., and R. Grupen. Harmonic control. In *Proceedings of the 1992 International Symposium on Intelligent Control*, pages 498–502, Glasgow, Scotland, August 1992. IEEE.
- [64] Connolly, C., and R. Grupen. On the applications of harmonic functions to robotics. *Journal of Robotics Systems*, 10(7):931–946, 1993.
- [65] Craig, J. *Introduction to Robotics: Mechanics and Control*. 2nd ed. Addison Wesley, Reading, MA, 1986.
- [66] Cutkosky, M., R. Howe, and W. Provancher. Force and tactile sensors. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 455–476. Springer, Cham, Switzerland, 2008.
- [67] Cutkosky, M. R. On grasp choice, models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, June 1989.
- [68] Cutkosky, M. R., and P. K. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. In *Proceedings of 1986 Conference on Robotics and Automation*, volume 3, pages 1533–1539, San Francisco, 1986. IEEE.
- [69] Dario, P., D. De Rossi, C. Domenici, and R. Francesconi. Ferroelectric polymer tactile sensors with anthropomorphic features. In *Proceedings of the 1984 Conference on Robotics*, pages 332–340, Atlanta, Georgia, March 1984. IEEE.
- [70] Dennett, D. Styles of mental representation. In *Proceedings of the Aristotelian Society*, volume 83, pages 213–226, London, UK, 1982.
- [71] Dominguez, M., and R. Jacobs. Developmental constraints aid the acquisition of binocular disparity sensitivities. *Neural Computation*, 15(1):161–182, 2003.

- [72] Drescher, G. *Made-up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, Cambridge, MA, 1991.
- [73] Edin, B., L. Beccai, L. Ascari, S. Roccella, J. Cabibihan, and M. Carrozza. A bio-inspired approach for the design and characterization of a tactile sensory system for a cybernetic prosthetic hand. In *Proceedings of the International Conference on Robotics and Automation*, pages 1354–1358 Orlando, FL. 2006. IEEE.
- [74] Edwards, S., D. Buckland, and J. McCoy-Powlen. *Developmental & Functional Hand Grasps*. SLACK Incorporated, Thorofare, NJ, 2002.
- [75] Elliot, T., and N. Shadbolt. Developmental robotics: Manifesto and applications. *Philosophical Transaction: Mathematical, Physical, and Engineering Sciences*, 361:2187–2206, 2003.
- [76] Ellis, R. E. Acquiring tactile data for the recognition of planar objects. In *Proceedings of the 1987 Conference on Robotics and Automation*, volume 2, pages 1799–1805, Raleigh, NC, April 1987. IEEE.
- [77] Ellis, R. E. Extraction of tactile features by passive and active sensing. *SPIE Intelligent Robots and Computer Vision*, 521:289–295, 1984.
- [78] Elman, J. Learning and development in neural networks: The importance of starting small. *Cognition*, 48:71–99, 1993.
- [79] Ernst, H. A. *MH-1, A computer-operated mechanical hand*. PhD thesis, MIT, Cambridge, MA, December 1961.
- [80] Fearing, R. S. Implementing a force strategy for object re-orientation. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 1, pages 96–102, San Francisco, April 1986. IEEE.
- [81] Fearing, R. S. Simplified grasping and manipulation with dextrous robot hands. *IEEE Journal of Robotics Research*, 2(4):188–195, January 1983.
- [82] Fearing, R. S. *Touch processing for determining a stable grasp*. Master’s thesis, MIT, Department of Electrical Engineering and Computer Science, Boston, September 1983.
- [83] Fearing, R. S., and J. M. Hollerbach. Basic solid mechanics for tactile sensing. In *Proceedings of the 1984 Conference on Robotics*, pages 266–275, Atlanta, March 1984. IEEE.
- [84] Feynman, R. *Lectures on Physics*, volume 1. Addison-Wesley, Reading, MA, 1963.
- [85] Fikes, R. E., and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4):189–208, 1971.
- [86] Fikes, R. E., and N. J. Nilsson. Strips, a retrospective. *Artificial Intelligence*, 59(1–2):227–232, 1993.
- [87] Fiorentino, Mary R. *A Basis for Sensorimotor Development—Normal and Abnormal*. Charles C. Thomas, Springfield, IL, 1981.
- [88] Frei, W., and C. Chen. Fast boundary detection: A generalization and a new algorithm. *IEEE Computer*, 26:988–999, 1977.
- [89] Gelb, A., editor. *Applied Optimal Estimation*. The Analytical Sciences Corporation, MIT Press, Cambridge, MA, 1986.
- [90] Gergely, G. What should a robot learn from an infant? Mechanisms of action interpretation and observational learning in infancy. In Christopher G. Prince, Luc Berthouze, Hideki Kozima, Daniel Bullock, Georgi Stojanov, and Christian Balkenius, editors, *Proceedings of the Workshop on Epigenetic Robotics*, pages 13–24, Boston, August 2003. Lund University Cognitive Studies.

- [91] Gibbs, G. J., and H. L. Colston. The cognitive psychological reality of image schemas and their transforms. *Cognitive Linguistics*, 6(4):347–378, 1995.
- [92] Gibson, E. J., and E. S. Spelke. *The Development of Perception*. 4th ed. Wiley, 1983.
- [93] Gibson, E. J., and R. D. Walk. The visual cliff. *Scientific American*, 202:64–71, 1960.
- [94] Giszter, S., F. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog’s spinal cord. *Journal of Neuroscience*, 13:467–491, 1993.
- [95] Goldman-Rakic, P. Organization development and plasticity of primate prefrontal cortex. *Neuroscience Abstracts*, 10(10), 1982.
- [96] Golub, G. H., and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1983.
- [97] Gordon, G. *Active Touch*. Pergamon Press, Elmsford, NY, 1978.
- [98] Grupen, R., and K. Souccar. Manipulability-based spatial isotropy: A kinematic reflex. In *Workshop on Mechatronical Computer Systems for Perception and Action*, Halmstad, Sweden, June 1–3, 1993.
- [99] Grupen, R. A., K. Biggers, T. C. Henderson, and S. Meek. Task defined internal grasp wrenches. Technical Report UUCS-88-001, Department of Computer Science, University of Utah, 1988.
- [100] Halder, G., P. Callaerts, and W. Gehring. New perspectives on eye evolution. *Current Opinions on Genetic Development*, 5:602–609, 1995.
- [101] Hanafusa, H., and H. Asada. A robot hand with elastic fingers and its application to assembly process. In *IFAC Symposium on Information and Control Problems in Manufacturing Technology*, pages 127–138, Tokyo, 1977.
- [102] Hanafusa, H., and H. Asada. Stable prehension by a robot hand with elastic fingers. In *Proceedings of the 7th International Symposium on Industrial Robots*, pages 361–368, Tokyo, October 1977. SME.
- [103] Harmon, L. D. Automated tactile sensing. Technical Report MSR82-02, Society of Manufacturing Engineers, Dearborn, MI, 1982.
- [104] Harmon, L. D. Automated touch sensing: A brief perspective and several new approaches. In *Proceedings of the 1984 Conference on Robotics*, pages 326–331, Atlanta, 1984. IEEE.
- [105] Harmon, L. D. Robotic taction for industrial assembly. *Communications of the Journal of Robotics Research*, 3(1):72–76, Spring 1984.
- [106] Harmon, L. D. Touch-sensing technology: A review. Technical Report MSR80-03, Society of Manufacturing Engineers, Dearborn, MI, 1980.
- [107] Hart, S., and R. Grupen. Natural task decomposition with intrinsic potential fields. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2507–2512, San Diego, CA, 2007.
- [108] Hart, S., S. Sen, and R. Grupen. Generalization and transfer in robot control. In *International Conference on Epigenetic Robotics*, Brighton, UK, 2008.
- [109] Hart, S., S. Sen, and R. Grupen. Intrinsically motivated hierarchical manipulation. In *International Conference on Robotics and Automation (ICRA)*, pages 3814–3819, Pasadena, CA, 2008. IEEE.
- [110] Hawkins, J., and S. Blakeslee. *On Intelligence*. Holt, New York, 2004.

- [111] Hebert, M., J. M. Wong, and R. Grupen. Phase lag bounded velocity planning for high performance path tracking. In *International Conference on Humanoid Robots (Humanoids)*, pages 947–952, IEEE-RAS, 2015.
- [112] Heimer, L. *The Human Brain and Spinal Cord*. Springer-Verlag, New York, 1983.
- [113] Hendriks-Jensen, H. *Catching Ourselves in the Act*. MIT Press, Cambridge, MA, 1996.
- [114] Hillis, W. D. A high resolution image touch sensor. *Journal of Robotics Research*, 1(2):33–44, Summer 1982.
- [115] Hogan, N. Impedance control: An approach to manipulation. I. Theory, II. Implementation, III. Applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–24, March 1985.
- [116] Holmes, M., and C. Isbell. Schema learning: Experience-based construction of predictive action models. In *Advances in Neural Information Processing Systems (NIPS) 17*, MIT Press, 2004.
- [117] Huber, M. *A hybrid architecture for adaptive robot control*. PhD thesis, University of Massachusetts Amherst, May 2000.
- [118] Huber, M., and R. Grupen. A feedback control structure for on-line learning tasks. *Journal of Robots and Autonomous Systems*, 22(3–4):303–315, 1997.
- [119] Huber, M., and R. Grupen. A hybrid architecture for learning robot control tasks. In *Spring Symposium Series: Hybrid Systems and AI: Modeling, Analysis and Control of Discrete + Continuous Systems*, 96–100, Stanford, CA, 1999. AAAI.
- [120] Huber, M., and R. Grupen. Learning to coordinate controllers—reinforcement learning on a control basis. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1366–1371, Nagoya, Japan, San Francisco, August 1997. Morgan Kaufmann.
- [121] Huber, M., and R. A. Grupen. A hybrid discrete event dynamic systems approach to robot control. Technical Report 96-43, CS Department, University of Massachusetts Amherst, October 1996.
- [122] Huber, M., W. MacDonald, and R. Grupen. A control basis for multilegged walking. In *Proceedings of the Conference on Robotics and Automation*, volume 4, pages 2988–2993, Minneapolis, MN, April 1996. IEEE.
- [123] Hutchinson, S., and A. Kak. Spar: A planner that satisfies operational and geometric goals in uncertain environments. *AI Magazine*, 11(1):30–61, 1990.
- [124] Huxley, A. F. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7:255–318, 1957.
- [125] Huxley, A. F. Muscular contraction. *Journal of Physiology*, 243:1–43, 1974.
- [126] Iberall, T. Grasp planning for human prehension. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1153–1156, Milan, Italy, 1987, IJCAI.
- [127] Ijspeert, A., J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1547–1554. MIT Press, Cambridge, MA, 2003.
- [128] Jacob, H., S. Feder, and J. Slotine. Real-time path planning using harmonic potential functions in dynamic environments. In *International Conference on Robotics and Automation*, pages 874–881, Albuquerque, NM, 1997. IEEE.

- [129] Jacobsen, S. C., E. K. Iversen, D. F. Knutti, R. T. Johnson, and K. B. Biggers. Design of the Utah/MIT dextrous hand. In *Proceedings of the 1986 Conference on Robotics and Automation*, pages 1520–1532, San Francisco, CA, April 1986. IEEE.
- [130] Jacobsen, S. C., I. D. McCammon, K. B. Biggers, and R. P. Phillips. Tactile sensing system design issues in machine manipulation. In *Proceedings of the 1987 Conference on Robotics and Automation*, pages 2087–2096, Raleigh, NC, April 1987, IEEE.
- [131] Jacobsen, S. C., C. C. Smith, K. B. Biggers, and E. K. Iversen. Behavior based design of robot effectors. In *Proceedings of the 1987 International Symposium on Robotics Research*, pages 41–55, MIT Press, Cambridge, MA, 1988.
- [132] Jacobsen, S. C., J. E. Wood, D. F. Knutti, and K. B. Biggers. The Utah/MIT dextrous hand. In *International Robotics Research Symposium*, pages 601–654, Bretton Woods, NH, August 1983.
- [133] Jacobsen, S. C., J. E. Wood, D. F. Knutti, K. B. Biggers, and E. K. Iversen. The version I Utah/MIT dextrous hand. In *Proceedings of the Second International Symposium on Robotics Research*. MIT Press, Cambridge, MA, 1984.
- [134] Kaelbling, L., and T. Lozano-Perez. Hierarchical task and motion planning in the now. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1470–1477, Shanghai, China, 2011.
- [135] Kakugo, A., S. Sugimoto, J. P. Gong, and Y. Osada. Gel machines constructed from chemically cross-linked actins and myosins. *Advanced Materials*, 14:1124–1126, 2002.
- [136] Kandel, E. R., and J. H. Schwartz. *Principles of Neural Science*. Elsevier, New York, 1981.
- [137] Kao, I., and M. R. Cutkosky. Quasistatic manipulation with compliance and sliding. *International Journal of Robotics Research*, 11(1):20–40, 1992.
- [138] Kao, I., K. Lynch, and J. Burdick. Contact modeling and manipulation. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Switzerland, Springer-Verlag, 2008.
- [139] Kapandji, I. *The Physiology of the Joints*, volume 1. Churchill Livingstone, New York, 1970.
- [140] Kelly, A. *Mobile Robotics: Mathematics, Models, and Methods*. Cambridge University Press, New York, 2013.
- [141] Kelso, S. *Dynamic Patterns*. MIT Press, Cambridge, MA, 1995.
- [142] Kerr, J., and B. Roth. Analysis of multifingered hands. *Journal of Robotics Research*, 4(4):3–17, Winter 1986.
- [143] Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In *International Conference on Robotics and Automation*, pages 500–505, St. Louis, MO, March 1985. IEEE.
- [144] Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1), 1986.
- [145] Kim, J., and P. Khosla. Real-time obstacle avoidance using harmonic potential functions. In *International Conference on Robotics and Automation*, pages 790–796, April 1991. IEEE.
- [146] King, A., M. Hutchings, D. Moore, and C. Blakemore. Developmental plasticity in the visual and auditory representations in the mammalian superior colliculus. *Nature*, 332:73–76, 1988.
- [147] Klein, C., and B. Blaho. Dexterity measures for the design and control of kinematically redundant manipulators. *Journal of Robotics Research*, 6(2):72–83, Summer 1987.
- [148] Kobayashi, H. Control and geometrical considerations for an articulated robot hand. *Journal of Robotics Research*, 4(1):3–12, Spring 1985.

- [149] Kobayashi, H. Grasping and manipulation of objects by articulated hands. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 3, pages 1514–1519, San Francisco, April 1986. IEEE.
- [150] Koditschek, D. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings of the International Conference on Robotics and Automation*, volume 97, pages 211–223, 1987. IEEE.
- [151] Koenderink, J. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [152] Korf, R. Macro-operators: A weak method for learning. *Artificial Intelligence*, 26(1):35–77, April 1985.
- [153] Kořecká, J., and L. Bogoni. Application of discrete event systems for modeling and controlling robotic agents. In *Proceedings of the International Conference on Robotics and Automation*, pages 2557–2562, San Diego, CA, May 1994. IEEE.
- [154] Krogh, B. A generalized potential field approach to obstacle avoidance control. In *Proceedings of the SME Conference on Robotics Research: The Next Five Years and Beyond*, pages 11–22, Bethlehem, PA, 1984.
- [155] Ku, L., E. Learned-Miller, and R. Grupen. An aspect representation for object manipulation based on convolutional neural networks. In *Proceedings of the International Conference on Robotics and Automation*, pages 794–800, May 2017. IEEE.
- [156] Ku, L., E. Learned-Miller, and R. Grupen. Modeling objects as aspect transition graphs to support manipulation. In *Proceedings of the International Symposium on Robotics Research*, Sestri Levante, Italy, August 2015.
- [157] Ku, L., S. Sen, E. Learned-Miller, and R. Grupen. Action-based models for belief-space planning. In *Robotics: Science and Systems, Workshop on Information-Based Grasp and Manipulation Planning*, May 2014.
- [158] Ku, L., S. Sen, E. Learned-Miller, and R. Grupen. Aspect transition graph: An affordance-based model. In *European Conference on Computer Vision, Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis*, pages 459–465, Zurich, Switzerland, 2014. Springer.
- [159] Kuindersma, S., R. Grupen, and A. Barto. Variable risk dynamic mobile manipulation. In *RSS 2012 Mobile Manipulation Workshop*, Sydney, Australia, July 2012.
- [160] Kuipers, B., P. Beeson, J. Modayil, and J. Provost. Bootstrap learning of foundational representations. *Connection Science*, 18(2):145–158, 2006.
- [161] Laird, J., P. Rosenbloom, and A. Newell. Chunking in SOAR: The anatomy of a general learning mechanism. *Journal of Machine Learning*, 1:11–46, 1986.
- [162] Lakoff, G., and M. Johnson. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Basic Books, New York, 1999.
- [163] Langley, L. L., I. R. Telford, and J. B. Christensen. *Dynamic Anatomy and Physiology*. McGraw-Hill, New York, 1974.
- [164] Lanighan, M., T. Takahashi, and R. Grupen. Planning robust manual tasks in hierarchical belief spaces manipulation skills. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 459–467, Delft, Netherlands, June 2018.
- [165] Larson, P., and S. Stensaas. Pedineurologic exam: A neurodevelopmental approach. https://neurologicexam.med.utah.edu/pediatric/html/home_exam.html, 2003.
- [166] Latombe, T. *Robot Motion Planning*. Kluwer, Boston, 1991.

- [167] LaValle, S. *Planning Algorithms*. Cambridge University Press, New York, 2006.
- [168] Law, J., M. Lee, M. Hülse, and A. Tomassett. The infant development timeline and its application to robot shaping. *Adaptive Behavior*, 19(5):335–358, 2011.
- [169] Lee, M. Tactile sensing: New directions, new challenges. *Journal of Robotics Research*, 19(7):636–643, 2000.
- [170] Lee, M., and H. Nicholls. Tactile sensing for mechatronics—a state of the art survey. *Mechatronics*, 9(1):1–31, 1999.
- [171] Li, Z., and S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4(1):32–44, February 1988.
- [172] Li, Z., and S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Transactions Systems, Man, and Cybernetics*, 11(10):681–689, 1988.
- [173] Lian, D., S. Peterson, and M. Donath. A three-fingered, articulated hand. In *Proceedings of the 13th International Symposium on Industrial Robots*, volume 2, pages 18-91–18-101, Chicago, April 1983. SME.
- [174] Liégeois, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, December 1977.
- [175] Lindeberg, T. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings of the CERN School of Computing*, pages 1–12, Netherlands, September 1996. Egmond aan Zee.
- [176] Lindeberg, T. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [177] Lindeberg, T., and B. ter Haar Romeny. Linear scale-space. In B. ter Haar Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, pages 1–77. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [178] Lord Corporation. Product literature for the Lord tactile sensor LTS series. Technical Report, Lord Corporate Development Center, Cary, NC, 1984.
- [179] Lungarella, M., G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: A survey. *Connection Science*, 15(4):151–190, 2003.
- [180] Luo, R. C., F. Wang, and Y. X. Liu. An imaging tactile sensor with magnetostrictive transduction. *SPIE Intelligent Robots and Computer Vision*, 521:264–270, 1984.
- [181] Lyons, D. Tagged potential fields: An approach to specification of complex manipulator configurations. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 3, pages 1749–1754, San Francisco, April 1986. IEEE.
- [182] MacDonald, W. S., and R. A. Grupen. Building walking gaits for irregular terrain from basis controllers. In *Proceedings of the Conference on Robotics and Automation*, Albuquerque, NM, April 1997. IEEE.
- [183] Mahowald, M., and C. Mead. The silicon retina. *Scientific American*, 264(5):76–82, 1991.
- [184] Malcolm, C., and T. Smithers. Symbol grounding via a hybrid architecture in an autonomous assembly system. In P. Maes, editor, *Designing Autonomous Agents*, pages 123–144. MIT Press, Cambridge, MA, 1990.
- [185] Mandler, Jean M. How to build a baby: On the development of an accessible representational system. *Cognitive Development*, 3:113–136, 1988.

- [186] Mandler, Jean M. How to build a baby II: Conceptual primitives. *Psychological Review*, 99(4):587–604, 1992.
- [187] Marzke, M., J. Longhill, and S. Rasmussen. Gluteus maximus muscle function and the origin of hominid bipedality. *American Journal of Physical Anthropology*, 7:519–528, 1988.
- [188] Mason, M. T., and J. K. Salisbury. *Robot Hands and the Mechanics of Manipulation*. MIT Press, Cambridge, MA, 1985.
- [189] McCann, G., and C. Wilts. Application of electric-analog computers to heat-transfer and fluid-flow problems. *Journal of Applied Mechanics*, 16(3):247–258, September 1949.
- [190] McCarty, M. E., R. K. Clifton, and R. R. Collard. Problem solving in infancy: The emergence of an action plan. *Developmental Psychology*, 35(4):1091–1101, 1999.
- [191] McCormick, S., editor. *Multigrid Methods (Frontiers in Applied Mathematics)*, volume 3. SIAM Books, Philadelphia, 1987.
- [192] McMahon, M. *Muscles, Reflexes, and Locomotion*. Princeton University Press, Princeton, NJ, 1984.
- [193] Metta, G., G. Sandini, L. Natale, and F. Panerai. Development and robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pages 33–42, Tokyo, Japan, 2001.
- [194] Meystel, A. Planning in a hierarchical nested controller for autonomous robots. In *Proceedings of the 25th Conference on Decision and Control*, pages 1237–1249, Athens, Greece, 1986. IEEE.
- [195] Minton, S. Constraint-based generalization: Learning game-playing plans from single examples. In *Proceedings of the National Conference on Artificial Intelligence*, pages 251–254, Austin, TX, 1984. Morgan Kaufmann.
- [196] Mishra, B., J. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.
- [197] Mitchell, T. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.
- [198] Modayil, J., and B. Kuipers. Autonomous development of a grounded object ontology by a learning robot. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence*, pages 1095–1101, Vancouver, BC, Canada, 2007. AAAI.
- [199] Modayil, J., and B. Kuipers. The initial development of object knowledge by a learning robot. *Robotics and Autonomous Systems*, 56(11):879–890, 2008.
- [200] Mountcastle, V. An organizational principle for cerebral function: The unit model and the distributed system. In G. Edelman and V. Mountcastle, editors, *The Mindful Brain*. MIT Press, Cambridge, MA, 1978.
- [201] Murase, H., and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.
- [202] Murray, R. M., Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, CA, 1994.
- [203] Mussa-Ivaldi, F. Modular features of motor control and learning. *Current Opinion in Neurobiology*, 9(6):713–717, 1999.
- [204] Mussa-Ivaldi, F. Nonlinear force fields: A distributed system of control primitives for representing and learning movements. In *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*, pages 84–90, San Mateo, CA, 1997. IEEE.

- [205] Mussa-Ivaldi, F., E. Bizzi, and S. Giszter. Transforming plans into action by tuning passive behavior: A field approximation approach. In *Proceedings of the 1991 International Symposium on Intelligent Control*, pages 101–109, Arlington, VA, August 1991. IEEE.
- [206] Muthukrishnan, C., D. Smith, D. Myers, J. Rebman, and A. Koivo. Edge detection in tactile images. In *Proceedings of 1987 Conference on Robotics and Automation*, pages 1500–1505, Raleigh, NC, April 1987. IEEE.
- [207] Myers, J. Multiarm collision avoidance using potential field approach. In *Space Station Automation*, pages 78–87, 1985. SPIE.
- [208] Nakamura, Y. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, Reading, MA, 1991.
- [209] Nakamura, Y., and H. Hanafusa. Optimal redundancy control of robot manipulators. *Journal of Robotics Research*, 6(1), Spring 1987.
- [210] Napier, J. *Hands*. Pantheon Books, New York, 1980.
- [211] Napier, J. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38b(4):902–913, November 1956.
- [212] Natale, L. *Linking action to perception in a humanoid robot: A developmental approach to grasping*. PhD thesis, LIRA-Lab, DIST, University of Genoa, 2004.
- [213] Newman, W., and N. Hogan. High speed control and obstacle avoidance using dynamic potential functions. In *IEEE International Conference on Robotics and Automation*, pages 14–24, Raleigh, NC, March 1987.
- [214] Newport, E. Maturation constraints on language learning. *Cognitive Science*, 14:11–28, 1990.
- [215] Nicolescu, M., and M. Mataric. A hierarchical architecture for behavior-based robots. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 227–233, Bologna, Italy, July 2002. ACM.
- [216] Okada, T. Computer control of multijointed finger system for precise object-handling. *IEEE Journal of Systems, Man, and Cybernetics*, 12(3):289–299, 1982.
- [217] Okada, T. Object-handling system for manual industry. *IEEE Journal of Systems, Man, and Cybernetics*, 9(2):79–89, February 1979.
- [218] Otero, T. F., and M. T. Cortes. Artificial muscles with tactile sensitivity. *Advanced Materials*, 15:279–282, 2003.
- [219] Özveren, C. M., and A. S. Willsky. Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, 35(7):797–806, 1990.
- [220] Paul, B. *Kinematics and Dynamics of Planar Machinery*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [221] Paul, R. *Robot Manipulators: Mathematics, Programming, and Control: The Computer Control of Robot Manipulators*. MIT Press, Cambridge, MA, 1981.
- [222] Penrose, R. A generalized inverse for matrices. In *Proceedings of the Cambridge Philosophical Society*, volume 51, pages 406–413, Cambridge, UK, 1955.
- [223] Pew, R. W., and S. B. Van Hemel, editors. *Technology for Adaptive Aging*. National Academies Press, Washington, DC, 2003.
- [224] Pfeifer, R., and J. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA, 2006.

- [225] Phillips, C. and R. Harbor. *Feedback Control Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [226] Piaget, J. *The Construction of Reality in the Child*. Basic Books, New York, 1954.
- [227] Piaget, J. *The Origins of Intelligence in Childhood*. International Universities Press, 1952.
- [228] Piater, J. H., and R. A. Grupen. Constructive feature learning and the development of visual expertise. In *Proceedings of the Seventeenth International Conference on Machine Learning*, Stanford, CA, 2000.
- [229] Pieper, D. *The kinematics of manipulators under computer control*. PhD thesis, Stanford University, 1968.
- [230] Pinker, S. *How the Mind Works*. Norton, New York, 1997.
- [231] Plantinga, H., and C. Dyer. Visibility, occlusion, and the aspect graph. *Computer Vision*, 5(2):137–169, 1990.
- [232] Plantinga, W., and C. Dyer. An algorithm for constructing the aspect graph. Technical Report 627, University of Wisconsin, 1985.
- [233] Platt, R. *Learning and generalizing control based grasping and manipulation skills*. PhD thesis, University of Massachusetts Amherst, September 2006.
- [234] Poulin-Dubois, D., A. Lepage, and D. Ferland. Infants' concept of animacy. *Cognitive Development*, 11:19–36, January–March 1996.
- [235] Prattichizzo, D., and J. Trinkle. Grasping. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Springer-Verlag, Secaucus, NJ, 2008.
- [236] Puterman, M. *Markov Decision Processes*. Wiley, New York, 1994.
- [237] Quinn, P., P. Eimas, and S. Rosenkrantz. Evidence for representations of perceptually similar natural categories by 3- and 4-month-old infants. *Perception*, 22:463–475, 1993.
- [238] Raibert, M. H. An all digital VLSI tactile array sensor. In *Proceedings 1984 Conference on Robotics*, pages 314–319, Atlanta, March 1984. IEEE.
- [239] Ramadge, P. J., and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–97, January 1989.
- [240] Rebman, J., and M. W. Trull. A robust tactile sensor for robotic applications. In *Proceedings of Computers in Engineering*, volume 2, pages 109–114, Chicago, 1983.
- [241] Reuleaux, F. *The Kinematics of Machinery*. Translated and annotated by A. B. W. Kennedy. 1876. Reprint, Dover, New York, 1963.
- [242] Rimon, E., and J. Burdick. New bounds on the number of frictionless fingers required to immobilize planar objects. *Journal of Robotic Systems*, 12(6):433–451, 1995.
- [243] Rimon, E., and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.
- [244] Rimon, E., and D. Koditschek. Exact robot navigation using cost functions: The case of distinct spherical boundaries in e^n . In *Proceedings of the International Conference on Robotics and Automation*, pages 1791–1796, April 1988. IEEE.
- [245] Rosenstein, M., and R. Grupen. Velocity-dependent dynamic manipulability. In *Conference on Robotics and Automation*, pages 2424–2429, Washington, DC, May 2002. IEEE.
- [246] Roy, D. Semiotic schemas: A framework for grounding language in the action and perception. *Artificial Intelligence*, 167:170–205, 2005.

- [247] Roy, N., G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montelermo, J. Pineau, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment (WIRE)*, 2000. <http://groups.csail.mit.edu/rrg/papers/wire2000.pdf>
- [248] Ruiken, D., J. Wong, T. Liu, M. Hebert, T. Takahashi, M. Lanighan, and R. Grupen. Affordance-based active belief recognition using visual and manual actions. In *International Conference on Intelligent Robots and Systems*, pages 5312–5317, Daejeon, Korea, October 2016. IEEE/RSJ.
- [249] Salisbury, J. K. Design and control of an articulated hand. In *Proceedings of the International Symposium on Design and Synthesis*, Tokyo, July 1984.
- [250] Salisbury, J. K. Interpretation of contact geometries from force measurements. In *Proceedings 1984 Conference on Robotics*, pages 240–247, Atlanta, March 1984. IEEE.
- [251] Salisbury, J. K. *Kinematic and force analysis of articulated hands*. PhD thesis, Stanford University, May 1982.
- [252] Salisbury, J. K. The Stanford/JPL hand: Mechanical specifications. Technical Report, Salisbury Robotics, Palo Alto, CA, 1984.
- [253] Sandini, G., G. Metta, and J. Konczak. Human sensorimotor development and artificial systems. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Intellectual Human Activity Support for Applications*, pages 303–314, 1997.
- [254] Scassellati, B. Building behaviors developmentally: A new formalism. In *AAAI Spring Symposium on Integrating Robotics Research*, Stanford, CA, 1998.
- [255] Schiff, W., and E. Foulke, editors. *Tactual Perception: A Sourcebook*. Cambridge University Press, New York, 1982.
- [256] Schlender, B. Intel’s Andy Grove: The next battles in tech. *Fortune*, pages 80–81, May 2003.
- [257] Schmidhuber, J. Self-motivated development through rewards for predictor errors/improvements. In D. Blank, L. Meeden, S. Franklin, O. Sporns, and J. Weng, editors, *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, Stanford, CA, 2005.
- [258] Sen, S. *Bridging the gap between autonomous skill learning and task specific planning*. PhD thesis, University of Massachusetts Amherst, September 2012.
- [259] Shepherd, G. *The Synaptic Organization of the Brain*. 2nd ed. Oxford University Press, New York, 1979.
- [260] Siegel, D. M., S. M. Drucker, and I. Garabieta. Performance analysis of a tactile sensor. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 3, pages 1493–1499, San Francisco, April 1986. IEEE.
- [261] Sininger, Y., K. Doyle, and J. Moore. The case for early identification of hearing loss in children. *Pediatric Clinics of North America*, 46(1):1–14, 2009.
- [262] Slater, A., and S. Johnson. Visual sensory and perceptual abilities of the newborn: Beyond the blooming, buzzing confusion. In F. Simion and G. Butterworth, editors, *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy: From Sensation to Cognition*, pages 121–141. Psychology Press, Hove, UK, 1997.
- [263] Sobh, M., J. C. Owen, K. P. Valvanis, and D. Gracani. A subject-indexed bibliography of discrete event dynamic systems. *IEEE Robotics & Automation Magazine*, 1(2):14–20, 1994.
- [264] Somov, P. Über schrauhengeschwindigkeiten eines festen körpers bei verschiedener zahl von stützflächen. *Zeitschrift für Mathematik und Physik*, 42:133–153, 1897.

- [265] Spelke, E. S., P. Vishton, and C. von Hofsten. Object perception, object-directed action, and physical knowledge in infancy. In M. S. Gazzaniga, editor, *The Cognitive Neurosciences*, pages 165–180. MIT Press, Cambridge, MA, 1995.
- [266] Stansfield, S. A. Primitives, features, and exploratory procedures: Building a robot tactile perception system. In *Proceedings 1986 Conference on Robotics and Automation*, volume 2, pages 1274–1279, San Francisco, April 1986. IEEE.
- [267] Steels, L. The autotelic principle. In I. Fumiya, R. Pfeifer, L. Steels, and K. Kunyoshi, editors, *Embodied Artificial Intelligence*, volume 3139 of Lecture Notes in Artificial Intelligence, pages 231–242. Springer-Verlag, 2004.
- [268] Sutton, R., and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [269] Takens, F. Detecting strange attractors in turbulence. In D. Rand and L. Young, editors, *Dynamical Systems and Turbulence*, volume 898 of Lecture Notes in Math, pages 366–381. Springer-Verlag: Lecture Notes in Math, 1981.
- [270] Tanaka, T. Gels. *Scientific American*, pages 124–138, January 1981.
- [271] Tarassenko, L., and A. Blake. Analogue computation of collision-free paths. In *International Conference on Robotics and Automation*, pages 540–545, April 1991. IEEE.
- [272] Thelen, E. Treadmill elicited stepping in seven month old infants. *Journal of Child Development*, 57:1498–1506, 1994.
- [273] Thelen, E., D. Corbetta, K. Kamm, J. Spencer, K. Schneider, and R. Zernicke. *The transition to reaching: Mapping intention and intrinsic dynamics*. Unpublished, 1992.
- [274] Thelen, E., and L. Smith. *A Dynamic Systems Approach to the Development of Cognition and Action*. MIT Press, Cambridge, MA, 1994.
- [275] Thompson, E. *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Harvard University Press, Cambridge, MA, 2007.
- [276] Thompson, R. *The Brain: An Introduction to Neuroscience*. W. H. Freeman and Company, New York, 1985.
- [277] Titze, I. *Principles of Voice Production*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [278] Todes, S. *Body and World*. MIT Press, Cambridge, MA, 2001.
- [279] Tubiana, R. *The Hand*. Saunders, 1981.
- [280] Turing, A. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.
- [281] Valdastri, P., S. Roccella, L. Beccai, E. Cattin, A. Menciassi, M. Carrozza, and P. Dario. Characterization of a novel hybrid silicon three-axial force sensor. *Sensors and Actuators A*, 123–124:249–257, 2005.
- [282] Varela, F., E. Thompson, and E. Rosch. *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA, 1991.
- [283] Vranish, J. Magnetoinductive skin for robots. In *Proceedings of the 1986 Conference on Robotics and Automation*, volume 2, pages 1292–1318, San Francisco, April 1986. IEEE.
- [284] Vukobratović, M., and B. Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.
- [285] Waldron, K., and J. Schiedeler. Kinematics. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 9–33. Springer-Verlag New York, Secaucus, NJ, 2008.

- [286] Waldron, K., and J. Schmiechler. Kinematics. In *Springer Handbook of Robotics*, pages 9–33. Springer-Verlag New York, Secaucus, NJ, 2007.
- [287] Wang, C., W. Timoszyk, and J. Bobrow. Payload maximization for open chained manipulators: Finding weightlifting motions for a PUMA 762 robot. *IEEE Transactions on Robotics and Automation*, 17(2):218–224, 2001.
- [288] Watkins, C. *Learning from delayed rewards*. PhD thesis, Cambridge University, 1989.
- [289] Watkins, C. J. C. H., and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.
- [290] Weng, J., J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504), 2001.
- [291] Westermann, G. *Constructivist neural network models of cognitive development*. PhD thesis, University of Edinburgh, 2000.
- [292] Whitcomb, L., D. Koditschek, and J. Cabrera. Toward the automatic control of robot assembly tasks via potential functions: The case of 2-d sphere assemblies. In *Proceedings of the Conference on Robotics and Automation*, volume 3, pages 2186–2191, Nice, France, May 1992. IEEE.
- [293] Wilson, F. R. *The Hand: How Its Use Shapes the Brain, Language, and Human Culture*. Vintage Books, New York, 1999.
- [294] Witkin, A. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1023, Karlsruhe, Germany, August 1983. IJCAI.
- [295] Wray, K., D. Ruiken, R. Grupen, and S. Zilberstein. Log-space harmonic function path planning. In *IROS, 2016*. IEEE/RSJ.
- [296] Yoshikawa, T. Analysis and control of robot manipulators with redundancy. In *Robotics Research: The First International Symposium*, pages 735–747, Bretton Woods, NH, 1984. MIT Press.
- [297] Yoshikawa, T. Dynamic manipulability of robotic manipulators. *Journal of Robotics Systems*, 2(1):113–124, 1985.
- [298] Yoshikawa, T. *Foundations of Robotics: Analysis and Control*. MIT Press, Cambridge, MA, 1990.
- [299] Yoshikawa, T. Manipulability of robotic mechanisms. *Journal of Robotics Research*, 4(2):3–9, Summer 1985.
- [300] Younger, B., and L. Cohen. Developmental changes in infants’ perception of correlations among attributes. *Child Development*, 57(3):803–815, June 1986.
- [301] Zalucky, A. ITA Interim Technical Report No. 5. AFWAL/MLTC Wright-Patterson AFB, Adept Technology, 1984.
- [302] Zanone, P. G., and J. A. S. Kelso. Relative timing from the perspective of dynamic pattern theory: Stability and instability. In J. Fagard and P. H. Wolff, editors, *The Development of Timing Control and Temporal Organization in Coordinated Action*. Elsevier Science, 1991.
- [303] Zelazo, N., P. R. Zelazo, K. Cohen, and P. D. Zelazo. Specificity of practice effects on elementary neuromotor patterns. *Child Psychology*, 29:686–691, 1993.
- [304] Zelinsky-Wibbelt, C. *Discourse and the Continuity of Reference: Representing Mental Categorization*. Walter de Gruyter, 2000.
- [305] Zhang, Q., H. Li, M. Poh, F. Xia, Z. Cheng, H. Xu, and C. Huang. An all-organic composite actuator material with a high dielectric constant. *Nature*, 419:284–287, 2002.

Index

- acceleration polytope, 160
- actin, 17
- actuators, 24
 - backdrivability, 34, 35, 37, 40
 - Bucky tubes, 42
 - chemical gels, 41
 - DC motor, 24–28, 30, 31
 - electroactive polymers, 41
 - hydraulic, 34, 37
 - pneumatic, 37
 - polymeric, 41
 - shape memory alloy, 40
 - synthetic muscle, 42
- amphibian reflex, 239, 244, 245
- analytic functions, 101
- angular momentum, 149, 153, 154
- annihilator, 260
- artificial intelligence, 277
- asymmetric tonic neck reflex (ATNR), 234, 244, 245
- asymptotic stability, 55
- auditory cortex, 224
- Australopithecus afarensis (Lucy), 116, 120

- Babinski sign, 236
- backdrivability, 34, 35, 37, 40
- back EMF, 25
- Bellman optimality condition, 275
- Bernstein, Nikolai, 2
 - biomechanics, 2
 - degrees of freedom problem, 2
 - dexterity, 2, 119
- black body radiation, 170

blink, [47](#), [231](#), [247](#)
Boyle, Robert, [225](#)
brachiation, [119](#)
Braitenberg, Valentino, [10](#)
buckytubes, [42](#)

center of mass, [151](#)
chemical gel actuator, [41](#)
Churchland, Paul, [81](#)
clasp reflex, [232](#). *See* [Moro](#)
closed-loop control, [11](#), [12](#), [47](#), [51](#), [63](#), [252](#), [259](#), [263](#), [285](#)
closed-loop transfer function, [62](#), [63](#), [65](#), [68](#), [327](#)
cogging, [28](#)
cognitive science, [2](#)
common ground, [3](#)
commutation, [25](#)
complex number, [316](#)
computed torque equation, [153](#), [154](#)
 analysis, [158](#)
 Coriolis and centrifugal force, [159](#), [334](#), [336](#)
 dynamic manipulability ellipsoid, [158](#)
 feedforward compensation, [157](#)
 generalized inertia matrix, [156](#)
 simulation, [157](#)
 state-space form, [154](#)
configuration space, [80](#)
 toroidal, [100](#)
configuration variable, [79](#)
conformal deformation, [254](#)
contact type, [133](#)
control basis, [12](#), [252](#), [256](#), [266](#)
 action, [251](#), [252](#), [256](#)
 combinatorial basis for, [252](#), [256](#)
 taxonomy, [251](#)
 aspect, [269](#), [270](#)
 Aspect Transition Graph (ATG), [270](#)
 basin of attraction, [265](#)
 co-articulation, [260](#)
 Markov Decision Process (MDP), [274](#)
 multi-objective control, [252](#), [260](#)
 null space, [260](#)
 “subject-to” operator, [260](#), [273](#)
 primitives, [260](#), [262](#), [283](#)
 contact control, [272](#), [273](#)
 fovea control, [272](#)
 isotropic conditioning control, [257](#)
 localizability control, [272](#), [273](#)
 manipulability control, [257](#), [272](#), [273](#)

- moment residual control, [265](#), [283](#)
- path control, [283](#)
- range of motion control, [272](#), [273](#), [283](#)
- skills
 - barcode reading policy, [272](#)
 - grasp policy, [265](#), [272](#)
- state, [264](#), [269](#), [270](#)
 - membership function, [269](#)
- convex, [253](#)
- convolution, [319](#)
- edge, [202](#)
 - gradient magnitude, [202](#)
 - orientation, [202](#)
- high-pass filter, [202](#)
- operators, [199](#)
- theorem, [197](#)
- critical point, [253](#)
- cross product, [301](#)
- curse of dimensionality, [5](#)
- Cutkosky, Mark, [121](#)

- DC motor, [24](#)
 - back EMF, [25](#)
 - cogging, [28](#)
 - commutation, [25](#)
 - efficiency, [31](#)
 - electrodynamics, [28](#)
 - gearheads, [33](#)
 - gear ratio, [34](#)
 - Lorentz force, [24](#)
 - motor constants, [26](#), [28](#)
 - power output, [31](#)
 - steady state velocity, [27](#)
 - torque, [27](#)
 - torque-current curve, [31](#)
 - torque-speed curve, [30](#)
- deep structure, [12](#)
- definiteness, [303](#)
- degenerate critical point, [253](#)
- degree of freedom (DOF), [80](#)
- dermatome, [49](#)
- developed muscular tension, [21](#)
- developmental curriculum, [251](#), [268](#)
- developmental reflexes
 - brainstem, [232](#)
 - asymmetric tonic neck reflex (ATNR), [234](#), [244](#), [245](#)
 - Babkin, [233](#)
 - Moro, [232](#), [244](#)

- palmar grasp reflex, 230, 233, 237, 244, 245
- placing, 235, 244
- plantar, 233, 236, 244
- primary stepping, 229, 235
- rooting, 232, 246
- startle, 233. *See* Strauss
- Strauss, 233
- sucking, 232, 244
- tongue retrusion, 232
- bridge, 236
 - avoid response, 236, 244, 245
 - doll's eye, 237. *See* oculocephalic
 - instinctive grasp, 237
 - Landau, 237, 244, 245
 - oculocephalic, 237, 244
 - radial release, 236, 244
 - symmetric tonic neck reflex (STNR), 238, 244, 245
 - traction, 236, 244, 245
 - ulnar release, 236, 244, 245
- limbic, 229–230
 - acoustic stapedius, 231
 - audito-oculogyric, 231
 - blink, 47, 231, 247
 - breathing, 229
 - elimination, 231
 - flexor-withdrawal, 49, 226, 230
 - gag, 231
 - laryngeal, 231
 - oculomotor, 231
 - orbicularis, 231
 - orienting, 231
 - pharyngeal, 231
 - sneeze, 231
 - swallowing, 229
- postural, 238
 - amphibian, 239, 244, 245
 - body-on-head, 239. *See* righting
 - labyrinthine, 237, 238
 - labyrinthine-head righting (LHRR), 239, 243–245
 - lateral propping, 239, 245
 - optical-head righting (OHRR), 239, 243–245
 - parachute, 239, 244
 - positive support, 239, 244, 245
 - righting, 239, 244, 245
 - segmental rolling, 239, 244, 245
 - vestibulo-ocular, 241
- spinal, 237
 - amphibian, 235

- flexor-withdrawal, 49, 226, 230
- Galant, 234, 245
- palmar grasp reflex, 230, 233, 237, 244, 245
- primary stepping, 229, 235
- segmental, 49, 252
- segmental rolling, 235
- stretch, 48, 50, 51
- developmental robotics, 6
- dexterity, 2, 119, 121
- dexterous workspace, 90
- dot product, 299
- Du Lian, L., 126
- dynamic conditioning
 - acceleration polytope, 160
 - dynamic manipulability ellipsoid, 160
 - generalized inertia ellipsoid, 158
 - gravitational bias, 160
 - velocity bias, 159, 160
- dynamic coupling, 147, 155–158
- dynamic manipulability ellipsoid, 160
- dynamic programming
 - full backup, 275
- dynamical systems theory, 6

- elastic potential, 252
- embodiment, 3, 5, 10
 - philosophy, 5
- equilibrium point theory, 253
- equivalence principle, 345
- Ernst, Heinrich, 123
- Euler's equation, 149
- Euler's formula, 316, 325
- exceptional surfaces, 140
- exteroception, 191
- eye, 174. *See* human eye
- eyespots, 174

- Feynman, Richard, 10
- fixed point, 254
- flexor-withdrawal reflex, 49, 226, 230
- force closure, 139, 140
- form closure, 139
- forward kinematics, 81
- Fourier transform, 191, 201, 316, 317
 - convolution theorem, 319
 - inverse, 192, 317
 - shift theorem, 318
 - spectral coefficients, 316

free body diagram, 53, 154
Frobenius inner product, 203

Galant reflex, 234, 245
generalized grasp Jacobian, 136
generalized inertia ellipsoid, 156, 158
Gergely, György, 248
glial cells, 241
grasp
 contact type, 133
 Jacobian, 132
 palmar grasp reflex, 230, 233, 237, 244, 245
 pincer grasp, 230
 selection matrix, 134
 taxonomy, 121
grasp Jacobian, 132, 133
gross motor skills, 281

Hanafusa, Hideo, 124
haptics, 178
harmonic function
 hitting probability, 255, 356
 mean-value property, 351
 numerical relaxation, 351
 occupancy grid, 283, 285, 294, 353–355
 resolution-complete, 355
harmonic oscillator
 amplitude and phase response, 326, 327
 boundary conditions, 323
 characteristic equation, 61
 complex conjugate roots, 321
 distinct real roots, 321
 repeated real roots, 322
 over-, under-, and critically damped, 67, 323–325
 response bandwidth, 327
 time-domain response, 321
 transfer function
 admittance, 330
 closed-loop transfer function (CLTF), 327
 compliance, 329
 impedance, 329, 330
 impedance control, 330
Hawkins, Jeff, 224
Hessian, 253, 254, 304
hierarchical action, 273
hierarchy, 221, 223, 226, 277
 human nervous system, 226
 in AI, 277

- in developmental psychology, 277
- in the central nervous system (CNS), 277
- macro-operator, 277
- neocortex, 223
- schema, 277
- high-pass filter, 203
- hitting probability, 255
- homogeneous transform, 85, 86
 - composition, 88
 - Denavit-Hartenberg, 88
 - inverse, 87
- Hooke's law, 252
- human eye
 - accommodation reflex, 176
 - anatomy, 175
 - evolution, 174
 - pupillary light reflex, 176
 - retina, 174
 - rods and cones, 174
 - spectral sensitivity, 174
 - squinting, 176
 - variations in other species, 177
- human hand, 119
 - finger, 120
 - palmar grasp reflex, 230, 233, 237, 244, 245
 - pincer grasp, 230
 - thumb, 121
- human voice, 192
 - formants, 193
 - resonance, 193
 - spectrogram, 194
- Huxley muscle model, 20
- hydraulic actuator, 34
 - jumping spider, 37

- Iberall, Thia, 134
- inertia tensor, 148, 149, 331, 333
 - rotation, 153
 - translation, 152
- inertial coordinate frame, 147, 334
- inertial coupling, 147, 155–158
- infant
 - hearing, 247
 - pain, 247
 - smell, 247
 - taste, 247
 - temperature, 246
 - touch, 246

- vision, 247
 - binocular depth, 247
 - motion, 247
 - visual cliff, 248
- instinctive grasp response, 237

Jacobi, Carl Gustav, 101

Jacobian, 101

- conditioning
 - amplitude, 105
 - precision, 105
- conditioning ellipsoid, 104
- control, 255
- force ellipsoid, 105
- inverse, 103
- manipulability ellipsoid, 104
- manipulator, 101
- nonsquare, 109
- nullspace, 109
- precision ellipsoid, 105
- pseudoinverse, 111
- redundant, 111
- self-motion manifold, 109
- singular, 103
- velocity ellipsoid, 104

kinematic chain, 79

kinematics, 79

- conditioning, 101
- dexterous workspace, 90
- forward, 81, 88
- hand-eye coordination, 98
- human arm, 91
- inverse, 88, 91
 - closed-form, 91
 - Pieper's solution, 91
- Jacobian, 101
- manipulator, 88
- nonholonomic constraints, 83
- of stereo reconstruction, 94
- pinhole camera, 95
- planar 2R manipulator, 88
- projective geometry, 95
- PUMA 560, 91
- reachable workspace, 90
- redundancy, 109

kinodynamic lenses, 165

knowledge

- explicit, [3](#), [274](#)
- implicit, [4](#), [274](#)
- situated, [5](#)
- tacit, [4](#), [10](#), [273](#), [278](#)

Labyrinthine-head righting reflex (LHRR), [239](#), [245](#)

Labyrinthine reflexes, [238](#), [244](#)

Lagrange's equation, [155](#)

Lagrangian, [155](#)

Lagrangian equation, [349](#)

Lagrangian mechanics, [349](#)

Landau reflex, [237](#), [244](#), [245](#)

landscape of attractors, [251](#)

Laplace's equation, [254](#)

- numerical solution, [351](#)

Laplace transform, [59](#), [60](#), [316](#), [320](#)

- complex frequency, [320](#)
- Euler's formula, [316](#)
- harmonic oscillator, [321](#)
- inverse, [320](#)
- time-domain solutions, [321](#)
- unit step, [321](#)

lateral propping reflex, [239](#), [245](#)

light, [170](#)

- black body, [170](#)
- luminosity function, [170](#)
- Poynting vector, [171](#)
- reflection, [170](#)
- refraction, [170](#)

linear independence, [299](#)

Linear integral transform, [315](#)

- bijection, [315](#)
- Fourier transform, [191](#)
 - inverse, [192](#)
- Laplace transform, [320](#)

Longuet-Higgins, Christopher, [2](#)

- cognitive science, [2](#)

lower motor neurons, [49](#)

low-pass filter, [201](#)

Lucy (*Australopithecus afarensis*), [116](#), [120](#)

luminosity function, [170](#), [208](#), [214](#), [317](#)

Lyapunov, Aleksandr, [54](#)

- asymptotically stable, [55](#)
- direct method, [54](#), [56](#)

manipulability, [104](#), [257](#)

Markov Decision Process, [274](#)

matrix, [310](#)

- cofactors, 302
- condition number, 307, 309
- determinant, 307
- determinate, 302
- geometric mean, 310
- inverse, 302
- minimum singular value, 309
- norm, 305
- null space, 308
- pseudoinverse, 312
- singular values, 309
- singular vectors, 309
- span, 308
- maturation, 240, 251
- mechanism, 79
 - closed-chain, 80
 - joint, 79
 - prismatic, 79
 - revolute, 79
 - link, 79
 - ground link, 80
 - unitary, 80
 - open-chain, 80
- membership function, 264, 265, 267, 269
- moment of inertia, 33, 148, 149
- Moment residual control, 265
- Moore-Penrose pseudoinverse, 256, 261
- Moro reflex, 232, 244
- motor cortex, 226
- motor nuclei, 49
- motor unit, 47, 50, 51, 58, 59, 61, 65, 221, 226, 252, 255
 - lower motor nuclei, 49
- mouthing behavior, 246
- muscle, 17–23
 - actin, 17
 - bulk viscoelastic model, 23
 - developed tension, 21
 - fused/unfused tetanus, 21
 - Huxley model, 20
 - myocytes, 19
 - myofibril, 19
 - myosin, 17, 18
 - sarcomere, 19
 - sliding filament model, 18
 - twitch, 21
- muscle stretch reflex, 50, 51
- myelin, 241
- myelination, 240, 241

- corticospinal tract, 241
 - proximo-distal, 243, 246
 - ulnar to radial, 246
- vestibular nerve, 241
- vestibulo-ocular, 241
- myocytes, 19
- myofibril, 19
- myosin, 17, 18
- myotome, 49

- nautilus, 174
- navigation function, 255
- negative definite, 254
- negative feedback, 47, 51
- neonate, 229
- neural tube, 221
- neuroanatomy
 - basal ganglia, 226
 - brainstem, 222, 228
 - cerebellum, 222, 226, 228
 - cerebral cortex, 222. *See* cerebrum
 - cerebrum, 222
 - auditory cortex, 224
 - cerebral hemispheres, 223, 224
 - gray matter, 223, 224
 - motor cortex, 226
 - pyramidal neurons, 224
 - somatosensory cortex, 224
 - visual cortex, 224, 225
 - white matter, 224
 - cortex, 222. *See* cerebrum
 - midbrain
 - lateral geniculate nucleus, 225
 - thalamus, 224–226
 - neocortex, 222. *See* cerebrum
 - spinal cord, 222, 226, 228
- neurons, 241
 - axons, 241
 - dendrites, 241
- Newton-Euler iterations, 155, 333, 337, 341
- Newton's laws, 147
- nonholonomic constraints, 83
- null space, 256
- Nyquist, Harry, 199
- Nyquist sampling theorem, 198

- object permanence, 248
- occupancy grid, 283, 285, 294, 353–355

- FREESPACE, 285
- oculo-cephalic reflex, 237, 244
- odometry, 81
- Okada, Takashi, 125
- Open-closed resonator, 192. *See* quarter wave resonator
- optical-head righting reflex (OHRR), 239, 244, 245
- optics, 170
 - biconvex lens, 172
 - evolution, 174
 - Gaussian lens formula, 172
 - index of refraction, 171
 - reflective, 170
 - refractive, 170
 - Snell's law, 171
 - thin lens, 172
 - thin lens equation, 173
- orthonormal, 299
- outer product, 302
- outward-inward iterations, 333. *See* Newtown-Euler iterations

- palmar grasp reflex, 230, 233, 237, 244, 245
- parachute reflex, 240, 244
- parallel axis theorem, 151
- pathfinder, 229
- perspective
 - vanishing point, 96
- perspective projection
 - orthographic projection, 96
 - shallow structure, 96
- phase portrait, 264, 265
- phylum chordata, 222
- Piaget, Jean, 277
 - accomodation, 277, 281
 - assimilation, 277, 281
 - schema, 277
- pinhole camera, 95, 170
 - depth of field, 171
 - history, 95
 - perspective, 95
 - perspective projection, 96
- Pinker, Steven, 119
- pit organ, 174
- placing reaction, 235, 244
- plantar reflex, 233, 236, 244
- plasticity, 242
- pneumatic actuator, 37
 - jet-pipe servo valve, 37
 - McKibben air muscle, 39

- polymeric actuator, 41
- Popplestone, Robin, 124
- positive definite, 253
- positive support reflex, 239, 244, 245
- potential function, 252
 - differential geometry, 253
 - elastic, 252
 - navigation function, 255
 - region of attraction, 268
 - streamlines, 254
- power, 133
- preadaptation primitives, 230
- primary stepping reflex, 229, 235
- products of inertia, 149, 333
- projective geometry, 96
- proprioception, 191
- pseudoinverse, 111, 256, 261, 312

- q-learning, 274
 - value iteration, 275
- quadratic forms, 305
- quarter wave resonator, 192, 193

- reachable workspace, 90
- reinforcement learning, 274
 - sampled backup, 275
- release reflexes, 236, 244, 245
- right-handed coordinate systems, 301
- righting reflex, 239, 244, 245
- robot hands, 122–126, 128
 - barrett hand, 270
 - DLR II hand, 128
 - Freddy, 123
 - Hanafusa hand, 124
 - Lian manipulator, 126
 - Okada hand, 125
 - parallel jaw gripper, 124
 - Stanford/JPL hand, 126
- robots
 - Barrett Whole Arm Manipulator (WAM), 270
 - dexter, 272
 - PUMA 560, 91
 - Sarcos General, Large Robot Arm (GRLA), 35
 - Shadow Hand-B, 40
 - Thing, 281
 - Utah/MIT dexterous hand, 257
- Roger-the-Crab
 - coordinate frames, 82

- kinematics, 81
 - nonholonomic base, 83
- rooting reflex, 232, 246
- rotating coordinate systems, 334
- rotation, 84
 - direction cosine matrix, 85
- saddle point, 254
- sampling, 191, 196
 - continuous signals, 191
 - convolution theorem, 197
 - Dirac delta function, 196
 - Nyquist sampling theorem, 198
 - sifting, 196
- sampling continuous signals, 191
- sarcomere, 19
- scalar product, 299
- screw systems, 129, 132
 - power, 132
 - twist, 129
 - wrench, 132
- second-order system, 321. *See* harmonic oscillator
- segmental rolling reflex, 239, 244, 245
- sensors
 - vestibular, 238
- Shannon, Claude, 199
- shape memory alloy, 40
- signal processing
 - convolution
 - differential geometry, 207
 - Gaussian kernel, 209
 - Gaussian operators, 208
 - Gaussian pyramid, 209
 - intrinsic scale, 209
 - Laplacian, 207
 - Laplacian of Gaussian, 212
 - deep structure, 12, 208
 - differential structure, 205
 - N -jet, 205
 - intrinsic scale, 208
 - scale space, 208
 - causality, 205, 209
 - deep structure, 209
 - intrinsic scale, 212
 - intrinsic structure, 209
 - structural information theory, 208
- singular value decomposition, 307
- skills

- locomotion, 283
- sliding filament model, 18
- Snell's law, 171
- somatosensory cortex, 224
- spectral coefficients, 201
- spectral filter, 201
 - high-pass, 202, 203
 - low-pass, 201
- spinal cord, 48
 - afferents, 222, 224
 - dorsal roots, 222, 224
 - efferents, 222
 - ventral roots, 222
- spring-mass-damper (SMD), 47
- stability, 51, 55
 - asymptotic, 55, 67, 253
 - BIBO, 54
 - critical points, 253
 - Lyapunov, 55
 - marginal, 67
 - negative feedback, 51
- startle, 233. *See* Strauss reflex
- Stefan-Boltzmann equation, 170
- stereo
 - triangulation, 100
- stereo localization, 107
- stereo vision, 94
- Strauss reflex, 233
- streamlines, 254
- stretch reflex, 48
- sucking reflex, 232, 244
- symmetric tonic neck reflex (STNR), 238, 244, 245
- synaptogenesis, 241
- synthetic muscle, 42

- tactile sensing, 181
 - addressing, 182
 - bandwidth, 182
 - hysteresis, 182
 - sensitivity, 182
 - spatial resolution, 181
- tactile sensors
 - binary contact switch, 182
 - capacitive, 185
 - conductive elastomers, 183
 - load cells, 182
 - magnetic, 186
 - multimodal, 186

- optical, 184
- teleological agent, 248
- tensor, 148, 149, 331
- tetanus, fused/unfused, 21
- touch, 178
 - mechanoreceptors, 179
 - articular surfaces, 179, 247
 - free-ended nerve fibers, 49, 180, 246
 - Golgi tendon organ, 179
 - hair, 180
 - Meissner corpuscles, 180
 - Merkel disks, 180
 - muscle spindles, 179
 - Pacinian corpuscles, 180
 - performance, 181
 - Ruffini corpuscles, 180
 - touch blend, 181
- trace, 254
- transfer function, 61, 62
 - admittance, 330
 - closed-loop transfer function (CLTF), 62, 63, 65, 68, 327
 - compliance, 329
 - impedance, 329, 330
 - stiffness, 329
- translation, 84
- twist, 129
- twitch, 21

- value iteration, 275
- vector
 - direction, 299
 - magnitude, 299
- vector product, 301
- vertebrate, 222
- vestibulo-ocular reflex, 241
- virtual finger, 134
- viscoelastic muscle model, 23
- vision, 207. *See* [signal processing](#)
 - binocular, 96
 - disparity, 97
 - triangulation, 96
 - CMOS image capture, 178
 - cortex, 224
 - disparity, 225
 - edges, 225
 - high-pass filter, 202
 - image formation, 170
 - luminosity function, 170, 208, 214, 317

- motion, [225](#)
- optic nerve, [225](#)
- scale space
 - blob, [209](#)
- stereo localization, [96](#)
- stereo triangulation
 - localizability, [107](#)

Wilson, Frank, [119](#)

withdrawal reflex, [226](#). *See* [flexor-withdrawal reflex](#)

workspace

- dexterous, [90](#)

- reachable, [90](#)

wrench, [132](#), [134](#)

Intelligent Robotics and Autonomous Agents

Edited by Ronald C. Arkin

Billard, Aude, Sina Mirrazavi, and Nadia Figueroa, *Learning for Adaptive and Reactive Robot Control*

Dorigo, Marco, and Marco Colombetti, *Robot Shaping: An Experiment in Behavior Engineering*

Arkin, Ronald C., *Behavior-Based Robotics*

Stone, Peter, *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*

Wooldridge, Michael, *Reasoning About Rational Agents*

Murphy, Robin R., *Introduction to AI Robotics*

Mason, Matthew T., *Mechanics of Robotic Manipulation*

Kraus, Sarit, *Strategic Negotiation in Multiagent Environments*

Nolfi, Stefano, and Dario Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*

Siegwart, Roland, and Illah R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*

Breazeal, Cynthia L., *Designing Sociable Robots*

Bekey, George A., *Autonomous Robots: From Biological Inspiration to Implementation and Control*

Choset, Howie, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*

Mataric, Maja J., *The Robotics Primer*

Wellman, Michael P., Amy Greenwald, and Peter Stone, *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*

Floreano, Dario and Claudio Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*.

Sterling, Leon S. and Kuldar Taveter, *The Art of Agent-Oriented Modeling*

Stoy, Kasper, David Brandt, and David J. Christensen, *An Introduction to Self-Reconfigurable Robots*

Lin, Patrick, Keith Abney, and George A. Bekey, editors, *Robot Ethics: The Ethical and Social Implications of Robotics*

Weiss, Gerhard, editor, *Multiagent Systems, second edition*

Vargas, Patricia A., Ezequiel A. Di Paolo, Inman Harvey, and Phil Husbands, editors, *The Horizons of Evolutionary Robotics*

Murphy, Robin R., *Disaster Robotics*

Cangelosi, Angelo and Matthew Schlesinger, *Developmental Robotics: From Babies to Robots*

Everett, H. R., *Unmanned Systems of World Wars I and II*

Sitti, Metin, *Mobile Microrobotics*

Murphy, Robin R., *Introduction to AI Robotics, second edition*

Gruppen, Roderic, *The Developmental Organization of Robot Behavior*

Boissier, Olivier Rafael H. Bordini, Jomi F. Hübner, and Alessandro Ricci, *Multi-Agent Oriented Programming*

Cangelosi, Angelo and Minoru Asada, *Cognitive Robotics*