



From the MixCache.com library

SAMPLE COPY

Control Theory for Modern Robotics

MixCache.com

SAMPLE COPY

Table of Contents

- **Introduction**
- **Chapter 1** Mathematical Foundations for Robotic Control
- **Chapter 2** Modeling Rigid-Body Dynamics and Actuation
- **Chapter 3** Sensors, Noise, and State Estimation Basics
- **Chapter 4** PID Control: Design, Tuning, and Anti-Windup
- **Chapter 5** Frequency-Domain Tools: Bode, Nyquist, and Loop Shaping
- **Chapter 6** State-Space Systems: Realization, Controllability, Observability
- **Chapter 7** Stability Analysis: Lyapunov Methods, Invariance, and Passivity
- **Chapter 8** Optimal Control Foundations: Variational Principles and PMP
- **Chapter 9** LQR and LQG: Regulation, Tracking, and the Separation Principle
- **Chapter 10** Model Predictive Control: Constraints, Warm Starts, and Real-Time MPC
- **Chapter 11** Robust Control: H_∞ , μ -Synthesis, and Disturbance Rejection
- **Chapter 12** System Identification for Robotics: From First Principles to Data-Driven Models
- **Chapter 13** Trajectory Optimization and Differential Dynamic Programming
- **Chapter 14** Impedance and Force Control for Manipulation
- **Chapter 15** Hybrid Systems and Contact-Rich Dynamics
- **Chapter 16** Legged Locomotion Control: Templates, Gaits, and Whole-Body Control
- **Chapter 17** Underactuated and Aerial Robot Control
- **Chapter 18** Learning for Control: Supervised, Imitation, and Reinforcement Learning
- **Chapter 19** Safe Learning: Control Barrier Functions, Risk, and Constraints
- **Chapter 20** Gaussian Processes and Bayesian Optimization in Control
- **Chapter 21** Residual and Hybrid Controllers: Combining Models with Learned Policies
- **Chapter 22** Sim-to-Real Transfer and Domain Randomization
- **Chapter 23** Tuning and Autotuning: Practical Performance Engineering
- **Chapter 24** Real-Time Implementation: Middleware, Scheduling, and Hardware-in-the-Loop
- **Chapter 25** Verification, Validation, and Benchmarking of Robotic Controllers

Introduction

Robotics has always lived at the intersection of elegant mathematics and stubborn reality. Motors saturate, gears flex, sensors drift, and contact with the world is discontinuous and unforgiving. Yet, when a robot deftly threads a needle or bounds across uneven terrain, it does so because its control system transforms models, data, and constraints into purposeful action. This book is about that transformation. It bridges classical control—whose guarantees and interpretability have guided engineers for decades—with learning-based methods that adapt, generalize, and uncover structure that models alone may miss.

Our approach is pragmatic. We begin with the pillars of classical practice: PID control and frequency-domain tools for shaping loops and margins, followed by state-space thinking, stability analysis, and optimal control. These methods give you the language to reason about dynamics, uncertainty, and performance. From there we develop modern, constraint-aware controllers such as model predictive control, and we study robustness to the disturbances, modeling errors, and delays that every real robot must endure.

Learning-based controllers enter not as replacements for models but as companions. Supervised and imitation learning help capture complex mappings where analytic models are brittle; reinforcement learning optimizes behavior when performance criteria are hard to encode analytically; Gaussian processes and Bayesian optimization enable data-efficient adaptation; and hybrid designs—such as residual learning atop nominal model-based controllers—leverage the best of both worlds. Throughout, we foreground safety: stability certificates via Lyapunov and passivity, constraint satisfaction via predictive control and barrier functions, and risk-aware decision making.

Examples anchor the theory. For manipulators, we examine impedance and force control for contact-rich tasks, robust trajectory tracking under actuator limits, and strategies for handling model mismatch and unmodeled flexibilities. For legged robots, we progress from template models to whole-body control under intermittent contact, exploring gait generation, disturbance rejection, and terrain adaptation. Along the way, we connect the math to implementation details—state estimation pipelines, timing and scheduling in real-time systems, and how solver choices and numerical conditioning influence what actually runs on hardware.

This is a nonfiction text intended for graduate students, researchers, and practitioners in robotics, control, and allied fields. Readers are expected to be comfortable with linear algebra, differential equations, and basic probability, and to have some

programming experience for simulation and experimentation. The narrative emphasizes insight and design trade-offs: why a particular controller is chosen, what can go wrong, and how to diagnose and fix it. Proofs are used when they illuminate design, and pointers are offered for deeper theoretical study.

The chapters are organized to support multiple entry points. If you are new to control, start with modeling and PID, then progress through state-space, stability, and optimal control before tackling MPC and robustness. If you come from machine learning, you may prefer to skim the classical foundations and dive into learning for control, safe exploration, and hybrid architectures, circling back to stability tools as needed. Application-focused readers can jump ahead to manipulation, hybrid dynamics, and legged locomotion, using the earlier chapters as reference.

Our central thesis is simple: precise robot behavior emerges when models, data, and constraints are treated as coequal citizens. The challenge is not merely to make a controller work in simulation, but to make it work on a physical robot, repeatedly, safely, and fast enough for the task at hand. By the end of this book, you will be able to design, analyze, and implement controllers that do exactly that—combining classical structure with learning-driven adaptability to meet the demands of modern robotics.

CHAPTER ONE: Mathematical Foundations for Robotic Control

To command a robot is to speak its language, and that language, at its core, is mathematics. Before we can delve into the intricacies of PID loops or the adaptive magic of learning, we must establish a common ground of mathematical tools. Think of this chapter as your linguistic boot camp, equipping you with the fundamental vocabulary and grammar to articulate control problems and decipher their solutions. We'll revisit concepts you might have encountered before, but with a specific robotic flavor, emphasizing how these abstract ideas manifest in the tangible world of joints, forces, and sensors.

We begin with vectors and matrices, the bedrock of nearly all robotic representations. A robot's state – its position, orientation, velocity – is almost invariably described by vectors. The transformations between different coordinate frames, the relationships between joint torques and end-effector forces, and the propagation of sensor noise are all elegantly captured by matrix operations. We'll move beyond simple definitions to explore concepts like linear independence, basis vectors, and vector spaces, understanding how these ideas allow us to define the "workspace" of a robot and the directions it can move. Eigenvalues and eigenvectors, those often-mysterious mathematical entities, will emerge as powerful tools for analyzing system stability and understanding how systems respond to inputs. They tell us about the natural modes of a system, revealing whether a robot arm will gracefully settle into position or oscillate wildly after a command.

Next, we'll dive into the world of calculus, but with a practical twist. Derivatives, in our context, aren't just about slopes of curves; they represent velocities and accelerations, the very essence of robot motion. We'll explore how to take derivatives of vector-valued functions, crucial for understanding the time evolution of a robot's state. Integrals, conversely, will allow us to move from velocities back to positions, or from accelerations to velocities. We'll pay particular attention to ordinary differential equations (ODEs), which are the universal language for describing dynamic systems. Every robot, from the simplest wheeled platform to the most complex humanoid, can be characterized by a set of ODEs that dictate how its state changes over time in response to forces, torques, and external disturbances. We'll look at first-order and second-order systems, understanding how their solutions reveal fundamental response characteristics like damping and oscillation, which are critical for designing smooth, stable robot motions.

Moving beyond scalar calculus, we'll embrace multivariate calculus, specifically partial

derivatives and gradients. When a robot has multiple joints or operates in a high-dimensional space, its behavior depends on many variables simultaneously. Partial derivatives help us understand the isolated effect of changing one variable while holding others constant. The gradient, a vector of partial derivatives, will become our compass, pointing in the direction of the steepest ascent or descent of a function. This is immensely valuable in optimization problems, where we often want to minimize errors or maximize performance by nudging various control parameters. Imagine trying to find the quickest path for a robot arm through a cluttered environment; the gradient of a cost function (representing time or collision risk) will guide the arm's trajectory.

Linear algebra and calculus converge in the Jacobian matrix, a concept so fundamental to robotics that it deserves special attention. The Jacobian is essentially a matrix of partial derivatives that relates velocities in one coordinate space to velocities in another. For a robot manipulator, the Jacobian connects the angular velocities of its joints to the linear and angular velocities of its end-effector. This matrix allows us to answer questions like: "If I want my robot's gripper to move at this speed in this direction, how fast should each joint rotate?" Or, conversely, "If these are my joint speeds, what is the resulting motion of the gripper?" The inverse Jacobian will prove equally important for inverse kinematics and resolving redundancies in robot motion. We will explore its properties, including its rank and null space, which offer insights into a robot's dexterity and its ability to achieve certain motions or avoid others. Singularities of the Jacobian, those infamous points where a robot loses a degree of freedom, will be discussed, highlighting their practical implications for robot design and control.

Tensors, while perhaps sounding intimidating, are simply generalizations of scalars, vectors, and matrices. In robotics, they become particularly relevant when dealing with concepts like inertia, stress, or strain. While we won't delve into the full mathematical abstraction of tensors in this chapter, we'll introduce the idea of a second-order tensor as a way to represent quantities that have both magnitude and direction, and whose effect depends on the direction of application. The inertia tensor, for instance, describes how resistant a rigid body is to rotation about different axes. Understanding these representations is crucial for accurately modeling robot dynamics, as we'll see in subsequent chapters.

Finally, we'll touch upon the essentials of probability and statistics. No robot operates in a perfect, deterministic world. Sensors are noisy, actuators have uncertainties, and the environment itself is unpredictable. Probability theory provides the framework for quantifying and reasoning about this uncertainty. We'll introduce basic concepts like probability distributions, expected values, and variance. These ideas are indispensable for state estimation, where we try to infer a robot's true state from noisy sensor readings, and for designing robust controllers that can tolerate imperfections. Gaussian distributions, in particular, will become a recurring theme due to their

prevalence in modeling noise and their convenient mathematical properties. We won't be building complex probabilistic models just yet, but laying the groundwork for how uncertainty is rigorously handled in modern control systems, especially when we venture into the realm of learning-based approaches.

Throughout this chapter, our goal is not just to present definitions, but to build intuition. We'll illustrate these mathematical concepts with simple robotic examples, showing how a vector can represent a robot's position, a matrix can rotate its orientation, a derivative can track its speed, and a probability distribution can capture the uncertainty in its measurements. By the end, you should feel comfortable speaking the mathematical language of robotics, ready to engage with the more advanced control strategies that follow. This foundation will be the lens through which we analyze system behavior, design controllers, and ultimately, bring intelligent machines to life. Consider this your essential toolkit; master these tools, and you'll be well-prepared for the engineering challenges ahead.

SAMPLE COPY

This is a sample preview. Purchase the book to read the full content.

Visit MixCache.com to purchase the complete book.

SAMPLE COPY