Advanced Textbooks in Control and Signal Processing

Eugene Lavretsky
Kevin A. Wise

Robust and Adaptive Control

With Aerospace Applications

Springer

Advanced Textbooks in Control and Signal Processing

Series Editors

Professor Michael J. Grimble, Professor of Industrial Systems and Director
Professor Michael A. Johnson, Professor of Control Systems and Deputy Director

Industrial Control Centre, Department of Electronic and Electrical Engineering,
University of Strathclyde, Graham Hills Building, 50 George Street, Glasgow G1 1QE, UK

For further volumes:
http://www.springer.com/series/4045

Eugene Lavretsky • Kevin A. Wise

Robust and Adaptive Control

with Aerospace Applications

Springer

Eugene Lavretsky
The Boeing Company
Huntington Beach, California, USA

Kevin A. Wise
The Boeing Company
Boeing Phantom Works
St. Louis, Missouri, USA

Additional material to this book can be downloaded from http://extras.springer.com
using the following password: 978-1-4471-4395-6

ISBN 978-1-4471-4395-6 ISBN 978-1-4471-4396-3 (eBook)
DOI 10.1007/978-1-4471-4396-3
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2012951336

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Series Editors' Foreword

The topics of control engineering and signal processing continue to flourish and develop. In common with general scientific investigation, new ideas, concepts, and interpretations emerge quite spontaneously, and these are then discussed, used, discarded, or subsumed into the prevailing subject paradigm. Sometimes, these innovative concepts coalesce into a new subdiscipline within the broad subject tapestry of control and signal processing. This preliminary battle between old and new usually takes place at conferences, through the Internet and in the journals of the discipline. After a little more maturity has been acquired by the new concepts, then archival publication as a scientific or engineering monograph may occur.

A new concept in control and signal processing is known to have arrived when sufficient material has evolved for the topic to be taught as a specialized tutorial workshop or as a course to undergraduate, graduate, or industrial engineers. *Advanced Textbooks in Control and Signal Processing* is designed as a vehicle for the systematic presentation of course material for both popular and innovative topics in the discipline. It is hoped that prospective authors will welcome the opportunity to publish a structured and systematic presentation of some of the newer emerging control and signal processing technologies in the textbook series.

An aim of the *Advanced Textbooks in Control and Signal Processing* series is to create a library that covers all the main subjects to be found in the control and signal processing fields. It is a growing but select series of high-quality books that now covers some fundamental topics and many more advanced topics in these areas. We are therefore very fortunate to have this textbook from Eugene Lavretsky and Kevin Wise on *Robust and Adaptive Control with Aerospace Applications* enter the series. In many ways, this textbook is a departure for the series since it deals with the fundamental topics of robust and adaptive control and has very strong material from the aerospace applications field. Thus, it is possible to see clearly how the stringent performance requirements of the applications motivate and are met by the control theory developments. From the aerospace control applications, the reader will appreciate the industrial context where aircraft operates across a wide range of flight conditions, giving rise to many design points. The aerospace industry solves this problem by using many strategically selected control design points

and gain schedules the resulting controllers. Another part of the design context is the need for control designs that are able to track a range of reference signals while remaining robust to both parametric and nonparametric model uncertainties. These issues are all considered in Part I of the text under the title of *Robust Control*.

One of the questions arising from the application of gain schedules is whether it is possible to reduce the number of controllers being used and still meet the system performance requirements. This is a serious practical question, and the authors find a solution in the methods of model reference adaptive control (MRAC) that form Part II of the textbook: *Robust Adaptive Control*. These chapters follow a sequence of developments, each one dealing with some practical aspect of MRAC, and illustrated by very appropriate case study examples. The techniques of Lyapunov stability theory are important tools in these developments and the whole of Chap. 8 is devoted to this topic. These tools are then used to provide a variety of performance guarantees for the adaptive control algorithms. Beginning from state feedback MRAC, the chapter sequence moves on, adding integral control, followed by inculcating robustness, and then improving the adaptation dynamics and culminating in the use of output feedback, which is the contribution of the last chapter (Chap. 14). All the chapters are supported by reference lists and sets of exercises for the reader.

Since it is not often that the textbook series contains a volume from writers who are based in industry, it is also fitting to say something about the authors in this Foreword.

Dr. Eugene Lavretsky is a Boeing Senior Technical Fellow at Boeing Research and Technology, Huntington Beach, California, USA. He has been responsible for the flight control systems of several advanced aircraft and has published numerous technical articles on control and aerospace topics. He is a senior member of the IEEE and was a recipient of the AIAA Mechanics and Control of Flight Award (2009), the IEEE Control System Magazine Outstanding Paper Award (2011), and the AACC Control Engineering Practice Award (2012), for his work in the aerospace field.

Dr. Kevin A. Wise is a Boeing Senior Technical Fellow at Boeing Phantom Works, St. Louis, Missouri, USA. He has been responsible for a wide range of aerospace developments including flight control systems, ejector seat systems, and autonomy in unmanned aerial vehicles. Dr. Wise is a fellow of the IEEE and has received both IFAC (2007) and AIAA (2004) awards for his work and publications. Both authors have taught the material of this textbook in graduate-level courses at US universities.

This new textbook is an excellent addition to the *Advanced Textbooks in Control and Signal Processing* series.

Industrial Control Centre
Glasgow, Scotland, U.K.
May 2012

M.J. Grimble
M.A. Johnson

Preface

After working in the aerospace industry for close to a quarter of a century, both of us felt strongly about writing this book, with the main goal to share our lessons learned and design insights into the development and analysis of robust and adaptive control systems. Our focus is on the systems that are practical yet have a formal basis for performing their design, analysis, and performance evaluations. During our professional careers at the Boeing Company, we have had a multitude of opportunities to design and flight test guidance, navigation, and control (GN&C) algorithms for a variety of platforms, ranging from commercial aircraft to fully autonomous experimental aerial vehicles. Over time and after numerous trade studies, we have collected a number of GN&C methods that have performed well on a variety of aircraft systems. So, we decided to write this book and share with the reader our experiences and lessons learned in the design, analysis, and evaluation of control technologies, with an emphasis on flight systems. The latter is not a prerequisite for understanding the book material, as these methods and design insights apply to all control systems. Aerospace applications and examples presented in this book are rather a motivation to challenges in constructing reliable and numerically efficient control algorithms.

Many parts of this book are based on undergraduate and graduate control courses that we have taught over the years at the Washington University in Saint Louis, the University of Missouri – Rolla, the Southern Illinois University in Edwardsville MO, and at the California Institute of Technology (Caltech). As such, the book material is quite suitable for senior undergraduate and graduate students, as well as for practicing engineers and research scientists, who have had an exposure to basic principles in controls and dynamics, such as an undergraduate level control course, covering classical methods (root locus, Bode diagrams, and Nyquist plots). In addition, we assume that the reader has a basic understanding of linear algebra, ordinary differential equations, and is familiar with using state space methods for analysis and numerical modeling of dynamical systems. These are the prerequisites.

Motivated and driven by aerospace applications, this book focuses on systems whose dynamics are continuous. Extensions of these methodologies to discrete and

hybrid systems are possible and can be found elsewhere in the vast literature devoted to control of dynamical systems.

Overall, this book is self-contained while covering theoretical development and practical applications of formal methods in robust and optimal linear control, robust stability analysis, Lyapunov stability theory, and model reference adaptive control (MRAC). Throughout this book, we present detailed simulation examples and case studies to illustrate key design steps and the benefits of applying robust and adaptive control methodologies to transport aircraft and experimental aerial platforms.

There are two major parts in this book. Part I presents robust control design and analysis methods for linear time-invariant systems. Part II focuses on MRAC methods for systems with nonlinear and uncertain dynamics.

Readers will benefit from the two-part distinct structure of this book. Such an arrangement enables a seamless transition from the classical linear control concepts to the state of the art in adaptive systems while illustrating each design with realistic aerospace applications. Also, the two-part book organization allows us to present self-contained material, covering linear and robust adaptive control techniques for dynamical systems that operate in the presence of uncertainties. Toward that end, we consistently give structured descriptions of both classical and advanced control techniques, key design procedures and guidelines, worked examples, and Matlab simulations. Each part ends with a set of educational and challenging exercises that are directly related to the material presented. All these features constitute the book's educational value.

Part I begins with an introduction to challenges in control design, analysis, and simulation of aerial vehicles. General aviation background and current trends that lead to the need for more advanced control are discussed. Also presented is a brief survey of control-theoretic methods for existing and future aerial vehicles. The theoretical portion of Part I starts with the introduction of robust and optimal linear control methods for linear systems. Command tracking using linear quadratic regulators (LQR) with integral action is presented. This part also covers two output feedback design methods, such as projective control and linear quadratic Gaussian control with Loop Transfer Recovery (LQG/LTR). These algorithms are employed to develop baseline control architectures for linear systems with known dynamics.

Part II begins with self-contained material on the design and analysis of adaptive state feedback controllers for linear and nonlinear uncertain dynamical systems in continuous-time domain. An overview of Lyapunov stability theory is given, followed by theoretical fundamentals for MRAC systems. Next, approximation properties of artificial neural networks and their applications to the design of direct adaptive systems are introduced, and several approximation-based MRAC methods are discussed. The part proceeds with the development of state feedback adaptive augmentation architectures for robust baseline linear controllers, followed by extensions and modifications to achieve transient performance in adaptive systems, as well as to accommodate output feedback constraints. In this part, we also present adaptive augmentation design methods to combine robust baseline controllers with adaptive feedback.

Throughout this book, we discuss motivations to the design, analysis, and implementation of robust and adaptive controllers, with the aim to addressing realistic challenges that often arise in the flight control of aerial vehicles and other systems.

Acknowledgments

First and foremost, we would like to thank our families for their love, understanding, and patience. Often, the book writing took us away from our family lives. Thank you for your infinite support in allowing us to complete such an undertaking.

We are indebted to professional colleagues and friends, whom we have had the privilege to closely interact during our professional and social activities, before and while writing this book. We would like to express our gratitude to Anuradha Annaswamy, Anthony Calise, Petros Ioannou, Irene Gregory, Jonathan How, Naira Hovakimyan, Wassim Haddad, Richard Murray, Dennis Bernstein, Gary Balas, Siva Banda, David Doman, Zac Dydek, Travis Gibson, Ross Gadient, Michael Steinhour, Joseph Brinker, Ken Buckholtz, and many others who have helped us better understand and appreciate the ever changing and challenging field of controls and dynamics.

We are grateful to Oliver Jackson and Michael Johnson for their editorial comments and suggestions toward improving readability, style, and clarity of the book material.

Finally, we would like to acknowledge The Boeing Company for giving us opportunities, associated challenges, and responsibilities in leading the development of flight systems for aircraft, helicopters, spacecraft, and other aerial platforms.

Eugene Lavretsky
Los Angeles, California

Kevin A. Wise
Saint Louis, Missouri

Contents

Part I
Robust Control

Chapter 1
Introduction

1.1 Why Robust and Adaptive Control?

Robust control can be thought of as an online policy capable of regulating systems (plants) whose dynamics may contain bounded (in some sense) uncertainties. Such an algorithm would often utilize feedback–feedforward state-output connections to generate appropriate control inputs so that the plant output moves along the prescribed "trajectories." The main idea here is to design a control system that would work satisfactory for a set of plants, whether linear or nonlinear, while assuming the worst-case conditions on the "unknown unknowns" in the system dynamics.

Discarding ad hoc designs, it would be safe to say that all formal and reliable control methods are model based. We often start with a mathematical model that resembles the process of interest in a selected domain of operation. The model may or may not be accurate in capturing significant and other effects in the process dynamics. In order to overcome potential modeling deficiencies, we seek a robust solution, designed based on the model, yet capable of controlling the real process, and not just the model. We would also want a controller whose performance "gracefully degrades" in the presence of uncertainties. The graceful degradation property is highly desirable, since it becomes the only assurance that the controller would not abruptly break down, if and when the system encounters slightly unprecedented events during its intended operation.

Embedding robustness properties into a control solution should be treated as one of the main criteria in any design. For example, achieving closed-loop stability and tracking performance, while providing adequate stability margins, are the main goals, especially when dealing with linear system approximations of real processes. In Part I, we will present various methods and techniques to achieve this goal.

Once a robust control solution is found, one may wonder if its robustness properties and applicability domains can be further extended to cover a wider class of uncertainties in the process dynamics. In Part II of this book, we will attempt to address this problem using methods from adaptive systems. We shall

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_1,

employ nonlinear design tools and show that indeed it is possible to construct adaptive controllers that would cope with unbounded state-dependent nonlinear uncertainties.

What is the difference between robust and adaptive controllers? A robust controller is designed to operate under the worst-case condition assumption. Such a controller may use excessive actions to regulate the process. In contrast, an adaptive controller would try to perform an online estimation of the process uncertainty and then produce a control input to anticipate, overcome, or minimize the undesirable deviations from the prescribed closed-loop plant behavior. In addition to their adaptive properties, these controllers can be constructed to "learn" or equivalently to remember. Learning refers to remembering/recognizing certain patterns and acting based on prior knowledge or "memory." A tracking error integrator in the feedback loop is a simple example of a learning controller. It accumulates and integrates regulation errors based on previous and current data. Adaptive controllers are nonlinear extensions of linear feedback integrators. In other words, adaptive loops form their output by integrating nonlinear functions of the system tracking errors.

We would like to emphasize that adaptive control is not the ultimate solution for all problems. This concept represents merely another formal method to design controllers for a wide class of process uncertainties and with performance guarantees.

Our professional experience comes from the design of robust and adaptive flight controllers for a variety of airborne platforms. Most of them are in operation today. Over the years, we have found that it is not robust versus adaptive but rather a combination of both controllers that works best, in the sense of maintaining closed-loop stability, enforcing robustness to uncertainties, and delivering the desired performance in the presence of unanticipated events. The (Robust + Adaptive) architecture combination is our "secret" control design recipe that we would like to share with the reader.

1.2 About This Book

The book is written to provide a self-contained introduction to linear robust control methods, followed by an exploration of adaptive systems. Part I is solely devoted to robust control methods for linear-time-invariant continuous systems. This part can be taught in a semester-long course to students who have had a basic introduction to control systems. Part II covers a series of topics in adaptive control in a progressive complexity, starting with the detailed introduction to model reference adaptive controllers for linear systems and ending with the adaptive output feedback control methods for a class of nonlinear uncertain dynamics. The mathematical prerequisites for this part consist of basic concepts in linear algebra and ordinary

differential equations. Prior to adaptive control, we give an introduction and an overview of the Lyapunov's stability theory, which becomes the essential tool for the development of all design and analysis methods in this part of the book. The contents here can be covered within a semester. It is also possible to condense the material for use in a 9- to 10-week graduate-level course.

The two parts of the book can be combined in a single course arrangement, whereby selected linear control methods from Part I are taught, followed by a subset of adaptive control techniques. For example, one may elect to start with linear optimal control and then discuss methods to combine robust and adaptive controllers into a single system capable of mitigating a wide range of uncertainties.

Finally, the book can serve as an ample reference for research scientists and control practitioners, who are interested in the development and application of robust linear and/or adaptive control methods to control a wide variety of systems.

1.3 Aircraft Flight Dynamics Equations of Motion

Inspired and motivated by aerospace applications, we would like to introduce the reader to modeling aerial vehicles, such as aircraft. In our opinion, these models give rise to many interesting and challenging control problems. Even if the reader is not all too familiar with aerospace-related applications, we believe that the material of this section would still be beneficial. It can serve as an example, revealing the intricacies and complex nature of modeling to support control synthesis for realistic systems and processes.

Toward that end, we begin with the rigid aircraft six-degrees-of-freedom (6-DoF) equations of motion [1–3]. These dynamics can be obtained based on Newton's second law of motion, written in the aircraft-fixed body axes coordinate system, as shown in Fig. 1.1.

The aircraft dynamics (treated as a rigid body) are comprised of three translational and three rotational degrees of freedom, resulting in the 6-DoF motion.

The translational motion is described by (1) the forward velocity u (positive along the fuselage-body x-axis), (2) the lateral velocity v (positive along the right-wing-body y-axis), and (3) the vertical velocity w (positive down and along the body z-axis).

The three rotational degrees of freedom are represented by (1) the body roll rate p (around the body x-axis), (2) the body pitch rate q (around the body y-axis), and (3) the body yaw rate r (around the body z-axis). Positive angular rates (p, q, r) result in the counterclockwise rotations around their respective axis (x, y, z).

With the body axes coordinate frame fixed at the aircraft center of gravity (CG), the 6-DoF aircraft equations of motion can be written as

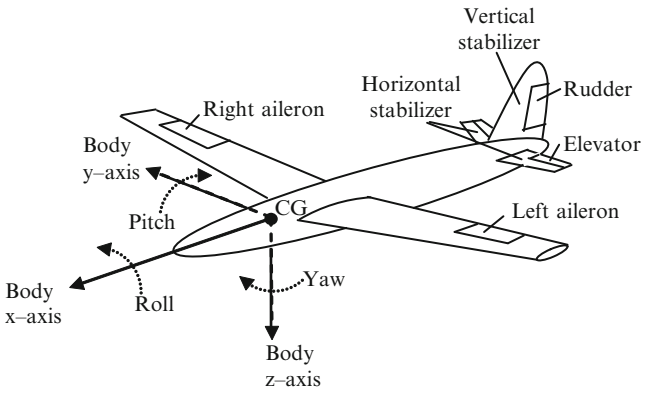


Fig. 1.1 Definition of rigid aircraft configuration, controls, axes, and degrees of freedom

$$\text{Translational DoF}: m\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} = -\left[\begin{pmatrix}p\\ q\\ r\end{pmatrix}\times\begin{pmatrix}u\\ v\\ w\end{pmatrix}\right]+\begin{pmatrix}F_x\\ F_y\\ F_z\end{pmatrix} + m\underbrace{\|\vec{g}\|}_{g}\underbrace{\begin{pmatrix}-\sin\theta\\ \cos\theta\ \sin\varphi\\ \cos\theta\ \cos\varphi\end{pmatrix}}_{\vec{g}}$$

$$\text{Rotational DoF}: J\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = -\left[\begin{pmatrix}p\\ q\\ r\end{pmatrix}\times J\begin{pmatrix}p\\ q\\ r\end{pmatrix}\right]+\begin{pmatrix}\bar{L}\\ M\\ N\end{pmatrix} \tag{1.1}$$

where m is the aircraft mass, $J \in R^{3\times 3}$ is the vehicle inertia matrix, and $(F_x,\ F_y,\ F_z)$ and $(\bar{L},\ M,\ N)$ are the body (x, y, z) – components of the forces and moments (due to aerodynamics and propulsion) acting on the aircraft. An expression with the square brackets $[a \times b]$ represents the cross-product of two vectors a and b. This expression is also known as the vector product. This is a vector whose magnitude is $\|a\|\|b\|\ \sin\theta$, where θ is the angle between a and b (positive counterclockwise from a to b). Its direction is perpendicular to the plane of a and b and is given by the right-hand rule.

In (1.1), $\vec{g}$ is the gravity vector, and $g = \|\vec{g}\|$ denotes its magnitude. The gravity vector is expressed in the aircraft-fixed body axes coordinates, in terms of the vehicle bank angle φ (positive – aircraft right wing down), the pitch angle θ (positive – aircraft nose up), and the true heading angle Ψ (positive – clockwise rotation of the aircraft nose from the true north direction) (Fig. 1.2).

The three Euler angles $(\varphi,\ \theta,\ \psi)$ give inertial angular orientation of the aircraft, if treated as a rigid body moving in the three-dimensional inertial space [1–3]. In other words, these angles describe the instantaneous orientation of the aircraft

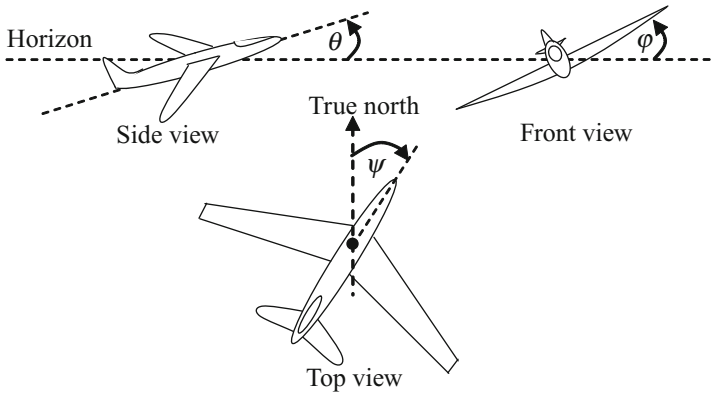


Fig. 1.2 Front, side, and top views of an aircraft in a bank, pitch, and yaw, respectively

body-fixed coordinate system with respect to the Earth-fixed (inertial) frame of reference. The following kinematic relations describe dynamics of the Euler angles versus the aircraft body angular rates $(p,\ q,\ r)$:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\varphi\ \tan\theta & \cos\varphi\ \tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \frac{\sin\varphi}{\cos\theta} & \frac{\cos\varphi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{1.2}$$

According to (1.1) and (1.2), the aircraft 6-DoF system state vector is

$$\vec{x} = (u,\ v,\ w,\ p,\ q,\ r,\ \varphi,\ \theta,\ \psi)^T$$

with $n_x = 9$ states in the system dynamics.

Shown in Fig. 1.1, the aircraft primary control inputs consist of (1) the left and right ailerons $\left(\delta_{left\,ail},\ \delta_{right\,ail}\right)$, (2) the horizontal stabilizer δ_h, (3) the elevator δ_e, (3) the vertical stabilizer δ_v, and (4) the rudder δ_r. Unless the aircraft is a glider, it would have yet another primary control input – the thrust force δ_{th}, which is created either by propellers or jet engines, mounted at specific locations on the vehicle airframe.

The horizontal stabilizer is a slow-movable surface, whose main purpose is to trim/equalize the aircraft longitudinal forces and moments while the elevator controls the pitching motion of the aircraft. The vertical stabilizer is a fixed surface designed to enforce lateral–directional stability, while the rudder controls the aircraft yawing motion.

For a conventional aircraft, the differential aileron $\delta_a = \delta_{left\,ail} - \delta_{right\,ail}$ is the primary roll control device, the elevator δ_e is for pitch control, the rudder δ_r controls the yaw motion, and the thrust force δ_{th} provides airspeed control. Disregarding the slow-moving horizontal stabilizer and the vertical rudder surfaces, the vector

$$\vec{u} = (\delta_{th},\ \delta_a,\ \delta_e,\ \delta_r)^T$$

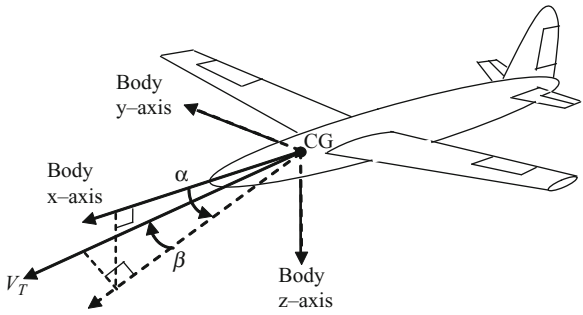


Fig. 1.3 Aircraft aero measurements: true airspeed, angle of attack, and angle of sideslip

with $n_u = 4$ defines the aircraft primary control inputs for airspeed, roll, pitch, and yaw, in that order. Through appropriate selection of individual controls, the aircraft dynamics can be modified to fly and maneuver the aircraft.

The system output signals can be defined based on the availability of physical measurement devices that are installed on the aircraft. For example, the body angular rates (p, q, r) are measured by rate gyroscopes that are usually located near the vehicle CG. Also, the same devices would be configured to provide the three Euler angles (φ, θ, ψ). In addition, every aircraft is typically equipped with at least three accelerometer devices that provide online measurements of longitudinal, lateral, and vertical loads, denoted by (A_x, A_y, A_z). Each device measures an acceleration component (in ft/s/s or g-s) at the point of installation and along its corresponding axis

$$A_x = \frac{F_x}{m\,g},\ A_y = \frac{F_y}{m\,g},\ A_z = \frac{F_z}{m\,g}$$

Furthermore, an aircraft measurement system would include three aero-data measurements. They are (1) the true airspeed V_T, (2) the angle-of-attack (AOA) α, and (3) the angle of sideslip (AOS) β. Their formal definitions (disregarding wind-gust components) and pictorial representations are given below (Fig. 1.3).

$$V_\mathrm{T} = \sqrt{u^2 + v^2 + w^2},\ \ \alpha = \arctan\left(\frac{w}{u}\right),\ \ \beta = \arcsin\left(\frac{v}{V_\mathrm{T}}\right)$$

Combining all these measurements gives the system output

$$\vec{y} = \left(A_x,\ A_y,\ A_z,\ V_\mathrm{T},\ \beta,\ \alpha,\ p,\ q,\ r,\ \varphi,\ \theta,\ \psi\right)$$

with $n_y = 12$ components.

The aerodynamic forces are often resolved into two perpendicular components, the lift and the drag, as shown in Fig. 1.4.

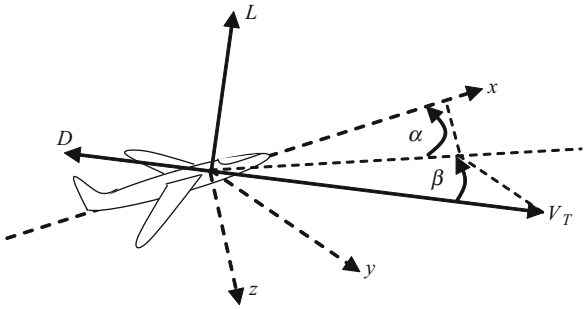


Fig. 1.4 Aerodynamic lift and drag forces

The aerodynamic lift force is perpendicular to the vehicle true airspeed vector V_T, while the drag force resists the vehicle motion along the airspeed direction. Both forces primarily depend on the angle-of-attack α, dynamical pressure $\bar{Q} = \frac{1}{2}\,\rho\, V_\mathrm{T}^2$, where ρ is the air density, altitude h, as well as on the aircraft control settings. If we now decompose the total forces into aerodynamic and propulsive components,

$$F_x = X_a + X_T, \quad F_y = Y_a + Y_T, \quad F_z = Z_a + Z_T \tag{1.3}$$

then the aerodynamic forces $(X_a,\ Y_a,\ Z_a)$can easily be written in terms of lift and drag:

$$\begin{aligned} X_\mathrm{a} &= L\,\sin\alpha - D\,\cos\beta\,\cos\alpha \\ Y_\mathrm{a} &= D\,\sin\beta \\ Z_\mathrm{a} &= -\,L\,\cos\alpha - D\,\cos\beta\,\sin\alpha \end{aligned} \tag{1.4}$$

It is not too difficult to rewrite the aircraft translational dynamics (1.1) in terms of lift, drag, true airspeed, and angle of attack. The corresponding equations can be found in any textbook on flight dynamics. We shall use these relations in deriving simplified models for control.

In general, the aircraft equations of motion (1.1) represent a continuous dynamical multi-input multi-output system in the form

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}) \\ \vec{y} &= h(\vec{x}, \vec{u}) \end{aligned} \tag{1.5}$$

with the state $\vec{x} \in R^9$, with the control input $\vec{u} \in R^4$, and with the output $\vec{y} \in R^{12}$.

Strictly speaking, another set of equations ought to be added to the aircraft dynamics. These are the three relations that connect the aircraft body-fixed velocities $(u,\ v,\ w)$ with the northeast-altitude inertial velocities $\left(\dot{x},\ \dot{y},\ \dot{h}\right)$. In essence, the inertial velocities are computed by transforming the body-fixed velocity vector $\begin{pmatrix} u & v & w \end{pmatrix}^T$ from the body-fixed into the Earth-fixed coordinates,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ -\dot{h} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

and of course, the aircraft inertial positions (x, y, h) are derived as the integrals of the corresponding velocities. The inertial speeds and positions are needed to design guidance algorithms for steering the vehicle along the prescribed trajectories. Also, these quantities become important during landing and takeoff phases of flight. The three inertial velocities and positions can be added to the system output $\vec{y}$. In that case, the aircraft dynamics would become 12-dimensional, with the extended state vector

$$\vec{X} = \left(\underbrace{u,\ v,\ w,\ p,\ q,\ r,\ \varphi,\ \theta,\ \psi}_{\vec{x}},\ x,\ y,\ h \right)^T$$

and with the redefined 18-dimensional system output

$$\vec{Y} = \left(\underbrace{A_x,\ A_y,\ A_z,\ V_T,\ \beta,\ \alpha,\ p,\ q,\ r,\ \varphi,\ \theta,\ \psi}_{\vec{y}},\ \dot{x},\ \dot{y},\ \dot{h},\ x,\ y,\ h \right)$$

It turns out that an attempt to use the fully coupled aircraft 6-DoF model (1.1) for control design would most likely result in an impractical control solution of unnecessary complexity and with an undesirable high sensitivity due to model parameters. This phenomenon immediately presents a modeling-for-control challenge: How detailed does a control-oriented model need to be so that the resulting control solution is simple, robust, effective, and works per design specifications, when applied to the real process or system? The answer to this question of course depends on the application of interest. In the next section, we will construct simplified flight dynamics models for control design purposes.

1.4 Simplified Flight Dynamics for Control Design

The aircraft 6-DoF motion (1.1) can be decomposed into a mean or a steady-state around an operating point (trim) and perturbation dynamics around the trim conditions. Such a decomposition allows one to reduce the overall nonlinear fully coupled 6-DoF aircraft dynamics into a tractable form, suitable for control design and analysis. The notion of "trimming an aircraft" refers to finding a balance, or equilibrium, between aerodynamic, propulsive, and gravitational forces and moments that are constantly acting on the vehicle. In flight, an aircraft is trimmed

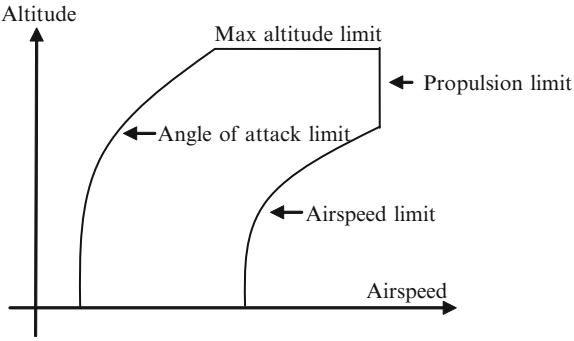


Fig. 1.5 Aircraft operational flight envelope, as a function of altitude and airspeed

by setting its primary controls to values that would result in the desired steady-state flight conditions. The trim function would be performed by a pilot or by an automatic flight control system.

In mathematical terms, we are looking for a system equilibrium pair $(\vec{x}_{eq},\ \vec{u}_{eq})$ in (1.5), such that the translational and angular accelerations are zeroed out,

$$\begin{aligned} \text{Translational DoF}: 0 &= -\left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix}\right] + \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} + m\,\vec{g} \\ \text{Rotational DoF}: 0 &= -\left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times J \begin{pmatrix} p \\ q \\ r \end{pmatrix}\right] + \begin{pmatrix} \bar{L} \\ M \\ N \end{pmatrix} \end{aligned} \tag{1.6}$$

or, equivalently $0 = f\left(\vec{x}_{\text{eq}},\ \vec{u}_{\text{eq}}\right)$. An aircraft would have many distinct equilibrium throughout the vehicle flight operational envelope (Fig. 1.5).

These trim points would depend first hand on altitude and airspeed. Based on available trim flight conditions, the main idea behind constructing control-oriented models and then performing flight control design consists of several distinct steps. These are the following:

1. Cover the flight envelope with a dense set of trim points.
2. Write simplified linear models around each of the trim point.
3. Use these dynamics to design fixed-point flight controllers per point.
4. Interpolate (i.e., gain schedule based on flight conditions) to combine linear controllers.

The result is a gain-scheduled flight control system that would be valid for the entire operational envelope. In what follows, we will concentrate on Step 2 and derive linear models (deviation dynamics from equilibrium) for a selected trim point.

When a conventional aircraft is trimmed wings-level, at selected flight conditions, the vehicle dynamics naturally decouples into longitudinal and lateral–directional modes. We are going to derive each of these separately.

1.4.1 Longitudinal Dynamics

The aircraft longitudinal dynamics describe changes in forward, vertical, and pitching motion of the vehicle. These dynamics can be further decomposed into fast and slow components or modes. The former is called the short period, and the latter is the phugoid. Typically, there would be a timescale separation between the two modes. The short period describes fast coupling between the aircraft angle of attack and the pitch rate. On the other hand, the phugoid represents a much slower (when compared to the short period) dynamical interchange between the vehicle altitude and the airspeed or, equivalently, between the aircraft potential and kinetic energy levels.

The short-period and the phugoid modes can be revealed after the aircraft model is linearized around a trim point (an equilibrium). For clarity of presentation, we assume that the thrust line is aligned with the vehicle x-axis. Then, the aircraft longitudinal equations of motion are

$$\begin{pmatrix} \dot{v}_{\mathrm{T}} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} X_{\mathrm{V}} & X_{\alpha} & 0 & -g\cos\gamma_0 \\ \frac{Z_{\mathrm{V}}}{V_0} & \frac{Z_{\alpha}}{V_0} & 1+\frac{Z_q}{V_0} & -\frac{g\sin\gamma_0}{V_0} \\ M_{\mathrm{V}} & M_{\alpha} & M_q & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_{\mathrm{T}} \\ \alpha \\ q \\ \theta \end{pmatrix} + \begin{pmatrix} X_{\delta_{\mathrm{th}}}\cos\alpha_0 & X_{\delta_{\mathrm{e}}} \\ -X_{\delta_{\mathrm{th}}}\sin\alpha_0 & \frac{Z_{\delta_{\mathrm{e}}}}{V_0} \\ M_{\delta_{\mathrm{th}}} & M_{\delta_{\mathrm{e}}} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_{\mathrm{th}} \\ \delta_{\mathrm{e}} \end{pmatrix} \tag{1.7}$$

where V_0 is the trimmed airspeed and α_0 is trimmed angle of attack, $\gamma_0 = \theta_0 - \alpha_0$ is the trimmed flight path angle (see Fig. 1.6), and θ_0 is the trimmed pitch angle. The model states $(v_{\mathrm{T}},\ \alpha,\ q,\ \theta)$ and the control inputs $(\delta_{\mathrm{th}},\ \delta_{\mathrm{e}})$ are incremental due to their trimmed values. Also, in (1.7), the matrix components represent constant (for fixed flight conditions) stability and control derivatives of the aircraft forces and moments, with respect to the longitudinal states and control inputs. When aircraft specific values of these derivatives are substituted into the model (1.7), most often the open-loop system eigenvalues will consist of a fast (short-period) and a slow (phugoid) pairs of complex conjugate numbers. Such a modal decomposition explains the timescale separation in the longitudinal dynamics of an aircraft, such as (1.7).

The short-period mode is defined by the dynamics of α and q. Extracting those from the model (1.7), yields

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_{\alpha}}{V_0} & 1+\frac{Z_q}{V_0} \\ M_{\alpha} & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_{\delta_{\mathrm{e}}}}{V_0} \\ M_{\delta_{\mathrm{e}}} \end{pmatrix} \delta_{\mathrm{e}} \tag{1.8}$$

These dynamics describe aircraft motion on a short interval of time, due to elevator input. Throughout the book, we shall utilize the short-period system quite often in our exploration of robust and adaptive control design and analysis methods.

The aircraft phugoid motion can be derived by setting $\dot{\alpha} = \dot{q} = 0$ in (1.7), solving for the corresponding "fast steady-state" values (α, q), and then substituting them into the remaining dynamics of v_T and θ.

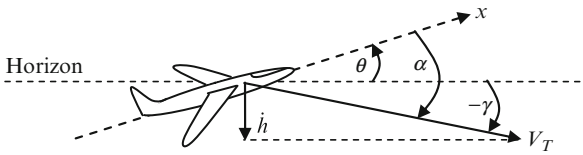


Fig. 1.6 Aircraft longitudinal motion and related angles

We leave the phugoid derivations to the reader and turn our attention back to the short-period dynamics (1.8). Let us now introduce the flight path angle γ. This is the angle between the aircraft airspeed vector and the horizon. For small angles, the following relationship connects the angle-of-attack α, the pitch angle θ, and flight path angle γ,

$$\alpha = \theta - \gamma \tag{1.9}$$

and it is depicted in Fig. 1.6 below.

Multiplying both sides of (1.9) by true airspeed V_T gives the vertical speed in inertial space:

$$\dot{h} = V_T \gamma = V_T (\theta - \alpha) \tag{1.10}$$

For small angles and assuming that the true airspeed is constant, we can differentiate (1.10) with respect to time, use α-dynamics from (1.8), and finally compute (approximately) the vertical acceleration a_z in body axes:

$$A_z \approx -\ddot{h} = -V \left(\dot{\theta} - \dot{\alpha}\right) = V (\dot{\alpha} - q) = Z_\alpha \alpha + Z_\delta \delta_e \tag{1.11}$$

In several of our upcoming design studies and examples, we shall treat this signal as an output of the aircraft longitudinal dynamics (1.8):

$$A_z = \begin{pmatrix} Z_\alpha & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + Z_\delta \delta_e \tag{1.12}$$

Sometimes, we choose to utilize A_z, rather than α, as the preferred state component. Differentiating (1.11) and solving for α in (1.12)

$$\alpha = \frac{A_z - Z_\delta \delta_e}{Z_\alpha} \tag{1.13}$$

gives

$$\begin{aligned} \dot{A}_z &= Z_\alpha \dot{\alpha} + Z_\delta \dot{\delta}_e = \frac{Z_\alpha}{V} \underbrace{(Z_\alpha \alpha + Z_\delta \delta_e)}_{a_z} + Z_\alpha q + Z_\delta \dot{\delta}_e \\ &= \frac{Z_\alpha}{V} A_z + Z_\alpha q + Z_\delta \dot{\delta}_e \end{aligned} \tag{1.14}$$

and then the pitch dynamics become

$$\dot{q} = M_\alpha\,\alpha + M_q\,q + M_\delta\,\delta_e = M_\alpha\left(\frac{A_z - Z_\delta\,\delta_\mathrm{e}}{Z_\alpha}\right) + M_q\,q + M_\delta\,\delta_e$$
$$= \frac{M_\alpha}{Z_\alpha}\,A_z + M_q\,q + \left(M_\delta - \frac{M_\alpha\,Z_\delta}{Z_\alpha}\right)\delta_e \tag{1.15}$$

Collecting (1.14) and (1.15), we can rewrite the short-period dynamics (1.8) in terms of the new state components $(A_z,\, q)$:

$$\begin{pmatrix}\dot{A}_z\\ \dot{q}\end{pmatrix} = \begin{pmatrix}\frac{Z_\alpha}{V} & Z_\alpha\\ \frac{M_\alpha}{Z_\alpha} & M_q\end{pmatrix}\begin{pmatrix}A_z\\ q\end{pmatrix} + \begin{pmatrix}0\\ M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\end{pmatrix}\delta_e + \begin{pmatrix}Z_\delta\\ 0\end{pmatrix}\dot{\delta}_e \tag{1.16}$$

We immediately note that presence of the control rate $\dot{\delta}_e$ in (1.16) will require the addition of an actuator model. The latter can be modeled by a second-order ordinary differential equation, with a specified natural frequency ω and a damping ratio ξ. The model is driven by the elevator command δ_e^{cmd}; its dynamics are

$$\ddot{\delta}_\mathrm{e} = -2\,\xi\,\omega\,\dot{\delta}_\mathrm{e} + \omega^2\left(\delta_e^{cmd} - \delta_e\right) \tag{1.17}$$

Combining (1.16) with (1.17), we arrive at the following four-dimensional system

$$\begin{pmatrix}\dot{A}_z\\ \dot{q}\\ \dot{\delta}_\mathrm{e}\\ \ddot{\delta}_\mathrm{e}\end{pmatrix} = \begin{pmatrix}\frac{Z_\alpha}{V} & Z_\alpha & 0 & Z_\delta\\ \frac{M_\alpha}{Z_\alpha} & M_q & M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -\omega^2 & -2\,\xi\,\omega\end{pmatrix}\begin{pmatrix}A_z\\ q\\ \delta_\mathrm{e}\\ \dot{\delta}_\mathrm{e}\end{pmatrix} + \begin{pmatrix}0\\ 0\\ 0\\ \omega^2\end{pmatrix}\delta_e^{cmd} \tag{1.18}$$

that describes the short-period dynamics, driven by an elevator command through actuation. Such a model is very helpful in flight control applications whereby angle-of-attack measurements are not available (or deemed unreliable). The trade-off here is that the model order has increased twice. In addition, the actuator position and the rate may not be available as measurements.

1.4.2 Lateral–Directional Dynamics

We begin with the kinematics of the Euler roll equation from (1.2):

$$\dot{\varphi} = p + \tan\,\theta\,(q\,\sin\,\varphi + r\,\cos\,\varphi) \tag{1.19}$$

Let θ_0 denote the trimmed pitch angle. Then, linear approximation of (1.19) around $\varphi_0 = p_0 = q_0 = r_0 = 0$ can be written as

$$\dot{\varphi} = p + r \tan \theta_0 \tag{1.20}$$

Stability axis roll and yaw rates $(p_s,\ r_s)$ are related to the body axis roll and yaw rates $(p,\ r)$ in the following way:

$$\begin{aligned} p_s &= p \cos \alpha + r \sin \alpha \\ r_s &= r \cos \alpha - p \sin \alpha \end{aligned} \tag{1.21}$$

Let α_0 denote the trimmed angle of attack (AOA). Then, a linear approximation of (1.21) is of the form

$$\begin{aligned} p_s &= p \cos \alpha_0 + r \sin \alpha_0 \\ r_s &= r \cos \alpha_0 - p \sin \alpha_0 \end{aligned} \tag{1.22}$$

Solving (1.22) for $(p,\ r)$ yields

$$\begin{aligned} p &= p_s \cos \alpha_0 - r_s \sin \alpha_0 \\ r &= r_s \cos \alpha_0 + p_s \sin \alpha_0 \end{aligned} \tag{1.23}$$

Substituting (1.23) into (1.20) results in

$$\begin{aligned} \dot{\varphi} &= p_s \cos \alpha_0 - r_s \sin \alpha_0 + (r_s \cos \alpha_0 + p_s \sin \alpha_0) \tan \theta_0 \\ &= (\cos \alpha_0 + \sin \alpha_0 \tan \theta_0) p_s + (\cos \alpha_0 \tan \theta_0 - \sin \alpha_0) r_s \end{aligned} \tag{1.24}$$

As we have previously noted, the following relation exists between the flight path angle, the pitch angle, and the angle of attack (at zero bank and sideslip angles):

$$\alpha_0 = \theta_0 - \gamma_0 \tag{1.25}$$

Substituting (1.25) into (1.24) gives

$$\begin{aligned} \dot{\varphi} &= \underbrace{(\cos \alpha_0 + \sin \alpha_0 \tan \theta_0)}_{\frac{\cos \gamma_0}{\cos \theta_0}} p_s + \underbrace{(\cos \alpha_0 \tan \theta_0 - \sin \alpha_0)}_{\frac{\sin \gamma_0}{\cos \theta_0}} r_s \\ &= \frac{\cos \gamma_0}{\cos \theta_0} p_s + \frac{\sin \gamma_0}{\cos \theta_0} r_s \end{aligned} \tag{1.26}$$

Assuming small angles, the angle of sideslip dynamics can be written as

$$\dot{\beta} = \frac{1}{V_0}\left(Y_\beta\,\beta + Y_p\,p_s + Y_r\,r_s + Y_{\delta_{ail}}\,\delta_{ail} + Y_{\delta_{rud}}\,\delta_{rud}\right) + \left(\frac{g\,\cos\,\theta_0}{V_0}\right)\varphi - r_s \quad (1.27)$$

where the right-hand side of the equation depends on the derivatives of the side-force Y, computed with respect to the lateral–directional states $(\beta,\ p_s,\ r_s,\ \varphi)$ and the control inputs $(\delta_{\text{ail}},\ \delta_{\text{rud}})$. Using (1.26) and (1.27), the aircraft lateral–directional linearized dynamics are

$$\begin{aligned}
\dot{\varphi} &= \frac{\cos\,\gamma_0}{\cos\,\theta_0}\,p_s + \frac{\sin\,\gamma_0}{\cos\,\theta_0}\,r_s \\
\dot{\beta} &= \frac{g\,\cos\,\theta_0}{V}\,\varphi + \frac{Y_\beta}{V}\,\beta + \frac{Y_p}{V}\,p_s + \left(\frac{Y_r}{V} - 1\right) r_s + \frac{Y_{\delta_{ail}}}{V}\,\delta_{ail} + \frac{Y_{\delta_{rud}}}{V}\,\delta_{rud} \\
\dot{p}_s &= L_\beta\,\beta + L_p\,p_s + L_r\,r_s + L_{\delta_{ail}}\,\delta_{ail} + L_{\delta_{rud}}\,\delta_{rud} \\
\dot{r}_s &= N_\beta\,\beta + N_p\,p_s + N_r\,r_s + N_{\delta_{ail}}\,\delta_{ail} + N_{\delta_{rud}}\,\delta_{rud}
\end{aligned} \quad (1.28)$$

We can easily rewrite (1.28) in matrix form:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{\cos\gamma_0}{\cos\theta_0} & \frac{\sin\gamma_0}{\cos\theta_0} \\ \frac{g\cos\theta_0}{V_0} & \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ 0 & L_\beta & L_p & L_r \\ 0 & N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \varphi \\ \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix} \begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix} \quad (1.29)$$

When the airspeed is sufficiently high, the gravity term in (1.29) becomes negligible: $\frac{g\,\cos\,\theta_0}{V_0} \approx 0$. In this case, the bank dynamics can be eliminated:

$$\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix} \begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix} \quad (1.30)$$

The resulting third-order lateral–directional linear model would be suitable for a control design where the goal is to regulate the vehicle roll and yaw rates, as well as the angle of sideslip.

1.4.3 Model Generalizations for Adaptive Control Design

The aircraft short-period dynamics (1.8), as well as the lateral–directional models (1.29) and (1.30), represent linear-time-invariant controllable systems:

$$\dot{x} = A\,x + B\,u \quad (1.31)$$

with the n-dimensional state x, the m-dimensional control u, the p-dimensional output

$$y = C\,x + D\,u \tag{1.32}$$

and with the matrices (A, B, C, D) of the corresponding dimensions. In Part I, we will discuss various methods to design robust linear controllers and to analyze their performance.

Then, in Part II, we will focus our attention on adaptive control techniques, with the goal of maintaining closed-loop stability and robustness in the presence of unexpected events. Specifically, we shall insert uncertainties into (1.31) and (1.32) and consider a class of dynamical systems in the form

$$\dot{x} = A\,x + B\,\Lambda\,(u + f(x)) \tag{1.33}$$

where the $(m \times m)$ matrix Λ models control actuation failures and the m-dimensional vector function $f(x)$ represents all other "unknown unknowns" in the system dynamics. The uncertain model (1.33) is our attempt to embed an extra realism into the "ideal" system (1.31). The uncertainties in (1.33) are called "matched," in the sense that they enter the system dynamics through control channels. So, as long as Λ is invertible, the system controllability property is not affected. It so happens that the matched uncertainty assumption implies existence of at least one control solution, capable of steering the system state along the desired trajectories.

We shall also consider regulation problems with non-matched but bounded uncertainties, such as time-dependent noise and environmental disturbances, represented by an n-dimensional uniformly bounded piecewise continuous vector function $\xi(t)$:

$$\dot{x} = A\,x + B\,\Lambda\,(u + f(x)) + \xi(t) \tag{1.34}$$

Again, we would like to point out that the assumed boundedness of $\xi(t)$ does not destroy the system controllability. So, the unwanted effects caused by bounded noise and disturbances can be mitigated through proper control synthesis. In Part II, we will explore robust and adaptive methods to control uncertain systems, such as (1.33) and (1.34).

Readers who are interested in adaptive control may find the matched uncertainty assumption to be quite restrictive. Some may even argue that there are many dynamical systems arising from realistic applications that do not satisfy the matching conditions. Be as it may, in aerospace applications, matched uncertainties are of primary concern, and that explains our interest in the control of uncertain systems such as (1.34). Finally, we would like to note that most of the adaptive control methods presented in this book can be extended to handle systems with non-matched uncertain dynamics, but these extensions are outside of the book scope.

1.5 Control-Oriented Models for Linear-Time-Invariant Systems

In this section, we present state space linear-time-invariant plant and controller models for control design and analysis that will be used in subsequent chapters. Models for closed-loop simulation and frequency domain analysis at both the plant input and plant output are also derived.

The plant model is

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u \end{aligned} \tag{1.35}$$

where $x \in R^{n_x}$ is the state, $u \in R^{n_u}$ the control, and $y \in R^{n_y}$ the output. The real matrices $\left(A_p, B_p, C_p, D_p\right)$ are of appropriate dimension and describe the dynamics of the plant.

The controller model is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{1.36}$$

where $x_c \in R^{n_{x_c}}$ is the controller state vector, u the control input from (1.35), y the output from (1.35), and $r \in R^{n_r}$ represents the external, possibly time-varying, command vector. The real matrices $\left(A_c, B_{c_1}, B_{c_2}, C_c, D_{c_1}, D_{c_2}\right)$ are of appropriate dimension and describe the controller dynamics (1.36).

In order to cast a commonly used static proportional state feedback controller such as

$$u = -K x \tag{1.37}$$

into the form of (1.36), we first choose $C_p = I_{n_x \times n_x}$, $D_p = 0_{n_x \times n_u}$ in (1.35). This gives $y = x$. Then, we define $C_c = -K$ in (1.36) and set the rest of the matrices in that equation to zero.

Next, we will connect the generic controller (1.36) to the plant model (1.35) and then derive state space models for the closed-loop system and the loop gain at the plant input and output break points. When building these models, one should simulate the closed-loop system to make sure the model is correctly connected with the minus signs inserted where appropriate (to represent negative feedback). After that, we would use the system loop gain models to compute the necessary frequency responses. Note that both the plant and the controller may have feedforward connections, with nonzero D matrices. The feedforward terms must be properly handled when forming the closed-loop system dynamics.

We will start with substituting the plant output equation into the control law:

$$\begin{aligned} u &= C_c x_c + D_{c1} y + D_{c2} r \\ u &= C_c x_c + D_{c1}\left(C_p x + D_p u\right) + D_{c2} r \\ \underbrace{\left(I - D_{c1} D_p\right)}_{Z} u &= C_c x_c + D_{c1} C_p x + D_{c2} r \\ u &= Z^{-1}\left(C_c x_c + D_{c1} C_p x + D_{c2} r\right) \end{aligned} \tag{1.38}$$

We must assume that matrix Z in (1.38) is invertible. This makes the overall problem formulation well-posed. Substituting (1.38) into the plant model (1.35) yields

$$\begin{aligned} \dot{x} &= A_p x + B_p Z^{-1}\left(C_c x_c + D_{c1} C_p x + D_{c2} r\right) \\ \dot{x} &= \left(A_p + B_p Z^{-1} D_{c1} C_p\right) x + B_p Z^{-1} C_c x_c + B_p Z^{-1} D_{c2} r \end{aligned} \tag{1.39}$$

We can also substitute the system output y into the controller:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1}\left(C_p x + D_p u\right) + B_{c2} r \\ \dot{x}_c &= A_c x_c + B_{c1}\left(C_p x + D_p Z^{-1}\left(C_c x_c + D_{c1} C_p x + D_{c2} r\right)\right) + B_{c2} r \\ \dot{x}_c &= \left(A_c + B_{c1} D_p Z^{-1} C_c\right) x_c + B_{c1}\left(I + D_p Z^{-1} D_{c1}\right) C_p x + \left(B_{c2} + B_{c1} D_p Z^{-1} D_{c2}\right) r \end{aligned} \tag{1.40}$$

Let us define an augmented state vector in the form

$$x_a = \begin{bmatrix} x^T & x_c^T \end{bmatrix}^T \tag{1.41}$$

Then, the closed-loop system dynamics can be written as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} &= \underbrace{\begin{bmatrix} A_p + B_p Z^{-1} D_{c1} C_p & B_p Z^{-1} C_c \\ B_{c1}\left(I + D_p Z^{-1} D_{c1}\right) C_p & A_c + B_{c1} D_p Z^{-1} C_c \end{bmatrix}}_{A_{cl}} \begin{bmatrix} x \\ x_c \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} B_p Z^{-1} D_{c2} \\ B_{c2} + B_{c1} D_p Z^{-1} D_{c2} \end{bmatrix}}_{B_{cl}} r \end{aligned} \tag{1.42}$$

or equivalently

$$\dot{x}_a = A_{cl}\, x_a + B_{cl}\, r \tag{1.43}$$

This is the closed-loop model. Its output can be easily defined as follows:

$$\begin{aligned} y &= C_p x + D_p u \\ &= C_p x + D_p Z^{-1}\left(C_c x_c + D_{c1} C_p x + D_{c2} r\right) \\ &= \left[C_p + D_p Z^{-1} D_{c1} C_p \;\; D_p Z^{-1} C_c\right] \begin{bmatrix} x \\ x_c \end{bmatrix} + \left[D_p Z^{-1} D_{c2}\right] r \\ &\underbrace{\left[\left(I + D_p Z^{-1} D_{c1}\right) C_p \;\; D_p Z^{-1} C_c\right]}_{C_{cl}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\left[D_p Z^{-1} D_{c2}\right]}_{D_{cl}} r \end{aligned} \tag{1.44}$$

and so the closed-loop system output becomes

$$y = C_{cl}\, x_a + D_{cl}\, r \tag{1.45}$$

Equations (1.43) and (1.45) give the state space model $(A_{cl}, B_{cl}, C_{cl}, D_{cl})$ for the closed-loop system.

The loop gain model at the plant input is formed to support frequency domain analysis of the design at the plant input loop break point. In this model, we treat the control input to the plant as the model input u_{in}. The control output from the controller becomes the model output u_{out}. Also, we neglect the command vector r. In this case, the plant and controller models are

$$\begin{aligned} \dot{x} &= A_p x + B_p u_{\text{in}} \\ y &= C_p x + D_p u_{\text{in}} \end{aligned} \tag{1.46}$$

and

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y \\ u_{\text{out}} &= C_c x_c + D_{c1} y \end{aligned} \tag{1.47}$$

We can connect these two systems with u_{in} as the input and u_{out} as the output

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1}\left(C_p x + D_p u_{\text{in}}\right) = A_c x_c + B_{c1} C_p x + B_{c1} D_p u_{in} \\ u_{\text{out}} &= C_c x_c + D_{c1}\left(C_p x + D_p u_{\text{in}}\right) + D_{c2} r = C_c x_c + D_{c1} C_p x + D_{c1} D_p u_{in} \end{aligned} \tag{1.48}$$

and rewrite these relations in matrix form.

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} &= \underbrace{\begin{bmatrix} A_p & 0 \\ B_{c1} C_p & A_c \end{bmatrix}}_{A_{Li}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\begin{bmatrix} B_p \\ B_{c1} D_p \end{bmatrix}}_{B_{Li}} u_{in} \\ u_{out} &= \underbrace{\begin{bmatrix} D_{c1} C_p & C_c \end{bmatrix}}_{C_{Li}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\left[D_{c1} D_p\right]}_{D_{Li}} u_{\text{in}} \end{aligned} \tag{1.49}$$

The system loop gain at the plant input is

$$L_i(s) = C_{Li}(sI - A_{Li})^{-1}B_{\mathrm{Li}} + D_{\mathrm{Li}} \tag{1.50}$$

and it is defined by the matrix quadruple $(A_{Li}, B_{Li}, C_{Li}, D_{Li})$.

Similarly, the loop gain model at the plant output is formed to support frequency domain analysis of the design at the plant output loop break point. In this model, we treat the plant output feeding the controller as the model input, y_{in}, the plant output from the plant as the model output, y_{out}, and neglect the command vector r. The plant and controller models are

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y_{\mathrm{out}} &= C_p x + D_p u \end{aligned} \tag{1.51}$$

and

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y_{in} \\ u &= C_c x_c + D_{c1} y_{in} \end{aligned} \tag{1.52}$$

Connecting these two systems with y_{in} as the input and y_{out} as the output yields

$$\begin{aligned} &\dot{x} = A_p x + B_p(C_c x_c + D_{c1} y_{\mathrm{in}}) = A_p x + B_p C_c x_c + B_p D_{c1} y_{\mathrm{in}} \\ &y_{\mathrm{out}} = C_p x + D_p(C_c x_c + D_{c1} y_{\mathrm{in}}) = C_p x + D_p C_c x_c + D_p D_{c1} y_{\mathrm{in}} \\ &\Downarrow \\ &\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \underbrace{\begin{bmatrix} A_p & B_p C_c \\ 0_p & A_c \end{bmatrix}}_{A_{\mathrm{Lo}}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\begin{bmatrix} B_p D_{c1} \\ B_{c1} \end{bmatrix}}_{B_{\mathrm{Lo}}} y_{\mathrm{in}} \\ &y_{\mathrm{out}} = \underbrace{\begin{bmatrix} C_p & D_p C_c \end{bmatrix}}_{C_{\mathrm{Lo}}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\begin{bmatrix} D_p D_{c1} \end{bmatrix}}_{D_{\mathrm{Lo}}} y_{\mathrm{in}} \end{aligned} \tag{1.53}$$

So, the loop gain at the plant output is defined by $(A_{Lo}, B_{Lo}, C_{Lo}, D_{Lo})$:

$$L_{\mathrm{o}}(s) = C_{Lo}(sI - A_{Lo})^{-1}B_{\mathrm{Lo}} + D_{\mathrm{Lo}} \tag{1.54}$$

The derived loop gains, (1.50) and (1.54), become essential tools to analyze relative stability properties of closed-loop linear systems in frequency domain.

1.6 Norms of Vectors and Matrices in Euclidean Spaces

This chapter presents a brief overview of norms for vectors and matrices. We shall use these concepts very often throughout the book.

The n-dimensional Euclidean space R^n is a collection of all n-dimensional vectors $x = (x_1 \quad \ldots \quad x_n)^T$, whose components x_i are real numbers, where the upper-script "T" denotes the transposition operator, which turns a row vector into a column vector and vice versa. If $n = 1$, we get the one-dimensional Euclidean space of real numbers $R = R^1$.

The set of all $(n \times m)$-real matrices, with n rows and m columns, defines the $(n \times m)$-dimensional Euclidean space $R^{n \times m}$. Elements of a Euclidean space can be added, subtracted, and multiplied by a scalar.

The inner product of two vectors x and y from R^n equals the sum of products of their corresponding components: $x^T y = \sum_{i=1}^{n} x_i y_i$. The product of two matrices $A = R^{n \times m}$ and $B = R^{m \times p}$ is the matrix $C = R^{n \times p}$, whose $(i, j)^{\text{th}}$ element is the inner product of the i^{th} row of A and j^{th} column of B.

For a vector $x \in R^n$, its length (or magnitude) is given by the norm $\|x\|$ – a real-valued function from R^n to R, with the following properties:

1. For any $x \in R^n$, $\|x\| \geq 0$.
2. $\|x\| = 0$ if and only if x is the zero vector in R^n.
3. For any two vectors x and y from R^n, the triangular inequality holds $\|x + y\| \leq \|x\| + \|y\|$.
4. For any real constant $\lambda \in R$ and any vector $x \in R^n$, $\|\lambda x\| = |\lambda| \ \|x\|$.

In the forthcoming design and analysis of adaptive controllers, we will encounter the class of vector p-norms:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty \tag{1.55}$$

For notational sake, we would often drop the lower-script "p" and write $\|x\|$.

Given a vector p-norm $\|x\|$, the induced matrix norm

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\| \tag{1.56}$$

clearly depends on the selected vector p-norm.

For a matrix $A = \left[a_{ij}\right] \in R^{n \times m}$, the Frobenius norm is defined by

$$\|A\|_{\text{F}} = \sqrt{\text{tr}\left(A^T A\right)} = \sqrt{\sum_{i,j} a_{ij}^2} \tag{1.57}$$

with tr() denoting the trace of a matrix, which is equal to the sum of the matrix diagonal elements.

The following statements are well-known [9] and are listed here without proof:

- For the vector 1-norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$, the corresponding induced matrix norm is equal to the maximum absolute column sum, that is, $\|A\|_1 = \max_{1 \le j \le m} \left| \sum_{i=1}^{n} a_{ij} \right|$.
- For the vector 2-norm $\|x\|_2 = \sum_{i=1}^{n} x_i^2$, the induced matrix norm is equal to the maximum singular value of A, that is, $\|A\|_2 = \sigma_{\max}(A)$.
- For the vector ∞-norm $\|x\|_\infty = \max_{1 \le i \le n} |x_i|$, the induced matrix norm is equal to the maximum absolute row sum, that is, $\|A\|_\infty = \max_{1 \le i \le n} \left| \sum_{j=1}^{m} a_{ij} \right|$.
- The induced matrix norm satisfies $\|A\,x\|_p \le \|A\|_p \, \|x\|_p$, and for any two compatibly dimensioned matrices, A and B, one also has $\|A\,B\|_p \le \|A\|_p \, \|B\|_p$.
- The Frobenius norm *is not* an induced norm of any vector norm, but it is compatible with the 2-norm in the sense that $\|A\,x\|_2 \le \|A\|_{\mathrm{F}} \, \|x\|_2$.
- For any two compatibly dimensioned matrices A and B, the Frobenius inner product is defined as $\langle A, B \rangle_{\mathrm{F}} = \mathrm{trace}\big(A^T B\big)$.
- According to the Schwartz inequality,

$$\left|\mathrm{trace}\big(A^T B\big)\right| = \left|\langle A, B \rangle_{\mathrm{F}}\right| \le \|A\|_{\mathrm{F}} \, \|B\|_{\mathrm{F}} \tag{1.58}$$

- For any two co-dimensional vectors a and b, the trace identity relation is

$$a^T b = \mathrm{tr}\big(b\,a^T\big) \tag{1.59}$$

1.7 Summary

Robust and adaptive control of continuous dynamical systems is the focus of this book. We have presented a concise self-contained introduction into the underlying theory and methods while emphasizing how to design and analyze practical control systems for multi-input multi-output systems with nonlinear and uncertain dynamics. Our true inspiration comes from aerospace applications. During our professional careers, we have been fortunate to have had the opportunity to design control systems for various types of aerial platforms, most of which were tested in flight and others went into production. In this chapter, we have added examples of flight dynamics models to later demonstrate a variety of robust and adaptive control technologies. We hope that readers would find these dynamics interesting and ever challenging.

1.8 Exercises

Exercise 1.1. For the aircraft longitudinal dynamics (1.7), compute (analytically) open-loop system eigenvalues and eigenvectors. Determine the short-period and the phugoid modes of the system. Find sufficient conditions for a timescale separation between the two longitudinal modes. Compute the short-period eigenvalues using the open-loop approximation of the short-period dynamics (1.8). Compare the original short-period modes versus their approximations.

Exercise 1.2. Consider the longitudinal dynamics

$$\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 \\ -0.001 & -0.632 & 1 & 0 \\ 0 & -0.759 & -0.518 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} V \\ \alpha \\ q \\ \theta \end{pmatrix} + \begin{pmatrix} 10.1 & 0 \\ 0 & -0.0086 \\ 0.025 & -0.011 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_{\text{th}} \\ \delta_{\text{e}} \end{pmatrix}$$

representative of a transport aircraft, trimmed at $V_0 = 250$ ft/s, and flying at a low altitude. In the model, all angles and angular rates are in radians, airspeed is in ft/s, throttle is in lbs, and elevator deflections are in radians. Compute open-loop system eigenvalues. Extract the short-period dynamics. Compute and compare the approximated short-period modes to the original ones. Also, compare the numerically computed modes to the analytical predictions from Exercise 1.1. Simulate open-loop system responses due to elevator and thrust step inputs. Identify (numerically) a timescale separation between the short period and phugoid. Introduce vertical acceleration A_z, as defined in (1.12), and simulate its response due to a negative step input in the elevator (trailing edge up). Observe the initial tendency of A_z. When the elevator is deflected trailing edge up to pitch the vehicle nose up, there is a small instant decrease in the vertical acceleration. Then, A_z starts to increase, resulting in the aircraft pitch-up motion. This transient is caused by the elevator deflecting upward and creating a small negative lift increment. As a result, the vertical acceleration momentarily goes into the "wrong" direction before it reverses and builds up. These dynamics can also be explained by the fact that there is a nonminimum phase zero (with a positive real part) in the transfer function from δ_{e} to A_z. It is important to understand that all tail-driven aerial vehicles have similar characteristics. This phenomenon becomes very important during control design.

Exercise 1.3. For the aircraft lateral–directional dynamics (1.29), compute (analytically) the open-loop system eigenvalues and eigenvectors. Also, compute the modal characteristics for the simplified dynamics (1.30). Compare the original data versus their approximations.

Exercise 1.4. The lateral–directional dynamics of a passenger aircraft, in a cruise configuration, are given below:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.0487 & -0.0829 & 0 & -1 \\ 0 & -4.546 & -1.699 & 0.1717 \\ 0 & 3.382 & -0.0654 & -0.0893 \end{pmatrix} \begin{pmatrix} \varphi \\ \beta \\ p \\ r \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0.0116 \\ 27.276 & 0.5758 \\ 0.3952 & -1.362 \end{pmatrix} \begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}$$

where the roll and sideslip angles are in radians, the angular rates are in rad/s, and the aileron and rudder deflections are in radians. Compute open-loop system eigenvalues and compare the data to the analytical predictions from Exercise 1.3. Simulate open-loop system response due to aileron and rudder step inputs. Observe the roll rate response due to aileron and the coupling between the roll and yaw rates (called the "Dutch roll" mode). These dynamics are fast when compared to the much slower changes in the roll angle (called the "roll subsidence" mode). Similar to short period, the roll rate and the Dutch roll modes are the main quantities for stabilization and regulation. This task is often accomplished during the so-called inner-loop control design phase, where the angular rates are stabilized via feedback connections, driving the aileron and the rudder. For the inner-loop design, the bank dynamics are ignored, and the three-dimensional lateral–directional model (1.30) is utilized. Extract these dynamics from the model data and simulate responses of the simplified model due to the same step inputs in aileron and rudder. Compare and discuss simulation results.

References

1. Etkin, B.: Dynamics of Flight. Stability and Control, 2nd edn. Wiley, New York (1982)
2. McRuer, D., Ashkenas, I., Graham, D.: Aircraft Dynamics and Automatic Control. Princeton University Press, Princeton (1990)
3. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation. Wiley, New York (1992)

Chapter 2
Optimal Control and the Linear Quadratic Regulator

2.1 Introduction

Control systems must provide stability and performance in the presence of model uncertainty and neglected dynamics. This has proven to be a significant challenge, and as our understanding of dynamics and control has improved, aerospace has been able to develop new aircraft designs that are faster, have greater performance, and perform robustly in very large flight envelopes. These advancements built upon the foundation created by classical methods but were powered by computer-aided design tools which greatly expanded the engineer's ability to solve larger, more complex problems using advanced techniques.

In general, designing flight control systems using conventional (classical) analytical methods involves iterative single-loop design analyses that are costly in time and manpower. These systems were often designed by discretizing the flight envelope at specific points, designing the control system at these points, and guaranteeing robustness to parameter variations by designing large single-loop stability margins and evaluating the design through simulation. These methods worked well on aircraft that were open-loop stable, but as new designs emerged that were open-loop unstable in multiple axes, multi-input multi-output (MIMO) design methods were needed.

In the 1970s and 1980s, the question of robust stability and performance was raised, and new control system design and analysis methods emerged, called modern control. These advancements provided the theoretical mathematics required for optimizing the controller design for MIMO systems, with evaluation of stability and robustness to parameter uncertainties. Using methods for characterizing model uncertainties, controller robustness properties were evaluated, and iterative design tools emerged to achieve robust stability and performance. These modern methods allowed the control system designer to understand and directly address stability and robustness concerns for open-loop unstable MIMO systems. With computer-aided design tools, engineers could readily pose and solve "optimal control" problems for complex systems and implement the control across a large flight envelope using gain scheduling.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_2,

Optimal control problems arise in designing a control in order to minimize a performance index. There are many classes of problems for nonlinear or linear systems, dealing with time-variant or time-invariant dynamics, over a fixed time interval or infinite time and with different types of performance indexes. Optimal control problems are in general very difficult to solve, except for linear systems with a quadratic performance index. These problems are well understood and produce control laws that have very interesting properties, such as excellent gain and phase margins.

One of the key challenges in using optimal control theory is transforming frequency domain performance and stability requirements from classical control into time domain requirements. A multivariable optimal controller design using a quadratic performance index optimizes the design in the time domain. Satisfying frequency domain requirements, such as bandwidth, noise sensitivity, etc., using the optimization performance index, is a challenge. Similarly, quantifying the degree of robustness required to overcome parameter uncertainties is not well-posed in the problem setup.

The key to using optimal control theory is to develop a method to tune the design parameters to achieve the desired performance and stability in the control system (and robustness). This is the goal for this chapter and the next. This chapter introduces optimal control theory, the linear quadratic regulator, and the all important matrix Riccati equation. We will discuss in detail some of the excellent properties that optimal controllers produce, which makes them a favorite in many aerospace control problems. Chapter 3 takes the optimal control principles and the regulator framework and extends them to command following design problems. It is this command following challenge that is most common in aerospace flight control systems.

2.2 Optimal Control and the Hamilton–Jacobi–Bellman Equation

The derivation of the Hamilton–Jacobi–Bellman (HJB) partial differential equation for optimal control problems will allow us to understand how optimal control regulator problems are posed and how we can form an optimal control from a performance index minimization problem. Optimal control problems are in general very difficult to solve. There are many books available on the subject. Athans and Falb [1], Kwakernaak and Sivan [2], and Anderson and Moore [3] are three excellent textbooks that deal with necessary and sufficiency conditions, differentiability and continuity assumptions, problem setup, derivations, and solutions for most optimal control problems that can be solved analytically. We will begin by deriving the HJB partial differential equation in a general setting and will then focus on linear systems with quadratic performance indices. We shall only provide a generic framework for derivation of optimal control policies. Readers interested in details are referred to now-classical control textbooks, such as [1–3].

2.2.1 The Hamilton–Jacobi–Bellman Equation

We begin by considering a dynamic nonlinear system in the form

$$\dot{x} = f(x, u, t), \quad x(t_0) = x_0 \tag{2.1}$$

where $x \in R^{n_x}$ is the system state and $u \in R^{n_u}$ is the control input. The system starts at time t_0, with the initial state x_0. We suppose that $f(x, u, t)$ is continuously differentiable in all arguments. This assumption is sufficient for the initial value problem (2.1) to have the unique solution on a finite time interval [4]. We also assume that T is small enough to reside within the time interval, where the system solutions are defined. We are interested in "optimally" controlling the system dynamics, starting from x_0 and driving the system state to a designated location. The notion of optimality is defined through the integral cost performance index

$$J = \int_{t_0}^{T} L(x(\tau), u(\tau), \tau) d\tau + S(x(T)) \tag{2.2}$$

evaluated along the system trajectories $x(t)$ due to applied control input $u(t)$. The instant cost $L(x, u, \tau)$ and the terminal cost $S(x(T))$ are defined as scalar nonnegative functions of their arguments. Essentially, the cost J is our cumulative measure of the overall efforts (controls) and the state-energy spent to steer the system from its initial state x_0 to a neighborhood of the terminal manifold $S(x(T)) = 0$.

For example, if the system dynamics are scalar, then we can utilize quadratic instant and terminal costs, $L(x, u, \tau) = x^2(\tau) + q\, u^2(\tau)$ and $S(x(T)) = w\, x^2(T)$, with positive weights $(q,\ w)$. In this case, $L(x, u, \tau)$ can be thought of as the instant kinetic energy of the system, while the terminal cost $S(x(T))$ measures how close we can drive the system state to the origin in T seconds or less. So, by appropriately choosing the weights $(q,\ w)$, we can emphasize the importance of minimizing the kinetic energy spent, while regulating the system state to zero. Later on, we shall address optimal control problems with quadratic cost.

Given the system dynamics (2.1), the control challenge of interest is to find an optimal control policy u^* to minimize the cost index J over the time interval $[t_0, T]$. When used in (2.1), the optimal control u^* produces the optimal state trajectory x^* over $[t_0, T]$. Clearly, the cost index J in (2.2) depends on the system initial state $x(t_0)$, the control policy $u(\bullet) = u(t)|_{t_0 \leqslant t \leqslant T}$, and on the initial time t_0:

$$J = J(x(t_0),\ u(\bullet),\ t_0) \tag{2.3}$$

Let J^* denote the optimal (minimum) cost, when using the optimal control policy u^*:

$$J^*(x_0,\ t_0) = \int_{t_0}^{T} L(x^*(\tau), u^*(\tau), \tau)d\tau + S(x^*(T)) = \min_{u_{[t_0,T]}} J(x_0,\ u,\ t_0) \tag{2.4}$$

We see that the optimal performance index J^* is also a function of the control $u_{[t_0,T]}$, the initial state, and time:

$$J^*(x_0, t_0) = \min_{u_{[t_0,T]}} J(x(t_0),\ u(\bullet),\ t_0) = \min_{u_{[t_0,T]}} \left[\int_{t_0}^{T} L(x, u, \tau)d\tau + S(x(T))\right] \tag{2.5}$$

Suppose that we start the system at an arbitrary initial condition x and at time t. Then, the optimal cost to go from x is

$$J^*(x, t) = \min_{u_{[t,T]}} \left[\int_{t}^{T} L(x, u, \tau)d\tau + S(x(T))\right] \tag{2.6}$$

We can break (2.6) into two integrals, from $[t, t_1]$ to $[t_1, T]$,

$$J^*(x, t) = \min_{u_{[t,T]}} \left[\int_{t}^{t_1} L(x, u, \tau)d\tau + \int_{t_1}^{T} L(x, u, \tau)d\tau + S(x(T))\right] \tag{2.7}$$

and then explicitly write the minimization operation over the two intervals:

$$J^*(x, t) = \min_{u_{[t,t_1]}} \min_{u_{[t_1,T]}} \left[\int_{t}^{t_1} L(x, u, \tau)d\tau + \int_{t_1}^{T} L(x, u, \tau)d\tau + S(x(T))\right] \tag{2.8}$$

The main idea here is to divide the integral into time slices and then at each slice choose the optimal control that minimizes the overall cost J. This argument leads the Principle of Optimality developed by Richard Ernest Bellman in the late 1950s. Here is the original formulation of the principle, as it appears in [5]:

Principle of Optimality. An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

The Principle of Optimality tells us that the optimal cost to go from x at time t to a terminal state $x(T)$ can be computed by minimizing the sum of (a) the cost to go from $x = x(t)$ to $x_1 = x(t_1)$ and (b) the optimal cost from x_1 onward. So formally speaking, we can move the min operation inside the brackets:

$$J^*(x,t) = \min_{u_{[t,t_1]}} \left[\int_t^{t_1} L(x,u,\tau)d\tau + \underbrace{\min_{u_{[t_1,T]}} \int_{t_1}^{T} L(x,u,\tau)d\tau + S(x(T))}_{J^*(x_1,t_1)} \right] \tag{2.9}$$

Now, we can see that inside the brackets, the second integral is the optimal cost to go from x_1 to $x(T)$:

$$J^*(x,t) = \min_{u_{[t,t_1]}} \left[\int_t^{t_1} L(x,u,\tau)d\tau + J^*(x_1,t_1) \right] \tag{2.10}$$

We define $t_1 = t + \Delta t$ and substitute it into (2.10):

$$J^*(x,\ t) = \min_{u_{[t,t+\Delta t]}} \left[\int_t^{t+\Delta t} L(x,u,\tau)d\tau + J^*(x(t+\Delta t), t+\Delta t) \right] \tag{2.11}$$

Assuming that all functions are smooth, we can expand the right-hand side of (2.11) in a Taylor series

$$J^*(x(t),t) = \min_{u_{[t,t+\Delta t]}} \left[L(x,u,\tau)\Delta t + J^*(x,t) + \left(\frac{\partial J^*(x,t)}{\partial x}\right)^T \Delta x + \frac{\partial J^*(x,t)}{\partial t}\Delta t + \underbrace{\mathrm{O}(\Delta t^2)}_{\text{H.O.T.}} \right] \tag{2.12}$$

where $\mathrm{O}(\Delta t^2)$ denotes high-order terms (H.O.T.) in the Taylor expansion. Here, we define

$$\frac{\partial J^*}{\partial x}(x,t) = \left[\frac{\partial J^*}{\partial x_1} \quad \cdots \quad \frac{\partial J^*}{\partial x_{n_x}} \right] \in R^{1\times n_x} \tag{2.13}$$

as a row vector. We will denote the transpose as $\nabla_x J^*(x,t)$ a column vector. We can cancel $J^*(x,t)$ on each side, since it does not depend on $u[t, t+\Delta t]$, divide both sides by Δt, and get

$$0 = \min_{u_{[t,\ t+\Delta t]}} \left[L(x,u,\tau) + \frac{\partial J^*(x,t)}{\partial x}\ \frac{\Delta x}{\Delta t} + \frac{\partial J^*(x,t)}{\partial t} + \mathrm{O}(\Delta t) \right] \tag{2.14}$$

Letting $\Delta t \to 0$ gives

$$-\frac{\partial J^*(x,t)}{\partial t} = \min_u \left[L(x(t),u,\tau(t)) + \frac{\partial J^*(x,t)}{\partial x} \underbrace{\dot{x}(t)}_{f(x,u,t)} \right] \tag{2.15}$$

where the system state is defined by (2.1). We now introduce the Hamiltonian

$$H(x, \nabla_x J^*(x,t), u, t) = L(x,u,\tau) + \frac{\partial J^*(x,\ t)}{\partial x} f(x,u,t). \tag{2.16}$$

and rewrite (2.15) as

$$-\frac{\partial J^*(x,t)}{\partial t} = \min_u H(x, \nabla_x J^*(x,t), u, t) \tag{2.17}$$

Due to the assumed smoothness of all the functions, it follows that to minimize H with respect to the control u, we can compute the function gradient and then equate it to zero:

$$\nabla H_u(x, \nabla_x J^*(x,t), u, t) = 0 \tag{2.18}$$

That is, every component of the gradient vector ∇H_u must vanish at the point of optimum. In addition, one needs to check if the derivative of the gradient (a matrix) $\frac{\partial^2 H}{\partial u^2}$ is positive semidefinite, which would indicate that the point of optimum is the true minima of the Hamiltonian. This inequality is known as the Legendre–Clebsch condition.

Formulation (2.18) allows the functional minimization problem, such as (2.4), to be transformed into a function minimization, which can be solved using ordinary calculus. Let

$$H^*(x, \nabla_x J^*(x,t), t) \equiv \min_u \left[H(x, \nabla_x J^*(x,t), u, t) \right] \tag{2.19}$$

If we can solve (2.18) for the optimal control $u = u^*$ and substitute it back into (2.17), we get the HJB partial differential equation (PDE), whose solution is the optimal cost $J^*(x(t),t)$:

$$-\frac{\partial J^*(x,t)}{\partial t} = H^*(x,\ \nabla_x J^*(x,t),\ t) \tag{2.20}$$

We need a boundary condition for (2.20) to be well-posed. Setting $t_0 = T$ in the cost index (2.2) yields

$$J^*(x(T),T) = S(x(T)) \tag{2.21}$$

Using Bellman's Principle of Optimality, we have arrived at sufficient conditions for optimal control solution to exist. The latter is defined by the HJB equation (2.20) together with its boundary condition (2.21). The sufficiency of (2.20), and (2.21) for control optimality means that if we can solve the former for $J^*(x,t)$ and calculate $u^*(t)$, then the latter constitutes the optimal control policy for the system (2.1), with respect to the cost index (2.2).

In most optimal control problems, we would be interested in the calculation of the optimal control policy u^* rather than the optimal cost J^*. Solving (2.20) is still quite difficult, even for low-order problems, in that we still must solve a PDE for the cost function $J^*(x,t)$.

As derived, the optimal policy $u^*(t)$ represents an open-loop control strategy, in the sense that u^* is computed as a function of time t. For practical applications, we would really want a feedback control policy, such as $u^* = u^*(x)$, to enforce robustness and reduce sensitivity of the solution to uncertainties that may exist in the system dynamics. We will see that if the dynamics are linear and the performance index penalty function $L(x,u,\tau)$ is quadratic, then the problem is easily solved, and the resulting optimal feedback control and the closed-loop system have very useful properties, with formal assurances of stability, performance, and robustness. In the forthcoming chapters, we shall derive and exploit these properties in our use of optimal control to maximize performance and robustness, while minimizing the control effort.

2.2.2 Summary

Dynamics	$: \dot{x} = f(x,u) \qquad x(t_0) = x_0$
Performance index	$: J(x,u,t) = \int_t^T L(x,u,\tau)d\tau + S(x(T))$
Optimal Cost	$: J^*(x,t) = \min_{u_{[t,T]}} [J]$
Hamiltonian	$: H(x,u,t) = L(x,u,t) + \frac{\partial J^*(x,t)}{\partial x} f(x,u)$
Optimal control	$: \nabla H_u(x,u,t) = 0 \Rightarrow \boxed{u^*(t)} \Rightarrow H^*(x,u^*,t)$
HJB - Equation	$: \begin{cases} -\frac{\partial J^*(x,t)}{\partial t} = H^*(x, \nabla_x J^*(x,t), t) \\ J^*(x(T),T) = S(x(T)) \end{cases}$

Example 2.1 In this example, we will set up (but not solve) the HJB equation. Consider the system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + u \end{aligned} \tag{2.22}$$

where $x(0) = [\,1 \quad 2\,]^T$, with the performance index

$$J = \int_0^1 \left(x_1^4 + u^2\right) dt + x_1^2(1) + x_2^2(1) \tag{2.23}$$

For this problem, $L(x,u,t) = x_1^4 + u^2$ and $S(x(T)) = x_1^2(1) + x_2^2(1) = x^T(1)x(1)$. The Hamiltonian is

$$\begin{aligned} H(x,u,\nabla_x J^*,t) &= L(x,u,t) + \frac{\partial J^*}{\partial x} f(x,u) \\ &= x_1^4 + u^2 + \frac{\partial J^*}{\partial x_1}\dot{x}_1 + \frac{\partial J^*}{\partial x_2}\dot{x}_2 \\ &= x_1^4 + u^2 + \frac{\partial J^*}{\partial x_1}(x_2) + \frac{\partial J^*}{\partial x_2}(-2x_1 - 3x_2 + u) \end{aligned} \tag{2.24}$$

Now, we minimize the Hamiltonian by differentiating the right-hand side of (2.24) with respect to the control and then equating the resulting derivative to zero. Thus,

$$\left(\nabla H_u = 0 = 2u^* + \frac{\partial J^*}{\partial x_2}\right) \Rightarrow \left(u^* = -\frac{1}{2}\frac{\partial J^*}{\partial x_2}\right) \tag{2.25}$$

Substituting the optimal solution back into (2.24), we get

$$H^*(x,\nabla_x J^*,t) = x_1^4 + \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - 2\frac{\partial J^*}{\partial x_2}x_1 - 3\frac{\partial J^*}{\partial x_2}x_2 - \frac{1}{2}\frac{\partial J^*}{\partial x_2} \tag{2.26}$$

The HJB equation is then

$$\begin{aligned} -\frac{\partial J^*}{\partial t} &= H^*(x,\nabla_x J^*(x,t),t) \\ &= x_1^4 + \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - \frac{\partial J^*}{\partial x_2}\left(2x_1 - 3x_2 - \frac{1}{2}\right) \end{aligned} \tag{2.27}$$

with the boundary condition $J^*(x,T) = x_1^2(T) + x_2^2(T)$.

2.3 Linear Quadratic Regulator

The linear quadratic regulator (LQR) is one of the most widely used control design methods in aerospace. Trade studies have been performed comparing properties of controllers (performance, robustness, control usage) in many different applications. For example, we have found that flight control systems designed using the LQR method have excellent performance, robustness, and minimize the control usage. This method is easily extended (in the next chapter) to produce command tracking controllers.

Consider the linear nonautonomous system

$$\dot{x} = A(t)x + B(t)u \qquad x(t_0) = x_0 \qquad x \in R^{n_x}, u \in R^{n_u} \tag{2.28}$$

with the quadratic performance index

$$J = \int_{t_0}^{T} \left(x^T Q\, x + u^T R\, u\right) d\tau + x^T(T)\, Q_T\, x(T) \tag{2.29}$$

where the cost weight matrices $(Q,\ R,\ Q_T)$ are symmetric positive semidefinite, positive definite, and positive semidefinite, respectively:

$$Q = Q^T \geqslant 0, \quad R = R^T > 0, \quad Q_T = {Q_T}^T \geqslant 0 \tag{2.30}$$

The weights Q and R can be time varying if needed. To have a well-posed problem, we would require the pair (A, B) to be controllable and the pair $\left(A, Q^{1/2}\right)$ to be observable. Weaker conditions, such as stabilizable $(A,\ B)$ and detectable $\left(A, Q^{1/2}\right)$, are also acceptable. The need for controllability of the system dynamics should be obvious. Clearly, the control cannot stabilize the system and perform as desired if the dynamics are not controllable. Detectability of modes through the performance index guarantees that the unstable modes are penalized, producing a control that will minimize their contribution to the index. We will see that the numerical choices of the matrices Q and R are very important in achieving performance and robustness in the closed-loop system.

Following (2.16), the LQR Hamiltonian is

$$H = x^T Q x + u^T R u + \frac{\partial J^*}{\partial x}\left(A(t)x + B(t)u\right) \tag{2.31}$$

Taking the gradient of H with respect to u and equating it to zero produces

$$\frac{\partial H}{\partial u} = 2Ru + B^T \nabla_x J^*(x, t) = 0 \tag{2.32}$$

where the optimal control is

$$u^* = -\tfrac{1}{2}R^{-1}B^T\nabla_x J^*(x,t) \tag{2.33}$$

Substituting u^* back into (2.20) yields the HJB equation

$$-\frac{\partial J^*}{\partial t} = x^T Q x + \tfrac{1}{4}\frac{\partial J^*}{\partial x}BR^{-1}B^T\nabla_x J^* + \frac{\partial J^*}{\partial x}Ax - \tfrac{1}{2}\frac{\partial J^*}{\partial x}BR^{-1}B^T\nabla_x J^* \tag{2.34}$$

which, in this form, is still quite difficult to solve. Fortunately, one can show that the optimal cost J^* is a quadratic time-varying function of the system state [3, Sect. 2.3]:

$$J^* = J^*(x(t),\, t) = x(t)^T P(t)\; x(t) \tag{2.35}$$

where $P(t) = P^T(t) > 0$. Substituting (2.35) into (2.34), we get

$$\frac{\partial J^*(x,\, t)}{\partial t} = x^T\dot{P}(t)x \qquad \nabla_x J^*(x,t) = 2\,P(t)x \tag{2.36}$$

Substituting (2.36) back into (2.34) and factoring out x on both sides, we get

$$x^T\left[-\dot{P}(t) - P(t)A - A^T P(t) - Q + P(t)B\,R^{-1}B^T P(t)\right]\; x = 0 \tag{2.37}$$

with the boundary condition $P(T) = Q_T$. Since this must be satisfied for any state x, the following initial value problem must be true:

$$\begin{aligned} -\dot{P}(t) &= P(t)A + A^T P(t) + Q - P(t)B\,R^{-1}B^T P(t) \\ P(T) &= Q_T \end{aligned} \tag{2.38}$$

The time-varying matrix ordinary differential equation in (2.38) is called the Riccati equation. Substituting (2.36) into (2.33) yields the optimal control policy

$$u^*(x,\, t) = -\tfrac{1}{2}R^{-1}B^T\frac{\partial J^*(x,\, t)}{\partial x} = -\underbrace{R^{-1}B^T P(t)}_{K(t)}\,x = -K(t)\,x \tag{2.39}$$

in state feedback form. Note that in (2.38), the Riccati equation is integrated backward in time. Then, the optimal control $u^*(x,\, t)$ with the feedback gains $K(t)$ are formed using (2.39). For real time operations, these gains must be stored in a lookup table, and the feedback control law would be implemented by looking up the gains in the table. This is gain scheduling.

Continuing on, we substitute the optimal control (2.39) into the system dynamics (2.28) and obtain the closed-loop system:

$$\begin{aligned}
&\dot{x} = A(t)x + B(t)u \qquad x(t_0) = x_0 \\
&u(t) = -K(t)x \\
&\dot{x} = (A(t) - B(t)K(t))x
\end{aligned} \tag{2.40}$$

2.3.1 Summary

Dynamics	$: \dot{x} = A(t)x + B(t)u \qquad x(t_0) = x_0$
Performance index	$: J = \int_{t_0}^{T} (x^T Q x + u^T R u)\mathrm{d}\tau + x^T(T) Q_T x(T)$
Riccati Equation	$: \begin{cases} -\dot{P}(t) = P(t)A + A^T P(t) + Q - P(t)B\,R^{-1}B^T P(t) \\ P(T) = Q_T \end{cases}$
Optimal Control	$: u^* = -R^{-1}B^T P(t)x = -K(t)x$
Closed - Loop System	$: \dot{x} = (A(t) - B(t)K(t))x, x(0) = x_0$

2.4 Infinite-Time LQR Problem

In this section, we consider the quadratic performance index on an infinite time interval:

$$J = \int_0^{\infty} \left(x^T Q\, x + u^T R\, u\right) d\tau \qquad Q = Q^T \geqslant 0, \;\; R = R^T > 0 \tag{2.41}$$

where the final time $T = \infty$. The state dynamics for this problem are assumed to be linear-time-invariant:

$$\dot{x} = Ax + Bu \qquad A, B - \text{constant} \qquad x \in R^{n_x}, u \in R^{n_u} \tag{2.42}$$

with (A, B) stabilizable and $\left(A, Q^{1/2}\right)$ detectable. It is possible to show that the corresponding Riccati equation

$$-\dot{P}(t) = P(t)A + A^T P(t) + Q - P(t)B\,R^{-1}B^T P(t) \tag{2.43}$$

with the limiting boundary condition

$$\lim_{T \to \infty} P(T) = 0_{n_x \times n_x} \tag{2.44}$$

has the unique solution [3]. Moreover, in the limit as $(t \to -\infty)$, this solution tends to a constant symmetric positive definite matrix, which can be found by solving the algebraic Riccati equation (ARE)

$$PA + A^T P - P B R^{-1} B^T P + Q = 0 \tag{2.45}$$

with the LQR-optimal control policy in feedback form

$$u = -\underbrace{R^{-1} B^T P}_{K} x = -K x \tag{2.46}$$

where $K \in R^{n_u \times n_x}$ is a constant matrix of the LQR-optimal feedback gains.

In industrial applications with nonlinear process control, the system models are often linearized at the designated operating conditions. Then, LQR-based controllers can be designed at each operating point. The resulting constant feedback gains K would be stored in a table and recalled (looked up) in real time for implementation. As we have already mentioned, this is the gain-scheduling control concept.

Substituting the optimal feedback control (2.46) into the open-loop dynamics (2.42) gives the closed-loop system:

$$\dot{x} = \underbrace{(A - BK)}_{A_{\text{cl}}} x = A_{\text{cl}} x \tag{2.47}$$

The LQR formulation guarantees the closed-loop system (2.47), whose dynamics are described by the constant matrix A_{cl}, to be stable [3]. This means the eigenvalues of A_{cl} lie in the left half complex plane $\text{Re}(\lambda(A_{\text{cl}}))<0$. The system state is regulated to zero, $x \to 0$ as $t \to \infty$, which implies $u \to 0$ as $t \to \infty$.

It is often desirable when simulating the dynamics to compute and examine the peak values of the optimal control u and its rate $\dot{u}$. If we differentiate u, we get

$$\dot{u} = -K\dot{x} = -K(A - BK)x = -K A_{\text{cl}} x \tag{2.48}$$

We can form a closed-loop simulation model, with outputs x, u, and $\dot{u}$, as

$$\begin{aligned} \dot{x} &= Ax + Bu \quad u = -Kx \\ \dot{x} &= (A - BK) \;\; x = A_{\text{cl}} x \\ y &= \begin{bmatrix} x \\ u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} I \\ -K \\ -KA_{\text{cl}} \end{bmatrix} x \end{aligned} \tag{2.49}$$

In real-life applications, and especially in flight control, it is critical to prevent saturation of the control surface positions and rates. When this happens, nonlinear effects begin to dominate the system response, stability is no longer guaranteed, and the system could depart. We can see from (2.49) that large gains K may cause large

control positions and rates. Thus, high gains are undesirable in most industrial control applications. From (2.46), we see that K gets large as P gets large. From (2.45), we see that it is the choice of the weights Q and R in the ARE that determines how large the gains will be.

2.4.1 Summary

Dynamics	$: \dot{x} = Ax + Bu \qquad x(0) = x_0$
Performance index	$: J = \int_0^\infty (x^T Q x + u^T R u) \mathrm{d}\tau$
Algebraic Riccati Equation	$: PA + A^T P - P B R^{-1} B^T P + Q = 0$
Optimal Control	$: u = -R^{-1} B^T P x = -Kx$
Closed - Loop System	$: \dot{x} = (A - BK)x, x(0) = x_0$
Simulation output	$: y = \begin{bmatrix} x \\ u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} I \\ -K \\ -KA_{\mathrm{cl}} \end{bmatrix} x$

Example 2.2 In this example, we wish to solve for the optimal control and examine the properties of the closed-loop system. Consider the following linear-time-invariant model

$$\dot{x} = Ax + Bu \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.50}$$

with the performance index

$$J = \int_0^\infty \left(x_1^2 + ru^2\right) d\tau \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad R = r \tag{2.51}$$

The eigenvalues of the open-loop system are $\lambda = 0$ and $\lambda = -1$. In the performance index, the state penalty matrix Q penalizes the first state of the system. The control penalty r is left as a parameter so we can see how small and large values of r change the closed-loop dynamics. It is always important to check if the design problem is well-posed. Conditions on the plant and on the performance index for a well-posed problem require to check if the unstable modes of the system are

controllable and if the unstable modes are observable through the state penalty matrix. In other words, we need to verify if (A, B) is stabilizable and $\left(A, Q^{\frac{1}{2}}\right)$ is detectable. First, we compute the controllability matrix:

$$P_c = [B \quad AB] = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}_{RK=2} \tag{2.52}$$

Since this matrix has full rank, the system is controllable. So any unstable modes are controllable. Next, we can factor the state penalty matrix into square roots

$$Q = \left(Q^{\frac{1}{2}}\right)^T Q^{\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.53}$$

and then check the observability using the square root of Q:

$$\begin{bmatrix} Q^{\frac{1}{2}} \\ Q^{\frac{1}{2}}A \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}_{\text{RK}=2} \tag{2.54}$$

Since this matrix also has full rank, all modes of the system are observable through the penalty matrix. Now, we can solve the ARE

$$\text{PA} + A^T P - P B R^{-1} B^T P + Q = 0 \tag{2.55}$$

for P, using A, B, Q, and $R = r$. Let $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$. Then the ARE is

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} - \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{r} [0 \quad 1] \times \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = 0 \tag{2.56}$$

Since the Riccati matrix P must be real symmetric and positive definite, from (2.56) we can derive three equations for p_1, p_2, and p_3. These are

$$\begin{aligned} -\frac{p_2^2}{r} + 1 &= 0 \\ l - p_2 - \frac{p_2 p_3}{r} &= 0 \\ 2(p_2 - p_3) - \frac{p_3^2}{r} &= 0 \end{aligned} \tag{2.57}$$

The first equation gives $p_2 = \sqrt{r}$ (both positive and negative values of m must be checked to see which is the solution). Using $p_2 = \sqrt{r}$, p_1 and p_3 are

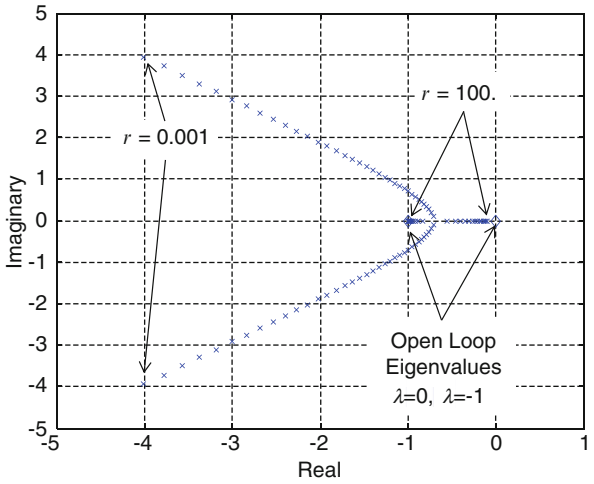


Fig. 2.1 Example 2.2 root locus varying the LQR control penalty parameter

$$p_3 = r\left(\sqrt{1+\frac{2}{\sqrt{r}}}-1\right)$$
$$p_1 = \sqrt{r}\sqrt{1+\frac{2}{\sqrt{r}}} \tag{2.58}$$

The constant state feedback gain matrix is

$$K = R^{-1}B^T P = \left[\frac{2}{\sqrt{r}} \quad \sqrt{1+\frac{2}{\sqrt{r}}}-1\right] \tag{2.59}$$

The closed-loop state dynamics are always stable with characteristic equations

$$\phi_{\mathrm{cl}}(s) = s^2 + s\sqrt{1+\frac{2}{\sqrt{r}}} + \frac{1}{\sqrt{r}} \tag{2.60}$$

By varying the control penalty r in (2.60), we can compute a root locus (Fig 2.1) to show how the numerical choice of R impacts the closed-loop system dynamics.

The root locus data in Fig. 2.1 are the result of changing the control penalty r from 0.001 to 100. For large values of r (small gains), the closed-loop poles are near the open-loop poles ($r = 100$, $K = [0.1 \quad 0.0954]$), producing a slow system response. For small values of r (large gains), the roots follow asymptotes into the left half plane, and the response gets fast ($r = 0.01$, $K = [31.6228 \quad 7.0153]$). In general, the values of the optimal feedback gains are proportional to the relative magnitude of Q and R. For a fixed R, large values of Q heavily penalize the state

(relative to the control), the resulting optimal feedback gains grow large, and the closed-loop system response gets fast. On the other hand, small values of Q penalize the control more than the state, resulting in smaller control efforts. This also keeps the gains small, producing a slower response. □

2.5 Guaranteed Stability Margins for State Feedback LQR

The LQR solution has excellent stability robustness properties at the input to the plant. This can be shown by examining the return difference matrix in the frequency domain. Readers who are not familiar with frequency domain analysis of multi-input multi-output linear-time-invariant systems should see Chap. 5 and then return to this section.

Consider the following LTI system:

$$\dot{x} = Ax + Bu \quad x \in R^{n_x} \quad u \in R^{n_u} \tag{2.61}$$

along with the infinite-time LQR problem

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) \mathrm{d}\tau \tag{2.62}$$

with the weights $Q = Q^T \geq 0$ and $R = R^T > 0$. Suppose that (A, B) is stabilizable and $\left(A, Q^{1/2}\right)$ is detectable. The ARE is

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{2.63}$$

and the corresponding optimal state feedback control is given by

$$u = -R^{-1}B^T P x = -Kx \tag{2.64}$$

Substituting (2.64) into (2.61) yields the closed-loop system

$$\dot{x} = (A - BK)x = A_{\mathrm{cl}} x \tag{2.65}$$

Of particular interest are frequency domain properties provided by the LQR state feedback controller (2.64). First, we introduce the loop transfer function $L(s)$, with the loop break point at the plant input. For the state feedback system shown in Fig 2.2, the loop gain is

$$L(s) = K(sI - A)^{-1}B \tag{2.66}$$

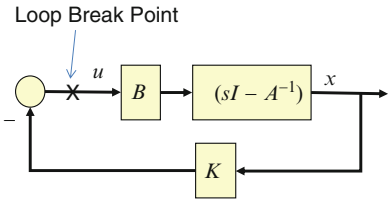


Fig. 2.2 Block diagram of the state feedback architecture

We define the open-loop dynamics $\Phi(s) = (sI - A)^{-1}$ and its transposed conjugate $\Phi^* = \Phi^T(-s)$. Using the ARE (2.63), we add and subtract $(s\,P)$ from both sides and rearrange the terms. We get

$$P(sI - A) + \left(-sI - A^T\right)P + PBR^{-1}B^TP = Q \tag{2.67}$$

insert $\Phi = (sI - A)^{-1}$, left-multiply by $B\,\Phi^*$, where $(\bullet)^*$ denotes complex conjugate transpose, right-multiply by ΦB, and arrive at

$$B^TP\Phi B + B^T\Phi^*\mathrm{PB} + B^T\Phi^*PBR^{-1}B^TP\Phi B = B^T\Phi^*Q\Phi B \tag{2.68}$$

We then add $R>0$ from the performance index (2.62) to both sides and note that $K = R^{-1}B^TP$ and $L(s) = K(sI - A)^{-1}B = K\Phi B = R^{-1}B^TP\Phi B$. Substituting $L(s)$ into (2.68) gives

$$\begin{aligned} &R\underbrace{R^{-1}B^TP\Phi B}_{L(s)} + \underbrace{B^T\Phi^*PBR^{-1}}_{L^T(-s)}R + \underbrace{B^T\Phi^*PBR^{-1}}_{L^T(-s)}R\underbrace{R^{-1}B^TP\Phi B}_{L(s)} + R \\ &= B^T\Phi^*Q\Phi B + R \end{aligned} \tag{2.69}$$

Simplifying (2.69), we obtain

$$R + RL(s) + L^T(-s)R + L^T(-s)RL(s) = B^T\Phi^*Q\Phi B + R \tag{2.70}$$

which can be further reduced to

$$(I + L(s))^*R(I + L(s)) = R + B^T\Phi^*Q\Phi B \tag{2.71}$$

where $(I + L(s))$ is the return difference matrix, computed at the system input break point. The term $B^T\Phi^*Q\Phi B$ is a Hermitian positive semidefinite matrix. By removing this term on the right side, we form the inequality

$$(I + L(s))^*R(I + L(s)) \geq R \tag{2.72}$$

If we assume an equal penalty on each control, that is, $R = \rho I$, $\rho>0$, then,

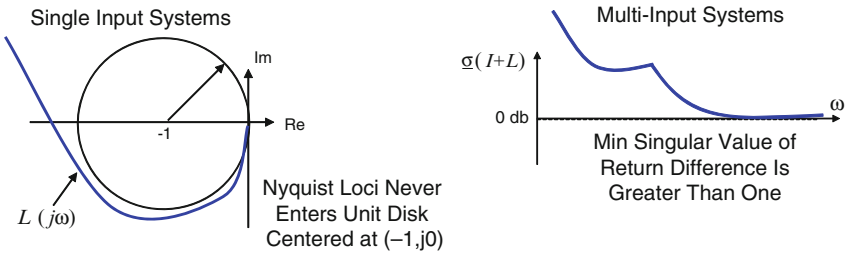


Fig. 2.3 Frequency domain analysis of optimal state feedback loop transfer functions

$$(I + L(s))^*(I + L(s)) \geq I \tag{2.73}$$

which tells us information about the magnitude of the return difference matrix. For single input systems, where $n_u = 1$, this is equivalent to the Nyquist locus not entering a unit circle centered about $(-1, j0)$. For multi-input system, where $n_u > 1$, this says that the minimum singular value of the return difference matrix (versus frequency) always has a magnitude greater than one. These properties are shown in Fig. 2.3.

This implies a minimum gain margin of $[-6, +\infty]$ dB and a phase margin of at least 60°. This property is what makes the LQR so attractive in industrial applications. Often, for open-loop unstable design problems, it is very difficult to achieve the desired gain and phase margins. This property is guaranteed (under the assumptions shown) for any choice of Q matrix. However, experience has shown that large feedback gains seldom work in practice. Because of modeling errors, unmodeled dynamics, noise, actuator rate saturation, and other disturbances, these excellent margins are not always realized in the physical system. So, special care must be taken to avoid large control gains (leading to a high bandwidth design) in most physical systems, and especially in aerospace applications.

There are many practical "rules-of-thumb" for selecting the LQR weight matrices. In the next chapter, a design method for tuning the LQR solution to achieve the desired performance and robustness without large gains will be given. We will also discuss how the eigenvalues of the closed-loop system evolve with the numerical choices made for the penalty matrices.

2.6 LQR Design and Asymptotic Properties

The numerical values in the LQR penalty matrices Q and R determine the eigenstructure of the closed-loop system $(A - BK_c)V = V\Lambda$. This eigenstructure specifies the system performance and robustness properties. It is very important to properly choose the numerical values for elements in Q and R, and more importantly, it is quintessential to learn how to exploit these matrices to tune the control

feedback gains and to achieve the desired performance and robustness in the resulting closed-loop system.

In this section, we shall investigate how the eigenstructure evolves, as the weighting matrices are varied numerically. Readers interested in detailed asymptotic analysis may find it in Kwakernaak and Sivan [1].

Consider the following LTI system:

$$\dot{x} = Ax + Bu \quad x \in R^{n_x} \quad u \in R^{n_u} \tag{2.74}$$

with the infinite-time quadratic cost index

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) \mathrm{d}\tau \tag{2.75}$$

where $Q = Q^T \geq 0, \quad R = R^T > 0$, (A, B) stabilizable, and $\left(A, Q^{1/2}\right)$ detectable. We assume that there are no transmission zeros on the $j\omega$ axis. Then, the ARE for this optimal control problem is

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{2.76}$$

Associated with this ARE is the $2n_x \times 2n_x$ Hamiltonian matrix H given by

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{2.77}$$

which can be used to determine the solution to the ARE. The optimal state feedback control is given by

$$u = -R^{-1}B^T P x = -Kx \tag{2.78}$$

which when substituted into (2.74) yields the closed-loop system

$$\dot{x} = (A - BK)x = A_{\mathrm{cl}} x \tag{2.79}$$

The n_x eigenvalues of the closed-loop system $\lambda(A_{\mathrm{cl}})$ are the stable eigenvalues of the Hamiltonian matrix H. In fact, the Hamiltonian matrix H has $2n_x$ eigenvalues of which n_x have negative real parts (stable) and n_x have positive real parts (unstable, but stable backward in time). Let

$$\phi_{\mathrm{cl}}(s) = \det[sI - A + BK] \tag{2.80}$$

then,

$$\det[sI - H] = \phi_{\text{cl}}(s)\phi_{\text{cl}}(-s) \tag{2.81}$$

The asymptotic properties we desire to explore are those associated with the migration of these eigenvalues, as the numerical values in the LQR penalty matrices Q and R are varied. We can examine these eigenvalues (roots of $\phi_{\text{cl}}(s)$) through the polynomial formed by expanding the $\det[sI - H]$. We begin with some elementary row and column operations on H. First, we multiply the first row of H by $-Q(sI - A)H^{-1}$ and add it to the second row. This yields

$$\begin{aligned}\det[sI - H] &= \det\begin{bmatrix} sI - A & BR^{-1}B^T \\ Q & sI + A^T \end{bmatrix} \\ &= \det\begin{bmatrix} sI - A & BR^{-1}B^T \\ 0 & \left(sI + A^T\right) - Q(sI - A)^{-1}BR^{-1}B^T \end{bmatrix}\end{aligned} \tag{2.82}$$

Then,

$$\begin{aligned}\det[sI - H] &= \det[sI - A]\det\left[\left(sI + A^T\right) - Q(sI - A)^{-1}BR^{-1}B^T\right] \\ &= \det[sI - A]\det\left[\left(sI + A^T\right)\left\{I - \left(sI + A^T\right)^{-1}Q(sI - A)^{-1}BR^{-1}B^T\right\}\right] \\ &= \det[sI - A]\det\left[sI + A^T\right]\det\left[I - \left(sI + A^T\right)^{-1}Q(sI - A)^{-1}BR^{-1}B^T\right]\end{aligned} \tag{2.83}$$

We factor the Q and $BR^{-1}B^T$ into products of two square roots: $Q = Q_1^T Q_1$ and $BR^{-1}B^T = R_1 R_1^T$. Next, using the identity $\det[I - AB] = \det[I - BA]$, we get

$$\begin{aligned}&\det\left[I - \underbrace{\left(sI + A^T\right)^{-1}Q_1^T}_{B}\underbrace{Q_1(sI - A)^{-1}R_1R_1^T}_{A}\right] \\ &\quad = \det\left[I - Q_1(sI - A)^{-1}R_1R_1^T\left(sI + A^T\right)^{-1}Q_1^T\right]\end{aligned} \tag{2.84}$$

and so,

$$\begin{aligned}&\det[sI - H] \\ &= \det[sI - A]\det\left[sI + A^T\right]\det\left[I - \underbrace{Q_1(sI - A)^{-1}R_1}_{H_1(s)}R_1^T\left(sI + A^T\right)^{-1}Q_1^T\right] \\ &= \phi(s)(-1)^{n_x}\phi(-s)\det\left[I + H_1(s)H_1^T(-s)\right]\end{aligned} \tag{2.85}$$

where $\phi(s) = \det[sI - A]$. Thus,

$$\phi_{\text{cl}}(s)\phi_{\text{cl}}(-s) = \phi(s)\phi(-s)\det\left[I + H_1(s)H_1^T(-s)\right] \tag{2.86}$$

Let

$$\det[H_1(s)] = \frac{\psi(s)}{\phi(s)} \tag{2.87}$$

and consider the performance cost index

$$J = \int_0^\infty \left(x^T Q x + \rho^2 u^T R u\right) d\tau \tag{2.88}$$

with a positive scalar weight $\rho > 0$. We are interested in the behavior as $\rho \to 0$ and as $\rho \to \infty$. The zeros of (2.86) are also the zeros of

$$\phi(s)\phi(-s)\det[\rho I + H_1(s)H_1(-s)] \tag{2.89}$$

As $\rho \to 0$, some of the roots will go to infinity. Those that stay finite will approach the transmission zeros of the transfer function matrix $H_1(s)$ and their negative values. These finite zeros control the dynamic response of the optimal regulator. As $\rho \to \infty$, the roots of $\phi_{\text{cl}}(s)$ are the n_x stable roots of $\phi(s)\phi(-s)$. That is, if the roots of $\phi(s)$ have positive real part, then the mirror image of them in $\phi(-s)$ will become the stable roots in $\phi_{\text{cl}}(s)$.

We see that shaping the zeros of $H_1(s)$ plays a crucial role in the design of the optimal control. This is done through selection of the LQR penalty matrix weights Q and R. Later in Chap. 3, we will use this fact to tune the design of optimal controllers to achieve performance and robustness.

2.7 Conclusions

In this chapter, we briefly discussed optimal control theory and the linear quadratic regulator. Many control systems today are designed using this method due to the frequency domain guarantees and the ease of the design. In the next chapter, we shall extend the regulator architecture to command tracking systems.

2.8 Exercises

Exercise 2.1. Consider

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + u \end{aligned} \qquad \begin{aligned} t_0 &= 0 \\ T &= 1 \end{aligned} \qquad x(0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$J = \int_0^T \left(x_1^4 + u^2\right) d\tau + x_1^2(T) + x_2^2(T)$$

Set up (but not solve) the HJB equation with the corresponding boundary conditions.

Exercise 2.2. Given

$$\dot{x} = -x + u$$

$$J = \int_0^1 \left(x^2 + u^2\right) d\tau + x^2(1)$$

use the ARE to find the optimal feedback control $u^*(t) = -K\,x(t)$. Draw the closed-loop system block diagram.

Exercise 2.3. For

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 + u \end{aligned}$$

$$J = \int_0^\infty \left(x_1^2 + u^2\right) d\tau$$

find the LQR-optimal control policy u to minimize the cost J.

Exercise 2.4. Consider the longitudinal aircraft dynamics given in Chap. 1, Exercise 1.2. This linear model represents the aircraft incremental dynamics, with respect to a trim condition. Design an infinite-time LQR to regulate the state vector to zero. Simulate the design with an initial state vector $x(0) = [\,10 \text{ ft/s} \quad 0.1 \text{ rad} \quad 0.1 \text{ rad/s} \quad 0 \text{ rad}\,]$.

Exercise 2.5. Consider a second-order system modeled by the input-output equation $\ddot{y} = u$. A feedback controller $u = -k_1 y - k_2 \dot{y}$ is to be designed such that the performance index

$$J = \int_0^\infty \left(4y^2 + u^2\right) d\tau$$

is minimized.

1. Find k_1 and k_2.
2. What are the closed-loop eigenvalues?

Exercise 2.6. Consider the linear system $\dot{x} = Ax + Bu$ with LQR performance index

$$J = \tfrac{1}{2} \int_0^\infty \left(q\, x^T Q x + \rho\, u^T R u\right) \mathrm{d}\tau$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, and $\left(A, Q^{\frac{1}{2}}\right)$ observable. Use the state feedback control $u = -Kx$ with $K = R^{-1}B^T P$, where P is the ARE solution matrix. What happens to the eigenvalues of the closed-loop system as

1.
$$q \to 0$$

2.
$$\rho \to 0$$

Exercise 2.7. Consider the following scalar linear quadratic command tracking problem:

$$\dot{x} = u$$
$$J = \frac{1}{2} \int_0^T \left[(x - r)^2 + \rho u^2\right] d\tau + \frac{1}{2} q_T (x(T) - r(T))^2$$

with $q_T > 0$, $\rho > 0$, T fixed, and a known reference (command) input $r(t)$.

1. What is the HJB equation for this problem? (Eliminate u). Include boundary conditions.
2. Find a solution for J^* in the form

$$J^*(x, t) = \tfrac{1}{2} P(t) x^2 + g(t) x + w(t)$$

Find differential equations for P, g, and w, such that the HJB equation is satisfied. Include boundary conditions. Derive but do not solve the related equations.

References

1. Athans, M., Falb, P.L.: Optimal Control: An Introduction to the Theory and Its Applications. Dover Books on Engineering, New York (2006)
2. Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems. Wiley, New York (1972)
3. Anderson, B.D.O., Moore, J.B.: Optimal Control, Linear Quadratic Methods. Dover, Mineola (1990)
4. Khalil, H.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (1996). 07458
5. Bellman, R.E.: Dynamic Programming. Dover, Mineola (2003)

Chapter 3
Command Tracking and the Robust Servomechanism

3.1 Introduction

Most industrial control problems require the control system to accurately track commands. This requirement distinguishes these problems from regulation in which the state is driven to zero. From classical control theory, we know that in order to track a constant command with zero error, we need to add integral error control action into the controller. For single-input single-output (SISO) systems, the loop transfer function $L(s)$ can be written as

$$L(s) = \frac{K(b_0 s^m + \cdots + b_{m-1}s + 1)}{s^p(a_0 s^n + \cdots + a_{n-1}s + 1)} \tag{3.1}$$

where the gain K and the polynomial coefficients a_i and b_i are real constants. The *type* of the control system depends upon the order p of the pole of $L(s)$ at $s = 0$. The number of finite zeros, their location, or the location of the poles are not important to specify the system type. The system type p, where $p = 0, 1, 2, \cdots$ indicates how many integrators are present in the control system. We know that in order to track a constant command $r(t) = \text{constant}$, and to produce zero steady-state error, an integrator is needed, $p \geq 1$, creating (at a minimum) a type 1 system. In order to track a type 1 input, the control system will need two integrators, creating a type 2 system. Thus, to track commands accurately, the class of commanded signals must be known, and the controller must be augmented with enough integrators to produce zero steady-state errors.

When these integrators are added to the control system for command tracking, they also provide disturbance rejection within the same class, that is, a type 1 control system can track constant commands and reject constant disturbances. Similarly, a type 2 system can track ramp inputs and reject ramp disturbances.

Basically, the augmentation of the system with these integrators for command tracking requires embedding into the system a model of the class of signals that the system will track. This is often referred to as the *internal model principle* [1].

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_3,

For instance, when tracking a constant command and adding a single integrator, we have embedded the command generation internal model $\dot{r} = 0$ into the system.

In the previous chapter, we have illustrated the use of linear quadratic optimal control theory to design a controller and examined the excellent stability properties provided by that method. The linear quadratic regulator (LQR) forces the system state to go to zero, forming a type 0 control system. If one wants to track a constant command using such an LQR controller, the system would have a steady-state offset error to the command. We know from Eq. (3.1) that in order to track a constant command with zero error, we need to add an integrator, creating a type 1 control system.

A natural extension of the LQR method presented in the previous chapter would be to add an integral control action into the controller to produce zero steady errors, while tracking constant commands. The number of integrators that would need to be added depends upon the commanded signal (whether it is a constant, a ramp, or other type of signal).

This chapter presents a systematic process for building an augmented state space model called the *servomechanism design model* [2]. This state space description embeds a model of the class of signals to be tracked, such that when optimal control theory is applied, the state regulation provides accurate tracking of the selected class of external commands. This system is then decomposed into two parts: a servo tracking controller for command following and a state feedback component for stabilization. In aerospace, this approach is often used to design flight control systems for both manned and unmanned aerial vehicles. The resulting control architecture provides accurate command tracking and a robust control system design with predictable and robust performance. The meaning of controller robustness was introduced in Chap. 2. It requires the minimum singular value of the return difference matrix having magnitude greater than 1. This topic of robustness in the frequency domain is covered in great detail later in Chap. 5.

3.2 The Servomechanism Design Model

Consider the following finite dimensional linear-time-invariant state space model:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew \\ y &= Cx + Du \end{aligned} \tag{3.2}$$

with an unknown bounded disturbance w and with the signals $x \in R^{n_x}$, $u \in R^{n_u}$, and $y \in R^{n_y}$ representing the system state, control, and output, respectively. We assume that the system is both controllable and observable. We would like a preselected subset of the output vector y to track the command input vector $r \in R^{n_r}$, and we assume that the dimension of r is less than or equal to the number of the system outputs (i.e., $n_y \geq n_r$). It is also assumed that the p^{th} order differential equation for $r(t)$ is given, with the following model:

Table 3.1 Internal models for command generation

Command signal $r(t)$	Differential equation	Model parameters
Constant	$\dot{r} = 0$	$p = 1, a_1 = 0$
Ramp	$\ddot{r} = 0$	$p = 2, a_1 = a_2 = 0$
Sinusoid	$\ddot{r} = -\omega_0^2 r$	$p = 2, a_1 = 0, a_2 = -\omega_0^2$

$$\overset{(p)}{r} = \sum_{i=1}^{p} a_i \overset{(p-i)}{r} \tag{3.3}$$

where the scalar coefficients a_i are known and the superscript (i) denotes the i^{th} derivative. Using the model in (3.3), examples for typical signals are shown in Table 3.1.

The polynomial formed by the Laplace transformation of (3.3) is

$$a(s) = s^p + \sum_{i=1}^{p} a_i s^{p-i}, \tag{3.4}$$

and it gives a known class of inputs without the knowledge of their magnitudes. Our control goal is to track this command with zero steady-state error. For disturbance inputs, we assume the same model as $r(t)$:

$$\overset{(p)}{w} = \sum_{i=1}^{p} a_i \overset{(p-i)}{w} \tag{3.5}$$

where $w_0 = w(0)$ is unknown.

Let us define the tracking error signal as

$$e = y_c - r \tag{3.6}$$

where $y_c \in R^{n_r}$ is the subset of the output y to be controlled and $e \in R^{n_r}$. The error signal is defined here as $e = y_c - r$ so that we can apply negative feedback of the errors and their derivatives in forming the feedback control. We will also arrange the output vector so that the first n_r variables in y define y_c. Thus,

$$y = \begin{bmatrix} y_c^T & y_{nc}^T \end{bmatrix}^T \tag{3.7}$$

where y_{nc} are output variables that are not controlled. The model for $y_c \in R^{n_r}$ is

$$y_c = C_c x + D_c u \tag{3.8}$$

This is the *regulated* system output. Tracking in y_c is the same as regulation in e; therefore, the objective is to make the error approach zero $e \to 0\,(y_c \to r)$, as $t \to \infty$, in the presence of unmeasurable disturbance w, in a robust manner with respect to the plant description. Taking (3.6) and differentiating p times, the resulting differential equation for the error may be written as

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = \overset{(p)}{y_c} - \sum_{i=1}^{p} a_i \overset{(p-i)}{y_c} - \left(\overset{(p)}{r} - \sum_{i=1}^{p} a_i \overset{(p-i)}{r} \right) \tag{3.9}$$

From (3.3), the bracketed term in the right side of (3.9) will be zero. Using (3.8), we have

$$\overset{(p-i)}{y_c} = C_c \overset{(p-i)}{x} + D_c \overset{(p-i)}{u} \tag{3.10}$$

Substituting this into (3.9) yields

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = C_c \left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right] + D_c \left[\overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \right] \tag{3.11}$$

This system represents a set of coupled ordinary differential equations. Let ξ and μ be defined as

$$\xi = \overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \tag{3.12}$$

$$\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \tag{3.13}$$

The error equation is

$$\overset{(p)}{e} - \sum_{i=1}^{p} a_i \overset{(p-i)}{e} = C_c \xi + D_c \mu \tag{3.14}$$

Differentiating (3.12), we get

$$\dot{\xi} = \overset{(p+1)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i+1)}{x} \tag{3.15}$$

Using (3.2) for $\dot{x}$ results in

$$\dot{\xi} = A \underbrace{\left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right]}_{\xi} + B \underbrace{\left[\overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} \right]}_{\mu} + E \underbrace{\left[\overset{(p)}{w} - \sum_{i=1}^{p} a_i \overset{(p-i)}{w} \right]}_{=0} \tag{3.16}$$

where (3.5) shows the last term to be zero. We can rewrite (3.16) as

$$\dot{\xi} = A\xi + B\mu \tag{3.17}$$

which is the original system model minus the disturbances.

The servomechanism design model is formed by creating a new state space model, containing the error dynamics and the system model from (3.17). The new state vector is z, and its components are the errors $e, \cdots, \overset{(p-1)}{e}$, with the vector ξ. The error is a linear combination of ξ and μ from (3.14). Augmenting z with these derivatives and ξ defines z to be

$$z = \begin{bmatrix} e \\ \dot{e} \\ \vdots \\ \overset{(p-1)}{e} \\ \xi \end{bmatrix} \tag{3.18}$$

This new state vector z has dimension $(n_x + p \times n_r)$. Differentiating (3.18) yields the *servomechanism design model*:

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.19}$$

where $\tilde{A}$ and $\tilde{B}$ are given by

$$\tilde{A} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & 0 & I & 0 \\ a_p I & a_{p-1} I & \cdots & a_2 I & a_1 I & C_c \\ 0 & \cdots & \cdots & \cdots & 0 & A \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ D_c \\ B \end{bmatrix} \tag{3.20}$$

The robust servomechanism LQR solution is obtained by applying linear quadratic regulator theory to (3.19). By regulating z, we regulate to zero both e, its $(p-1)$ derivatives, and ξ. In steady state, this allows the state vector x to be nonzero, in which case, $C_c x + D_c u = r$. This control formulation adds the desired integral control action acting on the command error.

3.2.1 Controllability of the Servomechanism Design Model

If we apply the Hautus controllability tests to the servomechanism design model in (3.19), for the system to be controllable, we must have

$$\operatorname{rank}\left[sI - \tilde{A} | \tilde{B}\right] = n_x + n_r \times p \tag{3.21}$$

where s evaluates at each of the eigenvalues of $\tilde{A}$. This matrix has $n_x + n_r \times p$ rows that must all be independent to have full rank. To derive this requirement, simply use elementary row and column operations to transform $\left[sI - \tilde{A} | \tilde{B}\right]$ into the following:

$$\left[\begin{array}{cc} sI - \tilde{A} & \tilde{B} \end{array}\right] = \left[\begin{array}{cccccc} sI & -I & 0 & 0 & 0 & 0 \\ 0 & sI & -I & 0 & 0 & 0 \\ 0 & 0 & sI & -I & 0 & 0 \\ 0 & 0 & 0 & \frac{a(s)}{s^p} I & -C & D \\ 0 & 0 & 0 & 0 & sI - A & B \end{array}\right]$$

$$a(s) = s^p - \sum_{i=1}^{p} a_i s^{p-i} \tag{3.22}$$

Clearly, the first $n_r \times (p-1)$ rows are independent. From the last row, $[sI - A | B]$ must be full rank which says that the original system model must be controllable. Considering the last two rows, if $s = s_i$ such that $a(s_i) = 0$ (a zero of $a(s)$), then we must have

$$\operatorname{rank}\left[\left[\begin{array}{cc} -C_c & D_c \\ s_i I - A & B \end{array}\right]\right] = n_x + n_r \tag{3.23}$$

For this to occur, the multivariable zeros or transmission zeros of the original system must not equal any zeros of $a(s)$ and $n_u \geq n_r$. To summarize are the following:

1. The original system (A, B) must be controllable.
2. The number of controls must be greater than the number of signals to track, $n_u \geq n_r$.
3. The original system (A, B, C_c, D_c) must not have any transmission zeros common with the polynomial $a(s)$.

For control design, we can often relax the controllability requirement to that of stabilizability. For stabilizability, the original system (A, B) must be stabilizable, and conditions (2) and (3) above must also be satisfied.

Example 3.1 Constant Command Tracking Consider a constant command r. According to (3.4), this gives $\dot{r} = 0$ $(p = 1)$, with $a_1 = 0$. The command error is $e = y_c - r$. The servomechanism design model using (3.19) is given by

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \dot{x} \end{bmatrix}, \mu = \dot{u}$$
$$\tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} D_c \\ B \end{bmatrix} \tag{3.24}$$

Example 3.2 Sinusoidal Command Tracking Consider a sinusoidal command $r(t) = \sin(\omega t)$. This gives $\ddot{r} = -\omega^2 r$, $(p = 2)$, with $a_1 = 0$, $a_2 = -\omega_0^2$, (see Eq. (3.4)). The command error is $e = y_c - r$. The state space system using (3.19) is given by

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \xi \end{bmatrix}, \xi = \ddot{x} - \omega^2 x, \mu = \ddot{u} - \omega^2 u,$$
$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & C_c \\ 0 & 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ D_c \\ B \end{bmatrix} \tag{3.25}$$

Example 3.3 Constant Command Tracking in a Scalar System Knowledge from classical control tells us that a type 1 controller is needed to track a constant command. Using a scalar system, this example will build a state space model and illustrate how to design an integral control for tracking constant commands. Consider the following scalar system:

$$\dot{x} = -2x + u + w$$
$$y = x \tag{3.26}$$

where x is the state, u is the control, and w is a nonmeasurable constant disturbance. Hence, $A = [-2]$, $B = [1]$, $E = [1]$, $C = [1]$, and $D = [0]$. The goal is for the output y (same as the state x) to track a constant command r, with zero steady-state error. The constant command is modeled using (3.3) as

$$\dot{r} = 0, p = 1, a_1 = 0 \tag{3.27}$$

The robust servo design model is

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.28}$$

with

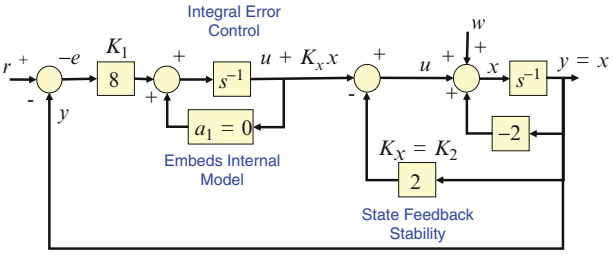


Fig. 3.1 Example 3.3 block diagram of the control and system dynamics

$$\tilde{A} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}; \tilde{B} = \begin{bmatrix} D \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.29}$$

in which we see that $(\tilde{A}, \tilde{B})$ form a controllable pair. The feedback control law is $\mu = -Kz$. It is desired that the closed-loop dynamics have a characteristic polynomial of $\phi_{\text{cl}}(s) = (s+2)^2 + 4 = s^2 + 4s + 8$ (pole placement problem). The feedback control is

$$\mu = -[K_1 \quad K_2] \begin{bmatrix} e \\ \dot{x} \end{bmatrix} \tag{3.30}$$

The closed-loop system is $\dot{z} = (A - BK)z$ with characteristic polynomial $\phi_{\text{cl}}(s) = \det[sI - \tilde{A} + \tilde{B}K]$. Substitute for $(\tilde{A}, \tilde{B})$ keeping the gains as parameters, expand the determinant, and equate to the desired closed-loop characteristic polynomial

$$\det[sI - \tilde{A} + \tilde{B}K] = s^2 + (2 + K_2)s + K_1 = s^2 + 4s + 8 \tag{3.31}$$

Equating coefficients of s yields two equations in the two unknown gains that can be solved for the gains $[K_1 \quad K_2] = [8 \quad 2]$. The control $u = \int \mu$ and is

$$\begin{aligned} u = -K \int z \, \mathrm{d}t &= -[8 \quad 2] \begin{bmatrix} \int e \, dt \\ x \end{bmatrix} \\ &= -8 \int e \, dt - 2x + \text{constant of integration} \end{aligned} \tag{3.32}$$

In the implementation, the constant of integration is ignored. Figure 3.1 illustrates the system, (controller, plant, and disturbance).

3.3 The Robust Servomechanism LQR

In Chap. 2, it was shown that the state feedback infinite-time linear quadratic regulator has excellent stability and robustness properties. In this section, this approach is applied to the servomechanism design model from the previous section

to form the robust servomechanism LQR (RSLQR). The RSLQR gain matrix K_c that is produced from the solution of the algebraic Riccati equation forms the state feedback control given as

$$\mu = -K_c z \tag{3.33}$$

which, when integrated p times, is implemented using integral control for command tracking and state feedback for stabilization.

The RSLQR design problem uses the servomechanism design model written as

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.34}$$

where z and μ are defined in (3.18) and (3.13), respectively. LQR control theory is applied to (3.34), using the performance index (PI),

$$J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) \mathrm{d}\tau \tag{3.35}$$

where $Q = Q^T \geq 0, R = R^T > 0$, $\left(\tilde{A}, \tilde{B}\right)$ is stabilizable, and $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ is detectable. For the infinite-time problem, the optimal steady-state control law for μ using state feedback is formed by solving the algebraic Riccati equation (ARE) using Q and R from (3.35), given as

$$P\tilde{A} + \tilde{A}^T P - P\tilde{B}R^{-1}\tilde{B}^T P + Q = 0 \tag{3.36}$$

The resulting steady-state $n_u \times (n_r + n_x)$-dimensional feedback controller gain matrix is

$$K_c = R^{-1}\tilde{B}^T P \tag{3.37}$$

with the state feedback control given as $\mu = -K_c z$. The gain matrix K_c is partitioned in the same manner as the vector z in (3.18), written as

$$K_c = \begin{bmatrix} K_p & K_{p-1} & \cdots & K_1 & K_x \end{bmatrix} \tag{3.38}$$

Substituting the definition of z into (3.33) yields

$$\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u} = -\sum_{i=1}^{p} K_i \overset{(p-i)}{e} - K_x \left[\overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x} \right] \tag{3.39}$$

Integrating (3.39) p-times gives the control solution u for the original system model in (3.2) as

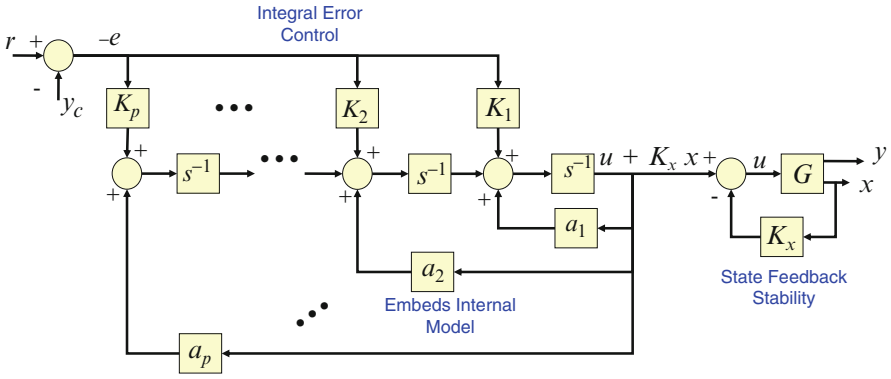


Fig. 3.2 Robust servomechanism block diagram

$$u = -K_x x + \sum_{i=1}^{p} s^{-i}\left(a_i\left(\overset{(p-i)}{u} + K_x \overset{(p-i)}{x}\right) - K_i \overset{(p-i)}{e}\right) \tag{3.40}$$

Figure 3.2 is a block diagram illustrating the system of (3.2) (represented as G) connected to the robust servomechanism state feedback control law.

The state feedback term $(-K_x x)$ enforces closed-loop stability of the plant. The p integrators and their gains provide integral error control, and the coefficients a_i embed the internal model of the signal being tracked. So, the closed-loop system is

$$\dot{z} = \left(\tilde{A} - \tilde{B}K_c\right)z + Fr \tag{3.41}$$

where $F = \left[-I_{n_u \times n_u} \quad 0_{n_u \times n_x}\right]^T$. The RSLQR closed-loop design using state feedback is guaranteed to be globally exponentially stable, and it will force the system-regulated output track the command signal $r(t)$, with zero steady-state error.

In Chap. 1, we introduced plant (1.35) and controller (1.36) state space models. These models were then coupled to form a closed-loop simulation model and loop gain frequency domain analysis models. We want to implement the RSLQR control from (3.40) using the controller given by

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{3.42}$$

The control in (3.40) is a state feedback control $(y = x)$. Substituting into (3.42), we have

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} x + B_{c2} r \\ u &= C_c x_c + D_{c1} x + D_{c2} r \end{aligned} \tag{3.43}$$

with

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \left[\begin{bmatrix} 0 & I_{n_r} & \cdots & 0 \\ \vdots & \ddots & \cdots & 0 \\ 0 & 0 & \cdots & I_{n_r} \\ a_p I_{n_r} - D_c K_p & \cdots & \cdots & a_1 I_{n_r} - D_c K_1 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_c - D_c K_x \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I_{n_r} \end{bmatrix} \right] \\ \begin{bmatrix} -K_p & \cdots & -K_2 & -K_1 \end{bmatrix} \quad [-K_x] \quad [0] \tag{3.44}$$

Example 3.4 The Robust Servo Controller for Example 3.3 In Example 3.3, the control u was given as

$$u = -8 \int e \, dt - 2x \tag{3.45}$$

The state space model for the controller using (3.44) is

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1] & [-1] \\ [-8] & [-2] & [0] \end{bmatrix} \tag{3.46}$$

In industrial applications, the commanded signal $r(t)$ is often assumed to be a constant. For example, in flight control, such a command could represent the stick force coming from a pilot or the guidance command coming from the outer-loop steering algorithms. Even though these command signals are not actually constant, designing and implementing a type 1 control system has proven very effective in most applications, and the RSLQR will provide zero steady-state error command tracking.

To achieve good transient response characteristics, tuning of the LQR PI matrices Q and R is required. Understanding how these matrices affect the control gains and how the control gains influence the closed-loop system response is key to achieving a good design.

It is important in the design of a realistic control system to be mindful of the "size" of the feedback gains in K_c. In aerospace applications, gains that are too large amplify sensor noise, drive the actuators with high rates, and cause issues and challenges with flexible body dynamics, called structural mode interaction. The feedback gains K_c depend upon the numerical values in Q and R. As $\|Q\|_2$ becomes large, the gains get large; as $\|R\|_2$ is made small, the gains get large; thus, $\|K_c\|_2 \sim \|Q\|_2 / \|R\|_2$.

3.3.1 Summary

Dynamics: $\dot{x} = Ax + Bu + Ew$

$y = Cx + Du$

Command model: $\overset{(p)}{r} = \sum_{i=1}^{p} a_i \overset{(p-i)}{r}$; Disturbance model: $\overset{(p)}{w} = \sum_{i=1}^{p} a_i \overset{(p-i)}{w}$

State model: $\xi = \overset{(p)}{x} - \sum_{i=1}^{p} a_i \overset{(p-i)}{x}$; Control model: $\mu = \overset{(p)}{u} - \sum_{i=1}^{p} a_i \overset{(p-i)}{u}$

Augmented state vector: $z = \begin{bmatrix} e & \dot{e} & \cdots & \overset{(p-1)}{e} & \xi \end{bmatrix}$

Performance index: $J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) d\tau \qquad Q = Q^T \geq 0, R = R^T > 0$

Control design model: $\dot{z} = \tilde{A}z + \tilde{B}\mu$; $\left(\tilde{A}, \tilde{B}\right)$ controllable. $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ detectable.

$$\tilde{A} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & 0 & I & 0 \\ \alpha_p I & \alpha_{p-1} I & \cdots & \alpha_2 I & \alpha_1 I & C \\ 0 & \cdots & \cdots & \cdots & 0 & A \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ D \\ B \end{bmatrix}$$

ARE: $P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P = 0 \qquad \mu = -R^{-1}\tilde{B}^T P z = -K_c \mathrm{z}$

Control: $u = -K_x x + \sum_{i=1}^{p} s^{-i} \left(a_i \left(\overset{(p-i)}{u} + K_x \overset{(p-i)}{x} \right) - K_i \overset{(p-i)}{e} \right)$

Controller: $\dot{x}_c = A_c x_c + B_{c1} x + B_{c2} r$

$u = C_c x_c + D_{c1} x + D_{c2} r$

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & I_{n_r} & \cdots & 0 \\ \vdots & \ddots & \cdots & 0 \\ 0 & 0 & \cdots & I_{n_r} \\ a_p I_{n_r} - D_c K_p & \cdots & \cdots & a_1 I_{n_r} - D_c K_1 \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_c - D_c K_x \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I_{n_r} \end{bmatrix} \\ \begin{bmatrix} -K_p & \cdots & -K_2 & -K_1 \end{bmatrix} & [-K_x] & [0] \end{bmatrix}$$

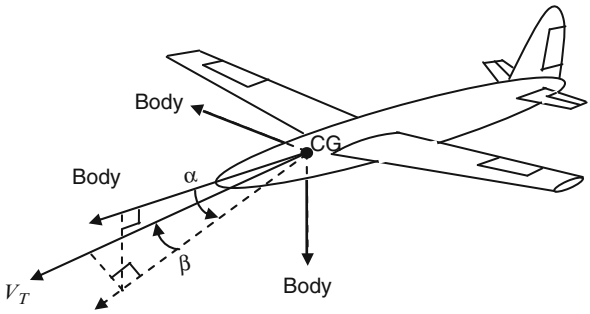


Fig. 3.3 Unmanned aircraft

The following example will illustrate how to choose parameters within Q and how to select a design that performs well, has a reasonable bandwidth, and does not result in high actuator rates. The processes for selecting the LQR penalty weights form LQR design charts that show important time domain and frequency domain metrics plotted versus loop gain crossover frequency. Viewing this information, while using the design charts, allows the control system engineer to select the desired bandwidth of the design and to perform the necessary trade studies required to meet the desired closed-loop system performance design goals. This process also prevents large feedback gains from being selected, which can introduce challenges later in the simulation and analysis of the control system.

Example 3.5 LQR Q-Matrix Parameter Selection Using Design Charts Consider the pitch-plane dynamics of an unmanned aircraft (Fig. 3.3), given as

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta + M_q q\end{aligned} \tag{3.47}$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant and will design an RSLQR controller with integral control. We will design a constant gain matrix K_c for a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration $A_z(\text{ft}/s^2)$ is given by

$$A_z = -V\dot{\gamma} = VZ_\alpha \alpha + VZ_\delta \delta \tag{3.48}$$

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate (3.48) to form the differential equation for $\dot{A}_z$ and then substitute for $\dot{\alpha}$ from (3.47). This produces

$$\begin{aligned}\dot{A}_z &= Z_\alpha A_z + VZ_\alpha q + VZ_\delta \dot{\delta}_e \\ \dot{q} &= \frac{M_\alpha}{VZ_\alpha}A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_e\end{aligned} \tag{3.49}$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_e = -2\zeta_a \omega_a \dot{\delta}_e + {\omega_a}^2(\delta_c - \delta_e) \tag{3.50}$$

Combining (3.49) and (3.50) forms the plant model written in matrix form as

$$\begin{bmatrix} \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha & VZ_\alpha & 0 & VZ_\delta \\ M_\alpha/VZ_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \delta_c \tag{3.51}$$

Since $r = \text{constant}$, $\dot{r} = 0$, and $p = 1$, then we need to add an integrator to form our type 1 controller. The state vector (Eq. 3.18) for the robust servomechanism design model is

$$z = \begin{bmatrix} e & \dot{x}^T \end{bmatrix}^T \tag{3.52}$$

with the design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ given as

$$\begin{bmatrix} \dot{e} \\ \ddot{A}_z \\ \ddot{q} \\ \dddot{\delta}_e \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha & \mathrm{VZ}_\alpha & 0 & \mathrm{VZ}_\delta \\ 0 & M_\alpha/\mathrm{VZ}_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \ddot{\delta}_e \\ \dddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \dot{\delta}_c \tag{3.53}$$

where $z = \begin{bmatrix} e & \dot{A}_z & \dot{q} & \ddot{\delta}_e & \dddot{\delta}_e \end{bmatrix}^T \in R^5$. At a flight condition of Mach 0.3, 5,000 ft altitude, and a trim angle-of-attack α of 5 degrees, the plant model data (stability and control derivatives) are

$$\begin{aligned} Z_\alpha &= -1.05273(1/\mathrm{s}) \\ Z_\delta &= -0.0343(1/\mathrm{s}) \\ M_\alpha &= -2.3294\left(1/\mathrm{s}^2\right) \\ M_q &= -1.03341\left(1/\mathrm{s}^2\right) \\ M_\delta &= -1.1684\left(1/\mathrm{s}^2\right) \\ V &= 329.127(\mathrm{ft/s}) \\ \omega_a &= 2\pi * 13.(\mathrm{rad/s}) \\ \zeta_a &= 0.6 \end{aligned} \tag{3.54}$$

Substituting the data into (3.53) yields

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -6672. & -98.02 \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6672. \end{bmatrix} \tag{3.55}$$

If we check the controllability of the pair $\left(\tilde{A}, \tilde{B}\right)$, we find the system to be controllable.

The objective in the design of the gain matrix is to track the acceleration command with zero error without using large gains. The design begins by equating $R = 1$ and selecting a Q matrix that penalizes the error state e in (3.53). Thus, the performance index in (3.35) is

$$J = \int_0^\infty \left(z^T Q z + \mu^2\right) \mathrm{d}\tau \tag{3.56}$$

We start by inserting the parameter q_{11} in the (1,1) element

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ & 0 & & 0 & \\ & & 0 & & \\ & 0 & & 0 & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{q} \\ A_z \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}, \tag{3.57}$$

and set the other matrix elements to zero. This will penalize the error in tracking the command. Substituting (3.57) into (3.56) gives the performance index as

$$J = \int_0^\infty \left(q_{11} e^2 + \mu^2\right) d\tau \tag{3.58}$$

If we check the observability of the pair $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$, we find the system to be observable through this choice of Q.

The LQR design charts are formed by sweeping q_{11} values from small to large, solving for the feedback gains for each value of q_{11}, and examining the closed-loop system properties. The computation steps are the following:

1. Set the value of q_{11} in Q from (3.57).
2. Solve the ARE in (3.36) for P.
3. Compute the feedback gain matrix K_c in (3.37).
4. Form the closed-loop system in (3.41).

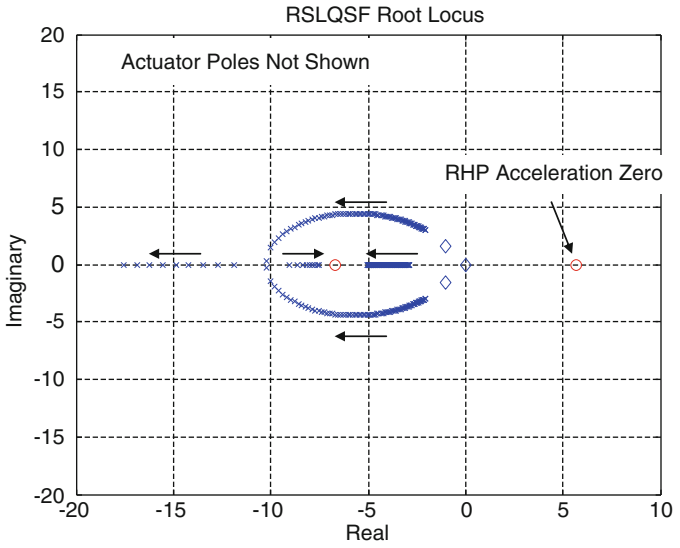


Fig. 3.4 RSLQR short-period dynamics root locus. Actuator poles at $-49.0 \pm 65.3j$ not shown

5. Simulate the closed-loop system to a step command and extract time domain performance metrics. These are rise time, settling time, percent command overshoot, percent command undershoot, max control, and control rate.
6. Evaluate the loop transfer function at the plant input and extract frequency domain metrics. These are loop gain crossover frequency, minimum singular values of $I + L$ and $I + L^{-1}$ (L at the plant input) versus frequency, and $\|S\|_\infty$ and $\|T\|_\infty$ for the commanded variable (S and T are formed using L at the plant output).
7. Loop back to step 1 and increase q_{11} until the numerical range is complete.

For this command tracking system, it is desired to track the acceleration command with zero error, and minimize the rise time and settling time, all in response to the command, without driving the control surface actuators with large gains. Large gains will cause large actuator deflections and rates, which are not desirable. This creates a trade study, in which the bandwidth must be limited in order not to exceed actuator limitations. Also, large gains amplify sensor noise, reduce stability margins, and make the system sensitive to unmodeled high-frequency dynamics (like flexible body modes).

For this flight condition, the range of the LQR penalty q_{11} is selected to be $q_{11} = \left[10^{-2}, \quad 10^{0.5}\right]$, using 100 design points. For a linear system, the response will depend upon the location of the closed-loop poles in the s-plane. Looping through the above calculations, the eigenvalues of the closed-loop system matrix $\left(\tilde{A} - \tilde{B}K_c\right)$ are plotted to form a root locus. The data are shown in Fig. 3.4.

Also plotted are the open-loop poles (diamonds), the commanded variable A_z/δ_e, and the system transfer function zeros, which include a nonminimum phase zero (right half plane (RHP) zero). The open-loop dynamics are stable at this flight condition, with the poles located in the left half plane (LHP). The two finite zeros of the acceleration transfer function are -6.73 and 5.69. As discussed in Chap. 2 on asymptotic properties of regulators and the root square locus, Fig. 3.4 shows the RHP zero at 5.69 mirrored into the LHP. Two of the closed-loop poles, one from the integrator and the other from the short period, are approaching this region on the negative real axis. The remaining short-period pole moves out to infinity along the negative real axis. The control actuator poles, not shown in the figure, move toward infinity along asymptotes at 45 degrees.

The time domain performance metrics of interest here are 63% rise time, 95% settling time, percent overshoot, percent undershoot (because the system is nonminimum phase), max actuator deflection, and max actuator rate in response to a constant step command. The frequency domain performance metrics are loop gain crossover frequency ω_c in Hz, the minimum of the minimum singular value of the return difference dynamics, denoted $\underline{\sigma}(I+L)$, and the minimum of the minimum singular value of the stability robustness matrix $I+L^{-1}$, denoted $\underline{\sigma}\left(I+L^{-1}\right)$. The metric $\underline{\sigma}(I+L) = 1/\|S\|_\infty$ and $\underline{\sigma}\left(I+L^{-1}\right) = 1/\|T\|_\infty$ (see Chap. 5, Sect. 5.2 for definitions). These metrics, plotted versus ω_c, are used to determine how the increasing bandwidth of the system affects the system characteristics, indicating a desired value for q_{11}.

As with most control system design procedures, there is not a single answer to determining a set of gains that are acceptable. It is for the designer to make a reasonable selection. Once a suitable design is chosen, the associated gain matrix K_c is then stored in a table to create a gain-scheduled control for real-time implementation. Figure 3.5 shows the rise time and settling time plotted against loop gain crossover frequency ω_c.

As ω_c increases, the system responds more quickly to the step command. As seen from the figure, there is a diminishing return in terms of speed of response as the bandwidth increases. This is also evident from the root locus in Fig. 3.4. As the dominant poles approach the zero locations at -6.73 and -5.69, the change in the pole location diminishes with the increasing gains. The poles headed toward infinity along the asymptotes continue to move, but their contribution to the response $\left(e^{\lambda t}\right)$ dies quickly as the eigenvalues get large and negative. This indicates that large gains are not needed to make the system respond quickly.

Figure 3.6 shows the percent overshoot, percent undershoot, max elevon (tail actuated control surface) deflection per g commanded, and max elevon rate per g commanded versus the loop gain crossover frequency ω_c.

At lower values of ω_c, the response slightly overshoots the command, causing an overshoot. Command overshoot in flight control systems needs to be minimized in order to maintain limits and placards on the aircraft. As the integrator gain increases (as q_{11} increases), above 2.1 Hz ω_c, the response has no command overshoot. This metric by itself indicates a desire for larger gains. The percent undershoot,

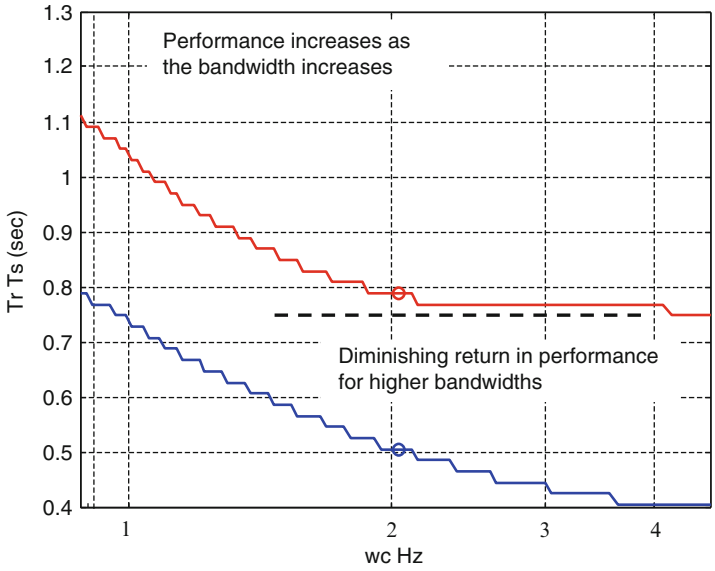


Fig. 3.5 Rise time (*blue*) and settling time (*red*) versus loop gain crossover frequency

characteristic of nonminimum phase responses, continues to increase with increasing ω_c. This response characteristic is undesirable and also needs to be minimized. Unfortunately, it increases with increasing ω_c. This metric indicates a desire for lower gains. Both the max deflection and max rate increase with increasing ω_c. It is critical in flight control systems not to have excessive deflections and rates in response to changes in the command. Electric actuators typically used in unmanned aircraft systems draw current proportional to the peak rate (at these normal operating conditions). High rates then cause significant power draw. Also, if the surface becomes rate saturated, this nonlinear effect can significantly degrade stability. As shown in the figure, the deflection and rate increase almost exponentially with increasing ω_c. These metrics also indicate a need for lower gains. As seen in this figure, some of the metrics tend toward increasing the gains, and some tend toward decreasing the gains.

Figure 3.7 shows two frequency response metrics: the minimum of the minimum singular value of the return difference dynamics $\underline{\sigma}(I + L)$ and the minimum of the minimum singular value of the stability robustness matrix $\underline{\sigma}\left(I + L^{-1}\right)$.

As is characteristic of LQR state feedback designs (discussed in Chap. 2), the $\underline{\sigma}(I + L)$ is equal to unity for all q_{11} design values. This metric is not particularly useful for developing state feedback designs but is critical when output feedback is used. The $\underline{\sigma}\left(I + L^{-1}\right)$, which is the inverse of the infinity norm of the complementary sensitivity function, is a measure of the damping in the dominant poles of the closed-loop system. We would like to maximize $\underline{\sigma}\left(I + L^{-1}\right)$. The figure shows that this metric tends to favor larger gains.

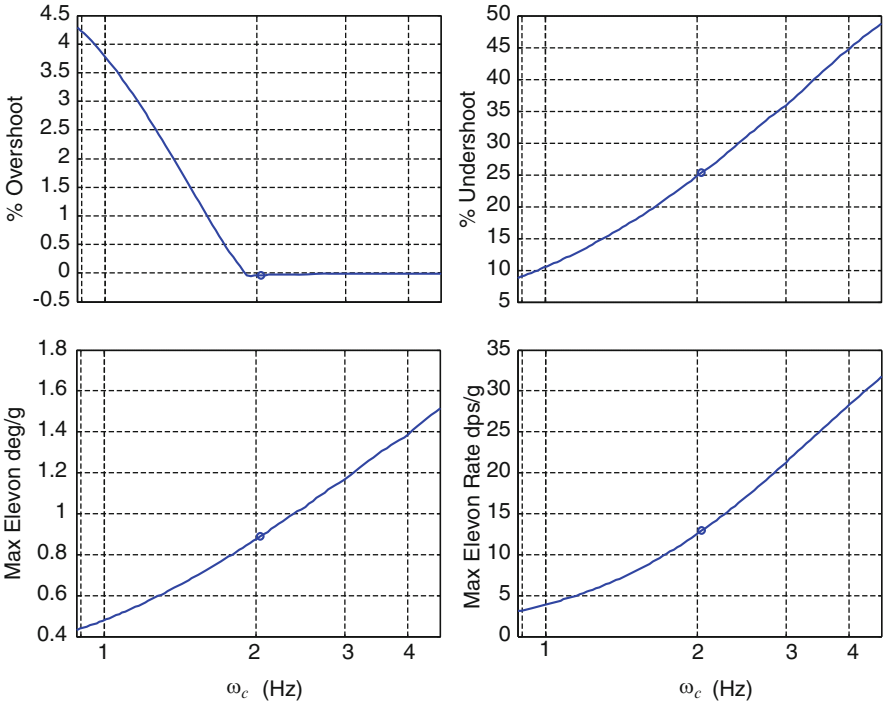


Fig. 3.6 Percent overshoot, undershoot, and max elevon deflection and rate versus loop gain crossover frequency ω_c

In balancing the positive and negative trends indicated by these metrics, a design condition $q_{11} = 0.2448$ was selected. This is the value of q_{11} where the percent overshoot first approaches zero. For this design condition, the states A_z, q, δ_e, and $\dot{\delta}_e$ are plotted versus time in Fig. 3.8.

Note that there is no overshoot to the unit command. For this approach flight condition, the response is quick, without the use of large gains.

The gain matrixK_c is

$$K_c = [0.4948\ 0.1790 - 14.0605\ 2.2089\ 0.0018] \tag{3.59}$$

The controller implementing this design is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{3.60}$$

where $y = \begin{bmatrix} A_z & q & \delta_e & \dot{\delta}_e \end{bmatrix}^T$, $r = A_{zc}$, and

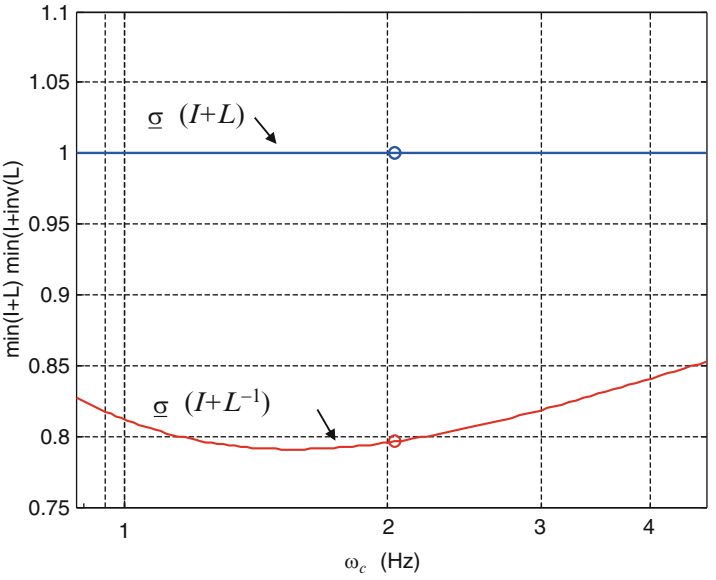


Fig. 3.7 Singular value frequency domain metrics versus loop gain crossover frequency

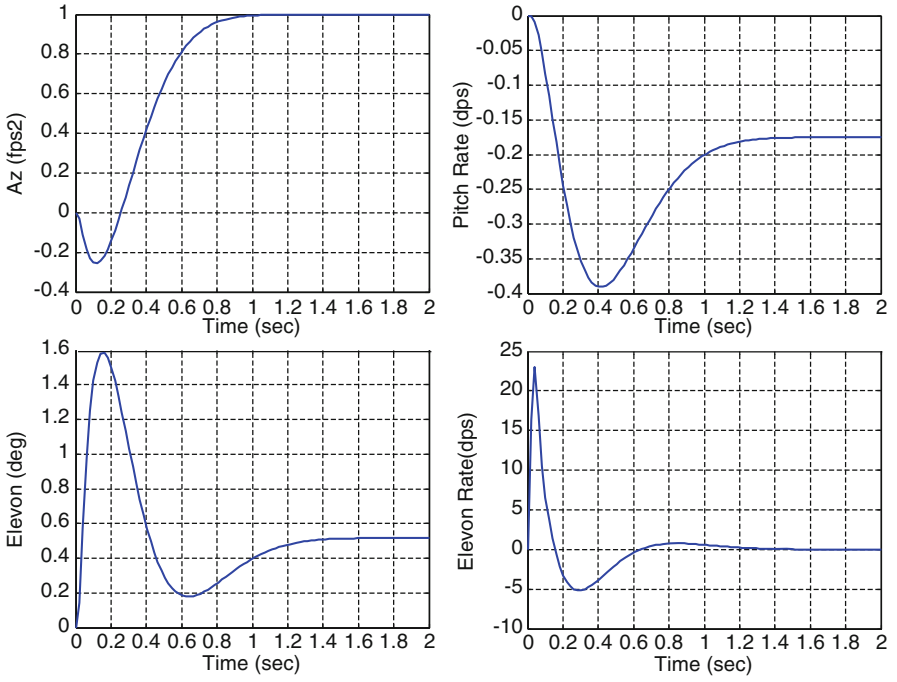


Fig. 3.8 States of the system responding to a unit acceleration step command

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1 \quad 0 \quad 0 \quad 0] & [-1] \\ [-0.4948] & [-0.179 \quad 14.0605 \quad -2.2089 \quad -0.0018] & [0] \end{bmatrix} \tag{3.61}$$

3.4 Conclusions

Depending upon the signal to be tracked, a certain number of integrators are needed to provide zero steady-state tracking error. In this chapter, we discussed how to formulate this problem within a state space framework and how to use optimal control to design the command tracking control system. In Chap. 2, we discussed the excellent frequency domain properties of LQR controllers. For our robust servomechanism controllers, we have these same excellent properties.

One of the key takeaways from the chapter should be the development of design charts for selecting numerical weights in optimal control problems. It is very easy to use too large of numerical weightings in the LQR performance index, and these large weights would lead to high gains. It is critical to be able to determine the bandwidth that is needed in the design to meet performance requirements and not to drive the control actuation system too hard.

3.5 Exercises

Exercise 3.1. A linearized suspended ball model is described by

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

(a) Use state feedback to stabilize the system producing closed-loop eigenvalues at $-1, -1/2$.
(b) The ball position x_1 can be measured using a photocell, but the velocity x_2 is more difficult to obtain. Suppose, therefore, that $y = x_1$. Design a full-order observer having poles at -4 and -5 and use the observer feedback to produce closed-loop eigenvalues at $-1/2, -1, -4, -5$.
(c) Repeat (b) using a first-order observer with pole at -6. Give a block diagram showing the controller as a single transfer function.
(d) Repeat this same design problem using the robust servo approach, obtaining integral control.

Exercise 3.2. Consider the design of a longitudinal (pitch-plane) autopilot. Using the robust servo formulation, design a pitch autopilot commanding a constant angle-of-attack α. Use the following dynamic model as the nominal plant model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{bmatrix} \delta_\mathrm{e}$$

and use data for $\frac{Z_\alpha}{V} = -1.21;\ \frac{Z_\delta}{V} = -0.1987; M_\alpha = 44.2506; (M_\delta = -97.2313)$.

(a) Design the autopilot to track a constant angle-of-attack command. Use the LQR approach outlined in Sect. 3.2.
(b) Design an autopilot to track a sinusoidal angle-of-attack command.

Exercise 3.3. Consider the longitudinal dynamics of a transport aircraft as given in Chap. 1, Exercise 1.2. Design a robust servo LQR control to track a constant speed command and a constant angle-of-attack command.

Exercise 3.4. Consider the lateral-directional dynamics of a transport aircraft as given in Chap. 1, Exercise 1.4. Design a robust servo LQR control to track a constant stability axis roll-rate p_s command (see Eq. (1.22)). Assume $\alpha_0 = 6\,\mathrm{deg}$.

References

1. Francis, B.A., Wonham, W.M.: The internal model principle of control theory. Automatica **12**(5), 457–465 (1976)
2. Davidson, E.J., Copeland, B.: Gain margin and time lag tolerance constraints applied to the stabilization problem and robust servomechanism problem. IEEE Trans. Autom. Control **30**(3), 229–239 (1985)

Chapter 4
State Feedback H_∞ Optimal Control

4.1 Introduction

Since control theory became an engineering discipline, mathematicians and engineers have searched for control system design methods that would simultaneously satisfy stability, performance, and robustness requirements in a single design step. In the 1980s, this problem was posed for multi-input multi-output (MIMO) systems and the design method called H_∞ optimal control emerged. This method allows the engineer to design, using state space models, a controller that satisfies important frequency domain requirements, often referred to as loop shaping. These requirements include shaping the sensitivity function $S(s)$, complementary sensitivity $T(s)$, the control activity $U(s)$, the loop gain $L(s)$, and its associated crossover frequency ω_c (rad/s).

In flight control systems for manned and unmanned aircraft configurations, robust performance and stability requirements necessitate the use of optimally designed flight control systems to achieve stability, to command tracking performance, to minimize control effort, and to be robust to inaccuracies in the model description. Robust performance requirements are generally driven by high maneuver rates needed for agile flight. Robust stability requirements are often related to large flight envelopes and uncertainties in the plant dynamics created by uncertain aerodynamics, actuation, and flexible body dynamics.

H_∞ optimal control allows the control system engineer to address these challenges in the design of the flight control system. The topology of a general H_∞ controller design problem is shown in Fig. 4.1. A state space model for the plant is

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew \\ z &= Cx + D_1 u + D_2 w \end{aligned} \tag{4.1}$$

where $x \in R^{n_x}$ is the state, $u \in R^{n_u}$ the control, $w \in R^{n_w}$ the exogenous disturbance, and $z \in R^{n_z}$ a collection of variables to be regulated. The design goal is to minimize the regulated variables z in response to the exogenous input w while providing

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_4,

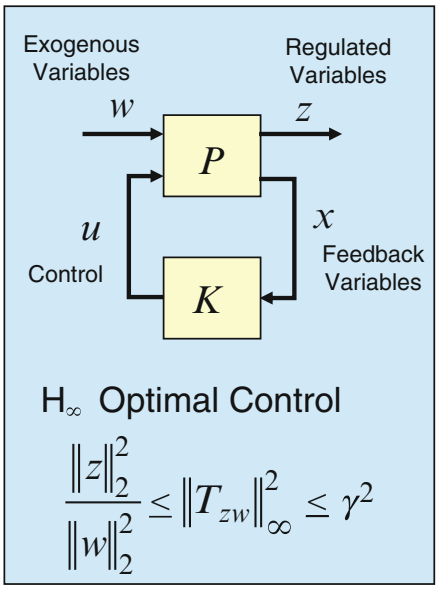


Fig. 4.1 H_∞ optimal control block diagram

internal stability. This is equivalent to minimizing the infinity norm of the transfer function matrix. If all the states are available for feedback, then the resulting H_∞ problem is referred to as a full information feedback problem. The solution results in a feedback compensator whose feedback gains are calculated by solving a single algebraic Riccati equation.

The state space solution of linear H_∞ optimal control problems can be found in Doyle et al. [1]. This same problem of reducing the H_∞ norm of a closed-loop system has been viewed as a two-person zero-sum differential game in Basar and Bernhard [2], where the solution is related to certain algebraic Riccati equations. This approach for nonlinear systems has been pursued in Basar and Bernhard [2] and in Helton [3]. For nonlinear systems, the Riccati equation is replaced with a particular Hamilton–Jacobi equation known as Isaacs equations ([4], p. 67, Eq. (4.2.1)). This type of optimal control is referred to as nonlinear H_∞ and/or $\mathcal{L}_2$-gain optimal control. A design example can be found in Wise and Sedwick [5].

We begin with a review of common norms for signal and systems, proceed to show how to engineer both stability and performance specifications in the frequency domain, and then demonstrate how to achieve loop shaping using frequency-dependent weights. The loop shaping ideas presented here are very similar to using lead-lag filters, low-pass filters, notch filters, etc., from classical control theory. If the reader is not familiar with frequency domain analysis, then Chap. 5 should be reviewed prior to working through this chapter. Once an understanding of how the state space design model is engineered, the full information state feedback controller is derived. A flight control design example using an unmanned aircraft pitch autopilot is presented to show how to implement the concepts of this chapter.

4.2 Norms for Signals and Systems

In control system design, we are usually concerned with the "size" of certain signals within the system. These signals may be commands, errors, states, outputs, or internal variables within the dynamics. Consider piecewise continuous scalar signals $u(t)$ which map $(-\infty, \infty)$ to $(-\infty, \infty)$. Define the 1-norm, 2-norm, and ∞-norm as

$$\begin{aligned} \|u\|_1 &= \int_{-\infty}^{\infty} |u(t)|dt \\ \|u\|_2 &= \left(\int_{-\infty}^{\infty} |u(t)|dt \right)^{\frac{1}{2}} \\ \|u\|_\infty &= \sup_t |u(t)| \end{aligned} \tag{4.2}$$

Suppose u is a current through a 1-ohm resistor. Then, the power is equal to u^2, and the total energy is the integral of u^2, which is the norm $\|u\|_2^2$.

4.2.1 Power Signals

The average power of a signal is the average of its instantaneous power. The average power of u is

$$\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u^2(t)dt \tag{4.3}$$

If this limit exists, then the signal is called a power signal, and we denote the limit as $pow(u)$, given as

$$pow(u) = \left(\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u^2(t)dt \right)^{\frac{1}{2}} \tag{4.4}$$

The $pow(u) = \left(\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u^2(t)dt \right)^{\frac{1}{2}}$ is not a norm. It does not satisfy the axiom that $\|u\| = 0 \rightarrow u(t) = 0$ for all $t \in (-\infty, \infty)$. Nonzero signals can have zero average power.

Example 4.1 If $\|u\|_2<\infty$, then $pow(u) = 0$. Consider $\|u\|_2<\infty$, then

$$\frac{1}{2T}\int_{-T}^{T} u^2(t)dt \leq \underbrace{\frac{1}{2T}}_{\to 0}\underbrace{\|u\|_2^2}_{T\to\infty} \tag{4.5}$$

Since $\|u\|_2^2<\infty$, taking the limit $T \to \infty$ yields $pow(u) = 0$.

Example 4.2 If $\|u\|_\infty<\infty$, then $pow(u) \leq \|u\|_\infty$. Using (4.5), we get

$$\frac{1}{2T}\int_{-T}^{T} u^2(t)dt \leq \frac{1}{2T}\int_{-T}^{T} \|u\|_\infty^2 dt = \|u\|_\infty^2 \frac{1}{2T}\int_{-T}^{T} dt = \|u\|_\infty^2 \tag{4.6}$$

4.2.2 Norms for Systems

Consider the norms for stable scalar transfer functions in which

$$\frac{1}{2T}\int_{-T}^{T} u^2(t)dt \leq \frac{1}{2T}\int_{-T}^{T} \|u\|_\infty^2 dt = \|u\|_\infty^2 \frac{1}{2T}\int_{-T}^{T} dt = \|u\|_\infty^2$$

The convolution response of the system is

$$y = G^* u \quad \to \quad y(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)d\tau \tag{4.7}$$

Typical terms for the transfer function are:

- G stable $\to$ that G is analytic in the closed RHP (Re $s \geq 0$).
- G proper $\to$ $G(j\omega)$ is finite (order of the denominator $\geq$ order of numerator).
- G strictly proper $\to$ $G(j\infty) = 0$ (order of denominator $>$ order of numerator).
- $G(j\infty) = 0$ biproper $\to$ G and G^{-1} are both proper.

From Parseval's theorem, for a stable G, we have

$$\|G\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega\right)^{\frac{1}{2}} = \left(\int_{-\infty}^{\infty} g^2(t)dt\right)^{\frac{1}{2}} = \|g\|_2 \tag{4.8}$$

Table 4.1 Output signal norms for stable transfer functions with specific input signals

	$u(t) = \delta(t)$	$u(t) = \sin(\omega t)$
$\|\|y\|\|_2$	$\|\|G\|\|_2$	∞
$\|\|y\|\|_\infty$	$\|\|G\|\|_\infty$	$\|G(j\omega)\|$
$pow(y)$	0	$\frac{1}{\sqrt{2}}\|G(j\omega)\|$

For a stable G, $\|G\|_2$ is finite if and only if G is strictly proper with no poles on the $j\omega$ axis. For a strictly proper G with no poles on the $j\omega$ axis, the $\|G\|_2^2$ can be expressed as

$$\|G\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} G(-s)G(s)ds = \frac{1}{2\pi j}\oint G(-s)G(s)ds \tag{4.9}$$

which can be evaluated using residues of the transfer function. $\|G\|_\infty$ is the peak of the Bode plot of G. $\|G\|_\infty$ is finite if and only if G is proper with no poles on $j\omega$ axis. Also, there is a sub-multiplicative property of the ∞-norm: $\|GH\|_\infty \leq \|G\|_\infty \|H\|_\infty$, which allows us to bound the combined system via norms on its elements. The above norms for signals and systems allow us to form and understand the amplification or attenuation in the responses of systems and signals of interest.

If we know how big the input signal u is, how big will the output y be? The table below illustrates this norm relationship for a stable strictly proper G (Table 4.1).

Example 4.3 The (1,1) table entry above is formed as follows:

$$y(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)d\tau = \int_{-\infty}^{\infty} g(t-\tau)\delta(\tau)d\tau = \int_{-\infty}^{\infty} g(t)d\tau = g(t)$$
$$\|y\|_2 = \|G\|_2 = \|g\|_2 \tag{4.10}$$

Suppose u is not fixed as in the above table but can be any signal with 2-norm ≤ 1. The result is often called the system gain and is equal to $\|G\|_\infty$. The following table illustrates the response for finite 2-norm, ∞-norm, and *pow* signals (Table 4.2):

Table 4.2 Output signal norms for stable transfer functions with specific input signals

	$\|u\|_2$	$\|u\|_\infty$	$pow(u)$
$\|y\|_2$	$\|G\|_\infty$	∞	∞
$\|y\|_\infty$	$\|G\|_2$	$\|G\|_1$	∞
$pow(y)$	0	$\leq \|G\|_\infty$	$\|G\|_\infty$

Example 4.4 The (1,1) table entry above is formed as follows. For $\|u\|_2 \leq 1$, we want $\|y\|_2$:

$$\|y\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 |u(j\omega)|^2 d\omega \leq \|G\|_\infty^2 \frac{1}{2\pi}\int_{-\infty}^{\infty} |u(j\omega)|^2 d\omega = \|G\|_\infty^2 \|u\|_2^2 \tag{4.11}$$

4.2.3 Computing Norms for Systems

For single-input single-output linear-time-invariant systems

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= cx \end{aligned} \tag{4.12}$$

the transfer function is $G(s) = c(sI - A)^{-1}b$. If the system matrix A is stable, the matrix exponential

$$e^{At} = I + tA + \tfrac{t^2}{2!}A^2 + \cdots \tag{4.13}$$

converges uniformly in time. Let

$$P = \int_0^{\infty} e^{A\tau} b b^T e^{A^T \tau} d\tau \tag{4.14}$$

then

$$AP + PA^T + bb^T = 0 \tag{4.15}$$

and the 2-norm of G is given by

$$\|G\|_2 = \left(cPc^T\right)^{1/2} \tag{4.16}$$

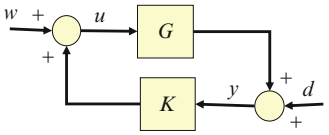


Fig. 4.2 Block diagram of a linear closed-loop system

Proof.

$$G(t) = ce^{At}b$$

$$\|G\|_2^2 = \int_0^\infty ce^{A\tau}bb^T e^{A^T\tau}c^T d\tau = c\int_0^\infty e^{A\tau}\mathrm{bb}^T e^{A^T\tau}d\tau c^T = cPc^T \tag{4.17}$$

4.2.4 Well-Posedness and Stability

Consider the system interconnection shown in the block diagram of Fig. 4.2.

The loop equations for the system shown in Fig. 4.2 are

$$\begin{aligned} w &= u - Ky \\ d &= y - Gu \end{aligned} \tag{4.18}$$

Arranging inputs and outputs into a vector yields

$$\begin{bmatrix} w \\ d \end{bmatrix} = \begin{bmatrix} I & -K \\ -G & I \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} \tag{4.19}$$

Solving for the outputs yields

$$\begin{bmatrix} u \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1}}_{H} \begin{bmatrix} w \\ d \end{bmatrix} \tag{4.20}$$

Suppose G and K are proper, and let H denote the closed-loop transfer function matrix. In this case, the feedback system is well posed if and only if

$$\det(I - G(\infty)K(\infty)) \neq 0$$

The system is internally stable if and only if H is stable.

Proof.

$$\det\begin{bmatrix} I & -K \\ -G & I \end{bmatrix} = \det\begin{bmatrix} I & 0 \\ -G & I - GK \end{bmatrix} \det\begin{bmatrix} I & -K \\ 0 & I \end{bmatrix} = \det[I - GK] \tag{4.21}$$

So, it is easy to see that the closed-loop transfer function H will be proper if and only if $\det(I - G(\infty)K(\infty)) \neq 0$.

4.3 Stability and Performance Specifications in the Frequency Domain

Classical control system design methods using transfer functions were focused around achieving certain frequency response characteristics, viewed from either Bode, Nyquist, or Nichols charts. For single-input single-output systems, these classical methods easily incorporated command tracking performance, stability margin, and plant roll-off design features that achieved the desired system response characteristics. When control system design using state space methods was introduced to address multi-input multi-output control design challenges, one of the complaints raised about the design methods was the lack of focus, or attention, toward achieving frequency domain properties. In this section, we will discuss how to achieve frequency response design goals within a state space format. Readers not familiar with frequency response analysis should review Chap. 5 before proceeding into designing controllers using H_∞ optimal control.

Consider the control system shown in Fig. 4.3. For this system loop transfer function, the loop gain at the plant input $L(s) = K(s)G(s)$ is a square matrix that has dimension equal to the number of inputs for the system, with units equal to those variables in the control vector u. Figure 4.4 illustrates frequency domain requirements for $L(s)$.

In order to track commands at low frequency, the loop gain must have sufficient magnitude. In order to be robust to high-frequency noise and unmodeled high-frequency dynamics, the loop gain must roll off and be sufficiently small. In the frequency band between these conflicting requirements is where the loop gain crosses 0 dB and defines the loop gain crossover frequency, ω_c. As illustrated in the figure, the singular values of $L(s)$ are $\sigma_i(L)$, with $\underline{\sigma}(L)$ and $\bar{\sigma}(L)$ denoting the minimum and maximum, respectively. We refer to the frequency at which $\bar{\sigma}(L)$ crosses 0 dB as the loop gain crossover frequency ω_c.

For SISO systems, $L(s)$ is a scalar with singular value $\underline{\sigma}(L) = \bar{\sigma}(L) = |L|$. Large gain at low frequencies would be obtained by using a large proportional gain and/or integral control (type 1 control). Roll-off at high frequencies would be obtained by using low-pass or elliptical filters, depending upon the amount of roll-off needed.

Stability margins would be computed from $L(s)$ to indicate the robustness of the design. For MIMO systems, singular value margins are computed from the sensitivity $S(s)$ and complementary sensitivity $T(s)$.

Command tracking performance can also be viewed by examining the sensitivity function $S(s)$, given by

$$e(s) = (I + L(s))^{-1} r(s) = S(s)r(s) \tag{4.22}$$

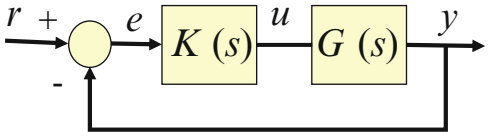


Fig. 4.3 Feedback control system

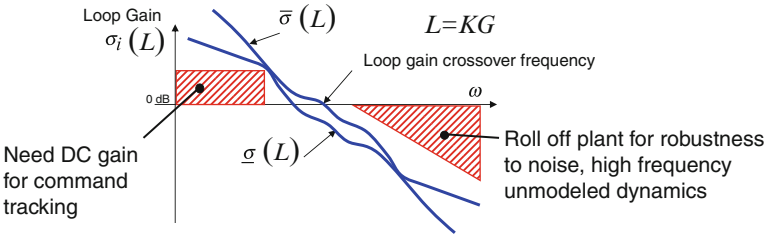


Fig. 4.4 Singular value frequency response requirements for a loop transfer function

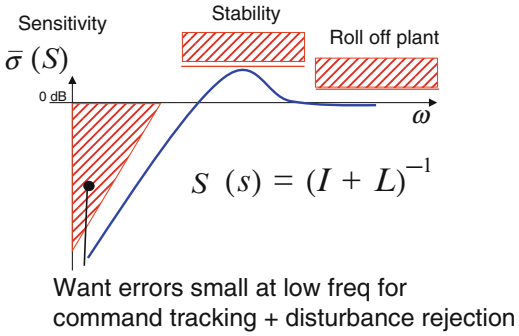


Fig. 4.5 Singular value frequency response requirements for the sensitivity function

Figure 4.5 illustrates singular value frequency response requirements for the sensitivity function $S(s)$.

At low frequency, where $L(s)$ needs to be large, $S(s)$ needs to be small. At high frequencies where $L(s)$ needs to be small, $S(s)$ is near unity. From Chap. 5, we know that stability margins are determined from the near singularity of the return difference $\left(I + L = S^{-1}\right)$ as measured by its minimum of $\underline{\sigma}(I + L)$ versus frequency, which equates to the peak of the sensitivity, or $\|S\|_\infty$. We know from the Bode integral log theorem that as we push the sensitivity lower in magnitude to achieve faster response, the peak pops creating a system that is more sensitive and less stable.

The complementary sensitivity $T(s)$ is defined as

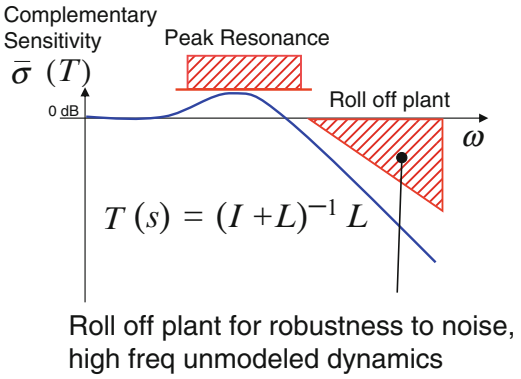


Fig. 4.6 Singular value frequency response requirements for the complementary sensitivity function

$$y(s) = (I + L)^{-1} L r(s) = T(s) r(s) \tag{4.23}$$

Figure 4.6 illustrates the complementary sensitivity function $T(s)$ singular value frequency response requirements.

This is also known as the closed-loop transfer function. At low frequencies where the loop gain is large, $T(s)$ is near unity. At high frequencies where the loop gain must roll off to be robust to high-frequency noise and unmodeled dynamics, $T(s)$ is small. At the peak of $T(s)$, $\|T\|_\infty$, the closed-loop system has a resonance in which frequencies at the peak are amplified by the system. If we were to approximate the system with an equivalent second-order system, the peak indicates low damping. This indicates that the dominant poles of the system are close to the $j\omega$ axis.

The control activity is amount of control used in responding to commands and rejecting disturbances. In general, it is desirable to minimize control usage at all frequencies, making sure that the actuators responding to the control signals are not position or rate saturated. In the frequency domain, constant weight on penalizing the control activity is usually used.

4.4 Loop Shaping Using Frequency-Dependent Weights

Figures 4.4, 4.5, and 4.6 show how to shape the loop gain, sensitivity, and complementary sensitivity to achieve command tracking performance, robustness to high-frequency noise, and unmodeled dynamics, as well as acceptable stability margins. These concepts are central to designing H_∞ optimal controllers. The design procedure minimizes the ∞-norm of a system response matrix. This response matrix contains a frequency-weighted sensitivity, complementary sensitivity, and control activity. The procedure for building a state space design model is straightforward.

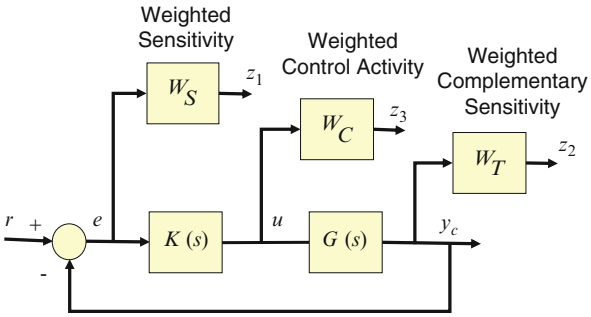


Fig. 4.7 Block diagram showing weighted sensitivity, control activity, and complementary sensitivity

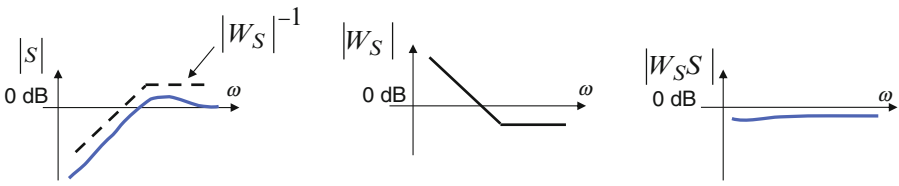


Fig. 4.8 Process for shaping the sensitivity function through a shaping filter

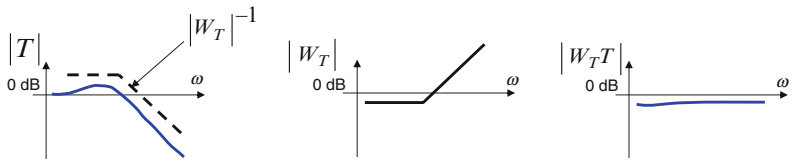


Fig. 4.9 Process for shaping the complementary sensitivity function through a shaping filter

The weighting filters used to shape the loops in the H_∞ optimal control design should be selected to be minimum order. Each state in the weighting filters adds a state to the controller. In gain-scheduled flight control applications, high-order controllers can introduce transients in the response as the scheduling variables change. Thus, low-order controllers are typically desirable.

Consider the block diagram shown in Fig. 4.7 for the plant model in (4.1). The scalar variable z_1 is a weighted error variable to be regulated. The idea is to weight the error response to a command, $e = Sr$, with a weighting filter W_s that is the inverse of the desired shape for S, so that when the $\|W_S S\|_\infty$ is minimized, it will shape S. Figure 4.8 illustrates this design process.

The second regulated variable in Fig. 4.7 is z_2 which is the weighted complementary sensitivity. Figure 4.9 illustrates the design process for shaping the complementary sensitivity.

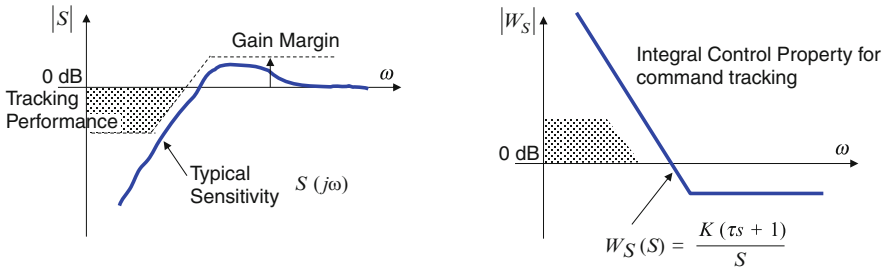


Fig. 4.10 First-order sensitivity weighting filter design

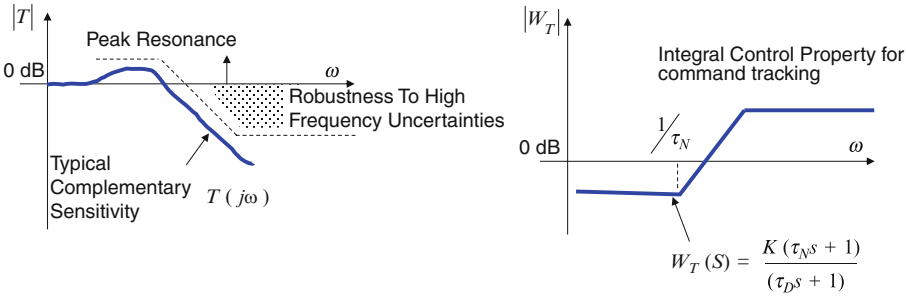


Fig. 4.11 First-order complementary sensitivity weighting filter design

The third regulated variable in Fig. 4.7 is z_3 which is a weighted control activity. This variable is multiplied here by a constant to penalize control activity at all frequencies. If it was needed to penalize some frequencies more than others, the process for selecting the weighting filter would be similar to that shown in Figs. 4.8 and 4.9.

The numerical choice of weighting filters will change with each application. Designing these filters is often difficult, with the degree of difficulty being comparable to selecting lead-lag filters in classical control design for improving gain or phase margins. Figure 4.10 illustrates a typical sensitivity frequency response with low-frequency command tracking and stability margin requirements and a typical first-order weighting filter for achieving the shape. The low-frequency behavior of W_s demonstrates an integrator property with a slope of 20 dB per decade. The gain K in W_s is chosen to produce a magnitude of W_s of -3 dB at the desired loop gain crossover frequency ω_c. The zero in W_s is chosen at the desired ω_c. The -3 dB magnitude will limit the peak of S $\left(\|S\|_\infty\right)$, thus producing adequate stability margins. These design rules and model for W_s can be used to shape S and keep the order of the weighting filter to a low number.

Figure 4.11 illustrates a typical complementary sensitivity frequency response which constrains the peak (peak resonance) and adds roll-off for robustness to uncertain and unmodeled high-frequency dynamics. The first-order weighting filter

used here is similar in shape to a lead-lag filter. The low-frequency behavior of W_T demonstrates a flat profile versus frequency which will constrain the peak resonance. The zero in W_T can be chosen smaller than the desired ω_c, with the gain K chosen so that the $|W_T|$ is 0 dB at ω_c. Some iteration of this may be needed to converge and obtain a desirable constraint for the peak resonance. The pole in W_T is chosen somewhat arbitrarily and should be chosen high enough in frequency to provide the minimum attenuation needed in the high-frequency range. One should not make it too high in frequency so as to keep the digital implementation (via a computer) reasonable.

4.5 State Feedback H_∞ Optimal Control

In this section, the state feedback control law is synthesized using an algebraic Riccati equation approach called γ-iteration. Consider the following linear-time-invariant model

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew \\ z &= Cx + D_1 u + D_2 w \end{aligned} \tag{4.24}$$

and cost function

$$J(u,w) = \frac{1}{2}\int_{t_0}^{T} \left(z^T z - \gamma^2 w^T w\right) d\tau \tag{4.25}$$

where $\gamma \geq 0$ and with t_0, x_0 given and T, $x(T)$ free. Our goal is to find the optimal control (minimizing control) u^* and maximizing disturbance w^* such that

$$J(u^*, w) \leq J(u^*, w^*) \leq J(u, w^*) \tag{4.26}$$

Examine the response z in (4.24) from the exogenous variable w

$$\|z\|_{RMS}^2 \leq \|T_{zw}\|_\infty^2 \|w\|_{RMS}^2 \tag{4.27}$$

where T_{zw} is the closed-loop transfer function model from w to z. Choose a positive γ such that $\gamma \geq \|T_{zw}\|_\infty$. Then, from (4.27),

$$\|z\|_{RMS} - \gamma^2 \|w\|_{RMS} \leq 0 \tag{4.28}$$

Using this γ, substitute (4.24) into (4.25) to obtain

$$\begin{aligned} J(u,w) &= \frac{1}{2}\int_{t_0}^{T}\left(z^T z - \gamma^2 w^T w\right)d\tau \\ &= \frac{1}{2}\int_{t_0}^{T}\left([Cx + D_1 u + D_2 w]^T [Cx + D_1 u + D_2 w] - \gamma^2 w^T w\right)d\tau \\ &= \frac{1}{2}\int_{t_0}^{T}\left(\begin{array}{l} x^T C^T C x + 2x^T \begin{bmatrix} C^T D_1 & C^T D_2 \end{bmatrix}\begin{bmatrix} u \\ w \end{bmatrix} + \\ \quad \begin{bmatrix} u \\ w \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}\begin{bmatrix} u \\ w \end{bmatrix} \end{array}\right)d\tau \end{aligned} \tag{4.29}$$

Let

$$S = \begin{bmatrix} C^T D_1 & C^T D_2 \end{bmatrix}, R = \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}, \tilde{u} = \begin{bmatrix} u \\ w \end{bmatrix} \tag{4.30}$$

Then, (4.29) becomes

$$J(u,w) = \frac{1}{2}\int_{t_0}^{T}\left(x^T C^T C x + 2x^T S\tilde{u} + \tilde{u}^T R\tilde{u}\right)d\tau \tag{4.31}$$

which is an LQR problem that has cross terms between the state x and extended control $\tilde{u}$. Next, rewrite the plant model in (4.24) using the extended control as

$$\dot{x} = Ax + \tilde{B}\tilde{u} \tag{4.32}$$

where $\tilde{B} = \begin{bmatrix} B & E \end{bmatrix}$. We can write the Hamiltonian for this LQR problem as

$$H = \frac{1}{2}\left(x^T C^T C x + 2x^T S\tilde{u} + \tilde{u}^T R\tilde{u}\right) + p^T\left(Ax + \tilde{B}\tilde{u}\right) \tag{4.33}$$

The necessary condition for the optimal control $\tilde{u}^*$ is

$$\nabla H_{\tilde{u}} = 0 = R\tilde{u} + S^T x + \tilde{B}^T \tilde{u} \tag{4.34}$$

Solving for the optimal $\tilde{u}^*$ gives

$$\tilde{u}^* = -R^{-1}\left(S^T x + \tilde{B}^T p\right) \tag{4.35}$$

The differential equation for the costate is

$$\dot{p} = -\nabla H_x = -C^T C x - A^T p - S\tilde{u} \tag{4.36}$$

with $p(T) = 0$. Substituting (4.35) into (4.32) and combining with (4.36), we can write the Hamiltonian system as

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -\tilde{B}R^{-1}\tilde{B}^T \\ -C^T C + SR^{-1}S & -A^T + SR\tilde{B}^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \tag{4.37}$$

The next step is to manipulate this first-order differential equation to eliminate the costate p and create a Riccati equation whose solution will give (4.35). The solution to (4.37) is derived from the state-transition matrix. Assume the state-transition matrix for (4.37) is

$$\Phi(T,t) = \begin{bmatrix} \phi_{xx}(T,t) & \phi_{xp}(T,t) \\ \phi_{px}(T,t) & \phi_{pp}(T,t) \end{bmatrix} \tag{4.38}$$

Then, $p(T) = \phi_{px}x + \phi_{pp}p$. Solving for p yields

$$p = \underbrace{\phi_{pp}^{-1}\phi_{px}}_{P} x = Px \tag{4.39}$$

Differentiating results in

$$\dot{p} = \dot{P}x + P\dot{x} \tag{4.40}$$

From (4.37), we have

$$\dot{P}x + P\dot{x} = \left(-C^T C + SR^{-1}S\right)x + \left(-A^T + SR\tilde{B}^T\right)p \tag{4.41}$$

Substituting for $\dot{x}$ using (4.32) and replacing p using (4.39) and factoring out x on the right yields the Riccati equation

$$-\dot{P} = PA + A^T P + C^T C - \left[P\tilde{B} + S\right]R^{-1}\left[\tilde{B}^T P + S^T\right] \tag{4.42}$$

whose solution P is used to form the state feedback control law as

$$\tilde{u} = -R^{-1}\left(B^T P + S^T\right)x \tag{4.43}$$

For the infinite time problem, (4.42) becomes an algebraic Riccati equation (ARE). The ARE is used in most applications. The following theorem summarizes the assumptions needed for the problem to be well-posed.

Theorem 4.1 [1] *Consider the linear-time-invariant system described in (4.24) where* $x \in R^{n_x}$, $u \in R^{n_u}$, $w \in R^{n_w}$, *and* $z \in R^{n_z}$. *Assume:*

1. (A, B, C, D_1) *has no zeros on the* $j\omega$ *axis.*
2. (A, B) *stabilizable.*
3. D_1 *is injective* $\left(\left(D_1^T D_1\right)^{-1} \text{ exists}\right)$.

Then, the following statements are equivalent:

1. There exists a state feedback control $\tilde{u} = -K_\infty x$ such that the closed-loop system is internally stable and $\|T_{zw}\|_\infty < \gamma$.
2. $D_2^T D_2 < \gamma^2 I$ and there exists a $P \geq 0$ that solves the following ARE:

$$PA + A^T P + C^T C - \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix} = 0 \tag{4.44}$$

and the optimal control u *is*

$$u = [I_{n_u} \quad 0]\tilde{u} = -[I_{n_u} \quad 0]R^{-1}\left(B^T P + S^T\right)x = -K_\infty x \tag{4.45}$$

4.6 Controller Design Using γ-Iteration

In this section, we build a control design model that embeds the sensitivity, complementary sensitivity, and control activity weighting filters from Sect. 4.3 into a state space model and then solves for the state feedback gain matrix (4.45) using a method called γ-iteration. The design model needs to be of the form of (4.24). Define the regulated variables in vector z to comprise sensitivity, complementary sensitivity, and control activity variables.

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

z_1 ← Sensitivity– to track commands.

z_2 ← Complementary Sensitivity – to roll-off plant, limit bandwidth.

z_3 ← Control Activity – minimize control usage.

From Sect. 4.3, the weighting filter W_s should be designed to be the inverse of the desired loop shape for $S(s)$, the weighting filter W_T should be designed to be the inverse of the desired loop shape for $T(s)$, and the control activity penalty to penalize control activity in the desired frequency range. To build the H_∞-controller

state space design model, the plant and weighting filters all need to be represented in a state space format. The plant dynamics are modeled as

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u \end{aligned} \tag{4.46}$$

where y is the variable to be commanded. The sensitivity weighting filter W_s is modeled as

$$\begin{aligned} \dot{x}_s &= A_S x_s + B_S(y - r) \\ z_1 &= C_S x_s + D_S(y - r) \end{aligned} \tag{4.47}$$

where r is the command. The variable z_1 is the weighted sensitivity. The complementary sensitivity weighting filter W_T is modeled as

$$\begin{aligned} \dot{x}_T &= A_T x_T + B_T y \\ z_2 &= C_T x_T + D_T y \end{aligned} \tag{4.48}$$

where the variable z_2 is the weighted complementary sensitivity. The control activity model must be selected in such a way to satisfy the requirement that D_1 matrix from (4.24) is injective. How to select this variable is demonstrated in *Example 4.5*.

The γ-iteration algorithm used here is summarized in the following five steps:

Algorithm 4.1 H_∞ Control γ-Iteration Method

1. *Pick a starting γ larger than what is anticipated as the optimal γ. This will start the binary search used to converge to the optimal value.*
2. *Form the LQR matrices using γ from (4.31).*
3. *Solve the algebraic Riccati Eq. (4.44) for the matrix P.*
4. *Check that $P > 0$ and that $Re(\lambda(A_{CL})) < 0$. If these tests pass, reduce γ, and go back to step 2. If the test fail, increase γ, and go back to step 2. A minimum step size needs to be established and used to determine when γ has converged to γ_{min}.*
5. *Once the bisection search has converged to a γ_{min}, form the feedback control using (4.45).*

When using the above process, care must be exercised as γ approaches $\gamma_{\min}$. It is typical that the R matrix in (4.44) becomes ill-conditioned as γ approaches $\gamma_{\min}$. The ARE solvers in most commercial tools are sensitive to this, and the resulting $P > 0$ matrix actually does not solve the ARE. This is easily tested by forming (4.44) and computing the norm on the result. The result should be a zero matrix, with 2-norm less than 10^{-5}. We have found that once the algorithm has converged to $\gamma_{\min}$, it is prudent to increase from the $\gamma_{\min}$ value slightly to reduce the feedback gain magnitudes and improve the accuracy of the solution to the ARE. We will demonstrate this in the upcoming example.

4.6.1 Summary

Dynamics: $\dot{x} = Ax + Bu + Ew$

$z = Cx + D_1 u + D_2 w$

Performance index: $J = \frac{1}{2}\int_0^\infty \left(x^T C^T C x + 2x^T S\tilde{u} + \tilde{u}^T R\tilde{u}\right) d\tau$

$$S = \begin{bmatrix} C^T D_1 & C^T D_2 \end{bmatrix}, R = \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}, \tilde{u} = \begin{bmatrix} u \\ w \end{bmatrix}$$

Controller Design

1. Pick a starting γ
2. Form LQR matrices
3. Solve

$$PA + A^T P + C^T C - \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix} = 0$$

4. Check $P > 0$. Compute feedback gains: $K_\infty = [I_{n_u} \quad 0] R^{-1}\left(B^T P + S^T\right)$

Check eigenvalues of closed loop system are stable: $\mathrm{Re}(\lambda(A - BK_\infty)) < 0$

5. Decrease γ until $\gamma_{\min}$

Example 4.5 H_∞ Flight Control Design Consider the design of the longitudinal flight control system for the unmanned aircraft shown in Fig. 4.12.

The pitch-plane dynamics are given as

$$\begin{aligned} \dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta + M_q q \end{aligned} \tag{4.49}$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant, and we will design a H_∞ controller using full state feedback. The feedback control law will consist of a constant gain matrix K_c at a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration $A_z(ft/s^2)$ is given by

$$A_z = -V\dot{\gamma} = Z_\alpha \alpha + Z_\delta \delta \tag{4.50}$$

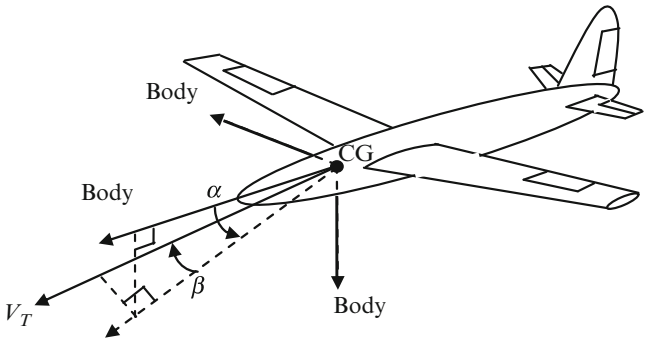


Fig. 4.12 Unmanned aircraft

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate Eq. (4.50) to form the differential equation for $\dot{A}_z$ and then substitute for $\dot{\alpha}$ from Eq. (4.49). This produces

$$\begin{aligned}\dot{A}_z &= Z_\alpha A_z + VZ_\alpha q + VZ_\delta \dot{\delta}_\mathrm{e} \\ \dot{q} &= \frac{M_\alpha}{VZ_\alpha} A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_\mathrm{e}\end{aligned} \tag{4.51}$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_\mathrm{e} = -2\zeta_\mathrm{a}\omega_\mathrm{a}\dot{\delta}_\mathrm{e} + {\omega_\mathrm{a}}^2(\delta_\mathrm{c} - \delta_\mathrm{e}) \tag{4.52}$$

where δ is the angular position and δ_c command. Combining Eqs. (4.51) and (4.52) forms our plant model written in state space form as

$$\begin{bmatrix}\dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e\end{bmatrix} = \begin{bmatrix} Z_\alpha & VZ_\alpha & 0 & VZ_\delta \\ M_\alpha/VZ_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix}\begin{bmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix}\delta_c \tag{4.53}$$

Assume each of the state variables is available for feedback. Equation (4.53) represents the aircraft's dynamics for the plant model expressed in (4.46). This model needs to be combined with the sensitivity weighting filter, complementary sensitivity weighting filter, and the control activity penalty. To satisfy Theorem 4.1 requirements for D_1 to be injective, the control activity is penalized by weighting

the variable $\ddot{\delta}$ in (4.53) as the control activity. This defines the third regulated variable z_3 expressed as

$$z_3 = W_C \ddot{\delta}_e \tag{4.54}$$

where $\ddot{\delta}$ is formed as

$$\ddot{\delta}_e = \underbrace{[0 \quad 0 \quad 0 \quad 1]}_{C_{\ddot{\delta}}} \dot{x} = C_{\ddot{\delta}}\left(A_p x + B_p u\right) \tag{4.55}$$

which gives z_3 as

$$z_3 = W_C C_{\ddot{\delta}}\left(A_p x + B_p u\right) \tag{4.56}$$

To form the H_∞ controller design model, we combine the plant model (4.46) with the sensitivity weighting filter (4.47) and complementary sensitivity weighting filter (4.48) as

$$\begin{bmatrix} \dot{x} \\ \dot{x}_S \\ \dot{x}_T \end{bmatrix} = \begin{bmatrix} A_p & 0 & 0 \\ -B_S C_p & A_S & 0 \\ B_T C_p & 0 & A_T \end{bmatrix} \begin{bmatrix} x \\ x_S \\ x_T \end{bmatrix} + \begin{bmatrix} B_p \\ -B_S D_p \\ B_T D_p \end{bmatrix} \delta_c + \begin{bmatrix} 0 \\ B_S \\ 0 \end{bmatrix} r$$
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -D_S C_p & C_S & 0 \\ D_T C_p & 0 & C_T \\ W_C C_c A_p & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ x_S \\ x_T \end{bmatrix} + \begin{bmatrix} -D_S D_p \\ D_T D_p \\ W_C C_c B_p \end{bmatrix} \delta_c + \begin{bmatrix} D_S \\ 0 \\ 0 \end{bmatrix} r \tag{4.57}$$

which is of the form of (4.24). For this flight condition, the plant model data is

$$\begin{aligned} Z_\alpha &= -1.05273(1/\mathrm{s}); \\ Z_\delta &= -0.0343(1/\mathrm{s}); \\ M_\alpha &= -2.3294\left(1/s^2\right); \\ M_q &= -1.03341\left(1/s^2\right); \\ M_\delta &= -1.1684\left(1/s^2\right); \\ V &= 329.127(\mathrm{ft/s}); \\ \omega_\mathrm{a} &= 2\pi^* 13.(\mathrm{rad}/s); \\ \zeta_\mathrm{a} &= 0.6; \end{aligned} \tag{4.58}$$

The sensitivity and complementary weighting filter designs are created by first defining (selecting) the desired loop gain crossover frequency ω_c. For this flight

condition, we will set $\omega_c = 2(\text{Hz})$. The sensitivity weighting filter coefficients from Fig. 4.10 are formed by choosing ω_c and the gain K. W_s from Fig. 4.10 is

$$W_S(s) = \frac{K(\tau s + 1)}{s} \tag{4.59}$$

The time constant is computed as

$$\tau = \frac{1}{2\pi\omega_c}(\omega_c \text{ in Hz}) \tag{4.60}$$

The gain K is chosen to be

$$K = \frac{0.5}{\tau} \tag{4.61}$$

The state space model for (4.59) is

$$(A_S, B_S, C_S, D_S) = (0., 1., 6.2832, 0.5) \tag{4.62}$$

The blue curve in Fig. 4.13 shows the frequency response for (4.59). The integral action at low frequency will weight $S(s)$ to provide the desired command tracking. The complementary sensitivity weighting filter from Fig. 4.11 is

$$W_T(s) = \frac{K(\tau_N s + 1)}{(\tau_D s + 1)} \tag{4.63}$$

where

$$\begin{aligned} \tau_N &= \frac{1}{2\pi\omega_c}(\omega_c \text{ in Hz}) \\ \tau_D &= 0.005(s) \\ K &= 0.707 \end{aligned} \tag{4.64}$$

The green curve in Fig. 4.13 shows the frequency response for (4.63).
The state space model for (4.63) is

$$(A_T, B_T, C_T, D_T) = (-200., 1., -2109.1, 11.252) \tag{4.65}$$

This model was formed by using the *tf2ss* command in Matlab. The weight that penalizes the control activity, W_c, from (4.56) is

$$W_C = 0.1 \tag{4.66}$$

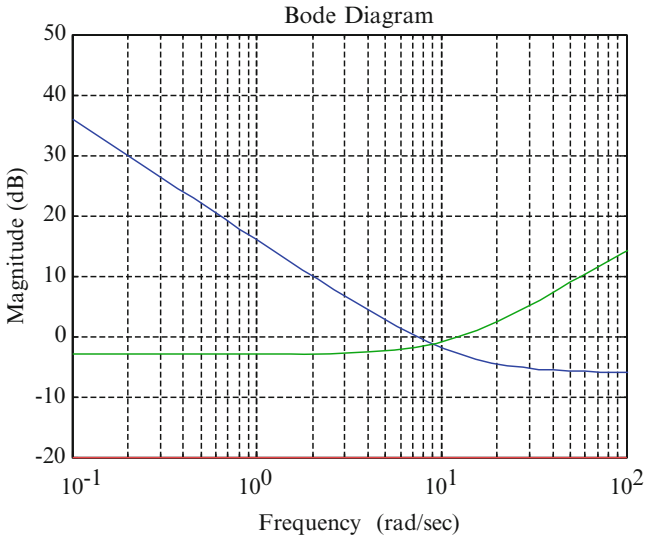


Fig. 4.13 Sensitivity, complementary sensitivity, and control activity weighting filter frequency responses

This penalty is plotted in Fig. 4.13 as the red curve.

Now that the desired loop shapes have been engineered, the γ-iteration process is used to form the state feedback control. The binary search algorithm was started with γ_max = 20 and γ_min = 1. The algorithm converged with the final γ of 2.544781744 (it is important to include at least 9 decimal places so that results can be reproduced by others). For this minimum γ, the six feedback gains are

$$\mathrm{K}_\infty = [-210764319.5 \;\; 14676259029.5 \;\; -3206078184.0 \;\; -29261950.9 \;\; -730137814.4 \;\; 44949185.4]$$

which are too large to be considered for implementation. The Riccati matrix P was substituted back into the ARE and the 2-norm of the sum computed. The 2-norm = 301572105986.4265, which shows that the ill-conditioning has destroyed the accuracy. By increasing γ slightly to $\gamma = 2.549781744$, the solution becomes accurate with 2-norm = 1.8849×10^{-5} with the feedback gains:

$$\mathrm{K}_\infty = [-0.92788164.7534 \;\; -13.1623 \;\; -0.11795 \;\; -3.210460.151197] \quad (4.68)$$

The H_∞ state feedback controller can be implemented in the following state space format:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \quad (4.69)$$

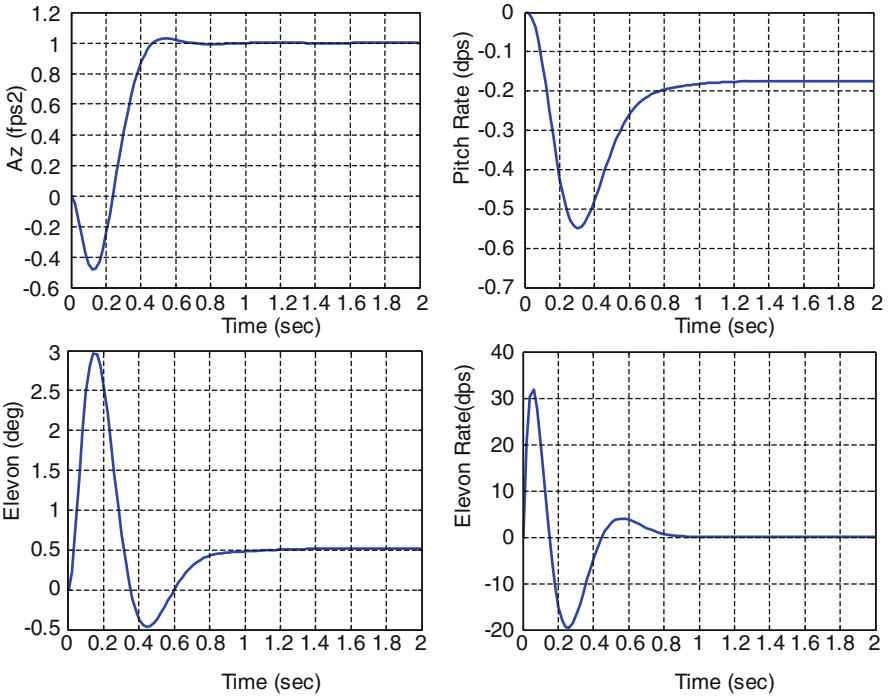


Fig. 4.14 States of the system responding to a unit acceleration step command

with

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} A_S & 0 \\ 0 & A_T \end{bmatrix} & \begin{bmatrix} B_S \\ B_T \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -B_S \\ 0 \end{bmatrix} \\ [-K_\infty(5:6)] & [-K_\infty(1:4)] & [0] \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -200 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\ [-3.2105 \; 0.1512] & [-0.9279 \; 64.7534 \; -13.1623 \; -0.1179] & [0] \end{bmatrix} \quad (4.70)$$

where $x_c = [x_T \quad x_S]^T$, $y = [A_z \quad q \quad \delta_e \quad \dot{\delta}_e]^T$, $r = A_{zc}$, and $u = \delta_c$.

To evaluate the design, a step simulation of the closed-loop system was performed. The states A_z, q, δ_e, and $\dot{\delta}_e$ are plotted versus time in Fig. 4.14. The controller is a second-order system (one state for W_s and one state for W_T). Note that there is no overshoot to the unit command. Using this approach flight condition, the response is quick without the use of large gains. This simulation can be compared with the robust servomechanism design Example 3.5 from Chap. 3. We see that the H∞ control design as responses with a slight overshoot in the acceleration response with an increase in the nonminimum phase undershoot at the initiation of the step command.

4.7 Conclusions

One of the most important developments in the 1980s was H_∞ optimal control and the understanding it gave engineers in performing trades between time domain requirements and frequency domain requirements. We presented the full information state feedback H_∞ optimal control, but output feedback versions also exist. We refer the student wishing to explore output feedback to see [2, 6].

4.8 Exercises

Exercise 4.1. Consider the design of a longitudinal (pitch-plane) autopilot. Using H_∞ state feedback, design a pitch autopilot commanding angle-of-attack α. Use the following dynamics model as the nominal plant model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_{\delta_e}}{V} \\ M_{\delta_e} \end{bmatrix} \delta_e$$

and use data for $\frac{Z_\alpha}{V} = -1.21;\ \frac{Z_\delta}{V} = -0.1987; M_\alpha = 44.2506; M_\delta = -97.2313$. Add second-order actuator dynamics for the elevator. Design the autopilot to track a constant angle-of-attack command. Use the γ-iteration approach outlined in Sect. 4.5.

Exercise 4.2. Consider the longitudinal dynamics of a transport aircraft as given in Chap. 1, Exercise 1.2. Design a H_∞ state feedback controller to track a constant speed command and a constant angle-of-attack command. Use the γ-iteration approach outlined in Sect. 4.5.

Exercise 4.3. Consider the lateral-directional dynamics of a transport aircraft as given in Chap. 1, Exercise 1.4. Design a H_∞ state feedback controller to track a constant stability axis roll rate p_s command (see Eq. (1.22)) and regulate sideslip angle β. Assume $\alpha_0 = 6\,\text{deg}$.

References

1. Doyle, J.C., Glover, K., Khargonekar, P.P., Francis, B.A.: State space solutions to standard H_2 and H_∞ control problems. IEEE Trans. Autom. Control **34**(8), 831–846 (1989)
2. Basar, T., Bernhard, P.: H_∞-Optimal Control and Related Minimax Design Problems, A Dynamic Game Approach. Birkhauser, Berlin (1990)
3. Ball, J.A., Helton, J.W.: Control for nonlinear plants: connection with differential games. In: Proceedings of the 28th Conference on Decision and Control (Tampa, FL), Inst. of Electrical and Electronics Engineers, Piscataway, NJ, pp. 956–962 (1989)
4. Isaacs, R.: Differential Games, p. 67. Wiley, New York (1965)
5. Wise, K.A., Sedwick, J.L.: Nonlinear H_∞ optimal control for Agile Missiles. J. Guidance Control Dyn. **19**(1), 157–165 (1996)
6. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Englewood Cliffs (1996)

Chapter 5
Frequency Domain Analysis

5.1 Introduction

Frequency domain analysis methods are among the most useful tools available for the development of control systems. When designing a control system, it is very important to understand the stability and robustness properties of the design. For linear systems, these properties are best analyzed, displayed, and understood in the frequency domain. For linear single-input single-output (SISO) systems, frequency domain methods for analysis, as well as techniques for synthesis of a controller, have been developed and used in industry since the 1950s. These analysis and design methods are often referred to as classical methods and include techniques like root locus, Bode, Nyquist, and Nichols charts. For multi-input multi-output (MIMO) systems, the analysis methods used are typically extensions of the methods used for SISO systems. In order to understand the methods for MIMO analysis, one should have a good grasp of classical SISO frequency domain methods.

Since the early 1980s, control system analysts have been focused upon determining the stability and robustness of MIMO feedback designs in the presence of uncertainties. In particular, this focus has been upon frequency domain techniques using methods which employ singular value frequency responses. These singular value-based methods of analysis join, and in some cases replace, the classical Bode and Nyquist techniques with multivariable generalizations and extend many modeling uncertainty capabilities. They have become widespread in industry as today's systems require MIMO analysis. This chapter presents an overview of the theory and methods available, connecting the classical and multivariable analysis methods and tools, and highlights aerospace control applications and analyses in the frequency domain.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_5,

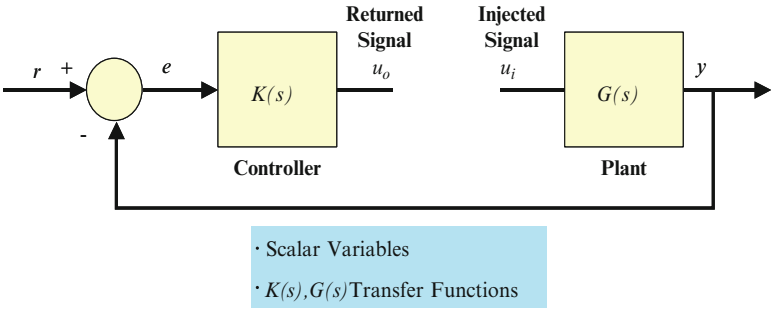


Fig. 5.1 Single-input single-output *KG* block diagram

5.2 Transfer Functions and Transfer Function Matrices

Many of the frequency domain analysis models for MIMO systems are natural extensions of transfer functions used to analyze SISO systems. However, unlike these transfer functions, MIMO analysis models have different sizes depending upon where the loop is broken for analysis. Consider the SISO system shown in the block diagram of Fig. 5.1.

The loop gain for this system can be calculated by breaking the loop at the control generation point (plant input) and injecting a signal u_i. The returned signal is

$$u_o = -\underbrace{K(s)G(s)}_{L(s)} u_i \tag{5.1}$$

in which $L(s)$ is the *loop gain transfer function*. Differencing the injected signal u_i and the returned signal u_o results in

$$\begin{aligned} u_i - u_o &= u_i + K(s)G(s)u_i = (1 + K(s)G(s))u_i \\ &= (1 + L(s))u_i \end{aligned} \tag{5.2}$$

which is the *return difference* for the loop. We will find later in this chapter that the *return difference matrix*, $I + L(s)$, plays a very important role in the development of stability robustness analysis tests for MIMO systems. The error transfer function for this system is

$$\frac{e(s)}{r(s)} = \frac{1}{1 + K(s)G(s)} = S(s) \tag{5.3}$$

where $S(s)$ is the *sensitivity* function, which describes the error dynamics. Note that the sensitivity is the inverse of the return difference. The closed-loop response to a command input is

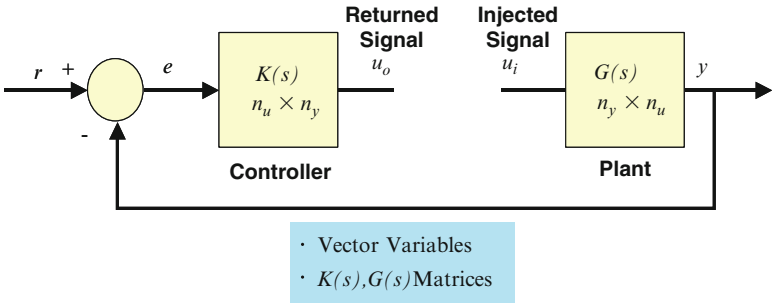


Fig. 5.2 Multi-input multi-output KG block diagram

$$\frac{y(s)}{r(s)} = \frac{K(s)G(s)}{1 + K(s)G(s)} = T(s) \tag{5.4}$$

where $T(s)$ is the *closed-loop transfer function*. The transfer function $T(s)$ is also called the *complementary sensitivity*, since $S(s)$ and $T(s)$ satisfy the identity

$$S(s) + T(s) = 1 \tag{5.5}$$

Now, consider the multivariable equivalent of Fig. 5.1 as shown in Fig. 5.2. In Fig. 5.2, the variables $r, e, u_i, u_o,$ and y are vectors, with the controller $K(s)$ a $n_u \times n_y$ matrix and the plant $G(s)$ a $n_y \times n_u$ matrix. The figure shows the loop broken at the plant input. The loop gain $L(s)$ is formed using the same procedure as in (5.1) where $L(s) = K(s)G(s)$ is a $n_u \times n_u$ matrix. Forming the return difference matrix yields

$$u_i - u_o = (I_{n_u} + K(s)G(s))u_i = (I_{n_u} + L(s))u_i \tag{5.6}$$

where $I_{n_u} + L(s)$ is also a $n_u \times n_u$ matrix. If this same procedure for calculating the loop gain is applied at the output of the plant, as shown in Fig. 5.2, the return difference dynamics are

$$u_i' - u_o' = \big(I_{n_y} + G(s)K(s)\big)u_i' \tag{5.7}$$

which produces a loop gain and return difference matrix that are $n_y \times n_y$ in dimension.

It is very important to learn that for MIMO systems, the loop gain is different at the plant input and plant output loop break points, which is unlike SISO systems. This dissimilarity is caused by the fact that matrices do not commute, but scalars do. Table 5.1 summarizes the loop gain, return difference, sensitivity, and complementary sensitivity transfer functions and matrices for the SISO and MIMO systems shown in Figs. 5.1 and 5.2.

Table 5.1 Summary of transfer functions and transfer function matrices used in frequency domain analysis

Function	SISO system (Fig. 5.1)	MIMO system (Fig. 5.2) at plant input	MIMO system (Fig. 5.2) at plant output
Loop gain	$L(s) = K(s)G(s) = G(s)K(s)$	$L(s) = K(s)\,G(s)$	$L(s) = G(s)\,K(s)$
Return difference	$1 + L(s)$	$I_{n_u} + L(s)$	$I_{n_y} + L(s)$
Sensitivity $S(s)$	$\dfrac{1}{1+L(s)}$	$(I_{n_u} + L(s))^{-1}$	$\left(I_{n_y} + L(s)\right)^{-1}$
Complementary sensitivity $T(s)$	$\dfrac{L(s)}{1+L(s)}$	$(I_{n_u} + L(s))^{-1}L(s)$	$\left(I_{n_y} + L(s)\right)^{-1}L(s)$

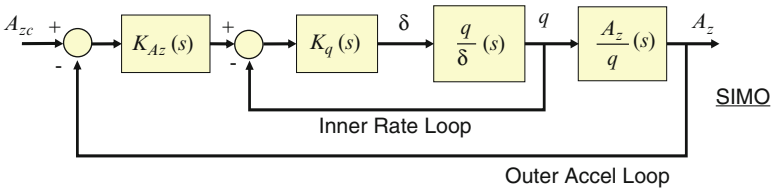


Fig. 5.3 Pitch-plane dynamics and autopilot controller

Table 5.1 lists the various matrices used to analyze MIMO control systems. In the remainder of this book, the subscript on the identity matrix indicating its dimension will be dropped for notational convenience.

Example 5.1 Consider the linear-time-invariant (LTI) pitch-plane dynamics of an unmanned aircraft shown in Fig. 5.3, controlled using a classical proportional-plus-integral control architecture. The pitch-plane short-period dynamics are given by (A, B, C, D) and can be written as

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix}}_{A} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{bmatrix}}_{B} \delta_e$$
$$\begin{bmatrix} A_z \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix}}_{C} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \underbrace{\begin{bmatrix} Z_\delta \\ 0 \end{bmatrix}}_{D} \delta_e \tag{5.8}$$

These dynamics form a single-input multi-output system. The transfer function matrix for the plant dynamics is

$$G(s) = C(sI - A)^{-1}B + D = \begin{bmatrix} \frac{A_z}{\delta_e} \\ \frac{q}{\delta_e} \end{bmatrix} \tag{5.9}$$

which is a 2×1 matrix. The autopilot (controller) for this plant contains proportional-plus-integral control elements in the inner rate loop closure and outer acceleration loop closure, given by

$$K_{A_z}(s) = \frac{K_{A_z}(s + a_z)}{s} \tag{5.10}$$

and

$$K_q(s) = \frac{K_q(s + a_q)}{s} \tag{5.11}$$

with the controller transfer function matrix given by

$$K(s) = [\, K_{A_z}(s) K_q(s) \quad K_q(s) \,] \tag{5.12}$$

which is a 2×1 matrix. A state-space model for this controller is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{5.13}$$

with matrices given as

$$A_c = \begin{bmatrix} 0 & 0 \\ K_q a_q & 0 \end{bmatrix};\ B_{c1} = \begin{bmatrix} -K_a a_z & 0 \\ -K_a K_q a_q & -K_q a_q \end{bmatrix};\ B_{c2} = \begin{bmatrix} K_a a_z \\ K_a K_q a_q \end{bmatrix}$$

$$C_c = [K_q \quad 1];\ D_{c1} = [\, -K_a K_q \quad -K_q \,];\ D_{c2} = \begin{bmatrix} K_a K_q \end{bmatrix} \tag{5.14}$$

The loop gain at the input to the plant is

$$L(s) = K(s)G(s) = K_{A_z}(s) K_q(s) \frac{A_z}{\delta_e} + K_q(s) \frac{q}{\delta_e} \tag{5.15}$$

which is a scalar transfer function. To analyze stability for this system, any SISO analysis technique can be applied. If we examine the loop gain at the plant output, then

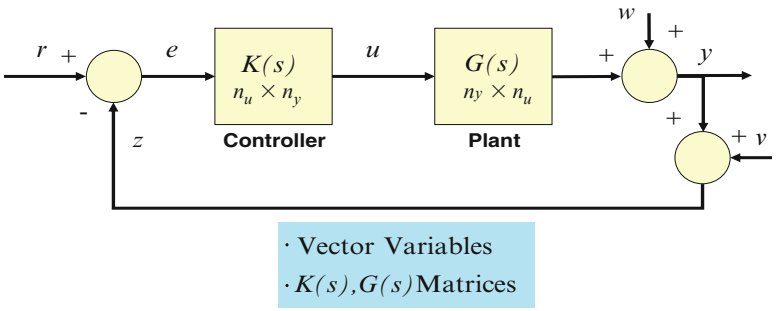


Fig. 5.4 MIMO system with disturbance and measurement noise

$$L(s) = G(s)K(s) = \begin{bmatrix} \frac{A_z}{\delta_e} K_{A_z}(s)K_q(s) & \frac{A_z}{\delta_e} K_q(s) \\ \frac{q}{\delta_e} K_{A_z}(s)K_q(s) & \frac{q}{\delta_e} K_q(s) \end{bmatrix} \tag{5.16}$$

which is a 2×2 matrix and is a singular matrix since it is the product of matrices that are $(2 \times 1) \times (1 \times 2)$ in dimension. It is typical in most aerospace applications that the plant and controller matrices are non-square. In this case, stability analysis should be conducted at the loop break point of minimum dimension.

Figure 5.4 illustrates a LTI MIMO system with command $r(t) \in R^{n_y}$, plant disturbance $w(t) \in R^{n_y}$, and measurement noise $v(t) \in R^{n_y}$. The output response from the system shown in Fig. 5.4 is

$$Y(s) = T(s)R(s) + S(s)W(s) + T(s)V(s) \tag{5.17}$$

This equation shows how the output response depends upon the sensitivity and complementary sensitivity functions. At frequencies $s = j\omega$ where commands are to be followed, we want $T(s) \to I$, which shows that sensor noise is also passed through the system into the output. It is not possible to reject sensor noise and track commands at the same frequencies. At frequencies where plant disturbances are to be rejected, we want $S(s) \to 0$.

The error response $E(s)$ can be formed by writing the following loop equations:

$$\begin{aligned} u &= Ke \\ y &= GKe + w \\ z &= GKe + w + v \\ e &= r + z = r + GKe + w + v \\ E(s) &= S(s)(R(s) + W(s) + V(s)) \end{aligned} \tag{5.18}$$

which shows that to make errors in tracking commands small, we want $S(s) \to 0$. Equations (5.17) and (5.18) illustrate the control design dilemma faced by engineers, that is, to make $S(s) \to 0$ at low frequencies for command tracking

and disturbance rejection and $T(s) \to 0$ at high frequencies for sensor noise rejection and robustness to high-frequency unmodeled dynamics. The dilemma is that $S(s) + T(s) = I$ at all frequencies, and as the sensitivity is made small, the complementary sensitivity is made unity and vice versa.

5.3 Multivariable Stability Margins

Classical stability margin analyses use frequency response methods (Bode and Nyquist) in determining the relative stability of SISO systems. These methods manipulate the loop transfer function of the system to derive gain and phase margins, typical measures of relative stability. In multivariable systems (MIMO systems), the loop transfer function of the system is a complex-valued matrix, making it more difficult to apply the same SISO methods to determine relative stability. The question of stability is easily answered by examining the poles of the closed-loop transfer function or the eigenvalues of the closed-loop matrix A_{cl}. It is the relative stability question, that is, the gain and phase margins for MIMO systems that is difficult.

In SISO systems, the gain of the loop transfer function is determined by computing the magnitude of the complex-valued transfer function versus frequency. For MIMO systems, the notion of gain or magnitude for the loop transfer function matrix becomes a question of determining the "magnitude" of a matrix versus frequency. To accomplish this task, the singular values of the matrix can be computed versus frequency and used as a measure of its magnitude.

In this section, we are concerned with deriving stability margins for multivariable systems. The robust stability analysis tests and stability margins formulas developed here are derived from application of the multivariable Nyquist theorem. These tests and formulas are natural extensions of the SISO tests reviewed in the previous section.

5.3.1 Singular Values

The singular value decomposition of a matrix $A \in C$ of dimension $n \times m$ is $A = U\Sigma V^*$, where * denotes complex conjugate transpose, and where $\Sigma \in R^{n\times m}$, $U \in C^{n\times n}$ and $V \in C^{m\times m}$ are unitary matrices, whose columns denote left and right singular vectors of the matrix A, respectively. (Note the similarity to an eigenvalue decomposition of a matrix.) Assuming that the matrix is of rank k, the nonzero portion of the singular value matrix is

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} \end{bmatrix}; \quad \Sigma_1 = diag[\,\sigma_1 \quad \cdots \quad \sigma_k\,] \tag{5.19}$$

with the singular values ordered in size with $\overline{\sigma} = \sigma_1$, the largest, and $\underline{\sigma} = \sigma_k$ the smallest. The use of singular values plays an important role in analyzing the near singularity of matrices. If A is a square singular matrix, then $\underline{\sigma} = 0$, and it is not invertible.

The maximum and minimum singular values of the matrix A can be defined as

$$\bar{\sigma}(A) = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2 \tag{5.20}$$

$$\underline{\sigma}(A) = \min_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

The maximum singular value of the matrix A (its 2-norm) represents how "big" the matrix is or how large the "gain" of the matrix is. The minimum singular value represents how nearly singular the matrix is. The condition number for a matrix, $\kappa(A)$, is the ratio of the maximum and minimum singular values, given by

$$\kappa(A) = \frac{\bar{\sigma}(A)}{\underline{\sigma}(A)} \tag{5.21}$$

and is used by numerical analyst to gain insight into how invertible a matrix is.

Associated with each singular value are singular vectors that describe the "direction" of the singular value. Consider the matrix $A \in C^{n \times m}$ with rank $k = \min(n, m)$. The k nonzero singular values of A, denoted as $\sigma_i(A)$, are the strictly positive square roots of the k nonzero eigenvalues of A^*A (or equivalently AA^*). This is expressed as

$$\sigma_i(A) = \sqrt{\lambda_i(A^*A)} = \sqrt{\lambda_i(AA^*)} > 0 \tag{5.22}$$

Each singular value has an input and output direction which can be determined by examining the singular vectors associated with the singular value decomposition (SVD) of the matrix. The SVD of a complex matrix $A \in C^{n \times m}$ is

$$A = U\Sigma V^* \tag{5.23}$$

where U is an $n \times n$ unitary matrix (i.e., $U^* = U^{-1}$) consisting of orthonormal column vectors u_i

$$U = [u_1 \quad \cdots \quad u_n] \tag{5.24}$$

which are referred to as the left singular vectors of the matrix, V is an unitary matrix consisting of orthonormal column vectors v_i

$$V = [v_1 \quad \cdots \quad v_m] \tag{5.25}$$

which are referred to as the right singular vectors of the matrix, and Σ is a real $n \times m$ matrix given by

$$\Sigma = \begin{bmatrix} \sigma_1 & & \underset{\sim}{0} & \\ & \sigma_2 & & \\ & & \ddots & \underset{\sim}{0} \\ \underset{\sim}{0} & & \sigma_k & \\ & \underset{\sim}{0} & & \underset{\sim}{0} \end{bmatrix} \tag{5.26}$$

The σ_i in (5.26) is the i-th singular value of the matrix A, with a corresponding left singular vector u_i (5.24) and right singular vector v_i (5.25). It is easy to show that

$$\begin{aligned} Av_i &= \sigma_i u_i \\ A^* u_i &= \sigma_i v_i \end{aligned} \tag{5.27}$$

The above equations can also be written as

$$\begin{aligned} A^* A v_i &= \sigma_i^2 v_i \\ A A^* u_i &= \sigma_i^2 u_i \end{aligned} \tag{5.28}$$

which shows that σ_i^2 is an eigenvalue of AA^* or A^*A and u_i is an eigenvector of AA^* and v_i is an eigenvector of A^*A.

Consider a square matrix $A \in C^{n \times n}$ having rank k. Using an SVD, the matrix A can be represented using a dyadic expansion as

$$A = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \cdots + \sigma_k u_k v_k^* = \sum_{i=1}^{k} \sigma_i u_i v_i^* \tag{5.29}$$

The SVD of a matrix describes the gain through the matrix, with the maximum gain equal to the 2-norm of the matrix $\left(\|A\|_2 = \sigma_1(A) = \bar{\sigma}(A)\right)$. In addition to the gain, the SVD describes the direction associated with the gain. The dyadic expansion in (5.29) indicates that the left and right singular vectors describe the direction of the gain. The maximum gain through the matrix occurs with the input direction from v_1 and output direction u_1.

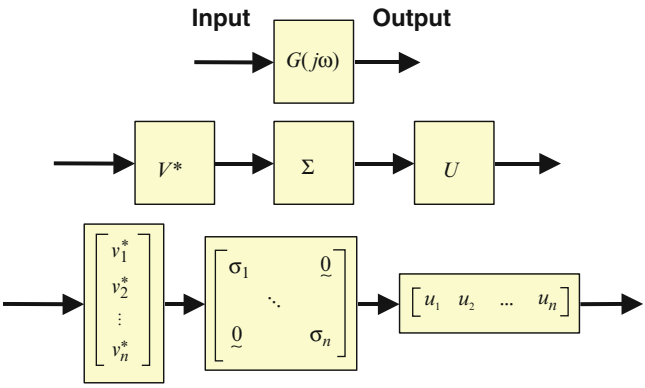


Fig. 5.5 Singular value decomposition of a transfer function matrix

Figure 5.5 illustrates the input-to-output mapping for a general transfer function matrix $G(j\omega) \in C^{n\times n}$. Here, the singular value expansion provides insight into the relative gain between input-to-output channels for a transfer function matrix.

5.3.2 *Singular Value Properties*

If the matrix A is invertible, that is, A^{-1} exists, then

$$\bar{\sigma}(A^{-1}) = \frac{1}{\underline{\sigma}(A)} \text{ and } \underline{\sigma}(A^{-1}) = \frac{1}{\bar{\sigma}(A)}.$$

$$\|A\|_2 = \bar{\sigma}(A)$$
$$\|A\|_F^2 = \sum_{i=1}^{n} \sigma_i^2(A)$$

where $\|\bullet\|_F$ denotes the Frobenius norm. If the matrices U and V are unitary, then

$$\sigma_i(UA) = \sigma_i(A)$$
$$\sigma_i(AV) = \sigma_i(A)$$

which says that unitary matrices preserve the singular values and $\|\bullet\|_2$ of a matrix.

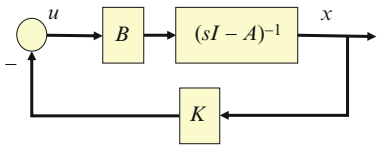


Fig. 5.6 State feedback block diagram

5.3.3 *Multivariable Nyquist Theory*

The multivariable Nyquist criterion gives a "yes or no" answer to the stability question. Other methods such as computing the eigenvalues of the system A matrix, examining the poles of the closed-loop transfer function, or solving a Lyapunov equation can also be used to answer the stability question. However, understanding the multivariable Nyquist criterion leads to important understanding of robustness analysis tests used to analyze model uncertainties. In addition, time delays, e^{sT}, are easily incorporated into the analysis in order to analyze MIMO systems with time delays.

The multivariable Nyquist criterion is derived from an application of the principle of the argument from complex variable theory.

Theorem 5.1.

Let Γ be a closed clockwise contour in the s-plane. Let f(s) *be a complex-valued function. Suppose that*

1. *$f(s)$ is analytic on Γ*
2. *$f(s)$ has Z zeros inside Γ*
3. *$f(s)$ has P poles inside Γ.*

Then, $f(s)$ will encircle the origin, 0, $Z - P$ times in a clockwise sense as s transverses Γ. ■

Let $N(p, f(s), \Gamma)$ denote the number of encirclements of the point p made by the function $f(s)$ as s transverses the closed clockwise contour Γ. If Γ equals the standard Nyquist D-contour (D_R), encircling the right half plane, and $f(s)$ is a rational function in s, then $N(0, f(s), D_R) = Z - P$.

If $f(s)$ is factored where $f(s) = f_1(s)f_2(s)$, then

$$\begin{aligned} N(0, f_1(s)f_2(s), D_R) &= N(0, f_1(s), D_R) + N(0, f_2(s), D_R) \\ &= (Z_1 - P_1) + (Z_2 - P_2) = Z - P \end{aligned} \tag{5.30}$$

Consider the feedback system shown in Fig. 5.6. The state equations for this system are

$$\dot{x} = Ax + Bu$$
$$u = -Kx$$

The closed-loop system is

$$\dot{x} = (A - BK)x$$

Let $L(s)$ denote the loop transfer function matrix for this system, written as

$$L(s) = K(sI - A)^{-1}B$$

The determinant of the return difference matrix, $\det[I + L(s)]$, is equal to the closed-loop characteristic polynomial divided by the open-loop characteristic polynomial, that is,

$$\det[I + L(s)] = \frac{\phi_{cl}(s)}{\phi_{ol}(s)}$$

This can be shown as

$$\begin{aligned}\phi_{cl}(s) &= \det[sI - A + BK] \\ &= \underbrace{\det[sI - A]}_{\phi_{ol}(s)} \det\left[I + (sI - A)^{-1}BK\right]\end{aligned} \tag{5.31}$$

Now, using the identity

$$\det\left[I_n + \underbrace{F}_{n\times m}\underbrace{G}_{m\times n}\right] = \det\left[I_m + \underbrace{G}_{m\times n}\underbrace{F}_{n\times m}\right]$$

Using this in (5.31) yields

$$\begin{aligned}\phi_{cl}(s) &= \phi_{ol}(s) \det\left[I + (sI - A)^{-1}\mathrm{BK}\right] \\ &= \phi_{ol}(s) \det\left[I + \underbrace{K(sI - A)^{-1}B}_{L(s)}\right] \\ &= \phi_{ol}(s) \det[I + L(s)]\end{aligned} \tag{5.32}$$

where $\phi_{ol}(s)$ is the open-loop system's characteristic polynomial and $\phi_{cl}(s)$ is the closed-loop system's characteristic polynomial. If $\phi_{cl}(s)$ is stable (the closed-loop system is stable), then $N(0, \phi_{cl}(s), D_R) = 0$. From (5.32) stability of $\phi_{cl}(s)$ requires that

$$N(0, \phi_{ol}(s), D_R) + N(0, \det[I + L(s)], D_R) = 0 \tag{5.33}$$

With this understanding, we can state the multivariable Nyquist theorem.

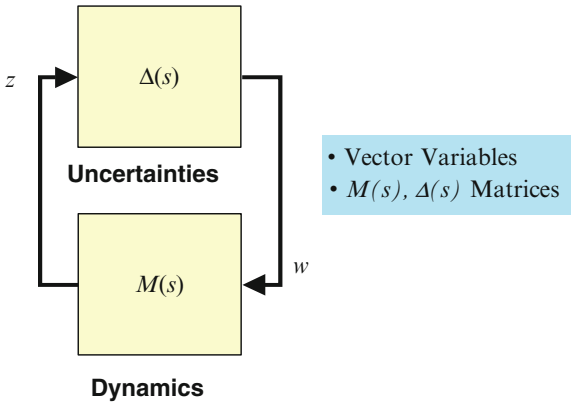


Fig. 5.7 ΔM analysis model

Theorem 5.2 Multivariable Nyquist Theorem

The feedback control system shown in Fig. 5.6 will be closed-loop stable in the sense that $\phi_{\mathrm{cl}}(s)$ *has no closed right half plane zeros if and only if for all R sufficiently large (radius of the* D*-contour)*

$$N(0, \det[I + L(s)], D_R) = -P_{ol} \tag{5.34}$$

or equivalently

$$N(-1, -1 + \det[I + L(s)], D_R) = -P_{ol}$$

where $P_{ol} = N(0, \phi_{ol}(s), D_R)$ *equals the number of open-loop right half plane poles.* ■

The multivariable Nyquist theorem (MNT) states that closed-loop stability requires the number of encirclements made by the determinant of the return difference matrix locus to be equal to the number of unstable open-loop poles. Encirclements can be counted relative to the origin (0, j0) or as in classical Nyquist diagrams about $(-1, \mathrm{j}0)$.

Stability margins for multivariable systems can be derived using the MNT by assuming that the controller $K(s)$ stabilizes the nominal plant $G(s)$ and that gain and phase uncertainties are large enough to change the number of encirclements made by the determinant of the return difference matrix locus. The assumption that the nominal plant is stabilized by the controller tells us that the return difference matrix encircles the origin P_{ol} times in the proper sense. Gain and phase margins can be computed by inserting a gain and phase variation $ke^{i\phi}$ in between the controller $K(s)$ and plant $G(s)$ and solving for the gain k (with $\phi = 0$) and phase θ (with $k = 1$) that destabilizes the system. To proceed in a more general manner, we consider the stability analysis model shown in Fig. 5.7 where the uncertainties in the system (gain and phase uncertainties) are represented in a block matrix $\Delta(s)$ and the

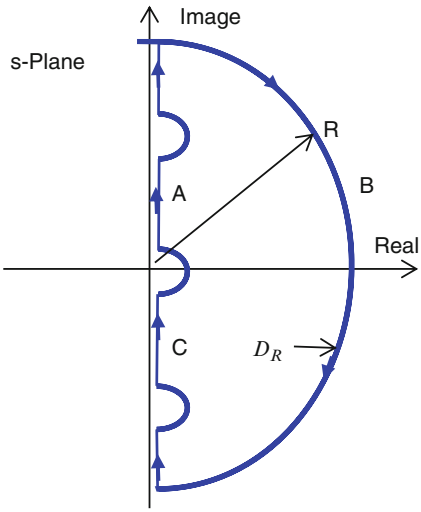


Fig. 5.8 Nyquist D_R contour

nominal plant and controller are represented in a matrix $M(s)$. Techniques for deriving this model will be presented in the next section.

The stability analysis question is "how large can the uncertainties $\Delta(s)$ become before the system becomes unstable?" The loop transfer function matrix $L(s)$ for this system is $L(s) = \Delta(s)M(s)$, with the return difference matrix given by $I + L(s) = I - \Delta(s)M(s)$. Using the MNT, for the system to become unstable, the uncertainties $\Delta(s)$ must change the number of encirclements made by the $\det[I + L(s)]$ locus. Note that the block diagram in Fig. 5.7 has no summing node with negative sign in most block diagrams. Thus, the return difference matrix for this system is written $I - \Delta M$.

As long as the return difference matrix $I + L(s)$ is nonsingular (for $s = j\omega$ along the D-contour), the number of encirclements made by the $\det[I + L(s)]$ locus will not change. This is best explained by examining the $\det[I + L(s)]$ locus as s transverses the D_R contour. Fundamental to this approach is the assumption that the nominal closed-loop system is stable, that is, the control design stabilizes the open-loop system.

Assuming that the nominal closed-loop system is stable, $\phi_{cl}(s)$ is a stable polynomial, and that it has no right half plane zeros. Let $f(s) = \det[I + L(s)]$, and represent $f(j\omega)$ with its magnitude and phase as

$$f(j\omega) = |f(j\omega)|e^{j\phi(\omega)} \tag{5.35}$$

as s transverses the D_R contour in the s-plane.

Consider the $j\omega$ axis path A shown in Fig. 5.8, where $0 \leq \omega \leq +\infty$. The section A locus of $f(j\omega)$ is shown in Fig. 5.9a. At low frequencies, the magnitude of $f(j\omega)$ is large due to the magnitude of $L(j\omega)$. As $\omega \to \infty$, the loop transfer matrix $L(j\omega) \to 0$, resulting in the $\det[I + L(j\omega)] = 1\,(1, j0)$. Along the infinite radius path B, $s = e^{j\psi}R$, with $R \to \infty$ and $-\frac{\pi}{2} \leq \psi \leq \frac{\pi}{2}$. When $R \to \infty$, $L(j\omega) \to 0$. This results in

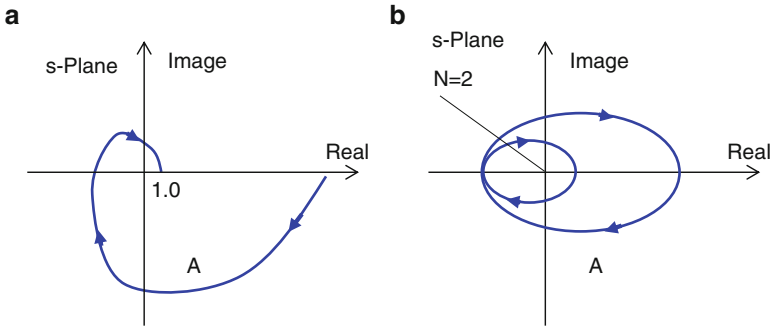


Fig. 5.9 Nyquist examples

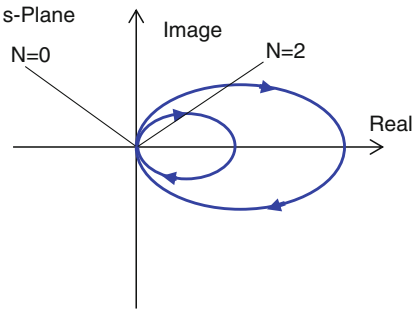


Fig. 5.10 Counting encirclements

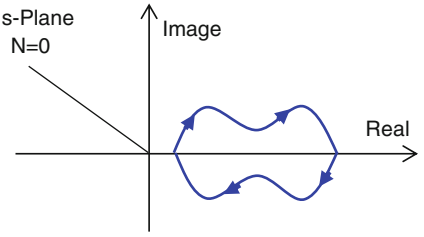


Fig. 5.11 Counting encirclements

encirclements of the point (1, *j*0). Section C will be the complex conjugate of the section A path. Figure 5.9b shows the entire locus and the number of encirclements N. Figure 5.9b shows there are two clockwise encirclements of the origin.

The number of encirclements N of the $\det[I + L(s)]$ locus must be equal to the number of open-loop unstable poles, P_{ol}, if the closed-loop system is to be stable. If the $\det[I + L(s)]$ were equal to zero then the number of encirclements would be indeterminate, or at least not equal to P_{ol}. This is shown in Fig. 5.10. In order for the number of encirclements to change, the $\det[I + L(s)]$ must equal zero at some frequency.

If $\phi_{ol}(s)$ is a stable polynomial, then $P_{ol} = 0$. An example $\det[I + L(s)]$ locus for this condition is shown in Fig. 5.11. In order for stable system to be destabilized by uncertainties Δ, the origin must be encircled.

5.3.4 Stability Margins for Multi-Input Multi-Output Systems

Uncertainty models used for stability analysis may be categorized as unstructured or structured. If the system uncertainty is modeled as a full single-block matrix, the uncertainty is unstructured. If the uncertainty is modeled as a block diagonal matrix, the uncertainty is structured. Both unstructured and structured uncertainty analysis procedures use singular value theory to measure the size of complex-valued matrices.

The following robustness theorems which are used to derive stability margins for multivariable systems are derived from an application of the multivariable Nyquist theorem. Consider the state feedback control system shown in Fig. 5.6. The basic problem is to determine the robustness of the design in the presence of uncertainties. This design has the state-space realization using the triple (A, B, K) with the loop transfer matrix (LTM) given by

$$L(s) = K(sI - A)^{-1}B \tag{5.36}$$

We wish to determine to what extent gain and phase uncertainty within the LTM can vary without compromising the stability of the closed-loop system. From the previous section (Eq. (5.32)), we know that

$$\det[I + L(s)] = \frac{\phi_{cl}(s)}{\phi_{ol}(s)} \tag{5.37}$$

where

$\phi_{ol}(s) = \det[sI - A]$: open-loop characteristic polynomial

$\phi_{cl}(s) = \det[sI - A + BK]$: closed-loop characteristic polynomial

Using the multivariable Nyquist theorem, stability for this system can be stated as follows:

The system of Fig. 5.6 will be closed-loop stable in the sense that $\phi_{cl}(s)$ has no closed right half plane zeros if and only if for all R sufficiently large

$$N(0, \det[I + L(s)], D_R) = -P_{ol} \tag{5.38}$$

or equivalently

$$N(-1, -1 + \det[I + L(s)], D_R) = -P_{ol} \tag{5.39}$$

where D_R is the standard Nyquist D-contour, which encloses all P_{ol} closed right half plane zeros of $\phi_{ol}(s)$. Note that $N(b_1, f(s), D)$ is indeterminate if $\phi(s_0) = b_1$ for some s_0 on the contour D.

The stability robustness of a multivariable system can be observed by the near singularity of its return difference matrix, $I + L(s)$, at some frequency $s = j\omega_0$. If $I + L(s)$ is nearly singular, then a small change in $L(s)$ could make $I + L(s)$ singular. From a single-input single-output viewpoint this is the distance from the $(-1, \text{j}0)$ point in the complex plane made by the gain loci $L(j\omega)$. If the gain loci then encircles the $(-1, \text{j}0)$, point instability results. The robustness theory discussed here gives an analogous distance measure for multivariable systems.

Application of the multivariable Nyquist theorem above is of little applicability as a robustness indicator because the $\det[I + L(s)]$ does not indicate the near singularity of $I + L(s)$. The multivariable Nyquist theorem only determines absolute stability. To determine the degree of robustness for a multivariable system, we determine how nearly singular the return difference matrix is by computing its singular values versus frequency.

Examining the magnitude of the singular values of the return difference matrix will indicate how close the matrix is to being singular. This measure of closeness to singularity is used in forming a multivariable gain margin, similar to the classical gain margin. However, as with many matrix norms, there is a restriction on the applicability of the singular value analysis. This restriction states that the compensated system described using the nominal $L(s)$ is closed-loop stable.

Classical gain and phase margins are used to measure the robustness of SISO systems to perturbations in the feedback loop. Singular values are used in measuring the robustness of multivariable systems. Let $L'(s)$ denote the perturbed LTM, which represents the actual system and differs from the nominal LTM $L(s)$ because of uncertainties in the open-loop plant model. Assume that $L'(s)$ has the state-space realization (A', B', K') and open and closed-loop polynomials given by

$$\phi'_{ol}(s) = \det[sI - A'] \tag{5.40}$$

$$\phi'_{cl}(s) = \det[sI - A' + B'K']$$

respectively. Define $\tilde{L}(s, \varepsilon)$ as a matrix of rational transfer functions with real coefficients which are continuous in ε for all ε such that $0 \leq \varepsilon \leq 1$ and for all $s \in D_R$, which satisfies $\tilde{L}(s, 0) = L(s)$ and $\tilde{L}(s, 1) = L'(s)$. Using these definitions of the perturbed model, we are ready to state the following fundamental robustness theorem.

Theorem 5.3

The polynomial $\phi'_{cl}(s)$ has no zeros in the closed right half plane and the perturbed feedback system is stable if the following hold:

1. *(a) $\phi_{ol}(s)$ and $\phi'_{ol}(s)$ have the same number of zeros in the closed right half plane.*
   (b) $\phi_{cl}(s)$ has no zeros in the closed right half plane.

2. $\det[I + L(s, \varepsilon)] = 0$ *for all* (s, ε) *in* $D_R \times [0, 1]$ *and for all* R *sufficiently large.* ■

This theorem states that the closed-loop perturbed system will be stable, if, by continuously deforming the Nyquist loci for the nominal system into that of the perturbed system $I + \tilde{L}(s, \varepsilon)$, that the number of encirclements of the critical point is the same for $L'(s)$ and $L(s)$, then no closed right half plane zeros were introduced into $\phi'_{cl}(s)$, resulting in a stable closed-loop system.

This theorem is used to develop simple tests for different types of model error characterizations. Just as there is no unique representation for dynamic systems, there are many different forms for describing their modeling errors. The most common model error characterizations are additive errors and multiplicative errors (also described as relative or absolute errors). The classical gain and phase margins are associated with multiplicative error models since these margins are multiplicative in nature. (See Doyle [1], Table 1, for representative types of uncertainty characterizations.)

Let $\Delta(s)$ denote the modeling error under consideration. The additive model error is given by

$$\Delta_a(s) = L'(s) - L(s) \tag{5.41}$$

and the multiplicative model error is given by

$$\Delta_m(s) = [L'(s) - L(s)]L^{-1}(s) \tag{5.42}$$

The perturbed LTM can be constructed using Eqs. (5.41) and (5.42). For the additive error model, we have

$$\tilde{L}(s, \varepsilon) = L(s) + \varepsilon\Delta_a(s) \tag{5.43}$$

and for the multiplicative error model, we have

$$\tilde{L}(s, \varepsilon) = [I + \varepsilon\Delta_m(s)]L(s) \tag{5.44}$$

Both Eqs. (5.41) and (5.42) imply the same $\tilde{L}(s, \varepsilon)$ using different model error characterizations. In both Eqs. (5.41) and (5.42), $\tilde{L}(s, \varepsilon)$ is given by

$$\tilde{L}(s, \varepsilon) = (1 - \varepsilon)L(s) + \varepsilon L'(s) \tag{5.45}$$

showing that $\tilde{L}(s, \varepsilon)$ is continuous in ε for $\varepsilon \in [0, 1]$ and for all $s \in D_R$.

We have now defined the true perturbed plant model in terms of its nominal design model and the uncertainty matrix. The fundamental robustness theorem uses the return difference matrix $I + \tilde{L}(s, \varepsilon)$ to determine if the number of encirclements of the critical point will change with the uncertainties. This happens when $I + \tilde{L}(s, \varepsilon)$ becomes singular, in which case the $\det\left[I + \tilde{L}(s, \varepsilon)\right] = 0$.

Using the multiplicative error characterization, the return difference matrix is

$$I + \tilde{L}(s, \varepsilon) = I + L(s) + \varepsilon \Delta_m(s) L(s) \tag{5.46}$$

or

$$I + \tilde{L}(s, \varepsilon) = A + B \tag{5.47}$$

with $A = I + L(s)$ and $B = \varepsilon \Delta_m(s) L(s)$. For the perturbed system to be unstable, viewed through a change in the number of encirclements of the $\det\left[I + \tilde{L}(s, \varepsilon)\right]$, the matrix $A + B$ must be singular for some $\varepsilon \in [0, 1]$ and $s \in D_R$. We know that $A = I + L(s)$ is nonsingular (the return difference matrix of the nominal design) since the nominal design is closed-loop stable. Thus, if the uncertainty is going to create instability, then the matrix $B = \varepsilon \Delta_m(s) L(s)$, when added to A, must make $A + B$ singular.

5.3.5 *A + B Argument*

The minimum singular value $\underline{\sigma}(A)$ measures the near singularity of the matrix A. Assume that the matrix $A + B$ is singular. If $A + B$ is singular then $A + B$ is rank deficient. Since $A + B$ is rank deficient, then there exists a vector $x \neq 0$ with unit magnitude $\left(\|x\|_2 = 1\right)$ such that $(A + B)x = 0$ ($x$is in the null space of $A + B$). This leads to $Ax = -Bx$ with $\|Ax\|_2 = \|Bx\|_2$. Using the above singular value definitions in (5.20) and $\|x\|_2 = 1$, we obtain the following inequality.

$$\underline{\sigma}(A) \leq \|Ax\|_2 = \|Bx\|_2 \leq \|B\|_2 = \bar{\sigma}(B) \tag{5.48}$$

If the matrix $A + B$ is singular, then $\bar{\sigma}(A) \leq \bar{\sigma}(B)$. For $A + B$ to be nonsingular, $\underline{\sigma}(A) > \bar{\sigma}(B)$. This is precisely how the stability robustness tests are derived.

Theorem 5.4 Stability Robustness Theorem: Additive Uncertainty Model

The polynomial $\phi'_{cl}(s)$ has no closed right half plane zeros and the perturbed feedback system is stable if the following hold:

1. *$\phi_{cl}(s)$ has no zeros in the closed right half plane.*
2. *$\underline{\sigma}(I + L(s)) > \bar{\sigma}(\Delta_a(s)) \forall s \in D_R$ and for all R sufficiently large, with $\Delta_a(s)$ given by (5.41).*

■

Theorem 5.5 Stability Robustness Theorem: Multiplicative Uncertainty Model

The polynomial $\phi'_{cl}(s)$ has no zeros in the closed right half plane and the perturbed feedback system is stable if the following hold:

1. *$\phi_{cl}(s)$ has no zeros in the closed right half plane.*
2. *$\underline{\sigma}(I + L^{-1}(s)) > \bar{\sigma}(\Delta_m(s)) \forall s \in D_R$ and for all R sufficiently large, with $\Delta_m(s)$ given by (5.42).*

■

The proof of this theorem uses the singularity of $A + B$ argument. Stability of the perturbed closed-loop system is guaranteed for a nonsingular $I + \tilde{L}(s, \varepsilon)$. Thus,

$$I + \tilde{L}(s, \varepsilon) = L(s)\big(I + L^{-1}(s) + \varepsilon\Delta_m(s)\big) \tag{5.49}$$

Here, we assume that $L^{-1}(s)$ exists. If $I + \tilde{L}(s, \varepsilon)$ is to be singular, then the matrix $I + L^{-1}(s) + \varepsilon\Delta_m(s)$ must be singular. Thus, to be nonsingular,

$$\underline{\sigma}\big(I + L^{-1}(s)\big) > \bar{\sigma}(\varepsilon\Delta_m(s)) \tag{5.50}$$

or

$$\underline{\sigma}\big(I + L^{-1}(s)\big) > |\varepsilon|\bar{\sigma}(\Delta_m(s))$$

$$\underline{\sigma}\big(I + L^{-1}(s)\big) > \bar{\sigma}(\Delta_m(s)) \tag{5.51}$$

Depending upon the model error characterization, either additive or multiplicative, the robustness test is different. Theorems 5.3 and 5.4 are sufficient tests for stability. As long as the singular value frequency responses do not overlap, stability is guaranteed.

Stability margins can be viewed as a multiplicative uncertainty, scaling the plant with some gain and phase. Singular value gain margins can be derived using the above theorems by assuming that the uncertainty matrix $\Delta_m(s)$ models these gain and phase uncertainties.

Consider the computation of a gain margin at the input to the plant. Place in each input channel a scalar gain $\varepsilon_i \in R$, with $E(s) = diag[\varepsilon_i] \in R^{n_u \times n_u}$ modeling these gains as a matrix. For the nominal condition with no uncertainty, $\varepsilon_i = 1$, the system is stable. Positive and negative gain margins would indicate how large and small, respectively, the scalar gain ε_i needs to be to destabilize the system. Our analysis problem will focus on independent uncertainties in each channel, with the gain margin relating to the smallest gain uncertainty that can destabilize the system. Figure 5.12 indicates how this gain uncertainty enters into the block diagram and how it can be represented using $\Delta(s)$.

Using the model indicated in Fig. 5.12, $\Delta(s) = E(s) - I$. For the nominal control system, let

$$\min_{\omega} \quad \underline{\sigma}(I + L^{-1}) = \beta_\sigma$$

From Theorem 5.5, stability is guaranteed if $\bar{\sigma}(I + L^{-1}(s)) > \bar{\sigma}(\Delta(s))$. For $\Delta(s) = E(s) - I$, $E(s) \in R^{n_u \times n_u}$, the singular values of $\Delta(s)$ are

$$\sigma_i(\Delta(s)) = \sigma_i(E(s) - I) = |\varepsilon_i - 1|$$

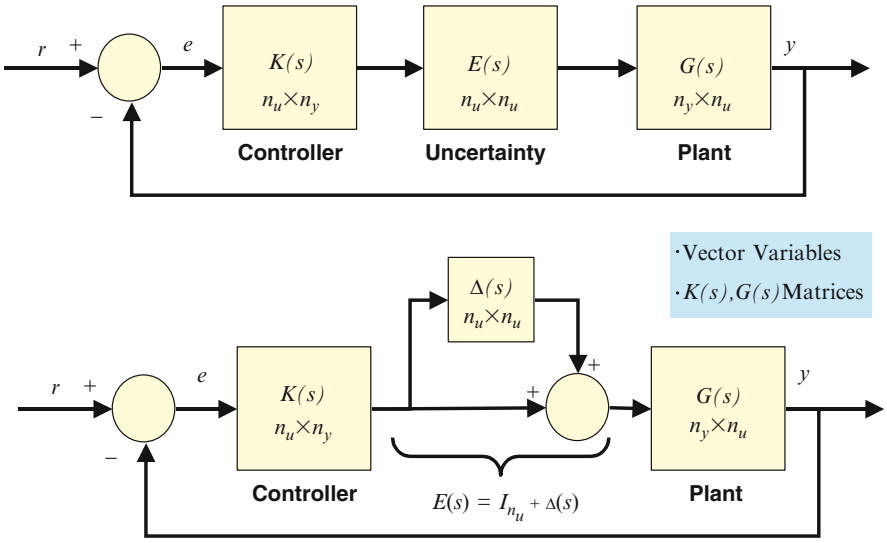


Fig. 5.12 Control system under uncertainty

If the largest $|\varepsilon_i - 1|$ is smaller than β_σ, then for $\varepsilon_i \in R$

$$1 - \beta_\sigma \leq \varepsilon_i \leq 1 + \beta_\sigma \tag{5.52}$$

which guarantees a gain margin of $[1 - \beta_\sigma, 1 + \beta_\sigma]$ for the system. If we consider the phase margin problem, $\varepsilon_i = \exp(j\phi_i(\omega))$, $\phi_i(\omega) \in R$, $E(s) = diag[\exp(j\phi_i(\omega))] \in C^{n_u \times n_u}$, then

$$\begin{aligned}
|\varepsilon_i - 1| &= \left|e^{j\phi_i(\omega)} - 1\right| \leq \beta_\sigma \\
&= |\cos(\phi_i(\omega)) - 1 + j\sin(\phi_i(\omega))| \leq \beta_\sigma \\
&= \left(\cos^2(\phi_i(\omega)) - 2\cos(\phi_i(\omega)) + 1 + \sin^2(\phi_i(\omega))\right)^{\frac{1}{2}} \leq \beta_\sigma \\
&= (2(1 - \cos(\phi_i(\omega))))^{\frac{1}{2}} \leq \beta_\sigma \\
&= \left(4\sin^2\left(\frac{\phi_i(\omega)}{2}\right)\right)^{\frac{1}{2}} \leq \beta_\sigma
\end{aligned}$$

which guarantees a phase margin of $\pm 2\sin^{-1}\frac{\beta_\sigma}{2}$ for the system.

5.3.6 Singular Value Stability Margins

1. Return Difference Matrix
   Let $\min_{\omega} \ \underline{\sigma}(I+L) = \alpha_\sigma$, then

$$\text{GM}_{I+L} = \left[\frac{1}{1+\alpha_\sigma}, \frac{1}{1-\alpha_\sigma}\right]; \quad \text{PM}_{I+L} = \pm 2\sin^{-1}\frac{\alpha_\sigma}{2} \tag{5.53}$$

2. Stability Robustness Matrix
   Let $\min_{\omega} \ \underline{\sigma}(I+L^{-1}) = \beta_\sigma$, then

$$\text{GM}_{I+L^{-1}} = [1-\beta_\sigma, 1+\beta_\sigma]; \quad \text{PM}_{I+L^{-1}} = \pm 2\sin^{-1}\frac{\beta_\sigma}{2} \tag{5.54}$$

$$\text{GM} = \text{GM}_{I+L} \cup \text{GM}_{I+L^{-1}}, \qquad \text{PM} = \text{PM}_{I+L} \cup \text{PM}_{I+L^{-1}} \tag{5.55}$$

Note that the best minimum singular value from the return difference matrix is $\min_{\omega} \ \underline{\sigma}(I+L) = \alpha_\sigma = 1$ (at high frequencies $L \to 0$). Substituting this into (5.53) produces a gain margin interval of $\text{GM}_{I+L} = \left[\frac{1}{2}, +\infty\right]$. Converting to decibels produces $\text{GM}_{I+L} = [-6, +\infty]$ dB. Similarly, the best minimum singular value from the stability robustness matrix is $\min_{\omega} \ \underline{\sigma}(I+L^{-1}) = \beta_\sigma = 1$ (at low frequencies $L^{-1} \to 0$). Substituting this into (5.54) produces a gain margin interval of $\text{GM}_{I+L^{-1}} = [0, 2]$. Converting to decibels produces $\text{GM}_{I+L^{-1}} = [-\infty, +6]$ dB.

Example 5.2 Gain and Phase Margins Using Singular Values Consider the unmanned aircraft presented in Example 5.1 which is controlled using the classical proportional-plus-integral control architecture shown in Fig. 5.3. Since the pitch-plane dynamics has a single input, we can use this example to compute both classical and singular value stability margins and relate them to each other. This will provide insight into how the singular value margins can be interpreted.

We will use a high-speed open-loop unstable flight condition and will add a second-order actuator model on the elevator. The dynamics are

$$\begin{aligned} \dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta_e + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta_e \\ \ddot{\delta}_e &= -2\zeta\omega_n\dot{\delta}_e - \omega_n^2(\delta_e - \delta_c) \end{aligned} \tag{5.56}$$

The actuator natural frequency is $\omega_n = 113$ rad/s with a damping factor $\zeta = 0.6$. The feedback variables from the inertial measurement unit are acceleration A_z and pitch rate q, where $A_z = Z_\alpha \alpha + Z_\delta \delta$. The plant model is

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u \end{aligned} \tag{5.57}$$

The numerical values for the matrices are

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \left[\begin{bmatrix} -1.3046 & 1.0 & -0.21420 & 0 \\ 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 0 & -1.2769.0 & -1.356 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12769.0 \end{bmatrix} \right.$$
$$\left. \begin{bmatrix} -1156.9 & 0 & -189.95 & 0 \\ 0 & 1.0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right] \tag{5.58}$$

The controller model is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{5.59}$$

with the matrices defined in (5.14). The gains are $K_a = -0.0015, K_q = -0.32$, $a_q = 2.0$ and $a_z = 6.0$. Substituting these values into (5.14) yields

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \left[\begin{bmatrix} 0 & 0 \\ -1.92 & 0 \end{bmatrix} \quad \begin{bmatrix} 0.0030 & 0 \\ -0.0029 & 1.92 \end{bmatrix} \quad \begin{bmatrix} -0.0030 \\ 0.0029 \end{bmatrix} \right.$$
$$\left. \begin{bmatrix} -0.32 & 1.0 \end{bmatrix} \quad \begin{bmatrix} -0.0005 & 0.32 \end{bmatrix} \quad [0.00048] \right] \tag{5.60}$$

Next, we will connect the controller to the plant model (see Eqs. (1.43) and (1.45)) and will simulate the closed-loop system to verify the model is correctly connected and then use the loop gain models (see Eqs. (1.49) and (1.53)) to compute the necessary frequency responses. Figure 5.13 shows a step response of the closed-loop system indicating the plant and controller are connected properly.

For this single-input single-output system, we will compute the Nyquist, Bode, $\underline{\sigma}(I + L)$, and $^{-}\sigma(I + L^{-1})$ at the plant input, and $\bar{\sigma}(S)$ and $\bar{\sigma}(T)$ at the plant output. Figures 5.14, 5.15, 5.16, and 5.17 show the plant input frequency response curves (Nyquist, Bode, $\underline{\sigma}(I + L)$, and $\underline{\sigma}(I + L^{-1})$). On the Nyquist plot in Fig. 5.14, we have drawn a circle centered at $(-1, j0)$ that has radius equal to the minimum of $\underline{\sigma}(I + L)$ (from Fig. 5.16). The classical gain and phase margins from Fig. 5.14 are 8.8 dB (2.7536) and 50°. These are also easily extracted from the Bode plot in Fig. 5.15. From Figs. 5.16 and 5.17, we have

$$\alpha_\sigma = \min \underline{\sigma}(I + L) = 0.5676; \quad \beta_\sigma = \min \underline{\sigma}(I + L^{-1}) = 0.7305 \tag{5.61}$$

These minimums versus frequency are substituted into the singular value gain and phase margins:

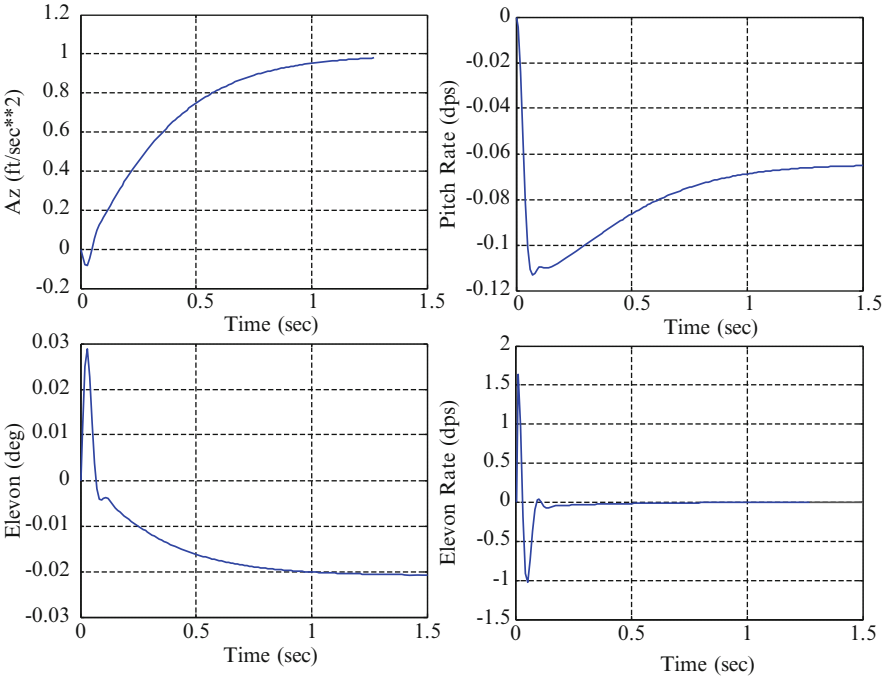


Fig. 5.13 Acceleration step response for Example 5.2

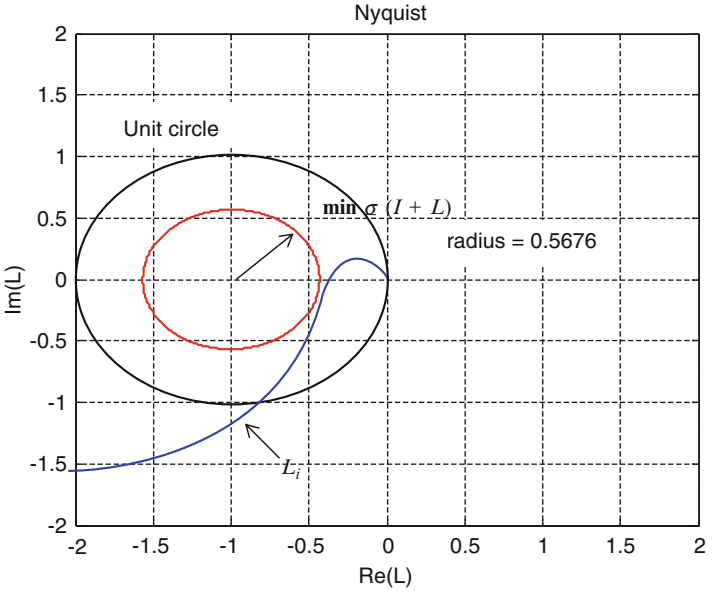


Fig. 5.14 Nyquist plot at the plant input loop break point

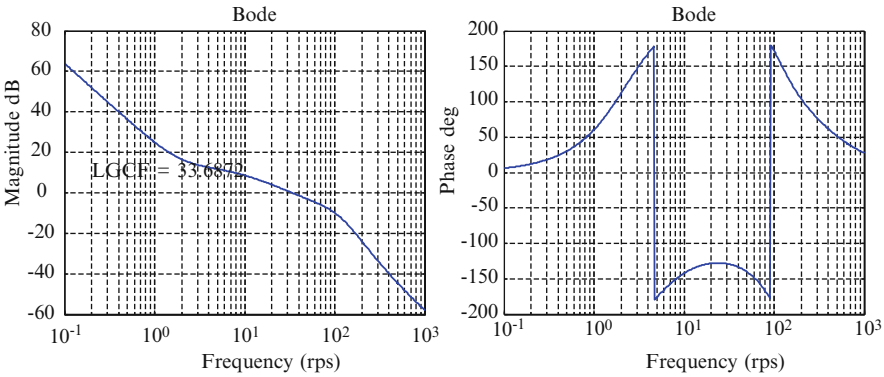


Fig. 5.15 Bode plot at the plant input loop break point

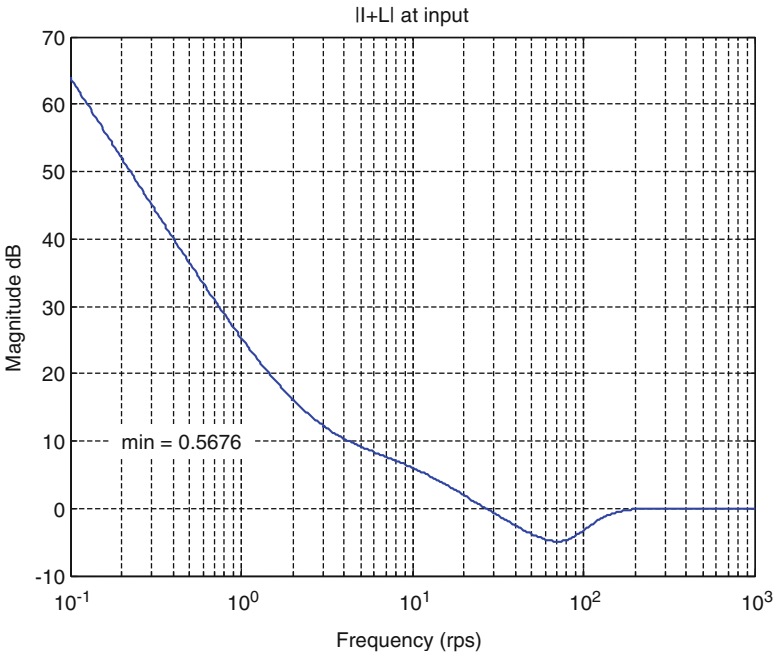


Fig. 5.16 $\underline{\sigma}(I+L)$ versus frequency at the plant input loop break point

$$\mathrm{GM}_{I+L} = \left[\frac{1}{1+\alpha_\sigma}, \frac{1}{1-\alpha_\sigma}\right]; \quad \mathrm{PM}_{I+L} = \pm 2\sin^{-1}\frac{\alpha_\sigma}{2} \tag{5.62}$$

$$\mathrm{GM}_{I+L^{-1}} = [1-\beta_\sigma, 1+\beta_\sigma]; \quad \mathrm{PM}_{I+L^{-1}} = \pm 2\sin^{-1}\frac{\beta_\sigma}{2} \tag{5.63}$$

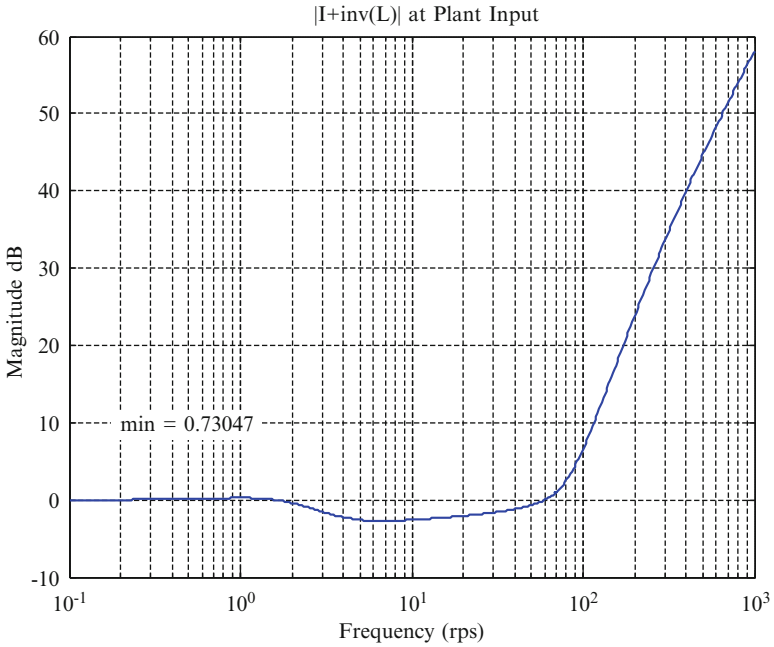


Fig. 5.17 $\underline{\sigma}(I + L^{-1})$ versus frequency at the plant input loop break point

Substituting yields

$$\begin{aligned} \text{GM}_{I+L} &= [0.6379 \quad 2.3127]; \\ &= [-3.9 \quad 7.28] \text{ dB} \end{aligned} \quad \text{PM}_{I+L} = \pm 32.97^\circ \tag{5.64}$$

$$\begin{aligned} \text{GM}_{I+L^{-1}} &= [0.4324 \quad 1.5676]; \\ &= [-7.28 \quad 3.90] \text{ dB} \end{aligned} \quad \text{PM}_{I+L^{-1}} = \pm 42.84^\circ \tag{5.65}$$

The singular value stability margins from (5.55) as computed from the singular values of $I + L$ and $I + L^{-1}$ are

$$\text{GM} = [-7.28 \quad 7.28] \text{dB}; \text{ PM} = \pm 42.84^\circ \tag{5.66}$$

Note that the classical margins from Figs. 5.14 and 5.15 are larger. The singular value stability margins are always more conservative than the single-loop classical margins.

Figures 5.18 and 5.19 show the sensitivity S and complementary sensitivity T output frequency response curves formed at the output loop break point. The sensitivity is the inverse of the return difference at the plant output. The infinity

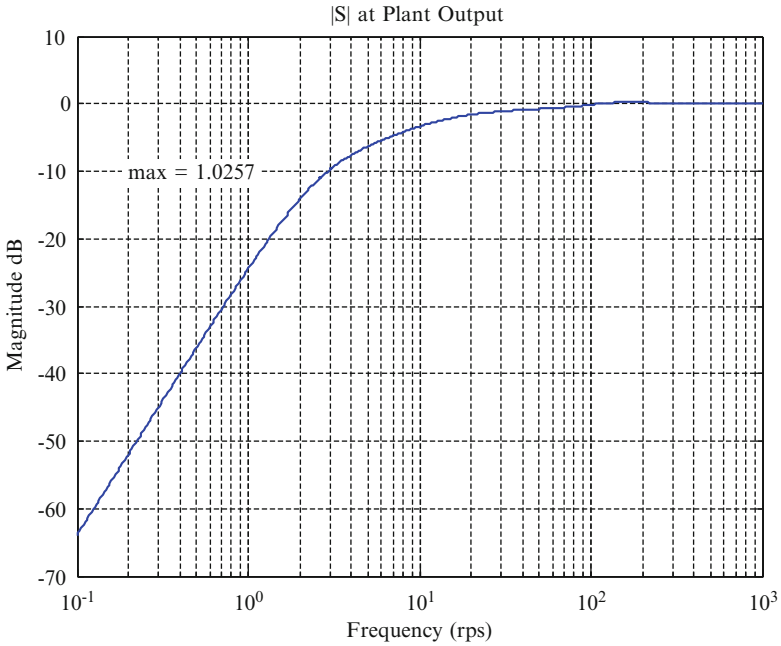


Fig. 5.18 $\bar{\sigma}(S)$ versus frequency at the plant output loop break point for this acceleration command system

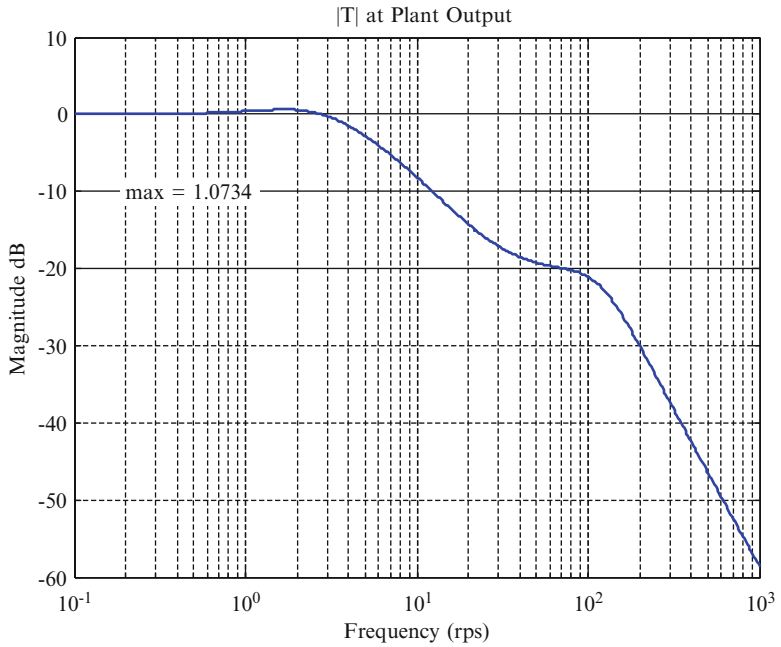


Fig. 5.19 $\bar{\sigma}(T)$ versus frequency at the plant output loop break point for this acceleration command system

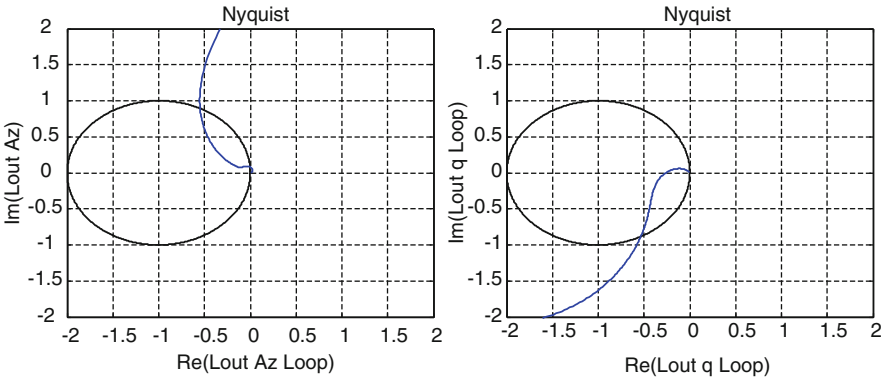


Fig. 5.20 Nyquist plots for the acceleration and pitch rate loops at the plant output

norm $\|S\|_\infty$ is equivalent to the minimum of $\underline{\sigma}(I + L_o)$ (they are inversely related). From Fig. 5.18, $\|S\|_\infty = 1.0257$ or 0.225 dB. This is a very small peak indicating good margins at the plant output. Figure 5.19 shows the complementary sensitivity $|T|$ which is the acceleration closed-loop transfer function. The $\|T\|_\infty = 1.0734$ (0.6152 dB) is a measure of the peak resonance in the acceleration loop. This is a small value also indicating good margins in this loop. If either $\|S\|_\infty$ or $\|T\|_\infty$ were large, this would indicate a problem in the design. In some multivariable systems, the margins at the plant input will be adequate, but at the plant output they are low. It is always prudent to check margins at all loop break points to make sure no sensitivity problems exist. Figure 5.20 shows Nyquist plots computed at the plant output for the acceleration and pitch rate feedback loops. Both plots show excellent stability margins. This directly relates to the excellent values of $\|S\|_\infty$ and $\|T\|_\infty$ in Figs. 5.18 and 5.19.

Figure 5.21 shows the frequency response of the controller $\bar{\sigma}(K)$. This figure indicates the amplification, or attenuation, of sensor noise through the controller. Although not directly related to stability margins, this frequency response should e examined to make sure the bandwidth of the controller is not too high and that high-frequency noise is not adversely amplified. The shape of the frequency response clearly shows the proportional-plus-integral control action that the controller is providing.

If noise amplification was a problem in the system, additional filtering using low-pass filters would be needed to clean up the feedback signals. These additional filters can be problematic as the phase lag causes issues with systems with low stability margins. Each integrator adds 90° of additional phase lag and can make the stabilization of unstable systems very difficult.

In the next section, we will explore the use of the structured singular value (SSV) μ in computing the robustness of control systems. The SSV is a very powerful analysis tool used to evaluate the stability robustness of systems to a variety of uncertainties.

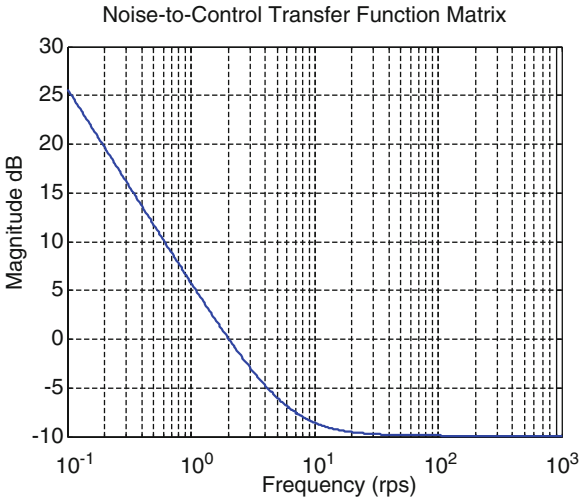


Fig. 5.21 $\bar{\sigma}(K)$ versus frequency

5.4 Control System Robustness Analysis

Control system sensitivity to uncertainties in dynamics has been a major focus for many years. In the past, the most widely used measure of stability robustness has been single-loop gain and phase margins derived from classical frequency response calculations. These methods and their singular value counterparts were discussed in the previous section.

Gain and phase margins provide significant insight into the robustness characteristics of the system. It has been proven many times in real-world applications that systems that have poor stability margins do not perform as desired. Requirements are typically levied onto the control design to provide 6-dB gain margin and at least 45° phase margin. Whether these margins be classical or singular value based, having adequate gain and phase margins, is an important aspect of control system design.

In the 1980s, significant research was performed on analyzing the robustness of control systems to neglected and mismodeled dynamics and real-parameter uncertainties. Analysis methods were developed to further analyze linear models to gain more insight into the controller's sensitivity to unmodeled dynamics, gain and phase uncertainties at different loop break points, and the sensitivity to parameter variations in the model. These methods all try to determine bounds on how large the uncertainties can be before the system would go unstable.

Many methods exist for solving the problem, all having differing amounts of conservatism in computing the robustness bounds. This conservatism exists due to the model of the uncertainties and how the uncertainties enter into the problem structure. Polynomial methods, singular value-based methods, and other frequency domain techniques were developed.

Table 5.2 Comparison of robustness analysis methods analyzing sensitivity to real-parameter variations

Robustness theory	Perturbation (%)
Small gain theorem	13.8
SSV μ	49.0
Stability hypersphere $Xp = \delta$	0.1
Stability hypersphere $p = A\alpha + b$	20.3
Stability hypersphere (Lyapunov unscaled)	0.007
Stability hypersphere (Lyapunov scaled)	0.02
Kharitonov's theorem	15.3
DeGaston–Safonov real margin	60.44
Monte Carlo eigen analysis	60–61

Table 5.2 illustrates several methods applied to a pitch autopilot analysis problem [2], investigating how much uncertainty can be tolerated in the aerodynamic parameters before the system goes unstable. The results in [2] use the same system model presented in Example 5.2. In this problem, each of the aerodynamic stability derivatives $(Z_\alpha/V, Z_\delta/V, M_\alpha, M_\delta)$ was allowed to vary simultaneously using

$$\left(\frac{Z_\alpha}{V}(1 \pm \delta_1), \frac{Z_\delta}{V}(1 \pm \delta_2), M_\alpha(1 \pm \delta_3), M_\delta(1 \pm \delta_4)\right) \tag{5.67}$$

It was desired to compute the smallest variation in these parameters that would cause the system to be unstable.

Table 5.2 shows various algorithms and modeling techniques applied to this analysis problem. The exact answer is 60.44 %. This answer was varied by a Monte Carlo analysis as well as inserting the predicted uncertainties back into the system model to show that the closed-loop system had $j\omega$ poles. As shown in the figure, some of the methods were found to be quite conservative. The small gain theorem, the structured singular value (SSV), and the DeGaston–Safanov [3] real stability margin all produced reasonable results when applied to this aerospace problem and will be further discussed in this section.

5.4.1 Analysis Models for Uncertain Systems

Multivariable stability margins, also called singular value stability margins, are a natural extension of classical gain and phase margins. Consider the SISO system shown in Fig. 5.22. Gain and phase margins for this system are computed by inserting a gain and phase variation $k_i e^{i\phi_i}$ in between the controller $K(s)$ and plant $G(s)$ and solving for the gain k_i (with $\phi_i = 0$) and phase ϕ_i (with $k_i = 1$) that destabilizes the system. The multivariable extension of this concept will use the ΔM analysis model created from the system matrices, as shown in Fig. 5.7.

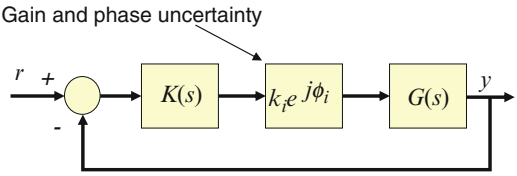


Fig. 5.22 System with plant input gain and phase uncertainty

Stability analysis models for multivariable systems can be formed to analyze gain and phase uncertainties, neglected and/or mismodeled dynamics, real-parameter uncertainties, and combinations thereof using methods identical to forming models for single-input single-output systems. These models can be easily formed using block diagram algebra, signal flow graph methods, or algebraic manipulation of loop equations. The resulting models will have a "structure" associated with them depending upon the specific problem, and the analysis will depend upon the structure. Within these models, the uncertainties in the system are usually isolated from the system models of the dynamics.

Figure 5.7 illustrated a general control system analysis model in which the matrix $\Delta(s)$ models uncertainties and $M(s)$ is a transfer function matrix modeling the dynamics between the output from the uncertainties to its input. We will use this ΔM representation of the dynamics for many of our stability analysis problems. The matrix $\Delta(s)$ will model gain and phase uncertainties in the system, that is, neglected and/or mismodeled dynamics, real-parameter uncertainties, or any combinations thereof. The matrix $M(s)$ will model the dynamics in the system that are assumed to be known.

For a control system under no uncertainty, the controller stabilizes the plant and the return difference matrix in nonsingular at all frequencies. Stability of the nominal system implies

$$\det[I + L(s)] \neq 0 \, \forall s \in D_R. \tag{5.68}$$

Using the ΔM analysis model shown in Fig. 5.7, (5.68) is equivalent to

$$\det[I - \Delta M(s)] \neq 0 \, \forall s \in D_R. \tag{5.69}$$

Under no uncertainty, $\Delta = 0$, clearly the system is stable. The analysis question is to determine how large can Δ be, while the system remains stable.

Example 5.3 Consider the ΔM analysis model shown in Fig. 5.7 and a stability analysis problem for a system as depicted in Fig. 5.23. The control system block diagram in Fig. 5.23 shows uncertainties Δ_1 at the input to the plant and uncertainties Δ_2 at the output of the plant. The uncertainties Δ_1 and Δ_2 can be constructed to model any type of uncertainty, depending upon the analysis question at hand. Δ_1 could model actuator uncertainties, unmodeled dynamics, time delays, or any plant input uncertainty, while Δ_2 could model sensor uncertainties, unmodeled dynamics, time delays, or any plant output uncertainty. Block diagram algebra is used to transform the system shown in Fig. 5.23 into the ΔM analysis

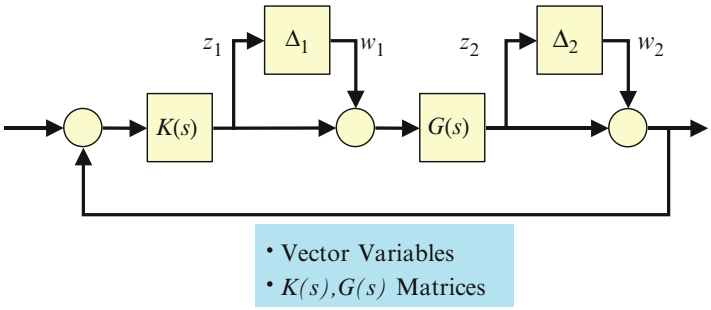


Fig. 5.23 Control system with simultaneous uncertainty at input and output

model. The matrix $\Delta(s)$ will be a block diagonal matrix, with each matrix or scalar uncertainty in the system located on the diagonal of $\Delta(s)$. The matrix $M(s)$ is a block matrix where the ij-th block is the transfer function matrix between from the output of the j-th uncertainty $\Delta_j(s)$ to the input of the i-th uncertainty $\Delta_i(s)$.

Consider the loop equations from Fig. 5.23 written as

$$\begin{aligned} z_1 &= K(s)(z_2 + w_2) \\ z_2 &= G(s)(z_1 + w_1) \end{aligned}$$

Substituting the z_2 expression into the z_1 equation and manipulating yields

$$\begin{aligned} z_1 &= K(s)(G(s)(z_1 + w_1) + w_2) \\ z_1 &= K(s)G(s)(z_1 + w_1) + K(s)w_2 \\ (I - K(s)G(s))z_1 &= K(s)G(s)w_1 + K(s)w_2 \\ z_1 &= (I - K(s)G(s))^{-1}K(s)G(s)w_1 + (I - K(s)G(s))^{-1}K(s)w_2 \end{aligned}$$

Substituting the z_1 expression into the z_2 equation and manipulating yields

$$\begin{aligned} z_2 &= G(s)(K(s)(z_2 + w_2) + w_1) \\ z_2 &= G(s)K(s)(z_2 + w_2) + G(s)w_1 \\ (I - G(s)K(s))z_2 &= G(s)K(s)w_2 + G(s)w_1 \\ z_2 &= (I - G(s)K(s))^{-1}G(s)w_1 + (I - G(s)K(s))^{-1}G(s)K(s)w_2 \end{aligned}$$

Combining these two expressions and writing in matrix form yields

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} (I - K(s)G(s))^{-1}K(s)G(s) & (I - K(s)G(s))^{-1}K(s) \\ (I - G(s)K(s))^{-1}G(s) & (I - G(s)K(s))^{-1}G(s)K(s) \end{bmatrix}}_{M(s)} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{5.70}$$

The loop equations for the uncertainties modeled in the system can be written as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}}_{\Delta(s)} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{5.71}$$

From this example, we see that the Δ matrix in (5.71) is block diagonal and has structure.

5.4.1.1 Real-Parameter Uncertainties

The above example essentially uses block diagram algebra to form the analysis model isolating the uncertainties from the nominal closed-loop system. When considering uncertainties in the real parameters contained in the state-space matrix A, working through the algebra can be tedious and error prone. This problem is easily solved for system linear in the uncertain parameters using a method [4, 5] that factors out the uncertain parameters in the closed-loop matrix A_{cl}, and decomposes the matrices using a singular value decomposition to form the B and C matrices for the model $M(s)$. The n uncertain parameters in A_{cl} are modeled as

$$p_i = \bar{p}_i(1 \pm \delta_i) \tag{5.72}$$

where p_i is the uncertain parameter with $\bar{p}_i$ its nominal value and δ_i the uncertainty. The closed-loop system is written isolating the uncertainties as follows:

$$A_{cl} = A_0 + \sum_{i=1}^{n} E_i \delta_i \tag{5.73}$$

where A_0 is the nominal closed-loop matrix and the matrices E_i factor out the uncertainties. The matrices E_i are the structural definitions for each of the parameter perturbations δ_i, with the rank of the matrix used to describe the parameter uncertainty. Using this model, decompose each $n \times n$ matrix E_i using a singular value decomposition. This gives

$$E_i = U \Sigma V^* \tag{5.74}$$

The matrix Σ will have k nonzero singular values, where k is equal to the rank of the matrix, with the remaining $n - k$ singular values equal to zero. Discard the zero singular values, and make Σ a $k \times k$ diagonal matrix containing only the nonzero singular values. We can write (5.74) with this new Σ as

$$E_i = \beta_i \alpha_i \tag{5.75}$$

where $\beta_i = U\Sigma^{1/2}$ and $\alpha_i = \Sigma^{1/2}V^*$. The matrices β_i and α_i in (5.75) depend only on the magnitude of the i-th nominal parameter. By using the decomposition described in (5.75) , we can replace $E_i\delta_i$ in (5.73) with $\beta_i\alpha_i\delta_i$. By using this modeling approach, we can separate out the parameter variations, δ_i, form $\Delta = diag[\delta_i]$ and create the nominally stable $M(s) = C_M(sI - A_M)^{-1}B_M$. The state-space triple (A_M, B_M, C_M) for *M(s)* is formed as follows.

Consider rank 1 perturbations only. β_i is $n \times 1$ and α_i is $1 \times n$. Then, (5.73) is

$$A_{cl} = A_0 + \sum_{i=1}^{n} \beta_i\alpha_i\delta_i \tag{5.76}$$

with $-1<\delta_i<1$. Write the closed-loop system as

$$\dot{x} = A_0x + \sum_{i=1}^{n} \beta_i u_i \tag{5.77}$$

where the u_i are input variables. Let the output y for this system be defined as

$$y = \sum_{i=1}^{n} \alpha_i x = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} x \tag{5.78}$$

Then, $u_i = \delta_i y_i$ and we can close the loop with

$$u_i = \delta_i y_i = \delta_i\alpha_i x \tag{5.79}$$

Substituting (5.79) into (5.77) yields

$$\begin{aligned} \dot{x} &= A_0x + \sum_{i=1}^{n} \beta_i\delta_i\alpha_i x \\ &= \left(A_0 + \sum_{i=1}^{n} \beta_i\delta_i\alpha_i\right)x = A_{cl}x \end{aligned} \tag{5.80}$$

which is the closed-loop system model. We can write a state-space triple (A_M, B_M, C_M) for this system as

$$A_M = A_0; B_M = [\beta_1 \cdots \beta_n]; C_M = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{5.81}$$

This triple describes the*M*matrix in the ΔM analysis model (Fig. 5.8).

Example 5.4 Consider the longitudinal dynamics from Example 5.2. An angle-of-attack controller is designed, and we desire to form an analysis model to analyze the uncertain aerodynamic parameters using the above method. The α control law is designed using the robust servomechanism approach from Chap. 3. This analysis will determine how sensitive the LQR control law is to knowing the aerodynamic stability derivatives used in the design model. The longitudinal dynamics with second-order actuator model are

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta_e + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta_e \\ \ddot{\delta}_e &= -2\zeta\omega_n\delta_e - \omega_n^2(\delta_e - \delta_c)\end{aligned} \tag{5.82}$$

The robust servomechanism model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ from Chap. 3 is

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta}{V} & 0 \\ 0 & M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_a^2 & -2\zeta_a\omega_a \end{bmatrix} = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.21420 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & -4624.0 & -81.6 \end{bmatrix} \tag{5.83}$$

$$\tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 4624.0 \end{bmatrix} \tag{5.84}$$

The LQR penalty matrices used to design the controller are

$$Q = diag[\,464.16 \quad 0 \quad 0 \quad 0 \quad 0\,]; R = 1 \tag{5.85}$$

with the resulting state feedback gain matrix given as

$$K_c = [\,-21.544 \quad -3.8421 \quad -0.29392 \quad 0.32045 \quad 0.0021463\,] \tag{5.86}$$

The nominal closed-loop system matrix is

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta}{V} & 0 \\ 0 & M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\omega_a^2 k_1 & -\omega_a^2 k_2 & -\omega_a^2 k_3 & -\omega_a^2(1+\omega_a^2 k_4) & -2\zeta_a\omega_a - \omega_a^2 k_5 \end{bmatrix}; \tag{5.87}$$

Consider real-parameter uncertainties in the four aerodynamic coefficients $(Z_\alpha/V, Z_\delta/V, M_\alpha, M_\delta)$ in (5.87). The parameter uncertainty model is $p_i = \bar{p}_i(1 \pm \delta_i)$. Using the closed-loop system uncertainty model from (5.73), we factor out each δ_i and form the matrices E_i:

$$E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; E_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{Z_\delta}{V} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$E_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & M_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; E_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \tag{5.88}$$

The singular value decomposition for E_1 is used to form the first column in B_M and first row in C_M. Substituting the numerical values into (5.88), we have

$$E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.3046 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= U\Sigma V = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1.3046] [0 \quad -1 \quad 0 \quad 0 \quad 0] = \underbrace{\begin{bmatrix} 0 \\ 1.1422 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\beta_1} \underbrace{[0 \quad -1.1422 \quad 0 \quad 0 \quad 0]}_{\alpha_1} \tag{5.89}$$

Using this same approach for each E_i, the columns and rows of the matrices B_M and C_M are populated. The state-space triple (A_M, B_M, C_M) is then

$$A_M = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.2142 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275100. & 49059. & 3753. & -16861. & -163. \end{bmatrix} \tag{5.90}$$

$$B_M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & 10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad C_M$$

$$= \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix}$$

Now, using the parameter uncertainty models presented in this section, we will investigate analysis methods to determine the robust stability.

5.4.2 Singular Value Robustness Tests

A very quick and useful analysis method for analyzing stability robustness is to apply the small gain theorem. This method is accurate when the uncertainty modeling matrix $\Delta(s)$ is a full complex-valued matrix. That is, when the matrix has no structure and is complex. When the matrix has structure, as in (5.71), the small gain theorem can be quite conservative. The more the structure deviates from a full complex-valued matrix, the more conservative is the result.

The structured singular value (SSV), denoted as μ, was developed to reduce the conservatism of evaluating stability robustness for problems like those in (5.71) that have structure. By structuring the uncertainty model into a block diagonal matrix form, and applying the SSV μ-test, a less conservative estimate of stability robustness is obtained.

Stability under the presence of uncertainty, assuming the nominal system is stable, requires the return difference matrix to become singular under the uncertainty. The stability test is described in (5.69). The following singular value robustness tests are designed to examine the return difference matrix and determine when it becomes singular.

5.4.2.1 The Small Gain Theorem

Consider the stability robustness analysis problem for the ΔM analysis model shown in Fig. 5.7. The return difference matrix for this system is $I + L = I - \Delta M$. The analysis problem is to determine the "size" of the matrix $\Delta(s)$ such that the system transitions from stable to unstable. This says that the return difference matrix transitions from nonsingular to singular under the uncertainty.

The matrix $\Delta(s)$ can be a block diagonal (BD) matrix, with each matrix entry $\Delta_i(s)$ on the diagonal corresponding to a matrix of perturbations occurring in the system. The matrix $M(s)$ is the transfer function between the output of the perturbation to its input. It depends upon the controller $K(s)$, the plant model $G(s)$, and the structure of the perturbations. Matrix $M(s)$ is a block matrix where the ij-th block $M_{i,j}(s)$ is the negative of the transfer function from the output of $\Delta_i(s)$ to the input of $\Delta_j(s)$.

We can intuitively define the bound on the norm of $\Delta(s)$ by using the $A + B$ argument of the preceding section. If $\det[I - \Delta M(s)] = 0$, then from the $A + B$ argument, we know that

$$\underline{\sigma}[I] > \bar{\sigma}[\Delta M] \tag{5.91}$$

Using $\bar{\sigma}[\Delta M] < \bar{\sigma}[\Delta]\bar{\sigma}[M]$ and the fact that $\bar{\sigma}[I] = 1$, we obtain the bound on the uncertainty as

$$\bar{\sigma}[\Delta] < 1/\bar{\sigma}[M] \tag{5.92}$$

which is referred to as the small gain theorem. The small gain theorem (SGT) is a *sufficient* test for stability. If it is violated, the system may still be stable. The conservatism is introduced in the step where $\bar{\sigma}[\Delta M]$ is bounded above by $\bar{\sigma}[\Delta]\bar{\sigma}[M]$. This step loses all structural information inherent in the matrices. It models a worst case scenario in which $\Delta(s)$ is a full complex-valued matrix.

5.4.2.2 The Structured Singular Value μ

The structured singular value (SSV) μ analysis was developed by Doyle [6] to reduce the conservatism of evaluating stability robustness using unstructured singular value computations like the small gain theorem. Consider the control system with input and output uncertainties as shown in Fig. 5.23. Stability for the perturbed system is guaranteed only when the return difference dynamics remains nonsingular, that is,

$$\det[I - K(I - \Delta_2)\,G(I - \Delta_1)\,] \neq 0\,\forall\,\Delta_1, \Delta_2, \text{ and } s \in D_R \tag{5.93}$$

which is equivalent to

$$\det[I - \Delta M] \neq 0\,\forall \Delta = diag[\Delta_1, \Delta_2], \text{ and } s \in D_R \tag{5.94}$$

The definition of the SSV μ is

$$\mu_\Delta(M) = \begin{cases} \dfrac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \Delta, \det[I - \Delta M] = 0\}} & M \in C^{n\times n} \\ 0, \text{ if no } \Delta \in \Delta \text{ makes } I - \Delta M \text{ singular} \end{cases} \tag{5.95}$$

The computation of the SSV μ in (5.95) utilizes the structure for $\Delta(s)$ to develop a less conservative answer to the bound on the destabilizing uncertainty.

Consider the simplest structure for the uncertainty Δ that is a diagonal matrix whose diagonal is a complex scalar, that is,

$$\Delta = \{\delta I_n : \delta \in C\} \tag{5.96}$$

Substitute this Δ model into (5.94). Assuming the uncertainty destabilizes the system, the return difference matrix is singular and can be written as

$$(I - \Delta M)w = (I - \delta I_n M)w = \delta\left(\frac{1}{\delta} I - M\right)w = 0 \tag{5.97}$$

for arbitrary vector w. This simplest structure defines an eigenproblem, with the SSV μ from (5.95) given as

$$\mu_\Delta(M) = \bar{\rho}(M). \tag{5.98}$$

where $\bar{\rho}(M)$ is the maximum spectral radius of the matrix M. When the uncertainty is a full complex matrix, as shown in the previous section, the small gain theorem produces an accurate bound on the uncertainty, with the SSV μ given as

$$\mu_\Delta(M) = \bar{\sigma}(M) \tag{5.99}$$

So, for problems of arbitrary structure, that is, for a block diagonal Δ, the SSV μ will be bounded above and below by

$$\bar{\rho}(M) \leq\ \mu_\Delta(M) \leq\ \bar{\sigma}(M) \tag{5.100}$$

Commercial software is available for computing the SSV μ in Matlab®. This software bounds the SSV μ through optimization given by

$$\max_{Q} |\lambda_{\max}(QM)| < \mu < \inf_{D} \bar{\sigma}\left(DMD^{-1}\right) \tag{5.101}$$

In analysis problems where real uncertainties are analyzed, or problems where real and complex (dynamic) uncertainties are analyzed together, there can exist a spread between the bounds in (5.101). This introduces some conservatism in bounding the uncertainty in some problems. A method discussed in the next section exactly computes the bound on Δ when Δ contains only real parameters.

Example 5.5 The SSV μ is a very powerful analysis tool used by researchers, scientists, and engineers. In this example, we will demonstrate its use as an analysis tool to understand the robustness of a MIMO flight control system, in which uncertainty is modeled at the plant input and output. When tools like the small gain theorem are used to predict the robustness to uncertainty, conservative robustness bounds are produced. By using the SSV μ, more accurate predictions of robust stability can be obtained.

Consider the following lateral-directional flight control system. It is desired to track a stability axis roll-rate command p_{sc} while keeping sideslip angle β small. In this example, we will compare two controllers and their associated robustness properties.

The state-space model $\dot{x} = Ax + Bu$ for the lateral-directional dynamics is

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Y_\beta & s(\alpha) & -c(\alpha) \\ L_\beta & 0 & 0 \\ N_\beta & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{5.102}$$

where $s(\alpha) = \sin(\alpha)$; $c(\alpha) = \cos(\alpha)$; the state $x = [\beta \quad p \quad r]^T$ contains the sideslip angle, roll rate, and yaw rate; and the control $u = [\delta_a \quad \delta_r]^T$ contains the aileron and rudder commands. The system and control distribution matrices are

$$A = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix} \tag{5.103}$$

We will use the robust servomechanism infinite-time LQR control from Chap. 3 to design the controllers. The first controller uses a single integrator to track stability axis roll-rate commands. The LQR design model is

$$\dot{z} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix} z + \begin{bmatrix} 0 \\ B \end{bmatrix} u \tag{5.104}$$

where $z = \left[\int e_{p_s} \quad \beta \quad p \quad r\right]^T$ and $C_c = [0 \quad c(\alpha) \quad s(\alpha) \quad 0]$. The LQR penalty matrices are

$$Q = \begin{bmatrix} 1.7 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.105}$$

where the (1,1) element of Q penalizes the error in tracking the stability axis roll-rate command and the (2,2) element of Q penalizes the sideslip angle. Solving the algebraic Riccati equation the resulting state feedback gain matrix is

$$K_c = \begin{bmatrix} -0.7852 & -2.0536 & -0.0797 & 0.0458 \\ 1.0409 & 3.8238 & 0.1280 & -0.1020 \end{bmatrix} \tag{5.106}$$

The closed-loop system matrix, $A_{cl} = \tilde{A} - \tilde{B}K_c$, is

$$A_{cl} = \begin{bmatrix} 0 & 0 & 0.99452 & 0.10453 \\ 0.38402 & 1.2836 & 0.14698 & -1.0283 \\ -76.704 & 337.46 & -8.5278 & 5.8489 \\ 704.55 & 2442.0 & 83.351 & -62.971 \end{bmatrix} \tag{5.107}$$

with eigenvalues $\lambda_{1,2} = -12.1814 \pm 22.1215j \; \lambda_{3,4} = -22.9261 \pm 11.8584j$. Figure 5.24 shows a step response commanding a stability axis roll rate $p_s = p\cos(\alpha) + r\sin(\alpha)$, and the response of the sideslip β, roll rate p, and yaw rate r. Figure 5.25 shows the frequency response analysis at the plant input where $\bar{\sigma}(L), \underline{\sigma}(I+L)$, and $\underline{\sigma}(I+L^{-1})$ are plotted versus frequency and the loop gain crossover frequency ω_c and singular value stability margins are computed. As discussed earlier in Chap. 2, the LQR state feedback design has excellent stability margins at the plant input. The dip in $\underline{\sigma}(I+L^{-1})$ has a minimum value or $\min \underline{\sigma}(I+L^{-1}) = 0.5815$. A typical requirement is to keep this minimum above 0.5.

We desire to investigate this control system robustness to simultaneous uncertainties at the plant input and plant output and demonstrate how the structure of the uncertainties impacts the analysis. We will use the ΔM analysis model for the block diagram shown in Fig. 5.23. The plant model is

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u \end{aligned} \tag{5.108}$$

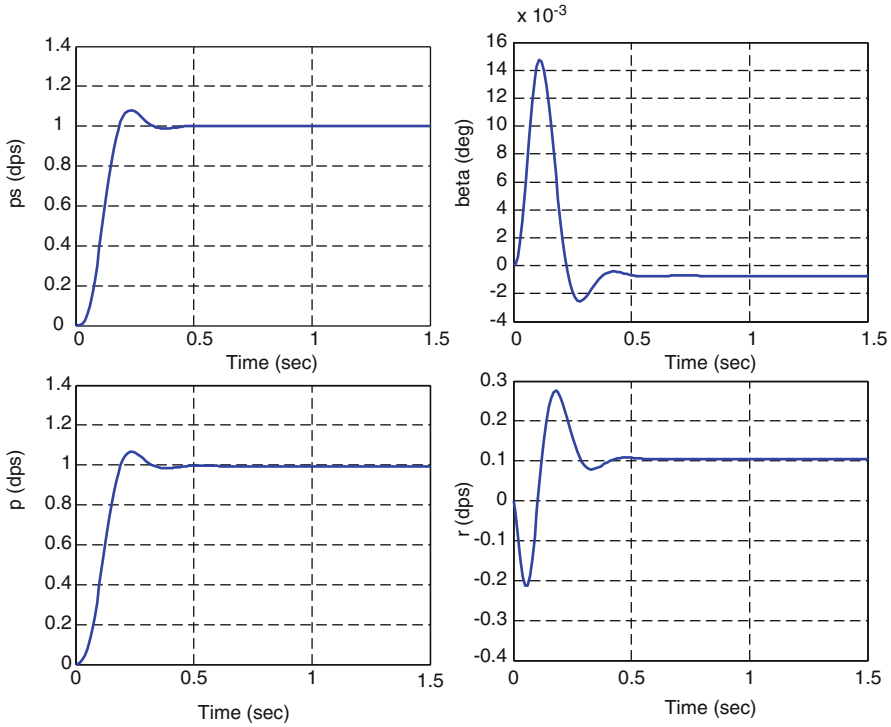


Fig. 5.24 Example 5.5 stability axis roll-rate step response time histories

where

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \left[\begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix} \right. \\ \left. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \right] \tag{5.109}$$

The RSLQR controller is modeled as

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{5.110}$$

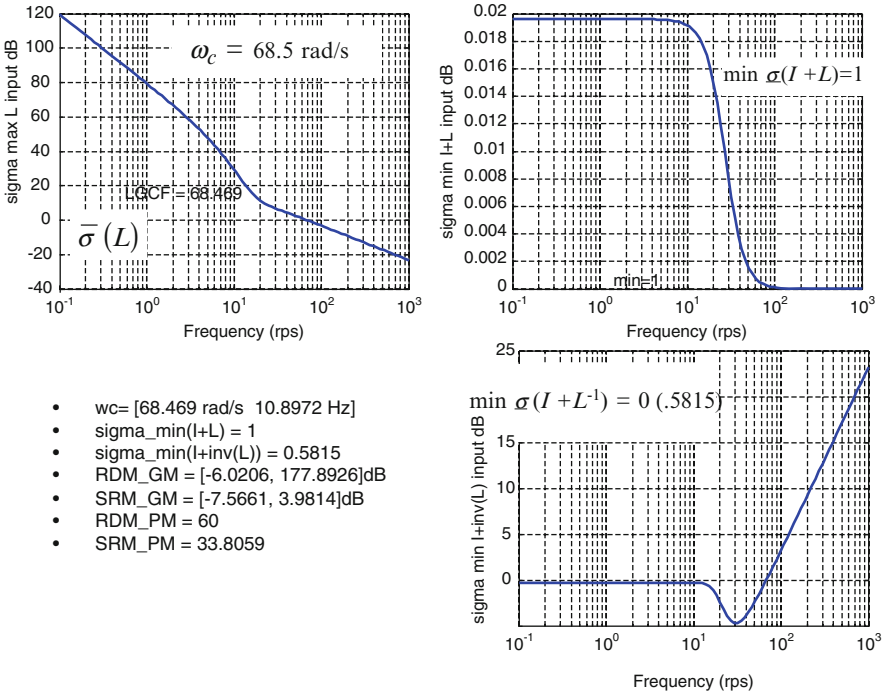


Fig. 5.25 Example 5.5 plant input frequency response analysis and stability margins

where

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & \begin{bmatrix} 0 & 0.9945 & 0.1045 \end{bmatrix} & \begin{bmatrix} -1 \end{bmatrix} \\ \begin{bmatrix} 0.7852 \\ -1.0409 \end{bmatrix} & \begin{bmatrix} 2.0536 & 0.0797 & -0.0458 \\ -3.8238 & -0.1280 & 0.1020 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix} \tag{5.111}$$

It is important when forming analysis models to check them for correctness. The plant (5.108) and controller (5.110) are connected to form a state-space model of M using the linear fractional transformation (*lft*) command in Matlab®. The closed-loop system has $n_x = 4$ states. The eigenvalues of the system are then compared to the eigenvalues of (5.107). This is a partial demonstration that the system is connected properly. Frequency domain models can also be checked (at the input and output) by evaluating the controller K and plant G at a given frequency and forming M as in (5.70). This matrix can be compared to the state-space model frequency response created using the (*lft*) command.

To begin the analysis, we will examine this MIMO control system robustness to plant input uncertainties Δ_1. The M matrix at this loop break point has dimensions 2×2. We will compute the SSVμ along with the small gain theorem (SGT) bound versus frequency. Figure 5.26 shows the SSVμ results for Δ_1 a full matrix and $\Delta_1 = diag$

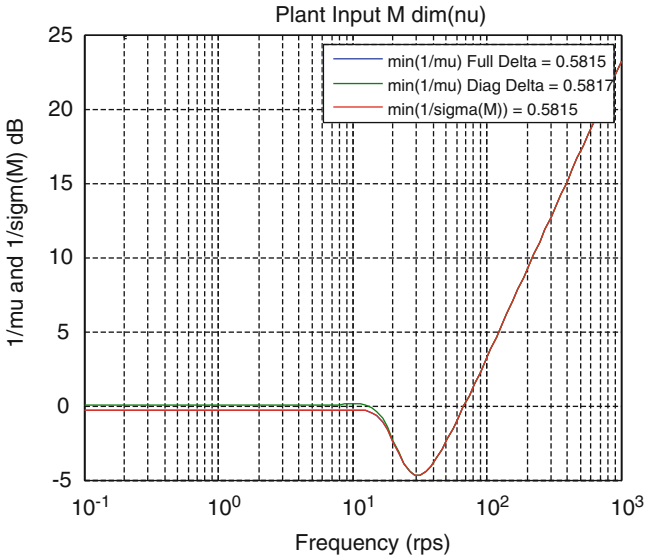


Fig. 5.26 Example 5.5 SSV μ and small gain theorem bounds at the plant input

$[\delta_1 \quad \delta_2]$ and the SGT bound $1/\bar{\sigma}(M)$. We do not see any difference at the plant input from the SSVμ and SGT bounds, and by restricting the uncertainty to be diagonal, further insight into this can be gained by examining the M matrix elements.

$$\begin{matrix} & \delta_a & \delta_r \\ \delta_a & M_{11} & M_{12} \\ \delta_r & M_{21} & M_{22} \end{matrix} \tag{5.112}$$

Figure 5.27 shows the magnitude of the elements of $M(j\omega)$. We see that the matrix is dominated by the (2,2) element. This implies that the directional β dynamics (versus the roll dynamics) driven by the rudder is establishing the bound. We see from Fig. 5.26 that by diagonalizing $\Delta_1 = diag[\delta_1 \quad \delta_2]$ does not alter the bound.

Next is the analysis at the plant output using just Δ_2. The M matrix at this loop break point has dimensions 3×3. Figure 5.28 shows the SSVμ results for Δ_2 a full matrix and $\Delta_2 = diag[\delta_1 \quad \delta_2 \quad \delta_3]$ and the SGT bound $1/\bar{\sigma}(M)$. Here, we see a much reduced bound as compared to the plant input loop break point. At the input, $\min(1/\mu) = 0.5815$ and was the same for both Δ_1 a full matrix and $\Delta_1 = \text{diag}[\delta_1 \quad \delta_2]$. Here, $\min(1/\mu) = 0.01389$ for Δ_2 a full matrix and $\min(1/\mu) = 0.18855$ when $\Delta_2 = diag[\delta_1 \quad \delta_2 \quad \delta_3]$. Restricting the Δ_2 matrix to be diagonal has a large impact. We would expect that the off-diagonal elements of $M(j\omega)$ are influencing the bound. Figure 5.29 shows the magnitude of the elements of $M(j\omega)$ for the diagonal element and off-diagonal elements and confirms this result. The M matrix at the plant output is

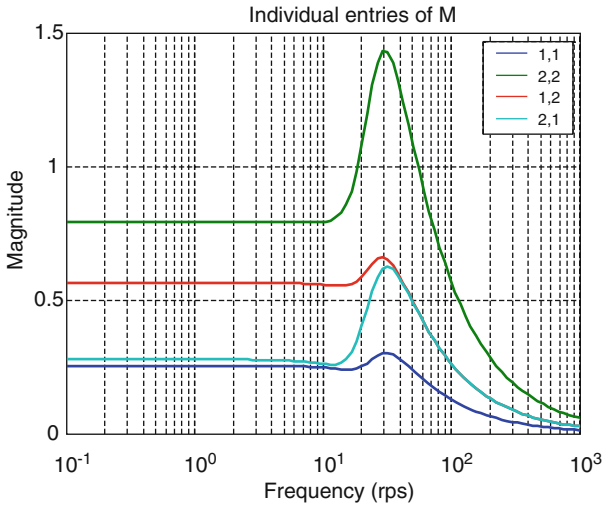


Fig. 5.27 Example 5.5 magnitudes of $M(j\omega)$ frequency response for plant input uncertainty

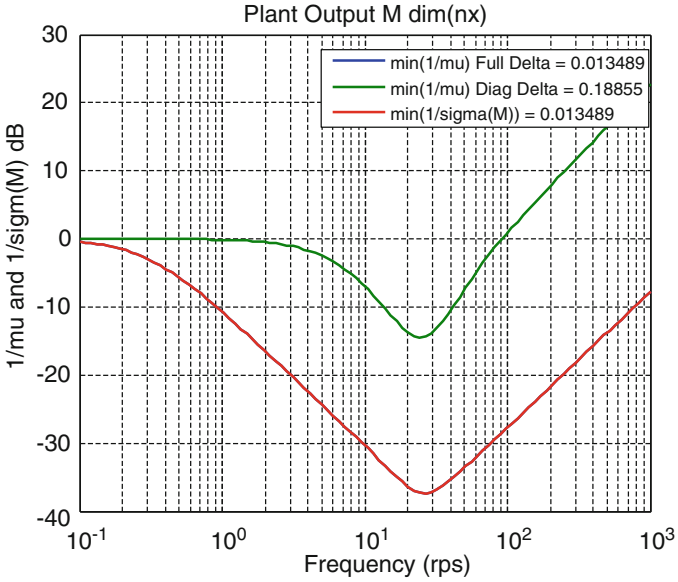


Fig. 5.28 Example 5.5 SSV μ and small gain theorem bounds at the plant input

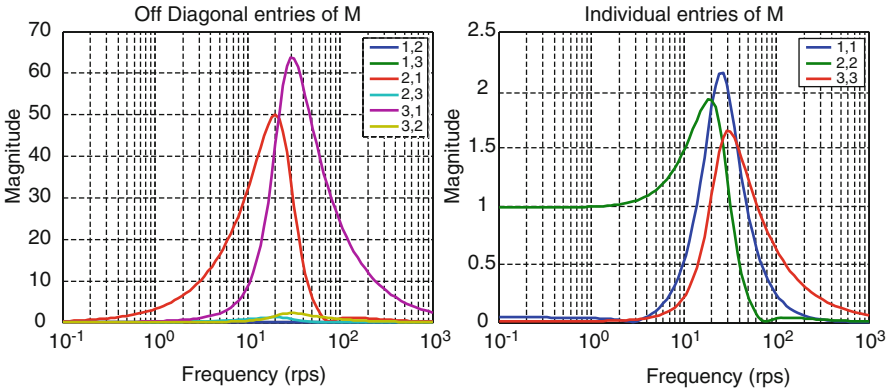


Fig. 5.29 Example 5.5 magnitudes of $M(j\omega)$ frequency response for plant output uncertainty

$$\begin{array}{c} \\ \beta \\ p \\ r \end{array} \begin{array}{c} \begin{array}{ccc} \beta & p & r \end{array} \\ \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \end{array} \tag{5.113}$$

We see from Fig. 5.29 that the 2,1 and 3,1 elements of $M(j\omega)$ are dominating the matrix. This also implies sensitivity in the directional β dynamics.

The last analysis is the combined plant input and plant output case, as shown in Fig. 5.23. The M matrix has dimensions 5×5. Figure 5.30 shows the SSVμ results for Δ a full matrix, a block diagonal matrix $\Delta = diag[\,\Delta_1 \quad \Delta_2\,]$, a diagonal matrix $\Delta = diag[\,\delta_1 \quad \cdots \quad \delta_5\,]$, and the SGT bound $1/\bar{\sigma}(M)$. The curves for the SSVμ analysis of Δ_2 a full matrix and the SGT bound are identical. Introducing structure in Δ only slightly improves the bound. To better understand this, we examine the frequency response $M(j\omega)$. Figure 5.31 shows the magnitude of the elements of $M(j\omega)$ for the diagonal elements and off-diagonal elements. We see that the dominant entry in $M(j\omega)$ is the (3,3) entry. This corresponds to the directional β dynamics, which are open-loop unstable. The rudder δ_r-to-β off-axis elements are the next largest elements. It is the rudder δ_r that primarily stabilizes the directional axis. This large (3,3) entry in $M(j\omega)$ is dominating the stability analysis and is why the block diagonal and diagonal Δ matrices did not produce larger bounds. This examination of $M(j\omega)$ confirms the robustness analysis indicating that this open-loop unstable vehicle is sensitive to uncertainties in the directional axis dynamics. Uncertainties in the air data system used to produce the feedback signal β should be examined in detail and in simulation.

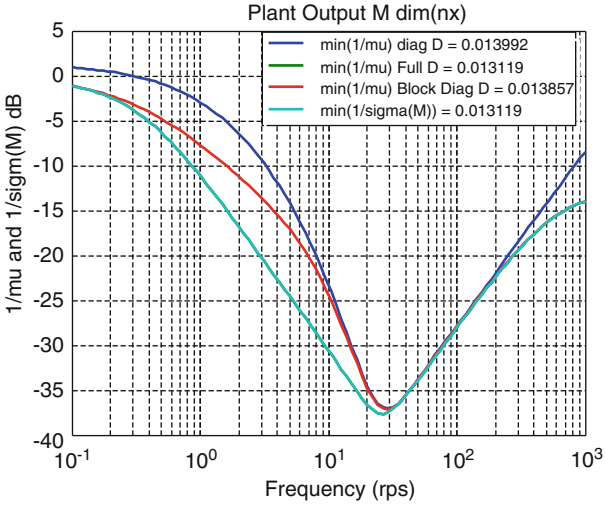


Fig. 5.30 Example 5.5 SSV μ and small gain theorem bounds for combined plant input and output uncertainties

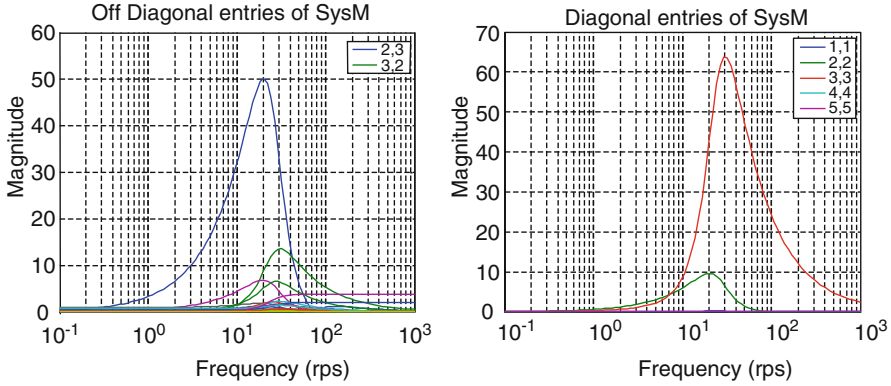


Fig. 5.31 Example 5.5 magnitudes of $M(j\omega)$ frequency response for combined plant input and output uncertainties

5.4.3 Real Stability Margin

An important question in the design of a control system is one that asks how well one must know the parameters in the model of the dynamics? This question arises due to the fact that the coefficients in the differential equations are seldom known exactly. A large gain margin gives some comfort that the system is robust, but it does not accurately predict the sensitivity of the controller to knowing the parameters.

The SSV μ discussed in the previous section is a method for measuring the sensitivity of the system to uncertainties. An additional measure called the real multiloop stability margin, or real margin, as defined by DeGaston and Safonov in [3], is a scalar quantity also interpreted as a gain margin. Its calculation gives a nonconservative measure of control system stability robustness to real-parameter variations modeled in the system dynamics.

In classical control system analyses, root locus plots can be used to graphically analyze a system's sensitivity to real-parameter variations. For a single parameter, this is a very easy and useful analysis to perform. In Kuo [7], the root locus approach is extended for multiple parameters. This approach has not been widely used due to its complexity and difficult graphical interpretation.

Consider a control system whose plant dynamics have uncertain dynamics. These uncertainties can arise from uncertain real-parameter variations, neglected/mismodeled dynamics, or combinations of both. In this section, we will focus on real-parameter uncertainty. To analyze the control system, we will transform the model into the ΔM analysis model in Fig. 5.7. The real-parameter uncertainties in the system are isolated and placed into a diagonal matrix Δ. The transfer matrix M describes nominal system characteristics that are stabilized by the controller. We assume that the system has adequate stability margins. Thus, for $\Delta = 0$ (no uncertainty), the system is stable, and the performance meets requirements.

Consider n uncertain real parameters represented by $\delta_i \in D_i \subseteq R$ where D_i is the domain of the i-th parameter. Let

$$\Delta = diag[\delta_1, \cdots, \delta_n] \tag{5.114}$$

and define the parameter space D as

$$D = D_1 \times D_2 \times \cdots \times D_n. \tag{5.115}$$

This parameter space describes the uncertain real parameter modeled in Fig. 5.7.

Using the multivariable Nyquist theorem in Sect. 5.2, the stability of the system described by Fig. 5.7 is implied by $\det[I - \Delta M] \neq 0$. The analysis problem is to find the largest parameter space D such that the system remains stable. This can be interpreted as finding the smallest uncertainty Δ that destabilizes the system. Consider the scalar stability margin k_m defined as

$$k_m = \min\{k \in [0, \infty) | \det[I - k\Delta M] = 0\} \tag{5.116}$$

If

$$(1/k_m)\delta_i \in D_i \quad \forall i \tag{5.117}$$

then ΔM remains stable for $\Delta \subseteq D$. This defines k_m as a multiloop stability margin.

A numerical algorithm for computing k_m converges by iterating lower and upper bounds on k_m, which are determined when either the convex hulls or interior points,

respectively, of certain image sets first intercept the origin. This development is made possible by the use of a mapping theorem taken from Zadeh and Desoer [8]. The multiloop stability margin is computed by finding the smallest k for which there exists $\Delta = diag[\delta_1, \ldots, \delta_n] \in D$ such that $\det[I - k\Delta M] = 0$.

The approach is as follows: Fix k. Map all the parameter space D into the complex plane with $\det[I - k\Delta M]$. If this region so mapped does not include the origin, then k is a lower bound on the stability margin k_m. Increment k positively until the origin is just included in the map. This would yield k_m exactly. However, computing the true image of D under the mapping $\det[I - kDM]$ is computationally prohibitive. To circumvent this problem, the convex hull of the image of D is used.

In general, the parameter space D will be an n-dimensional polytope having 2^n vertices. By scaling the parameter uncertainties and incorporating the scaling into M, a hypercube describing the parameter space may be used rather than a polytope. Define V_i as a vertex of the hypercube D, where $i = 1, \ldots, 2^n$ .The vertex V_i represents a corner of the hypercube. Let

$$V = \{V_1, V_2, \ldots, V_m\}, m = 2^n \tag{5.118}$$

denote the set of all hypercube vertices $V \subset D$. For example, consider three uncertain parameters. There are then $2^3 = 8$ vertices in the hypercube. They are

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix} \tag{5.119}$$

Let Δ_{V_i} be a matrix of parameter uncertainties made up of the vertex points $v_{i,j}$ as j is varied from 1 to n. This is described as

$$\Delta_{V_i} = diag\left[v_{i,j}, j = 1, \ldots, n\right] \tag{5.120}$$

From (5.119), $\Delta_{V_2} = diag[\,1 \quad 1 \quad -1\,]$. Define

$$\det[I - kDM] = \left\{ \begin{array}{l} z \in C | z = \det[I - k\Delta M] \forall \delta_i \in D_i, \\ \qquad\qquad i = 1, \ldots, n, \text{ with } k, M \text{ fixed} \end{array} \right\} \tag{5.121}$$

This set is a set of points that represents the entire hypercube solid being mapped into the complex plane through the determinant function. It describes the entire image of D (the image of the parameter uncertainties) in the Nyquist plane. Next, define

$$\det[I - kVM] = \{y_i \in C | y_i = \det[I - k\Delta_{V_i} M], i = 1, \ldots, n\} \tag{5.122}$$

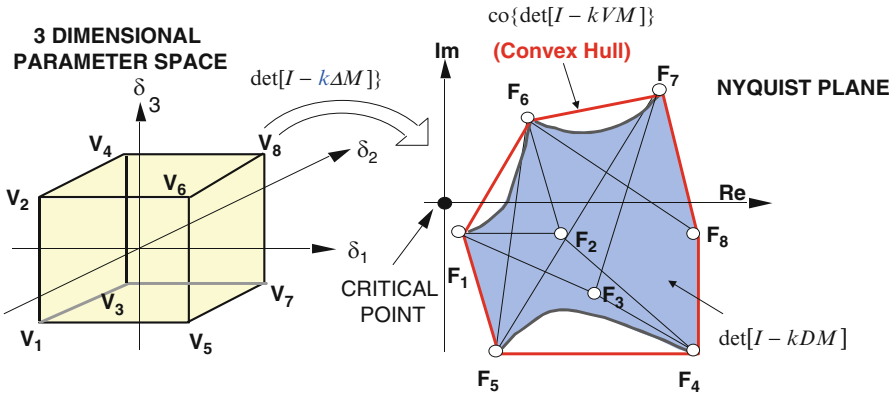


Fig. 5.32 Mapping the uncertain parameter space into the Nyquist plane

Equation (5.122) describes the set of points mapped into the complex plane made by the hypercube vertices. Let $F_i = \det[I - k\Delta_{V_i}M]$ be the mapping of the i-th vertex. F_i is a single point in the set $\det[I - kVM]$. With these definitions, we are now ready to state the following theorem.

Theorem 5.5 [3]. *Let* $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ *be defined as above. Fix* k. *Then,*

$$\det[I - kDM] \subset \text{co}\{\det[I - kVM]\} \tag{5.123}$$

■

This theorem states that the true image of the hypercube is contained in the convex hull created from the vertices. By mapping the 2^n vertices of D into the Nyquist plane and then constructing a convex hull about the 2^n points, a polygon is created that encompasses the $\det[I - kDM]$. Figure 5.32 illustrates this for a 3D hypercube.

The hypercube in Fig. 5.32 is the yellow box with the $2^3 = 8$ vertices arbitrarily labeled. Thus, $\Delta \subset D = D_1 \times D_2 \times D_3$ with $D_i = [\delta_{i_{\min}}, \delta_{i_{\max}}]$. The parameters $\delta_{i_{\min}}$ and $\delta_{i_{\max}}$ describe the lower and upper bounds of the parameter δ_i. By scaling these uncertainties and incorporating the scaling into M, we can model each parameter δ_i with limits of ± 1.

Figure 5.32 shows the mapping of this parameter-space hypercube into the complex plane using the determinant mapping function. The blue shaded region depicts the true image of the cube solid mapped into the Nyquist plane. If the origin was contained in the shaded region, then the system would be unstable. Since the origin is not in the shaded region, the gain margin k used in $\det[I - kDM]$ is smaller than the true stability margin and should be increased in magnitude until the origin is included.

The value of k such that the origin is just included in the shaded region is the exact stability margin k_m we seek to compute. In Fig. 5.32, the vertex points V_i are mapped into the F_i points. The convex hull containing the image of the hypercube is denoted as co{det$[I - kVM]$} and is shown as a heavy red border around the det $[I - kDM]$ image. We see from the figure that if co{det$[I - kVM]$} were used to determine k, conservatism would be present since the co{det$[I - kVM]$} contains more points than the true image of det$[I - kDM]$. This fact is used to define a lower bound on k_m, resulting in the following lemma.

Corollary 5.1. *Let M, D, and* det$[I - kDM]$ *be defined as given in Theorem 5.5. Then, for $k > k_0$,*

$$\det[I - k_0 DM] \subset \det[I - kDM] \tag{5.124}$$

$$\text{co}\{\det[I - k_0 DM]\} \subset \text{co}\{\det[I - kDM]\} \tag{5.125}$$

■

This corollary states that the image of the hypercube solid under the determinant mapping function, for k_0, is a subset of the image mapped using a larger k. Thus, the convex hull containing det$[I - k_0 DM]$ is contained in the convex hull co{det$[I - kDM]$}.

5.4.3.1 Lower Bound on the Stability Margin k_m

Application of Corollary 5.1 allows us to expand the co{det$[I - kDM]$} until the origin is enclosed. We show this graphically in Fig. 5.33. The solid lines represent co{det$[I - k_i VM]$} for $k_1, k_2,$ and k_3. As shown in the figure, for all $k < k_3$, the origin is not enclosed by co{det$[I - kVM]$}. Thus, $k_3 < k_m$ is a lower bound for the stability margin k_m. If k increases without co{det$[I - kVM]$} intercepting the origin, then $k_m = \infty$.

5.4.3.2 Upper Bound on the Stability Margin k_m

To compute the upper bound on k_m, the path between the vertices whose line segment intercepts the origin must be examined more closely. Define the following critical vertices:

Critical vertices: $F_i = \det[I - k\Delta_{V_i} M]$, $F_j = \det\left[I - k\Delta_{V_j} M\right]$, $i \neq j$, and $\beta \in [0, 1]$ such that $(1 - \beta)F_i + \beta F_j = 0$.
Isolated critical vertex (ICV): F_i is isolated if $F_i \neq F_j$, $i \neq j$.
Coincident critical vertex (CCV): F_i is coincident if $F_i = F_j$, $i \neq j$.

Critical vertices are defined as the two vertices whose line segment intercepts the origin. This is shown in Fig. 5.33 as the convex hull line segment $F_{53} - F_{63}$. These critical vertices are isolated if $F_i \neq F_j$. They are coincident if $F_i = F_j$. Let $m(i, j)$ be

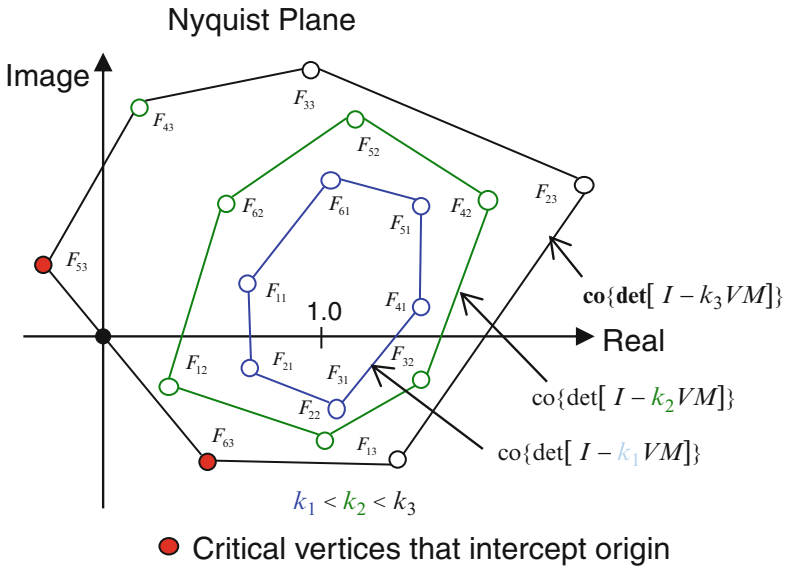


Fig. 5.33 Expanding the convex hulls until the origin is intercepted

equal to the number of differing coordinates of the two vertices V_i and V_j that are mapped by $\det[I - k\Delta_{V_i}M]$ to F_i and F_j. In Fig. 5.33, the critical vertices are F_{53} and F_{63}. The $m(i,j)$ is the minimum number of edges on the hypercube D from vertex V_i to V_j. In Fig. 5.33, $m(i,j) = 1$ examine the cube in Fig. 5.32 to see that vertex V_5 connects directly to V_6. The following corollary will aid in the calculation of the upper bound on k_m.

Corollary 5.2. *Any path along a single coordinate in D is mapped by* $\det[I - kDM]$ *to a straight line in the complex plane.*

■

For fixed M, the $\det[I - k\Delta M]$ for $\Delta = diag[\delta_1, \ldots, \delta_n] \in D$ is a polynomial in the variables δ_i and is affine with respect to each of the δ_i. This is true only for a diagonal Δ and is obtained by definition of the determinant. This affine relationship proves this lemma.

5.4.3.3 Comment

Corollary 5.2 guarantees that any point on the face of the hypercube D mapped into the Nyquist plane will be contained in the convex hull formed by the mapped vertices. This is true only for real parameters. If the parameters under variation were complex, any path along a single coordinate would trace an arc in the Nyquist plane. Thus, points contained on the face of a complex-parameter hypercube mapped into the Nyquist plane need not be contained in the convex hull formed

by the hypercube vertices. This fact precludes the use of this parameter-space method in analyzing complex-parameter variations.

Define a vertex path as any path between critical vertices F_i and F_j, consisting of $m(i,j)$ straight-line segments, defined by $\det[I - k\Delta_x M]$ as x progresses from V_i to V_j along the edges of the hypercube D. The first such vertex path to touch the origin defines a point in $\det[I - kDM]$, and the associated k is an upper bound on the stability margin k_m. The vertex path will determine the region in the parameter space that causes instability.

5.4.3.4 Convergence to k_m

The actual stability margin is computed by an iterative algorithm. It begins by examining the vertex paths between critical vertices (ICVs or CCVs) that intercept the origin. This defines the edges of the hypercube closest to the origin. The domain D is then split along this vertex path, creating sub-domains. Convex hulls around smaller and smaller sub-domains are computed. As the sub-domains become small, the union of all of the convex hulls for all of the sub-domains gets close to the actual image of the domain D. The accuracy in the computation of k_m is then dependent on how small the sub-domains are made. The following three lemmas are used to prove the convergence theorem that computes the exact multiloop real stability margin.

Lemma 5.1. *On a hypercube of dimension n with two vertices that differ by m coordinates, there are $m!$ paths between these two vertices along the edges of the hypercube. Each path between these two vertices will have $m + 1$ vertices along the path (including the original vertices).*

■

Lemma 5.2. *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as previously given. Let F_i and F_j be isolated critical vertices with $m(i,j) > 2$ and F_k denote the first vertex along a vertex path emerging from F_i. Define a point along the line segment between F_i and F_k as F_x, exclusive of the end points, that is, $F_x = (1 - \beta)F_i + \beta F_k, \beta \in (0,1)$. Let V_x be the associated point on the hypercube edge defined between V_i and V_k. Cut the domain D at V_x orthogonally to this edge to create two sub-domains D_1 and D_2, where $V_i \in D_1$ and $V_j \in D_2$. Then neither $\mathrm{co}\{\det[I - kD_1M]\}$ nor $\mathrm{co}\{\det[I - kD_2M]\}$ includes the origin.*

■

Lemma 5.3. *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as previously given. Then, there is at least one Δ_{V_i} associated with the stability margin k_m that assumes an extremal value.*

■

Lemma 5.1 is used to determine the number of vertex paths between critical vertices. These vertex paths define the coordinate direction in which the parameter-space domain D is split into sub-domains.

Lemma 5.2 is the heart of the convergence theorem used to obtain k_m. It is employed when $m(i,j)>2$. The utility of this lemma is best explained by an example. In Fig. 5.32, let vertex images F_1 and F_6 be isolated critical vertices, with k_l determined such that $\text{co}\{\det[I - k_l VM]\}$ intercepts the origin. For this case, $m(i,j) = m(1,6) = 2$. The convex hull enclosing the $\det[I - kDM]$ image has a larger area than the true image of the hypercube solid (shaded area). The area contained in $\text{co}\{\det[I - k_l VM]\}$ that is not contained in $\det[I - kDM]$ makes k_l a conservative estimate, that is, $k_l<k_m$. Lemma 5.4 says that if we split the parameter space into two sub-domains along one of the two vertex paths $(V_1 - V_2 - V_6)$ or $(V_1 - V_5 - V_6)$ and compute convex hulls about each of the images of the two sub-domains, then the origin will not be contained in either convex hull. This guarantees that we can converge to the true stability margin k_m by splitting the parameter space into sub-domains. As the sub-domains become smaller, we approach the true image of $\det[I - kDM]$.

Lemma 5.3 states that $k<k_m$ will not destabilize the system. Geometrically, this places the Δ_{V_i} on the boundary of D and guarantees a unique stability margin k_m. By using these Lemmas 5.1, 5.2, and 5.3, the convergence theorem [3] follows.

Theorem 5.6. *Let* $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ *be defined as previously given; then an iterative algorithm can be constructed that converges to* k_m. *If* k_m *is finite, then this procedure identifies the parameters* $\delta_i \in D$ *at which* k_m *is determined. There are three steps involved in determining* k_m*:*

1. *Determine the lower bound on* k_m.
2. *Determine the upper bound on* k_m.
3. *Iterate lower and upper bounds and converge on* k_m.

■

The actual procedure involved in each step is very problem dependent. As may be expected, there are several special cases concerning the critical vertices that vary the algorithm. For example, let $\text{co}\{\det[I - kVM]\}$ intercept the origin between two critical vertices F_i and F_j, one or both of which are coincident. For this case, different logic is required when splitting the domain into sub-domains. DeGaston and Safonov [3] present an excellent exposition of these special cases. They are briefly summarized here.

Special Case 1 The $\text{co}\{\det[I - kVM]\}$ intercepts the origin at a single isolated critical vertex (ICV) $F_i = \det[I - k_l \Delta_{V_i} M]$. Then $m(i,j) = 0$, $k_l = k_u = k_m$, and the algorithm stops. The parameters that cause instability are at the vertex V_i.

Special Case 2 The $\text{co}\{\det[I - kVM]\}$ intercepts the origin between two ICVs F_i and F_j where $m(i,j) = 1$. Both points F_i and F_j are contained in the mapped hypercube image $\det[I - kDM]$. With Corollary 5.2, the line segment connecting these two vertices is also contained in the mapped hypercube image $\det[I - kDM]$.

Thus, $k_l = k_u = k_m$, and the algorithm stops. The Δ along this edge of the hypercube that is destabilizing is given by

$$\Delta_\beta = diag\big[(1-\beta)V_i + \beta V_j\big]$$

where

$$\beta \in (0,1) \text{ such that } \det\big[I - k\Delta_\beta M\big] = 0 \tag{5.126}$$

If either of the above special cases is true, the application of step 1 determines k_m. Only if $m(i,j) \geq 2$ does the algorithm progress further.

Consider ICVs F_i and F_j with $m(i,j) \geq 2$. The upper bound on k_m is determined by examining the $m(i,j)!$ vertex paths between F_i and F_j. The upper bound k_u is determined by the largest k along one of the $m(i,j)!$ vertex paths that intercepts the origin. If k is increased and the origin is not intercepted, then $k_u = \infty$. Once the lower and upper bounds k_l and k_u have been determined, Lemmas 5.1, 5.2, and 5.3 are used to converge to k_m.

Special Case 3 The co$\{\det[I - kVM]\}$ intercepts the origin between two critical vertices F_i and F_j in which one or both are coincident.

Special Case 3a Consider the problem where F_i and F_j both intercept the origin, that is, $\det[I - k_l\Delta_{V_i}M] = \det\big[I - k_l\Delta_{V_j}M\big] = 0$. Then, $k_l = k_m$, and the stability margin is defined at multiple values of Δ_{V_i}.

Special Case 3b There are several coincident vertices located at F_i and several at F_j in which m_c is defined as follows:

$$m_c = \min\{m(i,j)\} = 1; i = \{a,b,\ldots\}\, j = \{s,t,\ldots\} \tag{5.127}$$

Pick an $i \in \{a,b,\ldots\}$ and $j = \{s,t,\ldots\}$. Thus, $m(i,j) = 1$ and k_m is determined as in special case 2.

Special Case 3c This is the same condition as in special case 3b, except that $m_c \geq 2$. For this case, domain splitting is used to divide the domain into sub-domains. This is repeated along each of the vertex paths to each coincident critical vertex.

Let the set $\{z\}$ contain z coincident critical vertices at F_z and the set $\{y\}$ contain y critical vertices at F_y. Take the first two elements of the set $\{z\}$, say, a, b. Then $m(a,b) \geq 1$, since both a, b are vertices of the hypercube. Split the domain along the edge between these two vertices with an orthogonal cut. This creates two sub-domains D_1 and D_2, each containing one of the critical vertices a and b. Continue this process $z-2$ times, creating z sub-domains, each having an isolated critical vertex at F_z. Repeat this same process for the critical vertices in $\{y\}$. This creates zy sub-domains, each having two critical vertices. Apply the procedures of the preceding special cases to each of these sub-domains.

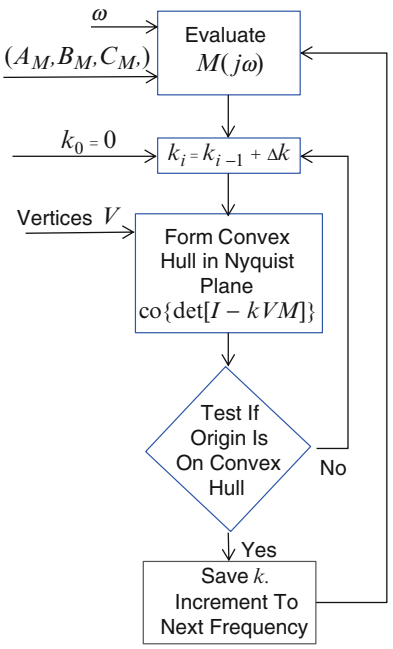


Fig. 5.34 Flow chart for computing the real margin k_m

5.4.3.5 Computing the Real Margin

Figure 5.34 outlines the calculation of the real margin k_m. The algorithm uses the convex hull-based lower bound from Theorem 5.6. The ΔM analysis model is created as defined in Sect. 5.3.1 following the example given in Example 5.4. The state-space model for M is

$$A_M = A_0; B_M = [\beta_1 \quad \cdots \quad \beta_n]; C_M = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{5.128}$$

and the uncertainty matrix Δ is

$$\Delta = diag[\delta_1 \quad \cdots \quad \delta_n] \tag{5.129}$$

A vector of frequencies is selected spanning the range in which the real margin is to be computed. As shown in Fig. 5.34, the matrix M is then evaluated at a given frequency, $M(j\omega) = C_M(sI - A_M)^{-1}B_M$, and the vertices of the hypercube are mapped into the Nyquist plane defined in V. Each vertex V_i (2^n of them) maps to a point F_i. A convex hull is then formed using the points F_i, and a zero-exclusion test is made to see if the origin is contained on the convex hull. If not, the value of k is increased, and the mapping/convex hull/zero-exclusion procedure is repeated. We know that for $k = 0$, the $\det[I - kDM] = 1.0$. As k is increased, the algorithm stops

when either the origin is on the convex hull or some upper limit is reached. This procedure is applied to each frequency in the vector ω, and then a plot of k versus frequency is made. The robustness bound is the minimum k versus frequency. The following example demonstrates this analysis.

Example 5.6 Autopilot Sensitivity to Real-Parameter Variations
Consider the longitudinal dynamics from Example 5.2 with the angle-of-attack controller from Example 5.4. The ΔM analysis model for this problem was formed in Example 5.4. This analysis will determine how sensitive the LQR control law is to accurate values for the aerodynamic stability derivatives used in the design model. We showed in Chap. 2 that LQR control laws have infinite margin at the plant input loop break point. This example will show that these control laws are sensitive to the accuracy of the model parameters.

The state-space triple for M from Example 5.4 is

$$A_M = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & \mathbf{-1.3046} & 1.0 & \mathbf{-0.2142} & 0 \\ 0 & \mathbf{47.711} & 0 & \mathbf{-104.83} & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275100. & 49059. & 3753. & -16861. & -163. \end{bmatrix} \tag{5.130}$$

$$B_M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & 10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; C_M$$

$$= \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix}$$

where the uncertain aerodynamic stability derivatives $(Z_\alpha/V, Z_\delta/V, M_\alpha, M_\delta)$ are in bold. The uncertainty matrix Δ that models the real-parameter uncertainties is a 4×4 diagonal matrix that models the parameter variations using (5.72). For $n = 4$, there are $2^4 = 16$ hypercube vertices, modeled as in (5.119).

Figure 5.35 shows a frequency sweep ($\omega = \text{logspace}(-2, 3, 100)$) where the minimum k was determined at each frequency by the $\text{co}\{\det[I - kVM]\}$ intersecting the origin of the Nyquist plane. Also included in the plot is the bound computed using the small gain theorem. The minimum k versus frequency is $k_m = 0.53831$, and it occurs at a frequency of $\omega = 8.7976$ rad/s. The small gain theorem bound is $\min(1/\bar{\sigma}(M)) = 0.27167$.

To determine if the bound $k_m = 0.53831$ is conservative or exact, we must examine the convex hull and the vertices used in forming the convex hull. Figure 5.36 is a plot of the 16 hypercube vertices at $\omega = 8.7976$ rad/s and $k = 0.53831$. The convex hull vertices and uncertainty matrices Δ_{V_i} are

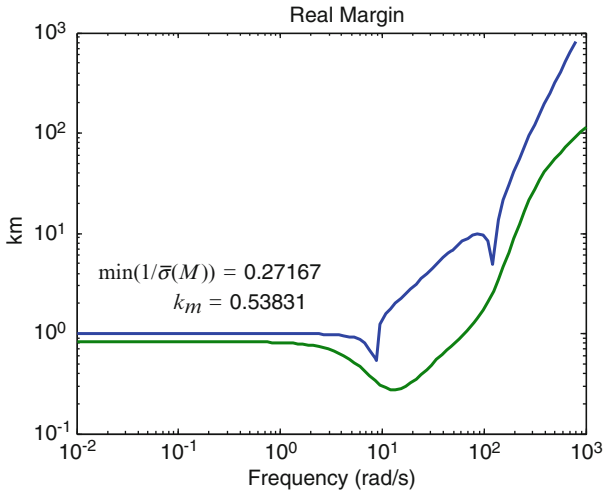


Fig. 5.35 Frequency sweep of the real margin and small gain uncertainty bounds

$$
\begin{aligned}
F_{13} &= 1.5867 - 0.0718\text{i}; & \Delta_{V_{13}} &= diag[-1 \quad -1 \quad 1 \quad 1] \\
F_{15} &= 1.8186 - 0.0700\text{i} & \Delta_{V_{15}} &= diag[-1 \quad -1 \quad -1 \quad 1] \\
F_{7} &= 2.0505 - 0.0314\text{i} & \Delta_{V_{7}} &= diag[1 \quad -1 \quad -1 \quad 1] \\
F_{3} &= 2.0752 - 0.0060\text{i} & \Delta_{V_{3}} &= diag[1 \quad 1 \quad -1 \quad 1] \\
F_{4} &= 0.3260 + 0.0787\text{i} & \Delta_{V_{4}} &= diag[1 \quad 1 \quad -1 \quad -1] \\
F_{2} &= 0.1062 + 0.0759\text{i} & \Delta_{V_{2}} &= diag[1 \quad 1 \quad 1 \quad -1] \\
F_{6} &= 0.0694 + 0.0515\text{i} & \Delta_{V_{6}} &= diag[1 \quad -1 \quad 1 \quad -1] \\
F_{14} &= -0.0000 - 0.0000\text{i} & \Delta_{V_{14}} &= diag[-1 \quad -1 \quad 1 \quad -1]
\end{aligned}
\tag{5.131}
$$

We see from Figs. 5.36 and (5.131) that $\Delta_{V_{14}}$ intercepts the origin. To show that the closed-loop system is destabilized using $\Delta = 0.53831\Delta_{V_{14}}$, we insert these uncertainties into the system matrix and compute the eigenvalues of A_{cl}. Doing so yields the following eigenvalues:

$$
\lambda(A_{\text{cl}}) = \begin{bmatrix} -76.0970 \pm 96.3341j \\ 0.0001 \pm 8.7976j \\ -11.4150 \end{bmatrix} \tag{5.132}
$$

which shows two roots just on the $j\omega$ axis.

This analysis shows that this design can tolerate a 53.8 % variation in the aerodynamic stability derivatives. We see that this "robust stability margin" is much less than the plant input margins guaranteed by using LQR controllers (at the plant input). It is well known that classical gain and phase margins, including

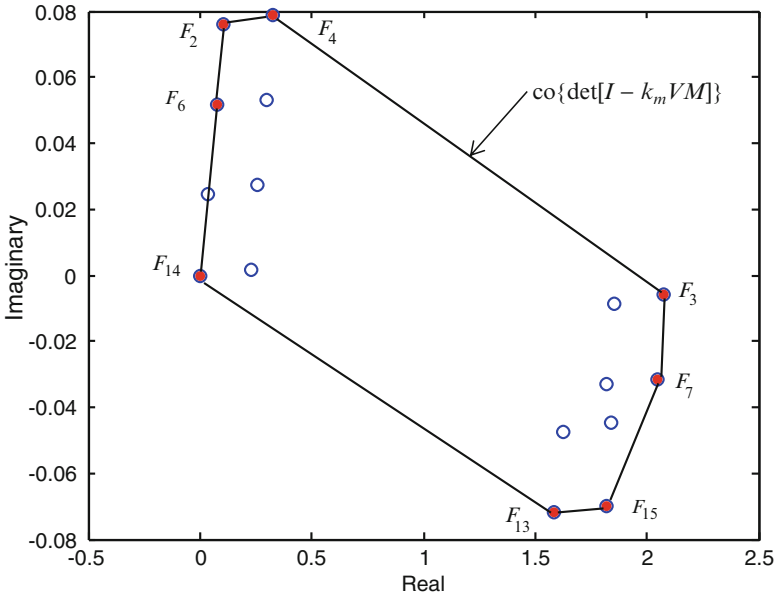


Fig. 5.36 Convex hull for the real margin bound

the vector margin, do not necessarily mean the system is robust to real-parameter uncertainties. This state feedback design has a scalar loop gain at the plant input described by $L(s) = K_c(sI - \tilde{A})\tilde{B}$. In transfer function form,

$$L(s) = \frac{27.4(s_i + 67.8 \pm 90.4j)(s_i + 7.5 \pm 5.1j)}{s(s + 7.6)(s - 6.3)(s_i + 67.8 \pm 90.4j)} = \frac{27.4(s_i + 7.5 \pm 5.1j)}{s(s + 7.6)(s - 6.3)} \quad (5.133)$$

Note that the open-loop system is unstable ($M_\alpha > 0$) and the actuator poles are exactly cancelled in $L(s)$ (at the plant input only). The gain margin at the plant input is $[-11, +\infty]$ dB and the phase margin $\pm 60°$. Next, consider a scalar real uncertainty δ_K at the plant input and compute a root locus, that is, zeros of $s(s + 7.6)(s - 6.3) + \delta_K 27.4(s_i + 7.5 \pm 5.1j)$. This is plotted in Fig. 5.37 and shows the system is stable for all gain values $\delta_K > 0.28$. This is achieved because of the zero dynamics that exist at this loop break point.

When we analyze the system under real-parameter uncertainty, the robustness bounds are determined by the zero dynamics that exist in the $M(s)$ matrix. This is the fundamental challenge in robust control. How do we design a controller that provides "robust" zero dynamics at multiple loop break points simultaneously? We can see this challenge by repeating this root locus analysis for the uncertain parameters. Consider varying just the M_δ stability derivative. The state-space triple for $M(s)$ is

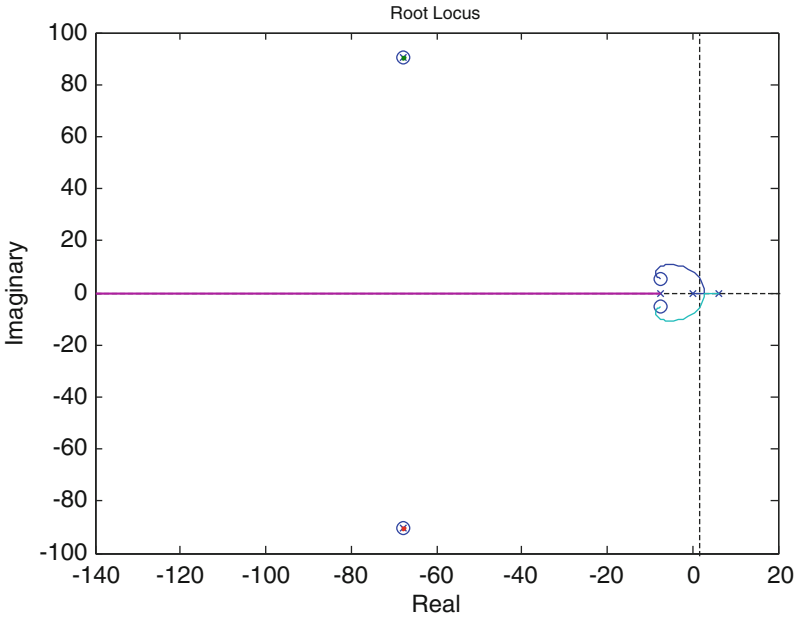


Fig. 5.37 Root locus of LQR $L(s)$ at plant input

$$A_M = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.2142 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275100. & 49059. & 3753. & -16861. & -163. \end{bmatrix} \tag{5.134}$$

$$B_M = \begin{bmatrix} 0 \\ 0 \\ 10.2389 \\ 0 \\ 0 \end{bmatrix}; C_M = \begin{bmatrix} 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix}$$

The transfer function for this system is

$$M(s) = \frac{-393449.(s_i + 7.2 \pm 4.7j)}{(s + 14.3)(s_i + 7.2 \pm 10.3j)(s_i + 67.8 \pm 90.4j)} \tag{5.135}$$

A root locus is shown in Fig. 5.38. The roots cross the $j\omega$ axis with a gain of $\delta_{M_\delta} = 0.68$. We see this is larger than the real margin bound $k_m = 0.53831$ from the example because only a single parameter is being varied.

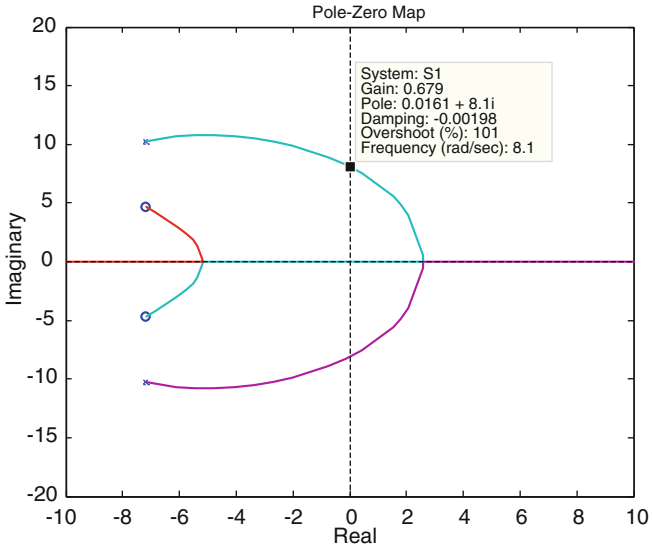


Fig. 5.38 Root locus for $M(s)$ varying only M_δ

The key point to be understood from this example is that for linear systems, the zero dynamics at each loop break point influences the resulting stability robustness. The control architecture as well as the magnitude of the gains in the design influence and change these zero dynamics. Thus, as the bandwidth of the control design changes, so does the sensitivity to accurate knowledge of the model parameters.

This fact has led us to combine robust control with adaptive control to improve the system's sensitivity to uncertainties and nonlinearities. The first half of this book is focused on optimal and robust control and how to design the best linear robust controller possible. "Best" in terms of meeting command tracking requirements and being robust to high-frequency unmodeled dynamics and sensor noise. These methods form the baseline control that is then augmented with an adaptive increment to further address uncertainties and nonlinearities that the robust control. The second half of the book covers the adaptive control linear and nonlinear systems. Together we have found these methods solve some of the most challenging problems in aerospace control.

5.5 Conclusions

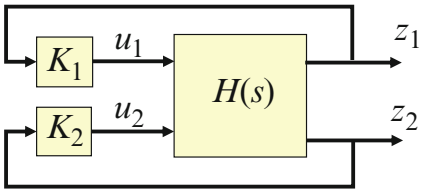
This chapter presented the theory and practice of using frequency domain methods to analyze robust stability. The ability to design controllers and analyze the stability of multivariable systems has been an enabling technology for the aerospace

industry. Prior to the development of this theory in 1980s, flight vehicles were predominantly designed to be stable. The advent of modern control theory, in which the methods in this chapter belong, enables control engineers to basically control any shape of aircraft.

Virtually all of the design methods used today for control system design are model dependent. The accuracy in which we know these models varies, and to measure the data accurately can significantly increase costs. Analyzing the control systems robustness to uncertainties gives the engineer very powerful tools to determine what data is needed to be known accurately and what data is not. Overall this reduces costs, but more importantly increases the quality of the design, making it perform better and be safer.

5.6 Exercises

Exercise 5.1. Consider the following block diagram



where

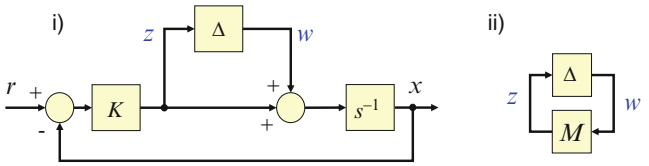
$$H(s) = \begin{bmatrix} \frac{3}{s} & \frac{-278}{s(s+6)(s+30)} \\ \frac{0.05}{s} & \frac{-206}{s(s+6)(s+30)} \end{bmatrix}$$

and $K_1 = 5$ and $K_2 = -10$.

(a) Apply the multivariable Nyquist theorem from Sect. 5.3.3 to this system, examining the return difference matrix $I + \mathrm{KH}$, where $K = diag[K_1 \quad K_2]$, and determine stability. Create the multivariable Nyquist plot. This is a plot of the $\det[I + \mathrm{KH}]$, and indicate the number of encirclements.

(b) Plot the singular values of the return difference matrix and stability robustness matrix versus frequency. Compute the singular value gain and phase margins for this system. This is a plot of $\underline{\sigma}[I + L]$ and $^{-}\sigma[I + L^{-1}]$ versus frequency. Plot these using a log scale for frequency and magnitude in dB.

Exercise 5.2. Consider the block diagrams shown below. Each block in the diagrams is a scalar.



(a) Derive a state-space model (A_M, B_M, C_M) for $M(s) = C_M(sI - A_M)^{-1}B_M$ shown in (ii), modeling the system shown in (i)
(b) For $\Delta = 0$, what is the range of gain K that provides closed-loop stability?
(c) For $K = 1$, sketch the small gain theorem applied to the system in (ii) using your model derived in part (a). This should be a magnitude versus frequency plot.
(d) What does the sketch in (c) indicate for the system's robustness to uncertainties that are constant, that is, $\Delta =$ constant?

Exercise 5.3. Consider the longitudinal airframe dynamics and classical control system described in Example 5.1. The model data is for (5.14) and (5.8)

$K_a = -0.0015$	$V = 886.78$ fps
$K_q = -0.32$	$Z_\alpha/V = -1.3046$
$a_z = 2.0$	Z_δ/V −0.2142
$a_q = 6.0$	$M_\alpha = 47.7109$
	$M_\delta = -104.8346$

(a) Form a closed-loop state-space model and simulate an acceleration step response to show the system is stable and correct.
(b) Form the loop transfer function matrix at the plant input. Compute stability margins.
(c) The actuator dynamics were neglected during the controller design. Derive a multiplicative error model for the neglected actuator dynamics, assuming that the actuator dynamics are modeled using the following transfer function:
(d) $\dfrac{\delta}{\delta_c} = \dfrac{1}{\tau s + 1}$
(e) Form a ΔM robustness analysis model for analyzing these neglected actuator dynamics.
(f) Determine the largest actuator time constant τ that results in a stable closed-loop system using the small gain theorem as the robustness test.

Exercise 5.4. Consider the longitudinal dynamics $\dot{x} = A_p x + B_p u$, $x = [\alpha \quad q]^T$, with

$$[A_p \quad B_p] = \left[\begin{bmatrix} -1.2100 & 1.0 \\ 44.2506 & 0 \end{bmatrix}\begin{bmatrix} -0.1987 \\ -97.2313 \end{bmatrix}\right].$$

Build a robust servomechanism model from Chap. 3:

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & & A_p \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 \\ B_p \end{bmatrix}$$

and close the loop using a state feedback control law $u = -K_c z, z = \begin{bmatrix} \int e_\alpha & \alpha & q \end{bmatrix}^T$, with gains

$$K_c = \begin{bmatrix} -2.1598 & -1.3301 & -0.1700 \end{bmatrix}$$

Analyze the stability of this system. Plot a Nyquist plot, Bode plot, $\underline{\sigma}[I + L]$, and $\underline{\sigma}[I + L^{-1}]$ versus frequency and compute singular value stability margins.

Exercise 5.5. Using the classical longitudinal control system from problem 5.2, analyze the closed-loop system's robustness to uncertainties in the aerodynamic parameters. Create the ΔM robustness analysis model using (5.81) as in Example 5.4, considering the uncertainties

$$\left(\frac{Z_\alpha}{V}(1 \pm \delta_1), \frac{Z_\delta}{V}(1 \pm \delta_2), M_\alpha(1 \pm \delta_3), M_\delta(1 \pm \delta_4)\right)$$

Compute the structured singular value (SSV) bound analyzing the sensitivity to these real-parameter uncertainties.

References

1. Doyle, J.C.: Robustness of multiloop linear feedback systems. In Proceedings of the IEEE Conference on Decision and Control, pp. 12–18. IEEE press, New York (1978)
2. Wise, K.A.: A comparison of six robustness tests evaluating missile autopilot robustness to uncertain aerodynamics. J. Guid. Control Dynamics **15**(4), 861–870 (1992)
3. DeGaston, R.R., Safonov, M.: Exact calculation of the multiloop stability margin. IEEE Trans. Autom. Control **33**(2), 156–171 (1988)
4. Morton, B.G., McAfoos, R.M.: A Mu-test for robustness analysis of a real-parameter variation problem. In Proceedings of the American Control Conference, American Automatic Control Council, Boston, MA. (1985)
5. Morton, B.G.: New applications of Mu to real parameter variation problems. In Proceedings of the 25th IEEE Conference on Decision and Control. IEEE press, New York (1985)
6. Doyle, J.C., Wall, J.E., Stein, G.: Performance and robustness analysis for structured uncertainty. In Proceedings of the 22nd IEEE Conference on Decision and Control, Orlando, Florida, pp. 629–636. (1982)
7. Kuo, B.C.: Automatic Control Systems, 6th edn. Prentice Hall, New York (1971)
8. Zadeh, L.A., Desoer, C.A.: Linear System Theory. McGraw-Hill, New York (1963)

Chapter 6
Output Feedback Control

6.1 Output Feedback Using Projective Controls

Projective control is an output feedback design method used to obtain a partial eigenstructure of a state feedback controller $(u = -K_x x, x \in R^{n_x}, u \in R^{n_u})$ using static and/or dynamic output feedback. The design retains the dominant performance and robustness properties of the state feedback design. For static output feedback $(u = -K_y y, y \in R^{n_y})$, n_y eigenvalues and associated eigenvectors of a state feedback design can be retained. For dynamic output feedback, a low-order compensator can be built that can retain additional eigenvalues and associated eigenvectors from the state feedback eigenstructure. The order of the compensator can be increased until the entire state feedback eigenstructure is obtained.

Linear quadratic regulator (LQR) designs generally give good performance characteristics and stability margins, with the availability of the states required for implementation. In many practical designs, not all the states are available for feedback. For these problems, dynamic compensators (observers, Kalman filters, state predictors) are required for implementation.

There have been numerous studies on constructing dynamic regulators. General procedures in the time domain are based upon observer theory and pole placement. An observer is a dynamic system whose output variables are the estimates of the states of another system. Pole placement is used to make the observer dynamics faster than the dynamics being estimated. Typically, using these theories does not provide the designer with a low-order dynamic compensator which yields satisfactory results.

An alternative approach is to use projective controls [1–3]. The projective control methodology combines optimal control (state feedback design) with eigenstructure assignment. The uniqueness of this method is that the designer has the option of choosing the order of the dynamic compensator (with some restrictions), rather than having a full-order compensator. The design goal using projective controls is to retain the dominant dynamics as if the states were available for feedback.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_6,

In this section, the projective controls method is used to design a bank-to-turn flight control for an unmanned aircraft implemented using static output feedback and, as required, a low-order dynamic compensator. The section presents the basic features of a design procedure using projective controls.

Projective control is a method of retaining the most dominant eigenstructure of a full state feedback design using only output feedback, where the optimal state feedback regulator serves as the reference solution (desired eigenstructure). Static projective controls are output feedback controls that retain an invariant subspace of the reference dynamics in the resulting closed-loop system.

Consider the problem of designing output feedback regulators for a linear-time-invariant system described by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{6.1}$$

with $C = \begin{bmatrix} 1_{n_y} & 0 \end{bmatrix}$, $x \in R^{n_x}, u \in R^{n_u}$, and $y \in R^{n_y}$, and the triple (A, B, C) controllable and observable. Let the resulting LQR state feedback control be characterized by

$$u = -R^{-1}B^T Px = -K_x x \tag{6.2}$$

where the positive definite matrix P satisfies

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{6.3}$$

where $Q \geq 0, R > 0$, and the pair $\left(A, Q^{\frac{1}{2}}\right)$ observable. Suppose using the state feedback control law, (6.2), the above system yields satisfactory closed-loop reference dynamics, described by

$$\dot{x} = \underbrace{(A - BK_x)}_{F} x = Fx \tag{6.4}$$

If there are n_y outputs available for feedback as described in (6.1) using the control law

$$u = -K_y y \tag{6.5}$$

Then, n_y eigenvalues $\left(\Lambda_{n_y}\right)$ and their associated eigenvectors $\left(X_{n_y}\right)$ of the state feedback design can be retained using the static output feedback gains given by

$$K_y = K_x X_{n_y} \left(CX_{n_y}\right)^{-1} \tag{6.6}$$

where the eigenvectors X_{n_y} and eigenvalues Λ_{n_y} satisfy the eigen equation for the state feedback system as

$$FX_{n_y} = X_{n_y}\Lambda_{n_y} \tag{6.7}$$

Using the output feedback control law from (6.5), the closed-loop system is

$$\dot{x} = \underbrace{\left(A - BK_yC\right)}_{A_{cl}} x = A_{cl}x \tag{6.8}$$

with eigenstructure

$$\begin{aligned} A_{cl}X_{n_y} &= \left(A - BK_yC\right)X_{n_y} \\ &= \left(A - BK_xX_{n_y}\left(CX_{n_y}\right)^{-1}C\right)X_{n_y} \\ &= (A - BK_x)X_{n_y} \\ &= X_{n_y}\Lambda_{n_y} \end{aligned} \tag{6.9}$$

which captures the n_y eigenvalues $\left(\Lambda_{n_y}\right)$ and their associated eigenvectors $\left(X_{n_y}\right)$ of the state feedback design. The remaining $\left(n_x - n_y\right)$ eigenvalues in (6.8) may not result in closed-loop stability or satisfactory performance. In this case, additional eigenstructure from A_{cl} can be retained using dynamic compensation.

Consider the closed-loop system given by (6.4). Introduce partitions in the matrices A and F as follows:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}; F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{6.10}$$

where A_{11} and $F_{11} \in R^{n_y \times n_y}$. Denote X and Λ as the eigenvector matrix and spectrum of F $(FX = X\Lambda)$, respectively, from (6.4). From this eigenstructure, select the desired dominant dynamics by selecting and placing their eigenvalues in Λ_{n_y} (n_y-eigenvalues) and associated eigenvectors in X_{n_y} $\left(FX_{n_y} = X_{n_y}\Lambda_{n_y}\right)$. The eigenstructure of the closed-loop output feedback system, using (6.5), has a spectrum

$$\Lambda(A_{cl}) = \Lambda_{n_y} \cup \Lambda(A_r)$$

where the residual dynamics satisfy

$$A_r = VAY \tag{6.11}$$

with $V \in C^{\left(n_x - n_y\right) \times n_x}$ and $Y \in C^{n_x \times \left(n_x - n_y\right)}$ satisfying the conditions $CY = 0, VX_{n_y} = 0$, and $VY = I$. If the static feedback projective control, (6.5), does not produce an acceptable result, a p-th order dynamic observer of the form

$$\begin{aligned} \dot{z} &= H_d z + D_d y \\ u &= -N_d z - K_d y \end{aligned} \tag{6.12}$$

can be used to extend the eigenstructure. Introduce an extended system by combining the observer (6.12) with the reference dynamics (6.10) described by

$$\dot{x}_e = F_e x_e; \quad F_e = \begin{bmatrix} H_d & D_d & 0 \\ 0 & F_{11} & F_{12} \\ 0 & F_{21} & F_{22} \end{bmatrix} \tag{6.13}$$

Using the observer states as additional outputs for feedback, an $(n_y + p)$ - dimensional invariant subspace can be retained. Denote (X_p, Λ_p) as the additional p-dimensional invariant subspace from the reference dynamics. The eigenstructure from the extended system can be decomposed as

$$\begin{bmatrix} H_d & D_d & 0 \\ 0 & F_{11} & F_{12} \\ 0 & F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} W_p & W_{n_y} \\ X_{p_1} & X_{n_{y1}} \\ X_{p_2} & X_{n_{y2}} \end{bmatrix} = \begin{bmatrix} W_p & W_{n_y} \\ X_{p_1} & X_{n_{y1}} \\ X_{p_2} & X_{n_{y2}} \end{bmatrix} \begin{bmatrix} \Lambda_p & 0 \\ 0 & \Lambda_{n_y} \end{bmatrix} \tag{6.14}$$

where W_p and W_{n_y} depend upon the observer matrices H_d and D_d. Define the following matrices:

$$\begin{aligned} N_0 &= X_{n_{y1}} \quad X_{n_{y2}}^{-1} \\ B_0 &= X_{p_2} - N_0 X_{p_1} \\ A_r &= A_{22} - N_0 A_{12} \end{aligned} \tag{6.15}$$

The dynamic compensator matrices (H_d, D_d, N_d, K_d) can all be parameterized by a free gain matrix P_0. Select P_0 to stabilize the residual dynamics, given by

$$A_{re} = A_r + B_0 P_0 A_{12} \tag{6.16}$$

which is an output feedback stabilization problem. The dynamic compensator gain matrices are then given by

$$\begin{aligned} H_d &= \Lambda_p + P_0(A_{12} - B_1 K_{x_2}) B_0 \\ D_d &= P_0(A_{11} - B_1 K_{x_1} + (A_{12} - B_1 K_{x_2}) N_0) - H_d P_0 \\ N_d &= K_{x_2} B_0 \\ K_d &= K_{x_1} + K_{x_2}(N_0 - B_0 P_0) \end{aligned} \tag{6.17}$$

where

$$K_x = [K_{x_1} \quad K_{x_2}]; B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

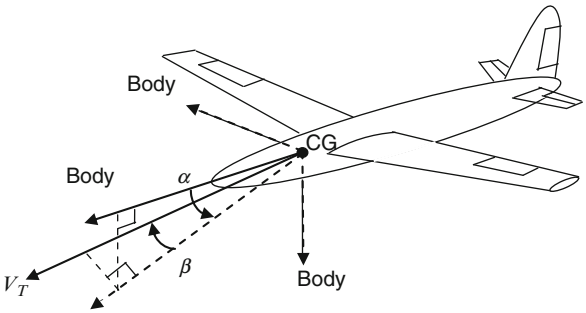


Fig. 6.1 Unmanned aircraft

Example 6.1 Static Output Feedback Design Using Projective Control. This example applies projective control theory to an air vehicle flight control design problem. The state feedback design (reference eigenstructure) to be retained with the projective control is the Robust Servo Linear Quadratic Regulator (RSLQR) design from Example 3.4 from Chap. 3. There are three basic steps to applying the projective control approach.

Step 1: Design the reference eigenstructure (using a state feedback approach).
Step 2: Design a static projective controller using output feedback. Evaluate the adequacy of the design. If not adequate, proceed to step 3.
Step 3: Design a low-order dynamic compensator recovering more of the entire eigenstructure of the state feedback design, iterate by adding to the eigenstructure, as required.

Step 3 is necessary only if the compensator designed in step 2 is not satisfactory. Time domain and frequency domain analyses are performed after each design step.

Consider the design of the longitudinal flight control system for an unmanned aircraft as shown in Fig 6.1. The pitch plane dynamics are given as

$$\begin{aligned} \dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta + M_q q \end{aligned} \tag{6.18}$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant and will design an RSLQR controller with integral control. We will design a constant gain matrix K_c for a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration $A_z(ft/s^2)$ is given by

$$A_z = -V\dot{\gamma} = Z_\alpha \alpha + Z_\delta \delta \tag{6.19}$$

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate (6.19) to form the differential equation for $\dot{A}_z$ and then substitute for $\dot{\alpha}$ from (6.18). This produces

$$\begin{aligned}\dot{A}_z &= \frac{Z_\alpha}{V}A_z + Z_\alpha q + Z_\delta \dot{\delta}_e \\ \dot{q} &= \frac{M_\alpha}{Z_\alpha}A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_e\end{aligned} \tag{6.20}$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_e = -2\zeta_a \omega_a \dot{\delta}_e + {\omega_a}^2(\delta_c - \delta_e) \tag{6.21}$$

Combining (6.20) and (6.21) forms the plant model written in matrix form as

$$\begin{bmatrix}\dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e\end{bmatrix} = \begin{bmatrix} Z_\alpha/V & Z_\alpha & 0 & Z_\delta \\ M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix}\begin{bmatrix}A_z \\ q \\ \delta_e \\ \dot{\delta}_e\end{bmatrix} + \begin{bmatrix}0 \\ 0 \\ 0 \\ {\omega_a}^2\end{bmatrix}\delta_c \tag{6.22}$$

Since $r = \text{constant}, \dot{r} = 0, p = 1$, and we need to add one integrator to form a type-1 controller. The state vector (3.17) for the robust servomechanism design model is

$$z = \begin{bmatrix} e & \dot{x}^T \end{bmatrix}^T \tag{6.23}$$

with the design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ given as

$$\begin{bmatrix}\dot{e} \\ \ddot{A}_z \\ \dddot{q} \\ \ddot{\delta}_e \\ \dddot{\delta}_e\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha/V & Z_\alpha & 0 & Z_\delta \\ 0 & M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix}\begin{bmatrix}e \\ \dot{A}_z \\ \ddot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e\end{bmatrix} + \begin{bmatrix}0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2\end{bmatrix}\dot{\delta}_c \tag{6.24}$$

At a flight condition of Mach 0.3, 5,000 ft altitude, and a trim angle-of-attack α of 5 °, the plant model is

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -6672. & -98.02 \end{bmatrix} \tilde{B} = \begin{bmatrix}0 \\ 0 \\ 0 \\ 0 \\ 6672.\end{bmatrix} \tag{6.25}$$

The objective in the design of the gain matrix is to track the acceleration command with zero error without using large gains. The design begins by setting $R = 1$ and selecting a Q matrix that penalizes the error state e in (6.24). Thus, the performance index in the equation is

$$J = \int_0^\infty \left(z^T Q z + \mu^2\right) d\tau \tag{6.26}$$

We start by inserting the parameter q_{11} in the (1,1) element,

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ & 0 & & \underset{\sim}{0} & \\ & & 0 & & \\ & \underset{\sim}{0} & & 0 & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}, \tag{6.27}$$

and setting the other matrix elements to zero. This will penalize the error in tracking the command. Substituting (6.27) into (6.26) gives the performance index as

$$J = \int_0^\infty \left(q_{11} e^2 + \mu^2\right) d\tau \tag{6.28}$$

LQR design charts were used in Chap. 3 to select the LQR penalty parameters in (6.28). We will use the same numerical value in this example to design the state feedback control and then will compare the projective control to the state feedback design.

Step 1: Design reference eigenstructure
From Chap. 3, $q_{11} = 0.2448$. This produces the state feedback gain matrix

$$K_x = \begin{bmatrix} 0.49482 & 0.17904 & -14.061 & 2.2089 & 1.8036 \times 10^{-3} \end{bmatrix} \tag{6.29}$$

For this design, the states A_z, q, δ_e, and $\dot{\delta}_e$ are plotted versus time in Fig. 6.2. Note that there is no overshoot to the unit command. For this approach flight condition, the response is quick without the use of large gains. It is desired to keep this same behavior, to the degree possible, in the output feedback projective control design.

Step 2: Design a static projective controller
For the static projective control output feedback design, the $\int e, A_z$, and q states are the desired feedbacks. The integral error state is part of the controller, so it is available. The acceleration and rate feedbacks come directly from the inertial measurement unit and are available. The actuator states in the model would require additional sensors within the actuator and are assumed not to be available for feedback. In the static projective controller design, these states will be projected out. The closed-loop matrix $F = \left(\tilde{A} - \tilde{B} K_x\right)$ from (6.4) is

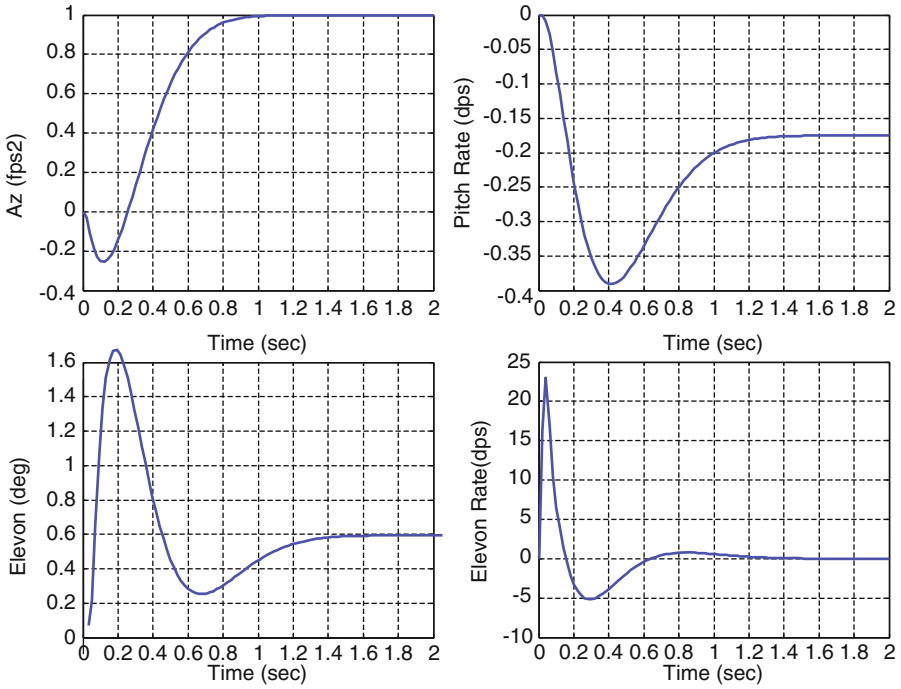


Fig. 6.2 States of the system responding to a unit acceleration step command

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -3301.3 & -1194.5 & 93810. & -21409. & -110.05 \end{bmatrix} \tag{6.30}$$

with eigenstructure

$$\Lambda = diag[\,-4.2466, \quad -4.9722 \pm 4.3962j, \quad -48.973 \pm 65.438j\,] \tag{6.31}$$

$$X = \begin{bmatrix} \begin{bmatrix} 2.2295\text{e}{-001} \\ -9.4676\text{e}{-001} \\ -1.6075\text{e}{-002} \\ -5.3104\text{e}{-002} \\ 2.2551\text{e}{-001} \end{bmatrix} & \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & -1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & -3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & -2.3622\text{e}{-002}j \end{bmatrix} & \begin{bmatrix} -1.0379\text{e}{-001} & +9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & +1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & +3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & +2.3622\text{e}{-002}j \end{bmatrix} \\ & \begin{bmatrix} 4.8477\text{e}{-004} & -1.6181\text{e}{-003}j \\ 8.2144\text{e}{-002} & +1.1097\text{e}{-001}j \\ 5.1005\text{e}{-005} & -1.6701\text{e}{-004}j \\ -7.2600\text{e}{-003} & -9.7008\text{e}{-003}j \\ 9.9035\text{e}{-001} & \end{bmatrix} & \begin{bmatrix} 4.8477\text{e}{-004} & +1.6181\text{e}{-003}j \\ 8.2144\text{e}{-002} & -1.1097\text{e}{-001}j \\ 5.1005\text{e}{-005} & +1.6701\text{e}{-004}j \\ -7.2600\text{e}{-003} & +9.7008\text{e}{-003}j \\ 9.9035\text{e}{-001} & \end{bmatrix} \end{bmatrix} \tag{6.32}$$

The dominant eigenvalues (short period plus integrator) are $\lambda_i = \{-4.2466, -4.9722 \pm 4.3962j,\}$ and are associated with the first three

eigenvectors in (6.32). These are the dynamics that are to be retained in the static output feedback design. From (6.6), the output matrix is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{6.33}$$

and the eigenvectors are

$$\begin{aligned} X_n = & \begin{bmatrix} \begin{bmatrix} 2.2295\text{e}{-001} \\ -9.4676\text{e}{-001} \\ -1.6075\text{e}{-002} \\ -5.3104\text{e}{-002} \\ 2.2551\text{e}{-001} \end{bmatrix} & \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & -1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & -3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & -2.3622\text{e}{-002}j \end{bmatrix} \end{bmatrix} \\ & \times \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & +1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & +3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & +2.3622\text{e}{-002}j \end{bmatrix} \end{aligned} \tag{6.34}$$

The static output feedback gain matrix is computed as

$$\begin{aligned} K_y &= K_x X_{n_y} \left(C X_{n_y}\right)^{-1} \\ &= [\,0.13327 \quad 0.050120 \quad -4.2100\,] \end{aligned} \tag{6.35}$$

Figure 6.3 shows a block diagram of the output feedback control. The closed-loop system using the static output feedback is $A_{cl} = A - BK_yC$. The eigenstructure for this system is

$$\Lambda = diag[\,-4.2466, \quad -4.9722 \pm 4.3962j, \quad -36.175 - 49.738\,] \tag{6.36}$$

$$X = \begin{bmatrix} \begin{bmatrix} 2.2295\text{e}-001 \\ -9.4676\text{e}-001 \\ -1.6075\text{e}-002 \\ -5.3104\text{e}-002 \\ 2.2551\text{e}-001 \end{bmatrix} & \begin{bmatrix} -1.0379\text{e}-001 - 9.1764\text{e}-002j \\ 9.1946\text{e}-001 \\ -1.4240\text{e}-003 - 1.0897\text{e}-002j \\ -4.3324\text{e}-002 - 3.3555\text{e}-002j \\ 3.6293\text{e}-001 - 2.3622\text{e}-002j \end{bmatrix} & \begin{bmatrix} -1.0379\text{e}-001 + 9.1764\text{e}-002j \\ 9.1946\text{e}-001 \\ -1.4240\text{e}-003 + 1.0897\text{e}-002j \\ -4.3324\text{e}-002 + 3.3555\text{e}-002j \\ 3.6293\text{e}-001 + 2.3622\text{e}-002j \end{bmatrix} \\ & \begin{bmatrix} 4.4768\text{e}-003 \\ -2.2266\text{e}-001 \\ 4.7032\text{e}-004 \\ 1.9597\text{e}-002 \\ -9.7469\text{e}-001 \end{bmatrix} & \begin{bmatrix} -8.2386\text{e}-003 \\ 2.9803\text{e}-001 \\ -8.7700\text{e}-004 \\ -2.6376\text{e}-002 \\ 9.5416\text{e}-001 \end{bmatrix} \end{bmatrix} \tag{6.37}$$

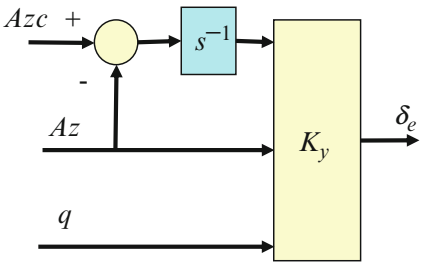


Fig. 6.3 Static output feedback acceleration command control

Note that the eigenvalues $\Lambda_i = diag[-4.2466, \quad -4.9722 \pm 4.3962j]$ and their associated eigenvectors from the state feedback design are retained in the closed-loop output feedback design system. The remaining two eigenvalues which forms the residual dynamics are $\Lambda_i = diag[-36.175, \quad -49.738]$.

The static output feedback controller shown in Fig. 6.3 implementing the RSLQR design can be implemented in the following state space format:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{6.38}$$

with

$$\begin{aligned} \begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} &= \begin{bmatrix} [0] & [1 \quad 0] & [-1] \\ [-K_y(1)] & [-K_y(2:3)] & [0] \end{bmatrix} \\ &= \begin{bmatrix} [0] & [1 \quad 0] & [-1] \\ [-0.13327] & [-0.050120 \quad 4.21] & [0] \end{bmatrix} \end{aligned} \tag{6.39}$$

where $x_c = \int e$, $y = [A_z \quad q]^T$, $r = A_{zc}$, and $u = \delta_c$.

For this static output feedback design method to be effective, care must be taken to keep the bandwidth reasonable and not destabilize the residual dynamics. As the bandwidth of the state feedback design is increased to have the system respond faster (an increase in the loop gain crossover frequency, Fig. 3.5), the larger gains destabilizes the residual dynamics. The next step in the design process is to compare this design with the state feedback design to determine changes in performance and stability robustness, if any, and to determine if they are acceptable. Figure 6.4 compares the state feedback and output feedback time histories. The static projective control response slightly lags the state feedback design and is acceptable from a time domain perspective. By capturing the dominant eigenvalues of the state feedback design in the output feedback design, the time response is very close. Next is to compare the design in the frequency domain. Figure 6.5 shows plots of the magnitude and phase of L, the magnitude of $I + L$, and the magnitude of $I + L^{-1}$, with the loop gain formed at the actuator command input. We see from the plots of $I + L$ and $I + L^{-1}$ that the output feedback design has a decrease in the stability robustness. This is also seen in the Nyquist plot, Figure 6.6, which shows the gain

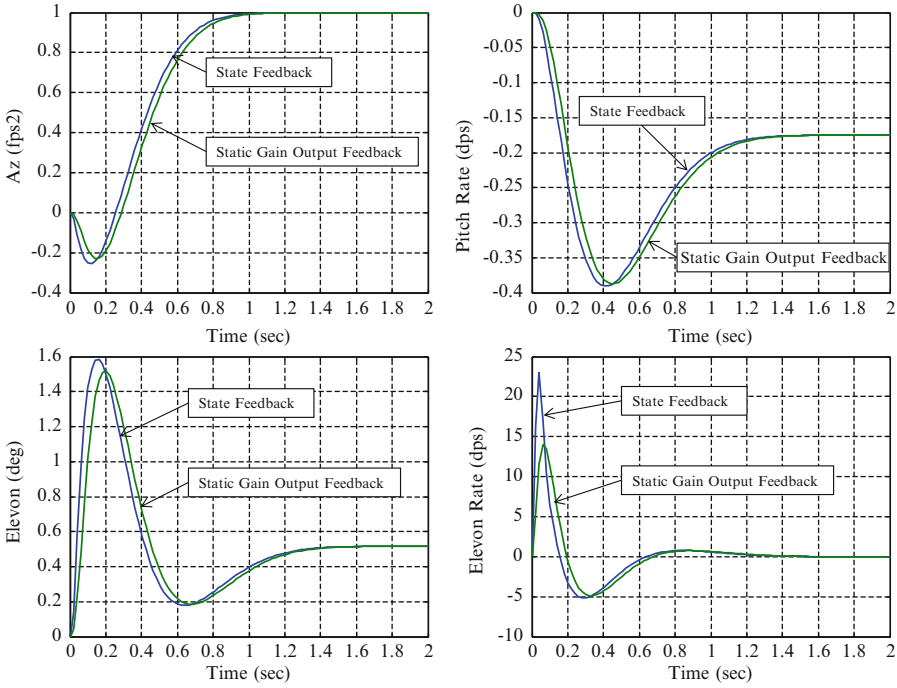


Fig. 6.4 Comparison of state feedback and output feedback time histories

margin and phase margin of the design. The output feedback design has a gain margin of 3.82 dB and phase margin of 28.3°. This would be unacceptable for flight, so the design would need to be improved by either (1) decreasing the bandwidth until acceptable stability margins were obtained or (2) designing a low-order dynamic compensator to recover the performance and margins of the state feedback design. We will select the second option and proceed to design a low-order dynamic compensator.

Step 3: Design a low-order dynamic compensator

For this example, a second-order compensator will retain the entire state feedback eigenstructure. To begin the design, we need to partition the matrices A and F as in (6.10) and the eigenvectors as in (6.14).

$$A_{11} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.053 & -346.5 \\ 0 & 0.007 & -1.033 \end{bmatrix} \quad A_{12} = \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix}$$
$$A_{21} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_{22} = \begin{bmatrix} 0 & 1 \\ -6672. & -98.02 \end{bmatrix} \tag{6.40}$$

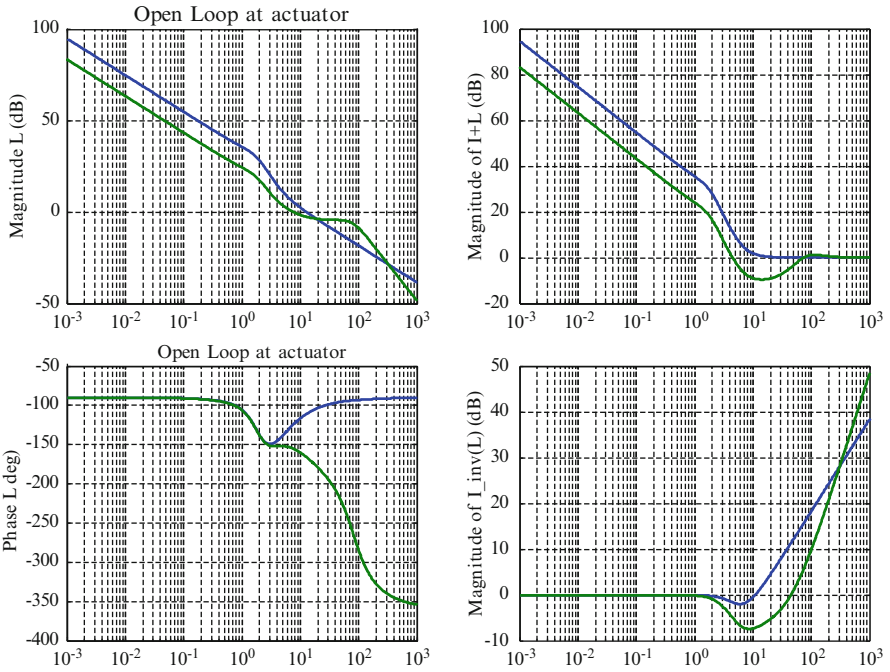


Fig. 6.5 Comparison of state feedback and output feedback frequency domain loop shapes

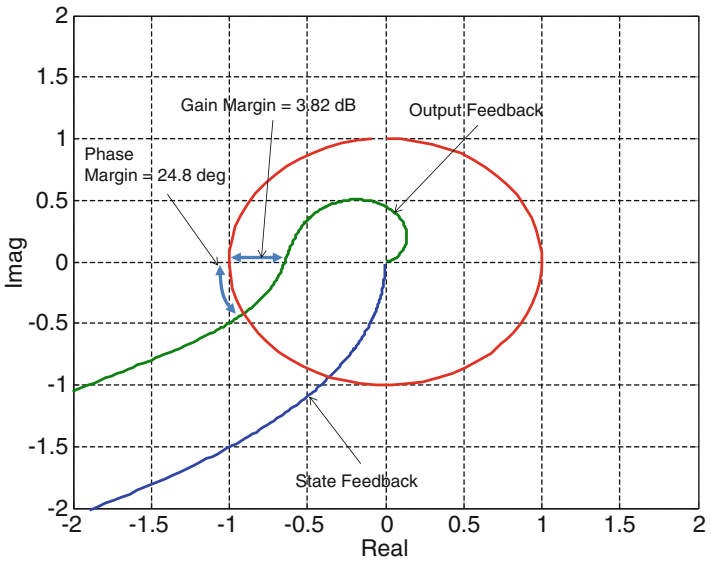


Fig. 6.6 Comparison of state feedback and output feedback

$$F_{11} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.053 & -346.5 \\ 0 & 0.007 & -1.033 \end{bmatrix} \quad F_{12} = \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix}$$
$$F_{21} = \begin{bmatrix} 0 & 0 & 0 \\ -3301.3 & -1194.5 & 93810. \end{bmatrix} \quad F_{22} = \begin{bmatrix} 0 & 1 \\ -21409. & -110.05 \end{bmatrix} \tag{6.41}$$

$$X_{n_{y1}} = \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}\text{i} & -1.0379\text{e}{-001} & +9.1764\text{e}{-002}\text{i} & 2.2295\text{e}{-001} \\ 9.1946\text{e}{-001} & & 9.1946\text{e}{-001} & & -9.4676\text{e}{-001} \\ -1.4240\text{e}{-003} & -1.0897\text{e}{-002}\text{i} & -1.4240\text{e}{-003} & +1.0897\text{e}{-002}\text{i} & -1.6075\text{e}{-002} \end{bmatrix}$$
$$X_{n_{y2}} = \begin{bmatrix} -4.3324\text{e}{-002} & -3.3555\text{e}{-002}\text{i} & -4.3324\text{e}{-002} & +3.3555\text{e}{-002}\text{i} & -5.3104\text{e}{-002} \\ 3.6293\text{e}{-001} & -2.3622\text{e}{-002}\text{i} & 3.6293\text{e}{-001} & +2.3622\text{e}{-002}\text{i} & 2.2551\text{e}{-001} \end{bmatrix}$$
$$X_{p_1} = \begin{bmatrix} 4.8477\text{e}{-004} & -1.6181\text{e}{-003} \\ 8.2144\text{e}{-002} & 1.1097\text{e}{-001} \\ 5.1005\text{e}{-005} & -1.6701\text{e}{-004} \end{bmatrix}$$
$$X_{p_2} = \begin{bmatrix} -7.2600\text{e}{-003} & -9.7008\text{e}{-003} \\ 9.9035\text{e}{-001} & 0 \end{bmatrix} \tag{6.42}$$

The compensator design (6.12) requires selecting a gain matrix P_0 such that the residual dynamics A_r in (6.16) are stable. The matrices needed to form A_r are N_0 and B_0.

$$\begin{aligned} N_0 &= X_{n_{y1}} X_{n_{y2}}^{-1} \\ &= \begin{bmatrix} -1.6565\text{e}{-001} & -5.8888\text{e}{-002} & 4.4744\text{e}{+000} \\ 2.4157\text{e}{+000} & 6.3925\text{e}{-001} & -1.8175\text{e}+001 \end{bmatrix} \end{aligned} \tag{6.43}$$

$$\begin{aligned} B_0 &= X_{p_2} - N_0 X_{p_1} \\ &= \begin{bmatrix} -2.5706\text{e}-003 & -2.6871\text{e}-003 \\ 9.3759\text{e}-001 & -7.0061\text{e}-002 \end{bmatrix} \end{aligned} \tag{6.44}$$

$$\begin{aligned} A_r &= A_{22} - N_0 A_{12} \\ &= \begin{bmatrix} 4.8883\text{e}+000 & 3.3521\text{e}-001 \\ -6.6917\text{e}+003 & -9.0801\text{e}+001 \end{bmatrix} \end{aligned} \tag{6.45}$$

Using the dynamic compensator in (6.12) with matrices defined in (6.17), the compensator is designed by choosing the free parameter matrix P_0 such that the residual dynamics in (6.16) are stable. For this example (6.46),

$$\begin{aligned} A_{re} &= A_r + B_0 P_0 A_{12} \\ &= \begin{bmatrix} 4.8883\text{e}+000 & 3.3521\text{e}-001 \\ -6.6917+003 & -9.0801\text{e}+001 \end{bmatrix} \\ &\quad + \begin{bmatrix} -2.5706\text{e}-003 & -2.6871\text{e}-003 \\ 9.3759\text{e}-001 & -7.0061\text{e}-002 \end{bmatrix} P_0 \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix} \end{aligned} \tag{6.46}$$

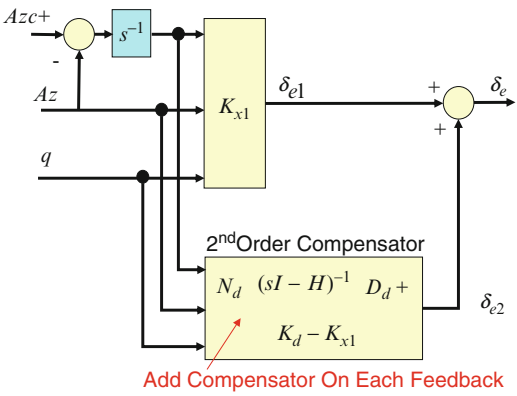


Fig. 6.7 Low-order dynamic projective controller block diagram

By multiplying out the matrices in (6.47), one can determine which elements of P_0 need to be chosen. This matrix is designed using a tuning process in which the elements are increased in magnitude until a suitable design is obtained (trial and error). After some tuning, the following matrix was obtained:

$$P_0 = \begin{bmatrix} 0 & 2 & -500 \\ 0 & 2 & -2000000 \end{bmatrix} \tag{6.47}$$

The zero elements in the first column were found not to matter. They were made zero to reduce the control usage. Substituting this P_0 into (6.47) yields

$$A_{re} = \begin{bmatrix} -5.8364\text{e}+002 & 4.5392\text{e}-001 \\ -2.1488\text{e}+004 & -1.1039\text{e}+002 \end{bmatrix}; \quad \lambda_i(A_{re}) = [\,-132.0 \quad -562.0\,] \tag{6.48}$$

Figure 6.7 illustrates a block diagram for the controller. Substituting P_0 into (6.17) yields

$$\begin{aligned} H_d &= \begin{bmatrix} -7.1546\text{e}+001 & 6.5552\text{e}+001 \\ -6.4828\text{e}+002 & -6.3452\text{e}+002 \end{bmatrix}; \\ D_d &= \begin{bmatrix} -1.4503\text{e}+002 & -4.0079\text{e}+001 & 1.3077\text{e}+007 \\ -3.6250\text{e}+004 & -1.1663\text{e}+004 & -1.2604\text{e}+008 \end{bmatrix} \\ N_d &= [\,-3.9870\text{e}-003 \quad -6.0617\text{e}-003\,]; \\ K_d &= [\,1.3327\text{e}-001 \quad 7.0218\text{e}-002 \quad -1.2185\text{e}+003\,] \end{aligned} \tag{6.49}$$

The dynamic output feedback controller shown in Fig. 6.7 capturing the eigenstructure of the RSLQR design can be implemented in the following state space format:

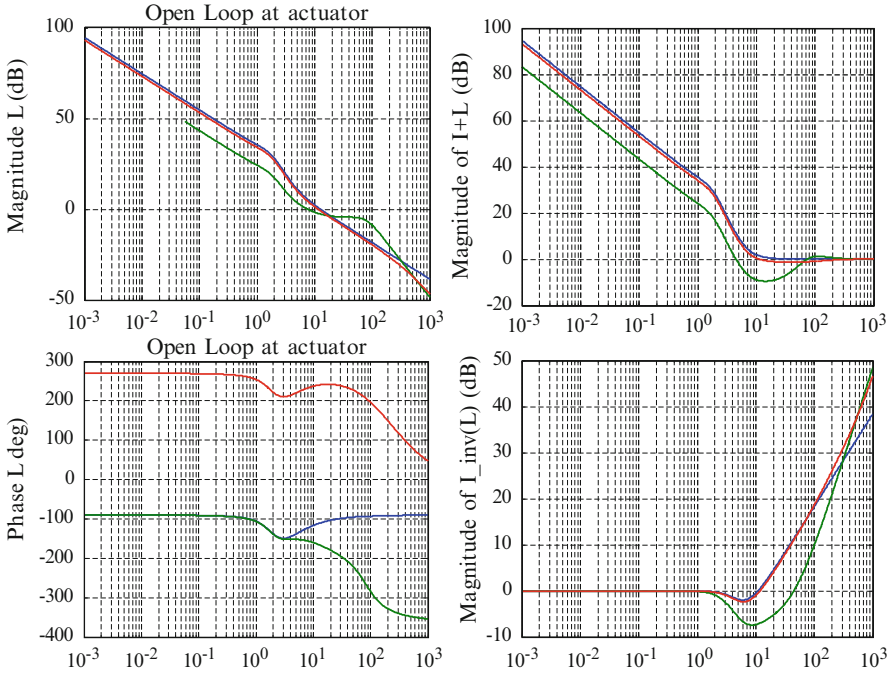


Fig. 6.8 Frequency response comparison between the state feedback, static, and dynamic output feedback projective controllers

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{6.50}$$

With

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ D_d(:,1)) & H_d \end{bmatrix} & \begin{bmatrix} C_c \\ D_d(:,2:3) \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} -K_d(1) & -N_d \end{bmatrix} & \begin{bmatrix} -K_d(2:3) \end{bmatrix} & [0] \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -145.03 & -71.546 & 65.552 \\ -3625.0 & -648.28 & -634.52 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ -40.079 & 1.3077\text{e}+007 \\ -1.1663\text{e}+004 & -1.2604\text{e}+008 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} -0.13327 & 0.0039870 & 0.0060617 \end{bmatrix} & \begin{bmatrix} -0.070218 & 1218.5 \end{bmatrix} & [0] \end{bmatrix} \tag{6.51}$$

where $x_c = \begin{bmatrix} \int e & x_{c2} & x_{c3} \end{bmatrix}^T$, $y = \begin{bmatrix} A_z & q \end{bmatrix}^T$, $r = A_{zc}$, and $u = \delta_c$.

A step-input simulation of the closed-loop system using the dynamic controller shows results that equal the state feedback design. Figure 6.8 compares the designs in the frequency domain showing plots of the magnitude and phase of L, the magnitude of $I + L$, and the magnitude of $I + L^{-1}$, with the loop gain formed at

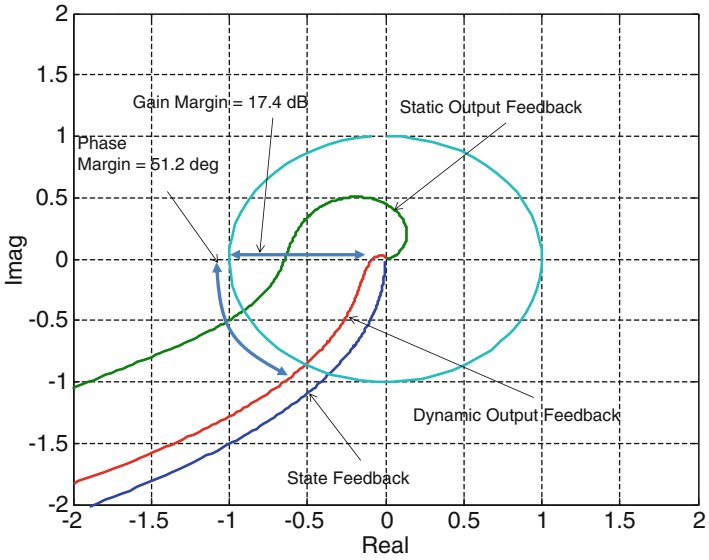


Fig. 6.9 Nyquist plot comparison between the state feedback, static, and dynamic output feedback projective controllers

the actuator command input. Figure 6.9 shows the Nyquist plot (L is a scalar), which shows the gain margin and phase margin of the design. The second-order compensator does an excellent job of recovering the state feedback design eigenstructure and properties using a low-order compensator with output feedback.

6.2 Linear Quadratic Gaussian with Loop Transfer Recovery

In Chaps. 2 and 3, optimal control was applied to the servomechanism problem to design a state feedback controller for command tracking. When only the output is available for feedback, a full-order observer can be designed to estimate the state. For LTI systems with Gaussian models for disturbances and measurement noise, the Kalman filter is the optimal state estimator. When optimal control (LQR) is combined with optimal state estimation (Kalman filter), the control design is called the linear quadratic Gaussian (LQG) problem.

The Kalman filter algorithm is an excellent state estimator. It is widely used in estimation problems, such as GPS navigation, where accurate state estimates are desired. However, when used to estimate the state in output feedback control design problems, the optimal state estimator (optimal in the sense of minimizing the error covariance) may not exhibit the best overall control properties. It is well known that the LQG controller captures the excellent time domain characteristics of the state feedback design, but the Kalman filter degrades the frequency domain properties

(stability margins) of the design. A tuning process called Loop Transfer Recovery (LTR) asymptotically recovers the state feedback frequency domain properties.

There are several methods available for applying LTR to the LQG problem (called LQG/LTR). Virtually all these methods introduce tuning mechanisms for recovering the frequency domain properties at the expense of using high gains somewhere in the control loop. Care must be taken to limit the gains and a thorough analysis performed to make sure the system is implementable.

When using LQG/LTR, the Kalman filter is no longer thought of as an optimal state estimator, but a dynamic compensator/observer tuned for performance and robustness. The fact that Riccati equations are used in the control and filter design makes the LQG/LTR method attractive and provides the mechanism for proving properties of the system analytically. In this section, we will use a popular tuning mechanism attributed to Doyle and Stein [4] and Doyle and Athans [5] and will demonstrate its use in a command tracking flight control example. We will consider the infinite-time design problem which uses steady-state gain matrices for the LQR and the Kalman filter.

Consider the following linear-time-invariant Gaussian design model:

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \end{aligned} \tag{6.52}$$

where w and v are zero mean, white, uncorrelated Gaussian random processes with covariances given by

$$\begin{aligned} E\{w(t)w^T(\tau)\} &= Q_0\delta(t-\tau) \\ E\{v(t)v^T(\tau)\} &= R_0\delta(t-\tau) \end{aligned} \tag{6.53}$$

The state estimate, $\hat{x}$, is formed using the following Kalman filter state estimator:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + K_f(y - \hat{y}) \\ K_f &= P_f C^T {R_0}^{-1} \\ 0 &= AP_f + P_f A^T + Q_0 - P_f C^T {R_0}^{-1} C P_f \end{aligned} \tag{6.54}$$

where $\hat{y}$ is the estimate of the output, $P_f = E\{xx^T\}$ is the steady-state error covariance, which results from solving the algebraic filter Riccati equation (covariance equation), and Q_0 and R_0 are the process and measurement noise covariances from (6.54), respectively. The optimal control is formed using the LQR state feedback control gain matrix K_c and the estimated state feedback $\hat{x}$, given as

$$u = -K_c\hat{x} \tag{6.55}$$

Figure 6.10 combines the LQR controller (Chap. 3) with the Kalman filter state estimator (6.55) into a block diagram. This is the LQG control architecture.

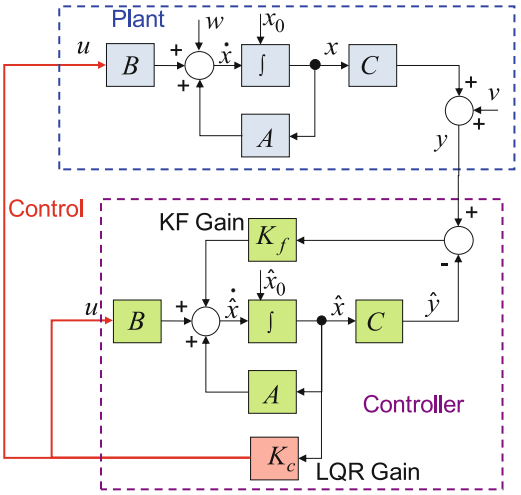


Fig. 6.10 Robust servo LQG using integral control-estimated state feedback

The frequency domain properties of the LQG system do not equal that of the LQR system primarily due to the observer dynamics introduced by the Kalman filter state estimator. For the state feedback controlled system, the LQR loop transfer function matrix (LTFM) at the plant input is

$$L_{LQR}(s) = K_c(sI - A)^{-1}B \tag{6.56}$$

For the output feedback controlled system, the LQG LTFM at the plant input is

$$L_{LQG}(s) = K_c\left(sI - A + BK_c + K_fC\right)^{-1}K_fC(sI - A)^{-1}B \tag{6.57}$$

Clearly, the dynamics introduced by the dynamic compensator alters the frequency domain characteristics for the LQG system.

For this control architecture, there are two approaches for applying LTR to the LQG control problem. One modifies the Kalman filter (state observer) to recover the state feedback loop properties, and the other modifies the LQR controller. Here, we will present the method of modifying the Kalman filter. This approach is taken from Doyle and Stein [4]. The tuning procedure consists of designing Kalman filters with the plant process disturbance covariance matrix Q_f parameterized with a scalarρ as

$$Q_f = Q_0 + \frac{1}{\rho}BB^T \tag{6.58}$$

where Q_0 is the nominal plant process disturbance covariance from (6.54), B is the control input distribution matrix, and $\rho$is the LTR filter compensation parameter. This parameter is adjusted to recover the LQR frequency domain characteristics

over the frequency range of interest. The modified matrix Q_f is used to compute the steady-state covariance P_f and filter gain matrix K_f to be used in the LQG controller.

Considering the loop broken at the plant input, LTR modifies K_f to create a system that has stability properties that asymptotically approach those of the LQR. The method uses a trial and error procedure in which the filter design is parameterized by a scalar $\rho > 0$ such that when $\rho \to 0$ we have $L_{\text{LQG}} \to L_{\text{LQR}}$ asymptotically but not necessarily uniformly. It is evident that the location of the Kalman filter eigenvalues, (6.58), alters the closed-loop frequency characteristics of the system.

The LQG/LTR approach requires that the controlled system (plant) be minimum phase (i.e., no RHP transmission zeros). The minimum phase requirement occurs because the LTR procedure asymptotically inverts the plant dynamics of the Kalman filter and substitutes the linear regulator dynamics. If there was an RHP transmission zero, an RHP pole would be created, causing an unstable system. The procedure may still be applied to nonminimum phase systems, but care must be taken to prevent instability in the LQG compensator. This limits the amount of recovery.

The LQG/LTR loop transfer function matrix at the plant input, L_{LQG}, will asymptotically recover the LQR frequency domain characteristics as $\rho \to 0$. This can be shown as follows. As $\rho \to 0$, the process covariance Q_f in (6.59) becomes largely dominated by the second term $\frac{1}{\rho}BB^T$. As these elements of Q_f get large, the covariance matrix P_f has elements that get large, resulting in the Kalman gain matrix K_f getting large with the following result:

$$\begin{aligned} L_{\text{LQG}}(s) &= K_c(sI - \tilde{A} + \tilde{B}K_c + K \longrightarrow_f C)^{-1} K \longleftarrow_f C(sI - \tilde{A})^{-1}\tilde{B} \\ L_{\text{LQG}}(s) &\approx K_c(sI - \tilde{A})^{-1}\tilde{B} \end{aligned} \tag{6.59}$$

It is this process that inverts the plant (within the Kalman filter) resulting in recovering the LQR L_{LQR}. It is important to note that as $\rho \to 0$, $\bar{\sigma}(P_f) \to \infty$ and $^-\sigma(P_f) \to 0$, creating a singular covariance matrix. In the next section, we will present the LTR method of Lavretsky [6] which prevents this condition from occurring during the recovery process.

The LQG controller transfer function matrix that relates the measurement y to the control u is

$$u = -K_c\left(sI - \tilde{A} + \tilde{B}K_c + K_f C\right)^{-1} K_f y \tag{6.60}$$

Substituting for the measurement $y = Cx + v$ and letting $\rho \to 0$ as in (6.60) yields

$$\begin{aligned} u &= -K_c(sI - \tilde{A} + \tilde{B}K_c + K_f C)^{-1} K_f (Cx + v) \\ &= -K_c(sI - \tilde{A} + \tilde{B}K_c + K^f C)^{-1} K_f Cx - K_c(sI - \tilde{A} + \tilde{B}K_c + K_f C)^{-1} K_f v \\ &= -K_c x - K_c(sI - \tilde{A} + \tilde{B}K_c + K_c F)^{-1} K_f v \end{aligned} \tag{6.61}$$

in which the first term is inverted and canceled $(K_f C)^{-1} K_f C = I$ resulting in $-K_c x$. However, the second term is not exactly canceled; $(K_f C)^{-1} K_f \neq I$, and the sensor noise v can be amplified. This feature limits the amount of recovery possible. In the use of this design method for making the LQG system robust, the sensor noise amplification in (6.62) must be examined.

The LQG/LTR controller design, examining the loop properties at the plant input, may be realized through the following synthesis technique:

Step 1: LQR controller design: K_c

Follow the robust servomechanism design approach outlined in Chap. 3. Design LQR weighting matrices Q and R such that the resulting LTFM $L_{\mathrm{LQR}}(s) = K_c \left(sI - \tilde{A}\right)^{-1} \tilde{B}$ meets performance and stability robustness requirements and exhibits the desired bandwidth. The frequency domain properties of the LQG system will not exceed those of the LQR system.

Step 2: Kalman filter design: K_f Design the Kalman filter state estimator using (6.55), with (6.59) defining the plant disturbance covariance. The LTR filter recovery parameter ρ is used to recover the LQR frequency domain characteristics over the frequency range of interest. Examine plant input and output frequency domain criteria and the sensor noise amplification in (6.62) and limit the LTR recovery so that the sensor noise is not amplified.

6.2.1 Summary

Dynamics: $\dot{x} = Ax + Bu + w \qquad x(t_0) = x_0$

$y = Cx + v$

$E\{x\}$

$E\{ww^T\} = Q_0 \delta(\tau);\ E\{vv^T\} = R_0 \delta(\tau)$

Robust Servomechanism LQR:

Command r

Controlled output (to follow r) : $y_c = C_c x$

$e = y_c - r; z = [e \quad \dot{x}]^T, \mu = \dot{u}$

Dynamics: $\dot{z} = \tilde{A}z + \tilde{B}\mu \quad \tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}$

Performance index: $J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) d\tau$

$\left(\tilde{A}, \tilde{B}\right)$ Stabilizable, $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ Detectable,

Algebraic Riccati Equation: $P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P$

Optimal Control: $\mu = -R^{-1}B^T Px = -K_c z$

$$u = \int \mu = -K_c \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix}$$

Kalman Filter State Estimator: $\dot{\hat{x}} = A\hat{x} + Bu + K_f(y_{\text{meas}} - \hat{y})$

$$Q_f = Q_0 + \frac{1}{\rho}\tilde{B}\tilde{B}^T$$

$$AP_f + P_f A^T + Q_f - P_f C^T R_0^{-1} C P_f = 0$$

$$K_f = P_f C^T R_0^{-1}$$

Example 6.2 LQG/LTR Design. This example applies LQG/LTR control theory to an air vehicle flight control design problem. The LQG/LTR design method combines an LQR state feedback control implemented using estimated states and a Kalman filter state estimator. The state feedback design is the Robust Servo Linear Quadratic Regulator (RSLQR) design from Example 3.4 in Chap. 3 and will be reused as the state feedback control. This is also the same design used in the previous example on projective control theory.

The RSLQR design model (6.24) is

$$\tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -6672. & -98.02 \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6672. \end{bmatrix} \tag{6.62}$$

Using the same RSLQR design, the state feedback gain matrix is

$$K_c = [0.49482 \quad 0.17904 \quad -14.061 \quad 2.2089 \quad 1.8036\text{e}-003] \tag{6.63}$$

The control law is implemented using

$$u = -K_c \begin{bmatrix} \int (A_{z_m} - r) & \hat{x} \end{bmatrix}^T \tag{6.64}$$

where the first gain in K_c multiplies the integral error, and the remaining gains multiply estimates of A_z, q, δ_e, and $\dot{\delta}_e$, respectively.

The measurements provided by an inertial measurement unit, A_{z_m} and q_m, are available for feedback. To design the Kalman filter state estimator, we need models of the process and measurement noise covariance matrices from (6.54). At this flight condition, the process noise modeled in the state equations is

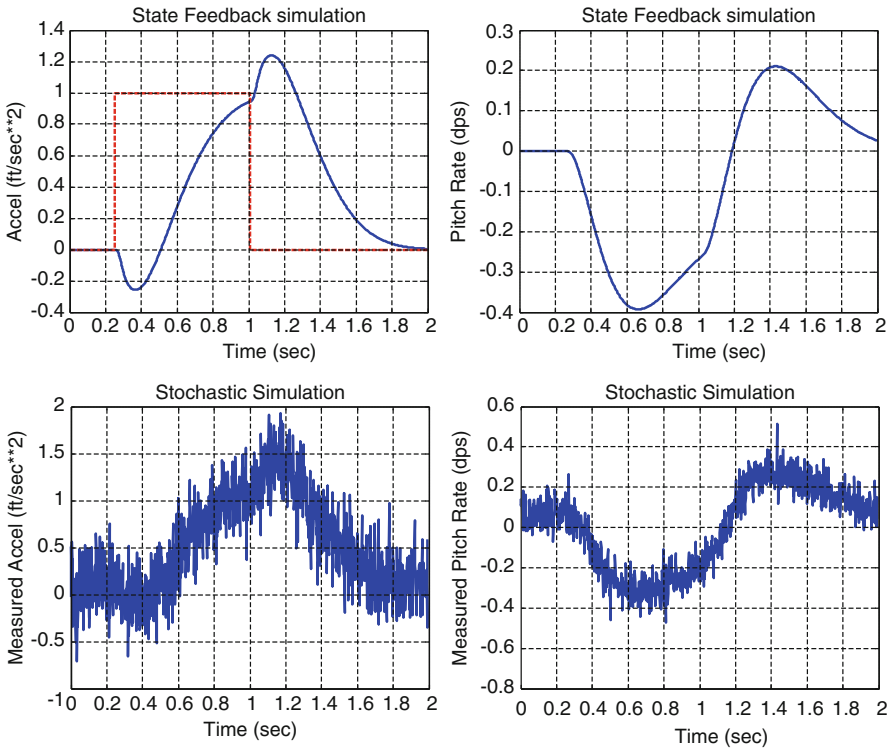


Fig. 6.11 State feedback and measured acceleration and pitch rate time histories

$$Q_0 = \begin{bmatrix} 1.94 \times 10^{-4} & 0 & 0 & 0 \\ 0 & 2.5 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 1.0 \times 10^{-8} & 0 \\ 0 & 0 & 0 & 1.0 \times 10^{-6} \end{bmatrix} \begin{bmatrix} (\text{fps})^2/\text{s} \\ (\text{rps})^2/\text{s} \\ (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \end{bmatrix} \tag{6.65}$$

The numerical values in Q_0 are often adjusted in the design process to tune the Kalman filter. For a typical inertial measurement unit, the measurement noise in A_{z_m} and q_m are modeled as

$$R_0 = \begin{bmatrix} 6.25 \times 10^{-2} & 0 \\ 0 & 1.0 \times 10^{-6} \end{bmatrix} \begin{bmatrix} (\text{fps})^2 \\ (\text{rps})^2 \end{bmatrix} \tag{6.66}$$

Figure 6.11 shows A_z and q simulation time histories of the state feedback controlled system without process and measurement noise, along with simulation time histories of the measured values that contain process and measurement noise. The Kalman filter state estimator is

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \tag{6.67}$$

where u is formed using (6.65) and was implemented using steady-state matrices obtained from the filter covariance equation

$$
\begin{aligned}
0 &= AP_f + P_f A^T + Q_0 - P_f C^T {R_0}^{-1} C P_f \\
K_f &= P_f C^T {R_0}^{-1} \\
P_f &= \begin{bmatrix} 2.0442\text{e}-003 & -6.0760\text{e}-006 & 2.4929\text{e}-010 & -5.6592\text{e}-008 \\ -6.0760-006 & 7.8188\text{e}-008 & -1.2264\text{e}-012 & 1.3375\text{e}-010 \\ 2.4929\text{e}-010 & -1.2264\text{e}-012 & 1.2523\text{e}-010 & -5.0000\text{e}-009 \\ -5.6592\text{e}-008 & 1.3375e-010 & -5.0000\text{e}-009 & 3.4544\text{e}-007 \end{bmatrix} \\
K_f &= \begin{bmatrix} 3.2707-002 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix}
\end{aligned}
\tag{6.68}
$$

The controller implementing the robust servomechanism integral control with the Kalman filter state estimator can be implemented in the following state space format:

$$
\begin{aligned}
\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\
u &= C_c x_c + D_{c1} y + D_{c2} r
\end{aligned}
\tag{6.69}
$$

with

$$
\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0_{1\times 4} \\ -B_p K_c(1) & A_p - B_p K_c - K_f C_p \\ & [-K_c] \end{bmatrix} & \begin{bmatrix} C_c \\ K_f \end{bmatrix} & \begin{bmatrix} -1 \\ 0_{4\times 1} \end{bmatrix} \\ [0 \quad 0] & & [0] \end{bmatrix}
\tag{6.70}
$$

where

$$
\begin{aligned}
\begin{bmatrix} A_c \\ C_c \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0854 & -340.41 & 0 & -11.289 \\ 0 & 6.8202\text{e}-003 & -1.1116\text{e}+000 & -1.0925 & 0 \\ 0 & -3.9887\text{e}-009 & 1.2264\text{e}-006 & 0 & 1.0 \\ -3301.0 & -1194.4 & 9.3804\text{e}+004 & -2.1408\text{e}+004 & -1.1005\text{e}+002 \end{bmatrix} \\ [-0.49477 \quad -0.17903 \quad 14.060 \quad -2.2087 \quad -0.0018035] \end{bmatrix} \\
\begin{bmatrix} B_{c1} & B_{c2} \\ D_{c1} & D_{c2} \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} 1.0000\text{e}+000 & 0 \\ 3.2707\text{e}+000 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ [0 \quad 0] & [0] \end{bmatrix}
\end{aligned}
\tag{6.71}
$$

and $x_c = \begin{bmatrix} \int e & \hat{A}_z & \hat{q} & \hat{\delta}_e & \dot{\hat{\delta}}_e \end{bmatrix}^T$, $y = [A_z \quad q]^T$, $r = A_{zc}$, and $u = \delta_c$.

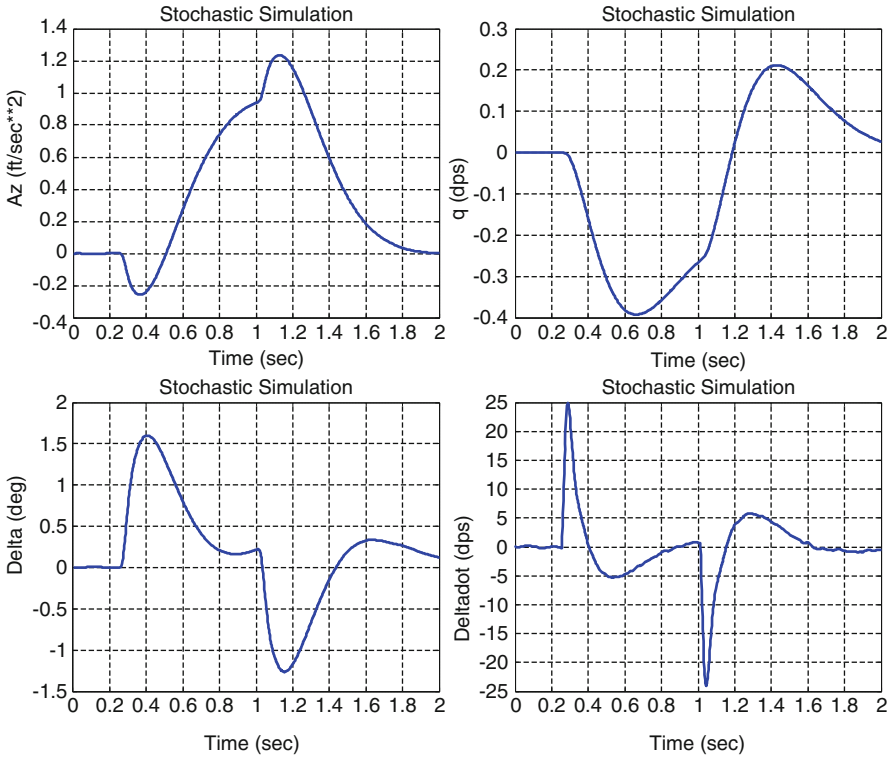


Fig. 6.12 State estimates using the nominal Kalman Filter process noise Q_0

Figure 6.12 shows state estimates using the estimator with the nominal process noise matrix Q_0 (LQG design). The Kalman filter does an excellent job estimating the states from the noisy measurements. However, the full-order observer (Kalman filter) has degraded the excellent frequency domain properties of the LQR state feedback design. To recover the frequency domain properties (at the plant input), Loop Transfer Recovery (LTR) is used. The LTR procedure consists of designing Kalman filters with the plant process covariance matrix Q_f parameterized with a scalar ρ as

$$Q_f = Q_0 + \frac{1}{\rho} BB^T \tag{6.72}$$

where Q_0 is the nominal covariance, B is the control input distribution matrix, and ρ is the LTR filter compensation parameter. This parameter is adjusted, $\rho \to 0$, to recover the LQR frequency domain characteristics over the frequency range of interest. The modified matrix Q_f is used to compute the steady-state covariance matrices P_f and filter gain matrices K_f to be used in the LQG controller. In this example, values of ρ were chosen to be

$$\rho = \begin{bmatrix} \infty & 10^5 & 10^4 & 10^3 & 10^2 \end{bmatrix} \tag{6.73}$$

The following controller combines the robust servo controller and Kalman filter estimator

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ -B_p K_c(1:n_r) & A_p - K_f C_p - B_p K_c(n_r+1:n_x) \end{bmatrix} \begin{bmatrix} x_1 \\ \hat{x} \end{bmatrix} \\ &\quad + \begin{bmatrix} 1 & 0 \\ & K_f \end{bmatrix} z_{meas} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} r \\ u &= -K_c \begin{bmatrix} x_c \\ \hat{x} \end{bmatrix} \end{aligned} \tag{6.74}$$

Note that (6.75) is valid for plant models with no D matrix, that is, $D_p = 0$. The first state x_1 is the robust servo integrator, the vector $\hat{x}$ is the estimated state, z_{meas} contains the acceleration and pitch rate measurements, and r is the acceleration command. Writing the controller in a generic form, we have

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c_1} z_{\text{meas}} + B_{c_2} r \\ u &= C_c x_c + D_{c_1} z_{\text{meas}} + D_{c_2} r \end{aligned} \tag{6.75}$$

For the LQG design ($\rho = \infty$)

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0854\text{e}+000 & -3.4041\text{e}+002 & 0 & -1.1289\text{e}+001 \\ 0 & 6.8202\text{e}-003 & -1.1116\text{e}+000 & -1.0925\text{e}+000 & 0 \\ 0 & -3.9887\text{e}-009 & 1.2264\text{e}-006 & 0 & 1.0 \\ -3.3010\text{e}+003 & -1.1944\text{e}+003 & 9.3804\text{e}+004 & -2.1408\text{e}+004 & -1.1005\text{e}+002 \end{bmatrix}$$

$$B_{c_1} = \begin{bmatrix} 1.0 & 0 \\ 3.2707e-002 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix}; B_{c_2} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$D_{c_1} = [0 \quad 0]; \; D_{c_2} = [0] \tag{6.76}$$

Note that in the above controller, the robust servo error, $e = y_c - r = A_{z_{\text{meas}}} - A_{z_{\text{cmd}}}$, is formed using the measured acceleration. This error is formed from the top row in B_{c_1} and B_{c_2}. An alternate controller would be to use the estimate of A_z from the Kalman filter, $e = \hat{A}_z - A_{z_{\text{cmd}}}$. This would change the control architecture significantly.

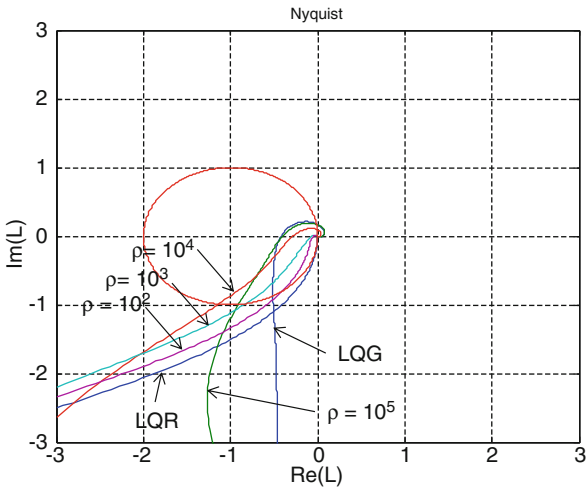


Fig. 6.13 Nyquist plot comparing LQR and LQG designs

Next, we will analyze the LQG/LTR design in the frequency domain and determine the desired amount of LTR to be applied at this flight condition. Figure 6.13 shows a Nyquist plot of the LQR, LQG, and LQG/LTR designs using values of ρ from (6.74). The red circle is a unit circle centered at $(-1,j0)$ for reference. The LQR locus (blue) demonstrates infinite gain margin (at the plant input) and excellent phase margin. The LQG design (blue) shows the decrease in gain margin and phase margin from inserting the Kalman filter state estimator into the controller. The locus for the LTR designs show initially, $\rho = 10^5$, that the margins are worse than those of the LQG. As the LTR parameter is reduced further, the margins improve and approach those of the LQR design. This demonstrates that the LTR recovery process is not uniform in its recovery. Figures 6.14 and 6.15 show the analysis results examining the return difference dynamics $I + L$ and stability robustness matrix $I + L^{-1}$ at the plant input, respectively. Both figures show the recovery of the LQR characteristics at the plant input.

To further examine the effects of LTR, we will examine the sensitivity and complementary sensitivity at the plant output and the noise transmission through the controller. The sensitivity and complementary sensitivity are given by

$$\begin{aligned} e &= S(s)r \\ y &= T(s)r \end{aligned} \tag{6.77}$$

The noise transmission through the controller, $u = G_{\text{Noise}}v$, is given by

$$\bar{\sigma}(G_{\text{Noise}}) = \bar{\sigma}\left(C_c(sI - A_c)^{-1}B_{c_1} + D_{c_1}\right) \tag{6.78}$$

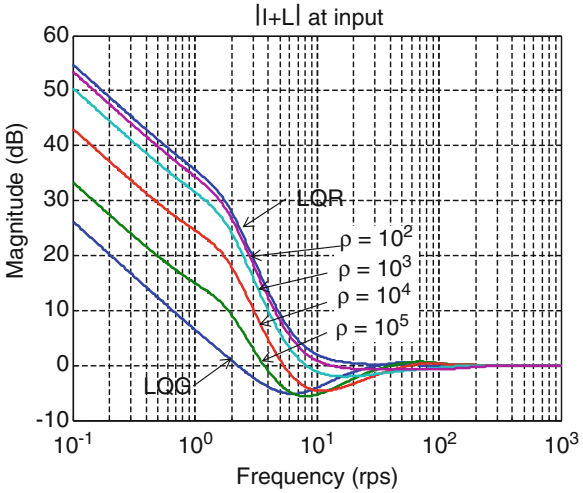


Fig. 6.14 Return difference dynamics $|I + L|$ at the plant input for LQR and LQG/LTR designs

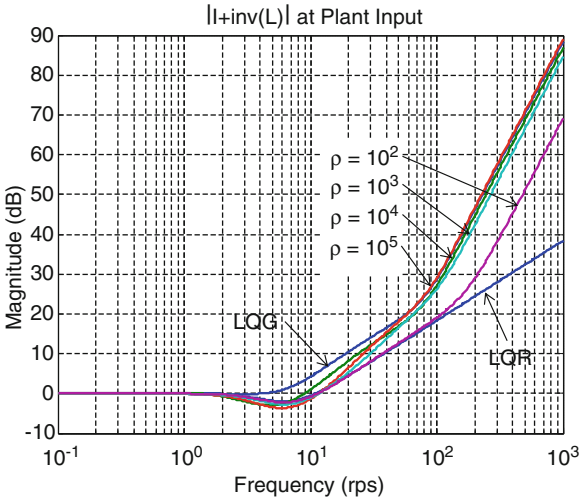


Fig. 6.15 Stability robustness $\left|I + L^{-1}\right|$ at the plant input for LQR and LQG/LTR designs

Figures 6.16, 6.17, and 6.18 show the analysis results at the plant output. The LTR process only guarantees recovery of the LQR properties at the plant input. The sensitivity function in Fig. 6.16 shows undesirable peaking in $S(s)$ as the recovery is made. From this figure, the value of ρ would need to be limited to 10^3. The complementary sensitivity function in Fig. 6.17 shows undesirable peaking in $T(s)$ as the recovery is made. This peak is similar to a peak resonance in under-damped

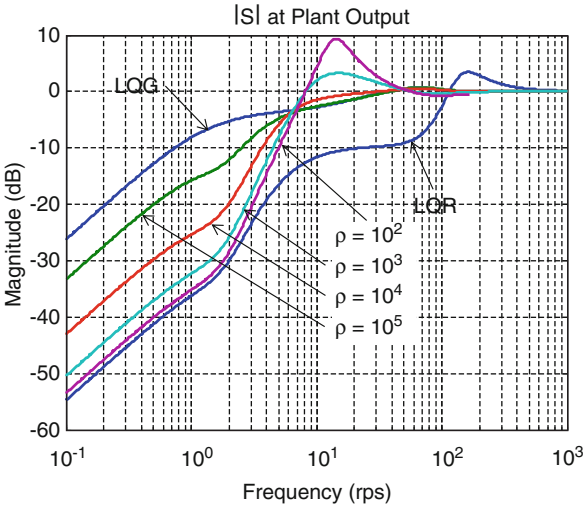


Fig. 6.16 Sensitivity $|S|$ at the plant output for LQR and LQG/LTR designs

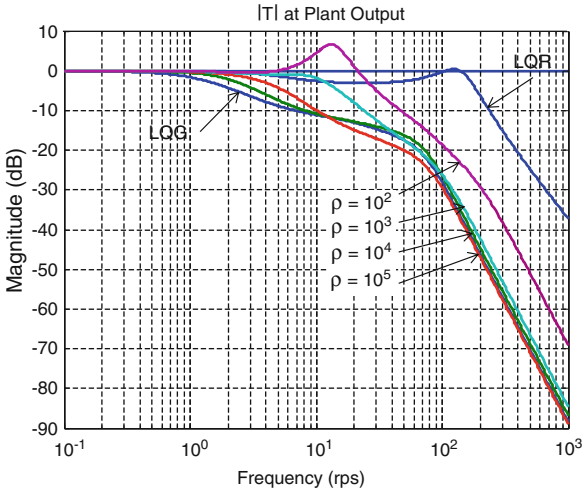


Fig. 6.17 Complementary sensitivity $|T|$ at the plant output for LQR and LQG/LTR designs

second-order systems. Even though the stability margins at the plant input are getting better with LTR, the margins at the plant output are getting worse. Figure 6.17 also shows the value of ρ would need to be limited to 10^3 to keep the peak small. Finally, Fig. 6.18 shows the noise transmission through the controller. We see that as $\rho \to 0$, the noise amplification increases. This would be quite undesirable. This figure indicates that the value of ρ would need to be limited to 10^4 or larger.

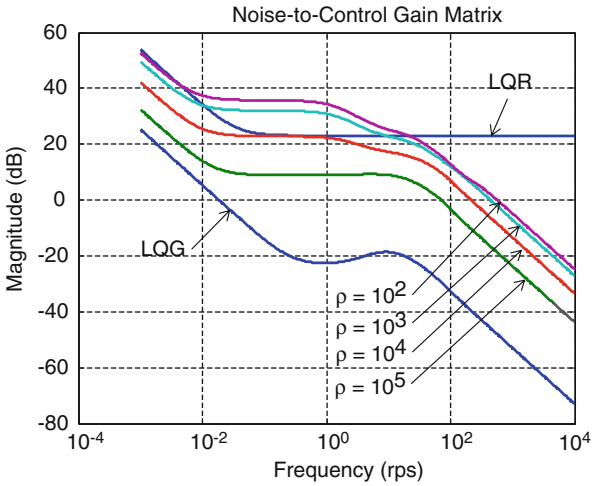


Fig. 6.18 Noise transmission through the controller for LQR and LQG/LTR designs

To finalize a choice of ρ, the decision should be made on maximizing $\underline{\sigma}(I+L)$ and $\underline{\sigma}(I+L^{-1})$ at the plant input, minimizing $\bar{\sigma}(S)$ and $\bar{\sigma}(T)$ at the plant output, and preventing noise amplification over a frequency range of interest. The following table summarizes these peak values:

Design	$\underline{\sigma}(I+L)$	$\underline{\sigma}(I+L^{-1})$	$\bar{\sigma}(S)$	$\bar{\sigma}(T)$
LQR	1.0000	0.7963	1.4936	1.0480
LQR	0.5506	0.9808	1.0791	1.0000
$\rho = 10^5$	0.5233	0.7136	1.0923	1.0000
$\rho = 10^4$	0.5853	0.6567	1.0599	1.0000
$\rho = 10^3$	0.7920	0.7301	1.4581	1.0000
$\rho = 10^2$	0.9160	0.7715	2.9361	2.1570

From the $\underline{\sigma}(I+L)$ values, we need $\rho \leqslant 10^4$ to meet plant input stability margin requirements. We would like $\underline{\sigma}(I+L^{-1})$ to be as large as possible, which is also satisfied by $\rho \leqslant 10^4$. We would like $\bar{\sigma}(S)$ to be minimized, which points to $\rho = 10^4$ as the desired recovery level. If $\rho = 10^3$, the peak in $\bar{\sigma}(S)$ would be too large. Thus, $\rho = 10^4$ is selected as the design. For comparison, the following table lists the Kalman filter gains:

Kalman filter gains	
LQG	LQG/LTR ρ = 10^4
3.2707e−002 −6.0760e + 000	6.9018e + 000 1.7004e + 002
−9.7217e−005 7.8188e−002	2.7206e−003 9.6745e + 000
3.9887e−009 −1.2264e−006	−5.5183e−001 −5.1037e + 001
−9.0547e−007 1.3375e−004	−2.8572e + 000 3.0166e + 003

It is evident that this method increases the gains to large values. Further analysis would be needed to determine if gains of this magnitude could actually be used in a real flight control system.

6.3 Loop Transfer Recovery Using the Lavretsky Method

In the previous section, we combined the optimal control (LQR) with the optimal state estimator (Kalman filter) to form the LQG controller. An LTR tuning process was then used to recover the LQR frequency domain properties in the LQG controlled system by inverting the filter dynamics. In this section, we shall explore an alternate method of LTR, referred to as the LTR method of Lavretsky (LTRLM) [7]. This method is also used later in this book to shape the transient dynamics in model reference adaptive control problems. Detailed derivations of the method and its related equations can be found in Chaps. 13 and 14. In this section, we are going to simply outline the key features of LTRLM and then demonstrate the method and its efficacy through a design example.

Achieving best possible performance and stability robustness properties for a process or a system via control design is the overall goal for the control system engineer to attain. Among linear-time-invariant systems, there is a special class of dynamics, called positive real (PR) and strictly positive real (SPR) [3, 4]. These systems have very interesting properties that enable robust output feedback control design. We present PR and SPR definitions as they are stated in [4].

Definition 6.1. *A $(p \times p)$ proper rational transfer function matrix $G(s)$ of the complex variable $s = \sigma + j\omega$ is called positive real if:*

1. *Poles of all elements of $G(s)$ are in the left half complex plane.*
2. *For all real ω for which $j\omega$ is not a pole of any element of $G(s)$, the matrix $G(j\omega) + G^T(-j\omega)$ is positive semidefinite.*
3. *Any pure imaginary pole $j\omega$ of any element of $G(s)$ is a simple pole, and the residue matrix $\lim_{s \to j\omega} (s - j\omega)G(s)$ is positive semidefinite Hermitian.*

Definition 6.2. *The transfer function $G(s)$ is called strictly positive real if $G(s - \varepsilon)$ is positive real, for some $\varepsilon > 0$.*

For scalar systems $(p = 1)$, PR and SPR dynamics have their Nyquist frequency response locus located entirely in the right half complex plane. This condition for $G(s)$ can be satisfied only if the system's relative degree is zero or one. Thus, encirclements of $(-1, j0)$ cannot occur. In other words, such a system will remain stable under a large set of uncertainties, which is a highly desirable property for any system to possess.

The relationship between PR, SPR transfer functions, and Lyapunov stability theory of the corresponding dynamical system has led to the development of several stability criteria for feedback systems with LTI and nonlinear components. These criteria include Popov's criterion and its variations [8]. The link between PR, SPR transfer function matrices and the existence of a Lyapunov function for studying stability can be established by the following two lemmas [4]:

Lemma 6.1 Positive Real Lemma. *Let $G(s) = C^T(sI - A)^{-1}B + D$ be a $(p \times p)$ transfer function matrix, where (A, B) is controllable and (A, C) is observable. Then, $G(s)$ is positive real if and only if there exist matrices $P = P^T > 0$, L, and W such that*

$$\begin{aligned} PA + A^T P &= -L^T L \\ PB &= C^T - L^T W \\ W^T W &= D + D^T \end{aligned} \tag{6.79}$$

Lemma 6.2 Kalman–Yakubovich–Popov (KYP) Lemma [4]. *Let,*

$$G(s) = C^T(sI - A)^{-1}B + D$$

be a $(p \times p)$ transfer function matrix, where (A, B) is controllable and (A, C) is observable. Then, $G(s)$ is strictly positive real if and only if there exist matrices $P = P^T > 0$, L, W, and a positive constant ε such that

$$\begin{aligned} PA + A^T P &= -L^T L - \varepsilon P \\ PB &= C^T - L^T W \\ W^T W &= D + D^T \end{aligned} \tag{6.80}$$

Clearly, if D is the zero matrix, then the SPR conditions (6.81) reduce to

$$\begin{aligned} PA + A^T P &= -L^T L - \varepsilon P \\ PB &= C^T \end{aligned} \tag{6.81}$$

and in this case, setting $\varepsilon = 0$, gives the PR conditions in the form

$$\begin{aligned} PA + A^T P &= -L^T L \\ PB &= C^T \end{aligned} \tag{6.82}$$

The first relation in (6.83) is the algebraic Lyapunov equation, and $V(x) = x^T P x$ is the Lyapunov function [4]. The second relation in (6.83) enables output feedback control design, whereby the system output $y = C x$ can be fed back into the input to control the system, while preserving closed-loop stability. Also, note that the matrices B and C define the transmission zeros of the system transfer function matrix $G(s) = C(sI - A)^{-1}B$.

We are going to modify the LQG/LTR design such that, for a class of restricted systems, the PR property is obtained asymptotically, $P_v B_v \to C^T$, with the positive tuning parameter $v \to 0$. In addition, we shall ensure that P_v remains symmetric and strictly positive definite, uniformly in v. These are the distinguishing features of LTRLM design. Similar to the previous section, in this design, the Kalman filter is no longer treated as a filter. It will continue to estimate the system state and serve as a dynamic compensator, tuned to improve the frequency domain properties of the system. The Gaussian covariance matrices for w and v are altered significantly to improve the controller robustness and to limit sensor noise amplification. So, these matrices no longer "model" the stochastic processes of the system.

We formulate the LTRLM design approach using the linear-time-invariant Gaussian design model,

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \end{aligned} \tag{6.83}$$

where w and v are zero mean, white, uncorrelated Gaussian random processes with covariances given by

$$\begin{aligned} E\{w(t)w^T(\tau)\} &= Q_0\delta(t-\tau) \\ E\{v(t)v^T(\tau)\} &= R_0\delta(t-\tau) \end{aligned} \tag{6.84}$$

The state estimate $\hat{x}$ is formed as before, using the state estimator,

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y_{\text{meas}} - \hat{y}) \tag{6.85}$$

and the control input is calculated using the LQR state feedback gain matrix K_c, with the estimated state feedback $\hat{x}$.

$$u = -K_c\hat{x} \tag{6.86}$$

In LTRLM, we parameterize the process and measurement noise covariance matrices using a positive scalar v,

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\,\bar{B}^T, \quad R_v = \frac{v}{v+1}R_0 \tag{6.87}$$

where $\bar{B}$ is a matrix formed by adding "fictitious" columns to B, to make $\bar{B} = [B \quad X]$ have its column rank equal to the row rank of C, such that $C\bar{B}$ becomes invertible and the corresponding extended system $C(sI - A)^{-1}\bar{B}$ is minimum phase, that is, all its transmission zeros are located in the left half complex plane. This is the "squaring-up" step of the method. Substituting the weights from (6.88) into the filter Riccati equation, we get

$$P_v A^T + A P_v - \left(1 + \frac{1}{v}\right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v}\right)\bar{B}\,\bar{B}^T = 0 \tag{6.88}$$

or, equivalently

$$P_v A^T + A P_v - P_v C^T R_0^{-1} C P_v + Q_0 + \bar{B}\,\bar{B}^T + \frac{1}{v}\left[\bar{B}\,\bar{B}^T - P_v C^T R_0^{-1} C P_v\right] = 0 \tag{6.89}$$

The gains in (6.86) are computed as

$$K_f = P_v C^T {R_v}^{-1} \tag{6.90}$$

Now as $v \to 0$, one can show that the filter covariance matrix P_v asymptotically approaches a constant symmetric positive definite matrix P_0, that is,

$$P_0 = \lim_{v \to 0} P_v = \lim_{v \to 0} P_v^T = P_0^T > 0 \tag{6.91}$$

This behavior is in contrast to the previous section, whereas the LTR parameter $\rho \to 0$, $\bar{\sigma}(P_f) \to \infty$, $\underline{\sigma}(P_f) \to 0$, and the $P_f$matrix became singular.

The important properties of P_0 in (6.92) are listed below without proof (see Chap. 13, Theorem 13.1 for formal derivations):

- *P_0 is the unique symmetric* strictly *positive definite solution of the following algebraic Lyapunov equation*

$$P_0\left(A - C^T R_0^{-1} C P_1\right)^T + \left(A - C^T R_0^{-1} C P_1\right) P_0 + Q_0 = 0 \tag{6.92}$$

- *There exists a unitary matrix* $W \in R^{m \times m}$ *such that*

$$P_0 C^T = \bar{B} W^T R_0^{\frac{1}{2}} \tag{6.93}$$

- *The unitary matrix* W *in (6.94) can be chosen as*

$$W = (U V)^T \tag{6.94}$$

where U *and* V *are two unitary matrices defined by the singular value decomposition,*

$$\bar{B}^T C^T R_0^{-\frac{1}{2}} = U \Sigma V \tag{6.95}$$

and Σ *represents the diagonal matrix of the corresponding singular values.*

For minimum phase systems, the SPR property is implied by (6.94). What the LTRLM design is trying to do is to shape the transmission zeros of the state estimator, such that the original system with the extended input becomes SPR asymptotically, as $v \rightarrow 0$. To do this, we "square-up the system" by adding extra columns to B (to form $\bar{B}$) and then apply the LTR tuning process, whereby we decrease the tuning parameter v in (6.88), until the system becomes almost SPR.

It was discussed earlier in Chap. 2 that in the LQR design problem, with the penalty matrix Q factored as $Q = Q^{\frac{T}{2}} Q^{\frac{1}{2}}$, the poles of the closed-loop system, $\lambda(A - BK_c)$, would approach the transmission zeros defined by $Q^{\frac{1}{2}}(sI - A)^{-1}B$ asymptotically as the gains grew large. If no finite transmission zeros existed, the roots would form a Butterworth pattern (or combinations of Butterworth patterns) in the left half complex plane. Thus, by the proper selection of Q, the designer places these zeros to achieve the desired response of the system. So, the selection of the LQR penalty matrix is a key tuning mechanism in the LQR controller design.

This same basic idea is in work under LTRLM. For the state estimator (aka Kalman filter), the process covariance Q_f is the equivalent to the LQR penalty matrix. Factoring the process covariance Q_f as $Q_f = L^T L$, the eigenvalues of the Kalman filter, $\lambda\left(A - K_f C\right)$, will approach the finite transmission zeros defined by $C(sI - A)^{-1}L$. Thus, the selection of the process covariance Q_f is an ideal tuning mechanism in the design of the LTRLM controller. Placing the zeros of the system in a desirable location is the key to achieving a robust design. This is achieved through the modified process covariance and measurement noise matrices in (6.88).

6.3.1 Summary

Dynamics: $\dot{x} = Ax + Bu + w \qquad x(t_0) = x_0$

$y = Cx + v$

$E\{x\} = \hat{x}, E\{w\} = 0, E\{v\} = 0,\ \mathrm{cov}(x) = P_v$

$E\{ww^T\} = Q_0\delta(\tau);\ E\{vv^T\} = R_0\delta(\tau)$

Robust Servomechanism LQR:

Command $r =$ constant.

Controlled output (to follow r) : $y_c = C_c x$

$e = y_c - r;\ z = [e \quad \dot{x}]^T, \mu = \dot{u}$

Dynamics: $\dot{z} = \tilde{A}z + \tilde{B}\mu \quad \tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}$

Performance index: $J = \int_0^\infty \left(z^T Q z + \mu^T R \mu\right) d\tau$

$\left(\tilde{A}, \tilde{B}\right)$ Stabilizable, $\left(\tilde{A}, Q^{\frac{1}{2}}\right)$ Detectable

Algebraic Riccati Equation: $P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P$

Optimal Control: $\mu = -R^{-1}B^T P x = -K_c z$

$u = \int \mu = -K_c \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix}$

Kalman Filter State Estimator: $\dot{\hat{x}} = A\hat{x} + Bu + K_f(y_{meas} - \hat{y})$

Square of the system: $C\begin{bmatrix} B & \bar{B} \end{bmatrix}$ has full rank

$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\bar{B}^T, \quad R_v = \frac{v}{v+1}R_0$

$P_v A^T + A P_v - \left(1 + \frac{1}{v}\right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v}\right)\bar{B}\bar{B}^T = 0$

$K_f = P_v C^T {R_v}^{-1}$

The LTRLM controller design, examining the loop properties at the plant input, may be realized through the following synthesis technique:

Step 1: LQR controller design: K_c

Follow the robust servomechanism design approach outlined in Chap. 3. Design LQR weighting matrices Q and R, such that the resulting loop gain $L_{\mathrm{LQR}}(s) = K_c\left(sI - \tilde{A}\right)^{-1}\tilde{B}$ meets performance and stability robustness requirements and exhibits the desired bandwidth.

Step 2: State estimator/Kalman filter design: K_f

Select columns X to make $\bar{B} = [B \quad X]$ have column rank equal to the row rank of C and to make the extended system minimum phase. Design the Kalman filter/state estimator using (6.89), with (6.88) defining the plant process and measurement noise covariance matrices. The LTR parameter v is used to recover the LQR frequency domain characteristics over the frequency range of interest. Ad hoc adjustment of the sensor noise covariance magnitude may be needed to scale the Kalman gains to prevent large gains from occurring. Examine plant input and output frequency domain criteria and the sensor noise amplification in and limit the LTR recovery so that the sensor noise is not amplified.

Example 6.3 LTRLM Design for Pitch Dynamics of an Aircraft. The flight control design example we have been using to demonstrate the various output feedback control design methods does not satisfy the requirements for using LTRLM. This method requires adding extra control columns X to B so that $C[B \quad X]$ is invertible, and the corresponding extended dynamics are minimum phase. We begin by approximating the aircraft model with minimum phase dynamics. This step is not required since we could have accomplished it through the squaring-up procedure. However, we have found that starting with minimum phase dynamics simplifies the overall design process. For the aircraft pitch dynamics, we can do this easily by neglecting the tail vertical force Z_δ which is small in most applications. Normally, the acceleration transfer function has an RHP zero. When we zero Z_δ, this transfer function no longer has any finite zeros.

To satisfy the requirement for $C[B \quad X]$ be invertible will require us to alter the design problem by removing the actuator model. This reduces the number of states, creating a second-order design model. In this example, we will compare the design using LTRLM with the LQG procedure, as well as with the conventional LQG/LTR method of the previous section.

Toward that end, we consider aircraft longitudinal dynamics in the form

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha\alpha + M_\delta\delta + M_q q\end{aligned} \tag{6.96}$$

where angle-of-attack α and pitch rate q are the states, and the elevator position δ represents the system control input. The measured outputs consist of the vertical acceleration A_z and the pitch rate q.

$$\begin{bmatrix} A_z \\ q \end{bmatrix} = \begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \delta \tag{6.97}$$

We will neglect the tail vertical force $(Z_\delta = 0)$ and form the minimum phase plant model (A_p, B_p, C_p, D_p) as

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \begin{bmatrix} -1.0527 & 1 & 0 \\ -2.3294 & -1.0334 & -1.1684 \\ -346.48 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{6.98}$$

First, we perform the LQR state feedback design. To track an acceleration command, the RSLQR state feedback design model (6.24) is

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} 0 & C_p(1,:) \\ 0 & A_p \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ B_p \end{bmatrix} \\ \tilde{A} &= \begin{bmatrix} 0 & -346.48 & 0 \\ 0 & -1.0527 & 1 \\ 0 & -2.3294 & -1.0334 \end{bmatrix} \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ -1.1684 \end{bmatrix} \end{aligned} \tag{6.99}$$

Using the weight matrices $Q = diag[\,0.2448 \quad 0 \quad 0\,]$ and $R = 1$, the LQR state feedback gain matrix is

$$K_c = [\,0.31623 \quad -33.261 \quad -6.7127\,] \tag{6.100}$$

The above control law is implemented as

$$u = -K_c \left[\int (A_{z_m} - r) \quad \alpha \quad q \right]^T \tag{6.101}$$

To analyze and compare our controllers, we will implement each one in our standard controller model:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{6.102}$$

Using the gains from (6.101), the state feedback controller is

$$\begin{aligned} \dot{x}_c &= [0] x_c + [\,1 \quad 0\,] y + [-1] r \\ u &= [-0.31623] x_c + [\,33.261 \quad 6.7127\,] y + [0] r \end{aligned} \tag{6.103}$$

To form the closed-loop system, the above controller is connected to the plant model.

Next is the LQG design. The nominal process and measurement noise covariance matrices are

$$Q_0 = \begin{bmatrix} 0.000196 & 0 \\ 0 & 0.0025 \end{bmatrix} \begin{bmatrix} (\text{rad})^2/s \\ (\text{rps})^2/s \end{bmatrix} \tag{6.104}$$

and

$$R_0 = \begin{bmatrix} 6.25 \times 10^{-2} & \\ & 1.0 \times 10^{-6} \end{bmatrix} \begin{bmatrix} (\text{fps})^2 \\ (\text{rps})^2 \end{bmatrix} \tag{6.105}$$

The Kalman filter state estimator is

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \tag{6.106}$$

where u is formed using (6.102) and was implemented using steady-state matrices obtained from the filter covariance equation

$$\begin{aligned} 0 &= AP_f + P_f A^T + Q_v - P_f C^T {R_v}^{-1} C P_f \\ K_f &= P_f C^T {R_v}^{-1} \end{aligned} \tag{6.107}$$

The steady-state covariance and Kalman filter gains design (using Q_0 and R_0) are

$$\begin{aligned} P_f &= \begin{bmatrix} 9.5843\text{e}-006 & 3.8344\text{e}-007 \\ 3.8344\text{e}-007 & 4.8957\text{e}-005 \end{bmatrix} \\ K_f &= \begin{bmatrix} -0.053132 & 0.38344 \\ -0.0021257 & 48.957 \end{bmatrix} \end{aligned} \tag{6.108}$$

To analyze this observer-based design (Kalman filter), we will implement the controller in our standard model (6.103). For the LQG controller, the RSLQR control law is given by

$$\begin{aligned} u &= -K_1 \int e - K_x \hat{x} \\ e &= y_c - r = A_z - A_{z_{cmd}} \end{aligned} \tag{6.109}$$

where $\hat{x}$ is the estimated state, and the RSLQR gain matrix is partitioned as $K_c = [K_1 \quad K_x]$. To form the estimated state, we need to substitute the control (6.110) into the state estimator (6.107). Doing so gives

$$\begin{aligned}
\dot{\hat{x}} &= A_p\hat{x} + B_p u + K_f(y - \hat{y}) \\
\dot{\hat{x}} &= A_p\hat{x} + B_p\left(-K_1\int e - K_x\hat{x}\right) + K_f\left(y - \left(C_p\hat{x} + D_p\left(-K_1\int e - K_x\hat{x}\right)\right)\right) \\
\dot{\hat{x}} &= \underbrace{\left(A_p - \left(B_p - K_f D_p\right)K_x - K_f C_p\right)}_{A_{22}}\hat{x} \underbrace{-\left(B_p - K_f D_p\right)K_1}_{A_{21}}\int e + K_f y
\end{aligned} \tag{6.110}$$

The LQG controller states are $x_c = \begin{bmatrix}\int e & \hat{x}\end{bmatrix}^T$. The controller state space model using (6.110) and (6.111) is

$$\begin{aligned}
\begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ K_f \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} r \\
u &= -\begin{bmatrix} K_1 & K_x \end{bmatrix}\begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0] y_{meas} + [0] r
\end{aligned} \tag{6.111}$$

where A_{21} and A_{22} are defined as in (6.111). Substituting the gains into (6.112), we have

$$\begin{aligned}
\begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -19.462 & 0.61656 \\ 0.36948 & -41.928 & -57.833 \end{bmatrix}\begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -0.053132 & 0.38344 \\ -0.0021257 & 48.957 \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} r \\
u &= [-0.31623\ 33.261\ 6.7127]\begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0\,0] y_{\text{meas}} + [0] r
\end{aligned} \tag{6.112}$$

Next is the LQG/LTR design. This method (from the previous section) adds a term to the process noise covariance matrix as

$$Q_f = Q_0 + \frac{1}{\rho} B_p B_p^T \tag{6.113}$$

We varied the LTR parameter ρ and selected a value of $\rho = 25$. This produces

$$Q_f = \begin{bmatrix} 0.000196 & 0 \\ 0 & 0.057106 \end{bmatrix} \tag{6.114}$$

which results in

$$P_f = \begin{bmatrix} 9.5933\text{e}-006 & 8.3416\text{e}-007 \\ 8.3416\text{e}-007 & 0.00023793 \end{bmatrix}$$
$$K_f = \begin{bmatrix} -0.053183 & 0.83416 \\ -0.0046243 & 237.93 \end{bmatrix} \tag{6.115}$$

Note the magnitude increase in the gain $K_f(2,2)$.

The last controller in this example uses the LTRLM. The first step in the design process is to design the LQR control law. We will use the RSLQR controller from (6.102). The second step is to select columns X to make $\bar{B} = [B_p \quad X]$ have column rank equal to the row rank of C_p. To complete this design, we must look at the numbers within these matrices:

$$C_p = \begin{bmatrix} -346.48 & 0 \\ 0 & 1 \end{bmatrix} B_p = \begin{bmatrix} 0 \\ -1.1684 \end{bmatrix} \bar{B} = \begin{bmatrix} B_p & \bar{b}_{21} \\ & \bar{b}_{22} \end{bmatrix} \tag{6.116}$$

To add a second column in $\bar{B}$, we see that any values are possible, except $\bar{b}_{21} = 0$. If $\bar{b}_{21} = 0$, then

$$\bar{B} = \begin{bmatrix} 0 & 0 \\ -1.1684 & \bar{b}_{22} \end{bmatrix}' \tag{6.117}$$

which is rank 1. To evaluate the effect of this selection, we will examine two designs, described

$$\bar{B}_1 = \begin{bmatrix} 0 & 1 \\ -1.1684 & 0 \end{bmatrix} \text{ and } \bar{B}_2 = \begin{bmatrix} 0 & 0.1 \\ -1.1684 & 10. \end{bmatrix} \tag{6.118}$$

To improve the numerical scaling between Q_f and R_f we will scale R_0 by 250. The process noise and measurement covariance matrices are given by

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) \bar{B}\,\bar{B}^T, \quad R_v = \frac{v}{v+1}\, 250 \tag{6.119}$$

For $v = 2.5$ and using $\bar{B} = \bar{B}_1$, we have

$$Q_v = \begin{bmatrix} 0.000196 & 0 \\ 0 & 0.0025 \end{bmatrix} + 1.4 \begin{bmatrix} 1 & 0 \\ 0 & 1.3652 \end{bmatrix} = \begin{bmatrix} 1.4002 & 0 \\ 0 & 1.9137 \end{bmatrix}$$
$$R_v = (0.71429)(250) \begin{bmatrix} 0.0625 & 0 \\ 0 & 1\text{e}-006 \end{bmatrix} = \begin{bmatrix} 11.161 & 0 \\ 0 & 0.00017857 \end{bmatrix} \tag{6.120}$$

Solving for the steady-state covariance and gain matrix from (6.108) yields

$$P_f = \begin{bmatrix} 0.011312 & -3.5561\text{e}-005 \\ -3.5561\text{e}-005 & 0.018303 \end{bmatrix}$$
$$K_f = \begin{bmatrix} -0.35116 & -0.19914 \\ 0.001104 & 102.5 \end{bmatrix} \tag{6.121}$$

The controller is formed by substituting the gains K_f into (6.112) and results in

$$\begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -19.48 & 0.16584 \\ 0.36948 & -42.794 & -246.8 \end{bmatrix} \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -0.053183 & 0.83416 \\ -0.0043243 & 237.93 \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} r$$
$$u = [-0.31623\ 33.261\ 6.7127] \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0\ 0] y_{\text{meas}} + [0] r \tag{6.122}$$

For $v = 2.5$ and using $\bar{B} = \bar{B}_2$, we obtain

$$Q_v = \begin{bmatrix} 0.000196 & 0 \\ 0 & 0.0025 \end{bmatrix} + 1.4 \begin{bmatrix} 0.01 & 1 \\ 1 & 101.37 \end{bmatrix} = \begin{bmatrix} 0.014196 & 1.4 \\ 1.4 & 141.91 \end{bmatrix}$$
$$R_v = (0.71429)(250) \begin{bmatrix} 0.0625 & 0 \\ 0 & 1\text{e}-006 \end{bmatrix} = \begin{bmatrix} 11.161 & 0 \\ 0 & 0.00017857 \end{bmatrix} \tag{6.123}$$

Solving for the steady-state covariance and gain matrices from (6.108) yields

$$P_f = \begin{bmatrix} 0.00017018 & 0.0017429 \\ 0.0017429 & 0.15898 \end{bmatrix}$$
$$K_f = \begin{bmatrix} -0.0052832 & 9.7605 \\ -0.054109 & 890.31 \end{bmatrix} \tag{6.124}$$

The controller is formed by substituting the gains K_f into (6.112) and results in

$$\begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -122.72 & 1.1991 \\ 0.36948 & -40.809 & -111.37 \end{bmatrix} \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -0.35116 & -0.19914 \\ 0.001104 & 102.5 \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} r$$
$$u = [-0.31623\ 33.261\ 6.7127] \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0\ 0] y_{meas} + [0] r \tag{6.125}$$

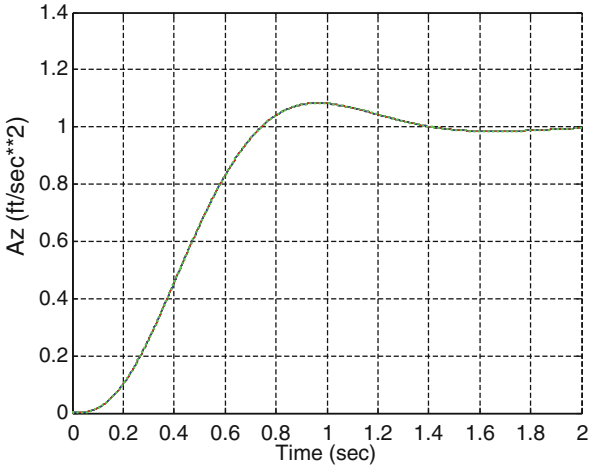


Fig. 6.19 Step response for the LQR, LQG, LQG/LTR, and LQG/LTRLM controller designs

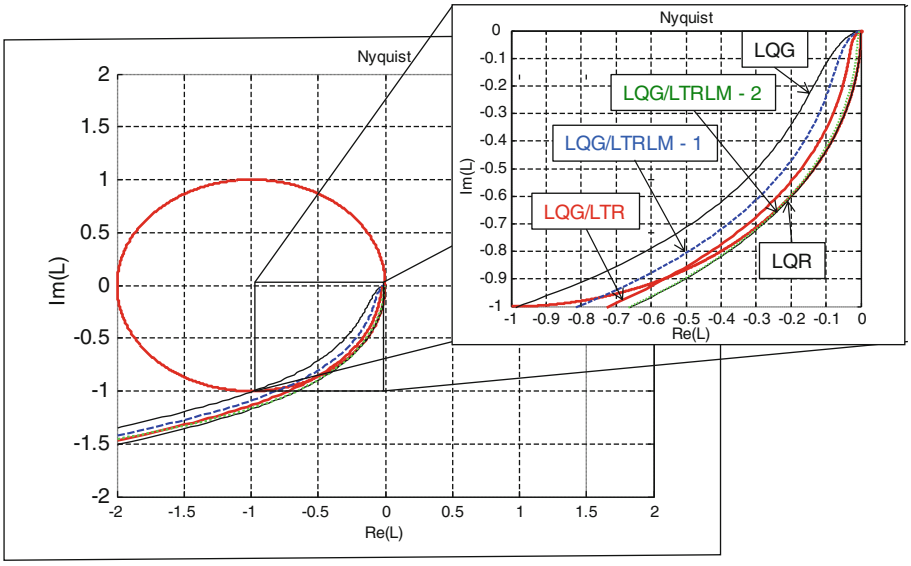


Fig. 6.20 Nyquist plot for the LQR, LQG, LQG/LTR, and LTRLM controllers

By varying the columns in $\bar{B}$, we get significantly different controllers. The second choice has a larger Q_v, which results in larger gains. We can expect that these gains will recover the LQR properties better than the designs with smaller gains.

Figure 6.19 shows a step response for the closed-loop system using all five controllers (LQR, LQG, LQG/LTR, and the two LTRLM controllers). The plot shows that the time domain simulation results are identical for all the designs. Figures 6.20, 6.21, 6.22, 6.23, and 6.24 show the frequency domain analysis of

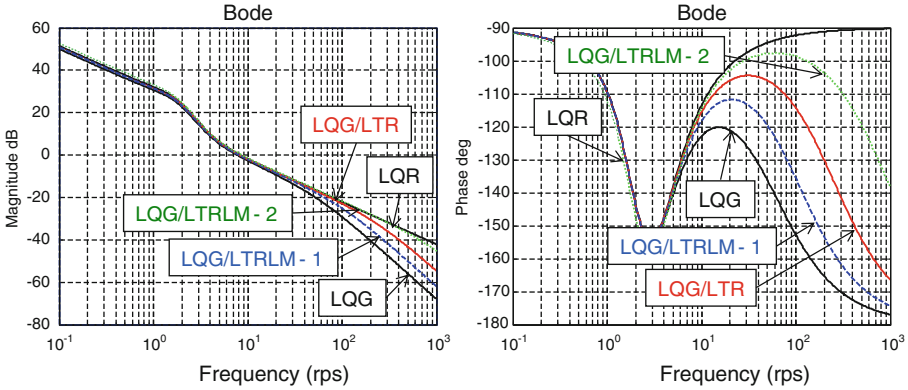


Fig. 6.21 Bode plots for the LQR, LQG, LQG/LTR, and LTRLM controllers

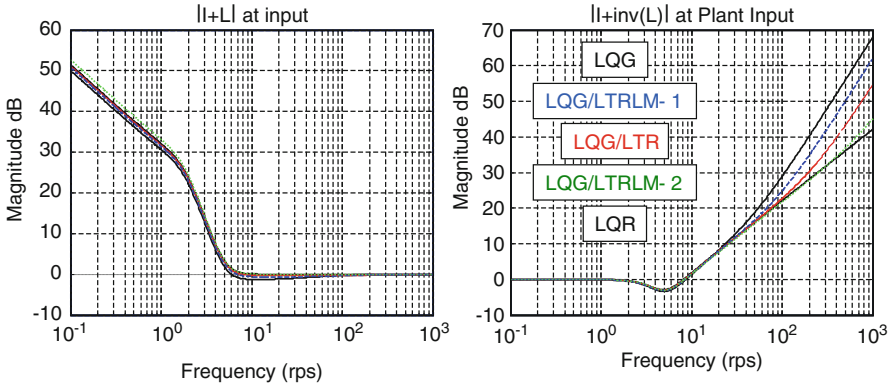


Fig. 6.22 $|I + L|$ and $\left|I + L^{-1}\right|$ at the plant input for the LQR, LQG, LQG/LTR, and LTRLM controllers

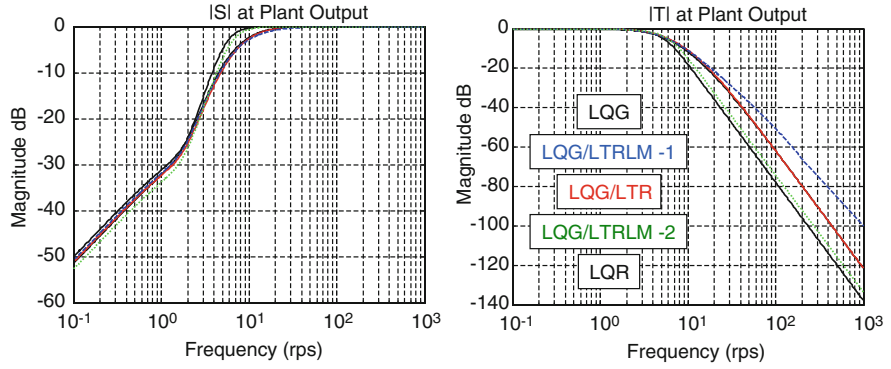


Fig. 6.23 $|S|$ and $|T|$ at the acceleration output for the LQR, LQG, LQG/LTR, and LTRLM controllers

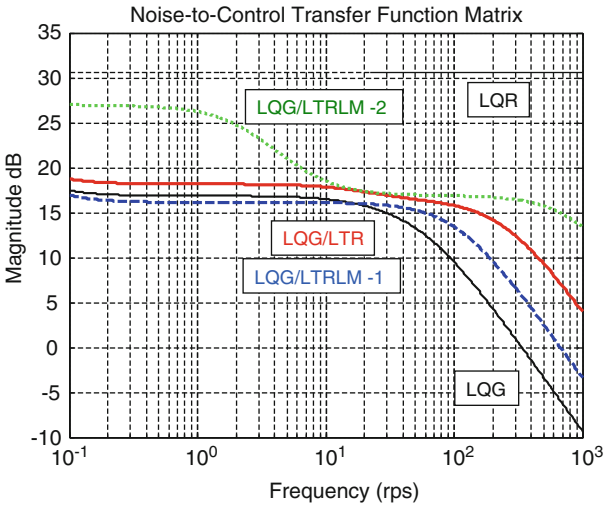


Fig. 6.24 Noise transmission through the controller for the LQR, LQG, LQG/LTR, and LTRLM controllers

these controllers. Figure 6.20 shows the Nyquist plot. We see that the LQR design (black line) does not enter the red unit circle centered at $(-1, j0)$. The LQG design (also black line) is the locus to the left which has the degraded gain margin and phase margins properties. Note that the LQG design here is not as bad as in the previous section due to the actuator being neglected within this model. The LQG/LTR design with LTR parameter $\rho = 25$ is the red curve. The two designs using the LTRLM approach (blue and green curves) bracket the LQG/LTR locus. The LTRLM method with $\bar{B} = \bar{B}_2$ (green curve) has the most recovery (closest to the LQR black curve).

We see from the figures that the LQG/LTR and LTRLM methods can all be tuned to recover the LQR properties. These methods all recover the properties by increasing the Kalman filter gains. Care must be taken to prevent the gains from getting too large. The LTRLM method can achieve the recovery with smaller overall gains as compared to the conventional LQG/LTR method. The LQG/LTR method adds $B_p B_p^T$ to Q_0, while the LTRLM method adds $\bar{B}\,\bar{B}^T$. For this example, these are

$$B_p B_p^T = \begin{bmatrix} 0 & 0 \\ 0 & 1.3652 \end{bmatrix}; \bar{B}\,\bar{B}^T = \begin{bmatrix} 0.01 & 1 \\ 1 & 101.37 \end{bmatrix} \tag{6.126}$$

The additional parameters offer an improvement by distributing the recovery into additional loops within the architecture.

6.4 Conclusions

In this chapter, we presented static output feedback and dynamic projective control and two LQG/LTR design methods. These design methods in no way capture all the available output feedback methods available to the engineer. We selected these methods because they have proven to be good design methods, and more importantly for the student, they demonstrate the insight needed to develop control system in practice.

The static projective control method has been found to be very effective at designing output feedback controllers. In flight control applications using gain scheduling, these controllers are low order, making them easy to implement.

LQG/LTR controllers require a dynamic observer for implementation. In flight control applications where gain scheduling is relied upon to compensate for a large flight envelope, these observers can introduce small transients as the observer parameters vary. The engineer must simulate and evaluate if these transients are acceptable.

The exercises that follow for this chapter take a longitudinal flight control problem and assign each design method. Any plant dynamics could be used for these exercises. The key is to learn how to design and compare designs so that both time domain performance and frequency domain robustness requirements are met.

6.5 Exercises

Exercise 6.1. Consider the unstable longitudinal dynamics model, as defined in Example 6.1, where $x = \begin{bmatrix} \alpha & q & \delta_e & \dot{\delta}_e \end{bmatrix}^T$. The matrices for the control design model $\dot{x} = A_p x + B_p u$ are

$$\begin{bmatrix} A_p & B_p \end{bmatrix} = \left[\begin{bmatrix} -1.3046\text{e} & 1.0 & -0.2.1420 & 0 \\ 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 0 & -12769. & -135.6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12769 \end{bmatrix} \right]$$

(a) Design a robust servomechanism LQR state feedback control to track a constant αcommand using the method of Chap. 3. Simulate the state feedback design to show the command tracking.

(b) It is desired not to feedback the elevator state and rate signals to improve reliability. Use the static projective control method of Sect. 6.1 to project out the actuator dynamics, keeping the dominant eigenstructure for command

tracking. Simulate the static projective control design to show the command tracking and compare with (a).

(c) Compute the eigenstructure for (a) and (b) to show that the dominant eigenvalues are retained. Analyze this design in the frequency domain. Compute Nyquist, Bode, $\underline{\sigma}[I+L]$, $\underline{\sigma}[I+L^{-1}]$ frequency responses for a) and b) at the plant input. Compute $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ frequency responses for a) and b) at the plant output for the α loop. Compute the loop gain crossover frequency and singular value stability margins for the design.

Exercise 6.2. Consider the unstable longitudinal dynamics from Exercise 6.1. The output signals available from the inertial measurement unit are $y = [A_z \quad q]^T$. The matrices for the output model $y = C_p x + D_p u$ are

$$[C_p \quad D_p] = \left[\begin{bmatrix} -1156.9 & 0 & -189.95 & 0 \\ 0 & 1.0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right]$$

(a) Design a robust servomechanism LQR state feedback control to track a constant αcommand using the method of Chap. 3. (Same controller from Exercise 6.1). Design a full state Kalman filter observer to estimate the states for feedback using the method outlined in Sect. 6.2. Use the following plant process and measurement noise covariance matrices for the Kalman filter design:

$$Q_0 = \begin{bmatrix} 1.0\times10^{-8} & 0 & 0 & 0 \\ 0 & 2.5\times10^{-7} & 0 & 0 \\ 0 & 0 & 1.0\times10^{-8} & 0 \\ 0 & 0 & 0 & 1.0\times10^{-6} \end{bmatrix}\begin{bmatrix} (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \\ (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 6.25\times10^{-2} & 0 \\ 0 & 1.0\times10^{-6} \end{bmatrix}\begin{bmatrix} (\text{fps})^2 \\ (\text{rps})^2 \end{bmatrix} \tag{6.127}$$

List all matrices used in the design.

(b) Simulate the LQG design and compare it to the state feedback design.

(c) Analyze this LQG design in the frequency domain. Compute Nyquist, Bode, $\underline{\sigma}[I+L]$, $\underline{\sigma}[I+L^{-1}]$ frequency responses for the LQG and state feedback at the plant input. Compute $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ frequency responses for (a) and (b) at the plant output for the α loop. Compute the loop gain crossover frequency and singular value stability margins for both designs. Determine the impact of using the Kalman filter estimator on the stability robustness of the system.

(d) Use the LTR method of Sect. 6.2 (6.59) to recover the frequency domain properties of the state feedback design in the LQG design. Evaluate the design in the frequency domain as in (c). Compute the maximum singular value of the noise-to-control transfer function matrix frequency response to examine the noise amplification in the resulting LQG/LTR design.

Exercise 6.3. Consider the unstable longitudinal dynamics in Example 6.3 and Exercise 6.1. The output signals available from the inertial measurement unit are $y = [A_z \quad q]^T$. The matrices for the output model $y = C_p x + D_p u$ are

$$[C_p \quad D_p] = \left[\begin{bmatrix} -1156.9 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right]$$

Note that the lift term due to the elevator has been zeroed.

(a) Design a robust servomechanism LQR state feedback control to track a constant α command using the method of Chap. 3. Design a full state Kalman filter observer to estimate the states for feedback using the method outlined in Sect. 6.3. Use the following plant process and measurement noise covariance matrices for the Kalman filter design:

$$Q_0 = \begin{bmatrix} 1.0 \times 10^{-8} & 0 & 0 & 0 \\ 0 & 2.5 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 1.0 \times 10^{-8} & 0 \\ 0 & 0 & 0 & 1.0 \times 10^{-6} \end{bmatrix}\begin{bmatrix} (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \\ (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 6.25 \times 10^{-2} & 0 \\ 0 & 1.0 \times 10^{-6} \end{bmatrix}\begin{bmatrix} (\text{fps})^2 \\ (\text{rps})^2 \end{bmatrix} \tag{6.128}$$

List all matrices used in the design.

(b) Simulate the LQG design and compare it to the state feedback design.

(c) Analyze this LQG design in the frequency domain. Compute Nyquist, Bode, $^{-}\sigma[I + L], ^{-}\sigma[I + L^{-1}]$ frequency responses for the LQG and state feedback at the plant input. Compute $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ frequency responses for (a) and (b) at the plant output for the α loop. Compute the loop gain crossover frequency and singular value stability margins for both designs. Determine the impact of using the Kalman filter estimator on the stability robustness of the system.

(d) Use the Loop Transfer Recovery method of Lavretsky, Sect. 6.2 (6.88), to recover the frequency domain properties of the state feedback design in the LQG design. Evaluate the design in the frequency domain as in (c). Compute the maximum singular value of the noise-to-control transfer function matrix frequency response to examine the noise amplification in the resulting LQG/LTR design.

References

1. Hopkins Jr., W.E., Medanic, J., Perkins, W.R.: Output feedback pole placement in the design of suboptimal linear quadratic regulators. Int. J. Contr. **34**, 593–612 (1981)
2. Medanic, J., Uskokovic, Z.: The design of optimal output regulators for linear multivariable systems with constant disturbances. Int. J. Contr. **37**, 809–830 (1983)

3. Tharp, H.S., Medanic, J.V., Perkins W.R.: Robust Projective Controls for Structured Perturbations. In: Proceedings of the American control conference, Seattle, WA, pp. 1833–1838
4. Doyle, J.C., Stein, G.: Multivariable feedback design: Concepts for a classical/modern synthesis. IEEE Trans. Automat. Contr. **26**(1), 4–16 (1981)
5. Stein, G., Athans, M.: The LQG/LTR procedure for multivariable feedback control design. IEEE Trans. Automat. Contr. **32**(2), 105–114 (1987)
6. Khalil, H.: Nonlinear Systems, 3rd edn, p. 07458. Prentice Hall, Upper Saddle River (1996)
7. Lavretsky, E.: Adaptive Output Feedback Design Using Asymptotic Properties of LQG/LTR Controllers," In: Proceedings of the AIAA guidance, navigation and control conference, Toronto, Ontario, Canada (2010).
8. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, New Jersey (1996)

Part II
Robust Adaptive Control

Chapter 7
Direct Model Reference Adaptive Control: Motivation and Introduction

7.1 Model Reference Control: Motivational Example

In the design of flight control systems, it is essential to provide closed-loop stability, adequate command tracking performance, as well as robustness to model uncertainties, control failures, and environmental disturbances. In the previous chapters, we considered optimal linear quadratic regulator (LQR) control design techniques that were suitable for flight control of aerial systems. These design methods relied on the inherent robustness properties of LQR optimal controllers. It was shown that with a proper selection of the LQR design tuning parameters (Q and R matrices), one could achieve 6 dB gain margin, and at least 60° phase margin, at the system control input break points.

It is also possible to show that LQR optimal controllers can tolerate time-state-dependent nonlinear uncertainties that might be present in the system control channels. These uncertainties are called "matched" since they appear only where control inputs exist in the system dynamics. The matching conditions imply that if the system uncertainties were known, a controller would have the ability to cancel them out.

In the presence of matched uncertainties, a deterioration of the system baseline closed-loop performance is inevitable. This is to be expected since the LQR controllers are designed to be robust to the entire class of matched uncertainties. However, they are not tuned to handle any specific uncertainty from this class. In other words, these LQR controllers may become overly conservative.

We pose the question: "Can we restore a given baseline closed-loop performance of the system, while operating under matched uncertainties?" The answer is "yes." This is the area where adaptive controllers are highly effective.

Throughout the chapters of Part II, we shall utilize the concept of a reference model for specifying the desired closed-loop tracking performance. Fixed-gain controllers, as well as adaptive systems, can be constructed using the reference model-based design concept. We shall begin our discussions with a motivational example.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_7,

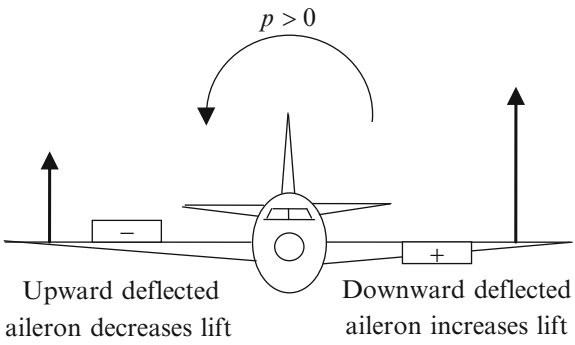


Fig. 7.1 Lift forces arising from positive differential aileron deflection cause aircraft to roll counterclockwise (positive roll rate)

Example 7.1 Fixed-Gain Model Reference Control of Aircraft Roll Dynamics. The roll dynamics of a conventional aircraft are controlled using differential motion of ailerons and spoilers. Ailerons are movable surfaces that are mounted outboards on the trailing edge of the wing, where they are placed symmetrically on each side of the wing, with respect to the aircraft centerline (Fig 7.1).

Deflected differentially (e.g., downward on one side and upward on the other), ailerons have the ability to increase the lift force on the downward deflected portion of the wing and to decrease it on the other side. The two distinct lift forces will create a rolling moment around the aircraft velocity vector placed at the aircraft center of gravity. While ailerons can move up and down, spoilers can only be deflected upward above the trailing edge of the wing to reduce the lift force and thus to aid ailerons in providing roll control. As a result, the aircraft rotates around its velocity vector. In this case, the aircraft roll dynamics can be approximated by a scalar (first-order) ordinary differential equation (ODE) in the form

$$\dot{p} = L_p\, p + L_{\delta_a}\, \delta_a \tag{7.1}$$

where p is the aircraft roll rate in stability axes (radians/s), δ_a is the total differential aileron-spoiler deflection (radians), L_p is the roll damping derivative, and L_{δ_a} is the dimensional rolling moment derivative with respect to differential aileron-spoiler deflection, (the aileron-to-roll control effectiveness). For a conventional open-loop-stable aircraft, the roll damping derivative L_p is negative, unless portions of the wing are stalled, in which case the roll damping may become positive. Positive differential aileron-spoiler deflection is defined to produce positive rolling moment, and as such, the aileron-to-roll control effectiveness L_{δ_a} typically has positive values.

Strictly speaking, the roll dynamics approximation above is valid only for sufficiently small values of p and δ_a. In addition, it is assumed that the aircraft yawing motion is suppressed by the rudder – a vertical tail mounted surface. Readers who might be unfamiliar with the flight mechanics nomenclature may consider (7.1) as a scalar ODE $\dot{x} = a\, x + b\, u$, with two constant parameters $a = L_p$, $b = L_{\delta_a}$, whose state and control input are $x = p$ and $u = \delta_a$, respectively.

The control task of interest is to force the aircraft to roll like the reference model,

$$\dot{p}_{ref} = a_{ref}\, p_{ref} + b_{ref}\, p_{cmd} \tag{7.2}$$

with the prescribed values of $a_{ref} < 0$ (the desired inverse time constant) and $b_{ref} > 0$ (the desired DC gain). The reference model (7.2) is driven by the commanded roll rate p_{cmd} and it calculates the reference roll rate p_{ref}. In essence, the reference model (7.2) imbeds and defines the desired closed-loop command tracking performance. The control task amounts to finding δ_a that would force the aircraft roll rate p track any bounded, possibly time-varying, reference command p_{ref}. This is the model reference control design task. Sometimes, it is also referred to as the model following control. Using this concept allows the designer to create controllers whose main task is to asymptotically match a given reference model behavior. Let us now explore details of the model reference control design.

Comparing the roll dynamics (7.1) to that of the reference model (7.2), it is easy to see that a control solution can be formulated in the feedback-feedforward form

$$\delta_a = \left(\frac{a_{ref} - L_p}{L_{\delta_a}}\right) p + \left(\frac{b_{ref}}{L_{\delta_a}}\right) p_{cmd} \tag{7.3}$$

where $k_p = \left(\frac{a_{ref} - L_p}{L_{\delta_a}}\right)$ is the roll rate feedback gain, and $k_{p_{cmd}} = \left(\frac{b_{ref}}{L_{\delta_a}}\right)$ is the command feedforward gain. In fact, substituting the controller (7.3) into the roll dynamics (7.1), gives the desired closed-loop system dynamics.

$$\dot{p} = a_{ref}\, p + b_{ref}\, p_{cmd} \tag{7.4}$$

In order to formally assess if (7.4) indeed converges to (7.2), we first define the roll rate tracking error,

$$e = p - p_{ref} \tag{7.5}$$

and then compute the tracking error dynamics by differentiating e with respect to time, while substituting (7.4) and (7.2).

$$\dot{e} = \dot{p} - \dot{p}_{ref} = a_{ref}\left(p - p_{ref}\right) = a_{ref}\, e \tag{7.6}$$

Since by definition $a_{ref} < 0$ (e.g., the reference model is exponentially stable), the error dynamics (7.6) are globally exponentially stable. Therefore, given any initial values $p(0)$ and $p_{ref}(0)$, the tracking error $e(t)$ will converge to the origin exponentially fast,

$$e(t) = \exp\left(a_{ref}\, t\right) e(0) \tag{7.7}$$

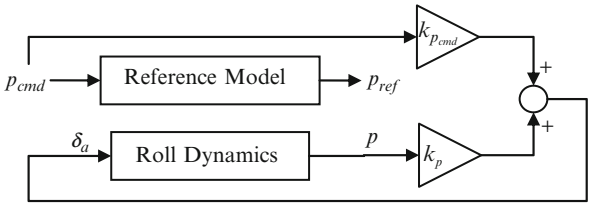


Fig. 7.2 Block diagram of the closed-loop roll dynamics with fixed-gain model reference controller obtained in Example 7.1

starting at any initial tracking error value $e(0) = p(0) - p_{ref}(0)$. So, the aircraft roll rate $p(t)$ will track the reference roll rate $p_{ref}(t)$, with the exponentially fast decaying tracking error $e(t)$,

$$p(t) = p_{ref}(t) + \exp\left(a_{ref}\, t\right)\left(p(0) - p_{ref}(0)\right) \tag{7.8}$$

and this closed-loop tracking performance is valid for any constant or bounded time-varying command $p_{cmd} = p_{cmd}(t)$. The command tracking problem is solved. The corresponding closed-loop system block diagram with the fixed-gain model reference controller (7.3) is shown in Fig. 7.2.

The model reference controller (7.3) is by no means unique in solving the command tracking problem of interest. Other solutions can be found. For example, any controller in the form

$$\delta_a = k_p\, p + k_{p_{cmd}}\, p_{cmd} - k_e\left(p - p_{ref}\right) \tag{7.9}$$

solves the same tracking problem, where $k_e \geq 0$ represents the error feedback gain.

However, does the error feedback in (7.9) give any advantage over the original controller (7.3)? In order to answer that question, let us calculate the error dynamics obtained using the modified controller (7.9).

$$\dot{e} = \left(a_{ref} - k_e\right) e \tag{7.10}$$

Consequently,

$$p(t) = p_{ref}(t) + \exp\left(\left(a_{ref} - k_e\right) t\right)\left(p(0) - p_{ref}(0)\right) \tag{7.11}$$

By definition, the error dynamics (7.10) define the transients that are incurred by the system while tracking a given reference command $p_{ref}(t)$. It is now evident that choosing $k_e > 0$ sufficiently large will allow the designer to obtain any desired (fast) transient dynamics. This constitutes the primary advantage of using an error feedback gain in the fixed-gain model reference controller (7.9). Figure 7.3 shows the resulting closed-loop system diagram.

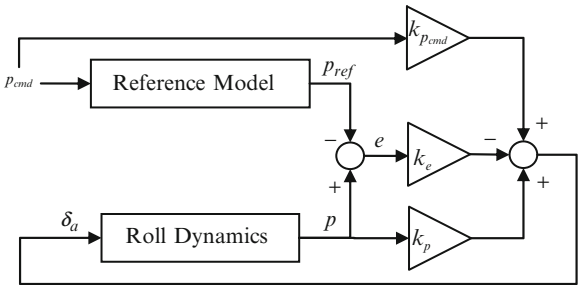


Fig. 7.3 Closed-loop system block diagram with fixed-gain model reference controller and error feedback obtained in Example 7.1

Of course, practical limitations, as well as stability robustness considerations, will place upper and lower limits on the selection of the controller gains. Eventually, these restrictions will dictate the trade-off between achievable transients in the closed-loop system and adequate stability robustness margins. □

7.2 Introduction to Direct Model Reference Adaptive Control

In the roll control example above, we have assumed that the system dynamics (7.1) (defined by the aircraft aerodynamics) were completely known. Then, we utilized the roll damping L_p and the aileron control effectiveness L_{δ_a} to design the two fixed-gain model reference controllers, (7.3) and (7.9).

In reality, aerodynamic parameters are rarely known exactly. This type of uncertainty is called parametric. If the true parameters are substantially different from their assumed constant values, controllers such as (7.9) can lead to instabilities in the system. Even when the system remains stable in the presence of parametric uncertainties, its closed-loop tracking performance may deteriorate to a point of becoming unacceptable.

Robustness considerations may not always solve the parameter sensitivity problem. Often, robust controllers will have a conservatism built into their design, and as such, they may not be able to provide adequate tracking performance, when operating under specific parametric uncertainties. This leads to the idea of adding a gain adaptation mechanism and arriving at model reference adaptive controllers.

Example 7.2 Model Reference Adaptive Control of Aircraft Roll Dynamics Suppose that the two aerodynamic parameters, L_p and L_{δ_a}, in the roll dynamics (7.1) are constant but otherwise completely unknown, with the exception that we do know the sign of the aileron control effectiveness L_{δ_a} (it is positive for a conventional aircraft). The control task remains the same as in Example 7.1 – we need to find δ_a such that p tracks p_{ref}, which in turn is driven by a bounded possibly time-varying command p_{cmd}.

The main control challenge here is to achieve the desired closed-loop tracking performance, specified by the reference model (7.2) while operating in the presence of constant parametric uncertainties L_p and L_{δ_a}.

In the forthcoming chapters, we will exploit Lyapunov-based methods that allow us to design adaptive controllers with formal guarantees of closed-loop stability, boundedness, and tracking performance. In the meantime, we shall outline main ideas in the design of adaptive systems.

If we knew the roll dynamics model parameters, then a feedback-feedforward controller in the form similar to (7.3)

$$\delta_a = k_p\, p + k_{p_{cmd}}\, p_{cmd} \tag{7.12}$$

would have solved the tracking problem. Since the system parameters are unknown, the ideal controller gains, k_p and $k_{p_{cmd}}$, cannot be computed directly as in Example 7.1. Instead, we consider an adaptive controller in the form

$$\delta_a = \hat{k}_p\, p + \hat{k}_{p_{cmd}}\, p_{cmd} \tag{7.13}$$

where $\left(\hat{k}_p,\ \hat{k}_{p_{cmd}}\right)$ represent the estimated feedback and feedforward gains, in that order. Substituting (7.13) into (7.1) gives the closed-loop system.

$$\dot{p} = \left(L_p + L_{\delta_a}\, \hat{k}_p\right) p + \left(L_{\delta_a}\, \hat{k}_{p_{cmd}}\right) p_{cmd} \tag{7.14}$$

Using parameterization (7.3), the reference model dynamics (7.2) can be equivalently written in terms of the ideal unknown gains as

$$\dot{p}_{\text{ref}} = \underbrace{\left(L_p + L_{\delta_a}\, k_p\right)}_{a_{ref}}\, p_{ref} + \underbrace{\left(L_{\delta_a}\, k_{p_{cmd}}\right)}_{b_{ref}}\, p_{cmd} \tag{7.15}$$

We now define the gain estimation errors,

$$\Delta k_p = \hat{k}_p - k_p, \quad \Delta k_{p_{cmd}} = \hat{k}_{p_{cmd}} - k_{p_{cmd}} \tag{7.16}$$

and rewrite the closed-loop system (7.14) in the following form:

$$\dot{p} = \underbrace{\left(L_p + L_{\delta_a}\, k_p\right)}_{a_{ref}}\, p + \underbrace{\left(L_{\delta_a}\, k_{p_{cmd}}\right)}_{b_{ref}}\, p_{cmd} + L_{\delta_a}\left(\Delta k_p\, p + \Delta k_{p_{cmd}}\, p_{cmd}\right) \tag{7.17}$$

Subtracting (7.15) from (7.17) gives the tracking error dynamics.

$$\dot{e} = a_{ref}\, e + L_{\delta_a}\left(\Delta k_p\, p + \Delta k_{p_{cmd}}\, p_{cmd}\right) \tag{7.18}$$

There are three error signals in the error dynamics (7.18): (1) the roll rate tracking error e, (2) the feedback gain estimation error Δk_p, and (3) the feedforward gain estimation error $\Delta k_{p_{cmd}}$. We are going to devise adaptive laws for changing the gains $\left(\hat{k}_p,\ \hat{k}_{p_{\text{cmd}}}\right)$, such that all these three errors tend to zero, globally and asymptotically.

In order to do that, we first define a scalar function V, representative of the total "kinetic energy" of all the errors in the system.

$$V\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) = \frac{e^2}{2} + \frac{|L_{\delta_a}|}{2\,\gamma_p}\,\Delta k_p^2 + \frac{|L_{\delta_a}|}{2\,\gamma_{p_{cmd}}}\,\Delta k_{p_{cmd}}^2 \tag{7.19}$$

The "energy" function represents a weighted sum of squares of all the errors in the system. This is the so-called Lyapunov function candidate, and the positive constant scalar weights $\left(\gamma_p,\ \gamma_{p_{cmd}}\right)$ will eventually become the rates of adaptation. We can easily evaluate the time derivative of V.

$$\dot{V}\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) = e\,\dot{e} + \frac{|L_{\delta_a}|}{\gamma_p}\,\Delta k_p\,\dot{\hat{k}}_p + \frac{|L_{\delta_a}|}{\gamma_{p_{cmd}}}\,\Delta k_{p_{cmd}}\,\dot{\hat{k}}_{p_{cmd}} \tag{7.20}$$

This is the system "power." Substituting (7.18) into (7.20) yields the time derivative of V, along the trajectories of the error dynamics (7.18) but without explicit knowledge of these trajectories.

$$\begin{aligned}\dot{V}\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) &= a_{ref}\,e^2 \\ &+ e\,L_{\delta_a}\left(\Delta k_p\,p + \Delta k_{p_{cmd}}\,p_{cmd}\right) + \frac{|L_{\delta_a}|}{\gamma_p}\,\Delta k_p\,\dot{\hat{k}}_p + \frac{|L_{\delta_a}|}{\gamma_{p_{cmd}}}\,\Delta k_{p_{cmd}}\,\dot{\hat{k}}_{p_{cmd}}\end{aligned} \tag{7.21}$$

Rearranging terms, we further get

$$\begin{aligned}\dot{V}\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) &= a_{ref}\,e^2 \\ &+ \Delta k_p\,|L_{\delta_a}|\left(\text{sgn}(L_{\delta_a})\,p\,e + \frac{\dot{\hat{k}}_p}{\gamma_p}\right) + \Delta k_{p_{cmd}}\,|L_{\delta_a}|\left(\text{sgn}(L_{\delta_a})\,p_{cmd}\,e + \frac{\dot{\hat{k}}_{p_{cmd}}}{\gamma_{p_{cmd}}}\right)\end{aligned} \tag{7.22}$$

We want the energy function V to dissipate in time. It is then sufficient to require that its derivative $\dot{V}$ (the system power) be nonpositive, when evaluated along the system trajectories. The nonpositivity of $\dot{V}$ can be easily achieved if we select the following adaptive laws:

$$\begin{aligned}\dot{\hat{k}}_p &= -\gamma_p\,p\ e\ \text{sgn}(L_{\delta_a}) \\ \dot{\hat{k}}_{p_{cmd}} &= -\gamma_{p_{cmd}}\,p_{cmd}\ e\ \text{sgn}(L_{\delta_a})\end{aligned} \tag{7.23}$$

or, equivalently,

$$\begin{aligned}\dot{\hat{k}}_p &= -\gamma_p\, p\; e \\ \dot{\hat{k}}_{p_{cmd}} &= -\gamma_{p_{cmd}}\, p_{cmd}\; e\end{aligned} \tag{7.24}$$

thus making the second and the third terms in (7.22) disappear. Then,

$$\dot{V}\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) = a_{ref}\, e^2 \leq 0 \tag{7.25}$$

and consequently, the system kinetic energy V is a nonincreasing function of time. This fact immediately implies that all the signals in the error dynamics (7.18), such as $\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right)$, are bounded functions of time. Furthermore, since the ideal gains $\left(k_p,\ k_{p_{cmd}}\right)$ are constant, the adaptive gains $\left(\hat{k}_p,\ \hat{k}_{p_{cmd}}\right)$ are also bounded.

The stable (by design) reference model (7.2), when driven by a bounded command p_{cmd}, gives a bounded output p_{ref}. Also, e was proven to be bounded. Then, the roll rate p is bounded. Consequently, the control input δ_a in (7.13) and the roll acceleration $\dot{p}$ in the system dynamics (7.1) are bounded. Furthermore, since $\dot{p}_{ref}$ is bounded, then $\dot{e}$ is bounded, and so

$$\ddot{V}\left(e,\ \Delta k_p,\ \Delta k_{p_{cmd}}\right) = 2\, a_{ref}\, e\, \dot{e} \tag{7.26}$$

is a uniformly bounded function of time. The latter implies that $\dot{V}$ is a uniformly continuous function of time.

By definition (7.19), $V \geq 0$ and because of (7.25), V is a nonincreasing function of time. Therefore, V tends to a limit as $t \to \infty$, where the function limiting value may not necessarily be zero.

We have shown that $0 \leq \lim_{t\to\infty} V\left(e(t),\ \Delta k_p(t),\ \Delta k_{p_{cmd}}(t)\right) < \infty$ and $\dot{V}$ are uniformly continuous. According to Barbalat's lemma (see Chap. 8), these two facts imply that the system power $\dot{V}$ in (7.25) asymptotically tends to zero, which in turn means

$$\lim_{t\to\infty} e(t) = 0 \tag{7.27}$$

Thus, the adaptive controller (7.13), along with the adaptive laws (7.24), forces p track its reference signal p_{ref} asymptotically and for any initial conditions (globally). At the same time, all signals in the corresponding closed-loop system remain uniformly bounded. These arguments prove closed-loop stability and tracking performance of the closed-loop system with the adaptive controller. The corresponding block diagram is shown in Fig. 7.4.

As seen from the figure, the closed-loop system is comprised of the original roll dynamics (7.1) operating under the adaptive controller (7.13), with the reference model dynamics (7.2), and using the adaptive laws (7.24). Here, the external input is the roll rate command p_{cmd}.

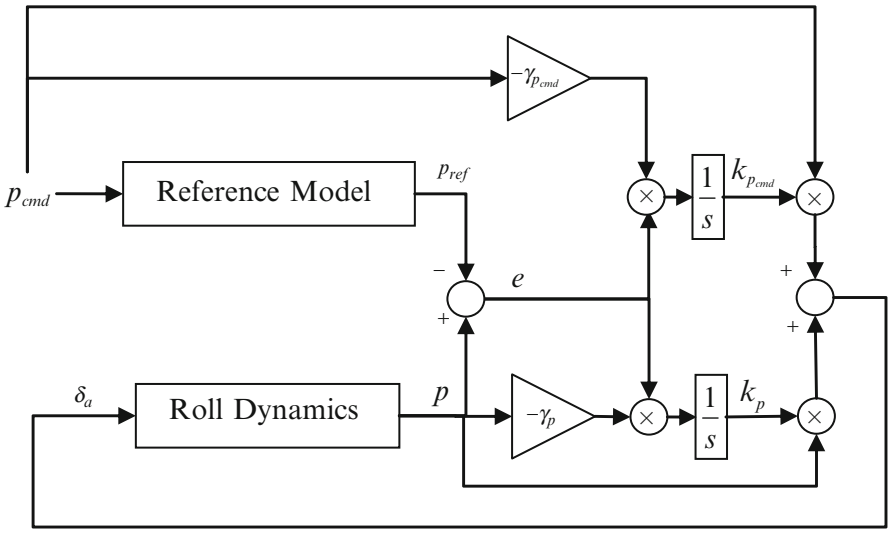


Fig. 7.4 Model reference adaptive controller obtained in Example 7.2

$$\begin{aligned}
\dot{p} &= \left(L_p + L_{\delta_a}\hat{k}_p\right)p + L_{\delta_a}\hat{k}_{p_{cmd}}\,p_{cmd} \\
\dot{p}_{\text{ref}} &= a_{ref}\,p_{ref} + b_{ref}\,p_{cmd} \\
\dot{\hat{k}}_p &= -\gamma_p\,p\,\left(p - p_{ref}\right) \\
\dot{\hat{k}}_{p_{cmd}} &= -\gamma_{p_{cmd}}\,p_{cmd}\,\left(p - p_{ref}\right)
\end{aligned} \tag{7.28}$$

Equivalently, this system can be written in terms of the tracking and parameter estimation errors.

$$\begin{aligned}
\dot{e} &= \left(a_{ref} + L_{\delta_a}\Delta k_p\right)e + L_{\delta_a}\left(\Delta k_p\,p_{ref} + \Delta k_{p_{cmd}}\,p_{cmd}\right) \\
\frac{d}{dt}\left(\Delta k_p\right) &= -\gamma_p\left(e + p_{ref}\right)\,e \\
\frac{d}{dt}\left(\Delta k_{cmd}\right) &= -\gamma_{p_{cmd}}\,p_{cmd}\,e
\end{aligned} \tag{7.29}$$

If instead of command tracking, the state regulation is of interest, then $p_{ref} = p_{cmd} = 0$, and so $\hat{k}_{p_{cmd}} = k_{p_{cmd}} = 0$. In this case, the closed-loop systems (7.28) and (7.29) simplify to the following time-invariant second-order inherently nonlinear dynamics,

$$\begin{aligned}
\dot{p} &= \left(L_p + L_{\delta_a}\hat{k}_p\right)p \\
\dot{\hat{k}}_p &= -\gamma_p\,p^2
\end{aligned} \tag{7.30}$$

These relations reveal the essential mechanism of adaptive control. The time-varying adaptive feedback gain $\hat{k}_p(t)$ will monotonically decrease its value until $\left(L_p + L_{\delta_a}\hat{k}_p\right)$ becomes negative, and as a result, the roll rate $p(t)$ will asymptotically converge to zero. In (7.30), the constant $\gamma_p > 0$ defines the rate of adaptation in the sense that large values of γ_p will force the adaptive gain $\hat{k}_p(t)$ to decrease faster.

In summary, using energy-based arguments, we have shown that the adaptive controller (7.12) and (7.24) provides the desired model reference-based closed-loop tracking performance for the system (7.1) while operating in the presence of the parametric uncertainties $\left(L_p,\ L_{\delta_a}\right)$. □

7.3 Direct Model Reference Adaptive Control of Scalar Linear Systems with Parametric Uncertainties

Let us now generalize and summarize the results obtained in Example 7.2 while restating them for a generic class of scalar linear-time-invariant uncertain systems in the form

$$\dot{x} = a\,x + b\,u \tag{7.31}$$

where $x \in R$ is the systems state, $u \in R$ is the control input, and $(a,\ b)$ represent the parametric uncertainties, (constant and unknown), with the known sgn$b$.

First, we choose the desired reference model,

$$\dot{x}_{ref} = a_{ref}\,x_{ref} + b_{ref}\,r \tag{7.32}$$

with $a_{ref} < 0$. This model is driven by any bounded, possibly time-varying, reference command r. The model parameters $\left(a_{ref},\ b_{ref}\right)$ must be chosen such that x_{ref} tracks r, with the designer specified criteria. For example, one might set $b_{ref} = -a_{\text{ref}}$ in order to enforce the unity DC gain from r to x_{ref}. Also, the value of $\left|a_{ref}\right|$ can be chosen such that the desired inverse time constant of the reference model is achieved.

Second, we define the model reference adaptive controller as a linear combination of feedback and feedforward terms,

$$u = \hat{k}_x\,x + \hat{k}_r\,r \tag{7.33}$$

where $\left(\hat{k}_x,\ \hat{k}_r\right)$ are the two adaptive gains, whose adaptive law dynamics are constructed similar to (7.24).

$$\begin{aligned}\dot{\hat{k}}_x &= -\gamma_x\,x\,\left(x - x_{ref}\right)\ \text{sgn}(b)\\ \dot{\hat{k}}_r &= -\gamma_r\,r\,\left(x - x_{ref}\right)\ \text{sgn}(b)\end{aligned} \tag{7.34}$$

In (7.34), positive scalars $(\gamma_x,\ \gamma_r)$ are called the rates of adaptation. The larger their values, the faster the system will adapt to the parametric uncertainties.

This particular controller is called "direct" to indicate that the controller gains are adapted in (7.34) directly in order to enforce the desired closed-loop tracking performance. Alternatively, indirect adaptive controllers can be designed to estimate the unknown plant parameters $(a,\ b)$ online and then use their estimated values to calculate controller gains.

Finally, using energy-based arguments, we can formally prove that the adaptive controller (7.33) and (7.34) provides the desired closed-loop tracking performance, in the sense that the system state x globally asymptotically tracks the state x_{ref} of the reference model (7.32) while keeping all signals in the corresponding closed-loop dynamics uniformly bounded in time.

A few immediate remarks are in order:

- The direct model reference adaptive controller (7.33) and (7.34) operates using only available (online measured) signals in the system. The latter consists of: (a) the system state x, (b) the state of the reference model x_{ref}, (c) the tracking error $e = x - x_{ref}$, and (d) the sign of the control effectiveness $\text{sgn}b$.
- All signals in the closed-loop system remain uniformly bounded in time.
- The system state x tracks the state of the reference model x_{ref}, globally and asymptotically. However, a characterization of the system transient dynamics in model reference adaptive control remains an open problem.
- The adaptive parameters $\left(\hat{k}_x,\ \hat{k}_r\right)$ are not guaranteed to converge to their true unknown values $(k_x,\ k_r)$ nor are they assured to converge to constant values in any way. All that is known is that these parameters remain uniformly bounded in time. Sufficient conditions for parameter convergence are known as persistency of excitation [1, 2]. It turns out that for a first-order linear system such as (7.1), persistent excitation is guaranteed if the commanded signal $r(t)$ contains at least one sinusoidal component. In this case, the two adaptive gains $\left(\hat{k}_x,\ \hat{k}_r\right)$ will converge to their true constant unknown values, exponentially fast.

7.4 Historical Roots and Foundations of Model Reference Adaptive Control

The adaptive control development was largely motivated in the early 1950s by the design of autopilots for aircraft that operated in a wide flight envelope, with a large range of speeds and altitudes. Different flight conditions caused the aircraft dynamics to change significantly. This phenomenon called for flight controllers that could accommodate drastic changes in the aircraft aerodynamic and propulsive forces and moments. Adaptive control was proposed as one of the design approaches to solving the flight control problem.

The concept of a model-reference adaptive system (MRAS) was originally proposed in 1958 by Whitaker et al. at MIT [3, 4]. The main idea behind this

concept was to specify the desired command-to-output performance of a servo-tracking system using a differential or a difference equation (the reference model) that would define the ideal response of the system due to external commands. This control concept was later called "explicit model following," and the corresponding architecture became known as the model reference adaptive control (MRAC).

Shortly after its introduction, the first proof of MRAC closed-loop stability using Lyapunov theory was given in 1965 by Butchart and Shackcloth, at the IFAC Symposium on Adaptive Control [5], and in 1966 by Parks [6].

In the following years, adaptive control theory for a broad class of multi-input multi-output uncertain dynamical systems was extensively developed and well documented in several now-classical textbooks [1, 2, 7, 8].

7.5 Exercises

Exercise 7.1. Consider the aircraft roll dynamics from Example 7.1. Given the roll damping $L_p = -0.8$ (s^{-1}) and the aileron effectiveness $L_{\delta_a} = 1.6$ (s^{-1}), design a fixed-gain model reference controller in the form of (7.3) to recover the reference model dynamics (7.2), *with* $a_{ref} = -2, b_{ref} = 2$. Also, design a fixed-gain controller with error feedback in the form of (7.9). Choose several bounded time-varying roll rate commands. Simulate the closed-loop system response, with each of the two controllers active (one at a time). Compare the two controllers and comment on the achieved closed-loop system stability, robustness, tracking, and transient properties.

Exercise 7.2. Derive relations (7.28), (7.29), and (7.30).

Exercise 7.3. Assume that the constant roll dynamics data $\left(L_p, L_{\delta_a}\right)$ from Example 7.1 are unknown and that only the sign of L_{δ_a} is known to be positive. Using the same reference model parameters, design an adaptive roll rate tracking controller in the form of (7.13), (7.14), (7.15), (7.16), (7.17), (7.18), (7.19), (7.20), (7.21), (7.22), and (7.23). Choose various roll rate commands and simulate the resultant closed-loop system performance. Compare fixed-gain versus adaptive controller performances and comment on your results.

Exercise 7.4. Consider a scalar dynamical system described by the first-order differential equation

$$\dot{x} = a\,x + b\,u, \quad x(0) = x_0$$

where $a = 2$ and $b = 3$ represent unknown constant parameters. It is assumed that $\text{sgn}b = 1$ is known. The goal is to design a controller such that the system state tracks the state of the reference model,

$$\dot{x}_{ref} = r(t) - x_{ref}$$

where $r = r(t)$ is the commanded reference input (a bounded signal). Assuming that the system dynamics are known, design a fixed-gain command tracking controller. Then, design a direct model reference adaptive controller. Simulate the closed-loop system dynamics for both controllers, starting from different initial conditions and using three different reference commands: (a) a step-input, (b) a series of steps, and (c) a sum of sinusoids. Tune your adaptive design (i.e., select rates of adaptation). Compare tracking performance of the two closed-loop systems and their corresponding control signals. Comment on your results.

References

1. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Control. Dover, New York (2005)
2. Ioannou, P., Fidan, P.: Adaptive Control Tutorial, SIAM, Advances in Design and control, SIAM, PA (2006)
3. Whitaker, H.P., Yamron, J., Kezer, A.: Design of Model Reference Adaptive Control Systems for Aircraft, Report R-164, Instrumentation Laboratory, MIT, Cambridge, MA (1958)
4. Whitaker, H.P., Osburn, P.V., Kezer A.: New Developments in the Design of Model Reference Adaptive Control Systems, Paper 61–39, Institute of the Aerospace Sciences, Easton, PA (1961)
5. Butchart, R.L., Shackcloth, B.: Synthesis of model reference adaptive control systems by Lyapunov's second method, In: Proceedings of the 1965 IFAC Symposium on Adaptive Control, Teddington, UK (1965)
6. Parks, P.D.: Lyapunov redesign of model reference adaptive systems. IEEE Trans. Automat. Contr. **11**, 362–367 (1966)
7. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, New Jersey 07632 (1991)
8. Krstic, M., Kanellakopoulos, I., Kokotovic, P.: Nonlinear and Adaptive Control Design. Wiley, New York (1995)

Chapter 8
Lyapunov Stability of Motion

8.1 Dynamical Systems

A dynamical system may be thought of as a collection of finite or infinite number of interconnected and time-dependent components. The system evolution is driven by an environment where the system operates. When subjected to an external time-dependent input $u(t)$, the system generates an output $y(t)$, which in turn may explicitly depend on the system internal properties, defined by the system states $x(t)$. The states describe the system inner-component connections, their dynamical response due to environmental stimulus, and their contributions to the system response.

In what follows, we consider a special class of dynamical systems that can be modeled by a finite number of coupled scalar ordinary differential equations in the form

$$\dot{x} = f(t, x, u) \tag{8.1}$$

In (8.1), $t \in R^+$ denotes time and $f: R \times R^n \times R^m \to R^n$ is a vector function. We call (8.1) the system dynamics, refer to $x \in R^n$ as the system state at time t, and define $u \in R^m$ as the control input (an externally supplied signal). The number of the state components n is called the order of the system.

A solution $x(t)$ of (8.1) (if one exists) corresponds to a curve in the system state space R^n, as t varies from an initial time t_0 to infinity. This curve is often referred to as the system state trajectory. Later in this chapter, we will formulate sufficient conditions guaranteeing existence and uniqueness of solutions for dynamical systems such as (8.1), starting from a given set of initial conditions $x(t_0) = x_0$.

In addition to the system dynamics (8.1), a set of algebraic equations may also be given,

$$y = h(t, x, u) \tag{8.2}$$

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_8,

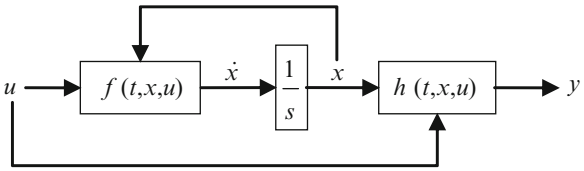


Fig. 8.1 State-space model block diagram

where $h : R \times R^n \times R^m \rightarrow R^p$ and $y \in R^p$. This is the system output. Together, Eqs. (8.1) and (8.2) form the system state space model, whose block diagram is shown in Fig 8.1.
A special case of (8.1), and (8.2) is the linear-in-control system,

$$\begin{aligned} \dot{x} &= f(t,\, x) + g(t,\, x)\, u \\ y &= h(t,\, x) + d(t,\, x)\, u \end{aligned} \tag{8.3}$$

where the functions g and d are of matching dimensions.

Letting $x = (\, x_1 \quad x_2 \quad \ldots \quad x_n \,)^T$, a particular class of nonlinear continuous-time dynamics is given by the systems in Brunovsky canonical form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\cdots\cdots\cdots \\ \dot{x}_n &= f(x) + g(x)\, u \\ y &= h(x) \end{aligned} \tag{8.4}$$

For linear time-variant (LTV) systems, the state space model (8.1), and (8.2) is

$$\begin{aligned} \dot{x} &= A(t)\, x + B(t)\, u \\ y &= C(t)\, x + D(t)\, u \end{aligned} \tag{8.5}$$

Finally, the class of linear time-invariant (LTI) systems is written in the familiar form

$$\begin{aligned} \dot{x} &= A\, x + B\, u \\ y &= C\, x + D\, u \end{aligned} \tag{8.6}$$

whose dynamic properties can be completely characterized by the matrix quadruple $(A,\ B,\ C,\ D)$.

If the model (8.1) does not contain the control input signal u,

$$\dot{x} = f(t,\, x) \tag{8.7}$$

then the resulting dynamics are called "unforced." If in addition, the function f does not depend explicitly on t, that is if

$$\dot{x} = f(x) \tag{8.8}$$

then the system unforced dynamics are called autonomous or time invariant. Systems that explicitly depend on time are nonautonomous (i.e., time variant).

8.2 Existence and Uniqueness of Solutions

Suppose that we initialize the state of the system (8.7),

$$x(t_0) = x_0 \in R^n \tag{8.9}$$

at a time instant $t_0 \geq 0$. Together, (8.7), (8.8), and (8.9) define the Cauchy problem, or equivalently, the initial value problem (IVP), whose solutions may or may not exist. Moreover, when a solution does exist, it may or may not be unique.

Besides theoretical demands, the question of existence and uniqueness become quite important for practitioners in simulation, dynamics, and control. For example, if the system (8.7) is constructed to emulate a real process that starts from an initial condition x_0, we need to know if and when the system unique solution would exist. Otherwise, the resulting simulation data may lead us to erroneous conclusions about the underlying process dynamics and control.

Contrary to LTI systems (8.6), existence and uniqueness of solutions for nonlinear equations (8.7) are not always guaranteed. In order to motivate our discussion, we consider several examples.

Example 8.1 The scalar nonlinear dynamics

$$\dot{x} = -\operatorname{sgn} x$$

has the discontinuous (at the origin) right-half side, which is defined by the sign function

$$\operatorname{sgn} x = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

The system phase portrait is easy to draw and is given in Fig. 8.2.

These data indicate that the system trajectories asymptotically approach either 1 or -1, depending on whether the initial conditions are negative or positive, respectively. The rate of change of the "kinetic energy" for this system is nonpositive:

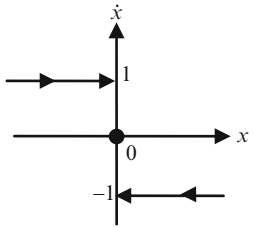


Fig. 8.2 Phase portrait of the system dynamics in Example 8.1

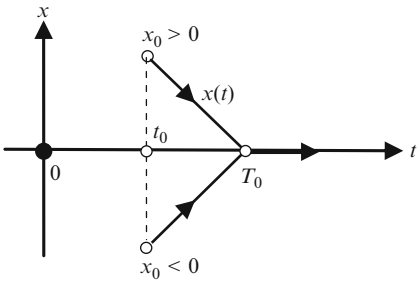


Fig. 8.3 System trajectories in Example 8.1

$$\frac{d}{dt}\left(\frac{x^2(t)}{2}\right) = x(t)\,\dot{x}(t) = -x(t)\ \operatorname{sgn} x(t) = -|x(t)| \leq 0$$

Therefore, the kinetic energy must dissipate, and so it seems that the system trajectories should asymptotically approach the origin, where $\operatorname{sgn} 0 = 0$ by the definition. To further investigate the system behavior, we can integrate the system dynamics on the interval from t_0 to t:

$$x(t) = x_0 - (t - t_0)\ \operatorname{sgn} x(t)$$

Even though this equation is implicit in x, we can easily sketch its solutions versus time (Fig. 8.3).

First, we note that every solution arrives at zero in finite time $T_0 = t_0 + |x_0|$, and it remains zero for all future times. Second, for every solution with $x_0 > 0$ there is the solution that starts at $(-x_0)$, and it meets the former solution at the same exact time T_0. Third, the system trajectories are not continuously differentiable at T_0. In fact, for all $t \geq T_0$ and as the system trajectory evolves along the t-axis, the system solutions will "jitter." This interesting phenomenon is solely caused by the discontinuity of the system dynamics at the origin. □

Example 8.2 Let $k > 0$ be a real number and consider the IVP:

$$\dot{x} = x^k, \quad x(t_0) = x_0$$

Using separation of variables, we can write the solution of this system,

$$x^{1-k}(t) = x_0^{1-k} + (1-k)(t-t_0)$$

and make several observations:

- For $0 < k < 2$, the system does not have solutions that start at $x_0 < 0$.
- Suppose $k = 2$, $x_0 = 1$, and $t_0 = 0$. Then, $x(t) = \frac{1}{1-t}$. This solution grows unbounded "blows up" in finite time $T = 1$, and it is not defined for $t \geq T$.
- For $k = \frac{2}{3}$ and $x_0 = t_0 = 0$, the IVP has not one but two solutions: $x(t) = \frac{t^3}{27}$ and $x(t) \equiv 0$. Let $T > 0$ denote a constant. As it turns out, this IVP has infinitely many solutions: □

$$x(t) = \begin{cases} 0 & , \quad 0 \leq t \leq T \\ \dfrac{1}{27}(t-T)^3 & , \quad t > T \end{cases} .$$

All of the above examples imply that both existence and uniqueness of IVP solutions for the dynamical system (8.7) depend on certain properties of the vector function $f(t, x)$.

We begin with a theorem that states sufficient conditions for the IVP problem to admit a solution which may not necessarily be unique [1].

Theorem 8.1. Peano *If $f(t, x)$: $R \times R^n \to R^n$ is continuous in a closed region,*

$$B = \{(t, x): |t - t_0| \leq T, \ \|x - x_0\| \leq r\} \subseteq R \times R^n \tag{8.10}$$

where T, r are some strictly positive constants, and $\|\cdot\|$ is the Euclidean vector norm (see (8.12)), then there exists $t_0 < t_1 \leq T$ such that the IVP (8.7), (8.8), and (8.9) has at least one continuously differentiable solution $x(t)$ on the interval $[t_0, \ T]$. ■

The assumed continuity of $f(t, x)$ in its arguments ensures that there is at least one solution of the IVP. Note however that this theorem does not guarantee the uniqueness of the solution. The key constraint that yields uniqueness is the so-called Lipschitz condition, whereby $f(t, x)$ satisfies the inequality

$$\|f(t, x) - f(t, y)\| \leq L \|x - y\| \tag{8.11}$$

for all (t, x) and (t, y) in some neighborhood of (t_0, x_0), with a finite constant $L > 0$. In (8.11) and everywhere else throughout the book, $\|x\|$ denotes the Euclidean vector norm of $x \in R^n$:

$$\|x\| = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty \\ \max\limits_{1 \leq i \leq n} |x_i| \quad , & p = \infty \end{cases} . \tag{8.12}$$

Lipschitz-based sufficient conditions for the unique existence of IVP solutions are stated below and without proof [2].

Theorem 8.2. Local Existence and Uniqueness *Let* $f(t, x)$: $R \times R^n \to R^n$ *be piece-wise continuous in t and satisfy the Lipschitz condition (8.11):*

$$\forall x, y \in B = \{x \in R^n : \|x - x_0\| \le r\}, \ \forall t \in [t_0, t_1] \tag{8.13}$$

Then, there exists some $\delta > 0$ *such that the IVP for the state equation* $\dot{x} = f(t, x)$ *with* $x(t_0) = x_0$ *has a unique solution over* $[t_0, t_0 + \delta]$. ■

Notice that the Lipschitz condition (8.11) is assumed to be valid locally in a neighborhood of (t_0, x_0) from a compact (closed and bounded) set B, as it is defined in (8.13).

We can try to extend the interval of existence and uniqueness over a given time interval $[t_0, t_0 + \delta]$ by taking $t_0 \triangleq t_0 + \delta$ as the new initial time and $x_0 \triangleq x(t_0 + \delta)$ as the new initial state. If the conditions of the theorem are satisfied at $(t_0 + \delta, \ x(t_0 + \delta))$, then there exists $\delta_2 > 0$ such that the IVP has a unique solution over $[t_0 + \delta, \ t_0 + \delta + \delta_2]$ that passes through the point $(t_0 + \delta, \ x(t_0 + \delta))$. We can now piece together the two solutions to establish the existence of a unique solution over the larger interval $[t_0, \ t_0 + \delta + \delta_2]$. This idea can be repeated to keep extending the IVP solution, arriving at the maximal IVP solution, which is defined on the maximal interval $[t_0, \ t_0 + \delta_{\max}]$, with finite or infinite $\delta_{\max}$. It is interesting to note that if $\delta_{\max}$ is finite, then the respective maximal solution tends to infinity [3], as the following example demonstrates.

Example 8.3 The unique solution of the scalar IVP,

$$\dot{x} = 1 + x^2, \ x(0) = 0$$

is $x(t) = \tan t$. Its maximal interval of existence is finite with $\delta_{\max} = \frac{\pi}{2}$, and, predictably, $\lim_{t \to \frac{\pi}{2}} x(t) \to \infty$; that is, this solution becomes unbounded in finite time. □

In process modeling applications, we are primarily interested in constructing IVP-s whose solutions are unique and exist for all $t \ge t_0$. The global uniqueness and existence requirements would ensure at least soundness of our models but not necessarily their validity. The latter would have to be verified by correlating model data with the application process under consideration.

The next theorem states that if the system dynamics function f satisfies global Lipschitz conditions, then the corresponding IVP has a unique solution over arbitrarily large time interval [2].

Theorem 8.3. Global Existence and Uniqueness. *Suppose that a vector function* $f(t, x)$: $R \times R^n \to R^n$ *is piece-wise continuous in t and globally Lipschitz in x,*

$$\|f(t, x) - f(t, y)\| \le L \|x - y\|, \quad \forall x, y \in R^n, \ \forall t \in [t_0, \ t_1] \tag{8.14}$$

with a finite constant $L > 0$. Then, the IVP (8.7), (8.8), and (8.9) has a unique solution over $[t_0,\ t_1]$, where the final time t_1 may be arbitrarily large. ■

We immediately note that the above stated global Lipschitz condition (8.14) is sufficient but not necessary as the next example shows.

Example 8.4 The system dynamics function in the scalar IVP

$$\dot{x} = -x^3, \quad x(0) = x_0$$

is not globally Lipschitz, yet the system has the unique solution

$$x(t) = \frac{x_0}{\sqrt{2\,x_0^2\,t + 1}}$$

which is defined for any initial condition x_0 globally and for all time $t \geq 0$. □

The next theorem is of particular interest to us. It presents sufficient conditions for extending IVP solutions indefinitely. Its detailed proof can be found in [2].

Theorem 8.4. Global Existence and Uniqueness on Unbounded Time Interval *Let $f(t,\ x)$: $R \times R^n \to R^n$ be piece-wise continuous in t, locally Lipschitz in x for all $t \geq 0$ and all x in a domain $D \subset R^n$. Let $W \subset D$ be a compact subset of D, $x_0 \in W$, and suppose it is known that every solution of the IVP (8.7), (8.8), and (8.9) lies entirely in W. Then, there is a unique solution that is defined for all $t \geq t_0$.* ■

In the forthcoming chapters, we will use Lyapunov's methods to check if system trajectories evolve inside a compact set. We will be able to do that without solving the system differential equation. Lyapunov's analysis methods generalize and extend the notion of energy, from mechanical systems to generic dynamics. For adaptive systems, we will show that suitable energy functions can be formed as sum of squares of the system state components. Then, we would compute the system power – the time derivative of the energy function, evaluated along the system trajectories. We will argue that if the system power is nonpositive, that is if the system energy dissipates, then every trajectory is bounded and exists globally for all time.

For now, let us illustrate the energy-based analysis using the dynamics from *Example 8.4*. Toward that end, we shall utilize the system "kinetic" energy,

$$V(x) = \frac{x^2}{2}$$

and compute its time derivative along the system dynamics (the system power):

$$\frac{d}{dt}\left(\frac{x^2}{2}\right) = x\,\dot{x} = x\left(-x^3\right) = -x^4 \leq 0$$

Since the power function is nonpositive, then the energy must decrease and consequently, the system state must be bounded for all time. Therefore and according

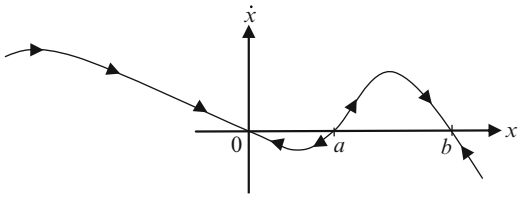


Fig. 8.4 Phase portrait of the system from Example 8.5

to Theorem 8.4, the system dynamics must have a unique solution starting from any initial condition at $t_0 = 0$ and extending indefinitely, for all $t \geq 0$.

Energy-based methods and Theorem 8.4 become extremely useful especially when the system dynamics cannot be integrated to obtain its IVP solutions explicitly.

Example 8.5 Consider the autonomous scalar dynamics $\dot{x} = f(x)$, whose phase portrait is shown in Fig. 8.4.

We assume that $f(0) = f(a) = f(b) = 0, f(x) > 0$ for all $x < 0$, and that the function is locally Lipschitz in x. Other than that, the function shape and its values are assumed to be completely unknown, and as such, these dynamics cannot be analytically integrated to compute the system solutions in their explicit form.

We now pose several questions and give their answers to demonstrate that explicit knowledge of IVP solutions is not required at all in order to assess if the system unique solutions exist. In addition, we will also assess their interval of existence.

Question: Will the IVP with a nonzero initial condition $x(0) \neq 0$ have a unique solution?

Answer: Since f is locally Lipschitz, then existence and uniqueness of the IVP solutions directly follows from Theorem 8.2.

Question: Is this solution defined for all time?

Answer: Anchored in Theorem 8.4, we can either employ energy-based arguments to show that all trajectories are bounded, or we can simply examine the system phase portrait shown in Fig. 8.4. Analyzing the latter, it becomes clear that starting from any nonzero initial condition, all trajectories of this system will enter the interval $[0,\ b]$ in finite time. Therefore, all these solutions are bounded, and because of Theorem 8.4, the IVP unique solutions are defined globally, for all $t \geq 0$. □

We have surveyed and discussed several well-known theorems concerning existence and uniqueness of IVP solutions for nonautonomous continuous dynamical systems. Basically, existence of IVP solutions is provided if the system dynamics are continuous in its arguments. However, in order to guarantee uniqueness, we have called on the Lipschitz assumption (local or global). As it turns out, the Lipschitz condition, even when local, is quite restrictive since the set of all Lipschitz-continuous functions represents a very small (called "meager") subset of all continuous functions. Such an observation might lead to a conjecture that only a very small set of IVP-s have unique solutions. Fortunately, this conjecture is incorrect. In 1932, the Polish mathematician Witold Orlicz proved that the set of all functions for which IVP-s have unique solutions is very large (a complement of a

meager set). Orlicz' theorem states that "almost all" differential equations with continuous right-hand sides have unique solutions. On the other hand, the set of IVP-s, for which we can formally characterize uniqueness of their solutions, is "almost nothing." This compelling argument suggests that there are very many classes of non-Lipschitz IVP-s with unique solutions that are yet to be discovered. Further details on the subject and the proof of Orlicz' theorem can be found in [3, Appendix A].

8.3 System Equilibrium

One of the central concepts in control and system theory is the concept of an equilibrium point. We will focus our discussions on nonautonomous unforced dynamical systems:

$$\dot{x} = f(t, x) \tag{8.15}$$

with the vector function $f : [0, \infty) \times D \to R^n$ which is piece-wise continuous in t and locally Lipschitz in x and with a domain $D \subset R^n$ that contains the origin $x = 0$.

Definition 8.1. *The origin in R^n is an equilibrium point for the unforced nonautonomous system (8.15) at $t_0 = 0$ if*

$$f(t, 0) = 0, \quad \forall t \geq 0 \tag{8.16}$$

It is not difficult to show that there is no loss of generality in using the origin and the zero initial time in the definition above. In fact, suppose we define a nonzero vector $x^* \in R^n$ to be an equilibrium point of (8.15) at a nonzero initial time $t = t_0$:

$$f(t, x^*) = 0, \quad \forall t \geq t_0$$

We can redefine time $\tau = t - t_0$, introduce the new state

$$z(\tau) = x(\tau + t_0) - x^*$$

and arrive at the transformed system dynamics

$$\boxed{\frac{d\, z(\tau)}{d\, \tau}} = \frac{d\, x(\tau + t_0)}{d\, t} = f(\tau + t_0,\, z(\tau) + x^*) = \boxed{g(\tau,\, z(\tau))}$$

with $g(0, 0) = f(t_0, x^*) = 0$. Thus, we have shifted the equilibrium point to the origin and the initial time to zero.

This idea can be further generalized. Suppose that we are given a trajectory $x^*(t)$ of (8.15) that starts at $t = t_0$:

$$\dot{x}^*(t) = f(t,\ x^*(t)), \quad t \geq t_0$$

We can again redefine time $\tau = t - t_0$, introduce the new state

$$z(\tau) = x(\tau + t_0) - x^*(\tau + t_0)$$

and rewrite the system dynamics

$$\boxed{\frac{d\,z(\tau)}{d\,\tau}} = \frac{d\,x(\tau + t_0)}{d\,t} - \frac{d\,x^*(\tau + t_0)}{d\,t}$$
$$= f(\tau + t_0,\, z(\tau) + x(\tau + t_0)) - f(\tau + t_0,\, z(\tau) + x^*(\tau + t_0)) = \boxed{g(\tau,\, z(\tau))}$$

with $g(0,\, 0) = 0$. Consequently, analyzing the redefined dynamics around the origin, as an equilibrium point, while starting at $t_0 = 0$, allows to determine the original system behavior around the original nonzero equilibrium x^*. This modification also allows us to assess the system relative dynamics with respect to any time-dependent trajectory $x^*(t)$, starting at an arbitrary initial time instant $t_0 \geqslant 0$.

A dynamical system can have multiple equilibrium points. Some of these equilibrium points might be isolated from each other, while others might form a continuum of equilibrium points. In either case, it is worth noting that whenever the system starts at an equilibrium point, it will remain there for all future times. The converse is also true and can be formally proven.

Example 8.6 The LTI system $\dot{x} = A\,x$ has an isolated equilibrium point at $x = 0$ if and only if $\det A \neq 0$ (A has no zero eigenvalues). Otherwise, the system has a continuum of equilibrium points. These are the only possible equilibrium patterns that a linear time-invariant system may have. □

Example 8.7 A nonlinear system can have multiple isolated equilibrium points. Consider the Bernoulli equation $\dot{x} = x\,(x - 1)$. It has two isolated equilibrium points, $x^* = 0$ and $x^* = 1$. The system phase portrait is shown in Fig. 8.5.

Clearly, all trajectories that start in the open interval $(-\infty,\ 1)$ will converge to the origin, while all other trajectories will diverge to $+\infty$. This phenomenon is typical for nonlinear dynamics, where, depending on the initial conditions, the system exhibits completely different behaviors. It is also clear that the system equilibrium at the origin is asymptotically stable (formal definition will be given later) in the sense that all trajectories that start in the open interval $(-\infty,\ 1)$ will converge back to the origin without leaving the interval. The other equilibrium $x^* = 1$ is unstable, meaning that there are trajectories that start arbitrarily close to 1, yet they move away from this equilibrium point. □

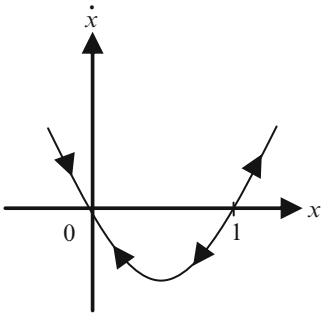


Fig. 8.5 Phase portrait of a Bernoulli equation from Example 8.7

8.4 Lyapunov Stability Definitions

The concept of Lyapunov stability is one of the most prominent and fundamental in dynamics and control. It is primarily concerned with analyzing behavior of system trajectories near equilibrium but without explicit computation of those solutions.

Theoretical foundations of what is known today as the Lyapunov stability theory are due to the Russian mathematician Alexander Mikhailovich Lyapunov (1857–1918). In 1892 at the University of Moscow, Lyapunov presented and subsequently defended his doctoral thesis "on the general problem of the stability of motion," where he had introduced basic definitions and fundamental theorems for studying the stability of solutions for a broad class of differential equations.

In 1908, Lyapunov's work was translated into French, reprinted by Princeton University Press in 1947, and gained wide acceptance in the West in the 1960s. Today, Lyapunov stability theory represents an indispensible tool that enables engineers and scientists analyze nonlinear systems and design controllers with stable and predictable performance.

System stability can be interpreted as a continuity of the system trajectories, with respect to initial conditions, over infinite time interval. The keywords here are "over infinite time interval." They highlight the difference between the notions of the stability and continuity on initial conditions. It is well-known that solutions of Lipschitz-continuous differential equations continuously depend on the system initial conditions [1–3]. However, the notion of stability requires that this continuity property holds infinitely in time.

Let $x(t;\ x_0)$ denote a solution of (8.15) with the initial condition $x(t_0) = x_0$. Suppose that this solution is unique and exists on a finite, possibly open-ended interval $[t_0,\ T)$. The continuity property of $x(t;\ x_0)$ due to changes in x_0 can be described as follows: Given any positive constant $\varepsilon > 0$, there must exist a sufficiently small positive constant $\delta > 0$, such that for all perturbed initial conditions $x_0 + \Delta x_0$ with $|\Delta x_0| \leq \delta$, the corresponding perturbed solution $x \times (t;\ x_0 + \Delta x_0)$ deviates from the original by no more than ε, that is, $\|x(t;\ x_0 + \Delta x_0) - x(t;\ x_0)\| \leq \varepsilon$, for all $t_0 \leq t < T$. Figure 8.6 illustrates the continuity property for a scalar system.

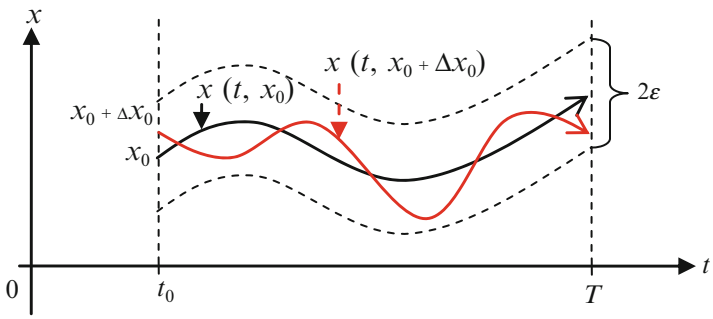


Fig. 8.6 Continuity of system solutions with respect to initial conditions

On the finite interval $[t_0,\ T)$, the perturbed trajectory $x(t;\ x_0 + \Delta x_0)$ will evolve within the $(2\,\varepsilon)$-strip relative to the original unperturbed solution $x(t;\ x_0)$, as long as the perturbed initial condition $(x_0 + \Delta x_0)$is located within the $(2\,\delta)$-strip of x_0.

In practice, we are often interested in analyzing system solutions that are defined on an infinite interval $[t_0,\ \infty)$. Will in this case the perturbed solution stay close to the original or will it deviate from the latter? A simple example demonstrates that both cases can occur.

Example 8.8 Starting at $t_0 = 0$ and from the initial condition $x_0 = \frac{1}{a}$, the linear time-invariant system

$$\dot{x} = a\,x - 1$$

has the steady-state solution $x\left(t;\ \frac{1}{a}\right) = \frac{1}{a}$. If the initial condition is perturbed by Δx_0, then the corresponding solution is

$$x\left(t;\ \frac{1}{a} + \Delta x_0\right) = \Delta x_0\, e^{a\,t} + \frac{1}{a}$$

Clearly, if $a<0$, then for any $\varepsilon>0$, $\left|x\left(t;\ \frac{1}{a} + \Delta x_0\right) - x\left(t;\ \frac{1}{a}\right)\right| = |\Delta x_0\, e^{a\,t}| \leq \varepsilon$, as long as $|\Delta x_0| \leq \delta = \varepsilon$, and this relation is valid for all $t \geq 0$. So, for any initial condition from the $(2\,\varepsilon)$-strip, the corresponding perturbed solution will remain within the same strip, which is centered around the steady-state solution $x(t) = \frac{1}{a}$. Note that in addition, the perturbed trajectory asymptotically approaches the original steady-state solution, as time tends to infinity. However, if $a > 0$, then no matter how small Δx_0 is the perturbed trajectory will become arbitrarily large in time, and as a result, it will deviate from the steady-state solution. □

A solution of (8.15) with the continuity property defined on an infinite interval is called stable. Otherwise, it is unstable.

Definition 8.2. Stability of Equilibrium in the Sense of Lyapunov *The equilibrium point* $x^* = 0$ *of the nonautonomous unforced dynamics (8.15) is stable if for*

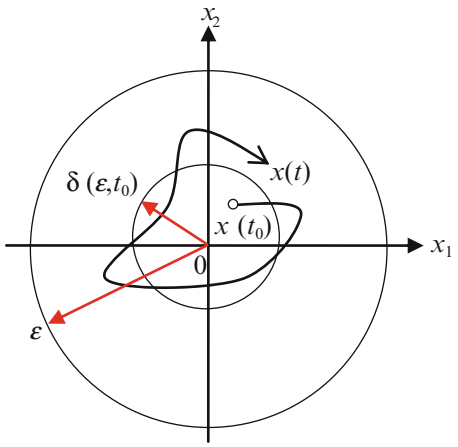


Fig. 8.7 Geometric interpretation of Lyapunov stability for two-dimensional dynamics

any $\varepsilon>0$ *and* $t_0 \geq 0$ *there exists* $\delta(\varepsilon,\ t_0)>0$ *such that for all initial conditions* $\|x(t_0)\|<\delta$ *and for all* $t \geq t_0 \geq 0$, *the corresponding system trajectories are bounded, as in* $\|x(t)\|<\varepsilon$. *The equilibrium is uniformly stable if it is stable and* δ *does not depend on* t_0. *Finally, the equilibrium is unstable if it is not stable.*

Using logical symbols such as $\wedge$ "and," $\forall$ "for any," $\exists$ "there exists," and $\Rightarrow$ "implies," we can formally define the meanings of stable, uniformly stable, and unstable equilibriums (note that the equilibrium under consideration is the origin in R^n):

$$\boxed{\text{Stable}}$$
$$\forall\varepsilon>0\ \forall t_0>0\ \exists\delta(\varepsilon,\ t_0)>0\ \forall t \geq t_0\ \boxed{\|x(t_0)\|<\delta(\varepsilon,\ t_0) \Rightarrow \|x(t)\|<\varepsilon}$$
$$\boxed{\text{Uniformly Stable}}$$
$$\forall\varepsilon>0\ \forall t_0>0\ \exists\delta(\varepsilon)>0\ \forall t \geq t_0\ \ \boxed{\|x(t_0)\|<\delta(\varepsilon) \Rightarrow \|x(t)\|<\varepsilon}$$
$$\boxed{\text{Unstable}}$$
$$\exists\varepsilon>0\ \exists t_0>0\ \forall\delta>0\ \exists T \geq t_0\ \ \boxed{\|x(t_0)\|<\delta \wedge \|x(T)\|>\varepsilon} \tag{8.17}$$

For two-dimensional dynamics, Lyapunov stability of the origin admits a simple geometrical interpretation (Fig. 8.7).

The origin is stable if given a sphere with a radius ε, one can find a smaller sphere whose radius is $\delta \leq \varepsilon$, such that all trajectories that start in the smaller sphere will continue to evolve within the larger sphere, for all $t \geq t_0$. The origin is uniformly stable if δ is independent of t_0. Finally, the origin is unstable if there exists an ε-sphere and an initial time t_0, such that no matter how close to the origin a trajectory starts, it will exit this sphere at some finite time T.

Such a geometrical explanation of Lyapunov stability can be easily extended to n-dimensional dynamics (8.15) (Fig. 8.8).

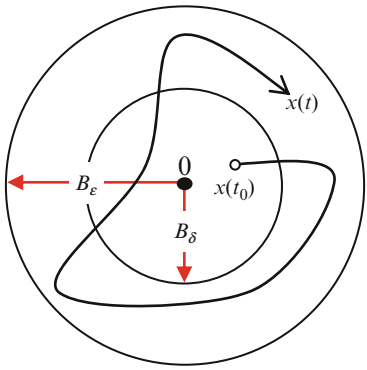


Fig. 8.8 Geometric interpretation of Lyapunov stability in n-dimensional state space

In essence, Lyapunov stability of the origin means that given an outer-sphere $B_\varepsilon = \{x \in R^n : \|x\| \leq \varepsilon\}$ in the system state space R^n, one can find an inner-sphere $B_\delta = \{x \in R^n : \|x\| \leq \delta\} \subset B_\varepsilon$, such that any trajectory that starts in the inner-sphere B_δ will evolve inside the outer-sphere B_ε, for all future times.

A unique feature of nonlinear dynamical systems is their ability to display a completely different behavior in various domains. For example, systems that are stable in a neighborhood of the origin may become unstable, or go to a different equilibrium, if their initial conditions are chosen outside of this neighborhood. For these reasons, we need to be able to clearly distinguish between local and global stability.

The local feature of Lyapunov stability definitions (8.17) is understood in the sense that for a given outer-sphere B_ε, one must find a set of initial conditions (an inner-sphere B_δ) such that the resulting trajectories stay within the outer-sphere B_ε. It is easy to see that if the origin is stable, then for an outer-sphere B_{ε_1} of a bigger radius $\varepsilon_1 > \varepsilon$, the same inner-sphere B_δ can be used to show stability of the system equilibrium. Suppose that the radius of the inner-sphere can be increased indefinitely, as the radius of the outer-sphere increases. In other words, let us suppose that $\delta(\varepsilon,\ t_0) \to \infty$, as $\varepsilon \to \infty$. This would indicate that the set of initial conditions, which lead to stable trajectories, is getting bigger. Eventually, one can declare that starting anywhere in R^n, a trajectory will not deviate too far from where it began and as a result, the stability property becomes global.

Definition 8.3. Global Stability *The origin is globally stable if it is stable and* $\lim_{\varepsilon \to \infty} \delta(\varepsilon,\ t_0) = \infty$.

Dependence of the system trajectories on a selected initial time t_0 is yet another unique feature of nonautonomous systems. This is in contrast to autonomous dynamics $\dot{x} = f(x)$, whose solutions depend only on the difference $(t - t_0)$. For nonautonomous systems, stability of an equilibrium will in general be dependent on the selected initial time t_0. That is why we had to introduce the notion of uniform stability. Also, in the definition (8.17), we emphasized that there would exist

equilibriums whose stability may or may not depend on the system initial conditions near the equilibrium. We had to also characterize the notion of instability which was merely a logic negation of the stability concept. The next example illustrates differences between the notions of stability and uniform stability.

Example 8.9 Stable but Not Uniformly Stable Equilibrium Consider the linear time-dependent dynamics

$$\dot{x}(t) = 2\,t\,(3\,\sin(t) - 1)\,x(t)$$

with the initial condition $x(t_0)$. The system solution is

$$\begin{aligned} x(t) &= x(t_0)\ \exp\left[\int_{t_0}^{t} 2\,\tau\,(3\,\sin(\tau) - 1)\,\mathrm{d}\tau\right] \\ &= x(t_0)\ \exp\left(6\ \sin t - 6\,t\ \cos t - t^2 - 6\ \sin t_0 + 6\,t_0\ \cos t_0 + t_0^2\right) \end{aligned}$$

The obvious inequality

$$6\ \sin t - 6\,t\ \cos t - t^2 \leq 6 + \underbrace{\left(t - t^2\right)}_{\leq \frac{1}{4}} \leq 6.25$$

implies

$$|x(t)| \leq |x(t_0)|\ \underbrace{\exp\left(6.25 - 6\ \sin t_0 + 6\,t_0\ \cos t_0 + t_0^2\right)}_{c(t_0)} = |x(t_0)|\,c(t_0)$$

Clearly, the origin is the system equilibrium. Is it stable? Since

$$|x(t)| \leq |x(t_0)|\,c(t_0)$$

it is evident that given any positive ε, we can select $\delta(\varepsilon,\ t_0) = \frac{\varepsilon}{c(t_0)}$ and immediately verify that for all $|x(t_0)|<\delta$, the relation

$$|x(t)| \leq |x(t_0)|\,c(t_0) \leq \frac{\varepsilon}{c(t_0)}\,c(t_0) = \varepsilon$$

takes place for all $t \geq t_0$. According to (8.17), we have proved stability of the origin. Is this equilibrium uniformly (in t_0) stable? In order to answer that question, we need to study sensitivity of the system solutions due to changes in t_0. Toward that end, let $t_0 = 2\,k\,\pi$, where k is a fixed positive integer. We can examine $x(t)$ at $t = t_0 + \pi = (2k + 1)\,\pi$ and get

$$x((2k + 1)\,\pi) = x(2\,k\,\pi)\ \exp((4\,k + 1)\,(6 - \pi)\,\pi)$$

or, equivalently,

$$\frac{x((2k+1)\,\pi)}{x(2\,k\,\pi)} = \exp((4\,k+1)\,(6-\pi)\,\pi) > 1, \quad \forall k \geq 1$$

So, the sequence $x(2\,k\,\pi)$ tends to infinity, as $k = 1, 2, \ldots, \infty$. In other words, there is an unboundedly increasing sequence of initial time instants $t_0(k) = 2\,k\,\pi$ which leads to an unboundedly increasing sequence of the initial values for the system solutions $x(t_0(k)) = x(2\,k\,\pi) \underset{k\to\infty}{\to} \infty$. Therefore, given any $\varepsilon > 0$, there is no $\delta(\varepsilon)$ independent of t_0 that would satisfy the uniform stability definition in (8.17).□

As shown in Example 8.9, in addition to being stable, perturbed trajectories may asymptotically converge back to the equilibrium. This observation naturally leads to the definitions of (a) asymptotic stability, (b) uniform asymptotic stability, and (c) global uniform asymptotic stability.

Definition 8.4. Asymptotic Stability *The equilibrium point $x^* = 0$ of (8.15) is asymptotically stable if it is stable and there exists a positive constant $c = c(t_0)$ such that $x(t) \to 0$ as $t \to \infty$, for all $\|x(t_0)\| \leq c$.*

Definition 8.5. Uniform Asymptotic Stability *The equilibrium point $x^* = 0$ of (8.15) is uniformly asymptotically stable if it is uniformly stable and there exists a positive constant c, independent of t_0, such that $x(t) \to 0$ as $t \to \infty$, for all $\|x(t_0)\| \leq c$, uniformly in t_0, where the limit uniformity is understood in the following sense:*

$$\exists c\ \forall \eta > 0\ \exists T(\eta)\ \forall t \geq t_0 + T(\eta)\ \forall \|x(t)\| \leq c\ \Rightarrow \|x(t)\| \leq \eta$$

Definition 8.6. Global Uniform Asymptotic Stability *The origin is globally uniformly asymptotically stable if it is uniformly asymptotically stable and* $\lim_{\varepsilon\to\infty} \delta(\varepsilon) = \infty$.

Achieving uniform asymptotic stability is a highly desirable property in any control design since asymptotically stable systems are able to maintain their closed-loop performance in the presence of perturbations and disturbances. We shall see that, in general, adaptive controllers achieve uniform stability and force the system tracking errors to converge to zero, asymptotically in time. This key property is lesser than uniform asymptotic stability, but it is greater than uniform stability; that is, in addition to being uniformly stable, certain signals (such as tracking errors) in the closed-loop system asymptotically tend to zero, while others are kept uniformly stable and bounded.

8.5 Lyapunov Stability Theorems

In his seminal work on stability of motion, A.M. Lyapunov introduced two theorems, known as Lyapunov's indirect (first) and direct (second) methods, for assessing stability of nominal solutions that arise in dynamical systems, which are governed

by a finite number of coupled ordinary differential equations. Lyapunov's methods provide verifiable sufficient conditions for stability of a nominal trajectory. Moreover, neither method requires an explicit knowledge of the system solutions.

Lyapunov's indirect method allows one to draw conclusions about the stability of an equilibrium point (the origin) for a nonlinear autonomous n-dimensional system $\dot{x} = f(x)$. The method is based on the linearization of the system dynamics around an equilibrium. In order for the original nonlinear system to be locally stable in the sense of Lyapunov, it is sufficient to show that the system Jacobian matrix $A = \frac{\partial f(x)}{\partial x}\Big|_{x=0}$ has all its eigenvalues $\{\lambda_i\}_{i=1,2,\ldots,n}$ in the complex open left-half plane: Re $\lambda_i < 0$, $\forall i = 1, 2, \ldots, n$. If Re $\lambda_i > 0$ for at least one eigenvalue of A, then the origin is unstable. If A has eigenvalues on the $j\omega$-axis, then the indirect method of Lyapunov does not apply. Further details, including formal proofs, can be found in [1–3]. From the control design point of view, the indirect method of Lyapunov provides the much needed theoretical foundation for application of linearization-based controllers in nonlinear systems.

Our main interest will be focused on Lyapunov's direct method. Specifically, we will discuss the method formulation and its applications to analyzing uniform stability of nonautonomous systems (8.15). We begin with the definitions of positive and negative-definite (semidefinite) functions. Subsequently, we will utilize these functions to constructively determine stability of an equilibrium point.

Definition 8.7. Positive-Definite and Semidefinite Functions *A scalar function $V(x) : R^n \to R$ of a vector argument $x \in R^n$ is called locally positive definite (semidefinite) if $V(0) = 0$, and there exists a constant $r > 0$ such that $V(x) > 0$ $(V(x) \geq 0)$, for all nonzero $x \in R^n$ from the r-neighborhood of the origin $B_r = \{x \in R^n : \|x\| \leq r\}$. The function is said to be globally positive definite if $B_r = R^n$.*

Definition 8.8. Negative-Definite and Semidefinite Functions *A scalar function of $V(x) : R^n \to R$ of a vector argument $x \in R^n$ is called locally (globally) negative definite (semidefinite) if the function $(-V(x))$ is locally (globally) positive definite (semidefinite).*

Example 8.10 Sign-Definite and Semidefinite Functions Consider a scalar function of a scalar argument: $V(x) = x^2(9 - x^2)$. A graphical sketch of this function is shown in Fig. 8.9.

It is easy to see that this function is locally positive definite on the open interval $(-3,\ 3)$, and it becomes positive semidefinite on the closed interval $[-3,\ 3]$. On the other hand, the function $V(x) = x^2$ is globally positive definite. Furthermore, if $P \in R^{n \times n}$ is a symmetric positive-definite (semidefinite) matrix, then the function $V(x) = x^T P x$ is globally positive definite (semidefinite), while $W(x) = -x^T P x$ represents a globally negative-definite (semidefinite) function. □

Next, we introduce the concept of the time derivative of a scalar function along the trajectories of a differential equation. Suppose that we are given a scalar continuously differentiable function $V(x)$, whose vector argument $x(t) \in R^n$ represents a time-varying trajectory of the nonautonomous system (8.15). We can compute the time derivative of $V(x(t))$ along the system solution $x(t)$:

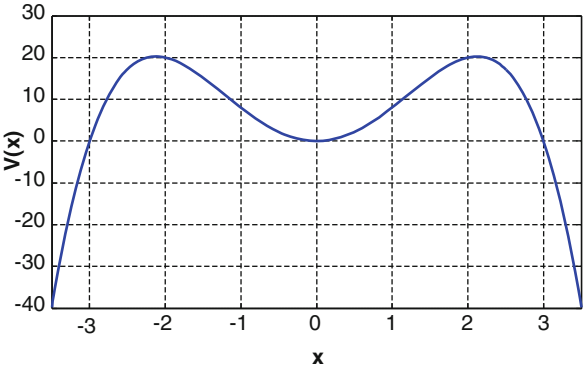


Fig. 8.9 Locally positive-definite function from Example 8.10

$$\boxed{\dot{V}(x)} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(t,\ x) = \boxed{\nabla V(x)\, f(t,\ x)} \tag{8.18}$$

where $\nabla V(x) = \left(\frac{\partial V}{\partial x_1},\ \frac{\partial V}{\partial x_2},\ \ldots,\ \frac{\partial V}{\partial x_n}\right)$ is the row vector gradient of $V(x)$ with respect to x. We immediately note that the time derivative of $V(x)$ along the trajectories of (8.15) depends not only on the function $V(x)$ but also on the system dynamics under consideration. Changing the latter while keeping the same V will in general yield a different $\dot{V}(x)$. We are now fully equipped to formulate the direct (second) method of Lyapunov.

Theorem 8.5. Lyapunov's Direct Method for Assessing Uniform Stability of Nonautonomous Systems *Let $x^* = 0 \in R^n$ be an equilibrium point for the nonautonomous dynamics (8.15), whose initial conditions are drawn from a domain $D \subset R^n$, with $x^* \in D$ and $t_0 = 0$. Suppose that on the domain D there exists a continuously differentiable locally positive-definite function $V(x) : D \to R$, whose time derivative along the system trajectories is locally negative semidefinite:*

$$\dot{V}(x) = \nabla V(x)\, f(t,\ x) \leq 0 \tag{8.19}$$

for all $t \geq 0$ and for all $x \in D$. Then, the system equilibrium $x^* = 0$ is locally uniformly stable in the sense of Lyapunov. If in (8.19) $\dot{V}(x) < 0$ for all nonzero x and for all $t \geq 0$ (the time derivative along the system trajectories is locally negative definite), then the origin is locally uniformly asymptotically stable. ■

We shall immediately note that Lyapunov's direct method presents sufficient conditions for appraising uniform stability (formal proof can be found in [2, Th. 4.8, pp. 151–153]). These sufficient conditions are expressed in terms of a locally positive-definite function $V(x)$, which is often called a Lyapunov function candidate. If in addition, the strict inequality (8.19) holds, then $V(x)$ becomes what is commonly referred to as a Lyapunov function. In terms of these concepts, Theorem 8.5 states that the origin is a uniformly stable if given the system dynamics,

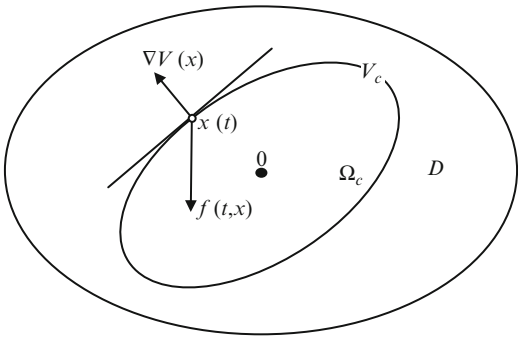


Fig. 8.10 Geometrical interpretation of Lyapunov's direct method

a Lyapunov function can be found. Conversely, if a Lyapunov function candidate does not satisfy the sufficient for stability requirement (8.19), no definite conclusions can be drawn and the search for a suitable Lyapunov function must continue.

Let us briefly discuss a geometric interpretation of Lyapunov's direct method. Choosing a sufficiently small positive constant c, we can ensure that the level set $V_c = \{x \in D : V(x) = c\}$ of the Lyapunov function $V(x)$ resides inside D (see Fig. 8.10).

Then, it is possible to show that the interior set $\Omega_c = \{x \in D : V(x)\}$, whose boundary is V_c, is closed and bounded (i.e., compact). For any $x \in V_c$, the gradient row vector $\nabla V(x)$ points perpendicular to the tangent hyperplane that touches the level set at x. Also, the inequality (8.19) implies that at any given time t and for any $x \in V_c$, the angle between the gradient vector $\nabla V(x)$ and the system dynamics $f(t, x)$ is no less than $\frac{\pi}{2}$. Therefore, the system trajectory will not leave Ω_c. Moreover, since $V(x(t))$ is nonincreasing, then $x(t)$ will remain in this set for all future times. If in addition it is assumed that $\dot{V}(x)<0$, then the system trajectories, starting anywhere in D, will evolve by entering a sequence of diminishing level sets $(V_{c_1} \supset \cdots \supset V_{c_k} \supset \cdots)$ with $(c>c_1>\cdots>c_k>\cdots)$, and as a result, these solutions will asymptotically approach the origin.

The Lyapunov function $V(x)$can now be viewed as an "energy-like" function for testing stability of a system. If the values of V do not increase along the system trajectories, then the origin is uniformly stable. If V strictly decreases, then in addition to being stable, the system trajectories will approach the origin asymptotically.

Example 8.11 Consider the scalar system

$$\dot{x} = f(t, x)$$

where $f(t, x)$ is locally Lipschitz on an open interval $(-a,\ a)$, $f(t, 0) = 0$, and $xf(t, x)<0$ for all $t \geq 0$ and all nonzero $x \in (-a,\ a)$; that is, the graph of $f(t, x)$ is located in the second and the fourth quadrants, uniformly in t and for all x from $(-a,\ a)$. The system dynamics are shown in Fig. 8.11.

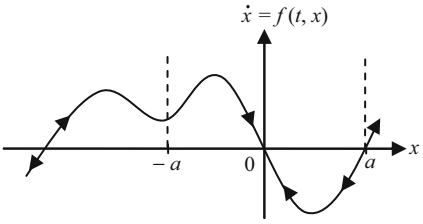


Fig. 8.11 System dynamics from Example 8.11

It is clear, that starting anywhere within the open interval $(-a,\ a)$, the system solutions will asymptotically converge to the origin. Let us now use Lyapunov's direct method to show that the origin is uniformly asymptotically stable. Toward that end, we consider a quadratic Lyapunov function candidate in the form $V(x) = x^2$. Its time derivative along the system trajectories is strictly negative for all nonzero $x \in (-a,\ a)$:

$$\dot{V}(x) = 2\,x\,\dot{x} = 2\,x f(t,\ x) < 0$$

Consequently, $V(x)$ is a Lyapunov function, and, according to Theorem 8.5, the origin is locally uniformly asymptotically stable. Of course, we already knew the answer since the system dynamics were scalar and the phase plane analysis method was readily applicable.

Suppose that the same system is n-dimensional and assume that the vector field $f(t,\ x)$ satisfies $x^T f(t,\ x) < 0$, uniformly in t and for all x from a domain $D \subset R^n$. We can use a quadratic Lyapunov function in the form $V(x) = x^T x$, show that its time derivative along the system trajectories is negative,

$$\dot{V}(x) = 2\,x^T\,\dot{x} = 2\,x^T f(t,\ x) < 0$$

and, thus, prove the uniform asymptotic stability property of the origin. Note that, in this case, the phase plane analysis does not apply. □

Lyapunov functions are by no means unique. Recalling the scalar dynamics in Example 8.11, let us assume that the system is autonomous. We can prove asymptotic stability using the same Lyapunov function as before. In order to show that it is not unique, let us consider the following Lyapunov function candidate:

$$V(x) = -\int_0^x f(y)\,\mathrm{d}y$$

Since $x f(x) < 0$ for all nonzero x, $V(x)$ is positive definite and $V(0) = 0$. Therefore, it represents a Lyapunov function candidate. The function time derivative along the system trajectories is negative:

$$\dot{V}(x) = -f(x)\,\dot{x} = -f^2(x) < 0$$

for all nonzero $x \in (-a, a)$. Consequently, the origin is uniformly asymptotically stable. □

The uniform asymptotic stability property calls for a subset of D. Starting there, the system solutions will converge to the origin. This subset is called the region of attraction. We shall study the case when the system domain and the region of attraction both equal R^n. This will lead to the concept of global uniform asymptotic stability.

Definition 8.9. *If the region of attraction of a uniformly asymptotically stable equilibrium is R^n, then the equilibrium is said to be globally uniformly asymptotically stable.*

The next definition leads to a verifiable condition for a Lyapunov function to yield global uniform stability properties.

Definition 8.10. *A Lyapunov function candidate $V(x)$: $R^n \to R$ defined such that $\lim_{\|x\|\to\infty} V(x) = \infty$ is called radially unbounded.*

Let $V_c = \{x \in R^n : V(x) = c\}$ denote a level set of a radially unbounded Lyapunov function candidate $V(x) : R^n \to R$, and let $\Omega_c = \{x \in R^n : V(x) \le c\}$ be the union of the interior set of V_c and V_c itself. Consider a converging sequence $\lim_{n\to\infty} x_n = a$, with all x_n from Ω_c. Then, the limit point a must also be in Ω_c. In fact, since $V(x)$ is continuous on R^n and $V(x_n) \le c$ for all $n = 1, 2, \ldots$, we get $c \ge \lim_{n\to\infty} V(x_n) = V(a)$, and consequently $a \in \Omega_c$. We have proved that every converging sequence in Ω_c has its limit point in the same set. Hence, Ω_c is a closed set. Moreover, we can prove that Ω_c is bounded. This fact can be shown by contradiction. Suppose that Ω_c is unbounded. Then, there must exist a sequence of points $\{x_n\} \in \Omega_c$, whose limit is infinity. Since $V(x)$ is continuous and radially unbounded, then $c \ge \lim_{n\to\infty} V(x_n) = \infty$, which is an obvious contradiction to the argument. Therefore, Ω_c is a bounded set. Since it is also closed and belongs to R^n, Ω_c is compact.

The next theorem states that if a radially unbounded Lyapunov function can be found, then the local uniform (asymptotic) stability properties from Theorem 8.5 become global. When applied to autonomous systems, this result is also known as Barbashin–Krasovskii–LaSalle theorem [1, 2].

Theorem 8.6. *Let $x = 0$ be an equilibrium point for (8.15). Let $V(x) : R^n \to R$ be a radially unbounded Lyapunov function of the system. Then, the system equilibrium is globally uniformly asymptotically stable.* ■

Simple examples of radially unbounded Lyapunov function candidates include quadratic functions of the form $V(x) = x^T P x$, where $P \in R^{n\times n}$ is a symmetric positive-definite matrix.

Example 8.12 The rotational motion of a rigid aircraft in three-dimensional space is governed by the following system of ordinary differential equations:

$$J\,\dot{\omega} = -[\omega \times J\,\omega] + M$$

where $\omega = (\ p \quad q \quad r\ )^T$ is the body angular velocity vector, with the roll (p), the pitch (q), and the yaw (r) velocity components, $J \in R^{3\times 3}$ is the aircraft inertia matrix:

$$J = \begin{pmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{xz} & 0 & J_{zz} \end{pmatrix}$$

with positive components $\left(J_{xx},\ J_{yy},\ J_{zz},\ J_{xz}\right)$, and $M \in R^3$ is the vector of aerodynamic/propulsive moments, computed with respect to the vehicle center of gravity. We assume that

$$\det J = J_{yy}\left(J_{xx}J_{zz} - J_{xz}^2\right) > 0$$

and also suppose that the moment vector M represents the system control input. The control task is to select M such that the aircraft rotational dynamics become globally uniformly asymptotically stable. We begin by considering a quadratic Lyapunov function candidate in the form

$$V(\omega) = \omega^T J\,\omega$$

This is indeed a Lyapunov function candidate since $V(0) = 0$ and J is symmetric and positive definite. We proceed to compute the time derivative of $V(\omega)$ along the trajectories of the aircraft rotational dynamics:

$$\dot{V}(\omega) = 2\,\omega^T J\,\dot{\omega} = 2\,\omega^T\left(-[\omega \times J\,\omega] + M\right) = 2\,\omega^T M$$

According to Theorem 8.5, we need $\dot{V}(\omega) < 0$. This can be easily achieved if we select the control input as a weighted negative feedback on ω,

$$M = -P\,\omega$$

with a symmetric positive-definite matrix of weights $P \in R^{3\times 3}$. Then, for any nonzero angular velocity $\omega \in R^3$,

$$\dot{V}(\omega) = -2\,\omega^T P\,\omega < 0$$

and so, the origin is uniformly asymptotically stable. Moreover, since $V(\omega)$ is radially unbounded, the achieved closed-loop uniform asymptotic stability property is global. This example illustrates both the practicality and the effectiveness of Lyapunov's direct method. Not only we were able to assert the desired stability property but we did so by using the "inverse" Lyapunov design arguments; that is, we chose our control input to enforce the sufficient conditions of Theorem 8.5. □

Example 8.13 For the linear time-invariant (LTI) n-dimensional dynamics,

$$\dot{x} = A\,x$$

with a Hurwitz (stable) matrix $A \in R^{n\times n}$, consider a quadratic Lyapunov function candidate $V(x) = x^T P\,x$, where $P \in R^{n\times n}$ is a symmetric positive-definite matrix. Let $Q \in R^{n\times n}$ be another symmetric positive-definite matrix. The time derivative of $V(x)$ along the system solutions is

$$\dot{V}(x) = x^T P\,\dot{x} + \dot{x}^T P\,x = x\left(P\,A + A^T P\right)x$$

If we can make this derivative negative for all nonzero $x \in R^n$, then we would prove global uniform asymptotic stability of the origin. In order to do that, we define P to be the solution of the so-called Lyapunov algebraic equation:

$$P\,A + A^T P = -Q$$

It turns out that given any symmetric positive definite Q, the Lyapunov algebraic equation has the unique symmetric positive-definite solution $P = P^T > 0$ if and only if A is Hurwitz [3]. Then,

$$\dot{V}(x) = -x^T Q\,x < 0$$

for all nonzero $x \in R^n$, which immediately proves global uniform asymptotic stability of the origin.

Evidently, since the system is linear and time invariant, we could have proven asymptotic stability by simply noting that A is Hurwitz. Nevertheless, the Lyapunov' arguments allow us to establish an important link between the stability of LTI systems and the Lyapunov's direct method. This link is given by the Lyapunov algebraic equation, and the latter will become the key design component for adaptive controllers. □

8.6 Uniform Ultimate Boundedness

The concepts of stability in the sense of Lyapunov are formulated with respect to an equilibrium or a nominal trajectory. Often, systems are designed to operate in the presence of disturbances and other uncertainties. As a result, the "ideal" definition of an equilibrium may not apply. Consider the nonautonomous system

$$\dot{x} = f(t, x) + \xi(t), \quad x(t_0) = x_0 \tag{8.20}$$

subject to a bounded disturbance $\xi(t) \in R^n$, with $\|\xi(t)\| \leq \xi_{\max}$. Suppose that $f : [0, \infty) \times D \to R^n$ is piece-wise continuous in t, locally Lipschitz in x on $[0, \infty) \times D$, and $D \subset R^n$ is a domain that contains the origin $x = 0$. Also, suppose that $f(t, 0) = 0, \quad \forall\, t \geq 0$. It is easy to see that no matter how small the disturbance bound $\xi_{\max}$ is, the origin is no longer an equilibrium point of the system. Nevertheless, we can still use Lyapunov' direct method to study the system behavior outside of the sphere $B_{\xi_{\max}} = \{x \in R^n : \|x\| \leq \xi_{\max}\}$, as if the origin is the system equilibrium. The main idea is to find a Lyapunov-like function $V(x)$ for all x outside of a bigger sphere $B_r \supset B_{\xi_{\max}}$ and then show that in finite time T the system trajectories enter B_r and remain there for all $t \geq T$. This thought will eventually lead us to the concept of uniform ultimate boundedness (UUB).

Example 8.13 Consider the scalar nonautonomous dynamics

$$\dot{x} = -x + \xi(t), \quad x(t_0) = x_0 > \xi_{\max} > 0$$

where $\xi(t)$ is a time-varying unknown bounded by $\xi_{\max}$ disturbance. Clearly, the system has no equilibrium points. The system solutions can easily be found:

$$x(t) = e^{-(t-t_0)}\, x_0 + \int_{t_0}^{t} e^{-(t-\tau)}\, \xi(\tau)\, \mathrm{d}\tau$$

We can also compute an upper bound,

$$|x(t)| \leq e^{-(t-t_0)}\, |x_0| + \left(1 - e^{-(t-t_0)}\right) \xi_{\max} \leq e^{-(t-t_0)}\, |x_0| + \xi_{\max}$$

and show that for any initial condition x_0 and any given $r > 0$, there must exist a finite time $0 \leq T(x_0, r) < \infty$, such that $|x(t)| \leq \xi_{\max} + r$, for all $t \geq t_0 + T(x_0, r)$. In fact, if $|x_0| \leq r$, then $|x(t)| \leq e^{-(t-t_0)}\, r + \xi_{\max} \leq r + \xi_{\max}$, and so $T(x_0, r) = 0$. If, on the other hand, $|x_0| > r$, then it is sufficient to choose $T(x_0, r) = \ln \frac{|x_0|}{r}$. This simple argument shows that the system trajectories enter a neighborhood of the origin $B_r = \{x \in R : |x| \leq r\}$ in finite time $T(x_0, r)$ and continue to evolve within the neighborhood afterward.

Alternatively, we can also exploit Lyapunov's direct method to show uniform ultimate boundedness of the system solutions. Let us utilize $V(x) = x^2$, which in this case is not a Lyapunov function candidate since the origin is not an equilibrium of the system. Nevertheless, we proceed to calculate the function time derivative along the system trajectories:

$$\begin{aligned} \dot{V}(x) = 2\, x\, \dot{x} &= 2\, x\, (-x + \xi(t)) = -2\, x^2 + 2\, x\, \xi(t) \\ &\leq -2\, x^2 + 2\, \xi_{\max}\, |x| = -2\, |x|\, (|x| - \xi_{\max}) \end{aligned}$$

Given any positive constant $\varepsilon > 0$, it is evident that

$$\dot{V}(x) < 0, \quad \forall\, |x| \geq \xi_{\max} + \varepsilon$$

Let $r = \xi_{\max} + \varepsilon$. Then, the time derivative of V is negative outside of the closed interval $B_r = [-r,\ r]$. Next, we are going to show that all solutions that start outside of B_r will reenter the interval within a finite time and will remain there forward in time. Since $\dot{V}$ is negative for all $|x| \geq r$, then the solutions starting inside B_r will remain there. Hence, these trajectories are uniformly bounded in time, that is, $|x(t)| \leq r$, $\forall t \geq t_0$. Starting from any $|x_0| > r$, $\dot{V}$ is strictly negative in the annulus set $\left\{r^2 \leq V(x) \leq x_0^2\right\}$, which implies that in this set $V(x(t))$ will continue to decrease monotonically until the solution enters $B_r = \{|x| \leq r\} = \left\{V(x) \leq r^2\right\}$, at some finite time $T(x_0,\ r)$. From that time on, the solution will evolve within B_r since $\dot{V}$ is strictly negative on its boundary $V(x) = r^2$. So again, we conclude that the system solutions are UUB with the ultimate bound $|x(t)| \leq r$. Similar to proving stability, the main advantage of applying Lyapunov's direct method to establish UUB of trajectories is the fact that the method does not require the knowledge of an explicit form of the system solutions. □

We now give a formal definition of the UUB concept as it is stated in [2].

Definition 8.11. *The solutions of (8.20) are uniformly ultimately bounded with ultimate bound b if there exist positive constants b and c, independent of $t_0 \geq 0$, and for every $a \in (0,\ c)$, there is $T = T(a,\ b)$, independent of t_0, such that*

$$\|x(t_0)\| \leq a \Rightarrow \|x(t)\| \leq b, \quad \forall\, t \geq t_0 + T \tag{8.21}$$

These solutions are said to be globally uniformly ultimately bounded if (8.21) holds for arbitrarily large a.

Graphical interpretation of the UUB concept is shown in Fig. 8.12.

In the definition above, the term "uniform" indicates that the bound b does not depend on t_0. The term "ultimate" means that boundedness holds after the lapse of a finite time T. The constant c defines a neighborhood of the origin, independent of t_0, such that all trajectories starting in the neighborhood will remain bounded in time. If c can be chosen arbitrarily large, then the local UUB property becomes global.

The notion of UUB can be considered as a "milder" form of stability in the sense of Lyapunov (SISL). A brief comparison between the SISL and the UUB concepts is given below:

- SISL is defined with respect to an equilibrium, while UUB is not.
- Asymptotic SISL is a strong property that is very difficult to achieve in practical dynamical systems.
- SISL requires the ability to keep the state arbitrarily close to the system equilibrium by starting sufficiently close to it. This is still too strong a requirement for practical systems operating in the presence of uncertainties and unknown disturbances.
- A bound b in the UUB concept cannot be made arbitrarily small by starting closer to the system equilibrium (if it has one) or to the origin.

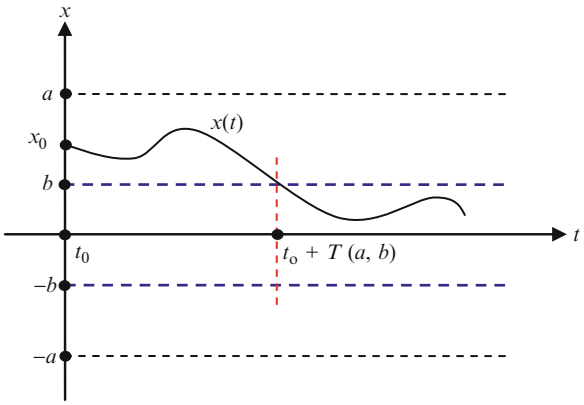


Fig. 8.12 Graphical interpretation of the UUB concept for nonautonomous dynamics

Next, we present a Lyapunov-based analysis of UUB properties. Suppose that for a given continuously differentiable positive-definite function $V(x)$, we can choose two finite positive constants $0 < \varepsilon < c < \infty$, such that the sets $\Omega_\varepsilon = \{V(x) \leq \varepsilon\}$ and $\Omega_c = \{V(x) \leq c\}$ are closed and bounded (i.e., compact). This would be true if, for example, $V(x) = x^T P x$ and P is a symmetric positive-definite matrix. Consider the annulus set in R^n,

$$\Lambda = \{x \in R^n : \varepsilon \leq V(x) \leq c\} = \Omega_c - \Omega_\varepsilon$$

and presume that the time derivative of $V(x(t))$ along the trajectories of the nonautonomous dynamical system (8.20) is strictly negative definite inside Λ:

$$\dot{V}(x(t)) < 0, \quad \forall x \in \Lambda, \quad \forall t \geq t_0$$

Then, a trajectory that starts in the annulus would have to move in a direction where $V(x(t))$ is decreasing. Since the annulus boundary consists of the function level sets, the trajectory would be trapped between the two sets and it would have to move toward the origin. Thus, inside the annulus, the system solution behaves as if the origin is a uniformly asymptotically stable equilibrium, which it is not.

Starting from an initial condition $x_0 = x(t_0)$ at a time instant $t_0 \geq 0$, the corresponding system trajectory $x(t)$ will evolve such that the function $V(x(t))$ decreases until the trajectory enters (in finite time T) the set Ω_ε, where it will remain afterward.

This argument proves the UUB property of the system solutions, with the ultimate bound $b = \max_{x \in \Omega_\varepsilon} \|x\| = \max_{x \in \partial\Omega_\varepsilon} \|x\|$, achieved on the boundary $\partial\Omega_\varepsilon$ of the set Ω_ε. The three sets Λ, Ω_c, Ω_ε and the UUB bound b are shown in Fig. 8.13. □

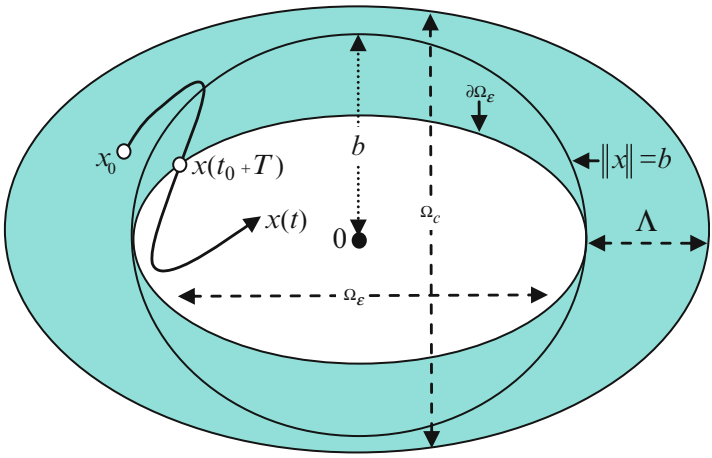


Fig. 8.13 Compact sets in the UUB analysis

Example 8.14 Let D denote a domain in R^n, where the system dynamics are defined as

$$\dot{x} = A\,x + B\,\varepsilon(t,\,x), \quad x(t_0) = x_0 \tag{8.22}$$

with the state $x \in R^n$, a Hurwitz matrix $A \in R^{n\times n}$, a constant matrix $B \in R^{n\times m}$, and with a bounded function $\varepsilon(t,\,x)$: $R \times R^n \to R^m$, $\|\varepsilon(t,\,x)\| \le \varepsilon_{\max}$, which is assumed to hold for all $t \ge t_0$ and $x \in D$. Let us choose $Q = Q^T > 0$ and consider a quadratic positive-definite function in the form

$$V(x) = x^T P\,x \tag{8.23}$$

where $P = P^T > 0$ is the unique positive-definite symmetric solution of the algebraic Lyapunov equation

$$P\,A + A^T P = -Q \tag{8.24}$$

Such a solution exists for any symmetric positive definite Q since A is Hurwitz. Due to the latter, it is intuitively clear that the trajectories of (8.22) are UUB. Let us formally prove it.

The time derivative of V along the system trajectories satisfies the following relation for all $(t \ge t_0,\ x \in D)$,

$$\dot{V}(x) = -x^T Q x + 2x^T P B\,\varepsilon(t,x) \le -\|x\|\,(\lambda_{\min}(Q)\,\|x\| - 2\,\lambda_{\max}(P)\,\|B\|\,\varepsilon_{\max}) \tag{8.25}$$

where $\lambda_{\min}(Q)$, $\lambda_{\max}(P)$ are the minimum and the maximum eigenvalues of Q and P, respectively. From (8.25), it follows that $\dot{V}(x)<0$ for all x that are located outside of the compact set,

$$B_r = \left\{ x \in D : \|x\| \leq 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\|B\|\,\varepsilon_{\max} = r \right\} \tag{8.26}$$

where we have assumed a sufficiently small $\varepsilon_{\max} > 0$ for the inclusion $B_r \subset D$ to hold.

We can define the maximal level set of $V(x)$ in D:

$$\Omega_{\max} = \max_c \{x \in D\colon V(x) = c\} = \{x \in D : V(x) = c_{\max}\} \tag{8.27}$$

If the domain D is bounded, then $c_{\max} > 0$ is finite. This follows from the fact that $V(x)$ is a continuous quadratic function of $x$and as such, its maximum on a bounded domain exists and is finite. On the other hand, if D is unbounded, then $c_{\max} = \infty$, and consequently, $\Omega_{\max}$ is unbounded as well. Either way, existence of this set is guaranteed.

Let us also define the minimal level set of $V(x)$ that contains B_r:

$$\Omega_{\min} = \min_c \{x \in B_r\colon V(x) = c\} = \{x \in \partial B_r\colon V(x) = c_{\min}\} \tag{8.28}$$

where ∂B_r denotes the boundary set of B_r. Existence of $\Omega_{\min}$ is guaranteed since B_r is compact and $V(x)$ is a continuous function with its minimum value achieved on the set boundary.

According to (8.25), $\dot{V}(x)<0$ for all x from the annulus

$$\Lambda = \{x \in R^n : c_{\min} \leq V(x) \leq c_{\max}\} \tag{8.29}$$

Figure 8.14 shows inclusion of the level sets.

Consequently, any trajectory that starts in Λ will have to enter the interior set of $\Omega_{\min}$ in finite time T, and it will remain there for all $t \geq t_0 + T$. This proves the UUB property of the system trajectories.

Next, we are going to estimate the corresponding ultimate bound b. In order to do this, we introduce the smallest sphere that contains $\Omega_{\min}$ (see Fig. 8.15):

$$B_R = \min_c \{x \in \Omega_{\min} : \|x\| \leq c\} \tag{8.30}$$

Since for all $x \in R^n$,

$$\lambda_{\min}(P)\,\|x\|^2 \leq \underbrace{x^T P x}_{V(x)} \leq \lambda_{\max}(P)\,\|x\|^2 \tag{8.31}$$

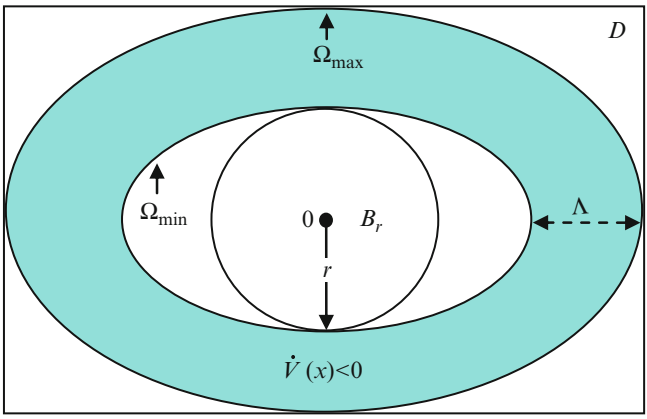


Fig. 8.14 Level sets from Example 8.14

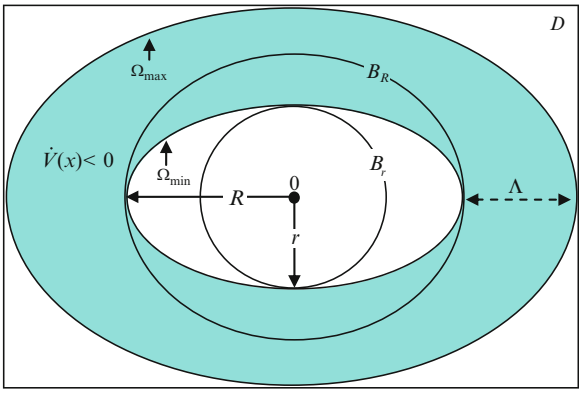


Fig. 8.15 Level sets and spheres from Example 8.14

then for all $x \in B_r$,

$$V(x) \leq \lambda_{\max}(P)\, r^2 \tag{8.32}$$

and, so

$$\Omega_{\min} = \left\{x \in \partial B_r : V(x) = \lambda_{\max}(P)\, r^2 = c_{\min}\right\} \tag{8.33}$$

For all $x \in \Omega_{\min}$, we have

$$\lambda_{\min}(P)\, \|x\|^2 \leq x^T P\, x = \lambda_{\max}(P)\, r^2 \tag{8.34}$$

and therefore,

$$\|x\|^2 \leq \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\, r^2 = R^2 \tag{8.35}$$

Table 8.1 UUB related assumptions and conclusions for the system from Example 8.14

Plant dynamics	$\dot{x} = A\,x + B\,\varepsilon(t,\,x), \quad x(t_0) = x_0$
Lyapunov equation	$P\,A + A^T\,P = -Q$
Lyapunov-like function	$V(x) = x^T\,P\,x$
Assumptions	A is Hurwitz $\|\varepsilon(t,\,x)\| \le \varepsilon_{\max}, \quad \forall x \in D \subset R^n \left\{ x \in D : \|x\| \le 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\|B\|\,\varepsilon_{\max} \right\} \subset D$
UUB	$\|x(t)\| \le 2\,\|B\|\,\varepsilon_{\max}\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}, \quad \forall t \ge t_0 + T$

In other words, the radius of the smallest sphere B_R that surrounds $\Omega_{\min}$ is

$$R = r\,\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \tag{8.36}$$

Clearly, $R > r$ as it is shown in Fig. 8.15. Finally, substituting the definition of r from (8.26) into (8.36), we get the ultimate bound for the system trajectories:

$$b = R = 2\,\|B\|\,\varepsilon_{\max}\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \tag{8.37}$$

Table 8.1 presents all of the key assumptions and relations that were utilized to establish the UUB result.

In summary, we have established the UUB property of the system (8.22). Our analysis was based on Lyapunov's direct method. Specifically, we have shown that all trajectories that start in $\Omega_{\max}$ will enter the interior of $\Omega_{\min}$ in finite time and will evolve inside of this set afterward, with the ultimate bound (8.37). □

8.7 Barbalat's Lemma

We now turn our attention to n-dimensional nonautonomous systems of the form

$$\dot{x} = f(t,\,x), \quad x(t_0) = x_0, \quad f(t,\,0) = 0 \tag{8.38}$$

with the vector field $f : R \times D \to R^n$ defined on a domain $D \subset R^n$. In order to ensure existence and uniqueness of the system solutions, we assume that $f(t, x)$ is piecewise continuous in t and locally Lipschitz-continuous in x, uniformly in t.

Suppose that we have a Lyapunov function candidate $V(x) : D \to R$, whose time derivative along the trajectories of (8.38) satisfies

$$\dot{V}(x) = \nabla V(x) f(t, x) \leq -W(x) \leq 0 \tag{8.39}$$

for all $x \in D$, where $W(x) : D \to R$ is a continuous positive semidefinite function on D. Then, according to Lyapunov's direct method, the origin is stable but not necessarily asymptotically stable, since $W(x)$ is not strictly positive definite. Let

$$E = \{x \in D : W(x) = 0\} \tag{8.40}$$

be a set of points in D where W is zero. Outside of E, $\dot{V}(x) < 0$. So, one may conjecture that the system trajectories that start outside of E will have to approach E, as time tends to infinity. This property (if it holds) would be equivalent to the LaSalle's invariance theorem [1–3], which is valid for autonomous systems only. Moreover, if our conjecture holds and if $E = \{0\}$, then the origin would become asymptotically stable. Before we go any further, let us consider an example.

Example 8.15 In adaptive control, we will often encounter nonautonomous systems, such as

$$\begin{aligned} \dot{e} &= -e + (\theta - \theta_*)\, \varphi(t) \\ \dot{\theta} &= -e\, \varphi(t) \end{aligned}$$

where θ_* is a constant and $\varphi(t)$ is a bounded function of time t. This system has multiple equilibrium points of the form $(0,\ \theta_*)^T$. As it turns out later on, this particular system represents closed-loop tracking error dynamics of an adaptive controller for a first-order plant. We would like to prove that the error tends to zero, $e(t) \underset{t \to \infty}{\to} 0$, while the parameter $\theta(t)$ remains uniformly bounded in time. Consider a radially unbounded quadratic Lyapunov function candidate in the form

$$V(e,\ \theta) = e^2 + (\theta - \theta_*)^2$$

and compute its time derivative along the system trajectories:

$$\begin{aligned} \dot{V}(e,\ \theta) &= 2\, e\, \dot{e} + 2\, (\theta - \theta_*)\, \dot{\theta} \\ &= 2\, e\, (-e + (\theta - \theta_*)\, \varphi(t)) + 2\, (\theta - \theta_*)\, (-e\, \varphi(t)) = -2\, e^2 \leq 0 \end{aligned}$$

So $V(x(t))$ is decreasing, as a function of time, and therefore, both $e(t)$ and $\theta(t)$ are uniformly bounded. Note that V will continue to decrease until $e \neq 0$. Since V is

lower-bounded and decreasing, it must tend to a limit, which may not necessarily be zero. If we can prove that in addition to the function having a limit its derivative tends to zero, we could argue that since the derivative is proportional to e^2, then e tends to zero as well. The property that relates functions with a limit and their derivatives is given by Barbalat's lemma. Essentially, the lemma states that if a time-dependent function tends to a limit and if its time derivative is uniformly continuous, then the derivative tends to zero. We shall formulate Barbalat's lemma and then return to complete the example. □

We begin with the definition of uniform continuity for a scalar function.

Definition 8.12. Uniform Continuity *A function* $f(t) : R \to R$ *is said to be uniformly continuous if*

$$\forall \varepsilon > 0 \quad \exists\, \delta = \delta(\varepsilon) > 0 \quad \forall\, |t_2 - t_1| \leq \delta \Rightarrow |f(t_2) - f(t_1)| \leq \varepsilon$$

Note that t_1 and t_2 play a symmetric role in the definition above. The uniform continuity concept should be compared to the definition of continuity at a point t, where $\delta = \delta(\varepsilon,\ t)$ becomes t-dependent.

It is not difficult to show that for a scalar continuously differentiable function to become uniformly continuous, it is sufficient to verify that the function derivative is bounded (Exercise 8.9). This fact becomes important during stability proofs for adaptive controllers.

We now state Barbalat's lemma, whose formal proof can be found in [2].

Lemma 8.1. (Barbalat) *Let* $f : R \to R$ *be a uniformly continuous function on* $[0,\ \infty)$. *Suppose that* $\lim_{t\to\infty} \int_0^t f(\tau)\, \mathrm{d}\tau$ *exists and is finite. Then,* $\lim_{t\to\infty} f(t) = 0$. ■

It is interesting to note that Barbalat's lemma is in some ways analogous to the well-known fact for converging infinite series $\sum_{k=1}^{\infty} a_k$, where $\lim_{k\to\infty} a_k = 0$ represents a necessary condition for the series to have a finite value.

For continuously differentiable functions, Barbalat's lemma can be restated as follows.

Lemma 8.2. *Let* $f : R \to R$ *be continuously differentiable on* $[0,\ \infty)$, *and suppose that* $\lim_{t\to\infty} f(t)$ *exists and is finite. If the function derivative* $\dot{f}(t)$ *is uniformly continuous on* $[0,\ \infty)$, *then* $\lim_{t\to\infty} \dot{f}(t) = 0$. ■

An immediate and a very practical corollary of Barbalat's lemma can now be stated.

Corollary 8.1. *If a scalar function* $f : R \to R$ *is twice continuously differentiable on* $[0,\ \infty)$ *and has a finite limit,* $\lim_{t\to\infty} f(t) < \infty$, *and the function second derivative is bounded, then* $\lim_{t\to\infty} \dot{f}(t) = 0$. ■

In general, the fact that derivative of a function tends to zero does not imply that the function itself has a limit. Also, the converse is not true. In fact, as the following examples show, there are no generic relations between functions and their derivatives.

Example 8.16 As $t \to \infty$, $f(t) = \sin(\ln t)$ does not have a limit, yet its derivative $\dot{f}(t) = \frac{\cos(\ln t)}{t}$ tends to zero. On the other hand, $f(t) = e^{-t}\sin(e^{2t})$ does tend to zero as $t \to \infty$. However, its derivative $\dot{f}(t) = -e^{-t}\sin(e^{2t}) + e^{t}\cos(e^{2t})$ tends to infinity. □

Example 8.15 (continued) Previously, we have shown that the time derivative of the Lyapunov function candidate $V(e, \theta) = e^2 + (\theta - \theta_*)^2$ along the system trajectories was negative semidefinite: $\dot{V}(e, \theta) = -2e^2 \leq 0$. The second time derivative of V is $\ddot{V}(e, \theta) = -4e\dot{e} = -4e(-e + (\theta - \theta_*)\varphi(t))$. Since $\varphi(t)$ is bounded by hypothesis, and $e(t)$and $\theta(t)$ were shown to be bounded, it is clear that $\ddot{V}(x(t))$ is uniformly bounded. Hence, $\dot{V}(x(t))$ is uniformly continuous. Also, it was shown that $V(x(t))$, as a function of time, tends to a limit. Then, by Barbalat's lemma, $\dot{V}(x(t)) \underset{t\to\infty}{\to} 0$, which in turn indicates that $e(t)$ tends to zero, as $t \to \infty$. □

We now return to completion of stability analysis for the system (8.38), where we have assumed that a Lyapunov function was found to satisfy the inequality in (8.39). We proceed with Lyapunov-based arguments. Since $V(x(t))$ is bounded from below and $\dot{V}(x(t)) \leq 0$, then the function has a limit, as $t \to \infty$, and the system state x is uniformly bounded. Next, we show that $\dot{V}(x(t))$ is uniformly continuous. We cannot differentiate an inequality such as (8.39). Instead, we integrate it from t_0 to t:

$$V(x(t)) - V(x(t_0)) \leq -\int_{t_0}^{t} W(x(\tau))\, d\tau \leq 0 \tag{8.41}$$

Rearranging terms gives

$$\int_{t_0}^{t} W(x(\tau))\, \mathrm{d}\tau \leq V(x(t_0)) - V(x(t)) \leq V(x(t_0)) < \infty \tag{8.42}$$

and, consequently,

$$\lim_{t\to\infty} \int_{t_0}^{t} W(x(\tau))\, d\tau < \infty \tag{8.43}$$

Since $x(t)$ is uniformly bounded and $f(t, x(t))$ is Lipschitz-continuous uniformly in x, then $x(t)$ is uniformly continuous in t. Moreover, in view of the fact that $W(x(t))$ is continuous in x, $W(x(t))$ becomes uniformly continuous in t. This property, coupled with (8.43), allows for direct application of Barbalat's lemma, which in this case states that $\lim_{t\to\infty} W(x(t)) = 0$. In other words, the system trajectories asymptotically approach the set E defined in (8.40), uniformly in time. We have just proved a special case of LaSalle–Yoshizawa theorem [1], with a Lyapunov function that did not explicitly depend on time.

Theorem 8.7. LaSalle–Yoshizawa *Starting anywhere in a domain D, all trajectories of the nonautonomous dynamics (8.38), with a Lyapunov function satisfying (8.39), uniformly asymptotically approach the set E from (8.40).* ■

Example 8.17 In adaptive control design, we will encounter n-dimensional nonautonomous systems in the form

$$\dot{e} = A\,e + b\,\underbrace{\left(\hat{K} - K\right)^T}_{\Delta K}\,\Phi(t)$$
$$\dot{\hat{K}} = -\gamma\,\Phi(t)\,e^T P\,b$$

where $e \in R^n$ is the system tracking error, $\hat{K} \in R^N$ is the adaptive n-dimensional vector of gains, γ is a constant positive adaptation rate, $K \in R^N$ is a constant vector of ideal (unknown) gains, $A \in R^{n\times n}$ is Hurwitz, $b \in R^n$ is a constant vector chosen such that the pair $(A,\ b)$ is controllable, and $\Phi(t) \in R^N$ is the so-called regressor vector, which is assumed to be uniformly bounded. Finally, $P \in R^{n\times n}$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation

$$P\,A + A^T P = -Q$$

with a symmetric positive-definite matrix Q.

These dynamics can be viewed as a generalization of the scalar system that was presented and analyzed in Example 8.15. Our immediate goal is to prove uniform boundedness of all signals and global uniform stability of the origin. Let us consider a quadratic radially unbounded Lyapunov function candidate in the form

$$V(e,\,\Delta K) = e^T P\,e + \Delta K^T\,\Delta K$$

and compute its time derivative along the system trajectories:

$$\begin{aligned}\dot{V}(e,\,\Delta K) &= \dot{e}^T P\,e + e^T P\,\dot{e} + 2\,\Delta K^T \dot{\hat{K}} = \\ &= \left(A\,e + b\,\Delta K^T\,\Phi\right)^T P\,e + e^T P\left(A\,e + b\,\Delta K^T\,\Phi\right) - 2\,\Delta K^T\,\Phi(e,\,r(t))\,e^T P\,b \\ &= -e^T Q\,e \le 0\end{aligned}$$

According to Lyapunov's direct method, this inequality implies global uniform stability of the origin, as well as uniform boundedness of $e(t)$ and $\Delta K(t)$. Then, because of the system dynamics, $\dot{e}(t)$ is also uniformly bounded, and so the second time derivative of the Lyapunov function

$$\ddot{V}(e,\,\Delta K) = -2\,e^T Q\,\dot{e}$$

is uniformly bounded. Therefore, $\dot{V}(e(t),\, \Delta K(t))$ is uniformly continuous in t. At the same time, since $V(e(t),\, \Delta K(t)) \geq 0$ and $\dot{V}(e(t),\, \Delta K(t)) \leq 0$, then the Lyapunov function itself tends to a limit. Lastly, applying Barbalat's lemma (in the form of Corollary 8.1) gives

$$\lim_{t\to\infty} \left[e^T(t)\, P\, e(t)\right] = \lim_{t\to\infty} \left[\dot{V}(e(t),\, \Delta K(t))\right] = 0$$

and, consequently $\lim_{t\to\infty} \|e(t)\| = 0$; that is, the system tracking error globally uniformly and asymptotically tends to the origin, while the rest of the signals remain uniformly bounded. In the forthcoming chapters, this key property will enable us to design stable robust adaptive controllers with predictable closed-loop performance. □

8.8 Summary and Historical Remarks

Theoretical foundations of stability theory for a general class of nonlinear differential equations were developed and published by Alexander Mikhailovich Lyapunov in his doctoral thesis on "the general problem of the stability of motion", which he defended at the University of Moscow in 1892. Lyapunov's stability, along with its extensions due to LaSalle, Yoshizawa, Barbashin, and Krasovskii, provided the necessary framework for the development of adaptive control. For dynamical systems without equilibrium, the notion of uniform ultimate boundedness was introduced and analyzed using Lyapunov's second method.

We would like to emphasize yet again that Barbalat's lemma constitutes the corner stone of proving stability for adaptive systems. This lemma allows to assert asymptotic stability of the system tracking error based on two facts: (a) The error is square integrable and (b) the error time derivative is uniformly bounded. Both statements come from application of Lyapunov's second method to examine stability of the system error dynamics. Barbalat's lemma has been independently derived by many authors, but the original work was attributed to Barbalat by V.M. Popov in his book "Hyperstability of Control Systems," published by Springer-Verlag in 1973.

8.9 Exercises

Exercise 8.1. Starting at different initial conditions, simulate the system dynamics in Example 8.1. Comment on the system behavior near and at the origin.

Exercise 8.2. Derive the system solution in Example 8.1 and prove the stated three properties.

Exercise 8.3. Derive the system solution in Example 8.4 and draw the system phase portrait. Given an initial condition, find a local Lipschitz constant. Prove that the system dynamics is not globally Lipschitz.

Exercise 8.4. Prove that trajectories of any scalar autonomous ODE (assuming that they exist) are monotonic functions of time.

Exercise 8.5. Prove the statement from Example 8.6.

Exercise 8.6. For a scalar nonautonomous differential equation in the form $\dot{x} = -a(t)\,x$, define sufficient conditions on $a(t)$, so that the equilibrium of the scalar dynamics is (a) stable, (b) asymptotically stable, and (c) uniformly asymptotically stable.

Exercise 8.7. Consider the system

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -g(x_1)\,(x_1 + x_2)$$

where g is locally Lipschitz and $g(y) \geq 1$ for all $y \in R$. Verify that

$$V(x) = \int_0^{x_1} y\, g(y)\, \mathrm{d}y + x_1\, x_2 + x_2^2$$

is globally positive definite and radially unbounded. Use $V(x)$ to show that the system equilibrium point $x_* = 0$ is globally asymptotically stable.

Exercise 8.8. There are theoretical extensions that deal with existence and uniqueness of IVP-s whose system dynamics are discontinuous in x. Show that the IVP

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_2 - \mathrm{sgn}\,(x_1 + x_2) \end{cases}, \; x(0) = x_0$$

does not satisfy the sufficient conditions for existence and uniqueness of its solution. Nevertheless, a solution does exist. Simulate the system starting from different initial conditions. Construct phase portrait of the system and argue that (a) the manifold $c(x) = x_1 + x_2 = 0$ is the system global attractor, (b) all system trajectories reach this manifold in finite time, and (c) the solution "slides" down the manifold toward the origin.

Exercise 8.9. Prove that a scalar continuously differentiable function is uniformly continuous if the function derivative is bounded. Using this fact, prove Corollary 8.1.

Exercise 8.10. Consider the n-dimensional LTI controllable system

$$\dot{x} = A\,x + b\left(u - K_x^T\,x\right)$$

with a Hurwitz matrix A. Suppose that $K_x \in R^{n\times 1}$ is constant and unknown. Let $r(t) \in R$ denote a bounded external command for the system output $y = C\,x$ to follow. The system control input u is chosen as

$$u = \hat{K}(t)\,x + K_r\,r(t)$$

where $K_r = -C\,A^{-1}\,b$ is the command feedforward gain, and $\hat{K}(t)$ is the time-variant state feedback gain, whose dynamics are given by the adaptive laws,

$$\dot{\hat{K}} = -\gamma\,x\,e^T P\,b$$

with a positive scalar γ and using the unique positive-definite solution $P \in R^{n\times n}$of the Lyapunov algebraic equation $P\,A + A^T\,P = -Q$, where $Q \in R^{n\times n}$ is symmetric and positive definite. Let

$$\dot{x}_{ref} = A\,x_{ref} + b\,r$$

define the desired dynamics. The system tracking error is $e = x - x_{ref}$. Write down the tracking error dynamics. Formulate the total closed-loop dynamics by combining the tracking error dynamics with the adaptive laws. Prove that for any bounded command r, any constant positive adaptation rate γ, and any symmetric positive-definite matrix Q, the tracking error $e(t)$ tends to zero globally and asymptotically; that is, the system state x tracks the desired state x_{ref},with diminishing errors. Argue that in this case, the system output $y = C\,x$ tracks the command $r(t)$ with bounded errors and in the presence of any constant uncertain vector parameter K_x (hint: Use the stability arguments from Example 8.17).

Exercise 8.11. Using the system and control equations from Example 8.10, choose $n = 2$, select your own data, and simulate the corresponding closed-loop system dynamics. Verify the theoretical predictions of stability and tracking while driving the desired dynamics with various external bounded commands. Demonstrate (via simulation tests) the closed-loop system tracking performance in the presence of a constant uncertain parameter K_x. Discuss your results.

References

1. Haddad, W.M., Chellaboina, V.: Nonlinear Dynamical Systems and Control. A Lyapunov-Based Approach, p. 08540. Princeton University Press, Princeton (2008)
2. Khalil, H.: Nonlinear Systems, 3rd edn, p. 07458. Prentice Hall, Upper Saddle River (1996)
3. Vidyasagar, M.: Nonlinear Systems Analysis, p. 07632. Prentice Hall, New Jersey (1993)

Chapter 9
State Feedback Direct Model Reference Adaptive Control

9.1 Introduction

For over 50 years, adaptive systems have decisively remained in the mainstream of controls and dynamics research. As a result, adaptive control has grown to become a well-formed scientific discipline. One of the reasons for the continuing popularity and rapid growth of adaptive control is its clearly defined goal – to enable control of dynamical systems that operate in the presence of unknown parameters.

Adaptive control research was initiated in the early 1950s. At that time, the interest in adaptive systems was primarily driven by the design of autopilots for high-performance aircraft. This was no surprise since newly designed aerial platforms required control solutions that would provide stable and predictable flight operations throughout the aircraft's vast envelope, ranging from subsonic to supersonic and even to hypersonic regions.

The last decade has witnessed the development of a coherent theory for adaptive control, which has led to many practical applications in the areas such as aerospace, robotics, chemical processes, ship steering, bioengineering, and many others.

A few historical remarks are in order. The original concept of a model reference adaptive system was proposed by Whitaker et al. in [1, 2]. The main idea behind this concept was to specify the desired command-to-output performance of a servo-tracking system that would eventually define the ideal response of the system output due to external commands. This control concept was later called the "explicit model following," and the corresponding architecture became known as the model reference adaptive control, or in short MRAC. Soon after its introduction, the first proof of MRAC closed-loop stability using Lyapunov theory was given in 1965 by Butchart and Shackcloth [3] and also in 1966 by Parks [4]. In the years that followed, adaptive control theory for a broad class of dynamical uncertain systems was developed and well documented in several now-classical textbooks [5–8].

A generic block diagram of a system (plant) operating under MRAC controller is shown in Fig. 9.1.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_9,

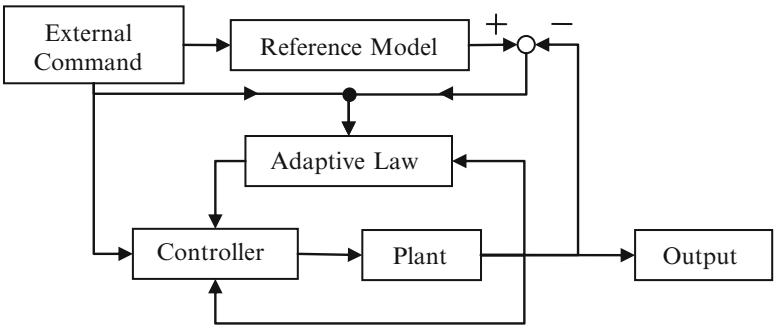


Fig. 9.1 An MRAC closed-loop block diagram

In essence, a MRAC system consists of a controller whose parameters (gains) are updated online using an adaptive law. The latter operates on the system output and on an external command (a.k.a. the reference input). The command also drives the reference model that specifies the desired trajectories for the system to follow. The difference between the reference model output and the system output constitutes the tracking error, which subsequently is sent to the adaptive law for online parameter adjustments. Finally, the controller computes its commands based on the reference input, the system output, and the online adjusted parameters from the adaptive law. Per design, the adaptive controller forces the system output to follow the desired external commands while operating in the presence of the plant uncertainties. For itself, the controller main objective is to maintain consistent performance of the closed-loop system in the presence of uncertainties and unknown variations in plant parameters.

When the true plant parameters are unknown, one might attempt to estimate control gains online using available measurements. This approach is referred to as the "direct." Alternatively, the gains can be approximated online by solving system design equations that relate the plant uncertainties to the known signals in the system. This is called the "indirect" method. MRAC systems can be designed using either direct or indirect approaches. There are also design methods available that merge the two, leading to combined (direct + indirect) MRAC architectures.

In this chapter, our focus will be on the design, analysis, and evaluation of direct MRAC systems for continuous plants with uncertain dynamics and full state measurements.

9.2 Command Tracking

We shall consider command tracking algorithms for continuous dynamic plants

$$\begin{aligned} \dot{x} &= f(t, x, u, \Theta, \xi) \\ y &= h(t, x, u, \Theta, \xi) \end{aligned} \tag{9.1}$$

with vector-parametric constant uncertainties Θ and with bounded environmental disturbances $\xi(t)$. In (9.1), $x \in R^n$ denotes the system state, $u \in R^m$ is the control input, and $y \in R^m$ is the regulated output. It is assumed that the entire system state vector x is available for control synthesis. In other words, the system state can be measured online.

The problem of tracking a command involves the design of the system control input u so that the regulated output $y(t)$ tracks a given bounded reference signal $r(t) \in R^m$, in the presence of the system uncertainties Θ and environmental disturbances $\xi(t)$. Specifically, we are looking for a control input that would force the output tracking error

$$e_y(t) = y(t) - r(t) \tag{9.2}$$

to become sufficiently small, as $t \to \infty$. Moreover, it is required that during tracking, all the signals in the corresponding closed-loop system remain uniformly bounded in time.

If $e_y(t) \underset{t\to\infty}{\to} 0$, then we assert that asymptotic output tracking has been achieved. In general, asymptotic tracking may not be feasible, and in that case, our goal would be to achieve uniform ultimate boundedness of the tracking error

$$\left\| e_y(t) \right\| \le \varepsilon, \quad \forall\, t \ge T \tag{9.3}$$

where $\varepsilon > 0$ is the desired tracking tolerance, T is a finite time instant, and $\|\bullet\|$ denotes a vector norm. A brief review of vector norms was given in Chap. 1.

9.3 Direct MRAC Design for Scalar Systems

We begin with a scalar plant whose dynamics are of the form

$$\dot{x} = a\,x + b\,(u + f(x)) \tag{9.4}$$

where x is the system state and u is the control input, while a and b represent unknown constant parameters. We assume that the sign of b is known, which is equivalent to saying that the system is controllable. The system dynamics depend on the unknown function $f(x)$ defined as a linear combination of N known basis functions $\varphi_i(x)$ with N unknown constants θ_i

$$f(x) = \sum_{i=1}^{N} \theta_i\, \varphi_i(x) = \theta^T\, \Phi(x) \tag{9.5}$$

where $\Phi(x) = (\,\varphi_i(x) \quad \ldots \quad \varphi_i(x)\,)^T \in R^N$ denotes the known regressor vector, whose components $\varphi_i(x)$ are assumed to be Lipschitz-continuous in x. So, the scalar model we consider here is

$$\dot{x} = a\,x + b\left(u + \theta^T \Phi(x)\right) \tag{9.6}$$

A stable reference model is given. Its dynamics are described by a first-order differential equation in the form

$$\dot{x}_{ref} = a_{ref}\,x_{ref} + b_{ref}\,r(t) \tag{9.7}$$

where $a_{ref}<0$ and b_{ref} are the desired constants and $r(t)$ is the reference input command. The reference model parameters must be chosen to represent the desired response due to bounded commands. For example, the designer may select $b_{ref} = -a_{ref}$ so that the DC gain of the reference dynamics becomes unity, and then select a_{ref} such that the reference system time constant is as small as desired.

The control objective of interest is to asymptotically track the state x_{ref} of the reference model (9.7), which can be driven by any bounded command $r(t)$. In other words, we need to design a control law $u(t)$, such that the state tracking error $e(t) = x(t) - x_{ref}(t)$ globally uniformly asymptotically tends to zero, as $t \to \infty$, while all signals in the corresponding closed-loop system remain uniformly ultimately bounded in time.

The required command tracking task must be accomplished in the presence of $(N + 2)$ unknown constant parameters $\{a,\ b,\ \theta_1,\ \ldots\ ,\ \theta_N\}$.

First, we define the "ideal" control solution, as if the unknown parameters were known. The ideal control is composed using the (feedback + feedforward) architecture

$$u_{ideal} = k_x\,x + k_r\,r - \theta^T\,\Phi(x) \tag{9.8}$$

where k_x and k_r represent the ideal feedback and feedforward gains, respectively. Substituting (9.8) into (9.6) gives the system closed-loop dynamics:

$$\dot{x} = (a + b\,k_x)\,x + b\,k_r\,r(t) \tag{9.9}$$

Comparing (9.9) with the desired reference model dynamics (9.7), it follows that the ideal gains k_x and k_r must satisfy the following two algebraic equations:

$$a + b\,k_x = a_{ref} \quad b\,k_r = b_{ref} \tag{9.10}$$

These relations are called the matching conditions. It is clear that for scalar plants, the unknown ideal gains, k_x and k_r, in (9.10) always exist. As we shall see later, this will not be the case for multidimensional dynamics.

Based on (9.8), we propose a tracking control solution in the form

$$u = \hat{k}_x\,x + \hat{k}_r\,r - \hat{\theta}^T\,\Phi(x) \tag{9.11}$$

where the feedback gain $\hat{k}_x$, the feedforward gain $\hat{k}_r$, and the estimated vector of parameters $\hat{\theta}$ will be determined to achieve global uniform asymptotic tracking of the reference model trajectories. Toward that end, we substitute (9.11) into the system dynamics (9.6)

$$\dot{x} = \left(a + b\,\hat{k}_x\right) x + b\left(\hat{k}_r\, r - \left(\hat{\theta} - \theta\right)^T \Phi(x)\right) \tag{9.12}$$

and rewrite the latter using the matching conditions (9.10)

$$\dot{x} = a_{ref}\, x + \underbrace{b\, k_r}_{b_m}\, r + b\,\underbrace{\left(\hat{k}_x - k_x\right)}_{\Delta k_x} x + b\,\underbrace{\left(\hat{k}_r - k_r\right)}_{\Delta k_r} r - b\,\underbrace{\left(\hat{\theta} - \theta\right)^T}_{\Delta\theta} \Phi(x) \tag{9.13}$$

where

$$\Delta k_x = \hat{k}_x - k_x, \quad \Delta k_r = \hat{k}_r - k_r, \quad \Delta\theta = \hat{\theta} - \theta \tag{9.14}$$

denote parameter estimation errors. Then, the closed-loop dynamics of the system tracking error signal

$$e(t) = x(t) - x_{ref}(t) \tag{9.15}$$

can be obtained by subtracting (9.7) from (9.13):

$$\dot{e}(t) = a_{ref}\, e + b\left(\Delta k_x\, x + \Delta k_r\, r - \Delta\theta^T \Phi(x)\right) \tag{9.16}$$

We are going to choose adaptive gains $\left(\hat{k}_x,\ \hat{k}_r,\ \hat{\theta}\right)$ to enforce global uniform asymptotic stability of the origin. This will be accomplished through the inverse Lyapunov design approach, where we would choose a Lyapunov function candidate and then select adaptive laws such that the function time derivative becomes nonpositive, when evaluated along the trajectories of the error dynamics (9.16). As a result, the tracking error would asymptotically converge to the origin, and so the system state would asymptotically track the state of the reference model.

Let us consider a quadratic Lyapunov function candidate in the form

$$V(e,\ \Delta k_x,\ \Delta k_r,\ \Delta\theta) = e^2 + |b|\left(\gamma_x^{-1}\,\Delta k_x^2 + \gamma_r^{-1}\,\Delta k_r^2 + \Delta\theta^T\,\Gamma_\theta^{-1}\,\Delta\theta\right) \tag{9.17}$$

where scalars $\gamma_x > 0$, $\gamma_r > 0$, and a constant symmetric positive-definite matrix $\Gamma_\theta \in R^{N\times N}$ are the rates of adaptation. Taking the time derivative of V, along the trajectories of (9.16), gives

$$\begin{aligned}\dot{V}(e,\,\Delta k_x,\,\Delta k_r,\,\Delta\theta) &= 2\,e\,\dot{e} + 2\,|b|\Big(\gamma_x^{-1}\,\Delta k_x\,\dot{\hat{k}}_x + \gamma_r^{-1}\,\Delta k_r\,\dot{\hat{k}}_r + \Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\Big)\\ &= 2\,e\big(a_{\text{ref}}\,e + b\,\left(\Delta k_x\,x + \Delta k_r\,r - \Delta\theta^T\,\Phi(x)\right)\big)\\ &\quad + 2\,|b|\Big(\gamma_x^{-1}\,\Delta k_x\,\dot{\hat{k}}_x + \gamma_r^{-1}\,\Delta k_r\,\dot{\hat{k}}_r + \Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\Big)\\ &= 2\,a_{ref}\,e^2 + 2\,|b|\Big(\Delta k_x\,\Big(x\,e\,\text{sgn}(b) + \gamma_x^{-1}\,\dot{\hat{k}}_x\Big)\Big)\\ &\quad + 2\,|b|\Big(\Delta k_r\,\Big(r\,e\,\text{sgn}(b) + \gamma_r^{-1}\,\dot{\hat{k}}_r\Big)\Big)\\ &\quad + 2\;|b|\,\Delta\theta^T\Big(-\Phi(x)\,e\,\text{sgn}(b) + \Gamma_\theta^{-1}\,\dot{\hat{\theta}}\Big)\end{aligned} \tag{9.18}$$

In order to enforce closed-loop stability, it is sufficient to choose adaptive laws such that $\dot{V}(e,\,\Delta k_x,\,\Delta k_r,\,\Delta\theta) \leq 0$. Indeed, if we select

$$\begin{aligned}\dot{\hat{k}}_x &= -\gamma_x\,x\,e\,\text{sgn}(b)\\ \dot{\hat{k}}_r &= -\gamma_r\,r\,e\,\text{sgn}(b)\\ \dot{\hat{\theta}} &= \Gamma_\theta\,\Phi(x)\,e\,\text{sgn}(b)\end{aligned} \tag{9.19}$$

then the time derivative of V, computed along the trajectories of (9.16), becomes negative semidefinite

$$\dot{V}(e,\;\Delta k_x,\;\Delta k_r,\;\Delta\theta) = 2\underbrace{a_{ref}}_{<0}\,e(t)^2 \leq 0 \tag{9.20}$$

which immediately implies that the signals $(e,\;\Delta k_x,\;\Delta k_r,\;\Delta\theta)$ are uniformly bounded in time. The latter, coupled with the fact that $(x_{\text{ref}},\;r)$ are bounded and θ is a constant vector, means that the system state x and the estimated vector of parameters $\hat{\theta}$ are uniformly bounded. Moreover, since the components $\varphi_i(x)$ of the regressor vector $\Phi(x)$ are Lipschitz-continuous functions of x, which was proven to be bounded, then the regressor components themselves are uniformly bounded. Hence, the control signal u in (9.11) is uniformly bounded as well. Consequently, both $\dot{x}$ and $\dot{x}_{\text{ref}}$ are uniformly bounded.

Differentiating (9.20) results in

$$\ddot{V}(e,\;\Delta k_x,\;\Delta k_r,\;\Delta\theta) = 4\,a_{ref}\,e(t)\,\dot{e}(t) \tag{9.21}$$

Therefore, $\ddot{V}$ is bounded, and consequently, $\dot{V}$ is a uniformly continuous function of time. Since V is lower bounded and $\dot{V}$ is negative semidefinite, then V, as a function of time, must have a finite limit. We can now use Barbalat's lemma to arrive at

$$\lim_{t\to\infty}\dot{V}(t) = 0 \tag{9.22}$$

Table 9.1 Direct MRAC design summary for a scalar plant

Open-loop plant	$\dot{x} = a\,x + b\left(u + \theta^T \Phi(x)\right)$
Reference model	$\dot{x}_{ref} = a_{ref}\,x_{ref} + b_{ref}\,r$
Tracking error	$e = x - x_{ref}$
Control input	$u = \hat{k}_x\,x + \hat{k}_r\,r - \hat{\theta}^T\,\Phi(x)$
Direct MRAC laws	$\dot{\hat{k}}_x = -\gamma_x\,x\,e\,\mathrm{sgn}(b)$ $\dot{\hat{k}}_r = -\gamma_r\,r\,e\,\mathrm{sgn}(b)$ $\dot{\hat{\theta}} = \Gamma_\theta\,\Phi(x)\,e\,\mathrm{sgn}(b)$

and because of (9.20), we conclude that the tracking error $e(t)$ tends to zero asymptotically, as $t \to \infty$.

Since the Lyapunov function (9.17) is radially unbounded and it does not depend explicitly on time, the attained asymptotic stability property is global and uniform, that is, the closed-loop tracking error dynamics are globally uniformly asymptotically stable. The command tracking problem is solved. We now recap our formally proven results in the Theorem 9.1.

Theorem 9.1. *For the uncertain scalar dynamical system (9.6), with the controller (9.11), and the adaptive laws (9.19), the system state $x(t)$ asymptotically tracks the state $x_{\mathrm{ref}}(t)$ of the reference model (9.7), driven by any bounded command $r(t)$, while all the signals in the closed-loop system remain uniformly bounded. Moreover, the closed-loop tracking error dynamics (9.16) are globally uniformly asymptotically stable.* ■

MRAC design equations for a scalar plant are summarized in the Table 9.1.

It is necessary to make a remark about dynamic behavior of the estimated parameters $\hat{\theta}$. The fact that the system tracking error e asymptotically tends to zero does not automatically imply that $\hat{\theta}$ converges to its ideal unknown parameter vector θ. What is certain is that the estimated parameters will remain uniformly bounded during tracking. Nevertheless, there are cases when parameter convergence will take place alongside the desired tracking. A sufficient condition for parameter convergence is given by the persistency of excitation (PE) [7–8], which imposes certain restrictions on the commanded signal $r(t)$. We shall define and discuss PE conditions at a later time.

As in any other control design method, MRAC has its own "tuning knobs." They are the rates of adaptations, represented by two positive constants $(\gamma_x,\ \gamma_r)$ and a symmetric positive-definite matrix Γ_θ. As seen from (9.19), the larger the rates, the faster the adaptive laws will evolve. One may conjecture that large rates would result in better and faster closed-loop tracking performance. This is partially true. Indeed, large rates of adaptation will yield fast tracking. However, this will also

lead to undesirable oscillations during transient times, when the system regulated output is trying to get closer to its command. The trade-off between fast tracking and smooth transients presents a design challenge.

Example 9.1 Helicopter Pitch Dynamics and Control During Hover Unlike a fixed-wing airplane, angular motion control of a helicopter is achieved by tilting its main rotor and as a result altering the direction of the rotor thrust vector. This action induces a change in angular moments acting on the vehicle and results in pitch, roll, and yaw angular motion.

In hover, the helicopter pitch dynamics depend primarily on the vehicle pitch rate q and on the applied (by a pilot or an automatic system) longitudinal control input δ, which is equivalent in its effect (also induces a pitching motion) to an elevator for a fixed-wing vehicle. Assuming constant thrust, while neglecting small forward and vertical speed components, pitch dynamics of a helicopter during hover can be approximated by the following scalar differential equation:

$$\dot{q} = M_q\, q + M_\delta\, (\delta + f(q))$$

where M_q represents the vehicle pitch damping and M_δ is the elevator effectiveness. The system also depends on the unknown function $f(q)$, which models inherent uncertainties in the helicopter dynamics, both linear and nonlinear.

For simulation purposes, we assume model parameters (unknown constants) that are representative of a hovering transport helicopter: $M_q = -0.61$ (rad/s) and $M_\delta = -6.65$ (rad/s^2). We also define

$$f(q) = \underbrace{-0.01}_{\theta}\ \underbrace{\tanh\left(\frac{360}{\pi}\, q\right)}_{\Phi(q)} = \theta\ \Phi(q)$$

where $\theta = -0.01$ is unknown and $\Phi(q)$ is the known regressor, and arrive at the helicopter dynamics:

$$\dot{q} = -0.61\, q - 6.65\left(\delta - 0.01\ \tanh\left(\frac{360}{\pi}\, q\right)\right)$$

Clearly, the origin of the open-loop $(\delta = 0)$ pitch dynamics becomes locally unstable, as shown in Fig. 9.2.

Such a system would certainly require active control for stabilization and command tracking.

Our particular selection of the system parameters in this example is purely academic. It merely supports the main objective here – to design an MRAC system and to show its efficacy in coping with linear and nonlinear uncertainties of various forms and shapes. Toward that end, we use MRAC design equations from Table 9.1 and construct the following adaptive pitch controller:

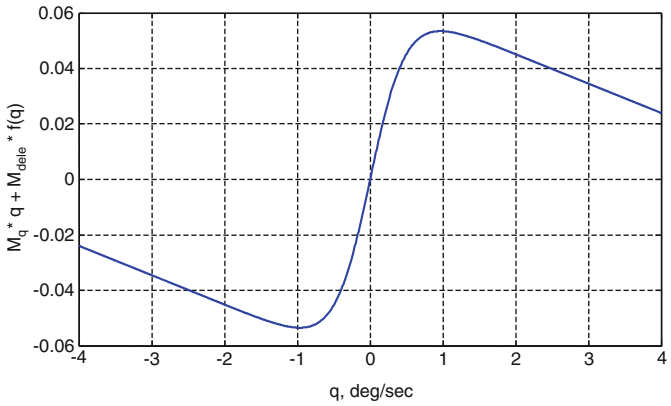


Fig. 9.2 Helicopter open-loop pitch dynamics from Example 9.1

$$\delta = \hat{k}_q\, q + \hat{k}_{q_{cmd}}\, q_{cmd} - \hat{\theta}^T\, \Phi(q)$$

with the adaptive laws

$$\begin{aligned} \dot{\hat{k}}_q &= \gamma_q\, q\, (q - q_{ref}) \\ \dot{\hat{k}}_{q_{cmd}} &= \gamma_{q_{cmd}}\, q_{cmd}\, (q - q_{ref}) \\ \dot{\hat{\theta}} &= -\Gamma_\theta\, \Phi(q)\, (q - q_{ref}) \end{aligned}$$

where q_{ref} is the desired pitch rate signal generated by the reference model

$$\dot{q}_{ref} = 4\, (q_{cmd} - q_{ref})$$

driven by any bounded time-varying pitch rate command $q_{cmd} = q_{cmd}(t)$. Here, we have selected $a_{ref} = -b_{ref} = -4$.

After several design iterations, we have chosen the rates of adaptation to be $\gamma_q = \gamma_{q_{cmd}} = 6000,\ \Gamma_\theta = 8$. Figure 9.3 shows the closed-loop system response (pitch rate, deg/s) and the required control effort (elevator deflection, deg) for tracking a series of step-input commands of different magnitudes.

During this event, the adaptive parameters (solid green, Fig. 9.4) remain bounded and approach their true unknown values (dashed blue, Fig. 9.4).

The observed parameter convergence is not guaranteed to always take place. For example, suppose that the same system is required to track a sinusoidal command (Fig. 9.5).

In this case, the adaptive parameters do not converge to their ideal values (Fig. 9.6). However, they do remain uniformly bounded, as expected.

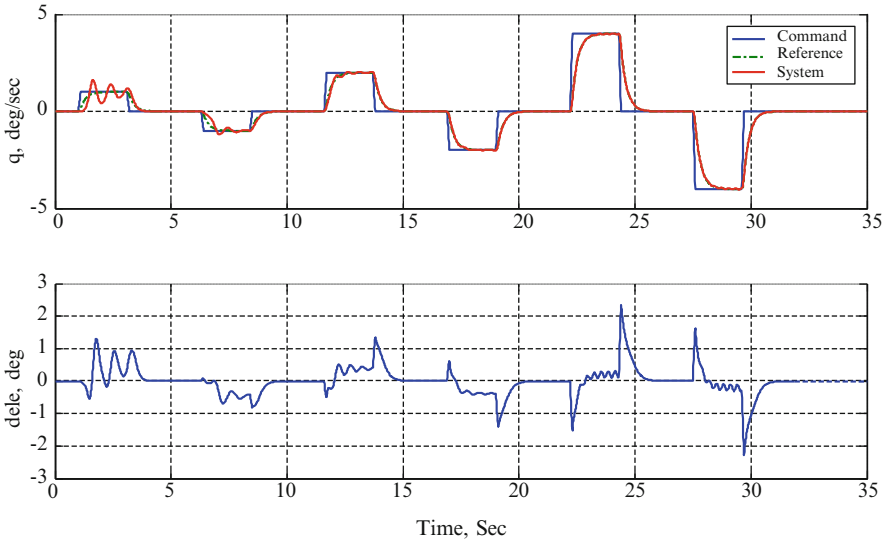


Fig. 9.3 Closed-loop pitch rate command tracking performance in Example 9.1

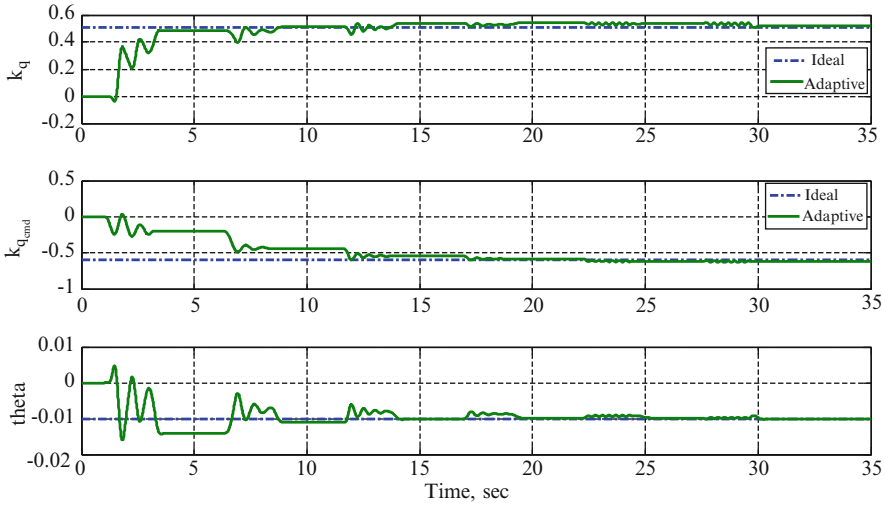


Fig. 9.4 Adaptive parameters from Example 9.1 converge to their ideal values

It is also interesting to compare the adaptive elevator input (Fig. 9.3) against the ideal signal generated by the fixed-gain controller:

$$\delta_{ideal} = \frac{1}{M_\delta}\left(\left(a_{ref} - M_q\right) q + b_{ref}\, q_{cmd}\right) - f(q)$$

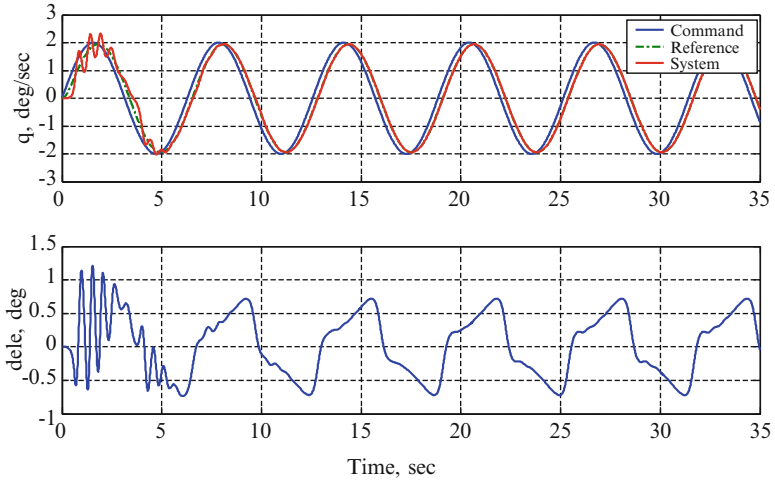


Fig. 9.5 Sinusoidal command tracking from Example 9.1

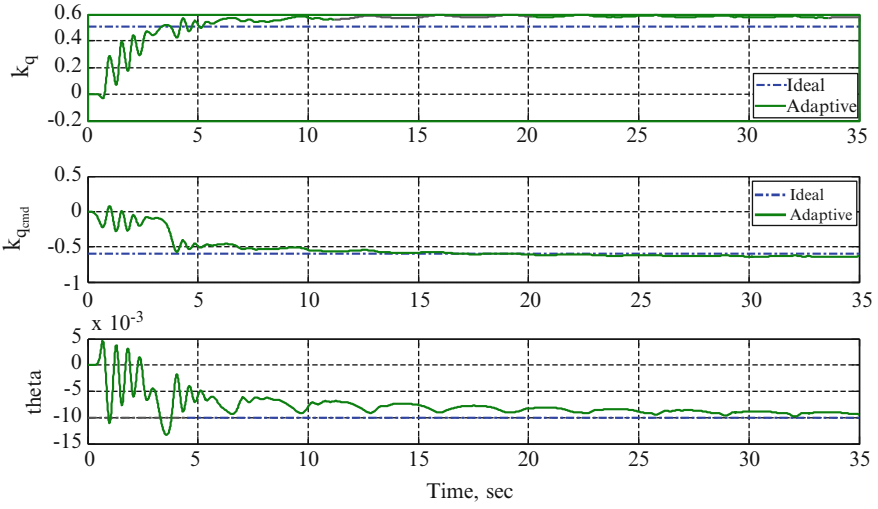


Fig. 9.6 Adaptive parameters during sinusoidal command tracking from Example 9.1

Fig. 9.7 shows the comparison data.

In spite of clearly visible similarities, the two control signals also exhibit subtle differences. Even after the transients have subsided, the MRAC signal has a tendency to oscillate (Fig. 9.8), while the fixed-gain controller does not.

Summarizing our discussions, we end this example with an observation that in the two simulation scenarios considered, the pitch rate MRAC system was able to provide adequate closed-loop command tracking performance while operating in the presence of linear and nonlinear uncertainties. □

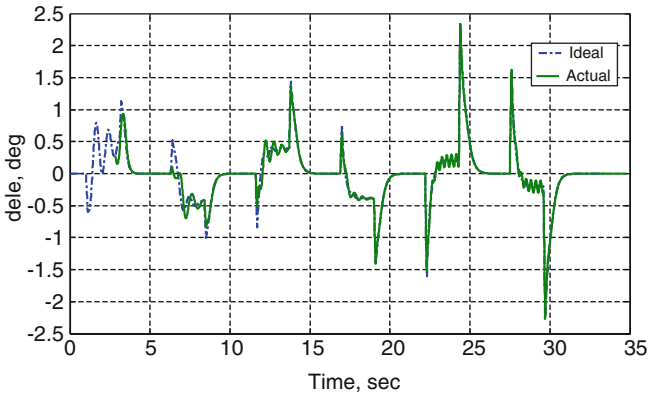


Fig. 9.7 Adaptive and ideal fixed-gain controllers from Example 9.1

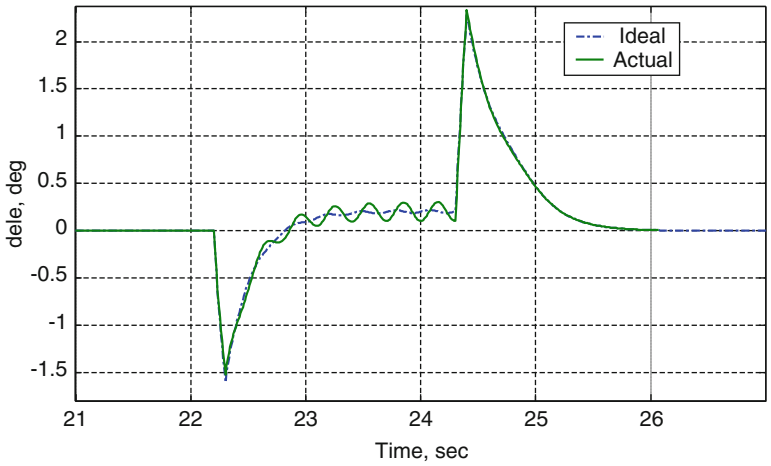


Fig. 9.8 MRAC signal oscillations in Example 9.1

9.4 Dynamic Inversion MRAC Design for Scalar Systems

Dynamic inversion (DI) control for systems with known dynamics represents a well-known method [9]. In this section, we will demonstrate the design of a DI-based MRAC system for the scalar uncertain dynamical system

$$\dot{x} = a\,x + b\,u + \underbrace{\theta^T \Phi(x)}_{f(x)} \tag{9.23}$$

with two unknown constants $(a,\ b)$ and with an unknown function $f(x)$ in the form of (9.5). Again we assume that the constant vector of ideal parameters θ is not known, while the regressor components $\varphi_i(x)$ represent a known set of Lipschitz-continuous basis functions. We also assume that sgn$b$ is known and that the system is controllable, that is, $|b| \geq b_{\min}>0$, where $b_{\min}$ represents a known lower bound of $|b|$.

The reference model dynamics are given by (9.7), and the design task remains the same – find a control input u to force the system state x asymptotically track the state of the reference model, which is in turn driven by any bounded time-varying command r.

This particular control problem was addressed and solved in the previous section, where we derived a direct MRAC system. Here, we shall present an alternative solution and then compare the two approaches.

We begin by rewriting the system dynamics

$$\dot{x} = \hat{a}\,x + \hat{b}\,u + \hat{f}(x) - \underbrace{(\hat{a}-a)}_{\Delta a}\,x - \underbrace{(\hat{b}-b)}_{\Delta b}\,u - \underbrace{\left(\hat{f}(x)-f(x)\right)}_{\Delta f(x)} \tag{9.24}$$

where $\hat{a}$, $\hat{b}$ are the estimated values and

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\theta}_i\,\varphi_i(x) = \hat{\theta}^T\,\Phi(x) \tag{9.25}$$

is the function approximator. All these quantities will be constructed during the design process. Also, in (9.24), Δa, Δb, and $\Delta f(x)$ represent the parameter and the function approximation errors, respectively. Using (9.25) gives the function approximation error:

$$\Delta\hat{f}(x) = \hat{f}(x) - f(x) = \sum_{i=1}^{N} \underbrace{\left(\hat{\theta}_i - \theta_i\right)}_{\Delta\theta_i}\,\varphi_i(x) = \Delta\theta^T\,\Phi(x) \tag{9.26}$$

Following the DI method, let us consider a controller in the form:

$$u = \frac{1}{\hat{b}}\left(\left(a_{ref} - \hat{a}\right)x + b_{ref}\,r - \hat{\theta}^T\,\Phi(x)\right) \tag{9.27}$$

Substituting (9.27) into the second term of (9.24) yields

$$\dot{x} = a_{ref}\,x + b_{ref}\,r - \Delta a\,x - \Delta b\,u - \Delta\theta^T\,\Phi(x) \tag{9.28}$$

With the tracking error signal e defined in (9.15), we can now compute the system tracking error dynamics. Subtracting (9.7) from (9.28) gives

$$\dot{e} = a_{ref}\, e - \Delta a\, x - \Delta b\, u - \Delta\theta^T\, \Phi(x) \tag{9.29}$$

Consider the following Lyapunov function candidate:

$$V(e,\ \Delta a,\ \Delta b,\ \Delta\theta) = e^2 + \gamma_a^{-1}\, \Delta a^2 + \gamma_b^{-1}\, \Delta b^2 + \Delta\theta^T\, \Gamma_\theta^{-1}\, \Delta\theta \tag{9.30}$$

where $\gamma_a > 0, \gamma_b > 0, \Gamma_\theta = \Gamma_\theta^T > 0$ are the adaptation rates. The time derivative of V, evaluated along the trajectories of the error dynamics (9.29), can be computed as

$$\begin{aligned}
\dot{V}(e,\ \Delta a,\ \Delta b,\ \Delta\theta) =& 2\, e\, \dot{e} + 2\Big(\gamma_a^{-1}\, \Delta a\, \dot{\hat{a}} + \gamma_b^{-1}\, \Delta b\, \dot{\hat{b}} + \Delta\theta^T\, \Gamma_\theta^{-1}\, \dot{\hat{\theta}}\Big) \\
=& 2\, e\big(a_{ref}\, e - \Delta a\, x - \Delta b\, u - \Delta\theta^T\, \Phi(x)\big) \\
&+ 2\Big(\gamma_a^{-1}\, \Delta a\, \dot{\hat{a}} + \gamma_b^{-1}\, \Delta b\, \dot{\hat{b}} + \Delta\theta^T\, \Gamma_\theta^{-1}\, \dot{\hat{\theta}}\Big) \\
=& 2\, a_{ref}\, e^2 + \Delta a\big(\gamma_a^{-1}\, \dot{\hat{a}} - x\, e\big) + \Delta b\Big(\gamma_b^{-1}\, \dot{\hat{b}} - u\, e\Big) \\
&+ \Delta\theta^T\Big(\Gamma_\theta^{-1}\, \dot{\hat{\theta}} - \Phi(x)\, e\Big)
\end{aligned} \tag{9.31}$$

Based on (9.31) and in order to make $\dot{V} \leq 0$, the adaptive laws are chosen as

$$\dot{\hat{a}} = \gamma_a\, x\, e, \quad \dot{\hat{b}} = \gamma_b\, u\, e, \quad \dot{\hat{\theta}} = \Gamma_\theta\, \Phi(x)\, e \tag{9.32}$$

Then,

$$\dot{V}(e,\ \Delta a,\ \Delta b,\ \Delta\theta) = 2\, a_{ref}\, e^2 \leq 0 \tag{9.33}$$

and consequently, the four signals $(e,\ \Delta a,\ \Delta b,\ \Delta\theta)$ are uniformly bounded. Since $r(t)$ is bounded and $a_{ref} < 0$, then x_{ref} is also uniformly bounded and because of that the system state x as well as the three estimated signals $\Big(\hat{a},\ \hat{b},\ \hat{\theta}\Big)$ are uniformly bounded.

In order to claim uniform boundedness of u from (9.27), we need to protect the controller from "blowing up" due to the division by $\hat{b}$. In other words, we need to modify the adaptive laws (9.32) and enforce boundedness of the estimated parameter $\hat{b}$.

Let us consider the following modification of the second equation in the adaptive laws (9.32):

$$\dot{\hat{b}} = \begin{cases} \gamma_b\, u\, e, & \text{if } \big|\hat{b}\big| > b_{\min} \vee \big[\hat{b} = b_{\min} \text{sgn}\, b \wedge (u\, e)\, \text{sgn}\, b > 0\big] \\ 0, & \text{if } \big|\hat{b}\big| = b_{\min} \wedge (u\, e)\, \text{sgn}\, b < 0 \end{cases}. \tag{9.34}$$

The main motive here is to stop adaptation of $\hat{b}$ if the parameter reaches its lower absolute limit value $b_{\min}$, with a nonzero time derivative $\dot{\hat{b}}$. In this case, we would

prevent the estimated parameter $\hat{b}$ from crossing the known lower absolute value bound $b_{\min}$.

Let us argue that the modification (9.34) does indeed prevent $\hat{b}$ from crossing its allowable bound and at the same time, it preserves closed-loop system stability.

Suppose that $\text{sgn} b > 0$. Then, according to (9.34), it is easy to see that starting with any initial condition $\hat{b}(0) > b_{\min}$, the estimated parameter will satisfy the desired lower bound relation $\hat{b}(t) \geq b_{\min}$, for all future times. In addition, we must verify that the proposed modification (9.34) does not adversely affect closed-loop stability of the tracking error dynamics (9.29). In particular, we need to ensure that the inequality (9.33) remains in effect. For this to be true, it is sufficient to show that

$$\Delta b \left(\gamma_b^{-1} \dot{\hat{b}} - u\, e \right) \leq 0 \tag{9.35}$$

Let us argue that with the adaptive law modification (9.34), the above relation does indeed hold.

When $\hat{b} > b_{\min}$, the adaptive law (9.34) is the same as the corresponding law in (9.32) and, therefore, $\dot{V} \leq 2\, a_{ref}\, e^2 \leq 0$. Suppose that there exists $0 \leq T < \infty$ such that $\hat{b}(T) = b_{\min}$. Since $b \geq b_{\min}$, then $\Delta b(T) = \hat{b}(T) - b = b_{\min} - b \leq 0$. If $u(T)\, e(T) \geq 0$, then $\dot{V} = 2\, a_{ref}\, e^2 \leq 0$, while $\dot{\hat{b}}(T) = \gamma_b\, u\, e \geq 0$ implying that $\hat{b}(t)$ increases locally for $t \geq T$. On other hand, if $u(T)\, e(T) < 0$, then according to (9.34), at $t = T$: $\Delta b \left(\gamma_b^{-1}\, \dot{\hat{b}} - u\, e \right) = -\underbrace{\Delta b}_{\leq 0}\, \underbrace{u\, e}_{\leq 0} \leq 0$, and so again, $\dot{V} \leq 2\, a_{ref}\, e^2 \leq 0$. This proves the desired properties of (9.34) for $\text{sgn}\, b > 0$. For $\text{sgn}\, b < 0$, formal arguments are similar, and therefore, they will be left as an exercise for the reader.

The adaptive law modification (9.34) enforces the nonpositive sign of $\dot{V}$, and as such, it contributes to achieving closed-loop system stability. The parameter adaptation dynamics in (9.34) represent a special case of the Projection Operator [6], whose continuous version will be introduced in Chap. 11.

With the proposed adjustment (9.34), the DI-based adaptive laws (9.32) become

$$\begin{aligned} \dot{\hat{a}} &= \gamma_a\, x\, e \\ \dot{\hat{b}} &= \begin{cases} \gamma_b\, u\, e, & \text{if } \left|\hat{b}\right| > b_{\min} \vee \left[\hat{b} = b_{\min}\, \text{sgn}\, b \wedge (u\, e)\, \text{sgn}\, b > 0\right] \\ 0, & \text{if } \left|\hat{b}\right| = b_{\min} \wedge (u\, e)\, \text{sgn}\, b < 0 \end{cases} \\ \dot{\hat{\theta}} &= \Gamma_\theta\, \Phi(x)\, e \end{aligned} \tag{9.36}$$

Next, we are going to formally prove that the DI-based adaptive controller in (9.27) provides global uniform asymptotic tracking of the reference model state. Since $\dot{V} \leq 0$, then $(e,\ \Delta a,\ \Delta b,\ \Delta\theta)$ are uniformly bounded. The latter implies that $\left(x,\ \hat{a},\ \hat{b},\ \hat{\theta}\right)$ are also uniformly bounded. Due to (9.34), $\hat{b} \geq b_0$ and

Table 9.2 DI-based MRAC design summary

Open-loop plant	$\dot{x} = a\,x + b\,u + \theta^T \Phi(x)$
Reference model	$\dot{x}_{ref} = a_{ref}\,x_{ref} + b_{ref}\,r$
Tracking error	$e = x - x_{ref}$
Control input	$u = \frac{1}{\hat{b}}\left(\left(a_{ref} - \hat{a}\right)x + b_{ref}\,r - \hat{\theta}^T\,\Phi(x)\right)$
DI-based MRAC laws	$\dot{\hat{a}} = \gamma_a\,x\,e$ $\dot{\hat{b}} = \begin{cases} \gamma_b\,u\,e, & \text{if } \lvert\hat{b}\rvert > b_{\min} \vee \left[\hat{b} = b_{\min}\,\text{sgn}\,b \wedge (u\,e)\,\text{sgn}\,b > 0\right] \\ 0, & \text{if } \lvert\hat{b}\rvert = b_{\min} \wedge (u\,e)\,\text{sgn}\,b < 0 \end{cases}$ $\dot{\hat{\theta}} = \Gamma_\theta\,\Phi(x)\,e$

consequently, u is uniformly bounded, and so is $\dot{x}$. Since r is bounded, then $\dot{x}_{ref}$ is bounded, and consequently, $\dot{e}$ is bounded as well. Because of (9.34)

$$\dot{V}(e,\ \Delta a,\ \Delta b,\ \Delta\theta) \leq -2\left|a_{ref}\right| e^2 \leq 0 \tag{9.37}$$

for all $t \geq 0$. Since V is positive definite and its derivative is semi-negative definite, then V converges to a limit, as a function of time. Integrating both sides of (9.37) yields

$$V(t) - V(0) \leq -2\left|a_{ref}\right| \int_0^t e^2(\tau)\,d\tau \leq 0 \tag{9.38}$$

or, equivalently:

$$\int_0^t e^2(\tau)\,d\tau \leq \frac{1}{2\left|a_{ref}\right|}\,(V(0) - V(t)) < \infty \tag{9.39}$$

Let $W(t) = \int_0^t e^2(\tau)\,d\tau$. From (9.39), it follows that $W(t)$ tends to a finite limit, as $t \to \infty$. At the same time, its time derivative is $\dot{W}(t) = e^2(t)$, and so its second time derivative is bounded: $\ddot{W}(t) = 2\,e\,\dot{e}(t) < \infty$. Then, $\dot{W}(t)$ is uniformly continuous. Finally, we can apply Barbalat's lemma to conclude that $\lim_{t\to\infty} \dot{W}(t) = 0$, which immediately implies $\lim_{t\to\infty} e^2(t) = 0$. The tracking problem is solved, and the DI-based MRAC design is summarized in Table 9.2.

Let us now illustrate the DI-based MRAC procedure by redesigning the pitch controller using the helicopter pitch dynamics data from Example 9.1.

Example 9.2 DI-Based MRAC Design for Helicopter Pitch Dynamics For the helicopter pitch dynamics (see Example 9.1)

$$\dot{q} = -0.61\, q - 6.65\, \delta + \underbrace{\left(\underbrace{0.0665}_{\theta}\ \underbrace{\tanh\left(\frac{360}{\pi}\, q\right)}_{\Phi(q)} \right)}_{f(q)}$$

the DI-based MRAC system is constructed using the design equations from Table 9.2. The resulting adaptive pitch controller

$$\delta = \frac{1}{\hat{b}} \left(\left(a_{ref} - \hat{a}\right) q + b_{ref}\, q_{cmd} - \hat{\theta}^T\, \Phi(q) \right)$$

with the reference model parameters $a_{\text{ref}} = -b_{ref} = -4$, and with the adaptive laws

$$\dot{\hat{a}} = \gamma_a\, q \left(q - q_{ref}\right)$$

$$\dot{\hat{b}} = \begin{cases} \gamma_b\, \delta \left(q - q_{ref}\right), & \text{if } \hat{b} < -b_{\min} \vee \left[\hat{b} = -b_{\min} \wedge \left(\delta \left(q - q_{ref}\right)\right) < 0\right] \\ 0, & \text{if } \hat{b} = -b_{\min} \wedge \left(\delta \left(q - q_{ref}\right)\right) > 0 \end{cases}$$

$$\dot{\hat{\theta}} = \Gamma_\theta\, \Phi(q) \left(q - q_{ref}\right)$$

were given the task to track the same exact step-input commands from Example 9.1. We assumed $|M_\delta| \geq b_{\min} = 1$ to be the known lower bound and selected $\hat{b}(0) = -4$ to represent the initial value for the estimated elevator effectiveness. With the adaptation rates $\gamma_a = \gamma_b = \Gamma_\theta = 200$, the closed-loop system pitch rate response and the corresponding elevator input are shown in Fig. 9.9.

Similar to Fig. 9.3, these simulation data also show adequate command tracking performance and achievable control input (elevator) values. Comparison of the DI-based MRAC signal with the ideal fixed-gain controller (Fig. 9.10) reveals transient oscillations in the MRAC signal.

Moreover, it is interesting to note that the estimated parameters are nowhere near their ideal unknown values (Fig. 9.11).

For example, the estimated pitch damping $\hat{a} = \hat{M}_q$ remains predominantly near zero, while its true value $a = M_q$ is negative and much larger than this estimate. Nevertheless, as predicted by the design, all of the estimated parameters stay uniformly bounded in time, while the system state tracks the state of the desired reference model. □

After reviewing the two simulation examples presented in this section, the reader should be able to appreciate inherent nonlinear features of MRAC systems. Even

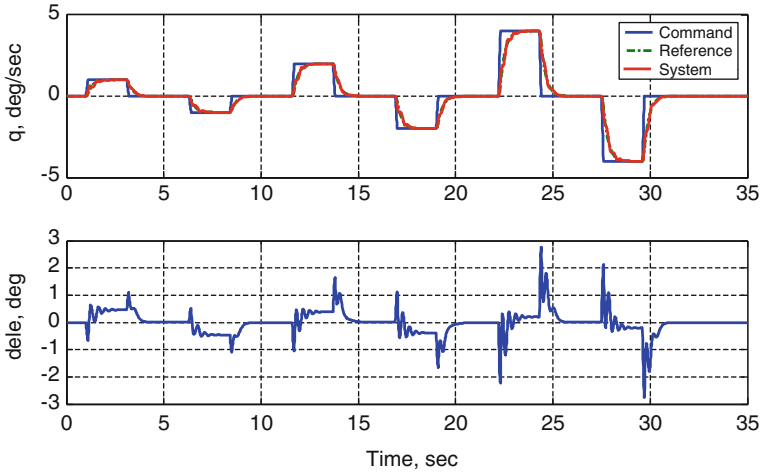


Fig. 9.9 Closed-loop tracking performance in Example 9.2

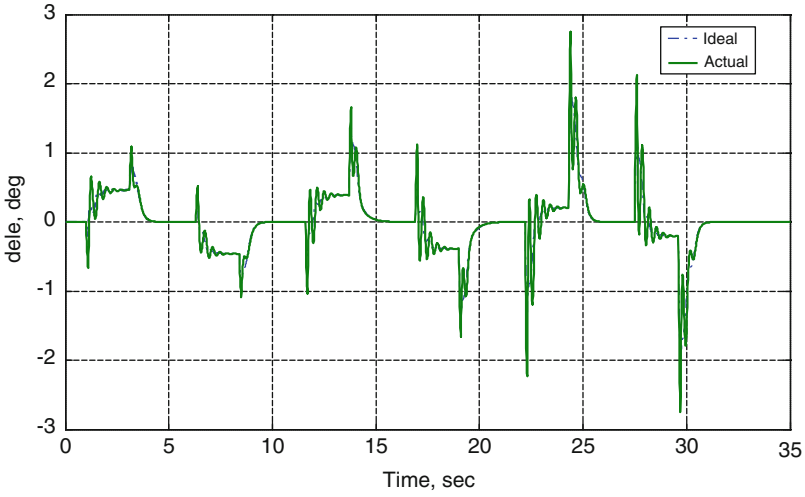


Fig. 9.10 DI-based MRAC and ideal fixed-gain controller from Example 9.2

for scalar dynamics, these controllers may yield transient oscillations and adaptive gain values that do not resemble the true unknowns in the system dynamics. However, these "undesirable" features are not in conflict with the formally derived MRAC design. Achieving smooth transients or having adaptive parameters converge to their ideal values was not formulated as the design goals. Only asymptotic command tracking was of interest and that goal was fully achieved despite the system uncertainties.

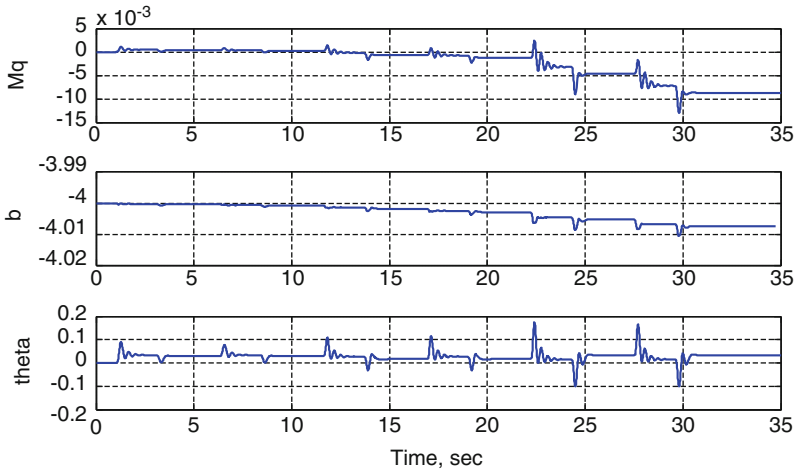


Fig. 9.11 Estimated parameters in Example 9.2

9.5 MRAC Design for Multi-Input Multi-Output Systems

In this section, we will extend applicability of the MRAC design from scalar dynamics to multi-input multi-output (MIMO) nonlinear systems in the form

$$\dot{x} = A\,x + B\,\Lambda\,(u + f(x)) \tag{9.40}$$

where $x \in R^n$ is the system state, $u \in R^m$ is the control input, and $B \in R^{n\times m}$ is the known control matrix, while $A \in R^{n\times n}$ and $\Lambda \in R^{m\times m}$ are unknown constant matrices. In addition, it is assumed that Λ is diagonal, its elements λ_i are strictly positive, and the pair $(A,\ B\,\Lambda)$ is controllable. The uncertainty in Λ is introduced to model control failures or modeling errors, in the sense that there may exist uncertain control gains or the designer may have incorrectly estimated the system control effectiveness.

In (9.40), the unknown possibly nonlinear vector-function $f(x) : R^n \to R^m$ represents the system matched uncertainty. It is assumed that each individual component $f_i(x)$ of $f(x)$can be written as a linear combination of N known locally Lipschitz-continuous basis functions $\varphi_i(x)$, with unknown constant coefficients. So, we write

$$f(x) = \Theta^T\,\Phi(x) \tag{9.41}$$

where $\Theta \in R^{N\times m}$ is a constant matrix of the unknown coefficients and $\Phi(x) = (\,\varphi_1(x) \quad \ldots \quad \varphi_N(x))^T \in R^N$ is the known regressor vector.

We are interested in the design of a MIMO state feedback adaptive control law such that the system state x globally uniformly asymptotically tracks the state $x_{ref} \in R^n$ of the reference model

$$\dot{x}_{ref} = A_{ref}\, x_{ref} + B_{ref}\, r(t) \tag{9.42}$$

where $A_{ref} \in R^{n\times n}$ is Hurwitz, $B_{ref} \in R^{n\times m}$, and $r(t) \in R^m$ is the external bounded command vector.

We also require that during tracking, all signals in the closed-loop system remain uniformly bounded. Thus, given any bounded command $r(t)$, the control input u needs to be chosen such that the state tracking error

$$e(t) = x(t) - x_{ref}(t) \tag{9.43}$$

globally uniformly asymptotically tends to zero, that is,

$$\lim_{t\to\infty} \left\| x(t) - x_{ref}(t) \right\| = 0 \tag{9.44}$$

If matrices A and Λ were known, one could have calculated and applied the ideal fixed-gain control law

$$u = K_x^T\, x + K_r^T\, r - \Theta^T\, \Phi(x) \tag{9.45}$$

and obtain the closed-loop system:

$$\dot{x} = \left(A + B\,\Lambda\, K_x^T\right) x + B\,\Lambda\, K_r^T\, r \tag{9.46}$$

Comparing (9.46) with the desired reference dynamics (9.42), it follows that for existence of a controller in the form of (9.45), the ideal unknown control gains, K_x and K_r, must satisfy the matching conditions

$$\begin{aligned} A + B\,\Lambda\, K_x^T &= A_{ref} \\ B\,\Lambda\, K_r^T &= B_{ref} \end{aligned} \tag{9.47}$$

Assuming that these matching conditions hold, it is easy to see that using (9.45) yields the closed-loop system which is exactly the same as the reference model. Consequently, for any bounded reference input signal $r(t)$, the fixed-gain controller (9.45) provides global uniform asymptotic tracking performance.

Let us at once note that given $\left(A,\ B,\ \Lambda,\ A_{ref},\ B_{ref}\right)$, there is no guarantee that the ideal gains $K_x,\ K_r$ exist such that the matching conditions (9.47) are satisfied. In other words, the control law (9.45) may not be able to meet the design objective. However often in practice, the structure of A is known, and the reference model matrices $A_{\text{ref}},\ B_{\text{ref}}$ are chosen so that the system (9.47) has at least one ideal solution pair $(K_x,\ K_r)$.

Assuming that $K_x,\ K_r$ in (9.47) do exist, we consider the following control law:

$$u = \hat{K}_x^T\, x + \hat{K}_r^T\, r - \hat{\Theta}^T\, \Phi(x) \tag{9.48}$$

where $\hat{K}_x \in R^{n\times m}$, $\hat{K}_r \in R^{m\times m}$, $\hat{\Theta} \in R^{N\times n}$ are the estimates of the ideal unknown matrices K_x, K_r, Θ, respectively. These estimated parameters will be generated online through the inverse Lyapunov analysis. Substituting (9.48) into (9.40), the closed-loop system dynamics can be written as

$$\dot{x} = \left(A + B\,\Lambda\,\hat{K}_x^T\right) x + B\Lambda\left(\hat{K}_r^T\, r - \left(\hat{\Theta} - \Theta\right)^T \Phi(x)\right) \tag{9.49}$$

Subtracting (9.42) from (9.49), we compute the closed-loop dynamics of the n-dimensional tracking error vector $e(t) = x(t) - x_{ref}(t)$:

$$\dot{e} = \left(A + B\,\Lambda\,\hat{K}_x^T\right) x + B\,\Lambda\left(\hat{K}_r^T\, r - \left(\hat{\Theta} - \Theta\right)^T \Phi(x)\right) - A_{ref}\, x_{ref} - B_{ref}\, r \tag{9.50}$$

With the matching conditions (9.47) in place, we further get

$$\begin{aligned}\dot{e} =& \left(A_{ref} + B\,\Lambda\left(\hat{K}_x - K_x\right)\right) x - A_{ref}\, x_{ref} \\ &+ B\,\Lambda\left(\hat{K}_r - K_r\right) r - B\,\Lambda\left(\hat{\Theta} - \Theta\right)^T \Phi(x) \\ =& A_{ref}\, e + B\,\Lambda\left[\left(\hat{K}_x - K_x\right)^T x + \left(\hat{K}_r - K_r\right)^T r - \left(\hat{\Theta} - \Theta\right)^T \Phi(x)\right]\end{aligned} \tag{9.51}$$

Let $\Delta K_x = \hat{K}_x - K_x$, $\Delta K_r = \hat{K}_r - K_r$, and $\Delta\Theta = \hat{\Theta} - \Theta$ represent the parameter estimation errors. In terms of the latter, the tracking error dynamics become

$$\dot{e} = A_{ref}\, e + B\,\Lambda\left[\Delta K_x^T\, x + \Delta K_r^T\, r - \Delta\Theta^T\, \Phi(x)\right] \tag{9.52}$$

We introduce rates of adaptation: $\Gamma_x = \Gamma_x^T > 0, \quad \Gamma_r = \Gamma_r^T > 0, \quad \Gamma_\Theta = \Gamma_\Theta^T > 0$. Going back to analyzing stability of the tracking error dynamics (9.52), let us consider a globally radially unbounded quadratic Lyapunov function candidate in the form

$$\begin{aligned}&V(e,\, \Delta K_x,\, \Delta K_r,\, \Delta\Theta) \\ &= e^T P\, e + \mathrm{tr}\left(\left[\Delta K_x^T\, \Gamma_x^{-1}\, \Delta K_x + \Delta K_r^T\, \Gamma_r^{-1}\, \Delta K_r + \Delta\Theta^T\, \Gamma_\Theta^{-1}\, \Delta\Theta\right] \Lambda\right)\end{aligned} \tag{9.53}$$

where $P = P^T > 0$ satisfies the algebraic Lyapunov equation

$$P\, A_{ref} + A_{ref}^T\, P = -Q \tag{9.54}$$

for some $Q = Q^T > 0$. Then, the time derivative of V, evaluated along the trajectories of (9.52), can be calculated:

$$\begin{aligned}\dot{V} &= \dot{e}^T P e + e^T P \dot{e} + 2\,\mathrm{tr}\left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}}\right] \Lambda\right) \\ &= \left(A_{\text{ref}}\, e + B\,\Lambda\left(\Delta K_x^T x + \Delta K_r^T r - \Delta\Theta^T \Phi(x)\right)\right)^T P e \\ &\quad + e^T P \left(A_{\text{ref}}\, e + B\,\Lambda\left(\Delta K_x^T x + \Delta K_r^T r - \Delta\Theta^T \Phi(x)\right)\right) \\ &\quad + 2\,\mathrm{tr}\left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}}\right] \Lambda\right) \\ &= e^T \left(A_{\text{ref}}^T P + P A_{\text{ref}}\right) e + 2\, e^T P B \Lambda \left(\Delta K_x^T x + \Delta K_r^T r - \Delta\Theta^T \Phi(x)\right) \\ &\quad + 2\,\mathrm{tr}\left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}}\right] \Lambda\right)\end{aligned} \tag{9.55}$$

Using (9.54) further yields

$$\begin{aligned}\dot{V} = -e^T Q\, e &+ \left[2\, e^T P B \Lambda \Delta K_x^T x + 2\,\mathrm{tr}\left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x \Lambda\right)\right] \\ &+ \left[2\, e^T P B \Lambda \Delta K_r^T r + 2\,\mathrm{tr}\left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r \Lambda\right)\right] \\ &+ \left[-2\, e^T P B \Lambda \Delta\Theta^T \Phi(x) + 2\,\mathrm{tr}\left(\Delta\Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \Lambda\right)\right]\end{aligned} \tag{9.56}$$

Via the vector trace identity (defined in Chap. 1),

$$\begin{aligned}\underbrace{e^T P B \Lambda}_{a^T}\ \underbrace{\Delta K_x^T x}_{b} &= \mathrm{tr}\left(\underbrace{\Delta K_x^T x}_{b}\ \underbrace{e^T P B \Lambda}_{a^T}\right) \\ \underbrace{e^T P B \Lambda}_{a^T}\ \underbrace{\Delta K_r^T r}_{b} &= \mathrm{tr}\left(\underbrace{\Delta K_r^T r}_{b}\ \underbrace{e^T P B \Lambda}_{a^T}\right) \\ \underbrace{e^T P B \Lambda}_{a^T}\ \underbrace{\Delta\Theta^T \Phi(x)}_{b} &= \mathrm{tr}\left(\underbrace{\Delta\Theta^T \Phi(x)}_{b}\ \underbrace{e^T P B \Lambda}_{a^T}\right)\end{aligned} \tag{9.57}$$

Substituting (9.57) into (9.56) results in

$$\begin{aligned}\dot{V} = &- e^T Q\, e + 2\,\mathrm{tr}\left(\Delta K_x^T \left[\Gamma_x^{-1} \dot{\hat{K}}_x + x\, e^T P B\right] \Lambda\right) \\ &+ 2\,\mathrm{tr}\left(\Delta K_r^T \left[\Gamma_r^{-1} \dot{\hat{K}}_r + r\, e^T P B\right] \Lambda\right) + 2\,\mathrm{tr}\left(\Delta\Theta^T \left[\Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi(x)\, e^T P B\right] \Lambda\right)\end{aligned} \tag{9.58}$$

If the adaptive laws are selected as

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x\, x\, e^T P B \\ \dot{\hat{K}}_r &= -\Gamma_r\, r(t)\, e^T P B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta\, \Phi(x)\, e^T P B\end{aligned} \tag{9.59}$$

then the time derivative of V in (9.58) becomes globally negative semidefinite:

$$\dot{V} = -e^T Q\, e \leq 0 \tag{9.60}$$

Therefore, the closed-loop error dynamics are uniformly stable. So, the tracking error $e(t)$ and the parameter estimation errors $\Delta K_x(t)$, $\Delta K_r(t)$, and $\Delta\Theta(t)$ are uniformly bounded and so are the parameter estimates $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$. Since $r(t)$ is bounded and A_{ref} is Hurwitz, then $x_{\text{ref}}(t)$ and $\dot{x}_{\text{ref}}(t)$ are bounded. Hence, the system state $x(t)$ is uniformly bounded, and the control input $u(t)$ in (9.48) is bounded as well. The latter implies that $\dot{x}(t)$ is bounded, and thus, $\dot{e}(t)$ is bounded. Furthermore, the second time derivative of $V(t)$

$$\ddot{V} = -2\, e^T Q\, \dot{e} \tag{9.61}$$

is bounded, and so $\dot{V}(t)$ is uniformly continuous. Since in addition, $V(t)$ is lower bounded and $\dot{V}(t) \leq 0$, then using Barbalat's lemma gives $\lim_{t\to\infty} \dot{V}(t) = 0$. We have formally proven that the state tracking error $e(t)$ tends to the origin globally, uniformly, and asymptotically: $\lim_{t\to\infty} \left\| x(t) - x_{ref}(t) \right\| = 0$. The MIMO command tracking problem for the system dynamics (9.40) is solved. We now formulate our obtained results as a theorem.

Theorem 9.2. *Given MIMO dynamics (9.40) with a control uncertainty* Λ *and a matched unknown function* $f(x)$ *from (9.41), the MRAC system (9.48), (9.49), (9.50), (9.51), (9.52), (9.53), (9.54), (9.55), (9.56), (9.57), (9.58), and (9.59) enforces global uniform asymptotic tracking performance of the reference model dynamics (9.42), driven by any bounded time-varying command* $r(t)$*. Moreover, all signals in the corresponding closed-loop system remain uniformly bounded in time.* ■

Table 9.3 summarizes the MIMO MRAC design equations.

Example 9.3 MRAC Control of Delta Wing Dynamics at High Angle of Attack A sketch of an aircraft equipped with a delta wing is shown in Fig. 9.12.

The wing sweeps sharply back from the fuselage with the angle between the wing leading edge often as high as 80° and the angle between the fuselage and the trailing edge of the wing at around 90°. Delta wings are known to be unstable, especially at high angle of attack (the angle between the aircraft velocity vector and the fuselage centerline). Yet their primary advantage is aerodynamic efficiency in high-speed flight [10].

Table 9.3 MIMO MRAC design equations

Open-loop plant	$\dot{x} = A\,x + B\,\Lambda\left(u + \Theta^T \Phi(x)\right)$
Reference model	$\dot{x}_{ref} = A_{ref}\,x_{ref} + B_{ref}\,r$
Model matching conditions	$A + B\,\Lambda\,K_x^T = A_{ref}, \quad B\,\Lambda\,K_r^T = B_{ref}$
Tracking error	$e = x - x_{ref}$
Control input	$u = \hat{K}_x^T\,x + \hat{K}_r^T\,r - \hat{\Theta}^T\,\Phi(x)$
Algebraic Lyapunov equation	$P\,A_{ref} + A_{ref}^T\,P = -Q$
MIMO MRAC laws	$\dot{\hat{K}}_x = -\Gamma_x\,x\,e^T P\,B$ $\dot{\hat{K}}_r = -\Gamma_r\,r(t)\,e^T P\,B$ $\dot{\hat{\Theta}} = \Gamma_\Theta\,\Phi(x)\,e^T P\,B$

A delta wing aircraft flying at high angle of attack is open-loop unstable in roll. This instability is called the "wing rock phenomenon." It is induced by unsteady aerodynamic effects acting on the delta wing asymmetrically. As a result, the aircraft undergoes an unstable rocking motion that needs to be actively controlled.

In this example, we consider a delta wing aircraft whose roll dynamics can be regulated by ailerons – the movable surfaces that are located symmetrically on the outboard portions of the aircraft left and right wing segments. Moving the left aileron down (positive deflection) and the right one up (negative deflection) induces the right-wing-down rolling motion (positive roll rate) of the aircraft. The difference between the left and right aileron positions is called the "differential aileron." This is the primary control input signal for regulating the aircraft roll dynamics.

We shall make use of a generic delta wing rock dynamic model in the form

$$\begin{aligned}\dot{\varphi} &= p\\ \dot{p} &= \theta_1\,\varphi + \theta_2\,p + (\theta_3\,|\varphi| + \theta_4\,|p|)p + \theta_5\,\varphi^3 + \theta_6\,\delta_a\end{aligned}$$

where φ is the aircraft roll angle (rad), p is the roll rate (rad/s), and δ_a is the differential aileron (control input, rad). The unknown constant parameters are

$$\theta_1 = -0.018, \quad \theta_2 = 0.015, \quad \theta_3 = -0.062, \quad \theta_4 = 0.009, \quad \theta_5 = 0.021, \quad \theta_6 = 0.75$$

Rewriting the model in the form of (9.40) gives

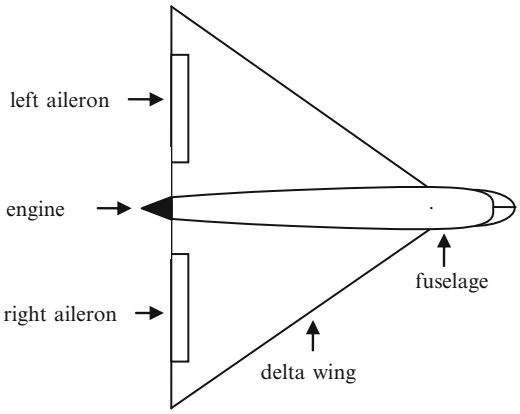


Fig. 9.12 Delta wing aircraft

$$\underbrace{\begin{pmatrix}\dot{\varphi}\\ \dot{p}\end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix}0 & 1\\ \theta_1 & \theta_2\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}\varphi\\ p\end{pmatrix}}_{x} + \underbrace{\begin{pmatrix}0\\ 1\end{pmatrix}}_{B}\underbrace{\theta_6}_{\Lambda}\left(\underbrace{\delta_a}_{u} + \underbrace{\frac{1}{\theta_6}\left((\theta_3\,|\varphi| + \theta_4\,|p|)p + \theta_5\,\varphi^3\right)}_{f(x)=\Theta^T\Phi(x)}\right)$$

where the uncertain state-dependent function $f(x)$ is represented by a constant unknown parameter vector Θ and the known regressor vector $\Phi(x)$:

$$\begin{aligned} f(x) &= \underbrace{\frac{1}{\theta_6}\begin{pmatrix}\theta_3 & \theta_4 & \theta_5\end{pmatrix}}_{\Theta^T}\underbrace{\begin{pmatrix}|\varphi|\,p & |p|\,p & \varphi^3\end{pmatrix}^T}_{\Phi(x)}\\ &= -0.0827\,|\varphi|\,p + 0.012\,|p|\,p + 0.028\,\varphi^3 \end{aligned}$$

The system control effectiveness $\Lambda = \theta_6$ is assumed to be constant and unknown. Also unknown are the second row coefficients $(\theta_1,\ \theta_2)$ in A, as well as the state-dependent function $f(x)$.

The reference roll dynamics are defined by the second-order transfer function

$$\frac{\varphi_{ref}}{\varphi_{cmd}} = \frac{\omega_n^2}{s^2 + 2\,\xi\,\omega_n\,s + \omega_n^2}$$

which represents the desired command-to-response roll angle behavior (using the Laplace transform). Here, φ_{ref} is the reference roll angle, φ_{cmd} is the commanded roll angle, and $(\omega_n,\ \xi)$ are the desired natural frequency and the damping ratio,

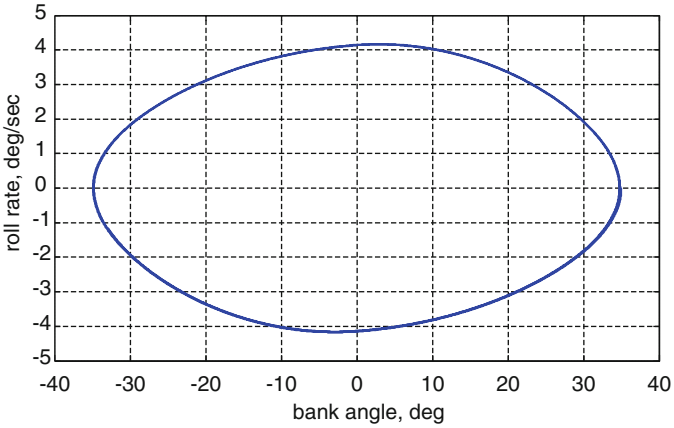


Fig. 9.13 Open-loop limit cycle for delta wing roll dynamics in Example 9.3

respectively. Let $p_{\text{ref}} = \dot{\varphi}_{ref}$ denote the reference roll rate. In state space form, the reference roll dynamics can be easily written as

$$\underbrace{\begin{pmatrix} \dot{\varphi}_{ref} \\ \dot{p}_{ref} \end{pmatrix}}_{\dot{x}_{ref}} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\,\xi\,\omega_n \end{pmatrix}}_{A_{ref}} \underbrace{\begin{pmatrix} \varphi_{ref} \\ p_{ref} \end{pmatrix}}_{x_{ref}} + \underbrace{\begin{pmatrix} 0 \\ \omega_n^2 \end{pmatrix}}_{B_{ref}} \underbrace{\varphi_{cmd}}_{r}$$

Clearly, the matching conditions (9.47) hold. In fact, from

$$\begin{pmatrix} 0 & 1 \\ \theta_1 & \theta_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \theta_6 K_x^T = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\,\xi\,\omega_n \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \theta_6 K_r^T = \begin{pmatrix} 0 \\ \omega_n^2 \end{pmatrix}$$

it follows that the ideal unknown feedback and feedforward gains are

$$K_x^T = -\frac{1}{\theta_6}\begin{pmatrix} \omega_n^2 + \theta_1 & 2\,\xi\,\omega_n + \theta_2 \end{pmatrix}, \quad K_r^T = \frac{\omega_n^2}{\theta_6}$$

In this example, we have selected $\omega_n = 1,\ \text{rad/s}$ and $\xi = 0.7$. So, the ideal gains are $K_x = \begin{pmatrix} -1.3093 & -1.8867 \end{pmatrix}^T$ and $K_r = 1.3333$.

One can verify that the open-loop system, with $\delta_a = 0$, has an unstable equilibrium at the origin and a limit cycle near $\varphi = 35^\circ$. The limit cycle attracts all open-loop system trajectories that start on the inside of its boundary, and it repels all the trajectories with the initial conditions on the outside (see Fig. 9.13).

We use the design equations from Table 9.3 to construct an MRAC system. The design "tuning knobs" consist of symmetric positive-definite matrices Q, Γ_x, Γ_r, and

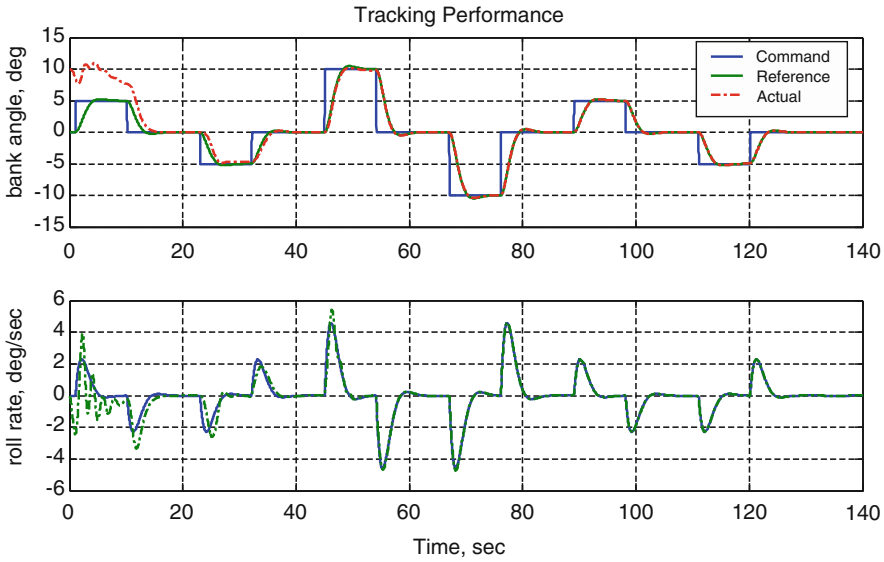


Fig. 9.14 Closed-loop system tracking performance in Example 9.3

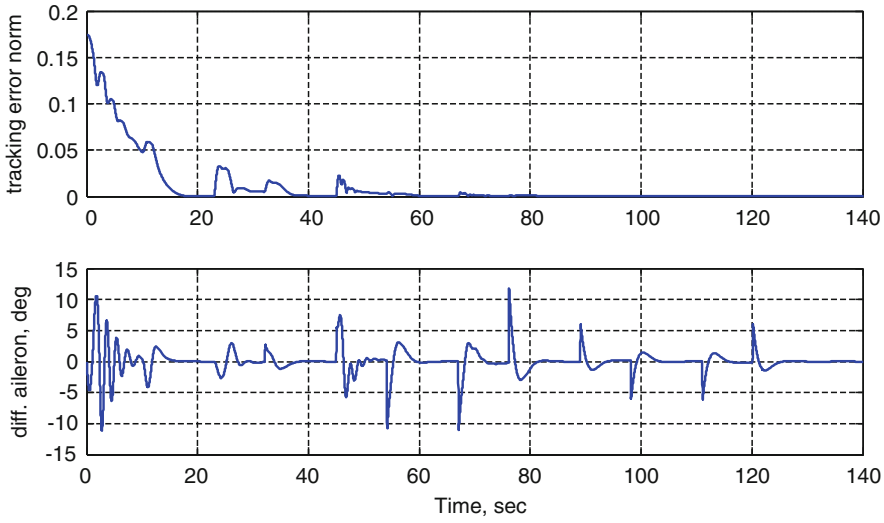


Fig. 9.15 Tracking error and differential aileron in Example 9.3

Γ_θ, with the last three quantities representing adaptation rates for the adaptive parameters $\hat{K}_x$, $\hat{K}_r$, and $\hat{\Theta}$, respectively. After several iterations, we have selected the following data:

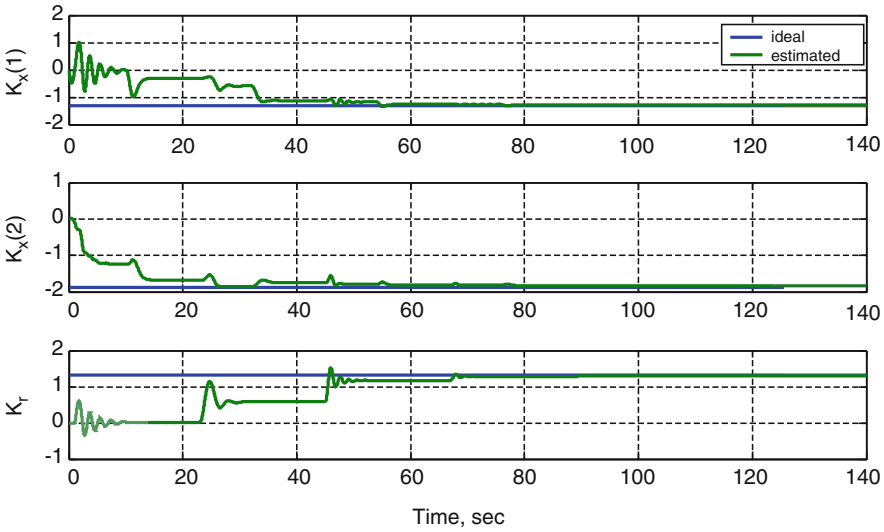


Fig. 9.16 Estimated feedback and feedforward gains in Example 9.3

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \quad \Gamma_x = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}, \quad \Gamma_r = 100, \quad \Gamma_\theta = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}$$

Figure 9.14 shows the system closed-loop response in tracking a series of step-input commands, with the initial bank angle set to 10°.

The system tracking error quickly dissipates (Fig. 9.15), while the required control input (differential aileron position) stays within achievable and reasonable limits.

It is interesting to observe (see Fig. 9.16) that in this case, the estimated feedback and feedforward gains $\left(\hat{K}_x,\ \hat{K}_r\right)$ converge to their true unknown values.

However, the estimated parameters $\left(\hat{\theta}_3,\ \hat{\theta}_4,\ \hat{\theta}_5\right)$ that correspond to the nonlinear regressor components have dissimilar tendencies: The first two are quite different from their ideal counterparts, while the third one does converge to its ideal value (Fig. 9.17).

Once again, we would like to remark that in general, parameter convergence is not guaranteed by an MRAC controller (see Theorem 9.2). Only uniform boundedness of all signals in the closed-loop system is certain. In order to emphasize this point, we encourage the reader to rerun this exact design but with a different bank angle command profile. For example, choosing $\varphi_{cmd} = 0.1745 \sin t$ will result in all adaptive parameters being very different from their ideal values, yet the closed-loop system tracking performance will remain acceptable. □

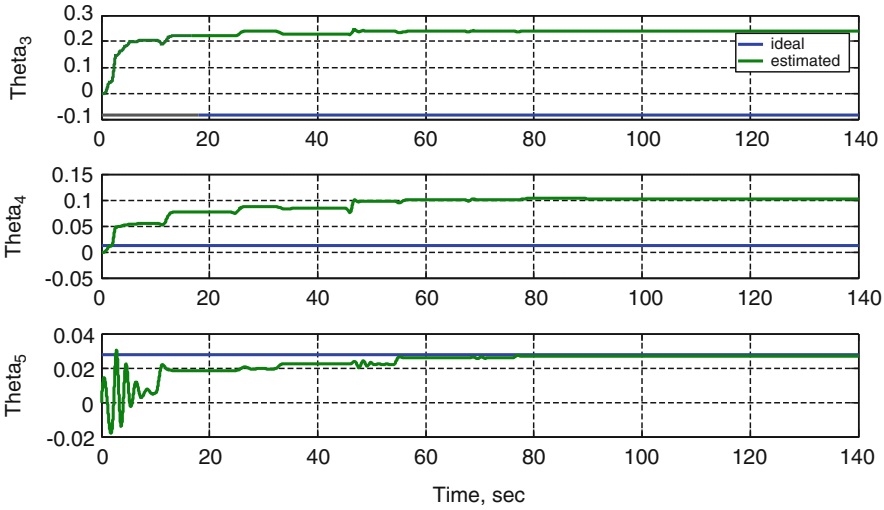


Fig. 9.17 Estimated parameters for nonlinear regressor components in Example 9.3

9.6 Summary

Adaptive control was initiated, inspired, and originally motivated by aerospace applications in the 1950s. The interest in the design of adaptive self-tuning controllers for practical systems that operate in uncertain environment has never diminished. The last decade has witnessed many successful demonstrations of adaptive control technology in aerospace, robotics, auto industry, and bioengineering.

Adaptive controllers have one common goal – to enable a dynamical system to track external commands while operating in realistic and often uncertain environment. In this chapter, we have shown how to achieve this goal for a specific class of multi-input multi-output dynamical systems with matched uncertainties. We have also demonstrated efficacy of adaptive control through simulation examples such as (a) helicopter pitch dynamics in hover and (b) wing rock dynamics of a delta wing at high angle of attack. All simulation data confirmed our theoretical predictions for MRAC in achieving desired tracking performance and keeping all signals in the corresponding closed-loop system uniformly bounded in time.

9.7 Exercises

Exercise 9.1. Prove that if some of the diagonal elements λ_i of the unknown diagonal matrix Λ in the system dynamics (9.40) are negative and the signs of all of them are known, then the adaptive laws

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x x e^T P B \operatorname{sgn}\Lambda \\ \dot{\hat{K}}_r &= -\Gamma_r r(t) e^T P B \operatorname{sgn}\Lambda \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi(x) e^T P B \operatorname{sgn}\Lambda \end{aligned} \tag{9.62}$$

solve the MIMO tracking problem, where $\operatorname{sgn}\Lambda = \operatorname{diag}[\operatorname{sgn}\lambda_1, \ \ldots, \ \operatorname{sgn}\lambda_m]$.

Exercise 9.2. Implement and simulate the system from Example 9.1. Test the MRAC controller (redesign, if needed) in the presence of various uncertainties and external commands of your choice. Comment on the system tracking performance. Discuss adaptive parameter dynamics and convergence of the estimated parameters to their true values.

Exercise 9.3. Repeat all tasks from Exercise 9.2 for the DI-based MRAC system in Example 9.2. Compare closed-loop performance of the two controllers, including their respective control efforts.

Exercise 9.4. Repeat all tasks from Exercise 9.2 for the MRAC controller and the system in Example 9.3. Find external commands that would cause the adaptive parameters to (a) converge to their true values, (b) converge to some constant values, and (c) not have limits. Is there a benefit for the adaptive parameters to converge to their true values? For all these cases, compare and discuss the system tracking performance.

References

1. Whitaker, H.P., Yamron, J., Kezer, A.: Design of Model Reference Adaptive Control Systems for Aircraft, Report R-164, Instrumentation Laboratory, MIT, Cambridge, MA (1958)
2. Whitaker, H.P., Osburn, P.V., Kezer A.: New Developments in the Design of Model Reference Adaptive Control Systems, Paper 61–39, Institute of the Aerospace Sciences, Easton, PA (1961)
3. Butchart, R.L., Shackcloth, B.: Synthesis of Model Reference Adaptive Control Systems by Lyapunov's Second Method. In: Proceedings of the 1965 IFAC symposium on adaptive control, Teddington (1965)
4. Parks, P.D.: Lyapunov redesign of model reference adaptive systems. IEEE Trans. Automat. Contr. **11**, 362–367 (1966)
5. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Control. Dover, New York (2005)
6. Ioannou, P., Fidan, P.: Adaptive Control Tutorial, SIAM, Advances in Design and control, SIAM, PA (2006)
7. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, New Jersey (1995)
8. Krstic, M., Kanellakopoulos, I., Kokotovic, P.: Nonlinear and Adaptive Control Design. Wiley, New York (1995)
9. Khalil, H.: Nonlinear Systems, 3rd edn, p. 07458. Prentice Hall, Upper Saddle River (1996)
10. Hsu, C., Lan, C.E.: Theory of wing rock. J. Aircraft **22**(10), 920–924 (1985)

Chapter 10
Model Reference Adaptive Control with Integral Feedback Connections

10.1 Introduction

We begin by considering a class of MIMO uncertain systems in the form

$$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + \overbrace{\Theta^T \Phi(x_p)}^{f(x_p)} \right) \tag{10.1}$$

where $x_p \in R^{n_p}$ is the system state vector, $u \in R^m$ is the control input,

$$f(x_p) = \Theta^T \Phi(x_p) \in R^m \tag{10.2}$$

is the linear-in-parameter state-dependent matched uncertainty, $\Theta \in R^{N \times m}$ is the matrix of unknown constant parameters, and $\Phi(x_p) \in R^N$ is the known N-dimensional regressor vector, whose components are locally Lipschitz-continuous functions of x_p. Also, in (10.1), $B_p \in R^{n \times m}$ is constant and known, $A_p \in R^{n \times n}$ is constant and unknown, and $\Lambda \in R^{m \times m}$ is a constant diagonal unknown matrix with positive diagonal elements. We assume that the pair $(A_p, (B_p \Lambda))$ is controllable.

The control goal of interest is bounded command tracking, that is, we need to design u such that the system regulated output

$$y = C_p x_p \in R^m \tag{10.3}$$

tracks any bounded possibly time-varying command $y_{\text{cmd}}(t) \in R^m$, with bounded errors and in the presence of the system uncertainties $\{A_p,\ \Lambda,\ \Theta_d\}$, where the system output matrix C_p is known and constant.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_10,

Let

$$e_y(t) = y(t) - y_{cmd}(t) \tag{10.4}$$

denote the system output tracking error. Augmenting (10.1) with the integrated output tracking error,

$$\left(e_{yI}(t) = \int_0^t e_y(\tau) d\tau \right) \Leftrightarrow \left(e_{yI} = \frac{e_y}{s} \right) \tag{10.5}$$

yields the extended open-loop dynamics

$$\dot{x} = A\,x + B\,\Lambda\left(u + f\left(x_p\right)\right) + B_{ref}\,y_{cmd} \tag{10.6}$$

where $x = \begin{pmatrix} e_{yI}^T & x_p^T \end{pmatrix}^T \in R^n$ is the extended system state vector, whose dimension is $n = n_p + m$. The extended open-loop system matrices are

$$A = \begin{pmatrix} 0_{m\times m} & C_p \\ 0_{n_p\times m} & A_p \end{pmatrix},\ B = \begin{pmatrix} 0_{m\times m} \\ B_p \end{pmatrix},\ B_{ref} = \begin{pmatrix} -I_{m\times m} \\ 0_{n_p\times m} \end{pmatrix} \tag{10.7}$$

and

$$y = \underbrace{\begin{pmatrix} 0_{m\times m} & C_p \end{pmatrix}}_{C} x = C\,x \tag{10.8}$$

represents the extended system controlled output. We will require preservation of controllability for the extended pair of matrices $(A,\ (B\,\Lambda))$ in (10.7). It is not difficult to show that the extended pair is controllable if and only if the original pair $\left(A_p,\ \left(B_p\,\Lambda\right)\right)$ is controllable and $\det\begin{pmatrix} A_p & B_p\,\Lambda \\ C_p & 0_{m\times m} \end{pmatrix} \neq 0$.

To summarize, we are interested in the state feedback output regulation problem for a generic class of MIMO uncertain dynamical systems in the form

$$\begin{aligned} \dot{x} &= A\,x + B\,\Lambda\left(u + \Theta^T\,\Phi\left(x_p\right)\right) + B_{ref}\,y_{cmd}(t) \\ y &= C\,x \end{aligned} \tag{10.9}$$

with known constant matrices $A \in R^{n\times n}$, $B \in R^{n\times m}$, $B_{ref} \in R^{n\times m}$, and $C \in R^{m\times n}$; an unknown constant diagonal positive-definite matrix $\Lambda \in R^{m\times m}$; and an unknown matrix of constant parameters $\Theta \in R^{N\times m}$.

The control goal is to force the system regulated output $y(t) \in R^{m\times 1}$ to track any bounded time-varying reference signal $y_{cmd}(t) \in R^{m\times 1}$, with bounded errors and in the presence of constant parametric uncertainties $(A,\ \Lambda,\ \Theta)$. We shall also require that the rest of the signals in the corresponding closed-loop system remain uniformly bounded in time.

10.2 Control Design

We commence with the assumption about the existence of an adaptive solution to the MIMO command tracking problem of interest.

Assumption 10.1 Model Matching Conditions. Given a reference Hurwitz matrix A_{ref} and an unknown positive-definite diagonal constant matrix Λ, there exists a constant (possibly unknown) gain matrix $K_x \in R^{n\times m}$, such that

$$A_{ref} = A + B\,\Lambda\,K_x^T \tag{10.10}$$

Using (10.10), we can rewrite the open-loop extended system dynamics (10.9) in the form

$$\dot{x} = A_{ref}\,x + B\,\Lambda\left(u - K_x^T\,x + \Theta^T\,\Phi\left(x_p\right)\right) + B_c\,y_{cmd} \tag{10.11}$$

and then choose

$$u = \hat{K}_x^T\,x - \hat{\Theta}^T\,\Phi\left(x_p\right) \tag{10.12}$$

where $\hat{K}_x(t) \in R^{n\times m}$ and $\hat{\Theta}(t) \in R^{N\times m}$ are adaptive gains whose dynamics will be defined later. Substituting (10.12) into (10.11) yields

$$\begin{aligned}\dot{x} &= A_{ref}\,x + B\,\Lambda\left(\underbrace{\left(\hat{K}_x - K_x\right)^T}_{\Delta K_x}x - \underbrace{\left(\hat{\Theta} - \Theta\right)^T}_{\Delta\Theta}\Phi\left(x_p\right)\right) + B_{ref}\,y_{cmd}\\ &= A_{ref}\,x + B\,\Lambda\left(\Delta K_x^T x - \Delta\Theta^T\,\Phi\left(x_p\right)\right) + B_{ref}\,y_{cmd}\end{aligned} \tag{10.13}$$

Based on (10.13), we consider the following reference model:

$$\dot{x}_{\text{ref}} = A_{ref}\,x_{ref} + B_{ref}\,y_{cmd}, \quad y_{ref} = C\,x_{ref} \tag{10.14}$$

It is easy to verify that the transfer function $G_{ref}(s)$ from y_{cmd} to y_{ref}

$$y_{ref} = \underbrace{\left[C\left(s\,I_{n\times n} - A_{ref}\right)^{-1}B_{ref}\right]}_{G_{ref}(s)}\,y_{cmd} \tag{10.15}$$

has the unity DC gain, where s is the Laplace variable. This feature formally prescribes the desired output regulation behavior for constant external commands.

We define the state tracking error

$$e = x - x_{ref} \tag{10.16}$$

and then subtract (10.14) from (10.13) to obtain the tracking error dynamics:

$$\dot{e} = A_{ref}\, e + B\, \Lambda \left(\Delta K_x^T x - \Delta \Theta^T\, \Phi(x_p) \right) \tag{10.17}$$

We now proceed with the Lyapunov-based approach, eventually leading to the design of stable adaptive laws and a verifiable closed-loop system tracking performance. Toward that end, let us consider a radially unbounded Lyapunov function candidate

$$\begin{aligned} V(e,\ \Delta K_x,\ \Delta \Theta) &= e^T\, P_{ref}\, e \\ &+ \text{trace}\left(\Delta K_x^T\, \Gamma_x^{-1}\, \Delta K_x\, \Lambda \right) + \text{trace}\left(\Delta \Theta^T\, \Gamma_\Theta^{-1}\, \Delta \Theta\, \Lambda \right) \end{aligned} \tag{10.18}$$

where $\Gamma_x = \Gamma_x^T > 0$ and $\Gamma_\Theta = \Gamma_\Theta^T > 0$ are rates of adaptation and $P_{ref} = P_{ref}^T > 0$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation

$$P A_{ref} + A_{ref}^T P = -Q \tag{10.19}$$

with some $Q = Q^T > 0$. The time derivative of V, along the trajectories of (10.17), is

$$\begin{aligned} \dot{V}(e,\ \Delta K_x,\ \Delta \Theta) &= -e^T\, Q\, e + 2\, e^T\, P\, B\, \Lambda \left(\Delta K_x^T\, x - \Delta \Theta^T\, \Phi(x_p) \right) \\ &+ 2\, \text{trace}\left(\Delta K_x^T\, \Gamma_x^{-1} \dot{\hat{K}}_x\, \Lambda \right) + 2\, \text{trace}\left(\Delta \Theta^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}}\, \Lambda \right) \end{aligned} \tag{10.20}$$

Applying the vector trace identity (valid for any two co-dimensional vectors a and b),

$$a^T\, b = \text{trace}\left(b\, a^T \right) \tag{10.21}$$

results in

$$\begin{aligned} \dot{V}(e,\ \Delta K_x,\ \Delta \Theta) &= -e^T\, Q\, e + 2\, \text{trace}\left(\Delta K_x^T \left\{ \Gamma_x^{-1} \dot{\hat{K}}_x + x\, e^T\, P\, B \right\} \Lambda \right) \\ &+ 2\, \text{trace}\left(\Delta \Theta^T \left\{ \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}} - \Phi(x_p)\, e^T\, P\, B \right\} \Lambda \right) \end{aligned} \tag{10.22}$$

If adaptive laws are selected in the form

$$\begin{aligned} \dot{\hat{K}}_x &= -\Gamma_x\, x\, e^T\, P\, B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta\, \Phi\, e^T\, P\, B \end{aligned} \tag{10.23}$$

then

$$\dot{V}(e,\ \Delta K_x,\ \Delta \Theta) = -e^T\, Q\, e \leq 0 \tag{10.24}$$

which, in turn, proves uniform ultimate boundedness of $(e,\ \Delta K_x,\ \Delta \Theta)$.

Let L_2 and L_∞ define the set of all square integrable and bounded functions in a Euclidean space of interest, respectively.

Relation (10.24) implies that the tracking error signal is square integrable: $e \in L_2$. Since $y_{cmd} \in L_\infty$, then $x_{ref} \in L_\infty$, and consequently $x \in L_\infty$. Since the ideal (unknown) parameters $(K_x,\ \Theta)$ are constant and their estimation errors $(\Delta K_x,\ \Delta\Theta)$ are bounded, then the corresponding estimated values are bounded as well, that is, $\left(\hat{K}_x,\ \hat{\Theta}\right) \in L_\infty$.

Since all components of the regressor vector $\Phi(x_p)$ are locally Lipschitz-continuous functions of $x_p \in L_\infty$, then the regressor components are also bounded. Hence, $u \in L_\infty$ and $\dot{x} \in L_\infty$. Thus, $\dot{e} \in L_\infty$, which implies that $\ddot{V} \in L_\infty$. Therefore, $\dot{V}$ is a uniformly continuous function of time.

Since V is lower bounded, $\dot{V} \leq 0$, and $\dot{V}$ is uniformly continuous, then V tends to a limit, while its derivative $\dot{V}$ tends to zero (Barbalat's lemma). Consequently, the tracking error, e, tends to zero asymptotically, as $t \to \infty$. Moreover, since the Lyapunov function (10.18) is radially unbounded, then the asymptotic convergence is global, that is, the closed-loop tracking error dynamics are globally asymptotically stable.

We have shown that the system state x globally asymptotically tracks the state x_{ref} of the reference model, and therefore, the system output $y = C\,x$ globally asymptotically tracks the reference model output $y_{ref} = C\,x_{ref}$. At the same time, the reference model dynamics are chosen such that y_{ref} tracks an external bounded command $y_{cmd}(t)$, with bounded errors. Therefore, y must also track y_{cmd} with bounded errors. The MIMO command tracking problem is solved.

The equation summary is given in Table 10.1, and the end result is stated in Theorem 10.1.

Theorem 10.1. *Consider the uncertain system dynamics in (10.9), operating under the MRAC controller (10.12), with the adaptive laws (10.23). Suppose that the matching condition (10.10) holds. Let the reference model (10.14) be driven by a bounded external command $y_{cmd}(t)$. Then, for any symmetric positive-definite matrices $\left(\Gamma_x,\ \Gamma_\Theta,\ Q_{ref}\right)$, all signals in the closed-loop system*

$$\begin{cases} \dot{x} = A\,x + B\,\Lambda\left(\hat{K}_x^T\,x - \hat{\Theta}^T\,\Phi(x_p) + \Theta^T\,\Phi(x_p)\right) + B_{ref}\,y_{cmd}(t) \\ \dot{x}_{ref} = A_{ref}\,x_{\text{ref}} + B_{ref}\,y_{cmd}(t) \\ \dot{\hat{K}}_x = -\Gamma_x\,x\left(x - x_{ref}\right)^T P\,B \\ \dot{\hat{\Theta}} = \Gamma_\Theta\,\Phi(x_p)\left(x - x_{ref}\right)^T P\,B \end{cases} \tag{10.25}$$

are uniformly ultimately bounded in time, where P represents the unique symmetric positive-definite solution of the algebraic Lyapunov equation (10.19). Moreover, the tracking error signal $e = x - x_{ref}$ is uniformly ultimately bounded, square integrable, and tends to the origin globally and asymptotically, that is, $\lim_{t\to\infty} \|e(t)\| = 0$. ■

In order to illustrate both usefulness and practicality of MRAC systems with integral feedback, we shall consider an example.

Table 10.1 Design summary for MRAC with integral action

Open-loop plant	$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + \Theta^T \Phi(x_p)\right)$
	$y = C_p x_p$
Integrated output tracking error and extended state	$\dot{e}_{Iy} = y - y_{cmd}, \quad x = \begin{pmatrix} e_{yI}^T & x_p^T \end{pmatrix}^T$
Open-loop extended plant	$\dot{x} = A x + B \Lambda \left(u + \Theta^T \Phi(x_p)\right) + B_{ref}\, y_{cmd}$
	$y = C x$
Reference model	$\dot{x}_{ref} = A_{\text{ref}}\, x_{\text{ref}} + B_{ref}\, y_{cmd}, \quad y_{ref} = C\, x_{ref}$
Tracking error	$e = x - x_{ref}$
Control input	$u = \hat{K}_x^T x - \hat{\Theta}^T \Phi(x_p)$
Algebraic Lyapunov equation	$P A_{ref} + A_{ref}^T P = -Q$
MRAC laws	$\dot{\hat{K}}_x = -\Gamma_x\, x\, e^T P B$
	$\dot{\hat{\Theta}} = \Gamma_\Theta\, \Phi(x_p)\, e^T P B$

Example 10.1 Aircraft Short-Period Dynamics and Control Longitudinal motion of a conventional aircraft is controlled by engine throttles and elevators (movable trailing edge tail surfaces, δ_e). While throttles are the primary inputs for regulating airspeed, deflecting elevators up or down will change the aircraft pitch rate $q = \dot{\theta}$ and, as a consequence, its orientation with respect to the horizon (the pitch angle, θ). At the same time, elevator movements will also affect the aircraft angle of attack α (the angle between the velocity vector and the fuselage longitudinal axis x). Figure 10.1 shows an aircraft sketch, with all the relevant degrees of freedom.

Coupled relations between α and q, driven by the elevator deflection δ_e, constitute the aircraft short-period dynamics. Assuming fixed throttle setting and constant airspeed, the aircraft short-period dynamics can be approximated by a second-order differential equation in the form

$$\underbrace{\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Z_\alpha}{V} & 1 + \frac{Z_q}{V} \\ M_\alpha & M_q \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \alpha \\ q \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{pmatrix}}_{B_p} \Lambda \left(\underbrace{\delta_e}_{u} + f(x_p) \right) \tag{10.26}$$

where α (rad) is the aircraft angle of attack, q (rad/s) is the pitch rate, V (ft/s) is the true airspeed (assumed constant), δ_e (rad) is the elevator deflection (the control input), and $\left(Z_\alpha,\ Z_q,\ Z_\delta,\ M_\alpha,\ M_q,\ M_\delta\right)$ are the aircraft stability derivatives [1, 2]. In (10.26), $\Lambda > 0$ represents a loss-of-control effectiveness, and $f\left(x_p\right)$ is the matched uncertainty in the system dynamics.

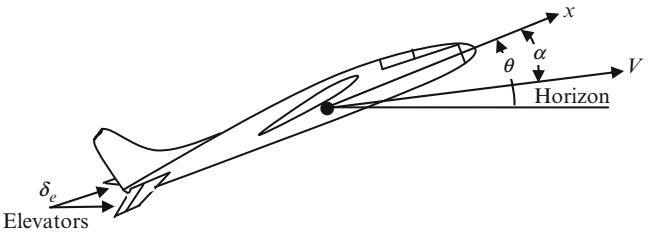


Fig. 10.1 Aircraft short-period motion in Example 10.1

We make use of generic transport aircraft (DC-8) cruise data from [1, p.712] to populate the short-period model

$$A_p = \begin{pmatrix} -0.8060 & 1.0 \\ -9.1486 & -4.59 \end{pmatrix}, \quad B_p = \begin{pmatrix} -0.04 \\ -4.59 \end{pmatrix}$$

and define the aircraft angle of attack α to be the system regulated output:

$$y = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C_p} x_p = \alpha$$

The open-loop system has its eigenvalues in the left half plane. So, the short-period dynamics are open-loop stable ($\omega_n = 3.58$ rad/s, $\zeta = 0.753$).

This model is augmented with the integrated output (angle of attack) tracking error. The extended open-loop matrices are

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -0.8060 & 1 \\ 0 & -9.1486 & -4.59 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ -0.04 \\ -4.59 \end{pmatrix}, \quad B_{ref} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$$

Assuming that these linear data are known, our next step is to construct a suitable reference model. We can use a linear control design technique to accomplish this task. Such an approach would allow us to automatically satisfy the required matching conditions (10.10) and at the same time to construct a reference model with the desired transient characteristics. We choose the linear quadratic regulator (LQR) method [3] as our baseline control design tool. Since the open-loop dynamics are already stable and sufficiently fast, we pick LQR weight matrices

$$Q_{LQR} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad R_{LQR} = 1$$

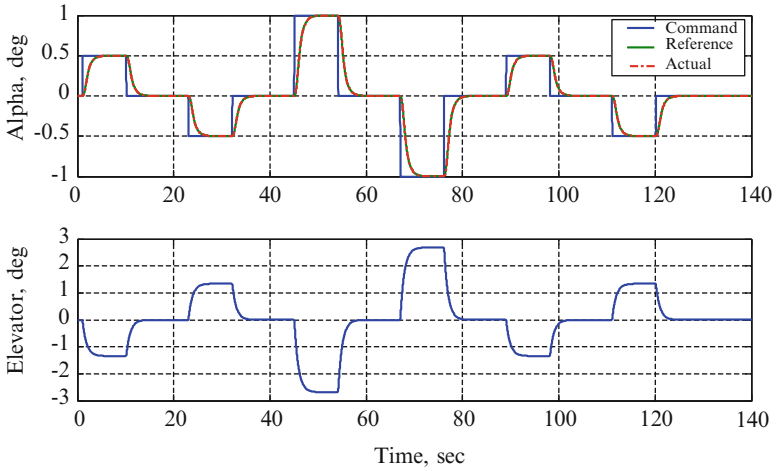


Fig. 10.2 Closed-loop baseline (no uncertainties) tracking performance in Example 10.1

and arrive at the desired reference model, which represents the baseline closed-loop short-period dynamics

$$A_{ref} = A + B \underbrace{\begin{pmatrix} 3.1623 & 1.1016 & 0.2152 \end{pmatrix}}_{-K_{LQR}^T} = \begin{pmatrix} 0 & 1 & 0 \\ -0.1328 & -0.8522 & 0.9910 \\ -14.5149 & -14.2048 & -5.5779 \end{pmatrix}$$

achieved via the LQR control feedback:

$$u_{LQR} = -K_{LQR}^T x$$

The reference model natural frequency and damping are quite close to those of the open-loop $(\omega_{n\,\mathrm{LQR}} = 3.57\ \mathrm{rad/s},\ \zeta_{LQR} = 0.734)$. The integrator pole represents the closed-loop system dominant eigenvalue. The pole is placed at $\lambda = -1.1873$ to enable adequate tracking performance with a reasonable control (elevator deflection) effort (see Fig. 10.2).

In this case, the reference and the actual system responses coincide with each other. This is to be expected since the system is simulated without uncertainties.

Next, we introduce linear state-dependent uncertainties into the system dynamics

$$f(x_p) = f(\alpha, q) = k_\alpha \alpha + k_q q$$

and choose $\Lambda = 0.5$, $k_\alpha = 1.5 M_\alpha$, and $k_q = 0.5 M_q$. Our particular selection corresponds to simultaneous changes in (a) the control effectiveness M_δ, (b) the static stability M_α, and (c) the pitch damping M_q. These uncertainties are intentionally chosen to destabilize the LQR closed-loop short-period open-loop

dynamics. In fact, with these uncertainties, the perturbed open-loop dynamics become

$$\dot{x} = A\,x + B\,\Lambda\left(u + k_\alpha\,\alpha + k_q\,q\right) = A_{ref}\,x + B\,\Lambda\left(u + \Lambda^{-1}\,K_{LQR}^T\,x + k_\alpha\,\alpha + k_q\,q\right)$$

and so, the ideal unknown controller gains are

$$K_{x\,\text{ideal}} = -\left(\Lambda^{-1}\,K_{LQR}^T + \begin{pmatrix} 0 & k_\alpha & k_q \end{pmatrix}\right)^T = \begin{pmatrix} 6.3246 & 15.9261 & 2.7254 \end{pmatrix}^T$$

In other words, if we knew the uncertainties, then the linear feedback

$$u_{ideal} = K_{x\,ideal}^T\,x$$

would have enforced the desired reference dynamics.

The next step is to design a MRAC system in order to recover the desired closed-loop performance, without any information about the parametric uncertainties. Since $f\left(x_p\right)$ is linear in x_p, only the adaptive gains $\hat{K}_x$ are required. Here, the design "tuning knobs" consist of two symmetric positive definite (3×3) – matrices Q and Γ_x. After several iterations, we have selected

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}, \quad \Gamma_x = \begin{pmatrix} 2000 & 0 & 0 \\ 0 & 2000 & 0 \\ 0 & 0 & 200 \end{pmatrix}$$

Our iterative design focus was on reducing unwanted transient oscillations, while providing adequate command tracking performance. Utilizing MRAC design equations from Table 10.1 and with the uncertainties turned on, the corresponding simulated closed-loop system tracking performance data are shown below (Fig. 10.3).

Clearly, the MRAC design is able to recover the baseline closed-loop dynamics. However, the control effort is significantly larger than before, and the uncertainties are the driving factor. Dynamics of the corresponding adaptive gains are shown in Fig. 10.4.

The three gains approach their ideal values. This is a "bonus," since parameter convergence is not guaranteed by the MRAC design.

In order to demonstrate good tracking without parameter convergence, we select $y_{cmd} = \sin(0.1\,t)$ and simulate the closed-loop system without any other changes. As expected, the output tracking performance remains of good quality (Fig. 10.5).

On the other hand, the adaptive gains are different from the ideal values $K_{x\,ideal}$ that are defined by the corresponding matching conditions (Fig. 10.6).

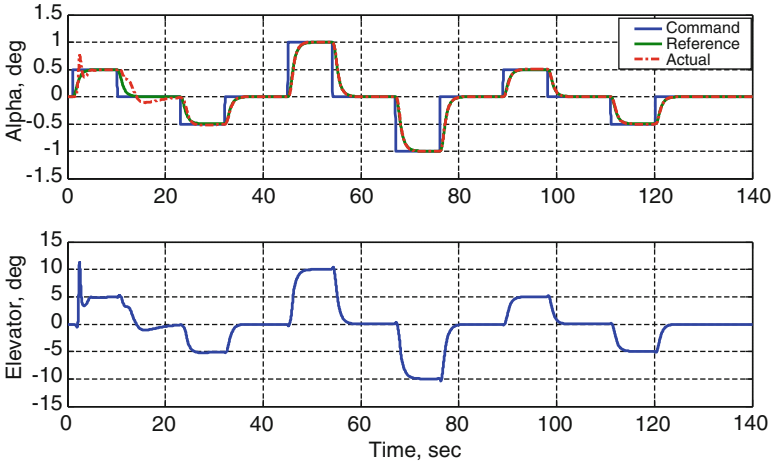


Fig. 10.3 Closed-loop performance recovery under MRAC system in Example 10.1

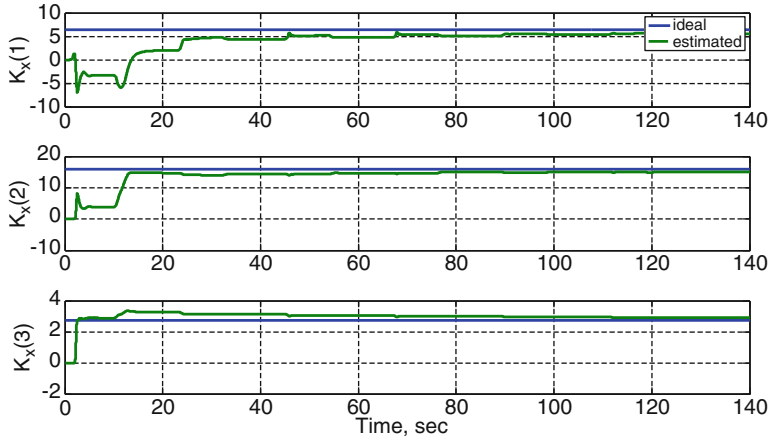


Fig. 10.4 Evolution of adaptive gains in Example 10.1

Parameter convergence in adaptive control depends on the persistency of excitation (PE) conditions [4, 5]. Basically, the external command needs to "persistently excite" the closed-loop system dynamics. For linear dynamical systems with linear-in-parameter uncertainties (such as those considered in this example), the PE conditions are satisfied if the system external command is chosen as a sum of sinusoids with different frequencies. Then, a single frequency would give exponential convergence of two adaptive gains to their corresponding unknown constant ideal values. For nonlinear systems, this rule no longer holds and the generic PE conditions are hard to verify numerically. □

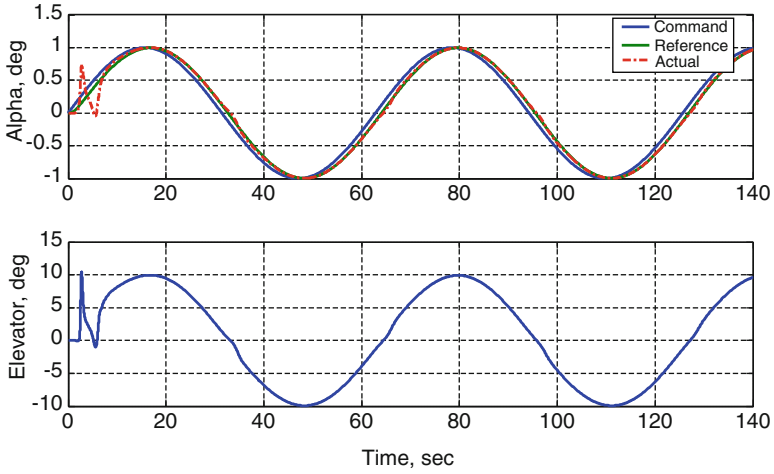


Fig. 10.5 Output tracking of a sinusoidal command in Example 10.1

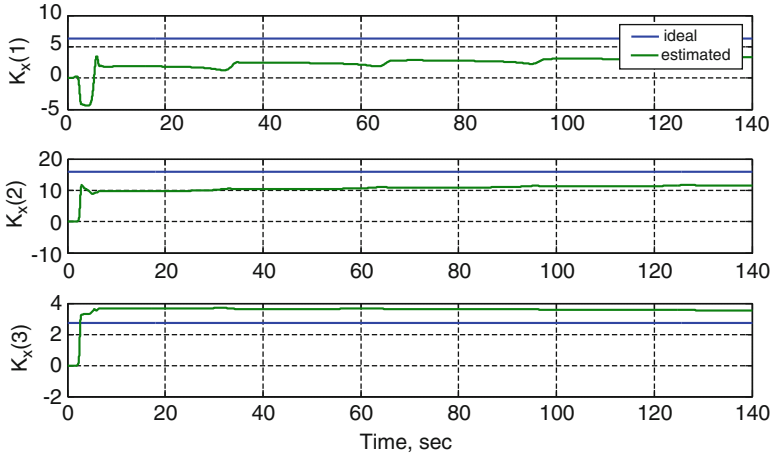


Fig. 10.6 Evolution of adaptive gains during tracking of a sinusoid in Example 10.1

10.3 MRAC Augmentation of an Optimal Baseline Controller

The adaptive design developed in the previous section can be modified to augment a baseline linear controller with the (Proportional + Integral) (PI) feedback architecture. The rational for using an augmentation approach (as oppose to all adaptive) stems from the fact that in most realistic applications, a system may already have a baseline controller, which often is designed to contain proportional as well as integral feedback connections. Such a baseline controller would have been intended to operate under nominal conditions (no uncertainties), where it would asymptotically

reject constant unknown disturbances and track constant commands with zero errors. If adding uncertainties destroys the expected baseline closed-loop performance, then one might attempt to recover the desired performance by augmenting the baseline controller with an adaptive element.

We consider the same class of n-dimensional MIMO nonlinear systems with m controls (as defined in (10.1)), whose plant dynamics are linearly parameterized, the uncertainties satisfy matching conditions, and the system state is measurable (i.e., available online for control synthesis). The system dynamics are

$$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + f\left(x_p\right)\right) \tag{10.27}$$

where n_p and m are the dimensions of the system state x_p and of the control u, respectively. Also, we assume that $A_p \in R^{n_p \times n_p}$ and $B_p \in R^{n \times m}$ are known, while $\Lambda \in R^{m \times m}$ is an unknown diagonal matrix with strictly positive diagonal elements λ_i. The pair $\left(A_p, \left(B_p \Lambda\right)\right)$ is presumed controllable, and the constant uncertainty Λ is introduced to model possible imperfections in the system control channels.

The unknown nonlinear function $f\left(x_p\right) : R^{n_p} \rightarrow R^m$ represents the system matched uncertainty. It is assumed that this function can be written as a linear combination of N known basis functions, with unknown constant coefficients:

$$f\left(x_p\right) = \Theta^T \Phi\left(x_p\right) \tag{10.28}$$

In (10.28), $\Theta \in R^{N \times m}$ is the unknown constant matrix of ideal parameters, and $\Phi\left(x_p\right) \in R^N$ represents the known locally Lipschitz-continuous regressor vector. Thus, we consider a generic class of MIMO systems in the form

$$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + \Theta^T \Phi\left(x_p\right)\right) \tag{10.29}$$

with the regulated output

$$y = C_p x_p + D_p \Lambda \left(u + \Theta^T \Phi\left(x_p\right)\right) \tag{10.30}$$

where $C_p \in R^{m \times n_p}$ and $D_p \in R^{m \times m}$ are known and constant.

Let $y_{cmd}(t) \in R^m$ denote a bounded command for the system output $y \in R^m$ to follow. This task is to be accomplished using the system control input $u \in R^m$, in the form of a full state feedback.

We define the output tracking error

$$e_y(t) = y(t) - y_{cmd}(t) \tag{10.31}$$

its integral e_{yI}

$$\dot{e}_{yI} = e_y = y - y_{cmd} \tag{10.32}$$

and formulate the extended open-loop dynamics

$$\underbrace{\begin{pmatrix}\dot{e}_{yI}\\ \dot{x}_p\end{pmatrix}}_{\dot{x}}=\underbrace{\begin{pmatrix}0_{m\times m} & C_p\\ 0_{n_p\times m} & A_p\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}e_{yI}\\ x_p\end{pmatrix}}_{x}+\underbrace{\begin{pmatrix}D_p\\ B_p\end{pmatrix}}_{B}\Lambda\left(u+f\left(x_p\right)\right)+\underbrace{\begin{pmatrix}-I_{m\times m}\\ 0_{n_p\times m}\end{pmatrix}}_{B_{ref}}y_{cmd} \tag{10.33}$$

or, equivalently,

$$\dot{x}=A\,x+B\,\Lambda\left(u+\Theta^T\,\Phi\left(x_p\right)\right)+B_{ref}\,y_{cmd} \tag{10.34}$$

In terms of (10.34), the system regulated output y in (10.30) can be written as

$$y=\underbrace{(0\;\;C_p)}_{C}\underbrace{\begin{pmatrix}e_{yI}\\ x_p\end{pmatrix}}_{x}+\underbrace{D_p}_{D}\Lambda\left(u+\Theta^T\,\Phi\left(x_p\right)\right)u=Cx+D\Lambda\left(u+\Theta^T\,\Phi\left(x_p\right)\right)u \tag{10.35}$$

The control problem of interest is bounded tracking in the presence of the system constant parametric uncertainties Λ and Θ. Specifically, we need to design the control input u, so that the system regulated output y tracks any bounded time-varying command y_{cmd}, with bounded tracking errors, while the rest of the signals in the corresponding closed-loop dynamics remain bounded.

We begin with the design of a baseline linear controller. Setting $\Lambda=I_{m\times m}$, $\Theta=0_{N\times m}$ in (10.34), results in the linear baseline open-loop dynamics:

$$\begin{aligned}\dot{x}&=A\,x+B\,u+B_{ref}\,y_{cmd}\\ y&=C\,x+D\,u\end{aligned} \tag{10.36}$$

Assuming constant command y_{cmd}, we can use the linear quadratic regulator (LQR) method, with Proportional + Integral (PI) feedback connections, to design the baseline LQ optimal control law, in the form of an LQR PI servomechanism. This design is outlined below.

We first calculate the optimal stabilizing controller for

$$\dot{z}=A\,z+B\,v \tag{10.37}$$

where

$$z=\dot{x}=\begin{pmatrix}\dot{e}_{yI}\\ \dot{x}_p\end{pmatrix},\quad v=\dot{u} \tag{10.38}$$

and the control input v is designed to minimize the linear quadratic cost index

$$J(v)=\int_0^\infty\left(z^TQ\,z+v^TR\,v\right)dt \tag{10.39}$$

with the appropriately selected symmetric positive-definite matrices Q and R. It is well-known that the corresponding optimal LQR solution is given in feedback form

$$v = \dot{u} = -\underbrace{R^{-1} B^T P}_{K_x^T} z = -\begin{pmatrix} K_I & K_P \end{pmatrix} \begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix} \tag{10.40}$$

In (10.40), P is the unique symmetric positive-definite solution of the algebraic Riccati equation

$$A^T P + P A + Q - P B R^{-1} B^T P = 0 \tag{10.41}$$

which is solved using an appropriately chosen $Q = Q^T \geq 0$. Integrating (10.40) yields the baseline LQR PI controller

$$u_{bl} = -K_x^T x = -K_I e_{yI} - K_P x = K_I \frac{(y_{cmd} - y)}{s} - K_P x_p \tag{10.42}$$

where the optimal gain matrix

$$K_x^T = \begin{pmatrix} K_I & K_P \end{pmatrix} \tag{10.43}$$

is partitioned into the integral gain K_I and the proportional gain K_P. The corresponding baseline LQR PI control block diagram is shown in Fig. 10.7.

In the presence of the system uncertainties Λ and Θ, the baseline tracking performance will often deteriorate. In order to restore the expected baseline behavior, we augment the baseline system with an adaptive element. This process consists of (a) the reference model definition, (b) the tracking dynamics formulation, and (c) the design of adaptive laws.

First, we define the reference model to represent the baseline closed-loop system dynamics, which are obtained by substituting the baseline controller (10.42) into the linear system (10.36). The resulting reference model dynamics become

$$\dot{x}_{ref} = A_{ref} x_{ref} + B_{ref} y_{cmd}, \quad y_{ref} = C_{ref} x_{ref} \tag{10.44}$$

where

$$A_{ref} = A - B K_x^T, \quad C_{ref} = C - D K_x^T \tag{10.45}$$

and A_{ref} is Hurwitz by design.

Then, we synthesize the total control input as the sum of the baseline LQR PI component (10.42) and its adaptive augmentation u_{ad} (to be constructed):

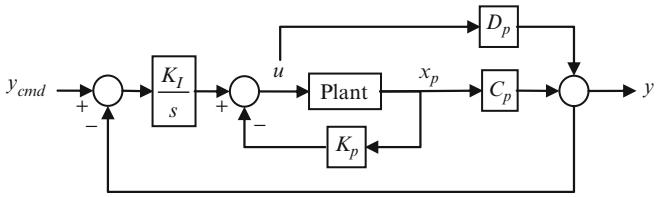


Fig. 10.7 Baseline servomechanism LQR PI control block diagram

$$u = \underbrace{-K_x^T x}_{u_{bl}} + u_{ad} = u_{bl} + u_{ad} \tag{10.46}$$

Substituting (10.46) into the original system dynamics (10.34) gives

$$\dot{x} = A_{ref}\, x + B\,\Lambda \left(u_{ad} + \underbrace{\overbrace{\left(I_{m\times m} - \Lambda^{-1}\right)}^{K_u^T} u_{bl} + \Theta^T \Phi\left(x_p\right)}_{\bar{\Theta}^T \bar{\Phi}\left(u_{bl}, x_p\right)} \right) + B_{ref}\, y_{cmd}$$

$$y = C_{ref}\, x + D\,\Lambda \left(u_{ad} + \bar{\Theta}^T \bar{\Phi}\left(u_{bl},\, x_p\right) \right) \tag{10.47}$$

Or, equivalently,

$$\begin{aligned} \dot{x} &= A_{ref}\, x + B\,\Lambda \left(u_{ad} + \bar{\Theta}^T \bar{\Phi}\left(u_{bl},\, x_p\right) \right) + B_{ref}\, y_{cmd} \\ y &= C_{ref}\, x + D\,\Lambda \left(u_{ad} + \bar{\Theta}^T \bar{\Phi}\left(u_{bl},\, x_p\right) \right) \end{aligned} \tag{10.48}$$

with the redefined regressor vector

$$\bar{\Phi}\left(u_{bl},\, x_p\right) = \begin{pmatrix} u_{bl}^T & \Phi^T\left(x_p\right) \end{pmatrix}^T \tag{10.49}$$

and with the extended matrix of unknown/ideal parameters:

$$\bar{\Theta} = \begin{pmatrix} K_u^T & \Theta^T \end{pmatrix}^T \tag{10.50}$$

The adaptive component u_{ad} is chosen to dominate the system matched uncertainty $\bar{\Theta}^T \bar{\Phi}\left(u_{bl},\, x_p\right)$

$$u_{ad} = -\hat{\bar{\Theta}}^T \bar{\Phi}\left(u_{bl},\, x_p\right) \tag{10.51}$$

where $\hat{\bar{\Theta}} \in R^{(n+N)\times m}$ is the matrix of adaptive parameters. Substituting (10.51) into (10.48) results in

$$\dot{x} = A_{ref}\, x - B\, \Lambda \underbrace{\left(\hat{\bar{\Theta}} - \bar{\Theta}\right)^T}_{\Delta\bar{\Theta}} \bar{\Phi} + B_{ref}\, y_{cmd}$$
$$y = C_{ref}\, x - D\, \Lambda\, \Delta\bar{\Theta}^T\, \bar{\Phi} \tag{10.52}$$

where

$$\Delta\bar{\Theta} = \hat{\bar{\Theta}} - \bar{\Theta} \tag{10.53}$$

is the matrix of parameter estimation errors. We now introduce the state tracking error

$$e = x - x_{ref} \tag{10.54}$$

and calculate the tracking error dynamics by subtracting the reference system dynamics (10.44) from the extended open-loop system dynamics (10.52):

$$\dot{e} = A_{ref}\, e - B\, \Lambda\, \Delta\bar{\Theta}^T \bar{\Phi} \tag{10.55}$$

In order to design MRAC laws and at the same time enforce closed-loop stability of the error dynamics, we consider a radially unbounded quadratic Lyapunov function candidate such as

$$V\left(e,\ \Delta\bar{\Theta}\right) = e^T\, P_{ref}\, e + \text{trace}\left(\Delta\bar{\Theta}^T\, \Gamma_{\bar{\Theta}}^{-1}\, \Delta\bar{\Theta}\, \Lambda\right) \tag{10.56}$$

where elements of $\Gamma_{\bar{\Theta}} = \Gamma_{\bar{\Theta}}^T > 0$ represent rates of adaptation and $P_{ref} = P_{ref}^T > 0$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation

$$A_{ref}^T P_{ref} + P_{ref} A_{ref} = -Q_{ref} \tag{10.57}$$

with some appropriately chosen matrix $Q_{ref} = Q_{ref}^T > 0$. Time-differentiating V, along the trajectories of (10.55), gives

$$\dot{V}\left(e, \Delta\bar{\Theta}\right) = -e^T Q_{ref}\, e - 2e^T P_{ref} B \Lambda \Delta\bar{\Theta}^T \bar{\Phi} + 2\text{trace}\left(\Delta\bar{\Theta}^T\, \Gamma_{\bar{\Theta}}^{-1}\, \dot{\hat{\bar{\Theta}}}\, \Lambda\right) \tag{10.58}$$

Applying the vector trace identity

$$a^T\, b = trace\left(b\, a^T\right) \tag{10.59}$$

further yields

$$\dot{V}\left(e,\ \Delta\bar{\Theta}\right) = -e^T Q_{ref}\, e + 2\,\text{trace}\left(\Delta\bar{\Theta}^T \left\{\Gamma_{\bar{\Theta}}^{-1}\,\dot{\hat{\bar{\Theta}}} - \bar{\Phi}\, e^T P_{ref}\, B\right\}\Lambda\right) \tag{10.60}$$

If adaptive laws are selected in the form

$$\dot{\hat{\bar{\Theta}}} = \Gamma_{\bar{\Theta}}\,\bar{\Phi}\left(u_{bl},\, x_p\right)\, e^T P_{ref}\, B, \tag{10.61}$$

then

$$\dot{V}\left(e,\ \Delta\bar{\Theta}\right) = -e^T Q_{ref}\, e \leq 0 \tag{10.62}$$

which immediately proves uniform ultimate boundedness of $\left(e,\ \Delta\bar{\Theta}\right)$.

Moreover, it follows from (10.62) that the tracking error signal is square integrable, $e \in L_2$. Since $y_{cmd} \in L_\infty$, then $x_{ref} \in L_\infty$, and consequently, $x \in L_\infty$ and $\left(u_{bl},\ x_p\right) \in L_\infty$. Since the ideal (unknown) matrix of parameters $\bar{\Theta}$ is constant and the estimation errors $\Delta\bar{\Theta}$ are bounded, then their estimated values are bounded as well, that is, $\hat{\bar{\Theta}} \in L_\infty$. Since components of the regressor vector $\bar{\Phi}\left(u_{bl},\, x_p\right)$ are locally Lipschitz continuous, and $\left(u_{bl},\ x_p\right) \in L_\infty$, then the regressor components are bounded. Hence, $u \in L_\infty$ and $\dot{x} \in L_\infty$. Thus, $\dot{e} \in L_\infty$, which implies that $\ddot{V} \in L_\infty$. Therefore, $\dot{V}$ is a uniformly continuous function of time. Since V is lower bounded, $\dot{V} \leq 0$, and $\dot{V}$ is uniformly continuous, then V tends to a limit, while its derivative $\dot{V}$ tends to zero (see Barbalat's lemma, Chap. 8). Consequently, the tracking error e tends to zero asymptotically, as $t \to \infty$.

Moreover, since the Lyapunov function (10.56) is radially unbounded, then the asymptotic convergence is global, that is, the closed-loop tracking error dynamics (10.55) are globally asymptotically stable.

Using the error dynamics (10.55), it is easy to check that $\ddot{e} \in L_\infty$. Then, $\dot{e}(t)$ is uniformly continuous. Since in addition $e(t)$ tends to zero, then using Barbalat's lemma, we conclude that $\lim_{t\to\infty} \|\dot{e}(t)\| = 0$. Consequently,

$$\lim_{t\to\infty} \left\|\Delta\bar{\Theta}^T(t)\ \bar{\Phi}(u_{bl}(t),\ x(t))\right\| = 0 \tag{10.63}$$

and

$$y = C\,x - D\,\Lambda\,\underbrace{\left(\Delta\bar{\Theta}^T\,\bar{\Phi}\right)}_{\to 0} \to C_{ref}\,x \to C_{ref}\,x_{ref} = y_{ref} \tag{10.64}$$

We have proven that for any bounded command y_{cmd}, the closed-loop system output from (10.52) globally asymptotically tracks the reference model output from (10.44), as $t \to \infty$. At the same time, the reference model dynamics (10.44) are chosen such that y_{ref} tracks any external bounded command $y_{cmd}(t)$, with bounded

errors. Therefore, y must also track y_{cmd} with bounded errors. The MIMO command tracking problem is solved.

The adaptive laws (10.61) can be written in terms of the system original parameters. Partition

$$\Gamma_{\bar{\Theta}} = \begin{pmatrix} \Gamma_u & 0_{n\times m} \\ 0_{N\times m} & \Gamma_\Theta \end{pmatrix} \tag{10.65}$$

where $(\Gamma_u,\ \Gamma_\Theta)$ denote rates of adaptation for uncertainties that correspond to x and $\Phi(x_p)$. Using (10.49), (10.50), and (10.65), the adaptive laws (10.61) become

$$\begin{aligned} \dot{\hat{K}}_u &= \Gamma_u\, u_{bl}\, e^T P_{ref}\, B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta\, \Phi(x_p)\, e^T P_{ref}\, B \end{aligned} \tag{10.66}$$

Also, the (LQR PI Baseline + Adaptive) total control input (10.46) is

$$u = u_{bl} + u_{ad} = \underbrace{\boxed{-K_x^T x}}_{u_{bl}=\text{Baseline}} + \underbrace{\boxed{\left[-\hat{K}_u^T u_{bl} - \hat{\Theta}^T \Phi(x_p)\right]}}_{u_{ad}=\text{Adaptive Augmentation}} \tag{10.67}$$

or, equivalently,

$$\begin{aligned} u &= (I_{m\times m} - \hat{K}_u^T) u_{bl} - \hat{\Theta}^T \Phi(x_p) = -(I_{m\times m} - \hat{K}_u^T) K_x^T x - \hat{\Theta}^T \Phi(x_p) \\ &= (I_{m\times m} - \hat{K}_u^T) \left(K_I \frac{(y_{cmd} - y)}{s} - K_P x_p \right) - \hat{\Theta}^T \Phi(x_p) \end{aligned} \tag{10.68}$$

Table 10.2 summarizes the developed adaptive augmentation procedure of a LQR PI baseline controller.

By design, this controller does not have a feedforward component. Also, note that in the adaptive laws (10.66), the parameter initial values are arbitrary, and as such, they can be set to zero. The following flight control design example illustrates the developed methodology.

Example 10.2 Adaptive Augmentation Design for DC-8 Short-Period Dynamics In Example 10.1, we designed a baseline optimal (LQR PI) controller for regulating short-period dynamics of the DC-8 transport aircraft. Our reference model was selected to represent the closed-loop system that was achieved under the baseline controller. Matched uncertainties were introduced to destabilize the baseline system. After that, we constructed an MRAC controller to recover the desired reference closed-loop performance, with the uncertainties turned on.

We now take a different approach. Instead of using an all-adaptive control solution, we demonstrate how to achieve the same closed-loop performance recovery by utilizing an adaptive augmentation design from Table 10.2. Such an

Table 10.2 MRAC augmentation of a LQR PI baseline system

Open-loop plant	$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + \Theta^T \Phi(x_p)\right)$
	$y = C_p x_p + D_p \Lambda \left(u + \Theta^T \Phi(x_p)\right)$
Integrated output tracking error and extended state	$\dot{e}_{I_y} = y - y_{cmd},\ x = \begin{pmatrix} e_{yI}^T & x_p^T \end{pmatrix}^T$
Open-loop extended plant	$\dot{x} = A\,x + B\,\Lambda \left(u + \Theta^T \Phi(x_p)\right) + B_{ref}\, y_{cmd}$
	$y = C\,x + D\,\Lambda \left(u + \Theta^T \Phi(x_p)\right)$
Reference model	$\dot{x}_{\text{ref}} = A_{ref}\, x_{ref} + B_{ref}\, y_{cmd}$
	$y_{ref} = C_{ref}\, x_{ref}$
Tracking error	$e = x - x_{ref}$
Riccati equation for LQR PI controller	$A^T P + P A - P B R^{-1} B^T P + Q = 0$
Baseline control input	$u_{\text{bl}} = -R^{-1} P B x$
Lyapunov equation for adaptive laws	$P_{ref} A_{ref} + A_{ref}^T P_{ref} = -Q_{ref}$
Total control input	$u = \left(I_{m\times} - \hat{K}_u^T\right) u_{bl} - \hat{\Theta}^T \Phi(x_p)$
MRAC laws	$\dot{\hat{K}}_u = \Gamma_u\, u_{bl}\, e^T P_{ref} B$
	$\dot{\hat{\Theta}} = \Gamma_\Theta\, \Phi(x_p)\, e^T P_{ref} B$

Table 10.3 Adaptive augmentation parameters for DC-8 short-period dynamics in Example 10.2

Q matrix for adaptive laws	$Q_{ref} = \text{diag}(100,\ 100,\ 100)$
Rates of adaptation	$\Gamma_u = \Gamma_\Theta = 800$
Regressor vector	$\Phi(x_p) = \begin{pmatrix} \alpha & q \end{pmatrix}^T$

approach would allow us to retain the baseline controller, instead of performing a complete redesign of the system.

After a few design iterations, we have selected appropriate values for adaptive tuning "knobs" (Table 10.3).

Using the design equations form Table 10.2, the system closed-loop dynamics are simulated with the uncertainties from Example 10.1. Figure 10.8 shows the results.

In comparison to the all-adaptive solution (see Fig. 10.3), the adaptive augmentation design also yields adequate tracking performance and a similar to the previous case control activity. There are also three adaptive gains, whose dynamics along with their corresponding ideal (unknown) values are shown below (Fig. 10.9).

Notwithstanding parameter convergence, the adaptive gains are well-behaved and remain bounded throughout the maneuver, as predicted by the theory. □

Let us now elaborate on the usefulness of an augmentation-based control design approach. In control engineering applications, a control designer is often faced with a preexisting controller, which constitutes and provides the baseline (i.e., expected) closed-loop tracking performance. Because of that, the control task at hand is to enhance the baseline system performance instead of replacing it with yet another system. Our adaptive augmentation procedure aims exactly at solving this particular task. Using control-theoretic arguments, we have developed a (Baseline + Adaptive) control system, capable of restoring the desired tracking characteristics

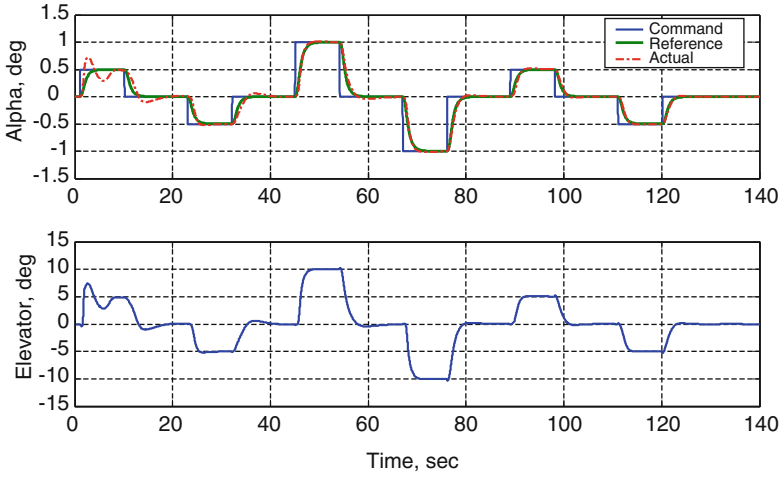


Fig. 10.8 Tracking performance and control effort in Example 10.2

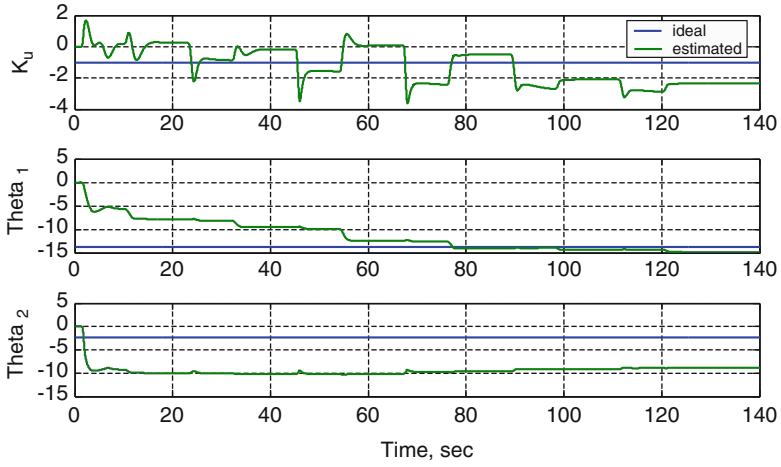


Fig. 10.9 Adaptive gains in Example 10.2

when matched uncertainties are prevalent in the system dynamics. Without the uncertainties, the system resorts to the baseline controller, while its adaptive component becomes inactive.

The overall (Baseline + Adaptive) control block diagram is shown in Fig. 10.10.

The red-dotted line in the figure denotes the adaptive nature of the gains $\hat{K}_u$ and $\hat{\Theta}$, whose dynamics are driven by the tracking error $e = x - x_{ref}$, and according to the adaptive laws (10.66), where x is the state of the extended system (10.33). Per

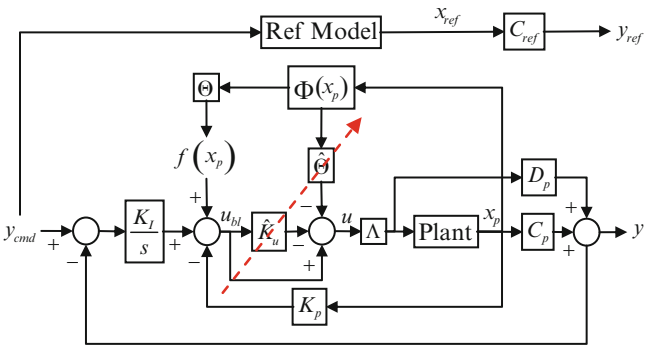


Fig. 10.10 Block diagram: adaptive augmentation of a baseline PI controller

design, the (Baseline + Adaptive) controller will force the system output y asymptotically track the reference model output y_{ref}, in spite of the matched uncertainties $f(x_p) = \Theta^T \Phi(x_p)$ and the unknown control gain Λ.

We have constructed the reference model to represent the desired closed-loop system operating under the baseline PI controller:

$$u_{bl} = -K_P x_p - K_I \left(\frac{y - y_{cmd}}{s} \right)$$

If there are no uncertainties and if the adaptive gains are initialized at zero, the tracking error will vanish asymptotically. Consequently, the adaptive gains will be constant and small. Then, the adaptive component

$$u_{ad} = -\hat{K}_u^T u_{bl} - \hat{\Theta}^T \Phi(x_p)$$

will become small as well, and as a result, the system will operate mostly under the baseline controller.

In the presence of uncertainties, the adaptive component becomes active, and it will provide an incremental signal (augmentation) to the baseline PI controller. In other words, anytime when the tracking error is sufficiently large, the total control signal is

$$u = u_{bl} + u_{ad}$$

which represents the (Baseline + Adaptive) architecture shown in Fig. 10.10.

It is interesting to note that if the system uncertainties fade away after being active, the adaptive gains will "freeze" and their values will remain constant until the tracking error becomes nonzero again. However, the adaptive component will not be necessarily zero. In fact, the adaptive signal u_{ad} becomes representative of a nonlinear controller with fixed gains, and as such, it will continue to add nonzero values to the baseline controller u_{bl}.

10.4 Summary

We have demonstrated how to embed fixed gain linear integral controllers into MRAC design. This leads to adaptive systems with integral action and provides a capability of tracking time-varying bounded commands without feedforward connections. We have also illustrated the design steps and its associated benefits using short-period dynamics of a generic transport aircraft.

In essence, we have offered a design procedure to combine a baseline linear (Proportional + Integral) controller with an MRAC system. The specific MRAC augmentation method discussed in this chapter allows a designer to merge a linear baseline system with an adaptive controller yet without "canceling" the former. Such an architecture is relevant in industrial applications where stability, performance, and robustness of preexisting baseline controllers can be enhanced through direct adaptation. This would result in the preservation and a recovery of the system baseline closed-loop performance, while operating in the presence of significant uncertainties that may exist in the process dynamics.

10.5 Exercises

Exercise 10.1. Verify that the transfer function $G_{ref}(s) = C\left(sI_{n\times n} - A_{ref}\right)^{-1} B_{ref}$ in (10.15) and (10.44) have the unity DC gain, that is, $G_{ref}(0) = -C\,A_{ref}^{-1} B_{ref} = I_{m\times m}$.

Exercise 10.2. Table 10.2 presents an adaptive augmentation design. Show that an alternative way to construct an adaptive augmentation of a baseline linear controller $u_{bl} = -K_x^T\, x$ is to start with the adaptive controller (10.12), and then, initialize the adaptive gain $\hat{K}_x$ from (10.23) such that $\hat{K}_x(0) = K_x$.

Exercise 10.3. Prove (10.63).

Exercise 10.4. For the delta wing dynamics from Example 9.3, assume that A and B matrices are known. The system uncertainties are represented by Λ and $f(x)$. The system regulated output is the bank angle φ. Design a baseline LQR PI controller u_{bl}, and then, augment it with an MRAC signal u_{ad}, via equations from Table 10.2. Use Table 10.1 to design a pure adaptive controller, and initialize its adaptive state gains at their corresponding baseline (LQR PI) values. Simulate both controllers. Compare and discuss their tracking performance and the associated control efforts.

References

1. McRuer, D., Ashkenas, I., Graham, D.: Aircraft Dynamics and Automatic control. Princeton University Press, Princeton (1990)
2. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation. Wiley, Hoboken (1992)
3. Anderson, B.D.O., Moore, J.B.: Optimal Control, Linear Quadratic Methods. Dover-Publications, Mineola (2007)
4. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Control. Dover, New York (2005)
5. Loannou, P., Fidan, P.: Adaptive Control Tutorial, SIAM, Advances in Design and control, SIAM, PA (2006)

Chapter 11
Robust Adaptive Control

11.1 MRAC Design in the Presence of Bounded Disturbances

Our starting point is the MIMO dynamical system

$$\dot{x} = A_{ref}\, x + B\, \Lambda \left(u + \Theta^T\, \Phi(x)\right) + B_{ref}\, y_{cmd} + \xi(t) \tag{11.1}$$

whose regulated output is

$$y = C_{ref}\, x \tag{11.2}$$

The system is operating in the presence of a uniformly bounded time-dependent disturbance $\xi(t) \in R^n$,

$$\|\xi(t)\| \leq \xi_{\max} \tag{11.3}$$

with its known and constant upper bound $\xi_{\max} \geq 0$. The system matched uncertainties are represented by a diagonal positive-definite matrix $\Lambda \in R^{m \times m}$ and a constant matrix $\Theta \in R^{N \times m}$. We assume that the constant matrices $\left(A_{ref},\ B,\ B_{ref},\ C_{ref}\right)$ are known, the pair $\left(A_{ref},\ B\,\Lambda\right)$ is controllable, and A_{ref} is Hurwitz.

The control objective is to design a state feedback MRAC system to enable bounded tracking of the reference model dynamics

$$\begin{aligned} \dot{x}_{ref} &= A_{ref}\, x_{ref} + B_{ref}\, y_{cmd} \\ y_{ref} &= C_{ref}\, x_{ref} \end{aligned} \tag{11.4}$$

with the output y_{ref}. The reference model is driven by a bounded time-dependent command $y_{cmd} \in R^m$, and the control goal consists of finding a state feedback controller u to force the system output y track a command y_{cmd}, in the presence of the

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_11,

system parametric uncertainties and while keeping the rest of the signals uniformly bounded in time.

Based on (11.1), we choose the control input to be

$$u = -\hat{\Theta}^T \Phi(x) \tag{11.5}$$

where $\hat{\Theta} \in R^{N \times m}$ is the matrix of adaptive parameters to be determined at a later time. Substituting (11.5) into (11.1) gives

$$\dot{x} = A_{ref}\, x - B\, \Lambda\, \Delta\Theta^T\, \Phi(x) + B_{ref}\, y_{cmd} + \xi(t) \tag{11.6}$$

where

$$\Delta\Theta = \hat{\Theta} - \Theta \tag{11.7}$$

is the matrix of the parameter estimation errors. Let

$$e = x - x_{ref} \tag{11.8}$$

be the state tracking error. Subtracting the reference model dynamics (11.4) from that of the system (11.1) yields the tracking error dynamics:

$$\dot{e} = A_{ref}\, e - B\, \Lambda\, \Delta\Theta^T\, \Phi(x) + \xi(t) \tag{11.9}$$

A radially unbounded quadratic Lyapunov function candidate is selected in the familiar form

$$V(e,\ \Delta\Theta) = e^T\, P\, e + \text{trace}\left(\Delta\Theta^T\, \Gamma_\Theta^{-1}\, \Delta\Theta\, \Lambda\right) \tag{11.10}$$

where $\Gamma_\Theta = \Gamma_\Theta^T > 0$ denotes constant rates of adaptation and $P = P^T > 0$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation

$$P\, A_{ref} + A_{ref}^T P = -Q \tag{11.11}$$

with $Q = Q^T > 0$. Time-differentiating V, along the trajectories of (11.9), gives

$$\begin{aligned} \dot{V}(e,\ \Delta\Theta) &= -e^T\, Q\, e \\ &- 2\, e^T\, P\, B\, \Lambda\, \Delta\Theta^T\, \Phi(x) + 2\, e^T\, P\, \xi(t) + 2\, \text{trace}\left(\Delta\Theta^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}}\, \Lambda\right) \end{aligned} \tag{11.12}$$

Applying the vector trace identity

$$a^T\, b = trace\left(b\, a^T\right) \tag{11.13}$$

further yields

$$\dot{V}(e,\Delta\Theta) = -e^T Q e + 2\text{trace}\left(\Delta\Theta^T \left\{\Gamma_\Theta^{-1}\dot{\hat{\Theta}} - \Phi e^T P B\right\}\Lambda\right) + 2e^T P \xi(t) \quad (11.14)$$

Suppose that we use the same adaptive laws as in the previous sections, that is,

$$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x)\ e^T P B \quad (11.15)$$

Then,

$$\dot{V}(e,\Delta\Theta) = -e^T Q e + 2e^T P \xi(t) \le -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\,\lambda_{\max}(P)\,\xi_{\max} \quad (11.16)$$

and, consequently, $\dot{V} < 0$ outside of the set

$$E_0 = \left\{(e,\ \Delta\Theta): \quad \|e\| \le 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\xi_{\max} = e_0\right\} \quad (11.17)$$

According to [1, Theorem 4.18, p. 172], trajectories $e(t)$ of the error dynamics (11.9) enter a compact set $(\Omega_0 \supset E_0) \subset R^n$ in finite time and will remain there for all future times. However, Ω_0 is not compact in the $(e,\ \Delta\Theta)$ space. In fact, Ω_0 is unbounded since the parameter estimation errors $\Delta\Theta$ are not restricted at all. Therefore, inside Ω_0, $\dot{V}$ can become positive, and, as a consequence, the parameter errors $\Delta\Theta$ can grow unbounded, even though the tracking error norm remains finite at all times. This phenomenon is known as the "parameter drift." It is caused by the disturbance term $\xi(t)$. This argument shows that the MRAC laws (11.15) are not robust to bounded disturbances, no matter how small the latter are.

11.2 MRAC Design Modifications for Robustness

In this section, we introduce three design modifications to enforce robustness of MRAC laws in the presence of unmatched disturbances, such as bounded process noise. These modifications are (1) the dead zone, (2) the e-modification, and (3) the σ-modification.

11.2.1 The Dead-Zone Modification

In order to enforce robustness, we consider adaptive laws with the dead-zone modification:

$$\dot{\hat{\Theta}} = \begin{cases} \Gamma_\Theta \Phi(x)\ e^T P B, & \text{if } \|e\| > e_0 \\ 0_{N\times m}, & \text{if } \|e\| \le e_0 \end{cases} \quad (11.18)$$

Proposed by B.B. Peterson and K.S Narendra in [2], the dead-zone modification stops the adaptation process when the norm of the tracking error becomes smaller than the prescribed value e_0. This assures uniform ultimate boundedness (UUB) of $\Delta\Theta$ (in addition to UUB of e). We are going to formally prove this claim.

Suppose that $\|e\| > e_0$, then the adaptive law is defined by (11.15), and it results in the upper bound (11.16). Consequently, $e(t)$ enters Ω_0 in finite time T and will reside within the set for all $t \geq T$. From that time forward, the adaptive parameter dynamics are frozen, that is, $\hat{\Theta}(t+T) = 0_{N\times m}$. This proves UUB of the error dynamics (11.9), and it also proves boundedness (but not necessarily UUB) of the adaptive parameter estimation errors, $\|\Delta\Theta(t)\| < \infty$, uniformly in time.

The tracking error bound e_0 in (11.17) depends on the eigenvalue ratio $\dfrac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}$. It is not too difficult to show (see [3], pp. 92–93) that the minimum of this ratio is achieved for $Q = I_{n\times n}$. Thus, the computable tracking error upper bound is proportional to $2\,\lambda_{\max}(P)\,\xi_{\max}$, where $P = P^T > 0$ is the unique solution of the Lyapunov equation $P A_{ref} + A_{ref}^T P = -I_{n\times n}$. However, even when the disturbance vanishes, with the dead-zone modification being active, asymptotic stability of the tracking error cannot be recovered.

The dead-zone modification is not Lipschitz, and as such, it may cause chattering (high-frequency oscillations) and other undesirable effects, especially when the tracking error is at or near the dead-zone boundary. A smooth version of the dead-zone modification was introduced by Slotine and Coetsee in [4]. Motivated by this idea, we choose a constant $0 < \delta < 1$ and consider a Lipschitz-continuous modulation function in the form

$$\mu(\|e\|) = \max\left(0,\ \min\left(1,\ \frac{\|e\| - \delta\, e_0}{(1-\delta)\, e_0}\right)\right) \tag{11.19}$$

A sketch of this function is shown in Fig. 11.1.

Adaptive laws with the continuous dead-zone modification are defined as

$$\dot{\hat{\Theta}} = \Gamma_\Theta\, \Phi(x)\ \mu(\|e\|)\ e^T P B \tag{11.20}$$

With these laws of adaptation, one can use Lyapunov-based arguments to prove bounded tracking and UUB of all signals [2].

Example 11.1 MRAC with the Dead-Zone Modification for Aircraft Roll Dynamics

We shall illustrate an MRAC design with the dead-zone modification using the aircraft roll dynamics (a scalar system from Example 10.1)

$$\dot{p} = L_p\, p + L_{\delta_a}\, \delta_a + \xi(t)$$

subjected to a bounded environmental disturbance $\xi(t)$, which in this case may represent the rotational component of a gust. Also, in the model, p is the aircraft roll rate (rad/s), δ_a is the differential aileron deflection (rad), L_p is the aerodynamic roll damping (s^{-1}), and L_{δ_a} is the aileron effectiveness (s^{-1}).

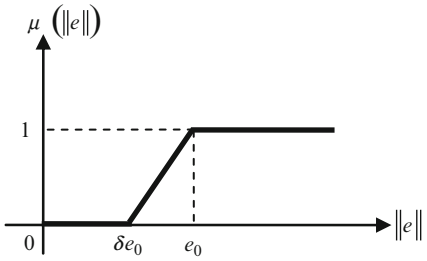


Fig. 11.1 The dead-zone modulation function

For a midsize airplane cruising at high altitude, typical values of the aerodynamic parameters are $L_p = -0.8,\ L_{\delta_a} = 1.6$. These are the two constant unknowns in the system. The goal is to design an MRAC state feedback–feedforward controller with the dead-zone modification and to enable bounded tracking of the reference model

$$\dot{p}_{ref} = A_{ref}\, p_{ref} + B_{ref}\, p_{cmd}$$

which is subsequently driven by a bounded roll rate command $p_{cmd}(t)$.

The roll dynamics can be easily rewritten in the form of (11.1):

$$\dot{p} = A_{ref}\, p + \underbrace{B}_{1}\ \underbrace{L_{\delta_a}}_{\Lambda > 0}\left(\underbrace{\delta_a}_{u} + \underbrace{\left(\frac{L_p - A_{ref}}{L_{\delta_a}}\right) p - \frac{B_{ref}}{L_{\delta_a}}\, p_{cmd}}_{\Theta^T\, \Phi(p,\, p_{cmd})}\right) + B_{ref}\, p_{cmd} + \xi(t)$$

where $\Theta^T = \dfrac{1}{L_{\delta_a}}\left(L_p - A_{ref} \quad -B_{ref}\right)$ is the vector of unknown constant parameters and $\Phi^T = \begin{pmatrix} p & p_{cmd} \end{pmatrix}$ is the known regressor vector, which depends on the system state p and the external command p_{cmd}. This model differs from (11.1) where the regressor is a state-dependent function. Even so, it is not difficult to repeat Lyapunov-based stability arguments and show that the same adaptive laws (11.18) apply, with the state- and command-dependent regressor vector $\Phi = \Phi(x,\ y_{cmd})$.

Therefore, according to (11.5) and (11.18), the MRAC roll rate tracking controller computes differential aileron deflections in the form

$$\delta_a = -\hat{k}_p\, p - \hat{k}_{p_{cmd}}\, p_{cmd}$$

where $\hat{\Theta}^T = \begin{pmatrix} \hat{k}_p & \hat{k}_{p_{cmd}} \end{pmatrix}$ are the adaptive gains, whose dynamics are specified by the adaptive laws shown below, with the discontinuous dead-zone modification:

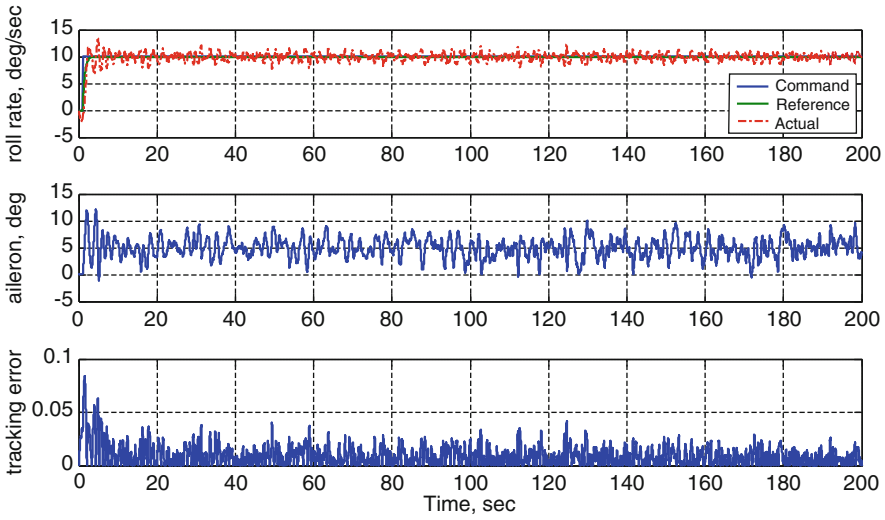


Fig. 11.2 Step-input roll rate tracking without the dead-zone modification in Example 11.1

$$\dot{\hat{k}}_p = \begin{cases} \gamma_p p\ (p - p_{ref}), & \text{if } |p - p_{ref}| > e_0 \\ 0, & \text{if } |p - p_{ref}| \le e_0 \end{cases}$$

$$\dot{\hat{k}}_{p_{cmd}} = \begin{cases} \gamma_{p_{cmd}} p_{cmd}\ (p - p_{ref}), & \text{if } |p - p_{ref}| > e_0 \\ 0, & \text{if } |p - p_{ref}| \le e_0 \end{cases}$$

For simulation, we have selected the following parameters:

$$A_{ref} = -B_{ref} = -2, \quad \gamma_p = \gamma_{p_{cmd}} = 100$$

The rotational gust component $\xi(t)$ was modeled as a random process noise, uniformly distributed on the interval $\frac{\pi}{180}[-10 \quad 10]$.

For a step-input roll rate command of 10°/s and without the dead-zone modification, that is, setting $e_0 = 0$, the system closed-loop tracking performance and the MRAC control effort (the aileron deflection) are adequate (Fig. 11.2).

As expected, the norm of the system tracking error is not zero and it is primarily driven by the process noise $\xi(t)$. However, the adaptive parameters exhibit the undesirable drift phenomenon (Fig. 11.3).

Rerunning the same case but with the dead-zone tolerance $e_0 = 0.0524$, we maintain good tracking performance (Fig. 11.4).

At the same time, the dead-zone modification prevents the adaptive parameters from drifting (Fig. 11.5).

As seen from Fig. 11.5, the adaptive parameters tend to their ideal unknown values. This can be attributed to an apparent level of persistency of excitation in the system dynamics, which is induced by the process noise. □

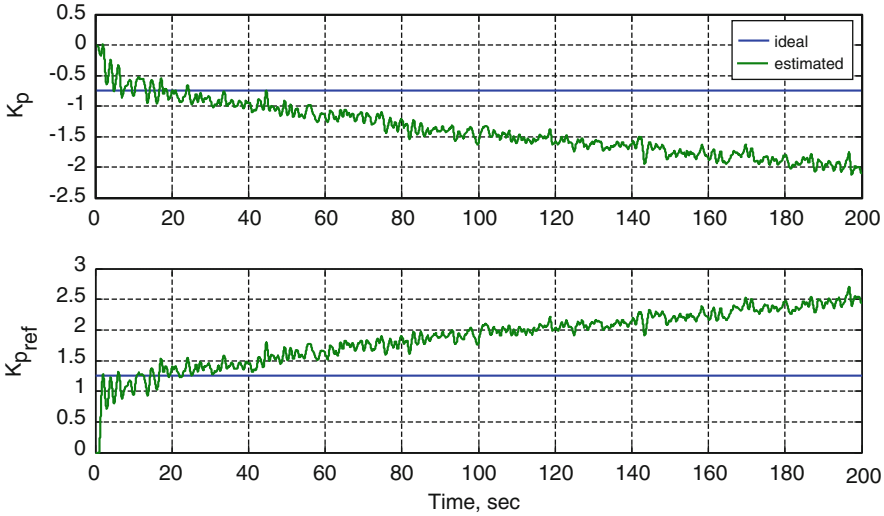


Fig. 11.3 Parameter drift without the dead-zone modification in Example 11.1

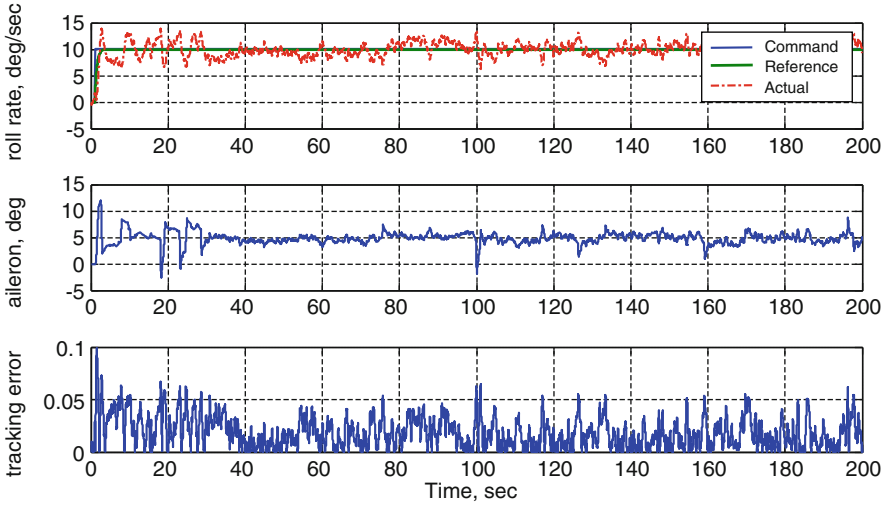


Fig. 11.4 Step-input roll rate tracking with the dead-zone modification in Example 11.1

11.2.2 The σ-Modification

Earlier, we have assumed prior knowledge of an upper bound $\xi_{\max}$ for the system disturbance $\xi(t)$. The σ-modification scheme, developed by Ioannou and Kokotovic [5, 8], does not require any prior information on the system disturbance upper bounds. The adaptive law with the σ-modification is

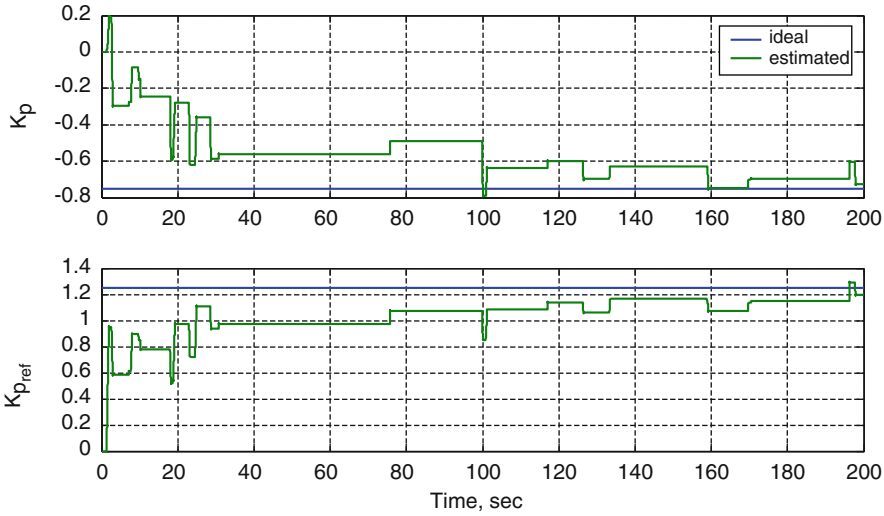


Fig. 11.5 Adaptive parameters with the dead-zone modification in Example 11.1

$$\dot{\hat{\Theta}} = \Gamma_\Theta \left(\Phi(x)\ e^T P B - \sigma \hat{\Theta} \right) \tag{11.21}$$

where σ is a strictly positive constant. In essence, this modification adds damping to the ideal adaptive law (11.15).

In order to prove UUB of all signals, we again consider the Lyapunov function candidate (11.10) and compute its time derivative along the trajectories of the tracking error dynamics (11.9):

$$\begin{aligned}
\dot{V}(e,\ \Delta\Theta) &= -e^T Q\, e + 2\,\mathrm{trace}\left(\Delta\Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi\, e^T P B \right\} \Lambda \right) + 2\, e^T P\, \xi(t) \\
&= -e^T Q\, e - 2\,\sigma\,\mathrm{trace}\left(\Delta\Theta^T \underbrace{\hat{\Theta}}_{\Theta + \Delta\Theta} \Lambda \right) + 2\, e^T P\, \xi(t) \\
&= -e^T Q\, e - 2\,\sigma\,\mathrm{trace}\left(\Delta\Theta^T\, \Delta\Theta\, \Lambda \right) - 2\,\sigma\,\mathrm{trace}\left(\Delta\Theta^T\, \Theta\, \Lambda \right) + 2\, e^T P\, \xi(t)
\end{aligned} \tag{11.22}$$

By definition,

$$\mathrm{trace}\left(\Delta\Theta^T\, \Delta\Theta\, \Lambda \right) = \sum_{i=1}^{N} \sum_{j=1}^{m} \Delta\Theta_{ij}^2\, \Lambda_{ii} \geq \|\Delta\Theta\|_F^2\, \Lambda_{\min} \tag{11.23}$$

where $\|\Delta\Theta\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \Delta\Theta_{ij}^2$ is the Frobenius norm of $\Delta\Theta$ and $\Lambda_{\min}$ is the minimum diagonal element of Λ. Moreover, using the Schwarz inequality gives

$$\left|\text{trace}\left(\Delta\Theta^T\,\Theta\,\Lambda\right)\right| \le \left\|\Delta\Theta^T\,\Theta\right\|_F \|\Lambda\|_F \le \|\Delta\Theta\|_F \|\Theta\|_F \|\Lambda\|_F \tag{11.24}$$

Substituting (11.23) and (11.24) into (11.22) results in

$$\begin{aligned}\dot{V}(e,\ \Delta\Theta) \le &-\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\,\lambda_{\max}(P)\,\xi_{\max}\\ &- 2\,\sigma\,\|\Delta\Theta\|_F^2\,\Lambda_{\min} + 2\,\sigma\,\|\Delta\Theta\|_F\,\|\Theta\|_F\,\|\Lambda\|_F\end{aligned} \tag{11.25}$$

Using $2\,a\,b \le a^2 + b^2$ for any a and b, we write

$$\begin{aligned}\dot{V}(e,\Delta\Theta) \le &-\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\,\lambda_{\max}(P)\,\xi_{\max}\\ &-2\sigma\|\Delta\Theta\|_F^2\Lambda_{\min} + \sigma\left(\|\Delta\Theta\|_F^2 + \|\Theta\|_F^2\right)\|\Lambda\|_F\\ = &-\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\,\lambda_{\max}(P)\,\xi_{\max} - \sigma\|\Delta\Theta\|_F^2\left(2\Lambda_{\min} + \|\Lambda\|_F\right) + \sigma\|\Theta\|_F^2\|\Lambda\|_F\end{aligned} \tag{11.26}$$

Hence, $\dot{V}(e,\ \Delta\Theta)<0$ if

$$\|e\|^2 - 2\,\|e\|\,\underbrace{\left(\frac{\lambda_{\max}(P)\,\xi_{\max}}{\lambda_{\min}(Q)}\right)}_{c_1} - \underbrace{\left(\frac{\sigma\,\|\Theta\|_F^2\,\|\Lambda\|_F}{\lambda_{\min}(Q)}\right)}_{c_2} > 0 \tag{11.27}$$

or, equivalently, when

$$\begin{cases}\|\Delta\Theta\|_F^2 > \dfrac{\|\Theta\|_F^2\,\|\Lambda\|_F}{\left(2\,\Lambda_{\min} + \|\Lambda\|_F\right)} = c_3\\ \|e\| > 2\left(\dfrac{\lambda_{\max}(P)\,\xi_{\max}}{\lambda_{\min}(Q)}\right) = 2\,c_1.\end{cases} \tag{11.28}$$

In other words, $\dot{V}<0$ outside of the compact (closed and bounded) set $\Omega \subset \left(R^n \times R^{N\times m}\right)$ defined below:

$$\begin{aligned}\Omega &= \left\{(e,\ \Delta\Theta) : \left[\|e\| \le c_1 + \sqrt{c_1^2 + c_2}\right] \wedge \left[\|e\| < 2\,c_1\right] \wedge \left[\|\Delta\Theta\|_F^2 \le c_3\right]\right\}\\ &= \left\{(e,\ \Delta\Theta) : \left[\|e\| < 2\,c_1\right] \wedge \left[\|\Delta\Theta\|_F^2 \le c_3\right]\right\}\\ &= \left\{(e,\ \Delta\Theta) : \left[\|e\| < 2\,\frac{\lambda_{\max}(P)\,\xi_{\max}}{\lambda_{\min}(Q)}\right] \wedge \left[\|\Delta\Theta\|_F^2 \le \frac{\|\Theta\|_F^2\,\|\Lambda\|_F}{\left(2\,\Lambda_{\min} + \|\Lambda\|_F\right)}\right]\right\}\end{aligned} \tag{11.29}$$

This argument immediately proves UUB of all signals in the closed-loop dynamics. In particular, (11.29) proves UUB tracking of the external command

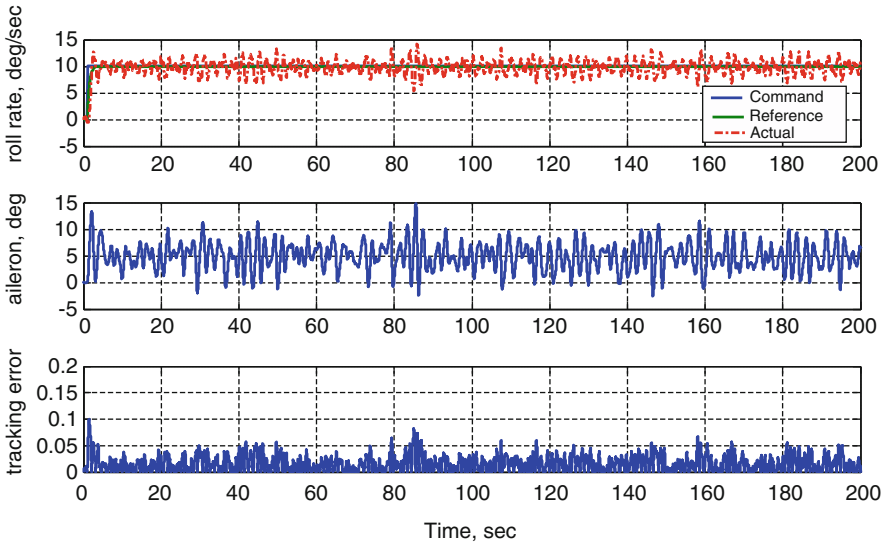


Fig. 11.6 Step-input roll rate tracking with the σ-modification in Example 11.2

$y_{cmd}(t)$ by the system output $y(t)$. Note that in this case, command tracking is achieved in the presence of parametric uncertainties $(\Lambda,\ \Theta)$ and nonparametric bounded time-varying disturbances $\xi(t)$. Next, we illustrate the σ-modification features and benefits for the scalar roll dynamics from Example 11.1.

Example 11.2 MRAC with the σ-Modification for Aircraft Roll Dynamics

Continuing with the roll dynamics model from Example 11.1, we utilize (11.21) and write the adaptive laws with the σ-modification:

$$\dot{\hat{k}}_p = \gamma_p \left(p\ \left(p - p_{ref}\right) - \sigma\ \hat{k}_p\right)$$
$$\dot{\hat{k}}_{p_{cmd}} = \gamma_{p_{cmd}}\ \left(p_{cmd}\ \left(p - p_{ref}\right) - \sigma\ \hat{k}_{p_{cmd}}\right)$$

We then select $\sigma = 0.1$ and simulate the same roll rate step-input response as in Example 11.1 but with the σ-modification turned on. This design also gives adequate roll rate command tracking performance (Fig. 11.6).

The data are comparable to the simulation results achieved using the dead-zone modification in Example 11.1 (see Fig. 11.4). In addition, the corresponding adaptive gains are bounded (Fig. 11.7), and potential drift-due-to-noise tendencies are completely prevented.

We make a note that in this case, the adaptive gains are oscillatory which may not be desirable. The oscillations are driven by the process noise, and the data reveal noise sensitivity of the adaptive law dynamics. □

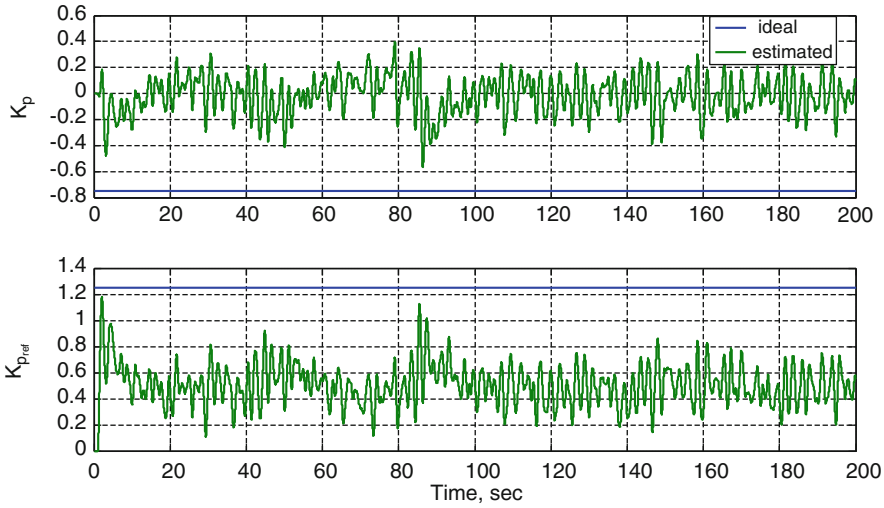


Fig. 11.7 Adaptive gains with the σ-modification in Example 11.2

11.3 The *e*-Modification

There are performance-related drawbacks to applying the σ-modification. When the tracking error becomes small, the adaptive law dynamics (11.21) can be approximately written as $\dot{\hat{\Theta}} \approx -\Gamma_\Theta \, \sigma \, \hat{\Theta}$. Hence, for small tracking errors, the adaptive parameters have a tendency to return to the origin, that is, they "unlearn" the gain values that caused the tracking error to become small in the first place. Furthermore, even if the disturbance $\xi(t)$ is removed from the system dynamics (11.1), and if the reference command y_{cmd} is persistently exciting [6], the parameter errors $\Delta\Theta(t)$ do not converge to the origin.

In order to overcome these undesirable effects, Narendra and Annaswamy introduced the *e*-modification [6]. Originally called the e_1 -modification, the method's main idea is to replace the constant damping gain σ in (11.21) with a term proportional to a linear combination of the system tracking errors, such as $\|e^T P B\|$. The rational for using an error-dependent damping is that it tends to 0, as the regulated output error diminishes. The adaptive laws with *e*-modification are

$$\dot{\hat{\Theta}} = \Gamma_\Theta \left(\Phi(x) \; e^T P B - \sigma \left\| e^T P B \right\| \hat{\Theta} \right) \tag{11.30}$$

As seen from (11.30), the *e*-modification adds a tracking error-dependent damping $\sigma \, \|e^T P B\|$ to the adaptive dynamics.

Using these laws, one can compute the time derivative of the Lyapunov function candidate (11.10), along the trajectories of the tracking error dynamics (11.9), and then repeat similar derivations that lead to (11.26). The only difference here is that instead of a constant parameter σ, we have an error-dependent damping term

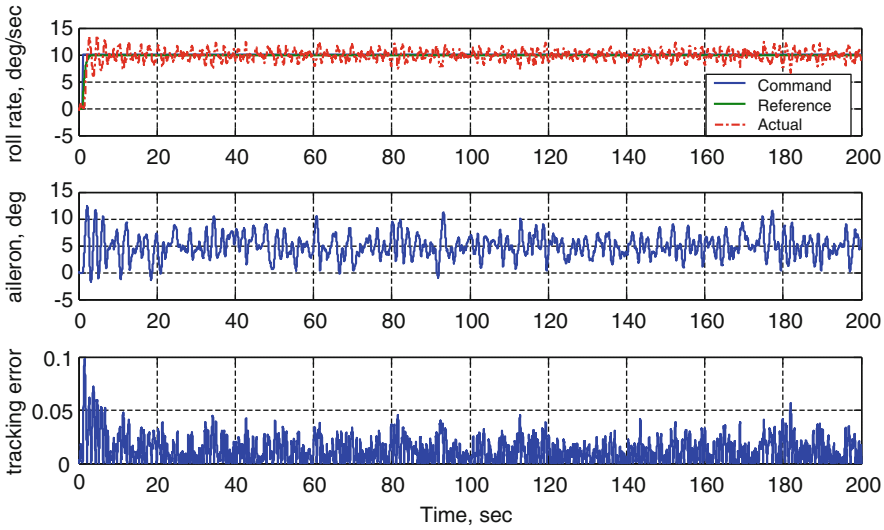


Fig. 11.8 Step-input roll rate tracking with the *e*-modification in Example 11.3

$\sigma \|e^T P B\|$. So, c_2 in (11.27) is no longer constant, but it remains nonnegative. This fact allows to arrive at the same compact set as in (11.29), outside of which $\dot{V}(e,\ \Delta\Theta)<0$. Once again, we can claim UUB of all trajectories. This completes the stability analysis for the *e*-modification with a guaranteed UUB-type output tracking performance.

Example 11.3 MRAC with the e-Modification for Aircraft Roll Dynamics

We now apply the *e*-modification design to the roll dynamics that was introduced in Example 11.1 and subsequently reused in Example 11.2. According to (11.30), the adaptive laws with the *e*-modification are

$$\dot{\hat{k}}_p = \gamma_p \left(p\ \left(p - p_{ref}\right) - \sigma \left|p - p_{ref}\right| \hat{k}_p\right)$$
$$\dot{\hat{k}}_{p_{cmd}} = \gamma_{p_{cmd}} \left(p_{cmd}\ \left(p - p_{ref}\right) - \sigma \left|p - p_{ref}\right| \hat{k}_{p_{cmd}}\right)$$

So now, the damping term $\sigma \left|p - p_{ref}\right|$ depends on the tracking error $e = p - p_{ref}$, and it will diminish if e becomes small.

Fig. 11.8 shows the system closed-loop tracking performance, with *e*-modification gain $\sigma = 1$.

Once again, we obtained adequate step-input command tracking in the presence of noise. The results are comparable to those shown in Figs. 11.4 and 11.6. The corresponding adaptive parameters are shown below (Fig. 11.9).

It is interesting to note that in this simulation scenario, the *e*-modification kept the adaptive parameters uniformly bounded, and in addition, it also forced them to approach their ideal values. However, such a tendency would not be possible

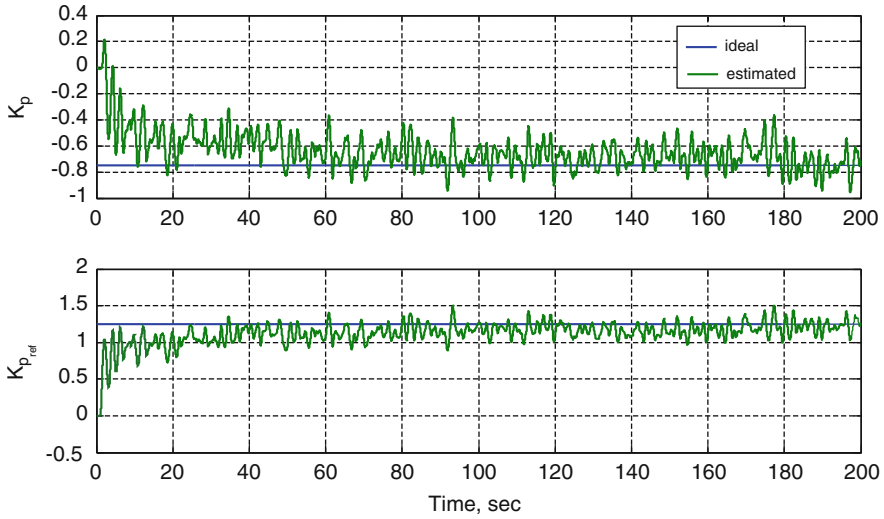


Fig. 11.9 Adaptive parameters with the e-modification in Example 11.3

without persistency of excitation induced by the process noise into the system dynamics. □

It is easy to see that for large tracking errors, the dead zone, the σ-modification, and the e-modification slow down (i.e., dampen) the adaptation. Often, such an effect is considered detrimental since it may contradict the control goal of reducing the tracking error as fast as possible.

11.4 The Projection Operator

In this section, we shall introduce a Lipschitz-continuous version of the Projection Operator [7, 8]. This concept is essential for enabling the adaptive laws (11.15) to achieve robustness with respect to parametric and nonparametric uncertainties that might exist in the system dynamics. We show that the Projection Operator tolerates fast adaptation, enforces uniform boundedness of the adaptive parameters, and maintains closed-loop stability of the corresponding error dynamics and of the original system. The selected version of the Projection Operator can be thought of as a direct extension of a projection-like modification that was originally proposed by Kreisselmeier and Narendra in [9].

To reiterate, our overall design goal is to continuously modify adaptive laws (11.15) in order to maintain negative semi-definiteness of the Lyapunov function time derivative in (11.14)

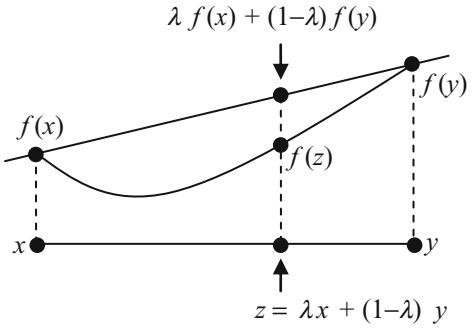


Fig. 11.10 Graph of a convex function

$$\text{trace}\left(\Delta\Theta^T\left\{\Gamma_\Theta^{-1}\dot{\hat{\Theta}} - \Phi\, e^T P B\right\}\Lambda\right) \leq 0 \tag{11.31}$$

and, at the same time, to keep the adaptive parameters $\hat{\Theta}(t)$ uniformly bounded in time. These two design objectives will be achieved through the introduction of the Projection Operator into the adaptive law dynamics.

We begin with basic definitions of convex sets and functions. These concepts will facilitate proper introduction of the Projection Operator.

Definition 11.1. *A subset $\Omega \subset R^n$ is convex if*

$$[\forall x,\, y \in \Omega \subset R^n] \Rightarrow [\lambda x + (1-\lambda)\, y = z \in \Omega], \quad \forall 0 \leq \lambda \leq 1 \tag{11.32}$$

Relation (11.32) states that if two points belong to a convex subset Ω, then all the points on the connecting line also belong to Ω.

Definition 11.2. *A function $f : R^n \to R$ is convex on R^n if*

$$f(\lambda x + (1-\lambda)\, y) \leq \lambda f(x) + (1-\lambda) f(y), \quad \forall 0 \leq \lambda \leq 1, \quad \forall x,\, y \in R^n \tag{11.33}$$

Inequality (11.33) is illustrated in Fig. 11.10. It shows that the graph of a convex function must be located below the straight line, which connects the two corresponding function values.

Lemma 11.1. *Let $f(x) : R^n \to R$ be convex. Then, for any constant $\delta > 0$, the subset $\Omega_\delta = \{\theta \in R^n \,|f(\theta) \leq \delta\}$ is convex.* ■

Proof of Lemma 11.1. Let $\theta_1,\ \theta_2 \in \Omega_\delta$. Then, $f(\theta_1) \leq \delta$ and $f(\theta_2) \leq \delta$. Since $f(x)$ is convex, then for any $0 \leq \lambda \leq 1$,

$$f\left(\underbrace{\lambda\,\theta_1 + (1-\lambda)\,\theta_2}_{\theta}\right) \leq \lambda\,\underbrace{f(\theta_1)}_{\leq\delta} + (1-\lambda)\,\underbrace{f(\theta_2)}_{\leq\delta} \leq \lambda\,\delta + (1-\lambda)\,\delta = \delta$$

Therefore, $f(\theta) \leq \delta$ and, consequently, $\theta \in \Omega_\delta$ which completes the proof. ■

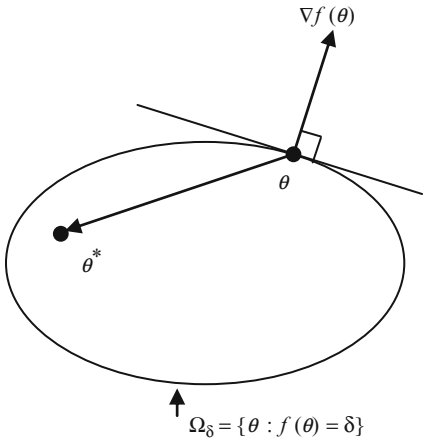


Fig. 11.11 Gradient vector on the boundary of a convex set

Lemma 11.2. *Let $f(x) : R^n \to R$ be a differentiable convex function. Choose a constant $\delta > 0$ and consider the subset*

$$\Omega_\delta = \{\theta \in R^n \,|\, f(\theta) \leq \delta\} \subset R^n$$

Let $\theta^* \in \Omega_\delta$ and assume that $f(\theta^*) < \delta$, that is, θ^* is an interior point (i.e., not on the boundary) of Ω_δ. Also, let $\theta \in \Omega_\delta$ and assume that $f(\theta) = \delta$, that is, θ lays on the boundary of Ω_δ. Then, the following inequality holds

$$(\theta^* - \theta)^T \nabla f(\theta) \leq 0 \tag{11.34}$$

where $\nabla f(\theta) = \left(\dfrac{\partial f(\theta)}{\partial \theta_1} \quad \ldots \quad \dfrac{\partial f(\theta)}{\partial \theta_n} \right)^T \in R^n$ is the gradient vector of f evaluated at θ. ▪

Relation (11.34) is illustrated in Fig. 11.11. It shows that the gradient vector of a function, evaluated at the boundary of a convex level set generated by this function, always points away from the set.

Proof of Lemma 11.2. Since $f(x)$ is convex, then

$$f(\lambda\,\theta^* + (1-\lambda)\,\theta) \leq \lambda f(\theta^*) + (1-\lambda) f(\theta)$$

Or, equivalently,

$$f(\theta + \lambda\,(\theta^* - \theta)) \leq f(\theta) + \lambda\,(f(\theta^*) - f(\theta))$$

Consequently, for any nonzero $0 < \lambda \leq 1$,

$$\frac{f(\theta+\lambda(\theta^*-\theta))-f(\theta)}{\lambda} \leq \underbrace{f(\theta^*)}_{<\delta} - \underbrace{f(\theta)}_{\delta} < \delta-\delta=0$$

Taking the limit as $\lambda \to 0$ yields relation (11.34) and completes the proof. ■

Suppose that a parameter vector θ belongs to a convex set Ω_0:

$$\Omega_0 = \{\theta \in R^n \,|\, f(\theta) \leq 0\} \tag{11.35}$$

Let us introduce another convex set:

$$\Omega_1 = \{\theta \in R^n \,|\, f(\theta) \leq 1\} \tag{11.36}$$

Then, it becomes obvious that $\Omega_0 \subseteq \Omega_1$.

We may now define the continuous Projection Operator

$$\begin{aligned}&\text{Proj}\,(\theta,\ y)\\ &= \begin{cases} y - \frac{\Gamma\nabla f(\theta)\,(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2_\Gamma}\, y f(\theta), & \text{if } [f(\theta)>0 \wedge y^T\nabla f(\theta)>0] \\ y & \text{, if not} \end{cases}\end{aligned} \tag{11.37}$$

where $\Gamma \in R^{n\times n}$ is any constant symmetric positive-definite matrix and $\|\nabla f\|^2_\Gamma = (\nabla f)^T\,\Gamma\,\nabla f$ is the weighed Euclidean squared norm of ∇f.

Let us graphically illustrate the Projection Operator in (11.37). To simplify the discussion, we set Γ to be the identity matrix. As seen from the definition (11.37), $\text{Proj}\,(\theta,\ y)$ does not alter the vector y if θ belongs to the convex set Ω_0 from (11.35). In the annulus set $\{0 \leq f(\theta) \leq 1\}$, the Projection Operator subtracts a vector normal to the boundary $\{f(\theta)=\lambda\}$ from y. As a result, we get a smooth transformation from the original vector field y for $\lambda=0$ to the tangent to the boundary vector for $\lambda=1$. The Projection Operator concept is shown in Fig. 11.12.

For an arbitrary positive-definite symmetric matrix Γ, a similar sketch can be drawn.

Next, we derive an important convex property of the Projection Operator.

Lemma 11.3. *For any symmetric positive-definite matrix* $\Gamma \in R^{n\times n}$,

$$(\theta-\theta^*)^T\left(\Gamma^{-1}\,\text{Proj}\,(\theta,\ \Gamma y) - y\right) \leq 0 \tag{11.38}$$

Proof of Lemma 11.3. Using (11.34) and (11.37) gives

$$\begin{aligned}&(\theta-\theta^*)^T\left(\Gamma^{-1}\,\text{Proj}\,(\theta,\ \Gamma y) - y\right)\\ &= \left\{ \begin{array}{ll} -\frac{\overbrace{(\theta-\theta^*)^T\nabla f}^{>0}}{\|\nabla f\|^2_\Gamma}\overbrace{\left[(\nabla f)^T\,\Gamma\, y\right]}^{>0}\overbrace{f}^{>0}, & \text{if } [f>0 \wedge y^T\Gamma\,\nabla f>0] \\ 0 & \text{, if not} \end{array} \right\} < 0\end{aligned} \tag{11.39}$$

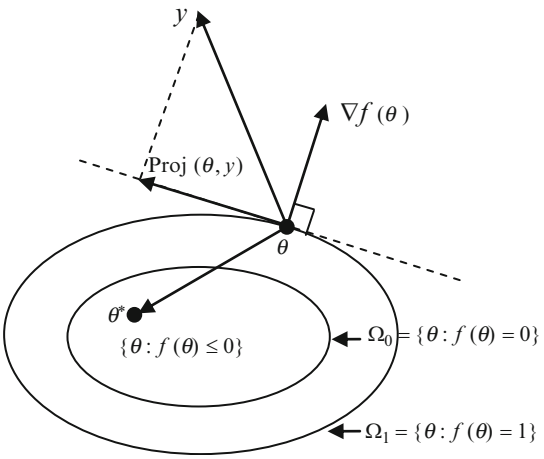


Fig. 11.12 The Projection Operator

and the proof is complete. ▪

We now state and prove yet another result of conceptual importance to the forthcoming development of adaptive controllers.

Lemma 11.4. *Let $f(\theta)$ be a convex continuously differentiable map from R^n to R. Using (11.37), consider the n-dimensional dynamics*

$$\dot{\theta} = \text{Proj}\,(\theta, y) \tag{11.40}$$

where $\theta \in R^n$ is the system state and $y \in R^n$ is a time-varying piecewise continuous vector. Then, starting from any initial condition $\theta(0) = \theta_0$ within the set

$$\Omega_0 = \{\theta \in R^n \,|f(\theta) \leq 0\} \tag{11.41}$$

the system trajectory $\theta(t)$ will remain in the set

$$\Omega_1 = \{\theta \in R^n \,|f(\theta) \leq 1\} \tag{11.42}$$

for all $t \geq 0$. ▪

Proof of Lemma 11.4. Existence and uniqueness of the system (11.40) solutions are provided by the fact that the Projection Operator is locally Lipchitz in θ, while the system external input $y(t)$ is piecewise continuous in time.

To prove the lemma, we need to show that the following relation holds

$$\underbrace{[f(\theta_0) \leq 0]}_{\theta_0 \in \Omega_0} \Rightarrow \underbrace{[f(\theta(t)) \leq 1]}_{\theta(t) \in \Omega_1}, \quad \forall t \geq 0 \tag{11.43}$$

Toward that end, we evaluate the time derivative of $f(\theta(t))$ along the trajectories of the system dynamics (11.40). Based on the definition (11.37), we obtain

$$\begin{aligned} \dot{f}(\theta) &= (\nabla f(\theta))^T \,\text{Proj}\,(\theta, y) \\ &= \begin{cases} (\nabla f(\theta))^T y(1 - f(\theta)), & \text{if } [f(\theta) > 0 \wedge y^T \nabla f(\theta) > 0] \\ (\nabla f(\theta))^T y, & \text{if not} \end{cases} \end{aligned} \tag{11.44}$$

Consequently,

$$\begin{aligned} &\dot{f}(\theta) > 0, \quad \text{if } \left[0 < f(\theta) < 1 \wedge y^T \nabla f(\theta) > 0\right] \\ &\dot{f}(\theta) = 0, \quad \text{if } \left[f(\theta) = 1 \wedge y^T \nabla f(\theta) > 0\right] \\ &\dot{f}(\theta) \leq 0, \quad \text{if } \left[f(\theta) \leq 0 \vee y^T \nabla f(\theta) \leq 0\right] \end{aligned} \tag{11.45}$$

The first and the second relations in (11.45) imply that if $f(\theta(0)) > 0$, then $f(\theta(t))$ monotonically increases in time for all $t \geq 0$, but it will never exceed 1. Also, the third condition in (11.45) states that if $f(\theta(0)) \leq 0$, then $f(\theta(t))$ is monotonically decreasing for all $t \geq 0$. Therefore, irrespective of initial values (as long as they are negative), $f(\theta(t)) \leq 1$ for all $t \geq 0$, which completes the proof of the lemma. ▪

The next example shows how to use the Projection Operator to construct actuator models with position and rate constraints.

Example 11.4 Actuator Dynamics with Position and Rate Constraints In control engineering applications, one often needs to account for mechanical, hydraulic, or electrical control actuation devices. Their dynamics are frequently modeled by a scalar system

$$\tau \dot{u} = u_{cmd} - u$$

where u_{cmd} is the actuator-commanded position, u is the actuator-achieved position, and τ is the actuator time constant. Since these devices have inherent position limits, the latter must be introduced into the model and analyzed appropriately.

In this example, we shall demonstrate how to create a dynamical model of an actuator with position constraints $|u| \leq u_{\max}$. Let ε be a constant such that $\varepsilon \in (0,\ 1)$. We introduce

$$\bar{u}_{\max} = \frac{u_{\max}}{\sqrt{1 + \varepsilon}}$$

and then embed the actuator constraints into the Projection Operator definition (11.37), by selecting a convex function in the form

$$f(u) = \frac{u^2 - \bar{u}_{\max}^2}{\varepsilon\ \bar{u}_{\max}^2} = \frac{(1+\varepsilon)\,u^2 - u_{\max}^2}{\varepsilon\ u_{\max}^2}$$

In this case, the two convex sets from (11.35) and (11.36) become

$$\Omega_0 = \{u \in R:\ f(u) \le 0\} = \left\{u \in R:\ |u| \le \frac{u_{\max}}{\sqrt{1+\varepsilon}}\right\}$$
$$\Omega_1 = \{u \in R:\ f(u) \le 1\} = \{u \in R:\ |u| \le u_{\max}\}$$

Using (11.37), we can now define the following projection-based first-order actuator model with position constraints:

$$\dot{u} = \text{Proj}\left(u,\ \frac{u_{cmd} - u}{\tau}\right)$$
$$= \begin{cases} \left(\frac{u_{cmd}-u}{\tau}\right)(1 - f(u)), & \text{if } [f(u) > 0\ \wedge (u_{cmd} - u)\,u > 0] \\ \left(\frac{u_{cmd}-u}{\tau}\right), & \text{if not} \end{cases}$$

According to Theorem 11.1, starting anywhere within the "conservative" position limits $\left(\pm\frac{u_{\max}}{\sqrt{1+\varepsilon}}\right)$, the actuator-achieved position $u(t)$ will never exceed the original limits $(\pm\ u_{\max})$, even if it is commanded to do so. In other words, there is no need to limit the commanded position. No matter what the actuator command is, the achieved position will remain within the prespecified limits.

One can make further modifications to the derived actuator model and enforce rate limit constraints $(\pm\ \dot{u}_{\max})$, in addition to position limits. For example, the following model

$$\dot{u} = \dot{u}_{\max}\,\text{sat}\left(\frac{1}{\dot{u}_{\max}}\ \text{Proj}\left(u,\ \frac{u_{cmd} - u}{\tau}\right)\right)$$

uses the saturation function $y = \text{sat}(x) = \max(-1,\ \min(x,\ 1))$, along with the Projection Operator. It is easy to see that these two modifications will keep both the actuator position and its rate contained within their desired limits.

Frequently in practice, actuator requirements are specified in terms of their natural frequencies and damping ratios. This leads to consideration of a second-order actuator model in the form

$$\ddot{u} + 2\,\xi\,\omega\,\dot{u} + \omega^2\,u = \omega^2\,u_{cmd}$$

where $(\omega,\ \xi)$ denote the actuator natural frequency and its damping ratio, correspondingly. Rewriting the model in state space gives

$$\underbrace{\begin{pmatrix}\dot{u}_1\\ \dot{u}_2\end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix}0 & 1\\ -\omega^2 & -2\,\xi\,\omega\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}u_1\\ u_2\end{pmatrix}}_{x} + \underbrace{\begin{pmatrix}0\\ \omega^2\end{pmatrix}}_{B} u_{cmd}$$

where $u_1 = u$ is the actuator-achieved position and $u_2 = \dot{u}$ is the respective rate.

In order to impose position and rate constraints $\pm(u_{\max},\ \dot{u}_{\max})$, we shall again use the Projection Operator (11.37) and modify the actuator dynamics as follows:

$$\dot{x} = \text{Proj}(x,\ A\,x + B\,u_{cmd})$$

For this model, a convex function $f(x) = f(u,\ \dot{u})$, which defines the Projection Operator domain, can be selected as

$$f(u,\ \dot{u}) = \frac{(1+\varepsilon)\left(\dfrac{u^2}{u_{\max}^2} + \dfrac{\dot{u}^2}{\dot{u}_{\max}^2}\right) - 1}{\varepsilon}$$

This leads to the following two convex sets:

$$\Omega_0 = \{f(u,\ \dot{u}) \le 0\} = \left\{|u| \le \frac{u_{\max}}{\sqrt{1+\varepsilon}} \wedge |\dot{u}| \le \frac{\dot{u}_{\max}}{\sqrt{1+\varepsilon}}\right\}$$
$$\Omega_1 = \{f(u,\ \dot{u}) \le 1\} = \{|u| \le u_{\max} \wedge |\dot{u}| \le \dot{u}_{\max}\}$$

From Lemma 11.4, we can assert that starting with any initial conditions from Ω_0, which satisfy the actuator position and rate bounds, the actuator model will produce trajectories evolving within the prescribed bounds in Ω_1. □

In the next section, we shall employ the Projection Operator (11.37) to construct provably stable adaptive laws in the form

$$\dot{\theta} = \text{Proj}(\theta,\ \Gamma\,y) = \Gamma\begin{cases} y - \dfrac{\nabla f\,(\nabla f)^T}{\|\nabla f\|_\Gamma^2}\,\Gamma\,y f, & \text{if } \left[f > 0 \wedge y^T\,\Gamma\,\nabla f > 0\right]\\ y, & \text{if not}\end{cases} \tag{11.46}$$

where θ denotes the estimated parameter vector, whose dynamics are driven by the time-varying external vector $y = y(t)$.

Based on (11.46), we can introduce a matrix version of the Projection Operator, when both Y and Θ are matrices of the same dimensions:

$$Y = (\,\vec{y}_1 \quad \ldots \quad \vec{y}_N\,) \in R^{n\times N}, \quad \Theta = (\,\vec{\theta}_1 \quad \ldots \quad \vec{\theta}_N\,) \in R^{n\times N} \tag{11.47}$$

In this case, the Projection Operator is defined column-wise:

$$\text{Proj}\left(\Theta,\ \Gamma Y\right) = \left(\text{Proj}\left(\vec{\theta}_1,\ \Gamma \vec{y}_1\right) \quad \ldots \quad \text{Proj}\left(\vec{\theta}_N,\ \Gamma \vec{y}_N\right)\right) \tag{11.48}$$

We can also generalize the convex inequality (11.38):

$$\text{tr}\left(\Delta\Theta^T\left(\Gamma^{-1}\text{Proj}\left(\hat{\Theta},\,\Gamma Y\right) - Y\right)\right) = \sum_{j=1}^{m} \underbrace{\left(\hat{\Theta}-\Theta\right)_j^T\left(\Gamma^{-1}\text{Proj}\left(\hat{\Theta},\,\Gamma Y_j\right) - Y_j\right)}_{\leq 0} \leq 0 \tag{11.49}$$

In addition, one can show that for all matrices $\Theta(0)$, whose columns belong to the set Ω_0 from (11.41), the corresponding trajectory $\hat{\Theta}(t)$ of the matrix differential equation

$$\dot{\Theta} = \text{Proj}\left(\Theta,\ \Gamma Y\right) \tag{11.50}$$

will have its columns evolving within the set Ω_1 from (11.42), for all $t \geq 0$. This statement directly follows from Lemma 11.4.

11.5 Projection-Based MRAC Design

In Sect. 11.1, we have designed robust MRAC systems for MIMO dynamics (11.1), with matched parametric uncertainties and a bounded process noise. These designs were carried out to force time derivatives of the selected Lyapunov function (11.10), computed along the trajectories of the error dynamics (11.9), to become negative semidefinite outside of a compact set. For example, in (11.14), we had

$$\dot{V}(e,\,\Delta\Theta) = -e^T Q e + 2\,\text{trace}\left(\Delta\Theta^T\left\{\Gamma_\Theta^{-1}\dot{\hat{\Theta}} - \Phi e^T P B\right\}\Lambda\right) + 2\,e^T P\,\xi(t), \tag{11.51}$$

and the design task was to choose $\dot{\hat{\Theta}}$ such that the trace term in (11.51) became nonpositive, while the adaptive parameters $\hat{\Theta}(t)$ remained uniformly bounded functions of time.

In what follows, we shall investigate how to force the trace term to be semi-negative via the matrix version of the Projection Operator (11.50), with its convex property (11.49), while enforcing uniform boundedness of the corresponding solutions $\hat{\Theta}(t)$. Since

$$\begin{aligned}&\text{tr}\left(\underbrace{\Delta\Theta^T}_{\left(\hat{\Theta}-\Theta\right)^T}\left[\Gamma_\Theta^{-1}\quad\underbrace{\dot{\hat{\Theta}}}_{Proj\left(\hat{\Theta},\,\Gamma_\Theta Y\right)}\quad-\underbrace{\Phi\,e^T\,P\,B}_{Y}\right]\Lambda\right)\\&=\sum_{j=1}^{m}\underbrace{\left(\hat{\Theta}-\Theta\right)_j^T\left(\Gamma_\Theta^{-1}\,\text{Proj}\left(\hat{\Theta},\;\Gamma_\Theta\,Y_j\right)-Y_j\right)}_{\leq 0}\;\underbrace{\lambda_j}_{\geq 0}\leq 0\end{aligned}\tag{11.52}$$

then we can define the following projection-based adaptive law

$$\dot{\hat{\Theta}}=\text{Proj}\Big(\hat{\Theta},\;\Gamma_\Theta\,\Phi\,e^T\,P\,B\Big)\tag{11.53}$$

to guarantee uniform boundedness of the adaptive gains column-wise (Lemma 11.4). Essentially, the Projection Operator ensures that the columns $\hat{\Theta}_j$ of the adaptive time-dependent parameter matrix $\hat{\Theta}(t)$ do not exceed their prespecified bounds $\Theta_j^{\max}$. At the same time and because of (11.52), it is easy to see that the operator contributes to the negative semi-definiteness of the Lyapunov function (11.51). Indeed,

$$\begin{aligned}&\dot{V}(e,\;\Delta\Theta)\leq-e^T\,Q\,e+2\,e^T\,P\,\xi(t)\leq-\lambda_{\min}(Q)\,\|e\|^2+2\,\|e\|\,\lambda_{\max}(P)\,\xi_{\max}\\&=-\lambda_{\min}(Q)\,\|e\|\left(\|e\|-2\,\frac{\lambda_{\max}(P)\,\xi_{\max}}{\lambda_{\min}(Q)}\right)\end{aligned}\tag{11.54}$$

and, consequently, $\dot{V}(e,\;\Delta\Theta)<0$ outside of the compact set

$$\Omega=\left\{(e,\;\Delta\Theta)\in R^n\times R^{N\times m}:\quad\|e\|\leq 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\xi_{\max}\wedge\|\Delta\Theta\|_F\leq\Delta\Theta_{\max}\right\}\tag{11.55}$$

where

$$\Delta\Theta_{\max}=2\,\underbrace{\left(\Theta_1^{\max}\quad\ldots\quad\Theta_m^{\max}\right)}_{\Theta_{\max}}=2\,\Theta_{\max}\tag{11.56}$$

and $\Theta_j^{\max}$ is the maximum allowable bound for the j^{th} column $\hat{\Theta}_j(t)$. This formal argument proves the UUB property of all signals in the corresponding closed-loop system. In particular, we have proven that the system regulated output $y(t)$ can track any external bounded command $y_{cmd}(t)$ with bounded errors.

Next, we show how to construct the convex vector function $f=\left(f_1\quad\ldots\quad f_m\right)^T$ and the related m-convex sets $\left\{\Omega_\delta^j\right\}_{j=1,\ldots,m}$. These are the sets that define the

Projection Operator domains for each column of adaptive parameters $\hat{\Theta}_j(t)$. Both the function and the convex set definitions will be constructed based on the desired column-wise upper bounds $\left\|\hat{\Theta}_j(t)\right\| \leq \Theta_j^{\max}$.

For the j^{th}column $\hat{\Theta}_j$ of the adaptive parameter matrix $\hat{\Theta} \in R^{N\times m}$, we introduce the projection tolerance $\varepsilon_j^{\Theta} > 0$ and choose a convex function in the form

$$f_j = f\left(\hat{\Theta}_j\right) = \frac{\left(1+\varepsilon_j^{\Theta}\right)\left\|\hat{\Theta}_j\right\|^2 - \left(\Theta_j^{\max}\right)^2}{\varepsilon_j^{\Theta}\left(\Theta_j^{\max}\right)^2} \tag{11.57}$$

The idea here is very similar to the one in Example 11.4. The two convex sets are defined for each $j = 1, \ldots, m$:

$$\begin{aligned}
\Omega_0^j &= \left\{\hat{\Theta}_j \in R^{N\times 1}:\ f\left(\hat{\Theta}_j\right) \leq 0\right\} = \left\{\hat{\Theta}_j \in R^{N\times 1}:\ \left\|\hat{\Theta}_j\right\| \leq \frac{\Theta_j^{\max}}{\sqrt{1+\varepsilon_j^{\Theta}}}\right\} \\
\Omega_1^j &= \left\{\hat{\Theta}_j \in R^{N\times 1}:\ f\left(\hat{\Theta}_j\right) \leq 1\right\} = \left\{\hat{\Theta}_j \in R^{N\times 1}:\ \left\|\hat{\Theta}_j\right\| \leq \Theta_j^{\max}\right\}
\end{aligned} \tag{11.58}$$

The gradient of the j^{th} convex function (11.57) can be easily computed as

$$\nabla f_j = \frac{\left(1+\varepsilon_j^{\Theta}\right)}{\varepsilon_j^{\Theta}\left(\Theta_j^{\max}\right)^2}\nabla\left[\left\|\hat{\Theta}_j\right\|^2\right] = \frac{2\left(1+\varepsilon_j^{\Theta}\right)}{\varepsilon_j^{\Theta}\,\Theta_j^{\max}}\,\hat{\Theta}_j \tag{11.59}$$

Via (11.53), the adaptive law for $\hat{\Theta}_j$ becomes

$$\dot{\hat{\Theta}}_j = \Gamma_\Theta \begin{cases} \left(\Phi\, e^T P B\right)_j - \dfrac{\nabla f_j \nabla f_j^T}{\|\nabla f\|^2_{\Gamma_\Theta}}\,\Gamma_\theta\left(\Phi\, e^T P B\right)_j f_j \\ \qquad\qquad , \quad \text{if} \quad \left[f_j > 0 \wedge \left(\Phi\, e^T P B\right)_j^T \Gamma_\theta \nabla f_j < 0\right] \\ \left(\Phi\, e^T P B\right)_j, \quad \text{if not} \end{cases} \tag{11.60}$$

By construction, the adaptation process in (11.60) ensures uniform boundedness of the adaptive time-dependent parameter matrix $\hat{\Theta}(t)$ forward in time, that is,

$$\left\{\left\|\hat{\Theta}_j(0)\right\| \leq \frac{\Theta_j^{\max}}{\sqrt{\left(1+\varepsilon_j^{\Theta}\right)}}\right\} \Rightarrow \left\{\left\|\hat{\Theta}_j(t)\right\| \leq \Theta_j^{\max}, \quad \forall t \geq 0, \quad 1 \leq j \leq m\right\} \tag{11.61}$$

Table 11.1 MRAC design with robustness modifications

Open-loop plant	$\dot{x} = A_{ref}\, x + B\, \Lambda \left(u + \Theta^T\, \Phi(x)\right) + B_{ref}\, y_{cmd} + \xi(t)$ $y = C_{ref}\, x$
Reference model	$\dot{x}_{ref} = A_{ref}\, x_{ref} + B_{ref}\, y_{cmd}, \quad y_{ref} = C_{ref}\, x_{ref}$
State tracking error	$e = x - x_{ref}$
Lyapunov equation	$P\, A_{ref} + A_{ref}^T\, P = -Q$
Total control input	$u = -\hat{\Theta}^T\, \Phi(x)$
MRAC with dead zone	$\dot{\hat{\Theta}} = \Gamma_\Theta\, \Phi(x)\; \mu(\|e\|)\; e^T\, P\, B$
MRAC with σ-mod	$\dot{\hat{\Theta}} = \Gamma_\Theta \left(\Phi(x)\; e^T\, P\, B - \sigma\, \hat{\Theta}\right)$
MRAC with e-mod	$\dot{\hat{\Theta}} = \Gamma_\Theta \left(\Phi(x)\; e^T\, P\, B - \sigma\, \|e^T\, P\, B\|\, \hat{\Theta}\right)$
MRAC with Projection Operator	$\dot{\hat{\Theta}} = \mathrm{Proj}\left(\hat{\Theta},\; \Gamma_\Theta\, \Phi\, e^T\, P\, B\right)$

Consequently, the adaptive parameter errors $\Delta\Theta(t)$ and the state tracking error $e(t)$ enter a compact set that contains the set Ω from (11.55) in finite time. The MIMO-bounded command tracking problem is solved.

Table 11.1 gives a summary of the four robustness modifications that were introduced in this chapter.

Table 11.2 presents an overview of the continuous Projection Operator, which acts on a pair of n-dimensional vectors θ and y.

The next example illustrates key design points in application of the projection-based MRAC to lateral-directional dynamics of an aircraft.

Example 11.5 Aircraft Lateral-Directional Dynamics and Control

Lateral-directional motion of a conventional aircraft is controlled by vertical tail panels (rudders) and wing-mounted surfaces (ailerons). Figure 11.13 shows a sketch.

The rudder (δ_r) is the primary control device for turning the aircraft, thus regulating the vehicle yaw rate r and the sideslip angle β. Moving ailerons differentially (i.e., left aileron trailing edge down and right aileron trailing edge up, δ_a) will force the aircraft to roll (right wing down), changing (increasing) its roll rate p, and thus the bank angle φ, with some induced coupling into the yaw and sideslip dynamics.

For small angles, the aircraft lateral-directional dynamics can be approximated by a linear time-invariant system in the form

Table 11.2 The Projection Operator design summary

Max parameter bounds	$\|\theta\| \le \theta^{\max}$
Convex function	$f\left(\hat{\theta}\right) = \dfrac{(1+\varepsilon)\|\theta\|^2 - (\theta^{\max})^2}{\varepsilon\,(\theta^{\max})^2}$
Two convex sets	$\Omega_0 = \{\theta: \quad f(\theta) \le 0\} = \left\{\theta: \quad \|\theta\| \le \dfrac{\theta^{\max}}{\sqrt{1+\varepsilon}}\right\}$ $\Omega_1 = \{\theta: \quad f(\theta) \le 1\} = \{\theta: \quad \|\theta\| \le \theta^{\max}\}$
Projection Operator	$\text{Proj}\,(\theta,\ y) = \begin{cases} y - \dfrac{\Gamma\,\nabla f\,(\nabla f)^T}{(\nabla f)^T \Gamma\,\nabla f}\,yf, \\ \qquad \text{if } [f > 0 \wedge (y^T\,\nabla f) > 0] \\ y, \quad \text{if not} \end{cases}$
Convex inequality for proof of stability	$(\theta - \theta^*)^T \left(\Gamma^{-1}\,\text{Proj}\,(\theta,\ \Gamma\,y) - y\right) \le 0,$ $\forall \theta^* \in \Omega_0,\ \theta \in \Omega_1,\ y \in R^n$
Uniform boundedness of parameters	$\dot{\theta} = \text{Proj}(\theta,\ \Gamma\,y)$ $[\theta(0) \in \Omega_0] \Rightarrow [\,\theta(t) \in \Omega_1,\ \forall t \ge 0]$

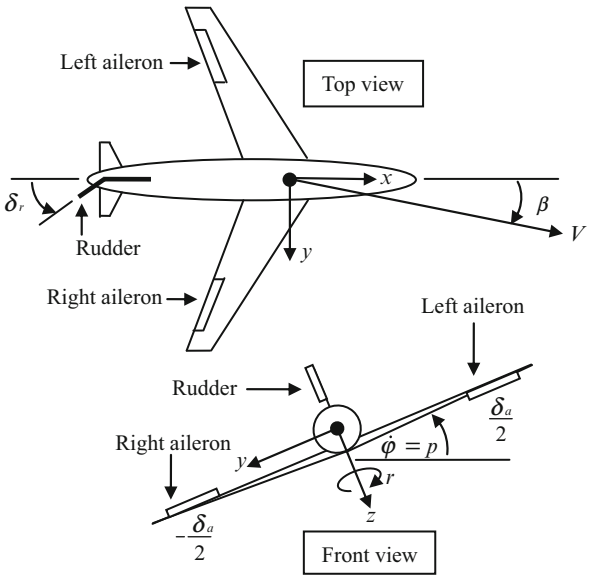


Fig. 11.13 Top and front views of a conventional aircraft in Example 11.5

Table 11.3 Nominal open-loop vehicle eigenvalues in Example 11.5

Eigenvalue	Damping	Frequency (rad/s)
$-0.0464 \pm 1.88\,j$	0.0247	1.88
0.00135	-1	0.00135
-1.78	1	1.78

$$\underbrace{\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{g}{V} & \frac{Y_\beta}{V} & \frac{Y_p}{V} & \frac{Y_r}{V} - 1 \\ 0 & L_\beta & L_p & Y_r \\ 0 & N_\beta & N_p & N_r \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \varphi \\ \beta \\ p \\ r \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_a}}{V} & \frac{Y_{\delta_r}}{V} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{pmatrix}}_{B_p} \underbrace{\begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}}_{u}$$

where $g = 32.174$ is the acceleration due to gravity (ft/s^2), V is the trimmed airspeed (positive constant, ft/s), and the system matrices $\left(A_p,\ B_p\right)$ are comprised of the vehicle aerodynamic stability and control derivatives.

For a small passenger aircraft in a cruise configuration, typical values of these parameters are [11, p. 357]

$$A_p = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.0487 & -0.0829 & 0 & -1 \\ 0 & -4.546 & -1.699 & 0.1717 \\ 0 & 3.382 & -0.0654 & -0.0893 \end{pmatrix}, \quad B_p = \begin{pmatrix} 0 & 0 \\ 0 & 0.0116 \\ 27.276 & 0.5758 \\ 0.3952 & -1.362 \end{pmatrix}$$

where the units for all angles and angular rates are expressed in rad and rad/s, respectively. Also, negligible coefficients in the β-dynamics are zeroed out.

A typical (for lateral-directional dynamics) regulated output would consist of the vehicle bank and sideslip angles

$$y = \begin{pmatrix} \varphi \\ \beta \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{C_p} x_p$$

while the available control inputs are represented by the differential aileron and the rudder deflections, both expressed in radians:

$$u = \begin{pmatrix} \delta_a & \delta_r \end{pmatrix}^T$$

The control task is to design u to enable independent and simultaneous tracking of bounded time-varying bank and sideslip commands that are stored in the vector $y_{cmd} = \begin{pmatrix} \varphi_{cmd} & \beta_{cmd} \end{pmatrix}^T$.

The nominal open-loop vehicle dynamics are unstable with the corresponding eigenvalues shown below (Table 11.3).

Table 11.4 Nominal closed-loop vehicle eigenvalues in Example 11.5

Eigenvalue	Damping	Frequency (rad/s)
$-1.34 \pm 1.29\,j$	0.72	1.86
$-1.25 \pm 1.17\,j$	0.73	1.71
−1.33	1	1.33
−8.84	1	8.84

In order to stabilize these dynamics and regulate the selected two outputs, we are going to design a baseline LQR tracking controller with Proportional + Integral (PI) action. Toward that end, we augment the system with two integrated tracking errors and obtain the baseline/nominal extended open-loop system

$$\underbrace{\begin{pmatrix} \dot{e}_{\varphi I} \\ \dot{e}_{\beta I} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{2\times 2} & C_p \\ 0_{4\times 2} & A_p \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} e_{\varphi I} \\ e_{\beta I} \\ x_p \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0_{2\times 2} \\ B_p \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}}_{u} + \underbrace{\begin{pmatrix} -I_{2\times 2} \\ 0_{4\times 2} \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} \varphi_{cmd} \\ \beta_{cmd} \end{pmatrix}}_{y_{cmd}}$$

$$y = \underbrace{\begin{pmatrix} 0_{2\times 2} & C_p \end{pmatrix}}_{C} x = \begin{pmatrix} \varphi & \beta \end{pmatrix}^T$$

where

$$\dot{e}_{\varphi I} = \varphi - \varphi_{cmd}, \quad \dot{e}_{\beta I} = \beta - \beta_{cmd}$$

are the dynamics of the two integrated tracking error signals. After several design iterations, we have selected diagonal LQR weights:

$$Q = \text{diag}\begin{pmatrix} 1 & 10 & 0 & 0 & 0.1 & 5 \end{pmatrix}, \quad R = I_{2\times 2}$$

The first two diagonal elements of Q give adequate natural frequencies, while the last two yield desired damping ratios in both regulated output channels (Table 11.4).

The resulting baseline LQR PI state feedback solution is

$$u_{bl} = -\underbrace{\begin{pmatrix} 0.9987 & 0.1627 & 0.9184 & 0.0896 & 0.3529 & -0.0166 \\ -0.0514 & 3.1581 & 0.0755 & 2.2907 & 0.0487 & -2.7885 \end{pmatrix}}_{K_{xLQR}^T} x = -K_{xLQR}^T x$$

and the closed-loop simulation results are shown in Fig. 11.14, where we have tested the baseline LQR PI controller performance in tracking a series of step-input bank and sideslip commands, simultaneously.

There are three signals per plot that are shown in the figure above: (1) the command response, (2) the reference response, and (3) the actual system response. As in all our previous examples, the reference data represent the closed-loop

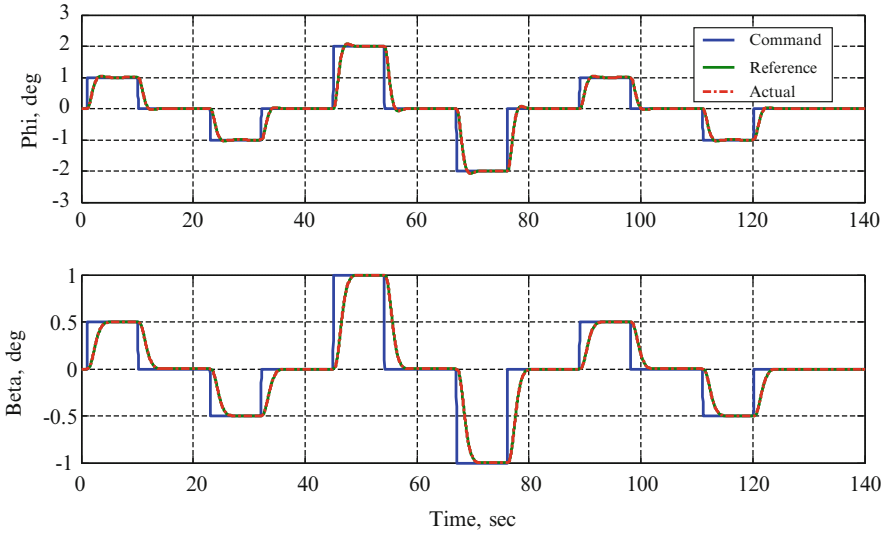


Fig. 11.14 Command tracking with baseline LQR PI Controller in Example 11.5

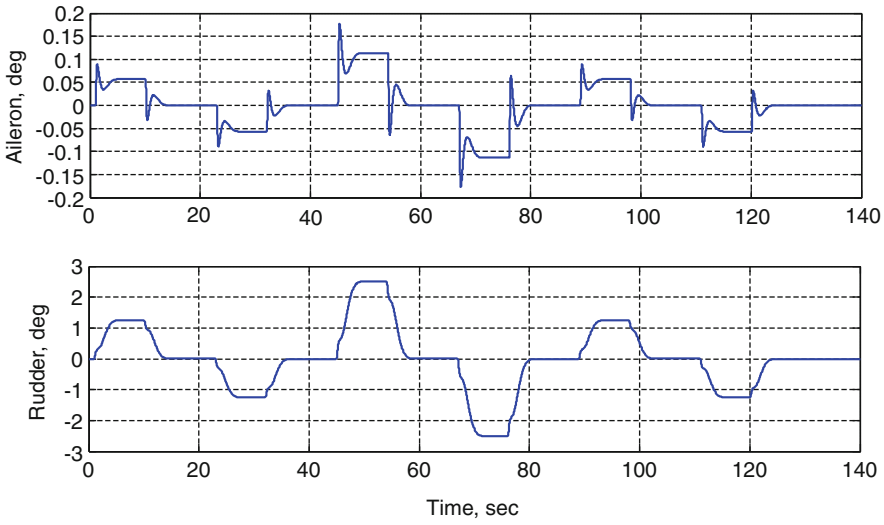


Fig. 11.15 Baseline aileron and rudder deflections in Example 11.5

vehicle behavior under the baseline LQR PI controller. Since there are no uncertainties in the baseline system dynamics, the reference and the actual responses are identical. The required aileron and rudder deflections (Fig. 11.15) are well behaved and definitely reside within realistic actuation limits.

Next, we introduce matched linear-in-parameter uncertainties into the system

$$\dot{x} = A\,x + B\,\Lambda\left(u + \Theta^T\,\Phi\big(x_p\big)\right) + B_{ref}\,y_{cmd}$$

embed the baseline LQR PI solution $u_{bl} = -K_{xLQR}^T\,x$, and arrive at the extended open-loop dynamics:

$$\dot{x} = \underbrace{\left(A - B\,K_{xLQR}^T\right)}_{A_{ref}}\,x + B\,\Lambda\left(u + \Lambda^{-1}K_{xLQR}^T\,x + \Theta^T\,\Phi\big(x_p\big)\right) + B_{ref}\,y_{cmd}$$

Let u_{ad} denote an adaptive control augmentation signal. With the total control input

$$u = u_{bl} + u_{ad}$$

the extended open-loop system becomes

$$\begin{aligned}\dot{x} &= A_{ref}\,x + B\,\Lambda\left(u_{ad} + \underbrace{\left(\Lambda^{-1} - I_{2\times 2}\right)K_{xLQR}^T\,x + \Theta^T\,\Phi\big(x_p\big)}_{\bar{\Theta}^T\,\bar{\Phi}(x)}\right) + B_{ref}\,y_{cmd}\\ &= A_{ref}\,x + B\,\Lambda\left(u_{ad} + \bar{\Theta}^T\,\bar{\Phi}(x)\right) + B_{ref}\,y_{cmd}\end{aligned}$$

which is in the same exact form as in Table 11.2.

For simulation studies, we have selected the following uncertainty-related parameters:

$$\Lambda = 0.5\,I_{2\times 2}, \quad \Phi\big(x_p\big) = \begin{pmatrix}\beta & p & r\end{pmatrix}^T$$
$$\Theta = \begin{pmatrix}4A_p(2,2) & 2A_p(2,3) & 2A_p(2,4)\\ 4A_p(2,1) & 2A_p(3,3) & 2A_p(3,4)\end{pmatrix}^T$$

With 50% control effectiveness reduction in aileron and rudder, these parameters emulate 200% change in the aircraft sideslip coefficients and 100% change in the vehicle roll and yaw stability derivatives. The perturbed system is open-loop unstable. Its command tracking responses under the baseline LQR PI controller become highly oscillatory and thus inadequate. The data are shown below (Fig. 11.16).

Although the baseline controller was able to stabilize the perturbed dynamics, the tracking performance is clearly unacceptable. Also, the corresponding aileron and rudder deflections exhibit the unwanted oscillations (Fig. 11.17).

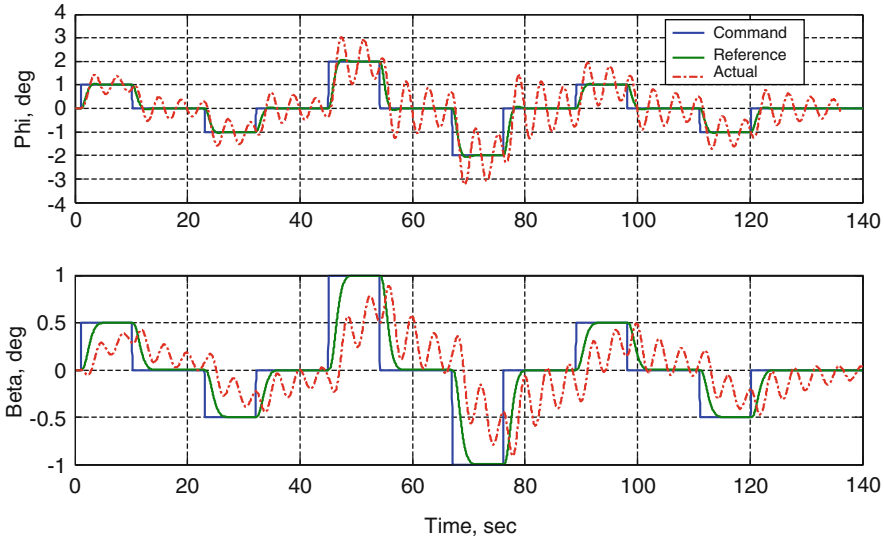


Fig. 11.16 Closed-loop response with uncertainties and LQR PI controller in Example 11.5

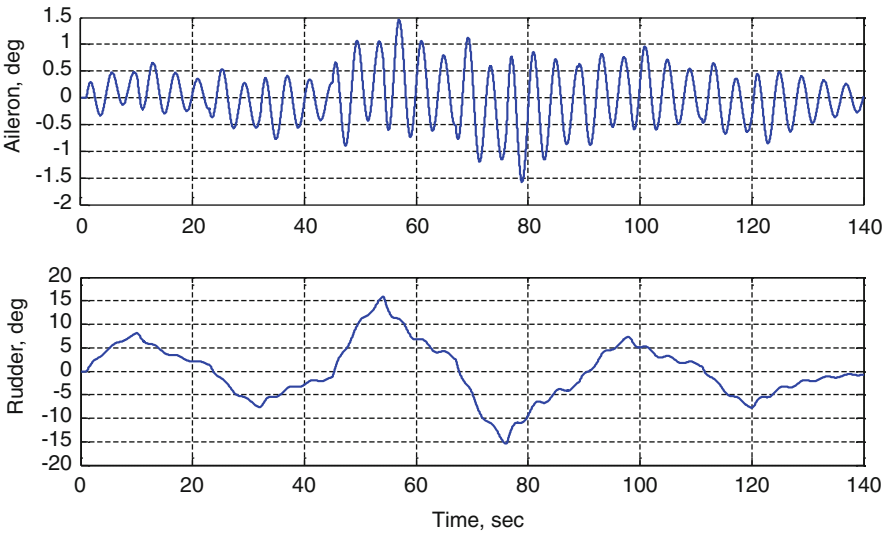


Fig. 11.17 LQR PI control inputs with uncertainties turned on in Example 11.5

In order to mitigate the system uncertainties, we add an adaptive augmentation component in the form

$$u_{ad} = -\hat{\bar{\Theta}}^T \bar{\Phi}(x)$$

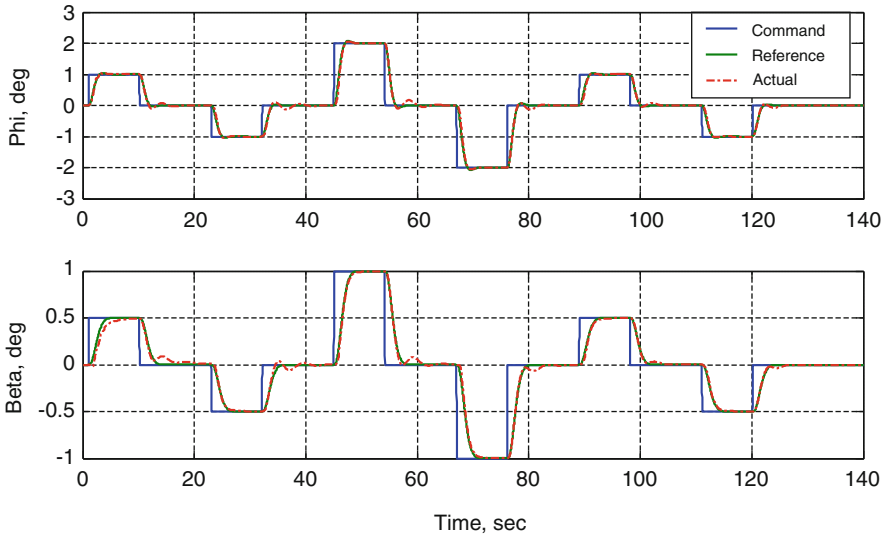


Fig. 11.18 Closed-loop tracking with (LQR PI + Adaptive) controller in Example 11.5

with the estimated parameters $\hat{\bar{\Theta}}(t)$ evolving according to the projection-based adaptive laws as shown in Table 11.2. The reference model is chosen to represent the closed-loop nominal system under the LQR PI controller and without uncertainties. The Q matrix in the Lyapunov algebraic equation is

$$Q = \text{diag}(\,0 \quad 0 \quad 0 \quad 0 \quad 10 \quad 800\,)$$

and the rates of adaptation are

$$\Gamma_\Theta = \text{diag}(\,100 \quad 100 \quad 600 \quad 600 \quad 600 \quad 600\,)$$

With the uncertainties turned on, the (LQR PI + Adaptive) controller recovers the desired closed-loop tracking performance (Fig. 11.18).

The required control effort is reasonable and well within the actuator capabilities of a generic aircraft such as the one considered (Fig. 11.19).

The magnitudes of the estimated parameters are shown in Fig. 11.20.

In this simulation, maximum allowable bounds for the adaptive parameters were set to 10, but the adaptive parameters never reached their bounds. So, it would be interesting to simulate a case when these bounds are reduced below their maximum achieved values. We set the aileron-related max bound to 0.5 and the rudder-related bound to 5. With the same uncertainties activated, Fig. 11.21 shows "graceful degradation" of the system closed-loop tracking performance.

Per design, the adaptive parameters evolve within the smaller projection bounds (Fig. 11.22), and because of that the system performance degraded slightly.

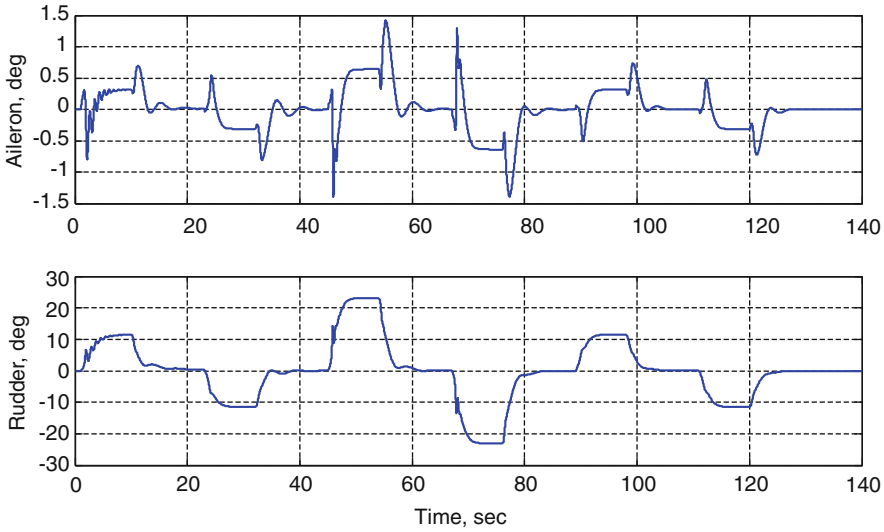


Fig. 11.19 (LQR PI + Adaptive) aileron and rudder deflections in Example 11.5

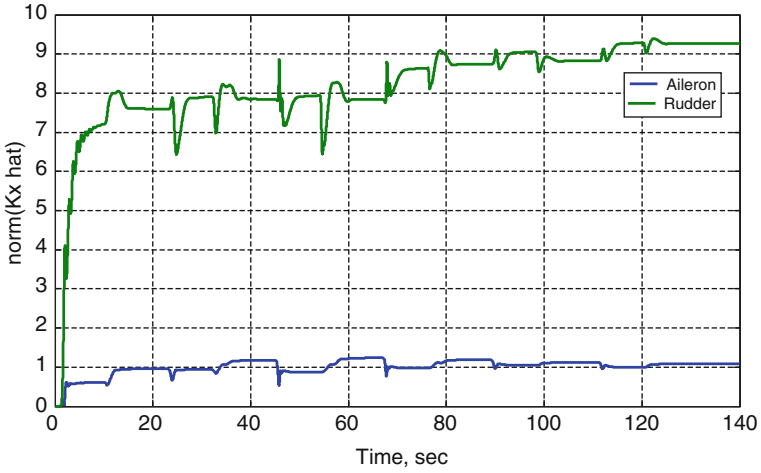


Fig. 11.20 Adaptive parameter dynamics in Example 11.5

However, the aileron and the rudder control activity (Fig. 11.23) remained very similar to the previous simulation case, where we had large projection bounds and thus attained a slightly better performance.

The main purpose of this simulation test is to verify that the Projection Operator has been implemented and functioned correctly. Additionally, we want to expose an

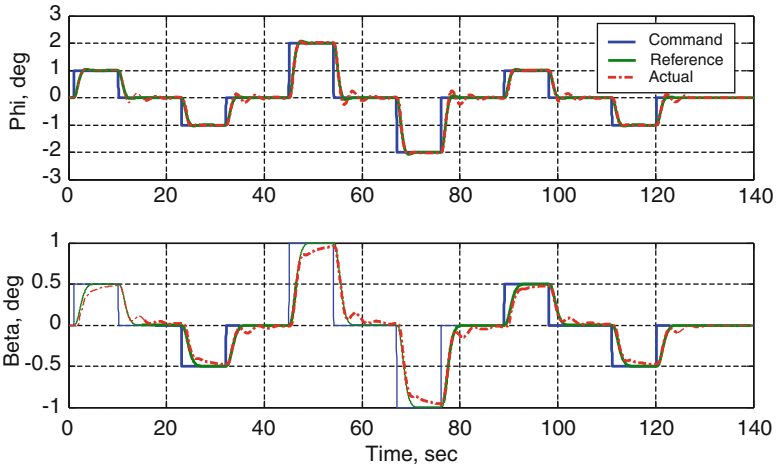


Fig. 11.21 Performance degradation with small projection limits in Example 11.5

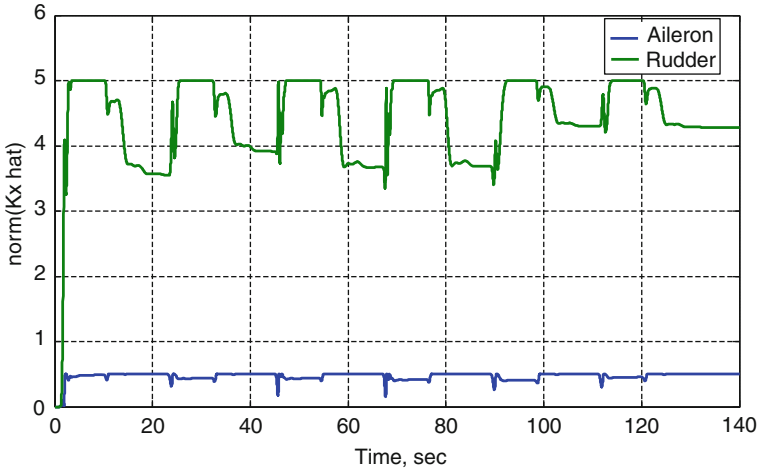


Fig. 11.22 Adaptive parameters with small projection bounds in Example 11.5

iterative nature of a control design process, such as MRAC. Based on theoretical predictions, the control designer is always expected to perform a trade-off study to find the best set of tuning parameters for the selected method, while performing an assessment of simulation trials versus theoretical predictions. □

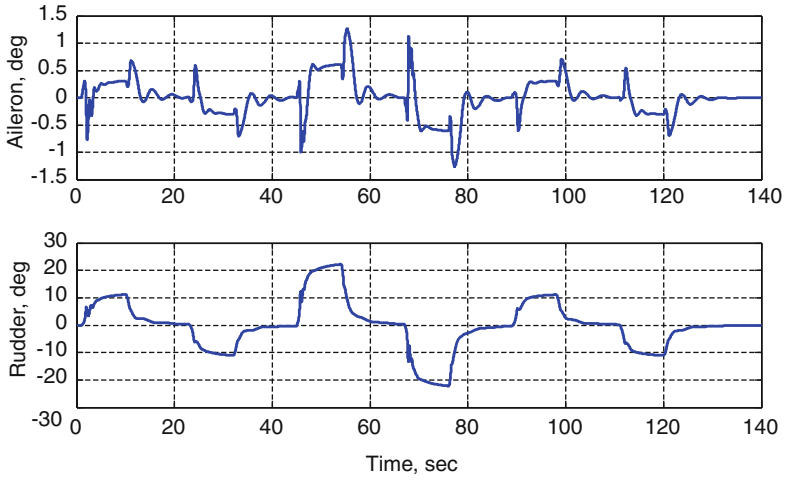


Fig. 11.23 Aileron and rudder deflections with small projection bounds in Example 11.5

11.6 Summary and Discussions

In conclusion, we would like to offer our opinion on the choice of robustness modifications in MRAC systems. These recommendations are not "theoretical" by any means. They are merely based on the authors' extensive experience during the design of MRAC systems for a multitude of aerospace applications.

In our view, any adaptive system must have the dead-zone modification (11.18) or its continuous version (11.19). The latter is the preferred choice since it avoids potential discontinuities in feedback connections. The "must-have" dead-zone modification will prevent adaptive parameters from drifting away.

As seen from (11.15), the adaptive law dynamics without robustness modifications are defined by integrating a nonlinear function, represented by the regressor vector $\Phi(x)$, multiplied by a linear combination of the state tracking errors $(e^T P B)$. This product is further multiplied by a constant matrix Γ_Θ (the integral gain), and finally, it is integrated to yield the adaptive parameters $\hat{\Theta}(t)$ (see Fig. 11.24).

As seen from the block diagram of Fig. 11.24, there is a chain of nonlinear integrators in a feedback loop, whose output constitutes the adaptive parameters. In all practical applications, feedback integrators must be "managed" in the sense that their output signals (i.e., the adaptive parameters) need to be constrained. This prevents integrators against "winding up" due to nonlinear saturation functions in the control channels, where the system achievable control limits are defined and enforced. Control techniques that prevent the integrator windup problems are called the "anti-windup" methods, and the Projection Operator is one of them. This is why we highly recommend using projection-based adaptive laws.

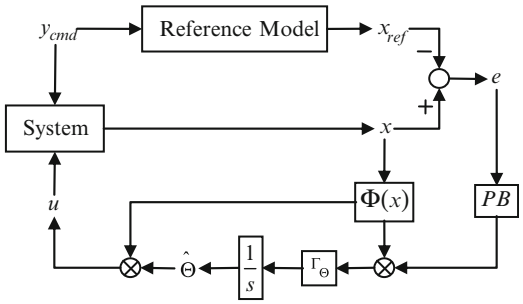


Fig. 11.24 Adaptive system viewed as a nonlinear integral feedback controller

In summary, our suggested MRAC architecture consists of the smoothed dead-zone modification coupled with the Projection Operator

$$\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta},\ \Gamma_\Theta \Phi\, \mu(\|e\|)\, e^T P B\right) \tag{11.62}$$

where $\mu(\|e\|)$ is the Lipchitz-continuous modulation function from (11.19). Essentially, the dead-zone modification protects the adaptive parameters from drifting due to noise, while the Projection Operator bounds the overall adaptive process, and at the same time, it prevents MRAC integrators against the undesirable windup phenomenon.

11.7 Exercises

Exercise 11.1. Prove that the Projection Operator (11.37) is locally Lipschitz.

Exercise 11.2. Simulate the two actuator models from Example 11.4. Select commands to violate position and rate constraints. Compare and discuss your results.

Exercise 11.3. ([10]). Consider a convex hypercube in R^n,

$$\Omega = \left\{\theta \in R^n : \left(\theta_i^{\min} \le \theta_i \le \theta_i^{\max}\right)_{i=1,2,\ldots,n}\right\}$$

where $\left(\theta_i^{\min},\ \theta_i^{\max}\right)$ represent the minimum and maximum bounds for the i^{th} component of the n-dimensional parameter vector θ. Choose a sufficiently small positive constant δ, and define another hypercube

$$\Omega_\delta = \left\{\theta \in R^n : \left(\theta_i^{\min} + \delta \le \theta_i \le \theta_i^{\max} - \delta\right)_{i=1,2,\ldots,n}\right\}$$

such that $\Omega_\delta \subset \Omega$.

For two n-dimensional vectors $(\theta,\ y)$, a rectangular version of the Projection Operator is defined component-wise as

$$\text{Proj}_i\,(\theta,\ y) = \begin{cases} \left(\dfrac{\theta_i^{\max} - \theta_i}{\delta}\right) y_i, & \left[\left(\theta_i > \theta_i^{\max} - \delta\right) \wedge (y_i > 0)\right] \\ \left(\dfrac{\theta_i - \theta_i^{\min}}{\delta}\right) y_i, & \left[\left(\theta_i < \theta_i^{\min} + \delta\right) \wedge (y_i < 0)\right] \\ y_i, \quad \text{otherwise} \end{cases}$$

Suppose that $\theta^* \in \Omega_\delta$ is a constant vector. Prove that for any $\theta \in \Omega$ and for any $y \in R^n$, the following inequality takes place:

$$(\theta - \theta^*)^T\,(\text{Proj}\,(\theta,\ y) - y) \leq 0$$

Let Γ be a positive-definite diagonal matrix. For the system dynamics (11.1), using the above inequality and adaptive laws (11.53) with the rectangular version of the Projection Operator, carry out stability proofs starting from (11.51), arriving at a UUB-type argument about the closed-loop system tracking performance.

Exercise 11.4. A second-order actuator model (transfer function) is given in the form

$$\delta = \left(\frac{\omega_n^2}{s^2 + 2\,\xi\,\omega_n\,s + \omega_n^2}\right)\delta_{cmd}$$

where $(\delta,\ \delta_{cmd})$ are the actual and commanded actuator positions (rad), while $(\xi,\ \omega_n)$ are the actuator model damping ratio and its natural frequency, respectively. Assume $\xi = 0.7$ and $\omega_n = 1$. Simulate the system response to a sinusoidal command. Introduce actuator position and rate limits. Use Projection Operator (11.37) to create an actuator model with position and rate constrains. Create another model using the rectangular version of the Projection Operator from Exercise 11.3. Select actuator commands to violate the actuator position constraints. Simulate both models and compare their performance.

Exercise 11.5. Implement the aircraft lateral-directional data from Example 11.5. Design an (LQR PI + Adaptive) controller using (11.62). Repeat simulation tests from Example 11.5. Compare and discuss your results.

References

1. Khalil, H.: Nonlinear Systems, 3rd edn, p. 07458. Prentice Hall, Upper Saddle River (1996)
2. Peterson, B.B., Narendra, K.S.: Bounded error adaptive control. IEEE Trans. Automat. Contr. **27**, 1161–1168 (1982)
3. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, New Jersey (1995)

4. Slotine, J.J.E., Coetsee, J.A.: Adaptive sliding controller synthesis for nonlinear systems. Int. J. Contr. (1986)
5. Ioannou, P., Kokotovic, P.: Adaptive Systems with Reduced Models. Springer, New York (1983)
6. Narendra, K.S., Annaswamy, A.M.: A new adaptive law for robust adaptive control without persistency of excitation. IEEE Trans.Automat Contr. **32**, 134–145, Feb. (1987)
7. Pomet, J.B., Praly, L.: Adaptive nonlinear regulation: Estimation from the Lyapunov equation. IEEE Trans. Automat. Contr. **37**, 729–740 (1992)
8. Ioannou, P., Fidan, P.: Adaptive Control Tutorial, SIAM, Advances in Design and Control, SIAM, PA (2006)
9. Kreisselmeier, G., Narendra, K.S.: Stable model reference adaptive control in the presence of bounded disturbances. IEEE Trans. Automat. Contr. **27**, 1169–1175 (1982)
10. Khalil, H.K.: Adaptive output feedback control of nonlinear systems represented by input – output models. IEEE Trans. Automat. Contr. **41**, 177–188 (1996)
11. McRuer, D., Ashkenas, I., Graham, D: Aircraft dynamics and automatic control, Princeton University Press, Princeton, New Jersey (1990)

Chapter 12
Approximation-Based Adaptive Control

12.1 Motivation

A typical control design starts with modeling, which is basically a procedure of constructing a mathematical description (such as a set of ordinary differential equations) for the physical system to be controlled. This selected model needs to reflect main features of the physical process. Accurate models are not always better. They may require unnecessarily complex control design and demand excessive computations. From a control point of view, the key in modeling is to capture the essential effects in the system dynamics within an operating range of interest. In addition, a good model should also provide some characterization of the system uncertainties – the so-called unknown unknowns in the physical process. Such a characterization can later be used to perform robust and/or adaptive design or to run Monte Carlo-based analysis, eventually leading to quantification and assessment of the closed-loop system stability, performance, and robustness.

In essence, model uncertainties symbolize the differences between the model and the real physical process. Uncertainties in the system-specific parameters are called "parametric," while all other uncertainties are "nonparametric."

Example 12.1 Point-Mass Dynamics with Parametric Uncertainties For the model of a controlled mass $m\ddot{x} = u$, the uncertainty in m is parametric, while the neglected motor dynamics, measurement noise, and sensor dynamics represent the nonparametric uncertainties. □

Example 12.2 Scalar Dynamics with Nonparametric Uncertainties Consider a scalar model with uncertain dynamics, such as $\dot{x} = f(x) + u + \xi(t)$, where x is the system state, u is the control input, $\xi(t)$ is the process noise, and the function $f(x)$ is unknown. Suppose that

$$f(x) = \sum_{i=1}^{N} \theta_i \varphi_i(x) + \varepsilon(x) = \underbrace{\theta^T \Phi(x)}_{Parametric} + \underbrace{\varepsilon(x)}_{Nonparametric}$$

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_12,

In other words, we assume that the unknown function $f(x)$ can be approximated by a finite linear combination of known basis functions $\varphi_i(x)$ and unknown constant parameters θ_i. In this case, the state-dependent function approximation error $\varepsilon(x)$ and the process noise $\xi(t)$ represent the nonparametric uncertainties, while the unknown constant parameters θ constitute the parametric uncertainty in the system dynamics. In order to characterize the latter, one needs to be able to find a good set of basis functions $\Phi(x)$, such that the approximation error $\varepsilon(x)$ becomes small on a compact (closed and bounded) set. Polynomials, Fourier series expansions, splines, and artificial feedforward neural networks can be used to represent and approximate functions on compact sets. □

In Sect. 12.4, we will design MRAC systems that can cope with both parametric and nonparametric uncertainties. In order to justify our design approach, we begin with a concise background material and an overview of important facts related to function approximation using artificial NNs to represent large classes of functions on given compact sets and within prespecified approximation tolerances.

12.2 Basic Definitions

An artificial feedforward NN is a multi-input multi-output static map composed of many interconnected nonlinear processing elements (neurons) operating in parallel. Figures 12.1 and 12.2 show sketches of two feedforward NNs.

An artificial feedforward NN consists of basic units called the "neurons" and their connections. A block diagram of a single artificial neuron is shown in Fig. 12.3.

Neurons, the basic processing elements of NNs, have two main components: (a) a weighted summer and (b) a nonlinear activation function. The activation functions of interest to us are the radial basis functions (RBFs) and the ridge functions, also called the "sigmoids."

Definition 12.1. Radial Basis Functions (RBFs) *An RBF is a Gaussian in the form*

$$\varphi(x, x_c) = e^{-(x-x_c)^T W (x-x_c)} = e^{-\|x-x_c\|_W^2} \tag{12.1}$$

In (12.1), $x \in R^n$ is the input, $x_c \in R^n$ is the center, and $W = W^T > 0$ is a positive-definite symmetric matrix of weights. Most often, we will write $\varphi(x, x_i) = \varphi_i(x)$ to abbreviate and to denote an RBF which is centered at the i^{th} center x_i.

Other definitions of RBFs are available in the literature [1, 2]. A generic RBF can be defined as $\varphi = \varphi\left(\|x - x_c\|_W\right)$, where $\|x\|_W = \sqrt{x^T W x}$ denotes the weighted Euclidean norm of a vector x. In addition, it is required that $\varphi(x)$ be integrable on R^n and $\int_{R^n} \varphi(x)\, dx \neq 0$. This activation function depends only on the weighted distance $r = \|x - x_c\|_W$ between its current input x and the center x_c. The Gaussian RBF in (12.1) is an example of this type of activation function.

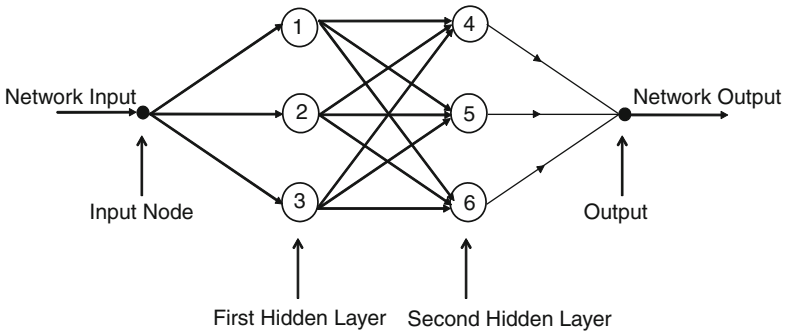


Fig. 12.1 Feedforward neural network with 2 hidden layers and 6 neurons

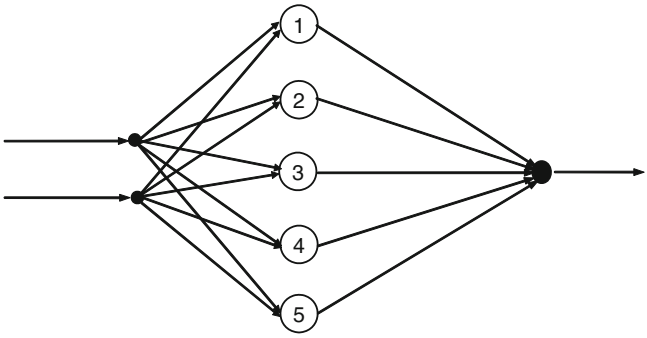


Fig. 12.2 Feedforward neural network with 1 hidden layer and 5 neurons

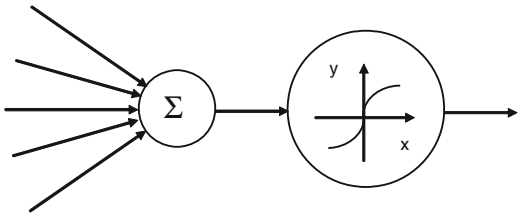


Fig. 12.3 Artificial neuron block diagram

Others include (a) multiquadrics, $\varphi(r) = \sqrt{(r^2 + c^2)}, \quad c > 0$, and (b) inverse multiquadrics, $\varphi(r) = \frac{1}{\sqrt{(r^2+c^2)}}$, $c > 0$.

Definition 12.2. Ridge Functions *A ridge function or a sigmoid is a nonlinear scalar map* $\sigma : R \to R$ *of the form*

$$\sigma = \sigma\left(w^T x + b\right) \tag{12.2}$$

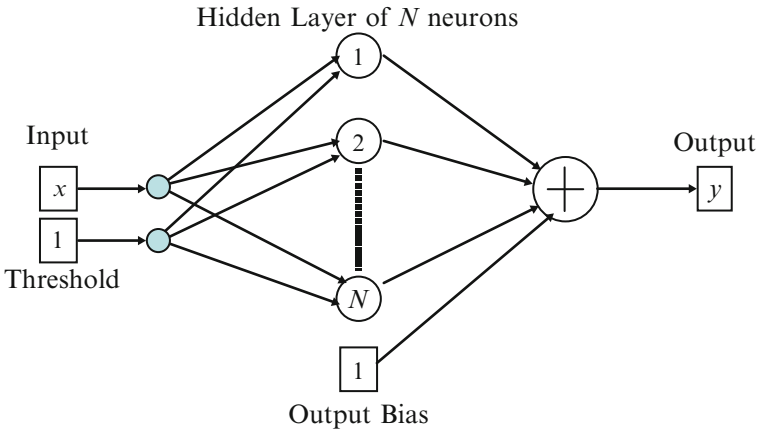


Fig. 12.4 Single-hidden-layer feedforward NN with N neurons

where $w \in R^n$ *denotes the vector of weights,* b *is a scalar threshold, and* $\sigma(\bullet)$ *is a scalar nonlinear function (not necessarily continuous) on* R*, with the following property:*

$$\lim_{v \to \pm\infty} \sigma(v) < \infty \tag{12.3}$$

The two most common examples of a ridge function are (a) the logistic sigmoid, $\sigma(v) = \frac{1}{1+e^{-v}}$, and (b) the hyperbolic tangent, $\sigma(s) = \frac{1-e^{-v}}{1+e^{-v}}$.

A feedforward NN with N neurons in its hidden layer is shown in Fig. 12.4.

Formally speaking, a feedforward NN is a map from R^n to R^m, that is,

$$y = NN(x) : R^n \to R^m \tag{12.4}$$

Definition 12.3. Sigmoidal Feedforward NNs *A sigmoidal feedforward NN with* N *neurons is a map from* R^{n} *to* R^{m} *in the form*

$$NN(x) = W^T \vec{\sigma}\left(V^T x + \theta\right) + b \tag{12.5}$$

where $W \in R^{N \times m}$ *is the matrix of the outer-layer weights;*

$$\vec{\sigma}(x) = \left(\sigma\left(V_1^T x + \theta_1\right) \quad \ldots \quad \sigma\left(V_N^T x + \theta_N\right)\right)^T \in R^N$$

is the vector of N *sigmoids;* $V \in R^{n \times N}$ *is the matrix of the inner-layer synaptic weights, with its* i^{th} *column denoted by* $V_i \in R^n$*;* $\theta \in R^N$ *is the vector of thresholds; and* $b \in R^m$ *denotes the NN bias vector.*

Definition 12.4. Feedforward RBF NNs *A feedforward RBF NN is a map from* R^{n} *to* R^{m} *in the form*

$$NN(x) = \theta^T \begin{pmatrix} \varphi\Big(\|x - C_1\|_{W_1}\Big) \\ \vdots \\ \varphi\Big(\|x - C_N\|_{W_N}\Big) \end{pmatrix} + b = \underbrace{\begin{pmatrix} \theta^T & b \end{pmatrix}}_{\Theta^T} \underbrace{\begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_N(x) \\ 1 \end{pmatrix}}_{\Phi(x)} = \Theta^T \Phi(x) \quad (12.6)$$

where $\Theta = \begin{pmatrix} \theta^T & b \end{pmatrix}^T \in R^{(N+1)\times m}$ *is the vector of weights,* $C_i \in R^n$ *is the center of the* i^{th} *receptive field,* $W_i = W_i^T > 0$ *is the norm weighting matrix,* $b \in R^m$ *is the NN bias, and* $\Phi(x) = \begin{pmatrix} \varphi_1(x) & \dots & \varphi_N(x) & 1 \end{pmatrix}^T \in R^{N+1}$ *is the regressor vector, whose components are the basis (activation) functions* $\varphi_i(x) = \varphi\Big(\|x - C_i\|_{W_i}\Big)$ *and the unit function.*

Often in practical applications, the symmetric positive-definite matrix W in (12.6) is chosen to be diagonal and in the form

$$W_i = \frac{1}{2\,\sigma_i^2}, \quad (i = 1, \dots, N)$$

where σ_i represents the width of the i^{th} Gaussian function. In this case,

$$\varphi_i(x) = e^{-\frac{\|x - C_i\|^2}{2\sigma_i^2}}$$

becomes the i^{th}component of the regressor vector $\Phi(x)$ in (12.6). Also, components of the regressor can be constructed using the Gaussian,

$$\varphi_i(x) = e^{-\left(\frac{N}{d_{\max}^2}\right)\|x - C_i\|^2}$$

whose standard deviation (width) σ is fixed according to the spread of the centers C_i, N is the number of centers, and $d_{\max}$ is the maximum distance between the chosen centers. Here, the standard deviation σ of all the isotropic Gaussian RBF components is fixed at

$$\sigma = \frac{d_{\max}}{\sqrt{2N}}$$

This formula ensures that the individual RBFs are not too peaked or too flat. Both of these two extreme conditions should be avoided.

12.3 Approximation Properties of Feedforward Neural Networks

Feedforward NNs have been shown to be capable of approximating generic classes of functions on compact sets and to within any prespecified tolerance. This property of feedforward NNs is often referred to as the universal approximation, while the NNs themselves are often called the universal approximators. Related theorems are stated below without proofs.

Theorem 12.1. Micchelli's Theorem [3] *Let* $\varphi = \varphi(r)$ *be the Gaussian, the multiquadrics, or the inverse multiquadrics function. Let* $\{x_i\}_{i=1}^{N}$ *be a set of distinct points in* R^n. *Then, the* $(N \times N)$ *interpolation matrix* Φ, *whose* $(i, j)^{th}$ *element is* $\varphi_{ij} = \varphi\left(\left\|x_i - x_j\right\|\right)$, *is nonsingular.* ■

There is a large class of RBFs that is covered by Micchelli's theorem. In fact, this theorem provides a theoretical basis for RBF-based function approximation and regression techniques. Specifically, using an RBF $\varphi = \varphi(r)$ and a finite set of N *points* $\{x_i\}_{i=1}^{N}$ in R^n, the above theorem assures that it is always possible to approximate functions $f(x)$ on a grid of points, using a linear combination of RBFs in the form $\hat{f}(x) = \sum_{i}^{N} \theta_i \varphi(x - x_i)$, such that $f(x_i) = \hat{f}(x_i)$ for all $\{x_i\}_{i=1}^{N}$.

Theorem 12.2. Universal Approximation Theorem for Sigmoidal NNs [4] *Any continuous function* $f(x) : R^n \to R$ *can be uniformly approximated by a single-hidden-layer NN,*

$$\forall \varepsilon > 0,\ \exists N,\ W,\ b,\ V,\ \theta,\ \forall x \in X \subset R^n : \left\| \underbrace{W^T \vec{\sigma}\left(V^T x + \theta\right) + b}_{NN(x)} - f(x) \right\|_{\infty} \leq \varepsilon \quad (12.7)$$

with a bounded monotone-increasing continuous activation vector function $\bar{\sigma}(\bullet)$ *on a compact domain* $X \subset R^n$. ■

The universal approximation theorem extends to the class of L_1 functions defined on compact sets. In that case, it is assumed that the selected activation function is a bounded measurable sigmoid, and the approximation is understood in terms of the L_1 functional norm.

Theorem 12.3. Rates of Approximation Theorem for Sigmoidal NNs [5] *Consider a class of functions* $f(x)$ *on* R^n *for which there is a Fourier representation of the form*

$$f(x) = \int_{R^n} e^{i\omega x} \tilde{f}(\omega)\, d\omega$$

for some complex-valued function $\tilde{f}(\omega)$ *for which* $\omega \tilde{f}(\omega)$ *is integrable and define*

$$C_f = \int_{R^n} \|\omega\| \left\|\tilde{f}(\omega)\right\| d\omega < \infty$$

Then, for every function $f(x)$ *with* C_f *finite and every* $N \geq 1$, *there exists a sigmoidal NN of the form (12.5), such that*

$$\|f(x) - NN(x)\|_{L_2}^2 = \int_{\|x\| \leq r} (f(x) - NN(x))^2 \, dx \leq \frac{\left(2\, r\, C_f\right)^2}{N}$$

■

Functions with C_f finite are continuously differentiable on R^d. Moreover, the NN approximation error is measured by the L_2-norm on the ball of radius r.

Theorem 12.4. Universal Approximation Theorem for RBF NNs [6] *Let* $\varphi(x) : R^n \to R$ *be an integrable bounded continuous function, and assume that*

$$\int_{R^n} \varphi(x) \, dx \neq 0$$

Then, for any continuous function f(x) and any $\varepsilon > 0$, *there is an RBF NN with N neurons, a set of centers* $\{C_i\}_{i=1}^N$, *and a common width* $\sigma > 0$,

$$\hat{f}(x) = \sum_{i=1}^{N} \theta_i \underbrace{\varphi\left(\frac{x - C_i}{\sigma}\right)}_{\varphi_i(x)} = \Theta^T \Phi(x)$$

such that

$$\|f(x) - NN(x)\|_{L_2}^2 = \int_{\|x\| \leq r} (f(x) - NN(x))^2 \, dx \leq \varepsilon = \mathrm{O}\left(N^{-\frac{1}{n}}\right)$$

■

In conclusion, we present a comparison of key features and properties possessed by the sigmoidal NNs and by the RBF NNs:

- Both RBF and sigmoidal NNs are universal approximators.
- An RBF NN depends on the Euclidean distances between the input vector x and the centers C_i. On the other hand, a sigmoidal NN depends on the sum of the inner product of the input vector x with its synaptic weight vectors V_i and a bias θ.
- Sigmoidal NNs provide $\mathrm{O}\left(N^{-\frac{1}{2}}\right)$ rate of approximation which does not explicitly depend on the dimension of x. The rate of approximation for the RBF NNs is of

order $O\left(N^{-\frac{1}{2n}}\right)$, and consequently, it decreases exponentially as the dimension of the input vector x increases. This phenomenon is called the "curse of dimensionality" (due to R. E. Bellman).

- An RBF has a local support, while a sigmoid does not. The local support implies learning and adaptation ability of RBF NNs. Sigmoidal NNs adapt but do not learn.

With the specific reference to artificial NNs in control, it is their ability to represent inherently nonlinear mappings and hence to model nonlinear dynamical systems, which is the feature to be most readily exploited in the synthesis of nonlinear controllers. This is the topic that we shall begin to address in the next section.

12.4 Adaptive Control with State Limiting Constraints

We are interested in the design of adaptive command tracking controllers for affine-in-control multi-input multi-output (MIMO) dynamical systems in the form

$$\dot{x} = A\,x + B\,\Lambda\,(u + f(x)) + \xi(t) \tag{12.8}$$

where $x \in R^n$ is the system state vector, $u \in R^m$ is the control input, $B \in R^{n\times m}$ is a known constant matrix, $A \in R^{n\times n}$ and $\Lambda \in R^{m\times m}$ (a diagonal matrix with positive elements) are unknown constant matrices, $f(x) : R^n \to R^m$ is a state-dependent (possibly nonlinear) uncertainty, and $\xi(t) \in R^n$ is a bounded time-varying unknown disturbance, whose upper bound

$$\|\xi(t)\| \leq \xi_{\max} \tag{12.9}$$

is known.

In the previous chapters, we have developed model reference adaptive control (MRAC) command tracking design methods, assuming that the matched nonlinear uncertainty admits an exact parameterization in the form $f(x) = \Theta^T\,\Phi(x)$, with constant unknown coefficients $\Theta \in R^{N\times m}$ and with a preselected known locally Lipschitz-continuous regressor vector $\Phi(x) \in R^N$.

In this section, we shall extend our design to nonlinear-in-parameter functions. Our main assumption here is that these uncertainties can be parameterized (i.e., approximated on a bounded closed set within a small tolerance) using artificial NNs, whose fixed basis functions are known (such as sigmoids with fixed inner-layer weights and thresholds or Gaussians with fixed centers).

In particular, using the universal approximation properties of artificial NNs, we shall assume that the unknown mapping $f(x) : R^n \to R^m$ can be approximated/represented on a known compact set $X \subset R^n$ by an NN with N fixed neurons $\varphi_i(x)$ and using unknown ideal constant connection weights that are stored in a matrix $\Theta \in R^{N\times m}$:

$$f(x) = \Theta^T\,\Phi(x) + \varepsilon(x) \tag{12.10}$$

Without a loss of generality, we define the approximation set

$$X = X_R = \{x \in R^n : \ \|x\| \leq R\} \tag{12.11}$$

to represent a sphere of a finite and known radius R. We shall also assume that inside the sphere, the ideal (unknown to the designer) approximation can be achieved within a known approximation tolerance $\varepsilon_0 > 0$:

$$\|\varepsilon(x)\| \leq \varepsilon_0, \quad \forall x \in X_R \tag{12.12}$$

Outside of X_R, we postulate that the approximation error can be upper-bounded (norm-wise) by a known possibly unbounded positive scalar function $\varepsilon_{\max}(x)$:

$$\|\varepsilon(x)\| \leq \varepsilon_{\max}(x), \quad \forall x \notin X_R \tag{12.13}$$

The control objective is to design a state feedback MRAC system, which guarantees boundedness of all signals in the corresponding closed-loop dynamics; while forcing the system state $x(t) \in R^n$, follow the state $x_{ref}(t) \in R^n$ of the desired exponentially stable reference model

$$\dot{x}_{ref} = A_{ref}\, x_{ref} + B_{ref}\, r(t) \tag{12.14}$$

driven by a known bounded time-varying reference command signal $r(t) \in R^m$,

$$\|r(t)\| \leq r_{\max}, \quad \forall t \geq 0 \tag{12.15}$$

whose maximum bound $r_{\max}$ is known.

We are going to construct an adaptive command tracking controller, capable of operating in the presence of the system structured and unstructured uncertainties, where the latter are represented by (a) the state-dependent function approximation error $\varepsilon(x) \in R^m$ in (12.12) and (b) the bounded disturbance $\xi(t) \in R^n$ in (12.9).

Let us immediately note that while the disturbance term $\xi(t)$ is uniformly bounded, the approximation error $\varepsilon(x)$ becomes bounded only if the system state $x(t)$ is located inside the sphere X_R. So, in addition to command tracking, we need a state limiter logic that would keep the system state within the approximation set X_R, or it would bring it back to X_R, if the state happens to be outside of the approximation set. This observation suggests a control law in the form

$$\begin{aligned} u &= \underbrace{\hat{K}_x^T\, x - \hat{\Theta}^T\, \Phi(x)}_{u_x} + (1 - \mu(x))\ \underbrace{\hat{K}_r^T\, r}_{u_r} + \mu(x)\, u_{sl} \\ &= u_x + (1 - \mu(x))\, u_r + \mu(x)\, u_{sl} \end{aligned} \tag{12.16}$$

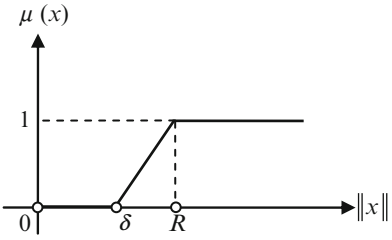


Fig. 12.5 State modulation function

where

$$u_x = \hat{K}_x^T x - \underbrace{\hat{\Theta}^T \Phi(x)}_{\hat{f}} \tag{12.17}$$

is the adaptive stabilizing term with adaptive gains $\hat{K}_x \in R^{n\times m}$ and $\hat{\Theta} \in R^{N\times m}$,

$$u_r = \hat{K}_r^T r \tag{12.18}$$

is the adaptive command tracking component with an adaptive feedforward command gain $\hat{K}_r \in R^{m\times m}$, $\mu(x)$ is the state modulation function, and u_{sl} is the state limiter.

This controller will be designed to operate as follows. The adaptive stabilizing term u_x will provide closed-loop stability for all $x \in X_R$. At the same time, the adaptive command tracking component u_r will force the system to follow commanded trajectories of the desired reference model. If the system state $x(t)$ starts outside of X_R or if the system disturbance $\xi(t)$ pushes it outside of the approximation set, then command tracking will subside and the state limiter u_{sl} will be responsible to bring x back into X_R (in finite time), where command tracking would resume.

These two modes of operation, tracking and state limiting, are governed by the state limiter modulation function $\mu(x)$, which essentially "gain-schedules" the controller (12.16) to smoothly transition between the adaptive tracking and the state limiting tasks.

We define the state limiter function as

$$\mu(x) = \max\left(0,\ \min\left(1,\ \frac{\|x\| - \delta R}{(1-\delta)R}\right)\right) \tag{12.19}$$

where $0 < \delta < 1$ is a constant. A sketch of this function is shown in Fig. 12.5.

By definition,

$$\mu(x) = \begin{cases} 0, & x \in X_{\delta R} \\ 1, & x \notin X_R. \end{cases} \tag{12.20}$$

and the positive constant δ defines the width of an annulus inside X_R where $0 \leq \mu(x) \leq 1$. According to (12.16), (12.19), and (12.20), the state limiter will turn the adaptive tracking on for all $x \in X_{\delta R}$, or it will turn the state limiter on for $x \notin X_R$. In the annulus set $\delta R \leq \|x\| \leq R$, both tasks are active, with one of them fading out and the other fading in, linearly in $\|x\|$.

Our choice of the modulation function in (12.19) is by no means unique. Other definitions can easily be constructed to accomplish the gain-scheduling feature of the adaptive controller (12.16).

In order for such a control solution to exist, the model matching conditions must hold:

$$\begin{aligned} A + B\,\Lambda\,K_x^T &= A_{ref} \\ B\,\Lambda\,K_r^T &= B_{ref} \end{aligned} \tag{12.21}$$

where K_x, K_r denote the ideal unknown constant feedback and feedforward gain matrices, respectively. Only existence of the ideal gains is assumed, whereas their knowledge will not be required to perform the design.

In (12.17), $\hat{f}(x) = \hat{\Theta}^T\,\Phi(x)$ is the function approximator. It is easy to see that the related function approximation error,

$$\Delta f(x) = \hat{f}(x) - f(x) \tag{12.22}$$

depends linearly on the parameter estimation error $\Delta\Theta = \hat{\Theta} - \Theta$:

$$\Delta f(x) = \underbrace{\left(\hat{\Theta} - \Theta\right)^T}_{\Delta\Theta}\,\Phi(x) - \varepsilon(x) = \Delta\Theta^T\,\Phi(x) - \varepsilon(x) \tag{12.23}$$

Using the model matching conditions (12.21), the open-loop system dynamics (12.8) can be written as

$$\dot{x} = A_{ref}\,x + B_{ref}\,r + B\,\Lambda\left(u - K_x^T\,x - K_r^T\,r + f(x)\right) + \xi(t) \tag{12.24}$$

Substituting (12.16) into (12.24) yields

$$\begin{aligned} \dot{x} &= A_{ref}\,x + B_{ref}\,r + B\,\Lambda\left(u_x + (1-\mu)\,u_r + \mu\,u_{sl} - K_x^T\,x - K_r^T\,r + f\right) + \xi \\ &= A_{ref}\,x + B_{ref}\,r \\ &\quad + B\,\Lambda\left(u_x - K_x^T\,x + f + (1-\mu)\left(u_r - K_r^T\,r\right) + \mu\left(u_{sl} - K_r^T\,r\right)\right) + \xi \end{aligned} \tag{12.25}$$

With (12.17) and (12.18), we get

$$\begin{aligned}\dot{x} &= A_{ref}\, x + B_{ref}\, r + \xi \\ &\quad + B\,\Lambda \left(\underbrace{\left(\hat{K}_x - K_x\right)^T}_{\Delta K_x} x - \underbrace{\left(\hat{\Theta} - \Theta\right)^T}_{\Delta\Theta} \Phi + \varepsilon + (1-\mu)\, \underbrace{\left(\hat{K}_r - K_r\right)^T}_{\Delta K_r} r \right) \\ &\quad + B\,\Lambda\,\mu\left(u_{sl} - K_r^T\, r\right)\end{aligned} \tag{12.26}$$

or, equivalently,

$$\begin{aligned}\dot{x} &= A_{ref}\, x + B_{ref}\, r + \xi \\ &\quad + B\,\Lambda\left(\Delta K_x^T\, x - \Delta\Theta^T\, \Phi + \varepsilon + (1-\mu)\,\Delta K_r^T\, r + \mu\left(u_{sl} - K_r^T\, r\right)\right)\end{aligned} \tag{12.27}$$

where

$$\Delta K_x = \hat{K}_x - K_x, \quad \Delta K_r = \hat{K}_r - K_r, \quad \Delta\Theta = \hat{\Theta} - \Theta \tag{12.28}$$

are the parameter estimation errors. Let

$$e = x - x_{ref} \tag{12.29}$$

denote the state tracking error. Subtracting (12.14) from (12.27) gives the state tracking error dynamics:

$$\begin{aligned}\dot{e} &= A_{ref}\, e + \xi \\ &\quad + B\,\Lambda\left(\Delta K_x^T\, x - \Delta\Theta^T\, \Phi + \varepsilon + (1-\mu)\,\Delta K_r^T\, r + \mu\left(u_{sl} - K_r^T\, r\right)\right)\end{aligned} \tag{12.30}$$

We introduce matrix $P \in R^{n\times n}$ to represent the unique positive-definite symmetric solution of the algebraic Lyapunov equation,

$$P\, A_{ref} + A_{ref}^T\, P = -Q, \quad Q = Q^T > 0 \tag{12.31}$$

and consider a quadratic radially unbounded Lyapunov function candidate in the form

$$\begin{aligned}V(e,\, \Delta K_x,\, \Delta K_r,\, \Delta\Theta) &= e^T P\, e \\ &\quad + \mathrm{tr}\left(\left[\Delta K_x^T\, \Gamma_x^{-1}\, \Delta K_x + \Delta K_r^T\, \Gamma_r^{-1}\, \Delta K_r + \Delta\Theta^T\, \Gamma_\Theta^{-1}\, \Delta\Theta\right]\Lambda\right)\end{aligned} \tag{12.32}$$

where $\Gamma_x = \Gamma_x^T > 0, \quad \Gamma_r = \Gamma_r^T > 0, \quad \Gamma_\Theta = \Gamma_\Theta^T > 0$ are the rates of adaptation. The time derivative of V, along the trajectories of the error dynamics (12.30), is given by

$$\begin{aligned}\dot{V} =& \dot{e}^T P\, e + e^T P\, \dot{e} + 2\,\mathrm{tr}\Big(\Big[\Delta K_x^T\, \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T\, \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta\Theta^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}}\Big]\, \Lambda\Big) \\ =& -e^T\, Q_{ref}\, e + 2\, e^T P\, \xi \\ &+ 2\, e^T P\, B\, \Lambda\, \big(\Delta K_x^T\, x - \Delta\Theta^T\, \Phi + \varepsilon + (1-\mu)\, \Delta K_r^T\, r + \mu\, (u_{sl} - K_r^T\, r)\big) \\ &+ 2\,\mathrm{tr}\Big(\Big[\Delta K_x^T\, \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T\, \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta\Theta^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}}\Big]\, \Lambda\Big) \end{aligned} \tag{12.33}$$

Regrouping terms further yields

$$\begin{aligned}\dot{V} =& -e^T\, Q\, e + 2\, e^T P\, \xi + 2\, e^T P\, B\, \Lambda\, \big(\mu\, (u_{sl} - K_r^T\, r) + \varepsilon\big) \\ &+ 2\, \Big[e^T P\, B\, \Lambda\, \Delta K_x^T\, x + \mathrm{tr}\Big(\Delta K_x^T\, \Gamma_x^{-1} \dot{\hat{K}}_x\, \Lambda\Big)\Big] \\ &+ 2\, \Big[(1-\mu)\, e^T P\, B\, \Lambda\, \Delta K_r^T\, r + \mathrm{tr}\Big(\Delta K_r^T\, \Gamma_r^{-1} \dot{\hat{K}}_r\, \Lambda\Big)\Big] \\ &+ 2\, \Big[-e^T P\, B\, \Lambda\, \Delta\Theta^T\, \Phi + 2\,\mathrm{tr}\Big(\Delta\Theta^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\Theta}}\, \Lambda\Big)\Big] \end{aligned} \tag{12.34}$$

Via the vector trace identity $a^T b = \mathrm{tr}(b\, a^T)$, which is valid for any two column vectors a and b, we obtain

$$\begin{aligned} \underbrace{e^T P\, B\, \Lambda}_{a^T}\, \underbrace{\Delta K_x^T\, x}_{b} &= \mathrm{tr}\left(\underbrace{\Delta K_x^T\, x}_{b}\, \underbrace{e^T P\, B\, \Lambda}_{a^T}\right) \\ \underbrace{e^T P\, B\, \Lambda}_{a^T}\, \underbrace{\Delta K_r^T\, r}_{b} &= \mathrm{tr}\left(\underbrace{\Delta K_r^T\, r}_{b}\, \underbrace{e^T P\, B\, \Lambda}_{a^T}\right) \\ \underbrace{e^T P\, B\, \Lambda}_{a^T}\, \underbrace{\Delta\Theta^T\, \Phi(x)}_{b} &= \mathrm{tr}\left(\underbrace{\Delta\Theta^T\, \Phi}_{b}\, \underbrace{e^T P\, B\, \Lambda}_{a^T}\right) \end{aligned} \tag{12.35}$$

Substituting (12.35) into (12.34) results in

$$\begin{aligned}\dot{V} =& -e^T\, Q\, e + 2\, e^T P\, \xi + 2\, e^T P\, B\, \Lambda\, \big(\mu\, (u_{sl} - K_r^T\, r) + \varepsilon\big) \\ &+ 2\,\mathrm{tr}\Big(\Delta K_x^T\, \Big[\Gamma_x^{-1} \dot{\hat{K}}_x + x\, e^T P\, B\Big]\, \Lambda\Big) \\ &+ 2\,\mathrm{tr}\Big(\Delta K_r^T\, \Big[\Gamma_r^{-1} \dot{\hat{K}}_r + (1-\mu)\, r\, e^T P\, B\Big]\, \Lambda\Big) \\ &+ 2\,\mathrm{tr}\Big(\Delta\Theta^T\, \Big[\Gamma_\Theta^{-1}\, \dot{\hat{\Theta}} - \Phi\, e^T P\, B\Big]\, \Lambda\Big) \end{aligned} \tag{12.36}$$

In order to keep the adaptive gains $\hat{K}_x,\ \hat{K}_r,\ \hat{\Theta}$ uniformly bounded, we shall employ projection-based adaptive laws (Sects. 11.3, and 11.4) in the form

$$\begin{aligned}\dot{\hat{K}}_x &= \text{Proj}\left(\hat{K}_x, -\Gamma_x\, x\, e^T P B\right)\\ \dot{\hat{K}}_r &= \text{Proj}\left(\hat{K}_r, -(1-\mu)\,\Gamma_r\, r\, e^T P B\right)\\ \dot{\hat{\Theta}} &= \text{Proj}\left(\hat{\Theta},\ \Gamma_\Theta\, \Phi\, e^T P B\right)\end{aligned} \tag{12.37}$$

where $\text{Proj}(\Theta,\ Y)$ is the Projection Operator, which maps two $(n \times N)$ matrices, $\Omega = \left[\,\vec{\theta}_1 \quad \ldots \quad \vec{\theta}_N\,\right] \in R^{n\times N}$ and $Y = \left[\,\vec{y}_1 \quad \ldots \quad \vec{y}_N\,\right] \in R^{n\times N}$, into a $(n \times N)$ matrix, denoted by $\text{Proj}(\Omega,\ Y)$. The operator is defined column-wise,

$$\text{Proj}\,(\Omega,\ Y) = \left(\text{Proj}\left(\vec{\theta}_1,\ \vec{y}_1\right) \quad \ldots \quad \text{Proj}\left(\vec{\theta}_N,\ \vec{y}_N\right)\right) \tag{12.38}$$

and its column vector components are

$$\text{Proj}\left(\vec{\theta}_j,\ \vec{y}_j\right) = \begin{cases} y - \dfrac{\Gamma\,\nabla f_j\left(\nabla f_j\right)^T}{\left(\nabla f_j\right)^T \Gamma\,\nabla f_j}\,\vec{y}_j f_j, & \text{if} \quad \left[f_j > 0 \wedge \left(\vec{y}_j^T\,\nabla f_j\right) > 0\right] \\ y & \text{if not} \end{cases} \tag{12.39}$$

where $f\left(\vec{\theta}_j\right) : R^n \to R$ is a convex function that defines the desired parameter domain. Given $\theta_j^{\max}$, the maximum allowable magnitude of the column vector $\vec{\theta}_j$ and a small constant $\varepsilon_j > 0$, the convex function is

$$f\left(\theta_j\right) = \frac{\left(1+\varepsilon_j\right)\left\|\theta_j\right\|^2 - \left(\theta_j^{\max}\right)^2}{\varepsilon_j\left(\theta_j^{\max}\right)^2} \tag{12.40}$$

With the adaptive laws (12.37) and because of the previously established convex properties of the Projection Operator, one can show that the derivative of the Lyapunov function (12.36) satisfies the following inequality:

$$\dot{V} \le -e^T Q\, e + 2\, e^T P\, \xi + 2\, e^T P B\, \Lambda\left(\mu\left(u_{sl} - K_r^T\, r\right) + \varepsilon\right) \tag{12.41}$$

In order to eventually prove stability and bounded command tracking, we need to analyze if $\dot{V}$ can be made nonpositive outside of a compact set. Toward that end, let us suppose that $x \notin X_R$. Then, $\mu(x) = 1$ and (12.41) becomes

$$\begin{aligned}\dot{V} &\le -\,e^T Q\, e + 2\, e^T P\, \xi + 2\, e^T P B\, \Lambda\left(u_{sl} - K_r^T\, r + \varepsilon\right)\\ &\le -\,\lambda_{\min}(Q)\,\|e\|^2 + 2\,\|e\|\,\lambda_{\max}(P)\,\xi_{\max} + 2\, e^T P B\, \Lambda\left(u_{sl} - K_r^T\, r + \varepsilon\right)\end{aligned} \tag{12.42}$$

In order to make the right-hand side of (12.42) nonpositive, we choose the state limiting control u_{sl} in the form

$$u_{sl} = -k_{sl}\,(x)\; sgn\big(B^T\,P\,e\big) \tag{12.43}$$

where $k_{sl}(x) > 0$ represents the state limiter gain, and the sign function is understood component-wise. Then,

$$e^T P\,B\,\Lambda\,u_{sl} = \sum_{i=1}^{m} \big(e^T P\,B\big)_i\,\lambda_i\,u_{sl\,i} = -k_{sl}(x)\;\sum_{i=1}^{m} \big|e^T P\,B\big|_i\,\lambda_i \tag{12.44}$$

and with (12.43) inserted into (12.42), we get

$$\begin{aligned}
\dot{V} &\le -\lambda_{\min}(Q)\,\|e\|^2 + 2\,\|e\|\,\lambda_{\max}(P)\,\xi_{\max} \\
&\quad -2\,k_{sl}(x)\;\sum_{i=1}^{m} \big|e^T P\,B\big|_i\,\lambda_i + e^T P\,B\,\Lambda\,\left(-K_r^T\,r + \varepsilon\right) \\
&= -\lambda_{\min}(Q)\,\|e\|^2 + 2\,\|e\|\,\lambda_{\max}(P)\,\xi_{\max} \\
&\quad -2\sum_{i=1}^{m} \big|e^T P\,B\big|_i\,\lambda_i\,\left(k_{sl}(x) - sgn\Big(\big(e^T P\,B\big)_i\Big)\,\left(K_r^T\,r - \varepsilon\right)\right)
\end{aligned} \tag{12.45}$$

If we now choose the state limiter gain to be large enough,

$$k_{sl}(x) = K_{r\;\max}\,r_{\max} + \varepsilon_{\max}(x) \tag{12.46}$$

where $K_{r\;\max} \ge \|K_r\|$ and $r_{\max} = \max\limits_{t\,\ge\,0} \|r(t)\|$, then

$$\begin{aligned}
\dot{V} &\le -\lambda_{\min}(Q)\,\|e\|^2 + 2\,\|e\|\,\lambda_{\max}(P)\,\xi_{\max} \\
&= -\lambda_{\min}(Q)\,\|e\|\,\left(\|e\| - 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\xi_{\max}\right) < 0
\end{aligned} \tag{12.47}$$

outside of the compact set:

$$E_0 = \left\{e \in R^n :\;\; \|e\| \le 2\,\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\,\xi_{\max} = e_0\right\} \tag{12.48}$$

Therefore, $e(t)$ enters a larger compact set $\tilde{E}_0 \supset E_0$, in finite time T [7–9]. Moreover, for all $t \ge T$, there must exist a positive constant $\bar{e}_0$, such that

$$\bar{e}_0 \ge \|e(t)\| = \big\|x(t) - x_{ref}(t)\big\| \ge \|x(t)\| - \big\|x_{ref}(t)\big\| \tag{12.49}$$

Hence,

$$\|x(t)\| \le \bar{e}_0 + \left\|x_{ref}(t)\right\| \le \bar{e}_0 + x_{ref\ \max} \tag{12.50}$$

where the upper bound $x_{ref\ \max}$ can be explicitly computed based on $r_{\max}$ and the properties of the reference model (12.14). So, if we choose the approximation set X_R to be large enough,

$$R > \bar{e}_0 + x_{ref\ \max} \tag{12.51}$$

then for all $t \ge T$,

$$\|x(t)\| \le R \tag{12.52}$$

that is, the system state $x(t)$ enters X_R in finite time T and remains there afterward.

Inside the set X_R, the state modulation function is zero, and the approximation error $\varepsilon(x)$ becomes small. Hence, the Lyapunov function time derivative from (12.41) can be upper-bounded as

$$\begin{aligned}\dot{V} &\le -e^T Q\, e + 2\, e^T P\, \xi + 2\, e^T P B\, \Lambda\, \varepsilon \\ &\le -\lambda_{\min}(Q)\, \|e\|^2 + 2\, \|e\|\, \lambda_{\max}(P)\, (\xi_{\max} + \|B\|\, \Lambda_{\max}\, \varepsilon_0) \\ &= -\lambda_{\min}(Q)\, \|e\| \left(\|e\| - 2\, \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\, (\xi_{\max} + \|B\|\, \Lambda_{\max}\, \varepsilon_0) \right)\end{aligned} \tag{12.53}$$

Consequently, $\dot{V} < 0$ outside of the compact set:

$$E_1 = \left\{ e \in R^n :\ \|e\| \le 2\, \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}\, (\xi_{\max} + \|B\|\, \Lambda_{\max}\, \varepsilon_0) = e_1 \right\} \tag{12.54}$$

Hence, $e(t)$ enters a compact set $\tilde{E}_1 \supset E_1$ in finite time T_1 [7–9], where it will remain afterward. Similarly to (12.49) and (12.50), we get an upper bound

$$\|x(t)\| \le \bar{e}_1 + x_{ref\ \max} \tag{12.55}$$

for some positive constant $\bar{e}_1 \ge \bar{e}_0$. In order to ensure that $x(t)$ remains inside X_R, it is sufficient to strengthen the inequality (12.51) and assume

$$R > \bar{e}_1 + x_{ref\ \max} \tag{12.56}$$

As we have already mentioned, the adaptive parameters will remain uniformly ultimately bounded (UUB). This property is due to the convexity of the Projection Operator (Sect. 11.3, Lemmas 11.3, 11.4). Consequently, all trajectories of the closed-loop system (12.8), (12.16), and (12.37) are UUB. Moreover, the tracking

Table 12.1 Projection-based MRAC design with state limiter constraints

Open-loop plant	$\dot{x} = A\,x + B\,\Lambda\,(u + f(x)) + \xi(t)$
Reference model	$\dot{x}_{ref} = A_{ref}\,x_{ref} + B_{ref}\,r(t)$
State tracking error	$e = x - x_{ref}$
Lyapunov equation for adaptive laws	$P\,A_{ref} + A_{ref}^T\,P = -Q$
Total control input	$u = u_x + (1 - \mu(x))\,u_r + \mu(x)\,u_{sl}$
State modulation function	$\mu(x) = \max\left(0,\ \min\left(1,\ \dfrac{\|x\| - \delta R}{(1-\delta)R}\right)\right)$
Adaptive stabilizing term	$u_x = \hat{K}_x^T\,x - \hat{\Theta}^T\,\Phi(x)$
Adaptive tracking term	$u_r = \hat{K}_r^T\,r$
State limiter	$u_{sl} = -k_{sl}(x)\ sgn\left(B^T\,P\,e\right)$
State limiter gain	$k_{sl}(x) = K_{r\ \max}\,r_{\max} + \varepsilon_{\max}(x)$
MRAC laws with Projection Operator	$\dot{\hat{K}}_x = \text{Proj}\left(\hat{K}_x, -\Gamma_x\,x\,e^T P\,B\right)$ $\dot{\hat{K}}_r = \text{Proj}\left(\hat{K}_r,\ \ (\mu(x) - 1)\ \Gamma_r\,r\ e^T P\,B\right)$ $\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta},\ \Gamma_\Theta\,\Phi\,e^T P\,B\right)$

error $e = x - x_{ref}$ enters a neighborhood of the origin, in finite time. The radius of this neighborhood (i.e., the tracking error ultimate bound) is determined by the minimum level set of the Lyapunov function V, which contains the set

$$\begin{aligned} E = &\{e \in R^n :\ \|e\| \le \bar{e}_1\} \\ &\times \left\{\hat{K}_x \in R^{n\times m} :\ \left\|\left(\hat{K}_x\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j,\quad 1 \le j \le m\right\} \\ &\times \left\{\hat{K}_r \in R^{m\times m} :\ \left\|\left(\hat{K}_m\right)_j\right\| \le \left(\hat{K}_r^{\max}\right)_j,\quad 1 \le j \le m\right\} \\ &\times \left\{\hat{\Theta} \in R^{N\times m} :\ \left\|\left(\hat{\theta}\right)_j\right\| \le \left(\hat{\theta}^{\max}\right)_j,\quad 1 \le j \le m\right\} \end{aligned} \tag{12.57}$$

outside of which $\dot{V} \le 0$.

This argument completes the design and analysis of the MRAC controller with state limiting constraints for MIMO dynamics with both structured and unstructured uncertainties. We summarize the derived design equations in Table 12.1.

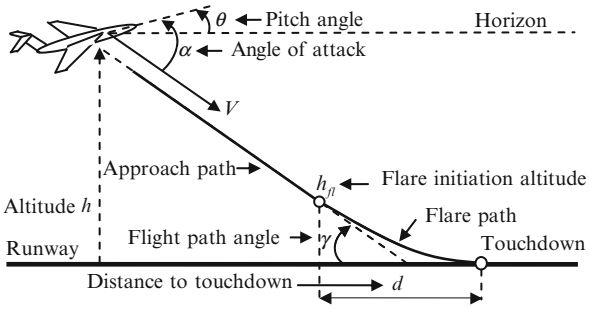


Fig. 12.6 Aircraft on final approach to landing

Next, as an illustrative example, we are going to design an adaptive automatic landing system for a generic medium-size transport aircraft.

Example 12.3 Automatic Landing System for a Medium-Size Transport Aircraft Modern transport aircraft are equipped with automatic landing systems whose sole purpose is to fly the vehicle along the desired trajectory, all the way until a predetermined touchdown point on a runway (Fig. 12.6).

On final approach to a landing, an aircraft would extend its wing leading edges (slats), move wing trailing edges (flaps) down, and deploy its landing gear. As a result, the vehicle aerodynamic drag increases, and the airspeed decreases. With the flaps and slats extended, the aircraft wing would be optimized (per design) to produce a sufficient lift force and to enable a low-speed landing, with a gentle touchdown at the designated runway touchdown point.

For clarity, we assume that the runway is parallel to the horizon and that the aircraft undergoes vertical and longitudinal motion only, that is, the vehicle can change its vertical and forward velocity components and it can also pitch up or down. In this case, the vehicle primary control inputs are engine thrust (δ_{th}, %) and elevators (collectively movable tail surfaces, δ_e, deg). The regulated outputs are represented by the true airspeed V (ft/s) and altitude above the runway (h, ft). From Fig. 12.6, it is not difficult to see that the aircraft angle of attack α (rad), the pitch angle (θ, deg), and the flight path angle γ (rad) satisfy the following equality:

$$\alpha = \theta - \gamma$$

Another important relation exists between the vehicle rate of climb $\dot{h}$ (ft/s), the runway velocity $\dot{d}$, the airspeed V, and the flight path angle γ. From Fig. 12.6 (for a small flight path angle), we get

$$\dot{h} = V \sin\gamma \approx V\gamma$$
$$\dot{d} = V \cos\gamma \approx V$$

The desired trajectory for the aircraft to follow consists of two segments: (a) straight line approach and (b) flare. Typical approach angles range between negative 2 and 3 degrees of γ, with the desired airspeed of 140–160 knots.

During the approach phase, the aircraft is commanded to fly a constant airspeed and a constant flight path angle. On the other hand, the main purpose of the flare is to slowdown the aircraft rate of descent (called the "sink rate") and to make a smooth transition from the selected approach glide slope (i.e., the flight path angle γ) to a shallow angle, at an altitude of approximately 50–65 f. above the runway.

A moderate flare can be described by a linear first-order differential equation such as

$$\dot{h} = -\frac{1}{\tau_h}\, h, \quad h\big(t_{fl}\big) = h_{fl}$$

where $\big(t_{fl},\ h_{fl}\big)$ are the flare initiation time and altitude, respectively. Also, τ_h is a positive time constant. The flare initiation altitude h_{fl} and the time constant τ_h can be chosen such that the vehicle would make a smooth transition from approach to flare and it would touch down within a predetermined distance along the runway. Let us formally define these two requirements.

A smooth transition from the approach phase ($V_{cmd} = V_0, \quad \gamma_{cmd} = \gamma_0$) to flare implies that at $t = t_{fl}$, the following relation must take place:

$$\dot{h}\big(t_{fl}\big) = V\gamma\big(t_{fl}\big) \approx V_{cmd}\,\gamma_{cmd} = -\frac{1}{\tau_h}\, h_{fl}$$

In addition, we impose a restriction on the runway distance traveled in $4\,\tau_{fl}$ seconds from the start of the flare maneuver,

$$\int_{t_{fl}}^{t_{fl}+4\,\tau_h} V(t)\, dt = d$$

where d is the desired distance to touchdown (see Fig. 12.2). Assuming constant airspeed throughout the entire maneuver, $V(t) \approx V_{cmd}$ gives

$$\int_{t_{fl}}^{t_{fl}+4\,\tau_h} V(t)\, \mathrm{d}t \approx V_{cmd}\, 4\,\tau_h = d$$

Then, the flare time constant is

$$\tau_h = \frac{d}{4\,V_{cmd}}$$

and the flare initiation altitude can be computed as

$$h_{fl} = -\tau_h V_{cmd}\, \gamma_{cmd} = -\frac{d}{4}\, \gamma_{cmd}$$

For simulation purposes, we consider a generic midsize transport aircraft flying wings-level at an altitude of $h_0 = 300$ ft above ground, with its landing gear down and with flaps/slats extended. The vehicle true airspeed is $V_0 = 250$ ft/s. The corresponding longitudinal linear (nominal) dynamics are of the form

$$\underbrace{\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 & 0 \\ -0.001 & -0.632 & 1 & 0 & 0 \\ 0 & -0.759 & -0.518 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -250 & 0 & 250 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} V \\ \alpha \\ q \\ \theta \\ h \end{pmatrix}}_{x}$$

$$+ \underbrace{\begin{pmatrix} 10.1 & 0 \\ 0 & -0.0086 \\ 0.025 & -0.011 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}}_{u} \Leftrightarrow \boxed{\dot{x} = A\,x + B\,u}$$

We wish to emphasize that our model represents a generic midsize aircraft and that the linear data are selected for the purposes of design, analysis, and simulation [11, p. 300].

As the aircraft approaches the runway, it will experience a significant increase in its aerodynamic lift force and the pitching moment. This phenomenon is called the "ground effect." Flying in close proximity to the ground drastically changes the airflow beneath and past the airplane. As a result, the ground effect tends to make the vehicle float along the runway.

In order to properly account for the ground effect, we need to modify the aircraft linear dynamics. The vehicle aerodynamic forces and moments depend on the relative motion of the aircraft with respect to the atmosphere. In our example, dynamics of these forces and moments are defined by the first three equations. The ground effect induces a change in the vertical (updraft) linear displacement of the air mass, and so, the aircraft aerodynamic forces and moments depend on the difference $\alpha - \alpha_g(h)$ between the aircraft angle of attack α and the angle of attack induced by the vertical updraft $\alpha_g(h)$, which in turn represents a uniformly bounded function of the ground proximity (i.e., altitude) h. So, the ground effect phenomenon can be embedded into the linear model as follows:

$$\underbrace{\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 & 0 \\ -0.001 & -0.632 & 1 & 0 & 0 \\ 0 & -0.759 & -0.518 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -250 & 0 & 250 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} V \\ \alpha \\ q \\ \theta \\ h \end{pmatrix}}_{x}$$
$$+ \underbrace{\begin{pmatrix} 10.1 & 0 \\ 0 & -0.0086 \\ 0.025 & -0.011 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}}_{u} + \underbrace{\begin{pmatrix} -18.984 \\ 0.632 \\ 0.759 \\ 0 \\ 0 \end{pmatrix}}_{B_g} \alpha_g(h)$$

or, equivalently,

$$\dot{x} = A\,x + B\,u + B_g\,\alpha_g(h)$$

where we have added an extra term $B_g\,\alpha_g(h)$, with a constant vector B_g, whose first three components are equal to the opposite of the corresponding values in the second column of A. This modification reflects our observation that the first three equations in the aircraft dynamics depend on the relative (with respect to the air mass) angle of attack $\alpha - \alpha_g(h)$.

It is not difficult to see that B_g can be reconstructed as a linear combination of the columns in B:

$$B \underbrace{\begin{pmatrix} -1.8796 \\ -73.2718 \end{pmatrix}}_{\theta_g} = B_g$$

In other words, the ground effect represents a matched uncertainty, and the resulting model takes the form of (12.8):

$$\dot{x} = A\,x + B\,\big(u + \theta_g\,\alpha_g(h)\big)$$

Let us make a quick remark about the ground effect matching condition: It is not a requirement for our design. Since $\alpha_g(h)$ is a uniformly bounded function of h, it can be treated similar to the bounded disturbance $\xi(t)$ in (12.8), as long as we can ensure that h is bounded. So, the ground effect unmatched effects on the aircraft dynamics can also be mitigated (see Exercise 12.3).

Continuing on, we define the system-regulated output to consist of the aircraft true airspeed and altitude (same as the aircraft height above the runway):

$$y = \begin{pmatrix} V \\ h \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{C} x = C\,x$$

Accurate aerodynamic data that describe the ground effect are often not available or highly uncertain, yet their undesirable influences on the aircraft landing performance must be taken into account. In order to mitigate these uncertainties, we shall design a (robust + adaptive) automatic landing flight controller, with predictable and quantifiable landing performance characteristics. Specifically, we are going to design a robust adaptive controller to simultaneously track commanded airspeed V_{cmd} and commanded altitude h_{cmd}. These two external commands are grouped into the external vector signal:

$$r = \begin{pmatrix} V_{cmd} & h_{cmd} \end{pmatrix}^T$$

Our selection of these two specific commands will enable automatic steering of the aircraft along a given flight path, all the way to a designated touchdown point on the runway. So, our main control goal is to design u to force y to follow the external vector signal r in the presence of unknown ground effects.

Beginning with the design of a baseline controller for automatic landing, we use the aircraft model without the ground effect uncertainty and employ the familiar LQR method. The baseline control input is

$$u_{bl} = K_x^T\,x + K_r^T\,r$$

where $K_x \in R^{5\times 2}$ and $K_r \in R^{2\times 2}$ are the baseline feedback and feedforward gain matrices, respectively. These gains can be calculated as follows. We choose

$$Q_{lqr} = \text{diag}(\,.02 \quad 0 \quad 0 \quad 0 \quad 1\,), \quad R_{lqr} = \text{diag}(\,20 \quad 20\,)$$

to compute K_x using the LQR method,

$$K_x^T = -\begin{pmatrix} 0.1173 & -89.1740 & 42.8761 & 140.0007 & 0.2340 \\ 0.0186 & -40.6065 & 4.3798 & 58.6016 & 0.2127 \end{pmatrix}$$

from the reference (nominal closed-loop) matrix,

$$A_{ref} = A + B\,K_x^T$$

and then, determine K_r such that the closed-loop baseline system DC gain, from the commanded input r to the regulated output y, is the (2×2) identity matrix:

$$\left[\text{DC Gain} = -C\,A_{ref}^{-1}\,B\,K_r^T = I_{2\times 2}\right] \Rightarrow \left[K_r^T = -\left(C\,A_{ref}^{-1}\,B\right)^{-1}\right]$$

Table 12.2 Reference model eigenvalues in Example 12.3

Eigenvalue	Damping, n/d	Frequency, rad/s
$-0.647 \pm 1.03\,j$	0.531	1.22
$-0.529 \pm 0.158\,j$	0.958	0.552
-1.39	1.0	1.39

This gives

$$K_r^T = \begin{pmatrix} -0.7753 & 0.8531 \\ 0.2340 & 0.2127 \end{pmatrix}$$

and the reference model dynamics, as in (12.14),

$$\dot{x}_{ref} = \underbrace{\left(A + B\,K_x^T\right)}_{A_{ref}} x_{ref} + \underbrace{\left(B\,K_r^T\right)}_{B_{ref}} r$$

whose eigenvalues are shown in Table 12.2.

We now turn our attention to the definition of the desired altitude profile. Given the approach airspeed $V_0 = 250$ (ft/s) and the target glide slope $\gamma_0 = -2.5$ (deg), we set the runway distance at $d = 3,000$ (ft), compute the corresponding flare time constant,

$$\tau_h = \frac{d}{4\,V_0} = 3.0\ (s)$$

and define the flare initiation altitude:

$$h_{fl} = -\frac{d}{4}\,\gamma_0 = 32.7249\ (\text{ft})$$

Then, we compute the commanded altitude trajectory, starting from the initial altitude $h_0 = 300$ (ft) and continuing all the way down to the runway touchdown point:

$$h_{cmd}(t) = \begin{cases} h_0, & \text{if } 0 \le t \le 1 \\ h_0 + V_0\,\gamma_0\,(t-1), & \text{if } h_{cmd} > h_{fl},\ t > 1 \\ e^{-\frac{1}{\tau_{fl}}\left(t-1-t_{fl}\right)}\,h_{fl}, & \text{if } h_{cmd} \le h_{fl},\ t > 1 \end{cases}$$

The resulting altitude command profile is shown in Fig. 12.7.

Note that during the first 1 s of flight, we set the altitude command constant. This will enable a smooth initiation of the landing sequence.

With the baseline controller turned on and without the ground effect, the baseline closed-loop system tracking performance is satisfactory (see Fig. 12.8).

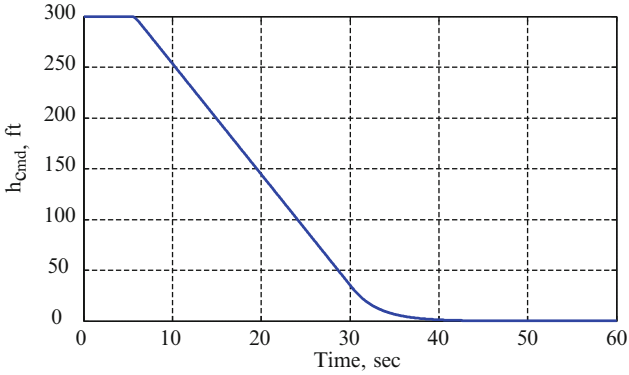


Fig. 12.7 Approach, flare, and landing altitude profile in Example 12.3

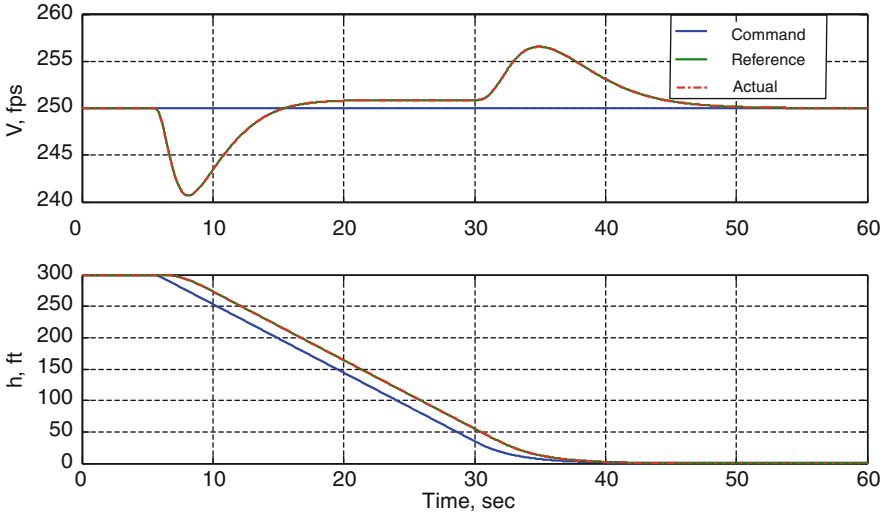


Fig. 12.8 Baseline closed-loop system performance (no ground effect) in Example 12.3

Required for the baseline landing maneuver, the elevator and thrust values are very benign (Fig. 12.9).

For pure academic purposes, we shall use the following equation to emulate the ground effect:

$$\alpha_g(h) = -0.0698\,(1 - \tanh(0.1\ (h - 60)))$$

The ground effect equation is plotted in Fig. 12.10.

As seen from the plot, the ground effect contributes to as much as 8 degree of angle of attack change (negative), as the aircraft approaches the runway.

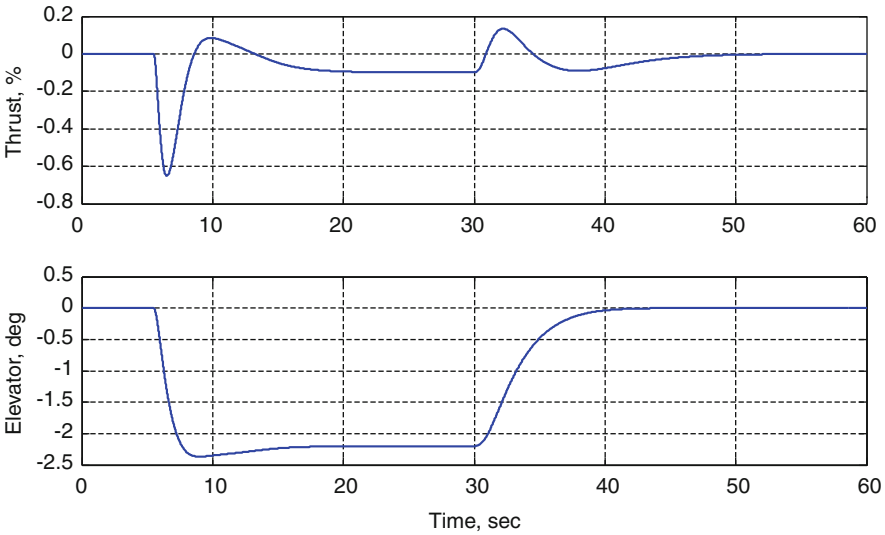


Fig. 12.9 Elevator and thrust data during baseline landing (no ground effect) in Example 12.3

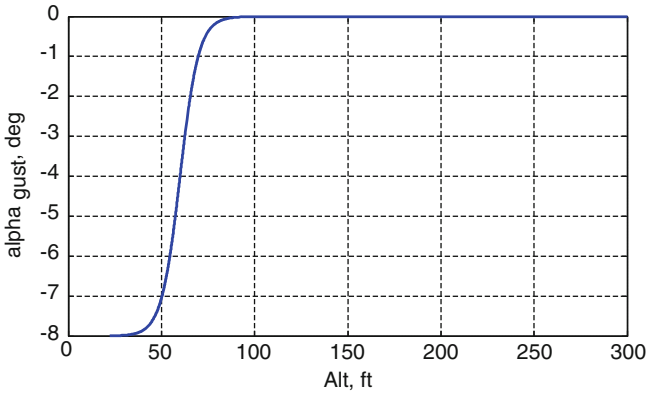


Fig. 12.10 Incremental angle of attack data due to ground effect in Example 12.3

Turning the ground effect on, while using only the baseline controller, results in a significant degradation of the aircraft landing performance (Fig. 12.11).

The simulation data show that while operating under the baseline controller only and in the presence of the ground effect, the vehicle floats along the runway, while its airspeed increases and deviates from its commanded value.

The corresponding elevator and thrust control inputs remain within reasonable limits (Fig. 12.12).

So, the baseline system attempts to counteract the unknown ground effect by reducing thrust to keep the speed down and by moving the elevator trailing edge up

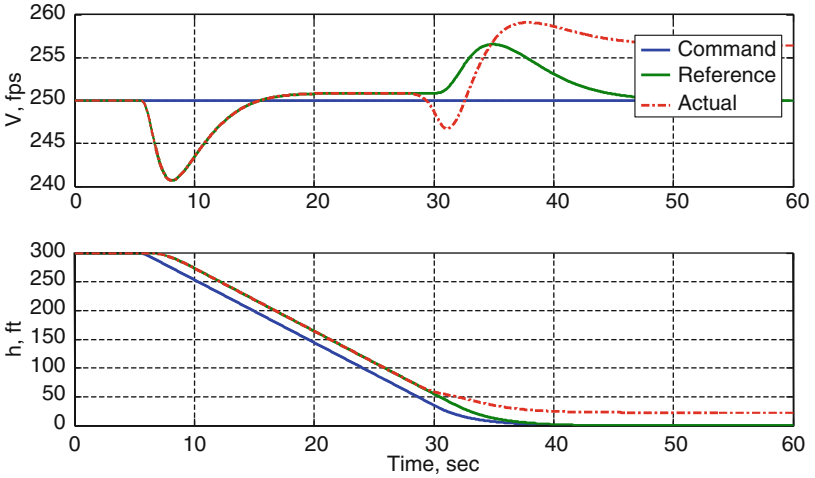


Fig. 12.11 Baseline closed-loop performance during landing with ground effect in Example 12.3

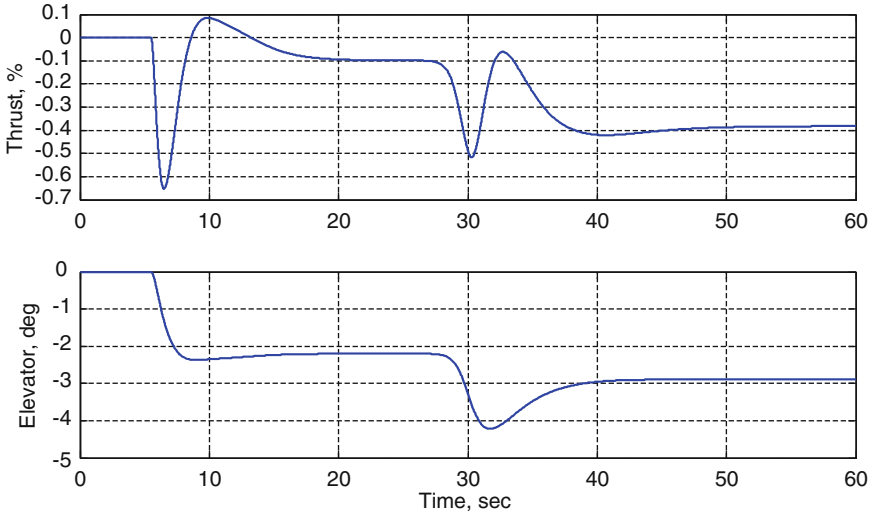


Fig. 12.12 Elevator and thrust data during baseline landing with ground effect in Example 12.3

to stabilize the aircraft pitching motion. Nevertheless, the baseline controller fails in the sense that overall, the vehicle landing performance is clearly unacceptable.

Next, we design an adaptive augmentation to help the baseline system cope with the ground effect-induced unknown effects. First, we choose the regressor vector with five altitude-dependent RBFs and with a single constant bias:

$$\Phi = \begin{pmatrix} \Phi_1(h) & \Phi_2(h) & \Phi_3(h) & \Phi_4(h) & \Phi_5(h) & 1 \end{pmatrix}^T$$

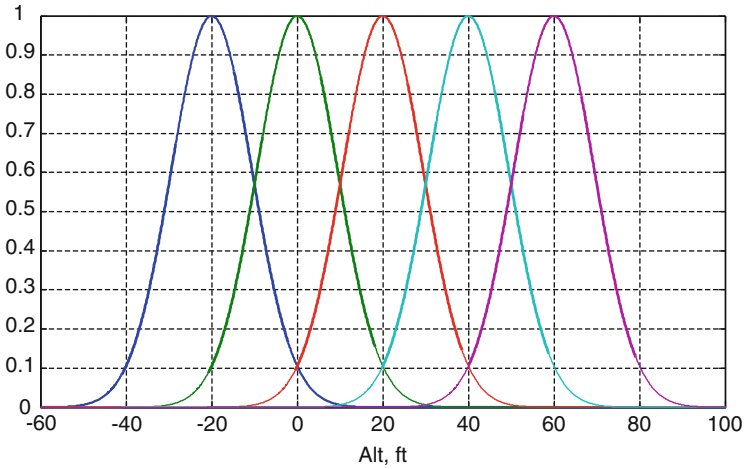


Fig. 12.13 RBF selection for adaptive control design in Example 12.3

The selected RBFs are uniformly distributed on the altitude interval $[-20,\ 60]$, with 20-ft separation from each other. All five RBFs have the same input scaling:

$$\Phi_i(h) = \exp\left(-0.0056\,(h - h_i)^2\right), \quad i = 1\,,\ \ldots\,,\ 5$$

When plotted versus altitude, these functions give a homogeneous coverage of the altitude range, where the ground effect is prevalent (Fig. 12.13).

Using the selected regressor vector Φ, it is possible to closely approximate the ground effect-induced angle of attack function $\alpha_g(h)$ on the interval of interest (see Exercise 12.3).

To design an adaptive augmentation, we choose

$$Q = \text{diag}(\,1 \quad 0 \quad 1 \quad 0 \quad 0\,)$$

to solve the algebraic Lyapunov Eq. (12.31) for P, select rates of adaptation,

$$\Gamma_x = \Gamma_r = 0, \quad \Gamma_\Theta = 20\,I_{6\times 6}$$

and form the adaptive laws per Table 12.1.

After several design trials, we have decided to set the modulation function $\mu(x)$ to zero, since its contribution to improving landing performance is negligible (in this case).

Total control is defined as an adaptive augmentation of the baseline LQR system,

$$u = u_{bl} + u_{ad}$$

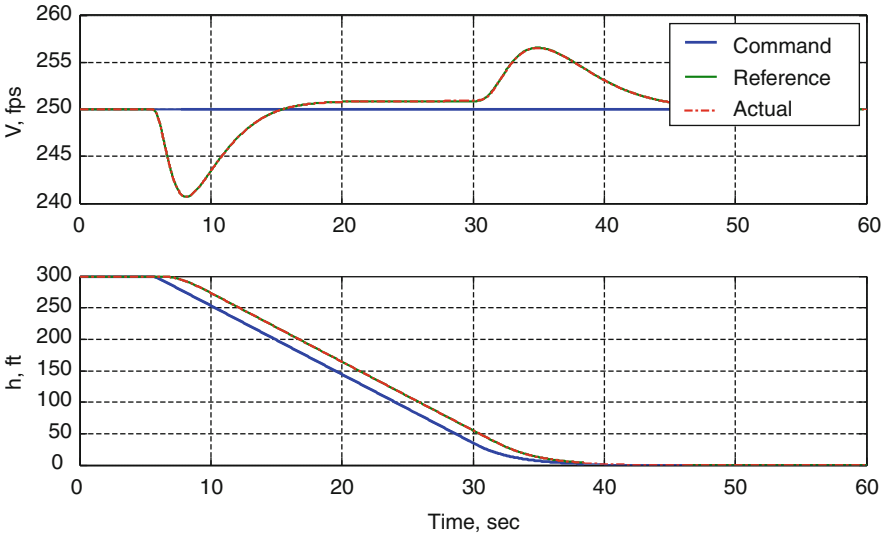


Fig. 12.14 (Baseline + Adaptive) closed-loop performance with ground effect in Example 12.3

where

$$u_{ad} = -\hat{\Theta}^T \Phi(h)$$

represents the adaptive component. This is a slight deviation from the design equations in Table 12.1, where we have an adaptive controller without a baseline system (see Exercise 12.2).

With the (baseline + adaptive) controller turned on and in the presence of the unknown ground effect, the system closed-loop performance is well recovered to that of the desired baseline (Fig. 12.14).

In fact, the data are almost indistinguishable from the baseline tracking (Fig. 12.8). However, once the aircraft descents below 60 ft, where the ground effect is active, the required control inputs (Fig. 12.15) differ from the baseline data (Fig. 12.9).

Yet, all controls remain smooth and reside within practical limits. In addition, the adaptive augmentation provides a sufficiently close estimate of the ground effect. This "bonus" outcome can be attributed to the fact that the ground effect persistently excites the vehicle dynamics, and as a result, the adaptive parameters converge to their constant unknown values (see Exercise 12.3). □

In conclusion, we note that the adaptive feedback/feedforward design method from Table 12.1 can be modified to incorporate a robust baseline controller with proportional and integral feedback (see Exercise 12.4). This would eliminate feedforward connections, which in its own right may become a desirable feature or even a requirement in some applications.

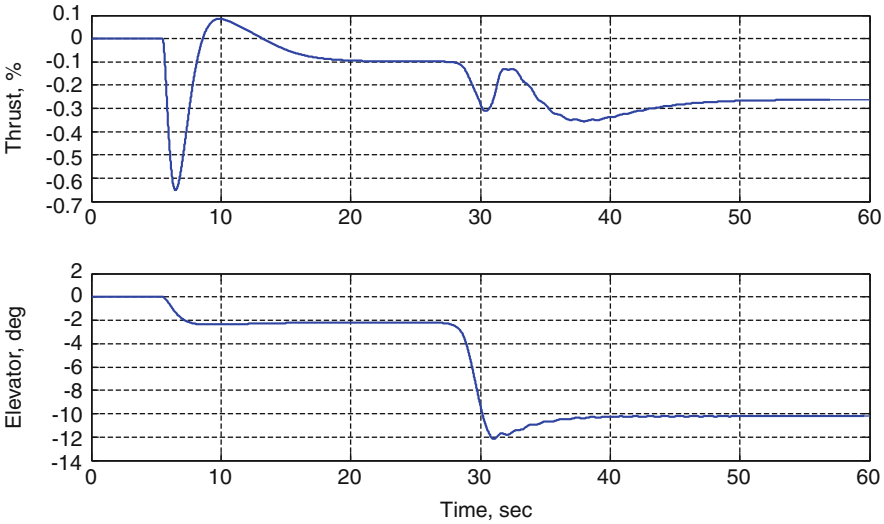


Fig. 12.15 Total control during landing with ground effect in Example 12.3

12.5 Summary

We have developed an adaptive design method to control MIMO dynamics in the presence of unstructured uncertainties, such as nonlinear state-dependent functions and bounded time-varying process noise. The resulting MRAC system represents an extension of the previously derived adaptive controllers for linear-in-parameter matched uncertainties.

Our current adaptive design includes a state limiter and a state modulation function. The state limiting logic was originally proposed in [10]. The state limiter keeps the system trajectories within predefined boundaries that define an operational envelope for the system. This is the set where we can represent the state-dependent uncertainties by linear-in-parameter RBF NNs. The state limiter is also capable of bringing the system state back into the operational envelope in finite time.

The state limiting mode is turned on or off by the state modulation function, which in turn provides a gain-scheduling feature between the command tracking and the state limiting modes of operation. In other words, the state limiter can seamlessly fade in and/or out the command tracking or the state limiting tasks, depending if the system state is located inside or outside of the operational envelope, respectively.

We have also presented a concise overview of function approximation properties using artificial NNs. This material justifies our model formulation and control design approaches for representation and attenuation of the system nonlinear-in-parameter uncertainties.

In summary, we have employed Lyapunov-based arguments and artificial NNs to attain UUB tracking performance for MIMO dynamics with both structured and unstructured uncertainties.

12.6 Exercises

Exercise 12.1. Select a scalar non-monotonic function. Use an off-line regression to approximate the selected function with sigmoidal and RBF neural networks. Increase the number of neurons and record the corresponding function errors. Plot the following data: (a) the function and the approximating NN and (b) the function approximation error versus the number of neurons. Repeat these tasks for a function of two independent variables. Comment on your results.

Exercise 12.2. Modify design equations in Table 12.1 to justify an adaptive augmentation-based design. Prove closed-loop system stability, show boundedness of all signals, and quantify tracking performance.

Exercise 12.3. Repeat the control design and all simulation steps from Example 12.3. Using the selected regressor Φ, perform off-line approximation of the ground effect-induced angle of attack function $\alpha_g(h)$. Compute the corresponding online approximation of the same function. Compare and discuss the off-line versus online approximation data. Modify the aircraft open-loop matrix A such that the ground effect matrix B_g is no longer matched. Introduce a control uncertainty $\Lambda \neq I_{m\times m}$ and add a uniformly bounded process noise $\xi(t)$. Redesign the controller (if needed) and rerun all simulation tests. Discuss robustness properties of the controller with respect to the unmatched ground effect uncertainties and the process noise.

Exercise 12.4. Similar to (12.37), derive an adaptive augmentation of a baseline proportional integral (PI) controller, (baseline PI + adaptive), with a state limiter modification similar to (12.16), for the extended open-loop system dynamics,

$$\dot{x} = A\,x + B\,\Lambda\left(u + f\left(x_p\right)\right) + B_{ref}\,y_{cmd} + \xi(t)$$

with m inputs u, m regulated outputs $y = C\,x$, m commands $y_{cmd} \in R^m$, and n uniformly bounded noise components $\xi(t) \in R^n$, where (A, B) is a controllable pair of unknown matrices, $\Lambda \in R^m$ is an unknown positive-definite diagonal matrix, and $f\left(x_p\right)$ denotes an unstructured matched state-dependent uncertainty. It is assumed that the first m components of the state vector x represent the integrated output tracking error, whose dynamics are $\dot{e}_{yI} = y - y_{cmd}$.

Exercise 12.5. Using the design equations from Exercise 12.4 and the aircraft data from Example 12.3, design and simulate a (baseline PI + adaptive) automatic landing system. Are there any advantages in using a PI baseline controller versus a feedback/feedforward system from Example 12.3? Compare and discuss your results.

References

1. Scarselli, F., Tsoi, C.: Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results. Neural Netw. **11**(1), 15–37 (1998)
2. Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J.: Neural networks for control systems – a survey. Automatica **28**(6), 1083–1112 (1992)
3. Micchelli, C.: Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr. Approx. **2**, 11–12 (1986)
4. Cybenko, G.: Approximation by superposition of a sigmoidal function. Math. Control Signals Syst. **2**, 303–314 (1983)
5. Barron, A.: Universal approximation bounds for superposition of a sigmoidal function. IEEE Trans. Info. Theory **3**, 930–945 (1993)
6. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. **3**(2), 246–257 (1991)
7. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Control. Dover, New York (2005)
8. Ioannou, P., Fidan, P.: Adaptive Control Tutorial. Advances in Design and control. SIAM, Philadelphia (2006)
9. Khalil, H.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (1996). 07458
10. Sanner, M.R., Slotine, J.-J.E.: Gaussian networks for direct adaptive control. IEEE Trans. Neural Netw. **3**(6), 837–863 (1992)
11. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation. Wiley, New York (1992)

Chapter 13
Adaptive Control with Improved Transient Dynamics

13.1 Motivation

Let us return to the original concept of the model reference adaptive control (MRAC), as it was first proposed in 1958 by Whitaker et al., at MIT [1]. The main idea was to specify the desired command-to-output performance of a servo-tracking system using a reference model that would define the ideal response of the system due to external commands. A generic block diagram of the MRAC system is shown in Fig. 13.1.

As seen from the diagram, the controller parameter adjustments (the adaptive law) are made based on the tracking error (the difference between the system actual response and its target specified by the reference model output), an output feedback from the process, and the external command.

Example 13.1 For clarity and to motivate further discussions, let us consider MRAC design equations for a scalar system shown below:

$$\begin{array}{ll} \text{Process} & : \dot{x} = a\,x + b\,u \\ \text{Ref. Model} & : \dot{x}_{ref} = a_{ref}\,x_{ref} + b_{ref}\,r \\ \text{Controller} & : u = \hat{k}_x\,x + \hat{k}_r\,r \\ \text{Adaptive Law} & : \begin{cases} \dot{\hat{k}}_x = -\gamma_x\,x\,(x - x_{ref}) \\ \dot{\hat{k}}_r = -\gamma_r\,r\,(x - x_{ref}). \end{cases} \end{array} \tag{13.1}$$

where a and b are unknown constant parameters in the process dynamics with the known $\operatorname{sgn} b > 0$. The control input u is selected such that the system state x follows the reference model state x_{ref}, driven by any bounded external command $r = r(t)$. Also in (13.1), the reference model data $a_{ref} < 0$ and b_{ref} are chosen to yield the desired speed of response and a DC gain (unity in most applications) from the reference model output $y_{ref} = x_{ref}$ to the system-regulated output $y = x$.

In this case, closed-loop system stability and global asymptotic tracking are achieved via a specific choice of the adaptive law in (13.1), with the adaptive gains

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_13,

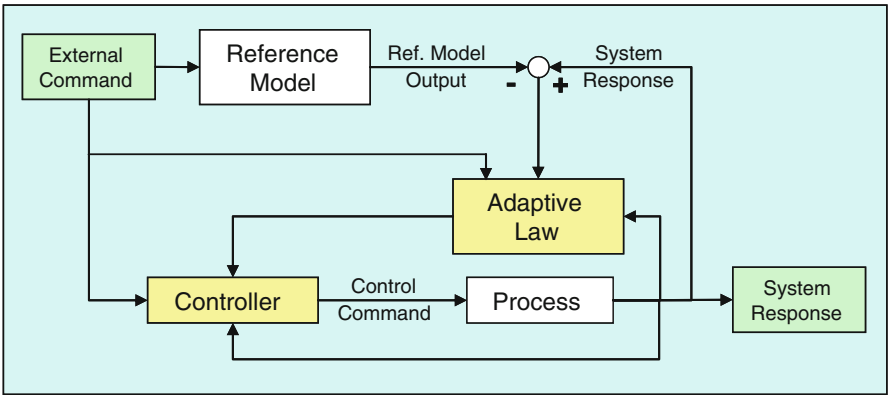


Fig. 13.1 MRAC block diagram

$(\hat{k}_x,\ \hat{k}_r)$, whose dynamics is influenced by two positive constant rates of adaptation $(\gamma_x,\ \gamma_r)$. As seen from (13.1), the state tracking error

$$e = x - x_{ref} \tag{13.2}$$

drives the adaptive laws. Existence of a servo-control solution for this particular scalar dynamics is provided by the matching conditions,

$$\begin{aligned} a_{ref} &= a + b\,k_x \\ b_{ref} &= b\,k_r \end{aligned} \tag{13.3}$$

where k_x and k_r denote the ideal unknown constant parameters (gains of the ideal controller). For scalar dynamics, such as the process in (13.1), it is clear that the matching relations (13.3) always have a solution.

Let

$$\Delta k_x = \hat{k}_x - k_x, \quad \Delta k_r = \hat{k}_r - k_r \tag{13.4}$$

represent the parameter estimation errors. Substituting the matching conditions (13.3) into (13.1), one can derive the tracking error dynamics,

$$\dot{e} = a_{ref}\,e + b\,(\Delta k_x\,x + \Delta k_r\,r) \tag{13.5}$$

which indeed define transients in the corresponding closed-loop system.

We emphasize that both the tracking error dynamics and the transient dynamics are indistinguishable. In other words, if and when e becomes small, the system output tracks the reference model with diminishing errors. On the other hand, the transient dynamics define what happens between the start of a maneuver and the time when the error gets small. We shall address this question in this chapter.

Returning to (13.5), we can employ Lyapunov arguments to prove global asymptotic stability of the tracking error dynamics. In fact, using a radially unbounded quadratic Lyapunov function candidate in the form

$$V(e,\ \Delta k_x,\ \Delta k_r) = e^2 + b\left[\frac{\Delta k_x^2}{\gamma_x} + \frac{\Delta k_r^2}{\gamma_r}\right] \tag{13.6}$$

it is not difficult to show that with the adaptive law (13.1), the time derivative of V, evaluated along the trajectories of the error dynamics (13.5), becomes nonpositive. This argument constitutes the inverse Lyapunov-based design. It provides (a) the adaptive law and (b) the required proof of closed-loop global asymptotic stability. As a result, we can formally show that for any initial condition, any bounded time-varying external command, and any positive rates of adaptation, the tracking error dynamics (13.5) are globally asymptotically stable,

$$\lim_{t\to\infty} |e(t)| = \lim_{t\to\infty} \left|x(t) - x_{ref}(t)\right| = 0 \tag{13.7}$$

and all signals in the corresponding closed-loop dynamics remain uniformly bounded, forward in time.

We immediately note that this adaptive controller solves the servo-tracking problem asymptotically in time, as t $\rightarrow \infty$, while it provides no uniformly guaranteed bounds on how large the transients might become prior to acquiring the command.

In the previous chapters, we have shown that in order to yield fast tracking and thus shorten transient times, one needs to increase the rates of adaptation $(\gamma_x,\ \gamma_r)$. However, experience shows that if these rates grow large, then unwanted transient oscillations will start to occur during the initial few seconds (the transient time) of operation. The balance between achieving fast tracking and avoiding undesired transients constitutes the MRAC design trade-off phenomenon. In essence, the rates of adaptation must be chosen large enough but not too large.

What also complicates the MRAC design tuning process is the direct dependence of the transient dynamics (13.5) on (a) the external command and (b) the initial conditions for the system and the adaptive controller. These parameters may too lead to undesirable transients.

Let us take a step back and look again at the error dynamics (13.5). We know that the time-varying signal

$$\varphi(t) = b\left(\Delta k_x(t)\ x(t) + \Delta k_r(t)\ r(t)\right) \tag{13.8}$$

is uniformly bounded and that the tracking error $e(t)$ globally asymptotically tends to zero, as shown in (13.7). Still, the time constant of the transient dynamics (13.5) $\tau_e = \frac{1}{|a_{ref}|}$ is exactly the same as the one for the reference model in (13.1).

Even though having the same time constant in both systems is theoretically correct, any control practitioner would want to have the transient dynamics (13.5)

evolve faster than the desired reference model. In other words, we want the transients to die out quickly, relative to the dynamics of the reference model trajectories. This design requirement is identical to the one that takes place during the construction of asymptotic state observers, originally developed by Luenberger in his Ph.D. thesis at the Stanford University (1963) and later published in [2]. Per Luenberger, the reference model in (13.1) represents an open-loop observer. So, just like in the closed-loop observer dynamics, we can add an error feedback term to the reference model and arrive at the observer-like reference model,

$$\dot{x}_{ref} = a_{ref}\, x_{ref} + b_{ref}\, r + \underbrace{\boxed{k_e\,(x - x_{\text{ref}})}}_{\text{Error Feedback Term}} \tag{13.9}$$

where $k_e > 0$ is the reference model feedback gain. The newly introduced error feedback term in (13.9) is equivalent to the output innovation feedback in a state observer. It is easy to see that in this case, the corresponding error dynamics become faster than the open-loop reference model from (13.1):

$$\dot{e} = \left(a_{ref} - k_e\right) e + b\left(\Delta k_x\, x + \Delta k_r\, r\right) \tag{13.10}$$

Once again, Lyapunov-based arguments can be easily repeated to prove (a) global asymptotic stability of the modified error dynamics (13.9) and (b) uniform boundedness of all signals in the related closed-loop system. For those readers who are familiar with the MRAC stability proof concept, we briefly note that using the same Lyapunov function candidate (13.6), one needs to compute its time derivative along the trajectories of (13.10), substitute the adaptive law from (13.1), and then show that the resulting time derivative is globally nonpositive. This will prove uniform boundedness of the tracking error e and of the parameter estimation errors (13.4). Furthermore, since in the observer-like reference model (13.9), $a_{ref} < 0$ and the error feedback term are bounded, then the model state x_{ref} is bounded as well. The rest of the proof follows standard (in MRAC) stability arguments, finally arriving at (13.7).

Revised block diagram with the observer-like reference model (13.9) is shown in Fig. 13.2.

Before proceeding any further, we would like to briefly present and discuss simulation comparison data for the observer-like reference model modification, while using the scalar process dynamics from (13.1) and the simulation parameters as indicated below:

$$\boxed{\begin{array}{ll} \text{Process} & : \dot{x} = x + 3\,u \\ \text{Ref. Model} & : \dot{x}_{ref} = -10\,x_{ref} + 10\,r + k_e\left(x - x_{ref}\right) \\ \text{Controller} & : u = \hat{k}_x\, x + \hat{k}_r\, r \\ \text{Adaptive Law} & : \begin{cases} \dot{\hat{k}}_x = -10\,x\left(x - x_{ref}\right) \\ \dot{\hat{k}}_r = -10\,r\left(x - x_{ref}\right). \end{cases} \end{array}} \tag{13.11}$$

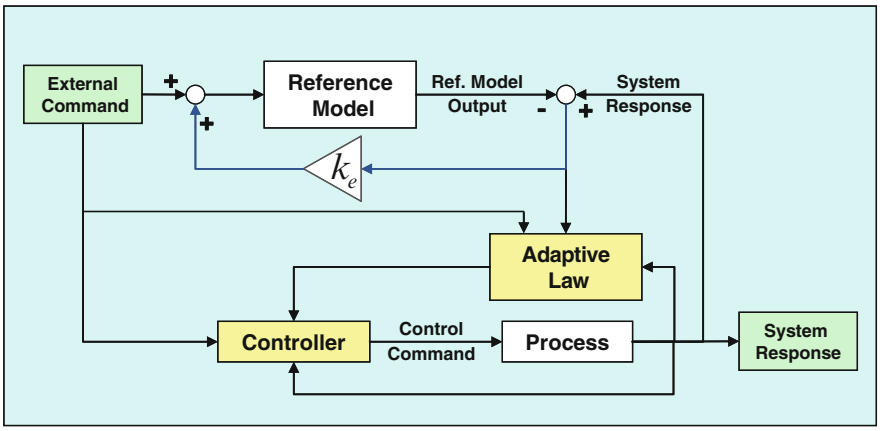


Fig. 13.2 MRAC block diagram with observer-like reference model in Example 13.1

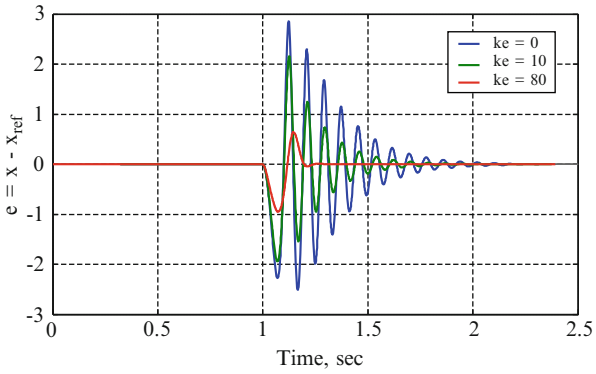


Fig. 13.3 MRAC transient dynamics due to step-input command in Example 13.1

In order to assess transient improvements, we perform three distinct simulation scenarios, where the error feedback gain k_e is set to 0 (standard MRAC case), 10, and 80. Figure 13.3 shows step-input response data for the three cases.

The original MRAC transient dynamics are quite oscillatory. As the reference model feedback gain k_e is increased, the transient dynamics become faster and the unwanted oscillations subside. Figure 13.4 presents simulation data comparison between the first and the third cases. Both the system state x and the control input u are shown. These responses were computed for a series of commanded step inputs of increased magnitude.

As seen from the simulation data, the use of the observer-like reference model (13.9) gives a predictable, scalable, and non-oscillatory (in transient) tracking performance (data shown in green).

Now, we shall pose the following question: Can the simulated transient improvements of the observer-like reference model be formally explained? We

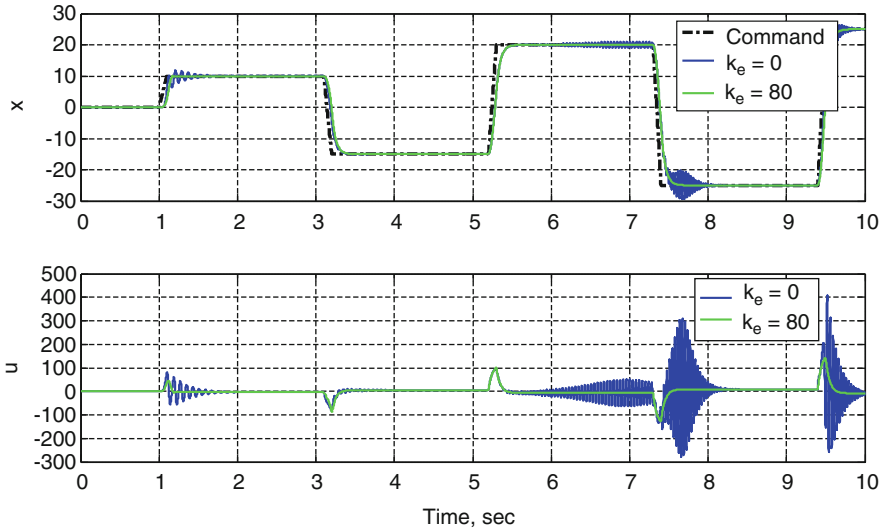


Fig. 13.4 MRAC tracking performance due to a sequence of step inputs in Example 13.1

claim that as the reference model error feedback gain k_e is increased, the system transient dynamics become less oscillatory.

In order to gain further insights into the transient behavior, we choose $k_0 > 0$, a small positive parameter ε, and redefine the reference model feedback gain:

$$k_e = \frac{k_0}{\varepsilon} \tag{13.12}$$

This allows to rewrite the modified error dynamics (13.10) in the form

$$\varepsilon \dot{e} = \left(\varepsilon a_{ref} - k_0\right) e + \varepsilon \underbrace{\left[b\left(\Delta k_x x + \Delta k_r r\right)\right]}_{\varphi(t)} \tag{13.13}$$

Since all signals in the corresponding closed-loop system are uniformly bounded, it is not difficult to find sufficient conditions so that there exists a strictly positive finite constant $0 < \varphi_{\max} < \infty$, such that for any $\varepsilon > 0$, the upper bound $|\varphi(t)| \leq \varphi_{\max}$ holds uniformly in time and ε. Furthermore, starting from an initial condition $e(0) = e_0$, the solution of (13.13) can be written explicitly:

$$e(t) = e^{\left(a_{ref} - \frac{k_0}{\varepsilon}\right) t} e(0) + \int_0^t e^{\left(a_{ref} - \frac{k_0}{\varepsilon}\right)(t-\tau)} \varphi(\tau)\, d\tau \tag{13.14}$$

We can compute an upper bound for this signal:

$$|e(t)| \leq e^{-k_0 \frac{t}{\varepsilon}} |e_0| + \frac{\varphi_{\max}}{k_0} \varepsilon \tag{13.15}$$

This relation is valid for any fixed $\varepsilon > 0$ uniformly in time. So, the system state $x(t)$ converges within $\left(\pm \frac{\varphi_{\max}}{k_0} \varepsilon \right)$ of the reference model state $x_{ref}(t)$ exponentially fast and at the rate which is no slower than $e^{-k_0 \frac{t}{\varepsilon}}$. This term gives an upper bound quantification for the decay rate of the MRAC transient dynamics, due to initial conditions mismatch, $x(0) \neq x_{\text{ref}}(0)$. Otherwise, the system transients would remain within ε-dependent bounds $\left(\pm \frac{\varphi_{\max}}{k_0} \varepsilon \right)$. Consequently, we can reduce the system transients by decreasing ε, which according to (13.12) corresponds to increasing the reference model feedback gain k_e. Being able to influence and shape the MRAC transient dynamics constitutes the essential benefit of the Luenberger-like reference model modification (13.9), (13.10), (13.11), and (13.12).

Let us give an alternative explanation for the noted transient improvements in scalar MRAC systems with observer-based reference models. The transient dynamics (13.13) can be analyzed using the singular perturbation methods [3]. Setting $\varepsilon = 0$, gives the so-called "slow" component

$$e = 0 \tag{13.16}$$

or, equivalently,

$$x = x_{ref} \tag{13.17}$$

Asymptotic stability of the slow component has already been established during Lyapunov-based proofs. Therefore, as $t \to \infty$,

$$\dot{x} = a_{ref}\, x + b_{ref}\, r + \mathrm{o}(1) \tag{13.18}$$

where the Small o-symbol o(1) denotes a function of time that asymptotically tends zero as $t \to \infty$. According to (13.18), the system state x asymptotically tracks the state of the observer-like reference model x_{ref}, with the latter asymptotically approaching the state of the original reference model from (13.1). We compute the "fast" dynamics by "stretching" time,

$$\tau = \frac{t}{\varepsilon} \tag{13.19}$$

rewrite (13.13) in the "fast" τ time scale, set $\varepsilon = 0$, and arrive at the exponentially stable fast dynamics:

$$\frac{d\, e(\tau)}{d\, \tau} = -k_0\, e(\tau) \tag{13.20}$$

It follows from (13.20) that during a finite transient time interval, the error dynamics (13.20) behave like a first-order exponentially stable system. This observation confirms our claim that for a sufficiently small ε, that is, for a large enough gain k_e, the resulting transient dynamics become smooth and approach the response of the scalar system (13.20), where k_0 defines inverse time constant of the transient system. This result can be formally summarized as follows: For a sufficiently small $\varepsilon > 0$, the state of the original system in (13.1) permits the following asymptotic expansion

$$x(t) = x_{ref}(t) + C\, e^{-\frac{k_0}{\varepsilon}t} + \mathrm{o}(1) \tag{13.21}$$

or, equivalently,

$$x(t) = x_{ref}(t) + C\, e^{-k_e t} + \mathrm{o}(1) \tag{13.22}$$

where $C > 0$ is a constant independent of k_e. The second term in (13.22) defines the transient (i.e., "fast") dynamics due to initial conditions. Consequently, with a large enough feedback gain k_e, MRAC transient dynamics can be quantified and forced to decay as fast as needed. We should immediately point out that since k_e is inversely proportional to ε, then the obvious "trade-off" in the modified design would be to avoid high gain effects in the reference model.

In the sections that follow, we will further exploit methods to analyze and enforce desired transient dynamics in adaptive control systems. But first, we present an overview of the mathematical preliminaries related to asymptotic expansions and their analysis. After that, we will generalize the observer-like reference model idea to a class of multi-input–multi-output (MIMO) dynamical systems with matched linear-in-parameter uncertainties. We shall conclude this chapter with practical observations and a summary of the derived results.

13.2 Asymptotic Orders and Singular Perturbations

Let R^n represent the Euclidean n-dimensional space, R^+ be the set of all positive real numbers, and let $R^{n\times m}$ denote the space of all n-by-m matrices, with integers n and m. For any $x \in R^n$, we write $\|x\|$ for a Euclidean vector norm of x and $\|A\|$ to be the corresponding induced matrix norm for $A \in R^{n\times m}$.

We shall use the Bachmann–Landau asymptotic order notation, denoted by the "Big O" and the "Small o" symbols [4, 5]. Given any two parameter-dependent functions (maps) $f(x;\ \varepsilon)$ and $g(x;\ \varepsilon)$, from a domain $X \subset R^n$ to another domain $Y \subset R^m$, with a scalar parameter $\varepsilon \in E \subset R$ from an interval E, we say that

$$f(x;\ \varepsilon) = \mathrm{O}(g(x;\ \varepsilon)) \tag{13.23}$$

if for each $x \in X$, there exists a positive scalar $k(x)$ such that

$$\|f(x;\ \varepsilon)\| \leq k(x)\|g(x;\ \varepsilon)\| \tag{13.24}$$

for all $\varepsilon \in E$. Choose $\varepsilon_0 \in E$ and suppose that the two limits

$$\begin{aligned} \lim_{\varepsilon \to \varepsilon_0} \|f(x;\ \varepsilon)\| &= \|f(x;\ \varepsilon_0)\| \\ \lim_{\varepsilon \to \varepsilon_0} \|g(x;\ \varepsilon)\| &= \|g(x;\ \varepsilon_0)\| \end{aligned} \tag{13.25}$$

exist. We write

$$f(x;\ \varepsilon) = \mathrm{O}(g(x;\ \varepsilon)), \quad \text{as } \varepsilon \to \varepsilon_0 \tag{13.26}$$

if for each $x \in X$, there exists a positive scalar $k(x)$ and a neighborhood/interval $N(x,\ \varepsilon_0)$ of $\varepsilon = \varepsilon_0$, such that (13.24) holds for all $\varepsilon \in N(x,\ \varepsilon_0)$. Without a loss of generality, we assume that $\varepsilon_0 = 0$. In this case, the asymptotic order relation (13.26) defines the convergence rate of $\|f(x;\ \varepsilon)\|$ to $\|f(x;\ 0)\|$, as $\varepsilon \to 0$, while holding x fixed. Specifically, for every fixed $x \in X$, $\|f(x;\ \varepsilon)\|$ converges to its limit $\|f(x;\ 0)\|$ no slower than $\|g(x;\ \varepsilon)\|$ converges to $\|g(x;\ 0)\|$, as $\varepsilon \to 0$. This convergence may hold uniformly in X, yet it could completely fail outside of X. The statement (13.23) is said to be uniformly valid in X if $k(x)$ is a finite constant independent of x. In addition, if the set $N(x,\ \varepsilon_0) = N(\varepsilon_0)$ is independent of x, then (13.26) is said to be uniformly valid in x. The relations (13.23) and (13.26) define the Big O symbol.

The Small o-symbol is defined as follows. For a given domain $X \subset R^n$, the statement

$$f(x;\ \varepsilon) = \mathrm{o}(g(x;\ \varepsilon)), \quad \text{as } \varepsilon \to 0 \tag{13.27}$$

means that for each $x \in X$ and any given $\delta > 0$, there exists an ε interval $N(x,\ \delta) = \{\varepsilon : \quad 0 < \varepsilon \leq \varepsilon_1(x,\ \delta)\}$ such that

$$\|f(x;\ \varepsilon)\| \leq \delta\, \|g(x;\ \varepsilon)\| \tag{13.28}$$

for all $\varepsilon \in N(x,\ \delta)$. We say that (13.27) is uniformly valid in X if $\varepsilon_1(x,\ \delta) = \varepsilon_1(\delta)$ is independent of x. Often, the notation $f \ll g$ is used to indicate (13.27).

The O and o symbols can be easily extended to parameter-dependent matrices $A(x;\ \varepsilon) \in R^{n \times n}$ using vector-induced matrix norms. For example, given a matrix $A_\varepsilon(x) = A(x;\ \varepsilon) \in R^{n \times n}$, the matrix asymptotic expansion

$$A_\varepsilon(x) = A_0(x) + A_1(x)\,\varepsilon + \mathrm{O}\left(\varepsilon^2\right), \quad \text{as } \varepsilon \to 0 \tag{13.29}$$

means that for every x from a domain $X \subset R^n$,

$$\lim_{\varepsilon \to 0} \|A_\varepsilon(x) - A_0(x) - A_1(x)\,\varepsilon\| = \lim_{\varepsilon \to 0} \left\|O\left(\varepsilon^2\right)\right\| = 0 \tag{13.30}$$

and the convergence rate in (13.30) is no slower than ε^2 for every fixed x. We immediately note that there is a difference between the asymptotic expansion (13.29) and, for example, the Taylor series expansion of the state-parameter-dependent matrix $A_\varepsilon(x)$. In fact, the Taylor series expansion may not even exist since differentiability of $A_\varepsilon(x)$, with respect to ε, is not assumed.

In our forthcoming derivations, we will encounter singular perturbation models [3–5]. These are dynamical systems with a small positive scale factor ε on some of the system state derivatives. For example, the transient dynamics (13.13) represent a singular perturbation model.

Generalizing (13.13), we get a singular perturbation model in the form

$$\varepsilon \dot{z} = \underbrace{(A_0 + O(\varepsilon))}_{A_\varepsilon} z + \varepsilon f(z,\, t,\, \varepsilon) \tag{13.31}$$

where $z \in R^n$ is the system state, $\varepsilon > 0$ is a constant parameter, $A_0 \in R^{n \times n}$ is Hurwitz, and

$$A_\varepsilon = (A_0 + O(\varepsilon)) \in R^{n \times n} \tag{13.32}$$

is Hurwitz, uniformly in ε. We also suppose that $f(z,\, t,\, \varepsilon) : R^n \times R^+ \times R^+ \to R^n$ is a uniformly bounded function of its arguments,

$$f(z,\, t,\, \varepsilon) = O(1) \tag{13.33}$$

or, equivalently,

$$\|f(z,\, t,\, \varepsilon)\| \leq f_{\max} < \infty \tag{13.34}$$

uniformly in $(z,\, t,\, \varepsilon)$, where $f_{\max}$ is a constant finite upper bound of the norm of f. In addition, we assume that $f(z,\, t,\, \varepsilon)$ is Lipschitz continuous in z and piecewise continuous in $(t,\, \varepsilon)$. It is not difficult to show that all of the above-stated assumptions assure existence and uniqueness of the system solutions, starting at any set of initial conditions $z_0 = z(0)$.

It is possible to show that for a sufficiently small ε, all trajectories of (13.31) converge to an $O(\varepsilon)$ neighborhood of the origin, exponentially fast. This fact is stated next.

Theorem 13.1. *Consider the singularly perturbed* n*-dimensional dynamics (13.31),*

$$\varepsilon \dot{z} = \underbrace{(A_0 + O(\varepsilon))}_{A_\varepsilon} z + \varepsilon f(z,\, t,\, \varepsilon)$$

where $\varepsilon > 0$ is a constant, $A_0 \in R^{n\times n}$ is a constant Hurwitz matrix, and $A_\varepsilon \in R^{n\times n}$ is a Hurwitz (uniformly in ε) matrix. Suppose that $f(z,\ t,\ \varepsilon) \in R^n$ is a uniformly bounded vector function, Lipschitz continuous in z*, and piecewise continuous in* t *and* ε*. Then, there exists a strictly positive constant $\gamma > 0$, independent of ε, such that the asymptotic relation,*

$$z(t) = \mathrm{O}\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + \mathrm{O}(\varepsilon), \quad (\varepsilon \to 0) \tag{13.35}$$

holds for all $t \geq 0$. If in addition to being uniformly bounded, the function $f(z,\ t,\ \varepsilon)$ asymptotically decays to zero in time (uniformly in z*), then*

$$z(t) = \mathrm{O}\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + \mathrm{O}(\varepsilon)\,\mathrm{o}(1), \quad (\varepsilon \to 0) \tag{13.36}$$

for all $t \geq 0$, where $\mathrm{o}(1) \underset{t\to\infty}{\to} 0$ is an asymptotically decaying time function. ■

Proof of Theorem 13.1 Since A_ε is Hurwitz uniformly in ε and

$$A_\varepsilon \frac{t}{\varepsilon} = (A_0 + \mathrm{O}(\varepsilon))\ \frac{t}{\varepsilon} = A_0\ \frac{t}{\varepsilon} + \mathrm{O}(1) \tag{13.37}$$

then following the proof arguments from [3, Lemma 9.9, pp. 369–371], we can claim existence of two strictly positive constants, k and γ, such that for a sufficiently small $\varepsilon > 0$, the induced 2-norm of the exponential matrix $e^{A_\varepsilon \frac{t}{\varepsilon}}$ satisfies

$$\left\| e^{A_\varepsilon \frac{t}{\varepsilon}} \right\| = \left\| e^{(A_0 + \mathrm{O}(\varepsilon))\frac{t}{\varepsilon}} \right\| \leq k\, e^{-\gamma \frac{t}{\varepsilon}} \tag{13.38}$$

where $(k,\ \gamma) > 0$ are independent of ε. This fact merely states that the 2-norm of a parameter-dependent Hurwitz matrix exponentially decays to zero, if the parameter is selected small enough. In terms of the asymptotic order notation, (13.38) implies

$$\left\| e^{A_\varepsilon \frac{t}{\varepsilon}} \right\| = \left\| e^{(A_0 + \mathrm{O}(\varepsilon))\frac{t}{\varepsilon}} \right\| = \mathrm{O}\left(e^{-\gamma \frac{t}{\varepsilon}}\right), \quad (\varepsilon \to 0) \tag{13.39}$$

for all $t \geq 0$.

Because of its specific form, the singular perturbation dynamics (13.31) can be analyzed directly by explicitly writing the system solution:

$$z(t) = e^{A_\varepsilon \frac{t}{\varepsilon}} z_0 + \int_0^t e^{A_\varepsilon \frac{(t-\tau)}{\varepsilon}} f(z(\tau),\ \tau,\ \varepsilon)\, d\tau \tag{13.40}$$

With the help of (13.39), we can easily derive an upper bound for the norm of the system solution (13.40):

$$\|z(t)\| \le \left\|e^{A_\varepsilon \frac{t}{\varepsilon}}\right\| \|z_0\| + f_{\max} \int_0^t \left\|e^{A_\varepsilon \frac{(t-\tau)}{\varepsilon}}\right\| \mathrm{d}\tau \le k \, \|z_0\| \, e^{-\gamma \frac{t}{\varepsilon}} + \left(\frac{k f_{\max}}{\gamma}\right) \varepsilon \quad (13.41)$$

Thus, we have proven the asymptotics (13.35),

$$z(t) = \mathrm{O}\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + \mathrm{O}(\varepsilon), \quad (\varepsilon \to 0)$$

for the system solutions (13.40), evolving on an infinite time interval.

In the context of singular perturbations, this relation implies that for a sufficiently small fixed $\varepsilon > 0$, all solutions of (13.31) converge to an ε neighborhood of the origin exponentially fast at the rate of no slower than $e^{-\gamma \frac{t}{\varepsilon}}$. The first term in (13.35) describes the "fast" (transient) dynamics of the system solutions, as they approach an ε neighborhood of the origin. Also in this case, $\frac{t}{\varepsilon}$ can be interpreted as the "stretched" time, which allows us to look at the system transients through a "magnifying time glass," so to speak.

Suppose that for a constant $\varepsilon > 0$,

$$f(z, t, \varepsilon) = \mathrm{o}(1) \underset{t \to \infty}{\to} 0 \quad (13.42)$$

uniformly in z. Repeating the previous arguments that have led us to (13.35), one can derive (13.36) (see Exercise 13.2),

$$z(t) = \mathrm{O}\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + \mathrm{O}(\varepsilon)\,\mathrm{o}(1), \quad (\varepsilon \to 0)$$

for all $t \ge 0$. So in this case, the system trajectories converge to a neighborhood of the origin exponentially fast, and after that, the solutions continue to asymptotically converge to the origin but at perhaps a much slower rate. The theorem proof is complete. □

Observe that setting $\varepsilon = 0$ reduces the differential Eq. (13.31) to an algebraic relation $z = 0$. This is the singularity phenomenon, whereby the origin becomes the "slow" manifold of the system. Overall, we have decomposed the system trajectories into "fast" and "slow" components, with the former describing the rate of convergence to the latter.

Let us mention that the asymptotic behavior (13.35) could have also been derived using the singular perturbation methods [3–5] for trajectory analysis of ordinary differential equations, such as the one in (13.31). We have decided to perform a direct analysis of the system trajectories (instead of using the singular perturbation techniques), only because for the system at hand, we could explicitly write solutions and estimate their norm upper bounds.

Later on in this chapter, we are going to utilize the two asymptotic relations (13.35) and (13.36) to aid in the design of adaptive output feedback controllers.

13.3 Asymptotic Properties of the Algebraic Riccati Equation

In our forthcoming design and analysis of MRAC transient dynamics, we will encounter parameter-dependent n-dimensional algebraic Riccati equations (ARE) in the form

$$P_v A + A^T P_v - P_v B\, R_v^{-1} B^T P_v + Q_v = 0 \tag{13.43}$$

where $v > 0$ is a constant parameter (A, B) and (A, C) are controllable and observable pairs of matrices, with $A \in R^{n\times n}$, $B \in R^{n\times m}$, $C \in R^{m\times n}$, and $m \leq n$. The ARE weight matrices are defined as

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) C^T C, \qquad R_v = \left(\frac{v}{v+1}\right) R_0 \tag{13.44}$$

where $Q_0 \in R^{n\times n}$ and $R_0 \in R^{m\times m}$ are both symmetric and strictly positive definite. This formulation appears in [6].

The well-known fact from optimal control of linear systems (with quadratic cost index) states that for any $v > 0$, the ARE (13.43) has the unique symmetric positive-definite solution $P_v > 0$. This ARE arises in the optimal linear quadratic regulator (LQR) control problems for linear time-invariant dynamics,

$$\dot{x} = A\,x + B\,u, \quad y = C\,x \tag{13.45}$$

with a quadratic minimization criterion in the form

$$\begin{aligned} J_v &= \int_0^\infty \left(x^T Q_v\, x + u^T R_v\, u\right) dt = \int_0^\infty \left(x^T \left(Q_0 + \frac{1}{\rho_v} C^T C\right) x + \rho_v\, u^T R_0\, u\right) dt \\ &= \int_0^\infty \left(x^T Q_0\, x + \frac{1}{\rho_v} y^T y + \rho_v\, u^T R_0\, u\right) dt \end{aligned} \tag{13.46}$$

where

$$\rho_v = \frac{v}{v+1} \tag{13.47}$$

is a positive constant.

Let us remark that the main difference of (13.46), from a typical cost function considered in classical textbooks on optimal control, is the presence of the second term, which is inversely proportional to ρ_v. This expression "punishes" the system output, as the system input is allowed to become large with ρ_v getting small. Substituting (13.44) into (13.43) gives

$$P_v A + A^T P_v - \left(1 + \frac{1}{v}\right) P_v B\, R_0^{-1} B^T P_v + Q_0 + \left(1 + \frac{1}{v}\right) C^T C = 0 \qquad (13.48)$$

or, equivalently,

$$P_v A + A^T P_v - P_v B R_0^{-1} B^T P_v + Q_0 + C^T C + \frac{1}{v}\left[C^T C - P_v B R_0^{-1} B^T P_v\right] = 0 \quad (13.49)$$

We are interested in analyzing asymptotic properties of the ARE solution P_v, as $v \to 0$. Hence, let us consider the following asymptotic expansion:

$$P_v = P_0 + P_1 v + \mathrm{O}\left(v^2\right), \quad \text{as } v \to 0 \qquad (13.50)$$

Similar to (13.29) and (13.30), the Big O symbol $\mathrm{O}(v^2)$ in (13.50) denotes a v-dependent $(n \times n)$ matrix, whose induced norm tends to zero no slower than v^2, as $v \to 0$, that is,

$$\lim_{v \to 0} \|P_v - P_0 - P_1 v\| = \lim_{v \to 0} \left\|\mathrm{O}\left(v^2\right)\right\| = 0 \qquad (13.51)$$

For matrices satisfying (13.50), we can also write

$$P_0 = \lim_{v \to 0} P_v$$

which means $\lim_{v \to 0} \|P_v - P_0\| = 0$, that is, limits of parameter-dependent matrices are understood in terms of their induced norms. Before proceeding any further, we need to introduce a square root of a matrix according to [7, p. 245].

Definition 13.1. *An* $(n \times n)$ *matrix* $S = P^{\frac{1}{2}} = \sqrt{P}$ *is called a square root of a symmetric positive-definite* $(n \times n)$ *matrix* P, *if* $P = S^T S$.

It is not so difficult to see that matrix square roots are by no means unique. However, we can define the unique square root by taking S to be symmetric.

Let us now state and prove several interesting asymptotic properties of a parameter-dependent ARE in the form of (13.49).

Theorem 13.2. *Consider the ARE (13.43) with any two controllable and observable matrix pairs,* (A, B) *and* (A, C), *and with the two symmetric positive-definite matrices* Q_v *and* R_v *from (13.44). Let* $A \in R^{n \times n}, B \in R^{m \times n}, C \in R^{p \times n}, Q_v \in R^{n \times n}$, *and*

$R_v \in R^{m\times m}$, where n, p, *and* m *are integers. Then, the ARE has the unique symmetric positive-definite solution P_v.*

Moreover, if $p = m$, $\det(CB) \neq 0$, *and the transfer function* $G(s) = C\,(s\,I_{n\times n} - A)^{-1}\,B$ *is minimum-phase, then the ARE solution* $\mathrm{P_v}$ *can be represented by the asymptotic expansion (13.50), while the following statements hold true:*

1. *P_0 and P_1 are symmetric.*
2. *P_0 is the unique symmetric strictly positive-definite solution of the following algebraic Lyapunov equation:*

$$P_0\left(A - B\,R_0^{-1}\,B^T\,P_1\right) + \left(A - B\,R_0^{-1}\,B^T\,P_1\right)^T P_0 + Q_0 = 0 \tag{13.52}$$

3. *There exists a unitary matrix $W \in R^{m\times m}$ such that*

$$P_0\,B = C^T\,W^T\sqrt{R_0} \tag{13.53}$$

4. *The unitary matrix W in (13.53) can be chosen as*

$$W = (U\,V)^T \tag{13.54}$$

where U and V are two unitary matrices, defined by the singular value decomposition,

$$C\,B\,R_0^{-\frac{1}{2}} = U\,\Lambda\,V \tag{13.55}$$

and Λ represents the diagonal matrix of the corresponding singular values.
5. *P_v is invertible for any $v \geq 0$ and for any unit vector $x \in R^n$,*

$$\lim_{v\to 0} x^T P_v\,x \geq \lambda_{\min}(P_0) > 0 \tag{13.56}$$

where $\lambda_{\min}(P_0)$ denotes the minimum eigenvalue of P_0.
6. *The following asymptotic relation holds*

$$P_v\,B = C^T\,W^T\sqrt{R_0} + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.57}$$

Before proving the theorem, an immediate remark is in order. Relations (13.52) and (13.53) imply that the transfer function

$$\begin{aligned} G_0(s) &= B^T\,P_0\left(s\,I_{n\times n} - A + B\,R_0^{-1}\,B^T\,P_1\right)^{-1}B \\ &= \sqrt{R_0}\,W\,C\left(s\,I_{n\times n} - A + B\,R_0^{-1}\,B^T\,P_1\right)^{-1}B \end{aligned}$$

becomes strictly positive real (SPR) [3] via feedback $u = -R_0^{-1} B^T P_1 x$, when the latter is applied to the linear dynamics (13.45). At the same time, the asymptotic expansions (13.50) and (13.57) mean that the transfer function,

$$G_v(s) = B^T P_v \left(s I_{n\times n} - A + B R_0^{-1} B^T P_v\right)^{-1} B \tag{13.58}$$

which is SPR by the design, approaches the transfer function,

$$G_y(s) = \sqrt{R_0}\, W C \left(s I_{n\times n} - A + B R_0^{-1} B^T P_0\right)^{-1} B \tag{13.59}$$

that is,

$$G_v(s) = G_y(s) + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.60}$$

uniformly in s.

Proof of Theorem 13.2. Existence and uniqueness of P_v is the well-known fact. We proceed by showing that matrices P_0 and P_1 in (13.50) are symmetric. Using (13.50) gives

$$P_0 = \lim_{v\to 0} P_v = \lim_{v\to 0} P_v^T = P_0^T \tag{13.61}$$

Consequently,

$$P_1 = \lim_{v\to 0} \frac{1}{v}(P_v - P_0) = \lim_{v\to 0} \frac{1}{v}\left(P_v^T - P_0^T\right) = P_1^T \tag{13.62}$$

Next, we substitute (13.50) into (13.49):

$$\begin{aligned} & v\left[\left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right) A + A^T \left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right)\right] \\ & - v\left[\left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right) B R_0^{-1} B^T \left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right) + Q_0 + C^T C\right] \\ & + C^T C - \left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right) B R_0^{-1} B^T \left(P_0 + P_1 v + \mathrm{O}\left(v^2\right)\right) = 0 \end{aligned} \tag{13.63}$$

Collecting the zero-order terms in v gives

$$C^T C - P_0 B R_0^{-1} B^T P_0 = 0 \tag{13.64}$$

The matrix solution (P_0 B) of (13.64) may be expressed as in (13.53), whose validity can be verified by its direct substitution into (13.64).

Collecting the first-order terms in v from (13.63) gives

$$P_0 \underbrace{\left(A - B\,R_0^{-1}\,B^T\,P_1\right)}_{\tilde{A}} + \left(A - B\,R_0^{-1}\,B^T\,P_1\right)^T P_0 - P_0\,B\,R_0^{-1}\,B^T\,P_0 + \underbrace{\left(Q_0 + C^T\,C\right)}_{\tilde{Q}} = 0 \tag{13.65}$$

or, equivalently,

$$P_0\,\tilde{A} + \tilde{A}^T\,P_0 - P_0\,B\,R_0^{-1}\,B^T\,P_0 + \tilde{Q} = 0 \tag{13.66}$$

Since a feedback connection, such as $u = -R_0^{-1}\,B^T\,P_1\,x$, does not change the controllability of (A, B), then $\left(\tilde{A},\, B\right)$ is also controllable. Moreover, since $\tilde{Q} = \tilde{Q}^T > 0$, then the ARE (13.66) has the unique symmetric positive-definite solution $P_0 = P_0^T > 0$. Finally, using (13.64) in (13.65) gives (13.52) and thus proves the second claim of the theorem.

Choosing the unitary matrix W as in (13.54), while using (13.55), results in

$$\begin{aligned} B^T P_0\,B &= B^T\,C^T\,W^T\sqrt{R_0} = \sqrt{R_0}\left(R^{-\frac{1}{2}}B^T\,C^T\right)W^T\sqrt{R_0} \\ &= \sqrt{R_0}\,V^T\,\Lambda\,\underbrace{U^T\,U}_{I_{m\times m}}\,V\,\sqrt{R_0} = \sqrt{R_0}\left(V^T\,\Lambda\,V\right)\sqrt{R_0} > 0 \end{aligned} \tag{13.67}$$

Note that this particular choice of W supports the established positive-definiteness property of P_0.

Let us select a unit vector $x \in R^n$. Then,

$$\lim_{v\to 0} x^T P_v\,x = \lim_{v\to 0} x^T\left[P_0 + \mathrm{O}(v)\right]x = x^T P_0\,x \geq \lambda_{\min}(P_0) > 0 \tag{13.68}$$

We know that the ARE solution P_v is invertible for any fixed $v > 0$. Also, from (13.68), it follows that for a sufficiently small $v \geq 0$, the eigenvalues of P_v are bounded away from zero. Therefore, P_v is invertible globally and for any $v \geq 0$. Finally, we note that (13.57) is a direct consequence of (13.50) and (13.53). The proof of the theorem is complete. □

Let us now make the following substitutions into the ARE (13.43):

$$A := A^T, \quad B := C^T \tag{13.69}$$

The resulting equation becomes

$$P_v\,A^T + A\,P_v - P_v\,C^T\,R_v^{-1}\,C\,P_v + Q_v = 0 \tag{13.70}$$

where according to (13.44),

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) B\,B^T, \quad R_v = \frac{v}{v+1}\,R_0 \tag{13.71}$$

The reader may have noticed that such an ARE arises in the design of Kalman filters and Luenberger observers. Substituting (13.71) into (13.70) gives

$$P_v\,A^T + A\,P_v - \left(1+\frac{1}{v}\right) P_v\,C^T\,R_0^{-1}\,C\,P_v + Q_0 + \left(1+\frac{1}{v}\right) B\,B^T = 0 \tag{13.72}$$

or, equivalently,

$$P_v\,A^T + A\,P_v - P_v\,C^T\,R_0^{-1}\,C\,P_v + Q_0 + B\,B^T + \frac{1}{v}\left[B\,B^T - P_v\,C^T\,R_0^{-1}\,C\,P_v\right] = 0 \tag{13.73}$$

For the parameter-dependent ARE in (13.73), all statements from the Theorem 13.2 can be easily reformulated. These claims are summarized (without proofs) below.

Corollary 13.1. *Suppose that all assumptions from Theorem 13.2 hold. Then, the unique positive-definite solution* $\mathrm{P_v}$ *of the ARE (13.70), with the weight matrices* $\mathrm{Q_v}$ *and* $\mathrm{R_v}$ *from (13.71), can be represented by the asymptotic expansion (13.50). Moreover, the following statements hold:*

1. *P_0 and P_1 are symmetric.*
2. *P_0 is the unique symmetric strictly positive-definite solution of the following algebraic Lyapunov equation:*

$$P_0\left(A - C^T\,R_0^{-1}\,C\,P_1\right)^T + \left(A - C^T\,R_0^{-1}\,C\,P_1\right)P_0 + Q_0 = 0 \tag{13.74}$$

3. *There exists a unitary matrix $W \in R^{m\times m}$ such that*

$$P_0\,C^T = B\;W^T\sqrt{R_0} \tag{13.75}$$

4. *The unitary matrix* W *in (13.53) can be chosen as*

$$W = (U\,V)^T \tag{13.76}$$

where U and V are two unitary matrices, defined by the singular value decomposition,

$$B^T\,C^T\,R_0^{-\frac{1}{2}} = U\,\Lambda\,V \tag{13.77}$$

and Λ *represents the diagonal matrix of the corresponding singular values.*
5. P_v *is invertible for any* $v \geq 0$*, and*

$$\lim_{v \to 0} x^T P_v x \geq \lambda_{\min}(P_0) > 0 \tag{13.78}$$

where $\lambda_{\min}(P_0)$ *denotes the minimum eigenvalue of* P_0*.*
6. The following asymptotic relation holds:

$$P_v C^T = B\ W^T \sqrt{R_0} + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.79}$$

Soon in this chapter, we shall use the above statements in our design of MRAC controllers with smooth transient dynamics, but at this moment, let us make the following remark: Since $\mathrm{P_v}$ is invertible for any $v \geq 0$, one can define the matrix inverse,

$$\tilde{P}_v = P_v^{-1} \tag{13.80}$$

and analyze its property using an asymptotic expansion in the form

$$\tilde{P}_v = \tilde{P}_0 + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.81}$$

Substituting (13.81) into $\tilde{P}_v P_v = I_{n\times n}$ gives

$$I_{n\times n} = \tilde{P}_v P_v = \left(\tilde{P}_0 + \mathrm{O}(v)\right)(P_0 + \mathrm{O}(v)) = \tilde{P}_0 P_0 + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.82}$$

Consequently,

$$I_{n\times n} = \lim_{v \to 0} \tilde{P}_v P_v = \tilde{P}_0 P_0 \tag{13.83}$$

and therefore,

$$\left\{\left[\tilde{P}_0 = P_0^{-1}\right] \Rightarrow \left[P_v^{-1} = P_0^{-1} + \mathrm{O}(v)\right]\right\}, \quad \text{as } v \to 0 \tag{13.84}$$

Using (13.84) and (13.79), yields

$$\begin{aligned} C^T &= \tilde{P}_v \left(B\ W^T \sqrt{R_0} + \mathrm{O}(v)\right) = \tilde{P}_v B\ W^T \sqrt{R_0} + \left(P_0^{-1} + \mathrm{O}(v)\right)\mathrm{O}(v) \\ &= \tilde{P}_v B\ W^T \sqrt{R_0} + \mathrm{O}(v) \end{aligned} \tag{13.85}$$

and as a result, we obtain the asymptotic relation,

$$\tilde{P}_v B = C^T R_0^{-\frac{1}{2}} W + \mathrm{O}(v) \tag{13.86}$$

which we shall employ in the design of adaptive output feedback controllers in Chap. 14. This concludes our asymptotic analysis of parameter-dependent ARE solutions.

13.4 System Dynamics and Control Problem Formulation

We are going to design an MRAC controller, with an observer-like reference dynamics, for a class of nonlinear MIMO uncertain dynamical systems in the form

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{m\times m} & C_p \\ 0_{n_p\times m} & A_p \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0_{m\times m} \\ B_p \end{pmatrix}}_{B} \Lambda \left(u + \overbrace{\Theta_d^T \Phi_d(x_p)}^{d(x_p)} \right)$$

$$+ \underbrace{\begin{pmatrix} -I_{m\times m} \\ 0_{n_p\times m} \end{pmatrix}}_{B_{ref}} y_{cmd} \; y = \underbrace{\begin{pmatrix} 0_{m\times m} & C_p \end{pmatrix}}_{C} x \tag{13.87}$$

The dynamics (13.87) incorporate an n_p-dimensional open-loop system with m control inputs u and m regulated outputs y. This is the original plant, whose state is $x_p \in R^{n_p}$. The plant is augmented by the m-dimensional integrated output tracking error dynamics, $\dot{e}_{yI} = C_p x_p - y_{\text{cmd}}$, where $C_p \in R^{m\times n_p}$ is a known constant matrix. The order of the complete system (13.87) is $n = n_p + m$. In addition, $x \in R^n$ is the system state vector, $u \in R^m$ is the control input, $y \in R^p$ is the regulated output, $y_{\text{cmd}} \in R^m$ is the commanded signal for y to follow, $d(x_p) = \Theta_d^T \Phi_d(x_p) \in R^m$ is a nonlinear state-dependent matched parametric uncertainty, $\Theta_d \in R^{N\times m}$ is the matrix of unknown constant "true" parameters, and $\Phi_d(x_p) \in R^N$ is the known N-dimensional regressor vector, whose components are locally Lipschitz continuous in x, that is, there exists a finite positive known constant $0 < L_{\Phi_d} < \infty$, such that for any $(x_1,\ x_2) \in R^{n_p}$ from a bounded neighborhood of the origin, the following inequality holds:

$$\|\Phi_d(x_1) - \Phi_d(x_2)\| \le L_{\Phi_d} \|x_1 - x_2\| \tag{13.88}$$

Also in (13.87), $A \in R^{n\times n}$, $B \in R^{n\times m}$, $B_{ref} \in R^{n\times m}$, and $C \in R^{m\times n}$ are constant known matrices, while $\Lambda \in R^{m\times m}$ is a constant diagonal unknown matrix with strictly positive diagonal elements.

Our choice of the process dynamics (13.87) is largely motivated by aerospace applications, where x_p models six degrees of freedom of an airborne platform and $d(x_p)$ represents uncertainties in the vehicle aerodynamic moments. By definition, the moment uncertainties appear together with the system control inputs, thus enforcing the matching conditions needed to justify mere existence of a control

solution. Moreover, control actuator uncertainties, control effectiveness reduction, and other control failures are modeled by an unknown constant matrix Λ. Finally, inclusion of the integrated output tracking error $\dot{e}_{yI} = C_p x_p - y_{\text{cmd}}$ into the open-loop system leads to the extended system formulation (13.87). This inclusion is optional, yet it allows the designer to explicitly account for baseline controllers with integral feedback, and it also allows to avoid feedforward terms in a control solution. Other dynamics, such as structural notch filters, sensors, and actuators, can also be added in the formulation of the extended open-loop system.

In order to control a dynamical system such as (13.87), we need the nominal system (no uncertainties) to be controllable.

Assumption 13.1 The nominal system matrix pair $(A_p,\ B_p)$ is controllable.

It is well known that controllability of $(A_p,\ B_p)$, coupled with the rank condition,

$$\text{rank}\begin{pmatrix} A_p & B_p \\ C_p & 0_{p\times m} \end{pmatrix} = n_p + m = n \tag{13.89}$$

ensures controllability of the extended pair (A, B).

Disregarding the system uncertainties, we form the ideal reference model dynamics,

$$\dot{x}_{ref\ ideal} = A_{ref}\, x_{ref\ ideal} + B_{ref}\, y_{cmd} \tag{13.90}$$

where

$$A_{ref} = A - B\ \underbrace{\left(R_{ref}^{-1} B^T P_{ref}\right)}_{K_{lqr}^T} \tag{13.91}$$

is Hurwitz, K_{lqr} is the baseline linear quadratic regulator (LQR) feedback gain, P_{ref} is the unique symmetric positive-definite solution of the ARE,

$$P_{ref} A + A^T P_{ref} - P_{ref} B R_{ref}^{-1} B^T P_{ref} + Q_{ref} = 0 \tag{13.92}$$

and (Q_{ref}, R_{ref}) are some appropriately chosen symmetric positive-definite matrices. Using the LQR design is not a requirement here. This is simply our preferred way to formulate ideal reference models and embed basic performance specifications into the system. Due to the inclusion of the integrated tracking error in (13.87), the DC gain of the reference model (13.90) is unity. Consequently, if $\Lambda = I_{m\times m}$ and $d(x) = 0_{m\times 1}$, then the LQR linear state feedback control $u_{lqr} = -K_{lqr}^T x$ enforces global exponential stability of the ideal reference model (13.90) and makes the regulated output y(t) track any bounded command $y_{cmd}(t)$ with bounded errors. Note that for a step-input command, the LQR controller provides global exponential tracking with zero steady-state errors. Also, it is easy to see that such a choice of the reference model enforces the model matching conditions stated below.

Assumption 13.2 Model Matching Conditions. Given a Hurwitz matrix A_{ref} and an unknown constant positive-definite diagonal matrix Λ, there exists a constant possibly unknown gain matrix K_x such that

$$A_{ref} = A - B\,\Lambda\,K_x^T \tag{13.93}$$

We shall note that existence of K_x is guaranteed for any controllable pair (A, B) and any nonsingular matrix Λ. In particular, relations (13.91) and (13.93) imply

$$K_x = K_{lqr}\,\Lambda^{-1} \tag{13.94}$$

Using (13.93), we rewrite the system dynamics (13.87) in the form

$$\dot{x} = A_{ref}\,x + B\,\Lambda\left(u + \underbrace{\left[K_x^T\,x + \Theta_d^T\,\Phi_d\left(x_p\right)\right]}_{\underbrace{\left(K_x^T \quad \Theta_d^T\right)}_{\Theta^T}\,\underbrace{\begin{pmatrix} x \\ \Phi_d\left(x_p\right)\end{pmatrix}}_{\Phi(x)}}\right) + B_{ref}\,y_{cmd} \tag{13.95}$$

and get

$$\dot{x} = A_{ref}\,x + B\,\Lambda\left(u + \Theta^T\,\Phi(x)\right) + B_{ref}\,y_{cmd} \tag{13.96}$$

The control goal of interest is bounded tracking of y_{cmd} in the presence of the system parametric uncertainties $\{\Lambda,\ \Theta\}$. Specifically, we need to find a control input u such that the regulated output $y = C\,x \in R^m$ tracks any bounded time-varying command $y_{cmd}(t) \in R^m$ with bounded errors, while the rest of the signals in the corresponding closed-loop system remain bounded. In addition, we shall require smooth and quantifiable transient characteristics in the closed-loop dynamics.

13.5 Observer-Like Model Reference Adaptive Control

Similar to (13.9) and for the system dynamics (13.96), we consider a Luenberger-like reference model in the form

$$\dot{x}_{ref} = A_{ref}\,x_{ref} + \underbrace{\boxed{L_y\left(x - x_{ref}\right)}}_{\text{Error Feedback Term}} + B_{ref}\,y_{cmd} \tag{13.97}$$

where $x_{ref} \in R^n$ is the reference model state and $L_v \in R^{n\times n}$ is the error feedback gain, parameterized by a positive scalar $v > 0$ (to be defined).

The system control input u is selected as

$$u = -\hat{\Theta}^T \Phi(x) \tag{13.98}$$

Substituting (13.98) into the system dynamics (13.96) gives

$$\dot{x} = A_{ref}\, x - B\,\Lambda \underbrace{\left(\hat{\Theta} - \Theta\right)^T}_{\Delta\Theta} \Phi(x) + B_{ref}\, y_{cmd} \tag{13.99}$$

where $\Delta\Theta \in R^{N\times m}$ denotes the matrix of parameter estimation errors.

In what follows, we are going to select $\left(L_v,\ \hat{\Theta}\right)$ such that the system state x globally asymptotically tracks x_{ref} – the state of the observer-like reference model (13.97) – and so $y \underset{t\to\infty}{\to} y_{ref}$. Also, we will show that x_{ref} tracks $x_{ref\,ideal}$, which in turn implies that $y_{ref} \underset{t\to\infty}{\to} y_{ref\,ideal}$. Furthermore, since the output of the ideal reference model (13.90) follows its command, $y_{ref\,ideal} \to y_{cmd}$, with bounded errors, and $y \underset{t\to\infty}{\to} y_{ref} \underset{t\to\infty}{\to} y_{ref\,ideal}$, then the system-regulated output y will also track y_{cmd} with bounded errors. This argument constitutes our design strategy.

We begin by choosing adaptive laws for $\hat{\Theta}$, so that x globally asymptotically tracks x_{ref}, in the presence of the system uncertainties. Let

$$e = x - x_{ref} \tag{13.100}$$

denote the state tracking error. Subtracting (13.97) from (13.99) gives the system transient dynamics:

$$\dot{e} = \left(A_{ref} - L_v\right)e - B\Lambda\, \Delta\Theta^T\, \Phi(x) \tag{13.101}$$

We choose the error feedback gain L_v as

$$L_v = P_v\, R_v^{-1} \tag{13.102}$$

where $P_v = P_v^T > 0$ is the unique solution of the following ARE:

$$P_v\, A_{ref}^T + A_{ref}\, P_v - P_v\, R_v^{-1}\, P_v + Q_v = 0 \tag{13.103}$$

with the ARE weight matrices $(Q_v,\ R_v)$selected as

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) I_{n\times n}, \quad R_v = \frac{v}{v+1}\, I_{n\times n} \tag{13.104}$$

using a constant parameter $v > 0$. This constant will eventually become our design "tuning knob": Small values of v will yield better MRAC transients. However, the corresponding feedback gain L_v will increase at the rate of $\frac{1}{v}$. In fact, we will show that as v tends to zero, the error feedback gain tends to infinity,

$$L_v = \left(1 + \frac{1}{v}\right) P_v = \mathrm{O}\left(\frac{1}{v}\right) \tag{13.105}$$

while the solution P_v of the ARE (13.103) tends to a constant positive-definite symmetric matrix P_0. It is easy to verify that the ARE (13.103) possesses the unique symmetric positive-definite solution P_v. Furthermore, because of (13.103), the observer closed-loop matrix,

$$A_v = A_{ref} - L_v = A_{ref} - P_v R_v^{-1} = A_{ref} - P_v \left(1 + \frac{1}{v}\right) \tag{13.106}$$

satisfies

$$\underbrace{P_v \left(A_{ref} - \underbrace{P_v R_v^{-1}}_{L_v}\right)^T}_{A_v} + \underbrace{\left(A_{ref} - \underbrace{P_v R_v^{-1}}_{L_v}\right)}_{A_v} P_v + P_v R_v^{-1} P_v + Q_v = 0 \tag{13.107}$$

or, equivalently,

$$P_v A_v^T + A_v P_v = -P_v R_v^{-1} P_v - Q_v < 0 \tag{13.108}$$

and therefore, A_v is Hurwitz for any $v > 0$.

Since P_v is the unique symmetric positive-definite solution of the ARE (13.103), then the matrix inverse $\tilde{P}_v = P_v^{-1}$ exists for any $v \geq 0$, and the following relation holds:

$$A_v^T \tilde{P}_v + \tilde{P}_v A_v = -R_v^{-1} - \tilde{P}_v Q_v \tilde{P}_v < 0 \tag{13.109}$$

The design task is to choose adaptive laws for $\hat{\Theta}$ so that the tracking error e globally asymptotically tends to the origin. We consider the following Lyapunov function candidate:

$$V(e,\ \Delta\Theta) = e^T \tilde{P}_v e + \mathrm{trace}\left(\Lambda \Delta\Theta^T \Gamma_\Theta^{-1} \Delta\Theta\right) \tag{13.110}$$

where $\Gamma_\Theta = \Gamma_\Theta^T > 0$ is the adaptation rate. The time derivative of V, along the trajectories of the error dynamics (13.101), can be computed as

$$\dot{V}(e,\Delta\Theta)=e^T\tilde{P}_v\dot{e}+\dot{e}^T\tilde{P}_v\dot{e}+2\text{trace}\left(\Lambda\Delta\Theta^T\Gamma_\Theta^{-1}\dot{\hat{\Theta}}\right)$$
$$=e^T\tilde{P}_v\left(A_ve-B\Lambda\Delta\Theta^T\Phi(x)\right)+\left(A_ve-B\Lambda\Delta\Theta^T\Phi(x)\right)^T\tilde{P}_v\dot{e}+2\text{trace}\left(\Lambda\Delta\Theta^T\Gamma_\Theta^{-1}\dot{\hat{\Theta}}\right)$$
$$=e^T\left(\tilde{P}_vA_v+A_v^T\tilde{P}_v\right)e-2e^T\tilde{P}_vB\Lambda\Delta\Theta^T\Phi(x)+2\text{trace}\left(\Lambda\Delta\Theta^T\Gamma_\Theta^{-1}\dot{\hat{\Theta}}\right) \tag{13.111}$$

Because of (13.108) and using the properties of the matrix trace operator, we get

$$\dot{V}(e,\ \Delta\Theta)=-e^T\left(R_v^{-1}+\tilde{P}_vQ_v\tilde{P}_v\right)e$$
$$+2\,\text{trace}\left(\Lambda\,\Delta\Theta^T\left(\Gamma_\Theta^{-1}\dot{\hat{\Theta}}-\Phi(x)\,e^T\tilde{P}_vB\right)\right) \tag{13.112}$$

If the adaptive laws are chosen as

$$\dot{\hat{\Theta}}=\Gamma_\Theta\Phi(x)\,e^T\tilde{P}_vB \tag{13.113}$$

then

$$\dot{V}(e,\ \Delta\Theta)=-e^T\left(R_v^{-1}+\tilde{P}_vQ_v\tilde{P}_v\right)e\le 0 \tag{13.114}$$

and hence, $V(e,\ \Delta\Theta)$ is the Lyapunov function for the error dynamics (13.101). For this reason, the tracking error signal e, as well as the parameter error matrix $\Delta\Theta$, is uniformly bounded in time, that is, $(e,\ \Delta\Theta)\in L_\infty$. Since A_{ref} in (13.97) is Hurwitz by design and $(e,\,y_{cmd})\in L_\infty$, then $\left(x_{ref},\,\dot{x}_{ref}\right)\in L_\infty$, and consequently $x\in L_\infty$. Since the unknown parameters Θ are constant and $\Delta\Theta\in L_\infty$, then $\hat{\Theta}\in L_\infty$. We assumed that the regressor vector $\Phi\left(x_p\right)$ is Lipschitz-continuous, and we have shown that $\left(x,\,\hat{\Theta}\right)\in L_\infty$. Therefore, from the definition (13.98), it follows that $u\in L_\infty$ and consequently $\dot{x}\in L_\infty$. Also, since $\dot{x}_{ref}\in L_\infty$, then $\dot{e}\in L_\infty$. Using (13.114) yields

$$\ddot{V}(e,\ \Delta\Theta)=-2\,e^T\left(R_v^{-1}+\tilde{P}_vQ_v\tilde{P}_v\right)\dot{e}\in L_\infty \tag{13.115}$$

The function V from (13.110) is lower bounded and has a nonincreasing time derivative as in (13.114). Thus, V tends to a limit, as $t\to\infty$. Also, the function second time derivative is uniformly bounded. Therefore, $\dot{V}$ is a uniformly continuous function of time. Using Barbalat's lemma, we immediately conclude that $\dot{V}$ tends to zero, as $t\to\infty$. Due to (13.114), we finally arrive at

$$\lim_{t\to\infty}\|e(t)\|=0 \tag{13.116}$$

which proves global asymptotic stability of the tracking error, attained by the adaptive controller (13.98), the adaptive laws (13.113), and the observer-like reference model (13.97).In order to show that x_{ref} asymptotically tracks $x_{ref\,ideal}$,

it is sufficient to subtract (13.90) from (13.97) and write the dynamics of the reference model error $e_{ref} = x_{ref} - x_{ref\,ideal}$:

$$\dot{e}_{ref} = A_{ref}\, e_{ref} + L_v \underbrace{e(t)}_{\mathrm{o}(1)} \tag{13.117}$$

Then (see Exercise 13.2),

$$e_{ref}(t) = \exp\left(A_{ref}\, t\right) e_{ref}(0) + \int_0^t \exp\left(A_{ref}\,(t-\tau)\right) L_v \underbrace{e(\tau)}_{\mathrm{o}(1)}\, d\tau = \mathrm{o}(1) \underset{t\to\infty}{\to} 0 \tag{13.118}$$

We have proven that $x \underset{t\to\infty}{\to} x_{ref} \underset{t\to\infty}{\to} x_{ref\,ideal}$, and so

$$(y = C\,x) \underset{t\to\infty}{\to} \left(y_{ref} = C\,x_{ref}\right) \underset{t\to\infty}{\to} \left(y_{ref\,ideal} = C\,x_{ref\,ideal}\right) \to y_{cmd}(t) \tag{13.119}$$

In other words, the system-regulated output y asymptotically tracks its ideal reference command $y_{ref\,ideal}$, and y also tracks its original command y_{cmd} with bounded errors.

13.6 Transient Dynamics Analysis

Let us now analyze the transient dynamics (13.101). To do that, we shall employ the results from Theorem 13.1 and singular perturbation techniques from Sects. 13.1 and 13.2.

Substituting (13.102) into (13.101), the transient error dynamics can be written as

$$\dot{e} = \underbrace{\left(A_{ref} - P_v R_v^{-1}\right)}_{\text{Hurwitz Matrix}} e - \underbrace{B\,\Lambda\,\Delta\Theta(t)^T\,\Phi(x(t))}_{\varphi(t)=\text{Uniformly Bounded Function of Time}} \tag{13.120}$$

Using (13.104) gives

$$\dot{e} = \left(A_{ref} - \left(1 + \frac{1}{v}\right) P_v\right) e - \varphi(t) \tag{13.121}$$

In Sect. 13.3, we have shown that the asymptotic relation

$$P_v = P_0 + \mathrm{O}(v), \quad \text{as } v \to 0 \tag{13.122}$$

holds with a constant positive-definite symmetric matrix P_0. Then,

$$\dot{e} = \left(A_{ref} - \left(1 + \frac{1}{\nu} \right) \left(P_0 + \mathrm{O}(\nu) \right) \right) e - \varphi(t) \tag{13.123}$$

or, equivalently,

$$\nu \dot{e} = \left(\nu A_{ref} - (\nu + 1) \left(P_0 + \mathrm{O}(\nu) \right) \right) e - \nu \varphi(t) \tag{13.124}$$

We can rewrite (13.124) as

$$\begin{aligned} \nu \dot{e} &= \left(\nu A_{ref} - (\nu + 1) \left(P_0 + \mathrm{O}(\nu) \right) \right) e - \nu \varphi(t) \\ &= \left(-P_0 + \underbrace{\left(\nu A_{ref} - \nu \left(P_0 + \mathrm{O}(\nu) \right) - \mathrm{O}(\nu) \right)}_{\mathrm{O}(\nu)} \right) e + \nu \varphi(t) \\ &= \left(-P_0 + \mathrm{O}(\nu) \right) e + \nu \varphi(t) \end{aligned} \tag{13.125}$$

and then compare it to (13.31). Then, according to Theorem 13.1, the trajectories of (13.123) satisfy the following asymptotics,

$$e(t) = \mathrm{O}\left(e^{-\gamma \frac{t}{\nu}} \right) + \mathrm{O}(\nu), \quad (\nu \to 0) \tag{13.126}$$

uniformly in time, with a positive constant γ, and for all sufficiently small $\nu > 0$. So, the transient dynamics exponentially decays to a neighborhood of the origin at the decay rate no slower than $\mathrm{O}\left(e^{-\gamma \frac{t}{\nu}} \right)$. Moreover, the "diameter" of the convergence set can be made smaller by choosing sufficiently small ν. This argument formally proves our claim about the transient dynamics improvement in MIMO MRAC systems with observer-like reference models.

Similar to the arguments from Sect. 13.2, we can offer an alternative way to analyze the transient dynamics in (13.124). This is a singularly perturbed system, and its dynamics are in the form of (13.31), where ν (instead of ε) is the small parameter. So, in order to understand the intricacies of the system behavior, we can employ the singular perturbation arguments yet again. Setting $\nu = 0$ gives the isolated root $e = 0$ for the corresponding reduced system, which describes asymptotic behavior as $t \to \infty$, that is for a sufficiently small $\nu > 0$, the error trajectories converge to a small neighborhood of the manifold $e \equiv 0$ and will evolve near this manifold thereafter.

In order to quantify and characterize the transient dynamics, we need to form the boundary-layer system. These dynamics are formed by "stretching" the time,

$$\tau = \frac{t}{\nu} \tag{13.127}$$

rewriting (13.124) in the "fast" timescale τ, and then setting $\nu = 0$. The resulting boundary-layer dynamics

$$\frac{d\,e}{d\,\tau} = -P_0\,e \tag{13.128}$$

are globally exponentially stable, since P_0 is symmetric and positive definite. According to Theorem 13.2, we claim that for a sufficiently small $\nu > 0$, the singular perturbation system (13.124) has a unique solution $e(t,\,\nu)$, defined on $[0,\ \infty)$, and the asymptotic relation

$$e(t,\,\nu) = \bar{e}\left(\frac{t}{\nu}\right) + \mathrm{O}(\nu) \tag{13.129}$$

holds uniformly on $[0,\ \infty)$, where $\bar{e}\left(\frac{t}{\nu}\right)$ is the solution of the boundary-layer system (13.128). Since

$$\bar{e}\left(\frac{t}{\nu}\right) = \exp(-P_0\,(t))\ \bar{e}(0) \tag{13.130}$$

then substituting (13.130) into (13.129) results in

$$e(t,\,\nu) = \exp\left(-P_0\left(\frac{t}{\nu}\right)\right)\ \left(x(0) - x_{ref}(0)\right) + \mathrm{O}(\nu) \tag{13.131}$$

This asymptotic relation is conservative. In fact, we have proven that the tracking error $e(t,\,\nu)$ asymptotically converges to the origin, starting from any initial condition. Consequently (see Exercise 13.3),

$$\varphi(t) = B\,\Lambda\,\underbrace{\left[\Delta\Theta(t)^T\,\Phi(x(t))\right]}_{\mathrm{o}(1)} = \mathrm{o}(1), \quad (t \to \infty) \tag{13.132}$$

and so, we can rewrite (13.131) as

$$x(t,\,\nu) = \underbrace{\boxed{\exp\left(-P_0\left(\frac{t}{\nu}\right)\right)\ \left(x(0) - x_{ref}(0)\right)}}_{\text{Transient Dynamics}} + \underbrace{\boxed{x_{ref}(t) + \mathrm{O}(\nu)\ \mathrm{o}(1)}}_{\text{Global Asymptotic Stability}} \tag{13.133}$$

where $\mathrm{o}(1)$ is a function of time, defined such that $\lim_{t\to\infty} \mathrm{o}(1) = 0$, while $\mathrm{O}(\nu)$ is a function of ν only, and it decays to zero no slower than ν.

Let us emphasize again that the asymptotic expansion (13.133) quantifies the transient dynamics due to the adaptive controller (13.98) and (13.113). Indeed, for a sufficiently small $\nu>0$, the transients in the error dynamics are described by the linear time-invariant globally exponentially stable system (13.128), whose solution is given by (13.130) and (13.133). The second term in (13.133) defines asymptotic behavior of

Table 13.1 Observer-like MRAC design summary

Open-loop plant	$\dot{x} = A_{ref}\, x + B\, \Lambda \left(u + \Theta^T\, \Phi(x)\right) + B_{ref}\, y_{cmd}$
Observer-like reference model	$\dot{x}_{ref} = A_{ref}\, x_{ref} + L_v \left(x - x_{ref}\right) + B_{ref}\, y_{cmd}$
State tracking error	$e = x - x_{ref}$
Riccati equation for adaptive laws	$P_v\, A_{\text{ref}}^T + A_{\text{ref}}\, P_v - P_v\, R_v^{-1}\, P_v + Q_v = 0$
ARE weight matrices	$Q_v = Q_0 + \left(\frac{v+1}{v}\right) I_{n\times n}, \quad R_v = \frac{v}{v+1}\, I_{n\times n}$
Observer gain	$L_v = P_v\, R_v^{-1}$
Total control input	$u = -\hat{\Theta}^T\, \Phi(x)$
MRAC laws	$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x)\, e^T\, P_v^{-1}\, B$

the tracking error, as $t \to \infty$. This fact constitutes the main benefit of the error feedback term in the observer-like reference model (13.97). Essentially, using a sufficiently small parameter $v > 0$ ensures quantifiable transient characteristics of the corresponding closed-loop tracking performance, and these transients are given by the first term in (13.133). A summary of the design is given in Table 13.1.

The system dynamics (13.87) and the corresponding control problem formulations can be modified to include nonparametric uncertainties, such as matched uncertainty approximation errors and bounded possibly non-matched process noise. In that case, one can use known robustification techniques (i.e., e modification, σ modification, and Projection Operator) to prove bounded tracking performance and then establish similar to (13.131) transient characteristics.

Finally, we would like to note that the state feedback MRAC design developed in this chapter, with an observer-like reference model, can be extended to adaptive output feedback controllers [6]. This topic will be addressed in the next chapter.

13.7 Summary

This section was devoted to the development and analysis of an observer-like modification to the reference dynamics formulation, within the MRAC state feedback framework. We draw a parallel between the derived modification and the theory of Luenberger observers. This modification allowed us to quantify and influence transient dynamics in adaptive control. Overall, the derived design represents a numerically efficient technique of reducing unwanted transient oscillations in state feedback/feedforward MRAC systems.

13.8 Exercises

Exercise 13.1. Show that if the external command $r(t)$ in (13.1) is continuously differentiable, and its rate $\dot{r}(t)$ is uniformly bounded in time, then the signal $\varphi(t)$ in (13.8) asymptotically tends to zero, as $t \to \infty$. (*Hint*: Differentiate the error dynamics (13.5) and show that $\ddot{e}(t)$ is uniformly bounded. Then, use Barbalat's lemma to establish asymptotic convergence of $\dot{e}(t)$ to zero).

Exercise 13.2. Prove (13.36). (*Hint*: Show that if (13.42) holds true, then $\|z(t)\|$ in (13.40) asymptotically in time tends to zero. Use [3, Lemma 9.6, p. 355] to aid in the proof).

Exercise 13.3. Show that for the extended dynamics (13.87), driven by the MRAC controller (13.98), the smoothness requirement on the command y_{cmd} can be removed, yet the signal $\varphi(t)$ in (13.132) will tend to zero asymptotically in time. This formally proves validity of using $o(1)$ in the asymptotic relation (13.133).

Exercise 13.4. Consider the aircraft dynamics and the MRAC design from Example 13.2. For the same system, design and simulate an MRAC controller with an observer-like dynamics. Compare and discuss the two designs.

References

1. Whitaker, H.P., Yamron, J., Kezer, A.: Design of model-reference control systems for aircraft. Rep. R-164. Instrumentation Laboratory, MIT, Cambridge, MA (1958)
2. Luenberger, D.G.: Observing the state of a linear system. IEEE Trans. Mil. Electron. **MIL-8**, 74–80 (1964)
3. Khalil, H.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (1996). 07458
4. Kevorkian, J., Cole, J.D.: Multiple Scale and Singular Perturbation Methods. Appl. Math. Sciences (Springer-Verlag New York Inc), vol. 114. Springer, New York (1996)
5. Murray, J.D.: Asymptotic Analysis. Appl. Math. Sciences (Springer-Verlag New York Inc), vol. 48. Springer, New York (1984)
6. Lavretsky, E.: Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. In: Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Ontario, Canada (2010)
7. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)

Chapter 14
Robust and Adaptive Control with Output Feedback

14.1 Introduction

The design of output feedback tracking controllers for nonlinear uncertain multi-input multi-output (MIMO) systems represents a challenging problem. In aerospace applications, such a challenge frequently arises in control of aerial vehicles whose dynamics contain flexible modes which cannot be ignored. Dynamics of these aerial vehicles exhibit almost no frequency separation between the vehicle primary and its flexible modes. The main challenge here arises when the flexible modes have low damping ratios and as such must be actively controlled or stabilized. In realistic applications, "flexible" vehicle state components are not available online, as the system measurements. In other words, not all of the system degrees-of-freedom are measured. In order to control such a system, one needs to construct a static or a dynamic output feedback. What complicates this situation is the fact that more often than not, flexible mode dynamics contain parametric uncertainties. For example, natural frequencies and damping ratios of these modes may not be known exactly, and they may depend on slowly varying parameters, such as the vehicle airspeed and gross weight. Also, uncertain aerodynamics and structurally inherent nonlinearities may influence the interconnections between the vehicle flexible and its primary modes.

In most practical applications, the number of measured output signals in these systems would exceed the number of control inputs. In order to yield desired input–output signal characteristics, the output measurements are often defined by a set of user-selected sensors that are placed at the desired locations on the vehicle [1]. At the same time, the system-regulated outputs are often determined based on a desired mission for the system to perform, and as such, their selection is often restricted and dictated by the system requirements. Thus, we have two sets of outputs in the system – the selectable set of measurements (sensors) and the prescribed regulated signals (mission requirements). Comprehensive surveys of input–output selection techniques can be found in [1, 2]. This brief discussion gives a motivation to the specific problem formulation that will be formally given in Sect. 14.4.

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_14,

An original framework for the design of adaptive output controllers was given in [3]. Many theoretical advancements were made since. Three notable contributions include (a) adaptive backstepping [4], (b) adaptive control with high gain observers [5–7], and (c) multiple model adaptive control [8]. Other ideas were also exploited, such as using time-delayed values to approximate system dynamics [9].

A command tracking controller for a dynamical system with uncertainties must be capable of (a) achieving desired tracking performance, (b) enforcing robustness, and (c) mitigating uncertainties. Finding such a control solution can be facilitated if one leverages known and well-established methods for MIMO linear-time-invariant (LTI) systems, with completely known dynamics. Indeed, for LTI dynamics with partial state measurements, there exist several formal (i.e., theoretically justified) design methods to construct output feedback controllers, both static and dynamic. Among those, the linear quadratic Gaussian synthesis with Loop Transfer Recovery (LQG/LTR) design, first given by Doyle and Stein in [10], represents one of the most frequently used methodologies for robust output feedback control design. The popularity of this technique is primarily based on its guaranteed properties, such as closed-loop stability and robustness to parametric uncertainties.

It would be safe to say that the development of the LQG/LTR design methodology was influenced by the seminal work of Kwakernaak and Sivan [11, 12], where the authors investigated "... the maximal achievable accuracy of linear optimal regulators," [11]. This asymptotic property of the LQR solutions allowed Doyle and Stein to develop their LQG/LTR technique [10], with subsequent extensions and interpretations reported elsewhere in the literature [13, 14]. In contrast to [11], where achieving zero cost was the goal, the LQG/LTR method aims at recovering the loop shapes of optimal full-state regulators. The state feedback loop recovery is achieved via specific choices of free design parameters, such as the process and measurement noise intensity matrices. Basically, these two matrices become the "tuning knobs" of the LQG/LTR design process.

Continuing this line of thoughts, we shall revisit and refine the results from [15], where a constructive design is proposed for composing adaptive output feedback controllers that are applicable to a generic class of uncertain MIMO systems. We shall also add a simulation case study to demonstrate key features and benefits of our methodology. Specifically, we will show that using a Luenberger-based state observer for a "squared-up" system [16] enables the design of a direct adaptive model reference output feedback controller for MIMO systems with matched uncertainties while regulating output signals whose dynamics may have high relative degree and are not necessarily minimum-phase.

This chapter material is organized as follows. In Sect. 14.2, we shall present mathematical preliminaries, including basic definitions and notations from singular perturbations. Section 14.3 defines MIMO systems of interest and the associated tracking control problem formulation. Our main result (Theorem 14.1) is given in Sect. 14.4, followed by a flight control case study in Sect. 14.5.

14.2 Mathematical Preliminaries

We write R^n to represent the Euclidean n-dimensional space and $R^{n\times m}$ to denote the space of all n-by-m matrices, where n and m are integers. For any $x \in R^n$, $\|x\|$ denotes the Euclidean vector norm of x, and for any $A \in R^{n\times m}$, $\|A\|$ is the corresponding induced matrix norm. Also, $\mathbb{C}^-$ symbolizes the open left half of the complex plane (excluding the $j\omega$ – axis).

We will need basic definitions of asymptotic orders and relations from Chap. 13, Sect. 13.2. Recall that given a matrix $A_\varepsilon(x) = A(x;\ \varepsilon) \in R^{n\times n}$, the asymptotic equation

$$A_\varepsilon(x) = A_0(x) + A_1(x)\,\varepsilon + \mathrm{O}\big(\varepsilon^2\big), \quad \text{as } \varepsilon \to 0 \tag{14.1}$$

means that for every x from a domain $X \subset R^n$,

$$\lim_{\varepsilon\to 0} \|A_\varepsilon(x) - A_0(x) - A_1(x)\,\varepsilon\| = \lim_{\varepsilon\to 0} \big\|\mathrm{O}\big(\varepsilon^2\big)\big\| = 0 \tag{14.2}$$

and the convergence rate in (14.2) is no slower than ε^2, for every fixed x. The "Big O" symbol in (14.1) and (14.2) comes from the Bachmann–Landau asymptotic order notations [17, 18].

Let $Q_0 \in R^{n\times n}$ and $R_0 \in R^{m\times m}$ be symmetric and positive definite. For two controllable and observable pairs of matrices, (A, B) and (A, C), where $A \in R^{n\times n}$, $B \in R^{n\times m}$, $C \in R^{m\times n}$, and $m \le n$, we choose a constant $v > 0$ and define two symmetric positive definite weight matrices:

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) B\,B^T, \quad R_v = \frac{v}{v+1}\,R_0 \tag{14.3}$$

Then, one can show that for any $v > 0$, the algebraic Riccati equation (ARE)

$$P_v\,A^T + A\,P_v - P_v\,C^T\,R_v^{-1}\,C\,P_v + Q_v = 0 \tag{14.4}$$

has the unique symmetric positive definite solution $P_v > 0$ [10, 11].

We are interested in the asymptotic behavior of P_v, as the positive constant parameter v tends to zero. In Chap. 13 (Corollary 13.1), we stated and proved several interesting asymptotic properties of the ARE solution P_v, using an asymptotic expansion in the form

$$P_v = P_0 + P_1\,v + \mathrm{O}\big(v^2\big) \tag{14.5}$$

for $v \to 0$. In fact, we have shown that under the following three conditions:

- The number of outputs and inputs in the system are the same: $p = m$.
- The system relative degree is one: $\det(CB) \neq 0$.
- The transfer function $G(s) = C\left(sI_{n\times n} - A\right)^{-1} B$ is minimum-phase.

the ARE solution P_v and its inverse P_v^{-1} exist, and both matrices are symmetric strictly positive definite, uniformly in $v \geq 0$. We have also shown that the asymptotic relation

$$P_v^{-1} B = C^T R_0^{-\frac{1}{2}} W + \mathrm{O}(v) \tag{14.6}$$

takes place as $v \to 0$, where $W = (UV)^T$; the two unitary matrices U and V are defined by the singular value decomposition $B^T C^T R_0^{-\frac{1}{2}} = U \Sigma V$, and Σ is the diagonal matrix of the corresponding singular values.

We shall use (14.6) to aid in the design of adaptive model reference output feedback controllers. In order to do that, we need to discuss a generic class of systems whose input–output dynamics are not square, that is, the number of the system inputs may or may not be the same as the number of its outputs. For these systems, the asymptotics (14.6) cannot be achieved. First, we are going to modify the system input–output dynamics to make it square.

Practical methods to "square-up" MIMO systems can be found in [16]. The squaring-up is accomplished by adding pseudo (i.e., fictitious) inputs or outputs. This procedure allows to enforce the three key assumptions that are listed below (14.5). Eventually, it leads to the desired asymptotic relation (14.6).

We shall deal with systems that have more outputs than inputs. This is a reasonable assumption since outputs represent sensors and their number and locations can be chosen by the system designer [1, 2]. The squaring-up problem for a non-square linear MIMO system, with m inputs and $(m < p)$ outputs,

$$\begin{array}{c} m - \text{Inputs} \\ \Downarrow \\ p - \text{Outputs} \Rightarrow \end{array} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in R^{(n+p)\times(n+m)}$$

can be stated as follows [16]: "Given the state matrix $A \in R^{n\times n}$, the input matrix $B \in R^{n\times m}$, and the output matrices $C \in R^{p\times n}$, $D \in R^{p\times m}$, with $(n > m,\ p > m)$, determine pseudo-input matrices $B_2 \in R^{n\times(p-m)}$ and $D_2 \in R^{p\times(p-m)}$, such that the resulting square system with p inputs and p outputs,

$$\begin{array}{c} p - \text{Inputs} \\ \Downarrow \\ p - \text{Outputs} \Rightarrow \end{array} \begin{pmatrix} A & (B, B_2) \\ C & (D, D_2) \end{pmatrix} \in R^{(n+p)\times(n+p)}$$

has its transmission zeros in the open left half complex plane, $\mathbb{C}^-$."

In [16], several constructive algorithms for solving the squaring-up problem are given. Systems that we shall encounter in this chapter will have no feedforward connections, that is, $D = 0_{p\times m}$. In this case, the squaring-up problem is reduced to finding a pseudo-input matrix $B_2 \in R^{n\times(p-m)}$, such that the square system

$$\begin{array}{c} \\ p-\text{Outputs} \Rightarrow \end{array} \begin{array}{c} p-\text{Inputs} \\ \Downarrow \\ \begin{pmatrix} A & (B, B_2) \\ C & 0_{p\times p} \end{pmatrix} \end{array} \in R^{(n+p)\times(n+p)}$$

has its transmission zeros in $\mathbb{C}^-$.

Let us mention that squaring-up problems have multiple solutions. Reference [16] gives two sufficient conditions for a solution to exist. They are:

1. (A, B) to be controllable.
2. $\text{rank}(CB) = m$.

Observe that when a squaring-up solution is found, the resulting system transfer function becomes minimum phase and has its relative degree equal to one.

In what follows, we shall exploit the squaring-up technique to enforce the asymptotic relation (14.6), which subsequently will allow us to design adaptive output feedback controllers with quantifiable performance and stability guarantees.

14.3 System Dynamics and Control Problem Formulation

We consider a class of nonlinear MIMO uncertain dynamical systems in the form

$$\begin{aligned} \dot{x} &= A\,x + B\,\Lambda\left(u + \Theta^T\,\Phi(x)\right) + B_{ref}\,z_{cmd} \\ y &= C\,x, \quad z = C_z\,x \end{aligned} \tag{14.7}$$

where $A \in R^{n\times n}$, $(B, B_{ref}) \in R^{n\times m}$, $C \in R^{p\times n}$, and $C_z \in R^{m\times n}$ are known matrices, (A, B) is controllable, and (A, C) is observable. The system state is $x \in R^n$, and the control input is $u \in R^m$. The system measurements are grouped into $y \in R^p$, the regulated output is $z \in R^m$, and $z_{cmd} \in R^m$ denotes an external bounded time-varying command for the regulated output z to follow. The system uncertainties are represented by a constant unknown non-singular diagonal matrix $\Lambda \in R^{m\times m}$, a constant unknown matrix $\Theta \in R^{N\times m}$, and a known regressor vector $\Phi(x) \in R^N$. It is assumed that the regressor is globally Lipschitz-continuous in x. The Lipschitz assumption implies that there exists a finite positive known constant $0 < L_\Phi < \infty$, such that

$$\|\Phi(x_1) - \Phi(x_2)\| \le L_\Phi\,\|x_1 - x_2\| \tag{14.8}$$

for any $x_1,\ x_2 \in R^n$. This assumption assures global existence and uniqueness of the system trajectories [5]. Also, we suppose that the number of the system output measurements p is greater than the number of the control inputs m, with rank $(C\,B) = m$.

The control goal of interest is to design u, based on the measurements y, such that z tracks z_{cmd} with bounded errors while operating in the presence of uncertain parameters $(\Lambda,\ \Theta)$.

Let us remind the reader that throughout this book, we have already considered systems of the form (14.7), on many occasions. For example, for a MIMO system with n_p states, one can add commands z_{cmd} into the problem formulation by augmenting the system with the integrated tracking error $\dot{e}_I = z - z_{cmd}$. In this case, $B_{ref} = \begin{pmatrix} -I_{m\times m} & 0_{n_p\times m} \end{pmatrix}^T$. Other formulations are possible, whereby filtered versions of the tracking error or of the system output can also be incorporated into the system dynamics.

Our control design approach will be reference-model-based, and as such, the first step here is to construct a desired reference model with target dynamics. Toward that end, we can employ the LQR method and compute an optimal state feedback gain matrix $K_{LQR} \in R^{n\times m}$ such that

$$A_{ref} = A - B\,K_{LQR}^T \tag{14.9}$$

is Hurwitz and has the desired modal characteristics, leading to the exponentially stable reference model

$$\dot{x}_{ref} = A_{ref}\,x_{ref} + B_{ref}\,z_{cmd}, \quad z_{ref} = C_z\,x_{ref} \tag{14.10}$$

whose output z_{ref} adequately tracks its command z_{cmd}, with bounded errors. Other methods to construct desired reference models can also be employed here.

Let u_{bl} denote a baseline controller and let u_{ad} be an adaptive augmentation (an incremental control signal). We define the total control as a sum

$$u = u_{bl} + u_{ad} \tag{14.11}$$

and then rewrite the system dynamics (14.7):

$$\dot{x} = A\,x + B\,u_{bl} + B\,\Lambda\left(u_{ad} + \Theta^T\,\Phi(x) + \left(I_{m\times m} - \Lambda^{-1}\right)u_{bl}\right) + B_{ref}\,z_{cmd} \tag{14.12}$$

Expressing the system matched uncertainty as

$$\begin{aligned}\Theta^T\,\Phi(x) + \left(I_{m\times m} - \Lambda^{-1}\right)u_{bl} &= \underbrace{\begin{pmatrix}\Theta^T & I_{m\times m} - \Lambda^{-1}\end{pmatrix}}_{\bar{\Theta}^T}\ \underbrace{\begin{pmatrix}\Phi(x)\\ u_{bl}\end{pmatrix}}_{\bar{\Phi}(x,\,u_{bl})}\\ &= \bar{\Theta}^T\,\bar{\Phi}(x,\,u_{bl})\end{aligned} \tag{14.13}$$

gives

$$\dot{x} = A\,x + B\,u_{bl} + B\,\Lambda\left(u_{ad} + \bar{\Theta}^T\,\bar{\Phi}(x,\,u_{bl})\right) + B_{ref}\,z_{cmd} \tag{14.14}$$

where $\bar{\Theta}$ and $\bar{\Phi}(x,\,u_{bl})$ are the extended parameter and regressor vectors, respectively.

The system (14.14) represents the open-loop dynamics that we shall exploit. Our overall strategy is to find u, in the form of (14.11), to force the system state x track the state of the reference model x_{ref}, with bounded errors. As a result, the system-regulated output z will track the external command z_{ref}, and consequently, z will also track z_{cmd}, which constitutes our main control goal.

From (14.8) and (14.13), it follows that the extended regressor vector $\bar{\Phi}(x,\,u_{bl})$ satisfies the Lipschitz condition

$$\left\|\bar{\Phi}(x_1,\,u_{bl}) - \bar{\Phi}(x_2,\,u_{bl})\right\| = \|\Phi(x_1) - \Phi(x_2)\| \leq L_\Phi\,\|x_1 - x_2\| \tag{14.15}$$

with the same Lipschitz constant L_Φ, as in (14.8).

Before proceeding further, let us discuss and motivate our specific selection of the original system definition (14.7) and its equivalent form (14.14). First, we note that these dynamics are as generic as the ones from the now-classical model reference adaptive control (MRAC) problem formulation [3, 19, 20]. The only difference here is that we have embedded the matching conditions assumption into the problem formulation.

Our selection of the control uncertainties in the form of $B\,\Lambda$ comes predominantly from aerospace applications, where control directions are usually known but their magnitudes are not. We have decided to introduce linear-in-parameters uncertainties only for the sake of presentation clarity. It is possible to redefine the system dynamics to include matched nonparametric uncertainties and to also add non-matched uncertainties, such as bounded process noise. We have dealt with similar constructs in the previous chapters and offered several modifications to account for these classes of systems.

Let us make another remark on the generality of the problem formulation. The selected system emulates flight dynamics of aerial platforms. In fact, our choice of the system dynamics is directly influenced and driven by standard (in aerospace) flight dynamics formulations for control design. Furthermore, the control design methodology developed in this chapter can be extended to systems with nonlinear dependence on uncertain parameters and to systems with time-varying uncertainties, as long as there is a known rate upper bound. In addition, process noise can be added to the system dynamics. All these modifications would reduce our method applicability from being global to that of a semi-global nature.

Moreover, the command feedforward term $B_{ref}\,z_{cmd}$ in (14.7) allows to encompass a specific class of dynamical systems, where integrated tracking errors or their filtered versions need to be inserted and accounted for during control design. Again, this is an option, not a requirement. One final comment: Our control solution will be

given in the form of an adaptive output feedback. Such a solution is not unique. Alternatively, robust controllers can also be developed to solve the posed tracking problem. The reader is encouraged to design a robust controller and then compare its performance with that of an adaptive system, whereby both controllers shall be constructed to perform the same tracking task.

We now summarize all of the assumptions for our method to be valid and then comment on their imposed restrictions.

Assumption 14.1.

- $(A,\ B)$ is controllable and $\text{rank}\, B = m$ (B has full column rank).
- $(A,\ C)$ is observable and $\text{rank}\, C = p$ (C has full row rank).
- The number of measured outputs is greater than the number of control inputs $(p > m)$ and $\text{rank}\,(CB) = m$ (same as the number of control inputs).

The first two assumptions are standard in dynamics and control [21]. The third assumption is very common for most practical systems in aerospace, automotive, and other industries, where the outputs (sensors) and the inputs (actuators) are defined by the vehicle designer and are placed at specific locations on the vehicle, in order to achieve desired input–output characteristics [1, 2]. We note that the assumed full rank condition does not constitute a restriction on the system-regulated output $z = C_z x$. These limitations are placed on the system measured output signals that are selected by system architects to produce a controllable and observable vehicle configuration. On the other hand, the system-regulated output is not often selectable, and its dynamics are allowed to be nonminimum-phase or have a high relative degree.

Under the above three assumptions, constructive numerical methods have been developed in [16] that solve the "squaring-up" problem of finding a constant matrix $B_2 \in R^{n\times(p-m)}$, such that $\det(C\bar{B}) \neq 0$ with $\bar{B} = (B \quad B_2)$ and the transfer function $C\,(s I_{n\times n} - A)^{-1}\bar{B}$ becomes minimum-phase (i.e., transmission zeros are located in $\mathbb{C}^-$). The added pseudo-control columns B_2 are in the sense "fictitious," meaning that they do not represent physical inputs in the system. We have already discussed the squaring-up problem in Sect. 14.2. □

In the next section, we are going to employ the squaring-up paradigm to aid in the design of an output feedback adaptive controller.

14.4 Adaptive Output Feedback Design and Analysis

Based on the system dynamics (14.14), we introduce a Luenberger-like state observer in the form:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + B u_{bl} + B\hat{\Lambda}\left(u_{ad} + \hat{\bar{\Theta}}^T \bar{\Phi}(\hat{x},\ u_{bl})\right) + L_v\,(y - \hat{y}) + B_{ref}\, z_{cmd} \\ \hat{y} &= C\hat{x}\end{aligned} \tag{14.16}$$

where $\hat{x} \in R^n$, $\hat{y} \in R^m$, and $L_v \in R^{n \times m}$ are the observer state, the predicted output signal, and the output prediction error feedback gain, respectively. Also, in (14.16), $\left(\hat{\Lambda} \in R^{m \times m},\ \hat{\bar{\Theta}} \in R^{(N+m) \times m}\right)$ represent the estimated parameters. The observer will be designed to estimate the system state $x(t)$, with bounded errors.

We select the baseline linear controller

$$u_{bl} = -K_{lqr}^T \hat{x} \tag{14.17}$$

and choose an adaptive augmentation in the form

$$u_{ad} = -\hat{\bar{\Theta}}^T \bar{\Phi}(\hat{x},\ u_{bl}) \tag{14.18}$$

with the intent to cancel/dominate estimation errors due to matched uncertainties in the observer dynamics (14.16). Substituting (14.18) into (14.16) gives the open-loop observer:

$$\begin{aligned} \dot{\hat{x}} &= A\,\hat{x} + B\,u_{bl} + L_v\,(y - \hat{y}) + B_{ref}\,z_{cmd} \\ \hat{y} &= C\,\hat{x} \end{aligned} \tag{14.19}$$

With (14.9) and (14.17), we obtain the observer closed-loop linear time-varying dynamics:

$$\begin{aligned} \dot{\hat{x}} &= A_{ref}\,\hat{x} + L_v\,(y - \hat{y}) + B_{ref}\,z_{cmd} \\ \hat{y} &= C\,\hat{x} \end{aligned} \tag{14.20}$$

It is evident that both the open-loop (14.19) and the closed-loop (14.20) observers do not explicitly depend on $\hat{\Lambda}$. Hence, the only parameter to be estimated is $\hat{\bar{\Theta}}$, which appears in the control input formulation (14.18). However, by the definition (14.13), the ideal unknown matrix $\bar{\Theta}$ contains $\left(I_{m \times m} - \Lambda^{-1}\right)$. So, any estimate of $\bar{\Theta}$ will certainly contain an estimate of Λ^{-1}. Hence, we would indirectly estimate Λ, after all.

Substituting (14.18) into the system (14.14) yields

$$\dot{x} = A\,x + B\,u_{bl} - B\,\Lambda\left(\hat{\bar{\Theta}}^T \bar{\Phi}(\hat{x},\ u_{bl}) - \bar{\Theta}^T \bar{\Phi}(x,\ u_{bl})\right) + B_{ref}\,z_{cmd} \tag{14.21}$$

Our task is to select $\left(L_v,\ \hat{\bar{\Theta}}\right)$, such that the state $\hat{x}$ of the observer (14.19) tracks the state x of the system (14.21), with bounded errors. Also, we will show that $\hat{x}$ tracks the state x_{ref} of the reference model (14.10) with bounded errors. Then, we will be able to prove that x tracks x_{ref} and z tracks z_{cmd}, with bounded errors. This is our design strategy.

We begin by choosing adaptive laws for $\hat{\bar{\Theta}}$ so that $\hat{x}$ tracks x, with bounded errors and in the presence of the system uncertainties. Let

$$e_x = \hat{x} - x \tag{14.22}$$

denote the state observation error. Note that e_x is not available, as the system measurement. However, the output observation error

$$e_y = \hat{y} - y = C\,(\hat{x} - x) = C\,e_x \tag{14.23}$$

represents the known (i.e., measured online) quantity. Subtracting (14.21) from (14.19) gives the observer error dynamics:

$$\dot{e}_x = (A - L_v\,C)\,e_x + B\,\Lambda\left(\hat{\bar{\Theta}}^T\,\bar{\Phi}(\hat{x},\,u_{bl}) - \bar{\Theta}^T\,\bar{\Phi}(x,\,u_{bl})\right) \tag{14.24}$$

We shall choose the observer gain matrix L_v to represent the steady-state Kalman filter gain

$$L_v = P_v\,C^T\,R_v^{-1} \tag{14.25}$$

where $P_v = P_v^T > 0$ is the unique solution of the ARE

$$P_v\,(A + \eta\,I_{n\times n})^T + (A + \eta\,I_{n\times n})\,P_v - P_v\,C^T\,R_v^{-1}\,C\,P_v + Q_v = 0 \tag{14.26}$$

$\eta > 0$ is a positive constant (defines a prescribed degree of stability), the weight matrices $(Q_v,\ R_v)$ are

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\,\bar{B}^T, \quad R_v = \frac{v}{v+1}\,R_0 \tag{14.27}$$

with a symmetric positive semidefinite $Q_0 \in R^{n\times n}$, a symmetric positive definite $R_0 \in R^{p\times p}$, and with

$$\bar{B} = (\,B \quad B_2\,) \tag{14.28}$$

where $B_2 \in R^{n\times(p-m)}$ is a constant matrix, selected such that the following two relations take place:

$$\det(C\,\bar{B}) \neq 0, \quad \text{zeros}\left[C\,(s\,I_{n\times n} - A)^{-1}\,\bar{B}\right] \in \mathbb{C}^- \tag{14.29}$$

As we have previously mentioned, calculation of B_2 to satisfy (14.29) can be achieved by solving the squaring-up problem [16] for the original triplet $(A,\ B,\ C)$, where due to Assumption 14.1, the output matrix C has more rows than the number of columns in B.

It is easy to verify that the ARE (14.26) possesses the unique symmetric positive definite solution P_v, for any positive η. Furthermore, because of (14.26), the observer matrix

$$A_v = A - L_v C = A - P_v C^T R_v^{-1} C \tag{14.30}$$

satisfies

$$\begin{aligned} &P_v \underbrace{\left(A - \underbrace{P_v C^T R_v^{-1}}_{L_v} C \right)^T}_{A_v} + \underbrace{\left(A - \underbrace{P_v C^T R_v^{-1}}_{L_v} C \right)}_{A_v} P_v \\ &+ P_v C^T R_v^{-1} C P_v + Q_v + 2\eta P_v = 0 \end{aligned} \tag{14.31}$$

or, equivalently,

$$P_v A_v^T + A_v P_v = -P_v C^T R_v^{-1} C P_v - Q_v - 2\eta P_v < 0 \tag{14.32}$$

and therefore, A_v is Hurwitz for any $v > 0$.

Based on the material from Sect. 14.2, one can show that the matrix inverse $\tilde{P}_v = P_v^{-1}$ exists for any $v \geq 0$, and the asymptotic relation

$$\tilde{P}_v \bar{B} = C^T R_0^{-\frac{1}{2}} W + \mathrm{O}(v) \tag{14.33}$$

holds uniformly for any $v > 0$. In (14.33),

$$W = (U\ V)^T \tag{14.34}$$

the two unitary matrices, U and V, are defined by the singular value decomposition

$$\bar{B}^T C^T R_0^{-\frac{1}{2}} = U \Sigma V \tag{14.35}$$

and Σ is the diagonal matrix of the corresponding singular values.

We have assumed that the number of the system outputs exceeds the number of inputs, that is, $p > m$. Let $S = \begin{pmatrix} I_{m\times m} & 0_{(p-m)\times m} \end{pmatrix}$. Then, from (14.33), we get

$$\tilde{P}_v B = C^T R_0^{-\frac{1}{2}} W S^T + \mathrm{O}(v) \tag{14.36}$$

In addition, the following relation holds:

$$A_v^T \tilde{P}_v + \tilde{P}_v A_v = -C^T R_v^{-1} C - \tilde{P}_v Q_v \tilde{P}_v - 2\eta \tilde{P}_v < 0 \tag{14.37}$$

Together, (14.36) and (14.37) imply that the transfer function

$$\underbrace{\left[B^T \tilde{P}_v \left(s I_{n\times n} - A_v\right)^{-1} B\right]}_{G_v(s)} \underset{v\to 0}{\rightarrow} \underbrace{\left[S W R_0^{-\frac{1}{2}} C \left(s I_{n\times n} - A_v\right)^{-1} B\right]}_{G_0(s)} \tag{14.38}$$

becomes strictly positive real (SPR) [3, 5, 19, 20], asymptotically as $v \to 0$.

Let us make an interesting remark regarding (14.38): The squaring-up approach, coupled with the ARE asymptotic properties, gives a unitary matrix $W \in R^{m\times m}$ in (14.34), such that the dynamics, from the system original input u to a linear combination of the original outputs, become "almost" SPR, for a sufficiently small v. So in essence, we have developed a constructive procedure to shape the transmission zeros of the exponentially stable transfer function $G_v(s)$ and make them approach $\mathbb{C}^-$ asymptotically, as $v \to 0$.

Next, we define the matrix of parameter estimation errors:

$$\Delta\bar{\Theta} = \hat{\bar{\Theta}} - \bar{\Theta} \tag{14.39}$$

Adding and subtracting $\hat{\bar{\Theta}}^T \bar{\Phi}(\hat{x}, u_{bl})$, the observer error dynamics (14.24) can be written as

$$\dot{e}_x = A_v e_x + B\Lambda \left[\Delta\bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) + \underbrace{\bar{\Theta}^T \left(\bar{\Phi}(\hat{x}, u_{bl}) - \bar{\Phi}(x, u_{bl})\right)}_{g(\hat{x}, x, u_{bl})}\right] \tag{14.40}$$

or, equivalently,

$$\dot{e}_x = A_v e_x + B\Lambda \left[\Delta\bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) + g(\hat{x}, x, u_{bl})\right] \tag{14.41}$$

Based on (14.15), it is straightforward to compute an upper bound

$$\|g(\hat{x}, x, u_{bl})\| = \left\|\bar{\Theta}^T \left(\bar{\Phi}(\hat{x}, u_{bl}) - \bar{\Phi}(x, u_{bl})\right)\right\| \leq \underbrace{\left(\bar{\Theta}_{\max} L_\Phi\right)}_{k_g} \|e_x\| = k_g \|e_x\| \tag{14.42}$$

where $k_g > 0$ represents a known computable constant and $\bar{\Theta}_{\max}$ is the known upper bound for $\bar{\Theta}$.

The design task now is to choose adaptive laws for $\hat{\bar{\Theta}}$ so that the observer error e_x becomes small, in finite time. The main challenge here is to construct adaptive laws based on the system output information. Hence, we consider the following Lyapunov function candidate:

$$V\left(e_x,\ \Delta\bar{\Theta}\right) = e_x^T \tilde{P}_v\, e_x + \text{trace}\left(\Lambda\, \Delta\bar{\Theta}^T\, \Gamma_\Theta^{-1}\, \Delta\bar{\Theta}\right) \tag{14.43}$$

where $\Gamma_\Theta = \Gamma_\Theta^T > 0$ is the adaptation rate. Using (14.37), the time derivative of $V\left(e,\ \Delta\bar{\Theta}\right)$, along the trajectories of (14.41), can be evaluated:

$$\begin{aligned}\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) = &-e_x^T \left(C^T\, R_v^{-1}\, C + \tilde{P}_v\, Q_v\, \tilde{P}_v + 2\,\eta\, \tilde{P}_v\right) e_x \\ &+ 2\, e_x^T\, \tilde{P}_v\, B\, \Lambda \left(\Delta\bar{\Theta}^T\, \bar{\Phi}(\hat{x},\ u_{bl}) + g(\hat{x},\ x,\ u_{bl})\right) \\ &+ 2\,\text{trace}\left(\Lambda\, \Delta\bar{\Theta}^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\bar{\Theta}}}\right)\end{aligned} \tag{14.44}$$

With the weight matrices from (14.27), we get

$$\begin{aligned}\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) = &-\left(1+\frac{1}{v}\right) e_y^T\, R_0^{-1}\, e_y - e_x^T\, \tilde{P}_v\, Q_0\, \tilde{P}_v\, e_x \\ &-\left(1+\frac{1}{v}\right) \left\|B^T \tilde{P}_v\, e_x\right\|^2 - 2\,\eta\, e_x^T\, \tilde{P}_v\, e_x + 2\, e_x^T\, \tilde{P}_v\, B\, \Lambda\, g \\ &+2\, e_x^T\, \tilde{P}_v\, B\, \Lambda\, \Delta\bar{\Theta}^T\, \bar{\Phi}(\hat{x},\ u_{bl}) + 2\,\text{trace}\left(\Lambda\, \Delta\bar{\Theta}^T\, \Gamma_\Theta^{-1}\, \dot{\hat{\bar{\Theta}}}\right)\end{aligned} \tag{14.45}$$

Substituting (14.36) into the sixth term and merging it with the seventh results in

$$\begin{aligned}\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) = &-\left(1+\frac{1}{v}\right) e_y^T\, R_0^{-1}\, e_y - e_x^T\, \tilde{P}_v\, Q_0\, \tilde{P}_v\, e_x \\ &-\left(1+\frac{1}{v}\right) \left\|B^T \tilde{P}_v\, e_x\right\|^2 - 2\,\eta\, e_x^T\, \tilde{P}_v\, e_x + 2\, e_x^T\, \tilde{P}_v\, B\, \Lambda\, g \\ &+ 2\,\text{trace}\left(\Lambda\, \Delta\bar{\Theta}^T \left\{\Gamma_\Theta^{-1}\, \dot{\hat{\bar{\Theta}}} + \bar{\Phi}(\hat{x},\ u_{bl})\, \underbrace{\left(e_x^T\, C^T\right)}_{e_y^T}\, R_0^{-\frac{1}{2}}\, W\, S^T\right\}\right) \\ &+ 2\, e_x^T\, \text{O}(v)\, \Lambda \left(\Delta\bar{\Theta}^T\, \bar{\Phi}(\hat{x},\ u_{bl}) + g\right)\end{aligned} \tag{14.46}$$

Adaptive law dynamics with the Projection Operator modification [22]

$$\dot{\hat{\bar{\Theta}}} = \text{Proj}\left(\hat{\bar{\Theta}}, -\Gamma_{\bar{\Theta}}\, \bar{\Phi}(\hat{x},\, u_{bl})\, e_y^T\, R_0^{-\frac{1}{2}}\, W\, S^T\right) \tag{14.47}$$

can be chosen such that the estimated parameters remain uniformly bounded, $\hat{\bar{\Theta}}(t) \in \Omega_{\bar{\Theta}} = \left\{\bar{\Theta} : \quad \left\|\bar{\Theta}\right\| \le \bar{\Theta}_{\max}\right\}$, for all $t \ge 0$. Then, (14.46) can be upper-bounded as

$$\begin{aligned}\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) \le & -2\,\eta\,\lambda_{\min}\left(\tilde{P}_v\right)\|e_x\|^2 - \lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\tilde{P}_v\right)\|e_x\|^2 \\ & - \left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)\left\|B^T\tilde{P}_v\, e_x\right\|^2 \\ & + 2\,\Lambda_{\max}\, k_g \left\|B^T\tilde{P}_v\, e_x\right\|\|e_x\| \\ & + 2\,v\,\|e_x\|\,k\,\Lambda_{\max}\left(\Delta\bar{\Theta}_{\max}\left\|\bar{\Phi}(\hat{x},\, u_{bl})\right\| + k_g\,\|e_x\|\right)\end{aligned} \tag{14.48}$$

where $k > 0$ is a constant, $0 < \|\Lambda\| \le \Lambda_{\max}$, and $0 \le \left\|\Delta\bar{\Theta}(t)\right\| \le \Delta\bar{\Theta}_{\max}$, with a finite constant $\Delta\bar{\Theta}_{\max} \le 2\,\bar{\Theta}_{\max}$.

Next, we define $w = \left\|B^T\,\tilde{P}_v\, e_x\right\|$ and rewrite (14.48):

$$\begin{aligned}&\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) \le -\left(2\,\eta + \lambda_{\min}(Q_0)\,\lambda_{\min}\left(\tilde{P}_v\right)\right)\ \lambda_{\min}\left(\tilde{P}_v\right)\ \|e_x\|^2 \\ &- \left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)\,w^2 + 2\,\Lambda_{\max}\,k_g w\,\|e_x\| \\ &+ 2\,v\,\|e_x\|\,k\,\Lambda_{\max}\left(\Delta\bar{\Theta}_{\max}\left\|\bar{\Phi}(\hat{x},\, u_{bl})\right\| + k_g\,\|e_x\|\right)\end{aligned} \tag{14.49}$$

Since $\lambda_{\min}\left(\tilde{P}_v\right) \ge \lambda_{\min}\left(\tilde{P}_0\right) > 0$, we get

$$\begin{aligned}&\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) \le -\left(2\,\eta + \lambda_{\min}(Q_0)\,\lambda_{\min}\left(\tilde{P}_0\right)\right)\ \lambda_{\min}\left(\tilde{P}_0\right)\ \|e_x\|^2 \\ &- \left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)\,w^2 + 2\,\Lambda_{\max}\,k_g w\,\|e_x\| \\ &+ 2\,v\,\|e_x\|\,k\,\Lambda_{\max}\left(\Delta\bar{\Theta}_{\max}\left\|\bar{\Phi}(\hat{x},\, u_{bl})\right\| + k_g\,\|e_x\|\right)\end{aligned} \tag{14.50}$$

An upper bound for the norm of the extended regressor vector can be computed. Since A_{ref} is Hurwitz in (14.20), then for any norm-bounded z_{cmd}, there must exist constants d_1 and d_2 (both may depend on the norm upper bound of z_{cmd}), such that $\|\hat{x}\| \le d_1 + d_2\,\|e_x\|$. Consequently, it is not difficult to show that

$$\left\|\bar{\Phi}(\hat{x},\, u_{bl})\right\| \le b_1 + b_2\,\|e_x\| \tag{14.51}$$

for some positive constants b_1 and b_2. In this case,

$$\begin{aligned}\dot{V}(e_x,\ \Delta\bar{\Theta}) \le &-\left(1+\frac{1}{\nu}\right)\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2\\ &-\left[\left(2\,\eta+\lambda_{\min}(Q_0)\,\lambda_{\min}\left(\tilde{P}_0\right)\right)\ \lambda_{\min}\left(\tilde{P}_0\right)-2\,\nu\,k\,\Lambda_{\max}\left(\Delta\bar{\Theta}_{\max}\,b_2+k_g\right)\right]\ \|e_x\|^2\\ &-\left(1+\frac{1}{\nu}\right)\ w^2+2\,\Lambda_{\max}\,k_g w\,\|e_x\|+2\,\nu\,\Lambda_{\max}\,k\,\Delta\bar{\Theta}_{\max}\,b_1\,\|e_x\|\end{aligned} \tag{14.52}$$

Let us introduce the following notation:

$$\begin{aligned}c_1 =&\lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\tilde{P}_0\right)-2\,\nu\,k\,\Lambda_{\max}\left(\Delta\Theta_{\max}\,b_2+k_g\right),\quad c_2=\Lambda_{\max}\,k_g\\ c_3 =&1+\frac{1}{\nu},\quad c_4=\nu\,\Lambda_{\max}\,k\,\Delta\Theta_{\max}\,b_1\end{aligned} \tag{14.53}$$

and rewrite (14.52) as

$$\begin{aligned}\dot{V}(e_x,\ \Delta\bar{\Theta}) \le &-c_3\,\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2-2\,\eta\,\|e_x\|^2\\ &-\underbrace{\left[c_1\ \|e_x\|^2-2\,c_2\,\|e_x\|\,w+c_3\,w^2-2\,c_4\,\|e_x\|\right]}_{\varphi(\|e_x\|,\,w)}\\ =&-c_3\,\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2-2\,\eta\,\|e_x\|^2-\varphi(\|e_x\|,\,w)\end{aligned} \tag{14.54}$$

One can show constructively the existence of a sufficiently small $\nu_0>0$ such that for all $0<\nu\le\nu_0$ the function $\varphi(\|e_x\|,\,w)$ in (14.54) has the unique nonpositive global minimum:

$$\varphi_{\min}(\nu)=\min_{\zeta=\begin{pmatrix}\|e_x\| & w\end{pmatrix}^T}\varphi(\zeta)=\mathrm{O}\left(\nu^2\right)\le 0 \tag{14.55}$$

Before proceeding further, let us prove this fact.

Lemma 14.1. *There exists $\nu_0>0$, such that for all $0<\nu\le\nu_0$, the function φ in (14.54) has the unique nonpositive global minimum:*

$$\varphi_{\min}(\nu)=\begin{cases}0 & ,\quad b_1=0\\ \min\limits_{\zeta=\begin{pmatrix}\|e_x\|\\ w\end{pmatrix}}\varphi(\zeta)=\mathrm{O}\left(\nu^2\right)<0, & b_1>0.\end{cases} \tag{14.56}$$

Proof of Lemma 14.1. The function

$$\varphi(\|e_x\|,\, w) = c_1\ \|e_x\|^2 - 2\,c_2\,\|e_x\|\,w + c_3\,w^2 - 2\,c_4\,\|e_x\| \tag{14.57}$$

in (14.54) can be written in matrix form

$$\begin{aligned}\varphi(\|e_x\|, w) &= \underbrace{\begin{pmatrix}\|e_x\| & w\end{pmatrix}}_{\zeta^T}\underbrace{\begin{pmatrix}c_1 & -c_2\\ -c_2 & c_3\end{pmatrix}}_{C}\underbrace{\begin{pmatrix}\|e_x\|\\ w\end{pmatrix}}_{\zeta} - 2\underbrace{\begin{pmatrix}\|e_x\| & w\end{pmatrix}}_{\zeta^T}\underbrace{\begin{pmatrix}c_4\\ 0\end{pmatrix}}_{b}\\ &= \zeta^T\,C\,\zeta - 2\,\zeta^T\,b\end{aligned} \tag{14.58}$$

From (14.58), it follows that $\varphi(\|e_x\|,\, w)$ has the unique global minimum if and only if the matrix

$$C = \begin{pmatrix}c_1 & -c_2\\ -c_2 & c_3\end{pmatrix} \tag{14.59}$$

is strictly positive definite, which in turn is equivalent to

$$c_1 > 0\,, \quad \det C = c_1\,c_3 - c_2^2 > 0 \tag{14.60}$$

The first inequality in (14.60) can be enforced if v is a sufficiently small positive constant. Specifically, it suffices to choose $0 < v < v_1$, where

$$v_1 = \frac{\lambda_{\min}(Q_0)\,\lambda_{\min}^2(\tilde{P}_0)}{2\,k\,\Lambda_{\max}\left(\Delta\bar{\Theta}_{\max}\,k_\Phi\,C_2 + k_g\right)} \tag{14.61}$$

The second relation in (14.60) can be written as

$$\begin{aligned}&\det C = c_1\,c_3 - c_2^2\\ &= \left(1 + \frac{1}{v}\right)\left[\lambda_{\min}(Q_0)\,\lambda_{\min}^2(\tilde{P}_0) - 2\,v\,k\,\Lambda_{\max}\left(\Delta\Theta_{\max}\,k_\Phi\,C_2 + k_g\right)\right]\\ &- \Lambda_{\max}^2\,k_g^2 = \mathrm{O}\left(\frac{1}{v}\right)\end{aligned} \tag{14.62}$$

Observe that $\lim_{v\to 0}[\det C] = +\infty$. So, for a sufficiently small positive $0 < v < v_2$, $\det C > 0$. Finally, choosing $v_0 = \min(v_1,\, v_2)$ enforces both inequalities in (14.60) and as a result guarantees uniqueness of the global nonpositive minimum for the function $\varphi(\zeta)$.

The location of the function minimum ξ_0 and the minimum value itself $\varphi_{\min} = \varphi(\zeta_0)$ can be easily computed by differentiating $\varphi(\zeta)$ with respect to ξ, setting the gradient vector to 0, solving for ξ_0, and then substituting the latter into (14.58):

$$\begin{aligned}&\left[\nabla\varphi(\zeta) = 2\,C\,\zeta - 2\,b = 0\right] \Rightarrow \left[\zeta_0 = C^{-1}\,b\right]\\ &\Rightarrow \left[\varphi_{\min} = \varphi(\zeta_0) = -\left(b^T\,C^{-1}\,b\right) < 0\right]\end{aligned} \tag{14.63}$$

This proves (14.55). Furthermore, because of (14.58) and (14.63), we get

$$\begin{aligned}\zeta_0 &= \begin{pmatrix}\|e_x\|_0\\ w_0\end{pmatrix} = C^{-1}\,b = \frac{1}{\det C}\begin{pmatrix}c_3 & c_2\\ c_2 & c_1\end{pmatrix}\begin{pmatrix}c_4\\ 0\end{pmatrix} = \frac{c_4}{c_1\,c_3 - c_2^2}\begin{pmatrix}c_3\\ c_2\end{pmatrix}\\ \varphi_{\min} &= -b^T\,C^{-1}\,b = -\begin{pmatrix}c_4 & 0\end{pmatrix}\frac{1}{\det C}\begin{pmatrix}c_3 & c_2\\ c_2 & c_1\end{pmatrix}\begin{pmatrix}c_4\\ 0\end{pmatrix} = -\frac{c_3\,c_4^2}{c_1\,c_3 - c_2^2}\end{aligned} \tag{14.64}$$

Due to (14.53), it easy to see that as $v \to 0$,

$$|\varphi_{\min}(v)| = \frac{c_3\,c_4^2}{c_1\,c_3 - c_2^2} = \frac{\mathrm{O}(v)}{\mathrm{O}\left(\frac{1}{v}\right)} = \mathrm{O}\left(v^2\right) \underset{v\to 0}{\to} 0 \tag{14.65}$$

So, if $b_1 = 0$, then $c_4 = 0$, and consequently, $|\varphi_{\min}(v)| = 0$. The proof of the lemma is complete. □

We can now return to the design of an output feedback adaptive control law. Continuing from (14.54) gives

$$\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) \le -\left(1 + \frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2 - 2\,\eta\,\|e_x\|^2 + |\varphi_{\min}(v)| \tag{14.66}$$

According to (14.55), $|\varphi_{\min}(v)| = \mathrm{O}(v^2) \underset{v\to 0}{\to} 0$, and therefore, $\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right) < 0$ outside of the compact set

$$\Omega_{e_x} = \left\{e_x:\quad \|e_x\|^2 \le \frac{|\varphi_{\min}(v)|}{2\,\eta} = r_{v,\eta}^2 = \mathrm{O}\left(\frac{v^2}{\eta}\right)\right\} \tag{14.67}$$

At the same time, the adaptive laws (14.47) ensure uniform boundedness of the estimated parameters $\hat{\bar{\Theta}}$ and of the corresponding estimation errors $\Delta\bar{\Theta}$. Consequently, $\dot{V}\left(e_x,\ \Delta\bar{\Theta}\right)$ is negative outside of the compact set $\Omega = \Omega_{e_x} \times \Omega_{\bar{\Theta}}$. This fact proves UUB of the observer error dynamics (14.41) [5]. Additionally, (14.67) implies that for a constant positive η, the radius $r_{v,\eta}$ of Ω_{e_x} decreases, as $v \to 0$. In other words, for any given constant η, the UUB tracking property "tends to" global asymptotic tracking, at the rate of $\mathrm{O}(v)$.

Let us now define the observer tracking error

$$e = \hat{x} - x_{ref} \tag{14.68}$$

subtract (14.10) from (14.20), and compute the observer tracking error dynamics:

$$\dot{e} = A_{ref}\,e - L_v\,e_y \tag{14.69}$$

Because of (14.5), (14.25), (14.27), and (14.67),

$$\begin{aligned}\dot{e} &= A_{ref}\, e + \left(1 + \frac{1}{v}\right)(P_0 + \mathrm{O}(v))\,\mathrm{O}\left(\frac{v}{\sqrt{\eta}}\right) \\ &= A_{ref}\, e + \mathrm{O}\left(\frac{v}{\sqrt{\eta}}\right) + \mathrm{O}\left(\frac{1}{\sqrt{\eta}}\right)\end{aligned} \tag{14.70}$$

and so, the observer tacking error $e(t)$ is uniformly bounded. Moreover,

$$\begin{aligned}\left\|x - x_{ref}\right\| &= \left\|x - \hat{x} + \hat{x} - x_{ref}\right\| \leq \underbrace{\left\|x - \hat{x}\right\|}_{\|e_x\|} + \underbrace{\left\|\hat{x} - x_{ref}\right\|}_{\|e\|} \\ &= \mathrm{O}\left(\frac{v}{\sqrt{\eta}}\right) + \mathrm{O}\left(\frac{1}{\sqrt{\eta}}\right)\end{aligned} \tag{14.71}$$

Consequently, as $t \to \infty$, the state tracking error either asymptotically converges to the origin, or it tends (in the UUB sense) to a compact neighborhood of the origin, with the neighborhood radius that can be made small by choosing a small v and a large η.

We have shown that the system state $x(t)$ tracks the state of the reference model $x_{ref}(t)$ with bounded errors. Therefore, the system-regulated output will $z(t) = C_z\, x(t)$ track the reference model regulated output $z_{ref}(t) = C_z\, x_{ref}(t)$. At the same time, $z_{ref}(t)$ tracks its commanded value $z_{cmd}(t)$. Then, $z(t)$ must track $z_{cmd}(t)$. This argument completes the design and stability analysis of the adaptive output tracking controller. We now summarize our formally derived results as Theorem 14.1.

Theorem 14.1. *Consider the MIMO system dynamics (14.7). Suppose Assumption 14.1 holds. Consider the state observer (14.19), whose feedback gain matrix L_v satisfies (14.25) and (14.26), with positive constants $(v,\ \eta)$, symmetric positive definite matrices $(Q_v,\ R_v)$ from (14.27) with $\bar{B} = (B \quad B_2)$, and with a constant matrix $B_2 \in R^{n\times(p-m)}$ chosen such that $\det(C\bar{B}) \neq 0$ and $C\,(s I_{n\times n} - A)^{-1}\bar{B}$ is minimum-phase. Then, there exists a sufficiently small positive parameter* v, *such that the (baseline + adaptive) dynamic output feedback controllers (14.11), (14.17), and (14.18), with the projection-based adaptive laws (14.47), enforce UUB of the closed-loop system trajectories. Moreover, the system-regulated output* z *tracks any bounded time-varying command z_{cmd} with bounded errors, while all other signals in the closed-loop system remain bounded. At the same time, $\hat{x}$ recovers the system state* x, *which in turn tracks the state x_{ref} of the reference model (14.10), with estimation and tracking errors entering in finite time a neighborhood of the origin, whose radius is of order $\mathrm{O}\left(\frac{v}{\sqrt{\eta}}\right) + \mathrm{O}\left(\frac{1}{\sqrt{\eta}}\right)$.* ■

A design synopsis, encapsulating the system dynamics and control equations, is given in Table 14.1.

Table 14.1 Adaptive output feedback control design summary

Open-loop plant	$\dot{x} = A\,x + B\,\Lambda\left(u + \Theta^T\,\Phi(x)\right) + B_{ref}\,z_{cmd}$
Measured and regulated outputs	$y = C\,x, \quad z = C_z\,x$
State observer	$\dot{\hat{x}} = A_{ref}\,\hat{x} + L_v\,(y - \hat{y}) + B_{ref}\,z_{cmd}, \quad \hat{y} = C\,\hat{x}$
Observer gain	$L_v = P_v\,C^T\,R_v^{-1}$
Squared-up B-matrix	$\bar{B} = (\,B \quad B_2\,) \Rightarrow \left\{\det(C\,\bar{B}) \neq 0\ \text{zeros}\left[C\,(s\,I_{n\times n} - A)^{-1}\,\bar{B}\right] \in \mathbb{C}^-\right.$
ARE weights	$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\,\bar{B}^T, \quad R_v = \frac{v}{v+1}\,R_0$
Algebraic Riccati equation	$P_v\,(A + \eta\,I_{n\times n})^T + (A + \eta\,I_{n\times n})\,P_v - P_v\,C^T R_v^{-1} C\,P_v + Q_v = 0$
Output tracking error	$e_y = \hat{y} - y$
Baseline control	$u_{bl} = -K_{LQR}^T\,\hat{x}$
Extended regressor	$\bar{\Phi}(x,\,u_{bl}) = \left(\,\Phi^T(x) \quad u_{bl}^T\,\right)^T$
Output selection matrix for adaptive laws	$S = \left(\,I_{m\times m} \quad 0_{(p-m)\times m}\,\right)$
Singular value decomposition	$B^T\,C^T\,R_0^{-\frac{1}{2}} = U\,\Sigma\,V$
Unitary matrix	$W = (U\,V)^T$
Projection-based MRAC laws	$\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta}, -\Gamma_\Theta\,\bar{\Phi}(\hat{x},\,u_{bl})\,e_y^T\,R_0^{-\frac{1}{2}}\,W\,S^T\right)$
Adaptive increment	$u_{ad} = -\hat{\Theta}^T\,\bar{\Phi}(\hat{x},\,u_{bl})$
Total control input	$u = u_{bl} + u_{ad}$

14.5 Adaptive Flight Control of a Flexible Transport Aircraft

In this study, we shall design robust and adaptive output feedback controllers for longitudinal dynamics of a large transport aircraft with flexible structure. The aircraft data (wings-level cruise configuration) are taken from [23], where all linear displacements, velocities, and accelerations are given in meters (m), m/s, and m/s^2, while all angles and angular rates are in radians (rad) and rad/s, respectively.

The vehicle model (plant) includes a short-period mode and four structural bending modes. Each mode is described by a complex-conjugate pair of eigenvalues. The system state $x_p \in R^{10\times 1}$ consists of the vehicle angle of attack α, pitch rate q, and four structural mode positions $(\xi_i)_{i=1,\ldots,4}$ and their rates $(\eta_i)_{i=1,\ldots,4}$:

$$x_p = (\,\alpha \quad q \quad \zeta_1 \quad \eta_1 \quad \zeta_2 \quad \eta_2 \quad \zeta_3 \quad \eta_3 \quad \zeta_4 \quad \eta_4\,)^T$$

There are two horizontal control surfaces available: (1) the elevator (δ_e), (an aft-mounted tail surface) and (2) the canard (δ_c) (a forward-mounted surface). The aircraft dynamics also depend on the vertical gust velocity vector $w_g \in R^{3\times 1}$. The gust enters the plant at three different locations, along the vehicle center line. So, the aircraft longitudinal dynamics

$$\dot{x}_p = A_p\, x_p + B_p\, \delta + B_g\, w_g$$

are driven by the two-dimensional control input $\delta = (\,\delta_e \quad \delta_c\,)^T$ and by the three-dimensional gust input w_g.We have modified the original data to make the open-loop system unstable in pitch. This necessitates a control action to restore and maintain basic stability of the vehicle. The open-loop plant matrices are

$$A_p = \begin{pmatrix} -1.60 & 1 & -1.1811 & -0.1181 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6.57 & -2.446 & -1.8130 & 1.1805 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7.196 & -0.445 & -56.82 & -5.53 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1.349 & 0.2466 & 0 & 0 & -231.52 & -1.712 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2.093 & 0.242 & 0 & 0 & 0 & 0 & -408.86 & -2.679 & -10.71 & -0.518 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.3073 & 0.05588 & 0 & 0 & 0 & 0 & -1.24 & -0.176 & -390.1 & -0.474 \end{pmatrix}$$

$$B_p = \begin{pmatrix} -0.070 & 3.726 & 0 & 0.572 & 0 & -0.465 & 0 & -0.582 & 0 & -0.112 \\ -0.006 & -0.28 & 0 & 0.019 & 0 & -0.054 & 0 & -0.0532 & 0 & 0.035 \end{pmatrix}^T$$

$$B_g = \begin{pmatrix} 0.0042 & 0.06 & 0 & 0.0105 & 0 & 0.0065 & 0 & -0.0045 & 0 & -0.0021 \\ 0.0037 & -0.0417 & 0 & 0.0393 & 0 & 0.0039 & 0 & 0.0101 & 0 & -0.0009 \\ 0.0012 & -0.056 & 0 & -0.0086 & 0 & 0.0059 & 0 & 0.0064 & 0 & 0.0012 \end{pmatrix}^T$$

The system output measurements include the pitch rate q and vertical accelerations $(a_{z\,i})_{i=1,2,3}$ from three distinct nodes on the vehicle centerline. The pitch rate is measured near the vehicle center of gravity (cg). The first vertical acceleration is taken near the tip of the aircraft nose, the second is near cg, and the third acceleration measurement comes from an aft cg location. These are the same three locations where the vertical gust w_g enters the system dynamics. So, the system measured output vector is

$$y_p = C_p\, x_p + D_p\, u$$

with

$$y_p = (\,q \quad a_{z\,1} \quad a_{z\,2} \quad a_{z\,3}\,)^T$$

and the output matrices shown below:

$$C_p = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -32.65 & -28.04 & 49.93 & 2.37 & -1405.41 & -10.39 & -1707.61 & -13.09 & -4472.18 & -7.53 \\ -11.69 & -35.68 & 65.02 & 6.27 & 117.33 & 0.87 & 895.25 & 5.94 & 191.7 & 1.34 \\ 3.68 & -38.04 & 153.63 & 17.68 & -739.42 & -5.47 & -821.74 & -5.66 & 672.66 & -1.83 \end{pmatrix}$$

$$D_p = \begin{pmatrix} 0 & -1.02 & 0 & -1.2 \\ 0 & -12.55 & -1.23 & 3.69 \end{pmatrix}^T$$

The system-regulated output is the vehicle pitch rate, q, shown below:

$$z = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{C_{p\,reg}} x_p = C_{p\,reg}\, x_p = q$$

In order to regulate z, we shall blend the two control surfaces, the elevator δ_e and the canard δ_c, and create a single longitudinal virtual control input:

$$u = 0.5\,(\delta_e - \delta_c)$$

Such a control mixing is standard in aerospace systems. It is called "control allocation." In order to incorporate u into the system dynamics, we introduce the control allocation matrix $G = \begin{pmatrix} 0.5 & -0.5 \end{pmatrix}^T$ and arrive at the plant dynamics

$$\dot{x}_p = A_p\, x_p + \left(B_p\, G\right)\, u + B_g\, w_g$$

with the virtual control input u and with its corresponding B-matrix:

$$\left(B_p\, G\right) = \begin{pmatrix} -0.032 & 2.003 & 0 & 0.2765 & 0 & -0.2055 & 0 & -0.2644 & 0 & -0.0385 \end{pmatrix}^T$$

Our first step is to create the desired reference model dynamics, and our tool of choice is the LQG/LTR method. In particular, we shall design a baseline output feedback controller $u(= u_{bl})$, such that q adequately tracks its commanded value $q_{cmd}(= z_{cmd})$, while operating only on the system output measurements. In order to do that, we augment the linear plant dynamics with the integrated pitch tracking error

$$\dot{e}_{q\,I} = q - q_{cmd}$$

and arrive at the extended open-loop system

$$\underbrace{\begin{pmatrix} \dot{e}_{q\,I} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & C_{p\,reg} \\ 0 & A_p \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} e_{q\,I} \\ x_p \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0 \\ B_p\, G \end{pmatrix}}_{B} u + \underbrace{\begin{pmatrix} -1 \\ 0 \end{pmatrix}}_{B_{cmd}} z_{cmd} + \underbrace{\begin{pmatrix} 0 \\ B_g \end{pmatrix}}_{B_g} w_g$$

with the output measurements

$$\begin{pmatrix} e_{qI} \\ y_p \end{pmatrix} = y = \underbrace{\begin{pmatrix} 1 & 0_{1\times 10} \\ 0_{4\times 1} & C_p \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} e_{qI} \\ x_p \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0 \\ D_p G \end{pmatrix}}_{D} u = C\,x + D\,u$$

that include the integrated pitch rate tracking error and the original system outputs. Adding the integrated output error will allow us to design a control input without command feedforward connections. In the context of the extended system, the regulated output can be expressed as

$$z = \underbrace{(\,0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\,)}_{C_z} x = C_z\,x$$

We now proceed to the design of an LQR (proportional + integral) (PI) state feedback controller. After several design iterations, we have selected the LQR weights as

$$Q_{lqr} = \text{diag}(\,2 \quad 0 \quad 0 \quad 0 \quad 0.0001 \quad 0 \quad 0.0001 \quad 0 \quad 0.0001 \quad 0 \quad 0.0001\,), \quad R_{lqr} = 1$$

Next, we choose the prescribed degree of stability $\eta = 0.2$, solve the ARE

$$P_{lqr}\,(A + 0.2\,I_{11\times 11}) + (A + 0.2\,I_{11\times 11})^T P_{lqr} - P_{lqr}\,B\,R_{lqr}^{-1}\,B^T\,P_{lqr} + Q_{lqr} = 0$$

for P_{lqr}, compute the LQR state feedback gains

$$K_{lqr} = R_{lqr}^{-1}\,B^T\,P_{lqr}$$

and arrive at the closed-loop poles of

$$A_{ref} = A - B\,K_{lqr}$$

with real parts no greater than -0.2 (we are using the LQR design modification with a prescribed degree of stability).

The next step is the design of a state observer. Note that the extended open-loop plant has one input u and five outputs y. According to our design methodology, we need to square-up the system dynamics, that is to say, we need to add four pseudo-inputs by building a matrix $B_2 \in R^{11\times 4}$ to enforce the square-up conditions:

$$\bar{B} = (\,B \quad B_2\,) \Rightarrow \begin{cases} \det(C\,\bar{B}) \neq 0 \\ \text{zeros}\left[C\,(s\,I_{n\times n} - A)^{-1}\,\bar{B}\right] \in \mathbb{C}^- \end{cases}$$

We could use the method from [16], or we can select B_2 directly to enforce the first square-up condition and then check if the second one holds true. Let us explore the direct approach. Since we need $\det(C\bar{B}) \neq 0$, it seems reasonable to select

$$B_2 = \begin{pmatrix} B_{cmd} & \frac{C_3^T}{\|C_3\|} & \frac{C_4^T}{\|C_4\|} & \frac{C_5^T}{\|C_5\|} \end{pmatrix}$$

form the matrix

$$\bar{B} = \begin{pmatrix} B & B_{cmd} & \frac{C_3^T}{\|C_3\|} & \frac{C_4^T}{\|C_4\|} & \frac{C_5^T}{\|C_5\|} \end{pmatrix} = \begin{pmatrix} B & B_2 \end{pmatrix}$$

and verify that the new system

$$\begin{matrix} & & 5-\text{Inputs} & \\ & & \Downarrow & \\ 5-\text{Outputs} & \Rightarrow & \begin{pmatrix} A & \bar{B} \\ C & 0_{p\times p} \end{pmatrix} & \in R^{16\times 16} \end{matrix}$$

with four fictitious pseudo-control columns in B_2 satisfies the two square-up conditions: (1) The system is minimum-phase, and (2) the relative degree is one.

With the selected matrix $\bar{B}$, we choose the observer weights similar to (14.3),

$$Q = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\,\bar{B}^T, \quad R = \frac{v}{v+1}R_0$$

with

$$Q_0 = I_{11\times 11}, \quad R_0 = 10^6 I_{6\times 6}, \quad v = 0.1$$

solve the ARE

$$P_v A^T + A P_v - P_v C^T R_v^{-1} C P_v + Q = 0$$

for P_v, compute the steady-state Kalman filter gain $L_v = R_v^{-1} C P_v$, and write the state observer dynamics

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + B u_{bl} + B_{cmd} z_{cmd} + L_v (y - \hat{y}) \\ \hat{y} &= C\hat{x} + D u_{bl} \end{aligned}$$

with the baseline control input

$$u_{bl} = -K_{lqr}^T \hat{x}$$

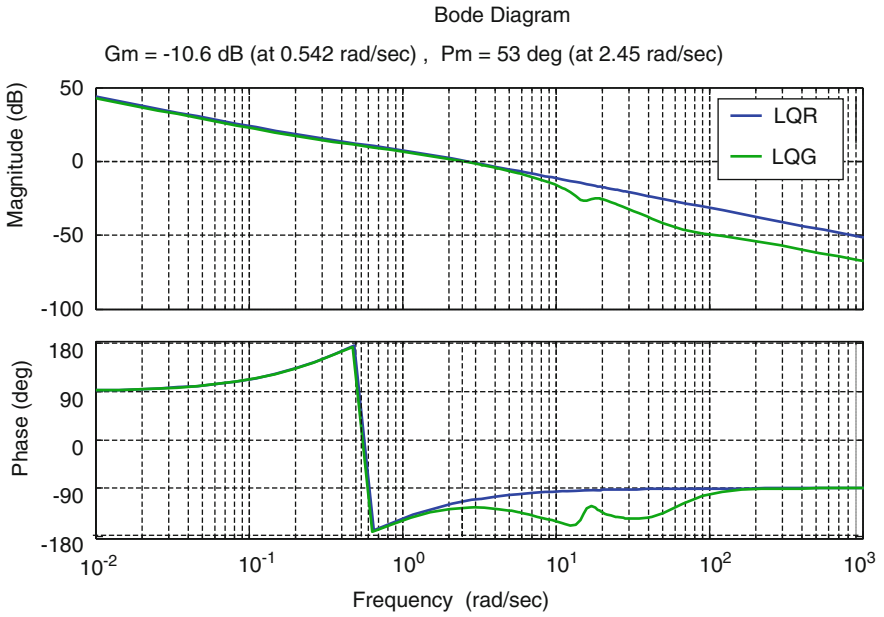


Fig. 14.1 LQR and LQG loop gain margins at virtual control input in Example 14.1

while utilizing the LQR-optimal feedback gains K_{lqr} on the observer states $\hat{x}$.

Let us comment on the choice of $\bar{B}$ and on our selection of the tuning parameter v. As we have previously discussed, $\bar{B}$ turns the original system with one input and five outputs into a (5×5) minimum-phase system. This is the squaring-up procedure for the observer design, where we have added four fictitious inputs into the second through the fifth columns of $\bar{B}$, respectively. The squaring-up modification allows us to recover the LQR state feedback margins and, at the same time, enforce the needed (for adaptive laws) asymptotics (14.6). The latter is achieved by setting the tuning knob v to be sufficiently small. However, if v becomes too small, then the observer gains may grow large, which is undesirable since the system noise sensitivity may increase. This is the trade-off in our design: We must find v small enough but not too small. Also, note that our selection of $(\bar{B},\ v)$ is by no means unique, yet it presents a straightforward way to recover optimal LQR state feedback margins (at the system input), with the assigned crossover frequencies, obtain reasonably small observer gains, and enforce the asymptotic relation (14.6).

With the selected pair $(\bar{B},\ v)$, we recover the gain and phase margins of the optimal LQR state feedback controller (Fig. 14.1).

The extended system gain and phase margins at the virtual input (including the observer dynamics) are very close to those of the LQR state feedback controller. We get a negative gain margin of -10.6 db at 0.542 rad/s, an infinite positive gain margin, and a phase margin of 53 deg, at 2.45 rad/s. These values are quite reasonable for the selected transport aircraft configuration.

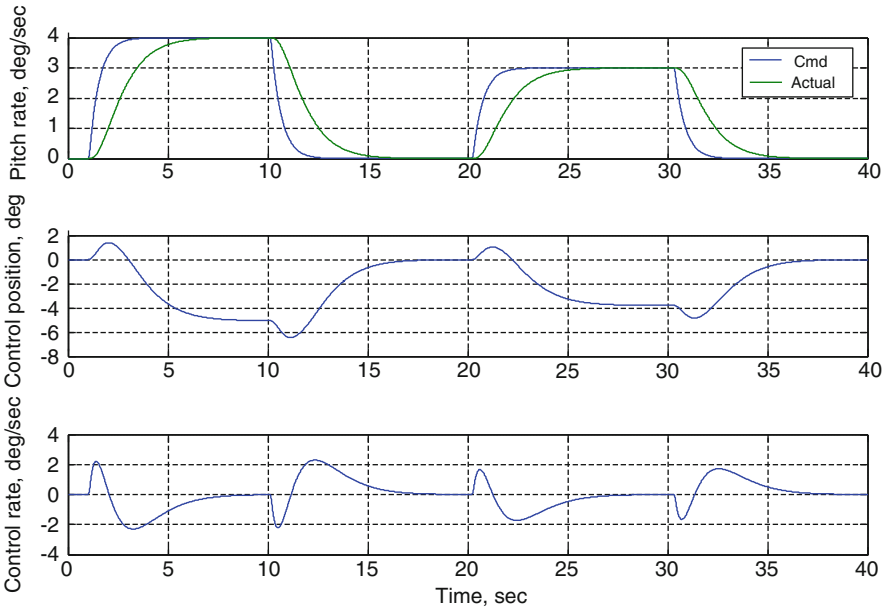


Fig. 14.2 Baseline LQG system tracking (without uncertainties) in Example 14.1

Note that that the loop gain of the LQG controller rolls off faster than the loop gain of the LQR state feedback system. So, the dynamic LQG solution is less sensitive to modeling uncertainties, and it has better disturbance rejection properties than the LQR state feedback. Of course, the faster roll-off at high frequencies in the LQG design is attributed to adding the dynamic state observer.

With the baseline LQG controller in the loop, the closed-loop system becomes

$$\begin{aligned} \dot{x} &= A\,x - B\,K_{lqr}^{T}\,\hat{x} + B_{cmd}\,z_{cmd} + \bar{B}_g\,w_g \\ \dot{\hat{x}} &= \left(A - B\,K_{lqr}^{T}\right)\hat{x} + B_{cmd}\,z_{cmd} + L_v\,(y - \hat{y}) \\ y &= \left(C - D\,K_{lqr}^{T}\right)x, \quad \hat{y} = \left(C - D\,K_{lqr}^{T}\right)\hat{x}, \quad z = C_z\,x \end{aligned},$$

and the resulting closed-loop eigenvalues are placed well within practical bounds that would be representative of a large transport aircraft in a cruise configuration.

Without uncertainties, the closed-loop system tracking performance is satisfactory. Representative data are shown in Fig. 14.2.

The baseline LQG controller forces the regulated output (pitch rate) $z = q$ to track its commanded values, and the required control effort lies well within the bandwidth of a typical aircraft actuation system.

We have also tested the LQG baseline system closed-loop tracking using various command shapes. All the results have shown adequate performance. For all these reasons, the closed-loop baseline system becomes our reference model for adaptive

control to achieve and maintain if and when uncertainties are present in the system dynamics. Specifically, we define the reference model matrices

$$\begin{aligned} A_{ref} &= A - B\,K_{lqr}^T, \quad B_{ref} = B_{cmd} \\ C_{ref} &= C - D\,K_{lqr}^T, \quad D_{ref} = 0_{5\times 1} \end{aligned}$$

and write the reference model dynamics in the form of (14.10):

$$\dot{x}_{ref} = A_{ref}\,x_{ref} + B_{ref}\,z_{cmd}, \quad z_{ref} = C_z\,x_{ref}$$

These are the desired dynamics. In other words, this is exactly how we want our system to respond to external commands. Note that the LQG state observer remains the same, as in (14.20):

$$\dot{\hat{x}} = A_{ref}\,\hat{x} + L_v\,(y - \hat{y}) + B_{ref}\,z_{cmd}$$

Before proceeding further, a remark is in order. As derived, our adaptive MRAC design is applicable to systems whose measured and regulated output have no direct feedforward control connections, that is, our method is applicable to systems with the zero D matrix in both the measurements and the regulated outputs. It is possible to broaden our design methodology to cover systems with feedforward control connections in their outputs. Derivations of such a method are similar to what we have presented in this chapter, and because of their similarity, we choose to omit formal proofs.

Let us now test the baseline system performance in the presence of uncertainties. So, we reduce the baseline controller gains by 75 %, set $\Lambda = 0.5$, and also introduce a matched nonparametric alpha-dependent uncertainty in the form of a Gaussian (an RBF), centered at $\alpha_c = 2$ deg, with the RBF sigma-width set to 0.011636, and with the function peak magnitude of -0.25.

With these uncertainties active and operating under the baseline controller only, the system tracking performance degrades significantly (Fig. 14.3).

It is evident from the test data shown in Fig. 14.3 that the baseline controller is unable to adequately track the reference pitch rate command signal.

In order to recover the desired baseline closed-loop performance, we shall add an adaptive output feedback u_{ad}, as shown in Table 14.1. We select adaptation rate matrix $\Gamma_{\bar{\Theta}}$ to be diagonal. Then, the adaptive laws can be written as

$$\begin{aligned} \dot{\hat{\Theta}} &= \text{Proj}\left(\hat{\Theta}, -\Gamma_\Theta\,\Phi(\hat{x})\,e_y^T\,R_0^{-\frac{1}{2}}\,W\,S^T\right) \\ \dot{\hat{K}}_u &= \text{Proj}\left(\hat{K}_u, -\Gamma_u\,u_{bl}\,e_y^T\,R_0^{-\frac{1}{2}}\,W\,S^T\right) \end{aligned}$$

where $\left(\Gamma_\Theta,\ \hat{K}_u\right)$ are the rates of adaptation for the original adaptive parameters $\hat{\Theta}$ and for the baseline control component u_{bl}, respectively. Also, in this case, the adaptive augmentation component is

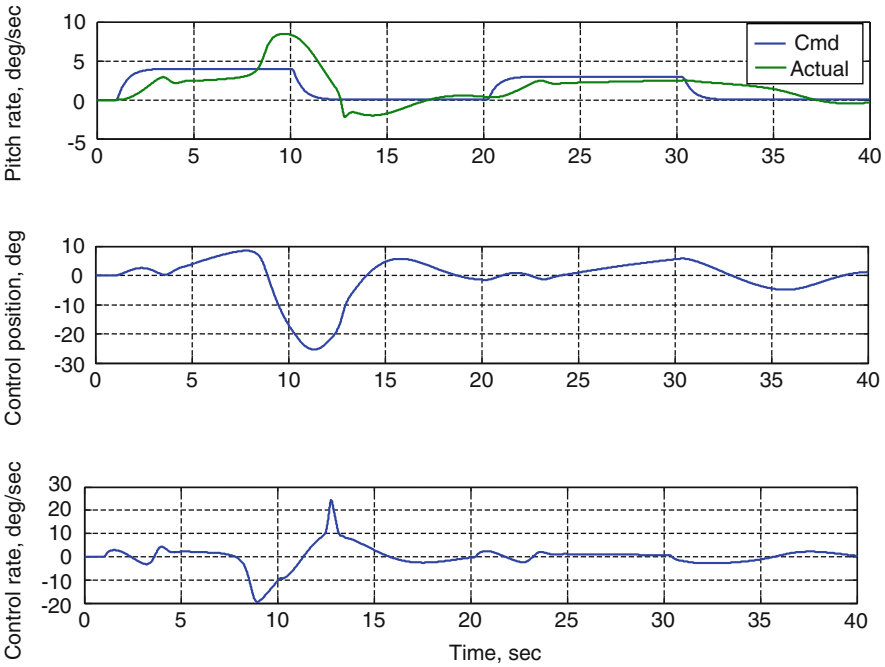


Fig. 14.3 Baseline controller tracking with uncertainties in Example 14.1

$$u_{ad} = -\hat{\Theta}^T \Phi(\hat{x}) - \hat{K}_u \underbrace{\left(-K_{lqr}\hat{x}\right)}_{u_{bl}}$$

It is interesting to note that according to (14.13), the adaptive gain $\hat{K}_u$ serves as an estimate of the constant parametric uncertainty $\left(I_{m\times m} - \Lambda^{-1}\right)$. Therefore, an estimate $\hat{\Lambda}$ of the system control effectiveness Λ can be reconstructed as

$$\hat{\Lambda} = \left(I_{m\times m} - \hat{K}_u\right)^{-1}$$

For the design study, we use $\Gamma_u = 100$ and select Γ_Θ to be diagonal, with all of its diagonal elements also set to 3,000. We define an α-dependent RBF regressor vector on the grid of breakpoints $[-10, \quad 10]\ \frac{\pi}{180}$, in two degree increments from each other. The RBF sigma widths are set to $\left(\frac{2}{3}\ \frac{2\pi}{180}\right)$. This value allows to position individual RBF-s such that any two consecutive functions overlap in the middle.

With the (baseline + adaptive) controller operating in the presence of the reduced control effectiveness, the scaled-down baseline control gains, and the α– dependent uncertainty, the closed-loop system performance is recovered rather well, using reasonable control deflections and rates (Fig. 14.4).

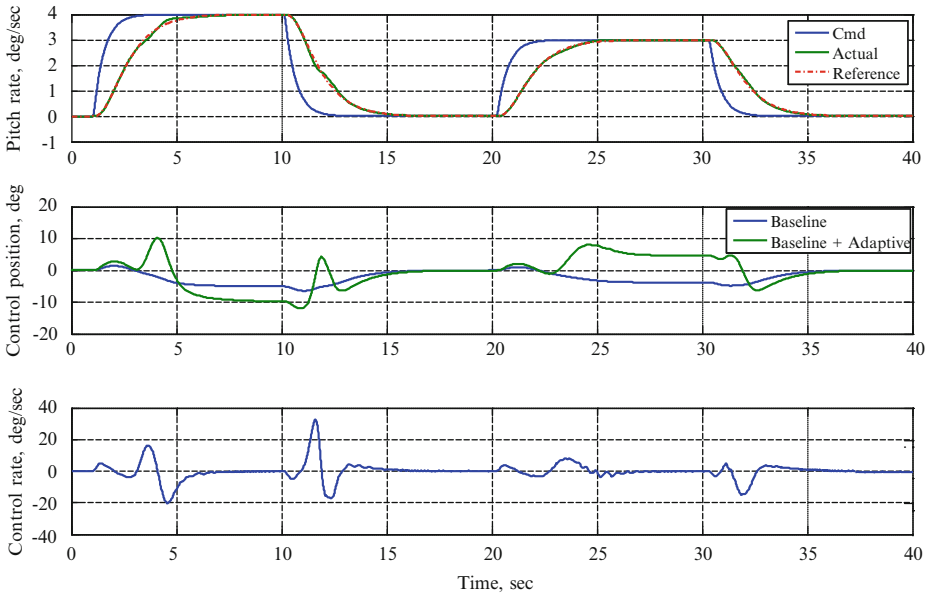


Fig. 14.4 (Baseline + Adaptive) controller tracking with uncertainties in Example 14.1

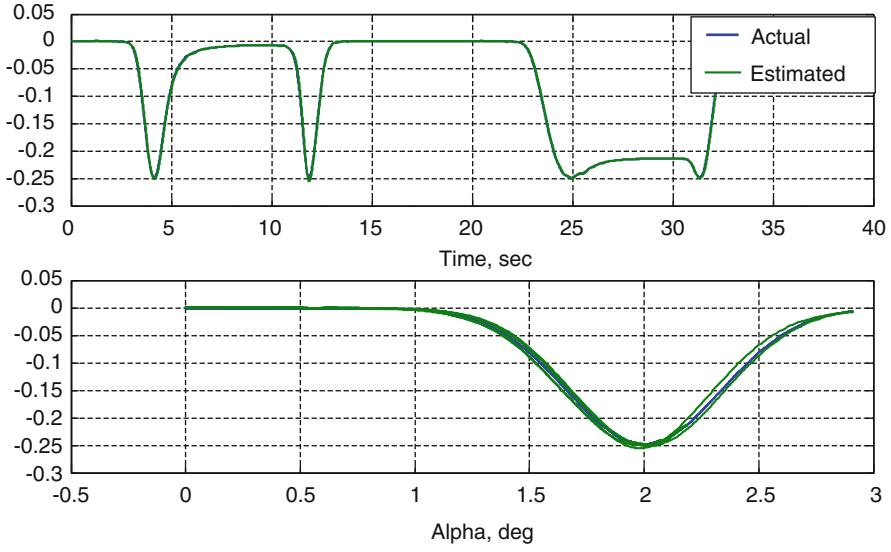


Fig. 14.5 RBF uncertainty estimation in Example 14.1

It turns out that the selected uncertainties persistently excite the system dynamics, and the adaptive system is able to approximate the unknown α-dependent RBF function, (Fig. 14.5).

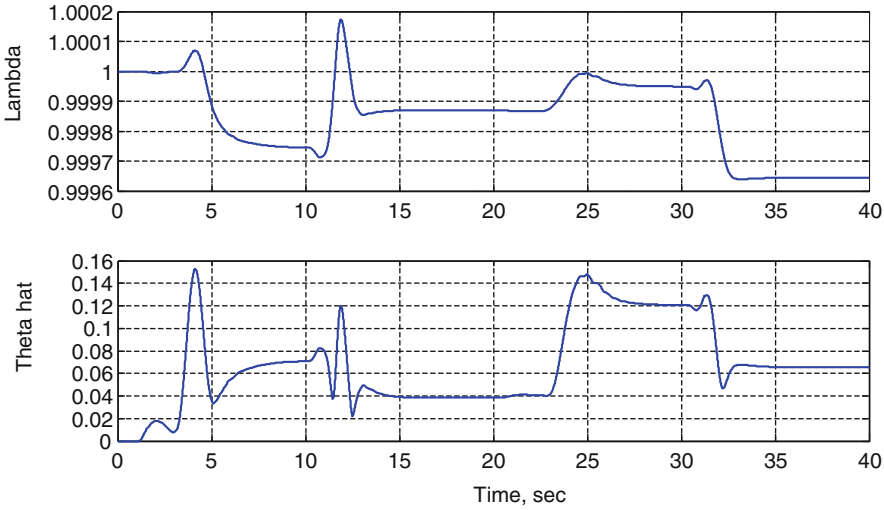


Fig. 14.6 Adaptive parameters in Example 14.1

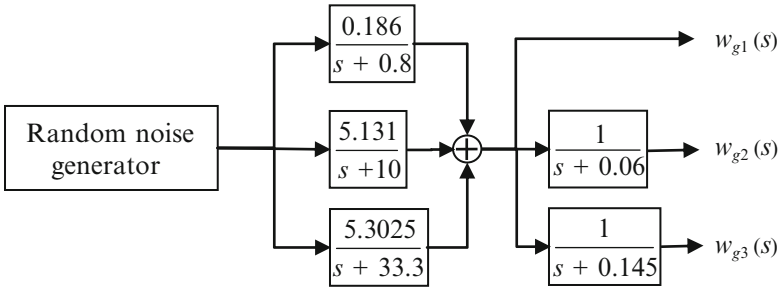


Fig. 14.7 Gust dynamic model in Example 14.1

Evolutions of $\hat{\Lambda}(t) = \left(1 - \hat{K}_u(t)\right)^{-1}$ and $\left\|\hat{\Theta}(t)\right\|$ are shown in Fig. 14.6.

The simulation test data indicate that system improvements mostly come through $\hat{\Theta}$ adaptation, with small changes to $\hat{K}_u$.

In order to add realism into simulation testing, we employ the gust model from [23]. The model is driven by a random noise, and it generates three separate gust profiles $\left(w_{g\,i}\right)_{i=1,2,3}$, according to the block diagram shown below (Fig. 14.7).

The intent of the model is to emulate time delays in gust propagation along the length of the aircraft.

For simulation testing, we select normally distributed zero-mean noise with standard deviation set to one and generate three light vertical gust profiles (Fig. 14.8).

Then, we evaluate closed-loop (baseline + adaptive) system tracking and gust rejection performance, in the presence of uncertainties (Fig. 14.9).

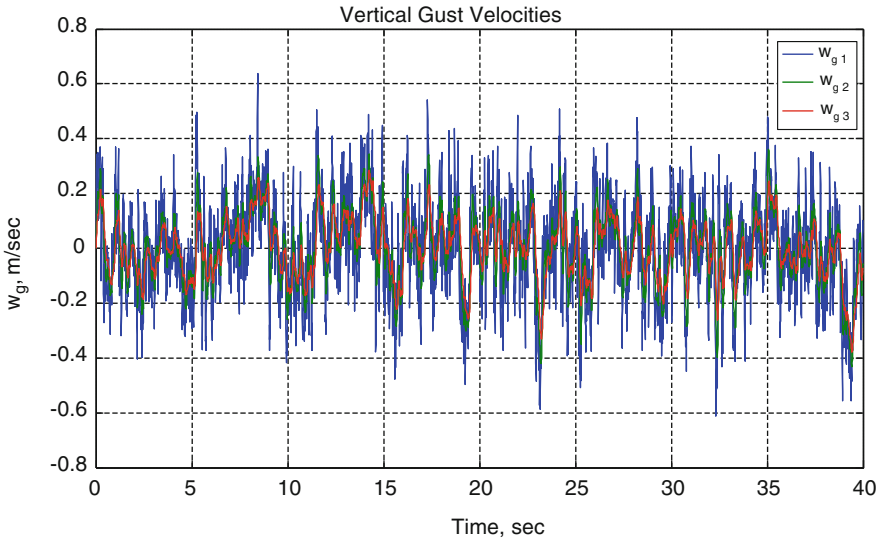


Fig. 14.8 Vertical gust velocities in Example 14.1

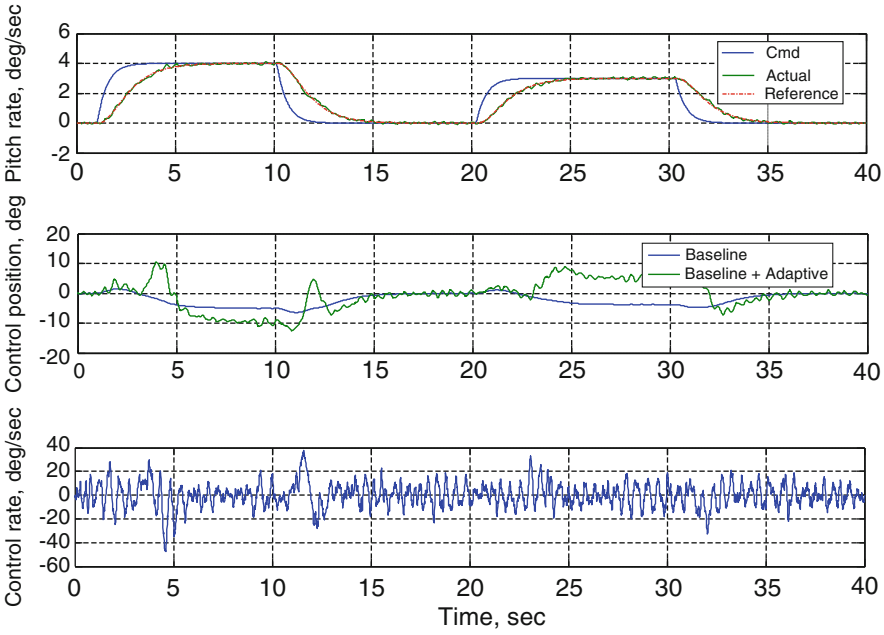


Fig. 14.9 Closed-loop tracking and gust rejection with uncertainties in Example 14.1

Clearly, the system is able to attenuate gust effects and to maintain its command tracking abilities in the presence of both gust and aerodynamic uncertainties.

We have also tested the exact same controller (without further retuning) in the presence of medium-to-high gust, nonlinear-in-parameter uncertainties, actuator dynamics, and with various pitch rate commands. All of the simulation trials have resulted in excellent and resilient to uncertainties closed-loop tracking performance. It is interesting to note that the collected simulation data indicate that the adaptive system requires about the same level of control effort as the baseline controller, yet the adaptive controller is able to cope quickly and efficiently with a variety of "unknown unknowns" in the system dynamics. □

14.6 Conclusions

In this chapter, we have presented an adaptive output feedback augmentation design for MIMO dynamical systems with matched uncertainties and with the number of output measurements exceeding the number of control inputs. The system-regulated output dynamics are allowed to be nonminimum phase and/or have a high relative degree. We have also developed a detailed design case study related to flight control of a large-size transport aircraft, with prominent structural dynamics, aerodynamic uncertainties, and environmental disturbances.

It is worthwhile to reflect back on the design procedure and summarize the quintessence of our method. After all proofs, derivations, and formal statements, it all comes down to the design of a baseline LQG/LTR dynamic output feedback controller, for the original system and without uncertainties. This linear system, with the baseline controller in the loop, defines the reference model. The key step to the design of an adaptive augmentation is the introduction of a small positive constant (the so-called tuning knob) into the observer ARE, followed by exploitation of the ARE asymptotic properties with respect to the tuning knob. It turns out that for a sufficiently small constant, one can create a linear combination of the system outputs, use it in an adaptive law, and then augment the baseline controller with an adaptive output feedback in such a way that all matched uncertainties in the system dynamics are mitigated. We have also noted that the derived design can be easily extended to a class of nonparametric and non-matched uncertainties, including bounded process noise.

In summary, our (baseline + adaptive) output feedback design consists of the following three main steps:

1. Using the system without uncertainties, square-up its input–output dynamics.
2. Select a sufficiently small tuning knob, design an LQG/LTR output feedback controller, and construct a linear combination of the system outputs for adaptive laws.
3. Compute an adaptive output feedback augmentation and add it to the baseline LQG/LTR controller.

What is interesting here is the fact that the design of a robust LQG/LTR controller paves the way to the design of a robust adaptive output feedback. Together, the two controllers provide seamless mitigation of a large class of uncertainties in the system dynamics while relying only on the system output measurements. So in a way, we have built an "output feedback bridge" between robust and adaptive control methodologies.

14.7 Exercises

Exercise 14.1. Suppose that all states are accessible, that is, let $y = x$. Write down the adaptive output feedback laws and compare them to a state feedback MRAC. Comment on the similarity of the derived solution with the observer-like adaptive output feedback design that was presented in this Section.

Exercise 14.2. Derive an adaptive output feedback solution, similar to the one from Sect. 14.3, but without a baseline linear controller.

Exercise 14.3. Consider the system dynamics (14.7), with the linear regressor vector $\Phi(x) = K_\Phi^T x$ and with an unknown constant matrix $K_\Phi \in R^{n\times m}$. Suppose that $B_{ref} = 0_{n\times m}$. Using the techniques from Sect. 14.3 and relying only on the system output measurements y, derive an adaptive controller to force the system-regulated output z track bounded commands.

Exercise 14.4. With the aircraft data from Sect. 14.5, perform a trade study in selecting appropriate values for the tuning knob ν. Show numerically that as ν gets smaller, the asymptotic relation (14.6) takes place. Plot $\left\| P_\nu^{-1} B - C^T R_0^{-\frac{1}{2}} W \right\|$ versus ν. Also, show that for small values of ν, the observer gains get large, eventually leading to high gain effects. Explain this phenomenon. Compute and plot (vs. ν) the associated crossover frequencies and MIMO gain/phase margins at the output breakpoint of the nominal system. Comment on your results.

Exercise 14.5. For the aircraft data from Sect. 14.5, select your own $(\bar{B},\ \nu)$, redesign the controller, rerun simulation tests, and compare your data with the original results. Test and comment on the system tracking and gust rejection performance. Select a matched nonparametric uncertainty and, without retuning the controller, rerun simulation tests with increasing gust magnitudes. Comment on your results.

Exercise 14.6. Using the aircraft data from Sect. 14.5, replace the system-regulated output q with α. Design a (baseline + adaptive) output feedback controller to track bounded angle of attack commands α_{cmd}. Simulate the closed-loop system using various uncertainties (parametric and nonparametric), as well as the gust model from Example 14.1.

References

1. Balas, G., Young, P.: Sensor selection via closed-loop control objectives. IEEE Trans. Autom. Control **7**(6), 692–704 (2003)
2. Wal, M., Jager, B.: A review of methods for input/output selection. Automatica **37**, 487–510 (2001)
3. Narendra, K.S., Annaswamy, A.M.: Stable Adaptive Systems. Dover, New York (2005)
4. Krstic, M., Kanellakopoulos, I., Kokotovic, P.: Nonlinear and Adaptive Control Design. Wiley, New York (1995)
5. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall Inc., New York (2002)
6. Seshagiri, S., Khalil, H.K.: Output feedback control of nonlinear systems using RBF neural networks. IEEE Trans. Neural Netw **11**(1), 69–79 (2000)
7. Freidovich, L.B., Khalil, H.K.: Performance recovery of feedback-linearization-based designs. IEEE Trans. Autom. Control **53**(10), 2324–2334 (2008)
8. Kuipers, M., Ioannou, P.: Multiple model adaptive control with mixing. IEEE Trans. Autom. Control **55**(8), 1822–1836 (2010)
9. Lavretsky, E., Hovakimyan, N., Calise, A.: Upper bounds for approximation of continuous-time dynamics using delayed outputs and feedforward neural networks. IEEE Trans. Autom. Control **48**(9), 1606–1610 (2003)
10. Doyle, J.C., Stein, G.: Multivariable feedback design: concepts for a classical/modern synthesis. IEEE Trans. Autom. Control **26**(1), 4–16 (1981)
11. Kwakernaak, H., Sivan, R.: The maximally achievable accuracy of linear optimal regulators and linear filters. IEEE Trans. Autom. Control **17**(1), 79–86 (1972)
12. Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems. Wiley-Interscience, New York (1972)
13. Safonov, M.G., Laub, A.J., Hurtmann, G.L.: Feedback properties of multivariable systems: the role and use of the return difference matrix. IEEE Trans. Autom. Control **26**(1), 47–65 (1981)
14. Stein, G., Athans, M.: The LQG/LTR procedure for multivariable feedback control design. IEEE Trans. Autom. Control **32**(2), 105–114 (1987)
15. Lavretsky, E.: Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. In: Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Ontario, Canada (2010)
16. Misra, P.: Numerical algorithms for squaring-up non-square systems, Part II: General case. In: Proceedings of American Control Conference, San Francisco, CA (1998)
17. Kevorkian, J., Cole, J.D.: Multiple Scale and Singular Perturbation Methods. Math. Sciences (Springer-Verlag New York Inc), vol. 114. Springer, New York (1996)
18. Murray, J.D.: Asymptotic Analysis. Appl. Math. Sciences (Springer-Verlag New York Inc). Springer, New York (1984)
19. Ioannou, P.A., Fidan, B.: Adaptive Control Tutorial. SIAM, Philadelphia (2006)
20. Slotine, J.-J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, Englewood Cliffs (1995)
21. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
22. Pomet, J.B., Praly, L.: Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Trans. Autom. Control **37**(6), 729–740 (1992)
23. McLean, D.: Gust-alleviation control systems for aircraft. Proc. Inst. Electr. Eng. Control Sci. **125**(7), 675–685 (1978)

ERRATA LIST

Robust and Adaptive Control with Aerospace Applications – 1st Edition, Springer, 2013
Eugene Lavretsky and Kevin A. Wise

Revised on 23 Jul 2013

Part I

Introduction

1) Page 10, change the top equation to:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ -\dot{h} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

Chapter 1

1) Page 21, remove boxes from around the equations

$$\begin{aligned} &\dot{x} = A_p x + B_p(C_c x_c + D_{c1} y_{\text{in}}) = A_p x + B_p C_c x_c + B_p D_{c1} y_{\text{in}} \\ &y_{\text{out}} = C_p x + D_p(C_c x_c + D_{c1} y_{\text{in}}) = C_p x + D_p C_c x_c + D_p D_{c1} y_{\text{in}} \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} &= \underbrace{\begin{bmatrix} A_p & B_p C_c \\ 0_p & A_c \end{bmatrix}}_{A_{\text{Lo}}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\begin{bmatrix} B_p D_{c1} \\ B_{c1} \end{bmatrix}}_{B_{\text{Lo}}} y_{\text{in}} \\ y_{\text{out}} &= \underbrace{\begin{bmatrix} C_p & D_p C_c \end{bmatrix}}_{C_{\text{Lo}}} \begin{bmatrix} x \\ x_c \end{bmatrix} + \underbrace{\begin{bmatrix} D_p D_{c1} \end{bmatrix}}_{D_{\text{Lo}}} y_{\text{in}} \end{aligned} \tag{1.53}$$

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3_15,

Chapter 2

1) Page 34, change

$$H^*(x,\nabla_x J^*,t) = x_1^4 + \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - 2\frac{\partial J^*}{\partial x_2}x_1 - 3\frac{\partial J^*}{\partial x_2}x_2 - \frac{1}{2}\frac{\partial J^*}{\partial x_2} \quad (2.26)$$

to

$$H^*(x,\nabla_x J^*,t) = x_1^4 + \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - 2\frac{\partial J^*}{\partial x_2}x_1 - 3\frac{\partial J^*}{\partial x_2}x_2 - \frac{1}{2}\left(\frac{\partial J^*}{\partial x_2}\right)^2$$

2) Page 34, change

$$\begin{aligned} -\frac{\partial J^*}{\partial t} &= H^*(x,\nabla_x J^*(x,t),t) \\ &= x_1^4 + \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - \frac{\partial J^*}{\partial x_2}\left(2x_1 - 3x_2 - \frac{1}{2}\right) \end{aligned} \quad (2.27)$$

to

$$\begin{aligned} -\frac{\partial J^*}{\partial t} &= H^*(x,\nabla_x J^*(x,t),t) \\ &= x_1^4 - \frac{1}{4}\left(\frac{\partial J^*}{\partial x_2}\right)^2 + \frac{\partial J^*}{\partial x_1}x_2 - \frac{\partial J^*}{\partial x_2}(2x_1 - 3x_2) \end{aligned}$$

3) Page 36, change

$$-\frac{\partial J^*}{\partial t} = x^T Q x + \tfrac{1}{4}\frac{\partial J^*}{\partial x}BR^{-1}B^T\nabla_x J^* + \frac{\partial J^*}{\partial x}Ax - \tfrac{1}{2}\frac{\partial J^*}{\partial x}BR^{-1}B^T\nabla_x J^* \quad (2.34)$$

to

$$-\frac{\partial J^*}{\partial t} = x^T Q x - \tfrac{1}{4}\frac{\partial J^*}{\partial x}BR^{-1}B^T\nabla_x J^* + \frac{\partial J^*}{\partial x}Ax$$

4) Page 36, change

$$u^*(x,\,t) = -\tfrac{1}{2}R^{-1}B^T\frac{\partial J^*(x,\,t)}{\partial x} = -\underbrace{R^{-1}B^TP(t)}_{K(t)}x = -K(t)\,x \qquad (2.39)$$

to

$$u^*(x,t) = -\frac{1}{2}R^{-1}B^T\nabla_x J^* = -\underbrace{R^{-1}B^TP(t)}_{K(t)}x = -K(t)x$$

5) Page 46, change

$$\det[sI - H] = \phi_{\rm cl}(s)\phi_{\rm cl}(-s) \qquad (2.81)$$

The asymptotic properties we desire to explore are those associated with the migration of these eigenvalues, as the numerical values in the LQR penalty matrices Q and R are varied. We can examine these eigenvalues (roots of $\phi_{\rm cl}(s)$) through the polynomial formed by expanding the $\det[sI - H]$. We begin with some elementary row and column operations on H. First, we multiply the first row of H by $-Q(sI - A)H^{-1}$ and add it to the second row. This yields

to

$$-Q(sI - A)^{-1}$$

Chapter 3

1) Page 57, change

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \dot{x} \end{bmatrix}, \mu = \dot{u}$$
$$\tilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} D_c \\ B \end{bmatrix} \qquad (3.24)$$

to (need comma and space)

$$\dot{z} = \widetilde{A}z + \widetilde{B}\mu, \quad z = \begin{bmatrix} e \\ \dot{x} \end{bmatrix}, \mu = \dot{u}$$
$$\widetilde{A} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}, \widetilde{B} = \begin{bmatrix} D_c \\ B \end{bmatrix}$$

2) Page 57, change

$$\dot{z} = \tilde{A}z + \tilde{B}\mu z = \begin{bmatrix} e \\ \xi \end{bmatrix}, \xi = \ddot{x} - \omega^2 x, \mu = \ddot{u} - \omega^2 u,$$
$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & C_c \\ 0 & 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ D_c \\ B \end{bmatrix} \tag{3.25}$$

to

$$\dot{z} = \widetilde{A}z + \widetilde{B}\mu, \quad z = \begin{bmatrix} e \\ \xi \end{bmatrix}, \xi = \ddot{x} - \omega^2 x, \mu = \ddot{u} - \omega^2 u,$$
$$\widetilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & C_c \\ 0 & 0 & A \end{bmatrix}, \widetilde{B} = \begin{bmatrix} 0 \\ D_c \\ B \end{bmatrix}$$

3) Page 59, change

$$P\tilde{A} + \tilde{A}^T P - P\tilde{B}R^{-1}\tilde{B}^T P + Q = 0 \tag{3.36}$$

The resulting steady-state $n_u \times (n_r + n_x)$-dimensional feedback controller gain matrix is

to

$$n_u \times (pn_r + n_x)$$

4) Page 60, change

$$u = -K_x x + \sum_{i=1}^{p} s^{-i}\left(a_i\left({}^{(p-i)}u + K_x\,{}^{(p-i)}x\right) - K_i\,{}^{(p-i)}e\right) \tag{3.40}$$

to

$$u = -K_x x + \sum_{i=1}^{p} s^{-i}(a_i(u + K_x x) - K_i e)$$

5) Page 60, change

$$\dot{z} = \left(\tilde{A} - \tilde{B}K_c\right)z + Fr \qquad (3.41)$$

$$\text{where} \quad F = \left[\, -I_{n_u \times n_u} \quad 0_{n_u \times n_x} \,\right]^T .$$

to

$$\dot{x}_e = \left(\widetilde{A} - \widetilde{B}K_c\right)x_e + Fr$$

where the extended state x_e is (3.18) integrated p-times, and $F = \left[\, -I_{n_u \times n_u} \quad 0_{n_u \times n_x} \,\right]^T$.

6) Page 63, change the following figure

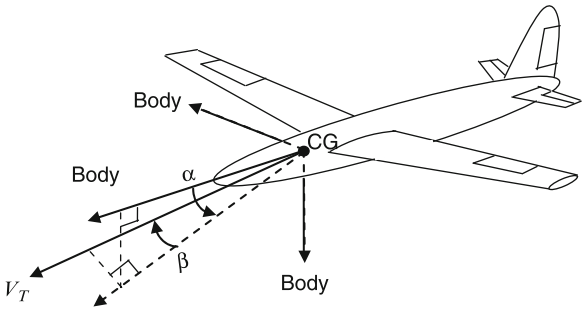
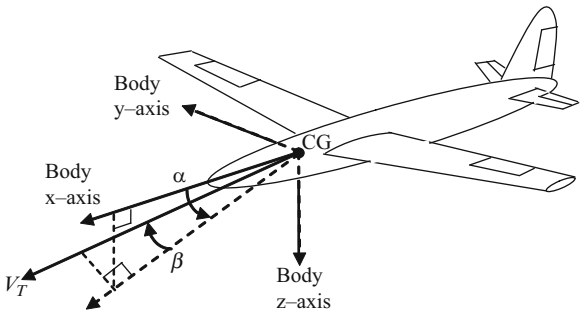


Fig. 3.3 Unmanned aircraft

to the same figure as in Figure 1.3, page 8.



7) Page 63, Change

$$A_z = -V\dot{\gamma} = VZ_\alpha\alpha + VZ_\delta\delta \tag{3.48}$$

to

$$A_z = -V\dot{\gamma} = Z_\alpha\alpha + Z_\delta\delta$$

8) Page 63, change

$$\begin{aligned}\dot{A}_z =& Z_\alpha A_z + VZ_\alpha q + VZ_\delta\dot{\delta}_e\\ \dot{q} =& \frac{M_\alpha}{VZ_\alpha}A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_e\end{aligned} \tag{3.49}$$

to

$$\begin{aligned}\dot{A}_z =& \frac{Z_\alpha}{V}A_z + Z_\alpha q + Z_\delta\dot{\delta}_e\\ \dot{q} =& \frac{M_\alpha}{Z_\alpha}A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right)\delta_e\end{aligned}$$

9) Page 64, change

$$\begin{bmatrix}\dot{A}_z\\ \dot{q}\\ \dot{\delta}_e\\ \ddot{\delta}_e\end{bmatrix} = \begin{bmatrix}Z_\alpha & VZ_\alpha & 0 & VZ_\delta\\ M_\alpha/VZ_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a\end{bmatrix}\begin{bmatrix}A_z\\ q\\ \delta_e\\ \dot{\delta}_e\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ {\omega_a}^2\end{bmatrix}\delta_c \tag{3.51}$$

to

$$\begin{bmatrix}\dot{A}_z\\ \dot{q}\\ \dot{\delta}_e\\ \ddot{\delta}_e\end{bmatrix} = \begin{bmatrix}\frac{Z_\alpha}{V} & Z_\alpha & 0 & Z_\delta\\ M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a\end{bmatrix}\begin{bmatrix}A_z\\ q\\ \delta_e\\ \dot{\delta}_e\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ {\omega_a}^2\end{bmatrix}\delta_c$$

10) Page 64, change

$$\begin{bmatrix} \dot{e} \\ \ddot{A}_z \\ \dddot{q} \\ \ddot{\delta}_e \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha & \mathrm{V}Z_\alpha & 0 & \mathrm{V}Z_\delta \\ 0 & M_\alpha/\mathrm{V}Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \ddot{q} \\ \dddot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \dot{\delta}_c \tag{3.53}$$

to

$$\begin{bmatrix} \dot{e} \\ \ddot{A}_z \\ \dddot{q} \\ \ddot{\delta}_e \\ \dddot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \dfrac{Z_\alpha}{V} & Z_\alpha & 0 & Z_\delta \\ 0 & M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \dddot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \dot{\delta}_c$$

11) Page 65, change

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ & 0 & & 0 & \\ & & 0 & & \\ & 0 & & 0 & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{q} \\ \dot{A}_z \\ \dddot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}, \tag{3.57}$$

to

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ & 0 & & \underset{\sim}{0} & \\ & & 0 & & \\ & \underset{\sim}{0} & & 0 & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \dddot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}$$

12) Page 67, change

The time domain performance metrics of interest here are 63% rise time, 95% settling time, percent overshoot, percent undershoot (because the system is nonminimum phase), max actuator deflection, and max actuator rate in response to a constant step command. The frequency domain performance metrics are loop gain crossover frequency ω_c in Hz, the minimum of the minimum singular value of the return difference dynamics, denoted $\underline{\sigma}(I+L)$, and the minimum of the minimum singular value of the stability robustness matrix $I+L^{-1}$, denoted $\underline{\sigma}\left(I+L^{-1}\right)$. The metric $\underline{\sigma}(I+L)=1/\|S\|_\infty$ and $\underline{\sigma}\left(I+L^{-1}\right)=1/\|T\|_\infty$ (see Chap. 5, Sect. 5.2 for definitions). These metrics, plotted versus ω_c, are used to determine how the increasing bandwidth of the system affects the system characteristics, indicating a desired value for q_{11}.

to

$$\underline{\sigma}(I+L)$$

13) Page 67, change

Figure 3.7 shows two frequency response metrics: the minimum of the minimum singular value of the return difference dynamics $\underline{\sigma}(I+L)$ and the minimum of the minimum singular value of the stability robustness matrix $\underline{\sigma}\left(I+L^{-1}\right)$.

As is characteristic of LQR state feedback designs (discussed in Chap. 2), the $\underline{\sigma}(I+L)$ is equal to unity for all q_{11} design values. This metric is not particularly useful for developing state feedback designs but is critical when output feedback is used. The $\underline{\sigma}\left(I+L^{-1}\right)$, which is the inverse of the infinity norm of the complementary sensitivity function, is a measure of the damping in the dominant poles of the closed-loop system. We would like to maximize $\underline{\sigma}\left(I+L^{-1}\right)$. The figure shows that this metric tends to favor larger gains.

to (keep together on same line)

$$\underline{\sigma}(I+L)$$

Chapter 4

1) Page 75, change

$$\begin{aligned}\|u\|_1 &= \int_{-\infty}^{\infty} |u(t)|dt \\ \|u\|_2 &= \left(\int_{-\infty}^{\infty} |u(t)|dt\right)^{\frac{1}{2}} \\ \|u\|_\infty &= \sup_t |u(t)|\end{aligned} \tag{4.2}$$

to

$$\begin{aligned}\|u\|_1 &= \int_{-\infty}^{\infty} |u(t)|dt \\ \|u\|_2 &= \left(\int_{-\infty}^{\infty} |u(t)|^2 dt\right)^{\frac{1}{2}} \\ \|u\|_\infty &= \sup_t |u(t)|\end{aligned}$$

2) Page 77, change

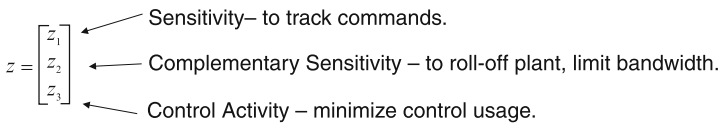
$$\|G\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} G(-s)G(s)ds = \frac{1}{2\pi j}\oint G(-s)G(s)ds \tag{4.9}$$

to

$$\|G\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} G(-s)G(s)ds = \frac{1}{2\pi j}\oint G(-s)G(s)ds \tag{4.9}$$

3) Page 88, change

In this section, we build a control design model that embeds the sensitivity, complementary sensitivity, and control activity weighting filters from Sect. 4.3 into a state space model and then solves for the state feedback gain matrix (4.45) using a method called γ-iteration. The design model needs to be of the form of (4.24). Define the regulated variables in vector z to comprise sensitivity, complementary sensitivity, and control activity variables.



From Sect. 4.3, the weighting filter W_s should be designed to be the inverse of the desired loop shape for $S(s)$, the weighting filter W_T should be designed to be the inverse of the desired loop shape for $T(s)$, and the control activity penalty to penalize control activity in the desired frequency range. To build the H_∞-controller

to

section 4.4

4) Page 94, change

$$\mathrm{K}_\infty = [-0.92788164.7534 \;\; -13.1623 \;\; -0.11795 \;\; -3.210460.151197] \quad (4.68)$$

to

$$\mathrm{K}_\infty = [-0.927881\ 64.7534 \;\; -13.1623 \;\; -0.11795 \;\; -3.21046\ 0.151197]$$

Chapter 5

1) Page 101 change

$$K(s) = [K_{A_z}(s)K_q(s) \quad K_q(s)] \quad (5.12)$$

which is a 2×1 matrix. A state-space model for this controller is

to

1×2

2) Page 102, change

$$\begin{aligned} u &= Ke \\ y &= GKe + w \\ z &= GKe + w + v \\ e &= r + z = r + GKe + w + v \\ E(s) &= S(s)(R(s) + W(s) + V(s)) \end{aligned} \tag{5.18}$$

to

$$\begin{aligned} u &= Ke \\ y &= GKe + w \\ z &= GKe + w + v \\ e &= r - z = r - GKe - w - v \\ E(s) &= S(s)(R(s) - W(s) - V(s)) \end{aligned}$$

3) Page 114, change

114 5 Frequency Domain Analysis

2. $\det[I + L(s, \varepsilon)] = 0$ *for all* (s, ε) *in* $D_R \times [0, 1]$ *and for all* R *sufficiently large*. ■

to

$$\det[I + L(s,\varepsilon)] \neq 0$$

4) Page 115, add space between x and is

5.3.5 A + B Argument

The minimum singular value $\underline{\sigma}(A)$ measures the near singularity of the matrix A. Assume that the matrix $A + B$ is singular. If $A + B$ is singular then $A + B$ is rank deficient. Since $A + B$ is rank deficient, then there exists a vector $x \neq 0$ with unit magnitude $\left(\|x\|_2 = 1\right)$ such that $(A + B)x = 0$ ($x$is in the null space of $A + B$). This leads to $Ax = -Bx$ with $\|Ax\|_2 = \|Bx\|_2$. Using the above singular value definitions in (5.20) and $\|x\|_2 = 1$, we obtain the following inequality.

5) Page 116, change

From Theorem 5.5, stability is guaranteed if $\bar{\sigma}(I + L^{-1}(s)) > \bar{\sigma}(\Delta(s))$. For $\Delta(s) = E(s) - I$, $E(s) \in R^{n_u \times n_u}$, the singular values of $\Delta(s)$ are

to

$$\underline{\sigma}\big(I + L^{-1}(s)\big) > \overline{\sigma}(\Delta(s))$$

6) Page 119, change

For this single-input single-output system, we will compute the Nyquist, Bode, $\underline{\sigma}(I + L)$, and $^{-}\sigma(I + L^{-1})$ at the plant input, and $\bar{\sigma}(S)$ and $\bar{\sigma}(T)$ at the plant output. Figures 5.14, 5.15, 5.16, and 5.17 show the plant input frequency response curves (Nyquist, Bode, $\underline{\sigma}(I + L)$, and $\underline{\sigma}(I + L^{-1})$). On the Nyquist plot in Fig. 5.14, we have drawn a circle centered at $(-1, j0)$ that has radius equal to the minimum of $\underline{\sigma}(I + L)$ (from Fig. 5.16). The classical gain and phase margins from Fig. 5.14 are 8.8 dB (2.7536) and 50°. These are also easily extracted from the Bode plot in Fig. 5.15. From Figs. 5.16 and 5.17, we have

to

$$\underline{\sigma}\big(I + L^{-1}\big)$$

7) Page 119,

with the matrices defined in (5.14). The gains are $K_a = -0.0015, K_q = -0.32, a_q = 2.0$ and $a_z = 6.0$. Substituting these values into (5.14) yields

to

$a_z = 2.0$ and $a_q = 2.0$

8) Page 122, change

$$\begin{aligned} \mathrm{GM}_{I+L^{-1}} &= [0.4324 \quad 1.5676] \\ &= [-7.28 \quad 3.90]\ \mathrm{dB}; \end{aligned} \quad \mathrm{PM}_{I+L^{-1}} = \pm 42.84^\circ \qquad (5.65)$$

to

$$\begin{aligned} GM_{I+L^{-1}} &= [0.2695 \quad 1.7305]; \\ &= [-11.4 \quad 4.7]\ \mathrm{dB} \end{aligned} \qquad PM_{I+L^{-1}} = \pm 42.84\ \mathrm{deg}$$

9) Page 122, change

$$\text{GM} = [-7.28 \quad 7.28]\text{dB; PM} = \pm 42.84^{\circ} \tag{5.66}$$

to

$$GM = [-11.4\ 7.28]\ \text{dB};\ \ PM = \pm\ 42.84\ \text{deg}$$

10) Page 124, change

Figure 5.21 shows the frequency response of the controller $\bar{\sigma}(K)$. This figure indicates the amplification, or attenuation, of sensor noise through the controller. Although not directly related to stability margins, this frequency response should e examined to make sure the bandwidth of the controller is not too high and that high-frequency noise is not adversely amplified. The shape of the frequency response clearly shows the proportional-plus-integral control action that the controller is providing.

to

should be examined

11) Page 127, change

For a control system under no uncertainty, the controller stabilizes the plant and the return difference matrix in nonsingular at all frequencies. Stability of the nominal system implies

$$\det[I + L(s)] \neq 0\, \forall s \in D_R. \tag{5.68}$$

to

is

12) Page 130, change

This triple describes the$M$matrix in the ΔM analysis model (Fig. 5.8).

to

add space in front and after M, and change 5.8 to 5.7

13) Page 133, change

$$B_M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & 10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad C_M$$

$$= \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix}$$

to

$$B_M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & 10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$C_M = \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix}$$

14) Page 136, change

$$A = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix} \quad (0.1)$$

to

$$A = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.7 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0.1228 & -0.2763 \\ 195.5 & -529.4 \\ -53.61 & 33.25 \end{bmatrix} \quad (0.2)$$

15) Page 138, change

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \left[\begin{bmatrix} \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \right] \tag{5.109}$$

to

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \left[\begin{bmatrix} \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \right]$$

16) Page 158, change

the $\det[I + KH]$, and indicate the number of encirclements.

(b) Plot the singular values of the return difference matrix and stability robustness matrix versus frequency. Compute the singular-value gain and phase margins for this system. This is a plot of $\underline{\sigma}[I + L]$ and $\bar{\sigma}[I + L^{-1}]$ versus frequency. Plot these using a log scale for frequency and magnitude in dB.

Exercise 5.2. Consider the block diagrams shown below. Each block in the diagrams is a scalar.

to

$\underline{\sigma}$

Chapter 6

1) Page 164, change

$$\begin{aligned} N_0 &= X_{n_{y1}} \quad X_{n_{y2}}^{-1} \\ B_0 &= X_{p_2} - N_0 X_{p_1} \\ A_r &= A_{22} - N_0 A_{12} \end{aligned} \tag{6.15}$$

to

$$N_0 = X_{n_{y2}} X_{n_{y1}}^{-1}$$

2) Page 166, change

$$\begin{bmatrix} \dot{e} \\ \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha/V & Z_\alpha & 0 & Z_\delta \\ 0 & M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_z Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -{\omega_a}^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {\omega_a}^2 \end{bmatrix} \dot{\delta}_c \tag{6.24}$$

to: make bracket same size as other brackets.

3) Page 169, change

$$\begin{aligned} X_n = \begin{bmatrix} \begin{bmatrix} 2.2295\text{e}{-001} \\ -9.4676\text{e}{-001} \\ -1.6075\text{e}{-002} \\ -5.3104\text{e}{-002} \\ 2.2551\text{e}{-001} \end{bmatrix} \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & -1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & -3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & -2.3622\text{e}{-002}j \end{bmatrix} \\ \times \begin{bmatrix} -1.0379\text{e}{-001} & -9.1764\text{e}{-002}j \\ 9.1946\text{e}{-001} & \\ -1.4240\text{e}{-003} & +1.0897\text{e}{-002}j \\ -4.3324\text{e}{-002} & +3.3555\text{e}{-002}j \\ 3.6293\text{e}{-001} & +2.3622\text{e}{-002}j \end{bmatrix} \end{bmatrix} \end{aligned} \tag{6.34}$$

to

$$X_{n_y} = \begin{bmatrix} \begin{bmatrix} 2.2295\text{e}-001 \\ -9.4676\text{e}-001 \\ -1.6075\text{e}-002 \\ -5.3104\text{e}-002 \\ 2.2551\text{e}-001 \end{bmatrix} \begin{bmatrix} -1.0379\text{e}-001-9.1764\text{e}-002j \\ 9.1946\text{e}-001 \\ -1.4240\text{e}-003-1.0897\text{e}-002j \\ -4.3324\text{e}-002-3.3555\text{e}-002j \\ 3.6293\text{e}-001-2.3622\text{e}-002j \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} -1.0379\text{e}-001+9.1764\text{e}-002j \\ 9.1946\text{e}-001 \\ -1.4240\text{e}-003+1.0897\text{e}-002j \\ -4.3324\text{e}-002+3.3555\text{e}-002j \\ 3.6293\text{e}-001+2.3622\text{e}-002j \end{bmatrix}$$

4) Page 173, change

The compensator design (6.12) requires selecting a gain matrix P_0 such that the residual dynamics A_r in (6.16) are stable. The matrices needed to form A_r are N_0 and B_0.

to

$$A_{re}$$

5) Page 173, change

$$N_0 = X_{n_{y1}} X_{n_{y2}}^{-1} = \begin{bmatrix} -1.6565\text{e}-001 & -5.8888\text{e}-002 & 4.4744\text{e}+000 \\ 2.4157\text{e}+000 & 6.3925\text{e}-001 & -1.8175\text{e}+001 \end{bmatrix} \tag{6.43}$$

to

$$N_0 = X_{n_{y2}} X_{n_{y1}}^{-1} = \begin{bmatrix} -1.6565\text{e}-001 & -5.8888\text{e}-002 & 4.4744\text{e}+000 \\ 2.4157\text{e}+000 & 6.3925\text{e}-001 & -1.8175\text{e}+001 \end{bmatrix}$$

6) Page 173, change

Using the dynamic compensator in (6.12) with matrices defined in (6.17), the compensator is designed by choosing the free parameter matrix P_0 such that the residual dynamics in (6.16) are stable. For this example (6.46),

to

delete (6.46)

7) Page 173, change

$$
\begin{aligned}
A_{re} &= A_r + B_0 P_0 A_{12} \\
&= \begin{bmatrix} 4.8883\text{e}+000 & 3.3521\text{e}-001 \\ -6.6917+003 & -9.0801\text{e}+001 \end{bmatrix} \\
&\quad + \begin{bmatrix} -2.5706\text{e}-003 & -2.6871\text{e}-003 \\ 9.3759\text{e}-001 & -7.0061\text{e}-002 \end{bmatrix} P_0 \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix}
\end{aligned} \tag{6.46}
$$

to

$-6.6917\text{e}+003$

8) Page 174, change

By multiplying out the matrices in (6.47), one can determine which elements of P_0 need to be chosen. This matrix is designed using a tuning process in which the elements are increased in magnitude until a suitable design is obtained (trial and error). After some tuning, the following matrix was obtained:

$$
P_0 = \begin{bmatrix} 0 & 2 & -500 \\ 0 & 2 & -2000000 \end{bmatrix} \tag{6.47}
$$

The zero elements in the first column were found not to matter. They were made zero to reduce the control usage. Substituting this P_0 into (6.47) yields

to

(6.46)

8) Page 177, change

covariance, which results from solving the algebraic filter Riccati equation (covariance equation), and Q_0 and R_0 are the process and measurement noise covariances from (6.54), respectively. The optimal control is formed using the LQR state feedback control gain matrix K_c and the estimated state feedback $\hat{x}$, given as

$$
u = -K_c \hat{x} \tag{6.55}
$$

to

(6.53)

9) Page 177, change

$$u = -K_c \hat{x} \tag{6.55}$$

Figure 6.10 combines the LQR controller (Chap. 3) with the Kalman filter state estimator (6.55) into a block diagram. This is the LQG control architecture.

to

(6.54)

10) Page 178, change

$$Q_f = Q_0 + \frac{1}{\rho} BB^T \tag{6.58}$$

where Q_0 is the nominal plant process disturbance covariance from (6.54), B is the control input distribution matrix, and $\rho$is the LTR filter compensation parameter. This parameter is adjusted to recover the LQR frequency domain characteristics

to

(6.53)

11) Page 179, change

Considering the loop broken at the plant input, LTR modifies K_f to create a system that has stability properties that asymptotically approach those of the LQR. The method uses a trial and error procedure in which the filter design is parameterized by a scalar $\rho > 0$ such that when $\rho \rightarrow 0$ we have $L_{\mathrm{LQG}} \rightarrow L_{\mathrm{LQR}}$ asymptotically but not necessarily uniformly. It is evident that the location of the Kalman filter eigenvalues, (6.58), alters the closed-loop frequency characteristics of the system.

The LQG/LTR approach requires that the controlled system (plant) be minimum phase (i.e., no RHP transmission zeros). The minimum phase requirement occurs

to

(6.57)

12) Page 179, change

The LQG/LTR loop transfer function matrix at the plant input, L_{LQG}, will asymptotically recover the LQR frequency domain characteristics as $\rho \to 0$. This can be shown as follows. As $\rho \to 0$, the process covariance Q_f in (6.59) becomes largely dominated by the second term $\frac{1}{\rho}BB^T$. As these elements of Q_f get large, the covariance matrix P_f has elements that get large, resulting in the Kalman gain matrix K_f getting large with the following result:

to

(6.58)

13) Page 179, change

$$u = -K_c\left(sI - \tilde{A} + \tilde{B}K_c + K_f C\right)^{-1} K_f y \tag{6.60}$$

Substituting for the measurement $y = Cx + v$ and letting $\rho \to 0$ as in (6.60) yields

to

(6.59)

14) Page 179, change

$$\begin{aligned} L_{\mathrm{LQG}}(s) &= K_c(sI - \tilde{A} + \tilde{B}K_c + K \longrightarrow_f C)^{-1} K \longleftarrow_f C(sI - \tilde{A})^{-1}\tilde{B} \\ L_{\mathrm{LQG}}(s) &\approx K_c(sI - \tilde{A})^{-1}\tilde{B} \end{aligned} \tag{6.59}$$

to

$$\begin{aligned} L_{LQG}(s) &= K_c\left(sI - A + BK_c + K_f C\right)^{-1} K_f C(sI - A)^{-1}B \\ L_{LQG}(s) &\approx K_c(sI - A)^{-1}B \end{aligned}$$

15) Page 179, change

It is this process that inverts the plant (within the Kalman filter) resulting in recovering the LQR L_{LQR}. It is important to note that as $\rho \to 0$, $\bar{\sigma}(P_f) \to \infty$ and $\underline{\sigma}(P_f) \to 0$, creating a singular covariance matrix. In the next section, we will present the LTR method of Lavretsky [6] which prevents this condition from occurring during the recovery process.

to

$$\underline{\sigma}(P_f) \to 0$$

16) Page 179, change

$$\begin{aligned} u &= -K_c(sI - \tilde{A} + \tilde{B}K_c + K_fC)^{-1}K_f(Cx + v) \\ &= -K_c(sI - \tilde{A} + \tilde{B}K_c + K_fC)^{-1}K_fCx - K_c(sI - \tilde{A} + \tilde{B}K_c + K_fC)^{-1}K_fv \\ &\approx -K_cx - K_c(sI - \tilde{A} + \tilde{B}K_c + K_cF)^{-1}K_fv \end{aligned} \tag{6.61}$$

to

$$\begin{aligned} u &= -K_c\left(sI - \widetilde{A} + \widetilde{B}K_c + K_fC\right)^{-1}K_f(Cx + v) \\ &= -K_c\left(sI - \widetilde{A} + \widetilde{B}K_c + K_fC\right)^{-1}K_fCx - K_c\left(sI - \widetilde{A} + \widetilde{B}K_c + K_fC\right)^{-1}K_fv \\ &\approx -K_cx - K_c\left(sI - \widetilde{A} + \widetilde{B}K_c + K_fC\right)^{-1}K_fv \end{aligned}$$

17) Page 180, change

in which the first term is inverted and canceled $\left(K_f C\right)^{-1} K_f C = I$ resulting in $-K_c x$. However, the second term is not exactly canceled; $\left(K_f C\right)^{-1} K_f \neq I$, and the sensor noise v can be amplified. This feature limits the amount of recovery possible. In the use of this design method for making the LQG system robust, the sensor noise amplification in (6.62) must be examined.

The LQG/LTR controller design, examining the loop properties at the plant input, may be realized through the following synthesis technique:

Step 1: LQR controller design: K_c

Follow the robust servomechanism design approach outlined in Chap. 3. Design LQR weighting matrices Q and R such that the resulting LTFM $L_{\mathrm{LQR}}(s) = K_c \left(sI - \tilde{A}\right)^{-1} \tilde{B}$ meets performance and stability robustness requirements and exhibits the desired bandwidth. The frequency domain properties of the LQG system will not exceed those of the LQR system.

Step 2: Kalman filter design: K_f Design the Kalman filter state estimator using (6.55), with (6.59) defining the plant disturbance covariance. The LTR filter recovery parameter ρ is used to recover the LQR frequency domain characteristics over the frequency range of interest. Examine plant input and output frequency domain criteria and the sensor noise amplification in (6.62) and limit the LTR recovery so that the sensor noise is not amplified.

to

(6.62) change to (6.51)
(6.55) change to (6.54)
(6.59) change to (6.58)
(6.62) change to (6.61)

18) Page 180, change the following text:

in which the first term is inverted and canceled $\left(K_f C\right)^{-1} K_f C = I$ resulting in $-K_c x$. However, the second term is not exactly canceled; $\left(K_f C\right)^{-1} K_f \neq I$, and the sensor noise v can be amplified. This feature limits the amount of recovery possible. In the use of this design method for making the LQG system robust, the sensor noise amplification in (6.62) must be examined.

to

in which the first term approximately equals the state feedback control law, but the second term amplifies the sensor noise. This feature limits the amount of recovery possible. In the use of this design method for making the LQG system robust, the sensor noise amplification in (6.62) must be examined.

19) Page 181, change

where the first gain in K_c multiplies the integral error, and the remaining gains multiply estimates of A_z, q, δ_e, and $\dot{\delta}_e$, respectively.

The measurements provided by an inertial measurement unit, A_{z_m} and q_m, are available for feedback. To design the Kalman filter state estimator, we need models of the process and measurement noise covariance matrices from (6.54). At this flight condition, the process noise modeled in the state equations is

to

(6.53)

20) Page 183, change

6.2 Linear Quadratic Gaussian with Loop Transfer Recovery

where u is formed using (6.65) and was implemented using steady-state matrices obtained from the filter covariance equation

to

(6.64)

21) Page 183, change

$$K_f = P_f C^T {R_0}^{-1}$$

$$P_f = \begin{bmatrix} 2.0442\text{e}-003 & -6.0760\text{e}-006 & 2.4929\text{e}-010 & -5.6592\text{e}-008 \\ -6.0760-006 & 7.8188\text{e}-008 & -1.2264\text{e}-012 & 1.3375\text{e}-010 \\ 2.4929\text{e}-010 & -1.2264\text{e}-012 & 1.2523\text{e}-010 & -5.0000\text{e}-009 \\ -5.6592\text{e}-008 & 1.3375e-010 & -5.0000\text{e}-009 & 3.4544\text{e}-007 \end{bmatrix}$$

$$K_f = \begin{bmatrix} 3.2707-002 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix} \tag{6.68}$$

The controller implementing the robust servomechanism integral control with the

to: need to add the “e” e-006, e-002 to these two numbers

22) Page 185, change

Note that (6.75) is valid for plant models with no D matrix, that is, $D_p = 0$. The first state x_1 is the robust servo integrator, the vector $\hat{x}$ is the estimated state, z_{meas} contains the acceleration and pitch rate measurements, and r is the acceleration command. Writing the controller in a generic form, we have

to

(6.74)

23) Page 185, change

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0854\text{e}+000 & -3.4041\text{e}+002 & 0 & -1.1289\text{e}+001 \\ 0 & 6.8202\text{e}-003 & -1.1116\text{e}+000 & -1.0925\text{e}+000 & 0 \\ 0 & -3.9887\text{e}-009 & 1.2264\text{e}-006 & 0 & 1.0 \\ -3.3010\text{e}+003 & -1.1944\text{e}+003 & 9.3804\text{e}+004 & -2.1408\text{e}+004 & -1.1005\text{e}+002 \end{bmatrix}$$

$$B_{c_1} = \begin{bmatrix} 1.0 & 0 \\ 3.2707e-002 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix}; B_{c_2} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$D_{c_1} = [0 \quad 0]; \; D_{c_2} = [0] \tag{6.76}$$

to

$$\begin{bmatrix} A_c \\ C_c \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0854\text{e}+000 & 3.4041\text{e}+002 & 0 & -1.1289\text{e}+001 \\ 0 & 6.8202\text{e}-003 & -1.1116\text{e}+000 & -1.0925\text{e}+000 & 0 \\ 0 & -3.9887\text{e}-009 & 1.2264\text{e}-006 & 0 & 1.0 \\ -3.3010\text{e}+003 & -1.1944\text{e}+003 & 9.3804\text{e}+004 & -2.1408\text{e}+004 & -1.1005\text{e}+002 \end{bmatrix} \\ [-4.9477\text{e}-001 \quad -1.7903\text{e}-001 \quad 1.4060\text{e}+001 \quad -2.2087\text{e}+000 \quad -1.8035\text{e}-003] \end{bmatrix}$$

$$\begin{bmatrix} B_{c_1} & B_{c_2} \\ D_{c_1} & D_{c_2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1.0 & 0 \\ 3.2707\text{e}-002 & -6.0760\text{e}+000 \\ -9.7217\text{e}-005 & 7.8188\text{e}-002 \\ 3.9887\text{e}-009 & -1.2264\text{e}-006 \\ -9.0547\text{e}-007 & 1.3375\text{e}-004 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ [0 \; 0] & [0] \end{bmatrix}$$

24) Page 186, change

Next, we will analyze the LQG/LTR design in the frequency domain and determine the desired amount of LTR to be applied at this flight condition. Figure 6.13 shows a Nyquist plot of the LQR, LQG, and LQG/LTR designs using values of ρ from (6.74). The red circle is a unit circle centered at (−1,j0) for reference. The LQR locus (blue) demonstrates infinite gain margin (at the plant

to

(6.73)

25) Page 189

To finalize a choice of ρ, the decision should be made on maximizing $\underline{\sigma}(I+L)$ and $\underline{\sigma}(I+L^{-1})$ at the plant input, minimizing $\bar{\sigma}(S)$ and $\bar{\sigma}(T)$ at the plant output, and preventing noise amplification over a frequency range of interest. The following table summarizes these peak values:

Design	$\underline{\sigma}(I+L)$	$\underline{\sigma}(I+L^{-1})$	$\bar{\sigma}(S)$	$\bar{\sigma}(T)$
LQR	1.0000	0.7963	1.4936	1.0480
LQR	0.5506	0.9808	1.0791	1.0000
$\rho = 10^5$	0.5233	0.7136	1.0923	1.0000
$\rho = 10^4$	0.5853	0.6567	1.0599	1.0000
$\rho = 10^3$	0.7920	0.7301	1.4581	1.0000
$\rho = 10^2$	0.9160	0.7715	2.9361	2.1570

From the $\underline{\sigma}(I+L)$ values, we need $\rho \leqslant 10^4$ to meet plant input stability margin requirements. We would like $\underline{\sigma}(I+L^{-1})$ to be as large as possible, which is also satisfied by $\rho \leqslant 10^4$. We would like $\bar{\sigma}(S)$ to be minimized, which points to $\rho = 10^4$ as the desired recovery level. If $\rho = 10^3$, the peak in $\bar{\sigma}(S)$ would be too large. Thus, $\rho = 10^4$ is selected as the design. For comparison, the following table lists the Kalman filter gains:

to

$\underline{\sigma}$

26) Page 190, change

Definition 6.2. *The transfer function $G(s)$ is called strictly positive real if $G(s-\varepsilon)$ is positive real, for some $\varepsilon > 0$.*

For scalar systems ($p = 1$), PR and SPR dynamics have their Nyquist frequency response locus located entirely in the right half complex plane. This condition for $G(s)$ can be satisfied only if the system's relative degree is zero or one. Thus, encirclements of $(-1, j0)$ cannot occur. In other words, such a system will remain stable under a large set of uncertainties, which is a highly desirable property for any system to possess.

to: This text should not be italic

27) Page 191, change

Clearly, if D is the zero matrix, then the SPR conditions (6.81) reduce to

$$\begin{aligned} PA + A^T P &= -L^T L - \varepsilon P \\ PB &= C^T \end{aligned} \tag{6.81}$$

and in this case, setting $\varepsilon = 0$, gives the PR conditions in the form

$$\begin{aligned} PA + A^T P &= -L^T L \\ PB &= C^T \end{aligned} \tag{6.82}$$

to: (6.80), and the text in the boxes should not be italic.

28) Page 192

The first relation in (6.83) is the algebraic Lyapunov equation, and $V(x) = x^T P x$ *is the Lyapunov function [4]. The second relation in (6.83) enables output feedback control design, whereby the system output* $y = Cx$ *can be fed back into the input to control the system, while preserving closed-loop stability. Also, note that the matrices* B *and* C *define the transmission zeros of the system transfer function matrix* $G(s) = C(sI - A)^{-1}B$.

We are going to modify the LQG/LTR design such that, for a class of restricted systems, the PR property is obtained asymptotically, $P_v B_v \to C^T$, *with the positive tuning parameter* $v \to 0$. *In addition, we shall ensure that* P_v *remains symmetric and strictly positive definite, uniformly in* v. *These are the distinguishing features of LTRLM design. Similar to the previous section, in this design, the Kalman filter is no longer treated as a filter. It will continue to estimate the system state and serve as a dynamic compensator, tuned to improve the frequency domain properties of the system. The Gaussian covariance matrices for* w *and* v *are altered significantly to improve the controller robustness and to limit sensor noise amplification. So, these matrices no longer "model" the stochastic processes of the system.*

We formulate the LTRLM design approach using the linear-time-invariant Gaussian design model,

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \end{aligned} \tag{6.83}$$

where w *and* v *are zero mean, white, uncorrelated Gaussian random processes with covariances given by*

$$\begin{aligned} E\{w(t)w^T(\tau)\} &= Q_0\delta(t-\tau) \\ E\{v(t)v^T(\tau)\} &= R_0\delta(t-\tau) \end{aligned} \tag{6.84}$$

The state estimate $\hat{x}$ *is formed as before, using the state estimator,*

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y_{\text{meas}} - \hat{y}) \tag{6.85}$$

and the control input is calculated using the LQR state feedback gain matrix K_c, *with the estimated state feedback* $\hat{x}$.

$$u = -K_c\hat{x} \tag{6.86}$$

In LTRLM, we parameterize the process and measurement noise covariance matrices using a positive scalar v,

to: Change (6.83) to (6.82). The text on this page should not be italic.

29) Page 193, change

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\,\bar{B}^T, \quad R_v = \frac{v}{v+1}R_0 \tag{6.87}$$

where $\bar{B}$ *is a matrix formed by adding "fictitious" columns to* B*, to make* $\bar{B} = [B \quad X]$ *have its column rank equal to the row rank of* C*, such that* $C\bar{B}$ *becomes invertible and the corresponding extended system* $C(sI - A)^{-1}\bar{B}$ *is minimum phase, that is, all its transmission zeros are located in the left half complex plane. This is the "squaring-up" step of the method. Substituting the weights from (6.88) into the filter Riccati equation, we get*

$$P_v A^T + A P_v - \left(1 + \frac{1}{v}\right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v}\right)\bar{B}\,\bar{B}^T = 0 \tag{6.88}$$

or, equivalently

$$P_v A^T + A P_v - P_v C^T R_0^{-1} C P_v + Q_0 + \bar{B}\,\bar{B}^T + \frac{1}{v}\left[\bar{B}\,\bar{B}^T - P_v C^T R_0^{-1} C P_v\right] = 0 \tag{6.89}$$

The gains in (6.86) are computed as

$$K_f = P_v C^T {R_v}^{-1} \tag{6.90}$$

Now as $v \to 0$*, one can show that the filter covariance matrix* P_v *asymptotically approaches a constant symmetric positive definite matrix* P_0*, that is,*

$$P_0 = \lim_{v \to 0} P_v = \lim_{v \to 0} P_v^T = P_0^T > 0 \tag{6.91}$$

This behavior is in contrast to the previous section, whereas the LTR parameter $\rho \to 0$*,* $\bar{\sigma}(P_f) \to \infty$*,* $\underline{\sigma}(P_f) \to 0$*, and the* P_f *matrix became singular.*

The important properties of P_0 *in (6.92) are listed below without proof (see Chap. 13, Theorem 13.1 for formal derivations):*

- P_0 *is the unique symmetric* strictly *positive definite solution of the following algebraic Lyapunov equation*

$$P_0\left(A - C^T R_0^{-1} C P_1\right)^T + \left(A - C^T R_0^{-1} C P_1\right) P_0 + Q_0 = 0 \tag{6.92}$$

to: (6.88) to (6.87), (6.86) to (6.85), (6.92) to (6.91). This page should not be italic.

30) Page 194, change

194 6 Output Feedback Control

- *There exists a unitary matrix* $W \in R^{m \times m}$ *such that*

$$P_0 C^T = \bar{B} W^T R_0^{\frac{1}{2}} \tag{6.93}$$

- *The unitary matrix* W *in (6.94) can be chosen as*

$$W = (U V)^T \tag{6.94}$$

where U *and* V *are two unitary matrices defined by the singular value decomposition,*

$$\bar{B}^T C^T R_0^{-\frac{1}{2}} = U \Sigma V \tag{6.95}$$

and Σ *represents the diagonal matrix of the corresponding singular values.*

For minimum phase systems, the SPR property is implied by (6.94). What the LTRLM design is trying to do is to shape the transmission zeros of the state estimator, such that the original system with the extended input becomes SPR asymptotically, as $v \rightarrow 0$. To do this, we "square-up the system" by adding extra columns to B (to form $\bar{B}$) and then apply the LTR tuning process, whereby we decrease the tuning parameter v in (6.88), until the system becomes almost SPR.

It was discussed earlier in Chap. 2 that in the LQR design problem, with the penalty matrix Q factored as $Q = Q^{\frac{T}{2}} Q^{\frac{1}{2}}$, the poles of the closed-loop system, $\lambda(A - BK_c)$, would approach the transmission zeros defined by $Q^{\frac{1}{2}}(sI - A)^{-1}B$ asymptotically as the gains grew large. If no finite transmission zeros existed, the roots would form a Butterworth pattern (or combinations of Butterworth patterns) in the left half complex plane. Thus, by the proper selection of Q, the designer places these zeros to achieve the desired response of the system. So, the selection of the LQR penalty matrix is a key tuning mechanism in the LQR controller design.

This same basic idea is in work under LTRLM. For the state estimator (aka Kalman filter), the process covariance Q_f is the equivalent to the LQR penalty matrix. Factoring the process covariance Q_f as $Q_f = L^T L$, the eigenvalues of the Kalman filter, $\lambda(A - K_f C)$, will approach the finite transmission zeros defined by $C(sI - A)^{-1}L$. Thus, the selection of the process covariance Q_f is an ideal tuning mechanism in the design of the LTRLM controller. Placing the zeros of the system in a desirable location is the key to achieving a robust design. This is achieved through the modified process covariance and measurement noise matrices in (6.88).

to: (6.94) to (6.93), (6.94) to (6.93), (6.88) to (6.87), (6.88) to (6.87). The text in the box should not be italic.

31) Page 196, change

Step 1: LQR controller design: K_c

Follow the robust servomechanism design approach outlined in Chap. 3. Design LQR weighting matrices Q and R, such that the resulting loop gain $L_{\mathrm{LQR}}(s) = K_c\left(sI - \tilde{A}\right)^{-1}\tilde{B}$ meets performance and stability robustness requirements and exhibits the desired bandwidth.

Step 2: State estimator/Kalman filter design: K_f

Select columns X to make $\bar{B} = [B \quad X]$ have column rank equal to the row rank of C and to make the extended system minimum phase. Design the Kalman filter/state estimator using (6.89), with (6.88) defining the plant process and measurement noise covariance matrices. The LTR parameter v is used to recover the LQR frequency domain characteristics over the frequency range of interest. Ad hoc adjustment of the sensor noise covariance magnitude may be needed to scale the Kalman gains to prevent large gains from occurring. Examine plant input and output frequency domain criteria and the sensor noise amplification in and limit the LTR recovery so that the sensor noise is not amplified.

to:), (6.89) to (6.88), (6.88) to (6.87).

32) Page 197, change

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \tag{6.102}$$

Using the gains from (6.101), the state feedback controller is

$$\begin{aligned} \dot{x}_c &= [0]x_c + [1 \quad 0]y + [-1]r \\ u &= [-0.31623]x_c + [33.261 \quad 6.7127]y + [0]r \end{aligned} \tag{6.103}$$

to: (6.101) to (6.100)

33) Page 198, change

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \tag{6.106}$$

where u is formed using (6.102) and was implemented using steady-state matrices obtained from the filter covariance equation

$$\begin{aligned} 0 &= AP_f + P_f A^T + Q_v - P_f C^T {R_v}^{-1} C P_f \\ K_f &= P_f C^T {R_v}^{-1} \end{aligned} \tag{6.107}$$

The steady-state covariance and Kalman filter gains design (using Q_0 and R_0) are

$$\begin{aligned} P_f &= \begin{bmatrix} 9.5843\text{e}-006 & 3.8344\text{e}-007 \\ 3.8344\text{e}-007 & 4.8957\text{e}-005 \end{bmatrix} \\ K_f &= \begin{bmatrix} -0.053132 & 0.38344 \\ -0.0021257 & 48.957 \end{bmatrix} \end{aligned} \tag{6.108}$$

To analyze this observer-based design (Kalman filter), we will implement the controller in our standard model (6.103). For the LQG controller, the RSLQR control law is given by

$$\begin{aligned} u &= -K_1 \int e - K_x \hat{x} \\ e &= y_c - r = A_z - A_{z_{cmd}} \end{aligned} \tag{6.109}$$

where $\hat{x}$ is the estimated state, and the RSLQR gain matrix is partitioned as $K_c = [K_1 \quad K_x]$. To form the estimated state, we need to substitute the control (6.110) into the state estimator (6.107). Doing so gives

to: (6.102) to (6.101), (6.103) to (6.102), (6.110) to (6.109), (6.107) to (6.106)

34) Page 199, change

The LQG controller states are $x_c = \begin{bmatrix} \int e & \hat{x} \end{bmatrix}^T$. The controller state space model using (6.110) and (6.111) is

$$\begin{aligned} \begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ K_f \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} r \\ u &= -[K_1 \quad K_x] \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0] y_{meas} + [0] r \end{aligned} \tag{6.111}$$

where A_{21} and A_{22} are defined as in (6.111). Substituting the gains into (6.112), we have

to: (6.110) to (6.109), (6.111) to (6.110), (6.111) to (6.110), (6.112) to (6.111)

35) Page 200, change

Note the magnitude increase in the gain $K_f(2,2)$.

The last controller in this example uses the LTRLM. The first step in the design process is to design the LQR control law. We will use the RSLQR controller from (6.102). The second step is to select columns X to make $\bar{B} = [B_p \quad X]$ have column rank equal to the row rank of C_p. To complete this design, we must look at the numbers within these matrices:

to: (6.102) to (6.101),

36) Page 201, change

Solving for the steady-state covariance and gain matrix from (6.108) yields

$$P_f = \begin{bmatrix} 0.011312 & -3.5561\text{e}-005 \\ -3.5561\text{e}-005 & 0.018303 \end{bmatrix}$$
$$K_f = \begin{bmatrix} -0.35116 & -0.19914 \\ 0.001104 & 102.5 \end{bmatrix} \tag{6.121}$$

to: (6.108) to (6.107)

37) Page 201, change

The controller is formed by substituting the gains K_f into (6.112) and results in

$$\begin{bmatrix} e \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -19.48 & 0.16584 \\ 0.36948 & -42.794 & -246.8 \end{bmatrix} \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -0.053183 & 0.83416 \\ -0.0043243 & 237.93 \end{bmatrix} y_{meas} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} r$$
$$u = [-0.31623 \; 33.261 \; 6.7127] \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix} + [0\,0] y_{\text{meas}} + [0] r \tag{6.122}$$

to: (6.112) to (6.111)

38) Page 201, change

Solving for the steady-state covariance and gain matrices from (6.108) yields

$$P_f = \begin{bmatrix} 0.00017018 & 0.0017429 \\ 0.0017429 & 0.15898 \end{bmatrix}$$
$$K_f = \begin{bmatrix} -0.0052832 & 9.7605 \\ -0.054109 & 890.31 \end{bmatrix} \tag{6.124}$$

The controller is formed by substituting the gains K_f into (6.112) and results in

to: (6.108) to (6.107), (6.112) to (6.111)

39) Page 205, change

Exercise 6.1. Consider the unstable longitudinal dynamics model, as defined in Example 6.1, where $x = \begin{bmatrix} \alpha & q & \delta_e & \dot{\delta}_e \end{bmatrix}^T$. The matrices for the control design model $\dot{x} = A_p x + B_p u$ are

$$\begin{bmatrix} A_p & B_p \end{bmatrix} = \left[\begin{bmatrix} -1.3046\text{e} & 1.0 & -0.2.1420 & 0 \\ 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 0 & -12769. & -135.6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12769 \end{bmatrix} \right]$$

to: remove the "e"

40) Page 206, change

(c) Compute the eigenstructure for (a) and (b) to show that the dominant eigenvalues are retained. Analyze this design in the frequency domain. Compute Nyquist, Bode, $\bar{\sigma}[I + L]$, $\bar{\sigma}[I + L^{-1}]$ frequency responses for a) and b) at the plant input. Compute $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ frequency responses for a) and b) at the plant output for the α loop. Compute the loop gain crossover frequency and singular value stability margins for the design.

to

$\underline{\sigma}$

41) Page 206 change

singular value stability margins for both designs. Determine the impact of using the Kalman filter estimator on the stability robustness of the system.

(d) Use the LTR method of Sect. 6.2, (6.59) to recover the frequency domain properties of the state feedback design in the LQG design. Evaluate the design in the frequency domain as in (c). Compute the maximum singular value of the noise-to-control transfer function matrix frequency response to examine the noise amplification in the resulting LQG/LTR design.

to: (6.59) to (6.58)

42) Page 207

(b) Simulate the LQG design and compare it to the state feedback design.

(c) Analyze this LQG design in the frequency domain. Compute Nyquist, Bode, $\underline{\sigma}[I + L]$, $\underline{\sigma}[I + L^{-1}]$ frequency responses for the LQG and state feedback at the plant input. Compute $\bar{\sigma}[S]$ and $\bar{\sigma}[T]$ frequency responses for (a) and (b) at the plant output for the α loop. Compute the loop gain crossover frequency and singular value stability margins for both designs. Determine the impact of using the Kalman filter estimator on the stability robustness of the system.

(d) Use the Loop Transfer Recovery method of Lavretsky, Sect. 6.2 (6.88), to recover the frequency domain properties of the state feedback design in the LQG design. Evaluate the design in the frequency domain as in (c). Compute the maximum singular value of the noise-to-control transfer function matrix frequency response to examine the noise amplification in the resulting LQG/LTR design.

to

$\underline{\sigma}$, (6.88) to (6.87)

ERRATA LIST

Robust and Adaptive Control with Aerospace Applications –
1st Edition, Springer, 2013
Eugene Lavretsky

Revised on 27 Jul 2013

Part II

Chapter 7

1. Page 222: Exercise 7.3: Change "(7.13), (7.14), (7.15), (7.16), (7.17), (7.18), (7.19), (7.20), (7.21), (7.22), and (7.23)." to "(7.13) through (7.23)."
2. Page 239: Example 8.9: Change $6\sin t - 6t\cos t - t^2 \le 6 + \underbrace{\left(t - t^2\right)}_{\le \frac{1}{4}} \le 6.25$ to

   $6\sin t - 6t\cos t - t^2 \le 6 + \underbrace{\left(6t - t^2\right)}_{\le 9} \le 15$
3. Page 239: Example 8.9: Change

   $|x(t)| \le |x(t_0)| \underbrace{\exp\left(6.25 - 6\sin t_0 + 6t_0\cos t_0 + t_0^2\right)}_{c(t_0)} = |x(t_0)|\, c(t_0)$ to

   $|x(t)| \le |x(t_0)| \underbrace{\exp\left(15 - 6\sin t_0 + 6t_0\cos t_0 + t_0^2\right)}_{c(t_0)} = |x(t_0)|\, c(t_0)$

Chapter 8

1. Page 261: Example 8.10: Change $\dot{x}_{ref} = Ax_{ref} + br$ to $\dot{x}_{ref} = Ax_{ref} + bK_r r$
2. Page 258: Change sentence

   "regressor vector", which is assumed to be uniformly bounded.

   to
   "regressor vector", which is assumed to be uniformly bounded and Lipschitz-continuous in t.

3. Page 258: Change $V(e, \Delta K) = e^T P\, e + \Delta K^T \Delta K$ to $V(e, \Delta K) = e^T P e + \frac{1}{\gamma} \Delta K^T \Delta K$.
4. Page 258: Change the next equation for $\dot{V} e, \Delta K = ...$ to

$$\dot{V}(e, \Delta K) = \dot{e}^T P\, e + e^T P\, \dot{e} + 2\frac{1}{\gamma} \Delta K^T \dot{\hat{K}} = \left(A\, e + b\, \Delta K^T \Phi\right)^T$$
$$P\, e + e^T P\left(A\, e + b\, \Delta K^T \Phi\right) - 2\Delta K^T \Phi e^T P\, b = -e^T Q\, e \leq 0$$

5. Page 261: Change equation $\dot{x}_{ref} = A x_{ref} + b\, r$ to $\dot{x}_{ref} = A x_{ref} + b\, K_r r$.

Chapter 9

1. Page 282: Eq. (9.47): Right-justify the equation number (i.e., move it to the right).

Chapter 10

1. Page 311: Table 2.10: Change the third equation from the bottom to: $u = \left(I_{m \times m} - \hat{K}_u^T\right) u_{bl} - \hat{\Theta}^T \Phi\left(x_p\right)$.
   This equation in the current book version has an incomplete sub-index in I.
2. Page 314: Exercise 10.1: In the problem statement, replace “have” with “has”.
3. Page 314: Exercise 10.2: Add the minus sign to the last equation in the problem statement: $\hat{K}_x(0) = -K_x$.

Chapter 11

1. Page 325: Replace text and equations starting from Eq. 11.26 through Eq. 11.29 with:

Using completion of squares, the sum of the first and the second terms in (11.25) can be transformed into

$$
\begin{aligned}
&-\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\lambda_{\max}(P)\xi_{\max} \\
&= -\lambda_{\min}(Q)\left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)}\right)^2 + \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)}
\end{aligned}
$$

Similarly, the sum of the third and the fourth terms in (11.25) can be written as

$$
\begin{aligned}
&-2\sigma\|\Delta\Theta\|_F^2\,\Lambda_{\min} + 2\sigma\|\Delta\Theta\|_F\|\Theta\|_F\|\Lambda\|_F \\
&= -2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}}
\end{aligned}
$$

Substituting these two expressions back into (11.25), gives

$$
\begin{aligned}
\dot{V}(e,\Delta\Theta) \le\ & -\lambda_{\min}(Q)\left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)}\right)^2 + \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} \\
& -2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}}
\end{aligned}
\tag{11.26}
$$

Hence, $\dot{V}(e,\Delta\Theta) < 0$ if at least one of the following two relations take place:

$$
\lambda_{\min}(Q)\left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)}\right)^2 - \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} - \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} > 0
$$

OR (11.27)

$$
2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 - \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} - \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} > 0
$$

or equivalently, when

$$\|e\| > \sqrt{\frac{1}{\lambda_{\min}(Q)}\left(\frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}}\right)} + \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)} = c_1$$

OR (11.28)

$$\|\Delta\Theta\|_F > \sqrt{\frac{1}{2\sigma\Lambda_{\min}}\left(\frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}}\right)} + \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}} = c_2$$

In other words, $\dot{V}(e, \Delta\Theta) < 0$ outside of the compact (closed and bounded) set $\Omega \subset \left(R^n \times R^{N\times m}\right)$ defined below.

$$\Omega = \left\{(e, \Delta\Theta) : \left(\|e\| \le c_1\right) \wedge \left(\|\Delta\Theta\|_F \le c_2\right)\right\} \quad (11.29)$$

2. Page 327: Change section number from "11.3" to subsection # "11.2.3"
3. Page 328: Replace the first sentence and the second sentences with the sentence shown below in yellow:

$\sigma\|e^T P B\|$. This fact allows to arrive at a compact set [4], outside of which $\dot{V}(e, \Delta\Theta) < 0$. Once again, we can claim UUB of all trajectories. This completes the stability analysis for the $e-$ modification with a guaranteed UUB-type output tracking performance.

4. Page 329: Change Eq. (11.59) to:

$$\nabla f_j = \frac{(1+\varepsilon_j^{\Theta})}{\varepsilon_j^{\Theta}(\Theta_j^{\max})^2}\nabla\left[||\,\widehat{\Theta}_j \wedge||^2\right] = \frac{2(1+\varepsilon_j^{\Theta})}{\varepsilon_j^{\Theta}(\Theta_j^{\max})^2}\,\widehat{\Theta}_j.$$

Chapter 12

1. Page 364: Replace δ with δR in Fig 12.5, as shown below:

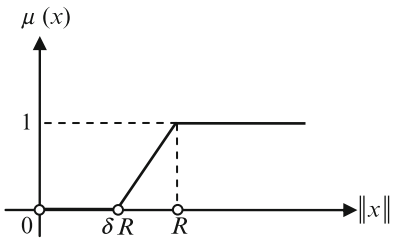


Fig. 12.5 State modulation function

2. Page 372: Change “Example 12.3” to “Example 12.1”
3. Page 373: In the sentence “In addition, we impose a restriction on the runway distance …” : Change τ_{fl} to $4\tau_h$.
4. Page 374: Change the second equation as shown below (change B(2,2) to 0.00044):

$$\underbrace{\begin{pmatrix}\dot{V}\\ \dot{\alpha}\\ \dot{q}\\ \dot{\theta}\\ \dot{h}\end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix}-0.038 & 18.984 & 0 & -32.174 & 0\\ -0.001 & -0.632 & 1 & 0 & 0\\ 0 & -0.759 & -0.518 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & -250 & 0 & 250 & 0\end{pmatrix}}_{A} \underbrace{\begin{pmatrix}V\\ \alpha\\ q\\ \theta\\ h\end{pmatrix}}_{x}$$
$$+ \underbrace{\begin{pmatrix}10.1 & 0\\ 0 & 0.00044\\ 0.025 & -0.011\\ 0 & 0\\ 0 & 0\end{pmatrix}}_{B} \underbrace{\begin{pmatrix}\delta_{th}\\ \delta_e\end{pmatrix}}_{u} \Leftrightarrow \boxed{\dot{x} = Ax + Bu}$$

5. Page 375: Change equation at the top of the page as shown below (change B(2,2) to 0.00044):

$$\underbrace{\begin{pmatrix}\dot{V}\\ \dot{\alpha}\\ \dot{q}\\ \dot{\theta}\\ \dot{h}\end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix}-0.038 & 18.984 & 0 & -32.174 & 0\\ -0.001 & -0.632 & 1 & 0 & 0\\ 0 & -0.759 & -0.518 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & -250 & 0 & 250 & 0\end{pmatrix}}_{A} \underbrace{\begin{pmatrix}V\\ \alpha\\ q\\ \theta\\ h\end{pmatrix}}_{x}$$
$$+ \underbrace{\begin{pmatrix}10.1 & 0\\ 0 & 0.00044\\ 0.025 & -0.011\\ 0 & 0\\ 0 & 0\end{pmatrix}}_{B} \underbrace{\begin{pmatrix}\delta_{th}\\ \delta_e\end{pmatrix}}_{u} + \underbrace{\begin{pmatrix}-18.984\\ 0.632\\ 0.759\\ 0\\ 0\end{pmatrix}}_{B_g} \alpha_g(h)$$

6. Page 376: Change 4th from the bottom equation as shown below:

$$Q_{lqr} = \text{diag}(\,0.2 \quad 0 \quad 0 \quad 0 \quad 1\,), \quad R_{lqr} = \text{diag}(\,10 \quad 10\,)$$

Chapter 13

1. Page 406: Insert space between "y_{cmd}" and "y" in **Eq. 13.87**. There are two equations there, not one, as shown below.

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{m\times m} & C_p \\ 0_{n_p\times m} & A_p \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0_{m\times m} \\ B_p \end{pmatrix}}_{B} \Lambda \left(u + \overbrace{\Theta_d^T \Phi_d(x_p)}^{d(x_p)} \right)$$

$$+ \underbrace{\begin{pmatrix} -I_{m\times m} \\ 0_{n_p\times m} \end{pmatrix}}_{B_{ref}} y_{cmd}\, y = \underbrace{\begin{pmatrix} 0_{m\times m} & C_p \end{pmatrix}}_{C} x \qquad (13.87)$$

2. Page 409: Move superscript "T" in Eq. 13.99 to the right and outside of parenthesis, as in:

$$\dot{x} = A_{ref}\, x - B\, \Lambda \underbrace{\left(\hat{\Theta} - \Theta\right)^T}_{\Delta\Theta} \Phi(x) + B_{ref}\, y_{cmd} \qquad (13.99)$$

3. Page 416: Exercise 13.4: Change "Example 13.3" to "Example 10.2".

Chapter 14

1. Page 429: **Eq. 14.46**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) = -\left(1 + \frac{1}{v}\right) e_y^T R_0^{-1} e_y - e_x^T \widetilde{P}_v Q_0 \widetilde{P}_v e_x$$

$$-\left(1 + \frac{1}{v}\right) \left\| B^T \widetilde{P}_v e_x \right\|^2 - 2\eta e_x^T \widetilde{P}_v e_x + 2 e_x^T \widetilde{P}_v B \Lambda g$$

$$+2\,\mathrm{trace}\left(\Lambda \Delta\overline{\Theta}^T \left\{ \Gamma_\Theta^{-1} \dot{\overline{\Theta}} + \overline{\Phi}(\hat{x}, u_{bl}) \underbrace{\left(e_x^T C^T\right)}_{e_y^T} R_0^{-\frac{1}{2}} W S^T \right\} \right)$$

$$+2 e_x^T \mathrm{O}(v) \Lambda \Delta\overline{\Theta}^T \overline{\Phi}(\hat{x}, u_{bl})$$

2. Page 430: **Eq. 14.48**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -2\eta\,\lambda_{\min}\left(\widetilde{P}_v\right)\|e_x\|^2 - \lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\widetilde{P}_v\right)\|e_x\|^2$$
$$-\left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)\left\|B^T\widetilde{P}_v\,e_x\right\|^2 + 2\Lambda_{\max}\,k_g\left\|B^T\widetilde{P}_v\,e_x\right\|\|e_x\|$$
$$+2v\,\|e_x\|\,k\,\Lambda_{\max}\,\Delta\overline{\Theta}_{\max}\left\|\overline{\Phi}(\hat{x}, u_{bl})\right\|$$

3. Page 430: **Eq. 14.49**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -\left(2\eta + \lambda_{\min}(Q_0)\,\lambda_{\min}\left(\widetilde{P}_v\right)\right)\lambda_{\min}\left(\widetilde{P}_v\right)\|e_x\|^2$$
$$-\left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)w^2 + 2\,\Lambda_{\max}\,k_g w\,\|e_x\|$$
$$+2v\,\|e_x\|\,k\,\Lambda_{\max}\,\Delta\overline{\Theta}_{\max}\left\|\overline{\Phi}(\hat{x}, u_{bl})\right\|$$

4. Page 430: **Eq. 14.50**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -\left(2\eta + \lambda_{\min}(Q_0)\,\lambda_{\min}\left(\widetilde{P}_0\right)\right)\lambda_{\min}\left(\widetilde{P}_0\right)\|e_x\|^2$$
$$-\left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2 - \left(1+\frac{1}{v}\right)w^2 + 2\,\Lambda_{\max}\,k_g w\,\|e_x\|$$
$$+2v\,\|e_x\|\,k\,\Lambda_{\max}\,\Delta\overline{\Theta}_{\max}\left\|\overline{\Phi}(\hat{x}, u_{bl})\right\|$$

5. Page 431: **Eq. 14.52**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -\left(1+\frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\|e_y\|^2$$
$$-\left[\left(2\eta + \lambda_{\min}(Q_0)\,\lambda_{\min}\left(\widetilde{P}_0\right)\right)\lambda_{\min}\left(\widetilde{P}_0\right) - 2v\,k\,\Lambda_{\max}\,\Delta\overline{\Theta}_{\max}\,b_2\right]\|e_x\|^2$$
$$-\left(1+\frac{1}{v}\right)w^2 + 2\,\Lambda_{\max}\,k_g w\,\|e_x\| + 2v\,\Lambda_{\max}\,k\,\Delta\overline{\Theta}_{\max}\,b_1\,\|e_x\|$$

6. Page 431: **Eq. 14.53**: Change to:

$$c_1 = \lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\widetilde{P}_0\right) - 2\,v\,k\,\Lambda_{\max}\,\Delta\Theta_{\max}\,b_2, \quad c_2 = \Lambda_{\max}\,k_g$$
$$c_3 = 1 + \frac{1}{v}, \quad c_4 = v\,\Lambda_{\max}\,k\,\Delta\Theta_{\max}\,b_1$$

7. Page 431: **Eq. 14.54**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -c_3\,\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2 - 2\,\eta\,\lambda_{\min}^2\left(\widetilde{P}_0\right)\|e_x\|^2$$
$$-\underbrace{\left[c_1\,\|e_x\|^2 - 2\,c_2\,\|e_x\|\,w + c_3\,w^2 - 2\,c_4\,\|e_x\|\right]}_{\varphi(\|e_x\|,w)}$$
$$= -c_3\,\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2 - 2\,\eta\,\lambda_{\min}^2\left(\widetilde{P}_0\right)\|e_x\|^2 - \varphi(\|e_x\|, w)$$

8. Page 432: **Eq. 14.61**: Change to:

$$v_1 = \frac{\lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\widetilde{P}_0\right)}{2\,k\,\Lambda_{\max}\,\Delta\overline{\Theta}_{\max}\,b_2}$$

9. Page 432: **Eq. 14.62**: Change to:

$$\det C = c_1\,c_3 - c_2^2$$
$$= \left(1 + \frac{1}{v}\right)\left[\lambda_{\min}(Q_0)\,\lambda_{\min}^2\left(\widetilde{P}_0\right) - 2\,v\,k\,\Lambda_{\max}\,\Delta\Theta_{\max}\,b_2\right] - \Lambda_{\max}^2\,k_g^2 = \mathrm{O}\left(\frac{1}{v}\right)$$

10. Page 433: **Eq. 14.66**: Change to:

$$\dot{V}\left(e_x, \Delta\overline{\Theta}\right) \leq -\left(1 + \frac{1}{v}\right)\lambda_{\min}\left(R_0^{-1}\right)\left\|e_y\right\|^2 - 2\,\eta\,\lambda_{\min}^2\left(\widetilde{P}_0\right)\|e_x\|^2 + |\varphi_{\min}(v)|$$

11. Page 433: **Eq. 14.67**: Change to:

$$\Omega_{e_x} = \left\{ e_x : \quad \|e_x\|^2 \leq \frac{|\varphi_{\min}(v)|}{2\eta\lambda_{\min}^2\left(\widetilde{P}_0\right)} = r_{v,\eta}^2 = \mathrm{O}\left(\frac{v^2}{\eta}\right) \right\}$$

12. Page 435: **Table 14.1**: Change 11th and 12th equations in the table as shown below:

Output selection matrix for adaptive laws	$S = \left(I_{m\times m} \quad 0_{m\times(p-m)}\right)$
Singular value decomposition	$\bar{B}^T C^T R_0^{-\frac{1}{2}} = U\Sigma V$

Back cover

Second paragraph: Change “The text is a **three**-part treatment” to “The text is a **two**-part treatment”."Errata List",6,1

Index

E. Lavretsky and K.A. Wise, *Robust and Adaptive Control*, Advanced Textbooks in Control and Signal Processing, DOI 10.1007/978-1-4471-4396-3,