LIUPING WANG

PID CONTROL SYSTEM DESIGN AND AUTOMATIC TUNING USING MATLAB/SIMULINK

with website

IEEE PRESS

WILEY

PID Control System Design and Automatic Tuning using MATLAB/Simulink

PID Control System Design and Automatic Tuning using MATLAB/Simulink

Liuping Wang
RMIT University
Australia

WILEY

IEEE PRESS

This edition first published 2020

Registered Offices
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office
The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Library of Congress Cataloging-in-Publication Data applied for

Hardback ISBN: 9781119469346

Cover Design: Wiley
Cover Images: © Panuwatccn/Shutterstock, © LogicBomb/Shutterstock

Set in 10/12pt WarnockPro by SPi Global, Chennai, India

Printed and bound by CPI Group (UK) Ltd, Croydon, CR0 4YY

10 9 8 7 6 5 4 3 2 1

For
Robin
Jianshe
and all my friends

Contents

Preface *xv*
Acknowledgment *xvii*
List of Symbols and Acronyms *xix*
About the Companion Website *xxi*

1 Basics of PID Control *1*
1.1 Introduction *1*
1.2 PID Controller Structure *1*
1.2.1 Proportional Controller *1*
1.2.2 Proportional Plus Derivative Controller *3*
1.2.3 Proportional Plus Integral Controller *5*
1.2.4 PID Controllers *9*
1.2.5 The Commercial PID Controller Structure *12*
1.2.6 Food for Thought *13*
1.3 Classical Tuning Rules for PID Controllers *13*
1.3.1 Ziegler–Nichols Oscillation Based Tuning Rules *13*
1.3.2 Tuning Rules based on the First Order Plus Delay Model *15*
1.3.3 Food for Thought *17*
1.4 Model Based PID Controller Tuning Rules *18*
1.4.1 IMC-PID Controller Tuning Rules *18*
1.4.2 Padula and Visioli Tuning Rules *19*
1.4.3 Wang and Cluett Tuning Rules *20*
1.4.4 Food for Thought *21*
1.5 Examples for Evaluations of the Tuning Rules *21*
1.5.1 Examples for Evaluating the Tuning Rules *21*
1.5.2 Fired Heater Control Example *25*
1.6 Summary *27*
1.7 Further Reading *28*
Problems *28*

2 Closed-loop Performance and Stability *31*
2.1 Introduction *31*
2.2 Routh–Hurwitz Stability Criterion *31*
2.2.1 Determining Closed-loop Poles *32*
2.2.2 Routh–Hurwitz Stability Criterion *33*

2.2.3 Food for Thought *36*
2.3 Nyquist Stability Criterion *36*
2.3.1 Nyquist Diagram *36*
2.3.1.1 Gain Margin *38*
2.3.1.2 Phase Margin *38*
2.3.1.3 Delay Margin *38*
2.3.2 Rework of Tuning Rules based PID Controllers *40*
2.3.3 Food for Thought *42*
2.4 Control System Structures and Sensitivity Functions *42*
2.4.1 One Degree of Freedom Control System Structure *43*
2.4.2 Two Degrees of Freedom Design *44*
2.4.2.1 Two degrees of freedom implementation of PI controllers *45*
2.4.3 Sensitivity Functions in Feedback Control *45*
2.4.4 Food for Thought *47*
2.5 Reference Following and Disturbance Rejection *47*
2.5.1 Closed-loop Bandwidth *47*
2.5.2 Reference Following and Disturbance Rejection with PID Controllers *50*
2.5.3 Reference Following and Disturbance Rejection with Resonant Controllers *53*
2.5.4 Food for Thought *54*
2.6 Disturbance Rejection and Noise Attenuation *54*
2.6.1 Conflict between Disturbance Rejection and Noise Attenuation *54*
2.6.2 PID Controller for Disturbance Rejection and Noise Attenuation *55*
2.6.3 Food for Thought *58*
2.7 Robust Stability and Robust Performance *59*
2.7.1 Modeling Errors *59*
2.7.2 Robust Stability *60*
2.7.3 Case Study: Robust Control of Polymer Reactor *62*
2.7.4 Food for Thought *65*
2.8 Summary *65*
2.9 Further Reading *67*
Problems *67*

3 Model-Based PID and Resonant Controller Design *71*
3.1 Introduction *71*
3.2 PI Controller Design *71*
3.2.1 Desired Closed-loop Performance Specification *71*
3.2.2 Model and Controller Structures *72*
3.2.3 Closed-loop Transfer Functions for Different Configurations *75*
3.2.4 Food for Thought *77*
3.3 Model Based Design for PID Controllers *78*
3.3.1 PD Controller Design *78*
3.3.2 Analytical Examples for Ideal PID with Pole-zero Cancellation *81*
3.3.3 Analytical Examples for PID Controllers with Filters *84*
3.3.4 PID Controller Design without Pole–Zero Cancellation *92*
3.3.5 MATLAB Tutorial on Solution of a PID Controller with Filter *94*
3.3.6 Food for Thought *95*

3.4 Resonant Controller Design 96
3.4.1 Resonant Controller Design 96
3.4.2 Steady-state Error Analysis 97
3.4.3 Pole–Zero Cancellation in the Design of a Resonant Controller 99
3.4.4 Food for Thought 101
3.5 Feedforward Control 102
3.5.1 Basic Ideas about Feedforward Control 102
3.5.2 Three Springs and Double Mass System 103
3.5.3 Food for Thought 108
3.6 Summary 108
3.7 Further Reading 108
Problems 109

4 Implementation of PID Controllers 113
4.1 Introduction 113
4.2 Scenario of a PID Controller at work 113
4.3 PID Controller Implementation using the Position Form 114
4.3.1 The Steady-state Information Needed 114
4.3.2 Discretization of a PID Controller 115
4.3.3 Food for Thought 116
4.4 PID Controller Implementation using the Velocity Form 117
4.4.1 Discretization of a PI Controller 117
4.4.2 Discretization of a PID Controller using the Velocity Form 119
4.4.3 Improving Accuracy in a Slower Sampling Environment 120
4.4.4 Food for Thought 122
4.5 Anti-windup Implementation using the Position Form 122
4.5.1 Integrator Windup Scenario 122
4.5.2 Anti-windup Mechanisms in the Position Form of PI Controllers 124
4.5.3 Food for Thought 125
4.6 Anti-windup Mechanisms in the Velocity Form 126
4.6.1 Anti-windup Mechanism on the Amplitude of the Control Signal 126
4.6.2 Limits on the Rate of Change of the Control Signal 129
4.6.3 Food for Thought 129
4.7 Tutorial on PID Anti-windup Implementation 130
4.8 Dealing with Other Implementation Issues 133
4.8.1 Plant Start-up 134
4.8.2 Dealing with Quantization Errors in PID Controller Implementation 135
4.9 Summary 136
4.10 Further Reading 137
Problems 137

5 Disturbance Observer- Based PID and Resonant Controller 139
5.1 Introduction 139
5.2 Disturbance observer-Based PI Controller 139
5.2.1 Estimation of Disturbance with Control 139
5.2.1.1 Choice of Proportional Controller K_1 140
5.2.1.2 Compensation of Steady-state Error 140

5.2.1.3 The closed-loop poles 141
5.2.1.4 Implementation procedure 142
5.2.2 Equivalence to PI controller 143
5.2.3 MATLAB Tutorial for Implementation of a PI Controller via Estimation 144
5.2.4 Examples for Estimator based PI Controllers 145
5.2.5 Food for Thought 148
5.3 Disturbance observer-Based PID Controller 149
5.3.1 Proportional Plus Derivative Control 149
5.3.2 Adding Integral Action 150
5.3.3 Equivalence to a PID Controller 151
5.3.4 MATLAB Tutorial on the Implementation of a disturbance observer-based PID Controller 153
5.3.5 Examples for Disturbance observer-based PID Controller 155
5.3.6 Food for Thought 156
5.4 Disturbance observer-Based Resonant Controller 156
5.4.1 Resonant Controller Design 156
5.4.2 Resonant Controller Implementation 158
5.4.3 Equivalence to a Resonant Controller 159
5.4.4 MATLAB Tutorial on Disturbance observer-Based Resonant Controller Implementation 160
5.4.5 Examples for Disturbance observer-Based Resonant Controllers 162
5.4.6 Food for Thought 167
5.5 Multi-frequency Resonant Controller 167
5.5.1 Adding Integral Action to the Resonant Controller 168
5.5.2 Adding More Periodic Components 170
5.5.3 Food for Thought 171
5.6 Summary 172
5.7 Further Reading 172
Problems 173

6 PID Control of Nonlinear Systems 179
6.1 Introduction 179
6.2 Linearization of the Nonlinear Model 179
6.2.1 Approximation of a Nonlinear Function 179
6.2.2 Linearization of nonlinear differential equations 181
6.2.3 Case Study: Linearization of the Coupled Tank Model 181
6.2.4 Case Study: Linearization of the Induction Motor Model 184
6.2.5 Food for Thought 186
6.3 Case Study: Ball and Plate Balancing System 187
6.3.1 Dynamics of the Ball and Plate Balancing System 187
6.3.2 Linearization of the Nonlinear Model 188
6.3.3 PID Controller Design 189
6.3.4 Implementation and Experimental Results 190
6.3.4.1 Disturbance Rejection 191
6.3.4.2 Making a Square Movement 192
6.3.4.3 Making a Circle Movement 192
6.3.4.4 Making more Complicated Movements 194

6.3.5 Food for Thought *194*
6.4 Gain Scheduled PID Control Systems *194*
6.4.1 The Weighting Parameters *194*
6.4.2 Gain Scheduled Implementation using PID Velocity Form *196*
6.4.3 Gain Scheduled Implementation using an Estimator Based PID Controller *197*
6.4.4 Food for Thought *199*
6.5 Summary *199*
6.6 Further Reading *199*
Problems *200*

7 Cascade PID Control Systems *203*
7.1 Introduction *203*
7.2 Design of a Cascade PID Control System *203*
7.2.1 Design Steps for a Cascade Control System *203*
7.2.2 Simple Design Examples *204*
7.2.3 Achieving Closed-loop Performance Invariance (Approximate) in a Cascade Structure *208*
7.2.4 Food for Thought *209*
7.3 Cascade Control System for Input Disturbance Rejection *209*
7.3.1 Frequency Characteristics for Disturbance Rejection *210*
7.3.2 Simulation Studies *211*
7.3.3 Food for Thought *213*
7.4 Cascade Control System for Actuator Nonlinearities *214*
7.4.1 Cascade Control for Actuator with a Deadzone *214*
7.4.2 Cascade Control for Actuators with Quantization Errors *218*
7.4.3 Cascade Control for Actuators with Backlash Nonlinearity *221*
7.4.4 Food for Thought *227*
7.5 Summary *230*
7.6 Further Reading *230*
Problems *231*

8 PID Controller Design for Complex Systems *233*
8.1 Introduction *233*
8.2 PI Controller Design via Gain and Phase Margins *233*
8.2.1 PI Controller Design Using Gain Margin and Phase Margin Specifications *233*
8.2.2 Design Examples *234*
8.2.3 Food for Thought *238*
8.3 PID Controller Design using Two Frequency Points *238*
8.3.1 Finding the PID Controller Parameters *238*
8.3.2 Desired Closed-loop Performance Specification using Two Frequency Points *240*
8.3.3 Design Examples *242*
8.3.4 MATLAB Tutorial on PID Controller Design Using two Frequency Points *243*
8.3.5 PID Controller Design for Beer Filtration Process *245*

8.3.6 Food for Thought *248*
8.4 PID Controller Design for Integrating Systems *249*
8.4.1 The Approximate Model *249*
8.4.2 Selection of Desired Closed-loop Performance *250*
8.4.3 Normalization of the Parameters and Empirical Rules *251*
8.4.4 Gain and Phase Margins *253*
8.4.5 Simulation Examples *253*
8.4.6 Food for Thought *256*
8.5 Summary *256*
8.6 Further Reading *257*
Problems *257*

9 Automatic Tuning of PID Controllers *259*
9.1 Introduction *259*
9.2 Relay Feedback Control *259*
9.2.1 Relay Control with Hysteresis *259*
9.2.2 Relay Control with Integrator *263*
9.2.3 Food for Thought *267*
9.3 Estimation of Frequency Response using the Fast Fourier Transform (FFT) *267*
9.3.1 FFT Estimation *268*
9.3.2 MATLAB Tutorial using the FFT for Estimation *269*
9.3.3 Monte-Carlo Simulation Studies *270*
9.3.4 Food for Thought *272*
9.4 Estimation of Frequency Response Using the frequency sampling filter (FSF) *273*
9.4.1 Frequency Sampling Filter Model *273*
9.4.2 MATLAB Tutorial on Estimation Using the FSF Model *276*
9.4.3 Monte-Carlo Simulation using the FSF Estimation *278*
9.4.4 Food for Thought *279*
9.5 Monte-Carlo Simulation Studies *279*
9.5.1 Effect of Unknown Constant Disturbance *279*
9.5.2 Effect of Unknown Low Frequency Disturbance *280*
9.5.3 Estimation of the Steady-state Value *282*
9.5.4 Food for Thought *283*
9.6 Auto-tuner Design for Stable Plant *283*
9.6.1 MATLAB Tutorial on Auto-tuner for Stable Plant *284*
9.6.2 Evaluation of the Auto-tuner for a Stable Plant *286*
9.6.2.1 PID Controller Parameters *287*
9.6.2.2 Nyquist Plots *287*
9.6.2.3 Closed-loop Simulation Results *288*
9.6.3 Comparative Studies *289*
9.6.4 Food for Thought *290*
9.7 Auto-tuner Design for a Plant with an Integrator *291*
9.7.1 Estimation of an Integrating Plus Delay Model *291*
9.7.2 Auto-tuner for Integrating Systems *292*
9.7.3 Auto-tuning of Cascade Control Systems *297*

9.7.4 Food for Thought 300
9.8 Summary 300
9.9 Further Reading 301
Problems 302

10 PID Control of Multi-rotor Unmanned Aerial Vehicles 305
10.1 Introduction 305
10.2 Multi-rotor Dynamics 305
10.2.1 Dynamic Models for Attitude Control 305
10.2.2 Actuator Dynamics for Quadrotor UAVs 307
10.2.3 Actuator Dynamics of Hexacopters 309
10.2.4 Food for Thought 311
10.3 Cascade Attitude Control of Multi-rotor UAVs 311
10.3.1 Linearized Model for the Secondary Plant 312
10.3.2 Linearized Model for the Primary Plant 313
10.3.3 Food for Thought 313
10.4 Automatic Tuning of Attitude Control Systems 313
10.4.1 Test Rigs for Auto-tuning Cascade PI Controllers of Multi-rotor UAVs 314
10.4.2 Experimental Results for Quadrotor UAV 314
10.4.3 Experimental Results for Hexacopter 320
10.4.4 Food for Thought 324
10.5 Summary 324
10.6 Further Reading 325
Problems 325

Suggestions to Food for Thought Questions 327

Bibliography 331

Index 341

Preface

PID control systems are the fundamental building blocks of classical and modern control systems. They have been used in the majority of industrial applications from chemical process control, mechanical process control, electro-mechanical process control, aerial vehicle control to electrical drive control and power converter control. Understanding these control systems and having the capability to design and implement them are paramount to a control engineer.

There are several key reasons for the continuing applications of PID controllers.

1. Simplicity in the design and analysis. There are three parameters to be chosen in the control systems. These parameters are easily understood and tuned by engineers.
2. Simplicity in the implementation. Although the PID control systems are designed and analyzed in the continuous-time, implementation is performed in discrete-time with control signal limits imposed.
3. The majority of the physical systems in the electrical, mechanical, aerospace and civil engineering fields can be decomposed in terms of components of first order or second order systems. For these first order and second order systems, the PID controller is a natural candidate because of its simplicity in design and implementation. For chemical process control, a complex system is often approximated using a first order plus delay model and a PID controller is commonly used.

This book is to present learning materials for students, instructors and engineers in various fields who wish to learn design, implementation and automatic tuning of PID control systems. The book begins with the basics in PID control systems (see Chapter 1), introducing the various PID control structures and the PID controller tuning rules. Chapter 2 presents the necessary tools for closed-loop stability and performance analysis and explains the roles of sensitivity functions in relation to disturbance rejection, reference following and measurement noise attenuation. In Chapter 3, pole-assignment controller design methods are introduced for PID controllers and resonant controllers that will track a sinusoidal reference signal and reject a sinusoidal disturbance. Feedforward compensation is introduced in this chapter. Many analytical examples and two MATLAB tutorials are given in this chapter to show the details of the designs. Chapter 4 discusses how a PID controller can be implemented in real-time with the topics of discretization, integrator windup problem, anti-windup mechanisms, and other implementation issues. A MATLAB real-time function is provided for PID controller implementation with anti-windup mechanism. Chapter 5 examines the PID controller design and resonant controller design from a different angle to the previous chapters. It introduces the integral mode and resonant modes through disturbance estimation. With the disturbance observer based approaches, the implementations of these control systems naturally incorporate anti-windup mechanisms when the control signal reaches its limit. MATLAB real-time functions are provided for the implementation of the PID controller and resonant controller with anti-windup mechanisms. In Chapter 6, PID control of nonlinear systems is discussed, which includes the topics of linearization, case study of a ball and plate balancing system with experimental validation, gain scheduled PID control systems and gain scheduled disturbance observer based control systems. Chapter 7 presents cascade PID control systems with the topics of cascade control system design, its roles in disturbance rejection and overcoming actuator's nonlinearities. Chapter 8 considers PID controller design for complex systems using frequency response data, which includes the topics of PID controller design using gain and phase margins, PID controller design using two frequency points with a specification on the desired sensitivity function, and empirical rules derived for PID control of integrator with time delay systems that have a performance specification and corresponding gain and phase margin measurement. MATLAB functions are given for the computation of the PID controller parameters using two frequency response points. Chapter 9 presents automatic tuning of PID controllers using relay feedback control. MATLAB real-time functions are created for relay feedback control and used for the Simulink simulations. The Fourier analysis and frequency sampling filter model are used, as two different methods, for the estimation of plant frequency response with the data generated from relay feedback control. The auto-tuners are created by linking the estimation to the PID controllers designed in the frequency domain as presented in Chapter 8. MATLAB functions are presented in a step-by-step manner for the estimation algorithms and for the auto-tuners. As case studies, Chapter 10 applies the PID control system design and the auto-tuner to multi-rotor unmanned aerial vehicles. This chapter is supported with experimental validations.

The book is self-contained with MATLAB/Simulink tutorials and supported with simulation and experimental results. Control system simulation and experimental implementation are emphasized in the book materials. The MATLAB real-time functions written for the use in Simulink simulations could be converted into C-codes for control system implementation with micro-controllers. For each section, there is a set of questions for us to reflect on. Some of them are easy and straightforward while others may require some thinking. At the end of each chapter, there is a set of problems for practicing the design and simulation of the control systems.

The book is suited for readers who have completed first three years engineering studies with some basic knowledge in block diagrams and Laplace transforms.

Liuping Wang
Melbourne, Australia

Acknowledgment

I wish to acknowledge the funding support from Mathworks Academic Support on the project entitled “PID Control Systems with Constraints: Design and Automatic Tuning using MATLAB/Simulink”. Particularly, I would like to thank Mr Bradley Horton from Mathworks for his help and support. I wish to thank Professors Shihua Li, Xisong Chen, Jun Yang and Dr Zhenhua Zhao in Southeast University, China, for interesting discussions on disturbance observer, during my visit to their university in 2014 and 2015. I wish to thank Dr Xi Chen and Dr Pakorn Poksawat previously at RMIT University Australia for their contributions on the automatic control of unmanned aerial vehicles.

For valuable comments towards improvement of this book, I wish to thank Professor Antonio Visioli at the University of Brescia, Italy, Dr John Tsing, who had worked in Measurex Corp. USA as a process control engineer and was an adjunct professor at San Jose State University, USA, Dr N. Leonard Segall of Sarnia, Ontario, Canada, Dr Chow Yin Lai, Dr Lasantha Meegahapola, Dr Arash Vahidnia, Dr Nuwantha Fernando, at RMIT University, Australia. I wish to thank Michelle Dunkley, Louis Vasanth Manoharan and Tessa Edmunds from Wiley and Sons Ltd for help and support during this book project, and Dipta Maitra for the book cover design.

I would like to thank my teaching team, Dr Robin Guan, Mr Long Tran Quang, Mr Junaid Saeed, Mr Luke McNabb and Mr Yifeng Sun, for their initiatives in teaching laboratory development. They have worked diligently to enhance students’ learning experience in the subject of Advanced Control Systems at RMIT University, Australia.

List of Symbols and Acronyms

Symbols

$A_{cl}(s)$	Closed-loop polynomial
$A_{cl}(s)^d$	Desired closed-loop polynomial
Δt	Sampling interval
$D_o(s)$	Laplace transform for output disturbance
$D_i(s)$	Laplace transform for input disturbance
$D_m(s)$	Laplace transform for measurement noise
$G(s)$	Transfer function model
j	Imaginary unit, $j = \sqrt{-1}$
K_c	Proportional control gain
$\lambda^l, \lambda^m, \lambda^h$	Scheduling parameters
q^{-i}	Backward shift operator, $q^{-i}[f(k)] = f(k-i)$
$S(s)$	Sensitivity function
$S^i(s)$	Input sensitivity function
$T(s)$	Complementary sensitivity function
$T_d(s)$	Desired complementary sensitivity function
τ_D	Derivative control gain
τ_f	Derivative control filter time constant
τ_I	Integral control time constant
τ_{cl}	Closed-loop time constant
$u(t)$	Control signal
u^{min}, u^{max}	Minimum and maximum limits for u
U_{ss}	Steady-state value for a control signal
$u_{act}(t)$	Actual control signal ($u_{act}(t) = U_{ss} + u(t)$)
u^0	Operating condition of u^0
w_n	Bandwidth or natural frequency in PID controller design (rad/s)
x^0	Operating condition of x
ξ	Damping coefficient in PID controller design
$y(t)$	Output signal
Y_{ss}	Steady-state value of the output signal
$y_{act}(t)$	Actual output signal ($y_{act}(t) = Y_{ss} + y(t)$)

Acronyms

FFT	Fast Fourier transform
FSF	Frequency sampling filter
IPD	Alternative PID controller structure with proportional control and derivative control on output only
PID	Proportional, integral and derivative
PWM	Pulse Width Modulation

About the Companion Website

This book is accompanied by a companion website:

www.wiley.com/go/Wang/PIDcontrolsystem

The website includes:

- Lecture slides
- MATLAB/Simulink programs

Scan this QR code to visit the companion website.

1

Basics of PID Control

1.1 Introduction

This chapter will introduce the basic ideas of PID control systems. It starts with an introduction to the roles of proportional control, integral control, and derivative control, followed by an introduction to the various tuning rules developed over the past several decades. These tuning rules are mainly developed for first order plus delay systems and are simple to use. However, they do not, in general, guarantee satisfactory control system performance. Simulation examples will be used to illustrate closed-loop performance.

This chapter is suitable for those who want to understand the very basics of PID control systems. By utilization of the tuning rules, it is possible to have an application of a PID control system without further exploration.

1.2 PID Controller Structure

There are four types of controllers that belong to the family of PID controllers: the proportional (P) controller, the proportional plus integral (PI) controller, the proportional plus derivative (PD) controller and the proportional plus integral plus derivative (PID) controller. To understand the roles of the controllers, in this section we will discuss each of the structures and the PID controller parameters. From the discussions, we will establish some basic knowledge about how to use these controllers in various applications.

1.2.1 Proportional Controller

The simplest controller is the proportional controller. With this term proportional, the feedback control signal $u(t)$ is computed in proportion to the feedback error $e(t)$ with the formula,

$$u(t) = K_c e(t) \tag{1.1}$$

where K_c is the proportional gain and the feedback error is the difference between the reference signal $r(t)$ and the output signal $y(t)$ ($e(t) = r(t) - y(t)$). The block diagram for the closed-loop feedback control configuration is shown in Figure 1.1 where $R(s)$, $E(s)$, $U(s)$, and $Y(s)$ are the Laplace transforms of the reference signal, feedback error, control

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

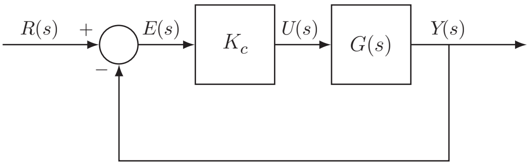


Figure 1.1 Proportional feedback control system.

signal, and output signal, respectively. $G(s)$ represents the Laplace transfer function of the plant.

Because of its simplicity, the proportional controller is often used in the cases when little information about the system is available and the required control performance in steady-state operation is not demanding. As the controller only involves one parameter to be determined, it is possible to choose K_c without detailed information about the plant.

One of the limitations of a simple proportional controller is that the steady-state error of the closed-loop control system will not be completely eliminated. We illustrate this point with the following example.

Example 1.1 *Suppose that the plant is a first order system with the following transfer function,*

$$G(s) = \frac{0.3}{s+1} \tag{1.2}$$

with proportional controller K_c ($K_c > 0$). Suppose that the reference signal is a step signal with amplitude 1 and its Laplace transform is $R(s) = \frac{1}{s}$. Find the steady-state value of the output with respect to the reference signal.

Solution. *From Figure 1.1, the closed-loop control system from the reference signal to the plant output has the transfer function,*

$$\frac{Y(s)}{R(s)} = \frac{K_cG(s)}{1+K_cG(s)} = \frac{0.3K_c}{s+1+0.3K_c}. \tag{1.3}$$

With any positive K_c, the closed-loop system is stable where its pole is determined by the solution of the polynomial equation[1],

$$s+1+0.3K_c = 0 \tag{1.4}$$

which is $-1-0.3K_c$.

The Laplace transform of the output, $Y(s)$, is

$$Y(s) = \frac{0.3K_c}{s+1+0.3K_c}R(s) = \frac{0.3K_c}{s(s+1+0.3K_c)}. \tag{1.5}$$

where $R(s) = \frac{1}{s}$. Applying final value theorem to the stable closed-loop system, we calculate

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} s \times \frac{0.3K_c}{s(s+1+0.3K_c)} = \frac{0.3K_c}{1+0.3K_c}. \tag{1.6}$$

[1] This polynomial equation is called a closed-loop characteristic equation.

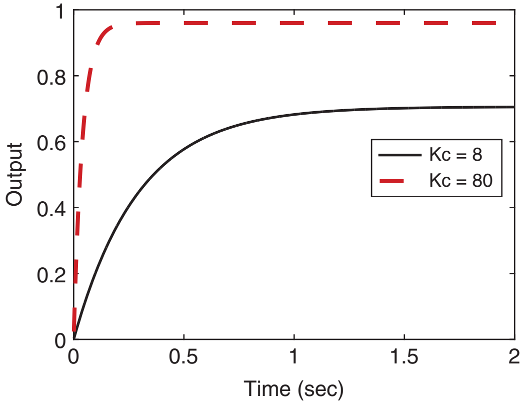


Figure 1.2 Closed-loop step response of a proportional control system (Example 1.1).

For any value of $0 < K_c < \infty$, $\lim_{t\to\infty} y(t) \neq 1$, *i.e. not equal to the desired value at the steady state response. Figure 1.2 shows the closed-loop step response with the proportional controller* $K_c = 8$ *and* $K_c = 80$, *respectively. It is seen that with the increased proportional gain* K_c, *the closed-loop response speed increases and the steady-state value becomes closer to the desired value 1.*

1.2.2 Proportional Plus Derivative Controller

In many applications, a proportional controller K_c is not sufficient to achieve a particular control objective such as stabilization or producing adequate damping for the closed-loop system. For instance, for a double integrator system with the transfer function:

$$G(s) = \frac{K}{s^2} \tag{1.7}$$

the closed-loop control system with a proportional controller K_c has a transfer function

$$\frac{Y(s)}{R(s)} = \frac{K_c K}{s^2 + K_c K}$$

which has a pair of closed-loop poles determined by the solutions of the polynomial equation:

$$s^2 + K_c K = 0.$$

These poles are at $\pm j\sqrt{K_c K}$. Thus, no matter what choice we make for K_c, the system still behaves in a sustained oscillatory manner because the pair of closed-loop poles are on the imaginary axis of the complex plane.

Now, assuming that we will additionally take the derivative of the feedback error signal into the control signal calculation, this leads to

$$u(t) = K_c e(t) + K_c \tau_D \dot{e}(t) \tag{1.8}$$

where τ_D is the derivative control gain.

The Laplace transfer function of (1.8) is calculated as

$$\frac{U(s)}{E(s)} = K_c + K_c \tau_D s.$$

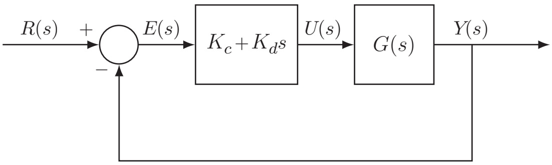


Figure 1.3 Proportional plus derivative feedback control system ($K_d = K_c\tau_D$).

This is what we called a proportional plus derivative (PD) controller.

The closed-loop feedback control configuration for a PD controller is shown in Figure 1.3. For the double integrator system (1.7), with the derivative term included in the controller, then the closed-loop transfer function becomes

$$\frac{Y(s)}{R(s)} = \frac{(K_c + K_c\tau_D s)G(s)}{1 + (K_c + K_c\tau_D s)G(s)} \tag{1.9}$$

$$= \frac{K_c K(1 + \tau_D s)}{s^2 + K_c K\tau_D s + K_c K}. \tag{1.10}$$

The closed-loop poles are determined by the solutions of the characteristic polynomial equation as

$$s^2 + K_c K\tau_D s + K_c K = 0$$

which are

$$s_{1,2} = \frac{-K_c K\tau_D \pm \sqrt{(K_c K\tau_D)^2 - 4K_c K}}{2}.$$

Clearly, we can choose the values of K_c and τ_D to achieve the desired closed-loop performance.

It is worthwhile emphasizing that almost without exception, the derivative term is different from the original form $K_c\tau_D s$ in the implementation. This is because the derivative term $K_c\tau_D(\frac{dr(t)}{dt} - \frac{dy(t)}{dt})$ is not practically implementable and the differentiation of the output signal $y(t)$ leads to amplification of the measurement noise. Thus, there are a few modifications with regard to the derivative terms. Firstly, in order to avoid the problem caused by a step reference trajectory change (the so-called derivative kick problem (Hägglund (2012))), the derivative action is only implemented on the output signal $y(t)$. Secondly, in order to avoid amplification of the measurement noise, a derivative filter is in a companionship with the derivative term $K_c\tau_D s$.

A commonly used derivative filter is a first order filter and has its time constant linked as a percentage to the actual derivative gain τ_D in the form:

$$F_D(s) = \frac{1}{\beta\tau_D s + 1} \tag{1.11}$$

where β is typically chosen to be 0.1 (10%). β is chosen to be larger if the measurement noise is severe.

With the derivative filter $F_D(s)$, a typical PD controller output is calculated as

$$u(t) = K_c(r(t) - y(t)) - K_c\tau_D\frac{dy_f(t)}{dt} \tag{1.12}$$

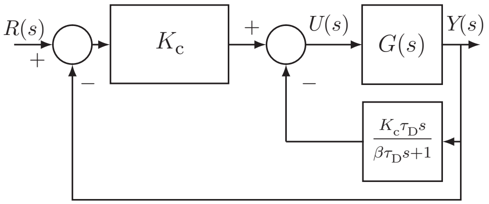


Figure 1.4 PD controller structure in implementation.

where $y_f(t)$ is the filtered output response using (1.11). In a Laplace transform, the control signal is expressed as

$$U(s) = K_c(R(s) - Y(s)) - \frac{K_c\tau_D s}{\beta\tau_D s + 1}Y(s). \tag{1.13}$$

Figure 1.4 shows the block diagram used for implementation of a PD controller with a filter.

If the derivative filter was not considered in the design, there is a certain degree of performance uncertainty due to the introduction of the filter. This may not be ideal for many applications. Designing a PD controller with the filter included will be discussed in Section 3.4.1.

1.2.3 Proportional Plus Integral Controller

A proportional plus integral (PI) controller is the most widely used controller among PID controllers. With the integral action, the steady-state error that had existed with the proportional controller alone (see Example 1.1) is completely eliminated. The output of the controller $u(t)$ is the sum of two terms, one from the proportional function and the other from the integral action, having the form,

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(\tau)\mathrm{d}\tau \tag{1.14}$$

where $e(t) = r(t) - y(t)$ is the error signal between the reference signal $r(t)$ and the output $y(t)$, K_c is the proportional gain, and τ_I is the integral time constant. The parameter τ_I is always positive, and its value is inversely proportional to the effect of the integral action taken by the PI controller. A smaller τ_I will result in a stronger effect of the integral action.

The Laplace transform of the controller output is

$$U(s) = K_c E(s) + \frac{K_c}{\tau_I s}E(s) \tag{1.15}$$

with $E(s)$ being the Laplace transform of the error signal $e(t)$. With this, the Laplace transfer function of the PI controller is expressed as

$$C(s) = \frac{U(s)}{E(s)} = \frac{K_c(\tau_I s + 1)}{\tau_I s} \tag{1.16}$$

Figure 1.5 shows a block diagram of the PI control system.

The example below is used to illustrate closed-loop control with a PI controller. For comparison purpose, we use the same plant as that used in Example 1.1.

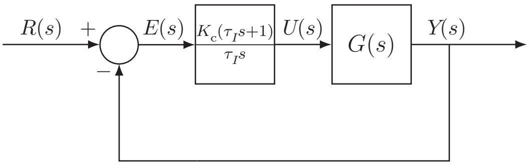


Figure 1.5 PI control system.

Example 1.2 *Assume that the plant is a first order system with the transfer function:*

$$G(s) = \frac{0.3}{s+1} \tag{1.17}$$

the PI controller has the proportional gain $K_c = 8$, and the integral time constant $\tau_I = 3$ and 0.5 respectively. Examine the locations of the closed-loop poles. With the reference signal $r(t)$ as a unit step signal, find the steady-state value of the closed-loop output $y(t)$.

Solution. *We calculate the closed-loop transfer function between the reference and output signals:*

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}. \tag{1.18}$$

With $C(s)$ given in (1.16) and $G(s)$ in (1.17), we have

$$\frac{Y(s)}{R(s)} = \frac{0.3K_c\tau_I s + 0.3K_c}{\tau_I s^2 + \tau_I(1 + 0.3K_c)s + 0.3K_c}. \tag{1.19}$$

The closed-loop poles of this system are determined by the solutions of the closed-loop characteristic equation,

$$\tau_I s^2 + \tau_I(1 + 0.3K_c)s + 0.3K_c = 0 \tag{1.20}$$

which are

$$s_{1,2} = -\frac{1 + 0.3K_c}{2} \pm \frac{1}{2}\sqrt{(1 + 0.3K_c)^2 - \frac{1.2K_c}{\tau_I}}. \tag{1.21}$$

If the quantity

$$(1 + 0.3K_c)^2 - \frac{1.2K_c}{\tau_I} = 0$$

then there are two identical real poles located at

$$s_{1,2} = -\frac{1 + 0.3K_c}{2}.$$

If the quantity

$$(1 + 0.3K_c)^2 - \frac{1.2K_c}{\tau_I} > 0$$

then there are two real poles located at

$$s_{1,2} = -\frac{1 + 0.3K_c}{2} \pm \frac{1}{2}\sqrt{(1 + 0.3K_c)^2 - \frac{1.2K_c}{\tau_I}}.$$

If the quantity

$$(1+0.3K_\mathrm{c})^2 - \frac{1.2K_\mathrm{c}}{\tau_\mathrm{I}} < 0$$

then there are two complex poles located at

$$s_{1,2} = -\frac{1+0.3K_\mathrm{c}}{2} \pm j\frac{1}{2}\sqrt{\frac{1.2K_\mathrm{c}}{\tau_\mathrm{I}} - (1+0.3K_\mathrm{c})^2}.$$

The closed-loop system is stable as long as K_c is positive and $0 < \tau_\mathrm{I} < \infty$.

Applying the final value theorem, we calculate

$$\begin{aligned}\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) &= \lim_{s\to 0} s \times \frac{0.3K_\mathrm{c}\tau_\mathrm{I}s + 0.3K_\mathrm{c}}{\tau_\mathrm{I}s^2 + \tau_\mathrm{I}(1+0.3K_\mathrm{c})s + 0.3K_\mathrm{c}}\frac{1}{s}\\ &= \frac{0.3K_\mathrm{c}}{0.3K_\mathrm{c}} = 1,\end{aligned} \tag{1.22}$$

where the steady-state value is equal to the reference signal, and it is independent of the value of integral time constant τ_I. Figure 1.6 shows for the same $K_\mathrm{c} = 8$ as in Example 1.1 the closed-loop step response with $\tau_\mathrm{I} = 3$ and $\tau_\mathrm{I} = 0.5$, respectively. It is seen that as τ_I reduces, the closed-loop response speed becomes faster. Nevertheless, the steady-state responses with both τ_I values are equal to one.

It is often the case that the output of a PI control system exhibits overshoot to a step reference signal. The percentage of overshoot increases as higher control performance demanded. This may cause a conflict in the PI control performance specifications: on the one hand a fast control system response is desired, and yet on the other hand, the overshoot is not desirable when step reference changes are performed. The overshooting problem in reference change could be reduced by a small change in the configuration of the PI controller. This small change is to put the proportional control on the output signal $y(t)$, instead of the feedback error $e(t) = r(t) - y(t)$. More specifically, the control signal $u(t)$ is calculated using the following relation,

$$u(t) = -K_\mathrm{c}y(t) + \frac{K_\mathrm{c}}{\tau_\mathrm{I}}\int_0^t (r(\tau) - y(\tau))\mathrm{d}\tau. \tag{1.23}$$

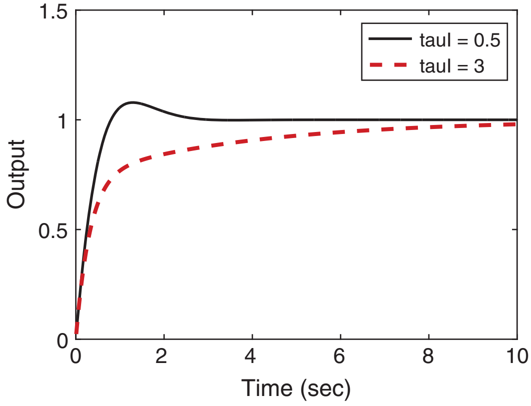


Figure 1.6 Closed-loop step response of a PI control system (Example 1.2).

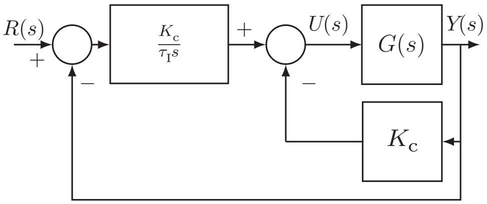


Figure 1.7 IP controller structure.

Applying a Laplace transform to this equation leads to the Laplace transform of the controller output in relation to the reference and the output as

$$U(s) = -K_c Y(s) + \frac{K_c}{\tau_I s}(R(s) - Y(s)). \tag{1.24}$$

Figure 1.7 shows a block diagram of this PI closed- loop control configuration. This type of implementation is called an IP controller in the literature, which is an alternative PI controller configuration. In Section 2.4, this PI controller structure is examined in the context of a reference filter within the framework of a two degrees of freedom control system. To demonstrate how this simple modification in the PI controller configuration can reduce the overshoot effect, we examine the following example.

Example 1.3 *Assume that the plant is described by the transfer function:*

$$G(s) = \frac{1}{s(s+1)^3} \tag{1.25}$$

and the PI controller has the parameters: $K_c = 0.56$, $\tau_I = 8$[2]*. Find the closed-loop transfer function between the reference signal $R(s)$ and the output signal $Y(s)$ for the original PI controller structure (see Figure 1.5) and the IP controller structure (see Figure 1.7), and compare their closed-loop step responses.*

Solution. *With the PI controller in the original structure, the closed-loop transfer function between the reference signal $R(s)$ and the output signal $Y(s)$ is calculated using,*

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}. \tag{1.26}$$

By substituting the plant transfer function (1.25) and the PI controller structure (1.16), the closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{K_c(\tau_I s + 1)}{\tau_I s^2 (s+1)^3 + K_c(\tau_I s + 1)}. \tag{1.27}$$

With the PI controller in the IP structure, the Laplace transform of the control signal $U(s)$ is defined by (1.24). By substituting this control signal into the Laplace transform of the output $Y(s)$ via the following equation,

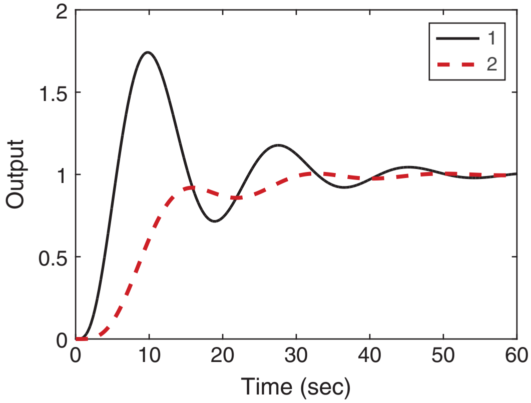
$$Y(s) = \frac{1}{s(s+1)^3} U(s) \tag{1.28}$$

re-grouping and simplification lead to the closed-loop transfer function:

$$\frac{Y(s)}{R(s)} = \frac{K_c}{\tau_I s^2 (s+1)^3 + K_c(\tau_I s + 1)}. \tag{1.29}$$

[2] This PI controller was designed using frequency response data in Wang and Cluett (2000).

Figure 1.8 Closed-loop step response of PI control system (Example 1.3). Key: line (1) response from the original structure; line (2) response from the IP structure.



By comparing the closed-loop transfer function (1.27) from the original PI controller structure with the one (1.29) from the alternative structure, we notice that both transfer functions have the same denominator, however, the one from the original structure has a zero at $-\frac{1}{\tau_I}$. *Because of this zero, the original closed-loop step response may have an overshoot.*

Indeed, the closed-loop step responses for both structures are simulated and compared in Figure 1.8, which shows that the original PI closed-loop control system has a large overshoot; in contrast, the IP closed-loop control system has reduced this overshoot. The penalty for reducing the overshoot is the slower reference response speed.

The closed-loop transfer function obtained with IP controller structure can also be interpreted as a two degrees of freedom control system with a reference filter $H(s) = \frac{1}{\tau_I s+1}$. *This topic will be further discussed in Section 2.4.2.*

Another form of PI controller, perhaps more convenient for model-based controller design as in Chapter 3, is described by:

$$C(s) = \frac{c_1 s + c_0}{s} \tag{1.30}$$

This form of PI controller is identical to the original PI controller structure when the parameters of K_c and τ_I are selected as

$$K_c = c_1; \quad \tau_I = \frac{c_1}{c_0} \tag{1.31}$$

1.2.4 PID Controllers

A PID controller consists of three terms: the proportional (P) term, the integral (I) term, and the derivative (D) term. In an ideal form, the output $u(t)$ of a PID controller is the sum of the three terms,

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(\tau)\mathrm{d}\tau + K_c\tau_D\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{1.32}$$

where $e(t) = r(t) - y(t)$ is the feedback error signal between the reference signal $r(t)$ and the output $y(t)$, and τ_D is the derivative control gain. The Laplace transfer function of the PID controller is

$$\frac{U(s)}{E(s)} = K_c(1 + \frac{1}{\tau_I s} + \tau_D s). \tag{1.33}$$

If the design is sound, the sign of τ_D is positive. If the sign of τ_D is negative, the derivative control term is to be neglected and instead a PI controller should be chosen.

Analogously to the proportional plus derivative controller described in Section 1.2.2, for most of the applications the derivative control is implemented on the output only with a derivative filter. For this reason, the control signal $U(s)$ is expressed in the following form:

$$U(s) = K_c(1 + \frac{1}{\tau_\mathrm{I} s})(R(s) - Y(s)) - \frac{K_c\tau_\mathrm{D} s}{\beta\tau_D s + 1} Y(s). \tag{1.34}$$

Figure 1.9 shows the block diagram of the PID controller structure.

To reduce the overshoot in output response to a step reference change, the proportional term in the PID controller may also be implemented on the plant output. In this case, the control signal is calculated using

$$u(t) = -K_c y(t) + \frac{K_c}{\tau_\mathrm{I}} \int_0^t (r(\tau) - y(\tau))\mathrm{d}\tau - K_c\tau_D \frac{\mathrm{d}y_f(t)}{\mathrm{d}t}. \tag{1.35}$$

Accordingly, the Laplace transform of the control signal is expressed as

$$\begin{aligned} U(s) &= -K_c Y(s) + \frac{K_c}{\tau_\mathrm{I} s}(R(s) - Y(s)) - \frac{K_c\tau_\mathrm{D} s}{\beta\tau_D s + 1} Y(s) \\ &= \frac{K_c}{\tau_\mathrm{I} s}(R(s) - Y(s)) - \frac{K_c(\tau_\mathrm{D}(\beta + 1)s + 1)}{\beta\tau_D s + 1} Y(s). \end{aligned} \tag{1.36}$$

Figure 1.10 shows a block diagram of the alternative PID controller structure (called an IPD controller).

The example below is used to illustrate the effect of the derivative term in the closed-loop control. The starting point is the PI controller designed in Example 1.3, based on which a derivative term is introduced.

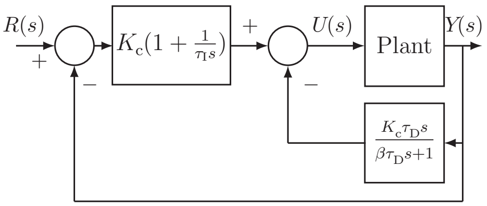

$R(s)$
+
−
$K_c(1+\frac{1}{\tau_\mathrm{I} s})$
+
$U(s)$
−
Plant
$Y(s)$
$\frac{K_c\tau_\mathrm{D} s}{\beta\tau_\mathrm{D} s+1}$

Figure 1.9 PID controller structure.

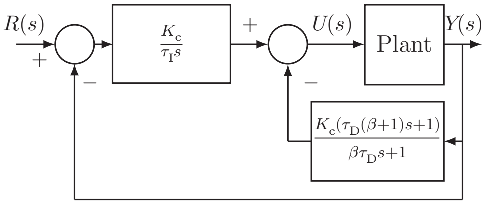

$R(s)$
+
−
$\frac{K_c}{\tau_\mathrm{I} s}$
+
$U(s)$
−
Plant
$Y(s)$
$\frac{K_c(\tau_\mathrm{D}(\beta+1)s+1)}{\beta\tau_\mathrm{D} s+1}$

Figure 1.10 IPD controller structure.

Example 1.4 *Suppose that the plant is described by the transfer function:*

$$G(s) = \frac{1}{s(s+1)^3} \tag{1.37}$$

and the PI part of the controller has the parameters: $K_c = 0.56$, $\tau_I = 8$. Choosing $\tau_D = 0.1$ and 1, find the closed-loop transfer function for the PID control system shown in Figure 1.9 and simulate its closed-loop performance. Also, find the closed-loop transfer function of the IPD control system shown in Figure 1.10 and simulate its closed-loop step response.

Solution. *The control signal from Figure 1.9 has the Laplace transform given by Equation (1.34). Substituting $U(s)$ into the following equation,*

$$Y(s) = \frac{1}{s(s+1)^3}U(s) \tag{1.38}$$

re-grouping and re-arranging lead to the closed-loop transfer function:

$$\frac{Y(s)}{R(s)} = \frac{K_c(\tau_I s+1)(\beta\tau_D s+1)}{\tau_I s^2(s+1)^3(\beta\tau_D s+1) + K_c(\tau_I s+1)(\beta\tau_D s+1) + K_c\tau_I\tau_D s^2}. \tag{1.39}$$

There are two zeros in the closed-loop transfer function: $-\frac{1}{\tau_I}$ caused by the integral control, and $-\frac{1}{\beta\tau_D}$ caused by the derivative control. Figure 1.11(a) shows the closed-loop step responses for $\tau_D = 0.1$ and $\tau_D = 1$ respectively. With the increase in τ_D, the oscillation in the closed-loop response is reduced. However, there is a large overshoot in the output response.

Using the IPD structure as shown in Figure 1.10, we calculate the closed-loop transfer function by using the Laplace transform of the controller output given in Equation (1.36). Substituting this control signal into the plant output (see Equation (1.38)), the closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{K_c(\beta\tau_D s+1)}{\tau_I s^2(s+1)^3(\beta\tau_D s+1) + K_c(\tau_I s+1)(\beta\tau_D s+1) + K_c\tau_I\tau_D s^2}. \tag{1.40}$$

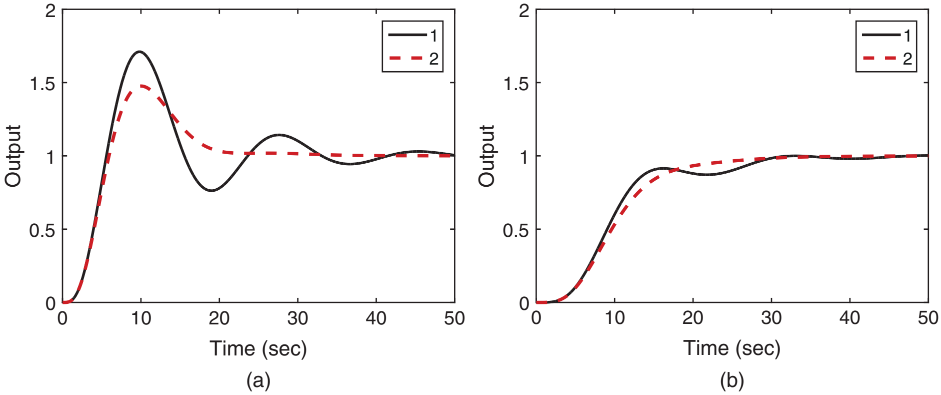


Figure 1.11 Step responses of PID control system (Example 1.4). (a) Responses for PID structure. (b) Responses for the IPD structure. Key: line (1) $\tau_D = 0.1$, line (2) $\tau_D = 1$ ($K_c = 0.56$, $\tau_I = 8$).

With this implementation, the denominator of the closed-loop transfer function is the same; however, there is only one zero at $-\frac{1}{\beta\tau_{\mathrm{D}}}$ *caused by the derivative control. Figure 1.11(b) shows the closed-loop step responses with an IPD controller. In comparison with the responses from the previous case, it is seen that the overshoot in the closed-loop responses has been eliminated, however their response speed becomes slower.*

1.2.5 The Commercial PID Controller Structure

In the PID controller design, the following structure is commonly used for determining the parameters K_{c}, τ_{I} and τ_{D}, where the controller transfer function $C(s)$ is expressed as

$$C(s) = K_{\mathrm{c}}(1 + \frac{1}{\tau_I s} + \tau_D s). \tag{1.41}$$

However, as demonstrated in this section, there are several variations in PID controller structure available for the realization of the control system, and different realization leads to different control system performance with the same set of PID controller parameters.

In order to be more flexible to the users, the commercial PID controllers (see Alfaro and Vilanova (2016)) from manufacturers such as ABB, Siemens, and National Instruments take the following general form with the Laplace transform of the control signal:

$$U(s) = K_{\mathrm{c}}\left(\gamma_1 R(s) - Y(s) + \frac{1}{\tau_{\mathrm{I}} s}(R(s) - Y(s)) + \frac{\tau_{\mathrm{D}} s}{\beta\tau_{\mathrm{D}} s + 1}(\gamma_2 R(s) - Y(s))\right). \tag{1.42}$$

where the coefficients $0 \leq \gamma_1 \leq 1$ and $0 \leq \gamma_2 \leq 1$ are for the weighting on the reference signal, and as before, the coefficient $0 \leq \beta \leq 1$ determines the appropriate derivative filtering action. There are several special combinations of the parameters γ_1, γ_2, and β that are commonly encountered.

1. When $\gamma_1 = 1, \gamma_2 = 0$, and $0 \leq \beta \leq 1$, the PID controller becomes identical to the case shown in Figure 1.9, where the derivative control with filter is implemented on the output only.
2. When $\gamma_1 = 0$ and $\gamma_2 = 0$, the PID controller becomes the IPD controller shown in Figure 1.10, where both the proportional control and derivative control are implemented on the output only.
3. When $\gamma_1 = 1$, $\gamma_2 = 1$, and $0 \leq \beta \leq 1$, the implementation of the PID controller puts proportional control, integral control, and derivative control with filter on the feedback error $R(s) - Y(s)$.
4. When $\gamma_1 = 1, \gamma_2 = 1$, and $\beta = 0$, the PID controller becomes the case where no derivative filter is used in the implementation. This will severely amplify the measurement noise.

It is worthwhile emphasizing that the parameters γ_1 and γ_2 only affect the closed-loop response to the reference signal $R(s)$ and they play no role in the closed-loop stability. We have examined the cases where γ_1 and γ_2 are either 0 or 1. However, we can extend the results to the situations where the parameters are between 0 and 1 and expect a compromised result. Upon understanding their roles, we can choose the appropriate coefficients according to the actual applications.

1.2.6 Food for Thought

1. The PID controllers are expressed in terms of the parameters K_c, τ_I and τ_D. What are the possible signs of K_c, τ_I and τ_D?
2. When you increase the magnitude of K_c, do you expect the action of proportional control to decrease or increase? When you increase τ_I, do you expect the action of integral control to decrease or increase? when you increase τ_D, do you expect the action of derivative control to decrease or increase?
3. What are the roles of integrator in a PID controller?
4. Can you implement the integrating control on output only? If not, explain the reason.
5. In many applications, we will put the proportional control on the feedback error, which is the original PI controller. Can you reduce the overshoot by using a ramp reference signal in the early part of the response?

1.3 Classical Tuning Rules for PID Controllers

This section will discuss the classical tuning rules that have existed for the past several decades and have withstood the test of time. Although all tuning rules are rule-based, there is still certain knowledge assumed for the system to be controlled.

1.3.1 Ziegler–Nichols Oscillation Based Tuning Rules

Ziegler–Nichols oscillation based tuning rules are to use closed-loop controlled testing to obtain the critical information needed for determining the PID controller parameters.

In the closed-loop control testing, the controller is set to proportional mode without integrator and derivative action. The sign of K_c must be the same as the steady-state gain of the plant for the reason of introducing negative feedback in the control system. With the proportional closed-loop control, the feedback gain K_c is set to be a very small value in magnitude to begin the experiment. The value of K_c is gradually increased until the control signal $u(t)$ exhibits sustained oscillation (see Figure 1.12). There are two parameters obtained from this test: the value of K_c that has caused the oscillation and the period of the oscillation. We denote this particular K_c as K_o and the period as P_o. With these two parameters, the PID controller parameters are presented for the Ziegler–Nichols tuning rules in Table 1.1.

Figure 1.12 Sustained closed-loop oscillation (control signal).

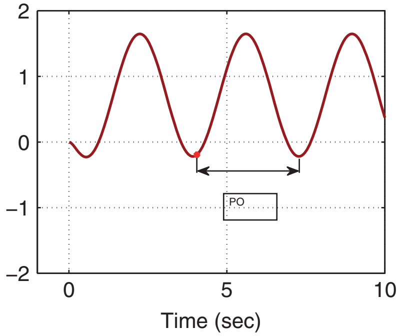


Table 1.1 Ziegler–Nichols tuning rule using oscillation testing data.

	K_c	τ_I	τ_D
P	$0.5K_o$		
PI	$0.45K_o$	$\frac{P_o}{1.2}$	
PID	$0.60K_o$	$\frac{P_o}{2}$	$\frac{P_o}{8}$

A proportional control will not cause sustained oscillation for first order plant and second order plant with a stable zero. Thus, the tuning rule is not applicable to these two classes of stable plants. The following example is used to illustrate an application of the tuning rule.

Example 1.5 *Assume that a continuous time plant has the Laplace transfer function:*

$$G(s) = \frac{s-2}{(s+1)(s+2)(s+3)}. \tag{1.43}$$

Find the PI and PID controller parameters using Ziegler–Nichols tuning rule and simulate the closed-loop control systems.

Solution. *We build a Simulink simulation program for proportional control as illustrated in Figure 1.1. Since this system has a negative steady-state gain of* $-\frac{1}{3}$*, so the feedback control gain should be negative.*[3] *Beginning the tuning process by setting* $K_c = -1$ *and decreasing gradually to* $K_c = -7.5$*, the closed-loop control system exhibits sustained oscillation as shown in Figure 1.12. From this figure, the period of oscillation reads as* 3.35*. Based on Table 1.1, the proportional gain for the PI controller is* $K_c = 0.45 \times (-7.5) = -3.38$ *and the integral time constant* $\tau_I = \frac{3.35}{1.2} = 2.79$*. The proportional gain for the PID controller is* $K_c = 0.6 \times (-7.5) = -4.5$, $\tau_I = \frac{3.35}{2} = 1.68$, *and* $\tau_D = \frac{3.35}{8} = 4.2$*. The PI and PID control systems are simulated where the reference signal is a unit step signal. Figure 1.13 compares the closed-loop output responses based on the PI and PID controller structures. Here the derivative control is implemented on the output only with a filter time constant* $0.1\tau_D$*. It is seen that with the derivative term, the closed-loop oscillation existing in the PI controller is reduced.*

In general, the Ziegler and Nichols tuning rules using the oscillation method lead to quite aggressive responses with oscillations in closed-loop responses that are undesirable for many control applications. In Tyreus and Luyben (1992), a smaller K_c and a larger τ_I are recommended to reduce the oscillations, which have the following values:

$$K_c = 0.313K_o; \quad \tau_I = 2.2P_o.$$

It is important to point out that generating sustained oscillation by increasing the controller gain is not a safe operation because a small error in the tuning process could

[3] We evaluate the steady-state gain of the transfer function by letting $s = 0$ and calculating the value of the transfer function.

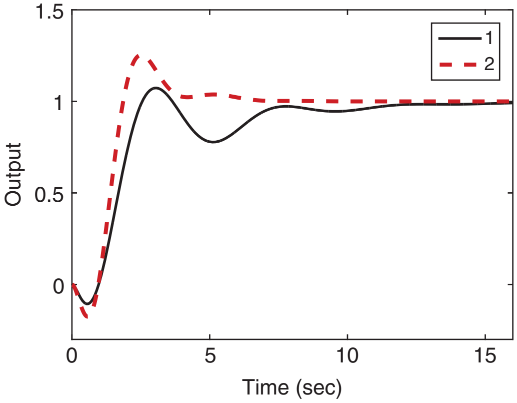


Figure 1.13 Comparison of closed-loop output response using Ziegler–Nichols rules (Example 1.5). Key: line (1) PI control; line (2) PID control.

cause the closed-loop system to become unstable. This unsafe procedure is replaced by using relay feedback control in Chapter 9, which also produces a sustained closed-loop oscillation.

1.3.2 Tuning Rules based on the First Order Plus Delay Model

The majority of tuning rules existing in the literature are based on a first order plus delay model, which has the following transfer function:

$$G(s) = \frac{K_{ss}e^{-ds}}{\tau_M s + 1} \tag{1.44}$$

where K_{ss} is the steady-state gain of the system, d is the time delay, and τ_M is the time constant. This is mainly because the primary applications of tuning rules are for process control where typically the process is stable with time delay. There are many methods available for obtaining a first order plus delay model. Among them is an incredibly simple procedure that is called fitting a reaction curve. This reaction curve is the so-called step response test.

The reaction curve is obtained by performing a plant step response test in open-loop operation, therefore we assume that the plant is stable, although it could be applied to integrating plant with caution. When performing this test, the plant input signal $u(t)$ takes a step change from an initial constant value U_0 to a normal operation value, U_s; the measurement of the plant output signal $y(t)$ in response to the step input change gives us the plant step response test data or the reaction curve. The response test completes when the value of the output signal reaches a constant or the signal fluctuates around a constant value due to noise and disturbances. Figure 1.14 illustrates a typical set of step response testing data, with the step change occurring at time $t = 0$. Figure 1.14(a) shows that the input takes a step change from the initial value U_0 to the final value U_s and Figure 1.14(b) shows the actual output response from the steady-state output position Y_0 to the steady-state output position Y_s.

The plant information obtained from the test includes the steady-state gain, defined as

$$K_{ss} = \frac{Y_s - Y_0}{U_s - U_0}. \tag{1.45}$$

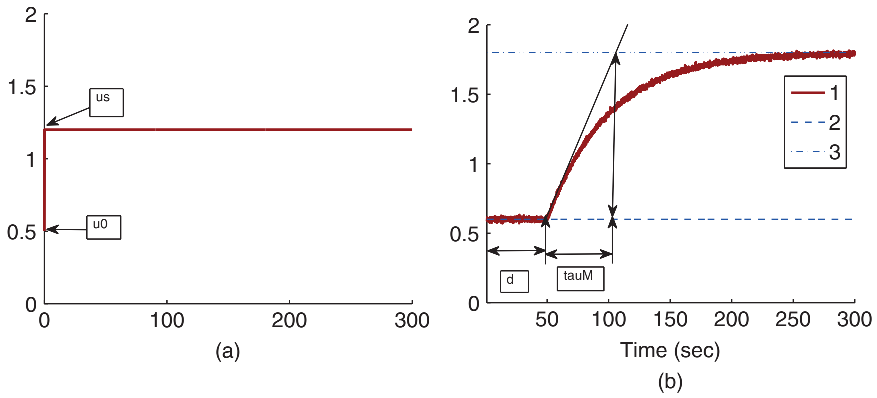


Figure 1.14 Step response data. (a) Input signal. (b) Output signal. Key: line (1) the output response; line (2) steady-state output position before the response (Y_0); line (3) steady-state output position in completion of the response (Y_s).

The time delay d is shown in the figure, which is the delayed time when the output responds to the change in the input signal. The parameter time delay d reflects the situation that the output response remains unchanged despite the step input signal being injected. Thus, it is estimated using the time difference between when the step reference change occurred ($t = 0$ for this figure) and when the output response moved away from its steady-state value (see the time interval in Figure 1.14(b) marked with the first set of arrows). A line with maximum slope is drawn on Figure 1.14(b), which is intersected with the line corresponding to the indicator of Y_s. The intersecting point shown in Figure 1.14(b) determines the value of τ_M that is a measurement of the dynamic response time.

Alternatively, because the step response of a first order system ($G(s) = \frac{K_{ss}}{\tau_M s+1}$) to a unit step input signal can be expressed as

$$g(t) = K_{ss}(1 - e^{\frac{-t}{\tau_M}})$$

and when the variable time $t = \tau_M$,

$$g(\tau_M) = K_{ss}(1 - e^{-1}) = 0.632K_{ss}$$

thus, we can determine the time constant τ_M using 63.2% of the rising time in the step response. This estimation of time constant gives a different value from the case when using the maximum slope approach. For the majority of the applications, this will result in a smaller time constant τ_M, and from the empirical tuning rules stated in the later part of the section, a smaller proportional gain K_c will follow. One can evaluate this approach as an exercise using Problem 1.2.

Essentially, the step response test gives the parameters in the first order plus delay description of the process as in (1.44).

There is a second set of Ziegler–Nichols tuning rules that is based on the plant step response test data. This is also called the Ziegler–Nichols tuning rules using reaction curve. With these parameters, Ziegler–Nichols tuning rules using a reaction curve are given in Table 1.2. By the nature of this testing procedure (open-loop testing), the tuning rules should apply to stable systems.

Table 1.2 Ziegler-Nichols tuning rules with a reaction curve.

	K_c	τ_I	τ_D
P	$\frac{\tau_M}{K_{ss}d}$		
PI	$0.9\frac{\tau_M}{K_{ss}d}$	$3d$	
PID	$1.2\frac{\tau_M}{K_{ss}d}$	$2d$	$0.5d$

Table 1.3 Cohen–Coon tuning rules with a reaction curve.

	K_c	τ_I	τ_D
P	$\frac{\tau_M}{K_{ss}d}\left(1+\frac{d}{3\tau_M}\right)$		
PI	$\frac{\tau_M}{K_{ss}d}\left(0.9+\frac{d}{12\tau_M}\right)$	$\frac{d(30\tau_M+3d)}{9\tau_M+20d}$	
PID	$\frac{\tau_M}{K_{ss}d}\left(\frac{4}{3}+\frac{d}{4\tau_M}\right)$	$\frac{d(32\tau_M+6d)}{13\tau_M+8d}$	$\frac{4d\tau_M}{11\tau_M+2d}$

There is another set of tuning rules that are derived based on the reaction curve, termed Cohen and Coon tuning rules. Table 1.3 gives the PID controller parameters calculated from Cohen and Coon tuning rules.

For the estimation of time delay d, τ_M, and K_{ss} when using MATLAB, it is a fairly straight forward procedure to draw the lines and pinpoint the data points. The MATLAB command for finding the point on a graph is called ginput. For example, by typing

```
[a,b]=ginput(1)
```

a cross hair will appear on the MATLAB figure and a double click on the point of interest will yield the exact values we need. This graphic procedure will be demonstrated in the example section (see Section 1.5).

1.3.3 Food for Thought

1. Can you apply Ziegler-Nichols oscillation tuning method to a first order system? Why?
2. Can you apply the reaction curve based tuning rules to unstable systems? Why?
3. How do we decide the sign of the proportional feedback controller gain when using the Ziegler-Nichols oscillation method?
4. Can you envisage any potential danger when using Ziegler- Nichols oscillation method?
5. How do you design a step response experiment?
6. What information will the step response experiment provide?

7. How do you determine steady-state gain, parameter τ_M and time delay d from a reaction curve?
8. What are your observations when comparing Ziegler-Nichols and Cohen-Coon tuning rules, in terms of signs and values of K_c, τ_I and τ_D?
9. Is there any desired closed-loop performance specification among the tuning rules?

1.4 Model Based PID Controller Tuning Rules

This section will discuss the PID controller tuning rules that are derived based on a first order plus delay model. These tuning rules worked well in applications.

1.4.1 IMC-PID Controller Tuning Rules

The internal model control (IMC)-PID tuning rules (Rivera et al. (1986)) are proposed on the basis of a first order plus delay model:

$$G_{\rm M}(s) = \frac{K_{\rm ss}{\rm e}^{-ds}}{\tau_{\rm M}s + 1}.$$

When using the IMC-PID tuning rules, a desired closed-loop response is specified by the transfer function from the reference signal to the output:

$$\frac{Y(s)}{R(s)} = \frac{{\rm e}^{-ds}}{\tau_{\rm cl}s + 1}$$

where $\tau_{\rm cl}$ is the desired time constant chosen by the user. The PI controller parameters are related to the first order plus delay model and the desired closed-loop time constant $\tau_{\rm cl}$, which are given as:

$$\begin{aligned} K_{\rm c} &= \frac{1}{K_{\rm ss}}\frac{\tau_{\rm M}}{\tau_{\rm cl} + d} \\ \tau_{\rm I} &= \tau_{\rm M}. \end{aligned} \tag{1.46}$$

If the system has a second order transfer function with time delay in the following form:

$$G_{\rm M}(s) = \frac{K_{\rm ss}{\rm e}^{-ds}}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

then a PID controller is recommended. Assuming that $\tau_2 \leq \tau_1$, then the PID controller parameters are calculated as

$$\begin{aligned} K_{\rm c} &= \frac{1}{K_{\rm ss}}\frac{\tau_1}{\tau_{\rm cl} + d} \\ \tau_{\rm I} &= \tau_1 \\ \tau_{\rm D} &= \tau_2. \end{aligned} \tag{1.47}$$

Later on, it was realized that the choice of $\tau_{\rm I}$ basically led to a pole-zero cancellation in the control system. Such a pole-zero cancellation will limit, as discussed in the next chapter, the control system performance in the disturbance rejection, particularly when τ_1 is large[4]. In Skogestad (2003), the IMC-PID tuning rules are modified to reduce the integral time constant as

$$\tau_I = \min[\tau_1, 4(\tau_{cl} + d)] \tag{1.48}$$

while K_c and τ_D are unchanged from (1.47).

The IMC-PID controller tuning rules are also extended to integrating systems in Skogestad (2003). Although the system has an integrator as part of its dynamics, integral control is still required for disturbance rejection (see Chapter 2).

Assuming that the system has the integrator with delay model:

$$G_M(s) = K_{ss}\frac{e^{-ds}}{s} \tag{1.49}$$

then a PI controller is recommended with the following parameters:

$$\begin{aligned} K_c &= \frac{1}{K_{ss}(\tau_{cl} + d)} \\ \tau_I &= 4(\tau_{cl} + d). \end{aligned} \tag{1.50}$$

If the transfer function for the integrating system has the form:

$$G_M(s) = K_{ss}\frac{e^{-ds}}{s(\tau_1 s + 1)} \tag{1.51}$$

then a PID controller is recommended to have the following parameters:

$$\begin{aligned} K_c &= \frac{1}{K_{ss}(\tau_{cl} + d)} \\ \tau_I &= 4(\tau_{cl} + d) \\ \tau_D &= \tau_1. \end{aligned} \tag{1.52}$$

If the system has a double integrator with the transfer function

$$G_M(s) = K_{ss}\frac{e^{-ds}}{s^2} \tag{1.53}$$

then a PID controller is recommended with the following parameters:

$$\begin{aligned} K_c &= \frac{1}{K_{ss}(\tau_{cl} + d)^2} \\ \tau_I &= 4(\tau_{cl} + d) \\ \tau_D &= 4(\tau_{cl} + d). \end{aligned} \tag{1.54}$$

The IMC-PID controller tuning rules will be studied in Examples 2.1 and 2.2.

1.4.2 Padula and Visioli Tuning Rules

Several sets of tuning rules were introduced in Padula and Visioli (2011) and Padula and Visioli (2012). These tuning rules are based on the first order plus delay model:

$$G_M(s) = \frac{K_{ss}e^{-ds}}{\tau_M s + 1}.$$

[4] This slow disturbance rejection problem will be analyzed using sensitivity analysis in Problem 2.8.

Table 1.4 Padula and Visioli tuning rules (PI controller).

	$M_s = 1.4$	**$M_s = 2$**
K_c	$\frac{1}{K_{ss}}\left(0.2958\left(\frac{d}{d+\tau_M}\right)^{-1.014} - 0.2021\right)$	$\frac{1}{K_{ss}}\left(0.5327\left(\frac{d}{d+\tau_M}\right)^{-1.029} - 0.2428\right)$
τ_I	$\tau_M\left(1.624\left(\frac{d}{\tau_M}\right)^{0.2269} - 0.5556\right)$	$\tau_M\left(1.44\left(\frac{d}{\tau_M}\right)^{0.4825} - 0.1019\right)$

Table 1.5 Padula and Visioli tuning rules (PID controller).

	$M_s = 1.4$	**$M_s = 2$**
K_c	$\frac{1}{K_{ss}}\left(0.1724\left(\frac{d}{d+\tau_M}\right)^{-1.259} - 0.05052\right)$	$\frac{1}{K_{ss}}\left(0.2002\left(\frac{d}{d+\tau_M}\right)^{-1.414} + 0.06139\right)$
τ_I	$\tau_M\left(0.5968\left(\frac{d}{\tau_M}\right)^{0.6388} + 0.07886\right)$	$\tau_M\left(0.446\left(\frac{d}{\tau_M}\right)^{0.9541} + 0.1804\right)$
τ_D	$\tau_M\left(0.5856\left(\frac{d}{\tau_M}\right)^{0.5004} - 0.1109\right)$	$\tau_M\left(0.6777\left(\frac{d}{\tau_M}\right)^{0.4968} - 0.1499\right)$

They were derived using optimization methods for minimizing an error function together with the sensitivity peak in the frequency domain (see Chapter 2).

Here, we only include two sets of the tuning rules introduced for disturbance rejection in their paper. Tables 1.4 and 1.5 present the tuning rules for PI and PID controllers, respectively. Each table contains two sets of rules. For the specification of $M_s = 1.4$ (M_s corresponds to the sensitivity peak (see Chapter 2)), the set of tuning rules is expected to produce a slower closed-loop response in comparison to that with $M_s = 2$. It is worthwhile mentioning from Tables (1.4) and (1.5) that both PI and PID controller parameters are invalid for the time delay $d = 0$ because as $d \rightarrow 0$, the proportional control gain $K_c \rightarrow \infty$.

A frequency response analysis for the Padula and Visioli tuning rules based control system will be presented in Chapter 2 where an example (see Example 2.4) will be given to show the sensitivity functions and their Nyquist diagrams.

1.4.3 Wang and Cluett Tuning Rules

In Wang and Cluett (2000), a first order plus delay model was used to derive several tuning rules for PID controllers. The rules were calculated using a frequency response analysis based on the ratio of the time constant τ_M to the time delay d, which is defined as $L = \tau_M/d$. A desired closed-loop time constant is chosen as a scale of the time delay d. Among them, which works quite well in applications, is a particular choice with the desired closed-loop time constant being equal to the time delay d. For this choice, the PID controller parameters are presented in Table 1.6.

Table 1.6 Wang–Cluett tuning rules with reaction curve ($L = \tau_M/d$).

	K_c	τ_I	τ_D
P	$\frac{0.13+0.51L}{K_{ss}}$		
PI	$\frac{0.13+0.51L}{K_{ss}}$	$\frac{d(0.25+0.96L)}{0.93+0.03L}$	
PID	$\frac{0.13+0.51L}{K_{ss}}$	$\frac{d(0.25+0.96L)}{0.93+0.03L}$	$\frac{d(-0.03+0.28L)}{0.25+L}$

1.4.4 Food for Thought

1. In the IMC-PID controller tuning rules, if a faster closed-loop response is desired, would you increase or decrease the desired closed-loop time constant τ_{cl}?
2. For the tuning rules derived by Padula and Visioli, are there any closed-loop performance parameters chosen by the user?
3. Intuitively, would you increase the proportional controller gain K_c if the PID controlled system is unstable when using the Padula and Visioli's tuning rules?
4. Would you increase the integral time constant τ_I if the PID controlled system is oscillatory when using the Padula and Visioli's tuning rules?

1.5 Examples for Evaluations of the Tuning Rules

Several examples are presented in this section for evaluation of the tuning rules that are based on the first order plus delay model.

1.5.1 Examples for Evaluating the Tuning Rules

The first example is based on a first order plus delay plant and the second example is based a high order plant so that an approximation is made during the graphic procedure.

Example 1.6 *The unit step response of a continuous time transfer function model*

$$G(s) = \frac{0.5e^{-20s}}{30s+1} \tag{1.55}$$

is shown in Figure 1.15. Instead of using the first order plus delay model directly, we will find the PI controller parameters using the values of τ_M, K_{ss}, and delay d.

Solution. *A Simulink simulator is built to collect the step response testing data and to produce a figure for the step response. On this figure, a line is drawn to reflect the maximum slope of the reaction curve; there are two arrows marking the points of interest. Using MATLAB command ginput(2), with a click on the bottom point, we find the coordinates $t_1 = 21, Y_0 = -0.02$; and with a click on the top point, we find $t_2 = 58, Y_s = 0.5$.*

From the readings of the two points, we find that

$$K_{ss} = \frac{Y_s - Y_0}{U_s - U_0} \approx 0.5 \tag{1.56}$$

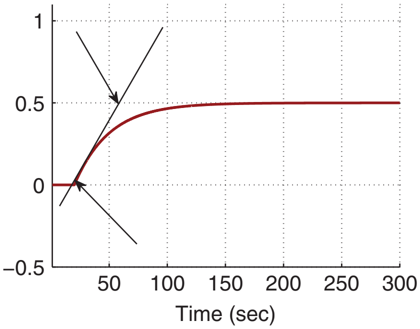


Figure 1.15 Unit step response (Example 1.6)

where $U_s - U_0$ is one since a unit step signal is used as the input. The time delay $d = t_1 = 21$, and the parameter $\tau_M = t_2 - t_1 = 58 - 21 = 37$. With these parameters, we calculate the PI controller parameters using the reaction curve based methods (see Tables 1.2, 1.3, and 1.6). The PI controller parameters are summarized in Table 1.7. Their closed-loop step responses are compared in Figure 1.16.

In reality, instead of a pure first order plus delay dynamics, there are more or less additional dynamics in the system. The tuning rules are applicable for a more complex system. We illustrate how to apply the tuning rules using the example below. This example also illustrates the fact that we need to be cautious when applying the tuning rules and be aware of their limitations.

Table 1.7 PI controller parameters with reaction curve.

	K_c	τ_I
Ziegler–Nichols	3.1714	63
Cohen–Coon	3.3381	32.7131
Wang–Cluett	2.0571	41.4811

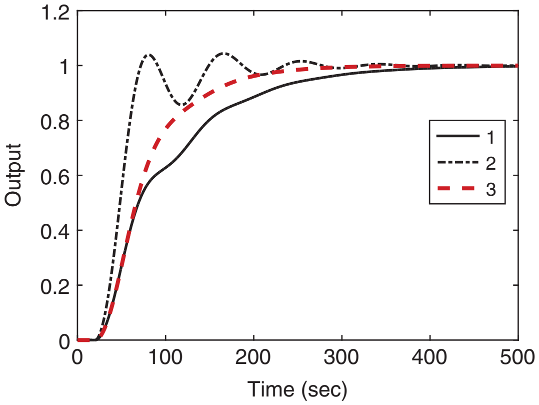


Figure 1.16 Closed-loop unit step response with PI controller (Example 1.6). Key: line (1) Ziegler–Nichols tuning rule; line (2) Cohen–Coon tuning rule; line (3) Wang–Cluett tuning rule.

Example 1.7 *A continuous time plant has the transfer function:*

$$G(s) = \frac{0.5e^{-20s}}{(30s+1)^3} \tag{1.57}$$

The unit step response of this transfer function model is shown in Figure 1.17. In this figure, a line is drawn to reflect the maximum slope of the reaction curve and the points of interest are marked by two arrows.

Using MATLAB command ginput(2), with a click on the bottom point, we will find the coordinates $t_1 = 36, Y_0 = -0.0022 \approx 0$*; and a click on the top point, we will find* $t_2 = 164, Y_s = 0.4981 \approx 0.5$*. Find the PI and PID controllers using the reaction curve based tuning rules.*

Solution. *The steady state gain* $K_{ss} = \frac{Y_s - Y_0}{1} = 0.5$*. The time delay is* $d = t_1 = 36$ *and the parameter* $\tau_M = t_2 - t_1 = 164 - 36 = 128$*. The PI controller parameters are calculated using the reaction curve based methods (see Tables 1.2, 1.3, and 1.6) and are summarized in Table 1.8. The closed-loop control systems with the PI controllers are simulated using the plant model (1.57). The unit closed-loop step responses are shown in Figure 1.18. From this figure, we can see that both PI controllers from Ziegler–Nichols and Cohen–Coon tuning rules failed to produce a stable closed-loop system. However, the PI controller using Wang–Cluett tuning rule gives a stable closed-loop system.*

The PI control systems will be subsequently used as examples for closed-loop stability analysis. The Nyquist plots of the PI controllers used in this example are analyzed in Example 2.3 of Chapter 2.

The following example illustrates the applications of the Padula and Visioli tuning rules.

Figure 1.17 Unit step response (Example 1.7).

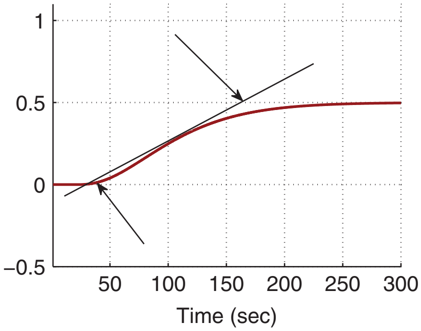


Table 1.8 PI controller parameters with reaction curve.

	K_c	τ_I
Ziegler–Nichols	6.4	108
Cohen–Coon	6.5667	75.9231
Wang–Cluett	3.8867	127.2154

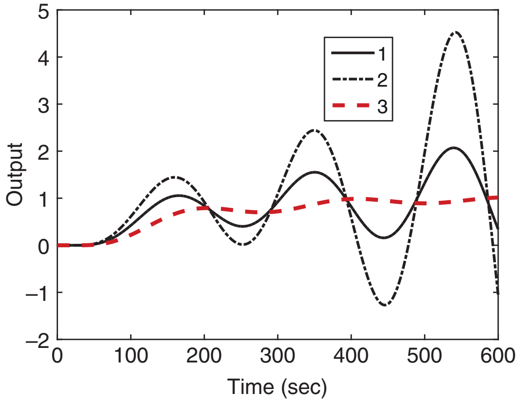


Figure 1.18 Closed-loop unit step response with PI controller (Example 1.7). Key: line (1) Ziegler–Nichols tuning rule; line (2) Cohen–Coon tuning rule; line (3) Wang–Cluett tuning rule.

Example 1.8 *Consider the same third order system with time delay used in Example 1.7. Find the PI and PID controller parameters using Padula and Visioli tuning rules and simulate their closed-loop step response.*

Solution. *The parameters used in the tuning rules are $K_{ss} = 0.5$, $d = 36$, and $\tau_M = 128$. To evaluate the PI controller performance, Table 1.4 is used to calculate the controller parameters. For $M_s = 1.4$, we have $K_c = 2.3487$ and $\tau_I = 84.7649$. For $M_s = 2$, we have $K_c = 4.5861$ and $\tau_I = 86.9015$.*

With sampling interval $\Delta t = 1$ (s), the closed-loop step responses are compared in Figure 1.19, where the IP controller structure is used to reduce overshoot to the step reference signal. It is clearly seen that the closed-loop system with $M_s = 1.4$ is stable; however, with $M_s = 2$ it is not.

Now, we evaluate the closed-loop performance for a PID controller with filter, where the filter time constant is chosen to be $0.1\tau_D$. Based on Table 1.5, the PID controller parameters are calculated as for $M_s = 1.4$, $K_c = 2.2253$, $\tau_I = 41.8298$, $\tau_D = 25.5365$, and for $M_s = 2$, $Kc = 3.54$, $\tau_I = 40.1098$, $\tau_D = 27.0037$.

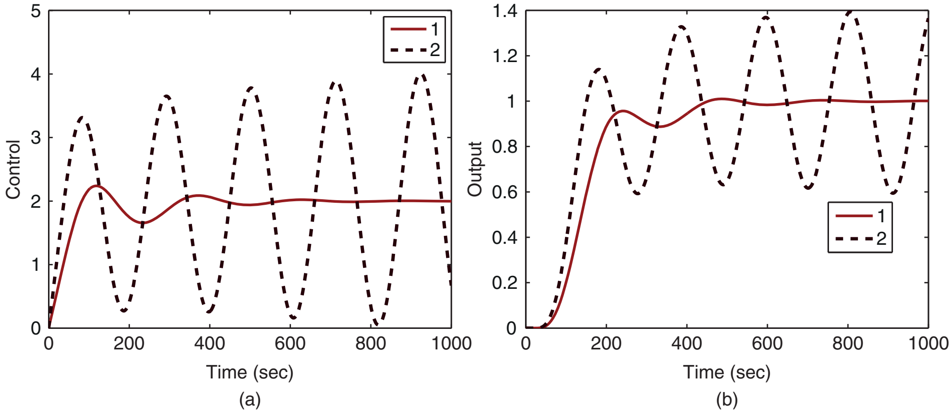


Figure 1.19 Comparison of closed-loop responses using Padula and Visioli PI controller (Example 1.8). (a) Control signal. (b) Output. Key: line (1) tuning rule with $M_s = 1.4$; line (2) tuning rule with $M_s = 2$.

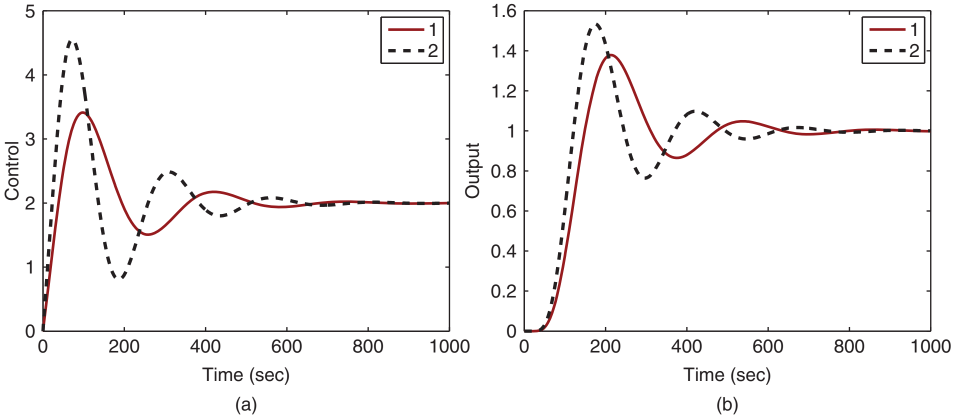


Figure 1.20 Comparison of closed-loop responses using Padula and Visioli PID controller (Example 1.8). (a) Control signal. (b) Output. Key: line (1) tuning rule with $M_s = 1.4$; line (2) tuning rule with $M_s = 2$.

With the same sampling interval $\Delta t = 1$, and both proportional and derivative control on output only (IPD structure) where the derivative filter time constant is selected as $0.1\tau_D$, the closed-loop responses are simulated. Figure 1.20 compares the closed-loop responses. Both tuning rules lead to stable closed-loop control systems. It is seen that there are overshoots in both reference responses, which was caused by the quite large derivative gains.

1.5.2 Fired Heater Control Example

A fired heater using gas fuel is a heating furnace that is typically used for household heating in the winter times. In this case study, the input to the heating furnace is the feed rate of the gas fuel and the output is the heater outlet or the room temperature in a house. Because the temperature sensors are located away from the heating source, there is a time delay in the measured temperature when the input feed rate changes. Additionally, depending on the operating conditions of the input feed rate, the dynamic response of the temperature is different. Two transfer function models are given in Ralhan and Badgwell (2000) to describe the operations of a gas fired heater at a low fuel operation and a high fuel operation. At the low fuel operating condition, the transfer function is described as

$$G_L(s) = \frac{3e^{-10s}}{(4s+1)^2} \frac{degC}{sm^3/s} \tag{1.58}$$

and at the high fuel operating condition,

$$G_H(s) = \frac{e^{-5s}}{(5s+1)^2} \frac{degC}{sm^3/s} \tag{1.59}$$

where the time constant is in minutes. Note that there are dramatically differences in time delay and the steady-state gain of the transfer function models.

In this study, we will introduce the effect of input disturbance in the closed-loop simulation by adding a step signal with negative magnitude to the control signal. This input disturbance represents a sudden change in the process that causes the output temperature drop. The effect of this disturbance is further discussed in Chapter 2.

Example 1.9 *In this example, we will show how to use the tuning rules to find the PID controller parameters for the fired heater at the lower operating condition using the transfer function (1.58) and simulate the closed-loop response with a step reference signal using sampling interval* $\Delta t = 5$ *(min) and a negative step disturbance entering at the half of the simulation time.*

The higher operating condition case is left as an exercise.

Solution. *Figure 1.21 shows the unit step response with the lines drawn to identify the time delay and time constant for a first order approximation. From the graph, the time delay is found as* 9.54 *min and the time constant* $\tau_M = 23 - 9.54 = 13.48$ *min. With the steady-state gain equal to* 3*, the approximation using first order plus model leads to the following transfer function:*

$$G(s) = \frac{3e^{-9.54s}}{13.48s + 1} \tag{1.60}$$

Now, applying the Ziegler–Nichols tuning rules (see Table 1.2), Cohen–Coon tuning rules (see Table 1.3) and Wang–Cluett tuning rules (see Table 1.6), we obtain the PI controller parameters for the fired heater process shown in Table 1.9. The PI controller parameters obtained are drastically different. The PI controllers using Ziegler–Nichols and Wang–Cluett tuning rules produce stable closed-loop system for the fired heater

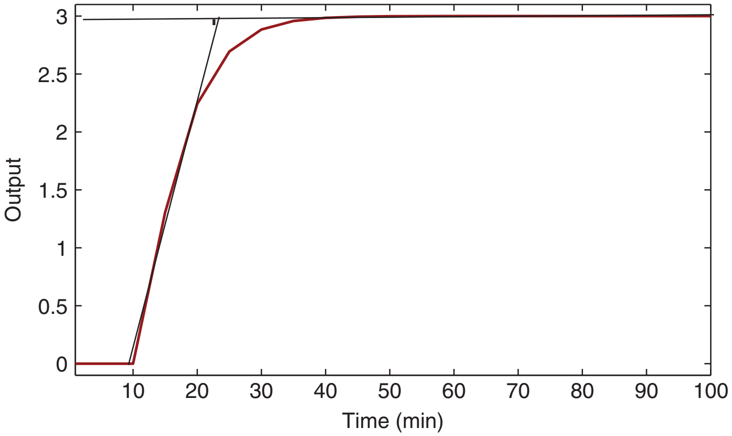


Figure 1.21 Unit step response of the fired heater process.

Table 1.9 PI controller parameters with reaction curve.

	K_c	τ_I
Ziegler–Nichols	0.4239	28.6200
Cohen–Coon	0.4517	13.2353
Wang–Cluett	0.2835	15.7610

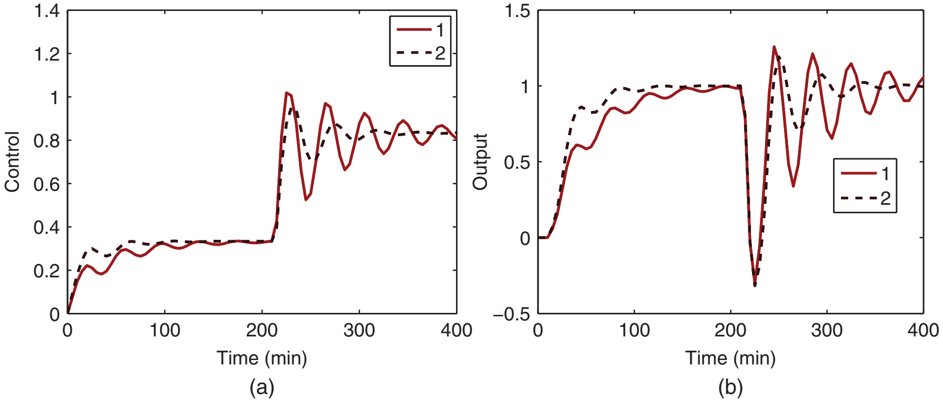


Figure 1.22 Comparison of closed-loop responses using Ziegler–Nichols and Wang–Cluett tuning rules (Example 1.9). (a) Control signal. (b) Output. Key: line (1) Ziegler–Nichols tuning rule; line (2) Wang–Cluett tuning rule.

process, however the PI controller using Cohen–Coon tuning rules does not lead to a stable closed-loop system, which was verified using closed-loop simulation. To evaluate the closed-loop control performance, a unit step input signal is used as a reference and a step input disturbance with magnitude of −0.5 *is added to the closed-loop simulation at half of the simulation time. Figure 1.22(a) shows the control signals generated by the PI controllers and Figure 1.22(b) shows the output responses to the reference change and the disturbance signal. Both closed-loop systems have oscillations, but in comparison, the controller using Wang–Cluett tuning rules leads to a slightly better closed-loop performance with less oscillations.*

1.6 Summary

This chapter has introduced the basics in PID control systems. It is important for us to understand the roles of proportional control, integral control and derivative control. The simple modification with the implementation of PID controller by putting the proportional control on output only reduces the overshoot in the step reference response. For some applications, avoiding the overshoot is important because the requirement is associated with the system's operational constraints while for other applications the overshoot is preferred if the PID controller is used in an inner-loop system (see Chapter 7). Additionally the derivative control should be implemented with a derivative filter to avoid amplification of measurement noise and the derivative control should be implemented on the output only to avoid the scenario of derivative 'kick'.

Several sets of tuning rules have been introduced in this chapter. These tuning rules are very simple and easy to use if the system can be approximated by a first-order-plus delay model. However, they offer no guarantees on the closed-loop performance as demonstrated by the simulation examples. Some of the examples used in this chapter will be analyzed using the Nyquist stability criterion and sensitivity functions in Chapter 2.

1.7 Further Reading

1. Text books in control engineering include, Franklin et al. (1998), Franklin et al. (1991), Ogata (2002), Golnaraghi and Kuo (2010), Goodwin et al. (2000) and Astrom and Murray (2008).
2. Process control books include Marlin (1995), Ogunnaike and Ray (1994), Seborg et al. (2010).
3. There are many books published on PID control, including Astrom and Hagglund (1995), Astrom and Hagglund (2006), Yu (2006), Johnson and Moradi (2005), Visioli (2006), Tan et al. (2012). PID control of multivariable systems is discussed in Wang et al. (2008).
4. Tuning rules are compiled as a book (O'Dwyer (2009)). PID controller tuning rules with performance specifications derived from frequency response analysis are introduced in Wang and Cluett (2000).
5. The survey and tutorial papers on PID control include Åström and Hägglund (2001), Ang et al. (2005), Li et al. (2006), Knospe (2006), Cominos and Munro (2002) and Visioli (2012), Blevins (2012).
6. A web-based laboratory for teaching PID control is introduced in Ko et al. (2001), Yeung and Huang (2003).
7. The issues associated with derivative filters in PID controller are addressed in Luyben (2001), Hägglund (2012), Hägglund (2013), Isaksson and Graebe (2002), Larsson and Hägglund (2011).
8. Improving reference tracking and reducing overshoot is discussed for an existing controller in an industrial environment (Visioli and Piazzi (2003)).
9. Ziegler-Nichols tuning formula is refined in Hang et al. (1991) and for unstable systems with time delay in De Paor and O'Malley (1989). The ZieglerNichols step response method is revisited from the point of view of robust loop shaping (Åström and Hägglund (2004)). There is a set of tuning rules for integrating plus delay model in Tyreus and Luyben (1992), which was extended to PID controllers in Luyben (1996).

Problems

1.1 *The following system is given to practice the Ziegler-Nichols tuning rules based on oscillation testing data to determine the PID controller parameters:*

$$G(s) = \frac{10}{(s+1)^3}$$

1. *Build a Simulink simulator using closed-loop proportional control with a controller K for the system. The sampling interval Δt is chosen to be* 0.1 *sec.*
2. *Find the PI and PID controller parameters using Table 1.1.*
3. *Implement the PI and PID controllers in Simulink with a step reference signal by putting both proportional and derivative control on the output only.*
4. *What are your observations with respect to the closed-loop performance of the PI and PID control system?*

1.2 *In the majority of the tuning rules, the key step is to find a first order plus delay approximate model for step response testing data. Because the step response of a first order system ($G(s) = \frac{K_{ss}}{\tau_M s+1}$) to a unit step input signal can be expressed as*

$$g(t) = K_{ss}(1 - e^{\frac{-t}{\tau_M}})$$

we can determine the time constant τ_M using 63.2 *percent of the rising time in the step response.*

1. *Why does the* 6.32 *percent of the rising time correspond to the time constant τ_M?*
2. *Construct a graphic method to find a first order plus time delay model for the following system:*

$$G(s) = \frac{0.5e^{-20s}}{(30s+1)^3}$$

3. *Compare this first order plus delay model with that found in Example 1.7.*
4. *Find the PID controller parameters using Table 1.2 and compare the closed-loop simulation results with the ones given in Example 1.7. What are your observations?*
5. *If the step response data contained severe measurement noise, would it be more difficult to determine the rising time?*

1.3 *The transfer function for the fired heater system with high operating condition introduced in Section 1.5.2 is given as*

$$G_H(s) = \frac{e^{-5s}}{(5s+1)^2}\frac{degC}{sm^3/s} \tag{1.61}$$

1. *Find the first order plus delay approximate model for this transfer function using the graphic method in Section 1.3.2. The sampling interval Δt is chosen to be* 1.
2. *Determine the PID controllers using Tables 1.2–1.6.*
3. *Evaluate their closed-loop performance by simulating their closed-loop unit step responses. What are your observations in terms of closed-loop performance with respect to the tuning rules?*

1.4 *The transfer function for the fired heater system with low operating condition introduced in Section 1.5.2 is given as*

$$G_L(s) = \frac{3e^{-10s}}{(4s+1)^2}\frac{degC}{sm^3/s} \tag{1.62}$$

1. *Design three PID controllers for this system using IMC-PID controller design equations shown in (1.47) where the desired closed-loop time constant $\tau_{cl} = 20$,* 30 *and* 40 *respectively.*
2. *Evaluate the closed-loop control system performance for the three PID controllers by simulating the closed-loop unit step response with sampling interval $\Delta t = 0.1$ sec.*
3. *What are your observations of the closed-loop performance when the desired closed-loop time constant τ_{cl} increases?*

1.5 *The two transfer functions obtained from the fired heater system are drastically different at the two operating conditions (see (1.61)–(1.62)). Hence, the PID controllers are different for the two operating conditions. Assume that we would only use one PID controller for both operating conditions.*

1. *Design IMC-PID controller ($C_H(s)$) for the fired heater system at high operating condition using $G_H(s)$ with $\tau_{cl} = 30$.*
2. *Evaluate the closed-loop performance by simulating the two PID control systems: (1) $C_H(s)$ and $G_H(s)$; (2) $C_H(s)$ and $G_L(s)$, respectively.*
3. *Let $C_L(s)$ denote the PID controller found from Problem 1.4 with $\tau_{cl} = 30$. Evaluate the closed-loop performance by simulating the two PID control systems: (1) $C_L(s)$ and $G_L(s)$; (2) $C_L(s)$ and $G_H(s)$.*
4. *Based on the simulation studies, which controller should we recommend?*
5. *Increase the desired time constant τ_{cl} to 40 and repeat the evaluations. What would be the recommendations for the choice of τ_{cl}?*

2

Closed-loop Performance and Stability

2.1 Introduction

Because feedback control can cause a closed-loop system to become unstable, ensuring closed-loop stability is paramount in control system design. In Section 2.2, we will discuss closed-loop stability by examining the locations of closed-loop poles and the applications of the Routh–Hurwitz stability criterion. In Section 2.3, the Nyquist stability criterion is presented based on a frequency response analysis, which leads to conclusions of closed-loop stability by examining the frequency response of the loop transfer functions, particularly convenient for analysis of systems with time delay.

To understand the roles of external signals playing in a feedback control system, Section 2.4 introduces control system structures with different degrees of freedom, which are related to the topic of reducing overshoot in reference response discussed in Chapter 1. Also in this section, the sensitivity functions are introduced in relation to various external signals in the closed-loop system. In Sections 2.5 and 2.6, we will examine the key issues existed in a feedback control system that are related to reference following, disturbance rejection, and noise attenuation from the angles of sensitivity analysis. The final section of this chapter will discuss the robust stability using frequency response analysis. Many examples presented in this chapter use PID controllers designed with the tuning rules given in Chapter 1.

2.2 Routh–Hurwitz Stability Criterion

Feedback control can cause a system to become unstable. Ensuring closed-loop stability is the most important aspect in control system design. Therefore, for every control system designed, its closed-loop stability is required to be checked as the top priority.

For linear time invariant systems, there are two main categories of methods that have been widely used to check closed-loop stability. The first type is based on the direct computation of the poles of the closed-loop transfer function. We call them the closed-loop poles. If all the closed-loop poles have negative real parts, namely with all poles strictly on the left half of the complex plane, then the closed-loop system is stable. If there are one or more poles on the right half of the complex plane, having a positive real component, then the closed-loop system is unstable. If there are one or more poles with real part equal to zero, namely on the imaginary axis of the complex plane, we call this type of system marginally stable. The second type is based on the open-loop

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

transfer function that includes the plant transfer function, sensor and actuator transfer functions, and the controller transfer function. The Nyquist stability criterion is one of the most widely used methods that is based on the open-loop frequency response analysis.

2.2.1 Determining Closed-loop Poles

The first step in the computation of closed-loop poles is to calculate the closed-loop transfer function. Once the closed-loop transfer function is determined, then the MATLAB function roots.m is used to find the zeros of the denominator of the closed-loop transfer function, which are the poles of the closed-loop transfer function.

Note that in the computation of the closed-loop transfer function, the sensor and actuator dynamics will be considered in addition to the controller transfer function.

Example 2.1 *Consider the process described by the transfer function:*

$$G(s) = \frac{1}{(s+1)(0.2s+1)} \tag{2.1}$$

where we assume that both sensor and actuator dynamics are included in the transfer function. Use IMC-PI controller tuning rules to determine the PI controller parameters and find the closed-loop poles.

Solution. *From Skogestad (2003), the second order system (2.1) is approximated by the first order plus delay model:*

$$G_{\mathrm{A}}(s) = \frac{e^{-ds}}{\tau_1 s + 1} = \frac{e^{-0.1s}}{1.1s + 1}$$

According to the IMC-PID controller tuning rules, the proportional gain and the integral time constant are calculated as

$$K_{\mathrm{c}} = \frac{\tau_1}{\tau_1 + d} = 0.9167; \quad \tau_{\mathrm{I}} = \min[\tau_1, 4(\tau_{\mathrm{cl}} + d)] = 1.1$$

where the closed-loop time constant τ_{cl} is selected as $\tau_{\mathrm{cl}} > 0.25\tau_1$.

Now, to calculate the closed-loop poles, we write the original the transfer function as

$$G(s) = \frac{B(s)}{A(s)}$$

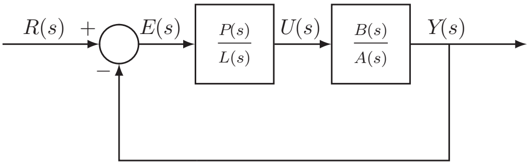
and the controller transfer function as

$$C(s) = \frac{P(s)}{L(s)} = \frac{K_{\mathrm{c}} s + \frac{K_{\mathrm{c}}}{\tau_{\mathrm{I}}}}{s} \tag{2.2}$$

$$= \frac{0.9167s + 0.8333}{s}. \tag{2.3}$$

Note that in the calculation of the closed-loop poles, we need to use the original process model (2.1), not the first order plus time delay model. The closed-loop block diagram is illustrated in Figure 2.1.

Figure 2.1 Closed-loop control system in transfer function form.



The closed-loop transfer function from the reference signal $R(s)$ to the output $Y(s)$ is calculated as

$$\frac{Y(s)}{R(s)} = \frac{P(s)B(s)}{L(s)A(s) + P(s)B(s)} \tag{2.4}$$

Clearly, the denominator of the closed-loop system is $L(s)A(s) + P(s)B(s)$ and the closed-loop poles are the solutions of the polynomial equation:

$$L(s)A(s) + P(s)B(s) = 0. \tag{2.5}$$

This polynomial equation is called the closed-loop characteristic equation. For this example, we have

$$\begin{aligned} L(s)A(s) + P(s)B(s) &= s(s+1)(0.2s+1) + 0.9167s + 0.8333 \\ &= 0.2s^3 + 1.2s^2 + 1.9167s + 0.8333. \end{aligned}$$

We use the MATLAB roots.m function as

```
den=[0.2000    1.2000    1.9167    0.8333];
roots(den)
```

and find the three closed-loop poles as $-3.7304, -1.5482, -0.7214$. Clearly, the closed-loop IMC-PI control system is stable with all poles on the left half of the complex plane. The closed-loop system has three closed-loop time constants. The dominant time constant is the largest calculated using the smallest magnitude of the closed-loop pole as

$$\tau = \frac{1}{0.7214} = 1.3862.$$

We pay attention to the largest time constant because it determines how fast the closed-loop response will be. For instance, this time constant is larger than the dominant process time constant, which is equal to 1. We may conclude that the closed-loop system response speed is slower than the open-loop process response speed because the closed-loop dominant time constant is larger.

2.2.2 Routh–Hurwitz Stability Criterion

It would not be a straightforward task if we had to calculate the closed-loop poles using pencil and paper for a third order system and above. The Routh–Hurwitz stability criterion was introduced in the development of control systems early on so that their closed-loop stability could be determined using pencil and paper. Without involving complex computation, using the Routh–Hurwitz stability criterion enables us to determine whether there are any closed-loop poles on the right half of the complex plane and how many of them there are. This simple calculation is performed using the coefficients of the denominator of the closed-loop transfer function.

Table 2.1 Routh–Hurwitz table.

s^n	a_n	a_{n-2}	a_{n-4}	$\dots$
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	$\dots$
s^{n-2}	$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$\dots$
s^{n-3}	$r_{3,1}$	$r_{3,2}$	$r_{3,3}$	$\dots$
$\vdots$	$\vdots$			
s^2	$r_{n-2,1}$	$r_{n-2,2}$		
s	$r_{n-1,1}$			
s^0	$r_{n,1}$			

In general, for the denominator of the closed-loop transfer function with the polynomial

$$A(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_0,$$

we use its coefficients to form the first two rows of the Routh–Hurwitz table (see Table 2.1), and based on the first two rows complete the rest of the elements in the table. The first element in the third row is

$$r_{2,1} = -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{bmatrix}$$

where it is seen that we used the nearest two previous rows to form the determinant scaled by the factor $-\frac{1}{a_{n-1}}$. The second element in the third row is

$$r_{2,2} = -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{bmatrix}$$

which is the same scaling but with the replacement of the next column in the determinant. With the same pattern,

$$r_{2,3} = -\frac{1}{a_{n-1}} \det \begin{bmatrix} a_n & a_{n-6} \\ a_{n-1} & a_{n-7} \end{bmatrix}.$$

For the fourth row, we will use the elements in the nearest two rows (second and third rows) with the same pattern, where

$$r_{3,1} = -\frac{1}{r_{2,1}} \det \begin{bmatrix} a_{n-1} & a_{n-3} \\ r_{2,1} & r_{2,2} \end{bmatrix}$$
$$r_{3,2} = -\frac{1}{r_{2,1}} \det \begin{bmatrix} a_{n-1} & a_{n-5} \\ r_{2,1} & r_{2,3} \end{bmatrix}.$$

The remainder of the elements are expressed in a general form as

$$r_{k,j} = -\frac{1}{r_{k-1,1}} \det \begin{bmatrix} r_{k-2,1} & r_{k-2,j+1} \\ r_{k-1,1} & r_{k-2,j+1} \end{bmatrix}$$

where $j = 1, 2, \dots$.

The elements in the first column of the Routh–Hurwitz table will determine whether the closed-loop system is stable. The number of roots of the polynomial $A(s)$ with real part greater than zero is equal to the number of sign changes in the first column of the Routh–Hurwitz table. Simply, if the first coefficient a_n in $A(s)$ is positive, then closed-loop stability will follow if all elements in the first column are positive.

Note that, in the calculation of the table, if an element in the first column becomes zero, then the calculation continues by replacing the zero element with a variable $|\delta|$. The remainder of the elements will be expressed in the function of $|\delta|$. Closed-loop stability will be determined by examining the coefficients in the first row using $|\delta| \rightarrow 0$.

If the coefficients of the closed-loop transfer function are constant and known, the closed-loop poles are simply calculated using the MATLAB function roots.m, as illustrated in Example 2.1. However, the Routh–Hurwitz stability criterion is very effective in determining closed-loop stability when some of the controller or process parameters have uncertainties.

Example 2.2 *Continuing from Example 2.1, determine the minimum value of τ_I in the IMC-PID tuning rule to produce a stable closed-loop system.*

Solution. *From the controller transfer function (2.3), we can express the τ_I variation in the controller parameters as*

$$C(s) = \frac{0.9167s + 0.9167k_I}{s}$$

where $k_I = \frac{1}{\tau_I}$. Therefore, the closed-loop polynomial becomes

$$L(s)A(s) + P(s)B(s) = 0.2s^3 + 1.2s^2 + 1.9167s + 0.9167k_I$$

From the closed-loop polynomial, we obtain the first two rows in the Routh–Hurwitz table (see Table 2.2).

The parameter x_1 is calculated using the following formulae:

$$x_1 = -\frac{1}{1.2}\det\left(\begin{bmatrix} 0.2 & 1.9167 \\ 1.2 & 0.9167k_I \end{bmatrix}\right) = -\frac{1}{1.2}(0.2 \times 0.9167k_I - 1.2 \times 1.9167).$$

To ensure closed-loop stability, we require $x_1 > 0$, which means that

$$k_I < \frac{1.2 \times 1.9167}{0.2 \times 0.9167} = 12.545.$$

Therefore $\tau_I > 1/12.545 = 0.07971$. The parameter x_2 is calculated as

$$x_2 = -\frac{1}{x_1}\det\left(\begin{bmatrix} 1.2 & 0.9167k_I \\ x_1 & 0 \end{bmatrix}\right) = 0.9167k_I.$$

Table 2.2 Routh–Hurwitz table for the third order system.

s^3	0.2	1.9167
s^2	1.2	$0.9167k_I$
s^1	x_1	0
s^0	x_2	0

The requirement of $x_2 > 0$ means that $k_I > 0$.

Combining the imposed conditions from x_1 and x_2, we conclude that closed-loop stability is ensured if $\tau_I > 0.07971$. One can verify using the MATLAB roots.m function that when $\tau_I = 0.07971$, the three closed-loop poles are: $-6, \pm j3.0957$. The closed-loop system is marginally stable because a pair of its poles is on the imaginary axis of the complex plane. For any τ_I less than 0.07971, the closed-loop system becomes unstable.

This example also indicates that there is a limit on what value we can select for the parameter τ_{cl} in the IMC-PID controller tuning rules.

2.2.3 Food for Thought

1. For a given system, when the closed-loop poles are found to be −1, −2 and −3, what kind of behavior do you expect the closed-loop output response to be?
2. When the closed-loop poles are found to be $-0.1 \pm j5$, what kind of output response do you expect?
3. Why do the complex poles always appear in a complex conjugate form?
4. Can we determine closed-loop stability if one of the coefficients in the leading column of Routh-Hurwitz stability criterion is zero?
5. Can we apply Routh-Hurwitz stability criterion to systems with time delay?

2.3 Nyquist Stability Criterion

The Nyquist stability criterion is one of the most widely used tools in analyzing closed-loop stability. Using MATLAB graphic tools, the Nyquist diagram is very easy to produce by calculating the frequency response of the loop transfer function. The gain margin, phase margin and delay margin provide valuable insight into closed-loop stability with respect to parameter variations.

2.3.1 Nyquist Diagram

The Nyquist stability criterion uses the frequency response of an open-loop system, including the plant, the sensors and the actuator, to determine closed-loop stability. Here, the open-loop system $M(s)$ is expressed as

$$M(s) = G(s)C(s)$$

where $G(s)$ represents the system dynamics including the plant dynamics, the actuator dynamics, and the sensor dynamics, and $C(s)$ represents the controller transfer function. More specifically, it is a graphic approach using real and imaginary values of the frequency response $M(j\omega)$ where $M(j\omega)$ is that of the loop transfer function. The benefit of being a graphic approach lies in the intuition as well as both quantitative and qualitative measures.

As an illustration, the Nyquist loci of a PI controlled system that has a stable open-loop transfer function with time delay is shown in Figure 2.2, where the following loop transfer function is used:

$$M(s) = \frac{3187.4}{s + 3355.2} \frac{594.9}{s + 1.956} \frac{0.0271s + 0.2737}{s} e^{-0.03s}. \tag{2.6}$$

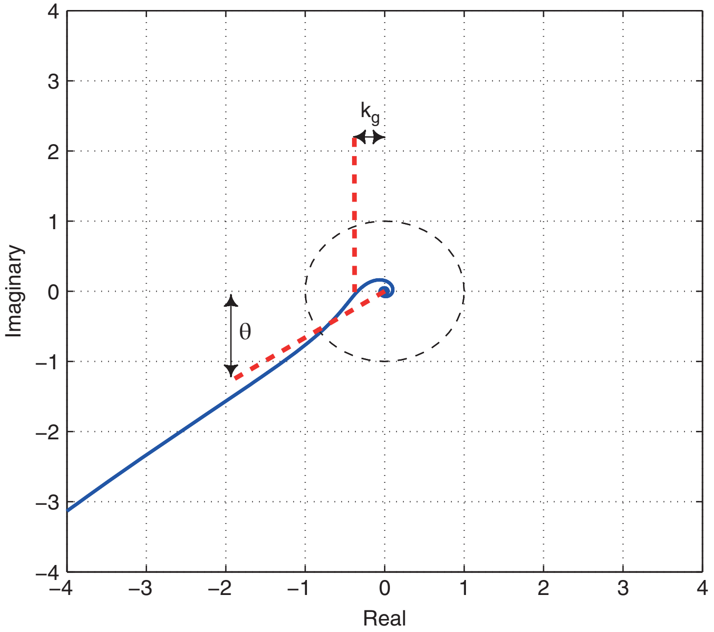


Figure 2.2 Nyquist plot with a unit circle for illustration of gain margin and phase margin. Solid line: Nyquist loci; dashed lines: pointers for the gain margin and phase margin.

We can specify the frequency vector ω and use the MATLAB function freqs.m to calculate the frequency response of the open-loop transfer function. The time delay component $e^{-j\omega d}$ is computed first using the MATLAB exponential function and multiplied by the frequency response of the rest of the open-loop transfer functions. Tutorial 2.1 shows the MATLAB program for the Nyquist plot.

The Nyquist criterion states that a feedback control system with single input and single output is stable if and only if, for the frequency response of the loop transfer function $M(s)$, number of counter clockwise encirclements of the $(-1, 0)$ point is equal to the number of poles of this loop transfer function with positive real parts. Note that this criterion presents both necessary and sufficient conditions for closed-loop stability using its open-loop transfer function. There are two comments related as below.

1. For the majority of PID controlled systems, this loop transfer function does not contain any poles that have positive real parts. Thus, for closed-loop stability of these classes of systems, the Nyquist stability criterion simply becomes that the frequency response should not encircle the $(-1, 0)$ point on the complex plane.
2. Because it uses the frequency response, closed-loop stability for systems with time delay will be examined without approximation. This is one of the most important advantages for using Nyquist loci to analyze control systems.

There are several quantitative, yet intuitive measurements that can be derived from the Nyquist loci. These quantities, termed gain margin, phase margin, and delay margin, are frequently used to assess the performance of the designed control system using the frequency information and safe-guard the closed-loop system against future model uncertainties.

2.3.1.1 Gain Margin

Gain margin is a quantity that is used to measure how much variation in gain the feedback control system could sustain before it became unstable. As illustrated in Figure 2.2, the gain margin is defined as GM $= \frac{1}{k_g}$, where k_g is the distance between the origin of the complex plane and the point that $M(j\omega)$ intersects the real axis (see the vertical dashed line in Figure 2.2). This means that if the loop gain were to exceed $\frac{1}{k_g}$, then the closed-loop system would become unstable. The parameter k_g can be easily determined from the Nyquist loci. Using the following MATLAB command:

```
[x,y]=ginput(1)
```

a cross hair appears on the Nyquist plot. By overlaying the center of the cross hair on the point that the Nyquist curve intersects the real axis, we obtain the coordinates as $x = -0.3226$ and $y = 8.8818e - 016$. The distance is $k_g = 0.3226$. Therefore, the gain margin is determined as $\frac{1}{k_g} = 3.1$. This means that if we were to increase the loop gain to three times the original value, then the closed-loop system would become unstable. We can associate this gain margin with the variations in the steady-state gain of the plant, the sensor, the actuator, or the controller gain K_c. It is the net effect of all the combined variations of gains.

2.3.1.2 Phase Margin

To identify the phase margin, we first draw a unit circle with its origin located at the origin of the complex plane, as shown in Figure 2.2, and a straight dashed line that connects the point when the circle intersects the Nyquist loci with the origin of the complex plane. The phase margin θ is the angle between the negative real axis and the dashed line. Clearly, it is the additional phase lag that could be associated with $M(j\omega)$ before the closed-loop system became unstable. The phase margin θ can be calculated using the following MATLAB function ginput.m. When using the following MATLAB function:

```
[x,y]=ginput(1)
```

a cross hair appears. Overlaying the center of the cross hair on the point that the unit circle intersects the Nyquist loci, we obtain the coordinates of x and y for that point. From Figure 2.2, the coordinates are $x = -0.8018$ and $y = -0.5664$. Then the phase margin θ is computed as

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) = \tan^{-1}\left(\frac{-0.5664}{-0.8018}\right) = 0.6150\,(\text{rad}).$$

2.3.1.3 Delay Margin

Although the phase margin θ represents how much additional phase lag can be added to the feedback control system before it becomes unstable, it does not directly convey the size of maximum time delay that can be added to the system. To determine the maximum time delay that can be tolerated, we let

$$e^{-j\theta} = e^{-jd_m\omega_p}$$

where d_m is the delay margin or the maximum delay to be tolerated and ω_p is the frequency when the unit circle intersects with the Nyquist loci. This yields

$$d_m = \frac{\theta}{\omega_p}$$

Clearly, a larger ω_p would lead to a smaller delay margin given the same phase margin θ. Thus, the frequency ω_p is an important parameter. To determine the frequency ω_p,

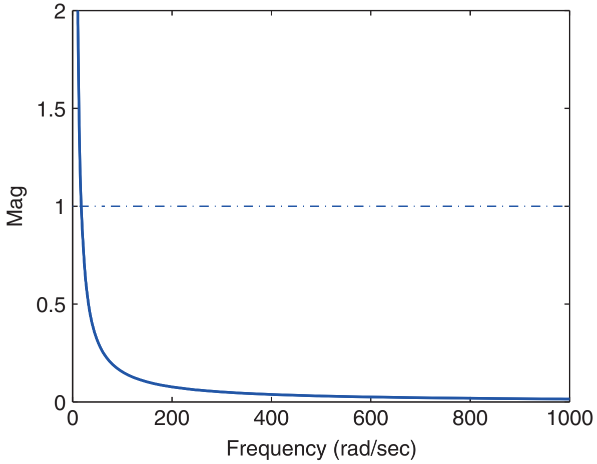


Figure 2.3 Magnitude of $M(\mathrm{j}\omega)$ (solid line) together with dashed line to determine ω_p.

we will plot $|M(\mathrm{j}\omega)|$, as shown in Figure 2.3. Using the function ginput.m, we identify the intersecting point of the dashed line with the solid line in Figure 2.3 that has the coordinates $x = 19.58$ and $y = 0.9998$. Thus, $\omega_p = 19.58$ rads^{-1}. From the parameter ω_p and the phase margin θ, we calculate the delay margin as

$$d_{\mathrm{m}} = \frac{\theta}{\omega_p} = \frac{0.6150}{19.58} = 0.0314\,(\mathrm{s}).$$

This means that the associated delay, which can be added to the system before it becomes unstable, is 0.0314 s.

Tutorial 2.1 *This tutorial is to illustrate how to produce a Nyquist diagram using MATLAB functions. The open-loop transfer function used in the illustration is given by (2.6).*

Step by Step

1. *Create a new file called NyquistPlot.m.*
2. *We produce the transfer function $M(s)$ using the three first order transfer functions with time delay. Enter the following program into the file:*

```
num1=3187.4;
den1=[1 3355.2];
num2=594.9;
den2=[1 1.956];
num3=[0.0271 0.2737];
den3=[1 0];
delay=0.03;
```

3. *Define the frequency vector and calculate the frequency responses of the three first order systems. Enter the following program into the file:*

```
w=logspace(-1,3,1000);
G1=freqs(num1,den1,w);
G2=freqs(num2,den2,w);
G3=freqs(num3,den3,w);
G12=G1.*G2;
G123=G12.*G3;
```

4. *Add the component of time delay to the frequency response. Enter the following program into the file:*

```
j=sqrt(-1);
Ny=G123.*exp(-j*w*delay);
```

5. *Present the two dimensional diagram with a unit circle that is centered at the origin of the complex plane. Enter the following program into the file:*

```
plot(real(Ny),imag(Ny),'b','linewidth',2)
hold on
th=[0:2*pi/60:(60)*2*pi/60];
R=exp(j*th);
plot(R)
axis([-4 4 -4 4])
xlabel('Real')
ylabel('Imag')
```

2.3.2 Rework of Tuning Rules based PID Controllers

Example 2.3 *We examine the PI controllers presented in Example 1.7, where the continuous time plant has the transfer function:*

$$G(s) = \frac{0.5e^{-20s}}{(30s+1)^3}. \tag{2.7}$$

Three sets of PI controller parameters were obtained by using the tuning rules of Ziegler–Nichols, Cohen–Coon and Wang–Cluett, as shown in Table 1.8. In this example, we analyze closed-loop stability for the three systems using Nyquist stability criterion, and modify the controller parameters to achieve closed-loop stability.

Solution. *With the PI controller parameters listed in Table 1.8, the loop frequency response $M(j\omega)$ is computed and the Nyquist diagrams for the three PI controlled system are compared in Figure 2.4. It is seen from their Nyquist diagrams that the first two tuning rules produced the Nyquist loci encircling the $(-1,0)$ point on the complex plane, hence they will lead to unstable closed-loop systems, which was confirmed by the Simulink simulation in Example 1.7. However, the Nyquist loci from the third tuning rule did not encircle the $(-1,0)$ point, hence it produced a stable closed-loop system, which was also confirmed through simulation in Example 1.7. Using the MATLAB ginput function with the cross hair we can identify the gain margin for the third Nyquist loci as*

$$GM = \frac{1}{k_g} = \frac{1}{0.8203} = 1.2191$$

and the phase margin as

$$\theta = \tan^{-1}\left(\frac{-0.3068}{-0.9309}\right) = 0.3184\,(rad)$$

The information derived from the Nyquist diagrams not only determines whether the closed-loop system is stable or not, but also provides us with an intuitive means to modify the controller parameters in order to achieve closed-loop stability. If we take 50% of the proportional controller gains from the first two tuning rules (i.e. $K_c = 3.2$ and $K_c = 3.26$ respectively), then all three PI controllers will produce a stable closed-loop system because all Nyquist plots will no longer encircle the $(-1,0)$ point on the complex plane

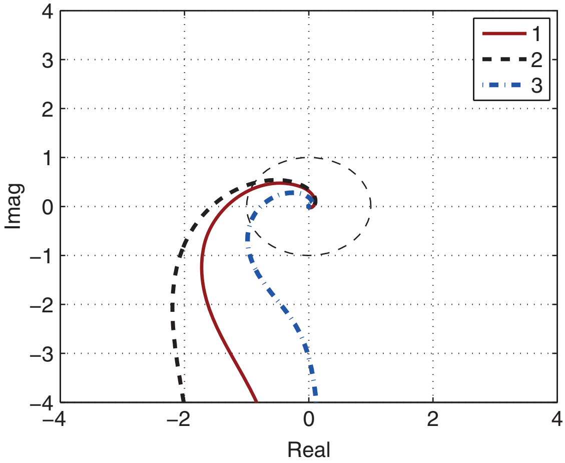


Figure 2.4 Nyquist plots with a unit circle (Example 2.3). Key: line (1) $M(j\omega)$ using Ziegler–Nichols tuning rules; line (2) Cohen–Coon tuning rules; and line (3) using Wang–Cluett tuning rules.

(see Figure 2.5). Table 2.3 summarizes the PI controller parameters and the associated gain margin, phase margin, and delay margin for this example, where Figure 2.6 is used to determine the delay margin.

Finally, the unit closed-loop step responses for the three PI control systems are computed using $\Delta t = 1$ (s). Figure 2.7 compares the three closed-loop step responses. In this set of closed-loop simulation studies, the proportional control is implemented on the output only.

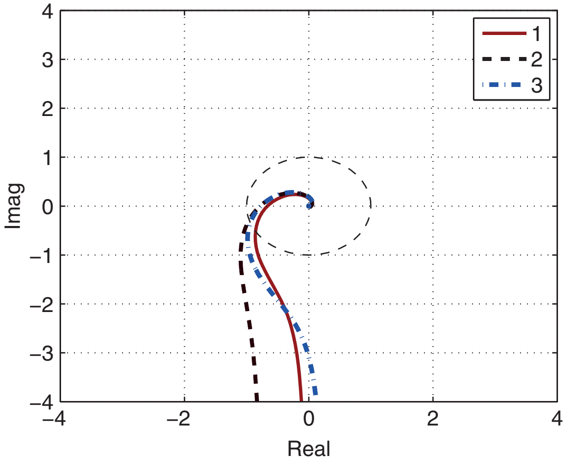


Figure 2.5 Nyquist plots for the modified controller with a unit circle (Example 2.3). Key: line (1) $M(j\omega)$ using Ziegler–Nichols tuning rules; line (2) Cohen–Coon tuning rules; and line (3) using Wang–Cluett tuning rules.

Table 2.3 Modified PI controller parameters with gain margin, phase margin, and delay margin.

	K_c	τ_I	Gain margin	Phase margin	Delay margin
Ziegler–Nichols	3.2	108	1.4914	0.5305	23.5778
Cohen–Coon	3.26	75.9231	1.258	0.2879	11.9461
Wang–Cluett	3.8867	127.2154	1.2191	0.3184	12.585

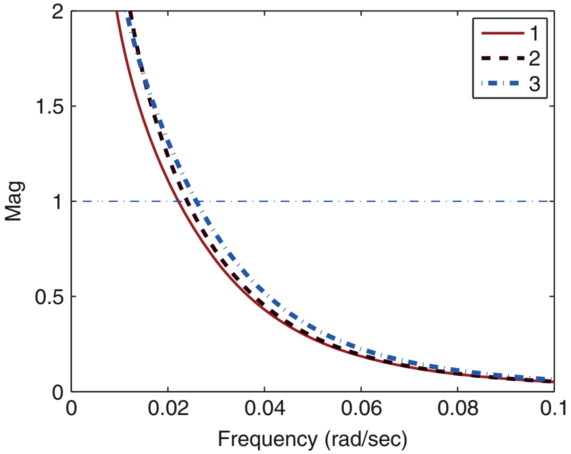


Figure 2.6 Magnitude of $M(j\omega)$ (solid line) together with dashed line to determine ω_p. Key: line (1) $|M(j\omega)|$ using Ziegler–Nichols tuning rules with reduced K_c; line (2) Cohen–Coon tuning rules with reduced K_c; and line (3) using Wang–Cluett tuning rules.

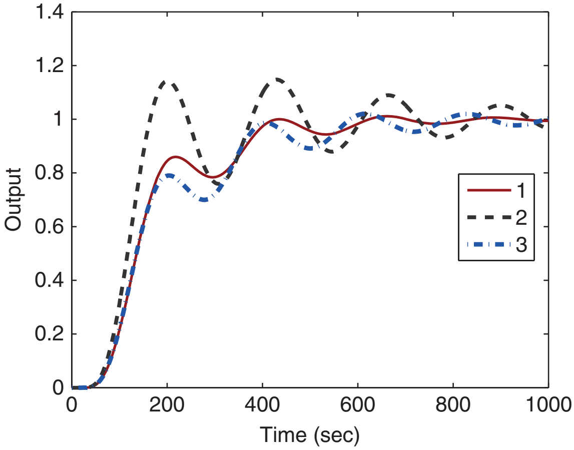


Figure 2.7 Comparison of closed-loop step responses (Example 2.3). Key: line (1) Ziegler–Nichols tuning rules with reduced K_c; line (2) Cohen–Coon tuning rules with reduced K_c; and line (3) using Wang–Cluett tuning rules.

Overall, with the Nyquist diagrams and the closed-loop step response evaluations, the Ziegler–Nichols tuning rules with the reduced K_c produced a slightly better control system. However, the Wang–Cluett tuning rules were applied as they were derived and not modified.

2.3.3 Food for Thought

1. What would happen to a closed-loop system if its gain margin is equal to 1?
2. Would you say that the Nyquist diagram is an effective means to check the performance of the PID controller designed when using the tuning rules?
3. What would you do if the gain margin of the control system designed is too small?
4. How would you be able to increase the phase margin of a control system?

2.4 Control System Structures and Sensitivity Functions

This section introduces control system structures with one and two degrees of freedom, which are related to the PID controller realization to reduce the overshoot in reference response discussed in Chapter 1. The external signals such as reference signal, disturbance signals and measurement noise are important in the analysis of control system performance. This section discusses these signals in relationships with the sensitivity functions.

2.4.1 One Degree of Freedom Control System Structure

A feedback control system of a one degree of freedom structure is represented by the block diagram, shown in Figure 2.8, where $R(s)$ is the reference signal, $Y(s)$ is the output, and $U(s)$ is the control signal. There is also an output disturbance in the system, denoted as $D_o(s)$. For simplicity, the transfer function $G(s)$ includes the plant dynamics, actuator dynamics, and sensor dynamics.

The Laplace transform of the error signal $E(s)$ is expressed as

$$\begin{aligned} E(s) &= R(s) - Y(s) = R(s) - G(s)U(s) - D_o(s) \\ &= R(s) - G(s)C(s)E(s) - D_o(s). \end{aligned} \tag{2.8}$$

Therefore, the error signal is expressed as

$$E(s) = \frac{R(s)}{1 + G(s)C(s)} - \frac{D_o(s)}{1 + G(s)C(s)}. \tag{2.9}$$

Then, the output of the control system is

$$\begin{aligned} Y(s) = R(s) - E(s) &= \left(1 - \frac{1}{1 + G(s)C(s)}\right) R(s) + \frac{D_o(s)}{1 + G(s)C(s)} \\ &= \frac{G(s)C(s)}{1 + G(s)C(s)} R(s) + \frac{D_o(s)}{1 + G(s)C(s)} \end{aligned} \tag{2.10}$$

and the control signal is

$$U(s) = C(s)E(s) = \frac{C(s)}{1 + G(s)C(s)} R(s) - \frac{C(s)}{1 + G(s)C(s)} D_o(s). \tag{2.11}$$

Assuming that $D_o(s)$ is zero, the transfer function between the reference signal and the plant output is

$$\frac{Y(s)}{R(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{2.12}$$

and the transfer function between the reference signal and the control signal is

$$\frac{U(s)}{R(s)} = \frac{C(s)}{1 + G(s)C(s)}. \tag{2.13}$$

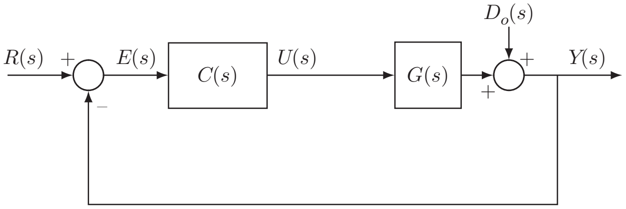


Figure 2.8 One degree of freedom control system structure.

Similarly, by assuming $R(s) = 0$, we derive the transfer functions between the output disturbance and the output, and the output disturbance and the control signal:

$$\frac{Y(s)}{D_o(s)} = \frac{1}{1 + G(s)C(s)} \tag{2.14}$$

$$\frac{U(s)}{D_o(s)} = -\frac{C(s)}{1 + G(s)C(s)}. \tag{2.15}$$

The properties of these closed-loop transfer functions are directly related to the closed-loop performance, determining the behavior of the output signal and the control signal in relation to reference signal $R(s)$ and the output disturbance $D_o(s)$. In this controller structure, once the controller $C(s)$ is selected, all four closed-loop transfer functions are fixed; only one degree of freedom is available to influence the output response $Y(s)$ to the reference signal $R(s)$ and to the disturbance $D_o(s)$. This is called one degree of freedom design.

2.4.2 Two Degrees of Freedom Design

A two degrees of freedom control system is shown in Figure 2.9. In this structure, an extra component $H(s)$ is placed after the reference signal $R(s)$, which will be used in the design. $D_o(s)$ denotes the output disturbance, D_i denotes the input disturbance, $D_m(s)$ denotes the measurement noise. How does this structure offer two degrees of freedom in the design? For this, with the assumption that $D_i(s) = 0$ and $D_m(s) = 0$, we calculate the output response $Y(s)$ in relation to the reference signal $R(s)$ and the output disturbance $D_o(s)$,

$$Y(s) = \frac{G(s)C(s)H(s)}{1 + G(s)C(s)}R(s) + \frac{D_o(s)}{1 + G(s)C(s)}. \tag{2.16}$$

From this, we have the two transfer functions

$$\frac{Y(s)}{R(s)} = \frac{G(s)C(s)H(s)}{1 + G(s)C(s)} \tag{2.17}$$

$$\frac{Y(s)}{D_o(s)} = \frac{1}{1 + G(s)C(s)}. \tag{2.18}$$

Transfer function $H(s)$ provides one more degree of freedom to shape the output response to the reference signal $R(s)$. This extra degree of freedom plus the original one degree of freedom gives the two degrees of freedom in the design. If the control system is configured as a, then we can shape, independently, the output response to the reference signal and to the disturbance.

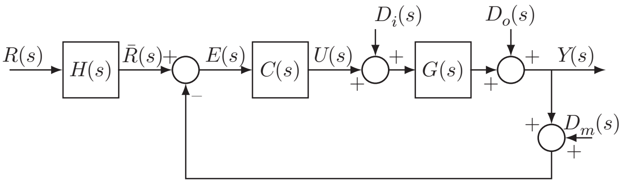


Figure 2.9 Two degrees of freedom control system structure.

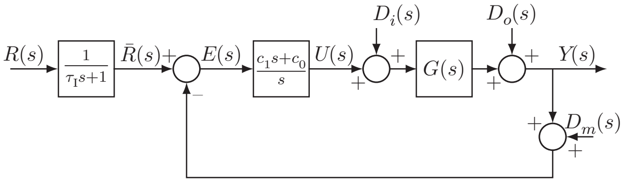


Figure 2.10 Two degrees of freedom PI control system structure, where $K_c = c_1$ and $\tau_I = \frac{c_1}{c_0}$.

2.4.2.1 Two degrees of freedom implementation of PI controllers

From Example 1.3, the IP controller structure illustrated in Figure 1.7 is in fact a two degrees of freedom implementation of a PI controller with a reference filter $H(s)$, where the reference filter is $H(s) = \frac{1}{\tau_I s+1}$, which is illustrated in Figure 2.10. One may argue that the choices of proportional controller gain K_c and the integral time constant τ_I have cemented the characteristics of disturbance rejection; however, the value of τ_I has provided an extra degree of freedom to influence reference response. The advantage of using the IP controller is that the implementation procedure is simpler, because there is no need for the implementation of reference signal filter. In the general framework of two degrees of freedom controller implementation, $H(s)$ may also be designed carefully to achieve desired effect of reference response.

2.4.3 Sensitivity Functions in Feedback Control

To understand the sensitivity functions and their roles in feedback control, we examine the block diagram of a closed-loop feedback control system illustrated in Figure 2.9.

Based on the block diagram, we calculate the feedback error of the closed-loop system firstly as

$$\begin{aligned}E(s) &= H(s)R(s) - (Y(s) + D_m(s))\\ &= H(s)R(s) - [G(s)(U(s) + D_i(s)) + D_o(s) + D_m(s)]\\ &= H(s)R(s) - G(s)C(s)E(s) - G(s)D_i(s) - D_o(s) - D_m(s).\end{aligned} \tag{2.19}$$

By re-arranging (2.19), the closed-loop feedback error is

$$E(s) = \frac{H(s)}{1+G(s)C(s)}R(s) - \frac{G(s)}{1+G(s)C(s)}D_i(s) - \frac{D_o(s)}{1+G(s)C(s)} - \frac{D_m(s)}{1+G(s)C(s)}. \tag{2.20}$$

Note that the feedback error is relation to the output via,

$$E(s) = H(s)R(s) - D_m(s) - Y(s). \tag{2.21}$$

By substituting (2.21) into (2.20), we obtain the expression of the closed-loop output $Y(s)$ as

$$\begin{aligned}Y(s) = &\frac{G(s)C(s)H(s)}{1+G(s)C(s)}R(s) + \frac{D_o(s)}{1+G(s)C(s)} + \frac{G(s)}{1+G(s)C(s)}D_i(s)\\ &- \frac{G(s)C(s)}{1+G(s)C(s)}D_m(s).\end{aligned} \tag{2.22}$$

Also, from the feedback error (2.20), we calculate the closed-loop control signal as

$$\begin{aligned} U(s) = C(s)E(s) = \frac{C(s)H(s)}{1+G(s)C(s)}R(s) - \frac{C(s)G(s)}{1+G(s)C(s)}D_{\mathrm{i}}(s) \\ - \frac{C(s)}{1+G(s)C(s)}D_o(s) - \frac{C(s)}{1+G(s)C(s)}D_{\mathrm{m}}(s). \end{aligned} \tag{2.23}$$

Based on these relationships, the following sensitivity functions are defined.

- Sensitivity function:

$$S(s) = \frac{1}{1+G(s)C(s)}.$$

- Complementary sensitivity function:

$$T(s) = \frac{G(s)C(s)}{1+G(s)C(s)}.$$

- Input disturbance sensitivity function:

$$S_{\mathrm{i}}(s) = \frac{G(s)}{1+G(s)C(s)}.$$

- Control sensitivity function:

$$S_{\mathrm{u}}(s) = \frac{C(s)}{1+G(s)C(s)}.$$

The sensitivity functions are related to each other in the following ways.

- The sensitivity plus complementary sensitivity is equal to one:

$$S(s) + T(s) = \frac{1}{1+G(s)C(s)} + \frac{G(s)C(s)}{1+G(s)C(s)} = 1. \tag{2.24}$$

- The input disturbance sensitivity is related to sensitivity:

$$S_{\mathrm{i}}(s) = \frac{G(s)}{1+G(s)C(s)} = S(s)G(s). \tag{2.25}$$

- The control sensitivity is related to sensitivity:

$$S_{\mathrm{u}}(s) = \frac{C(s)}{1+G(s)C(s)} = S(s)C(s). \tag{2.26}$$

With the sensitivity functions, we re-write the output of the closed-loop system (2.22) as

$$\begin{aligned} Y(s) &= H(s)T(s)R(s) + +S(s)D_{\mathrm{o}}(s) + S_{\mathrm{i}}(s)D_i(s) - T(s)D_{\mathrm{m}}(s) \\ &= H(s)T(s)R(s) + S(s)(D_{\mathrm{o}}(s) + G(s)D_{\mathrm{i}}(s)) - T(s)D_{\mathrm{m}}(s) \end{aligned} \tag{2.27}$$

and the control signal (2.23) as

$$U(s) = H(s)S_{\mathrm{u}}(s)R(s) - S_{\mathrm{u}}(s)D_{\mathrm{o}}(s) - S_{\mathrm{u}}(s)G(s)D_{\mathrm{i}}(s) - D_{\mathrm{m}}(s). \tag{2.28}$$

From these relationships, we can see that:

1. The complementary sensitivity function $T(s)$ represents the effect of both reference signal and measurement noise on the output.
2. The sensitivity $S(s)$ represents the effect of output disturbance on the output.
3. The input sensitivity $S_i(s)$ represents the effect of input disturbance on the output.

2.4.4 Food for Thought

1. In the two-degrees-of-freedom control system structure, must the reference filter $H(s)$ be stable?
2. In many applications, instead of designing the reference filter, it is simpler to design the reference signal by deploying a ramp signal between two constant references. Assuming that at time $t_0 = 0$, the reference signal $r(t_0) = 1$, we would like the reference signal to be increased to 3 in a duration of 10 seconds. What does this reference signal look on a piece of paper? If you would like to duplicate this signal using the reference filter $H(s)$, what would you do?
3. Can you think of a reason why we would use the IP controller instead of using the two-degrees-of- freedom approach when we need to overcome the reference overshoot problem in PID control systems?
4. If the IP controller you designed still resulted in overshoot in reference response, what would you do?
5. With respect to PID control, by examining the closed-loop transfer function (1.40) in Example 1.4, which reference filter would you use to reduce the overshoot in the closed-loop step response caused by the derivation control?

2.5 Reference Following and Disturbance Rejection

Aside from stabilization of unstable systems, the two most important purposes that a feedback control system serves are reference following and disturbance rejection. In this section, we will discuss these two topics in relation to the complementary sensitivity $T(s)$ and the sensitivity function $S(s)$. Their effects on the output will be measured and analyzed in the frequency domain. One caution is that closed-loop stability is a pre-requisite for the sensitivity analysis to be valid.

2.5.1 Closed-loop Bandwidth

For simplicity, we assume that $H(s) = 1$ to begin the discussion. From (2.27), the complementary sensitivity $T(s)$ represents the effect of reference following on the output and the sensitivity $S(s)$ represents the effect of disturbance rejection. Intuitively, for a control system designed with good reference following properties, we would like to see the complementary sensitivity $|T(j\omega)| = 1$ so that the output $|Y(j\omega)| = |R(j\omega)|$ for some designated frequencies, where $R(j\omega)$ is the frequency response of the reference signal. On the contrary, for disturbance rejection, intuitively, we would like the magnitude of the sensitivity function $|S(j\omega)| = 0$ so that the output $|Y(j\omega)| = 0$ for some frequencies contained in the output disturbance signal $D_o(j\omega)$ or the input disturbance signal $D_i(j\omega)$.

As from (2.24), the sum of $T(s)$ and $S(s)$ is equal to one. Two observations follow immediately.

1. If at a given frequency $\omega = \omega_0$, $|T(\mathrm{j}\omega)| = 1$, then $|S(\mathrm{j}\omega)| = 0$.
2. Similarly, if $|T(\mathrm{j}\omega)|$ is large, then $|S(\mathrm{j}\omega)|$ is small.

This basically says that the control objectives for reference tracking or disturbance rejection using feedback can be achieved using the same qualitative and quantitative measure of either complementary sensitivity function $T(\mathrm{j}\omega)$ or sensitivity function $S(\mathrm{j}\omega)$ because $S(\mathrm{j}\omega) + T(\mathrm{j}\omega) = 1$. The implication is that if a feedback control system has a good tracking performance for a reference signal, then it will also have a good disturbance rejection for the disturbance signals that have the identical frequency characteristics. In other words, the reference following and disturbance rejection in feedback control share the same goals in the design without conflicting.

We can also examine the reference tracking from feedback error analysis to reach the same conclusions as above. Consider a unit negative feedback control system with the reference pre-compensation $H(s) = 1$. The feedback error signal $E(s)$ is simply computed as

$$\begin{aligned} E(s) &= R(s) - Y(s) = R(s) - T(s)R(s) \\ &= (1 - T(s))R(s) = S(s)R(s). \end{aligned} \tag{2.29}$$

It is seen here that, for the reference tracking, the feedback error is directly related to the sensitivity function $S(s)$. Therefore, if $|S(\mathrm{j}\omega)|$ is small over some frequency band, then the feedback control system will yield, on the same frequency band, better reference tracking as well as disturbance rejection performance. This re-iterates that the control system design objectives for reference following and disturbance rejection share the same characteristics over either the complementary sensitivity or the sensitivity function.

One of the qualitative measures in sensitivity analysis is the characterization of closed-loop bandwidth ω_b. The parameter bandwidth ω_b is a frequency parameter either in the unit of Hertz (Hz) or rads^{-1}, which corresponds to the frequency when the complementary sensitivity function has the following value:

$$|T(\mathrm{j}\omega)| = \frac{|T(\mathrm{j}0)|}{\sqrt{2}}.$$

As illustrated in Figure 2.11, the magnitude of a complementary sensitivity function $T(\mathrm{j}\omega)$ is intersected with a dash-dotted horizontal line with a value of $\frac{1}{\sqrt{2}} \approx 0.707$, where the vertical dash-dotted line leads to the bandwidth of ω_b. As demonstrated in Section 2.3, one can easily use the MATLAB ginput.m function to identify the bandwidth from the plot of the complementary sensitivity function by putting the hair cross line at the point when the dash-dotted line intersects the magnitude of complementary sensitivity. For reference following, the bandwidth ω_b is interpreted as if the frequencies of the reference signal fall between 0 and ω_b, then the output signal will have the capacity to closely duplicate this reference signal. For disturbance rejection, because $|S(\mathrm{j}\omega)|$ is small between 0 and ω_b, if the frequencies of a disturbance signal fall into this range, then the output will have the capacity to suppress the disturbance.

Having said that minimization of the effect of disturbance will certainly lead to maximizing tracking performance of the same type of reference signals, in engineering applications there might be additional constraints on the reference tracking such as output

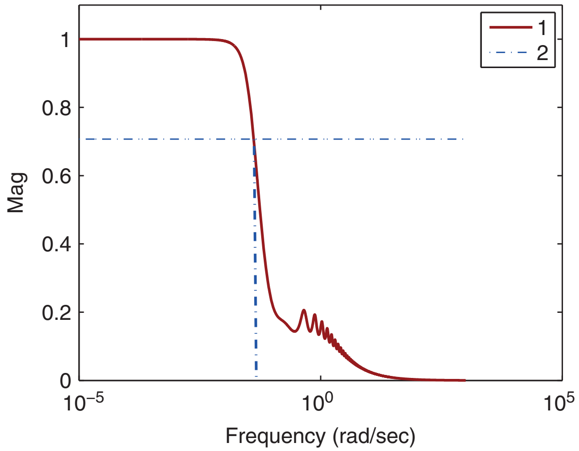


Figure 2.11 Complementary sensitivity function with bandwidth illustration. Key: solid line: $|T(j\omega)|$; dash-dotted lines: illustration of bandwidth.

overshoot of the reference signal. The two degrees of freedom control system implementation provides an additional means in an open-loop manner to shape the reference tracking properties by selecting the stable reference filter $H(s)$.

One exception to the shared common ground of disturbance rejection and reference following is the scenario when there is pole-zero cancellation in the controller structure. Pole-zero cancellation is a commonly used technique for control system design because it simplifies the controller parameter solutions. Assuming that the controller transfer function is $C(s) = \frac{P(s)}{L(s)}$ and the model transfer function is $G(s) = \frac{B(s)}{A(s)}$ from Figure 2.9, with $H(s) = 1$, the output response to the reference signal is

$$Y(s) = \frac{\frac{P(s)}{L(s)}\frac{B(s)}{A(s)}}{1 + \frac{P(s)}{L(s)}\frac{B(s)}{A(s)}} R(s) = T(s)R(s) \tag{2.30}$$

and to the input disturbance is

$$Y(s) = \frac{\frac{B(s)}{A(s)}}{1 + \frac{P(s)}{L(s)}\frac{B(s)}{A(s)}} D_{\mathrm{i}}(s) = S_{\mathrm{i}}(s)D_{\mathrm{i}}(s). \tag{2.31}$$

Note that in the complementary sensitivity function $T(s)$, $\frac{P(s)}{L(s)}$ and $\frac{B(s)}{A(s)}$ appear in pairs. Therefore, the poles or zeros that have been canceled in the design will disappear from the complementary sensitivity function, meaning that they will not affect the closed-loop performance of reference tracking. However, in the input disturbance sensitivity function $S_{\mathrm{i}}(s)$, $\frac{P(s)}{L(s)}$ and $\frac{B(s)}{A(s)}$ appear in pairs only for the denominator. Thus, if a pole from the system transfer function $G(s)$ is canceled in the controller design, it will re-appear as the same pole in the input disturbance sensitivity function because $\frac{P(s)}{L(s)}$ does not appear in the numerator. This means that if a system pole is canceled in the controller structure, a fast reference following response does not automatically imply a fast input disturbance rejection, depending on the location of the canceled pole. A detailed study of pole-zero cancellation on the effect of disturbance rejection is given in Example 3.7.

2.5.2 Reference Following and Disturbance Rejection with PID Controllers

PID controllers are the most widely used controllers in engineering applications. Their successful applications are aligned with the most commonly encountered reference signals and disturbance signals. The reference signals for PID controller applications are a step signal or a series of step signals. For instance, if we want to regulate a room temperature to 20 °C from the current temperature of 15 °C, then the reference signal is a step signal with amplitude of 5 °C. In other words, the output of a PID control system is regulated to a constant value. Note that the Laplace transform of a unit step signal is $R(s) = \frac{1}{s}$ with the magnitude of frequency response $|R(\mathrm{j}\omega)| = \frac{1}{|\omega|}$, which is infinity at $\omega = 0$. Hence, for the output to follow a step reference signal perfectly, the complementary sensitivity function $T(\mathrm{j}0) = 1$ at the zero frequency. This automatically implies that the sensitivity function $S(\mathrm{j}0) = 0$, which is required for tracking the step reference signal and rejecting a disturbance with its frequency contents concentrated at the zero frequency.

Now, on a back envelope calculation, if the plant model does not contain a zero at $s = 0$, the integrator contained in a PID controller will simultaneously yield $T(\mathrm{j}0) = 1$ and $S(\mathrm{j}0) = 0$ because the integrator in the controller creates an infinite gain for the loop transfer function $|C(\mathrm{j}\omega)G(\mathrm{j}\omega)|$ at $\omega = 0$. The sensitivity function $S(s)$ contains a zero at $s = 0$ as long as the plant model $G(s)$ does not contain a zero at $s = 0$.

Because the majority of the control systems are designed to track a constant reference signal or regulate the output to a constant value, additionally, the most disturbances occurred in the process control applications are rich in the low frequency region, the PID controllers are adequate for the commonly encountered applications.

Example 2.4 *Consider the first order plus time delay system described by the transfer function*

$$G(s) = \frac{0.8\mathrm{e}^{-20s}}{10s + 1}.$$

Use the Padula and Visioli tuning rules (Padula and Visioli (2011)) to determine the PID controller parameters with consideration of two cases: $M_\mathrm{s} = 1.4$ *and* $M_\mathrm{s} = 2$*, and analyze their properties for reference following and disturbance rejection using the complementary sensitivity* $T(\mathrm{j}\omega)$ *and sensitivity functions* $S(\mathrm{j}\omega)$*.*

Solution. *Using the Padula and Visioli tuning rules presented in Table 1.5, we find two sets of PID controller parameters. For* $M_\mathrm{s} = 1.4$*, we have*

$$K_\mathrm{c} = 0.2959; \quad \tau_\mathrm{I} = 9.9063; \quad \tau_\mathrm{D} = 7.1749$$

and for $M_\mathrm{s} = 2$*, we have*

$$K_\mathrm{c} = 0.5207; \quad \tau_\mathrm{I} = 10.4447; \tau_\mathrm{D} = 8.0639.$$

Because there is a derivative filter used in the PID controller, the controller transfer function $C(s)$ *becomes*

$$C(s) = K_\mathrm{c}(1 + \frac{1}{\tau_\mathrm{I} s} + \frac{\tau_\mathrm{D} s}{0.1\tau_\mathrm{D} s + 1})$$

where the derivative filter time constant is taken as $0.1\tau_\mathrm{D}$.

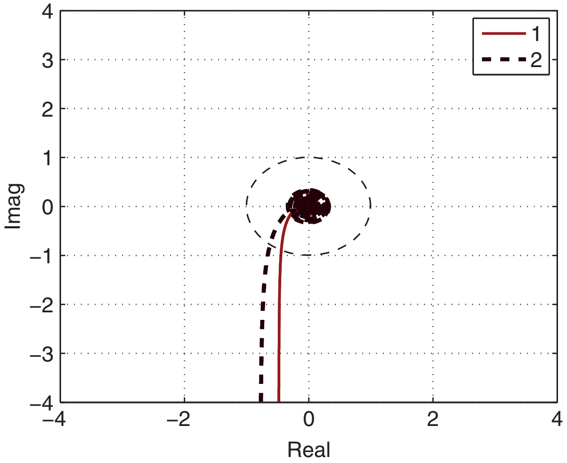


Figure 2.12 Nyquist diagrams using Padula and Visioli PID controller (Example 2.4). Key: line (1) $M_s = 1.4$; line (2) $M_s = 2$.

Firstly, we verify closed-loop stability of the two PID controlled systems by plotting their Nyquist diagrams shown in Figure 2.12. It is seen that the PID controllers produce sufficient gain margins and phase margins.

Next, we present the magnitudes of the complementary sensitivity functions in Figure 2.13 with illustration of the bandwidth. They are equal to 0.0403 rads^{-1} *and* 0.0657 rads^{-1}*, respectively. The bandwidth of the PID controller tuned using the* $M_s = 2$ *is larger among the two cases. The magnitudes of the sensitivity functions are presented in Figure 2.14. It is interesting to note that the sensitivity peak for the* $M_s = 1.4$ *case is close to the specification. However, for the* $M_s = 2$ *case, the sensitivity peak is less than the specified value of 2, but it is higher than that obtained in the previous case.*

To simulate the closed-loop response for reference following and disturbance rejection, we use the two degrees of freedom PID implementation to avoid overshoot in output when the reference signal is a unit step, where both the proportional control and the derivative control are implemented on the output only. By selecting the sampling interval $\Delta t = 1$ *s,*

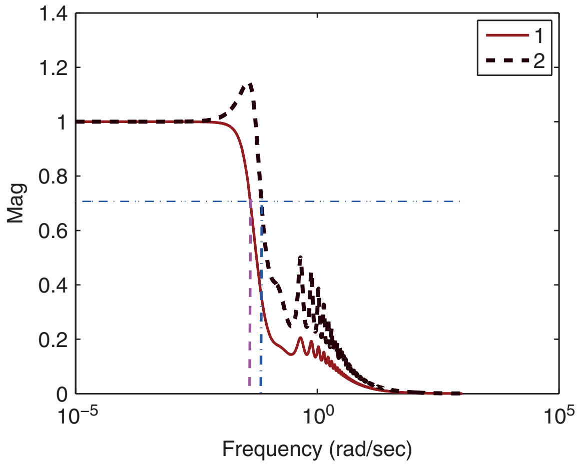


Figure 2.13 Complementary sensitivity function using the Padula and Visioli PID controller with bandwidth illustration. Key: line (1) $M_s = 1.4$; line (2) $M_s = 2$.

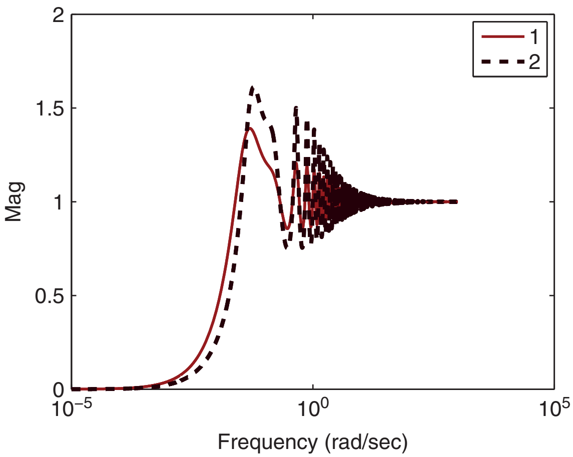


Figure 2.14 Sensitivity function using the Padula and Visioli PID controller. Key: line (1) $M_s = 1.4$; line (2) $M_s = 2$.

the closed-loop simulation is performed with a unit step signal for the reference and a step input disturbance with unknown amplitude entering the system at half of the simulation time. Figure 2.15(a) shows the closed-loop control signals for the reference following and disturbance rejection. It is seen that for the tuning rule with $M_s = 2$, the control amplitude variations are larger. Figure 2.15(b) shows the closed-loop output response. It confirms that the output from the tuning rule with $M_s = 2$ is faster among the two responses. The time domain performances are consistent with the frequency response analysis that the PID control system with the wider bandwidth produces a faster reference tracking as well as disturbance rejection. It is worthwhile mentioning that the two degrees of freedom implementation is crucial to reduce the overshoot for the reference change when using the tuning rules produced by Padula and Visioli. As an exercise, one is encouraged to implement the one degree of freedom PID controller and observe the substantial amount of overshoot occurring in the closed-loop step response for the reference change.

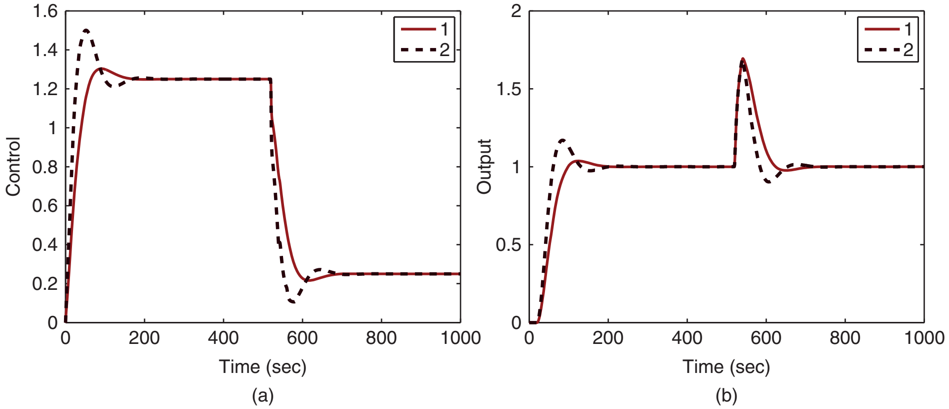


Figure 2.15 Comparison of closed-loop responses using Padula and Visioli PID controller (Example 2.4). (a) Control signal. (b) Output. Key: line (1) tuning rule with $M_s = 1.4$; line (2) tuning rule with $M_s = 2$.

2.5.3 Reference Following and Disturbance Rejection with Resonant Controllers

In the applications of control systems to power electronics, aerospace, and mechanical engineering, it is often required for the output of the closed-loop control system to track sinusoidal reference signal or to reject a sinusoidal disturbance. In these applications, we assume that the sinusoidal reference signal is $r(t) = A_{\mathrm{m}} \sin(\omega_0 t)$ with known parameters. However, for disturbance rejection, the disturbance signal is expressed as $d(t) = d_{\mathrm{m}} \sin(\omega_0 t)$, where the frequency ω_0 is known but the amplitude of the disturbance d_{m} is unknown.

Note that the Laplace transform of the reference sinusoidal signal with frequency ω_0 is

$$R(s) = \frac{A_{\mathrm{m}}\omega_0}{s^2 + \omega_0^2}$$

with frequency response

$$R(\mathrm{j}\omega) = \frac{A_{\mathrm{m}}\omega_0}{(\mathrm{j}\omega)^2 + \omega_0^2} = \frac{A_{\mathrm{m}}\omega_0}{-\omega^2 + \omega_0^2}$$

Thus, as $\omega \to \pm\omega_0$, $|R(\mathrm{j}\omega)| \to \infty$.

From the sensitivity analysis, in order for the feedback control system to have a good tracking performance for the sinusoidal reference signal, we need the complementary sensitivity function $|T(\mathrm{j}\omega)| = 1$ at $\omega = \omega_0$. Similarly, to reject the sinusoidal disturbance signal with unknown amplitude, we need to have the sensitivity function $|S(\mathrm{j}\omega)| = 0$ at $\omega = \omega_0$. In order to achieve these characteristics for the feedback control system, a quick calculation indicates that the feedback controller is required to embed the mode $\frac{1}{s^2+\omega_0^2}$ into its structure, assuming that the plant does not contain a pair of complex zeros at $\pm \mathrm{j}\omega_0$. In the literature, this type of controller is called either resonant controller or repetitive controller because of the periodic nature of the external signals.

We consider the case that $r(t)$ is a multi-frequency periodic signal, defined as $r(t) = A_0 \sin\omega_0 t + A_1 \sin\omega_1 t + \ldots + A_k \sin\omega_k t$ where the frequencies ω_1, ω_2, …, and ω_k are given. In order to track this multi-frequency periodic signal using a feedback controller, the complementary sensitivity function is required to satisfy $T(\mathrm{j}\omega) = 1$ at the frequencies $\omega = \omega_0, \omega_1, \ldots, \omega_k$. As a result, the sensitivity function automatically satisfies $S(\mathrm{j}\omega) = 0$ at the frequencies $\omega = \omega_0, \omega_1, \ldots, \omega_k$. Therefore, the controller designed for following the multi-frequency periodic reference signal will also reject a periodic disturbance with the same frequencies. The controller is required to embed the following components

$$\frac{1}{(s^2 + \omega_0^2)(s^2 + \omega_1^2)\ldots(s^2 + \omega_k^2)}$$

into its structure in order to achieve $T(\mathrm{j}\omega) = 1$ or $S(\mathrm{j}\omega) = 0$ at the frequencies $\omega = \omega_0, \omega_1, \ldots, \omega_k$. Here we need to assume that the plant does not have complex zeros at the corresponding frequencies.

In Section 3.5, pole-assignment controller design techniques will be introduced for the design of resonant controller. In Section 5.4, a resonant controller will be designed using a disturbance estimation technique together with anti-windup implementation. This disturbance estimation based design technique is extended to a multi-frequency sinusoidal reference/disturbance signal in Section 5.5.

2.5.4 Food for Thought

1. If the disturbance rejection is too slow, taking a long time for the output response to recover, would you decrease the closed-loop bandwidth?
2. For a PID controlled system with K_c, τ_I and τ_D, if the bandwidth ω_b were too small, would you increase K_c?
3. For the IMC-PID tuning rules (see Section 1.4.1), the original tuning-rules (Equation (1.47)) will lead to a slow response to input disturbance. Can you explain this using the characteristics of the input sensitivity function?
4. Why is the two-degrees-of-freedom PID controller implementation important in the context of reference following and disturbance rejection?
5. In many applications, rejecting load disturbance is the primary objective of the control systems. Load disturbance is often modelled as a constant input disturbance. To avoid a large disruption to the system when switching on a load is paramount. How can we design such a load profile so that it can be switched gradually?

2.6 Disturbance Rejection and Noise Attenuation

Both noise and disturbance co-exist in a physical system. A good closed-loop performance requires minimization of the effects of both disturbance and noise.

2.6.1 Conflict between Disturbance Rejection and Noise Attenuation

For minimization of the effects of both input and output disturbances, we make the magnitude of the output in frequency response

$$\begin{aligned} |Y_\mathrm{d}(\mathrm{j}\omega)| &= |S(\mathrm{j}\omega)(D_\mathrm{o}(\mathrm{j}\omega) + G(\mathrm{j}\omega)D_\mathrm{i}(\mathrm{j}\omega))| \\ &= |S(\mathrm{j}\omega)||(D_\mathrm{o}(\mathrm{j}\omega) + G(\mathrm{j}\omega)D_\mathrm{i}(\mathrm{j}\omega))| \end{aligned} \tag{2.32}$$

as small as possible. For minimization of the measurement noise, we make the magnitude of the output in frequency response

$$|Y_\mathrm{m}(\mathrm{j}\omega)| = |T(\mathrm{j}\omega)D_\mathrm{m}(\mathrm{j}\omega)| = |T(\mathrm{j}\omega)||D_\mathrm{m}(\mathrm{j}\omega)| \tag{2.33}$$

as small as possible, indicating that there is a conflict between disturbance rejection and noise attenuation.

We cannot alter the disturbances and noise because they already existed in the system. Thus, what we do is to make:

- the magnitude of sensitivity $S(\mathrm{j}\omega)$ ($|S(\mathrm{j}\omega)|$) small for disturbance rejection;
- the magnitude of complementary sensitivity $T(\mathrm{j}\omega)$ ($|T(\mathrm{j}\omega)|$) small for noise attenuation.

These are the basic design principles for control systems. However, noting that the relationship between the sensitivity and complementary sensitivity is constrained by

$$S(\mathrm{j}\omega) + T(\mathrm{j}\omega) = 1 \tag{2.34}$$

which says that we cannot make both $|S(\mathrm{j}\omega)|$ and $|T(\mathrm{j}\omega)|$ small over the same frequency bands. In other words, if the disturbance is minimized in a given frequency region where $|S(j\omega)|$ is small, then inevitably the measurement noise is not attenuated in the same frequency region where $|T(j\omega)|$ is large. So how are we going to design a closed-loop control system that will minimize the effects of disturbance and the measurement noise?

Note that the disturbances existing in the system correspond to slow movement of the variables or slow changes, therefore the frequency contents of the disturbance term $|D_o(j\omega) + G(j\omega)D_i(j\omega)|$ are concentrated in the low frequency region. In contrast, the measurement noise corresponds to fast movement of the variables or fast and frequent changes of the variables, therefore the frequency contents of the measurement noise $|D_m(j\omega)|$ are concentrated in the higher frequency region. This means that we can achieve disturbance rejection by choosing the sensitivity function $S(j\omega) \approx 0$ at the low frequency region, which implies $T(j\omega) \approx 1$ at the low frequency region, because $S(j\omega) + T(j\omega) = 1$. This is not too bad for noise attenuation because $|D_m(j\omega)|$ is small in the low frequency region. At the high frequency region, to avoid the amplification of measurement noise, we choose $|T(j\omega)| \approx 0$, which implies $|S(j\omega)| \approx 1$. This is not too bad for disturbance rejection because $|D_o(j\omega) + G(j\omega)D_i(j\omega)|$ is small in the high frequency region. In short, to achieve a trade-off relationship between disturbance rejection and noise attenuation, a closed-loop bandwidth ω_b should be carefully selected.

2.6.2 PID Controller for Disturbance Rejection and Noise Attenuation

Two examples are given in this section to illustrate the relationship between disturbance rejection and noise attenuation when using a PID controller.

Example 2.5 *Consider the transfer function model given in Ralhan and Badgwell (2000) to describe the operations of a gas fired heater at a low fuel operation:*

$$G_L(s) = \frac{3e^{-10s}}{(4s+1)^2}\frac{degC}{sm^3/s}. \tag{2.35}$$

This transfer function was approximated using the first order plus delay model in Section 1.5.2 as

$$G(s) = \frac{3\mathrm{e}^{-9.54s}}{13.48s+1} \tag{2.36}$$

Use the Padula and Visioli tuning rules to determine the PID controller parameters for the fired heater system, and evaluate the PID controllers' performance in disturbance rejection and noise attenuation.

Solution. *For the PID controller tuning rule that corresponds to $M_s = 1.4$, use of Table 1.5 together with the first order approximate model (2.36) leads to the following PID controller parameters:*

$$K_c = 0.1574; \tau_I = 7.2783; \tau_D = 5.1449.$$

The PID controller parameters corresponding to the sensitivity peak $M_s = 2$ are

$$K_c = 0.2524; \tau_I = 6.7547; \tau_D = 5.6731.$$

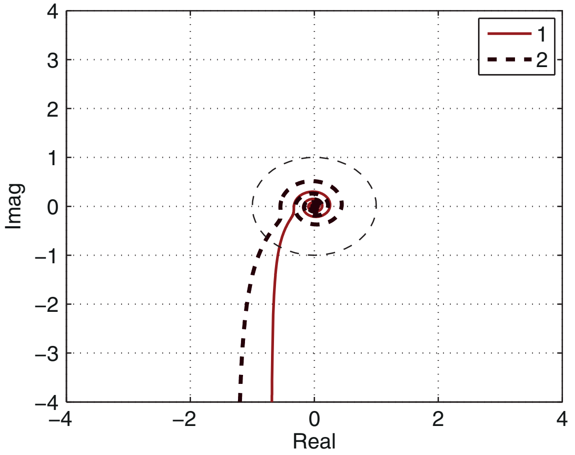


Figure 2.16 Nyquist diagrams using the Padula and Visioli PID controller (Example 2.5). Key: line (1) $M_s = 1.4$; line (2) $M_s = 2$.

We evaluate the closed-loop stability by finding the Nyquist loci of the PID controllers with the plant model (2.35) where the derivative filter time constant is selected as $0.1\tau_D$*. As shown in Figure 2.16, the* $(-1, 0)$ *point is not encircled, thus both PID control systems are stable with sufficient gain and phase margins. To evaluate the closed-loop performance for disturbance rejection, we examine the magnitude of the sensitivity functions (see Figure 2.17(a)). It is seen that* $|S(j\omega)|$ *is smaller for the design corresponding to* $M_s = 2$ *in the lower frequency region; however, it becomes very large in the medium frequency region with a peak greater than* 2*. We can conclude that the PID controller corresponding to* $M_s = 2$ *will have better disturbance suppression in the lower frequency region, however the performance will dramatically decay when the disturbance frequency increases to the medium to higher frequency region.*

To evaluate the closed-loop performance for noise attenuation, we examine the magnitude of the complementary sensitivity function (see Figure 2.17(b)). It is seen that the complementary sensitivity for the PID controller with $M_s = 2$ *has a large spike at the medium frequency region, also its magnitude is larger at the higher frequency region. It*

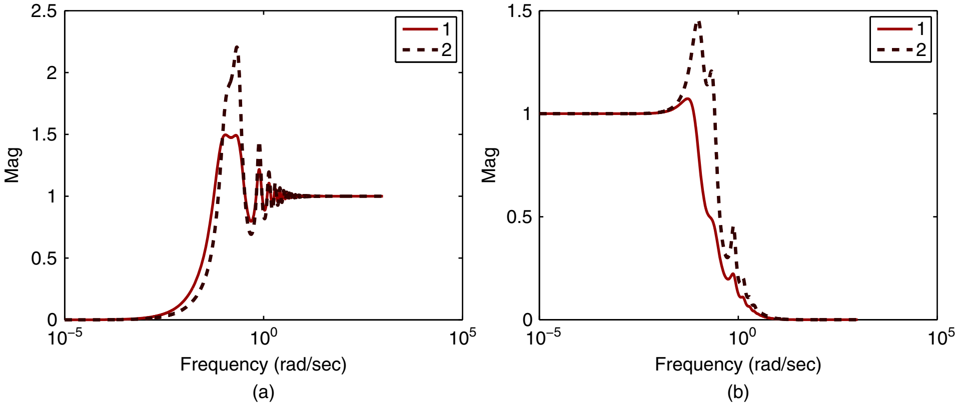


Figure 2.17 Sensitivity functions using Padula and Visioli PID controller (Example 2.5). (a) Sensitivity. (b) Complementary sensitivity. Key: line (1) $M_s = 1.4$; line (2) $M_s = 2$.

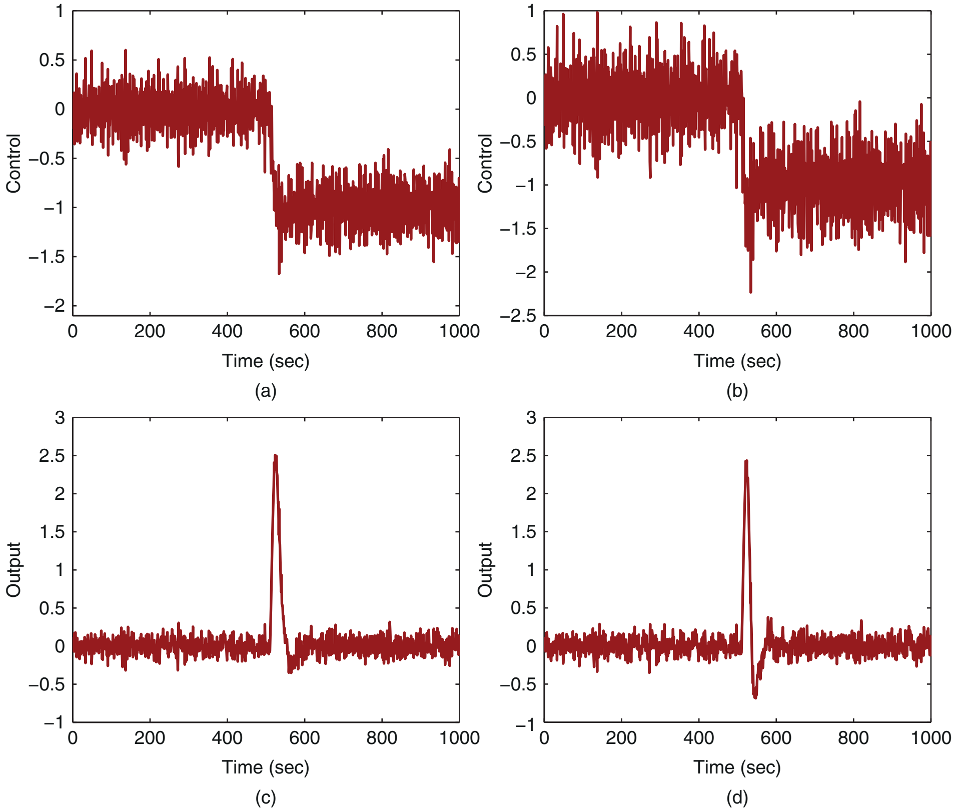


Figure 2.18 Closed-loop responses to disturbance and measurement noise using Padula and Visioli PID controller (Example 2.5). (a) Control ($M_s = 1.4$). (b) Control ($M_s = 2$). (c) Output ($M_s = 1.4$). (d) Output ($M_s = 2$).

is concluded that the PID controller with $M_s = 2$ *is more sensitive to noise. Closed-loop simulation is used to evaluate the time domain responses for disturbance rejection and noise attenuation. The sampling interval* Δt *is selected to be* 1 *s. A unit input step disturbance enters the closed-loop control system at half of the simulation time and there is a measurement noise with variance* 0.01 *added to the simulation. Figures 2.18(a) and (b) compare the closed-loop control signal responses to the disturbance and measurement noise. It is indeed seen that the control signal for the case* $M_s = 2$ *is more sensitive to the noise. By comparing the output responses (see Figures 2.18(c) and (d)), we find that the output response to the input disturbance is faster for the case that* $M_s = 2$*.*

A derivative filter plays an important role in noise attenuation for a PID controlled system because the derivative action will amplify the measurement noise. The following example is used to show its significance in the presence of measurement noise in conjunction with the choice of sampling interval in the implementation.

Example 2.6 *To emphasize that a derivative filter is paramount in the application of PID controllers, we continue from Example 2.5 by examining how the filter time constant affects the noise attenuation in the closed-loop system. In the implementation of a PID*

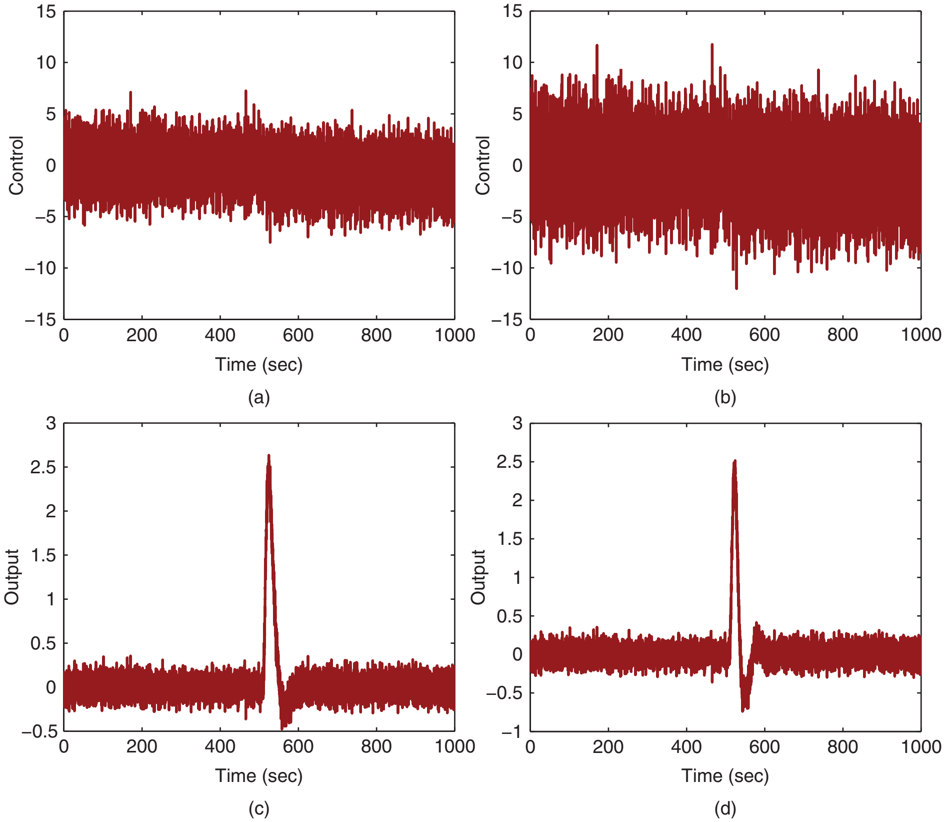


Figure 2.19 Closed-loop responses to disturbance and measurement noise using Padula and Visioli PID controller (Example 2.6). (a) Control ($M_s = 1.4$). (b) Control ($M_s = 2$). (c) Output ($M_s = 1.4$). (d) Output ($M_s = 2$).

controller, we will reduce the filter time constant from $0.1\tau_D$ to $0.01\tau_D$ and reduce the sampling interval accordingly. We leave the sensitivity function analysis as an exercise, instead we will present the simulation studies.

Solution. *With the filter time constant reduced, the sampling interval Δt needs to be reduced accordingly to avoid numerical instability in the simulation. In this simulation, Δt is reduced from 1 to 0.1. With the identical input disturbance and noise amplitude, the closed-loop response is simulated for disturbance rejection and noise attenuation. From Figures 2.19(a) and (b), it is seen that the control signal is completely distorted by the noise, although the output is less distorted. The main cause of the noise amplification is because the derivative term is particularly sensitive to the reduction of the filter time constant. Since the PID controller is designed in continuous time, a smaller sampling interval is required for a smaller time constant from the derivative filter.*

2.6.3 Food for Thought

1. Would you increase the bandwidth of the closed-loop control system if a low quality sensor is used in the measurement device and the measurement noise is severe?

2. When using a resonant controller, if the frequency ω_0 embedded into the controller is large, would you expect that the measurement noise will be amplified more than that with a PID controller?
3. In the implementation of PID controller, if the measurement noise is severe, would you decrease the derivative filter time constant?
4. Would you prefer to use a PI controller instead of a PID controller if the noise is severe?

2.7 Robust Stability and Robust Performance

Robust stability and robust performance are two important issues for a control engineer to consider when a feedback control system is designed and to be implemented. The phrase “robust control” is used to imply that the control system designed can withstand the uncertainties caused by the discrepancies between the model used for the controller design and the actual plant model.

2.7.1 Modeling Errors

Many of us, as a control engineers, have experienced a feedback control system designed using the correct methodologies and the closed-loop simulation has yielded satisfactory performance; however, the control system failed to produce stable closed-loop operation when it was implemented. More than often, the key reason behind the discrepancy between what is desired and what is reality is the existence of modeling errors between the model used for the control system design and the behavior of the actual system at specific operation conditions.

There are a few factors causing the existence of modeling errors in control system design depending on how the mathematical model is derived. In electrical engineering applications, such as electrical machine control and power converter control, the mathematical models are derived from physical laws using current and voltage (see Wang et al. (2015)). Similarly physical laws are used to derive the mathematical models for electro-mechanical systems such as unmanned aerial vehicles (see Chapter 10), and ball and plate balancing systems (see Chapter 6). The mathematical models derived using physical laws are referred mechanistical models, which are often in the form of nonlinear differential equations (see Chapter 6). The modeling errors for the electrical systems and the electro-mechanical systems are often caused by an inaccurate measurement of the physical parameters and the variations of the operating conditions.

In chemical process control applications, due to the lack of clearly defined physical laws or the complexities of the physical systems, the mathematical models are commonly obtained by directly conducting identification experiments on the plant and estimating a transfer function model based on the input and output measurement data (see the fired heater system (Ralhan and Badgwell (2000)) in Section 1.5.2). The mathematical models derived using the identification experiments (see Chapter 9) are referred empirical models. The modeling errors for the chemical process control applications are often caused by restricted identification experimental conditions including small input signal amplitude, corruptions of large measurement noise and disturbances, model estimation errors, as well as variations of operating conditions.

Furthermore, when using a restricted controller structure such as a PID controller, the mathematical models could be too complex for the design of a simple controller. Thus, model order reduction is required, which leads to another source of modeling errors. In the context of control system design, there is an unknown transfer function denoted by $G(s)^{\text{true}}$, which accurately describes the linear time invariant system at a given operating condition. However, for the various reasons discussed above, this $G(s)^{\text{true}}$ is seldom available to us. Instead, a transfer function model $G(s)$ is obtained from linearization (see Chapter 6) or from identification experiments and used in the control system design. This leads to the conceptual description of modeling errors that consist of the following forms in the literature:

$$\Delta G(s) = G(s)^{\text{true}} - G(s) \tag{2.37}$$

$$\Delta G_{\text{m}}(s) = \frac{G(s)^{\text{true}} - G(s)}{G(s)} \tag{2.38}$$

where $\Delta G(s)$ is called the additive modeling error and $\Delta G_{\text{m}}(s)$ is called the multiplicative modeling error. We assume that the modeling errors are stable.

Except that in the case of model order reduction for PID controller design, $G(s)^{\text{true}}$ is assumed unknown, thus the exact descriptions of $\Delta G(s)$ and $\Delta G_{\text{m}}(s)$ may not be available. However, in robust control, it is often that the bounds on the frequency responses are used to quantify the impact of the modeling errors, for $\omega \geq 0$,

$$|\Delta G(\text{j}\omega)| \leq \delta(\omega) \tag{2.39}$$

$$|\Delta G_{\text{m}}(\text{j}\omega)| \leq \delta_{\text{m}}(\omega) \tag{2.40}$$

In a simplified case, the modeling error bound may be chosen as a constant, which produces a conservative measure for the modeling errors.

2.7.2 Robust Stability

Robust stability for a control system is quantified and analyzed in the frequency domain in terms of the additive and multiplicative modeling errors defined, with its origin from the Nyquist stability criterion introduced in Section 2.3. The purpose is to assess closed-loop stability when the controller $C(s)$ is applied to the unknown system $G(s)^{\text{true}}$.

To start, we consider that the open-loop system $G(s)^{\text{true}}$ is stable and assume the following conditions are satisfied.

- The controller $C(s)$ is designed to stabilize the model $G(s)$ with all closed-loop poles strictly on the left half of the complex plane.
- The modeling errors, either additive or multiplicative, are stable.

Then, from the Nyquist stability criterion, the necessary and sufficient condition for closed-loop stability when the controller $C(s)$ is applied to the unknown system $G(s)^{\text{true}}$ is that the frequency response of the loop transfer function $G(\text{j}\omega)^{\text{true}}C(\text{j}\omega)$ will not encircle the $(-1, 0)$ point on the complex plane. This is translated into the following inequality:

$$|1 + G(\text{j}\omega)^{\text{true}}C(\text{j}\omega)| > 0 \tag{2.41}$$

for all $\omega \geq 0$. Because $G(\mathrm{j}\omega)^{\mathrm{true}}$ is unknown, in Equation (2.41) it is replaced by the model $G(\mathrm{j}\omega)$ and the additive modeling error $\Delta G(\mathrm{j}\omega)$, leading to

$$|1 + (G(\mathrm{j}\omega) + \Delta G(\mathrm{j}\omega))C(\mathrm{j}\omega)| > 0 \tag{2.42}$$

which is

$$\begin{aligned} 0 &< |(1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega))(1 + \frac{\Delta G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)})| \\ &\leq |(1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega))||(1 + \frac{\Delta G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)})|. \end{aligned} \tag{2.43}$$

Now, since the controller $C(s)$ is designed to stabilize the model $G(s)$, the Nyquist loci of $G(\mathrm{j}\omega)C(\mathrm{j}\omega)$ will not encircle the $(-1, 0)$ point, it ensures that

$$|1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)| > 0$$

for all $\omega \geq 0$.

Therefore, the inequality (2.41) is satisfied if

$$|1 + \frac{\Delta G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)}| > 0.$$

Or, more conservatively, for all $\omega \geq 0$,

$$1 - |\frac{\Delta G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)}| > 0.$$

This leads to the robust stability condition in the frequency domain as

$$|\frac{\Delta G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)}| < 1 \tag{2.44}$$

for all $\omega \geq 0$. This robust stability condition is a sufficient condition, which ensures the Nyquist loci of the $G(\mathrm{j}\omega)^{\mathrm{true}}C(\mathrm{j}\omega)$ not to encircle the $(-1, 0)$ point through the representation of additive modeling error.

The robust stability condition (2.44) can also be written in terms of the multiplicative modeling error ΔG_m by noting that

$$\Delta G(\mathrm{j}\omega) = \Delta G_m(\mathrm{j}\omega)G(\mathrm{j}\omega)$$

leading to

$$\begin{aligned} &|\frac{\Delta G_m(\mathrm{j}\omega)G(\mathrm{j}\omega)C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)}| \\ &= |\Delta G_m(\mathrm{j}\omega)T(\mathrm{j}\omega)| < 1. \end{aligned} \tag{2.45}$$

With the multiplicative modeling error, the robust stability condition is expressed in relation to the complementary sensitivity function $T(\mathrm{j}\omega)$.

With the frequency response bound defined as $\delta(\omega)$ in (2.39), the robust stability condition becomes:

$$|\frac{C(\mathrm{j}\omega)}{1 + G(\mathrm{j}\omega)C(\mathrm{j}\omega)}|\delta(\omega) < 1 \tag{2.46}$$

for all $\omega \geq 0$. Alternatively, with the multiplicative modeling error bound $\delta_{\mathrm{m}}(\omega)$ defined in (2.40), it has the following representation:

$$|T(\mathrm{j}\omega)|\delta_{\mathrm{m}}(\omega) < 1 \tag{2.47}$$

for all $\omega \geq 0$.

The first robust stability condition (2.46) is expressed in terms of the control sensitivity function whilst the second stability condition (2.47) is expressed in terms of the complementary sensitivity function, which is directly related to the closed-loop control performance specifications.

The robust stability condition (2.47) says that if the multiplicative modeling error $\delta_{\mathrm{m}}(\omega)$ is larger than 1 at a given frequency ω_0, then $|T(\mathrm{j}\omega_0)|$ needs to be less than 1 to ensure closed-loop stability. Conversely, if $|T(\mathrm{j}\omega)|$ is large at a certain frequency region, then a small modeling error is required in the same region to ensure closed-loop stability. Clearly, if the controller contains an integrator, then $|T(\mathrm{j}0)| = 1$, which indicates that $\delta_{\mathrm{m}}(0) < 1$ in order to guarantee closed-loop stability. Similarly, for the resonant controller that has embedded a sinusoidal mode at $\omega = \omega_0$, $|T(\mathrm{j}\omega_0)| = 1$, leading to the condition that $\delta_{\mathrm{m}}(\omega_0) < 1$ in order to guarantee robust closed-loop stability.

In summary, the existence of modeling error gives an additional constraint on the complementary sensitivity function. This constraint is translated into the selection of the desired closed-loop bandwidth as demonstrated in the following case study.

2.7.3 Case Study: Robust Control of Polymer Reactor

Example 2.7 *The mathematical model for the eighth reactor in a copolymerization reactor train is described by the following transfer function model (Madhuranthakam and Penlidis (2016)):*

$$Y(s) = \left[\frac{Ks + 1}{\tau_1^2 s^2 + 2\tau_1\tau_2 s + 1}\right]^8 U(s) \tag{2.48}$$

*where the input is the flow rate of the chain transfer agent (CTA) to the first reactor in the reactor train and the output is the weight-based average molecular weight (*MWw*). The parameters in the transfer function for the reactor are given as $K = 361.54$, $\tau_1 = 106.84$ and $\tau_2 = 1.72$.*

For this case study, we will perform the following tasks.

1. *We will find the first order plus delay approximate model using the step response data, and calculate the additive and multiplicative modeling errors.*
2. *Based on the IMC-PI controller tuning method, by choosing $\tau_{\mathrm{cl}} = d$, which is the time delay of the first order approximate model, we will calculate the PI controller parameters and check if the closed-loop system is robustly stable. As a comparison, we will increase τ_{cl} to $2d$ in order to observe its effect on the robustness of the control system.*

Solution. *With sampling interval $\Delta t = 1$ s, the unit step response of the transfer function (2.48) is calculated as shown in Figure 2.20. Clearly, it is seen that the steady-state gain is 1 and the system has a large time delay.*

By drawing a line to produce a maximum slope, which is then intersected with the steady-state line as shown in Figure 2.20, the time delay for this reactor is estimated as

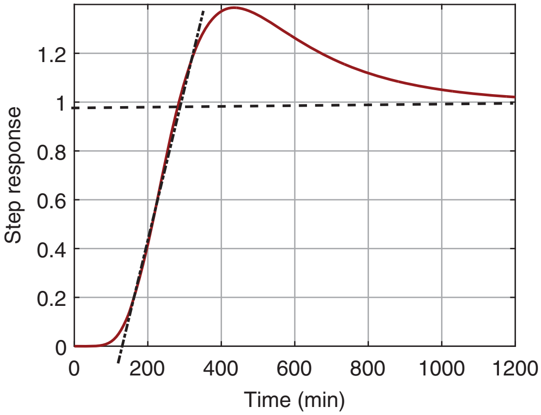


Figure 2.20 Unit step response of the eighth reactor with lines to assist obtaining first order plus delay model.

128.5 *(min) and the time constant is* 168.5 *(min). This graphic approach leads to the first order plus time delay model as*

$$G(s) = \frac{e^{-ds}}{\tau_M s + 1} = \frac{e^{-128.5s}}{168.5s + 1}. \tag{2.49}$$

The additive modeling error is calculated as

$$\Delta G(s) = \left[\frac{Ks + 1}{\tau_1^2 s^2 + 2\tau_1\tau_2 s + 1}\right]^8 - \frac{e^{-128.5s}}{168.5s + 1}$$

and the multiplicative modeling error is

$$\Delta G_m(s) = \frac{\Delta G(s)}{G(s)}.$$

Because the reactor's transfer function model has eight stable zeros, which becomes a large lead element in the dynamic response and has been neglected in the first order plus delay model, there is a large discrepancy between the system and the approximate model. This large discrepancy is reflected in the magnitudes of the frequency response of the modeling error both additive and multiplicative, as shown in Figures 2.21(a) and (b). In particular,

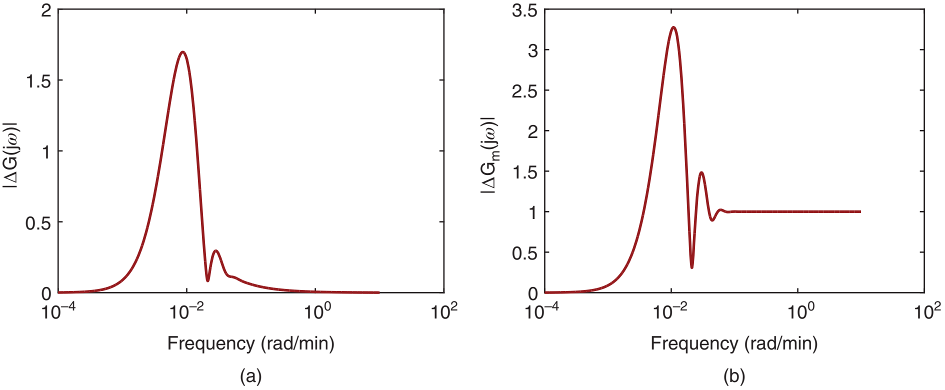


Figure 2.21 Magnitude of modeling errors with the first order plus delay model (Example 2.7). (a) Additive modeling error. (b) Multiplicative modeling error.

the magnitude of $\Delta G_m(j\omega)$ *exceeds* 3 *at* $\omega = 0.01\,\mathrm{rad\,min}^{-1}$*. This indicates that the desired bandwidth of the closed-loop control system needs to be quite small in order to guarantee closed-loop stability in the presence of a modeling error.*

For systems with a large time delay, the desired time constant is commonly selected to be larger than the time delay. We will start by choosing the desired time constant $\tau_{cl} = 128$ *(min), which leads to*

$$\begin{aligned} K_c &= \frac{1}{K_{ss}}\frac{\tau_M}{\tau_{cl} + d} = 0.7374 \\ \tau_I &= \tau_M = 168.5 \end{aligned} \tag{2.50}$$

where $K_{ss} = 1$, $\tau_M = 168.5$ *and* $d = 128.5$*. It can be checked using a Nyquist plot that this PI controller will lead to a stable closed-loop system when it is used to control the first order plus time delay model (2.49). So we check if the robust stability condition (2.47) with respect to the multiplicative modeling error is satisfied. Figure 2.22(a) shows the magnitude of the complementary sensitivity function. With the choice of* $\tau_{cl} = d$*, the magnitude of the complementary sensitivity decays rapidly (see the solid line), and as a result the robust stability condition is satisfied where the quantity* $|T(j\omega)|\delta_m(\omega)$ *is less than 1. Therefore, we can conclude that the IMC-PI controller will produce a stable closed-loop system for the polymer reactor model based on (2.48) when* $\tau_{cl} = 128$*. As a comparison, we increase the desired closed-loop time constant* τ_{cl} *to* $2d$*. Figures 2.22(a) and (b) show the further reductions of the magnitude of the complementary sensitivity function and the quantity* $|T(j\omega)|\delta_m(\omega)$*. Finally, the performance of the closed-loop control system is evaluated against the transfer function model (2.48) with the proportional control implemented on the output only. Figures 2.23(a) and (b) show the closed-loop step response of the control signal and output signal respectively. It is seen that when* $\tau_{cl} = d$*, the closed-loop responses are severely oscillatory, however when* τ_{cl} *is increased to* $2d$*, the oscillation is almost overcome. In summary, increasing the desired closed-loop time constant* τ_{cl}*, which essentially reduces the desired closed-loop bandwidth, will enhance the robustness of the control system when the plant is stable.*

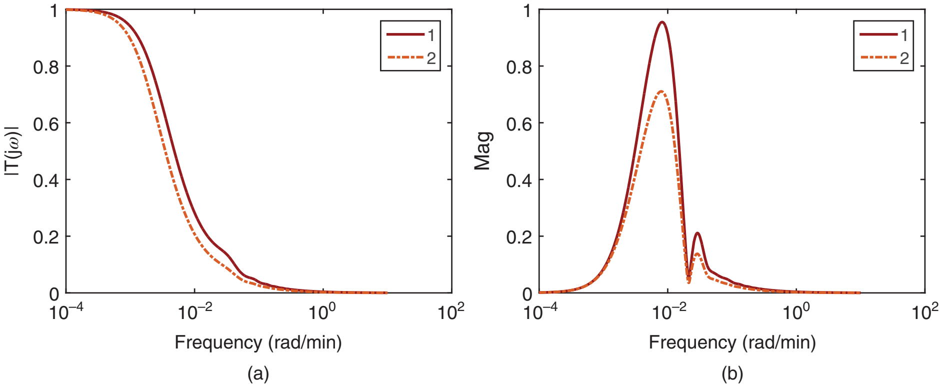


Figure 2.22 Complementary sensitivity function and graphic presentation of robust stability condition (Example 2.7). (a) Magnitude of complementary sensitivity function. (b) $|T(j\omega)|\delta_m(\omega)$. Key: line (1) $\tau_{cl} = 128$; line (2) $\tau_{cl} = 256$.

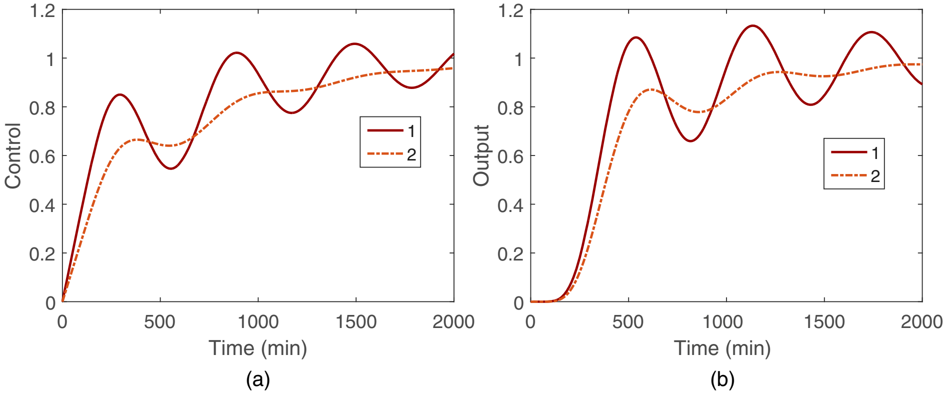


Figure 2.23 Closed-loop step responses (Example 2.7). (a) Control. (b) Output. Key: line (1) $\tau_{\rm cl} = 128$; line (2) $\tau_{\rm cl} = 256$.

This process is to be studied again as an exercise (see Problem 8.3) using the frequency response design technique introduced in Chapter 8, where closed-loop control performance improvement is expected.

Note that many of the empirical tuning rules for PID controllers do not have a performance tuning parameter to adjust the desired closed-loop performance in order to account for the modeling errors. This is the biggest limitation for the empirical PID controller tuning rules. For the IMC-PID controller, the desired closed-loop bandwidth is selected by adjusting the desired closed-loop time constant $\tau_{\rm cl}$.

2.7.4 Food for Thought

1. If we neglected a time delay d in the design, what would be the multiplicative modelling error?
2. If the Padula and Visioli PID controller tuning rules produced an unstable closed-loop system, would you try to reduce the proportional control gain K_c or increase the integral time constant τ_I?
3. Is it correct to say that the modelling error limits the closed-loop bandwidth?
4. Is it correct to say that in general we need a model with higher accuracy for a higher demand in the closed-loop performance for disturbance rejection and reference following?

2.8 Summary

In this chapter, we have presented the commonly used tools for closed-loop stability and performance analysis. Understanding the relationships between the closed-loop poles, closed-loop stability, and performance is crucial in control system design and analysis. In the later chapters, we will discuss model based controller design methods that specifically use the locations of the desired closed-loop poles as performance specification. The important aspects discussed in this chapter are:

- If the closed-loop transfer function is known with constant coefficients, then the closed-loop poles can be simply determined by finding the solutions of the characteristic equation. Numerically, this is achieved using MATLAB function root.m for the denominator of the transfer function.
- The Routh-Hurwitz stability criterion explicitly determines whether a closed-loop system is stable by examining the coefficients of the closed-loop characteristic polynomial. This is particularly useful when we need to determine the effect of the variation of a parameter on the closed-loop system.
- Nyquist stability criterion is based on the frequency response of the open-loop transfer function that includes the system, the controller, the actuator and the sensor. It involves the presentation of the frequency response graphically with real and imaginary parts (a two dimensional plot), revealing the important quantities such as gain margin, phase margin and delay margin. Because it is based on the frequency domain, time delay in the system can be easily captured.
- Sensitivity analysis based on the frequency domain is fundamental to the understanding of the characteristics of closed-loop control systems. In the context of sensitivity analysis, the parameter called closed-loop bandwidth is defined.
- Understanding the closed-loop performance in terms of disturbance rejection and reference following is based on the frequency response of the sensitivity function and complementary sensitivity function. Because the relationship between the complementary sensitivity function and sensitivity function is captured by:

  $$T(s) + S(s) = 1$$

  the performance requirements are consistent in the sense that a closed-loop system that has a fast reference following will also have a fast disturbance rejection.
- Understanding the closed-loop performance in terms of disturbance rejection (or reference following) and measurement noise attenuation is also based on the frequency response of the sensitivity function and complementary sensitivity function. However, the relationship

  $$T(s) + S(s) = 1$$

  captures the fundamental trade-off between disturbance rejection and measurement noise attenuation. There is a conflict between the two requirements, meaning that a fast disturbance rejection in a closed-loop control system will inevitably lead to the amplification of measurement noise. The existence of measurement noise limits the closed-loop control performance in terms of disturbance rejection and reference following.
- Modelling errors are often present in the PID control system design. Because of their existence, there is a difference in the closed-loop performance between what is desired and what is actually achieved. Their effects on the closed-loop performance are analyzed and quantified in the frequency domain using the errors and the complementary sensitivity function. Furthermore, the Nyquist stability criterion provides an effective means to access the information about gain margin, phase margin and delay margin, which are important to assess the impact of modelling errors on closed-loop stability.

2.9 Further Reading

1. Text books in control engineering include, Franklin et al. (1998), Franklin et al. (1991), Ogata (2002), Golnaraghi and Kuo (2010), Goodwin et al. (2000) and Astrom and Murray (2008).
2. Disturbance rejection, reference following and robustness are discussed in Garpinger et al. (2014), in Alcántara et al. (2013). Two-degrees-of-freedom PID controller design is proposed in Gorez (2003) and Yukitomo et al. (2004), Araki and Taguchi (2003). Load disturbance rejection with consideration on robustness of closed-loop system is proposed in Panagopoulos et al. (2002). Effect of noise for tuning the controller parameters is discussed in Fertik (1975).
3. The Ziegler-Nichols step response method is revisited from the point of view of robust loop shaping in Åström and Hägglund (2004).
4. Tuning of PID controllers based on gain and phase margin specifications was proposed in Ho et al. (1995), Ho and Xu (1998), Ho et al. (2000). The Ziegler-Nichols, and Cohen-Coon tuning formulas that optimize for load disturbance response are analyzed in Ho et al. (1996) for gain margin and phase margin. Gain and phase margins are used in selecting PID controller parameters with consideration of robustness and closed-loop performance in Ho et al. (1998).
5. Disturbance rejection for the modified IMC tuning rules is specifically addressed in Skogestad (2006). Robustness of IMC is discussed in Morari and Zafiriou (1989). Further discussions of tuning guidelines and robustness for IMC can be found in Vilanova (2008).

Problems

2.1 *In Example 2.3, three PI controllers (see Table 2.3) were found using tuning-rules together with analysis of Nyquist diagram for the following continuous-time system:*

$$G(s) = \frac{0.5e^{-20s}}{(30s+1)^3} \tag{2.51}$$

1. *Evaluate the closed-loop step response of the three PI control systems in one-degree-of-freedom using a Simulink simulation.*
2. *Evaluate the closed-loop step response of the PI control systems in two-degrees-of-freedom where the reference filter is selected as $H(s) = \frac{1}{\tau_I s+1}$, and compare the output responses with the simulation results presented in Figure 2.7.*

2.2 *In Example 1.8, Padula and Visioli tuning rules were used to find PID controller for the following system*

$$G(s) = \frac{0.5e^{-20s}}{(30s+1)^3}$$

where $Kc = 3.54$, $\tau_I = 40.1098$, $\tau_D = 27.0037$.

1. *Find the closed-loop transfer function between $R(s)$ and $Y(s)$ with the control signal $U(s)$ defined as*

$$U(s) = K_c(1 + \frac{1}{\tau_I s} + \frac{\tau_D s}{0.1\tau_D s + 1})(R(s) - Y(s))$$

Simulate the closed-loop unit step response for this one-degree-of-freedom control system with sampling interval $\Delta t = 1$ (sec).

2. *Find the closed-loop transfer function between $R(s)$ and $Y(s)$ with the control signal $U(s)$ defined as*

$$U(s) = \frac{K_c}{\tau_I s}(R(s) - Y(s)) - K_c Y(s) - \frac{K_c \tau_D s}{0.1\tau_D s + 1} Y(s)$$

Simulate the closed-loop step response for this IPD control system and compare the results with the ones from the one-degree-of-freedom controller structure.

3. *Observing from the closed-loop transfer functions, what reference filter should be chosen if the remainder of the overshoot from the IPD implementation were to be eliminated? Verify the results using the selected reference filter.*

2.3 *Use Routh-Hurwitz stability criterion to determine the range of the proportional controller K_c that will stabilize the systems with the following transfer functions.*

1. $G(s) = \dfrac{0.1}{(s+1)(s+3)}$
2. $G(s) = \dfrac{(-5s+1)}{(s+2)^2(s+10)}$
3. $G(s) = \dfrac{s+0.1}{(s-3)(s+6)(s+1)}$
4. $G(s) = \dfrac{-s+3)}{(s+3)(s^2+s+5)}$

2.4 *Use Routh-Hurwitz criterion to determine the range of the integral time constant τ_I that will produce a stable closed-loop system for the plant with transfer function $G(s) = \frac{1}{(s^2+s+6)(s+1)}$. Here we assume that the proportional gain $K_c = 2$.*

2.5 *Consider a system with the following transfer function:*

$$G(s) = \frac{(s-2)}{(s+2)^3(s+5)}.$$

1. *Assuming that the integral time constant $\tau_I = 0.2$, form the closed-loop characteristic equation in terms of the proportional control K_c, and numerically determine the variations of closed-loop poles with respect to K_c ($0 \leq K_c \leq 100$). This numerical procedure leads to root-locus analysis of the PI control system with respect to the variations of K_c.*
2. *Alternatively, we can use MATLAB function 'rlocus.m' for the root-locus analysis with the loop transfer function:*

$$G(s)C(s) = K_c \frac{(s-2)}{(s+2)^3(s+5)} \frac{s+5}{s}$$

3. *From the root-locus analysis, determine the range of the proportional gain K_c that will produce a stable closed-loop system.*
4. *What are your observations on the closed-loop pole variations with respect to the proportional gain K_c from the root-locus analysis?*

2.6 *This Problem is an extension to Example 2.7. Because the reactor's transfer function model has 8 stable zeros, it is difficult to design a PID controller using the PID tuning rules without desired closed-loop performance adjustment.*

1. *Use Padula and Visioli PID controller tuning rules based on the first order plus delay model shown in (2.49) to find two sets of PID controller parameters for $M_s = 1.4$ and $M_s = 2$ (see Table 1.5).*
2. *Plot the Nyquist diagrams for both PID controlled systems using the original reactor model (2.48) and verify if the closed-loop systems are stable.*
3. *Based on the Nyquist diagrams, propose what we would do to improve the gain margin and phase margin for the two PID controlled systems, and modify the PID controller parameters accordingly.*
4. *Simulate the closed-loop step response for both systems based on the original reactor model (2.48) using the modified PID controllers.*

2.7 *One of the simple and yet effective approaches to model order reduction is to neglect the small time constant ϵ in the transfer function model:*

$$G(s) = \frac{e^{-2s}}{(10s+1)(\epsilon s+1)}$$

in order to obtain a first order plus delay model.

1. *Determine the multiplicative modelling error $\Delta G_m(s)$ when the approximate model is taken as $G_A(s) = \frac{e^{-2s}}{(10s+1)}$.*
2. *Calculate $\Delta G_m(j\omega)$ for $\epsilon = 0.1, 1, 10$ and compare their magnitudes in a diagram. What are your observations?*
3. *Use Padula and Visioli PID controller tuning rules ($M_s = 2$) based on the first order plus delay model to find two sets of PID controller parameters.*
4. *Calculate the complementary sensitivity function $T(j\omega)$ for the PID controller.*
5. *Determine the closed-loop robust stability using the relationship*

   $$|T(j\omega)\Delta G_m(j\omega)| < 1$$

   for all ω. Is the closed-loop system stable for $\epsilon = 0.1, 1, 10$?
6. *Compare the closed-loop step responses, which are simulated based on the original second order plus delay model for $\epsilon = 0.1$ and $\epsilon = 1$ with sampling interval $\Delta t = 0.01$ and derivative filter time constant equal to $0.1\tau_D$. What are your observations?*

2.8 *The original IMC-PI tuning rules are given for a first order plus delay system:*

$$G(s) = \frac{K_{ss}e^{-ds}}{\tau s+1}$$

where τ_{cl} is the desired closed-loop time constant and

$$K_c = \frac{1}{K_{ss}}\frac{\tau}{\tau_{cl}+d}; \quad \tau_I = \tau \tag{2.52}$$

1. *Find the complementary sensitivity function $T(s)$, the input disturbance sensitivity function $S_i(s)$ for the IMC-PI controlled system. What are your observations in terms of the poles of the sensitivity functions?*
2. *For a system with $d = 2$, $\tau = 20$, $K_{ss} = 0.5$, calculate $|T(j\omega)|$ and $|S_i(j\omega)|$ for $\tau_{cl} = 5, 10$. Will $|T(j\omega)|$ change with respect to τ_{cl}? If so, what are the bandwidth parameters? Will $|S_i(j\omega)|$ change with respect to τ_{cl}?*
3. *Simulate the closed-loop step response for the two PI controlled systems with sampling interval $\Delta t = 0.01$. Does τ_{cl} affect the closed-loop step response speed?*
4. *Simulate the closed-loop input disturbance response for the two PI controlled systems, where the disturbance is a constant with magnitude of 3. Does τ_{cl} affect the closed-loop disturbance rejection?*
5. *What if the disturbance occurred at the output, would τ_{cl} make a difference on the disturbance rejection? Verify your intuition with the closed-loop simulation of output disturbance.*

3

Model-Based PID and Resonant Controller Design

3.1 Introduction

The models used in the design of PID controllers are limited to two particular types. One is a first order model while the other is a second order model. If the plant dynamics yield a higher order model, an approximation is often involved to obtain a first order or a second order model so that a PID controller can be designed using a model based approach.

When using model based design methods, a desired closed-loop performance specification is required before commencing. The desired performance is chosen in terms of the locations of the desired closed-loop poles, which reflect the closed- loop response time to reference change and disturbance rejection or in the frequency domain the bandwidth of the desired closed- loop control system. The desired closed-loop performance is often adjusted several times using closed-loop simulation and experimental validation before the designer finds the suitable closed-loop performance.

3.2 PI Controller Design

To design a PI controller, a first order model is used. Although the first order dynamics is the basic unit to form a system, it can also be used to describe a number of commonly encountered physical systems, such as the dynamic relationship between motor torque and angular velocity in the motor control problem, and fluid in-flow rate and fluid level in a fluid vessel control problem.

3.2.1 Desired Closed-loop Performance Specification

In the model-based design, a desired closed-loop performance is required to be specified. In the PI controller case, a second order transfer function is used in the specification,

$$T(s) = \frac{w_{\mathrm{n}}^2}{s^2 + 2\xi w_{\mathrm{n}} s + w_{\mathrm{n}}^2} \tag{3.1}$$

where w_{n} and ξ are the natural frequency and damping coefficient for the second order transfer function. These are the free parameters to be selected by the designer as the desired performance specification.

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

The parameter ξ is often chosen as 1 or 0.707. When $\xi = 1$, the poles of the desired closed-loop transfer function (3.1) are the solutions of the polynomial equation,

$$s^2 + 2w_n s + w_n^2 = 0 \tag{3.2}$$

which are $s_1 = s_2 = -w_n$. Namely, we have two identical poles when $\xi = 1$. With the second choice of $\xi = 0.707$, the poles are a pair of complex conjugate numbers determined by

$$s_{1,2} = \frac{-2\xi w_n \pm \sqrt{4\xi^2 w_n^2 - 4w_n^2}}{2} = -0.707w_n \pm j0.707w_n. \tag{3.3}$$

With the parameter ξ chosen (either 1 or 0.707), the natural frequency w_n becomes a closed-loop performance parameter that the user specifies according to the desired closed-loop response requirement. In general, when w_n is larger, the desired closed-loop response is faster. The parameter w_n is directly related to the closed-loop response time and the band limit of the closed-loop system, which provide us with the guidelines on its selection. These two aspects are examined.

From the simulation of a step response (3.1) (see Figure 3.1), the response time is inversely proportional to the parameter w_n. Figure 3.1(a) shows that with the damping coefficient $\xi = 0.707$, the total step response time is about $\frac{3}{w_n}$ and with $\xi = 1$, as shown in Figure 3.1(b), the total step response time is about $\frac{5}{w_n}$. There is another estimate of w_n that can be used as a guideline for the designer. Here the parameter w_n is related to the bandwidth of the desired closed-loop control system. For the desired closed-loop transfer function given by (3.1), when $\xi = 0.707$, it can be verified that $|T(\omega)| = \frac{1}{\sqrt{2}}$ at the frequency $\omega = w_n$. Hence, with the special choice of the damping coefficient $\xi = 0.707$, the natural frequency w_n is the bandwidth of the closed-loop system, which we can directly use for the closed-loop performance specification. With the choice of $\xi = 1$, the bandwidth of the closed-loop system is slightly smaller.

3.2.2 Model and Controller Structures

For the first order model, we assume that a first order time constant τ and a steady state gain K are known to form the Laplace transfer function,

$$G(s) = \frac{K}{\tau s + 1} \tag{3.4}$$

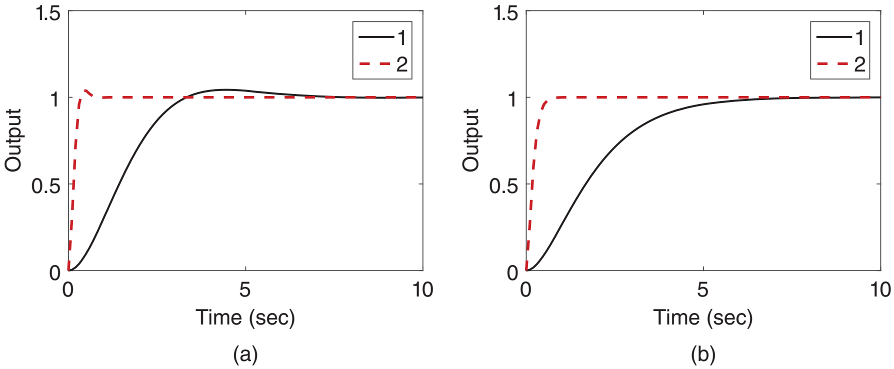


Figure 3.1 Step response of the desired closed-loop transfer function. (a) $\xi = 0.707$. (b) $\xi = 1$. Key: line (1) $w_n = 1$; line (2) $w_n = 10$.

which can also be expressed in the pole–zero form,

$$G(s) = \frac{b}{s+a} \tag{3.5}$$

where $a = 1/\tau$ and $b = K/\tau$.

For a PI controller, its transfer function is given by

$$C(s) = K_\mathrm{c}\left(1 + \frac{1}{\tau_\mathrm{I} s}\right) \tag{3.6}$$

which can be written in the transfer function form,

$$C(s) = \frac{c_1 s + c_0}{s} \tag{3.7}$$

where $K_\mathrm{c} = c_1$ and $\tau_\mathrm{I} = \frac{c_1}{c_0}$. We will first find the coefficients c_1 and c_0 based on the model (3.5), then convert these coefficients into the standard PI controller parameters K_c and τ_I.

The key to the solution of the PI controller parameters is to equate the desired closed-loop poles to the actual closed-loop poles. The locations of the closed-loop poles determine whether the closed-loop system is stable, the closed-loop response time, and the band limit of the closed-loop system.

To this end, we calculate the actual closed-loop system using the design model (3.5) and the controller model (3.7) via the closed-loop transfer function

$$\begin{aligned} T_\mathrm{cl} &= \frac{G(s)C(s)}{1 + G(s)C(s)} = \frac{\frac{b}{s+a}\frac{c_1 s+c_0}{s}}{1 + \frac{b}{s+a}\frac{c_1 s+c_0}{s}} \\ &= \frac{b(c_1 s + c_0)}{s(s+a) + b(c_1 s + c_0)}. \end{aligned} \tag{3.8}$$

The closed-loop poles of the actual system are the solutions of the polynomial equation with respect to s:

$$s(s+a) + b(c_1 s + c_0) = 0. \tag{3.9}$$

Equation (3.9) is called the closed-loop characteristic equation. Since the model parameters a and b are given, the free parameters in (3.9) are the controller parameters c_1 and c_0. To find the controller parameters c_1 and c_0, the following polynomial equation is set:

$$s(s+a) + b(c_1 s + c_0) = s^2 + 2\xi w_\mathrm{n} s + w_\mathrm{n}^2 \tag{3.10}$$

where the left-hand side of the equation (3.10) is the polynomial that determines the actual closed-loop poles and the right-hand side is the polynomial that determines the desired closed-loop poles. By equating these two polynomials, the actual closed-loop poles are assigned to the desired closed-loop poles. This controller design technique is called pole assignment controller design.

Now, we compare the coefficients of the polynomial equation (3.10) on both sides:

$$s^2 : 1 = 1 \tag{3.11}$$

$$s : a + bc_1 = 2\xi w_\mathrm{n} \tag{3.12}$$

$$s^0 : bc_0 = w_\mathrm{n}^2. \tag{3.13}$$

Solving (3.12) gives

$$c_1 = \frac{2\xi w_\mathrm{n} - a}{b} \tag{3.14}$$

and solving (3.13) gives

$$c_0 = \frac{w_\mathrm{n}^2}{b}. \tag{3.15}$$

With the relationships between c_1, c_0 and K_c, τ_I (see Equation (3.7)), we find the PI controller parameters as

$$K_\mathrm{c} = c_1 = \frac{2\xi w_\mathrm{n} - a}{b} \tag{3.16}$$

$$\tau_\mathrm{I} = \frac{c_1}{c_0} = \frac{2\xi w_\mathrm{n} - a}{w_\mathrm{n}^2}. \tag{3.17}$$

Example 3.1 *A first order system is used to describe the dynamic relationship between voltage change and velocity of a DC motor. Assume that a particular motor has the Laplace transfer function model,*

$$G(s) = \frac{0.1}{10s + 1}. \tag{3.18}$$

Find the PI controller parameters for velocity control, where the desired closed-loop performance is specified by two performance levels: one fast response $w_\mathrm{n} = 5$ and one slow response $w_\mathrm{n} = 0.5$; $\xi = 0.707$ in both cases. Simulate the closed-loop step responses with proportional control on the output only and compare the results.

Solution. *The model parameters needed for the PI controller design are $a = \frac{1}{10} = 0.1$ and $b = \frac{0.1}{10} = 0.01$. With these parameters and the closed-loop performance specification, based on equations (3.16 and 3.17) we find the controller parameters as, for $w_\mathrm{n} = 5$ and $\xi = 0.707$,*

$$K_\mathrm{c} = 697; \tau_\mathrm{I} = 0.2788,$$

for $w_\mathrm{n} = 0.5$ and $\xi = 0.707$

$$K_\mathrm{c} = 60.7; \tau_\mathrm{I} = 2.43.$$

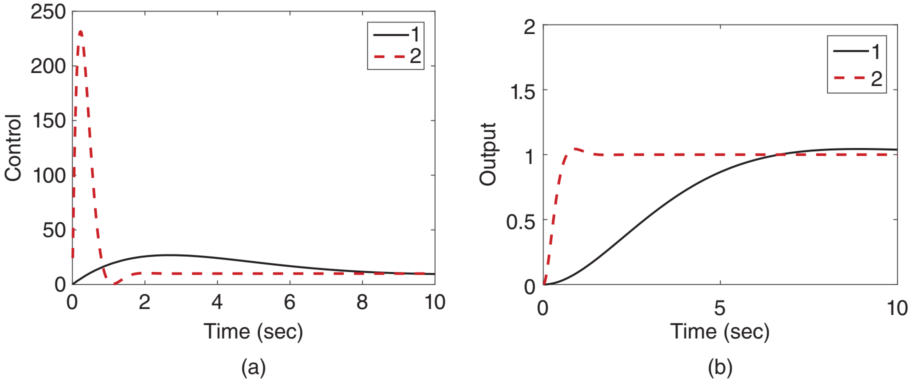


Figure 3.2 Closed-loop response (Example 3.1). (a) Control signal. (b) Output signal. Key: line (1) $w_\mathrm{n} = 0.5$; line (2) $w_\mathrm{n} = 5$.

As we can see, when w_n is larger, the proportional gain K_c is larger whilst the integral time constant τ_I is smaller.

We simulate the closed-loop unit step responses with the results shown in Figure 3.2. The closed-loop response speed is much faster when $w_n = 5$ in comparison with the case of $w_n = 0.5$, and the control signal has a much larger amplitude.

3.2.3 Closed-loop Transfer Functions for Different Configurations

With the PI controller designed, now we study the closed-loop transfer functions for using the traditional PI controller configuration and the IP controller configuration.

For the traditional PI controller configuration (see Figure 1.5), the Laplace transform of the control signal, $U(s)$, is expressed as the function of feedback error signal $E(s)$ using the relation

$$U(s) = C(s)E(s) = \frac{c_1 s + c_0}{s} E(s) \tag{3.19}$$

where $E(s) = R(s) - Y(s)$ and $C(s) = \frac{c_1 s + c_0}{s}$. The closed- loop transfer function between the reference signal $R(s)$ and the output signal $Y(s)$ is then

$$\frac{Y(s)}{R(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{3.20}$$

where $G(s)$ is the first order transfer function $\frac{b}{s+a}$. By substituting the transfer functions of controller ($C(s)$) and the plant transfer function $G(s)$ into (3.20), we obtain the closed-loop transfer function for the PI control system using the traditional implementation:

$$\frac{Y(s)}{R(s)} = \frac{b(c_1 s + c_0)}{s(s+a) + b(c_1 s + c_0)}. \tag{3.21}$$

Note that the denominator of (3.21) is used in the design of the PI controller (see (3.10)), which is made equal to the desired closed-loop characteristic polynomial $s^2 + 2\xi w_n s + w_n^2$. By substituting the controller parameters (see (3.14) and (3.15) and the desired closed-loop characteristic polynomial into (3.21), the closed-loop transfer function becomes:

$$\frac{Y(s)}{R(s)} = \frac{(2\xi w_n - a)s + w_n^2}{s^2 + 2\xi w_n s + w_n^2}. \tag{3.22}$$

Using the traditional PI controller configuration, the actual closed-loop transfer function between the reference signal and the output is not equal to the desired closed-loop transfer function specified in the design (see $T(s)$ in (3.8)). Instead, there is a zero in the closed-loop transfer function at the location determined by the polynomial equation:

$$(2\xi w_n - a)s + w_n^2 = 0 \tag{3.23}$$

which is at $s = -\frac{w_n^2}{2\xi w_n - a}$. The existence of this zero could cause some overshoot in the step response. One can verify that the closed-loop transfer function (3.22) has a bandwidth larger than w_n for the choice of $\xi = 0.707$.

For the IP controller configuration (see Figure 1.7), the control signal is expressed as

$$\begin{aligned} U(s) &= -K_c Y(s) + \frac{K_c}{\tau_I s} E(s) \\ &= -c_1 Y(s) + \frac{c_0}{s}(R(s) - Y(s)). \end{aligned} \tag{3.24}$$

The output signal $Y(s)$ is expressed as

$$Y(s) = \frac{b}{s+a} U(s). \tag{3.25}$$

By substituting (3.24) into (3.25), we find the closed-loop transfer function for the alternative configuration as

$$\frac{Y(s)}{R(s)} = \frac{bc_0}{s(s+a) + b(c_1 s + c_0)}. \tag{3.26}$$

By the design procedure, the denominator of this transfer function is $s^2 + 2\xi w_n s + w_n^2$ and the numerator $bc_0 = w_n^2$, therefore the closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2} \tag{3.27}$$

which is equal to the desired closed-loop transfer function we specified in the performance specification (see (3.1)).

As for disturbance rejection and measurement noise attenuation, both PI control system configurations have the identical closed- loop transfer function because the structural change introduced in the alternative configuration is only related to how the reference signal is introduced in the feedback loop. We illustrate this point by calculating the closed-loop transfer function between input disturbance and the plant output.

Suppose that an input disturbance has a Laplace transform $D_i(s)$ and it enters the system at the position of plant input. In this case, the plant output is expressed as

$$Y(s) = \frac{b}{s+a}(U(s) + D_i(s)). \tag{3.28}$$

To derive the closed-loop transfer function between the input disturbance and the plant output, we assume that the reference signal $R(s) = 0$ so to concentrate on the disturbance rejection. When the reference signal $R(s) = 0$, the control signals from both configurations (see (3.19) and (3.24)) become identical to the transform:

$$U(s) = -c_1 Y(s) - \frac{c_0}{s} Y(s). \tag{3.29}$$

Substituting (3.29) into (3.28), we obtain the closed-loop transfer function between the input disturbance $D_i(s)$ and output $Y(s)$ as

$$\begin{aligned} \frac{Y(s)}{D_i(s)} &= \frac{bs}{s(s+a) + b(c_1 s + c_0)} \\ &= \frac{bs}{s^2 + 2\xi w_n s + w_n^2} \end{aligned} \tag{3.30}$$

where we have used the design equation in (3.10).

Note that there is a factor s in the numerator of the closed-loop transfer function. This factor ensures that the closed-loop control system will reject a step input disturbance without steady- state error. This point will be made clear through the examples in the following section.

Example 3.2 *A continuous-time system is described by the following differential equation:*

$$J\frac{\mathrm{d}\omega(t)}{\mathrm{d}t} + B\omega(t) = kv(t) + T_{\mathrm{L}}(t) \tag{3.31}$$

where $v(t)$ and $\omega(t)$ are the control and output signal, $T_{\mathrm{L}}(t)$ is the unknown disturbance, and $J = 0.02$, $B = 0.001$, $k = 0.5$ are the physical parameters. The control objective is to maintain a desired output of the system while rejecting the disturbance. The desired closed-loop band limit is specified as $w_{\mathrm{n}} = 5$ (rad s^{-1}) with the damping coefficient $\xi = 0.707$. Design a PI controller to achieve the control objective and simulate the closed-loop response using both PI control system configurations.

Solution. *In order to obtain the transfer function model between the input v and the output $\omega(t)$, we take the Laplace transform of the dynamic model (3.31):*

$$Js\Omega(s) + B\Omega(s) = kV(s) + T_{\mathrm{L}}(s) \tag{3.32}$$

where $\Omega(s)$, $V(s)$ and $T_{\mathrm{L}}(s)$ are the Laplace transforms of the continuous time variables.

Equation (3.32) can also be expressed as

$$\begin{aligned}\Omega(s) &= \frac{k}{Js+B}V(s) + \frac{T_{\mathrm{L}}(s)}{Js+B}\\ &= \frac{k/J}{s+\frac{B}{J}}V(s) + \frac{T_{\mathrm{L}}(s)/J}{s+\frac{B}{J}}.\end{aligned} \tag{3.33}$$

The PI controller parameters are calculated as

$$K_{\mathrm{c}} = \frac{2\xi w_{\mathrm{n}} - a}{b} = 0.2808;\ \tau_{\mathrm{I}} = \frac{2\xi w_{\mathrm{n}} - a}{w_{\mathrm{n}}^2} = 0.2808$$

where $a = \frac{B}{J} = 0.05$ and $b = \frac{k}{J} = 25$, $w_{\mathrm{n}} = 5$ and $\xi = 0.707$.

The closed-loop response is simulated using Simulink. In the closed-loop simulation, a unit step input signal is used as the reference change at time $t = 0$ and a step input signal with magnitude 1.5 is used as the input disturbance, which occurred at time $t = 10$. Figures 3.3(a) and (b) compare the closed-loop output and control signals from the original PI controller configuration with those from the IP controller configuration. It is seen that the IP controller configuration avoids the overshoot that exhibits in the original configuration; however, both controller configurations provide the same responses in disturbance rejection.

3.2.4 Food for Thought

1. Would you use a second order model to design a PI controller?
2. If you found that the closed-loop response speed was too slow, would you decrease the parameter w_n?
3. Is it correct to say that the closed-loop bandwidth $\omega_b = w_n$ when the closed-loop system transfer function is characterized by $\frac{w_n^2}{s^2+2\xi w_n s+w_n^2}$ if the damping coefficient $\xi = 0.707$?

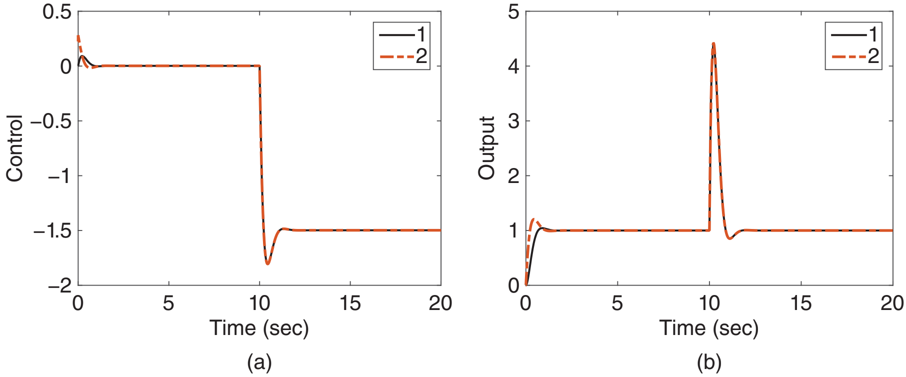


Figure 3.3 Closed-loop response of PI control system (Example 3.2). (a) Control signal. (b) Output response. Key: line (1) IP control system; line (2) PI control system.

4. Would you be able to list three physical systems that can be described by a first order model?
5. Has the IP controller configuration changed the characteristics of disturbance rejection when comparing with the original PI controller?

3.3 Model Based Design for PID Controllers

Second order models are used directly for the design of a PD or a PID controller. In addition, a first order plus delay model is approximated using a second order transfer function model, where the irrational transfer function e^{-ds} is approximated using a rational transfer function $\frac{-ds+2}{ds+2}$ that is called first order Padé approximation. If the mathematical model is of higher order, then approximation is made to obtain a second order model in order for the PID controller design to be carried out.

3.3.1 PD Controller Design

The combination of proportional and derivative (PD) control could be useful in the situation where stabilization of an unstable system is of main concern or in the situation where the system is severely oscillatory.

Because the derivative action will amplify measurement noise (see Chapter 2), a derivative filter is required for the implementation of a PD controller. Thus, the general form of a PD controller is given as

$$C(s) = K_c + \frac{K_d s}{\tau_f s + 1} \tag{3.34}$$

where K_c, K_d and τ_f are the proportional gain, the derivative control gain and the filter time constant, respectively.

For this type of controller design, we assume that the continuous time system is a second order with the transfer function:

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}. \tag{3.35}$$

It is not a straightforward task to choose the parameters K_c, K_d and τ_f based on the second order model (3.35). However, the PD controller can be converted into the classical lead-lag compensator that has the following form:

$$C(s) = \frac{p_1 s + p_0}{s + l_0} \tag{3.36}$$

where the parameters K_c, K_d are τ_f are related to p_1, p_0 and τ_f via the relations below:

$$\tau_\mathrm{f} = \frac{1}{l_0} \tag{3.37}$$

$$K_\mathrm{c} = \frac{p_0}{l_0} \tag{3.38}$$

$$K_\mathrm{d} = \frac{p_1}{l_0} - \frac{p_0}{l_0^2}. \tag{3.39}$$

The lead-lag compensator in (3.36) can be designed by positioning the desired closed-loop poles on the left half of the complex plane.

With the lead-lag compensator, the actual closed-loop characteristic polynomial is a third order, which is

$$\begin{aligned} A_\mathrm{cl}(s) &= (s + l_0)(s^2 + a_1 s + a_0) + (p_1 s + p_0)(b_1 s + b_0) \\ &= s^3 + (a_1 + l_0 + p_1 b_1)s^2 + (a_0 + l_0 a_1 + p_0 b_1 + b_0 p_1)s + (l_0 a_0 + b_0 p_0). \end{aligned}$$

By choosing a third order desired closed-loop characteristic polynomial having the following form,

$$A_\mathrm{cl}^\mathrm{d}(s) = s^3 + a_2^\mathrm{cl} s^2 + a_1^\mathrm{cl} s + a_0^\mathrm{cl}$$

and letting $A_\mathrm{cl}(s) = A_\mathrm{cl}^\mathrm{d}(s)$, we obtain the following linear equations:

$$\begin{aligned} a_1 + l_0 + p_1 b_1 &= a_2^\mathrm{cl} \\ a_0 + l_0 a_1 + p_0 b_1 + b_0 p_1 &= a_1^\mathrm{cl} \\ l_0 a_0 + b_0 p_0 &= a_0^\mathrm{cl}. \end{aligned}$$

The parameters are found via the solution of the linear equations as

$$\begin{bmatrix} l_0 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} 1 & b_1 & 0 \\ a_1 & b_0 & b_1 \\ a_0 & 0 & b_0 \end{bmatrix}^{-1} \begin{bmatrix} a_2^\mathrm{cl} - a_1 \\ a_1^\mathrm{cl} - a_0 \\ a_0^\mathrm{cl} \end{bmatrix}. \tag{3.40}$$

The polynomial equation $A_\mathrm{cl}(s) = A_\mathrm{cl}^\mathrm{d}(s)$ is called Diophantine equation, which is the essential step for finding the controller parameters in the pole assignment controller design. The matrix with the dimensions 3 × 3 in (3.40) is called the Sylvester matrix, which is required to be invertible in the pole assignment controller design.

The following tutorial summarizes the computational procedure for finding the parameters for the PD controller with filter. We will use this program later on for applications.

Tutorial 3.1 *In this tutorial, we will write a simple MATLAB program to calculate the PD controller parameters. The desired closed-loop characteristic polynomial is specified as*

$$A_\mathrm{cl} = s^3 + a_2^\mathrm{cl} s^2 + a_1^\mathrm{cl} s + a_0^\mathrm{cl}$$

and the model is a second order system with the transfer function

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}.$$

Step by Step

1. *Create a new file called PDplace.m for the MATLAB function.*
2. *Define the input and output variables for the MATLAB function, where* $a1$*,* $a0$*,* $b1$*, and* $b0$ *are the model parameters and Acl is the desired closed-loop characteristic polynomial. Enter the following program into the file:*

   ```
   function [Kc,Kd,tauf]=PDplace(a1,a0,b1,b0,Acl);
   ```

3. *Find the closed-loop performance parameters. Continue entering the following program into the file:*

   ```
   ac_2=Acl(2);
   ac_1=Acl(3);
   ac_0=Acl(4);
   ```

4. *Form the following matrix and vector for the solution of the PD controller parameters, and solve the linear equation. Continue entering the following program into the file:*

   ```
   S_matrix=[1 b1 0; a1 b0 b1; a0 0 b0];
   Vec=[ac_2-a1;ac_1-a0;ac_0];
   contr_p=inv(S_matrix)*Vec;
   ```

5. *Convert the parameters into a PD controller with derivative filter. Continue entering the following program into the file:*

   ```
   L0=contr_p(1);
   p1=contr_p(2);
   p0=contr_p(3);
   tauf=1/L0;
   Kc=p0/L0;
   Kd=p1/L0-p0/L0^2;
   ```

6. *Test this program using the double integrator system presented in Example 3.3, where we use the following codes:*

   ```
   a1=0;
   a0=0;
   b1=0;
   b0=0.1;
   Ac=conv([1 1],[1 1]);
   Acl=conv(Ac,[1 1]);
    [Kc,Kd,tauf]=PDplace(a1,a0,b1,b0,Acl)
   ```

For many applications, the second order model is simplified as

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0} \tag{3.41}$$

and the PD controller parameters have the following solutions:

$$l_0 = a_2^{\mathrm{cl}} - a_1 \tag{3.42}$$

$$p_1 = \frac{a_1^{\mathrm{cl}} - a_0 - l_0 a_1}{b_0} \tag{3.43}$$

$$p_0 = \frac{a_0^{\mathrm{cl}} - l_0 a_0}{b_0}. \tag{3.44}$$

Example 3.3 *A double integrator system is described by the Laplace transfer function*

$$G(s) = \frac{0.1}{s^2}.$$

Design a PD controller with filter to stabilize this system. All desired closed-loop poles are positioned at -1.

Solution. *From the double integrator model, we have* $a_1 = a_0 = 0$, $b_1 = 0$, *and* $b_0 = 0.1$. *The desired closed- loop polynomial is*

$$(s + 1)^3 = s^3 + 3s^2 + 3s + 1$$

Therefore, $a_2^{\mathrm{cl}} = 3$, $a_1^{\mathrm{cl}} = 3$, *and* $a_0^{\mathrm{cl}} = 1$. *The controller parameters are found using*

$$\begin{bmatrix} l_0 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix} \tag{3.45}$$

leading to $l_0 = 3$, $p_1 = 30$, *and* $p_0 = 10$. *Basically, the lead and lag compensator for this system becomes*

$$C(s) = \frac{30s + 10}{s + 3}.$$

For implementation of the PD controller, we use the equations (3.37)–(3.39) to find the corresponding proportional, derivative gains and the filter time constant as

$$\tau_{\mathrm{f}} = \frac{1}{3}; K_{\mathrm{c}} = \frac{10}{3}; K_{\mathrm{d}} = \frac{80}{9}; \tau_{\mathrm{D}} = \frac{K_{\mathrm{d}}}{K_{\mathrm{c}}} = \frac{8}{3}.$$

3.3.2 Analytical Examples for Ideal PID with Pole-zero Cancellation

When designing a PID controller, the pole–zero cancellation technique is widely used in the application field. The main reason behind this practice is that with the pole–zero cancellation technique, the controller parameter calculations become very simple and can be performed using pencil on the back of an envelope. However, there are two important rules regarding the pole–zero cancellation technique. Firstly, an unstable pole or zero in the system should not be canceled because the cancellation will lead to an internally unstable system, for the reason that the canceled pole or zero is still part of desired closed-loop poles (see Section 2.5). Secondly, a stable pole close to the imaginary axis, which is a pole corresponding to a large time constant, in the system should not be canceled because the slow pole will re-appear to become the dominant pole in the closed-loop response to input disturbance, and as a result, slow disturbance rejection will occur (see the sensitivity analysis in Chapter 2). As a general rule, a stable pole or zero could be canceled if its position is on the left-hand side of the desired closed-loop poles on the complex plane. Additionally, it is the faster stable pole in the plant model that gets canceled.

We assume that there are two poles in a second order model and both are real, stable poles. With these assumptions, the transfer function is expressed as

$$G(s) = \frac{b_0}{(s+\alpha_1)(s+\alpha_2)} \tag{3.46}$$

where $\alpha_2 > 0$ is positive and $\alpha_2 \geq \alpha_1$.

The PID controller is assumed to have the ideal structure,

$$C(s) = K_c(1 + \frac{1}{\tau_I s} + \tau_D s) \tag{3.47}$$

and an implementation filter will be added at the implementation stage with a small β parameter chosen by the designer (see Section 1.2). This means that the parameter β will not be considered in the design stage. We will deploy a technique called pole–zero cancellation in the PID controller design, and with this technique the controller parameter solutions become very simple.

We re-write the PID controller given in (3.47) into the transfer function form,

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s}. \tag{3.48}$$

By comparing (3.48) with (3.47), we have the relationships,

$$K_c = c_1; \;\; \tau_I = \frac{c_1}{c_0}; \;\; \tau_D = \frac{c_2}{c_1}. \tag{3.49}$$

Thus, we find the parameters in (3.48) first, then convert them into the PID controller parameters required in the implementation stage.

When using the pole–zero cancellation technique, we assume that the numerator of the controller $C(s)$ is factored and the controller now has the form,

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s} = \frac{c_2(s+\gamma_1)(s+\gamma_2)}{s}. \tag{3.50}$$

By choosing the zero of the controller $-\gamma_2$ equal to the pole of the model $-\alpha_2$ (i.e. $\gamma_2 = \alpha_2$), we canceled the pole in the model with the zero in the controller. The net effect of this is that the relationship $G(s)C(s)$ is simplified into

$$G(s)C(s) = \frac{b_0 c_2(s+\gamma_1)}{(s+\alpha_1)s} \tag{3.51}$$

the closed-loop transfer function between the reference signal $R(s)$ and the output $Y(s)$ becomes

$$\begin{aligned}\frac{Y(s)}{R(s)} &= \frac{C(s)G(s)}{1+C(s)G(s)} \\ &= \frac{b_0 c_2(s+\gamma_1)}{s(s+\alpha_1) + b_0 c_2(s+\gamma_1)}\end{aligned} \tag{3.52}$$

Note that the free parameters in (3.52) are c_2 and γ_1, respectively, and its denominator is a second order polynomial. The design becomes identical to the case when we designed the PI controller using pole assignment technique. Thus, by choosing the desired closed-loop characteristic polynomial as

$$A_{\rm cl}(s) = s^2 + 2\xi w_{\rm n} s + w_{\rm n}^2 \tag{3.53}$$

with $w_{\rm n} > 0$ and $\xi = 0.707$ or $\xi = 1$, and equating the desired closed-loop characteristic polynomial with the denominator of (3.52), we obtain the polynomial equation,

$$s(s + \alpha_1) + b_0 c_2 (s + \gamma_1) = s^2 + 2\xi w_{\rm n} s + w_{\rm n}^2 \tag{3.54}$$

By comparing both sides of (3.54), the free parameters are determined as

$$c_2 = \frac{2\xi w_{\rm n} - \alpha_1}{b_0}; \ \gamma_1 = \frac{w_{\rm n}^2}{c_2 b_0}. \tag{3.55}$$

With the parameters, the PID controller is re-constructed with the information $\gamma_2 = \alpha_2$ as

$$C(s) = \frac{c_2(s + \gamma_1)(s + \alpha_2)}{s}. \tag{3.56}$$

Its actual parameters are expressed using (3.49) as

$$K_{\rm c} = c_2\gamma_1 + c_2\alpha_2 = \frac{w_{\rm n}^2}{b_0} + \frac{(2\xi w_{\rm n} - \alpha_1)\alpha_2}{b_0} \tag{3.57}$$

$$\tau_{\rm I} = \frac{c_1}{c_0} = \frac{c_2\gamma_1 + c_2\alpha_2}{c_2\gamma_1\alpha_2} = \frac{1}{\alpha_2} + \frac{2\xi w_{\rm n} - \alpha_1}{w_{\rm n}^2} \tag{3.58}$$

$$\tau_{\rm D} = \frac{c_2}{c_1} = \frac{1}{\gamma_1 + \alpha_2} = \frac{2\xi w_{\rm n} - \alpha_1}{\alpha_2(2\xi w_{\rm n} - \alpha_1) + w_{\rm n}^2}. \tag{3.59}$$

Example 3.4 *Assume that a dynamic system has the second order transfer function*

$$G(s) = \frac{2}{(10s + 1)(0.5s + 1)}. \tag{3.60}$$

Design a PID controller using the pole assignment controller design technique with pole–zero cancellation. The closed-loop performance is specified by $\xi = 0.707$ with $w_{\rm n} = 1$. Simulate the closed-loop response and observe the characteristics of input disturbance rejection and the effect of measurement noise.

Solution. *First, we write the transfer function model (3.60) in the form used for the derivation of PID controller parameters:*

$$G(s) = \frac{0.4}{(s + 0.1)(s + 2)}. \tag{3.61}$$

From this equation, we find $b_0 = 0.4$, $\alpha_1 = 0.1$, and $\alpha_2 = 2$. Therefore, in the pole–zero cancellation, we will cancel the pole at $s = -2$. Using (3.57)–(3.59), we calculate the PID controller parameters as

$$K_{\rm c} = 9.07; \ \tau_{\rm I} = 1.814; \ \tau_{\rm D} = 0.3622.$$

With the implementation parameter $\beta = 0.1$, in the closed-loop simulation a unit step signal as the reference signal is introduced at time $t = 0$ and an input step disturbance with

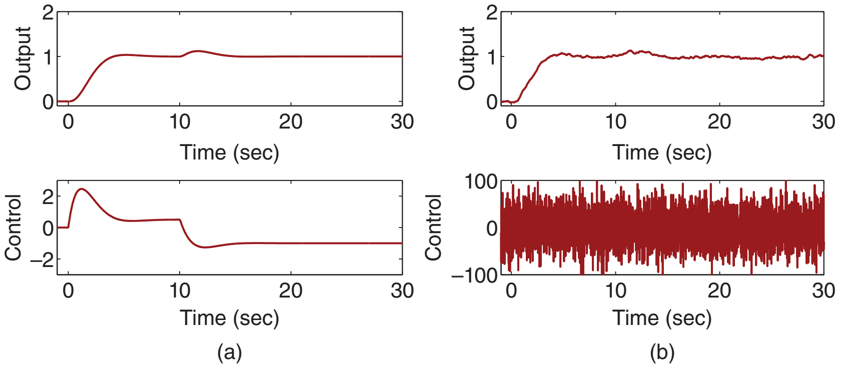


Figure 3.4 Closed-loop response of PID control system (Example 3.4). (a) PID controller in a noise free environment. (b) PID controller in noisy environment.

amplitude of 1.5 *is introduced at* $t = 10$. *Figure 3.4 (a) shows the closed-loop response in a noise-free environment. In order to observe the characteristics of the PID controller in a noisy environment, a band limit white noise with variance* 0.01 *is added to the closed-loop simulation and the closed-loop simulation results are shown in Figure 3.4 (b). In comparison with the noise free case, it is seen that the measurement noise has caused significant fluctuations in the control signal. This observation leads to some caution when we use a PID controller in a noisy environment.*

3.3.3 Analytical Examples for PID Controllers with Filters

As the system becomes more complicated, a derivative filter needs to be used as part of the PID controller design to give an extra degree of freedom in the solution of the controller parameters. This is important in eliminating the approximation introduced by choosing a filter time constant in the implementation stage, hence enhancing the robustness of the PID control system.

The general form of PID controller is defined by the transfer function

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)} \tag{3.62}$$

where l_0 is a parameter that needs to be utilized as part of the derivative filter. However, the industrial control systems are often defined in terms of the PID controller parameters, K_c, τ_I, τ_D, and a filter time constant τ_f. [1]

Thus, we need to find PID controller parameters that lead to a completely identical configuration between the controller $C(s)$ defined by (3.62) and an industrial PID controller. With this in mind, we will choose the PID controller parameters such that

$$C(s) = K_c(1 + \frac{1}{\tau_I s} + \frac{\tau_D s}{\tau_f s + 1}) \tag{3.63}$$

is identical to the PID controller in (3.62). The problem is solved using reverse engineering by expressing (3.63) as

$$C(s) = \frac{K_c(\tau_I s(\tau_f s + 1) + (\tau_f s + 1) + \tau_I \tau_D s^2)}{\tau_I s(\tau_f s + 1)} \tag{3.64}$$

[1] This τ_f was previously linked to the derivative gain τ_D as $\tau_f = \beta\tau_D$.

which should be exactly equal to (3.62). By comparing these two expressions, we obtain,

$$c_2 = \frac{K_c(\tau_I\tau_D + \tau_I\tau_f)}{\tau_I\tau_f} \tag{3.65}$$

$$c_1 = \frac{K_c(\tau_I + \tau_f)}{\tau_I\tau_f} \tag{3.66}$$

$$c_0 = \frac{K_c}{\tau_I\tau_f} \tag{3.67}$$

$$l_0 = \frac{1}{\tau_f} \tag{3.68}$$

We solve the PID controller parameters using these four linear equations and their values are,

$$\tau_f = \frac{1}{l_0} \tag{3.69}$$

$$\tau_I = \frac{c_1}{c_0} - \tau_f \tag{3.70}$$

$$K_c = \tau_I\tau_f c_0 \tag{3.71}$$

$$\tau_D = \frac{c_2\tau_f - K_c\tau_f}{K_c}. \tag{3.72}$$

Example 3.5 *We consider the Padé approximation of irrational transfer function* e^{-ds} *for a first order plus delay model:*

$$G(s) = \frac{K_p e^{-ds}}{\tau_p s + 1} \approx \frac{K_p(-ds + 2)}{(\tau_p s + 1)(ds + 2)}. \tag{3.73}$$

Find the analytical solution of the PID controller parameters with filter using pole-zero cancelation.

Solution. *The first order plus delay model is stable with* $\tau_p > 0$. *The transfer function model (3.73) can also be written in a second order model pole–zero form,*

$$G(s) = \frac{b_1 s + b_0}{(s + \alpha_1)(s + \alpha_2)} \tag{3.74}$$

where $b_1 = -\frac{K_p}{\tau_p}$, $b_0 = \frac{2K_p}{\tau_p d}$.

To find the analytical solution of the PID controller with filter, we will use the pole–zero cancellation technique. For this purpose, if $\frac{1}{\tau_p} < \frac{2}{d}$, *then we choose* $\alpha_1 = \frac{1}{\tau_p}$ *and* $\alpha_2 = \frac{2}{d}$. *In the case of a plant with a dominant time delay when* $\frac{1}{\tau_p} > \frac{2}{d}$, *we let* $\alpha_1 = \frac{2}{d}$ *and* $\alpha_2 = \frac{1}{\tau_p}$.

Because the zero is unstable located at $s = \frac{2}{d}$, *this zero should not be canceled in the controller design. However, we will cancel the pole* $-\alpha_2$, *which has a faster dynamics response. The PID controller structure has the transfer function form,*

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)} \tag{3.75}$$

which is an ideal PID controller with a filter. The filter pole $-l_0$ *will be used in the design. We also assume that the PID with filter has the zeros located at* $-\gamma_1$ *and* $-\alpha_2$, *where* α_2

corresponds to one of the poles in the model. The open- loop transfer function has the quantity,

$$L(s) = G(s)C(s) = \frac{b_1 s + b_0}{(s+\alpha_1)(s+\alpha_2)} \frac{c_2(s+\gamma_1)(s+\alpha_2)}{s(s+l_0)} \\ = \frac{c_2(b_1 s + b_0)(s+\gamma_1)}{s(s+\alpha_1)(s+l_0)}. \tag{3.76}$$

The closed-loop transfer function is

$$T(s) = \frac{L(s)}{1+L(s)} = \frac{c_2(b_1 s + b_0)(s+\gamma_1)}{s(s+\alpha_1)(s+l_0) + c_2(b_1 s + b_0)(s+\gamma_1)}. \tag{3.77}$$

Note that the denominator of (3.77) is a third order polynomial and it has three unknown controller parameters, l_0, c_2 and γ_1. Thus, the desired closed-loop characteristic polynomial $A_{cl}(s)$ must be a third order polynomial with its order to match the denominator of (3.77). To this end, we select

$$A_{cl}(s) = (s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1) \tag{3.78}$$

where $\lambda_1 > 0$ is a positive parameter. As before, we select the damping coefficient $\xi = 0.707$ and the natural frequency w_n to reflect the design requirements, such as closed-loop response time and bandwidth. The extra pole located at $-\lambda_1$ is often chosen to be far away to the left of the dominant poles $-\xi w_n \pm j\sqrt{1-\xi^2} w_n$ ($\xi = 0.707$). For example, we could select $\lambda_1 > 10w_n$.

The closed-loop characteristic polynomial equation is expressed as

$$s(s+\alpha_1)(s+l_0) + c_2(b_1 s + b_0)(s+\gamma_1) = (s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1) \tag{3.79}$$

where the left-hand side of this equation is the closed-loop characteristic polynomial and the right-hand side is the desired one. For an exercise, we work out the exact quantities for both sides as

$$s^3 + (\alpha_1 + l_0 + c_2 b_1)s^2 + (\alpha_1 l_0 + c_2(b_0 + b_1\gamma_1))s + c_2 b_0 \gamma_1 \\ = s^3 + (2\xi w_n + \lambda_1)s^2 + (w_n^2 + 2\lambda_1 \xi w_n)s + \lambda_1 w_n^2. \tag{3.80}$$

By comparing the coefficients of the both sides of the polynomials, three linear equations are obtained,

$$s^2 : \alpha_1 + l_0 + c_2 b_1 = 2\xi w_n + \lambda_1 \tag{3.81}$$

$$s : \alpha_1 l_0 + c_2 b_0 + c_2 b_1 \gamma_1 = w_n^2 + 2\lambda_1 \xi w_n \tag{3.82}$$

$$s^0 : c_2 b_0 \gamma_1 = \lambda_1 w_n^2. \tag{3.83}$$

We solve for $c_2\gamma_1$ based on (3.83):

$$c_2\gamma_1 = \frac{\lambda_1 w_n^2}{b_0}. \tag{3.84}$$

Then, the value of $c_2\gamma_1$ is substituted into (3.82), which becomes,

$$\alpha_1 l_0 + c_2 b_0 = w_n^2 + 2\lambda_1 \xi w_n - \frac{b_1 \lambda_1 w_n^2}{b_0}. \tag{3.85}$$

Note that both (3.81) and (3.85) contain the same pair of unknown variables (α_1, c_2), so we will solve these two together using these two equations. From (3.81), we find the value of l_0,

$$l_0 = -c_2 b_1 + 2\xi w_{\text{n}} + \lambda_1 - \alpha_1. \tag{3.86}$$

Substituting this l_0 into (3.85) and collecting the terms, we find c_2 as

$$c_2 = \frac{-2\xi w_{\text{n}}\alpha_1 - \lambda_1\alpha_1 + \alpha_1^2 + w_{\text{n}}^2 + 2\lambda_1\xi w_{\text{n}} - \frac{b_1\lambda_1 w_{\text{n}}^2}{b_0}}{b_0 - \alpha_1 b_1} \tag{3.87}$$

where we assume that $b_0 - \alpha_1 b_1 \neq 0$. The value of l_0 is found using (3.86) with c_2 given by (3.87). From c_2, we also find γ_1 as

$$\gamma_1 = \frac{\lambda_1 w_{\text{n}}^2}{b_0 c_2}. \tag{3.88}$$

With the parameters c_2, γ_1, and l_0 calculated, the PID controller with filter is re-constructed as,

$$C(s) = \frac{c_2(s+\gamma_1)(s+\alpha_2)}{s(s+l_0)} \tag{3.89}$$

where α_2 corresponds to the location of the pole in the model that we chose to cancel.

Equivalently, (3.89) is expressed in the more general form,

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s+l_0)} \tag{3.90}$$

where c_2 is calculated using (3.87), $c_1 = c_2(\gamma_1 + \alpha_2)$ and $c_0 = c_2\alpha_2\gamma_1$.

Example 3.6 *Given a first order plus delay system with the transfer function*

$$G(s) = \frac{10\text{e}^{-5s}}{10s+1} \tag{3.91}$$

find the PID controller parameters using the pole assignment design technique. The desired closed-loop performance is specified by $\xi = 0.707$, $\lambda_1 = 1$. To understand that the approximation of time delay using the transfer function model causes error between the actual plant and the model used for the design, find the PID controller parameters for $w_{\text{n}} = 0.4$ and then reduce it to $w_{\text{n}} = 0.2$, and simulate the closed-loop performance with a unit step reference signal and disturbance rejection of step signal with amplitude 0.2.

Solution. *The first order plus delay model is approximated using Padé approximation, leading to*

$$G(s) \approx \frac{-s+0.4}{(s+0.1)(s+0.4)}. \tag{3.92}$$

In the design, we cancel the pole from the time delay, and assign the values of α_1 and α_2 as $\alpha_1 = 0.1$ and $\alpha_2 = 0.4$. Also from (3.92), we find $b_1 = -1$ and $b_0 = 0.4$.

In the calculation, we first use $w_{\text{n}} = 0.4$. We calculate the value of c_2 as

$$c_2 = \frac{-2\xi w_{\text{n}}\alpha_1 - \lambda_1\alpha_1 + \alpha_1^2 + w_{\text{n}}^2 + 2\lambda_1\xi w_{\text{n}} - \frac{b_1\lambda_1 w_{\text{n}}^2}{b_0}}{b_0 - \alpha_1 b_1} = 1.9581 \tag{3.93}$$

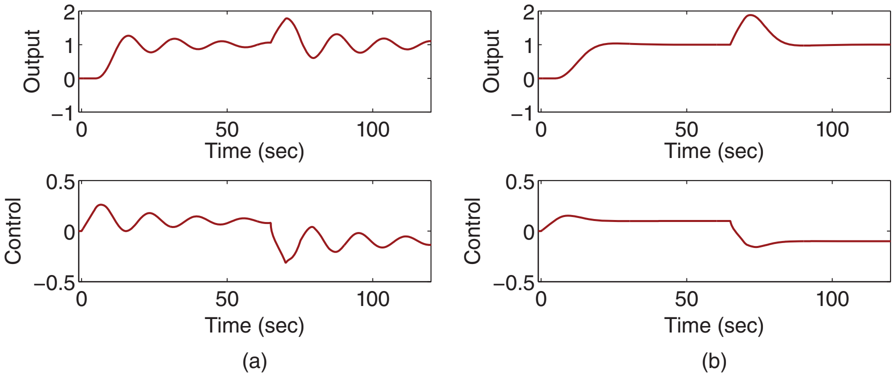


Figure 3.5 Closed-loop response of PID control system (Example 3.6). (a) PID control with $\xi = 0.707$, $\lambda_1 = 1$ and $w_n = 0.4$. (b) PID control with $\xi = 0.707$, $\lambda_1 = 1$ and $w_n = 0.2$.

and l_0 as

$$l_0 = -c_2 b_1 + 2\xi w_n + \lambda_1 - \alpha_1 = 3.4237 \tag{3.94}$$

$c_1 = c_2(\gamma_1 + \alpha_2) = 1.1832$, *and* $c_0 = c_2\alpha_2\gamma_1 = 0.16$. *From these parameters, we calculate the PID controller parameters using (3.69)–(3.72):*

$$K_c = 0.332; \quad \tau_I = 7.1; \quad \tau_D = 1.43; \quad \tau_f = 0.292.$$

Figure 3.5(a) shows the closed-loop responses. It is seen that both control signal and the plant output signal are oscillatory, which is due to the modeling error introduced by the approximation of the time delay. To demonstrate that reducing the desired bandwidth of the closed-loop system will reduce the oscillation, we reduce the performance parameter w_n *from* 0.4 *to* 0.2*. Following the same procedure, we obtain the PID controller parameters for* $w_n = 0.2$*, as*

$$K_c = 0.1793; \quad \tau_I = 8.0323; \quad \tau_D = 1.3375; \quad \tau_f = 0.5581.$$

Figure 3.5(b) shows the closed-loop responses with this reduced w_n*. It is seen indeed that the closed-loop oscillation is eliminated.*

The following example will examine the effect of pole–zero cancellation on reference following and disturbance rejection.

Example 3.7 *AC motor angular position control is required in the applications of high precision machining. Assume that an AC motor has the Laplace transfer function model,*

$$G(s) = \frac{0.05}{s(10s + 1)} \tag{3.95}$$

where the input to the model is the torque current and the output is the angular position of the motor. Design a PID controller with filter using the pole assignment design technique with pole–zero cancellation, where all the desired closed-loop poles are placed at −1*. Verify that the output of the closed-loop control system will track a constant reference input without steady- state error. Also investigate the dynamic response of the closed-loop system to an input disturbance with unknown amplitude of* d_m*.*

Solution. *Write the model in a pole and zero form,*

$$G(s) = \frac{0.005}{s(s+0.1)} \tag{3.96}$$

and the controller in the form,

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s+l_0)}. \tag{3.97}$$

Since we are going to cancel the pole -0.1 *in the AC motor model, the controller is parameterized as*

$$C(s) = \frac{c_2(s+\gamma_1)(s+0.1)}{s(s+l_0)}. \tag{3.98}$$

With this choice, we have the open-loop transfer function,

$$\begin{aligned} L_o(s) = C(s)G(s) &= \frac{c_2(s+\gamma_1)(s+0.1)}{s(s+l_0)} \frac{0.005}{s(s+0.1)} \\ &= \frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0)}. \end{aligned} \tag{3.99}$$

The closed-loop transfer function from the reference signal $R(s)$ *to the output* $Y(s)$ *is*

$$\begin{aligned} \frac{G(s)C(s)}{1+G(s)C(s)} &= \frac{\frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0)}}{1+\frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0)}} \\ &= \frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0)+0.005c_2(s+\gamma_1)}. \end{aligned} \tag{3.100}$$

The denominator of (3.100) is a third order polynomial, which is the actual closed-loop characteristic polynomial determining the closed- loop poles. Thus the number of closed-loop poles is 3*. With all closed-loop poles placed at* -1*, we select the desired closed- loop polynomial as*

$$A_{\mathrm{cl}}(s) = (s+1)^3 = s^3 + 3s^2 + 3s + 1. \tag{3.101}$$

The actual closed-loop characteristic polynomial is made equal to the desired one:

$$s^2(s+l_0) + 0.005c_2(s+\gamma_1) = s^3 + 3s^2 + 3s + 1. \tag{3.102}$$

This equation is expressed as

$$s^3 + l_0 s^2 + 0.005c_2 s + 0.005c_2\gamma_1 = s^3 + 3s^2 + 3s + 1. \tag{3.103}$$

Comparing both sides of the equation, we have

$$s^2 : l_0 = 3 \tag{3.104}$$

$$s : 0.005c_2 = 3 \tag{3.105}$$

$$s^0 : 0.005c_2\gamma_1 = 1. \tag{3.106}$$

These equations give the solution of the PID controller parameters, $l_0 = 3$*,* $c_2 = 600$*, and* $\gamma_1 = \frac{1}{3}$*. The controller* $C(s)$ *is constructed with these parameters and the canceled pole:*

$$C(s) = \frac{600(s+\frac{1}{3})(s+0.1)}{s(s+3)} = \frac{600s^2 + 260s + 20}{s(s+3)}. \tag{3.107}$$

The PID controller parameters are calculated based on $C(s)$ as

$$K_c = 84.4; \tau_I = 12.7; \tau_D = 0.16; \tau_f = 0.33.$$

The transfer function between the reference signal $R(s)$ and the output signal $Y(s)$ is

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}. \tag{3.108}$$

With the pole–zero cancellation, we obtain the closed-loop transfer function as

$$\begin{aligned}\frac{Y(s)}{R(s)} &= \frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0) + 0.005c_2(s+\gamma_1)} \\ &= \frac{3s+1}{(s+1)^3}\end{aligned} \tag{3.109}$$

where we have used the Diophantine equation (3.102).

Assume that the reference is a step signal with the Laplace transform $\frac{1}{s}$, then the output is

$$Y(s) = \frac{3s+1}{(s+1)^3}\frac{1}{s}. \tag{3.110}$$

Applying final value theorem, we calculate the final value of the output as

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0}\frac{3s+1}{(s+1)^3} = 1. \tag{3.111}$$

So, the output of the closed-loop control system will follow the step reference signal with zero steady-state error, and all closed- loop poles for this relationship are all positioned at -1 as the design required.

Figure 3.6 shows the reference responses for the three different controller configurations. Because the controller $C(s)$ has a filter, the order of its denominator equals the order of its numerator and it can be directly implemented on the feedforward path, namely $U(s) = C(s)(R(s) - Y(s))$. The output is shown by line (1) in Figure 3.6. It is seen that there is a large overshoot occurring in the output response. The output response illustrated by line (2) is the closed-loop response by putting both the proportional and derivative terms on the output only. In this implementation approach, there is no overshoot, but the

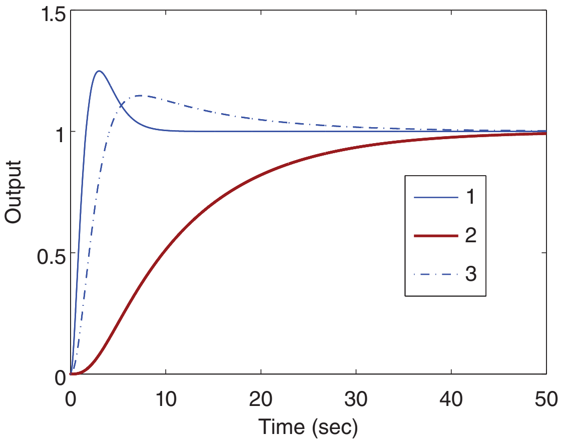


Figure 3.6 Reference response of the PID control system (Example 3.7). Key: line (1) implementation using $C(s)$ directly; line (2) PID controller with both proportional and derivative terms on the output; line (3) PID controller with derivative term on the output.

response is slower than the first approach. When only the derivative term is implemented on the output, the closed-loop response is shown by line (3), which has a smaller amount of overshoot and the closed- loop response speed is faster than the case indicated by line (2).

To investigate the dynamic characteristics of the output response with respect to the input disturbance, we write the closed-loop transfer function between the input disturbance $D_i(s)$ and the output $Y(s)$ as

$$\frac{Y(s)}{D_i(s)} = \frac{G(s)}{1 + G(s)C(s)} = \frac{\frac{0.005}{s(s+0.1)}}{1 + \frac{0.005c_2(s+\gamma_1)}{s^2(s+l_0)}} \tag{3.112}$$

which is, by substituting the values of c_2, γ_1, and l_0, and the Diophantine equation into (3.112),

$$\frac{Y(s)}{D_i(s)} = \frac{0.005s(s+3)}{(s+1)^3(s+0.1)}. \tag{3.113}$$

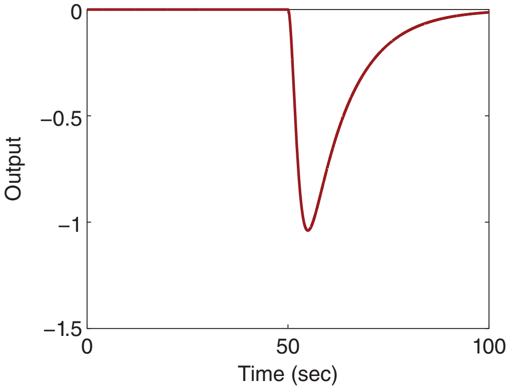
The denominator of (3.113) indicates that there are four poles for this closed-loop relationship: three of them are at the desired location -1 and one of them is at -0.1, which is the plant pole we canceled in the design. This is not good because the plant pole has re-appeared in the closed-loop system for input disturbance response and the magnitude of the plant pole was 10 times smaller than the desired closed-loop poles we selected in the design. This means that the actual closed-loop response speed for input disturbance rejection is much slower than the closed-loop performance we designed for, despite that the dynamic response to reference following has achieved the closed-loop performance we have designed for.

At the steady-state, using the final value theorem, we can verify that

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} s\frac{0.005s(s+3)}{(s+1)^3(s+0.1)}\frac{d_M}{s} = 0. \tag{3.114}$$

Therefore, the PID controller will completely reject the disturbance with zero steady-state error. Figure 3.7 shows the dynamic response to an input disturbance with unknown amplitude. The disturbance occurred at time 50 s, and it is seen that the closed-loop response to the disturbance is very slow and it takes about 50 s to bring the output response back to 0, which is the steady-state in this case.

Figure 3.7 Input disturbance rejection (Example 3.7). All controller structures produce identical disturbance responses.



When the plant is underdamped, the pole–zero cancellation technique we used before will be avoided for the reason that the plant pole that was canceled in the design will re-appear in the closed-loop system for input disturbance rejection.

3.3.4 PID Controller Design without Pole–Zero Cancellation

We assume that the PID controller has the form,

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)} \tag{3.115}$$

and the second order model has the form,

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}. \tag{3.116}$$

Without pole–zero cancellation, the open-loop transfer function is

$$L(s) = C(s)G(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)} \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}. \tag{3.117}$$

The closed-loop transfer function is

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{(c_2 s^2 + c_1 s + c_0)(b_1 s + b_0)}{s(s + l_0)(s^2 + a_1 s + a_0) + (c_2 s^2 + c_1 s + c_0)(b_1 s + b_0)}. \tag{3.118}$$

Note that the denominator of the closed-loop transfer function is a fourth order polynomial and there are four unknown controller parameters to be determined in the design. Thus, the desired closed-loop characteristic polynomial $A_{\text{cl}}(s)$ must be a fourth order polynomial with all zeros on the left half of the complex plane. For instance, we can assume that $A_{\text{cl}}(s)$ has the form,

$$A_{\text{cl}}(s) = (s^2 + 2\xi w_{\text{n}} s + w_{\text{n}}^2)(s + \lambda_1)^2 \tag{3.119}$$

where the dominant poles are $-\xi w_{\text{n}} \pm j\sqrt{1 - \xi^2} w_{\text{n}}$ ($\xi = 0.707$ or 1), and $\lambda_1 \geq w_{\text{n}} > 0$. For simplicity, the desired closed-loop characteristic polynomial $A_{\text{cl}}(s)$ is denoted as $s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0$. To assign the closed-loop poles to the desired locations, we solve the Diophantine equation,

$$s(s + l_0)(s^2 + a_1 s + a_0) + (c_2 s^2 + c_1 s + c_0)(b_1 s + b_0) = s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0. \tag{3.120}$$

By multiplication and collecting terms, we find the exact quantity on the left-hand side of (3.120), which is equal to its right-hand side:

$$\begin{aligned} &s^4 + (b_1 c_2 + a_1 + l_0) s^3 + (b_1 c_1 + b_0 c_2 + a_0 + a_1 l_0) s^2 \\ &\quad + (b_1 c_0 + b_0 c_1 + l_0 a_0) s + b_0 c_0 \\ &\quad = s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0. \end{aligned} \tag{3.121}$$

By comparing both sides of (3.121), a set of linear equations is formed,

$$s^3 : b_1 c_2 + a_1 + l_0 = t_3 \tag{3.122}$$

$$s^2 : b_1 c_1 + b_0 c_2 + a_0 + a_1 l_0 = t_2 \tag{3.123}$$

$$s : b_1 c_0 + b_0 c_1 + l_0 a_0 = t_1 \tag{3.124}$$

$$s^0 : b_0 c_0 = t_0. \tag{3.125}$$

This set of linear equations is expressed in matrix and vector form for convenience of solution,

$$\overbrace{\begin{bmatrix} 1 & b_1 & 0 & 0 \\ a_1 & b_0 & b_1 & 0 \\ a_0 & 0 & b_0 & b_1 \\ 0 & 0 & 0 & b_0 \end{bmatrix}}^{S_y} \begin{bmatrix} l_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} t_3 - a_1 \\ t_2 - a_0 \\ t_1 \\ t_0 \end{bmatrix} \tag{3.126}$$

where we assume that the square matrix S_y (called the Sylvester matrix) is invertible.

Example 3.8 *An inverted pendulum on a cart has external force as input $f(t)$ and its output is the angular position $\theta(t)$. The Laplace transfer function to describe the dynamics of a laboratory pendulum test bed has the form,*

$$G(s) = \frac{-0.1}{(s-1)(s+1)}. \tag{3.127}$$

Design a PID controller for this test bed, where the closed-loop performance is specified by $\xi = 0.707$, and $w_n = \lambda_1 = 10$. Simulate a closed-loop step response using a PID controller with the configuration of the derivative term on output while the proportional and integral terms are on the error signal, and with the configuration of both derivative term and proportional term on the output while only the integral term is on the error signal.

Solution. *For this example, $b_1 = 0$, $b_0 = -0.1$, $a_1 = 0$, and $a_0 = -1$. From (3.126), the linear equation formulated based on the pole assignment controller design technique is*

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.1 & 0 & 0 \\ -1 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & -0.1 \end{bmatrix} \begin{bmatrix} l_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 34 \\ 484 \\ 3414 \\ 10000 \end{bmatrix}. \tag{3.128}$$

Solution of this set of linear equations gives the controller designed using the pole assignment technique: $l_0 = 34.14$, $c_2 = -4834$, $c_1 = -34481$, $c_0 = -100000$. We convert these parameters into PID controller parameters as

$$\tau_f = \frac{1}{l_0} = 0.0293 \tag{3.129}$$

$$\tau_I = \frac{c_1}{c_0} - \tau_f = 0.3155 \tag{3.130}$$

$$K_c = \tau_I \tau_f c_0 = -904.2028 \tag{3.131}$$

$$\tau_D = \frac{c_2 \tau_I \tau_f - K_c \tau_I \tau_f}{K_c \tau_I} = 0.1240. \tag{3.132}$$

Note that the proportional gain K_c is negative, not positive, as one might think. This is because the system is unstable with a pole on the right-hand side of the complex plane and the proportional controller gain K_c does not follow the rules established for stable systems.

The closed-loop step response is simulated for both structures. Figure 3.8 shows both control signal and output signal for a unit step change at time $t = 0$. In comparison, the

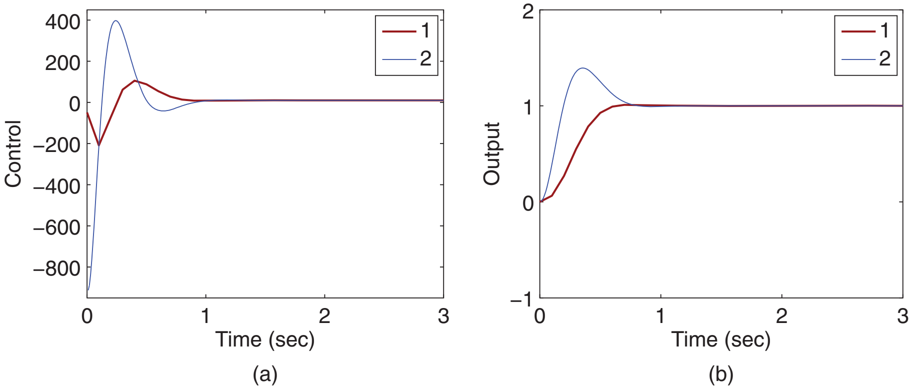


Figure 3.8 Closed-loop response (Example 3.8). (a) Control signal. (b) Output signal. Key: line (1) The IPD controller structure; line (2) The original PID controller structure.

IPD controller structure by implementing the proportional term on the output only significantly reduced the overshoot exhibited in the original structure.

3.3.5 MATLAB Tutorial on Solution of a PID Controller with Filter

Tutorial 3.2 *In this tutorial, we will write a simple MATLAB program to calculate the PID controller parameters. The desired closed-loop characteristic polynomial is specified as*

$$A_{\text{cl}} = s^4 + ac_3 s^3 + ac_2 s^2 + ac_1 s + ac_0$$

and the model is a second order system with the transfer function

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}.$$

Step by Step

1. *Create a new file called PIDplace.m for the MATLAB function.*
2. *Define the input and output variables for the MATLAB function, where $a1$, $a0$, $b1$, and $b0$ are the model parameters and A_{cl} is the desired closed-loop characteristic polynomial. Enter the following program into the file:*

```
function [Kc,tauI,tauD,tauf]=PIDplace(a1,a0,b1,b0,Acl);
```

3. *Find the closed-loop performance parameters. Continue entering the following program into the file:*

```
ac_3=Acl(2);
ac_2=Acl(3);
ac_1=Acl(4);
ac_0=Acl(5);
```

4. *Form the following matrix and vector for the solution of the PID controller parameters, and solve the linear equation. Continue entering the following program into the file:*

   ```
   S_matrix=[1 b1 0 0; a1 b0 b1 0; a0 0 b0 b1; 0 0 0 b0];
   Vec=[ac_3-a1;ac_2-a0;ac_1;ac_0];
   contr_p=inv(S_matrix)*Vec;
   ```

5. *Convert the parameters into a PID controller with derivative filter. Continue entering the following program into the file:*

   ```
   L0=contr_p(1);
   c2=contr_p(2);
   c1=contr_p(3);
   c0=contr_p(4);
   tauf=1/L0;
   tauI=c1/c0-tauf;
   Kc=tauI*tauf*c0;
   tauD=(c2*tauI*tauf-Kc*tauI*tauf)/(Kc*tauI);
   ```

6. *Test this program using the inverted pendulum system presented in Example 3.8, where we use the following codes:*

   ```
   b1=0;
   b0=-0.1;
   a1=0;
   a0=-1;
   Ac1=[1 0.707*20 100];
   Ac2=[1 20 100];
   Acl=conv(Ac1,Ac2);
   [Kc,tauI,tauD,tauf]=PIDplace(a1,a0,b1,b0,Acl)
   ```

3.3.6 Food for Thought

1. Would you design a PD or a PID controller for a first order system?
2. If a system has a pole located on the right half of the complex plane (unstable pole), can you choose a controller zero to cancel this unstable pole?
3. Is it correct to say that the open-loop pole that was cancelled in the design will re-appear at the output response to input disturbance?
4. Is it correct to say that the open-loop pole that was cancelled will re-appear at the output response to output disturbance?
5. Is it correct to say that the open-loop pole that was cancelled will re-appear at the output response to reference signal?
6. How would you deal with time delay when using pole-assignment controller design technique?
7. In the general pole-assignment controller design (without pole-zero cancellation), what is the condition required to obtain unique solutions of controller parameters?

3.4 Resonant Controller Design

The resonant controller, different from the PI or PID controller, incorporates the factor $s^2 + \omega_0^2$ in the denominator of the controller. With the embedded mode, the closed-loop feedback control system is designed to be stable, and at the steady-state the output of the control system will completely track the sinusoidal signal and/or reject a sinusoidal disturbance signal that contains the frequency ω_0 without any steady-state errors (see Section 2.5.3).

3.4.1 Resonant Controller Design

Consider a first order transfer function that is used to describe the dynamics of an AC motor with the form,

$$G(s) = \frac{b}{s+a} = \frac{B(s)}{A(s)} \tag{3.133}$$

where the input is the torque current and the output is velocity. The task is to design a controller $C(s)$ to reject a sinusoidal disturbance with frequency ω_0 (rad s^{-1}). Pole-assignment controller design will be used here.

For the first order system, the controller structure is chosen as

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2} = \frac{P(s)}{L(s)}. \tag{3.134}$$

Here, the denominator of the controller is second order (s^2), the numerator is also chosen to be second order to allow proportional control action. This choice leads to three unknown coefficients c_2, c_1, and c_0 to be determined. The actual closed-loop characteristic polynomial is

$$L(s)A(s) + P(s)B(s) = (s^2 + \omega_0^2)(s + a) + b(c_2 s^2 + c_1 s + c_0). \tag{3.135}$$

This is a third order polynomial. Therefore, the desired closed- loop characteristic polynomial should be third order and the number of desired closed-loop poles should be 3 accordingly. For instance, we can assume that the desired closed-loop characteristic polynomial $A_{\rm cl}(s)$ has the form (ξ, $w_{\rm n}$, $\lambda_1 > 0$)

$$A_{\rm cl}(s) = (s^2 + 2\xi w_{\rm n} s + w_{\rm n}^2)(s + \lambda_1) \tag{3.136}$$

where the dominant poles are $-\xi w_{\rm n} \pm j\sqrt{1-\xi^2} w_{\rm n}$ ($\xi = 1$ or 0.707), and $\lambda_1 \geq w_{\rm n} > 0$. For simplicity, the desired closed-loop characteristic polynomial $A_{\rm cl}(s)$ is denoted as $s^3 + t_2 s^2 + t_1 s + t_0$.

With the pole assignment controller design technique, we let the actual closed-loop characteristic polynomial equal the desired closed-loop characteristic polynomial, which leads to,

$$L(s)A(s) + P(s)B(s) = A_{\rm cl}(s). \tag{3.137}$$

This equation (3.137) is called the Diophantine equation. By substituting the expressions of $L(s)$, $A(s)$, $P(s)$, $B(s)$, and $A_{\rm cl}(s)$ into (3.137), the Diophantine equation is written as

$$s^3 + (a + bc_2)s^2 + (\omega_0^2 + bc_1)s + (a\omega_0^2 + bc_0) = s^3 + t_2 s^2 + t_1 s + t_0. \tag{3.138}$$

In order for the left-hand side of the equation to be equal to the right-hand side of the equation, we have the following linear equations:

$$s^2 : a + bc_2 = t_2 \tag{3.139}$$

$$s : \omega_0^2 + bc_1 = t_1 \tag{3.140}$$

$$s^0 : a\omega_0^2 + bc_0 = t_0. \tag{3.141}$$

Solving these linear equations gives the coefficients of the controller as

$$c_2 = \frac{t_2 - a}{b} \tag{3.142}$$

$$c_1 = \frac{t_1 - \omega_0^2}{b} \tag{3.143}$$

$$c_0 = \frac{t_0 - a\omega_0^2}{b}. \tag{3.144}$$

3.4.2 Steady-state Error Analysis

To show that the output of the closed-loop control system will follow a sinusoidal signal with frequency ω_0, we calculate the closed-loop feedback error signal $E(s) = R(s) - Y(s)$ in relation to the sinusoidal reference signal $R(s)$. Here, the control signal is

$$U(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2} E(s) \tag{3.145}$$

and the output signal is

$$Y(s) = \frac{b}{s+a} U(s) = \frac{b}{s+a} \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2} E(s). \tag{3.146}$$

Noting that $Y(s) = R(s) - E(s)$, by substituting this into (3.146), we obtain

$$(1 + \frac{b}{s+a} \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2}) E(s) = R(s). \tag{3.147}$$

The relationship between the reference signal $R(s)$ and the output signal $Y(s)$ is

$$\begin{aligned} E(s) &= \frac{(s+a)(s^2+\omega_0^2)}{(s^2+\omega_0^2)(s+a) + b(c_2 s^2 + c_1 s + c_0)} R(s) \\ &= \frac{(s+a)(s^2+\omega_0^2)}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} R(s) \end{aligned} \tag{3.148}$$

where we have used the Diophantine equation (3.137). When the reference signal $r(t) = R_m \sin(\omega_0 t)$, its Laplace transform is $R(s) = \frac{R_m \omega_0}{s^2+\omega_0^2}$. From (3.148), we have the Laplace transform of the feedback error signal,

$$\begin{aligned} E(s) &= \frac{(s+a)(s^2+\omega_0^2)}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} \frac{R_m \omega_0}{s^2+\omega_0^2} \\ &= \frac{(s+a) R_m \omega_0}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} \end{aligned} \tag{3.149}$$

where we have canceled the factor $s^2 + \omega_0^2$. Since the denominator of (3.149) contains all zeros on the left half of the complex plane, by applying final value theorem, we obtain

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} s \frac{(s+a)R_m\omega_0}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} = 0. \tag{3.150}$$

Because $e(t) = r(t) - y(t)$ and $\lim_{t\to\infty} e(t) = 0$, we conclude that the output $y(t)$ will converge to the reference signal $r(t)$.

To show that the closed-loop control system will completely reject a sinusoidal disturbance, we will find the relationship between the input disturbance and the output. Here, the transfer function between the input disturbance $D_i(s)$ and the output $Y(s)$ is

$$\begin{aligned}\frac{Y(s)}{D_i(s)} &= \frac{G(s)}{1+G(s)C(s)}\\ &= \frac{(s^2+\omega_0^2)b}{(s^2+\omega_0^2)(s+a) + b(c_2s^2 + c_1s + c_0)}\\ &= \frac{b(s^2+\omega_0^2)}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)}.\end{aligned} \tag{3.151}$$

Assume that the disturbance signal is a sinusoidal signal $d_i(t) = d_m \sin(\omega_0 t)$ with unknown amplitude d_m, and it has the Laplace transform $D_i(s) = \frac{d_m\omega_0}{s^2+\omega_0^2}$. Thus, the output in response to the input disturbance $D_i(s)$ is

$$\begin{aligned}Y(s) &= \frac{b(s^2+\omega_0^2)}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} D_i(s)\\ &= \frac{b(s^2+\omega_0^2)}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} \frac{d_m\omega_0}{s^2+\omega_0^2}\\ &= \frac{bd_m\omega_0}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)}.\end{aligned} \tag{3.152}$$

The zeros of the denominator are all on the left half of the complex plane as the parameters $\lambda_1 > 0$, $\xi = 0.707$, or 1 and $w_n > 0$. Thus, by applying final value theorem, we obtain

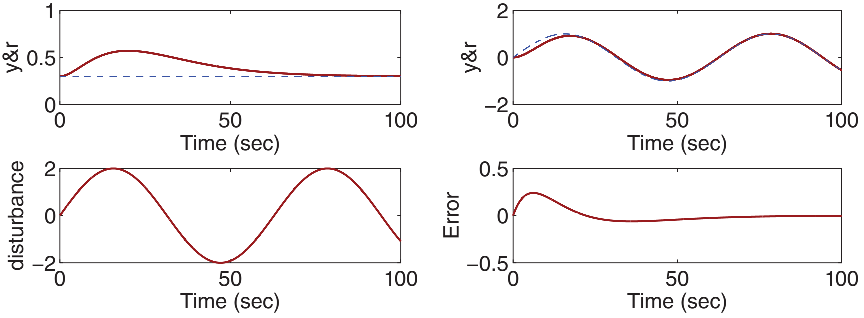
$$\begin{aligned}\lim_{t\to\infty} y(t) &= \lim_{s\to 0} sY(s)\\ &= \lim_{s\to 0} \frac{sbd_m\omega_0}{(s^2 + 2\xi w_n s + w_n^2)(s+\lambda_1)} = 0\end{aligned} \tag{3.153}$$

Therefore, the output $y(t)$ in response to the input disturbance $d_i(t)$ is zero at the steady-state. This means that the input disturbance $d_i(t)$ will be completely rejected by the closed- loop feedback control system.

Example 3.9 *For an AC motor with parameters* $a = 0.01$ *and* $b = 0.05$*, if the disturbance is at frequency* $\omega_0 = 0.1$*, and all three desired closed-loop poles are selected as* -0.1*, design the feedback controller for this AC motor so that the closed-loop control signal will reject the sinusoidal disturbance signal.*

Solution. *The desired closed-loop characteristic polynomial is*

$$A_{cl}(s) = (s+0.1)^3 = s^3 + 0.3s^2 + 0.03s + 0.001. \tag{3.154}$$



(a) (b)

Figure 3.9 Closed-loop response of resonant control (Example 3.9). (a) Disturbance rejection. (b) Reference following.

Here, $t_2 = 0.3$, $t_1 = 0.03$, and $t_0 = 0.001$. From the equations given in (3.138), we find the controller parameters as

$$c_2 = \frac{t_2 - a}{b} = \frac{0.3 - 0.01}{0.05} = 5.8 \tag{3.155}$$

$$c_1 = \frac{t_1 - \omega_0^2}{b} = \frac{0.03 - 0.01}{0.05} = 0.4 \tag{3.156}$$

$$c_0 = \frac{t_0 - a\omega_0^2}{b} = \frac{0.001 - (0.01)^2}{0.05} = 0.018. \tag{3.157}$$

First, we simulate the closed-loop disturbance rejection by operating the system at the steady-state value 0.3 *and at time $t = 0$, inject a disturbance signal $d_{in}(t) = 2\sin(0.1t)$. Figure 3.9(a) shows the output response to the disturbance signal. It is seen that the disturbance is completely rejected and the output returns to the reference signal. The same control system will also follow a sinusoidal reference signal, as shown in Figure 3.9(b), which shows that the output tracks a sinusoidal signal $r(t) = \sin(0.1t)$. Thus, the control system we designed for sinusoidal disturbance rejection will automatically follow a sinusoidal signal with the same frequency.*

3.4.3 Pole–Zero Cancellation in the Design of a Resonant Controller

Example 3.10 *An electric circuit has a second order transfer function:*

$$G(s) = \frac{1}{(s+1)(s+6)}. \tag{3.158}$$

This system has a sinusoidal disturbance with frequency $\omega_0 = 1$. Design a resonant controller to completely reject this disturbance at the steady-state. All desired closed-loop poles are positioned at -3.

Solution. *Since the positions of the closed-loop poles -3 are on the right-hand side of the fast pole of the system -6, we will choose to use the pole–zero cancellation technique in the design. Cancellation of this pole will not significantly alter the response speed of the input disturbance. However, because we do not know the exact order of the controller,* *we will start the design using the lowest order possible and see if we can find the unique controller parameter solutions.*

The simplest controller structure for a resonant controller is

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2}. \tag{3.159}$$

This structure is chosen because we know that in order to reject the sinusoidal disturbance the denominator of the controller will contain the factor $s^2 + \omega_0^2$, and the order of the numerator is to match the order of the denominator.

To implement the pole–zero cancellation, the controller is re- written as

$$C(s) = \frac{c_2(s + \gamma_1)(s + 6)}{s^2 + \omega_0^2}. \tag{3.160}$$

The open-loop transfer with the plant and the controller is

$$L_o(s) = G(s)C(s) = \frac{1}{(s+1)(s+6)} \frac{c_2(s + \gamma_1)(s + 6)}{s^2 + \omega_0^2}. \tag{3.161}$$

The closed-loop transfer function between the reference signal $R(s)$ and the output $Y(s)$ is

$$\begin{aligned} \frac{Y(s)}{R(s)} &= \frac{G(s)C(s)}{1 + G(s)C(s)} \\ &= \frac{c_2(s + \gamma_1)}{c_2(s + \gamma_1) + (s+1)(s^2 + \omega_0^2)}. \end{aligned} \tag{3.162}$$

Here, from (3.162), we can see that the denominator of the closed-loop transfer function is a third order polynomial and therefore there are three closed-loop poles in the system. When we look closely at (3.162), there are only two unknown controller coefficients, c_2 and γ_1. If we were to examine the Diophantine equation,

$$c_2(s + \gamma_1) + (s+1)(s^2 + \omega_0^2) = s^3 + t_2 s^2 + t_1 s + t_0 \tag{3.163}$$

where the right-hand side of the equation is the desired closed- loop polynomial, there are three linear equations in order to match the actual closed-loop poles to the desired closed-loop poles. Thus, there is no unique solution to determine the two unknown controller coefficients c_2 and γ_1 when there are three linear equations. This means that the controller structure is too simple for the system.

We increase the controller complexity by choosing

$$C(s) = \frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{(s^2 + \omega_0^2)(s + l_0)} \tag{3.164}$$

where both denominator and numerator have the same order of 3 *for maintaining proportional control and practical implementation. With the pole–zero cancellation, we re-write the controller structure as*

$$C(s) = \frac{c_3(s^2 + \gamma_1 s + \gamma_0)(s + 6)}{(s^2 + \omega_0^2)(s + l_0)}. \tag{3.165}$$

The open-loop transfer function is

$$L_0(s) = G(s)C(s) = \frac{1}{(s+1)(s+6)} \frac{c_3(s^2 + \gamma_1 s + \gamma_0)(s + 6)}{(s^2 + \omega_0^2)(s + l_0)}. \tag{3.166}$$

With the cancellation of the factor $s + 6$, *the closed-loop transfer function is*

$$\frac{Y(s)}{R(s)} = \frac{c_3(s^2 + \gamma_1 s + \gamma_0)}{c_3(s^2 + \gamma_1 s + \gamma_0) + (s+1)(s^2 + \omega_0^2)(s + l_0)}. \tag{3.167}$$

From (3.167), it is seen that the closed-loop characteristic polynomial is fourth order and there are four closed-loop poles, and there are four independent unknown controller parameters: c_3, γ_1, γ_0, *and* l_0. *As per the design specification, all closed-loop poles are positioned at* -3, *hence the desired closed-loop characteristic polynomial is* $(s + 3)^4$. *This leads to the Diophantine equation:*

$$c_3 s^2 + c_3\gamma_1 s + c_3\gamma_0 + (s+1)(s^2 + \omega_0^2)(s + l_0) = (s+3)^4 \tag{3.168}$$

which is

$$\begin{aligned} &s^4 + (1 + l_0)s^3 + (\omega_0^2 + l_0 + c_3)s^2 + (\omega_0^2 + l_0\omega_0^2 + c_3\gamma_1)s + l_0\omega_0^2 + c_3\gamma_0 \\ &\quad = s^4 + 12s^3 + 54s^2 + 108s + 81. \end{aligned} \tag{3.169}$$

Comparing both sides of the equation (3.169), we obtain the linear equations:

$$s^3 : 1 + l_0 = 12 \tag{3.170}$$

$$s^2 : \omega_0^2 + l_0 + c_3 = 54 \tag{3.171}$$

$$s : \omega_0^2 + l_0\omega_0^2 + c_3\gamma_1 = 108 \tag{3.172}$$

$$s^0 : l_0\omega_0^2 + c_3\gamma_0 = 81. \tag{3.173}$$

Solving these linear equations gives the values of the controller parameters: $l_0 = 11$, $c_3 = 54 - \omega_0^2 - l_0 = 42$, $\gamma_1 = (108 - \omega_0^2 - l_0\omega_0^2)/c_3 = 2.2857$, *and* $\gamma_0 = (81 - l_0\omega_0^2)/c_3 = 1.667$. *With these parameters, we re-construct the resonant controller as*

$$\begin{aligned} C(s) &= \frac{c_3(s^2 + \gamma_1 s + \gamma_0)(s+6)}{(s^2 + \omega_0^2)(s + l_0)} \\ &= \frac{42s^3 + 348s^2 + 657s + 486}{s^3 + 11s^2 + s + 11}. \end{aligned} \tag{3.174}$$

The closed-loop output response to an input disturbance $d_i(t) = 2\sin(t)$ *is simulated with the system operating at the steady state of* 1, *the disturbance entering the system at* $t = 0$. *Figure 3.10 shows the output response to the disturbance input. It is seen that it takes about three seconds for the closed-loop system to completely reject the sinusoidal disturbance.*

With the resonant controller, as shown in Section 2.5.3, the complementary sensitivity function is required to satisfy $T(\omega) = 1$ at the frequency $\omega = \omega_0$. Thus, this indicates that the bandwidth of the closed-loop resonant control system is to be greater than ω_0, at least. If ω_0 is large, from the analysis given in Chapter 2, this implies in general that a resonant control system requires a better mathematical model for robustness and better sensors for noise attenuation.

3.4.4 Food for Thought

1. A periodic reference signal has a known frequency 50 Hz. What is the frequency parameter ω_0 you would use in the design of resonant controller so that the reference signal could be tracked without steady-state error?

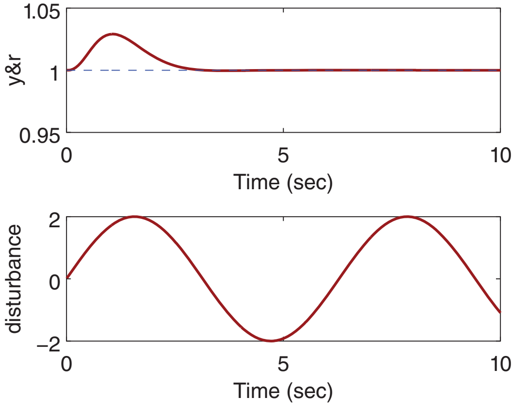


Figure 3.10 Sinusoidal input disturbance rejection (Example 3.10).

2. If a resonant controller is designed to track a sinusoidal signal with frequency ω_0 without steady-state error, will the same resonant controller reject a periodic input disturbance signal with the same frequency?
3. If the sinusoidal reference signal is known to have the frequency ω_0 and the disturbance is known to have a frequency $3\omega_0$, is it correct to choose the denominator of the resonant controller to contain the factor $(s^2 + \omega_0^2)(s^2 + 9\omega_0^2)$?
4. If the reference signal is a sinusoidal signal with frequency ω_0 plus a constant, what would be the key factors in the denominator of the resonant controller you would choose so that the closed-loop control system will track this reference signal?

3.5 Feedforward Control

Feed forward control is widely used in combination with either PID or resonant controllers. The starting point for feedforward compensation is that the feedforward variables used are either directly measured or estimated. Their effect is captured at the control signal from which it can be subtracted. It has various forms; however, the basic idea remains the same.

3.5.1 Basic Ideas about Feedforward Control

Assume that the output of a dynamic system is described by the Laplace transform:

$$Y(s) = G(s)U(s) + G_d(s)D_o(s) \tag{3.175}$$

where $U(s)$ is the Laplace transform of the control signal and $D_o(s)$ is the disturbance that is measured and used in the feedforward compensation. To introduce feedforward compensation, (3.175) is re-written as

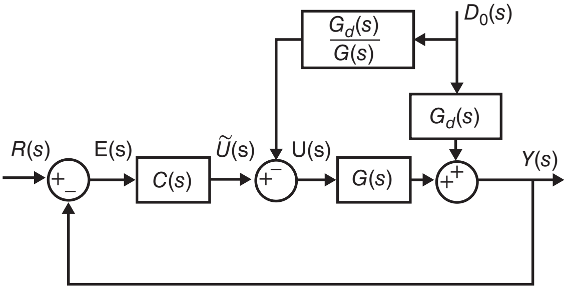
$$\begin{aligned} Y(s) &= G(s)(U(s) + \frac{G_d(s)}{G(s)}D_o(s)) \\ &= G(s)\tilde{U}(s) \end{aligned} \tag{3.176}$$

where an intermediate control signal $\tilde{U}(s)$ is defined as

$$\tilde{U}(s) = U(s) + \frac{G_d(s)}{G(s)}D_o(s).$$

Based on (3.176), a PID controller $C(s)$ is designed using the transfer function $G(s)$ to generate the intermediate control signal $\tilde{U}(s)$ from which the effect of the disturbance

Figure 3.11 Block diagram of the feedback and feedforward control system.



$D_\mathrm{o}(s)$ is subtracted. This explicitly leads to the actual control signal $U(s)$ to be expressed as

$$\begin{aligned} U(s) &= \tilde{U}(s) - \frac{G_\mathrm{d}(s)}{G(s)} D_\mathrm{o}(s) \\ &= C(s)(R(s) - Y(s)) - \frac{G_\mathrm{d}(s)}{G(s)} D_\mathrm{o}(s). \end{aligned} \tag{3.177}$$

Clearly, the assumption is that the transfer function $\frac{G_\mathrm{d}(s)}{G(s)}$ is stable and realizable in addition to $D_\mathrm{o}(s)$ being measured. Figure 3.11 illustrates the feedback and feedforward control system. As an example, if we assume that $G(s)$ and $G_\mathrm{d}(s)$ are represented by the following first order plus delay transfer functions:

$$G(s) = \frac{0.5\mathrm{e}^{-5s}}{2s+1}; \quad G_\mathrm{d}(s) = \frac{0.1\mathrm{e}^{-10s}}{6s+1}$$

then the control signal $U(s)$ is expressed as

$$U(s) = C(s)(R(s) - Y(s)) - \frac{0.2(2s+1)e^{-5s}}{6s+1} D_\mathrm{o}(s).$$

Because the time delay of the disturbance model is larger than the time delay of the plant model, the transfer function $\frac{G_\mathrm{d}(s)}{G(s)}$ is realizable, hence the feedforward compensation can be implemented for the application.

3.5.2 Three Springs and Double Mass System

To illustrate how to use a feedforward controller in conjunction with a PID controller, we will examine the PID control system design for a three springs and double mass system.

A three springs and double mass system Tongue (2002) is illustrated in Figure 3.12. In this figure, the two blocks are with mass M_1 and M_2, and there are two spring constants α_1 for the right and left springs, α_2 for the middle spring. The manipulated variables are the two applied forces u_1 and u_2 and the output variables are the distances of the mass block movement y_1 corresponding to block one and y_2 corresponding to block two.

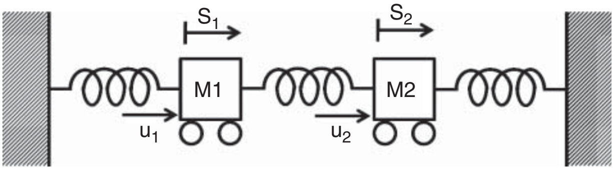


Figure 3.12 Three springs and double mass system.

We consider the simplified spring force f_1 where $f_1 = \alpha_1 d_1$ and d_1 is the distance of the mass movement, and apply Newton's law [2] to the first block to obtain the following equation:

$$u_1(t) - \alpha_1 y_1(t) - \alpha_2(y_1(t) - y_2(t)) = M_1 \frac{d^2 y_1(t)}{dt^2} \tag{3.178}$$

where the first term represents the first external force, the second and third terms represent the forces from spring 1 and spring 2, respectively. The right-hand side of (3.178) is the multiplication of the mass and acceleration of block 1. Similarly, by applying Newton's law to the second block, we obtain

$$u_2(t) - \alpha_1 y_2(t) + \alpha_2(y_1(t) - y_2(t)) = M_2 \frac{d^2 y_2(t)}{dt^2} \tag{3.179}$$

where the first term represents the second external force, and the second and the third terms are the forces generated from spring 3 and spring 2. The right-hand side of (3.179) is the multiplication of mass and acceleration of block 2.

To have a clear view of the dynamics, we re-write (3.178) and (3.179) as

$$\frac{d^2 y_1(t)}{dt^2} = -\frac{\alpha_1 + \alpha_2}{M_1} y_1(t) + \frac{\alpha_2}{M_1} y_2(t) + \frac{u_1(t)}{M_1} \tag{3.180}$$

$$\frac{d^2 y_2(t)}{dt^2} = -\frac{\alpha_1 + \alpha_2}{M_2} y_2(t) + \frac{\alpha_2}{M_2} y_1(t) + \frac{u_2(t)}{M_2}. \tag{3.181}$$

We have two second order systems for the three springs and double mass system. Additionally, there are interactions between the two systems.

As in Tongue (2002), the physical parameters are $M_1 = 2$ kg, $M_2 = 4$ kg, $\alpha_1 = 40$ N m^{-1} and $\alpha_2 = 100$ N m^{-1}. As an exercise, the parameter is varied to $\alpha_1 = 200$ N m^{-1}.

Example 3.11 *In this example, we assume that the second block is not controlled. The force from the second block acts as a disturbance on the closed-loop control of the first block. Find the natural frequency of the uncontrolled system and choose all closed-loop poles in relation to this natural frequency when designing the PID controller.*

Solution. *The Laplace transforms of the differential equations (3.180) and (3.181) are obtained as*

$$Y_1(s) = \frac{1/M_1}{s^2 + \frac{\alpha_1+\alpha_2}{M_1}}(\alpha_2 Y_2(s) + U_1(s)) \tag{3.182}$$

$$Y_2(s) = \frac{1/M_2}{s^2 + \frac{\alpha_1+\alpha_2}{M_2}}(\alpha_2 Y_1(s) + U_2(s)) \tag{3.183}$$

where we have assumed zero initial conditions of the variables $y_1(t)$ and $y_2(t)$. With these two transfer function models, we can build the Simulink simulator for the three springs double mass system using transfer function blocks.

To find the natural frequency for the first mass, (3.182) is considered, where the Laplace transform $Y_2(s)$ is the external disturbance. Thus, the open-loop system has a pair of poles on the imaginary axis, determined by the solution of the polynomial equation:

$$s^2 + \frac{\alpha_1 + \alpha_2}{M_1} = 0$$

[2] $F = Ma$, where F is the total force, M is the mass and a is the acceleration.

which is

$$s_{1,2} = \pm j\sqrt{\frac{\alpha_1 + \alpha_2}{M_1}} = \pm j\sqrt{\frac{40 + 100}{2}} = \pm j8.3667.$$

The natural frequency is 8.3667 N m^{-1} kg^{-1}.

PID Controller with Filter

The PID controller with filter is designed based on the transfer function model (3.182) using the MATLAB program PIDplace.m (see Tutorial 3.2). The term $\alpha_2 Y_2(s)$ is considered as disturbance. Two pairs of complex poles are used in the pole assignment PID controller design. With damping coefficient $\xi = 0.707$, the parameter w_n is used as a performance tuning parameter. Because of the interactions between the two outputs, the closed-loop dynamics from the first mass block are more complex. As a result, there is a minimum w_n required for the stabilization of the closed-loop system. Through simulation studies, this minimum w_n is approximately five times the natural frequency of the original system. This behavior is similar to that of controlling unstable systems where a minimum controller gain is required for robustness of the feedback system.

By selecting $w_n = 6 \times 8.3667 = 50.2$ and $\xi = 0.707$ for the two pairs of complex poles, the program PIDplace.m calculates the PID controller parameters as

$$K_c = 4269.8; \quad \tau_I = 0.0477; \quad \tau_D = 0.0260; \quad \tau_f = 0.0070.$$

Closed-loop Simulation Studies

In the first simulation, we assume $u_2(t) = 0$ for $t > 0$. However, because the second mass is connected to the first mass, the output $y_2(t)$ is not zero. With sampling interval $\Delta t = 0.0001$ (s), a step reference signal with amplitude 0.1 is added to the closed-loop control. Figure 3.13(a) shows the control signal used to move the first mass from the origin to 0.1 m and Figure 3.13(b) shows the output signal. It is clearly seen from Figure 3.13(a) that the second mass generates a sinusoidal disturbance and the feedback control signal tries to compensate it. The effect of the sinusoidal disturbance is hardly noticeable.

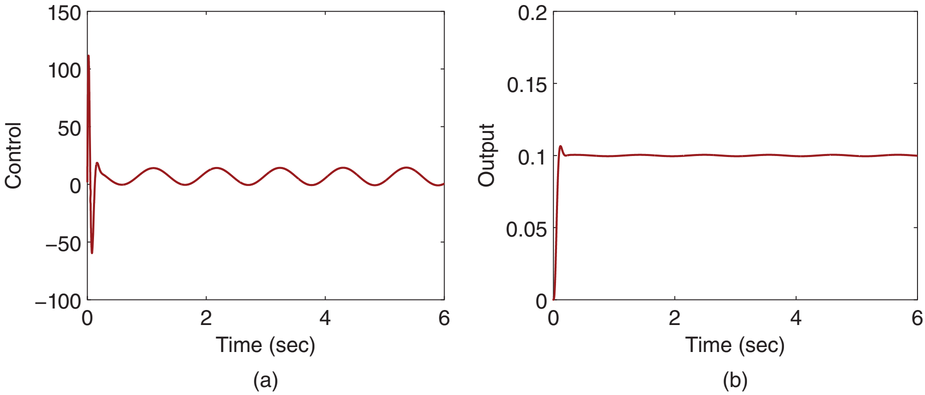


Figure 3.13 Closed-loop responses for three springs and double mass system with $u_2 = 0$ (Example 3.11). (a) Control signal. (b) Output signal.

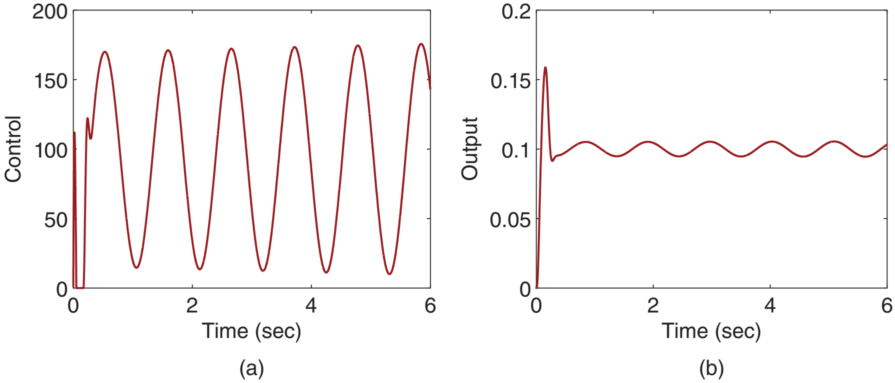


Figure 3.14 Closed-loop responses for three springs and double mass system with $u_2 = -120$ (Example 3.11). (a) Control signal. (b) Output signal.

Another simulation scenario is presented. Assume that there is a constant negative force acting on the second mass block, which is $u_2(t) = -120$. Additionally, the control signal $u_1(t)$ is constrained to positive values ($u_1(t) \geq 0$). Figure 3.14 shows the closed-loop responses for the three springs and double mass system with $u_2 = -120$. The control signal (see Figure 3.14(a)) still tries to reduce the effect of the disturbance, however because of its larger amplitude, the effect of the disturbance can be seen from the output response (see Figure 3.14(b)).

Example 3.12 *To overcome the sinusoidal disturbance generated by the second mass, we will design a disturbance feedforward control together with the PID feedback control and simulate the closed-loop control performance. We assume that the output $y_2(t)$ is measured.*

Solution. *To design the feedforward together with the PID controller, we consider the physical equation (3.180) that describes the dynamics of the first mass block:*

$$\begin{aligned}\frac{d^2y_1(t)}{dt^2} &= -\frac{\alpha_1+\alpha_2}{M_1}y_1(t) + \frac{1}{M_1}(\alpha_2 y_2(t) + u_1(t))\\ &= -\frac{\alpha_1+\alpha_2}{M_1}y_1(t) + \frac{1}{M_1}\tilde{u}_1(t) \qquad (3.184)\end{aligned}$$

where we have defined an intermediate control variable

$$\tilde{u}_1(t) = \alpha_2 y_2(t) + u_1(t).$$

Now, the PID controller presented in Example (3.11) is unchanged and is used here to calculate the intermediate control signal $\tilde{u}_1(t)$. The feedforward control is then realized as

$$u_1(t) = \tilde{u}_1(t) - \alpha_2 y_2(t). \qquad (3.185)$$

The PID controller with disturbance feedforward compensation is evaluated with the case of $u_2 = -120$. Figure 3.15(a) shows the closed-loop control signal, from which we can see that the control signal captures the oscillations caused by the second mass block. More importantly, because of the characteristics of the control signal, the output $y_1(t)$ no

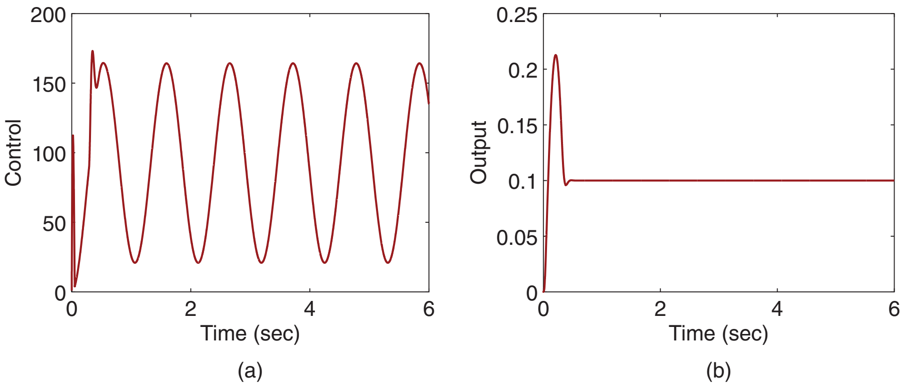


Figure 3.15 Closed-loop responses with disturbance feedforward control for three springs and double mass system with $u_2 = -120$ (Example 3.12). (a) Control signal. (b) Output signal.

longer has the periodic oscillations when compared with that shown in Figure 3.14(b). This means that the periodic disturbance is completely eliminated.

One needs to be cautious when using feedforward compensation. The effectiveness of feedforward compensation is dependent on the accuracy of the model used in the compensation. It is reasonable to say that this dependence is more severe than the cases when feedback control strategies are used. Here, the parameter α_2 plays an important role. To illustrate, we consider the following three cases:

$$u_1(t) = \begin{cases} \tilde{u}_1(t) - 0.5\alpha_2 y_2(t) & \text{(a)} \\ \tilde{u}_1(t) - 1.5\alpha_2 y_2(t) & \text{(b)} \\ \tilde{u}_1(t) + 0.5\alpha_2 y_2(t) & \text{(c).} \end{cases} \tag{3.186}$$

Figure 3.16 compares the output responses for the three feedforward compensations. It is clearly seen that the periodic disturbance is no longer completely compensated and the responses are different from each other in the steady-state behavior. Also, the worse case is when the sign of the compensation was wrong.

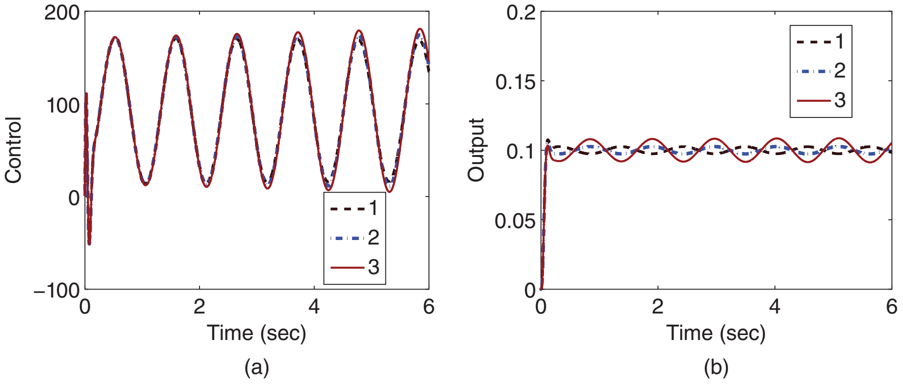


Figure 3.16 Comparison between the closed-loop output responses when using different feedforward compensations (Example 3.12). (a) Control signal. (b) Output signal. Key: line (1) feedforward using (3.186(a)); line (2) feedforward using (3.186(b)); line (3) feedforward using (3.186 (c)).

3.5.3 Food for Thought

1. In the feedforward control, if the feedforward variable is bounded, as shown in the three springs and double mass system, would you believe that the feedforward control will not cause closed-loop instability?
2. In most of the applications when using feedforward compensation, the feedforward variable is captured as part of the control signal as shown in the three springs and double mass system so it is conveniently subtracted. Can we extend this idea of feedforward compensation to the reference signal?
3. What are the main advantages of using feedforward control?
4. What are the drawbacks of using feedforward control?

3.6 Summary

This chapter has discussed one of the most widely used control system design methods for PID and resonant controllers. The central ideas behind the pole-assignment controller design are to specify the desired closed-loop performance via the locations of the closed-loop poles and to match the desired closed-loop poles with the actual closed-loop poles by finding the solutions of a polynomial equation. These methods are conceptually and computationally simple, and the design methodology can be extended to various controller structures and models. The other important aspects are summarized as follows.

- For a first order model, we can design a P controller or a PI controller or a resonant controller.
- For a second order model, we can design a PD controller or a PID controller or a resonant controller with a first order filter.
- For a higher order model, the structure of the controller needs to be selected so that the controller parameters can be determined uniquely.
- We can use the pole-zero cancellation technique for the purpose of getting simple analytical solutions. We can only cancel the poles corresponding to fast dynamic response. The canceled poles will re-appear in the closed-loop system in response to input disturbance. We can not cancel unstable poles or zeros in the design.
- For a higher order system, model order reduction is used to obtain first order or second order model for the PID controller design.
- The locations of the desired closed-loop poles are chosen, and they need to be adjusted to suit the requirement of closed-loop response speed for reference following, disturbance rejection, noise attenuation and robustness of the closed-loop system. It will be desirable to adjust them according to the Nyquist diagram and the sensitivity analysis presented in Chapter 2.

3.7 Further Reading

1. Internal model control is introduced in the book Morari and Zafiriou (1989).
2. Pole-assignment PID controller design together with several other design methods are introduced in Goodwin et al. (2000),Cominos and Munro (2002).

3. Robust pole-assignment controller design is proposed in Soh et al. (1987), Nurges (2006), Wang et al. (2009).
4. Pole-assignment in combination with sensitivity shaping is proposed in Langer and Landau (1999).
5. Self-tuning control using pole-assignment design is one of the original self-tuning controllers Wellstead et al. (1979).
6. Pole-assignment PID controller design with dominant pole consideration and analysis is introduced in Zítek et al. (2013).
7. With approximation of time-delay and specification of a desired closed-loop time constant, PI and PID controllers are found analytically using direct synthesis method for disturbance rejection (Chen and Seborg (2002)).
8. A simple relay nonlinear PD controller is proposed for controlling high precision motion system with frictions (Zheng et al. (2018)).
9. Current control in electrical AC drives and power converters typically uses PI control structures in the synchronous reference frame ($d-q$ reference frame) because their reference signals are constant (Wang et al. (2015)). If the stationary reference frame is used, then current control for the AC drives and power converters utilizes resonant controllers because their reference signals are sinusoidal (Wang et al. (2015)).

Problems

3.1 *Design PI controllers for the systems with the transfer function $G(s) = \frac{b}{s+a}$, where the parameters a and b are given as follows:*

1. $a = 0.1$ and $b = 3$;
2. $a = 10$ and $b = -0.1$;
3. $a = -3$ and $b = 1$.

In the design, the desired closed-loop characteristic polynomial is selected as $s^2 + 2\xi w_n s + w_n^2$ with $\xi = 0.707$ and choosing the natural frequency parameter w_n as $|a|$, $5|a|$ and $10|a|$. Compare the proportional controller gain K_c and the integral time constant τ_I for the three choices of the bandwidth. What are your observations?

3.2 *PD controllers are designed based on a second order model, which has the following form:*

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}$$

1. Find the PD controllers with a derivative filter for the following systems:
   (a) $b_1 = 0$, $b_0 = 3$, $a_1 = 0$, $a_0 = 4$.
   (b) $b_1 = -1$, $b_0 = 1$, $a_1 = 0.2$, $a_0 = 1$.
   (c) $b_1 = 0.1$, $b_0 = -1$, $a_1 = 1$, $a_0 = 0$.
   where all the desired closed-loop desired closed-loop polynomial is chosen to be $(s^2 + 2\xi w_n s + w_n^2)(s + 2w_n)$, where $\xi = 0.707$ and $w_n = 2$.
2. Write a Simulink simulation program to evaluate the performance of the three closed-loop PD control systems with a unit step reference signal using sampling

interval $\Delta t = 0.01$ (sec), where the derivative control is implemented on the output only. Will the closed-loop output follow the reference signal without steady-state error?

3.3 *In the design of a PI controller, it is often to approximate a higher order system with a first order transfer function where the dominant time constant is identified (largest time constant) and kept, while the other smaller time constants are neglected. This follows the procedure of writing the transfer function using the form*

$$G(s) = \frac{K}{(\tau s + 1)(\epsilon s + 1)} \approx \frac{K}{\tau s + 1}$$

where we assume that $\epsilon << \tau$.

1. *Design PI controllers for the following systems with appropriate approximation of the complex dynamics. The desired closed-loop performance is specified using the polynomial $s^2 + 2\xi w_n s + w_n^2$, where $\xi = 0.707$ and w_n is $5/\tau$.*
   (a) $G(s) = \frac{0.1}{(s+0.2)(s+3)}$;
   (b) $G(s) = \frac{-5}{(s+0.1)(s+6)^2}$;
   (c) $G(s) = \frac{e^{-0.1s}}{s+0.1}$. *(Hint: use Pade approximation to the time-delay in order to find the dominant dynamics, $e^{-ds} \approx \frac{-ds+2}{ds+2}$)*
2. *Evaluate the closed-loop stability by using Nyquist diagram. If the closed-loop system is unstable, reduce the parameter w_n to achieve stable closed-loop system.*
3. *Simulate the closed-loop step response for the PI controlled systems with sampling interval $\Delta t = 0.001$.*
4. *If the closed-loop response were oscillatory, reduce the parameter w_n until satisfactory performance is achieved.*

3.4 *Find the PID controller parameters using pole-assignment controller design technique and apply pole-zero cancellation to simplify the parameter solutions. Here, the PID controller structure is assumed to be $K_c + \frac{K_c}{\tau_I s} + K_c \tau_D s$. Take approximation of the complex dynamics if necessary. The desired closed-loop characteristic polynomial is $(s^2 + 2\xi w_n s + w_n^2)$ where $\xi = 0.707$ and w_n is adjusted for each case. The transfer functions with performance specification are given as below:*

1. $G(s) = \frac{10}{(s+20)s}$, *and* $w_n = 5$
2. $G(s) = \frac{2}{(s+3)(s-1)}$, *and* $w_n = 1$.
3. $G(s) = \frac{e^{-0.1s}}{(s+3)s}$, *and* $w_n = 1$.
4. $G(s) = \frac{s-3}{s(s+0.4)(s+10)}$, *and* $w_n = 0.2$.

3.5 *Find the parameters for PID controller with a filter using pole-assignment controller design technique. Use pole-zero cancellation technique to simplify the parameter solution and take approximation of the complex dynamics if necessary. With a pole-zero cancellation, the desired closed-loop polynomial is specified as $(s^2 + 2\xi w_n s + w_n^2)(s + 3w_n)$. The transfer functions with performance specification are given as below:*

1. $G(s) = \frac{e^{-s}}{(s+5)(s+2)(s+0.1)}$, $w_n = 1$ *and* $\xi = 0.707$.

2. $G(s) = \frac{s-1}{(s+10)(s+0.01)s}$, $w_n = 0.1$ *and* $\xi = 1$.

3.6 *For a system with the transfer function*

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0}$$

design a controller with the structure

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s}$$

where all three desired closed-loop poles are chosen to be $-\lambda$*. Convert this controller into an ideal PID controller that has the structure* $K_c + \frac{K_c}{\tau_I s} + K_c \tau_D s$ *so to find the proportional gain* K_c*, integral time constant* τ_I *and the derivative time* τ_D*. Use final value theorem to show that for a step reference signal with amplitude* R_0*, the closed-loop output response is* R_0 *as* $t \rightarrow \infty$*. The following three systems are used in this exercise.*

1. $G(s) = \frac{1}{s(s+2)}$, $\lambda = 3$, $R_0 = 1$.
2. $G(s) = \frac{-3}{s^2+3^2}$, $\lambda = 6$, $R_0 = -3$.
3. $G(s) = \frac{1}{s^2-1}$, $\lambda = 1$, $R_0 = 2$.

3.7 *For a system with the transfer function*

$$G(s) = \frac{b}{s + a}$$

design a resonant controller with the structure

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2 + \omega_0^2}$$

where all three desired closed-loop poles are chosen to be $-\lambda$*. Supposing that the reference signal is a sinusoidal signal* $r(t) = sin(\omega_0 t)$*, show that as* $t \rightarrow \infty$*, the feedback error* $r(t) - y(t) \rightarrow 0$*. The following three systems are used in this exercise.*

1. $G(s) = \frac{-1}{2s+1}$, $\omega_0 = 1$, *and* $\lambda = 2$.
2. $G(s) = \frac{0.5}{s}$, $\omega_0 = 0.1$, *and* $\lambda = 0.5$.
3. $G(s) = \frac{1}{s-1}$, $\omega_0 = 2$, *and* $\lambda = 1$.

3.8 *In order for the closed-loop system to track a reference signal* $r(t) = sin(\omega_0 t) + R_0$*, the controller needs to contain the factors s and* $s^2 + \omega_0^2$ *in its denominator. For a system with the transfer function*

$$G(s) = \frac{b}{s + a}$$

design a resonant controller plus integral action with the structure

$$C(s) = \frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s(s^2 + \omega_0^2)}$$

where all four desired closed-loop poles are chosen to be $-\lambda$*. Supposing that the reference signal is a sinusoidal signal* $r(t) = sin(\omega_0 t) + 1$*, show that as* $t \rightarrow \infty$*, the feedback error* $r(t) - y(t) \rightarrow 0$*. The following three systems are used in this exercise.*

1. $G(s) = \frac{0.1}{s+0.1}$, $\omega_0 = 1$, *and* $\lambda = 1$.

2. $G(s) = \frac{1}{5s+3}$, $\omega_0 = 0.1$, *and* $\lambda = 2$.
3. $G(s) = \frac{2}{s-2}$, $\omega_0 = 2$, *and* $\lambda = 2$

3.9 *A robot arm is described by the second order transfer function:*

$$G(s) = \frac{1}{s(s+6)} \tag{3.187}$$

where the input is the voltage and the output is the position of the arm on x-axis.

1. *Design a resonant control system such that the output of the robot arm will follow a sinusoidal reference signal* $r(t) = 3sin(t)$ *without steady-state error. All desired closed-loop poles are positioned at* -1 *(hint: use pole-zero cancellation technique to simplify the computation).*
2. *Verify your design by showing that the error signal* $(r(t) - y(t))$ *converges to zero as time* $t \to \infty$. *That is*

   $$\lim_{t\to\infty} e(t) = \lim_{t\to\infty}(r(t) - y(t)) = 0$$

   Write a Simulink simulation program to evaluate the closed-loop system performance for reference following of the sinusoidal signal where the sampling interval $\Delta t = 0.005$.
3. *Now, suppose that the reference signal contains a constant 10, leading to the new reference signal* $r(t) = 10 + 3sin(t)$. *Show that with this resonant controller, the error signal (*$r(t) - y(t)$*) also converges to zero as time* $t \to \infty$. *Modify the Simulink simulation program to include the new reference signal and confirm that the closed-loop output tracks the new reference signal without steady-state error.*
4. *Assume that the input disturbance* $d_i(t) = d_0 + d_m sin(t)$ *enters the system at half of the simulation time, where* $d_0 = -2$ *and* $d_m = 1$. *Modify the Simulink simulation program to include the disturbance and confirm that the resonant controller can not completely reject this input disturbance. What would you propose for the resonant controller structure so that it will reject this disturbance without steady-state error? Verify your answer using final value theorem.*

3.10 *An unstable system is described by the transfer function:*

$$G(s) = \frac{1}{(s-a)(s+p)} \tag{3.188}$$

where $a > 0$ *and* $p > 0$.

1. *Design a PI controller to control this system, where all desired closed-loop poles are positioned at* $-a$.
2. *In order to guarantee the closed-loop stability of the PI control system, the magnitude of the stable pole p has a relationship with the unstable pole a. Use Routh-Hurwitz stability criterion to find this relationship.*

4

Implementation of PID Controllers

4.1 Introduction

The key reason for the wide application of PID control systems is their simplicity of structure, design, and implementation. This chapter shows how to implement a PID controller with operational constraints. Operational constraints are used to ensure plant safe operation and protect the electronics from damage when the control signals exceed limits. Without exception, all control systems implemented in practical applications must contain the operational constraints.

4.2 Scenario of a PID Controller at work

To understand how the implementation of a PID controller is performed, we will first describe the general scenario of a digital control system at work.

A computer controlled system generally consists of a plant to be controlled (including sensors and actuators), and a computer that stores control algorithms and computes the input to the plant. In many applications, the computer is replaced by a micro-controller or a digital signal processor.

The majority of the physical systems are continuous time systems. The variables within the systems are inherently analog continuous signals. For instance, the pressure signals, temperature signals, voltage signals and current signals are the commonly encountered analog signals in physical systems. These analog signals are continuous in nature. Since the computers, micro-controllers and digital signal processors only take the digital signals for processing, there must be key devices that “bridge” the analog world (physical plants) to the digital world (computing instruments). These devices are called analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). An internal clock synchronizes the actions of the ADCs and DACs. Together, they form a computer controlled system or digital device controlled system.

Electronic sampling is the most frequently used method in analog-to-digital converters. The ADCs in the micro-controller and in the commonly used interface cards use electronic sampling. When using this method, a sensor is connected to physical plant, which generates an electronic output signal. This signal is passed through an electronic device that samples the output signal of the sensor and converts it to a digital signal. The ADC using electronic sampling is an integrated circuit that accepts an analog voltage as

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

its input, at the same time, it accepts a time signal that gives an interface between the outside environment and the synchronized environment inside a data processing unit.

A digital-to-analog converter is an integrated circuit that accepts voltage signals at high and low levels and produces an analog output proportional to the real number represented by the binary word at the input of the DAC. The DAC is an electronic implementation of the zero-order-hold. The input to the device is a binary word that is made available, by the clock signal, at the ith sample instant. The output signal of the DAC is a constant voltage over the interval $t_i \leq t \leq t_{i+1}$, where $t_i = i\Delta t$ and $t_{i+1} = (i+1)\Delta t$. A DAC is a circuit that accepts a digital signal and produces a quantized signal. The output signal consists of a series of steps and the width of the steps is the sampling interval Δt. Here, we assume that the input signal is a discrete signal and the output signal is a continuous-time signal in steps.

How does a computer controlled system work? To illustrate how in a general sense a computer controlled system works, a room temperature control is used as an example. Suppose that in a cold environment the appropriate room temperature is maintained through heating. The heating elements are from an oil furnace. The input to control the room temperature is the fuel injection rate to the furnace and the output is room temperature. The control objective is to maintain the room temperature at 19°C. The assumption is that the measurement and control action are synchronized in terms of a sequence of time intervals. The intervals between two executions are assumed to be the same, and it is called the sampling interval Δt. A typical cycle of measurement and control is summarized in the following steps.

1. Take the output measurement. At the time when the closed-loop control system is switched on, assumed as t_0, the temperature sensor reads the room temperature $y(t_0)$. Then this measurement is read by the computer or the digital device into its memory.
2. Compute feedback error. This temperature measurement at time t_0 is compared with the required room temperature. This comparison generates the feedback error $e(t_0)$.
3. Compute the controller output. The control algorithm in computer computes a control signal $u(t_0)$ based the feedback error $e(t_0)$ and the history of the feedback error, depending on the complexity of the control algorithm.
4. This digital value $u(t_0)$, through zero-order hold, is converted into the analog control signal $u(t)$ (fuel injection rate), where $u(t) = u(t_0)$, $t_0 \leq t \leq t_1$, and $t_1 = t_0 + \Delta t$.
5. This analog signal is the fuel injection rate for the time interval, $t_0 \leq t \leq t_1$.
6. The clock ticks to the next sample period, t_1, the temperature sensor reads the room temperature as $y(t_1)$, and the entire controlling process repeats the steps listed.

4.3 PID Controller Implementation using the Position Form

Position form implementation of PID controllers involves direct discretization of the continuous time controller.

4.3.1 The Steady-state Information Needed

The control signal $u(t)$ for a PID controller is computed using the equation,

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau) d\tau - K_c \tau_D \frac{dy_f(t)}{dt} \tag{4.1}$$

where $e(t) = r(t) - y(t)$ is the feedback error signal between the reference signal $r(t)$ and the output $y(t)$, and $y_f(t)$ is the filtered output signal. It is important to make it clear that all the signals used in the computation are not the actual physical variables in numbers, instead they are the deviation variables from the physical variables in a steady-state operation. In other words, the control signal $u(t)$, the reference signal $r(t)$, and the output signal $y(t)$ represent the changes to the corresponding physical variables in steady-state operation. For instance, in the application of a room temperature control in a wintery environment if all doors and windows were closed the room temperature was set to 18°C, which was maintained by the heat of a gas furnace with its valve opening at 40%. The steady-state of the output variable is $Y_{ss} = 18$°C, and the steady-state of the control variable is $U_{ss} = 40\%$. If we were happy with the room temperature ($r(t) = 0$) and the doors and windows remained closed, then there would be no change to the room temperature $y(t) = 0$ due to the steady operation of the gas furnace. At this steady-state operation, the control signal from the PID controller is $u(t) = 0$. If someone decided to change the room temperature setting from 18°C to 20°C at a time t_0, then the reference signal $r(t) = 2$°C at $t = t_0$ and the feedback error $e(t) = r(t) - y(t) = 2 - 0$ at $t = t_0$. Based on the feedback error $e(t)$, the controller produces a control signal $u(t) > 0$ at $t = t_0$. The gas furnace would change the valve opening percentage ($u_{act}(t)$: $u_{act}(t) = U_{ss} + u(t)$). The room temperature sensor reads the actual room temperature as $y_{act}(t)$ and the output signal $y(t)$ is updated with $y(t) = y_{act}(t) - Y_{ss}$. The control signal $u(t)$ changes with the new output signal $y(t)$.

The position form of the PID controller is to directly calculate the deviation control variable $u(t)$ based on the feedback error $e(t)$. Therefore, when using the position form of PID controller for implementation, it is vital to have a priori knowledge about the steady-state information of the control signal and the output signal, so that the actual control signal will be computed as $u_{act}(t) = u(t) + U_{ss}$; in reverse, the output signal $y(t) = y_{act}(t) - Y_{ss}$.

4.3.2 Discretization of a PID Controller

In the implementation of a PID controller when using micro- controllers or other computational devices, the input signals to the PID controller are digital signals and the output from the PID controller is also a digital signal. Therefore, we need to find the discrete form of the PID controller in order for it to be implemented. In the position form, the output of the PID controller is computed from the sum of the three terms (see 4.1). For simplicity of expression, we let the control signal

$$u(t) = u_P(t) + u_I(t) - u_D(t)$$

where $u_P(t)$, $u_I(t)$ and $u_D(t)$ represent the proportional, integral, and derivative control terms, respectively.

In the position form, the implementation of proportional (P) controller, proportional plus integral (PI) controller, and proportional plus derivative (PD) controller, and proportional plus integral plus derivative (PID) controller will be formed from using one single term or a combination of single terms from the discretized functions.

We assume that the discretization occurs in a uniformly sampling interval Δt, and the continuous time t is sampled as $t = 0, t_1, t_2, \dots, t_{i-1}, t_i, t_{i+1}, \dots,$. The proportional term is easiest to be discretized. At an arbitrary time t_i, the proportional control term $u_\mathrm{P}(t_i)$ is calculated as

$$u_\mathrm{P}(t_i) = K_\mathrm{c}(r(t_i) - y(t_i)). \tag{4.2}$$

The integral control term $u_\mathrm{I}(t_i)$ requires numerical approximation for the integral function, which is written as

$$u_\mathrm{I}(t_i) = \frac{K_\mathrm{c}}{\tau_\mathrm{I}} \sum_{k=0}^{i} e(t_k)\Delta t \tag{4.3}$$

where $\int_0^t e(\tau)\mathrm{d}\tau \approx \sum_{k=0}^{i} e(t_k)\Delta t$. We know that

$$\lim_{\Delta t \to 0} \sum_{k=0}^{i} e(t_k)\Delta t = \int_0^t e(\tau)\mathrm{d}\tau. \tag{4.4}$$

Therefore, the accuracy of the approximation increases as the sampling interval Δt reduces.

The discretization of the derivative term $u_\mathrm{D}(t)$ in the position form is identical to the procedure used in the velocity form, as described by Equations (4.22)–(4.25).

One critical problem with this implementation of a PID controller is that the integral term from the computation used in (4.3) could be too large, resulting in the numerical problem of variable "overflow". Because the integral term uses the accumulated sum from all the past feedback error signal, this problem will occur if the feedback error has a non-zero steady-state value. The PID implementation algorithm introduced in Section 4.4 avoids this problem with an iterative approach.

4.3.3 Food for Thought

1. Without detailed mathematical modelling, how would you propose to obtain steady-state information for a physical plant?
2. The position form of a PID controller is calculated using

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau)d\tau - K_c \tau_D \frac{dy_f(t)}{dt} \tag{4.5}$$

   Is it possible for the control variable $u(t)$ to become negative in the situation where the physical system obviously requires a positive control input signal (for instance, percentage opening of a valve)?
3. The PID controllers are designed in continuous-time, at what stage do we obtain the discretized form of the PID controller?
4. What would you do when you discovered that the PID control system you designed should be stable, and yet it was unstable in a perfect simulation environment in which sampled data were used?
5. From your experience in numerical analysis, with the same sampling interval Δt, which one is easier, the approximation of integration or the approximation of derivative?
6. Is it possible that the integral calculation becomes too large for this type of implementation?

4.4 PID Controller Implementation using the Velocity Form

There are only two types of PID controllers using the velocity form for implementation. One is proportional plus integral (PI) controller, the other is proportional plus integral plus derivative (PID) controller. Proportional (P) controllers and proportional plus derivative (PD) controllers cannot be implemented using the velocity form for the reason that the implementation involves direct manipulation of the integral term.

In the velocity form, the PI controller or PID controller is discretized without separation of individual terms, and the solution is focused on the derivative of the control variable $\frac{\mathrm{d}u(t)}{\mathrm{d}t}$, from which comes the term "velocity".

4.4.1 Discretization of a PI Controller

The Laplace transfer function of the control signal $U(s)$ is in relation to the feedback error $E(s)$:

$$\begin{aligned} U(s) &= K_c(1 + \frac{1}{\tau_I s})E(s) \\ &= \frac{K_c sE(s) + \frac{K_c}{\tau_I}E(s)}{s}. \end{aligned} \tag{4.6}$$

From (4.6), we have

$$sU(s) = K_c sE(s) + \frac{K_c}{\tau_I}E(s). \tag{4.7}$$

The inverse Laplace transform of (4.7) leads to the following differential equation:

$$\dot{u}(t) = K_c\dot{e}(t) + \frac{K_c}{\tau_I}e(t). \tag{4.8}$$

This is the equation that is used to calculate the derivative of the control signal $\dot{u}(t)$. This is termed the velocity form because on the left-hand side of the equation the derivative of the control signal is computed.

At a first glance of (4.8), we notice that the derivative of the error signal $\dot{e}(t)$ is involved in the computation. However, the derivative of the error signal $\dot{e}(t)$ will be eliminated in the discretization process.

Let us represent the derivatives of the control and the feedback error signals $\dot{u}(t)$ and $\dot{e}(t)$ by the first order approximation at sample time t_i :

$$\dot{u}(t_i) \approx \frac{u(t_i) - u(t_{i-1})}{\Delta t} \tag{4.9}$$

$$\dot{e}(t_i) \approx \frac{e(t_i) - e(t_{i-1})}{\Delta t}. \tag{4.10}$$

By substituting these approximations into (4.8), we obtain

$$u(t_i) - u(t_{i-1}) = K_c(e(t_i) - e(t_{i-1})) + \frac{K_c}{\tau_I}e(t_i)\Delta t. \tag{4.11}$$

By moving the term $u(t_{i-1})$ from the left-hand side to the right-hand side, the computation of the control signal becomes

$$u(t_i) = u(t_{i-1}) + K_c(e(t_i) - e(t_{i-1})) + \frac{K_c}{\tau_I}e(t_i)\Delta t. \tag{4.12}$$

It is useful to emphasize that the feedback error signal $e(t_i) = r(t_i) - y(t_i)$.

For the IP structure discussed in Chapter 1 where the proportional control term is only implemented on the output signal, a small modification of (4.12) is to replace the differenced feedback error $e(t_i) - e(t_{i-1})$ with the differenced output signal $-y(t_i) + y(t_{i-1})$. As a result, the implementation of the IP controller structure is based on the computational equation,

$$u(t_i) = u(t_{i-1}) + K_c(-y(t_i) + y(t_{i-1})) + \frac{K_c}{\tau_I}e(t_i)\Delta t. \tag{4.13}$$

Note that in the implementation of equation (4.13), the current control signal $u(t_i)$ at time t_i is a linear combination of past control signal $u(t_{i-1})$, current output $y(t_i)$, past output signal $y(t_{i-1})$, and current reference signal $r(t_i)$. All these signals on the right-hand side of (4.13) are available at the current sampling time t_i.

One of the key advantages of using the velocity form for implementation is the handling of steady-state information. As before, all the signals in (4.13) are the deviation signals. They are relative to their steady-state values. To this end, with the steady-state values of U_{ss}, Y_{ss} and R_{ss}, the actual signals corresponding to the plant operation are

$$u_{act}(t_i) = u(t_i) + U_{ss} \tag{4.14}$$
$$y_{act}(t_i) = y(t_i) + Y_{ss} \tag{4.15}$$
$$r_{act}(t_i) = r(t_i) + R_{ss}. \tag{4.16}$$

We consider the IP structure in (4.13) first, and then extend the procedure to the original PI structure in (4.12).

Now, we manipulate (4.13) by first adding the steady- state value of U_{ss} to both sides, adding and subtracting Y_{ss} to the second term that corresponds to the proportional control, which leads to its equivalent expression:

$$\begin{aligned} u(t_i) + U_{ss} &= u(t_{i-1}) + U_{ss} + K_c(-y(t_i) - Y_{ss} + Y_{ss} + y(t_{i-1})) \\ &\quad + \frac{K_c\Delta t}{\tau_I}(r(t_i) - y(t_i)). \end{aligned} \tag{4.17}$$

At this point, we need to make the assumption that the steady state of the reference signal R_{ss} is equal to the steady state of the output signal Y_{ss}, which is realistic for plant operations.

With the steady-state information, we replace the deviation variables with the actual physical variables, and obtain the computational equation for the actual implementation of the PI controller as,

$$u_{act}(t_i) = u_{act}(t_{i-1}) + K_c(-y_{act}(t_i) + y_{act}(t_{i-1})) + \frac{K_c\Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)). \tag{4.18}$$

With this formulation given by (4.18), all actual measurements of the physical variables are used in the updating of the control signal, and the control signal computed is the physical variable to be implemented. Perhaps more importantly, the implementation using (4.18) does not require the actual steady-state information of U_{ss} and Y_{ss}, which makes the implementation convenient and practical in control applications. The extension to the original PI structure in (4.12) follows from the same assumption that the steady state of the reference signal R_{ss} is equal to the steady state of the output signal Y_{ss}. Thus, the implementation equation for the original PI structure is

$$\begin{aligned} u_{\mathrm{act}}(t_i) &= u_{\mathrm{act}}(t_{i-1}) + K_c(r_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_i) - r_{\mathrm{act}}(t_{i-1}) + y_{\mathrm{act}}(t_{i-1})) \\ &\quad + \frac{K_c \Delta t}{\tau_I}(r_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_i)). \end{aligned} \tag{4.19}$$

Here the proportional control will directly act on the reference signal.

4.4.2 Discretization of a PID Controller using the Velocity Form

The Laplace transfer function of a PID controller is expressed as

$$U(s) = K_c E(s) + \frac{K_c}{\tau_I s} E(s) - \frac{K_c \tau_D s}{\tau_f s + 1} Y(s). \tag{4.20}$$

The first two terms of left-hand side of Equation (4.20) form the PI controller, which was discretized in the previous section. The question is how the third term corresponding to the derivative control will be discretized. Since we do not wish to use steady-state information of the output, a different approach for discretization of the derivative term is needed in the velocity form. From the transfer function of the derivative control,

$$U_D(s) = \frac{1}{\tau_f} \frac{K_c \tau_D s}{s + \frac{1}{\tau_f}} Y(s) \tag{4.21}$$

the differential equation that governs the relationship between the variables $u_D(t)$ and $y(t)$ is obtained as

$$\frac{du_D(t)}{dt} + \frac{1}{\tau_f} u_D(t) = \frac{K_c \tau_D}{\tau_f} \frac{dy(t)}{dt}. \tag{4.22}$$

By approximating

$$\frac{du_D(t)}{dt} \approx \frac{u_D(t_i) - u_D(t_{i-1})}{\Delta t}; \quad \frac{dy(t)}{dt} \approx \frac{y(t_i) - y(t_{i-1})}{\Delta t}$$

at time t_i, the differential equation (4.22) becomes:

$$\frac{u_D(t_i) - u_D(t_{i-1})}{\Delta t} = -\frac{1}{\tau_f} u_D(t_i) + \frac{K_c \tau_D}{\tau_f} \frac{y(t_i) - y(t_{i-1})}{\Delta t}. \tag{4.23}$$

By multiplying Δt on both sides of the equation, and re- arranging, we obtain

$$(1 + \frac{\Delta t}{\tau_f}) u_D(t_i) = u_D(t_{i-1}) + \frac{K_c \tau_D}{\tau_f}(y(t_i) - y(t_{i-1})). \tag{4.24}$$

The steady state of $u_D(t)$ is taken as zero, and by adding and subtracting the steady state of the output value to (4.24), we obtain the computation of the derivative control term using the actual output measurement,

$$u_D(t_i) = \frac{\tau_f}{\tau_f + \Delta t} u_D(t_{i-1}) + \frac{K_c \tau_D}{\tau_f + \Delta t}(y_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_{i-1})). \tag{4.25}$$

Now, going back to the expression of the PID controller given by (4.20), the derivative of the control signal is expressed as

$$\dot{u}(t) = K_c \dot{e}(t) + \frac{K_c}{\tau_I} e(t) - \dot{u}_D(t). \tag{4.26}$$

Combining the discretization of the derivative term (4.25) with the discretization of the PI controller part, the completed PID controller when using velocity form is expressed as

$$\begin{aligned} u_{\mathrm{act}}(t_i) &= u_{\mathrm{act}}(t_{i-1}) + K_{\mathrm{c}}(r_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_i) - r_{\mathrm{act}}(t_{i-1}) + y_{\mathrm{act}}(t_{i-1})) \\ &\quad + \frac{K_{\mathrm{c}}\Delta t}{\tau_{\mathrm{I}}}(r_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_i)) - u_{\mathrm{D}}(t_i) + u_{\mathrm{D}}(t_{i-1}). \end{aligned} \tag{4.27}$$

When implementing the proportional control directly on the output, the PID controller using velocity form is

$$\begin{aligned} u_{\mathrm{act}}(t_i) &= u_{\mathrm{act}}(t_{i-1}) + K_{\mathrm{c}}(-y_{\mathrm{act}}(t_i) + y_{\mathrm{act}}(t_{i-1})) \\ &\quad + \frac{K_{\mathrm{c}}\Delta t}{\tau_{\mathrm{I}}}(r_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_i)) - u_{\mathrm{D}}(t_i) + u_{\mathrm{D}}(t_{i-1}). \end{aligned} \tag{4.28}$$

Example 4.1 *A second order system is described by the transfer function:*

$$G(s) = \frac{-0.1}{(s+1)^2}. \tag{4.29}$$

Design a PID controller with filter to control this plant. The desired closed-loop polynomial is specified as $(s^2 + 2\xi w_{\mathrm{n}} s + w_{\mathrm{n}}^2)(s + \lambda_1)^2$, *where* $\xi = 0.707$, $w_{\mathrm{n}} = \lambda_1 = 5$. *Simulate the unit step closed-loop response using the discrete PID computational algorithm with sampling interval* Δt *chosen as* $\frac{1}{10w_{\mathrm{n}}} = 0.02$ *and* $\frac{1}{5w_{\mathrm{n}}} = 0.04$. *Illustrate that when* Δt *is increased to* 0.1, *the closed-loop system becomes unstable although the continuous time PID controller design produced a stable closed-loop system.*

Solution. *Using the MATLAB program pidplace.m (see Tutorial 3.2), we obtain the PID controller parameters as*

$$K_{\mathrm{c}} = -245.66;\ \tau_I = 0.59;\ \tau_{\mathrm{D}} = 0.176;\ \tau_{\mathrm{f}} = 0.066.$$

Figure 4.1 compares the closed-loop responses with the three cases of the sampling intervals. It is seen that the difference between the closed-loop responses using $\Delta t = 0.02$ *and* 0.04 *is negligible. However, when the sampling interval is increased to* 0.1, *the closed-loop system becomes unstable as shown in Figure 4.1.*

4.4.3 Improving Accuracy in a Slower Sampling Environment

Sampling rate $\frac{1}{\Delta t}$ is typically chosen to be 5 to 10 times the closed-loop bandwidth. In the PI control system, this bandwidth is the parameter w_n. In reality, this range of sampling rates will result in performance degradation from the desired closed-loop performance we aimed at. To make the sampling effect truly negligible, the sampling rate would be at least 10 times higher than a typical rate. Since PID controllers are designed in a continuous-time environment (i.e. the model is continuous-time differential equation and its parameters are chosen corresponding to the continuous-time system), in principle the sampling rate $\frac{1}{\Delta t}$ should be chosen as large as the computational power of the system permitted.

When the sampling rate is slow (or Δt is large relative to the closed-loop bandwidth), a possible way to improve the accuracy of the discretization of the PID controller is to

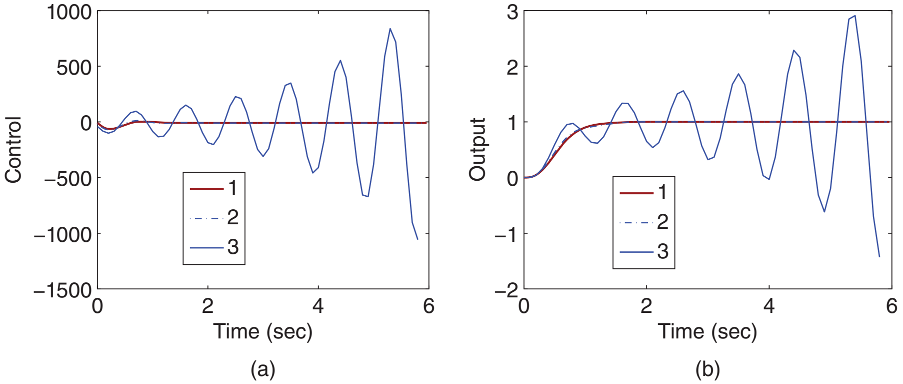


Figure 4.1 Closed-loop response (Example 4.1). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response with $\Delta t = 0.02$; line (2) closed-loop response with sampling interval $\Delta t = 0.04$; line (3) closed-loop response with sampling interval $\Delta t = 0.1$.

use a higher order discretization scheme for the derivatives of the signals (Burden and Faires (1989)).

Let

$$\dot{u}(t_i) = \frac{3u(t_i) - 4u(t_{i-1}) + u(t_{i-2})}{2\Delta t} \tag{4.30}$$

$$\dot{e}(t_i) = \frac{3e(t_i) - 4e(t_{i-1}) + e(t_{i-2})}{2\Delta t}. \tag{4.31}$$

Considering the original velocity form of the PI controller:

$$\dot{u}(t) = K_c\dot{e}(t) + \frac{K_c}{\tau_I}e(t) \tag{4.32}$$

by replacing the derivatives $\dot{u}(t)$ and $\dot{e}(t)$ with (4.30) and (4.31), we obtain the approximate equation to (4.32) at sampling time t_i :

$$\frac{3u(t_i) - 4u(t_{i-1}) + u(t_{i-2})}{2\Delta t} = K_c\frac{3e(t_i) - 4e(t_{i-1}) + e(t_{i-2})}{2\Delta t} + \frac{K_c}{\tau_I}e(t_i). \tag{4.33}$$

By adding and subtracting the quantity $4U_{ss}$ to the left side of (4.33), (4.33) becomes,

$$\begin{aligned}&\frac{3u(t_i) - 4u(t_{i-1}) + u(t_{i-2}) + 4U_{ss} - 4U_{ss}}{2\Delta t}\\ &= \frac{3u_{act}(t_i) - 4u_{act}(t_{i-1}) + u_{act}(t_{i-2})}{2\Delta t}\\ &= K_c\frac{3e(t_i) - 4e(t_{i-1}) + e(t_{i-2})}{2\Delta t} + \frac{K_c}{\tau_I}e(t_i)\end{aligned} \tag{4.34}$$

where $u_{act}(t_i) = u(t_i) + U_{ss}$, $u_{act}(t_{i-1}) = u(t_{i-1}) + U_{ss}$ and $u_{act}(t_{i-2}) = u(t_{i-2}) + U_{ss}$. Thus the actual control signal is

$$\begin{aligned}u_{act}(t_i) = &\frac{4}{3}u_{act}(t_{i-1}) - \frac{1}{3}u_{act}(t_{i-2}) + \frac{K_c}{3}(3e(t_i) - 4e(t_{i-1}) + e(t_{i-2}))\\ &+ \frac{2\Delta t}{3}\frac{K_c}{\tau_I}e(t_i).\end{aligned} \tag{4.35}$$

4.4.4 Food for Thought

1. In the velocity form of PID controller, at which step did we add the steady-state values of the signals in order to obtain the PID controller representation with the actual plant signals?
2. Is it correct to say that the implementation of the IP controller is simply to neglect the reference signal in the calculation of the proportional control term?
3. Under what plant operational condition could we assume that the steady-state of the derivative control signal $u_D(t)$ is zero?
4. There is a guideline in the selection of the sampling interval Δt based on the bandwidth of the closed-loop control system. Would you also consider that this guideline might not be adequate when the derivative filter time constant is much smaller than the closed-loop time constant?

4.5 Anti-windup Implementation using the Position Form

The implementation of PID controllers requires the consideration of an anti-windup mechanism in the presence of constraints on the control signals. There are many approaches available for implementing PID controllers with an anti-windup mechanism. This section will discuss the implementation when using the position form.

4.5.1 Integrator Windup Scenario

To understand the scenario of integrator wind-up, we look at the example below.

Example 4.2 *Consider the integrating plus delay plant with the transfer function,*

$$G(s) = \frac{1.8e^{-30s}}{s(10s+1)^2}. \tag{4.36}$$

The PI controller for this plant has proportional gain $K_c = 0.0065$ *and integral time constant* $\tau_I = 244.5$*. Simulate the closed-loop response for this PI control system with a unit step reference signal. Supposing that the control signal amplitude is not to exceed* 1.5×10^{-3}*, illustrate the scenario of integrator windup.*

Solution. *The simulation set-up is conducted using a Simulink program, where a saturation block is used to simulate the scenario of limits of control amplitude. Both proportional and integral terms are implemented on the feedback error signal. We first set the limits in the saturation blocks to be larger (*± 3*) than the maximum and minimum of the control signal amplitude. The control signal and output responses are shown in Figure 4.2 (see line (1)). The closed-loop responses behave reasonably well. By reducing the allowable control amplitude to* 1.5×10^{-3}*, the actual control signal to the plant, u, is limited. When this happens, the closed-loop response becomes oscillatory, as shown in Figure 4.2 (see line (2)). When investigating this further, we find that the controller output signal (*u_1*) continued to grow as shown in Figure 4.3 (see solid line) after the actual control signal to the plant was limited (see dash-dot line) because the feedback error is positive. The amplitude of* u_1 *reached its maximum when the feedback error* $e(t)$ *changes sign from positive to negative (see the top plot). The magnitude of the*

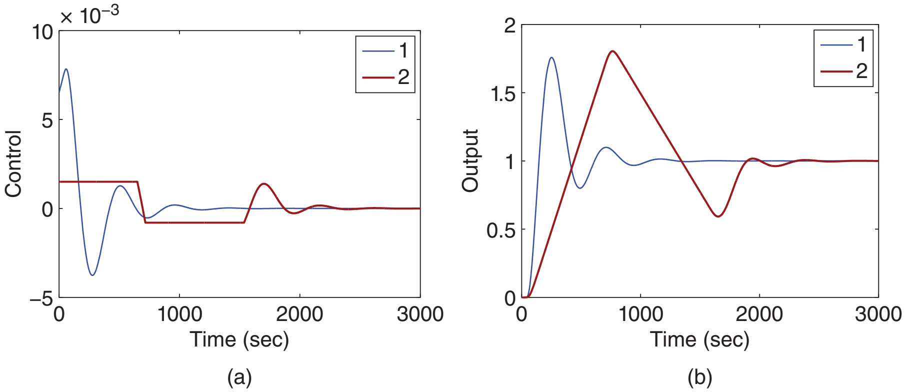
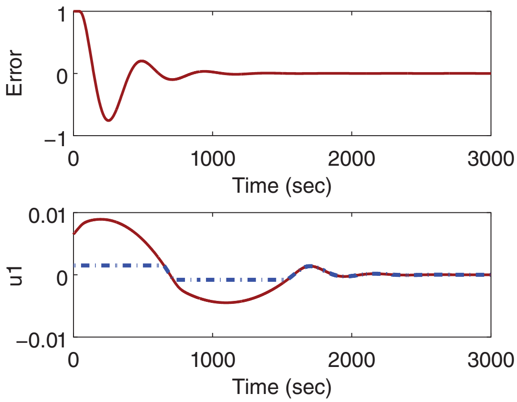


Figure 4.2 Closed-loop response (Example 4.2). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response without saturation; line (2) closed-loop response with saturation.

Figure 4.3 Error signal and control signal in the integrator windup case (Example 4.2).



control signal gradually reduced as the magnitude of the error increased, which is in the negative region. Eventually, the integral term became sufficiently small, and the control signal came within the allowable upper limit. When the control signal changed its sign, it encountered the negative constraint. As the result of control signal saturation, both the error signal and the output signal became oscillatory.

There are two things we noted from this example.

1. In the position form when implementing a PI controller, the control signal is calculated using the equation,

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(\tau)\mathrm{d}\tau \tag{4.37}$$

where the feedback error $e(t)$ is integrated. This integral term will continue to grow in magnitude as long as the sign of the feedback error remains the same (integration is about calculation of the area of the curve). As a result, the control signal $u(t)$ calculated will continue to grow as long as the sign of the feedback error remains the same.

2. Because of the saturation, the actual control signal implemented on the plant is not the same as the output of the controller when the saturation limits are reached. The controller is not informed of what is actually happening in the plant.

Those are the two key points needed to be taken into consideration when devising anti-windup schemes in PID controllers. In summary, the integral function should be stopped when the saturation limits are reached and the controller should be informed of what is actually happening in the plant.

4.5.2 Anti-windup Mechanisms in the Position Form of PI Controllers

There are many anti-windup mechanisms in PI controllers. More or less, they have the same principles by implementing the PI controller with a stable transfer function and letting the controller know what is actually happening in the plant. Figure 4.4 shows one of the anti-windup mechanisms in the position form of PI controller implementation with amplitude constraint. In this implementation, the parameters of the PI controller are c_1 and c_0, corresponding to the transfer function,

$$C(s) = \frac{c_1 s + c_0}{s}. \tag{4.38}$$

The assumption for using this implementation is that the controller has a stable zero, namely the ratio $\frac{c_0}{c_1} > 0$, or τ_I is positive. In this implementation, there is a positive feedback used in the system (positive feedback is seldom used, but here is an example). $\sum$ represents the saturation nonlinearity, which is defined by the following computation. If $u^{\min} < u_0(t) < u^{\max}$, then $u(t) = u_0(t)$; if $u_0(t) \le u^{\min}$, then $u(t) = u^{\min}$; if $u_0(t) \ge u^{\max}$, then $u(t) = u^{\max}$. The saturation block in the Simulink library can be used to represent the function $\sum$.

When the saturation limits are not reached, $\sum$ is a unity gain ($u(t) = u_0(t)$), then the transfer function from the error signal e to the control signal u is

$$\begin{aligned}\frac{U(s)}{E(s)} &= \frac{c_1}{1 - \frac{c_0 c_1}{c_1(c_1 s + c_0)}}\\ &= \frac{c_1 s + c_0}{s}\end{aligned} \tag{4.39}$$

which is the transfer function of a PI controller (4.38). The integral action in this configuration is achieved by putting positive feedback around a stable transfer function. If the control signal reaches a limit, for instance, $u(t) = u^{\max}$, because the transfer function $\frac{c_0}{c_1(c_1 s + c_0)}$ is stable by the assumption made, the feedback signal $\frac{c_0}{c_1(c_1 s + c_0)}$ will become a constant after a transient response. With this action, the integral action will be stopped,

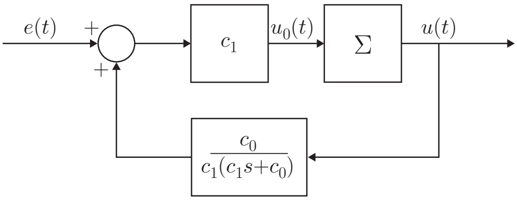


Figure 4.4 PI controller (position form) with anti-windup mechanism ($\sum$ represents the saturation nonlinearity).

also the controller calculation is fully informed of what is actually happening by this feedback link.

Example 4.3 *Apply the anti-windup control mechanism in the PI controller given in Example 4.2, and compare the simulation results with the results obtained from the previous example.*

Solution. *The parameters required for the anti-windup implementation are*

$$c_1 = 0.0065; c_0 = 2.6585e - 005.$$

Using the anti-windup mechanism, we simulated the closed-loop response in the same conditions as those in Example 4.2. Figure 4.5 shows the closed-loop response with the anti-windup mechanism. In comparison to the case without the anti-windup mechanism, the overshoot in the closed-loop output response has been significantly reduced. The control signal with the anti-windup scheme came out of the saturation faster than the control signal without it. Also, the duration of the second saturation time was shorter in the anti-windup scheme. As a result, the closed-loop performance is improved with the anti-windup mechanism.

4.5.3 Food for Thought

1. Could PID windup event occur if the control signal operates within its limits?
2. Could this windup event happen to a PD controller?
3. In the position form of PI controller anti-windup implementation, is it the case that each element in the implementation is stable?
4. If the steady-state value of the control signal is not zero, where would you propose to add this steady-state value in the implementation block diagram shown in Figure 4.4?
5. Could you use the anti-windup mechanism in the position form if the integral time constant τ_I is negative?

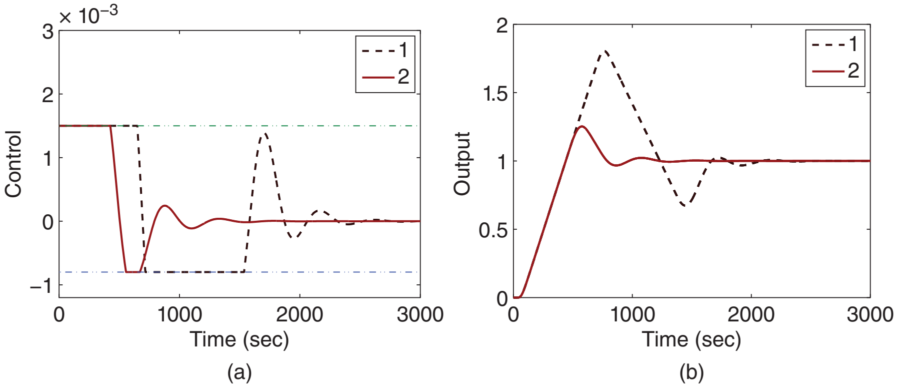


Figure 4.5 Closed-loop response (Example 4.3). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response without anti-windup mechanism; line (2) closed-loop response with anti-windup mechanism.

4.6 Anti-windup Mechanisms in the Velocity Form

It is straightforward to implement anti-windup mechanisms in the velocity form of PID controllers, mainly because the velocity form of a PID controller is derived for computer controlled systems in discrete time form. Similarly, the two key points in implementation of anti-windup mechanism are to stop integral action when the control signal reaches saturation and to make sure that the actual control signal equals the control signal computed. In the velocity form of PID controllers, not only can the anti-windup scheme be readily implemented on the amplitude of the control signal, but also on the derivative of the control signal.

4.6.1 Anti-windup Mechanism on the Amplitude of the Control Signal

We assume that the actual control variable to the plant is limited by $u^{\min}$ and $u^{\max}$. Namely, the actual control signal must satisfy the constraints:

$$u^{\min} \leq u_{\text{act}}(t) \leq u^{\max}.$$

From Section 4.4, we learned how to calculate the actual control signal $u_{\text{act}}(t_i)$ at sample time t_i using the equation,

$$\begin{aligned} u_{\text{act}}(t_i) &= u_{\text{act}}(t_{i-1}) + K_c(-y_{\text{act}}(t_i) + y_{\text{act}}(t_{i-1})) \\ &\quad + \frac{K_c\Delta t}{\tau_I}(r_{\text{act}}(t_i) - y_{\text{act}}(t_i)) - u_D(t_i) + u_D(t_{i-1}). \end{aligned} \tag{4.40}$$

With this formulation given by (4.40), all actual measurements of the physical variables are used in the updating of the control signal, and the control signal computed is the physical variable to be implemented. Thus, the implementation procedure naturally satisfies one of the requirements in an anti-windup mechanism that the actual control signal be equal to the computed control signal. In order to stop the integration, when the actual control signal reaches the limit, we impose the limits on the actual control signal with the computation that if $u_{\text{act}}(t_i) < u^{\min}$, then $u_{\text{act}}(t_i) = u^{\min}$; if $u_{\text{act}}(t_i) > u^{\max}$, then $u_{\text{act}}(t_i) = u^{\max}$. When the sample time t_i moves one step forward, $u_{\text{act}}(t_{i-1})$ carries the information of saturation at the previous sample time and the control signal computation is automatically informed of the saturation. Both requirements in an anti-windup mechanism are satisfied.

In summary, the implementation of an anti-windup scheme in the velocity form of a PI controller follows the steps below.

1. For the IPD controller structure, calculate the actual control signal using

$$\begin{aligned} u_{\text{act}}(t_i) &= u_{\text{act}}(t_{i-1}) + K_c(-y_{\text{act}}(t_i) + y_{\text{act}}(t_{i-1})) \\ &\quad + \frac{K_c\Delta t}{\tau_I}(r_{\text{act}}(t_i) - y_{\text{act}}(t_i)) - u_D(t_i) + u_D(t_{i-1}). \end{aligned} \tag{4.41}$$

Or for original PID controller structure, calculate

$$\begin{aligned} u_{\text{act}}(t_i) &= u_{\text{act}}(t_{i-1}) + K_c(e_{\text{act}}(t_i) - e_{\text{act}}(t_{i-1}) \\ &\quad + \frac{K_c\Delta t}{\tau_I}(e_{\text{act}}(t_i)) - u_D(t_i) + u_D(t_{i-1}) \end{aligned} \tag{4.42}$$

where $e_{\text{act}}(t_i) = r_{\text{act}}(t_i) - y_{\text{act}}(t_i)$.

2. Check if the control signal is within the limits

$$u^{\min} \leq u_{\text{act}}(t_i) \leq u^{\max}.$$

If the constraints are satisfied, then this is the actual control signal to the plant. If not, go to the next step,

3. If $u_{\text{act}}(t_i) < u^{\min}$, then $u_{\text{act}}(t_i) = u^{\min}$; if $u_{\text{act}}(t_i) > u^{\max}$, then $u_{\text{act}}(t_i) = u^{\max}$.

Example 4.4 *Consider a mechanical system with severe non-minimum phase behavior that is described by the transfer function:*

$$G(s) = \frac{s - 0.1}{(s + 1)(s + 2)}. \tag{4.43}$$

The operation of the control signal is required to be within the limits $(-30, 1)$. *There is an undershoot in the step response. For a unit step response, the maximum undershoot in magnitude is required not to exceed* 6. *Design a PID controller with an anti-windup mechanism to achieve the performance specification. The implementation is to use the IPD controller structure where the proportional control and derivative control are implemented on the output only.*

Solution. *We need to choose a desired closed-loop performance in the design. We choose the simplest formulation in the design, so that we can easily adjust the parameters:*

$$A_{cl} = (s^2 + 2\xi w_{\text{n}} s + w_{\text{n}}^2)(s + \lambda_1)^2 \tag{4.44}$$

where $\xi = 0.707$ *and* $w_{\text{n}} = \lambda_1$. *We need a few iterations in the design to get it right.*

First attempt in the design

We start with $w_{\text{n}} = 5$, *and design the PID controller using the pole-assignment design technique. The PID controller parameters are*

$$K_{\text{c}} = -3.0578; \tau_I = 1.4166; \tau_{\text{D}} = 0.3251; \tau_{\text{f}} = 3.4536e - 004$$

where the MATLAB program PIDplace.m is used to calculate the parameters (see Tutorial 3.2).

The closed-loop control system is simulated first without any limits on the control signal. Here we used a very small $\Delta t = 0.0002$ *to avoid the numerical errors induced by discretization of the PID controller (note that the filter time constant is very small). The simulation results are shown in Figure 4.6, which is seen to have a minimum of the control signal of about* -70, *which violates the lower limit; in addition, the undershoot of the output is about* -15, *which violates the undershoot requirement. We added the operational limits with the anti-windup mechanism, and compare the closed-loop responses with the previous case. It is seen that the limits of the control signal are satisfied. However, the undershoot of the output is reduced to* -7, *which still violates the requirement.*

Second attempt in the design

The parameter that we can make an adjustment is the desired closed-loop performance. It is known that if the desired closed-loop performance produces a slower closed-loop response, then the undershoot will be reduced. We choose $w_{\text{n}} = \lambda_1 = 2$ *in the second attempt. With this closed-loop performance specification, the PID controller parameters are:*

$$K_{\text{c}} = -2.8599; \tau_I = 1.4631; \tau_{\text{D}} = 0.3211; \tau_{\text{f}} = 0.0122.$$

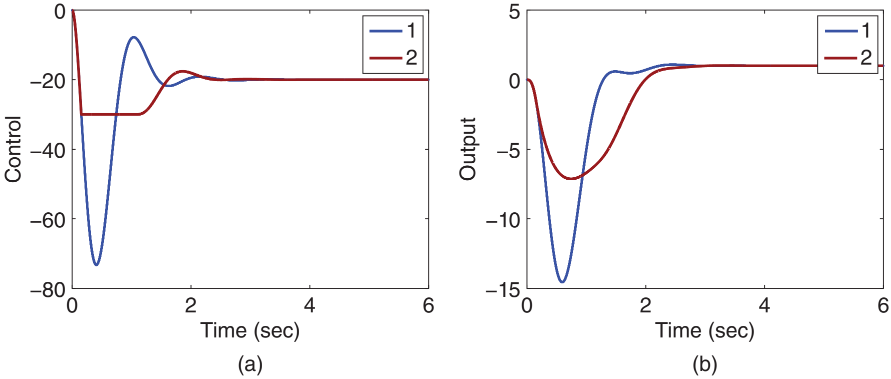


Figure 4.6 Closed-loop response (Example 4.4). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response without limits; line (2) closed-loop response with control signal saturation.

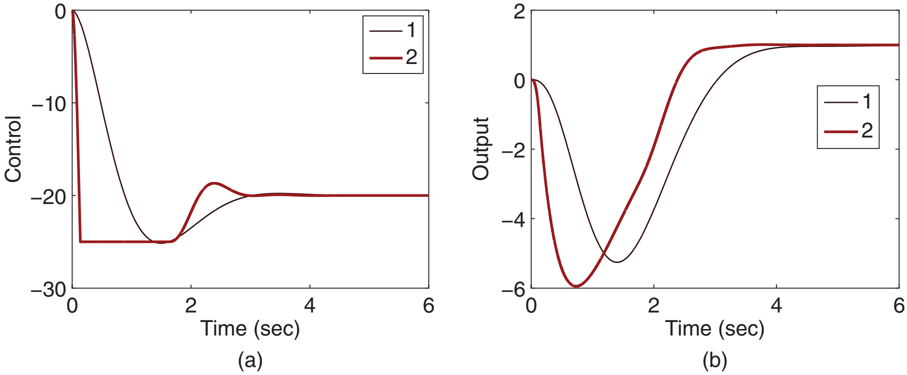


Figure 4.7 Closed-loop response (Example 4.4). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response with $w_n = 2$; line (2) closed-loop response using anti-windup control with $w_n = 5$.

The closed-loop simulation results are shown in Figure 4.7. With this choice of $w_n = 2$, *the minimum of the control signal has been reduced to* -25, *which is within the limit, also the magnitude of the undershoot has been reduced to be less than* 5.5. *Thus, all design requirements are satisfied.*

Third attempt in the design

From the first attempt, we learnt that by reducing the minimum value of the control signal, the magnitude of the undershoot would also be reduced. We could also change the limits of the control signal to $(-25, 1)$, *but maintain the desired closed-loop performance specification as* $w_n = \lambda_1 = 5$. *With the anti-windup scheme, the closed-loop response is compared in Figure 4.7. All limits of control signal and output signal are satisfied. In addition, the output response is faster in reaching the reference signal, in comparison to the results obtained in the second attempt.*

4.6.2 Limits on the Rate of Change of the Control Signal

Similarly, we can also impose limits on the rate of change on the control signal. This set of limits is typically specified as

$$Du^{\min} \le \dot{u}(t) \le Du^{\max}.$$

In the computation, we calculate $\dot{u}(t)$ using the approximation,

$$\begin{aligned}\dot{u}(t) &\approx \frac{u(t_i) - u(t_{i-1})}{\Delta t}\\ &= \frac{u_{\text{act}}(t_i) - u_{\text{act}}(t_{i-1})}{\Delta t}\end{aligned} \tag{4.45}$$

as the current sample $u_{\text{act}}(t_i)$ and the past sample $u_{\text{act}}(t_{i-1})$ share the same steady-state value. The idea is to calculate the actual control signal first and then check if the limits on the derivative are violated. If not, then the control signal is accepted. Otherwise, if the derivative is less than the lower limit ($\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} < Du^{\min}$), then the derivative is made equal to the lower limit ($\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} = Du^{\min}$), and the current control signal is calculated using this information; if the derivative is larger than the upper limit ($\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} > Du^{\max}$), then the derivative is made equal to the upper limit ($\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} = Du^{\max}$) and the current control signal is calculated based on the maximum derivative of the system allowed. With these computation equations, the requirements of an anti-windup mechanism are satisfied because the integral action is capped and the actual control signal is equal to the computed control signal with information of saturation embedded in the previous sample of control signal $u_{\text{act}}(t_{i-1})$.

The limits will be imposed using the following computational steps.

1. Calculate the actual control signal using

$$\begin{aligned}u_{\text{act}}(t_i) &= u_{\text{act}}(t_{i-1}) + K_{\text{c}}(-y_{\text{act}}(t_i) + y_{\text{act}}(t_{i-1}))\\ &\quad + \frac{K_{\text{c}}\Delta t}{\tau_{\text{I}}}(r_{\text{act}}(t_i) - y_{\text{act}}(t_i)) - u_{\text{D}}(t_i) + u_{\text{D}}(t_{i-1}).\end{aligned} \tag{4.46}$$

2. Check if the derivative of the control signal is within the limits

$$Du^{\min} \le \frac{u_{\text{act}}(t_i) - u_{\text{act}}(t_{i-1})}{\Delta t} \le Du^{\max}.$$

   If the constraints are satisfied, then this is the actual control signal to the plant. If not, we calculate the control signal using one of following steps.

   - If $\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} < Du^{\min}$, then
   $$u_{\text{act}}(t_i) = u_{\text{act}}(t_{i-1}) + Du^{\min}\Delta t;$$
   - if $\frac{u_{\text{act}}(t_i)-u_{\text{act}}(t_{i-1})}{\Delta t} > Du^{\max}$, then
   $$u_{\text{act}}(t_i) = u_{\text{act}}(t_{i-1}) + Du^{\max}\Delta t.$$

4.6.3 Food for Thought

1. What would be the essential elements in implementing the anti- windup mechanism for control amplitude limits when using a velocity form of a PID controller?

2. Is it correct to say that in the implementation using the velocity form, the integral action is automatically switched off when the control signal reaches its limit after two samples consecutively?
3. When the derivative of the control signal $\dot{u}(t)$ reaches its limit, what behavior would you anticipate for the control signal?
4. Is it possible for both the control signal and the derivative of the control signal reaching their limits at the same sampling time?

4.7 Tutorial on PID Anti-windup Implementation

Tutorial 4.1 *This tutorial is to illustrate how to implement the PID control algorithm in real-time. The core of this activity is to produce a MATLAB embedded function that can be used in a Simulink simulation as well as in xPC Target implementation. This embedded function is based on the PID controller velocity form discussed in the previous sections. The entire embedded MATLAB completes one cycle of computation for the control signal. For every sampling period, it will repeat the same computation procedure. The embedded function is written in a set of general variables and it is suitable for all PID control applications.*

Step by Step

1. *Create a new Simulink file called PIDV.slx*
2. *In Simulink's directory of User-Defined Functions, find the icon of the embedded MATLAB function and copy it to the PIDV model.*
3. *Click on the icon of the embedded function, and define the input and output variables and the controller parameters to the PIDV model so that the embedded function has the following form:*

   ```
   function uCurrent = PIDV(yCurrent,rCurrent,Kc,tauI,tauD,
   ...tauf,deltat,Umin,Umax,Dumin,Dumax)
   ```

   where uCurrrent is the calculated control signal at the sampling time t_i, the first two elements (yCurrent and rCurrent) among the input variables are the measurement of the output and the reference signals at sampling time t_i, Kc, tauI and tauD are the proportional control gain, integral time constant and derive control gain, tauf is the derivative filter time constant, deltat is the sampling interval and Umin and Umax are the upper and lower limits imposed on the control signal uCurrent, Dumin and Dumax are the upper and lower limits imposed on the derivative of the control signal.
4. *We need to edit the input and output data ports in order to let the embedded function know which input ports are the real-time variables and which are the parameters. This editing task is performed using Model Explorer.*
   - *click on 'yCurrent', on Scope, select 'input', assign port '1' and size '−1', complexity 'Inherited', type 'Inherit: Same as Simulink'. Repeat the same editing procedure for the reference signal 'rCurrent'.*
   - *The rest of the inputs to the embedded function are the parameters required in the computation. Click on 'Kc', on Scope, select 'Parameter' and click 'Tunable'*

and click 'Apply' to save the changes. Repeat the same editing procedure for the rest of the parameters.

- *To edit the output port from the embedded function, click on 'uCurrent', on Scope, select 'Output', Port '1', Size '−1', Sampling Model 'Sample based', Type 'Inherit: Same as Simulink', and click on 'Apply' to save the changes.*

5. *In the following, the program will declare those variables that are stored in the embedded function during each iteration for their dimensions and initial values. 'uPast' is the past control signal ($u(t_{i-1})$), 'yPast' is the past output signal ($y(t_{i-1})$). Because the PID controller implemented using the velocity form takes the actual measured variables, the past input and output variables should be initialized to the actual measured physical variables before the closed-loop control. These initial values can enter the computation as the extra parameters in the embedded function. Here, for simplicity of the programming, they are assigned to zero. Enter the following program into the file:*

```
persistent uPast
if isempty(uPast)
    uPast=0;
end
persistent yPast
if isempty(yPast)
    yPast=0;
end
persistent uDPast
if isempty(uDPast)
    uDPast=0;
end
persistent rPast
if isempty(rPast)
    rPast=0;
end
```

6. *Calculate the filtered derivative of the control signal (see 4.25). Enter the following program into the file:*

```
uDCurrent=tauf/(tauf+deltat)*uDPast+..
(Kc*tauD)/(tauf+deltat)*(yCurrent-yPast);
```

7. *Calculate the actual control signals (see (4.27)). Enter the following program into the file:*

```
 uCurrent=uPast+Kc*(rCurrent-yCurrent-rPast+yPast)+..
 (Kc*deltat)/tauI*(rCurrent-yCurrent)-uDCurrent+uDPast;
```

Alternatively, if one wishes to reduce overshoot in the output response by putting the proportional control on output only (see (4.28)), the following computation is used instead,

```
 uCurrent=uPast+Kc*(-yCurrent+yPast)+..
 (Kc*deltat)/tauI*(rCurrent-yCurrent)-uDCurrent+uDPast;
```

8. *Impose constraints on the derivative of the control signal. Enter the following program into the file:*

```
Du=(uCurrent-uPast)/deltat;
 if (Du>Dumax)  uCurrent=uPast+Dumax*deltat; Du=Dumax;end
 if (Du<Dumin) uCurrent=uPast+Dumin*deltat;Du=Dumin; end
```

9. *Impose constraints on the amplitude of the control signal. Enter the following program into the file:*

```
if (uCurrent>Umax) uCurrent=Umax;end
    if (uCurrent<Umin) uCurrent=Umin; end
```

10. *Update the past control and output signals. Updating uPast is part of the anti-windup implementation for the constraints so that when the control signal reaches its limit, the integral action is stopped. Enter the following program into the file:*

```
uDPast=uDCurrent;
uPast=uCurrent;
yPast=yCurrent;
rPast=rCurrent;
```

11. *Test this program with the DC motor control example (see Example 4.5).*

For PI controller implementation, the program is simply modified by removing the parameters τ_D and τ_f from the input, followed by deleting the calculation of derivative control (see Step 6) and removing it from the calculation of the control signal.

Example 4.5 *A DC motor model is given by the transfer function:*

$$G(s) = \frac{0.5}{(s+2)s} \tag{4.47}$$

where the input is voltage and the output is the angular position. The requirement is that the angular position follows a ramp signal of a unit slope without steady-state error, and the operational requirements are that the control signal is within the limits of $(-7, 5)$, and the derivative of the control signal is within the limits of ± 20. Design a PID controller with anti-windup mechanism for this system. The desired closed-loop performance is determined using the desired closed-loop polynomial $(s^2 + 2\xi w_n s + w_n^2)(s + \lambda_1)^2$, where $\xi = 0.707$ and $w_n = \lambda_1 = 3$. Also, investigate the approach that uses a smaller controller gain to reduce both $|u(t)|$ and $|\dot{u}(t)|$ and compare the results with the anti-windup control.

Solution. *With this desired closed-loop performance, we design the PID controller using the MATLAB program PIDplace.m and obtain the controller parameters as*

$$K_c = 19.9831; \tau_I = 1.0167; \tau_D = 0.2061; \tau_f = 0.1213.$$

We will use the PID controller structure with both proportional and integral terms implemented on the feedback error, while the derivative term is implemented on the output.

The limits given for the control signal are all violated when using this PID controller (see Figure 4.8). The anti-windup mechanism takes effect and the results are compared in Figure 4.8.

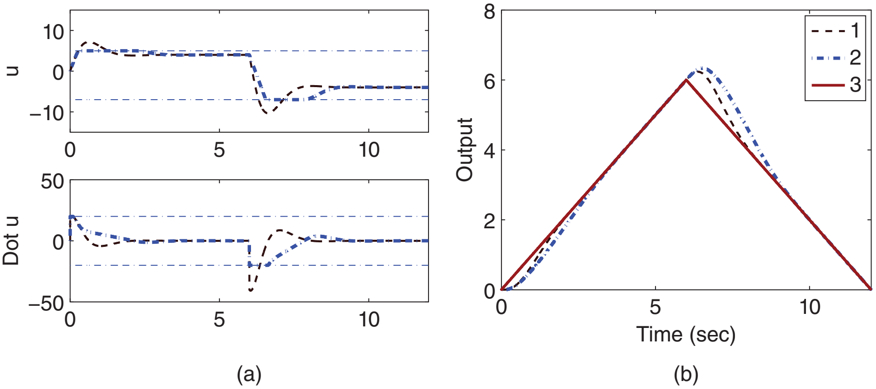


Figure 4.8 Closed-loop response (Example 4.5). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response without limits; line (2) closed-loop response using anti-windup control with $w_n = 3$; line (3) reference signal.

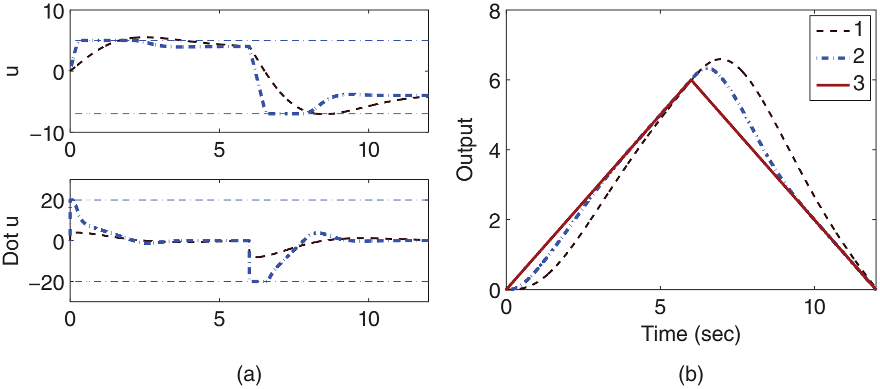


Figure 4.9 Closed-loop response (Example 4.5). (a) Input signal. (b) Output signal. Key: line (1) closed-loop response without limits ($w_n = 1$); line (2) closed-loop response using anti-windup control with $w_n = 3$; line (3) reference signal.

Without anti-windup control, we need to reduce the parameters w_n and λ_1 to reduce the magnitudes of the control signal and the derivative of the control signal. By selecting $w_n = \lambda_1 = 1$, both control signal and the derivative of the control signal are within the operational limits specified in the design. Figure 4.9 shows the closed-loop control results. It is seen that the track performance is significantly deteriorated in comparison with the results obtained from anti-windup control. The results show that the anti-windup control, if implemented properly, will allow us to use a higher gain in closed-loop control, with safety protection of the equipment.

4.8 Dealing with Other Implementation Issues

When using the velocity form for implementing the PID controllers, the issues, such as plant start-up from a non-zero steady-state operating conditions and quantization errors, can be handled in a relatively easy manner.

4.8.1 Plant Start-up

One of the issues in implementation of a PID control system is how to turn on the closed-loop control. This step is often linked with plant start-up, particularly when the plant has nonlinearities in the start-up region. There are a few possibilities in starting up the plant and the closed-loop PID controller, depending on whether the plant is stable and how much nonlinearity the plant has in the start-up region. In the start-up phase, accuracy in reference following and disturbance rejection are not the main concerns. For an industry plant, the time of start-up phase should be as short as possible to increase production rate and reduce plant downtime. There are a few scenarios discussed below.

- The plant is linear over the start-up phase and within the operating region so that the same linear controller can cover both the start-up operation and the normal operation. If all the scenarios are true, then the closed-loop PID control system can be turned on in the start-up phase.
- In many applications, at the start-up phase, the plant is extremely nonlinear, such as deadband within the start-up region. The PID controller designed for normal operation may not be working well in this region. If it is deadband we are dealing with, and if the plant is stable, a common strategy is to operate the plant in open loop without feedback control and inject a step input signal directly to the system (u is a step signal). By choosing a large amplitude for the step signal, the plant response is brought out from the deadband as quickly as possible. This is to avoid the scenario that the closed-loop system becomes unstable in the initial phase because of the plant nonlinearity. With open-loop start-up, the PID controller will be turned on when the system reaches the region for that it is designed.
- Proportional controller is a good candidate in the start-up region, if the plant is not stable or has slow dynamics. With a proportional controller, the proportional gain K_c can be chosen by trial-and-error, thus it is convenient to tune the closed-loop dynamics for the start-up region. The PID controller designed for normal operation will be switched on once the system response reaches the region for which the PID controller is designed.

When there is a need to switch between the open-loop operation to closed-loop control, or switch between two different controllers, there are two points that should be taken into consideration: the time for the switching and how the switching is to be conducted. Another important point is to make sure that when the switching occurs, the control signal from the previous operation should be close to the one used in the current operation so to avoid discontinuity.

To make sure the continuity between the open-loop operation and the closed-loop operation, the PID controller in velocity form is a good candidate. Because the PID controller in velocity form takes the actual plant input and output signals in the computation, without additional effort the transition from the two operational states can be made smooth (we could call it "bumpless transfer").

A simple switching policy is illustrated in the following. Here, a parameter λ is defined as the signal for switching. When $\lambda = 0$, the system operates in open-loop state, and when $\lambda = 1$, the PID controller controls the plant. The value of λ is assigned according to the operation of the plant. For instance, we can assign λ to 1 when the plant output variable $y_{\mathrm{act}}(t)$ reaches 50% of the full value. By defining $u_{\mathrm{op}}(t_i)$ as the open-loop control signal at the sampling time t_i, for example, a step input signal, the control signal with the switching from open-loop operation to closed-loop operation is calculated as

$$\begin{aligned} u_{\text{act}}(t_i) = (1 - \lambda)u_{\text{op}}(t_i) + \lambda[u_{\text{act}}(t_{i-1}) + K_{\text{c}}(-y_{\text{act}}(t_i) + y_{\text{act}}(t_{i-1})) \\ + \frac{K_{\text{c}}\Delta t}{\tau_{\text{I}}}(r_{\text{act}}(t_i) - y_{\text{act}}(t_i)) - u_{\text{D}}(t_i) + u_{\text{D}}(t_{i-1})] \end{aligned} \tag{4.48}$$

where the derivative term is computed using

$$u_{\text{D}}(t_i) = \frac{\tau_{\text{f}}}{\tau_{\text{f}} + \Delta t}u_{\text{D}}(t_{i-1}) + \frac{K_{\text{c}}\tau_{\text{D}}}{\tau_{\text{f}} + \Delta t}(y_{\text{act}}(t_i) - y_{\text{act}}(t_{i-1})). \tag{4.49}$$

This switching policy gives a switching between the two operations. When using it, we will define the parameter λ prior to the control application.

4.8.2 Dealing with Quantization Errors in PID Controller Implementation

Quantization errors occur in PID controller implementation when the actuators are restricted to a set of values. The topic of quantization error will be further discussed in Section 7.4. As an illustration, we assume that the actuator can only implement a control signal $u(t)$ with the step size of q. This quantization information can be embedded into the PID controller implementation as a constraint on $\Delta u(t_i) = u(t_i) - u(t_{i-1})$ so to reduce the effect of quantization errors on the PID controller performance.

We recall in Section 4.4 that the derivative of the control signal is expressed as

$$\dot{u}(t) = K_{\text{c}}\dot{e}(t) + \frac{K_{\text{c}}}{\tau_I}e(t) - \dot{u}_{\text{D}}(t) \tag{4.50}$$

At sampling time t_i, by approximating $\dot{u}(t_i)$ as $\frac{\Delta u(t_i)}{\Delta t}$, we obtain

$$\Delta u(t_i) = (K_{\text{c}}\dot{e}(t_i) + \frac{K_{\text{c}}}{\tau_I}e(t_i) - \dot{u}_{\text{D}}(t_i))\Delta t. \tag{4.51}$$

In the implementation, at sampling time t_i, the step size of the quantization q is checked against the computed incremental control signal. If

$$|(K_{\text{c}}\dot{e}(t_i) + \frac{K_{\text{c}}}{\tau_I}e(t_i) - \dot{u}_{\text{D}}(t_i))\Delta t| < q \tag{4.52}$$

i.e. the computed incremental control signal $|\Delta u(t_i)|$ is smaller than the step size q, then the actual incremental control signal $\Delta u(t_i) = 0$ and

$$u(t_i) = u(t_{i-1}).$$

However, if the computed incremental control signal $|\Delta u(t_i)|$ is greater than or equal to the step size q, which is,

$$|(K_{\text{c}}\dot{e}(t_i) + \frac{K_{\text{c}}}{\tau_I}e(t_i) - \dot{u}_{\text{D}}(t_i))\Delta t| \geq q \tag{4.53}$$

then $\Delta u(t_i)$ is calculated according to the constraint from the quantization step by using the following expression:

$$\Delta u(t_i) = q \times \text{round}\left(\frac{(K_{\text{c}}\dot{e}(t_i) + \frac{K_{\text{c}}}{\tau_I}e(t_i) - \dot{u}_{\text{D}}(t_i))\Delta t}{q}\right) \tag{4.54}$$

where the round(x) is to round the variable x to the nearest integer.

After the incremental control signal is computed with the quantization information, the control signal is updated as before using

$$u(t_i) = u(t_{i-1}) + \Delta u(t_i).$$

This implementation will reduce the impact of quantization error by limiting the error to each sampling instant and informing the controller with the actually implemented control signal. The PID controller implementation with quantization has been tested in the application of beer filtration process in Section 8.3.5.

With the variable expressions defined in Tutorial 4.1, small modification of the MATLAB codes PIDV.slx will implement a PID controller to reduce the effect of quantization errors. Firstly, we will add the quantization step q as an input parameter, and replace step 7 in Tutorial 4.1 by the following codes:

```
uDCurrent=tauf/(tauf+deltat)*uDPast+..
(Kc*tauD)/(tauf+deltat)*(yCurrent-yPast);
 Deltau=Kc*(-yCurrent+yPast)+..
(Kc*deltat)/tauI*(rCurrent-yCurrent)-uDCurrent+uDPast;
   if abs(Deltau)<q; Deltau=0;
else
   Deltau=q*round(Deltau/q);
 end
 uCurrent=uPast+Deltau;
```

where q is the quantization step.

4.9 Summary

Because PID controllers are continuous-time controllers, their implementations require the discretization process as discussed in this chapter. The discretization process is performed according to their time domain signals for the proportional control, the integral control and the derivative control with filter, respectively. This leads to a greater flexibility in choosing which the structures of PID controller we actually want to implement. For instance, if we wish to put the proportional controller on the output only, the control signal calculation from the proportional controller simply opted out the reference signal. Because the PID controllers are designed in the continuous time for discrete time implementation, the sampling rate needs to be selected as fast as possible. The other important aspects are summarized as follows.

- There are the position form and the velocity form in the choice of implementation algorithms. When using the position form, one needs to be careful with the steady-state control signal if it is not zero. The velocity form takes care of this problem by using the first sample of the actual control signal as an estimate to the steady-state control signal value.
- Integrator windup problem exists in PID controlled systems when the control signal reaches saturation limits. For the position form implementation, the PID controller needs to be re-written to produce a stable polynomial realization to avoid the integrator windup problem. For the velocity form implementation, the problem is solved by

simply including the control signal saturation information in the past control signal for the computation of current control signal. This effectively switches off the integral action when the control signal reaches the saturation limit.

- The strategies for plant start-up and dealing with quantization errors can be devised in a relatively simple manner when using the velocity form implementation.
- The Simulink program PIDV.slx produced in the MATLAB tutorial is used in the later chapters for closed-loop simulation studies. This program has also been successfully used in many applications where it is translated into C- program for micro-controllers.

4.10 Further Reading

1. A book on continuous-time control system design with discrete-time implementation is Middleton and Goodwin (1990).
2. A general anti-windup control system implementation is introduced and analyzed in Goodwin et al. (2000), in Zaccarian and Teel (2011). Unified framework is proposed for anti-windup control system implementation (Kothare et al. (1994)). An overview on anti-windup mechanisms is given in Peng et al. (1996), Tarbouriech and Turner (2009), and Galeani et al. (2009). An anti-windup implementation for PID controller is proposed in Visioli (2003).

Problems

4.1 *The Laplace transfer function of the control signal $U(s)$ is in relation to the feedback error $E(s)$:*

$$U(s) = 10(1 + \frac{1}{s})E(s)$$

1. *Assuming that the sampling interval is $\Delta t = 0.1$, write down the implementation equation for the PI controller with the velocity form.*
2. *Now, suppose that the feedback error $e(t_0) = -1$, $e(t_1) = -0.5$, $e(t_2) = 0.3$, $e(t_3) = -0.4$, $e(t_4) = 0.1$, and at t_0, $u(t_0)$= 0.5. We also assume that the control signal is limited to ± 0.6 ($-0.6 \leq u(t) \leq 0.6$). Calculate the control signal $u(t_1)$, $u(t_2)$, $u(t_3)$ and $u(t_4)$.*

4.2 *Following Tutorial 4.1 for PID controller implementation with an anti-windup mechanism, MATLAB real-time functions can be written for future Simulink based simulations. Verify the PIDV.slx program by generating the simulation results presented in Example 4.5.*

4.3 *A DC motor model is given by the transfer function:*

$$G(s) = \frac{0.6}{(s+3)s} \tag{4.55}$$

where the input is voltage and the output is the angular position. The requirement is that the angular position follows a ramp signal of a unit slope without steady-state error, and the operational requirements are that the control signal is within the limits of $(-7, 5)$*, and the derivative of the control signal is within the limits of* ± 20*. The desired closed-loop performance is determined using the desired closed-loop polynomial* $(s^2 + 2\xi w_n s + w_n^2)(s + \lambda_1)^2$*, where* $\xi = 0.707$ *and* $w_n = \lambda_1 = 3.5$*.*

1. *Design a PID controller for this system.*
2. *Implement the PID controller using velocity form with anti-windup mechanisms using PIDV.slx with both proportional and integral control on the reference signal.*
3. *Implement the IPD controller by putting integral control on the reference, both proportional and derivative control on the output. Will this IPD controller track the ramp reference signal without steady-state error? If not, where did the mistake occur?*

4.4 *For the same system presented in Problem 4.3, investigate the PID controller design that leads to a smaller controller gain to reduce both* $|u(t)|$ *and* $|\dot{u}(t)|$ *so that these signals are within the operational limits. Here we can reduce the parameters* w_n *and* λ_1 *to achieve the requirements. Simulate the closed-loop response and compare the results with the anti-windup control presented in Problem 4.3.*

4.5 *A continuous time system is described by the following transfer function:*

$$G(s) = \frac{0.2}{s^2 + 0.1s + 1}$$

1. *Design a PID controller with filter for this continuous time system, where all desired closed-loop poles are chosen to be* -1*.*
2. *Suppose that the actuator can only implement a control signal* $u(t)$ *with the step size of* $q = 0.1$*. With sampling interval* $\Delta t = 0.01$*, simulate the closed-loop response to reference following and disturbance rejection where a unit step reference signal enters the simulation at* $t = 0$ *and a negative unit step input disturbance enters the system at half of the simulation time.*

5

Disturbance Observer- Based PID and Resonant Controller

5.1 Introduction

In Chapter 1 to Chapter 4, we discussed PID control systems with explicit functions of proportional control, integral control, and derivative control. Because integral control embeds a marginally stable mode in the controller structure, which could cause the problem of integrator windup when the control signal reaches its saturation limits, the PID control system implementation requires modification to overcome this problem. Similar implementation problems to a worse degree are faced by the resonant controllers.

This chapter examines the PID controller design and resonant controller design from a different angle to the previous chapters. It introduces the integral mode and resonant modes through disturbance estimation. By doing so, the design becomes simpler, and more importantly, the implementations of the PID and resonant controllers flow naturally from the designs with anti-windup mechanisms. This development is particularly significant for resonant controllers because simplicity in the implementation with anti-windup mechanisms is paramount for practical applications.

5.2 Disturbance observer-Based PI Controller

This section introduces the idea of disturbance estimation that leads to an equivalent PI control system. The mathematical model used in this section is a first order model. For systems with higher order transfer functions, approximation is required.

5.2.1 Estimation of Disturbance with Control

We assume that there is a constant input disturbance $d(t)$, which is unknown. So the differential equation used to describe a first order system is given by

$$\dot{y}(t) = -ay(t) + b(u(t) + d(t)) \tag{5.1}$$

where a and b are model coefficients, and $u(t)$ and $y(t)$ are the input and output variables. Figure 5.1 illustrates the mathematical model to be used for the estimator based PI controller design.

Because of the assumption that $d(t)$ is a constant, we have

$$\dot{d}(t) = 0. \tag{5.2}$$

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

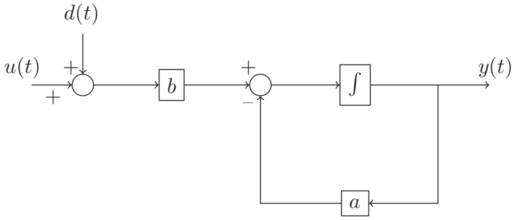


Figure 5.1 Block diagram of the system for a disturbance observer-based PI controller.

We would like to specify two desired closed-loop poles $-\alpha_1$ and $-\alpha_2$ where $\alpha_1 > 0, \quad \alpha_2 > 0$. In the proposed design, the choice of α_1 influences predominantly the proportional gain K_1 and α_2 predominantly the integral gain K_2.

5.2.1.1 Choice of Proportional Controller K_1

The choice of K_1 is straightforward. Firstly, we define:

$$\tilde{u}(t) = u(t) + d(t). \tag{5.3}$$

Then (5.1) becomes

$$\dot{y}(t) = -ay(t) + b\tilde{u}(t). \tag{5.4}$$

Together with proportional control

$$\tilde{u}(t) = -K_1 y(t)$$

we obtain the closed-loop system:

$$\dot{y}(t) = -(a + bK_1)y(t). \tag{5.5}$$

Equating the actual closed-loop pole $-(a + bK_1)$ to the desired closed-loop pole $-\alpha_1$, we find the proportional gain K_1 as

$$K_1 = \frac{\alpha_1 - a}{b}. \tag{5.6}$$

5.2.1.2 Compensation of Steady-state Error

We know that proportional control will lead to a closed-loop system with steady-state error for a constant reference signal or a disturbance signal. For overcoming the steady-state error, we need integral action incorporated into the control system. Here, we will estimate the steady-state error and then compensate it in the control signal. For this purpose, we extract the disturbance information from (5.1) leading to

$$bd(t) = \dot{y}(t) + ay(t) - bu(t). \tag{5.7}$$

One might attempt to directly calculate the unknown disturbance $d(t)$ using (5.7) and compensate for it in the control signal. However, it can be easily verified that such an approach fails to produce the control signal required because of uncertainties in the model parameters and other imperfections in practical applications. We will estimate the disturbance signal $d(t)$ with compensation on the error.

Let $\hat{d}(t)$ denote the estimated disturbance signal. The error, between what is given and what is to be estimated, is described by,

$$\epsilon(t) = bd(t) - b\hat{d}(t) = \dot{y}(t) + ay(t) - bu(t) - b\hat{d}(t). \tag{5.8}$$

With the gain K_2 weighted on the error $\epsilon(t)$, together with the assumption $\dot{d}(t) = 0$, we construct the estimation $\hat{d}(t)$ as

$$\frac{d\hat{d}(t)}{dt} = K_2(\dot{y}(t) + ay(t) - bu(t) - b\hat{d}(t)). \tag{5.9}$$

This is the so-called observer-equation. We will choose the gain K_2 such that the error $\tilde{d}(t) = d(t) - \hat{d}(t)$ converges to zero.

Note that

$$\frac{d\tilde{d}(t)}{dt} = -K_2 b\tilde{d}(t) \tag{5.10}$$

then, the parameter K_2 is chosen such that $-K_2 b = -\alpha_2$. This then leads to

$$K_2 = \frac{\alpha_2}{b}.$$

Hence,

$$\frac{d\tilde{d}(t)}{dt} = -\alpha_2\tilde{d}(t). \tag{5.11}$$

For any given initial condition $|\tilde{d}(0)| < \infty$ and $\alpha_2 > 0$, the estimation error $|\tilde{d}(t)| \to 0$ as $t \to \infty$. The convergence rate is dependent on the parameter α_2. The larger α_2 is, the faster the error will converge to zero.

Now, to calculate the control signal with compensation on the steady-state error, the unknown disturbance $d(t)$ in (5.3) is replaced by the estimated $\hat{d}(t)$ from (5.9), leading to the control signal calculated as

$$u(t) = -K_1 y(t) - \hat{d}(t). \tag{5.12}$$

5.2.1.3 The closed-loop poles

To verify indeed that the closed-loop poles for the control system are $-\alpha_1$ and $-\alpha_2$, substituting (5.12) into (5.1) gives

$$\dot{y}(t) = -(a + bK_1)y(t) + b\tilde{d}(t) = -\alpha_1 y(t) + b\tilde{d}(t) \tag{5.13}$$

where $\tilde{d}(t) = d(t) - \hat{d}(t)$. Together with (5.11), the closed-loop system equation is written as

$$\begin{bmatrix} \frac{dy(t)}{dt} \\ \frac{d\tilde{d}(t)}{dt} \end{bmatrix} = \overbrace{\begin{bmatrix} -\alpha_1 & b \\ 0 & -\alpha_2 \end{bmatrix}}^{A} \begin{bmatrix} y(t) \\ \tilde{d}(t) \end{bmatrix}. \tag{5.14}$$

Because A is an upper triangular matrix, the closed-loop poles (or eigenvalues) are simply calculated as the solution of the characteristic equation:

$$\det(sI - A) = (s + \alpha_1)(s + \alpha_2) = 0$$

where I is the identity matrix with dimensions 2 × 2, equal to $-\alpha_1$ and $-\alpha_2$.

5.2.1.4 Implementation procedure

Because the estimation equation (5.9) contains the derivative of the output signal $y(t)$, direct discretization requires information of $y(t_{i+1})$ at sampling time t_i, which is not available to us.

Let us define a variable $\hat{z}(t)$ as

$$\hat{z}(t) = \hat{d}(t) - K_2 y(t). \tag{5.15}$$

Then, substituting this variable into the estimation equation (5.9) yields

$$\begin{aligned}\frac{\mathrm{d}\hat{z}(t)}{\mathrm{d}t} &= -K_2 b\hat{z}(t) - (K_2^2 b - K_2 a)y(t) - K_2 bu(t)\\ &= -\alpha_2\hat{z}(t) - K_2(\alpha_2 - a)y(t) - \alpha_2 u(t).\end{aligned} \tag{5.16}$$

If the reference signal $r(t) \neq 0$, then both (5.15) and (5.16) are modified by replacing $y(t)$ with $y(t) - r(t)$. Furthermore, if the control signal $u(t)$ reaches its saturation limit, this saturation information is updated in the estimation of the disturbance $\hat{d}(t)$ via (5.16). Figure 5.2 illustrates the block diagram of the control system using disturbance observer. For discretizaton, at sampling time t_i, the derivative $\frac{\mathrm{d}\hat{z}(t)}{\mathrm{d}t}$ is approximated using the first order approximation as

$$\frac{\mathrm{d}\hat{z}(t)}{\mathrm{d}t} \approx \frac{\hat{z}(t_{i+1}) - \hat{z}(t_i)}{\Delta t}.$$

Then, it follows that

$$\hat{z}(t_{i+1}) = \hat{z}(t_i) - (\alpha_2\hat{z}(t_i) + \frac{\alpha_2(\alpha_2 - a)}{b}(y(t_i) - r(t_i)) + \alpha_2 u(t_i))\Delta t. \tag{5.17}$$

The calculation procedure for control signal is summarized as follows. Choose initial condition for $\hat{z}(t_0)$ at the beginning of the closed-loop operation. This could be selected to be zero if no other information is available. With the reference signal $r(t_i)$ and output signal $y(t_i)$ available at sampling time t_i, the control signal $u(t_i)$ is calculated recursively. The control signal is limited between $u^{\min}$ and $u^{\max}$.

1. Calculate the estimated disturbance signal at sample t_i as

$$\hat{d}(t_i) = \hat{z}(t_i) + K_2(y(t_i) - r(t_i)).$$

2. Calculate the control signal using the following equation

$$u(t_i) = -K_1(y(t_i) - r(t_i)) - \hat{d}(t_i).$$

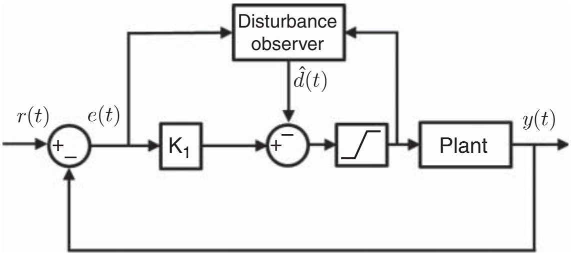


Figure 5.2 Block diagram of the control system using a disturbance observer.

3. Implement the control signal saturation:

$$u(t_i) = \begin{cases} u^{\min} & \text{if } u(t_i) < u^{\min} \\ u(t_i) & \text{if } u^{\min} \leq u(t_i) \leq u^{\max} \\ u^{\max} & \text{if } u(t_i) > u^{\max} \end{cases} \quad .$$

4. Update the estimation of disturbance $\hat{d}(t_{i+1})$ for the next sampling instant as

$$\hat{z}(t_{i+1}) = \hat{z}(t_i) - (\alpha_2 \hat{z}(t_i) + \frac{\alpha_2(\alpha_2 - a)}{b}(y(t_i) - r(t_i)) + \alpha_2 u(t_i))\Delta t.$$

5. Send the control signal $u(t_i)$ for implementation. When the next sampling period arrives, the new measurement of the output is taken and the computation is repeated from step 1.

5.2.2 Equivalence to PI controller

In order to calculate the equivalent PI controller, the Laplace transform of $\hat{Z}(s)$ is expressed as

$$\hat{Z}(s) = -\frac{K_2(\alpha_2 - a)}{(s + \alpha_2)} Y(s) - \frac{\alpha_2}{s + \alpha_2} U(s)$$

where $K_2 = \frac{\alpha_2}{b}$, and the Laplace transform $\hat{D}(s)$ is

$$\hat{D}(s) = \hat{Z}(s) + K_2 Y(s) = \frac{K_2(s + a)}{s + \alpha_2} Y(s) - \frac{\alpha_2}{s + \alpha_2} U(s). \tag{5.18}$$

The Laplace transform of the control signal $U(s)$ becomes

$$U(s) = -K_1 Y(s) - \frac{K_2(s + a)}{s + \alpha_2} Y(s) + \frac{\alpha_2}{s + \alpha_2} U(s)$$

which is

$$U(s) = -\left(\frac{K_1(s + \alpha_2)}{s} + \frac{K_2 s + K_2 a}{s} \right) Y(s) \tag{5.19}$$

Thus, the equivalent PI controller is revealed as

$$\begin{aligned} C(s) &= \frac{K_1(s + \alpha_2)}{s} + \frac{K_2 s + K_2 a}{s} \\ &= K_1 + K_2 + \frac{(K_1 \alpha_2 + K_2 a)}{s}. \end{aligned} \tag{5.20}$$

The PI controller parameters are

$$K_c = K_1 + K_2$$

$$\frac{K_c}{\tau_I} = K_1 \alpha_2 + K_2 a$$

where $K_1 = \frac{\alpha_1 - a}{b}$ and $K_2 = \frac{\alpha_2}{b}$. We can verify that the closed-loop poles are at $-\alpha_1$ and $-\alpha_2$, which was the design specification.

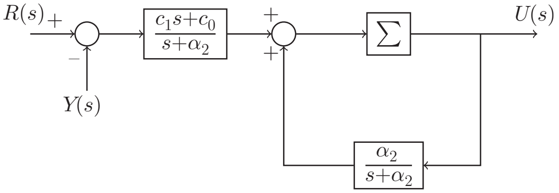

$R(s)$
+
−
$\frac{c_1 s + c_0}{s + \alpha_2}$
+
+
$\sum$
$U(s)$
$Y(s)$
$\frac{\alpha_2}{s + \alpha_2}$

Figure 5.3 Transfer function realization of the estimator based PI controller and $\sum$ is the saturation limiter.

Defining the coefficients $c_1 = K_c$ and $c_0 = \frac{K_c}{\tau_I}$ and the saturation limiter $\sum$, Figure 5.3 shows the transfer function realization of the PI controller with anti-windup mechanism, which calculates the control signal $U(s)$ based on the reference signal $R(s)$ and the output signal $Y(s)$.

5.2.3 MATLAB Tutorial for Implementation of a PI Controller via Estimation

The embedded PI controller via estimation will be used in the simulation studies with the Simulink environment.

Tutorial 5.1 *This tutorial is to illustrate how to implement the estimator based PI control algorithm in real-time. The core of this activity is to produce a MATLAB embedded function that can be used in a Simulink simulation as well as in MATLAB real time implementation. The entire embedded MATLAB completes one cycle of computation for the control signal. For every sampling period, it will repeat the same computation procedure.*

Step by Step

1. *Create a new Simulink file called PIEstim.slx*
2. *In Simulink's directory of User-Defined Functions, find the icon of embedded MATLAB function and copy it to the PIEstim model.*
3. *Click on the icon of the embedded function, and define the input and output variables to the PIEstim model so that the embedded function has the following form:*

   ```
   function uCur = PIEstim(r,y,K1,K2,alpha2,a,deltat,umin,umax)
   ```

   where uCur is the calculated control signal at the sampling time t_k, the first two elements (r and y) among the input variables are the measurements of the the reference and output signals at sampling time t_i, $K1$ and $K2$ are the controller gain and estimator gain, deltat is the sampling interval, and umin and umax are the lower and upper limits imposed on the control signal uCur.
4. *At the top of the embedded function, find Model Explorer among the Tools. When opening the Model Explorer, select Discrete for the Update method and input deltat into the Sample time; select Support variable-size arrays; select Saturate on integer overflow; select Fixed point. Click Apply to save the changes.*
5. *We need to edit the input and output data ports in order to let the embedded function know which input ports are the real time variables and which are the parameters. This editing task is performed using Model Explorer.*
   - *click on r, on Scope, select Input, assign port 1 and size -1 complexity, Inherited type Inherit: Same as Simulink. Repeat the same editing procedure for the output signal y.*

- *The remaining five inputs to the embedded function are the parameters required in the computation. Click on K1, on Scope, select Parameter and click Tunable and click Apply to save the changes. Repeat the same editing procedure for the rest of the parameters.*
- *To edit the output port from the embedded function, click on uCur, on Scope, select Output, Port 1, Size −1, Sampling Model Sample based, Type Inherit: Same as Simulink, and click on Apply to save the changes.*

6. *In the following, the program will declare the variable that is stored in the embedded function during each iteration for their dimensions and initial values. zhat is the estimated disturbance at sampling time t_i, which needs an initial value and is assigned to zero. Enter the following program into the file:*

```
persistent zhat
if isempty(zhat)
    zhat=0;
end
```

7. *Update the estimation of $\hat{z}(t_{i+1})$. Enter the following program into the file:*

```
 dhat=zhat+K2*(y-r);
```

8. *Update the control signal by combining the proportional control with the disturbance estimation. Enter the following program into the file:*

```
uCur= -K1*(y-r)-dhat;
```

9. *Implement saturation with anti-windup mechanism. Enter the following program into the file:*

```
if (uCur>umax)
 uCur=umax;
 end
if (uCur<umin)
 uCur=umin;
 end
```

10. *Calculate the $\hat{z}(t_{i+1})$ for the next sample. Enter the following program into the file:*

```
zhat=zhat-alpha2*zhat*deltat-(y-r)*(alpha2-a)*K2*deltat..
-alpha2*uCur*deltat;
```

5.2.4 Examples for Estimator based PI Controllers

Example 5.1 *A continuous-time system is approximated by the following first order model:*

$$G(s) = \frac{0.1}{T_1 s + 1} \tag{5.21}$$

where the time constant T_1 is 10 s. It is known that the system has a variable delay of T_D which has a maximum value of 1 s and a neglected time constant T_2 with a maximum value of 5 s. Design an estimator based PI controller for this system and simulate *the closed-loop control performance for unit step reference change and rejection of input disturbance having amplitude of* 20.

Solution. *Because this system has neglected time delay and time constant, this model uncertainty will limit the specification of desired closed-loop performance. A good starting point is to select the dominant closed-loop pole equal to the known pole of system, which is at* -0.1*. Hence,* $\alpha_1 = 0.1$*. The second desired closed-loop pole* $-\alpha_2$ *is determined using closed-loop simulations with the MATLAB real-time function presented in Section 5.2.3.*

With $a = 0.1$*,* $b = 0.01$*, and* $\alpha_1 = 0.1$*, the parameter* K_1 *is calculated as*

$$K_1 = \frac{\alpha_1 - a}{b} = 0.$$

The parameter K_2 *is calculated as*

$$K_2 = \frac{\alpha_2}{b}.$$

With the parameter α_2 *being selected as* 0.1, 0.2, *and* 0.3, *three values of* K_2 *are calculated as* 10, 20, *and* 30.

By selecting the sampling interval $\Delta t = 0.01$ *(s) and using the Simulink simulator with the PIEstim.slx function built in Section 5.2.3, closed-loop simulation results are obtained for the worst case scenario where the plant transfer function used in the simulation is*

$$G(s) = \frac{0.1e^{-s}}{(10s+1)(5s+1)}.$$

In the simulation, the unit step reference signal enters the system at $t = 0$ *and the input disturbance enters the system at half of the simulation time. It is seen from the comparative results illustrated in Figure 5.4 that with the increase in* α_2*, the closed-loop response speed for both reference following and disturbance rejection has increased. One can verify that further increase of* α_2 *will lead to the increased oscillations in the closed-loop response.* $\alpha_2 = 0.3$ *is a good choice with respect to the model uncertainties.*

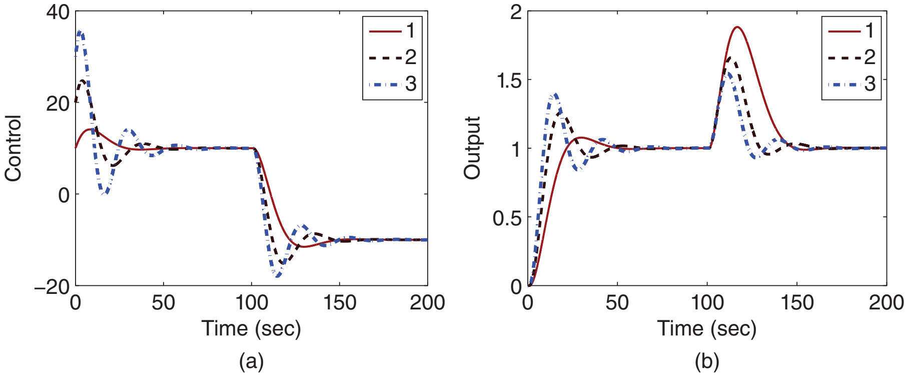


Figure 5.4 Comparison of closed-loop control performance using an estimator based PI controller with different α_2 values (Example 5.1). (a) Control signal. (b) Output. Key: line (1) $\alpha_2 = 0.1$; line (2) $\alpha_2 = 0.2$; line (3) $\alpha_2 = 0.3$.

It is apparent that much larger control amplitude is required when $\alpha_2 = 0.3$ is used for the estimator based PI controller design. In practice, this much larger control amplitude may not be realizable due to the physical limitations of the actuators. An interesting question arises as how the closed-loop response speed will change if the control signal amplitude is limited. The following example illustrates the comparative studies.

Example 5.2 *Continue from Example 5.1. Assuming that the control signal amplitude is constrained so that*

$$-11 \le u(t) \le 14. \tag{5.22}$$

Compare the closed-loop responses to a unit step reference signal and a step input disturbance with amplitude of 20.

Solution. *We examine the slowest response case with $\alpha_2 = 0.1$ and the fastest response case $\alpha_2 = 0.3$ where $\alpha_1 = 0.1$ is used for both cases.*

When $\alpha_2 = 0.1$, the maximum and minimum of the control signal are naturally satisfied except for a brief time in disturbance rejection (see Figure 5.5). Therefore, for the choice of $\alpha_2 = 0.1$ the closed-loop response to reference and disturbance signals remains unchanged with respect to the control signal constraints. However, when $\alpha_2 = 0.3$, the control signal amplitude exceeds its maximum and minimum values in reference following and disturbance rejection. In fact, the maximum value needs to be reduced from 35.60 *to* 14 *and the minimum value needs to be increased from* −17.92 *to* −11*. The constraints imposed will lead to a drastic reduction on the amplitude of the control signal when $\alpha_2 = 0.3$ is used in the controller design. Figure 5.5(a) shows the closed-loop control signals to the step reference signal and the disturbance signal. It is seen that indeed the control signals are within the limits. Figure 5.5(b) compares the closed-loop responses to the step reference signal and the disturbance signal. It is seen that the closed-loop responses to the step reference signal have a similar speed, however, the disturbance rejection with the larger α_2 is much faster.*

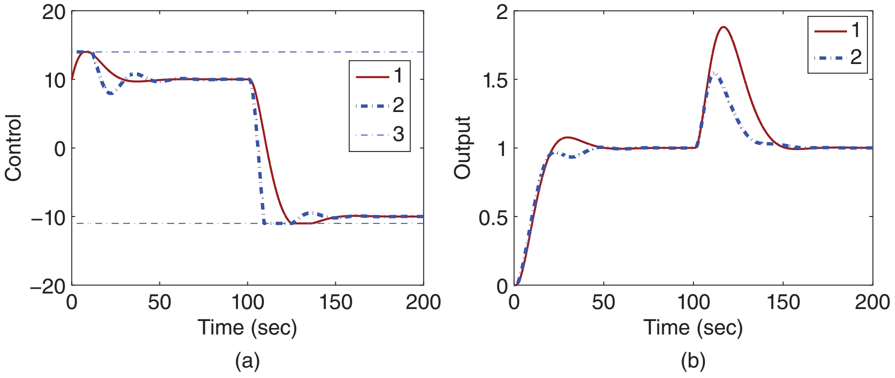


Figure 5.5 Comparison of closed-loop control performance using an estimator based PI controller with different α_2 values (Example 5.2). (a) Control signal. (b) Output. Key: line (1) $\alpha_2 = 0.1$; line (2) $\alpha_2 = 0.3$; and line (3) the limits of the control signal.

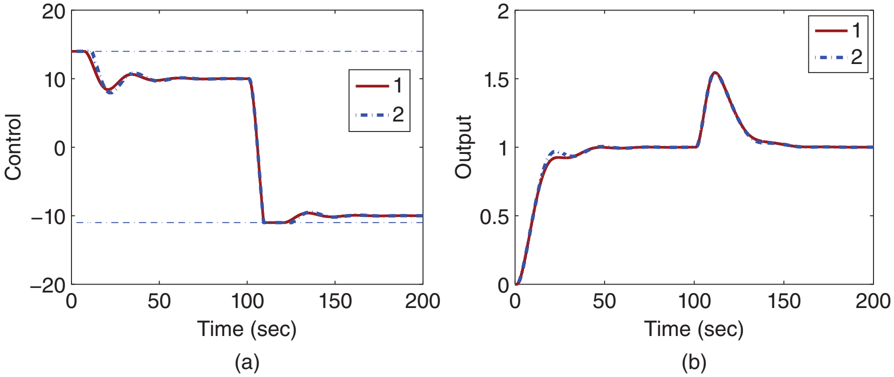


Figure 5.6 Comparison of closed-loop control performance between the PI controller in velocity form and the disturbance observer-based PI controller (Example 5.3). (a) Control signal. (b) Output. Key: line (1) the PI controller in velocity form; line (2) the disturbance observer-based PI controller.

The above two examples indicate that with the estimator based PI control system, the closed-loop performance limitation is limited by the model uncertainties due to unmodeled dynamics in the system. The performance limitation is reflected by the value of α_2. The control signal limits can be incorporated into the implementation of the control system with a naturally occurred anti-windup mechanism.

From Section 5.2.2 it is clear that there is an equivalent PI controller with parameters K_c and τ_I to this estimation based PI controller. The implementation of a PI controller using the velocity form was discussed with anti-windup mechanisms in Section 4.6. Naturally we wonder if these two implementations would lead to different outcomes. The following example compares the outcomes of these two PI controller with anti-windup mechanisms.

Example 5.3 *Continue from Examples 5.1 and 5.2. Evaluate the the original PI controller velocity form implementation with the estimation based PI controller implementation.*

Solution. *With* $\alpha_1 = 0.1$ *and* $\alpha_2 = 0.3$*, the proportional control gain and integral time constant are calculated as*

$$K_c = K_1 + K_2 = 30; \quad \frac{K_c}{\tau_I} = K_1\alpha_2 + K_2 a = 3; \quad \tau_I = 10$$

where $K_1 = 0$ *and* $K_2 = 30$*. Figures 5.6 (a)–(b) show that both implementations have incorporated an anti-windup mechanism although there is a tiny difference between the closed-loop performances when the saturation limit is reached.*

5.2.5 Food for Thought

1. In the design of a disturbance observer-based PI controller, what effect will the desired closed-loop poles $-\alpha_1$ and $-\alpha_2$ have on the closed-loop performance?
2. If the system has a large modelling error, would you decrease α_1 and α_2?

3. If the steady-state values of the control signal and output signal are not zero at the initial time, how would you enter these steady-state values in the disturbance observer-based PI controller implementation?
4. Is it possible to estimate the disturbance $d(t)$ without compensating it in the control system?
5. Is the implementation of integrator based on a stable system?

5.3 Disturbance observer-Based PID Controller

To design a PID controller using the estimation based approach, we consider a second order transfer function:

$$G(s) = \frac{b}{s^2 + a_1 s + a_0} = \frac{Y(s)}{U(s)} \tag{5.23}$$

where $U(s)$ and $Y(s)$ are the Laplace transforms of the input and output signals. We assume that $b \neq 0$. By assuming zero initial conditions, the corresponding differential equation is expressed as

$$\ddot{y}(t) = -a_1\dot{y}(t) - a_0 y(t) + bu(t).$$

In matrix form, it is expressed as

$$\begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u(t). \tag{5.24}$$

5.3.1 Proportional Plus Derivative Control

We will first design a proportional plus derivative controller. For this purpose, the feedback control signal $u(t)$ is

$$u(t) = -\begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}. \tag{5.25}$$

Substituting (5.25) into (5.24) leads to the closed-loop equation:

$$\begin{aligned} \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} &= \left(\begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ K_1 b & K_2 b \end{bmatrix} \right) \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -a_0 - K_1 b & -a_1 - K_2 b \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}. \end{aligned} \tag{5.26}$$

The closed-loop characteristic polynomial is then calculated as

$$\det\left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -a_0 - K_1 b & -a_1 - K_2 b \end{bmatrix} \right) = s(s + a_1 + K_2 b) + a_0 + K_1 b.$$

This is clearly a second order polynomial. We can specify a damping coefficient $\xi = 0.707$ and choose the parameter w_n as the closed-loop performance parameter. Alternatively, we could also choose two desired closed-loop poles as $-\alpha_1$ and $-\alpha_2$ where $\alpha_1 > 0$ and $\alpha_2 > 0$. In any case, to determine the proportional and the derivative controller gain, the actual closed-loop characteristic polynomial is made equal to the desired one, leading to

$$s^2 + (a_1 + K_2 b)s + a_0 + K_1 b = s^2 + 2\xi w_\mathrm{n} s + w_\mathrm{n}^2.$$

The solution of this polynomial equation gives the proportional control gain:

$$K_1 = \frac{w_\mathrm{n}^2 - a_0}{b} \tag{5.27}$$

and the derivative control gain:

$$K_2 = \frac{2\xi w_\mathrm{n} - a_1}{b}. \tag{5.28}$$

Since it is necessary to use a filter for the derivative action because of measurement noise, a quick way to calculate a first order filter time constant is to find the corresponding gain τ_D, which is

$$\tau_\mathrm{D} = \frac{K_2}{K_1}$$

based on (5.25). With this parameter, the derivative filter constant is

$$\tau_\mathrm{f} = \beta \tau_\mathrm{D}$$

where $\beta = 0.1$ is often selected. The filtered derivative output signal is expressed as

$$Y_\mathrm{df}(s) = \frac{s}{\tau_\mathrm{f} s + 1} Y(s).$$

For many applications, it is preferable to design the PD controller together with derivative filter τ_f to avoid the extra errors introduced from the approximation. A PD controller with filter design is discussed in detail in Section 3.4.1 with the filter constant calculated through the desired closed-loop performance specification.

5.3.2 Adding Integral Action

To add integral action to the PD controller, we assume that there is a constant input disturbance $d(t)$ so we have $\dot{d}(t) = 0$. The differential equation model (5.24) is modified to become:

$$\begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} (u(t) + d(t)). \tag{5.29}$$

Similar to the PI controller design via estimation, we will write the unknown disturbance term as

$$bd(t) = \ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) - bu(t). \tag{5.30}$$

Together with the assumption that $\dot{d}(t) = 0$, the estimation of $d(t)$ is constructed as

$$\frac{\mathrm{d}\hat{d}(t)}{\mathrm{d}t} = K_3(\ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) - bu(t) - b\hat{d}(t)). \tag{5.31}$$

We choose $\alpha_3 > 0$ and determine the estimator's gain K_3 using

$$K_3 = \frac{\alpha_3}{b}.$$

Because Equation (5.31) has double derivative and derivative of output signal $y(t)$, it is not suitable for computational purposes. To this end, we define a new variable

$$\hat{z}(t) = \hat{d}(t) + K_3\dot{y}(t)$$

and rewrite Equation (5.31) as function of $\hat{z}(t)$:

$$\frac{d\hat{z}(t)}{dt} = -\alpha_3\hat{z}(t) + K_3(a_1 - \alpha_3)\dot{y}(t) + K_3a_0y(t) - \alpha_3u(t). \tag{5.32}$$

Assuming that the reference signal is $r(t_i)$ at the sampling time t_i and the output signal and its derivative are measured as $y(t_i)$, $\dot{y}(t_i)$, with the sampling interval Δt, with saturation limits $u^{\min}$ and $u^{\max}$, the control signal is calculated using the following steps.

1. Update the estimated disturbance signal. An initial condition on $\hat{z}(t_0)$ will be given as the start up of the control algorithm.

$$\hat{d}(t_i) = \hat{z}(t_i) - K_3\dot{y}(t_i).$$

2. Calculate the control signal

$$u(t_i) = -K_1(y(t_i) - r(t_i)) - K_2\dot{y}(t_i) - \hat{d}(t_i).$$

3. Implement control signal saturation:

$$u(t_i) = \begin{cases} u^{\min} & if\ u(t_i) < u^{\min} \\ u(t_i) & if\ u^{\min} \leq u(t_i) \leq u^{\max} \\ u^{\max} & if\ u(t_i) > u^{\max} \end{cases} .$$

4. Update the disturbance estimator for t_{i+1}:

$$\hat{z}(t_{i+1}) = \hat{z}(t_i) + \Delta t(-\alpha_3\hat{z}(t_i) + K_3(a_1 - \alpha_3)\dot{y}(t_i)) \\ + \Delta t(K_3a_0(y(t_i) - r(t_i)) - \alpha_3u(t_i)).$$

5. When the next sample period arrives, repeat the computation at step 1.

Note that in the implementation of the PID controller, the derivative control is on the output only, avoiding producing a spike on the control signal when the reference signal signal $r(t)$ makes a step change. When a filter is used for the derivative signal $\dot{y}(t_i)$, the signal $\dot{y}(t_i)$ is replaced by $\dot{y}_f(t_i)$, which is illustrated in Tutorial 5.2.

5.3.3 Equivalence to a PID Controller

To find the equivalence of the proposed estimation based PID controller to the PID controller expressed in transfer function form, we examine the Laplace transform of the estimation equation (5.31), which is,

$$s\hat{D}(s) = K_3(s^2 + a_1s + a_0)Y(s) - \alpha_3U(s) - \alpha_3\hat{D}(s) \tag{5.33}$$

where $\alpha_3 = K_3b$. Solving for $\hat{D}(s)$ gives

$$\hat{D}(s) = \frac{1}{s + \alpha_3}(K_3(s^2 + a_1s + a_0)Y(s) - \alpha_3U(s)). \tag{5.34}$$

Note that the Laplace transform of the control signal is expressed as

$$U(s) = -K_1Y(s) - K_2sY(s) - \hat{D}(s). \tag{5.35}$$

By substituting (5.34) into (5.35), the Laplace transform of the control signal $U(s)$ is found as

$$U(s) = -\frac{s+\alpha_3}{s}(K_1 + K_2 s)Y(s) - \frac{K_3}{s}(s^2 + a_1 s + a_0)Y(s). \tag{5.36}$$

With the negative feedback, the equivalent controller transfer function $C(s)$ is obtained as

$$C(s) = \frac{(s+\alpha_3)(K_1 + K_2 s)}{s} + \frac{K_3(s^2 + a_1 s + a_0)}{s} = \frac{P(s)}{L(s)}. \tag{5.37}$$

Now, it can be verified that the closed-loop polynomial is indeed

$$A(s)L(s) + B(s)P(s) = (s+\alpha_3)(s^2 + 2\xi w_n s + w_n^2) \tag{5.38}$$

where $B(s)$ and $A(s)$ are the numerator and denominator of the transfer function model as given by (5.23). Additionally, it can be verified that the controller transfer function can be written in an equivalent form to a PID controller:

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s} \tag{5.39}$$

where the parameters c_2, c_1, and c_0 are calculated as

$$c_2 = K_2 + K_3 \tag{5.40}$$
$$c_1 = K_1 + K_2\alpha_3 + K_3 a_1 \tag{5.41}$$
$$c_0 = K_1\alpha_3 + K_3 a_0. \tag{5.42}$$

Figure 5.7 shows the transfer function realization of the disturbance observer-based PID controller where $\sum$ is the saturation limiter.

There are several comments related. Firstly, the relationship presented in (5.38) shows that there are three desired closed-loop poles for the PID controller designed. The pair of complex poles located at $-\xi w_n \pm j w_n\sqrt{1-\xi^2}$ with $\xi = 0.707$ or 1 is used to determine the parameters K_1 and K_2, and the pole located at $-\alpha_3$ is used to determine the parameter K_3. Secondly, the PID controller can be implemented using the estimation of the input disturbance, providing a stable implementation for the integrator. Thirdly, the controller transfer function shown in (5.39) can readily be used for frequency response analysis so that the gain margin, phase margin, and delay margin can be calculated for the controller designed.

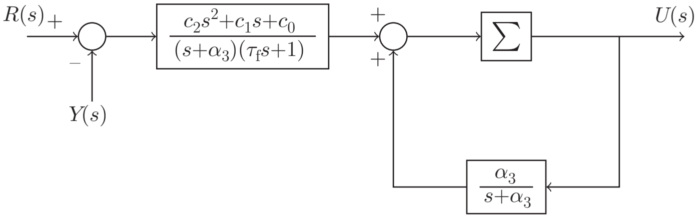

$R(s)$
+
−
$Y(s)$
$\frac{c_2s^2+c_1s+c_0}{(s+\alpha_3)(\tau_f s+1)}$
+
+
$\sum$
$U(s)$
$\frac{\alpha_3}{s+\alpha_3}$

Figure 5.7 Transfer function realization of the disturbance observer-based PID controller.

5.3.4 MATLAB Tutorial on the Implementation of a disturbance observer-based PID Controller

This section presents the MATLAB tutorial for the implementation of disturbance observer-based PID controller. This implementation will contain the anti-windup mechanism when the control signal reaches its maximum or minimum values. The embedded PID controller via estimation will be used in the simulation studies within the Simulink environment.

Tutorial 5.2 *This tutorial is to illustrate how to implement the disturbance observer-based PID control algorithm in real-time. The core of this activity is to produce a MATLAB embedded function that can be used in a Simulink simulation as well as in MATLAB Real-time implementation. The entire embedded MATLAB completes one cycle of computation for the control signal. For every sampling period, it will repeat the same computation procedure.*

Step by Step

1. *Create a new Simulink file called PIDEstim.slx*
2. *In Simulink's directory of User-Defined Functions, find the icon of embedded MATLAB function and copy it to the PIDEstim model.*
3. *Click on the icon of the embedded function, and define the input and output variables to the PIDEstim model so that the embedded function has the following form:*

   ```
   function uCur = PIDEstim(r,y,K1,K2,K3,tauf,alpha3,..
   a1,a0,deltat,umin,umax)
   ```

   where uCur is the calculated control signal at the sampling time t_i, the first two elements (r and y) among the input variables are the measurements of the the reference and output signals at sampling time t_i, $K1$, $K2$ and $K3$ are the PD controller gains and estimator gain, tauf is the time constant for the derivative filter, alpha3 is for the pole location of the estimator, deltat is the sampling interval, umin and umax are the lower and upper limits imposed on the control signal uCur.
4. *At the top of the embedded function, find Model Explorer among the Tools. When opening the Model Explorer, select discrete for the update method and input deltat into the sample time; select Support variable-size arrays; select Saturate on integer overflow; select Fixed point. Click Apply to save the changes.*
5. *We need to edit the input and output data ports in order to let the embedded function know which input ports are the real-time variables and which are the parameters. This editing task is performed using Model Explorer.*
   - *Click on r, on Scope, select input, assign port 1 and size -1, complexity Inherited, type Inherit: Same as Simulink. Repeat the same editing procedure for the output signal y.*
   - *The rest of 10 inputs to the embedded function are the parameters required in the computation. Click on $K1$, on Scope, select Parameter and click Tunable and click Apply to save the changes. Repeat the same editing procedure for the rest of the parameters.*

- *To edit the output port from the embedded function, click on uCur, on Scope, select Output, Port* 1*, Size* −1*, Sampling Model Sample based, Type Inherit: Same as Simulink, and click on Apply to save the changes.*

In the following, the program will declare the variables that are stored in the embedded function during each iteration for their dimensions and initial values. zhat is the estimated disturbance at sampling time t_i, which needs an initial value and is assigned to zero. We also declare the derivative of the filtered output and the past output signals that will be stored in the memory. Enter the following program into the file:

```
persistent zhat
if isempty(zhat)
    zhat=0;
end
persistent ydfpast
if isempty(ydfpast)
    ydfpast=0;
end
persistent ypast
if isempty(ypast)
    ypast=0;
end
```

6. *Update the filtered output derivative signal. Enter the following program into the file:*

```
ydf=tauf/(tauf+deltat)*ydfpast+1/(tauf+deltat)*(y-ypast);
```

7. *Update the estimated disturbance $\hat{d}(t_i)$. Enter the following program into the file:*

```
dhat=zhat+K3*ydf;
```

8. *Update the control signal by combining the proportional plus derivative control with the disturbance estimation. Enter the following program into the file:*

```
uCur= -K1*(y-r)-K2*ydf-dhat;
```

9. *Implement saturation with anti-windup mechanism. Enter the following program into the file:*

```
if (uCur>umax) uCur=umax;end
if (uCur<umin) uCur=umin;end
```

10. *Calculate $\hat{z}(t_{i+1})$ for the next sample. Enter the following program into the file:*

```
zhat=zhat-deltat*alpha3*zhat+...
deltat*(K3*(a1-alpha3)*ydf+K3*a0*(y-r)-alpha3*uCur);
```

11. *Update the past output signal and past filtered derivative output signal in preparation for the next sample period. Enter the following program into the file:*

```
ypast=y;
ydfpast=ydf;
```

This program is tested using the example presented in the next section.

5.3.5 Examples for Disturbance observer-based PID Controller

Example 5.4 *An unstable system is described by the transfer function:*

$$G(s) = \frac{0.1e^{-0.05s}}{(s+2)(s-2)} \tag{5.43}$$

where the small time delay is due to the dynamics from the actuator. Design a estimator based PID controller with the pair of dominant poles at -2 *and the estimator pole at* -10. *Evaluate the control system performance for reference following and disturbance rejection in the presence of control amplitude constraints.*

Solution. *Choosing* $\xi = 1$ *and* $w_n = 2$ *positions the closed-loop poles at* -2. *The model (5.43) gives the parameters* $a_1 = 0$, $a_0 = -4$, *and* $b = 0.1$. *The PD controller parameters are calculated as*

$$K_1 = \frac{w_n^2 - a_0}{b} = 80$$

and the derivative control gain

$$K_2 = \frac{2\xi w_n - a_1}{b} = 40.$$

Because it is an unstable system with neglected time delay, the selection of third pole $-\alpha_3$ *is considered with simulation studies. When* α_3 *is too large, the neglected time delay will lead to the instability of the closed-loop system. However, when* α_3 *is not large enough, and the control amplitude constraints could lead to instability. The range of* α_3 *is found to be between* 5 *and* 15 *to provide satisfactory closed-loop performance in the presence of control signal amplitude constraints. With* $\alpha_3 = 10$, *the parameter* K_3 *is calculated as*

$$K_3 = \frac{\alpha_3}{b} = 100.$$

A derivative filter time constant is calculated as

$$\tau_f = \frac{0.01K_2}{K_1} = 0.005.$$

The filter time constant used is quite small because it introduces additional dynamics in the system. If the system has severe noises and a larger filter time constant is required, then the derivative filter needs to be designed as part of the control system (see Section 3.4.1). The disturbance observer-based PID control system is evaluated through simulation studies using the MATLAB real-time function in Section 5.3.4 together with a Simulink program. Choosing the sampling interval $\Delta t = 0.001$ *(s), in the simulation, a unit step reference signal enters the closed-loop system at* $t = 0$ *and an input step disturbance with amplitude of* 20 *enters the system at half of the simulation time. The control signal amplitude is constrained between* -68 *and* 26. *Figures 5.8(a) and (b) show the closed-loop control performances in the presence of the control signal amplitude constraints. In comparison, the responses from the unconstrained control signal and the output signal are also illustrated in the same figures. It is seen that the control amplitude has been reduced from about* 100 *to* 26 *at the beginning of the simulation. For the disturbance rejection, the control signal was constrained from the original* -73

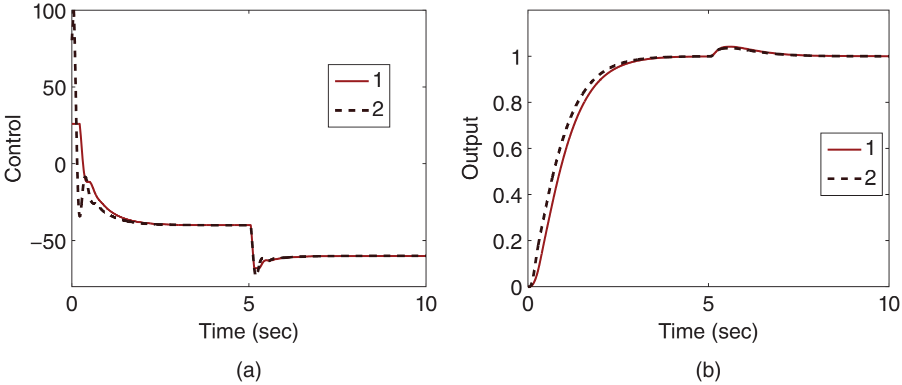


Figure 5.8 Closed-loop control performance using disturbance observer-based PID controller with control signal amplitude constraints (Example 5.4). (a) Control signal. (b) Output. Key: line (1) constrained responses; line (2) unconstrained responses.

to −68. The comparative results show that the constraints are met with little closed-loop performance deterioration.

One can also verify that the anti-windup PID controller implementation introduced in Section 4.6 will produce the same disturbance rejection results, however, will have a slower reference response if the proportional control is implemented on the output only. This is left as an exercise.

5.3.6 Food for Thought

1. In the design of disturbance observer-based PID controller, how would you modify Equations (5.29)–(5.32) to include the derivative filter?
2. Can you list three physical systems that have second order transfer functions and are suitable for a PD controller application?
3. How would you propose to include the steady-state values of control signal and output signal in the implementation of the disturbance observer-based PID controller?
4. Would you consider the possibility of adding the integral action after the PD controller has stabilize the system, which can be achieved subsequently by subtracting the disturbance term?

5.4 Disturbance observer-Based Resonant Controller

This section will investigate resonant controller design and implementation with anti-windup mechanism using the disturbance estimation approach.

5.4.1 Resonant Controller Design

Assume that a dynamic system is described by the differential equation:

$$\dot{y}(t) = -ay(t) + b(u(t) + d(t)) \tag{5.44}$$

where a and b are the coefficients; $u(t)$ and $y(t)$ are the input and output signals; $d(t)$ is the input disturbance signal. In particular, we assume that $d(t)$ is a sinusoidal signal with known frequency ω_0, but unknown amplitude d_m and phase angle ψ_0, which is expressed as

$$d(t) = d_\mathrm{m} \sin(\omega_0 t + \psi_0).$$

The resonant control law is expressed as

$$u(t) = -K_1(y(t) - r(t)) - \hat{d}(t)$$

where $\hat{d}(t)$ is an estimate of the unknown disturbance $d(t)$.

By choosing the desired closed-loop pole at $-\alpha_1$ and $\alpha_1 > 0$, the proportional feedback control gain K_1 is calculated as

$$K_1 = \frac{\alpha_1 - a}{b}.$$

Now, the next question is how to compute the estimated input sinusoidal disturbance signal $\hat{d}(t)$. The derivative of this disturbance signal is

$$\dot{d}(t) = d_\mathrm{m}\omega_0 \cos(\omega_0 t + \psi_0)$$

and its second derivative is

$$\ddot{d}(t) = -d_\mathrm{m}\omega_0^2 \sin(\omega_0 t + \psi_0) = -\omega_0^2 d(t).$$

Now, we choose $x_1(t) = d(t)$ and $x_2(t) = \dot{d}(t)$. With these choices, the following differential equations are used to describe the sinusoidal disturbance signal:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \tag{5.45}$$

In order to estimate the disturbance signal $d(t)$, from (5.44), we obtain

$$bd(t) = \begin{bmatrix} b & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \dot{y}(t) + ay(t) - bu(t)$$

which is the output equation for the estimation. Thus, the estimated variables $\hat{x}_1(t)$ and $\hat{x}_2(t)$ are constructed as

$$\begin{bmatrix} \dfrac{d\hat{x}_1(t)}{dt} \\ \dfrac{d\hat{x}_2(t)}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \left(\dot{y}(t) + ay(t) - bu(t) - \begin{bmatrix} b & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} \right) \tag{5.46}$$

where γ_1 and γ_2 are the estimator gains chosen for the design.

The next question is how to choose γ_1 and γ_2 such that the errors between the estimated and true disturbance signals are ensured to converge to zero as $t \to \infty$. For this purpose, we define $\tilde{x}_1(t) = x_1(t) - \hat{x}_1(t)$ and $\tilde{x}_2(t) = x_2(t) - \hat{x}_2(t)$. It can be verified as an exercise that we have the following error system:

$$\begin{bmatrix} \dfrac{d\tilde{x}_1(t)}{dt} \\ \dfrac{d\tilde{x}_2(t)}{dt} \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} - \begin{bmatrix} \gamma_1 b & 0 \\ \gamma_2 b & 0 \end{bmatrix} \right) \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} \tag{5.47}$$

where we have used the following relationship:

$$\begin{bmatrix} b & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \dot{y}(t) + ay(t) - bu(t).$$

Clearly, we can choose the parameters γ_1 and γ_2 such that the poles (or eigenvalues) of the error system are on the left half of the complex plane, which ensures its stability. The characteristic polynomial for the error system is calculated as

$$\det\left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -\gamma_1 b & 1 \\ -\omega_0^2 - \gamma_2 b & 0 \end{bmatrix}\right) = s^2 + \gamma_1 b + \omega_0^2 + \gamma_2 b.$$

Now, we choose a desired characteristic polynomial for performance specification as $s^2 + 2\xi w_n s + w_n^2$. Then, to make the characteristic polynomial for the error system equal to the desired characteristic polynomial, the coefficients γ_1 and γ_2 are found as,

$$\gamma_1 = \frac{2\xi w_n}{b};$$
$$\gamma_2 = \frac{w_n^2 - \omega_0^2}{b}. \tag{5.48}$$

In the applications, the damping parameter ξ is chosen to be 0.707 and the parameter w_n is adjusted for how fast we would like to see the errors converge to zero.

5.4.2 Resonant Controller Implementation

The calculation of the estimated input disturbance using (5.46) requires the derivative of the output signal $\dot{y}(t)$, which is not desirable in the implementation. To overcome the problem, we define a pair of new variables:

$$\hat{z}_1(t) = \hat{x}_1(t) - \gamma_1 y(t); \quad \hat{z}_2(t) = \hat{x}_2(t) - \gamma_2 y(t).$$

Then, from (5.46), the following two equations are obtained:

$$\frac{d\hat{z}_1(t)}{dt} = -2\xi w_n \hat{z}_1(t) + \hat{z}_2(t) + (a\gamma_1 + \gamma_2 - 2\xi w_n \gamma_1)y(t) - b\gamma_1 u(t) \tag{5.49}$$
$$\frac{d\hat{z}_2(t)}{dt} = -w_n^2 \hat{z}_1(t) + (a\gamma_2 - w_n^2\gamma_1)y(t) - b\gamma_2 u(t). \tag{5.50}$$

The control law presented above is a combination of proportional feedback and a disturbance observer. This section will present how this control law can be implemented in a discrete time environment with anti-windup mechanism.

The derivatives in the observer equations (5.49) and (5.50) are first discretized with a sampling interval Δt leading to their approximations at the sampling time t_i:

$$\frac{d\hat{z}_1(t)}{dt} \approx \frac{\hat{z}_1(t_{i+1}) - \hat{z}_1(t_i)}{\Delta t}; \quad \frac{d\hat{z}_2(t)}{dt} \approx \frac{\hat{z}_2(t_{i+1}) - \hat{z}_2(t_i)}{\Delta t}.$$

The following algorithm summarizes the computational procedure for the implementation of the resonant controller with anti-windup mechanism.

Assume that the control signal $u(t)$ is limited to $u^{\min}$ and $u^{\max}$, that is

$$u^{\min} \le u(t) \le u^{\max}.$$

Choosing the initial conditions for $\hat{z}_1$ and $\hat{z}_2$, the control signal is calculated iteratively according to the following steps, where $r(t_i)$ is the reference signal at sampling time t_i.

1. Calculate the estimated sinusoidal disturbance $\hat{d}(t_i)$ as
$$\hat{d}(t_i) = \hat{z}_1(t_i) + \gamma_1(y(t_i) - r(t_i)).$$
2. Calculate the control signal $u(t_i)$ as
$$u(t_i) = -K_1(y(t_i) - r(t_i)) - \hat{d}(t_i).$$
3. Implement the saturations on the control signal using
$$u(t_i) = \begin{cases} u^{\min} & \mathit{if}\ u(t_i) < u^{\min} \\ u(t_i) & \mathit{if}\ u^{\min} \le u(t_i) \le u^{\max} \\ u^{\max} & \mathit{if}\ u(t_i) > u^{\max} \end{cases} .$$
4. Update the estimated disturbance signals using the following equations:
$$\begin{aligned} \hat{z}_1(t_{i+1}) &= \hat{z}_1(t_i) + \Delta t(-2\xi w_\mathrm{n}\hat{z}_1(t_i) + \hat{z}_2(t_i)) \\ &\quad + \Delta t((a\gamma_1 + \gamma_2 - 2\xi w_\mathrm{n}\gamma_1)(y(t_i) - r(t_i)) - b\gamma_1 u(t_i)) \\ \hat{z}_2(t_{i+1}) &= \hat{z}_2(t_i) + \Delta t(-w_\mathrm{n}^2\hat{z}_1(t_i) + (a\gamma_2 - w_\mathrm{n}^2\gamma_1)(y(t_i) - r(t_i)) - b\gamma_2 u(t_i)). \end{aligned}$$
5. When the next sampling period arrives, repeat the computation from step 1.

5.4.3 Equivalence to a Resonant Controller

To find the Laplace transfer function of the resonant controller designed using disturbance estimation, we note that the Laplace transform of the control signal has the following expression:

$$U(s) = -K_1 Y(s) - \hat{D}(s) \tag{5.51}$$

where $\hat{D}(s) = \hat{X}_1(s)$ is the Laplace transform of the estimated sinusoidal disturbance. It can be verified that the Laplace transform of (5.46) gives

$$\begin{bmatrix} \hat{X}_1(s) \\ \hat{X}_2(s) \end{bmatrix} = \begin{bmatrix} s + 2\xi w_\mathrm{n} & -1 \\ w_\mathrm{n}^2 & s \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} (sY(s) + aY(s) - bU(s)). \tag{5.52}$$

Calculating the matrix inversion and multiplication gives

$$\hat{X}_1(s) = \frac{\gamma_1 s + \gamma_2}{s^2 + 2\xi w_\mathrm{n} s + w_\mathrm{n}^2}((s + a)Y(s) - bU(s)). \tag{5.53}$$

To find the Laplace transform of the control signal, we will substitute (5.53) into (5.51) and move the term containing $U(s)$ from the left-hand side to the right-hand side of the equation, which gives the expression of the control signal

$$U(s) = -K_1 \frac{s^2 + 2\xi w_\mathrm{n} s + w_\mathrm{n}^2}{s^2 + \omega_0^2} Y(s) - \frac{(\gamma_1 s + \gamma_2)(s + a)}{s^2 + \omega_0^2} Y(s) \tag{5.54}$$

where $b\gamma_1 = 2\xi w_\mathrm{n}$ and $b\gamma_2 = w_\mathrm{n}^2 - \omega_0^2$.

From (5.54), we find the Laplace transfer function of the resonant controller as

$$C(s) = K_1 \frac{s^2 + 2\xi w_\mathrm{n} s + w_\mathrm{n}^2}{s^2 + \omega_0^2} + \frac{(\gamma_1 s + \gamma_2)(s + a)}{s^2 + \omega_0^2}. \tag{5.55}$$

This controller has a pair of complex poles at $s_{1,2} = \pm j\omega_0$.

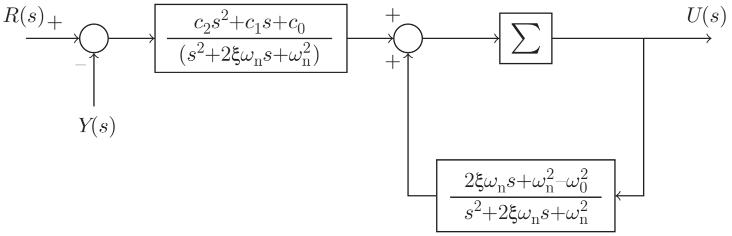

$R(s)$
+
−
$Y(s)$
$\frac{c_2s^2+c_1s+c_0}{(s^2+2\xi\omega_n s+\omega_n^2)}$
+
+
$\sum$
$U(s)$
$\frac{2\xi\omega_n s+\omega_n^2-\omega_0^2}{s^2+2\xi\omega_n s+\omega_n^2}$

Figure 5.9 Transfer function realization of a resonant controller with saturation limits.

To verify if the closed-loop poles are indeed at the locations of $-\alpha_1$ and $-\xi w_n \pm jw_n\sqrt{1-\xi^2}$ ($\xi = 1$ or 0.707), we calculate the closed-loop characteristic polynomial as

$$\begin{aligned}&(s+a)(s^2+\omega_0^2)+b(K_1(s^2+2\xi w_n s+w_n^2)+(\gamma_1 s+\gamma_2)(s+a))\\ &=(s^2+2\xi w_n s+w_n^2)(s+\alpha_1) \qquad (5.56)\end{aligned}$$

where $K_1 = \frac{\alpha_1-a}{b}$, $\gamma_1 = \frac{2\xi w_n}{b}$, and $\gamma_2 = \frac{w_n^2-\omega_0^2}{b}$. The closed-loop characteristic polynomial (5.56) leads to the conclusion that the closed-loop poles are at the locations as we specified.

Figure 5.9 shows the transfer function realization of the resonant controller with anti-windup mechanism.

5.4.4 MATLAB Tutorial on Disturbance observer-Based Resonant Controller Implementation

This section presents the MATLAB tutorial for implementation of the disturbance observer-based resonant controller together with the anti-windup mechanism when the control signal reaches its maximum or minimum values. The embedded resonant controller via estimation will be used in the simulation studies within the Simulink environment.

Tutorial 5.3 *The core of this activity is to produce a MATLAB embedded function that can be used in a Simulink simulation as well as in MATLAB real time implementation. The entire embedded MATLAB completes one cycle of computation for the control signal. For every sampling period, it will repeat the same computation procedure.*

Step by Step

1. *Create a new Simulink file called ResEstim.slx*
2. *In Simulink's directory of User-Defined Functions, find the icon of embedded MATLAB function and copy it to the ResEstim model.*
3. *Click on the icon of the embedded function, and define the input and output variables to the ResEstim model so that the embedded function has the following form:*

```
function uCur =ResEstim(r,y,K1,gamma1,gamma2,xi,wn,..
a,omega0,deltat,umin,umax)
```

where uCur is the calculated control signal at the sampling time t_i, the first two elements (r and y) among the input variables are the measurements of the the *reference and output signals at sampling time t_i, $K1$ is the proportional controller gain, gamma1, gamma2, xi, wn, a, omega0 are the parameters for the estimator, deltat is the sampling interval, umin and umax are the lower and upper limits imposed on the control signal uCur.*

4. *At the top of the embedded function, find Model Explorer among the Tools. When opening the Model Explorer, select discrete for the update method and input deltat into the sample time; select Support variable-size arrays; select Saturate on integer overflow; select Fixed point. Click Apply to save the changes.*
5. *We need to edit the input and output data ports in order to let the embedded function know which input ports are the real-time variables and which are the parameters. This editing task is performed using Model Explorer.*
   - *Click on r, on Scope, select input, assign port 1 and size -1 complexity, Inherited type Inherit: Same as Simulink. Repeat the same editing procedure for the output signal y.*
   - *The rest of 10 inputs to the embedded function are the parameters required in the computation. Click on $K1$, on Scope, select Parameter and click Tunable and click Apply to save the changes. Repeat the same editing procedure for the rest of the parameters.*
   - *To edit the output port from the embedded function, click on uCur, on Scope, select Output, Port 1, Size -1, Sampling Model Sample based, type Inherit: Same as Simulink, and click on Apply to save the changes.*

   In the following, the program will declare the variables that are stored in the embedded function during each iteration for their dimensions and initial values. zhat1 and zhat2 are the two estimated variables at sampling time t_i, which need initial values and are assigned to zero in this program. Enter the following program into the file:

```
persistent zhat1
if isempty(zhat1)
    zhat1=0;
end
persistent zhat2
if isempty(zhat2)
    zhat2=0;
end
```

6. *We calculate the following two constants using the parameters of the estimator. One could calculate the parameters outside the embedded function to save memory. Enter the following program into the file:*

```
cz1=a*gamma1-2*xi*wn*gamma1+gamma2;
cz2=2*xi*wn;
```

7. *Update the estimated disturbance $\hat{d}(t_i)$. Enter the following program into the file:*

```
dhat=zhat1+gamma1*(y-r);
```

8. *Update the control signal by combining the proportional control with the disturbance estimation. Enter the following program into the file:*

```
uCur= -K1*(y-r)-dhat;
```

9. *Implement saturation with anti-windup mechanism. Enter the following program into the file:*

```
if (uCur>umax) uCur=umax;end
if (uCur<umin) uCur=umin;end
```

10. *Calculate the* $\hat{z}_1(t_{i+1})$ *and* $\hat{z}_2(t_{i+1})$ *for the next sample. Enter the following program into the file:*

```
zhat1=zhat1+deltat*((-cz2*zhat1+zhat2)+cz1*(y-r)-cz2*uCur);
zhat2=zhat2+deltat*(-wn^2*zhat1..
+(a*gamma2-wn^2*gamma1)*(y-r)..
-(wn^2-omega0^2)*uCur);
```

This program is tested using the example presented in the next section.

5.4.5 Examples for Disturbance observer-Based Resonant Controllers

The following example is to illustrate the disturbance observer-based resonant controller design.

Example 5.5 *An electrical system is approximated by the following first order plus time delay model:*

$$G(s) = \frac{0.3e^{-0.0015s}}{0.001s + 1} \tag{5.57}$$

where the delay is used to describe the neglected time constants from the other electronic components in the system. The control objective is for the output of the system to follow a sinusoidal reference signal with frequency $\omega_0 = 2\pi \times 50$ rads^{-1}*. Design a resonant controller and simulate the closed-loop output with sampling interval* $\Delta t = 0.00001$ *(s).*

Solution. *The resonant controller is designed using the first order model, which gives the parameters* $a = 1/0.001 = 1000$ *and* $b = 0.3/1000 = 3000$*. Because the system has a time delay that is not used in the resonant controller design, it will affect the closed-loop stability and performance. Therefore, the proportional controller gain and the estimator gains need to be selected with regards to the actual first order plus delay system. A good starting point is to select the desired closed-loop pole* $-\alpha_1$ *for the controller gain* K_1 *equal to the model pole* $-a$*, leading to*

$$K_1 = \frac{\alpha_1 - a}{b} = 0.$$

Then, we select the parameters for the estimator to satisfy the closed-loop stability and performance in the presence of unmodeled time delay. By choosing $\xi = 0.707$*, the parameter* w_{n} *is used for adjusting the closed-loop response speed and robustness. The parameter selection can be performed using both closed-loop simulation and frequency response analysis.*

One can verify that using the Simulink embedded function created in Tutorial 5.3, the closed-loop system is unstable for the choice of $w_{\mathrm{n}} = 1000$*. Reducing* w_{n} *to* 500 *gives the estimator gains as*

$$\gamma_1 = \frac{2\xi w_{\mathrm{n}}}{b} = 2.3567; \quad \gamma_2 = \frac{w_{\mathrm{n}}^2 - \omega_0^2}{b} = 504.3465.$$

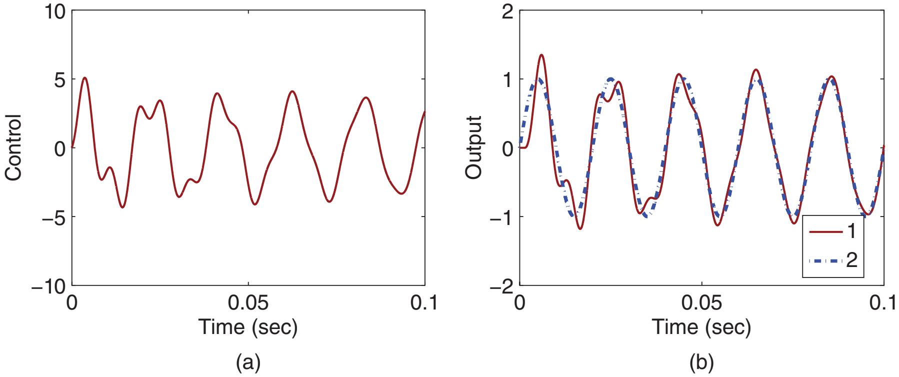


Figure 5.10 Closed-loop control response using a disturbance observer-based resonant controller (Example 5.5, $w_n = 500$). (a) Control signal. (b) Output. Key: line (1) output response; line (2) reference signal.

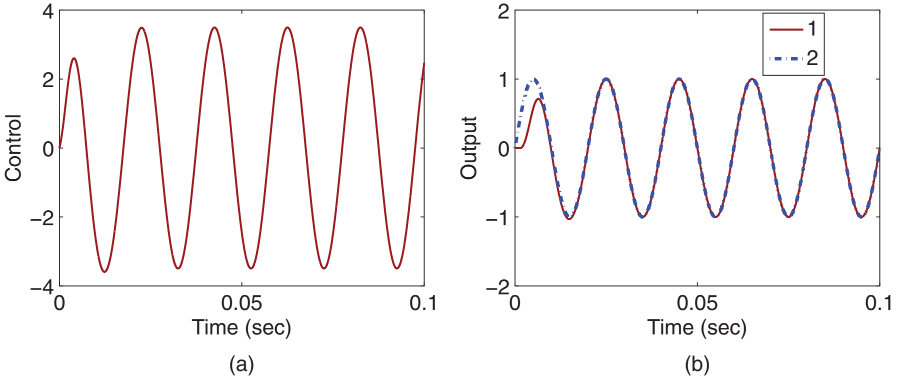


Figure 5.11 Closed-loop control response using disturbance observer-based resonant controller (Example 5.5, $\alpha_1 = 1000, w_n = 300$). (a) Control signal. (b) Output. Key: line (1) output response; line (2) reference signal.

Figures 5.10 (a) and (b) show the closed-loop control signal and the output signal to a sinusoidal reference signal with frequency $\omega_0 = 100\pi$. It is seen that the closed-loop system is stable. However the closed-loop responses are oscillatory in the initial period of responses evident from both the control signal and the output signal. The oscillation is caused by the neglected time delay in the plant.

Further reduction of w_n is necessary to eliminate the undesirable characteristics. With the choice of $w_n = 300$, the estimator gains become,

$$\gamma_1 = \frac{2\xi w_n}{b} = 1.4140; \quad \gamma_2 = \frac{w_n^2 - \omega_0^2}{b} = -28.9868.$$

Figures 5.11(a) and (b) show the closed-loop control signal and output signal to the sinusoidal reference signal. It is seen that the reduction of w_n has eliminated the undesired behavior in both control signal and output signal.

Another approach to the selection of parameters α_1 and w_n is to make $\alpha_1 = w_n$, which could simplify the resonant controller tuning process. It can be verified through simulation

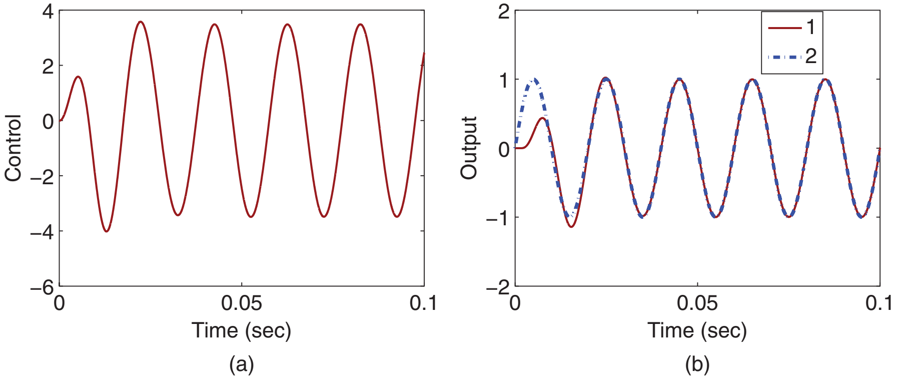


Figure 5.12 Closed-loop control response using a disturbance observer-based resonant controller (Example 5.5, $w_n = \alpha_1 = 400$). (a) Control signal. (b) Output. Key: line (1) output response; line (2) reference signal.

studies that when $\alpha_1 = w_n = 600$, the closed-loop system has oscillatory modes. When the parameters $\alpha_1 = w_n$ are reduced to 400, the controller and estimator gains are calculated as

$$K_1 = \frac{\alpha_1 - a}{b} = -2; \quad \gamma_1 = \frac{2\xi w_n}{b} = 1.8853; \quad \gamma_2 = \frac{w_n^2 - \omega_0^2}{b} = 204.3465.$$

Figures 5.12(a) and (b) show the responses of the control signal and the output signal to the sinusoidal reference. It is seen that with this selection of performance parameters, the closed-loop system has satisfactory responses.

One can easily analyze the effect of unmodeled dynamics using the complementary sensitivity function together with the time delay (see Section 2.7). The nominal complementary sensitivity function is calculated using the transfer function $G_0(s) = \frac{b}{s+a}$. If for all $\omega > 0$, the following condition is satisfied:

$$|\Gamma(j\omega)| = |\frac{G_0(j\omega)C(j\omega)}{1 + G_0(j\omega)C(j\omega)}||(e^{-j\omega d} - 1)| < 1$$

where $d = 0.0015$. Then, the closed-loop system is guaranteed to be stable. Figure 5.13 compares $|\Gamma(j\omega)|$ with respect to the two choices of the resonant controller parameters. It is seen that both closed-loop systems are robustly stable with $|\Gamma(j\omega)| < 1$ for all ω.

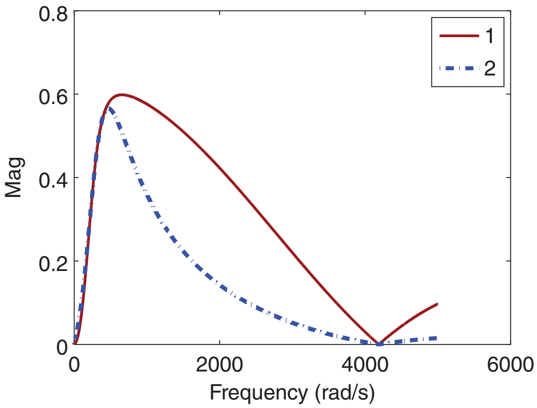


Figure 5.13 $|\Gamma(j\omega)|$ (Example 5.5). Key: line (1) $\alpha_1 = 1000$, $w_n = 300$; line (2) $w_n = \alpha_1 = 400$.

It is clear that these two approaches in selecting the parameters α_1 and w_n produced quite different controller parameters although the responses to the reference signal are similar. The question arises as how these two resonant controllers will behave in the presence of periodic disturbances. Clearly, if the disturbance has exactly the same frequency ω_0, then both resonant controllers will achieve the same disturbance rejection at the steady-state operation from the sensitivity analysis in Section 2.5. However, when the actual disturbance frequency is different from ω_0, there is a difference in the control system performance. The next example will compare the closed-loop performance of the two resonant controllers for the purpose of disturbance rejection.

Example 5.6 *Continue from Example 5.5. Assuming that there is a sinusoidal input disturbance with its frequency in the range between* $1.5\omega_0$ *and* $2\omega_0$*, we will evaluate the closed-loop performance of the two resonant controllers for the objective of disturbance rejection.*

Solution. *To measure how the resonant controllers will respond to disturbances in other frequency regions, we examine the sensitivity function*

$$S(j\omega) = \frac{1}{1 + G(j\omega)C(j\omega)}$$

where $G(j\omega)$ *is the frequency response of the system described by transfer function (5.57) and* $C(j\omega)$ *is the frequency response of the resonant controller described by the transfer function (5.55). Figure 5.14 compares the magnitude of the sensitivity functions for the two resonant control systems. Both sensitivity functions have a magnitude of 0 at* $\omega = 100\pi$ *from analytical solutions. For the choice of* $\alpha_1 = 1000$ *and* $w_n = 300$*, the sensitivity function has a lower gain at all frequency regions when comparing with the second case of* $w_n = \alpha_1 = 400$*. This implies that the first choice will work better for the purpose of disturbance rejections because smaller magnitude of sensitivity function leads to a better reduction of disturbance effect. Because the magnitude of the sensitivity function is*

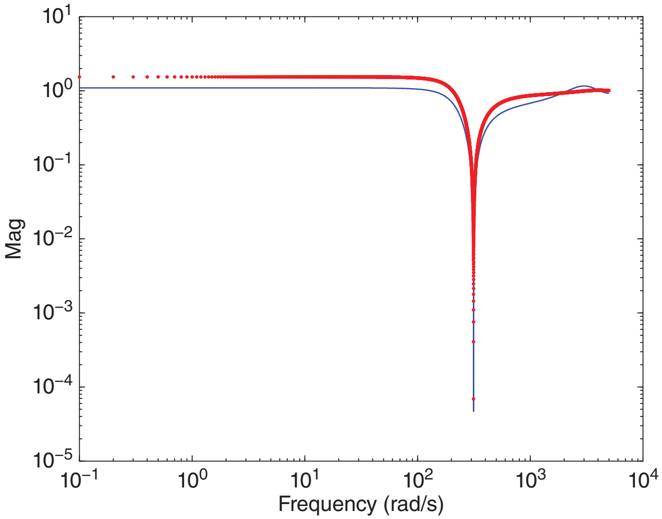


Figure 5.14 Magnitude of the sensitivity function (Example 5.6). Solid line: $\alpha_1 = 1000$, $w_n = 300$; dotted line: $w_n = \alpha_1 = 400$.

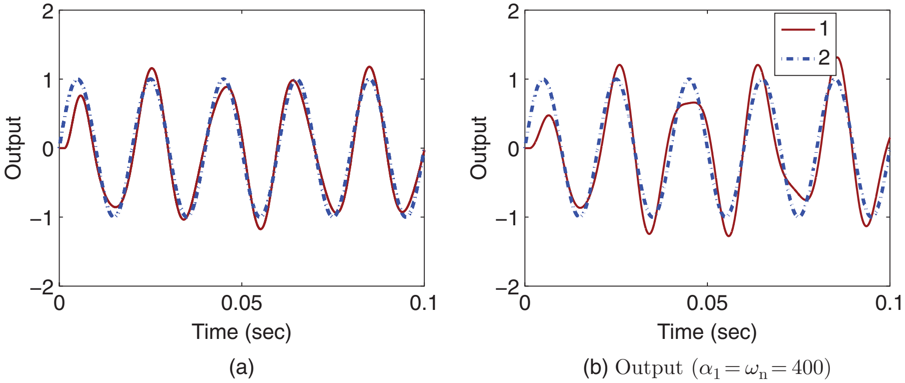


Figure 5.15 Closed-loop control response using disturbance observer-based resonant controller (Example 5.6). (a) Output ($\alpha_1 = 1000, w_n = 300$). (b) Output ($\alpha_1 = w_n = 400$). Key: line (1) $\alpha_1 = 1000, w_n = 300$, line (2) $\alpha_1 = w_n = 400$.

not zero at the disturbance frequency band $(1.5\omega_0, 2\omega_0)$*, it is not possible to completely eliminate the disturbance with the resonant controller. To evaluate the actual closed-loop performance, an input disturbance*

$$d_i(t) = 0.6sin(1.7\omega_0 t + 0.1)$$

is added to the Simulink simulation program used in Example 5.5 where $\omega_0 = 100\pi$ *rad s−1. The rest of the simulation parameters and conditions remains unchanged from the previous example. Figures 5.15(a) and (b) compare the closed-loop output responses with the two resonant controllers. It is seen that the resonant controller designed using* $\alpha_1 = 1000$ *and* $w_n = 300$ *produces much better reference tracking results than the one using* $\alpha_1 = w_n = 400$*. This improvement is the result of a better disturbance rejection property when the sensitivity function has a smaller magnitude at the disturbance frequency band.*

In summary, for better disturbance rejection, one should select α_1 *as large as possible, then adjust the bandwidth of the estimator* w_n *to achieve closed-loop stability and performance in the presence of unmodeled time delay. We emphasize that the reason why the periodic disturbance is not completely rejected at the steady-state is because the actual disturbance frequency is* $1.7\omega_0$*.*

The next example is to illustrate the effectiveness of the anti-windup mechanism contained in the resonant controller.

Example 5.7 *Continue from Examples 5.5 and 5.6. The magnitude of the control signal is about* 4.2 *in order for the closed-loop system to track the sinusoidal reference signal and reject the sinusoidal disturbance. Assume that the physical system only allows the control signal to vary between* ±3.6*, i.e.*

$$-3.6 \leq u(t) \leq 3.6.$$

Evaluate the resonant control system in the presence of control amplitude constraints.

Solution. *The MATLAB embedded function presented in Tutorial 5.3 is used with the maximum and minimum values of the control signals specified. The rest of the simulation conditions remains unchanged from Example 5.6.*

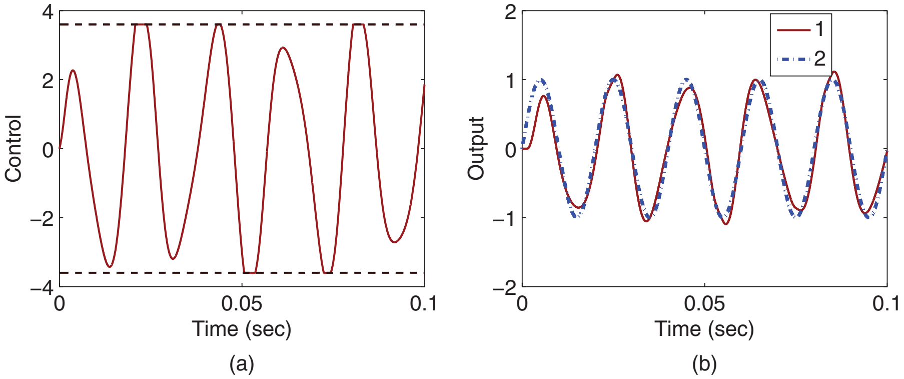


Figure 5.16 Closed-loop control response using disturbance observer-based resonant controller in the presence of control signal constraints (α_1 = 1000,w_n = 300). (a) Solid line- control signal, dashed line -control signal limits. (b) Output. Key: line (1) constrained control; line (2) unconstrained control.

Figures 5.16(a) and (b) show the closed-loop resonant control system responses to the sinusoidal reference signal and the sinusoidal disturbance signal. It is seen that the control signal is constrained within the limits specified and the anti-windup mechanism works very well. Figure 5.16(b) reveals that despite the constraints imposed on the control amplitude, there is little performance loss in the output response to the reference signal.

5.4.6 Food for Thought

1. Can you list three systems that have first order dynamics and require a resonant control system for disturbance rejection or reference following of a sinusoidal signal?
2. Is it correct that with the disturbance observer-based resonant controller, the complementary sensitivity function has a magnitude of one at the frequency $\omega = \omega_0$ where ω_0 is the frequency of the sinusoidal reference signal or the disturbance signal?
3. What do the closed-loop performance parameters α_1 and w_n represent in the design of disturbance observer-based resonant controller? If there are neglected dynamics in the system, how would you choose them?
4. What are the key components in the disturbance observer-based-resonant controller that lead to the anti-windup mechanism in its implementation?
5. Would you say that it is quite straightforward to implement the resonant controller with anti-windup mechanism?

5.5 Multi-frequency Resonant Controller

In many applications, the resonant controller containing a single frequency introduced in Section 5.4 may not be adequate to track the reference signal or reject the disturbance signal that contains multiple frequencies. The framework we used in the single frequency case can be extended to multiple frequencies.

5.5.1 Adding Integral Action to the Resonant Controller

We consider the following first order differential equation as in Section 5.4

$$\dot{y}(t) = -ay(t) + b(u(t) + d(t)) \tag{5.58}$$

where a and b are the coefficients; $u(t)$ and $y(t)$ are the input and output signals; $d(t)$ is the input disturbance signal. Different from the resonant controller design, we assume that $d(t)$ is a combination of a sinusoidal signal with a constant. For this purpose, $d(t)$ is expressed as

$$d(t) = d_1(t) + d_0$$

where $d_1(t) = d_\mathrm{m}\sin(\omega_0 t + \psi_0)$ and d_0 is an unknown constant. The feedback control law is determined as

$$u(t) = -K_1(y(t) - r(t)) - \hat{d}(t)$$

where $\hat{d}(t)$ is an estimate of the unknown disturbance $d(t)$.

With the desired closed-loop pole specified at $-\alpha_1$ and $\alpha_1 > 0$, the proportional feedback control gain K_1 is given as

$$K_1 = \frac{\alpha_1 - a}{b}.$$

Now, we will extend the estimation algorithm presented in Section 5.4 to include the estimation of an unknown constant. Let us choose $x_1(t) = d_1(t)$ and $x_2(t) = \dot{d}_1(t)$ and $x_3(t) = d_0$. Note that $\dot{x}_3(t) = 0$ as d_0 is a constant. The following differential equations will be used to describe the unknown disturbance $d(t)$ with the additional state $x_3(t)$:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega_0^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}. \tag{5.59}$$

In order to estimate the disturbance signal $d(t)$, from (5.58), we obtain

$$bd(t) = b\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \dot{y}(t) + ay(t) - bu(t)$$

which is the output equation for the estimation. Thus, the estimated variables $\hat{x}_1(t)$, $\hat{x}_2(t)$, $\hat{x}_3(t)$ are expressed as

$$\begin{bmatrix} \frac{\mathrm{d}\hat{x}_1(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\hat{x}_2(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\hat{x}_3(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega_0^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \left(\dot{y}(t) + ay(t) - bu(t) - b\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{bmatrix} \right) \tag{5.60}$$

where γ_1, γ_2 and γ_3 are the estimator gains chosen for the design.

As in Section 5.4, we will choose the parameters γ_1, γ_2 and γ_3 such that the poles of the error system are on the left half of the complex plane to ensure the convergence of the estimation errors. Here, the characteristic polynomial for the error system is computed as

$$\det M = \det\left(\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ -\omega_0^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \bar{\gamma}_1 \\ \bar{\gamma}_2 \\ \bar{\gamma}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}\right) \tag{5.61}$$

where $\bar{\gamma}_i = b\gamma_i$, $i = 1, 2, 3$ for simplicity in the computation.

There is an analytical expression for the determinant given by (5.61), which leads to analytical solutions for the parameters γ_1, γ_2, and γ_3. We firstly partition the M matrix into a block matrix as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

where

$$M_{11} = \begin{bmatrix} s + \bar{\gamma}_1 & -1 \\ \omega_0^2 + \bar{\gamma}_2 & s \end{bmatrix}; \quad M_{12} = \begin{bmatrix} \bar{\gamma}_1 \\ \bar{\gamma}_2 \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} \bar{\gamma}_3 & 0 \end{bmatrix}; \quad M_{22} = s + \bar{\gamma}_3.$$

With this partition, the determinant of the block matrix becomes

$$\det(M) = \det(M_{11}) \det(M_{22} - M_{21} M_{11}^{-1} M_{12}).$$

Note that M_{11} is exactly the same as the determinant described in the estimation of the sinusoidal signal in Section 5.4, which is

$$\det(M_{11}) = s^2 + \bar{\gamma}_1 s + \omega_0^2 + \bar{\gamma}_2$$

and the second determinant is

$$\det(M_{22} - M_{21} M_{11}^{-1} M_{12}) = \frac{s^3 + (\bar{\gamma}_1 + \bar{\gamma}_3)s^2 + (\omega_0^2 + \bar{\gamma}_2)s + \gamma_3 \omega_0^2}{s^2 + \bar{\gamma}_1 s + \omega_0^2 + \bar{\gamma}_2}.$$

The characteristic polynomial for the closed-loop error system becomes

$$\det(M) = s^3 + (\bar{\gamma}_1 + \bar{\gamma}_3)s^2 + (\omega_0^2 + \bar{\gamma}_2)s + \gamma_3 \omega_0^2. \tag{5.62}$$

With the choice of desired characteristic polynomial for the performance specification that consists of a pair of complex poles and a real pole having the following form: $(s + \alpha)(s^2 + 2\xi w_n s + w_n^2)$, we find the the coefficients $\bar{\gamma}_1$, $\bar{\gamma}_2$, and $\bar{\gamma}_3$ as

$$\bar{\gamma}_3 = \frac{\alpha w_n^2}{\omega_0^2}$$

$$\bar{\gamma}_1 = 2\xi w_n + \alpha - \bar{\gamma}_3;$$

$$\bar{\gamma}_2 = w_n^2 - \omega_0^2 + 2\xi w_n \alpha. \tag{5.63}$$

The actual gains used in the estimation are then scaled to yield: $\gamma_i = \frac{\bar{\gamma}_i}{b}$ for $i = 1, 2, 3$.

Defining $\hat{z}_1(t) = \hat{x}_1(t) - \gamma_1 y(t)$, $\hat{z}_2(t) = \hat{x}_2(t) - \gamma_2 y(t)$, and $\hat{z}_3(t) = \hat{x}_3(t) - \gamma_3 y(t)$, it can be verified that the implementation equation for the estimated disturbance becomes

$$\begin{bmatrix} \dfrac{\mathrm{d}\hat{z}_1(t)}{\mathrm{d}t} \\ \dfrac{\mathrm{d}\hat{z}_2(t)}{\mathrm{d}t} \\ \dfrac{\mathrm{d}\hat{z}_3(t)}{\mathrm{d}t} \end{bmatrix} = \Omega \begin{bmatrix} \hat{z}_1(t) \\ \hat{z}_2(t) \\ \hat{z}_3(t) \end{bmatrix} + \Omega \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} y(t) + a \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} y(t) - \begin{bmatrix} \bar{\gamma}_1 \\ \bar{\gamma}_2 \\ \bar{\gamma}_3 \end{bmatrix} u(t) \tag{5.64}$$

where Ω is the system matrix defined as

$$\Omega = \begin{bmatrix} -\bar{\gamma}_1 & 1 & -\bar{\gamma}_1 \\ -\omega_0^2 - \bar{\gamma}_2 & 0 & -\bar{\gamma}_2 \\ -\bar{\gamma}_3 & 0 & -\bar{\gamma}_3 \end{bmatrix}.$$

Ω has eigenvalues as the solutions of the characteristic equation:

$$(s + \alpha)(s^2 + 2\xi w_n s + w_n^2) = 0.$$

Hence, the implementation of the estimator using (5.64) is a stable realization.

From the estimated $\hat{z}_1(t)$ and $\hat{z}_3(t)$, we obtain

$$\hat{d}_1(t) = \hat{z}_1(t) + \gamma_1 y(t); \quad \hat{d}_0(t) = \hat{z}_3(t) + \gamma_3 y(t)$$

and $\hat{d}(t) = \hat{d}_1(t) + \hat{d}_0(t)$.

5.5.2 Adding More Periodic Components

If the system's disturbance or reference signal has more than one pair of periodic components, the disturbance estimator needs to be designed so to include those components.

We assume that the system is described by the differential equation (5.58). As before, $d(t)$ is the input disturbance signal and it is a combination of sinusoidal signals with a constant. For this purpose, $d(t)$ is expressed as

$$d(t) = d_1(t) + d_2(t) + d_0$$

where $d_1(t) = d_{m1}\sin(\omega_1 t + \psi_1)$, $d_2(t) = d_{m2}\sin(\omega_2 t + \psi_2)$ $(\omega_1 \neq \omega_2)$ and d_0 is an unknown constant.

Continuing from the design in Section 5.5.1, we choose $x_1(t) = d_1(t)$, $x_2(t) = \dot{d}_1(t)$, $x_3(t) = d_0$, $x_4(t) = d_2(t)$, and $x_5(t) = \dot{d}_2(t)$. With the two additional states, the following differential equation is used to describe the input disturbance $d(t)$:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega_1^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_2^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}. \tag{5.65}$$

To estimate the disturbance signal $d(t)$ from the dynamic system (5.58), the disturbance term is written as

$$bd(t) = b\begin{bmatrix}1 & 0 & 1 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\\ x_5(t)\end{bmatrix} = \dot{y}(t) + ay(t) - bu(t) = \eta(t).$$

The estimated variables $\hat{x}_1(t)$, $\hat{x}_2(t)$, $\hat{x}_3(t)$, $\hat{x}_4(t)$, and $\hat{x}_5(t)$ are written as

$$\begin{bmatrix}\frac{\mathrm{d}\hat{x}_1(t)}{\mathrm{d}t}\\ \frac{\mathrm{d}\hat{x}_2(t)}{\mathrm{d}t}\\ \frac{\mathrm{d}\hat{x}_3(t)}{\mathrm{d}t}\\ \frac{\mathrm{d}\hat{x}_4(t)}{\mathrm{d}t}\\ \frac{\mathrm{d}\hat{x}_5(t)}{\mathrm{d}t}\end{bmatrix} = \overbrace{\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ -\omega_1^2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & -\omega_2^2 & 0\end{bmatrix}}^{A}\begin{bmatrix}\hat{x}_1(t)\\ \hat{x}_2(t)\\ \hat{x}_3(t)\\ \hat{x}_4(t)\\ \hat{x}_5(t)\end{bmatrix} + \begin{bmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ \gamma_4\\ \gamma_5\end{bmatrix}\left(\eta(t) - b\overbrace{\begin{bmatrix}1 & 0 & 1 & 1 & 0\end{bmatrix}}^{C}\begin{bmatrix}\hat{x}_1(t)\\ \hat{x}_2(t)\\ \hat{x}_3(t)\\ \hat{x}_4(t)\\ \hat{x}_5(t)\end{bmatrix}\right) \tag{5.66}$$

where γ_1, γ_2, γ_3, γ_4, and γ_5 are the estimator gains chosen for the design.

To find the estimator gains, we will consider the pair of matrices, A and C. Because of their higher dimensions, it is no longer easy to work out the analytical solutions of the estimator gains. Instead, we can use MATLAB programs for finding the estimator's gain matrix. For this purpose, we define the A and C matrices as illustrated in (5.66), and we choose five desired closed-loop poles for the error system to ensure the convergence of the estimated variables. The MATLAB program place.m is used to compute the γ_i, $i = 1, 2, 3, 4, 5$, illustrated as below:

```
Gamma=place(A',C',P)'
```

where P contains the five desired closed-loop poles for the error system and gamma is the vector that contains γ_i, $i = 1, 2, 3, 4, 5$.

5.5.3 Food for Thought

1. Is it correct that in the design of disturbance observer-based resonant controller for the multi-frequency signal, the complexity of the estimator has increased in order to estimate the multi-frequency disturbance signal?
2. Would you say that in general a higher accuracy of the model is required for estimating more frequency components?
3. Can you devise a resonant controller scheme for the multi-frequency control so that you can sequentially enter the periodic component one at a time?

5.6 Summary

We have discussed the PID and resonant controller design from the angle of disturbance estimation. Using the disturbance observer-based approaches, the integral control or the resonant control is introduced through the estimation of an input disturbance with the assumption that the disturbance is a constant for the integral mode or the disturbance is a sinusoidal signal for the resonant mode. The advantages of the proposed approaches include the simplified design for the controller, and perhaps even more importantly a stable controller structure suitable for direct implementation with an anti-windup mechanism in the event of control signal saturation. The other important aspects of the chapter are summarized as follows.

- For a first order system with constant disturbance, the controller is a proportional control and the estimator is based on a first order model. This is equivalent to a PI controller with proportional and integral gains. The controller gain and estimator gains are selected with independent performance specifications.
- For a second order system with constant disturbance, the controller has the functions of proportional and derivative control, and the estimator is based on a first order model. This is equivalent to a PID controller with proportional, integral and derivative gains. The closed-loop performance for the controller and the estimator is specified separately through the locations of their closed-loop poles.
- For the resonant controller design, the system is assumed to have a first order model with input sinusoidal disturbance. The sinusoidal disturbance includes those with a single frequency or a multi-frequency. A proportional controller is used for the control function, and an estimator that embeds the sinusoidal modes is used to estimate the disturbance. This is equivalent to the resonant controller discussed in the previous chapters. Because the disturbance estimation is based on a stable system, the implementation of the proposed approach is numerically sound with a naturally embedded anti-windup mechanism if the control signal reaches its saturation limits.

For systems that have higher order dynamics, to use the design approaches proposed in this chapter, model order reduction is required.

5.7 Further Reading

1. A book is devoted to disturbance observer-based linear and nonlinear control systems (Li et al. (2014)). A survey is presented in Chen et al. (2016). The same research group has also worked on various topics of disturbance observer in control systems (Li et al. (2012), Yang et al. (2012)).
2. The early work on motion control using disturbance observer includes Komada et al. (1991). Disturbance observer is used for rigid mechanical systems in Schrijver and van Dijk (2002) and for robotic manipulator in Chen et al. (2000). Controller design for disturbance rejection is proposed using disturbance observer with estimation of frequency in Jia (2009). Stability and robust performance of motion control using disturbance observer is analyzed in Sariyildiz and Ohnishi (2015).
3. PID control and active disturbance rejection are introduced in Han (2009).

4. H_∞ control is compared with disturbance observer-based control methods in Mita et al. (1998).
5. Disturbance rejection in dual-stage feed drive control system is discussed in She et al. (2011).
6. A recent survey is presented for disturbance estimation and attenuation in PMSM drives (Yang et al. (2017)) using the framework of disturbance observer.
7. The disturbance observer-based resonant controller is used to control a single phase voltage converter with experimental validation in McNabb et al. (2017).

Problems

5.1 *Use the disturbance observer-based approach to design and implement a PI controller for the following systems:*

$$G(s) = \frac{1}{(s+1)(s+10)}$$

$$G(s) = \frac{2e^{-0.5s}}{(s+0.1)(s+10)^2}$$

$$G(s) = \frac{0.5e^{-2s}}{(s+0.01)(s+1)^3}$$

1. *Find the approximate first order model $G_A(s) = \frac{b}{s+a}$ by neglecting the relatively small time constant(s) and small time delay while maintaining the same steady-state gain.*
2. *Choose the desired closed-loop pole for the proportional controller K_1 as $-2a$ where a is the dominant pole for the system while the pole for the estimator is $-3a$ to obtain the estimator gain K_2.*
3. *Compute the frequency responses of the controller $C(j\omega)$, the sensitivity function $S(j\omega)$, the input disturbance sensitivity function $S_i(j\omega)$ and the complementary sensitivity function $T(j\omega)$. You can adjust the desired closed-loop poles for the controller and estimator, and observe their effects on the sensitivity functions.*
4. *Evaluate the closed-loop stability by using Nyquist diagram. If the closed-loop system is unstable, adjust the controller pole and the estimator pole to achieve stable closed-loop system.*
5. *Evaluate the robust stability condition for all $\omega > 0$:*

$$|T(j\omega)||\Delta G_m(j\omega)| < 1$$

6. *What are your observations from the Nyquist diagram and robust stability condition? What can we do to improve the robustness of the closed-loop system?*
7. *Build the MATLAB real-time function PIEstim.slx by following Tutorial 5.1 and simulate the closed-loop step response and input disturbance rejection. We choose sampling interval $\Delta t = 0.001$ and set the constraints on the control amplitude to be sufficiently large. A unit step reference signal is used in the simulation studies where a step input disturbance with amplitude of -1 enters the simulation at half of the simulation time.*

8. *Evaluate the effect of constraints on the control signal where the constraint parameters u^{max} and u^{min} are chosen to be 85 percent of the control signal's maximum amplitude from the previous step.*
9. *What are your observations from the constrained control simulations?*

5.2 *Use the disturbance observer-based approach to design a PID controller for the following systems:*

$$G(s) = \frac{2}{(s-1)(s+1)}$$
$$G(s) = \frac{3}{s^2}$$
$$G(s) = \frac{1}{s^2 + 0.1s + 3}$$

1. *Choose the desired closed-loop characteristic polynomial for the proportional plus derivative controller as $s^2 + 2\xi w_n s + w_n^2$ where $\xi = 0.707$ and $w_n = 3$, while the pole for the estimator is -4 to obtain the estimator gain K_3.*
2. *Build the MATLAB real-time function PIDEstim.slx by following Tutorial 5.2 and simulate the closed-loop step response and input disturbance rejection with the sampling interval $\Delta t = 0.001$ where the constraints on the control amplitude are set to be sufficiently large. In the simulations, the derivative filter time constant $\tau_f = 0.1\tau_D$. The reference signal is a unit step signal and the input disturbance signal has amplitude of -2 that enters the simulation at half of the simulation time.*
3. *Evaluate the effect of constraints on the control signal where the constraint parameters u^{max} and u^{min} are chosen to be 85 percent of the control signal's maximum amplitude from the previous step.*
4. *What are your observations from the constrained control simulations?*

5.3 *Consider the disturbance observer-based PID controller design for the three systems introduced in Problem 5.2. All the other conditions remain the same except designing the Proportional and Derivative controller with filter (see Section 3.4.1), where the filter time constant is considered in the process. The additional pole for the PD controller is chosen to be $-2w_n$.*

1. *What are your observations on the PD controller parameters in comparison with those obtained from Problem 5.2?*
2. *Repeat the simulation studies with constraints as described in Problem 5.2 and compare the simulation results with those obtained from Problem 5.2. What are your observations?*

5.4 *It is ideal to incorporate the derivative filter in the PID controller design and analysis. From Section 5.3, inclusion of the derivative filter τ_f will change the transfer function of the PID controller presented in (5.39). Find the PID controller transfer function*

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)}$$

that is equivalent to the disturbance observer-based PID controller.

5.5 *Equation (5.18) reveals that the disturbance observer-based PI control is equivalent to computing the disturbance transfer function using the input and output signal:*

$$\hat{D}(s) = \frac{\alpha_2}{s+\alpha_2}\frac{s+a}{b}Y(s) - \frac{\alpha_2}{s+\alpha_2}U(s) \tag{5.67}$$

which is, in essence, the inversion of the plant model together with a first order stable filter with unity steady-state gain for implementation, which is the essential component for disturbance observer-based approach (Li et al. (2014)).

1. *Propose a discretization scheme for the disturbance estimation based on (5.67).*
2. *Draw a closed-loop feedback diagram with limits on the control signal amplitude. Does this implementation incorporate the anti-windup mechanism?*

5.6 *Equation (5.34) shows that the disturbance observer-based PID controller is equivalent to accessing the disturbance information with the following transfer function form:*

$$\hat{D}(s) = \frac{\alpha_3}{s+\alpha_3}\frac{s^2+a_1s+a_0}{b}Y(s) - \frac{\alpha_3}{s+\alpha_3}U(s) \tag{5.68}$$

which is the inversion of the plant model with first order filter that has a unity steady-state gain.

1. *Propose a discretization scheme for computing the estimated disturbance based on the transfer function (5.68) with an additional first order filter so to make it realizable.*
2. *Draw the closed-loop diagram with control signal constraints.*

5.7 *From Problem 5.6, if the plant model has the following transfer function*

$$G(s) = \frac{b_1s+b_0}{s^2+a_1s+a_0}$$

where $b_1 \neq 0$, b_1 and b_0 have the same sign, meaning a stable zero.

1. *Express the $\hat{D}(s)$ in terms of the inversion of the new plant model.*
2. *Will this stable zero result in slow disturbance rejection if $|b_1| >> |b_0|$ or if $|b_1| << |b_0|$?*
3. *Verify your answer by examining the input disturbance sensitivity function $S_i(s)$ for both cases.*

5.8 *Consider the following systems with the given frequency ω_0 for a resonant controller design:*

$$G(s) = \frac{0.3e^{-0.01s}}{s+2}, \quad \omega_0 = 0.5;$$

$$G(s) = \frac{1}{(s+1)(s+10)}, \quad \omega_0 = 0.3;$$

$$G(s) = \frac{2}{(s+0.5)(s+10)^2}, \quad \omega_0 = 0.1$$

1. *Design the resonant controller based on a first order model $\frac{b}{s+a}$ by neglecting either the small time delay or small time constant (s), where the proportional controller pole is selected as* $-2a$ *and the desired closed-loop characteristic polynomial for the estimator as* $(s + w_n)^2$ *with* $w_n = 3a$.
2. *Calculate the frequency response of the complementary sensitivity function* $T(j\omega)$ *and the multiplicative modelling error* $\Delta G_m(j\omega)$ *and check if the robust stability condition, for all* $\omega > 0$:

$$|T(j\omega)||\Delta G_m(j\omega)| < 1$$

   is satisfied. If not, reduce the parameter w_n *until it is satisfied.*
3. *Build the MATLAB real-time function ResEstim.slx by following Tutorial 5.3 and simulate the closed-loop response to a sinusoidal reference signal* $r(t) = \sin(\omega_0 t)$, *where* $\Delta t = 0.001$.
4. *Add a sinusoidal input disturbance* $d_i = 3\cos(\omega_0 t)$ *at half of the simulation time and simulate the disturbance rejection of the closed-loop system.*
5. *In order to evaluate the anti-windup mechanism, you may choose the control signal limits* u^{max} *and* u^{min} *based on 85 percent of the maximum and minimum of the control signal in transient response.*

5.9 *In many applications, tracking of a ramp reference signal is required. The disturbance observer-based resonant controller introduced in Section 5.4 can be modified to track a ramp reference signal by assuming the frequency* $\omega_0 = 0$. *Assume that a given first order system is described by the following transfer function model:*

$$G(s) = \frac{0.3}{s+1}$$

1. *Modify the disturbance observer-based resonant controller to track a ramp input signal;*
2. *Find the proportional controller gain* K_1 *by positioning the desired closed-loop pole at* -2, *and find the estimator gains* γ_1 *and* γ_2 *by choosing the desired closed-loop characteristic polynomial as* $s^2 + 2\xi w_n s + w_n^2$, *where* $\xi = 1$ *and* $w_n = 2$.
3. *Use final value theorem to show that indeed this disturbance observer-based control system will track a ramp reference signal without steady-state error.*
4. *With the reference signal as* $0.5t$, *simulate the closed-loop response using the MATLAB real-time function ResEstim.slx where the sampling interval* $\Delta t = 0.01$.

5.10 *Consider the following first order system:*

$$G(s) = \frac{300}{s+100}$$

Design a resonant controller that will follow a reference signal $r(t) = 10 + \sin t$, *and reject an input disturbance* $d_i(t) = \cos 3t$, *where all desired closed-loop poles for both the controller and the estimator are positioned at* -200.

5.11 *One application for the resonant controller is in the area of a single phase AC current regulator. The objective is to accurately track a sinusoidal reference signal with a frequency that matches the power grid.*

The dynamic model for a single phase Voltage Source Inverter (VSI) coupled to a back electromotive force with an inductive-resistive filter network is expressed as

$$\frac{di(t)}{dt} = -\frac{R}{L}i(t) + \frac{1}{L}(v(t) - \epsilon_g(t)) \tag{5.69}$$

where $i(t)$ is the output current from the inverter, $v(t)$ is the input variable that is the pulse width modulated (PWM) switched voltage from the VSI, and $\epsilon_g(t)$ is the back EMF voltage that is naturally a sinusoid with a nominal frequency ω_0. The parameter R is the resistance and L is the inductance, associated with the inductive-resistive filter network.

The control signal $v(t)$ equals to the multiplication of a modulation signal $m(t)$ with the inverter DC link voltage V_{dc}:

$$v(t) = m(t)V_{dc} \tag{5.70}$$

In the mathematical model of a single phase Voltage Source Inverter as in McNabb et al. (2017), the physical parameters are $V_{dc} = 200V$, $L = 12 \times 10^{-3}H$, $R = 1\Omega$. The back EMF voltage is described by $\epsilon_g = 130V_{RMS}$ and $50Hz$ ($\omega_0 = 100\pi$).

1. *Design a resonant controller for the single phase Voltage source inverter, where $\omega_0 = 100\pi$. The closed-loop performance specifications are considered for the following two cases.*
   (a) *The feedback controller gain K_1 is selected such that the closed-loop control system pole is equal to the open-loop system pole. The parameters for the estimator are $w_n = 200\pi$ and $\xi = 0.707$.*
   (b) *The feedback controller gain K_1 is selected such that the closed-loop control system pole is twice of the open-loop system pole in magnitude. The parameters for the estimator are $w_n = 100\pi$ and $\xi = 0.707$.*
2. *Compare the sensitivity function $|S(j\omega)|$ and the complementary sensitivity function $|T(j\omega)|$ for the two cases. Determine the maximum time delay that the resonant control systems can tolerate. What are our observations? (Hint: the multiplicative modelling error is $\Delta G_m(j\omega) = 1 - e^{-jd\omega}$ for neglected time delay d).*
3. *Choosing sampling interval $\Delta t = 0.0001$ (sec), simulate the closed-loop response to reference following and disturbance rejection, in which the reference signal is $r(t) = 3\cos(\omega_0 t)$ and the disturbance is $\epsilon_g = 130\cos(\omega_0 t)$. In the simulation model, the control signal is $v(t)$ for simplification.*
4. *Impose the constraints on the amplitude of $v(t)$, where the maximum and minimum of the control signal are chosen to be 85 percent of the unconstrained control cases in transient responses.*
5. *Discuss the closed-loop control performances with respect to the two desired closed-loop performance specifications.*

6

PID Control of Nonlinear Systems

6.1 Introduction

One of the approaches to obtain the models for the control system design is based on analysis of the system dynamics using first principles, such as mass balance, Newton's laws, current law and voltage law. The majority of these types of models are nonlinear in nature. Thus, in order to use them for the PID controller design or other linear time invariant controller design, these nonlinear models need to be linearized around the operating conditions of the system.

This chapter will introduce linearization of nonlinear models through several examples and case studies. It will also show how to apply the PID controllers to a nonlinear plant using the technique called gain scheduled control.

6.2 Linearization of the Nonlinear Model

The starting point of designing a PID controller for a nonlinear plant is to obtain a linear time invariant model through linearization around the chosen operating conditions of the system.

6.2.1 Approximation of a Nonlinear Function

Assume that the nonlinear models have the general form:

$$\dot{x}(t) = f[x(t), u(t), t] \tag{6.1}$$

where$f[\cdot]$ is a nonlinear function. The purpose of linearization is to find a linear function (a set of linear functions) to describe the dynamics of the nonlinear model at a given operating condition.

In order to understand how the linearization is performed, we first examine the case of linearization of a nonlinear function. It begins with a Taylor series expansion and approximation of a nonlinear function. As we know, a function with variable $x, f(x)$ can be expressed in terms of a Taylor series expansion at $x = x^0$, where x^0 is a constant, as

$$f(x) = f(x^0) + \frac{\mathrm{d}f(x)}{\mathrm{d}x}|_{x=x^0}(x - x^0) + \frac{1}{2!}\frac{\mathrm{d}^2f(x)}{\mathrm{d}x^2}|_{x=x^0}(x - x^0)^2 + \dots \tag{6.2}$$

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

if the function $f(x)$ is smooth and its derivatives exist for all the orders. All the derivatives are evaluated at $x = x^0$.

Using the first two terms in the Taylor series expansion leads to the approximation of the original function $f(x)$ at a specific point x^0,

$$f(x) \approx f(x^0) + \frac{df(x)}{dx}|_{x=x^0}(x - x^0). \tag{6.3}$$

This first order Taylor series approximates the original nonlinear function $f(x)$ using the function evaluated at x^0 and its first derivative at $x = x^0$. The approximation holds in the vicinity of $x = x^0$. Figure 6.1 illustrates an example of a linear approximation of a nonlinear function where $x^0 = 5.3, f(x^0) = 140$ and $\frac{df(x)}{dx}|_{x=x^0} = 85$. It is seen that within the region where x is close to $x^0, f(x)$ is closely approximated by the first order Taylor series expansion (6.3). Intuitively, we can think of the original variable x as a "large" variable because it covers a large region, and the perturbed variable $x - x^0$ as a "small" variable because it covers a small region around x^0.

If the nonlinear function $f(x)$ contains n variables, meaning that $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$ is a vector with dimension n, then the function is approximated using the first $n + 1$ terms in the multi-variable Taylor series expansion as

$$f(x_1, x_2, x_3, \dots, x_n) \approx f(x_1^0, x_2^0, x_3^0, \dots, x_n^0) + \left.\frac{\partial f(x)}{\partial x_1}\right|_{x_1=x_1^0, x_2=x_2^0, \dots} (x_1 - x_1^0)$$
$$+ \left.\frac{\partial f(x)}{\partial x_2}\right|_{x_1=x_1^0, x_2=x_2^0, \dots} (x_2 - x_2^0) + \dots + \left.\frac{\partial f(x)}{\partial x_n}\right|_{x_1=x_1^0, x_2=x_2^0, \dots} (x_n - x_n^0). \tag{6.4}$$

Note that we need the partial derivatives of the nonlinear function against all its variables. Similar to the single variable case, the multi-variable Taylor series expansion consists of the constant term that is the nonlinear function evaluated at $x_1^0, x_2^0, , \dots, x_n^0$, followed by the partial derivatives with perturbations $x_1 - x_1^0, x_2 - x_2^0, , \dots, x_n - x_n^0$. Furthermore, the first order Taylor series expansion approximates well the original nonlinear function if the variables $x_1, x_2, , \dots, x_n$ are in the vicinities of $x_1^0, x_2^0, \dots, x_n^0$.

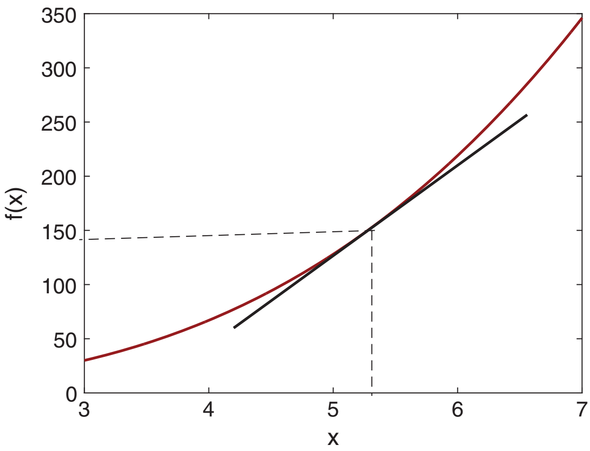


Figure 6.1 Approximation of a nonlinear function at $x^0 = 5.3$.

6.2.2 Linearization of nonlinear differential equations

The nonlinear models obtained from using first principles of the physical laws are differential equations. We assume that the nonlinear differential equation used to describe a physical system takes the general form:

$$\dot{x}(t) = f(x(t), u(t), t) \tag{6.5}$$

where $x(t)$ is a vector that represents the state variables of dimension n and $u(t)$ is a vector for the control signals of dimension m.

In linearization of a nonlinear dynamic system, we will firstly choose the constant vectors $x^0 = \left[x_1^0 \ x_2^0 \ \dots \ x_{\mathrm{n}}^0\right]^T$ and $u^0 = \left[u_1^0 \ u_2^0 \ \dots \ u_m^0\right]^T$, and apply the linearization procedure of the nonlinear functions as outlined in the previous section. We may write the final results in matrix and vector forms.

The constant vectors x^0 and u^0 play an important role in the linearized model. To make the linearized system truly linear, these vectors need to be selected carefully. The point of interest is called an equilibrium point for the nonlinear dynamic system (6.5), which was originated from nonlinear control (see Bay (1999)). These equilibrium points in control system design and implementation are often referred to as stationary points, which represent a steady-state solution to the dynamic equation (6.5). In general terms, an equilibrium point is defined by a constant vector x^{e} such that if $x(t_0) = x^{\mathrm{e}}$, $u(t) = 0$, then $x(t) = x^{\mathrm{e}}$.

Since an equilibrium point is a constant vector, the nonlinear differential equation (6.5) satisfies the following relationship:

$$\dot{x}(t) = f(x^{\mathrm{e}}, 0, t) = 0. \tag{6.6}$$

The concept of equilibrium points is extended to constant vectors x^0 and u^0 such that the following steady-state solution of the nonlinear differential equation (6.5) is true:

$$\dot{x}(t) = f(x^0, u^0) = 0. \tag{6.7}$$

These constant vectors are not dependent on time; however, they are allowed to vary as desired trajectories. In fact, they correspond to the steady-state values of the system that have been discussed previously in the PID control system implementation (see Chapter 4).

In control applications, it is often to choose the desired values for the state variables x^0, and solve the nonlinear algebraic equation given by (6.7) to determine the constant vector u^0. However, because of the uncertainty associated with the physical parameters and unknown disturbances, the characterization of the operating conditions of the system by the pair x^0 and u^0 could be quite inaccurate and far from reality.

The shortcoming caused by the inaccuracy of steady-state parameters can be overcome by the action of an integrator contained in the controller as the discrepancy is modeled as a constant input disturbance to the system. The integral action will automatically adjust the steady-state value of the input signal to achieve zero steady-state error in the output signal.

6.2.3 Case Study: Linearization of the Coupled Tank Model

Two cubic water tanks are connected in series as illustrated in Figure 6.2. Water flows into the first tank and flows out from the second tank. A pump controls the water in-flow

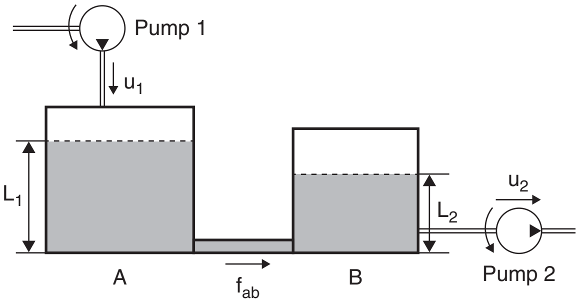


Figure 6.2 Schematic of a double tank.

rate $u_1(t)$ (m^3 s^{-1}) to the first tank; and another pump controls the water out-flow rate $u_2(t)$ (m^3 s^{-1}) from the second tank. Water flows from tank A to tank B, with a flow rate $f_{ab}(t)$. The unit for the flow rate is m^3 s^{-1} and the unit for the water level is m.

Using mass balance, the rate of change of water volume $V_1(t)$ in tank A is

$$\frac{\mathrm{d}V_1(t)}{\mathrm{d}t} = u_1(t) - f_{ab}(t). \tag{6.8}$$

The water volume can also be expressed as $V_1(t) = S_1L_1(t)$, where S_1 is the cross-sectional area of the tank A, and $L_1(t)$ is the water level in tank A. The dynamic equation to describe the rate of change in the water level $L_1(t)$ (tank A) is

$$S_1\frac{\mathrm{d}L_1(t)}{\mathrm{d}t} = u_1(t) - f_{ab}(t). \tag{6.9}$$

Likewise, the rate of change in the water level $L_2(t)$ is

$$S_2\frac{\mathrm{d}L_2(t)}{\mathrm{d}t} = f_{ab}(t) - u_2(t) \tag{6.10}$$

where S_2 is the cross-sectional area for tank B. For a small orifice with a cross-sectional area a_s (m^2), $f_{ab}(t)$ is linked to the tank levels $L_1(t)$ and $L_2(t)$ by the following relationship:

$$f_{ab}(t) = a_s\sqrt{2g(L_1(t) - L_2(t))} \tag{6.11}$$

where g is acceleration due to gravity (= 9.81 m s^{-2}); f_{ab} is the flow rate (m^3 s^{-1}).

By substituting (6.11) into (6.9) and (6.10), we obtain

$$\frac{\mathrm{d}L_1(t)}{\mathrm{d}t} = -\frac{a_s}{S_1}\sqrt{2g(L_1(t) - L_2(t))} + \frac{1}{S_1}u_1(t) \tag{6.12}$$

$$\frac{\mathrm{d}L_2(t)}{\mathrm{d}t} = \frac{a_s}{S_2}\sqrt{2g(L_1(t) - L_2(t))} - \frac{1}{S_2}u_2(t). \tag{6.13}$$

Both of these models are nonlinear.

Example 6.1 *Find the linearized models for the water tanks and discuss possible operating conditions and how they affect the linearized models.*

Solution. *In the linearization, the independent variables are $L_1(t)$, $L_2(t)$, $u_1(t)$, and $u_2(t)$. We will linearize the two equations (6.12) and (6.13) separately in terms of those independent variables. We let L_1^0 and L_2^0 denote the operating points for the tanks. The coefficients $\gamma_1 = \frac{a_s\sqrt{2g}}{S_1}$ and $\gamma_2 = \frac{a_s\sqrt{2g}}{S_2}$ are used to simplify the notations in both (6.12) and (6.13).*

The first term in (6.12) is approximated by the first order Taylor series expansion as

$$\gamma_1\sqrt{L_1(t)-L_2(t)} \approx \gamma_1\sqrt{L_1^0-L_2^0}+\gamma_1\frac{\partial(\sqrt{L_1(t)-L_2(t)})}{\partial L_1}|_{L_1^0,L_2^0}(L_1(t)-L_1^0)$$
$$+\gamma_1\frac{\partial(\sqrt{L_1(t)-L_2(t)})}{\partial L_2}|_{L_1^0,L_2^0}(L_2(t)-L_2^0). \tag{6.14}$$

Note that

$$\frac{\partial(\sqrt{L_1(t)-L_2(t)})}{\partial L_1}|_{L_1^0,\,L_2^0}=\frac{1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}} \tag{6.15}$$
$$\frac{\partial(\sqrt{L_1(t)-L_2(t)})}{\partial L_2}|_{L_1^0,\,L_2^0}=-\frac{1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}. \tag{6.16}$$

Therefore, (6.14) is written as

$$\gamma_1\sqrt{L_1(t)-L_2(t)}=\gamma_1\sqrt{L_1^0-L_2^0}+\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_1(t)-L_1^0)$$
$$-\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_2(t)-L_2^0). \tag{6.17}$$

The second term in the differential equation (6.12) is already linear in relation to $u_1(t)$, therefore, we keep it unchanged. By substituting the Taylor series approximation (6.17) into the differential equation (6.12), we obtain the linearized model for water tank A (do not forget that there is a negative sign):

$$\frac{\mathrm{d}L_1(t)}{\mathrm{d}t}=-\gamma_1\sqrt{L_1^0-L_2^0}-\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_1(t)-L_1^0)$$
$$+\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_2(t)-L_2^0)+\frac{1}{S_1}u_1(t). \tag{6.18}$$

Firstly, we notice that in order for the linearization to be valid, the operating points $L_1^0 > L_2^0$. Secondly, the first term is a constant that is not zero because $L_1^0 \neq L_2^0$. We can choose the steady-state value of $u_1(t)$ according to this constant. For this purpose, we re-write (6.18) as

$$\frac{\mathrm{d}L_1(t)}{\mathrm{d}t}=-\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_1(t)-L_1^0)$$
$$+\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0-L_2^0}}(L_2(t)-L_2^0)+\frac{1}{S_1}\left(u_1(t)-S_1\gamma_1\sqrt{L_1^0-L_2^0}\right). \tag{6.19}$$

To find the small signal model for tank A, we define the deviation variables as

$$\tilde{L}_1(t)=L_1(t)-L_1^0;\quad \tilde{L}_2(t)=L_2(t)-L_2^0;\quad \tilde{u}_1(t)=u_1(t)-S_1\gamma_1\sqrt{L_1^0-L_2^0}.$$

This leads to the linearized model for tank A as

$$\frac{\mathrm{d}\tilde{L}_1(t)}{\mathrm{d}t} = -\frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0 - L_2^0}}\tilde{L}_1(t) + \frac{\gamma_1}{2}\frac{1}{\sqrt{L_1^0 - L_2^0}}\tilde{L}_2(t) + \frac{1}{S_1}\tilde{u}_1(t). \tag{6.20}$$

Note that the steady-state value of the control signal $S_1\gamma_1\sqrt{L_1^0 - L_2^0}$ is a function of the system parameters S_1, γ_1. If there are errors in these parameters, then there is an error in the steady-state value of the control signal. This error could be modelled as an input disturbance that can be estimated using the estimator based PID controller (see Chapter 5).

The linearization of the nonlinear model for the tank B follows the same steps, and we leave the details as an exercise.

6.2.4 Case Study: Linearization of the Induction Motor Model

From the literature, several standard models of induction motor are available to use for control system design (Quang and Dittrich (2008), Wang et al. (2015)). Among them is a mathematical model with four differential equations in direct-quadrature (dq) coordination, which is originated from the field oriented control theory (Quang and Dittrich (2008), Wang et al. (2015)). When the parasitic effect such as eddy currents, magnetic field saturation are neglected, the dynamic model of an induction motor is governed by the following differential equations (Wang et al. (2015)):

$$i_{\mathrm{sd}}(t) + \tau'_\sigma\frac{\mathrm{d}i_{\mathrm{sd}}(t)}{\mathrm{d}t} = \tau'_\sigma\omega_{\mathrm{s}}(t)i_{\mathrm{sq}}(t) + \frac{k_{\mathrm{r}}}{r_\sigma\tau_{\mathrm{r}}}\psi_{\mathrm{rd}}(t) + \frac{1}{r_\sigma}u_{\mathrm{sd}}(t) \tag{6.21}$$

$$i_{\mathrm{sq}}(t) + \tau'_\sigma\frac{\mathrm{d}i_{\mathrm{sq}}(t)}{\mathrm{d}t} = -\tau'_\sigma\omega_{\mathrm{s}}(t)i_{\mathrm{sd}}(t) - \frac{k_{\mathrm{r}}}{r_\sigma}\omega(t)\psi_{\mathrm{rd}}(t) + \frac{1}{r_\sigma}u_{\mathrm{sq}}(t) \tag{6.22}$$

$$\psi_{\mathrm{rd}}(t) + \tau_{\mathrm{r}}\frac{\mathrm{d}\psi_{\mathrm{rd}}(t)}{\mathrm{d}t} = L_{\mathrm{h}}i_{\mathrm{sd}}(t) \tag{6.23}$$

$$f_{\mathrm{d}}\omega(t) + J_{\mathrm{E}}\frac{\mathrm{d}\omega(t)}{\mathrm{d}t} = \frac{3Z_{\mathrm{p}}L_{\mathrm{h}}}{2L_{\mathrm{r}}}i_{\mathrm{sq}}(t)\psi_{\mathrm{rd}}(t) - T_{\mathrm{L}}(t) \tag{6.24}$$

$$\omega_{\mathrm{s}}(t) = \omega(t) + \frac{L_{\mathrm{h}}i_{\mathrm{sq}}(t)}{\tau_{\mathrm{r}}\psi_{\mathrm{rd}}(t)} \tag{6.25}$$

where $i_{\mathrm{sd}}(t)$ and $i_{\mathrm{sq}}(t)$ are the stator currents in the dq coordination, $\psi_{\mathrm{rd}}(t)$ is the rotor flux in the d-axis and the input variables $u_{\mathrm{sd}}(t)$ and $u_{\mathrm{sq}}(t)$ represent the stator voltages in the dq coordination. $\omega_{\mathrm{s}}(t)$ and $\omega(t)$ are the synchronous and rotor velocity respectively. $T_{\mathrm{L}}(t)$ is the load torque that might change with respect to time. The rest of the parameters are physical parameters with descriptions in Wang et al. (2015). For example, R_{s} and L_{s} are the stator resistance and inductance, R_{r} and L_{r} are the rotor resistance and inductance, L_{h} is the mutual machine inductance;f_{d} is the friction coefficient, J_{E} is the inertia constant and Z_{p} is the number of pole pairs. The manipulated variables in the induction motor control problem are the stator voltages, $u_{\mathrm{sd}}(t)$ and $u_{\mathrm{sq}}(t)$, and the output variables are the rotor velocity $\omega(t)$ and the rotor flux in the d-axis, $\psi_{\mathrm{rd}}(t)$.

In the above model, there are four bilinear terms contained in (6.21)–(6.24). However, because the synchronous velocity $\omega_{\mathrm{s}}(t)$ is not a state variable, it needs to be replaced by the slip equation (6.25), which leads to the following nonlinear terms:

$$\omega_s(t)i_{sq}(t) = \omega(t)i_{sq}(t) + \frac{L_h}{\tau_r}\frac{i_{sq}^2(t)}{\psi_{rd}(t)} \tag{6.26}$$

$$\omega_s(t)i_{sd}(t) = \omega(t)i_{sd}(t) + \frac{L_h}{\tau_r}\frac{i_{sq}(t)i_{sd}(t)}{\psi_{rd}(t)}. \tag{6.27}$$

By pre-defining operating conditions and the steady-state parameters, ω_s^0, ω^0, i_{sq}^0, i_{sd}^0, ψ_{rd}^0, the nonlinear terms are approximated using a first order Taylor series expansion around the steady-state parameters. In particular, the following approximations are used in the derivation of the linearized model,

$$\omega(t)i_{sq}(t) \approx \omega^0 i_{sq}^0 + i_{sq}^0(\omega(t) - \omega^0) + \omega^0(i_{sq}(t) - i_{sq}^0) \tag{6.28}$$

$$\omega(t)i_{sd}(t) \approx \omega^0 i_{sd}^0 + i_{sd}^0(\omega(t) - \omega^0) + \omega^0(i_{sd}(t) - i_{sd}^0) \tag{6.29}$$

$$\omega(t)\psi_{rd}(t) \approx \omega^0 \psi_{rd}^0 + \psi_{rd}^0(\omega(t) - \omega^0) + \omega^0(\psi_{rd}(t) - \psi_{rd}^0) \tag{6.30}$$

$$i_{sq}(t)\psi_{rd}(t) \approx i_{sq}^0 \psi_{rd}^0 + \psi_{rd}^0(i_{sq}(t) - i_{sq}^0) + i_{sq}^0(\psi_{rd}(t) - \psi_{rd}^0) \tag{6.31}$$

$$\frac{i_{sq}^2(t)}{\psi_{rd}(t)} \approx i_{sq}^0 \psi_{rd}^0 + \frac{2i_{sq}^0}{\psi_{rd}^0}(i_{sq}(t) - i_{sq}^0) - \frac{(i_{sq}^0)^2}{(\psi_{rd}^0)^2}(\psi_{rd}(t) - \psi_{rd}^0) \tag{6.32}$$

$$\frac{i_{sq}(t)i_{sd}(t)}{\psi_{rd}(t)} \approx \frac{i_{sq}^0 i_{sd}^0}{\psi_{rd}^0} + \frac{i_{sq}^0}{\psi_{rd}^0}(i_{sd}(t) - i_{sd}^0) - \frac{i_{sq}^0 i_{sd}^0}{(\psi_{rd}^0)^2}(\psi_{rd}(t) - \psi_{rd}^0) + \frac{i_{sd}^0}{\psi_{rd}^0}(i_{sq}(t) - i_{sq}^0). \tag{6.33}$$

Although the variables $\omega(t)$, $i_{sq}(t)$, $i_{sd}(t)$, $\psi_{rd}(t)$ are the actual physical variables, not the deviation variables, the approximation relations are only valid in the vicinity of the steady-state conditions as they are based on a Taylor series expansion. From (6.28)–(6.33), it is seen that information about the steady-state values of ω^0, i_{sd}^0, i_{sq}^0, and ψ_{rd}^0 is required to obtain the parameters for the linearized terms. Since the output variables are $\omega(t)$ and ψ_{rd}, the steady-state parameters for these variables will be chosen to be equal to their desired reference signals. In particular, in the application of induction motor control, the reference signal to rotor flux is often fixed as a constant with its value dependent on the operating speed and load condition of the induction motor. For instance, the reference signal for ψ_{rd} is recommended to be 0.35 Wb for the energy efficient at the rated speed and load-free operating condition. The reference signal to the rotor velocity $\omega(t)$ changes according to operating conditions. Therefore, the steady state conditions for ψ_{rd}^0 and ω^0 are first determined according to the operating conditions of the induction motor. Next, from (6.23), by letting $\frac{d\psi_{rd}(t)}{dt} = 0$, the steady-state solution of i_{sd}^0 is determined via the steady-state calculation, $i_{sd}^0 = \frac{1}{L_h}\psi_{rd}^0$. Furthermore, by letting $\frac{d\omega(t)}{dt} = 0$, the steady-state operating condition for i_{sq} is calculated using (6.24) together with the linear approximation (6.31),

$$i_{sq}^0 = \frac{2L_r}{3Z_p L_h \psi_{rd}^0}(f_d\omega^0 + T_L^0). \tag{6.34}$$

With all the steady-state operating parameters defined, the next step is to substitute (6.28)–(6.33) into (6.21)–(6.25) in order to obtain the linear time-invariant (LTI) model that is valid at the operating condition specified by the steady-state parameters ω^0, i^0_{sq}, i^0_{sd}, ψ^0_{rd}. By gathering all the appropriate terms, it can be verified that the linear model has the form,

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = Ax(t) + Bu(t) + \mu^0 \tag{6.35}$$

where $x(t) = [i_{\text{sd}}(t) - i^0_{\text{sd}}\ i_{\text{sq}}(t) - i^0_{\text{sq}}\ \psi_{\text{rd}}(t) - \psi^0_{\text{rd}}\ \omega(t) - \omega^0]^T$, $u(t) = [u_{\text{sd}}(t) - u^0_{\text{sd}}\ u_{\text{sq}}(t) -u^0_{\text{sq}}]^T$, and with the coefficient $\kappa_t = \frac{3Z_{\text{p}}L_{\text{h}}}{2L_{\text{r}}J_{\text{E}}}$, the matrices A and B are defined as

$$A = \begin{bmatrix} -\dfrac{1}{\tau'_\sigma} & \omega^0 + \dfrac{2L_{\text{h}}}{\tau_{\text{r}}}\dfrac{i^0_{\text{sq}}}{\psi^0_{\text{rd}}} & \dfrac{k_{\text{r}}}{r_\sigma\tau_{\text{r}}\tau'_\sigma} - \dfrac{L_{\text{h}}}{\tau_{\text{r}}}\dfrac{(i^0_{\text{sq}})^2}{(\psi^0_{\text{rd}})^2} & i^0_{\text{sq}} \\ -\omega^0 - \dfrac{L_{\text{h}}}{\tau_{\text{r}}}\dfrac{i^0_{\text{sq}}}{\psi^0_{\text{rd}}} & -\dfrac{1}{\tau'_\sigma} - \dfrac{L_{\text{h}}}{\tau_{\text{r}}}\dfrac{i^0_{\text{sd}}}{\psi^0_{rd}} & -\dfrac{k_{\text{r}}}{r_\sigma\tau'_\sigma}\omega^0 + \dfrac{L_{\text{h}}}{\tau_{\text{r}}}\dfrac{i^0_{\text{sq}}i^0_{\text{sd}}}{(\psi^0_{\text{rd}})^2} & -\dfrac{k_{\text{r}}}{r_\sigma\tau'_\sigma}\psi^0_{\text{rd}} - i^0_{\text{sd}} \\ \dfrac{L_{\text{h}}}{\tau_{\text{r}}} & 0 & -\dfrac{1}{\tau_{\text{r}}} & 0 \\ 0 & \kappa_t\psi^0_{\text{rd}} & \kappa_t i^0_{\text{sq}} & -\dfrac{f_{\text{d}}}{J_{\text{E}}} \end{bmatrix};$$

$$B = \begin{bmatrix} \dfrac{1}{r_\sigma\tau'_\sigma} & 0 \\ 0 & \dfrac{1}{r_\sigma\tau'_\sigma} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The constant vector μ^0 represents uncertainties associated with the physical parameters and the variations of the steady-state parameters. It is seen that the uncertainties are captured as input constant disturbances to the system.

6.2.5 Food for Thought

1. Would you say that the first order Taylor series expansion underpins the linearization of nonlinear models?
2. Would you want to a use second order Taylor series expansion to improve the accuracy of approximation?
3. For the linear models derived from the double water tank system, what are the actual physical variables that are directly measured? What are the deviation variables we created?
4. Is it a good strategy to choose the steady-state values of a nonlinear system according to the reference signals to the closed-loop control system?
5. Would it be possible to find the steady-state values by solving the nonlinear dynamic equations, which can be achieved by building a Simulink program?
6. For a complex system, would you consider using experimental tests to find the actual operating conditions through steady-state experiments?

6.3 Case Study: Ball and Plate Balancing System

This section is based on a final year project performed by Mr John Lee (see Lee (2013)), who was previously a fourth year electrical engineering student at RMIT University, Australia. As part of the project, a prototype of a ball and plate balancing system was built and PID control systems were designed for this self-made system. More importantly, they were successfully implemented with experimental results. Details for this project can be found in Lee (2013).

6.3.1 Dynamics of the Ball and Plate Balancing System

A ball and plate balancing system is illustrated in Figure 6.3, which consists of a ball and a rigid plate together with a set of actuators, sensors, and a controller. The position of the ball on the top of the plate is controlled by manipulating the inclination of the plate about its x- and y-axes.

The system has four variables to be controlled. The first pair corresponds to the inclination of the plate in the x- and y-axes captured by the angles of the plate θ_x and θ_y from the x- and y-axes, and the second pair corresponds to the position of the ball in the x- and y-axes denoted by x and y. Two DC motor drivers are used to control the system. The relationship between the motor torque forces and the inclination of the plate is described by the following two differential equations:

$$\begin{aligned}(J_\mathrm{p} + J_\mathrm{b} + mx^2)\ddot{\theta}_x &+ 2mx\dot{x}\dot{\theta}_x + mxy\ddot{\theta}_y + m\dot{x}y\dot{\theta}_y + mx\dot{y}\dot{\theta}_y \\ &+ mgx\cos(\theta_x) = \tau_x \end{aligned} \tag{6.36}$$

$$\begin{aligned}(J_\mathrm{p} + J_\mathrm{b} + my^2)\ddot{\theta}_y &+ 2my\dot{y}\dot{\theta}_y + mxy\ddot{\theta}_x + m\dot{x}y\dot{\theta}_x + mx\dot{y}\dot{\theta}_x \\ &+ mgy\cos(\theta_y) = \tau_y \end{aligned} \tag{6.37}$$

where J_p and J_b are the mass moment of inertia of the plate and the ball respectively, m is the mass of the ball and g is the gravity constant ($g = 9.8$ m s^{-2}). The variables τ_x and τ_y are the torque forces in the x and y directions.

The movement of the ball on the plate is described the following two equations:

$$\left(m + \frac{J_\mathrm{b}}{R^2}\right)\ddot{x} - mx(\dot{\theta}_x)^2 - my\dot{\theta}_x\dot{\theta}_y = -mg\sin(\theta_x) \tag{6.38}$$

$$\left(m + \frac{J_\mathrm{b}}{R^2}\right)\ddot{y} - my(\dot{\theta}_y)^2 - mx\dot{\theta}_x\dot{\theta}_y = -mg\sin(\theta_y) \tag{6.39}$$

where R is the radius of the ball.

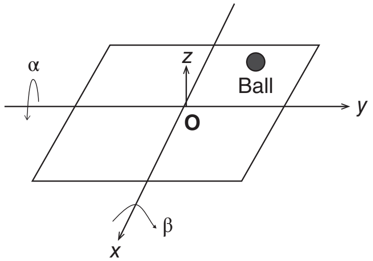


Figure 6.3 Schematic of the ball and plate balancing system.

In the control system implementation, a touch screen is used as the sensor to measure the ball's position on the plate. As a result, a heavy ball is used, which gives the inertial parameter J_b as

$$J_b = \frac{2}{5}mR^2.$$

Thus, the plate dynamic equations are re-written as

$$\frac{7m}{5}\ddot{x} - mx(\dot{\theta}_x)^2 - my\dot{\theta}_x\dot{\theta}_y = -mg\sin(\theta_x) \tag{6.40}$$

$$\frac{7m}{5}\ddot{y} - my(\dot{\theta}_y)^2 - mx\dot{\theta}_x\dot{\theta}_y = -mg\sin(\theta_y). \tag{6.41}$$

Clearly the equations (6.36) and (6.37) describe the actuator dynamics, which have severe nonlinearities, and the equations (6.40) and (6.41) describe the position outputs x and y in relation to the input variables θ_x and θ_y. From these two sets of equations, it is apparent that cascade control systems (see Chapter 7) should be deployed to control the actuators and the plant, respectively.

However, because in the control system implementation two DC motors are used as the actuators with their angular positions under control from the manufacturer, the dynamics from the actuators are neglected and, instead, steady-state relationships between the angular positions of the DC motors and the inclination of the plate (governed by θ_x and θ_y) are found. With this simplification, the control system design focuses on the dynamics of the plate as described by (6.40) and (6.41) where the inputs are θ_x and θ_y, and the outputs are the positions of the ball governed by x and y coordinates.

6.3.2 Linearization of the Nonlinear Model

The next step in the control system design is to derive the linear models for the nonlinear dynamic system. The operating conditions for the ball and plate balancing system are defined as follows.

1. At the equilibrium, the ball is stable at the center of the plate, which is $x^0 = 0$ and $y^0 = 0$.
2. The angle of the plate is zero in both x- and y-axes, which is translated to $\theta_x^0 = \theta_y^0 = 0$.
3. The angle of the plate is not changing, which leads to $\dot{\theta}_x^0 = \dot{\theta}_y^0 = 0$.

To obtain the linear model for (6.40), we consider the linearization procedure term by term.

The first term in (6.40) is linear by itself and does not require linearization. The nonlinear function in the second term is approximated by a Taylor series expansion as

$$\begin{aligned} x\dot{\theta}_x^2 &\approx x^0(\dot{\theta}_x^0)^2 + \frac{\partial(x\dot{\theta}_x^2)}{\partial x}|_{x=x^0,\dot{\theta}_x=\dot{\theta}_x^0}(x - x^0) + \frac{\partial(x\dot{\theta}_x^2)}{\partial\dot{\theta}_x}|_{x=x^0,\dot{\theta}_x=\dot{\theta}_x^0}(\dot{\theta}_x - \dot{\theta}_x^0) \\ &= x^0(\dot{\theta}_x^0)^2 + \dot{\theta}_x^2|_{x=x^0,\dot{\theta}_x=\dot{\theta}_x^0}(x - x^0) + 2x\dot{\theta}_x|_{x=x^0,\dot{\theta}_x=\dot{\theta}_x^0}(\dot{\theta}_x - \dot{\theta}_x^0) \\ &= 0 \end{aligned}$$

because $x^0 = \dot{\theta}_x^0 = 0$.

The quantity in the third term of (6.40) is approximated by a Taylor series expansion as

$$\begin{aligned} y\dot{\theta}_x\dot{\theta}_y \approx{} & y^0\dot{\theta}_x^0\dot{\theta}_y^0 + \frac{\partial(y\dot{\theta}_x\dot{\theta}_y)}{\partial y}|_{y=y^0,\dot{\theta}_x=\dot{\theta}_x^0,\dot{\theta}_y=\dot{\theta}_y^0}(y - y^0) \\ & + \frac{\partial(y\dot{\theta}_x\dot{\theta}_y)}{\partial\dot{\theta}_x}|_{y=y^0,\dot{\theta}_x=\dot{\theta}_x^0,\dot{\theta}_y=\dot{\theta}_y^0}(\dot{\theta}_x - \dot{\theta}_x^0) \\ & + \frac{\partial(y\dot{\theta}_x\dot{\theta}_y)}{\partial\dot{\theta}_y}|_{y=y^0,\dot{\theta}_x=\dot{\theta}_x^0,\dot{\theta}_y=\dot{\theta}_y^0}(\dot{\theta}_y - \dot{\theta}_y^0) = 0 \end{aligned}$$

because $y^0 = \dot{\theta}_x^0 = \dot{\theta}_y^0 = 0$.

The nonlinear quantity on the right-hand side of (6.40) is approximated as

$$\begin{aligned} \sin\theta_x &\approx \sin\theta_x^0 + \frac{\mathrm{d}\sin\theta_x}{\mathrm{d}\theta_x}|_{\theta_x=\theta_x^0}(\theta_x - \theta_x^0) \\ &= \sin\theta_x^0 + \cos\theta_x|_{\theta_x=\theta_x^0}(\theta_x - \theta_x^0) = \theta_x. \end{aligned}$$

Combining all the linearized quantities together yields the linear model that describes the dynamics of the ball and plate balancing system at its operating condition as,

$$\frac{7m}{5}\ddot{x} = -mg\theta_x \tag{6.42}$$

which says that at the equilibrium point, the ball and plate balancing system is a double integrator system. The input to the system is angle of the plate θ_x and the output is the position of the ball x on the plate.

The dynamic model for the y-axis is obtained through the linearization of the nonlinear model (6.41) as

$$\frac{7m}{5}\ddot{y} = -mg\theta_y. \tag{6.43}$$

It is seen that the linearized models are equal. Also the coupling relationships in the original nonlinear models are gone, meaning that two identical PID controllers can be used to control the x- and y-axes separately.

6.3.3 PID Controller Design

Because all the steady-state variables are zero in the ball and plate balancing system, the Laplace transfer function of (6.42) for the x-axis becomes

$$\frac{X(s)}{\Theta_x(s)} = -\frac{5}{7}g\frac{1}{s^2}. \tag{6.44}$$

Because the system is a double integrator system, one might attempt to use a PD controller for the position control because the self-contained integrator would take care of the tracking accuracy to step reference signals. Were the system a true double integrator, this option would be viable. Because the double integrator is the result of linearization of a nonlinear model, a small deviation from the operating condition, this characteristic would disappear, observed from the original physical model (6.40). Therefore, PID controller with derivative filter is selected to perform the position control.

Following from the PID controller design in Section 3.4, the controller structure is selected as

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s(s + l_0)} \tag{6.45}$$

and the desired closed-loop polynomial is selected as

$$\begin{aligned} A_{\text{cl}} &= (s^2 + 2 \times 0.707 w_{\text{n}} s + w_{\text{n}}^2)(s + w_{\text{n}})^2 \\ &= s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0 \end{aligned}$$

where the parameter w_{n} is a tuning parameter for the closed-loop performance.

With the second order model,

$$G(s) = \frac{b_0}{s^2}$$

where $b_0 = -\frac{5}{7}g$, the polynomial equation for solving the PID controller parameters becomes

$$\begin{aligned} &s^4 + l_0 s^3 + b_0 c_2 s^2 + b_0 c_1 s + b_0 c_0 \\ &\quad = s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0. \end{aligned} \tag{6.46}$$

The solution of (6.46) gives

$$l_0 = t_3; c_2 = \frac{t_2}{b_0}; c_1 = \frac{t_1}{b_0}; c_0 = \frac{t_0}{b_0}.$$

To determine the performance parameter w_{n}, using Simulink simulation, nonlinear system simulators were built with consideration of the nonlinear plant and actuator dynamics for various reference signals. It was found that $w_{\text{n}} = 3$ is a satisfactory choice from the simulation studies based on the nonlinear simulators, and this w_{n} was used in the actual implementation with a small adjustment for special cases.

It is noted that the linear models for the ball and plate balancing system are sufficiently accurate for the PID controller design with the parameter w_{n} calibrated against the nonlinear system. By adjusting w_{n}, we effectively adjust the desired closed-loop bandwidth against the actual physical system to reduce the effect of modeling errors.

However, when a resonant controller is designed using the linear model (6.44), the feedback controller could not stabilize the actual physical system despite numerous attempts. A possible key reason for this problem is that the resonant controller demands much higher accuracy from the plant model in the higher frequency region. The neglected dynamics from the actuators and the sensors will have a greater effect on the closed-loop stability and performance when a resonant controller is used.

As a result, in order to track a sinusoidal reference signal, the same PID controllers are used with feedforward compensation on the reference signals, in which the output errors at the steady-state operation are measured and then compensated at the reference signal by modifying its amplitude and phase. This compensation from reference signal no longer puts extra demand on the accuracy of the model.

6.3.4 Implementation and Experimental Results

In the implementation of the control system, the parameters in controller (6.45) are converted to K_{c}, τ_{I}, τ_{d} and τ_{f} for the discretization and a two degrees of freedom PID controller is implemented as shown in Section 3.4, which is to put the proportional control and the derivative control on the output only. The sampling interval used in the implementation is $\Delta t = 0.01$ (s).

Additionally, to protect the equipment, constraints on the derivative of the control signal and the amplitude of the control signal are imposed with anti-windup mechanisms, which was shown in Tutorial 4.1. In the implementation, the MATLAB real time function PIDV.slx in Tutorial 4.1 was converted into C-code for real time execution using a micro-controller.

It was a tremendous effort to design and implement the ball and plate balancing control system, which requires detailed considerations and executions in hardware electronics design and software design. The design and implementation were detailed in Lee's report (Lee (2013)). As a case study, selected experimental results are presented.

6.3.4.1 Disturbance Rejection

In the disturbance rejection experiment, the ball is positioned at the center of the plate, which is at the coordinates of $x = 0$ and $y = 0$. The reference signals to both x- and y-axes are zero and an external pulse disturbance was applied to the ball by using a finger to perturb the position of the ball. The x-axis and y-axis responses to this unknown disturbance are shown in Figures 6.4(a) and (b). Figure 6.4(c) shows the x–y plane plot for the

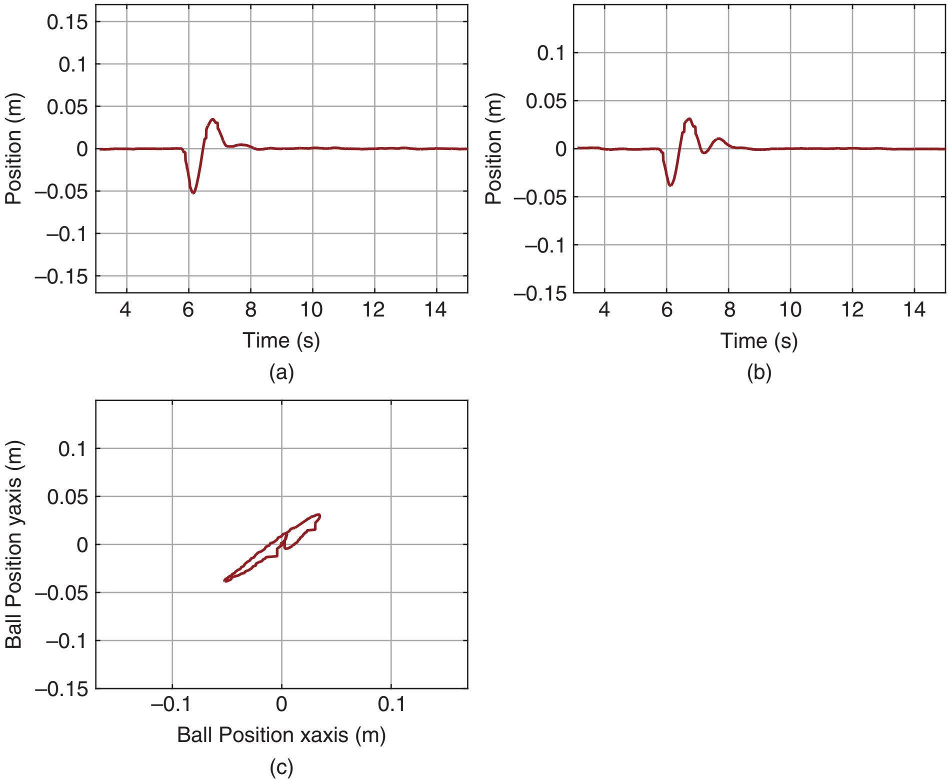


Figure 6.4 Disturbance rejection. (a) x-axis response. (b) y-axis response. (c) Ball position.

disturbance rejection. It is seen that the PID control system has successfully rejected the disturbance without steady-state errors.

6.3.4.2 Making a Square Movement

To make a square movement, two sets of step signals are chosen as the desired reference signals for the x-axis and y-axis, as shown by the dashed lines in Figures 6.5(a) and (b). The output responses to the x and y reference signals are compared in the same plots. Figure 6.5(c) shows the ball movement on the plate, which is seen as a square trajectory.

6.3.4.3 Making a Circle Movement

To make a circle movement, the reference signals to the x-axis and y-axis are chosen to be

$$x^*(t) = 0.075\cos\left(\frac{2\pi}{2.5}t\right)$$
$$y^*(t) = 0.075\sin\left(\frac{2\pi}{2.5}t\right).$$

Here, the desired angular velocity of the ball movement is $\frac{2\pi}{2.5}$ rad s−1 and the radius of the circle is 0.075 m.

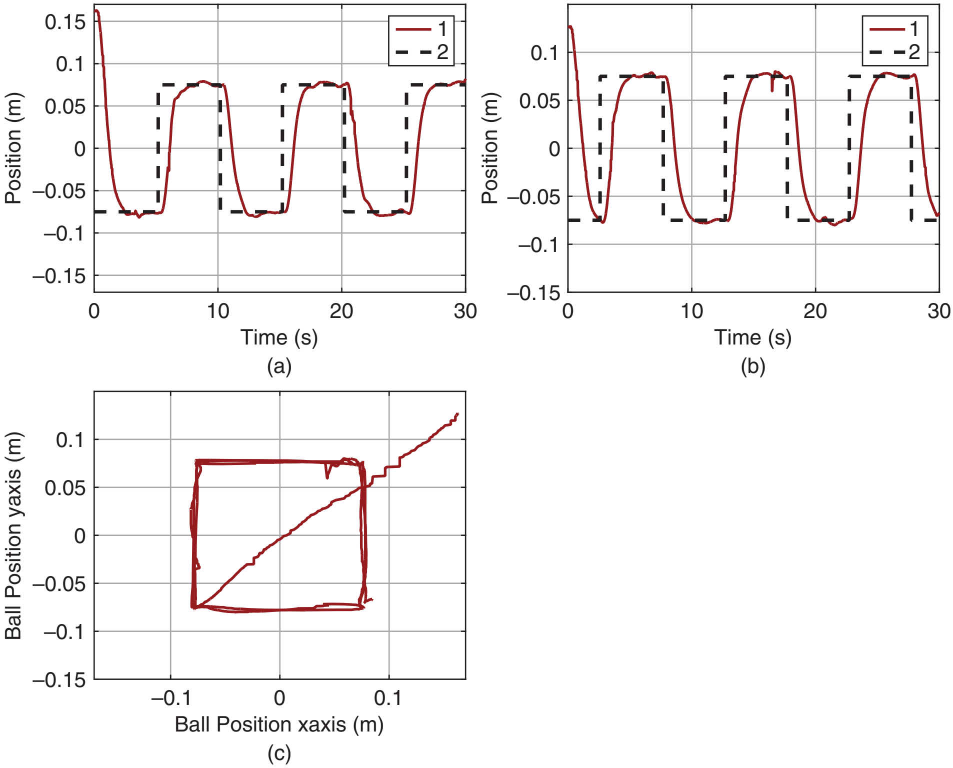


Figure 6.5 Making a square movement. (a) x-axis response. (b) y-axis response. (c) Ball position. Key: line (1) output response; line (2) reference signal.

As stated before, the closed-loop control using a resonant controller embedding the mode $s^2 + (\frac{2\pi}{2.5})^2$ into its denominator is not stable possibly due to modeling errors. The solution resorts to PID control with pre-compensation by modifying the reference signals.

To find the suitable reference signals for the circle movement, the desired sinusoidal reference signals $x^*(t)$ and $y^*(t)$ are applied to the same PID control systems. Then, at steady-state operation, the ratio between the peaks of the reference and output is calculated as 0.2566, and the phase lag between the desired sinusoidal signal $x^*(t)$ and $x(t)$ is 3.8973 rad. Therefore, the reference signals to the PID control systems are modified for the pre-compensation as

$$\bar{x}^*(t) = \frac{0.075}{0.2566}\cos\left(\frac{2\pi}{2.5}t - 3.8973\right)$$
$$\bar{y}^*(t) = \frac{0.075}{0.2566}\sin\left(\frac{2\pi}{2.5}t - 3.8973\right).$$

Figures 6.6(a) and (b) compare the x-axis and y-axis responses to the original desired sinusoidal reference signals. Indeed, with the pre-compensation, the output signals track the desired reference signals quite nicely. Figure 6.6(c) shows the x–y plane movement of the ball, which is seen to make a circle movement.

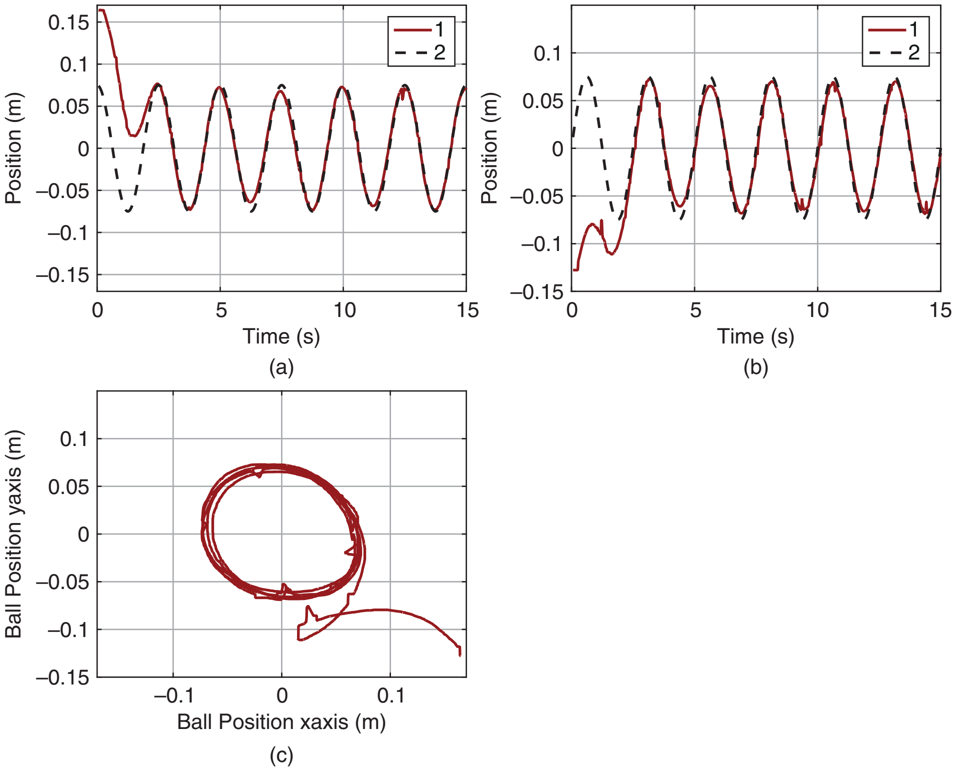


Figure 6.6 Making a circle movement. (a) x-axis response. (b) y-axis response. (c) Ball position. Key: line (1) output response; line (2) desired reference signal.

6.3.4.4 Making more Complicated Movements

In the final year project thesis (Lee (2013)), the control of ball's square movement was extended to more complicated maze movement by carefully choosing the step reference signals on the x- and y-axes. The control of ball's circle movement was extended to the movement of a figure "8". They are all successfully implemented and validated on the ball and plate balancing system. For those who are interested in the other more complicated cases, Lee's video demonstration can be found at https://www.youtube.com/watch?v=wiPWqmlN5N4.

6.3.5 Food for Thought

1. For the ball and plate balancing system, is it possible to choose other steady-state values for the linearization of the nonlinear model?
2. Was it a bit surprise to find that the linear model for such a complex system is a simple double integrator with gain easily determined?
3. Would you be able to list three factors that will result in modelling errors between the double integrator model and the actual ball and plate balancing system?
4. In order to overcome the modelling error, which parameter do you think that John has used to tune the closed-loop PID controller against the actual ball and plate balancing system?

6.4 Gain Scheduled PID Control Systems

Gain scheduled control for nonlinear plants has proven to be a successful design methodology in many engineering applications. The gain scheduled control system is to use linear control strategies to control a nonlinear plant, and the family of closed-loop linear systems is stable in the vicinity of each linear model.

There are four general steps involved in the design of a gain scheduled control system listed as below, which are to

1. identify the operating conditions of the nonlinear system and obtain a family of linear models with regard to these conditions;
2. perform linear control system design for the family of linear models with specified closed-loop performance for each linear system;
3. implement the actual gain scheduling that forms an interpolation between the family of the linear closed-loop control systems;
4. validate and simulate the gain scheduled control system.

We have discussed the tasks given in step 1 and step 2 in the previous chapters. In this section, we will focus on step 3. Step 4 has been illustrated with experimental validations for induction motor control (see Wang et al. (2015)) and for a fixed-wing unmanned aerial vehicle (see Poksawat et al. (2017)).

6.4.1 The Weighting Parameters

One approach used in the design of a gain scheduled control system is to assign a set of weighting parameters with values between 0 and 1 that will correspond to each operating condition of the nonlinear system. These weighting parameters will be used in the next section for the calculation of the control signal with the gain scheduled component.

As an example, the gain scheduled speed control system for AC motor control is used for visualization. Here, the parameters $\lambda^{\rm l}$, $\lambda^{\rm m}$, and $\lambda^{\rm h}$ are used as the weights for low speed, median speed, and high speed operations of a motor. The basic idea is to assign a correct value to the weighting parameters according to the operating conditions, which can be identified through the measurement of a physical variable. For the motor control application, its velocity is readily measured, which also influences the parameters of the linear model.

The first approach, also the simplest approach, is to assign the weighting parameters according to the reference signal of the system. We have following three cases. When we choose the reference signal ω^* to be $\omega^{\rm l}$ for low speed operation of the AC motor, then $\lambda^{\rm l} = 1$, $\lambda^{\rm m} = 0$ and $\lambda^{\rm h} = 0$. When we choose the reference signal ω^* to be $\omega^{\rm m}$ for median speed operation, $\lambda^{\rm l} = 0$, $\lambda^{\rm m} = 1$ and $\lambda^{\rm h} = 0$. when the desired velocity is at the high speed where $\omega^* = \omega^{\rm h}$, $\lambda^{\rm l} = 0$, $\lambda^{\rm m} = 0$ and $\lambda^{\rm h} = 1$.

This approach takes into consideration the changes in plant dynamics due to reference changes; however, it does not consider the possibility that disturbances could cause the significant changes in plant dynamics. Hence, with this simple approach, closed-loop instability could occur if severe disturbances were encountered in plant operation.

The more general approach is to compute the weighting parameters $\lambda^{\rm l}$, $\lambda^{\rm m}$, and $\lambda^{\rm h}$ according to the actual measurement of velocity ω. In order to avoid random triggering of the model changes in the presence of noises and transient responses, a band is formed around the desired speed. By assigning a tolerance constant δ to the desired speed ranges, the weighting constants $\lambda^{\rm l}$, $\lambda^{\rm m}$, and $\lambda^{\rm h}$ are defined as

$$\begin{aligned}
-\delta + \omega^{\rm l} \le \omega \le \omega^{\rm l} + \delta \quad & \lambda^{\rm l} = 1; \lambda^{\rm m} = 0; \lambda^{\rm h} = 0 \\
-\delta + \omega^{\rm m} \le \omega \le \omega^{\rm m} + \delta \quad & \lambda^{\rm l} = 0; \lambda^{\rm m} = 1; \lambda^{\rm h} = 0 \\
-\delta + \omega^{\rm h} \le \omega \le \omega^{\rm h} + \delta \quad & \lambda^{\rm l} = 0; \lambda^{\rm m} = 0; \lambda^{\rm h} = 1.
\end{aligned}$$

Outside the band of the desired speed, none of the linear models can accurately describe the dynamic system. A traditional method is to use a combination of these two models from the nearest regions. For instance, assuming that the actual operating condition is between the band of the desired median speed and that of the desired high speed ($\omega^{\rm m} + \delta \le \omega(t) < \omega^{\rm h} - \delta$), by defining $\lambda^{\rm h}$ ($0 \le \lambda^{\rm h} \le 1$) as a function of $\omega(t)$, $\lambda^{\rm h}(t)$ is calculated using the linear interpretation of the two boundaries between the median and high speeds given by:

$$\lambda^{\rm h}(t) = \frac{\omega(t) - \omega^{\rm m} - \delta}{\omega^{\rm h} - \omega^{\rm m} - 2\delta}. \tag{6.47}$$

The weighting parameter $\lambda^{\rm m}$ follows as $\lambda^{\rm m} = 1 - \lambda^{\rm h}$ ($0 \le \lambda^{\rm m} \le 1$), and $\lambda^{\rm l} = 0$ for this region. Similarly, for $\omega^{\rm l} + \delta \le \omega(t) < \omega^{\rm m} - \delta$,

$$\lambda^{\rm m}(t) = \frac{\omega(t) - \omega^{\rm l} - \delta}{\omega^{\rm m} - \omega^{\rm l} - 2\delta} \tag{6.48}$$

and $\lambda^{\rm l}(t) = 1 - \lambda^{\rm m}(t)$, $\lambda^{\rm h} = 0$.

Figure 6.7 illustrates the weighting parameters that have been used to represent the operating regions of the AC motor.

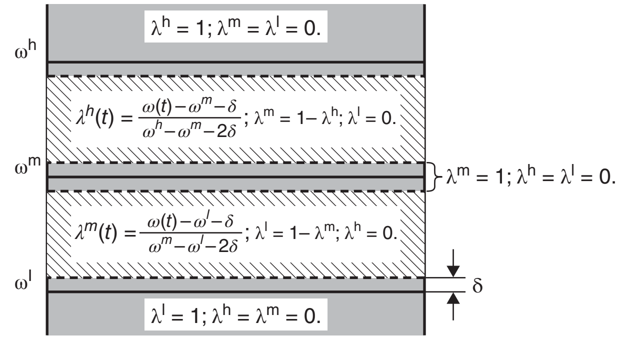

$\lambda^h = 1; \lambda^m = \lambda^l = 0.$
ω^h
$\lambda^h(t) = \frac{\omega(t)-\omega^m-\delta}{\omega^h-\omega^m-2\delta}; \lambda^m = 1-\lambda^h; \lambda^l = 0.$
ω^m
$\lambda^m = 1; \lambda^h = \lambda^l = 0.$
$\lambda^m(t) = \frac{\omega(t)-\omega^l-\delta}{\omega^m-\omega^l-2\delta}; \lambda^l = 1-\lambda^m; \lambda^h = 0.$
ω^l
δ
$\lambda^l = 1; \lambda^h = \lambda^m = 0.$

Figure 6.7 Weighting parameters.

6.4.2 Gain Scheduled Implementation using PID Velocity Form

One of the key elements in the implementation of a gain scheduled control system is that the calculated control signal is required to be the actual control signal, not the control signal in deviation variables. The difference between them is the steady-state value of the control signal U_{ss}. Because when the operating condition changes the value U_{ss} changes, as we noticed from the exercises in linearization. This means that if we calculate the deviation control variable, this steady-state value needs to be adapted when the operating condition changes.

However, in the PID controller implementation shown in Chapter 4, the control signal was discretized with its steady-state value added to give the actual control signal, for example, see (4.27), which is shown at the sampling time t_i, as

$$u_{\text{act}}(t_i) = u_{\text{act}}(t_{i-1}) + K_c(e(t_i) - e(t_{i-1})) + \frac{K_c \Delta t}{\tau_I} e(t_i) - u_{\text{D}}(t_i) + u_{\text{D}}(t_{i-1}) \tag{6.49}$$

where the feedback error is calculated as $e(t_i) = r_{\text{act}}(t_i) - y_{\text{act}}(t_i)$, and the derivative control signal $u_{\text{D}}(t_i)$ is expressed in relation to the actual output signal $y_{\text{act}}(t_i)$ as

$$u_{\text{D}}(t_i) = \frac{\tau_{\text{f}}}{\tau_{\text{f}} + \Delta t} u_{\text{D}}(t_{i-1}) + \frac{K_c \tau_{\text{D}}}{\tau_{\text{f}} + \Delta t}(y_{\text{act}}(t_i) - y_{\text{act}}(t_{i-1})).$$

Thus, the remaining task in the gain scheduled PID controller implementation is to perform the interpolation between the family of the linear closed-loop control systems.

We assume that with the three operating conditions selected, there are three PID controllers designed and simulated to obtain desired closed-loop performance for each operating condition. For clarity, we use the superscripts l, m, and h to denote the PID controller parameters obtained at the low, median, and high operating conditions. The identifiers λ^{l}, λ^{m} and λ^{h} are calculated in real time using the actual output $y_{\text{act}}(t_i)$, which is equivalent to the velocity $\omega(t_i)$ in the previous section. Then, the calculation of the actual control signal with a gain scheduled controller becomes:

$$\begin{aligned} u_{\text{act}}(t_i) &= u_{\text{act}}(t_{i-1}) \\ &\quad + \lambda^{\text{l}} \left[K_c^{\text{l}}(e(t_i) - e(t_{i-1})) + \frac{K_c^{\text{l}} \Delta t}{\tau_I^{\text{l}}} e(t_i) - u_{\text{D}}(t_i) + u_{\text{D}}(t_{i-1}) \right] \end{aligned}$$

$$
\begin{aligned}
&+ \lambda^{\mathrm{m}} \left[K_c^{\mathrm{m}}(e(t_i) - e(t_{i-1})) + \frac{K_c^{\mathrm{m}} \Delta t}{\tau_I^{\mathrm{m}}} e(t_i) - u_{\mathrm{D}}(t_i) + u_{\mathrm{D}}(t_{i-1}) \right] \\
&+ \lambda^{\mathrm{h}} \left[K_c^{\mathrm{h}}(e(t_i) - e(t_{i-1})) + \frac{K_c^{\mathrm{h}} \Delta t}{\tau_I^{\mathrm{h}}} e(t_i) - u_{\mathrm{D}}(t_i) + u_{\mathrm{D}}(t_{i-1}) \right]
\end{aligned}
\tag{6.50}
$$

where u_{D} is calculated using the following scheduled expression:

$$
\begin{aligned}
u_{\mathrm{D}}(t_i) = {} & \lambda^{\mathrm{l}} \left[\frac{\tau_{\mathrm{f}}^{\mathrm{l}}}{\tau_{\mathrm{f}}^{\mathrm{l}} + \Delta t} u_{\mathrm{D}}(t_{i-1}) + \frac{K_c^{\mathrm{l}} \tau_{\mathrm{D}}^{\mathrm{l}}}{\tau_{\mathrm{f}}^{\mathrm{l}} + \Delta t} (y_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_{i-1})) \right] \\
&+ \lambda^{\mathrm{m}} \left[\frac{\tau_{\mathrm{f}}^{\mathrm{m}}}{\tau_{\mathrm{f}}^{\mathrm{m}} + \Delta t} u_{\mathrm{D}}(t_{i-1}) + \frac{K_c^{\mathrm{m}} \tau_{\mathrm{D}}^{\mathrm{m}}}{\tau_{\mathrm{f}}^{\mathrm{m}} + \Delta t} (y_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_{i-1})) \right] \\
&+ \lambda^{\mathrm{h}} \left[\frac{\tau_{\mathrm{f}}^{\mathrm{h}}}{\tau_{\mathrm{f}}^{\mathrm{h}} + \Delta t} u_{\mathrm{D}}(t_{i-1}) + \frac{K_c^{\mathrm{h}} \tau_{\mathrm{D}}^{\mathrm{h}}}{\tau_{\mathrm{f}}^{\mathrm{h}} + \Delta t} (y_{\mathrm{act}}(t_i) - y_{\mathrm{act}}(t_{i-1})) \right].
\end{aligned}
\tag{6.51}
$$

To switch on the gain scheduled PID controller, the first sample of the control signal $u_{\mathrm{act}}(t_0)$ will use the actual open-loop control signal, which is the estimate of the steady-state value U_{ss} at an operating condition. The values of λ^{l}, λ^{m}, and λ^{h} are continuously updated using the actual output or other physical parameter to pin-point the operating condition of the nonlinear system.

6.4.3 Gain Scheduled Implementation using an Estimator Based PID Controller

As an illustration, we will discuss the estimator based PI controller for the gain scheduled implementation. The extension to PID controller using an estimator is left as an exercise.

We assume that the system has three operating conditions denoted by the superscripts l, m, and h. At each operating condition, a first order model is obtained together with the steady-state values of the input signal U_{ss} and output signal Y_{ss}.

At an operating condition, the steady-state values U_{ss} and Y_{ss} are used to obtain the first order differential equation:

$$
\dot{y}(t) = -ay(t) + b(u(t) + d(t)) \tag{6.52}
$$

where a and b are the model coefficients obtained at the operating condition, and $u(t)$ and $y(t)$ are the input and output deviation variables (or small signals), defined as

$$
u(t) = u_{\mathrm{act}}(t) - U_{\mathrm{ss}}; \quad y(t) = y_{\mathrm{act}}(t) - Y_{\mathrm{ss}}.
$$

Note that in this formulation, the steady-state value Y_{ss} can be chosen to correspond to the reference signal at the operating condition, which is known in applications. It is more difficult to accurately determine the value of U_{ss}, but as in the velocity controller implementation, we can give a rough estimate of U_{ss} by using the initial open-loop control signal. Afterwards, the estimated constant disturbance $\hat{d}(t)$ in (6.52) will provide the compensation for the error in the steady-state value of the control signal U_{ss}.

As in Chapter 5, two desired closed-loop poles $-\alpha_1$ and $-\alpha_2$ are specified where $\alpha_1 > 0, \alpha_2 > 0$. The controller and estimator parameters are calculated for the low speed operating condition as

$$\begin{aligned} K_1^{\mathrm{l}} &= \frac{\alpha_1 - a^{\mathrm{l}}}{b^{\mathrm{l}}} \\ K_2^{\mathrm{l}} &= \frac{\alpha_2}{b^{\mathrm{l}}} \end{aligned}$$

for median speed operating condition as

$$\begin{aligned} K_1^{\mathrm{m}} &= \frac{\alpha_1 - a^{\mathrm{m}}}{b^{\mathrm{m}}} \\ K_2^{\mathrm{m}} &= \frac{\alpha_2}{b^{\mathrm{m}}} \end{aligned}$$

and for the high speed operating condition as

$$\begin{aligned} K_1^{\mathrm{h}} &= \frac{\alpha_1 - a^{\mathrm{h}}}{b^{\mathrm{h}}} \\ K_2^{\mathrm{h}} &= \frac{\alpha_2}{b^{\mathrm{h}}}. \end{aligned}$$

Because the control signal and output signal are small signals, they will be calculated in terms of their operating conditions. For this purpose, at the sampling instance t_i, we define the output signals for the three operating conditions:

$$\begin{aligned} y^{\mathrm{l}}(t_i) &= y_{\mathrm{act}}(t_i) - Y_{\mathrm{ss}}^{\mathrm{l}} \\ y^{\mathrm{m}}(t_i) &= y_{\mathrm{act}}(t_i) - Y_{\mathrm{ss}}^{\mathrm{m}} \\ y^{\mathrm{h}}(t_i) &= y_{\mathrm{act}}(t_i) - Y_{\mathrm{ss}}^{\mathrm{h}}. \end{aligned}$$

The control signal in deviation form is calculated using the identifiers λ^{l}, λ^{m}, and λ^{h} introduced in Section 6.4.1 using the following form:

$$\begin{aligned} u(t_i) = {} & -\lambda^{\mathrm{l}}((K_1^{\mathrm{l}} + K_2^{\mathrm{l}})y^{\mathrm{l}}(t_i) + \hat{z}^{\mathrm{l}}(t_i)) - \lambda^{\mathrm{m}}((K_1^{\mathrm{m}} + K_2^{\mathrm{m}})y^{\mathrm{m}}(t_i) \\ & + \hat{z}^{\mathrm{m}}(t_i)) - \lambda^{\mathrm{h}}((K_1^{\mathrm{h}} + K_2^{\mathrm{h}})y^{\mathrm{h}}(t_i) + \hat{z}^{\mathrm{h}}(t_i)). \end{aligned} \tag{6.53}$$

At sampling time t_i, the three estimators will run in parallel to estimate the disturbance term at different operating conditions:

$$\begin{aligned} \hat{z}^{\mathrm{l}}(t_{i+1}) &= \hat{z}^{\mathrm{l}}(t_i) - (\alpha_2 \hat{z}^{\mathrm{l}}(t_i) + \frac{\alpha_2(\alpha_2 - a^{\mathrm{l}})}{b^{\mathrm{l}}} y^{\mathrm{l}}(t_i) + \alpha_2 u(t_i))\Delta t \\ \hat{z}^{\mathrm{m}}(t_{i+1}) &= \hat{z}^{\mathrm{m}}(t_i) - (\alpha_2 \hat{z}^{\mathrm{m}}(t_i) + \frac{\alpha_2(\alpha_2 - a^{\mathrm{m}})}{b^{\mathrm{m}}} y^{\mathrm{m}}(t_i) + \alpha_2 u(t_i))\Delta t \\ \hat{z}^{\mathrm{h}}(t_{i+1}) &= \hat{z}^{\mathrm{h}}(t_i) - (\alpha_2 \hat{z}^{\mathrm{h}}(t_i) + \frac{\alpha_2(\alpha_2 - a^{\mathrm{h}})}{b^{\mathrm{h}}} y^{\mathrm{h}}(t_i) + \alpha_2 u(t_i))\Delta t. \end{aligned}$$

The actual control signal is constructed using

$$u_{\mathrm{act}}(t_i) = u(t_i) + U_{\mathrm{ss}}.$$

The steady-state value U_{ss} does not need to be accurate because of the estimation of disturbance. At the start of the closed-loop control, this parameter could take the value of the open-loop control signal. When operating condition changes, it can use the converged steady-state value of the control signal from the previous operating condition as the new steady-state value for the changed operating condition.

6.4.4 Food for Thought

1. In the calculation of the weighting parameters, λ^l, λ^m and λ^h, can we use other physical parameters beside the output variable, to identify the operating conditions?
2. In the gain scheduled PID controller when using the velocity form, do you need the steady-state information of the plant input and output?
3. In the gain scheduled disturbance observer-based PID controller, do you need the steady-state information of the plant input and output?

6.5 Summary

Many physical models are represented by nonlinear differential equations. In order to design PID controllers for those physical systems, linearization of the nonlinear models is required. In this chapter, we have discussed how to obtain a linear model when a nonlinear model in the form of differential equations is given and how to design a gain scheduled control system for a nonlinear system. The other important aspects of this chapter are summarized as follows.

- An operating condition is selected to obtain a linear model for a given nonlinear model. A linear model is valid with respect to a chosen operating condition. If the operating condition changes, the linear model will change.
- In the linearization process, it is not always possible for us to find the correct equilibrium point due to parameter uncertainty. If this happens, we could model the uncertainty with a constant disturbance to the system. This constant disturbance will be overcome by the integral action in the controller or by subtracting it from the control signal with the disturbance estimation as proposed in Chapter 5.
- If the nonlinearity is severe, a gain scheduled control system is desired. With the gain scheduled PID control system, a family of linear models and controllers is obtained first, and the controller is smoothly switched when the operating condition changes. We can realize the gain scheduled PID control systems using the PID controller in velocity form or using the disturbance observer- based PID controllers.

6.6 Further Reading

1. Text books for nonlinear control include Isidori (2013), Khalil (2002), Nijmeijer and Van der Schaft (1990).
2. Henson and Seborg (1997) presents advanced topics in nonlinear process control.
3. pH neutralization process is a typical example for nonlinear systems (Henson and Seborg (1994), Kalafatis et al. (1995), Böling et al. (2007)). It is a Wiener nonlinear model with its static inverse identified in Kalafatis et al. (1997), then compensated with the inverse nonlinearity, leading to a gain-scheduled control system (Kalafatis et al. (2005)). Another successful application of Hamerstein-Wiener model with a gain-scheduled control system is runtime management of Quality of Service (QoS) performance and resource provisioning in shared resource software environments (Patikirikorala et al. (2012)).

4. The gain scheduled PID control algorithm presented in this chapter has been successfully used to attitude control of fixed-wing unmanned aerial vehicle with experimental validations (Poksawat et al. (2017)). The gain scheduled algorithm has an extension to continuous-time model predictive control, which was successfully validated on induction motor control (Wang et al. (2015)).

Problems

6.1 *Perform linearization of the following differential equations and find their transfer functions at the operating conditions.*

1. *A dynamic system is described by the following differential equation:*

$$\dot{x}(t) = 2x(t)u(t) + x^2(t)u^2(t) + u^2(t)$$

Find the linearized model at $x^0 = 1$ *and* $u^0 = 1$. *Is this system stable at this operating condition?*

2. *A dynamic system is described by the following differential equation:*

$$\ddot{x}(t) = 2\sqrt{x(t)+6} + x^2(t)u(t) + u^2(t)$$

Find the linearized model at $x^0 = -1$ *and* $u^0 = -1$. *Where are the poles of this linearized system?*

6.2 *Perform linearization of a differential equation for an inverted pendulum on a cart. The differential equations that describe the motion of an inverted pendulum are:*

$$(m+M)\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F \tag{6.54}$$

$$(I+ml^2)\ddot{\theta} + ml\ddot{x}cos\theta - mglsin\theta = 0 \tag{6.55}$$

where $\theta(t)$ *is the angle of the pendulum,* $x(t)$ *is the horizontal position of the cart,* $2l$ *is the length of the pendulum, M and m are the masses of the cart and the pendulum, respectively, and I is the moment of inertia of the pendulum about its center of gravity. F is a force applied to the body of the cart.*

The operating point of the pendulum is selected as, $\theta^0 = \dot{\theta}^0 = \ddot{\theta}^0 = 0$, *and* $x^0 = \dot{x}^0 = \ddot{x}^0 = 0$, *and* $F^0 = 0$.

1. *Verify the linearized differential equations at the operating point are:*

$$(m+M)\ddot{x} + ml\ddot{\theta} = F \tag{6.56}$$

$$(I+ml^2)\ddot{\theta} + ml\ddot{x} - mgl\theta = 0 \tag{6.57}$$

2. *Find the Laplace transfer function between the force F (the input variable) and the pendulum angle* θ.

6.3 *A Permanent Magnetic Synchronous Motor (PMSM) is described by the differential equations in the d-q rotating reference frame:*

$$\frac{di_d(t)}{dt} = \frac{1}{L_d}(v_d(t) - Ri_d(t) + \omega_e(t)L_q i_q(t)) \tag{6.58}$$

$$\frac{di_q(t)}{dt} = \frac{1}{L_q}(v_q(t) - Ri_q(t) - \omega_e(t)L_d i_d(t) - \omega_e(t)\phi_{mg}) \tag{6.59}$$

$$\frac{d\omega_e(t)}{dt} = \frac{p}{J}(T_e - \frac{B}{p}\omega_e(t) - T_L) \tag{6.60}$$

$$T_e = \frac{3}{2}p\phi_{mg}i_q \tag{6.61}$$

where ω_e is the angular electrical speed and is related to the rotor speed by $\omega_e = p\omega_m$ with p denoting the number of pole pairs, v_d and v_q represent the stator voltages in the d-q frame, i_d and i_q represent the stator currents in this frame, and T_L is load torque that is assumed to be zero if no load is attached to the motor.
The operating point for the PMSM is defined as i_d^0, i_q^0 and ω_e^0. Find the linearized differential equations that describe the dynamics of the motor at the operating point.

6.4 *A continuous-time system is described by the differential equation,*

$$\dot{x}(t) = -x(t)^2 + 2x(t)u(t) - u(t)^3 \tag{6.62}$$

Supposing that the operating point for the input signal is at $u^0 = 1$, the operating point for $x(t)$ could be determined by letting $\dot{x}(t) = 0$ and solving the algebraic equation with the value of $u^0 = 1$, leading to

$$-(x^0)^2 + 2(x^0) - 1 = 0 \tag{6.63}$$

1. *What is the operating point for $x(t)$? Find the linearized model against the operating points of x^0 and u^0.*
2. *Find the Laplace transfer function between the input variable $u(t) - u^0$ and the output variable $x(t) - x^0$. Is this system stable at the operating point?*
3. *Design a PI controller for this system, where the closed-loop poles are positioned at -3.*

6.5 *A dynamic system is described by the following differential equation:*

$$\ddot{y}(t) = -y(t) + y(t)|u(t)| + u^2(t)$$

1. *Find the linearized model at $y^0 = 1$ and $u^0 = 1$. Hint: we need to consider the case $u(t) > 0$ and $u(t) < 0$ separately to obtain two linear systems depending on the sign of the control signal $u(t)$.*
2. *Design two PID controllers for this system in terms of both positive $u(t)$ and negative $u(t)$, where all closed-loop poles are positioned at $-\alpha$ ($\alpha > 0$), which is selected to produce stable closed-loop systems with satisfactory performance. α could be different for the two controllers. A starting point is to select $\alpha = 1$.*
3. *Implement the gain scheduled PID control system using the sign of control signal $u(t)$ as the identifier of the operating condition. We modify the PIDV.slx real-time function to incorporate the gain scheduled control feature. In the simulation, the reference signal is chosen to be zero, and added to the simulation is an input disturbance, which is a square wave signal with appropriate period 2 and amplitude of ± 2. The sampling interval Δt is selected to be 0.01.*

6.6 *Consider the linearized model and the nonlinear plant given in Problem 6.5.*

1. *Design two disturbance observer-based PID controllers for the same nonlinear plant depending on the sign of the control signal, where all desired closed-loop poles for both controller and estimator are positioned at* $-\alpha$ ($\alpha > 0$)*. The parameter* α*, which could be different, is selected to produce stable closed-loop systems with satisfactory performance. A starting point is to select* $\alpha = 1$.
2. *Modify the MATLAB real-time function PIDEstim.slx introduced in Tutorial 5.2 to include the gain-scheduled control feature. Simulate the closed-loop control system for input disturbance rejection with the same simulation condition outlined in Problem 6.5.*

7

Cascade PID Control Systems

7.1 Introduction

Cascade PID control systems are very important for successful applications. Most physical systems are complex in dynamics and cascade PID control systems offer effective solutions to their control problems, where a complex system is decomposed into several smaller and simpler systems for which PID control systems are designed for each of them.

This chapter will introduce the general design procedures for a cascade control system. The advantages of using cascade control when there are disturbances and nonlinearities in actuators are investigated and illustrated through simulation studies.

7.2 Design of a Cascade PID Control System

The PID and resonant controllers introduced in the previous chapters are the building blocks in a cascade control system design. Thus, the design of a cascade control system is a combination of the simpler control systems.

7.2.1 Design Steps for a Cascade Control System

A typical system suitable for cascade control is shown in Figure 7.1. For this system, the mathematical model of the system is described by $G_s(s)G_p(s)$ and there is an input disturbance $d_i(t)$, where s is the differential operator in this time domain environment. More importantly, the variable between the transfer functions, $x_1(t)$, is measurable.

With the assumption that the variable $x_1(t)$ is measurable, the cascade feedback control system is configured as shown in Figure 7.2, where an inner-loop system is formed with the feedback from $x_1(t)$ with the reference signal $x_1^*(t)$ to the inner-loop generated by the outer-loop controller $C_p(s)$.

The inner-loop system is called the secondary system, which is, thus, denoted with transfer function model $G_s(s)$ and controller $C_s(s)$. The outer-loop system is called the primary system, denoted with transfer function model $G_p(s)$ and controller $C_p(s)$. Clearly, the link between the secondary and primary control systems is the reference signal $x_1^*(t)$, which is the output of the primary controller.

There are four steps involved with the design of a cascade control system as listed below.

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

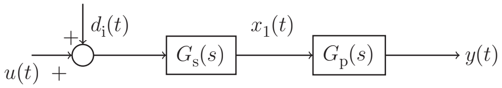


Figure 7.1 Block diagram for a system suitable for cascade control.

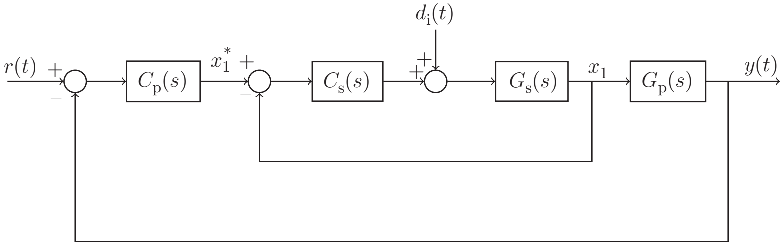


Figure 7.2 Block diagram of a cascade control system.

1. A complex system is decomposed into a series of first order or second order subsystems based on the considerations of physical relationships and availability of measurements.
2. Design P, PI, PID or PD controllers for each of the subsystems depending on the requirements. In general, the outer- loop systems are required to contain integral action to eliminate steady-state errors. In the design process, the inner-loop control system is designed first and the closed-loop transfer function for the inner-loop system is obtained. The outer-loop control system is designed based on the outer-loop system, where the relatively small time constants resulting from the inner closed- loop system are neglected, but its steady-state gain is taken into account in the outer-loop model. However, in the situation where the inner-loop control system dynamics could be captured and taken into the design of outer-loop controller, the closed-loop performance for the cascade control system should be improved in general (see Section 9.7.3).
3. Robust stability and performance analysis are performed, and closed-loop performances are adjusted using the bandwidths of the inner-loop and outer-loop systems. This step is important because there are neglected dynamics in the cascade control system. In principle, the bandwidth of the inner closed-loop control should be much wider than the one used in the outer closed-loop control. Namely, the inner-loop control system should have a much faster response speed for obtaining the closed-loop stability of the cascade control system.
4. In the implementation, a wider bandwidth for the secondary closed-loop system is desired and also achieved by putting proportional control K_c on the feedback error.

7.2.2 Simple Design Examples

Example 7.1 *A physical system is described by two continuous time transfer functions connected in series (see Figure 7.1), where $G_s(s)$ represents the actuator dynamics and $G_p(s)$ is the primary system to be controlled. The variable $x_1(t)$ is measurable and can be used as a feedback signal. Here, the transfer functions are*

$$G_s(s) = \frac{5}{s+10}; \quad G_p(s) = \frac{0.005}{s+0.05}.$$

Design a cascade control system with two PI controllers based on the configuration illustrated in Figure 7.2. For simplicity, we select the damping coefficient $\xi = 0.707$ for both inner- and outer-loop control systems and use the bandwidths w_{ns} and w_{np} as the tuning parameters of the inner (secondary) and outer-loop (primary) systems respectively.

Solution. *It is reasonable to choose the parameters w_{ns} and w_{np} based on the open-loop poles of the secondary and primary systems. For the inner-loop control system, we choose $w_{\text{ns}} = 5 \times 10 = 50$ leading to a pair of closed-loop poles at $-35.35 \pm j35.35$, and for the outer-loop system, we choose $w_{\text{np}} = 4 \times 0.05 = 0.2$ leading to a pair of closed-loop poles at $-0.1414 \pm \text{j}0.1414$. These selections give us the ratio of inner-loop bandwidth to outer-loop bandwidth of 250.*

In the design of cascade control system, we start the design from the inner-loop controller. From the pole assignment PI controller design outlined in Chapter 3, we obtain the PI controller parameters for the inner-loop controller as

$$K_{\text{cs}} = \frac{2\xi w_{\text{ns}} - a}{b} = \frac{2\xi w_{\text{ns}} - 10}{5} = 12.14;$$
$$\tau_{\text{Is}} = \frac{2\xi w_{\text{ns}} - a}{w_{\text{ns}}^2} = \frac{2\xi w_{\text{ns}} - 10}{w_{\text{ns}}^2} = 0.0243$$

where the parameters $a = 10$ and $b = 5$ for the inner-loop system, and $w_{\text{ns}} = 50$ with damping coefficient $\xi = 0.707$. The closed-loop transfer function between the reference signal $X_1^*(s)$ and the output signal $X_1(s)$ is calculated as,

$$\frac{X_1(s)}{X_1^*(s)} = \frac{(2\xi w_{\text{ns}} - 10)s + w_{\text{ns}}^2}{s^2 + 2\xi w_{\text{ns}} s + w_{\text{ns}}^2}. \tag{7.1}$$

To design the outer-loop controller, we consider the transfer function between $X_1^*(s)$ and the output $Y(s)$, which is

$$\frac{Y(s)}{X_1^*(s)} = \frac{(2\xi w_{\text{ns}} - 10)s + w_{\text{ns}}^2}{s^2 + 2\xi w_{\text{ns}} s + w_{\text{ns}}^2} \frac{0.005}{s + 0.05}. \tag{7.2}$$

In the design of the outer-loop controller, it is assumed that the dynamic response from the inner-loop system is much faster than the outer-loop (primary) system, the dynamics from the inner-loop system are neglected. For instance, when w_{ns} is 50, the inner-loop transfer function is approximated by a unit gain, that is

$$\frac{X_1(s)}{X_1^*(s)} = \frac{\frac{(2\xi w_{\text{ns}} - 10)}{w_{\text{ns}}^2} s + 1}{\frac{1}{w_{\text{ns}}^2} s^2 + \frac{2\xi}{w_{\text{ns}}} s + 1} \approx 1. \tag{7.3}$$

This approximation is based on the consideration that the inner-loop–closed-loop transfer function has much smaller time constants than those of the outer-loop system. Additionally, the integral action in the inner-loop controller ensures that the closed-loop transfer function has a unit steady-state value. Thus, the design of the outer-loop controller is simplified by considering the outer-loop model only. For this example, the PI controller parameters for the outer-loop controller are chosen as

$$K_{\text{cp}} = \frac{2\xi w_{\text{np}} - 0.05}{0.005} = 46.56; \quad \tau_{\text{Ip}} = \frac{2\xi w_{\text{np}} - 0.05}{w_{\text{np}}^2} = 5.82$$

where $w_{np} = 0.2$ and $\xi = 0.707$. Because in the design of the outer-loop controller we neglect the inner-loop dynamics, the robustness of the cascade control system is an issue that needs to be investigated. For this example, we can calculate the actual closed-loop poles of this cascade control system by including the inner-loop control systems.

One can verify that there are four closed-loop poles with the following values: $-35.2335 \pm j35.4441$ *and* $-0.1415 + \pm j0.1415$. *It is interesting to note that the pair of dominant closed-loop poles are almost equal to the performance specifications from the outer-loop control system and the remaining pair is close to the performance specification from the inner-loop control system. This means that the coupling between the inner-loop controller design and outer-loop controller design is almost broken because the approximation given by (7.3) is quite accurate.*

One can also verify that by choosing the ratio between the inner-loop bandwidth and the outer-loop bandwidth to be 250 with $w_{ns} = 25$ and $w_{np} = 0.1$, the cascade closed-loop control system will have four closed-loop poles: $-17.6293 \pm j17.7002$, $-0.0707 \pm j0.0708$, *which are very close to the pair of desired inner-loop control system poles* $-17.6750 \pm j17.6803$ *and the outer-loop poles* $-0.0707 \pm j0.0707$. *However, by choosing the ratio to be 10 with $w_{ns} = 25$ and $w_{np} = 2.5$, the cascade closed-loop control system has four closed-loop poles:* $-15.8796 \pm j18.4590$, $-1.8204 \pm j1.8096$ *in comparison to the two pairs of desired closed-loop poles:* -17.6750 ± 17.6803, $-1.7675 \pm j1.7680$. *It is clearly seen that there are some discrepancies between the closed-loop poles specified for the inner-loop and outer-loop systems and those that have actually been obtained in the cascade control system. However, with the ratio being* 10, *the closed-loop poles of the cascade control system remain close to those that were originally specified.*

In the applications of the cascade control system, the controller for the inner-loop system could be selected to be a proportional controller. Being a proportional controller, one can simply tune the controller parameter K_c. This is particularly useful in the situation where the model for the inner-loop system is nonlinear and not easy to obtain. The following example is to illustrate the cascade control using a proportional controller for the secondary plant and a PID controller for the primary plant.

Example 7.2 *The secondary system in a cascade control system is a motor that has the transfer function:*

$$G_s(s) = \frac{0.03}{s(s+30)} \tag{7.4}$$

where the output of the motor is angular position. The primary system is an undamped oscillator with the transfer function:

$$G_p(s) = \frac{0.6}{s^2+1}. \tag{7.5}$$

Design a cascade control system with inner-loop proportional control and outer-loop PID control. The outer-loop control system is specified with $\xi = 0.707$ and $w_{np} = 1$ and the remaining poles are placed at -2.

Solution. *This system is of fourth order with three poles on the imaginary axis. It would be very difficult to find a PID controller that could produce a stable closed-loop* *system. This example will illustrate how effectively two simple controllers will produce satisfactory closed-loop control results.*

The secondary system is approximated by the following integral model: $G_s(s) \approx \frac{0.001}{s}$ where the stable mode is neglected. Because the primary control system is required to have the natural frequency $w_{np} = 1$, we select the closed-loop pole for the secondary control system at -10, leading to the proportional controller $K_{cs} = 10000$. Roughly speaking, we select the ratio of inner-loop bandwidth to the outer-loop bandwidth as 10.

Because the secondary plant contains an integrator, and with the specification of fast closed-loop response, the inner-loop dynamics are neglected in the design of the outer-loop PID controller. The PID controller with filter is designed using the MATLAB function pidplace.m illustrated in Tutorial 3.2. In the design of a PID controller, the desired closed-loop polynomial is selected as

$$A_{cl}(s) = (s^2 + 2\xi w_{np}s + w_{np}^2)(s+2)^2$$

where $w_{np} = 1$ and $\xi = 0.707$. Using pidplace.m function, the PID controller parameters together with the derivative filter are found as

$$K_{cp} = 1.0784; \quad \tau_{Ip} = 0.8758; \quad \tau_{Dp} = 2.5717; \quad \tau_{fp} = 0.1847.$$

The closed-loop control performance is evaluated using closed-loop simulation studies with sampling interval $\Delta t = 0.01$ (s). A unit step reference signal is given for the closed-loop control at $t = 0$. At $t = 20$ (s), a step input disturbance with amplitude of 30 enters the simulation at the input signal to the secondary plant. Figure 7.3(a) shows the control signal to the secondary plant and Figure 7.3(b) shows the closed-loop output response. It is seen that the cascade closed-loop system is stable with characteristics similar to the performance specification given to the primary control system ($w_{np} = 1$, $\xi = 0.707$). It is also noted that the step input disturbance is completely rejected without steady-state error. There are three comments. For this particular example, both secondary and primary plants have poles on the imaginary axis. The key for the stabilization and maintaining the closed-loop performance is to choose the ratio of the control system bandwidths sufficiently large; 10 for this example. Because of the cascade control structure, the control system design is simplified into the design of a proportional

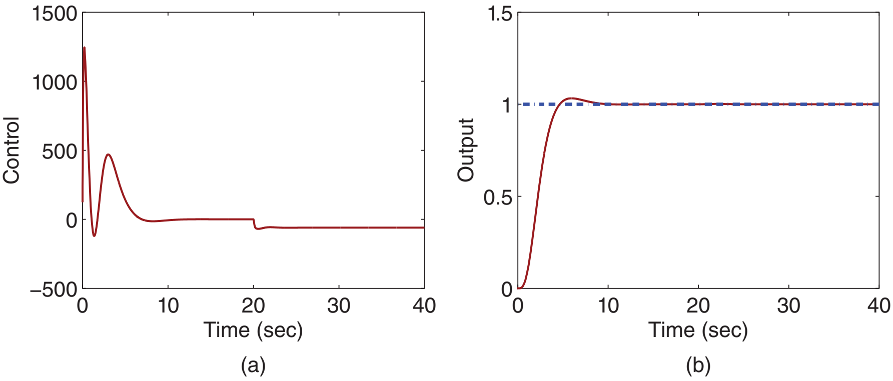


Figure 7.3 Cascade closed-loop response signals (Example 7.2, primary controller PID and secondary controller P). (a) Control signal. (b) Output.

controller and a PID controller, and both can be readily implemented. We can also verify that when the secondary plant has an integrator, a PI controller for the inner-loop system does not offer much additional advantage in a cascade structure. However, if the secondary plant does not have an integrator, there is an advantage when using a PI controller for the inner-loop system. Because the integral action in the secondary controller will produce unity of the steady-state gain from the inner-loop control system, the outer-loop controller design will only consider the dynamics from the primary plant.

7.2.3 Achieving Closed-loop Performance Invariance (Approximate) in a Cascade Structure

In the design of cascade control system, we have neglected the secondary closed-loop dynamics in the design of the primary control system, which was illustrated in Examples 7.1 and 7.2. Understanding how the neglected dynamics affect the closed-loop performance of the cascade control system is important for the design and applications.

Essentially, we will examine the variations of the closed-loop poles of the cascade control system. Assume that secondary and primary controllers and plants are described by the transfer functions illustrated in Figure 7.4.

The inner-loop transfer function between $X_1^*(s)$ and $X_1(s)$ is calculated as

$$\begin{aligned}\frac{X_1(s)}{X_1^*(s)} &= \frac{P_s(s)B_s(s)}{L_s(s)A_s(s) + P_s(s)B_s(s)} \\ &= \frac{P_s(s)B_s(s)}{A_{cls}(s)}\end{aligned} \tag{7.6}$$

where $A_{cls}(s)$ is the closed-loop polynomial for the secondary system. The closed-loop transfer function for the cascade control system becomes,

$$\frac{Y(s)}{R(s)} = \frac{P_p(s)B_p(s)P_s(s)B_s(s)}{L_p(s)A_p(s)A_{cls}(s) + P_p(s)B_p(s)P_s(s)B_s(s)}. \tag{7.7}$$

Thus, the actual closed-loop polynomial for the cascade control system is

$$\begin{aligned}A_{cl}(s) &= L_p(s)A_p(s)A_{cls}(s) + P_p(s)B_p(s)P_s(s)B_s(s) \\ &= A_{cls}(s)\left(L_p(s)A_p(s) + P_p(s)B_p(s)\frac{P_s(s)B_s(s)}{A_{cls}(s)}\right).\end{aligned} \tag{7.8}$$

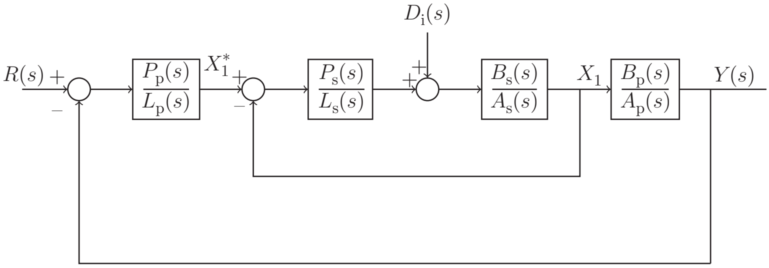


Figure 7.4 Closed-loop cascade control system.

Clearly, there are some couplings between the poles from the secondary and primary control systems, and this leads to uncertainties in the closed-loop performance of the cascade control system. However, in the design when neglecting the secondary closed-loop system, we have assumed that

$$\frac{P_s(s)B_s(s)}{A_{cls}(s)} = T(s)_s \approx 1$$

which is the closed-loop transfer function for the secondary plant, also its complementary sensitivity function. This approximation could be quantified in the frequency domain through the bandwidth of the secondary closed-loop system. If this approximation stands true, then the actual closed-loop polynomial $A_{cl}(s)$ is approximated using following expression:

$$\begin{aligned} A_{cl}(s) &\approx A_{cls}(s)(L_p(s)A_p(s) + P_p(s)B_p(s)) \\ &= A_{cls}(s)A_{clp}(s) \end{aligned} \tag{7.9}$$

where $A_{clp}(s)$ is the closed-loop polynomial for the primary control system. Therefore, the closed-loop poles of the cascade control system are found by letting

$$A_{cl}(s) \approx A_{cls}(s)A_{clp}(s) = 0$$

which is equivalent to

$$A_{cls}(s) = 0; \quad A_{clp}(s) = 0.$$

This means that in the design of a cascade control system, if the secondary control system has a sufficiently large bandwidth and has a unit steady-state gain, then the closed-loop poles of the cascade control system consist of the closed-loop poles from the secondary control system and the primary control system. We can design the secondary control system and the primary control system separately as long as the closed-loop transfer function from the secondary control system has much smaller time constants and is equal to one in steady state.

7.2.4 Food for Thought

1. Would you be able to list three control applications that have used a cascade control system structure?
2. Why have we neglected the closed-loop secondary system's dynamics in the design of PID controller for the primary system?
3. If the secondary system has no self-contained integral action and a proportional controller is used, should you consider the effect of steady-state error from the inner-loop when you design the outer-loop controller?
4. Would you say that a cascade control structure gives us a means to tackle the problem of PID control system design for a higher order and complex system?

7.3 Cascade Control System for Input Disturbance Rejection

One of the major advantages in using cascade control lies in its characteristics with regard to disturbance rejection. This section will explicitly examine the properties of input disturbance rejection in cascade control with both frequency response analysis and simulation studies.

7.3.1 Frequency Characteristics for Disturbance Rejection

To examine the effectiveness of disturbance rejection, we calculate the closed-loop transfer function between the disturbance $D_i(s)$ and the output $Y(s)$, as shown in Figure 7.4. Here, we assume the reference signal $R(s) = 0$.

The output $X_1(s)$ from the inner-loop system consists of the input disturbance $D_i(s)$ and the reference signal $X_1(s)^*$, which can be verified as,

$$\begin{aligned} X_1(s) = & \overbrace{\frac{B_s(s)P_s(s)}{A_s(s)L_s(s) + B_s(s)P_s(s)}}^{T(s)_s} X_1(s)^* \\ & + \overbrace{\frac{B_s(s)L_s(s)}{A_s(s)L_s(s) + B_s(s)P_s(s)}}^{S_i(s)_s} D_i(s). \end{aligned} \tag{7.10}$$

This relationship is governed by the complementary sensitivity function $T(s)_s$ and the input sensitivity function $S_i(s)_s$ for the secondary plant, which has the following compact form:

$$X_1(s) = T(s)_s X_1(s)^* + S_i(s)_s D_i(s). \tag{7.11}$$

The primary output $Y(s)$ is expressed as

$$\begin{aligned} Y(s) &= \frac{B_p(s)}{A_p(s)} X_1(s) \\ &= \frac{B_p(s)}{A_p(s)} (T(s)_s X_1(s)^* + S_i(s)_s D_i(s)). \end{aligned} \tag{7.12}$$

With the control signal $X_1(s)^*$ generated from the primary controller as

$$X_1(s)^* = -\frac{P_p(s)}{L_p(s)} Y(s)$$

we obtain the closed-loop transfer function from the input disturbance $D_i(s)$ to the output $Y(s)$:

$$\frac{Y(s)}{D_i(s)} = \frac{G_p(s)S_i(s)_s}{1 + G_p(s)C_p(s)T(s)_s} \tag{7.13}$$

where $G_p(s) = \frac{B_p(s)}{A_p(s)}$ and $C_p(s) = \frac{P_p(s)}{L_p(s)}$ are the transfer functions for the primary plant and primary controller, respectively.

In the frequency domain, the amplitude of the frequency response $|Y(j\omega)|$ becomes

$$\begin{aligned} |Y(j\omega)| &= |\frac{G_p(j\omega)S_i(j\omega)_s}{1 + G_p(j\omega)C_p(j\omega)T(j\omega)_s}||D_i(j\omega)| \\ &= |\frac{G_p(j\omega)}{1 + G_p(j\omega)C_p(j\omega)T(j\omega)_s}||S_i(j\omega)_s||D_i(j\omega)|. \end{aligned} \tag{7.14}$$

Note that the input sensitivity function for the secondary plant plays an important role in rejecting the disturbance occurring at the inner-loop system. In general, because the secondary control system has a much faster dynamic response than the primary control system, the magnitude of the input sensitivity function is expected to be very small in the low and medium frequency regions. Therefore, the cascade control structure is more effective in rejecting disturbance that occurs in the secondary plant.

7.3.2 Simulation Studies

The following example illustrates the characteristics of frequency response and simulation studies of the cascade control system for input disturbance rejection.

Example 7.3 *Consider the position control of a DC motor in the presence of unknown load T_{L}. The relationship between the input voltage $V(s)$ and the angular velocity of the motor $\Omega(s)$ is described by the normalized Laplace transfer function:*

$$\frac{\Omega(s)}{V(s)} = \frac{\mathrm{e}^{-ds}}{s+1} \tag{7.15}$$

where a small time delay $d = 0.0016$ (s) is used to model the delay induced by the sensing and actuation devices. The angular position $\Theta(s)$ is related to the angular velocity through integration:

$$\frac{\Theta(s)}{\Omega(s)} = \frac{1}{s}.$$

Design a cascade control system for the position control of the DC motor and show its advantages in terms of disturbance rejection of the unknown load.

Solution. *For the cascade control system design, the secondary transfer function is*

$$G_{\mathrm{s}}(s) = \frac{\mathrm{e}^{-ds}}{s+1}.$$

By neglecting the time delay and from the pole-assignment controller design, the proportional controller gain and the integral time constant are

$$K_{\mathrm{cs}} = 2\xi w_{\mathrm{ns}} - 1 = 34.35; \quad \tau_{\mathrm{Is}} = \frac{2\xi w_{\mathrm{ns}} - 1}{w_{\mathrm{ns}}^2} = 0.0550$$

where $\xi = 0.707$ and $w_{\mathrm{ns}} = 25$. The primary transfer function is

$$G_{\mathrm{p}}(s) = \frac{1}{s}$$

and the proportional controller gain and the integral time constant are

$$K_{\mathrm{cp}} = 2\xi w_{\mathrm{np}} = 3.535; \quad \tau_{\mathrm{Ip}} = \frac{2\xi}{w_{\mathrm{np}}} = 0.5656$$

where $\xi = 0.707$ and $w_{\mathrm{np}} = 2.5$.

With these PI controllers in a cascade structure, the sensitivity functions are calculated with their magnitudes of the frequency responses shown in Figures 7.5(a)–(d). The complementary sensitivity function for the secondary plant [see Figure 7.5(a)] shows that the inner-loop system has a very wide bandwidth, numerically determined as around 52 rad s^{-1}, which corresponds to the magnitude of $|T(\mathrm{j}\omega)_{\mathrm{s}}| = \frac{1}{\sqrt{2}}$. The input disturbance

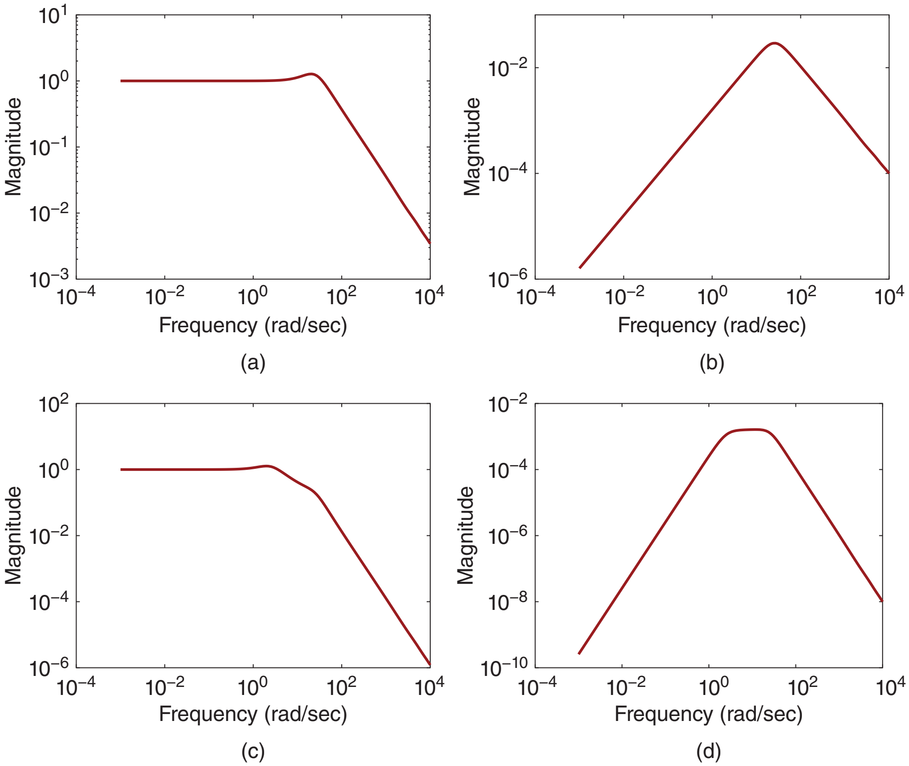


Figure 7.5 Sensitivity functions for the cascade control system (Example 7.3). (a) Complementary sensitivity (secondary). (b) Input sensitivity (secondary). (c) Complementary sensitivity (primary). (d) Input sensitivity (primary).

sensitivity for secondary plant [see Figure 7.5(b)] has a maximum amplitude of 0.029 *at* $\omega = 25$ *rad* s^{-1}*, and its magnitude is small in the low and medium frequency regions, indicating that the inner-loop controller has a good input disturbance rejection property for disturbance occurring in these regions. For the primary control system, the complementary sensitivity [see Figure 7.5(c)] shows that the closed-loop bandwidth is significantly reduced, which is numerically determined as* 5.2 *rad* s^{-1} *corresponding to the bandwidth of the primary control system without much variation. The interesting point is that the magnitude of the input sensitivity for the primary plant [see Figure 7.5(d)] has reduced further when comparing it with Figure 7.5(b). The maximum amplitude is about* 0.0016 *and almost constant in the region* $5 \leq \omega \leq 18$*, which is about* 5% *of the maximum amplitude of the input sensitivity for the inner-loop control system. This basically says that the cascaded control structure will produce superior performance in disturbance rejection for those occurring in the secondary system.*

Closed-loop simulation is performed with sampling interval $\Delta t = 0.0001$ *(s). The reference signal is chosen to be zero to simulate the situation when we are trying to maintain the motor angular position constant while adjusting the motor load. The load disturbance is chosen to be a square wave signal with amplitude changing between 0 and 100. Figure 7.6(a) shows the control signal for the cascade system required to reject*

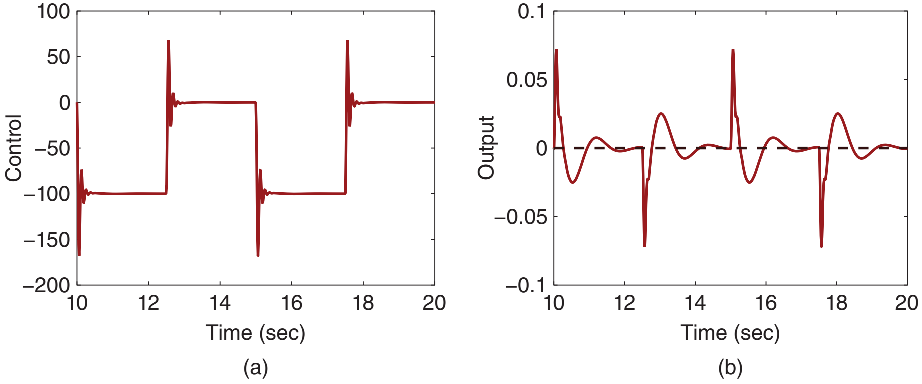


Figure 7.6 Cascade closed-loop response to square wave disturbance signal with amplitude 100 and period of 10 (Example 7.3). (a) Control signal. (b) Output signal.

this disturbance and Figure 7.6(b) shows the closed-loop output response to the load disturbance. Despite such a large load variation, the output variation is very small, indicating superior property in load disturbance rejection.

In the implementation of the PI controller for the secondary system, we always use a one-degree of freedom PI controller, which will put both proportional and integral control on the feedback error to the inner-loop system. This implementation will give us a wider bandwidth for the inner-loop complementary sensitivity $T(\mathrm{j}\omega)_\mathrm{s}$, leading to better closed-loop control results.

We leave as an exercise the comparative studies of this example without the cascade control structure, as illustrated by Figure 7.7. One might find that simple implementation of the primary controller in Example 7.3 would lead to an unstable system and one would want to try a PID controller with filter so to take into consideration the dynamics from the secondary system (see (7.15)).

7.3.3 Food for Thought

1. Would you be able to list three application examples where input disturbance existed due to uncertainties to the system?
2. Will the cascade control structure improve the rejection of output disturbance occurring at the inner-loop system?

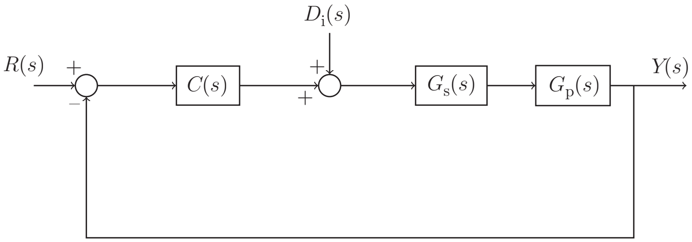


Figure 7.7 One controller for disturbance rejection.

3. If the inner-loop system has sensor bias error, will this error affect the primary control system? If so, in what way?
4. In general, in the implementation of cascade control system, we will not consider implementing the IP controller for the secondary controller. Namely, we will put both integral and proportional control on the feedback error of the inner-loop system. Why do you think this is the case?

7.4 Cascade Control System for Actuator Nonlinearities

One of the important applications when using a cascade control system is to take its advantages in dealing with actuator nonlinearities.

7.4.1 Cascade Control for Actuator with a Deadzone

A deadzone nonlinearity for an actuator, which is due to wear and tear, is described by the following equations:

$$x(t) = \begin{cases} e(t) - \delta & e(t) > \delta \\ 0 & -\delta \leq e(t) \leq \delta \\ e(t) + \delta & e(t) < \delta \end{cases} \tag{7.16}$$

where $x(t)$ is the output of the deadzone and $e(t)$ is the input to the deadzone. Figure 7.8 illustrates the deadzone nonlinearity with input signal $e(t)$ and output signal $x(t)$ from the deadzone nonlinearity. For this type of nonlinearity, if we know the size of δ, then it can be pre-compensated for with its inverse function as,

$$e(t) = \begin{cases} \hat{e}(t) + \delta & \hat{e}(t) \geq 0 \\ \hat{e}(t) - \delta & \hat{e}(t) \leq 0. \end{cases} \tag{7.17}$$

This pre-compensation will eliminate the effect of the deadzone on the control system. However, the pre-compensation requires information about the size of the deadzone that may not be readily available. Therefore, in order to obtain the parameter δ, nonlinear system identification is often used on the basis of experimental data collected from the actuator. Additionally, the size of deadzone characterized by the parameter δ may vary with respect to time. Thus, this parameter may need to be updated with experimental data from time to time so to form adaptation.

Because feedback control provides a means to the inversion of the nonlinearity, it is shown that through simulation studies it offers an efficient approach to compensate the

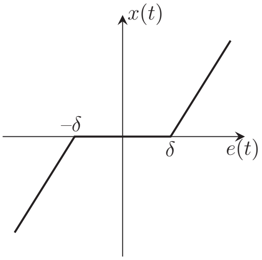


Figure 7.8 Deadzone nonlinearity.

effect of a deadzone when a secondary control loop is introduced for the actuator. This is particularly effective when the size of the zone is varying.

In order to highlight the importance of cascade feedback control, the first example is to illustrate the effect of a deadzone on closed-loop control performance without cascade control, and the second example is to show the performance improvement when the cascade control is introduced.

Example 7.4 *The actuator for a physical system is described by the transfer function*

$$G_s(s) = \frac{0.5}{s+15},$$

which is the secondary plant. The primary plant is described by the transfer function:

$$G_p(s) = \frac{0.8}{(0.1s+1)(s+0.1)}. \tag{7.18}$$

There is a deadzone associated with the actuator.

Design a PI controller to control the primary system while neglecting the actuator dynamics, and illustrate the performance deterioration due to the existing deadzone. The desired closed-loop performance is specified with natural frequency $w_n = 1$ *and damping coefficient* $\xi = 0.707$.

Solution. *The transfer function* $G_p(s)$ *has a dominant pole at* -0.1 *and small time constant* 0.1. *By neglecting this small time constant and taking into consideration the steady-state gain from the actuator, which is* $\frac{0.5}{15}$, *we obtain the approximate model for the PI controller design as*

$$G(s) = \frac{0.5}{15}\frac{0.8}{s+0.1} = \frac{b}{s+a}.$$

With $a = 0.1$, $b = 0.0267$, $w_n = 1$, *and* $\xi = 0.707$, *we calculate the PI controller parameters as*

$$K_c = \frac{2\xi w_n - a}{b} = 49.275; \quad \tau_I = \frac{2\xi w_n - a}{w_n^2} = 1.314.$$

The closed-loop control system performance is simulated with the sampling interval $\Delta t = 0.001$ *(s) and a unit step reference signal entering the system at t* $= 0$. *Figures 7.9(a) and (b) compare the closed-loop responses with and without the deadzone, where the sizes of the zones are* $\delta = 20$ *and* $\delta = 40$, *respectively. From these figures, it is interesting to note that with the PI controller there is no steady-state error in the output response and the control signal automatically finds a new steady-state value to compensate for the effect of the deadzone. However, the transient closed-loop control performance deteriorates with the introduction of the deadzone. Additionally, the larger the size of the deadzone* (δ), *the worse the transient closed-loop control performance is.*

Now, we assume that the actuator's output $x_1(t)$ is directly measured, which can be used to construct a cascade control system.

Example 7.5 *We continue from Example 7.4. Instead of neglecting the actuator dynamics, we use a PI controller to control the actuator and a PI controller for the primary plant.*

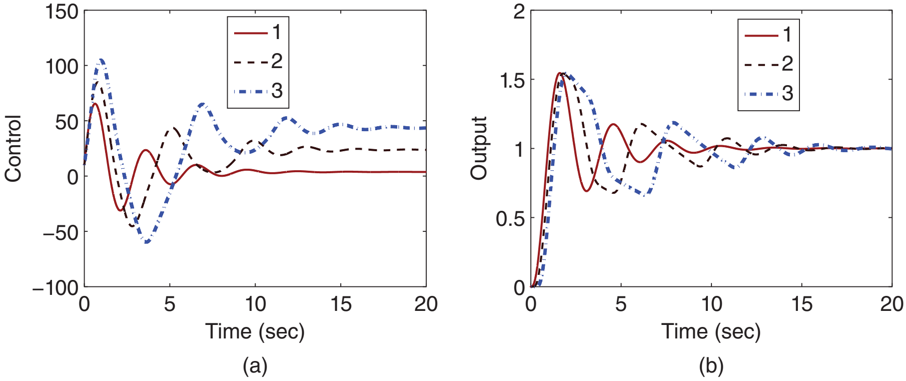


Figure 7.9 Closed-loop control response by neglecting the actuator dynamics (Example 7.4). (a) Control signal. (b) Output. Key: line (1) response without the deadzone; line (2) response with the deadzone ($\delta = 20$); line (3) response with the deadzone ($\delta = 40$).

To be consistent with Example 7.4, the natural frequency of the primary control loop is selected as 1 and $\xi = 0.707$, which is identical to the performance specification in the previous example. However, we will vary the natural frequency of the secondary control loop to illustrate the effect of cascade control on the closed-loop performance in the presence of a deadzone.

Solution.

Case A. *In the first case, we select the natural frequency for the secondary control system as $w_{\text{ns}} = 20$, which is 20 times that used for the primary control system. With this selection, the PI controller parameters are*

$$K_{\text{cs}} = \frac{2 \times 0.707 \times 20 - 15}{0.5} = 26.56;\ \tau_{\text{Is}} = \frac{2 \times 0.707 \times 20 - 15}{400} = 0.0332.$$

In the design of primary controller, the inner-loop dynamics are neglected. Therefore, the PI controller is designed using the transfer function (7.18) for the primary plant, leading to

$$K_{\text{cp}} = \frac{2 \times 0.707 \times 1 - 0.1}{0.8} = 1.6425;\ \tau_{\text{Ip}} = \frac{2 \times 0.707 \times 1 - 0.1}{1} = 1.314.$$

The cascade control system is configured and simulated using the Simulink program, as illustrated in Figure 7.10, where the PI controller with the velocity form from Tutorial 4.1 is used in the Simulink simulation. Note that in the inner-loop controller implementation, the proportional controller K_c is implemented on the feedback error to give an extra boost to the secondary closed-loop bandwidth. This implementation is used for all the cascade simulations. For simplicity, both input disturbance and measurement noise are assumed zero. With this cascade control structure, closed-loop simulation studies are performed with a deadzone where $\delta = 20$ and $\delta = 40$, respectively. Figure 7.11(a) shows the inner-loop control signal response in the presence of a deadzone nonlinearity. It is seen that the control signal converges to different values in steady state to compensate for the effect of the nonlinearity and it exhibits discontinuity. As a result of the cascade control, the effect of the deadzone is eliminated, as shown in the output response (see Figure 7.11(b)).

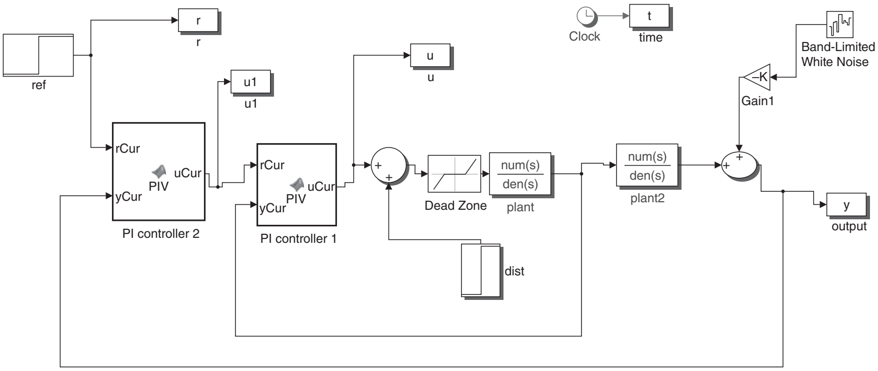


Figure 7.10 Simulink simulation program for the cascade control system with a deadzone nonlinearity in the actuator (Example 7.5).

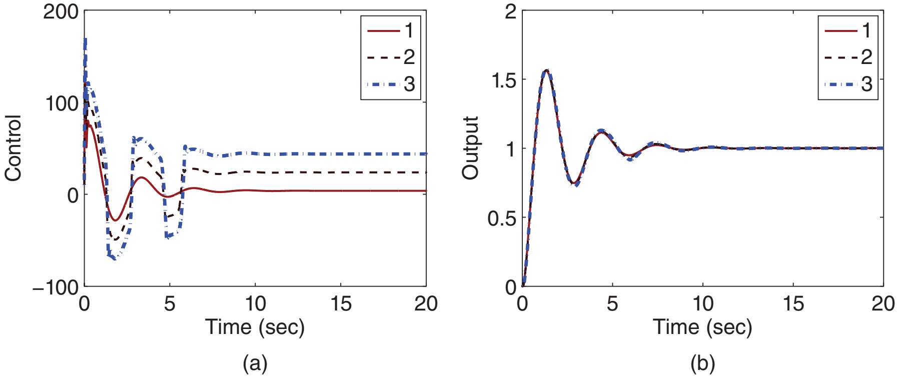


Figure 7.11 Closed-loop control response using cascade control (Example 7.5, $w_{\text{ns}} = 20$, $w_{\text{np}} = 1$). (a) Control signal. (b) Output. Key: line (1) response without a deadzone; line (2) response with a deadzone ($\delta = 20$); line (3) response with a deadzone ($\delta = 40$).

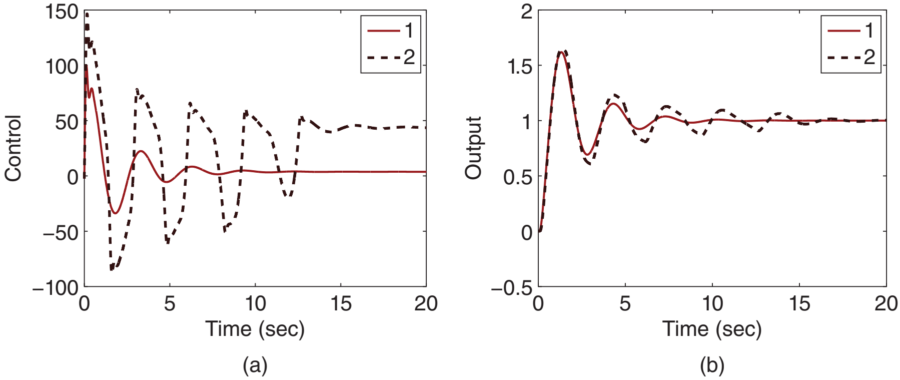


Figure 7.12 Closed-loop control response using cascade control (Example 7.5, $w_{\text{ns}} = 10$, $w_{\text{np}} = 1$). (a) Control signal. (b) Output. Key: line (1) response without a deadzone; line (2) response with a deadzone ($\delta = 40$).

Case B. *In the second case, we reduce the natural frequency for the inner-loop control system from $w_{\text{ns}} = 20$ to $w_{\text{ns}} = 10$. Without a deadzone, this reduction does not result in a significant change in the closed-loop control performance, as shown in Figures 7.12(a) and (b). However, when the deadzone nonlinearity is introduced and in the worst case where the size of the deadzone is $\delta = 40$, this smaller closed-loop bandwidth for the inner-loop system is no longer adequate to compensate for the effect of the deadzone nonlinearity, and as a result, the closed-loop output becomes oscillatory, as illustrated in Figure 7.12(b).*

7.4.2 Cascade Control for Actuators with Quantization Errors

Actuators with quantization errors are commonly encountered in control applications, particularly when using low cost actuation devices. Analyzing quantization errors was a classical topic (see Slaughter (1964), Miller et al. (1988)). Figure 7.13 illustrates quantization of the input signal $e(t)$ to produce the output signal $x(t)$.

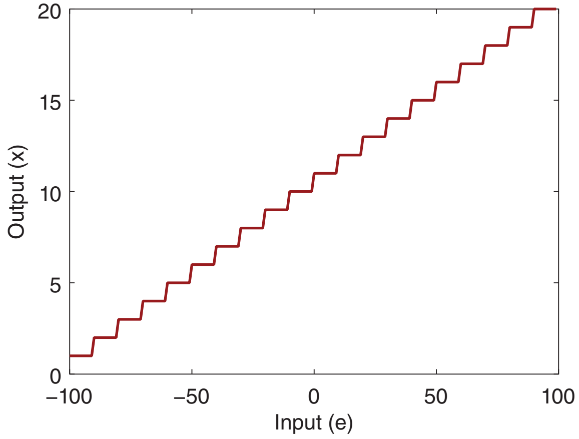


Figure 7.13 Illustration of a quantization of signal $e(t)$ with quantization interval $q = 1$.

In this section, we will show how the quantization errors affect closed-loop control performance and how a cascade control system will reduce the effects and improve the closed-loop performance.

Example 7.6 *Consider the system used in the original paper (Slaughter (1964)) for study of quantization error, which has the following continuous time transfer function:*

$$G(s) = \frac{4500}{s(s+10)(s+20)}. \tag{7.19}$$

Design a PI controller for this system with the natural frequency $w_\mathrm{n} = 1$ *and* $\xi = 0.707$. *Choosing sampling interval* $\Delta t = 0.01$ *(s), simulate the effect of quantization errors on the closed-loop performance.*

Solution. *Although the system is of third order, it can be approximated by the following first order transfer function by neglecting the two small time constants, which gives the design model for the PI controller.*

$$G(s) = \frac{4500}{s(s+10)(s+20)} = \frac{22.5}{s(0.1s+1)(0.05s+1)} \approx \frac{22.5}{s}.$$

With the choice of $w_\mathrm{n} = 1$ *and* $\xi = 0.707$, *the PI controller parameters are found as*

$$K_\mathrm{c} = \frac{2\xi w_\mathrm{n} - a}{b} = \frac{2 \times 0.707}{22.5} = 0.0628; \quad \tau_\mathrm{I} = \frac{2\xi w_\mathrm{n} - a}{w_\mathrm{n}^2} = 2 \times 0.707 = 1.414$$

where the natural frequency $w_\mathrm{n} = 1$, $\xi = 0.707$, $a = 0$, *and* $b = 22.5$.

Now, the closed-loop control system is simulated using a unit step reference signal entering the system at $t = 0$ *and a step input disturbance signal (amplitude*$= 0.5$*) entering the system at* $t = 20$ *(s). With* $\Delta t = 0.01$ *(s) and the quantization parameter* $q = 0.1$, *the quantizer function in Simulink is inserted into the Simulink simulator before the plant. Figure 7.14(a) shows the control signal calculated using the PI controller and Figure 7.14(b) shows the plant input signal after the quantization of the control signal with the addition of step disturbance. It is seen that the input signal only takes values in multiples of* $q = 0.1$. *Figure 7.14(c) shows the closed-loop output with the quantization. Due to the quantization process, there is a performance loss. A simulation study is*

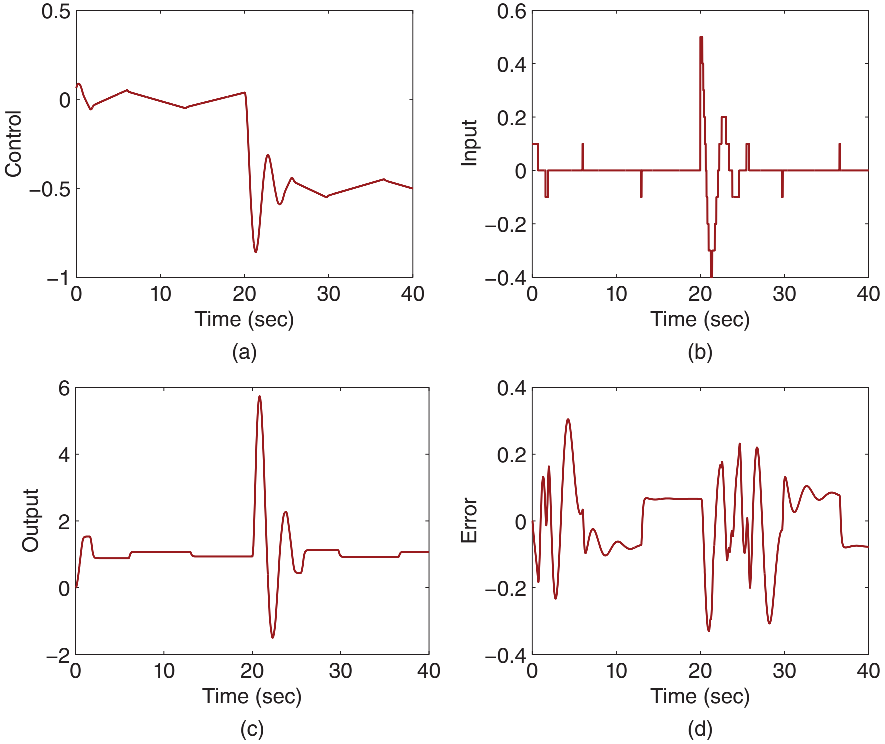


Figure 7.14 Closed-loop control response with quantization on the input signal (Example 7.6, $q = 0.1$). (a) Control signal. (b) Input signal. (c) Output. (d) Error.

performed for the control system without the quantizer. Figure 7.14(d) shows the error signal between the outputs without quantization and with quantization. It is seen that the magnitude of the error is about 0.3.

In the next example, we assume that the system in Example 7.6 can be decomposed into an actuator and a plant, and the output from the actuator can be measured to form a feedback control for the actuator. Then, we examine how the cascade control structure improves the closed-loop control performance with the quantization.

Example 7.7 *We assume that the actuator for the secondary plant has the transfer function*

$$G_s(s) = \frac{45}{s+10}$$

and the primary plant has the transfer function:

$$G_p(s) = \frac{100}{s(s+20)}. \tag{7.20}$$

The natural frequency for the primary control system is selected as $w_{np} = 1$ *and the secondary control system is selected as* $w_{ns} = 30$*. The damping coefficient is* $\xi = 0.707$ *for* *both loops. Show the effect of the cascade control structure on the closed-loop control performance with the same quantization as Example 7.6.*

Solution. *With the selection of the natural frequency* $w_{\mathrm{ns}} = 30$ *for the secondary control system, the PI controller parameters are calculated as*

$$K_{\mathrm{cs}} = \frac{2 \times 0.707 \times 30 - 10}{45} = 0.7204; \tau_{\mathrm{Is}} = \frac{2 \times 0.707 \times 30 - 10}{900} = 0.0360.$$

The inner-loop dynamics are neglected in the primary controller design; however, there is an approximation introduced to obtain the integrator model with gain equal to 5 *(see (7.20)). We calculate the PI controller parameters for the outer-loop system, with* $w_{\mathrm{np}} = 1$ *and* $\xi = 0.707$*, as*

$$K_{\mathrm{cp}} = \frac{2 \times 0.707}{5} = 0.2828; \tau_{\mathrm{Ip}} = 2 \times 0.707 = 1.414.$$

The cascade closed-loop control system is simulated with the same conditions as in Example 7.6 where a step reference signal enters the system at $t = 0$ *and a step input disturbance with amplitude* 0.5 *enters the system at* $t = 20$ *(s). The Simulink Quantizer function is inserted into the simulator before the secondary plant. Figure 7.15 shows the cascade control structure used in the Simulink simulation.*

Figure 7.16(a) shows the control signal generated by the secondary PI controller, and Figure 7.16(b) shows the quantized control signal with the added input disturbance signal, which becomes the input signal to the secondary plant. Figure 7.16(c) shows the plant output. By comparing the control signal to the input signal, it is seen that the cascade control system has taken quantization into consideration when calculating the controller output. As a result, the performance degradation is minimal, as illustrated by the error signal between the outputs of the cascade control systems without quantization and with quantization, as shown in Figure 7.16(d).

Comparing the cascade control system with that given by Example 7.6 there are two comments. The first is related to the disturbance rejection by comparing the output from Figure 7.14(c) with that from Figure 7.16(c). It is seen that without the cascade control, the effect of the disturbance is tremendous, which has a large magnitude of about 6 *and it takes about* 5 *s for the control system to reject the disturbance (see Figure 7.14(c)). In contrast, the effect of the disturbance in the cascade control is almost invisible in the output response [see Figure 7.16(c)]. The second comment is related to the performance deterioration due to the quantization by comparing Figure 7.14(d) with Figure 7.16(d). It is seen that without the cascade control, there is a large performance decay due to the quantization, where the error signal fluctuated with magnitude of* 0.3 *[see Figure 7.14(d)]. In contrast, the performance decay due to the quantization is reduced significantly, where the error signal fluctuated with magnitude of* 0.005 *[see Figure 7.16(d)]. In other words, without the cascade control, the error due to quantization alone is increased* 60 *times.*

7.4.3 Cascade Control for Actuators with Backlash Nonlinearity

A backlash nonlinearity is commonly encountered in actuation mechanisms. This type of nonlinearity has some shared characteristics with the deadzone nonlinearity, such as the existence of a deadband in the actuation. Assuming a deadband Δ for the backlash

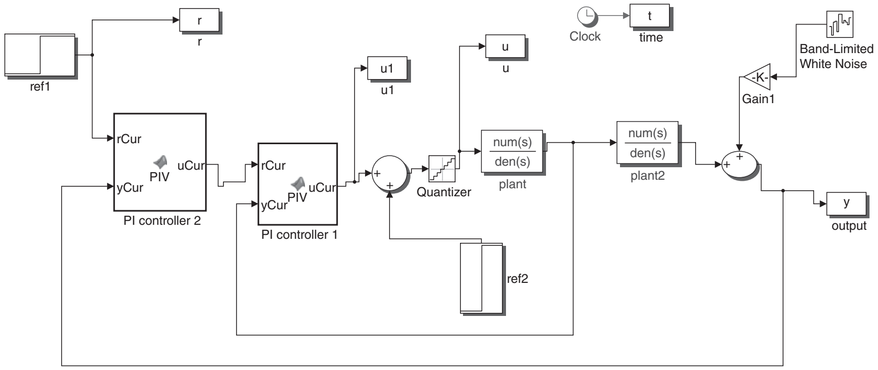


Figure 7.15 Simulink simulation program for cascade control with actuator quantization errors.

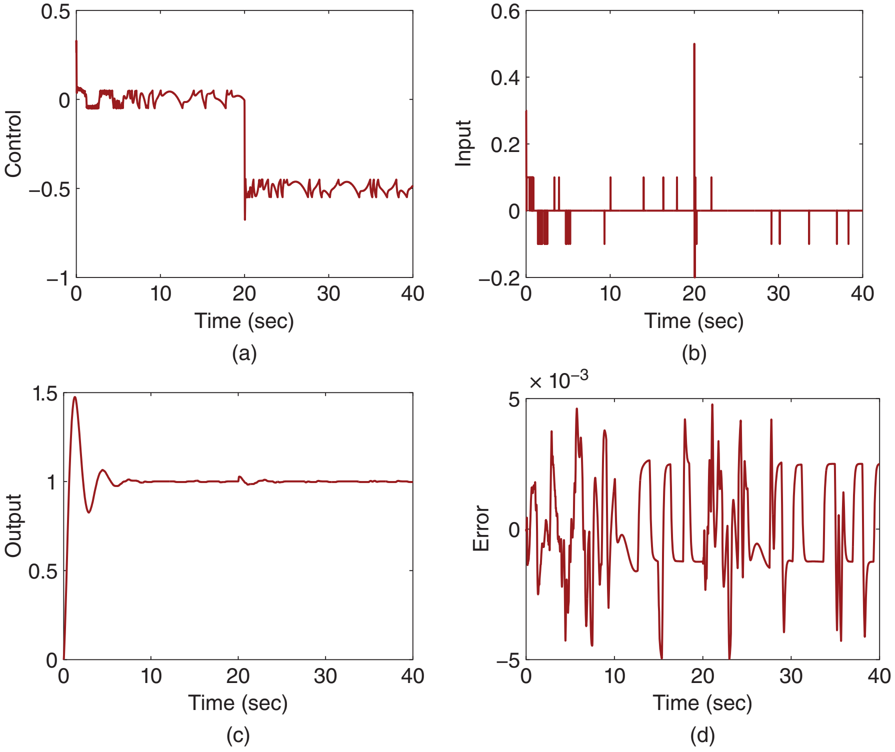


Figure 7.16 Cascade closed-loop control response with quantization on input signal (Example 7.7, $q = 0.1$). (a) Control signal. (b) Input signal. (c) Output. (d) Error.

nonlinearity, with the input signal to the backlash as $e(t)$ and output as $x(t)$, Figure 7.17 illustrates a backlash nonlinearity with gain ($k = 1$) and a deadzone ($\Delta = 40$). As illustrated in the figure, following the path with the arrows pointed upwards, the output signal is described by

$$x(t) = k\left(e(t) - \frac{\Delta}{2}\right)$$

and following the path with the arrows pointed downwards, the output signal is described by

$$x(t) = k\left(e(t) + \frac{\Delta}{2}\right).$$

The output signal $x(t)$ can switch between these two paths.

This type of nonlinearity causes sustained oscillations and affects the closed-loop control performance. The following example illustrates its effect on the closed-loop stability and performance.

Example 7.8 *In this example, the secondary system corresponding to the actuator dynamics is described by the following transfer function:*

$$G_s(s) = \frac{0.5}{s + 15}. \tag{7.21}$$

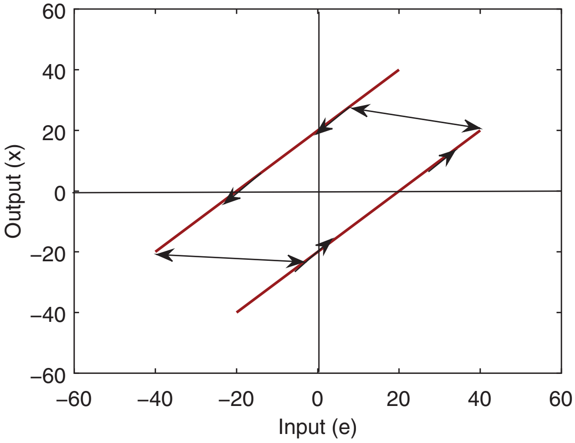


Figure 7.17 Illustration of a backlash nonlinearity with $k = 1$ and $\Delta = 40$.

The primary system is an integrator with a small time delay described by the transfer function:

$$G_{\mathrm{p}}(s) = \frac{0.01\mathrm{e}^{-0.3}}{s}. \tag{7.22}$$

The actuator has a backlash nonlinearity with deadband width $\Delta = 60$. *Design a PI controller with damping coefficient* $\xi = 0.707$ *and* $w_{\mathrm{n}} = 0.1$ *and* 0.3 *respectively. Use Simulink simulation to show the effect of the backlash nonlinearity on the closed-loop stability and performance. The simulation will be performed using sampling interval* $\Delta t = 0.01$ *(s) and Simulink backlash block function. The reference signal is a unit step.*

Solution. *When the actuator dynamics is neglected, its steady-state value is still considered in the design of a PI controller. For this purpose, the approximate model used for the design of a PI controller becomes:*

$$G(s) \approx \frac{3.333 \times 10^{-4}}{s}$$

where the time delay is also neglected in the design.

Case A *(*$w_{\mathrm{n}} = 0.1$*). With* $w_{\mathrm{n}} = 0.1$ *and* $b = 3.333 \times 10^{-4}$*, the PI controller parameters become:*

$$K_{\mathrm{c}} = \frac{2 \times 0.707 w_{\mathrm{n}}}{b} = 424.2; \tau_{\mathrm{I}} = \frac{2 \times 0.707 w_{\mathrm{n}}}{w_{\mathrm{n}}^2} = 14.14.$$

Figures 7.18(a)–(c) compare the closed-loop performance of the control systems with and without the backlash nonlinearity. It is seen that with the backlash nonlinearity, the closed-loop control signal [Figure 7.18(a)] and output signal [Figure 7.18(c)] become oscillatory. After the backlash, the actual input signal to actuator is a modulated control signal [see Figure 7.18(b)]. It is also interesting to note that the oscillation due to backlash is sustained, which is seen from the error between the two output signals illustrated in Figure 7.18(d). Here the error amplitude is 0.095 *and the period is* 36.4*.*

Case B *(*$w_{\mathrm{n}} = 0.3$*). Now, we increase the desired natural frequency* w_{n} *for the PI controller design. With* $w_{\mathrm{n}} = 0.3$ *and* $b = 3.333 \times 10^{-4}$*, the PI controller parameters are calculated as*

$$K_{\mathrm{c}} = 1272.6; \tau_I = 4.7133.$$

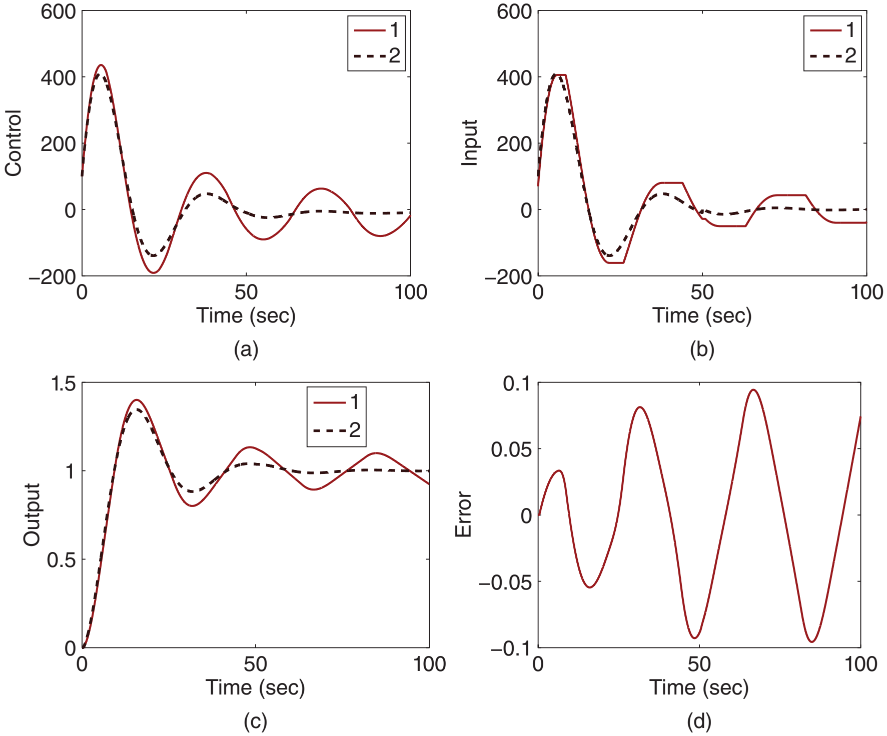


Figure 7.18 The effect of backlash on closed-loop performance (Example 7.8, $w_n = 0.1$). (a) Control signal. (b) Input signal. (c) Output. (d) Error. Key: line (1) control system with backlash; line (2) control system without backlash.

For this particular example, when compared with case A, K_c is three times larger and τ_I is three times smaller.

A closed-loop simulation is performed with and without the backlash nonlinearity. Figures 7.19(a)–(c) compare the closed-loop control results. It is seen that with the larger desired natural frequency, essentially a faster closed-loop response speed, sustained oscillation due to the backlash nonlinearity still exists; however, its amplitude and period have changed. By examining the error signal between the two outputs (see Figure 7.19(d)), we can find that the amplitude of the error is 0.042 *and the period of oscillation is* 11.44*. Therefore, by increasing the bandwidth of the closed-loop control system, the amplitude of the error caused by the backlash is reduced; however the frequency of the oscillation is increased.*

It is worthwhile noting that the bandwidth of the closed-loop system (w_n) in Example 7.8 is limited by the unmodeled dynamics in the system such as the neglected actuator dynamics and the time delay. One can verify that if w_n is increased to 1 for this particular system, the closed-loop system has a sustained oscillation even without the backlash, which is caused by the unmodeled dynamics in the system. When w_n exceeds 1, the closed-loop system becomes unstable even without the backlash nonlinearity.

Example 7.8 shows how the bandwidth of the closed-loop feedback control system affects the closed-loop performance in the presence of a backlash nonlinearity. The

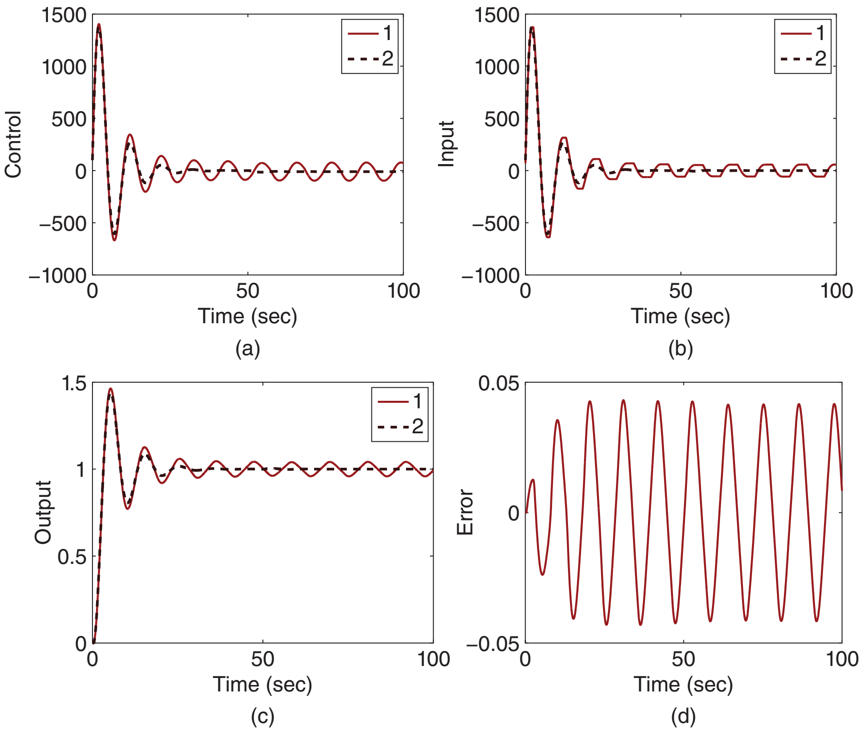


Figure 7.19 The effect of backlash on closed-loop performance (Example 7.8, $w_n = 0.3$). (a) Control signal. (b) Input signal. (c) Output. (d) Error. Key: line (1) control system with backlash; line (2) control system without backlash.

larger the closed-loop bandwidth is, the smaller the magnitude and the higher frequency of the periodic oscillation will be. The next example will illustrate that when using cascade control, a secondary controller is used to control the actuator with a much larger natural frequency, leading to a significant reduction in the amplitude of the oscillation and an increase of the oscillation frequency.

Example 7.9 *We continue to use the system described by the transfer functions with (7.21) and (7.22), together with the backlash nonlinearity with deadband* $\Delta = 60$*. Design a cascade control system with the bandwidth* $w_{ns} = 20$ *for the secondary system and the bandwidth* $w_{np} = 0.1$ *and* 0.3 *for the primary system, respectively. The damping coefficient is* $\xi = 0.707$ *for both secondary and primary control systems. Simulate the cascade closed-loop control system with the backlash nonlinearity, where the sampling interval* $\Delta t = 0.01$.

Solution. *From the transfer function model (7.21), we have* $b = 0.5$ *and* $a = 15$*. With the desired natural frequency* $w_{ns} = 20$*, the PI controller for the actuator has the following proportional gain and integral time constant,*

$$K_{cs} = \frac{2\xi w_{ns} - a}{b} = 26.56; \quad \tau_{Is} = \frac{2\xi w_{ns} - a}{w_{ns}^2} = 0.0332.$$

Case A *($w_{\rm np} = 0.1$). For the primary plant, with $w_{\rm np} = 0.1$, $a = 0$, and $b = 0.01$, the PI controller parameters are calculated as*

$$K_{\rm cp} = \frac{2\xi w_{\rm np}}{b} = 14.14; \quad \tau_{\rm Ip} = \frac{2\xi}{w_{\rm np}} = 14.14,$$

where the time-delay is neglected in the design and the steady-state gain of the inner-loop system is taken to be one.

The Simulink backlash function is inserted into the simulator before the secondary plant. Figure 7.20 shows the cascade control structure used in the Simulink simulation.

Now, with the reference signal being a unit step, sampling interval $\Delta t = 0.01$, and actuator backlash with deadband $\Delta = 60$, the cascade control system is simulated with the Simulink program illustrated in Figure 7.20. Figures 7.21(a)–(c) compare the cascade control system with and without the backlash nonlinearity. It appears that, in the presence of the backlash nonlinearity, both the control signal [see Figure 7.21(a)] and the input signal to the actuator [see Figure 7.21(b)] are noisy. However, the output of the primary plant is very close to that simulated without the backlash nonlinearity, and it is hard to distinguish them by visual inspection [see Figure 7.21(c)]. The error between the two output responses is shown in Figure 7.21(d). At the steady state, the amplitude of the error is 0.00015*, which is* 633 *times smaller than the control system without cascade control. In fact, the cascade control system does not eliminate the periodic oscillation caused by the backlash nonlinearity. However, because of the much wider bandwidth used in the secondary control system, the period of the sustained oscillation has been reduced to* 0.3*, as shown in Figure 7.22, by the segments of the control signal and the input signal. The control signal is still a sinusoidal signal and the input signal after the backlash is a modulated control signal.*

Case B *($w_{\rm np} = 0.3$). As an exercise, one can verify that when the natural frequency for the primary controller is increased to* 0.3*, the amplitude of the error between the outputs with and without the backlash nonlinearity is reduced to* 0.0001288 *and the period of the sustained oscillation is reduced to* 0.28*. Thus, it is less significant in error reduction when increasing the outer-loop bandwidth.*

One can also verify that when the bandwidth $w_{\rm ns}$ of the secondary controller is increased from 20 *to* 30 *while maintaining $w_{\rm np} = 0.1$, the amplitude of the error between the outputs with and without the backlash is reduced to* 0.00007898 *which is almost half of that presented in case A and the period of oscillation is reduced to* 0.16*.*

7.4.4 Food for Thought

1. In the cascade control structure, the nonlinearities are assumed in the actuator. Do you agree that one of the key reasons for the successful cascade nonlinear compensation is due to a much wider bandwidth used for the inner-loop control system?
2. In the three classes of nonlinearities considered, from the control signals to the primary plant, do you observe the nonlinear inversion compensation? Is this nonlinear inversion predominantly caused by high gain feedback control?
3. Would you say that the backlash nonlinearity is harder to deal with because of the periodic oscillation?
4. Which nonlinearity is easiest to compensate?

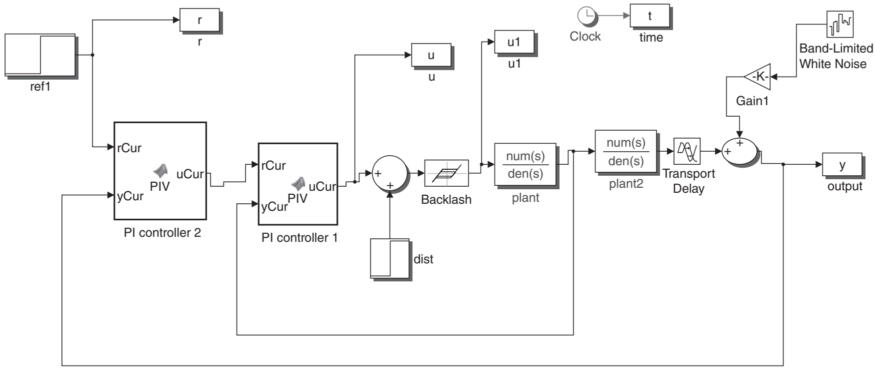


Figure 7.20 Simulink simulation program for cascade control with a backlash nonlinearity in the actuator.

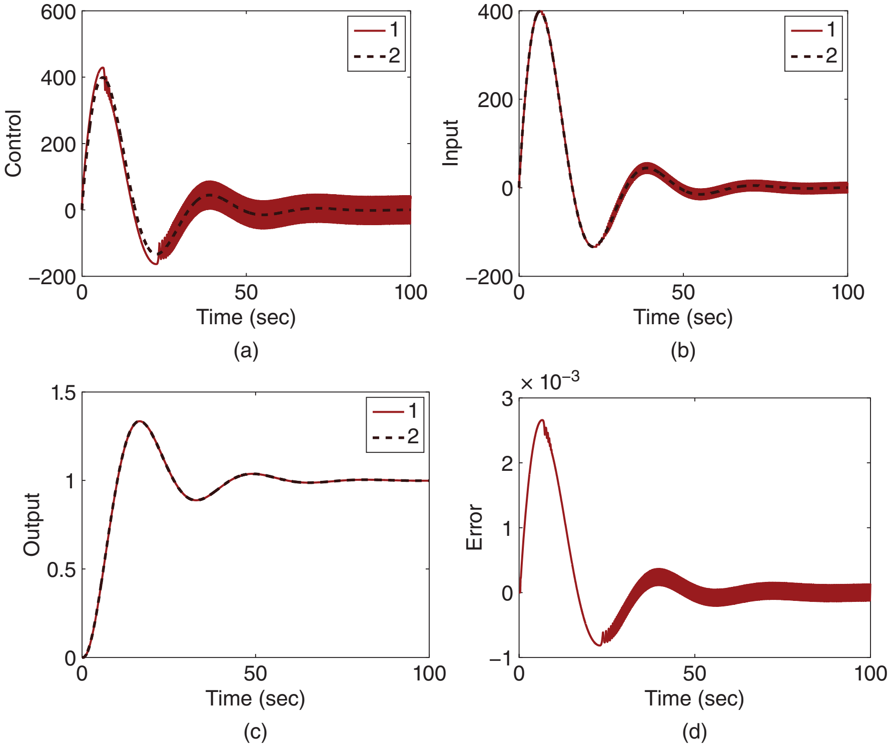


Figure 7.21 The effect of backlash on cascaded closed-loop performance (Example 7.9, w_{ns} = 20 and w_{np} = 0.1). (a) Control signal. (b) Input signal. (c) Output. (d) Error. Key: line (1) control system with backlash; line (2) control system without backlash.

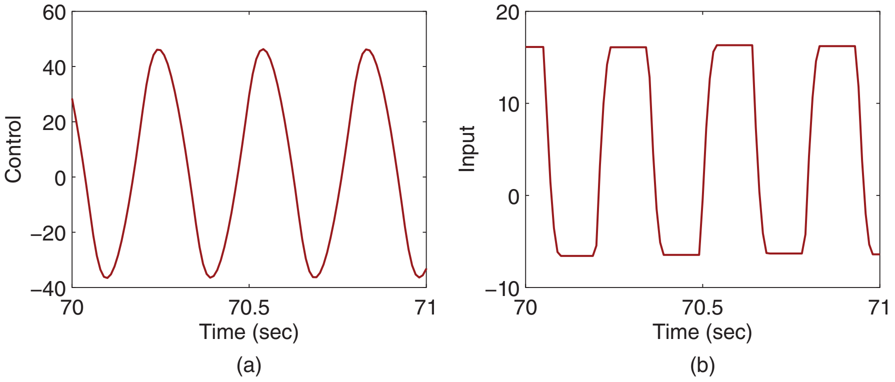


Figure 7.22 Segment of data to illustrate the effect of backlash on cascaded closed-loop performance (Example 7.9). (a) Control signal. (b) Input signal.

7.5 Summary

Cascade control systems are widely used in control engineering applications. The main reasons behind the utilization of cascade control systems include the simplification of control system design because a complex system is decomposed into several subsystems and each of them can be described by a first order or a second order model, superior performance in disturbance rejection and compensations of actuator nonlinearity.

The other important aspects of this chapter is summarized as follows.

- The design of a cascade control system starts at the control system design for the inner-loop system. The inner-loop closed-loop system dynamics are commonly neglected in the design of outer-loop control system. However, if a proportional controller is used for the inner-loop control, then the steady-state gain of the inner-loop system needs to be considered in the outer-loop controller design.
- The closed-loop performance of the cascade control system is dependent on the relative bandwidth between the inner-loop and outer-loop systems. For the inner-loop controller, we will choose the desired closed-loop poles corresponding to much faster dynamic response speed than those used in the outer-loop control system.
- The effect of actuator nonlinearities can be compensated using a PI controller designed for the actuator.
- The input disturbance occurred at the inner-loop system is much more effectively compensated in the cascade control structure.

7.6 Further Reading

1. Qualitative analysis of cascade control was introduced in Franks and Worley (1956). The improved disturbance rejection using cascade control was discussed in Krishnaswamy et al. (1990), Lee et al. (1998), Lee et al. (2002).
2. Quantization error analysis was introduced in Slaughter (1964) and Miller et al. (1988). Nonlinear oscillations from the backlash nonlinearity was analyzed in Moradi and Salarieh (2012). Adaptive control of systems with backlash was proposed in Tao and Kokotovic (1993). Adaptive nonlinear control of system with unknown deadzone was proposed in Zhou and Shen (2007). Nonlinear control of gear transmission servo systems with backlash nonlinearity was proposed in Shi and Zuo (2015). Application examples include cascade control of hydraulic actuator (Cunha et al. (2002)).
3. Tuning of cascade PID control systems was proposed in Wang et al. (1995), Visioli and Piazzi (2006), Dittmar et al. (2012) and Alfaro et al. (2009).
4. PID control of systems with hysteresis was designed and analyzed in Jayawardhana et al. (2008).
5. Electrical drives and power converters typically use cascade PI control structures (Wang et al. (2015)).
6. Cascade control system with inner-loop sliding mode control and outer-loop PI control was proposed for induction motor control in Mishra et al. (2018) and power converter control in Saeed et al. (2018).

Problems

7.1 *An electro-mechanical system is controlled by a cascade PI controller, where the mathematical models for the secondary and primary systems are,*

$$G_s(s) = \frac{1}{s+1}; \quad G_p(s) = \frac{0.1}{s+0.1}$$

Choose the proportional controller gain K such that the secondary system has a closed-loop pole at -6 and design the primary PI controller by finding its proportional gain K_c and integral time constant τ_I. All desired closed-loop poles for the outer-loop system are chosen to be $-\lambda$ ($\lambda > 0$).

7.2 *Continue from Prob 7.1. Because of the neglected dynamics from the inner-loop control system, there is a range of λ for the closed-loop stability. Use Routh-Hurwitz stability criterion to determine the range of λ for the stability of the closed-loop system.*

7.3 *Continue from Prob 7.1. Suppose that there is a step input disturbance entering the system at the inner-loop plant with amplitude 2. Find the transfer function between the input disturbance and the output (primary variable), and show that the disturbance will be rejected without steady-state errors if a correct λ is selected.*

7.4 *A dynamic system is described by the following transfer functions that are used for the design of a cascade control system,*

$$G_s(s) = \frac{-s+10}{(s+10)(s+3)}; G_p(s) = \frac{0.1}{s(s+2)}$$

1. *Find the proportional controller K such that the secondary control system has a pair of identical real poles on the left half of the complex plane.*
2. *Design a PID controller with filter for the outer-loop system with all closed-loop poles positioned at -1.*
3. *Use Nyquist stability criterion to check if the cascade control system is stable.*

7.5 *In the majority of applications, we would like to have a PI controller for the secondary control system as well as for the primary control system. Design cascade control systems for the following cases.*

1. *The system transfer functions are*

$$G_s(s) = \frac{1}{(s+1)}; G_p(s) = \frac{0.1}{s+0.1}.$$

The closed-loop poles for the secondary system are all positioned at -6; and for the primary system are all positioned at -1. Find the transfer function between the input disturbance and the output. Determine the gain margin and phase margin for the cascade control system.

2. *The system transfer functions are*

$$G_s(s) = \frac{-5}{(s+30)(s+2)}; G_p(s) = \frac{0.1}{s(s+4)}.$$

The closed-loop poles for the inner-loop system are all positioned at -6*; and for the outer-loop system are all positioned at* -0.6*. Find the transfer function between the input disturbance and the output. Determine the gain margin and phase margin for the cascade control system.*

7.6 *A type of heat exchanger systems is to transfer heat from a hot fluid to a cooler fluid in which the temperature control of outlet fluid is of prime importance. Based on experimental data, the mathematical models of a heat exchanger system were obtained in Khare and Singh (2010). The process model for the heater exchanger is,*

$$G_p(s) = \frac{50e^{-ds}}{30s+1}$$

the sensor dynamics with the thermocouple is,

$$G_m(s) = \frac{0.16}{10s+1}$$

the actuator is the combination of a converter and valve, having the transfer function model:

$$G_s(s) = \frac{0.75 \times 0.13}{3s+1}$$

1. *Design a cascade control system for this heater exchanger with all the closed-loop poles of the secondary system positioned at* -1.5 *and the primary system at* -0.1*. Here, the secondary PI control system is to control the actuator based on the transfer function* $G_s(s)$ *to overcome the nonlinearity of the valve such as quantization errors. The primary PID control system is designed based on the process model* $G_p(s)$ *and the sensor model* $G_m(s)$*. In the design, the time delay* d *is neglected.*
2. *Determine the gain margin, phase margin and delay margin for the cascade control system using Nyquist diagram.*
3. *Simulate the cascade closed-loop PID control performance for disturbance rejection and reference following with time delay* $d = 2$*. The sampling interval* $\Delta t = 1$*. The implementation of inner-loop PI controller will put both proportional and integral control on the feedback error. However, to reduce overshoot in the reference response, the implementation of the primary controller will put both proportional and derivative control on the output only.*

8

PID Controller Design for Complex Systems

8.1 Introduction

The PID controller design methods discussed in the previous chapters are either model based approaches or rule based approaches. It is clear that when using a model based approach, a first order model yields a PI controller and a second order model yields a PID controller. In some applications, the first order and second order models are basically an approximation to the actual physical systems. In other applications, the underlying physical systems are complex and are of higher order. This chapter studies how to design PID controllers for higher order systems directly using frequency response data.

8.2 PI Controller Design via Gain and Phase Margins

This section presents PID controller design via specification of gain margin and phase margin.

8.2.1 PI Controller Design Using Gain Margin and Phase Margin Specifications

The starting point is to assume that the frequency response of a desired open-loop transfer function $L_\mathrm{d}(\mathrm{j}\omega)$ at a specific frequency point $\omega = \omega_1$ is available. It is also assumed that the frequency response of the system $G(\mathrm{j}\omega)$ is available at $\omega = \omega_1$.

At the frequency ω_1, the actual open-loop frequency response with a PI controller is

$$L_\mathrm{d}(\mathrm{j}\omega_1) = \frac{c_1\mathrm{j}\omega_1 + c_0}{\mathrm{j}\omega_1}G(\mathrm{j}\omega_1).$$

Letting the actual open-loop frequency response equal its desired counterpart leads to

$$\frac{c_1\mathrm{j}\omega_1 + c_0}{\mathrm{j}\omega_1}G(\mathrm{j}\omega_1) = L_\mathrm{d}(j\omega_1) \tag{8.1}$$

which is

$$c_0 + \mathrm{j}c_1\omega_1 = \frac{\mathrm{j}\omega_1 L_\mathrm{d}(\mathrm{j}\omega_1)}{G(\mathrm{j}\omega_1)} = \mathrm{Real}\left[\frac{\mathrm{j}\omega_1 L_\mathrm{d}(\mathrm{j}\omega_1)}{G(\mathrm{j}\omega_1)}\right] + \mathrm{jImag}\left[\frac{\mathrm{j}\omega_1 L_\mathrm{d}(\mathrm{j}\omega_1)}{G(\mathrm{j}\omega_1)}\right]. \tag{8.2}$$

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

Comparing the left-hand side with the right-hand side of (8.2) gives

$$c_1 = \frac{1}{\omega_1}\text{Imag}\left[\frac{j\omega_1 L_d(j\omega_1)}{G(j\omega_1)}\right] \tag{8.3}$$

$$c_0 = \text{Real}\left[\frac{j\omega_1 L_d(j\omega_1)}{G(j\omega_1)}\right]. \tag{8.4}$$

From c_0 and c_1, the PI controller parameters are calculated using the following relationships

$$K_c = c_1; \quad \tau_I = \frac{c_1}{c_0}.$$

One of the choices for the frequency response of a desired open- loop transfer function $L_d(j\omega)$ is to specify the gain margin for the PI control system. Say, if one wishes to have a gain margin of 2, then $L_d(j\omega_1) = -0.5$. However, the frequency ω_1 still needs to be determined, where ω_1 in this specification is the cross-over frequency for the desired closed- loop system. Had a proportional controller $C(s) = K_c$ been used in the design, ω_1 could be chosen as the cross-over frequency for the $G(j\omega)$. Following the same practice, because of the $\frac{\pi}{2}$ phase-lag introduced by the integrator in the controller, a reasonable practice is to select ω_1 in the vicinity of the frequency $\omega_{\frac{\pi}{2}}$ where $G(j\omega)$ for the first time crosses the imaginary axis. Note that $\omega_1 \neq \omega_{\frac{\pi}{2}}$ because at $\omega_{\frac{\pi}{2}}$, $\text{Real}(G(j\omega)) = 0$, consequently resulting in $c_0 = 0$. A practice is to set $\omega_1 = 1.01\omega_{\frac{\pi}{2}}$.

Similar to the specification of the desired gain margin, the phase margin can also be used to specify the desired open-loop frequency response $L_d(j\omega_1)$. It is known that $|L_d(j\omega)| = 1$ at the frequency (say ω_1) that defines the phase margin. Hence, by denoting the phase margin as θ, then

$$L_d(j\omega_1) = -\cos\theta - j\sin\theta.$$

8.2.2 Design Examples

The following example demonstrates the performance of the PI controller when using gain margin and phase margin specifications.

Example 8.1 *Consider a system with the following transfer function,*

$$G(s) = \frac{1.2(-s+1)}{(2s+1)^2(s+1)}. \tag{8.5}$$

Choose the frequency $\omega_1 = \omega_{\frac{\pi}{2}} + 0.01\omega_{\frac{\pi}{2}}$.

1. *Selecting gain margin* $k_g = 2$ *and* 4 *respectively, design a PI controller with these two gain margin specifications and evaluate the closed-loop performance.*
2. *Selecting phase margin* $\theta = \frac{\pi}{3}$, *design a PI controller with the phase margin and simulate the closed-loop response.*

Solution. *We will first determine the frequency* ω_1 *by presenting the real part of* $G(j\omega)$ *as shown in Figure 8.1(a). By inspection of Figure 8.1(a), the parameter* ω_1 *is found to be* $0.28 \times 1.1 = 0.2828$. *With this frequency parameter,* $G(j\omega_1) = -0.009 - j0.9091$.

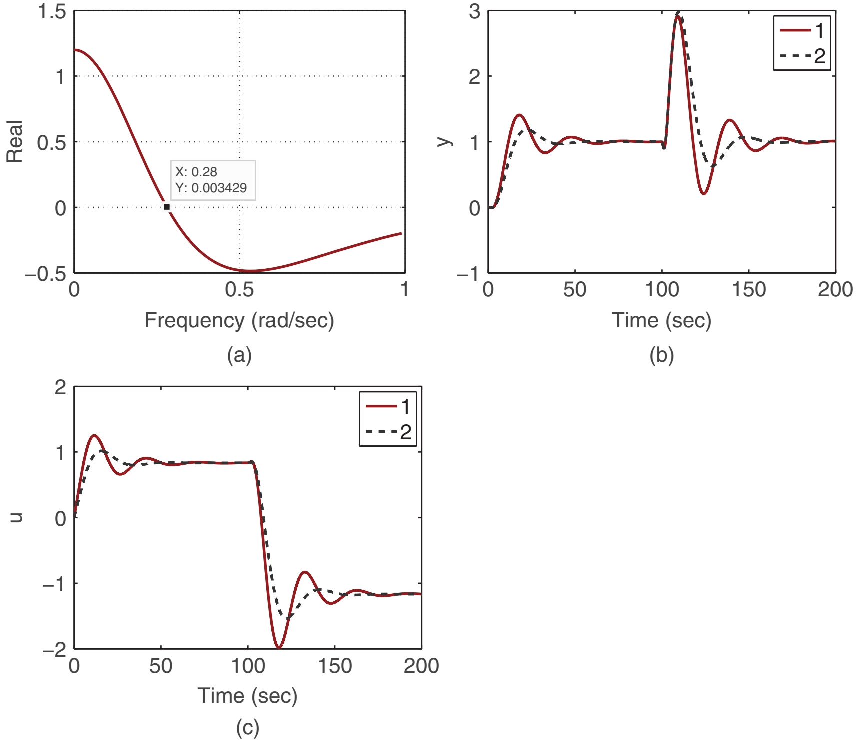


Figure 8.1 Illustration of the design process and closed-loop responses (Example 8.1) when using the gain margin. (a) Determining ω_1. (b) Output signal. (c) Control signal. Key: line (1) with a specified gain margin of 2; line (2) with a gain margin of 3.

Design using the gain margin. *The specification of a gain margin of 2 ($L_d(j\omega_1) = -0.5$) gives*

$$\frac{j\omega_1 L_d(j\omega_1)}{G(j\omega_1)} = 0.1555 + j0.0015$$

which leads to $c_0 = 0.1555$ and $c_1 = \frac{0.0015}{0.2828} = 0.0054$. Hence, the proportional controller gain $K_c = 0.0054$ and the integral time constant $\tau_I = 0.035$.

Repeating the above computation with the specification of a gain margin of 3 ($L_d(j\omega_1) = -1/3$), it can verified that

$$K_c = 0.0036; \tau_I = 0.035.$$

Here K_c has been reduced while τ_I stays the same.

Closed-loop simulation studies have been performed for both controllers where a unit step reference signal is introduced at time $t = 0$ and an input step disturbance with magnitude 2 is injected at $t = 100$ (s). Figures 8.1(b)–(c) show the closed-loop output responses and the control signal responses. It is seen that with a larger gain margin, the closed-loop system response is slower and less oscillatory.

Design using phase margin. *By choosing the desired phase margin $\theta = \frac{\pi}{3}$, the desired open-loop frequency response at ω_1 is*

$$L_\mathrm{d}(\mathrm{j}\omega) = -\cos\frac{\pi}{3} - \mathrm{j}\sin\frac{\pi}{3} = -0.5 - \mathrm{j}0.866.$$

Using this information together with the information of ω_1 and $G(\mathrm{j}\omega_1)$, we calculate

$$\frac{\mathrm{j}\omega_1 L_\mathrm{d}(\mathrm{j}\omega_1)}{G(\mathrm{j}\omega_1)} = 0.1529 + \mathrm{j}0.2709$$

yielding

$$c_0 = 0.1529; c_1 = \frac{0.2709}{0.2828} = 0.958.$$

The PI controller parameters are $K_\mathrm{c} = 0.958$ and $\tau_\mathrm{I} = \frac{c_1}{c_0} = 6.267$. Apparently, the design using phase margin specification gives a larger proportional gain and integral time constant. Closed-loop simulation studies have been performed for this PI control system for evaluation of reference following and input disturbance rejection. Figure 8.2(a) shows the output response and Figure 8.2(b) shows the control signal response. It is seen that the closed-loop response speed is faster than the closed-loop systems designed using the gain margin specification in the first part of this example.

The following example is to show how the PI controller is used to control a higher order complex system with time delay, which is very difficult if other design methods are used for this case. The closed-loop performance when using the specification of gain margin is left as an exercise (see Problem 8.1).

Example 8.2 *The transfer function of a complex system is given by*

$$G(s) = \frac{-3(-s^2 + s + 1)}{(10s + 1)(8s + 1)(6s + 1)(5s + 1)} \mathrm{e}^{-6s}. \tag{8.6}$$

Design a PI controller for this system by specifying the desired phase margin with $\theta = \frac{\pi}{3}$ and evaluate the closed- loop performance.

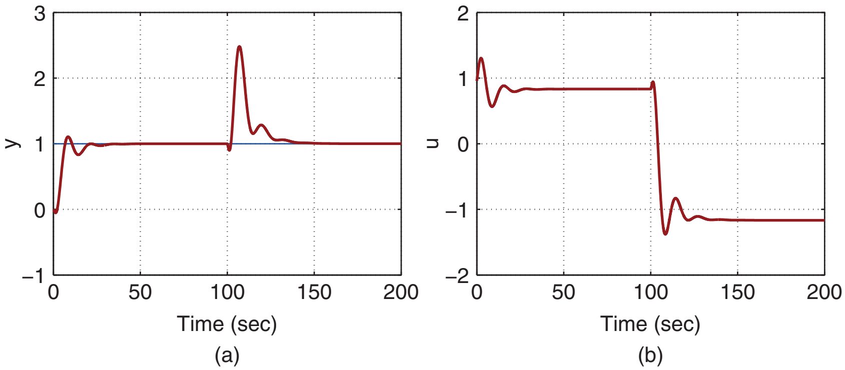


Figure 8.2 Closed-loop responses (Example 8.1) when using the phase margin. (a) Output signal. (b) Control signal.

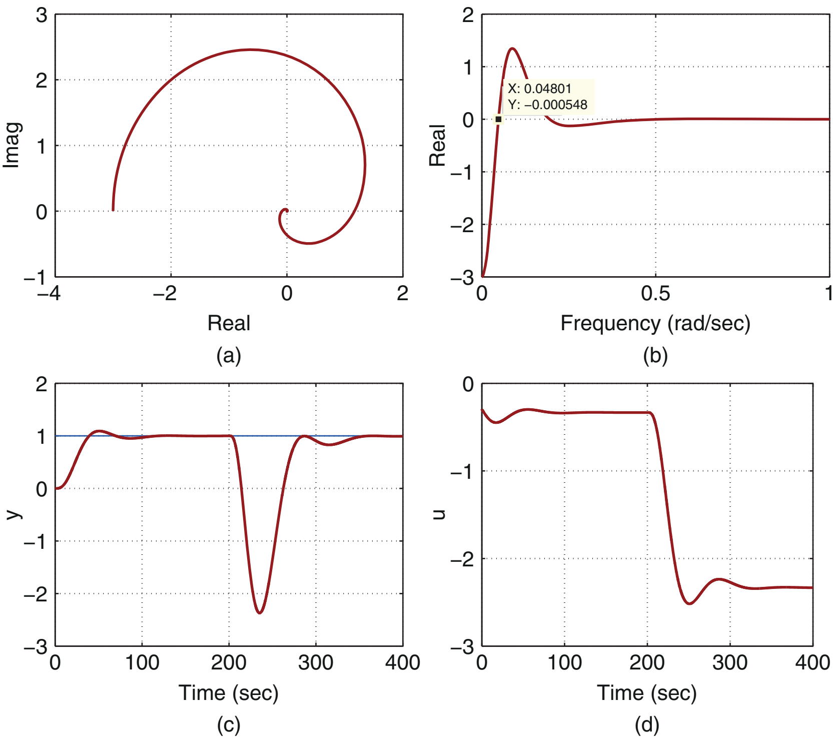


Figure 8.3 Illustration of the design process and closed-loop responses (Example 8.2) when using a phase margin. (a) $G(j\omega)$. (b) Determining ω_1. (c) Output. (d) Control.

Solution. *The frequency response* $G(j\omega)$ *is shown in Figure 8.3(a) and its real part is shown in Figure 8.3(b), from which the frequency* $\omega_1 = 0.048$ *is chosen, leading to* $G(j\omega_1) = -0.6442 + j2.2672$. *With the given desired phase margin* $\theta = \frac{\pi}{3}$, *the desired open-loop frequency response at* ω_1 *is specified as*

$$L_d(j\omega) = -\cos\frac{\pi}{3} - j\sin\frac{\pi}{3} = -0.5 - j0.866.$$

Then the following quantity is calculated:

$$\frac{j\omega_1 L_d(j\omega_1)}{G(j\omega_1)} = -0.0148 - j0.0143.$$

This gives $c_0 = -0.0148$ *and* $c_1 = -0.2955$. *Finally,* $K_c = -0.2955$ *and* $\tau_I = 20.0154$.

To evaluate the closed-loop performance of the PI control system, simulation studies are performed using unit step reference input and a step disturbance with magnitude of 2 entering the system at 200 (s). The proportional control is implemented on the output only. Figures 8.3(c) and (d) show the closed-loop control results. The results show that even though the process is higher order and complex, the PI controller produces stable and satisfactory results. However, the response to the input disturbance is quite slow.

It must be emphasized that the PI controller design using the specification of either gain margin or phase margin does not work for severely underdamped systems and unstable systems. This is because for these classes of systems, the desired open-loop frequency response $L_d(j\omega)$ must be calculated in a more sophisticated way to reflect the undesired process characteristics. Problem 8.1 is left for the verification of these statements.

8.2.3 Food for Thought

1. If we are to fit the frequency response of a higher order transfer function with a first order plus delay model or a second order plus delay model, how would you propose to estimate the time delay parameter d?
2. Would you say that the stability of the closed-loop system is not guaranteed when a PID controller is designed using an approximate model?
3. Is it correct that the closed-loop stability is checked against the original higher order system?
4. If the amplitude of the frequency response of the original system, $|G(j\omega)|$, asymptotically decays as ω increases, do you think that the PI controller design methods using the specification of gain margin or phase margin will guarantee closed-loop stability for the original system?
5. Will the amplitude of the frequency response of the original system containing a pair of underdamped modes asymptotically decay as ω increases?

8.3 PID Controller Design using Two Frequency Points

This section discusses an intuitive and simple approach to PID controller design from the perspective of curve fitting of the frequency response of the loop transfer function.

8.3.1 Finding the PID Controller Parameters

As we know from the PI controller design, the process frequency response information at one frequency ω_1 is sufficient to find the two parameters (K_c and τ_I). Because there are three parameters contained in a PID controller, naturally it requires the process frequency response information at two frequencies ω_1 and ω_2 ($\omega_1 \neq \omega_2$) to uniquely determine the three parameters (K_c, τ_I and τ_D).

The starting point is to assume that the desired open-loop frequency response $L_d(j\omega)$ is specified at $\omega = \omega_1$ and $\omega = \omega_2$, and the plant frequency response $G(j\omega)$ is also known at $\omega = \omega_1$ and $\omega = \omega_2$. Furthermore, we assume that $\omega_1 < \omega_2$. The specifications of the $L_d(j\omega)$, and the frequency points ω_1 and ω_2 will be discussed later.

For a PID control system, the actual open-loop frequency response at ω_1 is

$$L(j\omega_1) = \frac{c_2(j\omega_1)^2 + c_1 j\omega_1 + c_0}{j\omega_1} G(j\omega_1)$$

and at ω_2 is

$$L(j\omega_2) = \frac{c_2(j\omega_2)^2 + c_1 j\omega_2 + c_0}{j\omega_2} G(j\omega_2).$$

Thus, by letting

$$L(j\omega_1) = L_d(j\omega_1) \tag{8.7}$$

$$L(j\omega_2) = L_d(j\omega_2) \tag{8.8}$$

and comparing their real and imaginary components, the following linear equations hold:

$$c_1\omega_1 = \text{Imag}\,[X(j\omega_1)] \tag{8.9}$$

$$-c_2\omega_1^2 + c_0 = \text{Real}\,[X(j\omega_1)] \tag{8.10}$$

$$-c_2\omega_2^2 + c_0 = \text{Real}\,[X(j\omega_2)] \tag{8.11}$$

where for notational simplicity,

$$X(j\omega_1) = \frac{j\omega_1 L_d(j\omega_1)}{G(j\omega_1)} \tag{8.12}$$

$$X(j\omega_2) = \frac{j\omega_2 L_d(j\omega_2)}{G(j\omega_2)}. \tag{8.13}$$

From (8.9), the coefficient c_1 is calculated as

$$c_1 = \frac{\text{Imag}\,[X(j\omega_1)]}{\omega_1}. \tag{8.14}$$

From (8.10) and (8.11), the coefficients c_2 and c_0 are calculated by solving the two linear equations, giving

$$c_2 = \frac{\text{Real}\,[X(j\omega_2)] - \text{Real}\,[X(j\omega_1)]}{\omega_1^2 - \omega_2^2} \tag{8.15}$$

$$c_0 = c_2\omega_1^2 + \text{Real}\,[X(j\omega_1)]. \tag{8.16}$$

Finally, the PID controller parameters are given in relation to c_2, c_1, c_0 as

$$K_c = c_1; \quad \tau_I = \frac{c_1}{c_0}; \quad \tau_D = \frac{c_2}{c_1}.$$

Because there are two frequencies ω_1 and ω_2 used in the design, more thought is required in the selection of not only $L_d(j\omega)$ at ω_1 and ω_2, but also the frequencies ω_1 and ω_2 themselves. Naturally, one assumes that the desired gain margin and phase margin are good candidates for the selection of $L_d(j\omega)$. However, the challenge is to find the suitable values for the desired gain margin and phase margin together with ω_1 and ω_2.

The following example is used to demonstrate the difficulties in specifying $L_d(j\omega)$.

Example 8.3 *Consider PID controller design for the higher order system used in Example 8.2 [see the transfer function given by (8.6)]. Choosing $\omega_1 = 0.0485$, $\omega_2 = 0.108$ (cross-over frequency of $G(j\omega)$), and desired phase margin $\theta = \frac{\pi}{3}$, design a PID controller by varying the desired gain margin.*

Solution. *It can be easily verified that for any choice of desired gain margin less than $1/0.3 = 3.33$, the derivative gain τ_D is negative. As in practice, a derivative filter is often used in conjunction with the derivative gain τ_D, which has the form $\frac{\tau_D s}{0.1\tau_D s+1}$. Then with*

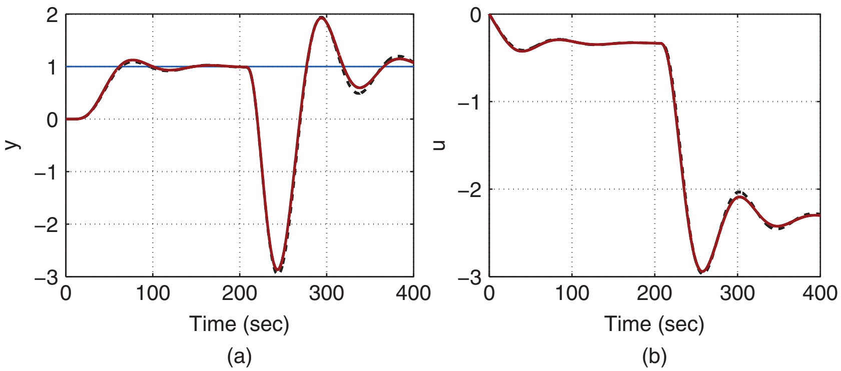


Figure 8.4 Comparison of closed-loop responses of PI and PID control systems (Example 8.3). (a) Output response. (b) Control signal. Key: PID control system (solid-line), PI control system (dashed-line).

the negative τ_D, the filter becomes unstable. This definitely renders the controller better off without the derivative term.

By increasing the desired gain margin to 5*, together with* $\omega_1 = 0.0485$*,* $\omega_2 = 0.108$*, and the desired phase margin* $\theta = \frac{\pi}{3}$*, the PID controller parameters are calculated as*

$$K_c = -0.2955; \tau_I = 18.8527; \tau_D = 1.3110.$$

In comparison to the PI controller designed in Example 8.2, the proportional gain is the same, however the integral time constant is reduced. Closed-loop system simulation is performed with proportional control implemented on the output only. The derivative filter with time constant $0.1\tau_D$ *is used in the implementation of derivative control. Figure 8.4 compares the closed-loop responses of the PID control system with those of the PI control system from Example 8.2. It is seen that the performance improvement using the PID controller is very small in comparison to the one with the PI controller. Again, the closed-loop response to input disturbance is oscillatory and slow.*

8.3.2 Desired Closed-loop Performance Specification using Two Frequency Points

It is apparent that the desired closed-loop performance specification via choice of $L_d(j\omega)$ plays an important role in the design of a PID controller using the frequency response. The parameters such as gain margin and phase margin are relatively easy to specify in terms of closed-loop stability; however, it is difficult to relate them to the actual closed-loop response performance for reference following and disturbance rejection.

Going back to the drawing board, it is necessary to find a systematic and yet a simple way to specify $L_d(j\omega)$ such that the closed-loop response performances for reference following and disturbance rejection are met. Another aspect in PID controller design apart from the performance specification is that, because of the limited complexity of the controller structure, there is a difference between what is desired and what is achievable. In other words, what we ask for in a PID control system is not necessarily achievable.

One of the effective ways to specify the desired open-loop frequency response $L_{\mathrm{d}}(\mathrm{j}\omega)$ is via the specification of the desired frequency response of the complementary sensitivity function $T_{\mathrm{d}}(\mathrm{j}\omega)$, where

$$T_{\mathrm{d}}(\mathrm{j}\omega) = \frac{L_{\mathrm{d}}(\mathrm{j}\omega)}{1 + L_{\mathrm{d}}(\mathrm{j}\omega)}.$$

Hence, if $T_{\mathrm{d}}(\mathrm{j}\omega)$ is specified, then $L_{\mathrm{d}}(\mathrm{j}\omega)$ is calculated as

$$L_{\mathrm{d}}(\mathrm{j}\omega) = \frac{T_{\mathrm{d}}(\mathrm{j}\omega)}{1 - T_{\mathrm{d}}(\mathrm{j}\omega)}. \tag{8.17}$$

The properties of $T_{\mathrm{d}}(\mathrm{j}\omega)$ are directly related to reference following and noise attenuation, as well as indirectly to disturbance rejection via the frequency response of the desired sensitivity function

$$S_{\mathrm{d}}(\mathrm{j}\omega) = 1 - T_{\mathrm{d}}(\mathrm{j}\omega).$$

What are the key characteristics of a complementary sensitivity function? There are four basic characteristics listed as below.

1. The desired complementary sensitivity function $T_{\mathrm{d}}(s)$ must have all poles on the left-hand side of the complex plane.
2. With a PID controller in the feedback control, the complementary sensitivity $T_{\mathrm{d}}(s)$ must be equal to unity at $s = 0$.
3. The plant unstable zeros contained in $G(s)$ will be in the presence of $T_{\mathrm{d}}(s)$ because the plant unstable zeros cannot be changed through feedback control.
4. The plant time delay e^{-ds} will appear in the desired complementary sensitivity function because the plant time delay can not be changed through feedback control.

All the characteristics can be easily verified with closed-loop transfer function calculations, which is left as an exercise.

In view of these characteristics of the desired complementary sensitivity function, without a complete knowledge about the system transfer function $G(s)$, it could be a difficult task to choose a suitable $T_{\mathrm{d}}(s)$ in its own right. This task could become even more difficult when the plant frequency information $G(\mathrm{j}\omega)$ is given at one or two frequency points.

The specification of $T_{\mathrm{d}}(s)$ is proposed as follows so that the PID controller design using frequency response data remains effective while maintaining the original simplicity. This specification was originally proposed in Wang et al. (1995) and was described in more detail in Wang and Cluett (2000).

We assume that the plant transfer function $G(s)$ is stable with all poles on the left-half complex plane and the system has no severely underdamped poles. With these assumptions, the behaviour of a control signal to a step reference signal in an over-damped closed-loop control system can be approximated by a first order response. This behaviour is then described by the desired control sensitivity function $S_u(s)$ with the first order transfer function:

$$S_u(s) = \frac{1}{K_p} \frac{\frac{1}{\beta}\tau_{cl}s + 1}{\tau_{cl}s + 1} \tag{8.18}$$

where $\tau_{cl} > 0$ is the desired closed-loop time constant for the control signal, the parameter β is selected so that $\frac{\tau_{cl}}{\beta} > 0$ is approximately equal to the dominant time constant of the system, K_p is the steady-state gain of the system. When the dominant time constant of the system is unknown, which is the case for using the plant frequency response data in the design, the parameter β is a tuning parameter.

The desired complementary sensitivity function follows from the desired control sensitivity function in the form:

$$T_d(s) = S_u(s)G(s) = \frac{1}{K_p} \frac{\frac{1}{\beta}\tau_{cl}s + 1}{\tau_{cl}s + 1} G(s) \tag{8.19}$$

Clearly, $T_d(s)$ is stable as $\tau_{cl} > 0$ and the transfer function $G(s)$ is assumed to be stable; $T_d(s)$ at steady-state ($s = 0$) is equal to unity because of the factor $\frac{1}{K_p}$, where K_p is the steady-state gain of $G(s)$; and the time delay or zeros in $G(s)$ are contained in $T_d(s)$. Therefore, all four characteristics of T_d have been included in this simple specification.

If the dominant time constant of the plant is estimated (or known) as τ_{op}, then the desired closed-loop time constant τ_{cl} is chosen to be $\beta\tau_{op}$, where $\beta = \frac{\tau_{cl}}{\tau_{op}}$.

8.3.3 Design Examples

Example 8.4 *Consider PID controller design for the higher order system used in Example 8.2 (see the transfer function given by (8.6)). Choosing $\omega_1 = 0.04801$, $\omega_2 = 0.108$ (cross-over frequency of $G(j\omega)$), which are the same as those used in Example 8.3. Fixing $\tau_{cl} = 3$, investigate the effect of β on the closed-loop performance in terms of reference following and disturbance rejection.*

Solution. *Because the system transfer function is known, the desired complementary sensitivity function $T_d(s)$ is seen to be, with the specification of $\tau_{cl} = 3$,*

$$T_d(s) = \frac{1}{K_p} \frac{\frac{\tau_{cl}}{\beta}s + 1}{\tau_{cl}s + 1} G(s) = \frac{\frac{3}{\beta}s + 1}{3s + 1} \frac{(-s^2 + s + 1)}{(10s + 1)(8s + 1)(6s + 1)(5s + 1)} e^{-6s}$$

Here, the choice of $\tau_{cl} = 3$ adds a small time constant to the desired closed-loop system, which essentially does not change much the desired closed-loop response speed. However, the specification of τ_{cl}/β will add a lead element to $T_d(s)$, which will have an impact on the desired closed-loop response speed.

We will first consider the selection of $\beta = \frac{3}{10}$. This leads to pole-zero cancelation for the pole corresponding to the largest time constant 10. *With this specification, the desired loop frequency responses at ω_1 and ω_2 are calculated using (8.17),*

$$L_d(j\omega_1) = -0.3986 - j0.6717; \quad L_d(j\omega_2) = -0.3420 - j0.1166.$$

The calculation of PID controller coefficients c_2, c_1, c_0 follows from (8.14)–(8.16) based on which the PID controller parameters are obtained as,

$$K_c = -0.2837; \quad \tau_I = 22.1468; \quad \tau_D = 7.2192. \tag{8.20}$$

For comparison purpose, we investigate the effect of β on the closed-loop response. With the choice of $\beta = \frac{3}{20}$, the PID controller parameters are calculated as

$$K_c = -0.4083; \tau_I = 20.3657; \tau_D = 8.4836.$$

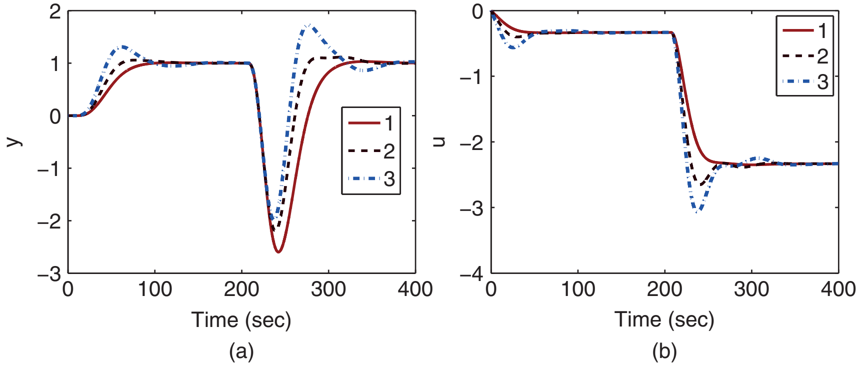


Figure 8.5 Comparison of closed-loop responses of PID control systems with different performance specifications (Example 8.4). (a) Output response. (b) Control signal. Key: line (1) $\tau_{cl} = 3$ and $\beta = \frac{3}{10}$; line (2) $\tau_{cl} = 3$ and $\beta = \frac{3}{20}$; line (3) $\tau_{cl} = 3$ and $\beta = \frac{3}{30}$.

Further decrease of β to $\beta = \frac{3}{30}$ leads to the PID controller parameters as

$$K_c = -0.4526; \tau_I = 14.4588; \tau_D = 10.4671.$$

Clearly, with the decrease of β, the proportional control gain K_c has increased in magnitude, and integral time constant τ_I has reduced and the derivative gain τ_D has increased. The parameters of the PID controller indicate that the closed-loop responses become faster as β decreases. Figure 8.5 compares the closed-loop simulation results for the three cases for a unit step response and an input disturbance rejection where the disturbance magnitude is 2 entering the system at $t = 200$ (sec). In comparison with the closed-loop response when using the gain and phase margin specifications (see Example 8.3 and Figure 8.4), it is clearly seen with this specification that the closed-loop response to the input disturbance has been significantly improved.

Note that a derivative filter with time constant $0.1\tau_D$ has been used in the simulation, and both proportional control and derivation control are implemented on the output only.

8.3.4 MATLAB Tutorial on PID Controller Design Using two Frequency Points

The objectives of the following two tutorials are to produce a MATLAB program for PID controller design using two frequency response points (see Tutorial 8.1) and to test this program using a simulation example (see Tutorial 8.2).

Tutorial 8.1 *We need plant frequency information at two frequencies that are denoted as 'w1' and 'w2'. The frequency response at 'w1' is denoted by 'Gjw1' and at 'w2' is denoted by 'Gjw2'. As in PI controller design, K is the estimated steady-state gain, 'beta' and 'taucl' are specified according to the control sensitivity function*

$$S_u(s) = \frac{1}{K} \frac{\frac{1}{\beta}\tau_{cl}s + 1}{\tau_{cl}s + 1}$$

Step by Step

1. *Create a new file called FR4PID.m for the MATLAB function.*
2. *Define the input and output variables for the MATLAB function. Enter the following program into the file:*

```
function [Kc,tauI,tauD]=FR4PID(beta,taucl,w1,w2,Gjw1,Gjw2,K)
```

3. *Calculate the frequency response of the control sensitivity function at ω_1 and ω_2. Continue entering the following program into the file:*

```
j=sqrt(-1);
Sujw1=(j*w1*taucl/beta+1)/(K*(j*w1*taucl+1));
Sujw2=(j*w2*taucl/beta+1)/(K*(j*w2*taucl+1));
```

4. *Calculate the frequency response of the complementary sensitivity function $T(j\omega)$ and the desired open-loop frequency response $L_d(j\omega)$ at ω_1 and ω_2. Continue entering the following program into the file:*

```
Tjw1=Sujw1*Gjw1;
Ljw1=Tjw1/(1-Tjw1);
Tjw2=Sujw2*Gjw2;
Ljw2=Tjw2/(1-Tjw2);
```

5. *Calculate $X(j\omega_1)$ and $X(j\omega_2)$ using (8.12) and (8.13). Continue entering the following program into the file:*

```
Xjw1=j*w1*Ljw1/Gjw1;
Xjw2=j*w2*Ljw2/Gjw2;
```

6. *Calculate the controller coefficients c_2, c_1 and c_0 using $X(j\omega_1)$ and $X(j\omega_2)$. Continue entering the following program into the file:*

```
c1=imag(Xjw1)/w1;
c2=-(real(Xjw2)-real(Xjw1))/(w2^2-w1^2);
c0=c2*w1^2+real(Xjw1);
```

7. *Finally, calculate the PID controller parameters, K_c, τ_I and τ_D. Continue entering the following program into the file:*

```
Kc=c1;
tauI=c1/c0;
tauD=c2/c1;
```

The program needs to be tested so that we can use it for applications.

Tutorial 8.2 *We select a system with dominant time delay that has the following transfer function,*

$$G(s) = \frac{0.4e^{-20s}}{(2s+1)^2(s+1)}.$$

Step by Step

1. *We create a new file called 'test4FR2PID.m'.*

2. *Define the system under test. Enter the following program into the file:*

```
delay=20;
K=0.4;
num=K;
tau1=2;
tau2=2;
tau3=1;
den1=conv([tau1 1],[tau2 1]);
den=conv(den1,[tau3 1]);
w=0.001:0.01:1;
Gjw0=freqs(num,den,w).*exp(-j*w*delay);
```

3. *plot the real and imaginary parts of 'Gjw0' as before to find* $\omega_{\frac{\pi}{2}} = 0.061$ *and* $\omega_{\pi} = 0.131$.
4. *These two frequencies will be used to find the PID controller parameters. We need to add time delay to the frequency calculation. Continue entering the following program into the file:*

```
w1=[0.061 0.131];
Gjw=freqs(num,den,w1).*exp(-j*w1*delay);
```

5. *We choose* $\beta = 5$ *and* $\tau_{cl} = 5$, *then use the function 'FR4PID.m' to calculate the PID controller parameters. Continue entering the following program into the file:*

```
taucl=5;
beta=5;
[Kc,tauI,tauD]=FR4PID(beta, taucl,w1(1),w1(2),Gjw(1),Gjw(2),K);
```

6. *The function gives* $K_c = 0.9924$, $\tau_I = 11.1598$ *and* $\tau_D = 3.3389$.
7. *If one wishes to have a slower closed-loop response, the desired closed-loop time constant* τ_{cl} *could be increased. When* $\tau_{cl} = 10$, $\beta = 5$, $K_c = 0.7994$, $\tau_I = 10.5643$ *and* $\tau_D = 1.7446$.

Closed-loop simulation of the PID control systems are performed using a derivative filter with the filter time constant being $0.1\tau_D$. Both the proportional control term and derivative control term are implemented on the output only. Figure 8.6 shows the closed-loop responses for reference following of a unit step signal and disturbance rejection. The input step disturbance with amplitude of 2 enters the system at $t = 100$ (s). It is seen that by increasing τ_{cl}, the closed-loop response speed is reduced, however, the slight oscillation with the smaller τ_{cl} is overcome.

Note that the MATLAB program FR4PID.m will be used for auto-tuner design in Chapter 9 where the plant frequency information at ω_1 and ω_2 will be found by the relay feedback experiments. Additionally, the PID controller will degrade to a PI controller if the derivative gain τ_D is either negative or is too small. In the case of the PI controller, the proportional gain K_c and τ_I remain unchanged from the calculation of the FR4PID.m program.

8.3.5 PID Controller Design for Beer Filtration Process

In the work by Lees and Wang (2015), two transfer function models were estimated for a beer filtration process at different operational conditions.

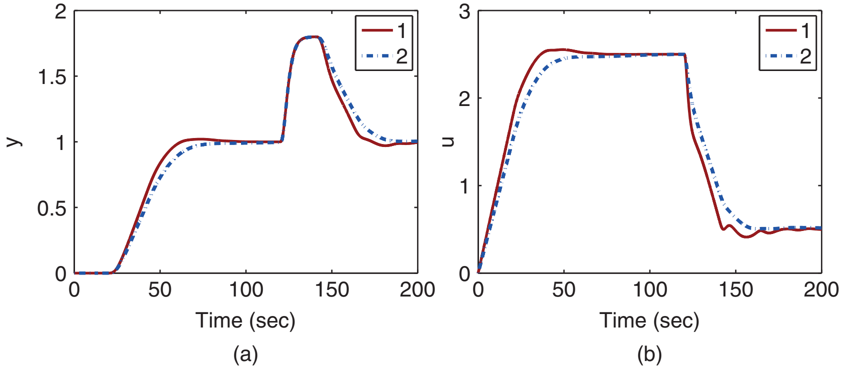


Figure 8.6 Comparison of closed-loop responses of PID control systems. (a) Output response. (b) Control signal. Key: line (1) $\tau_{cl} = 5$ and $\beta = 5$; line (2) $\tau_{cl} = 10$ and $\beta = 5$

For the first operational condition, step response experiments were conducted to obtain the estimated transfer function:

$$G_1(s) = \frac{0.0216s - 0.0031}{s^2 + 0.4576s + 0.0868} e^{-s/6}$$

For the second operational condition, the estimated transfer function is

$$G_2(s) = \frac{0.0174s - 0.0046}{s^2 + 0.5978s + 0.0445} e^{-s/6}$$

where the time unit for the transfer functions is minute, instead of second. The filtration process is clearly a nonlinear system, in which the system dynamics change with respect to operating conditions.

In order to design a single PID controller for the system, the frequency responses of the transfer function models are then averaged point-by-point. Figure 8.7 shows the frequency response of $G_1(j\omega)$, $G_2(j\omega)$ and the averaged frequency response $G_a(j\omega)$. To obtain the two frequency response points ω_1 and ω_2, the real and imaginary parts of $G_a(j\omega)$ are examined, where $\omega_1 = 0.133$ is identified as the point when the real part of $G_a(j\omega)$ changes sign from negative to positive and $\omega_2 = 0.432$ is identified as the point where the imaginary part changes sign from positive to negative. The corresponding frequency response $G_a(j\omega)$ at $\omega = \omega_1$ is $0.0009 + j0.0532$ and at ω_2 is $0.037 + j0.0001$.

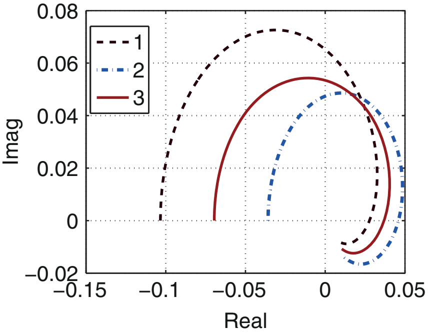


Figure 8.7 Frequency response. Key: line (1) $G_1(j\omega)$; line (2) $G_2(j\omega)$; line (3) $G_a(j\omega)$

The desired closed-loop performance is specified at ω_1 and ω_2 through the following relationship:

$$T_d(j\omega) = \frac{1}{K_p} \frac{\frac{1}{\beta}\tau_{cl} j\omega + 1}{\tau_{cl} j\omega + 1} G_a(j\omega) \tag{8.21}$$

where $K_p = -0.0696$, $\tau_{cl} = 7.187$ and $\beta = 0.625$. Note that we have selected $\frac{\tau_{cl}}{\beta} = 11.5$, which corresponds to the dominant time constant of $G_1(s)$.

Using the MATLAB function FR4PID.m produced in Tutorial 8.1, we calculate the PID controller parameters:

$$K_c = -11.795; \tau_I = 8.4069; \tau_D = 0.7832$$

Figure 8.8 shows the frequency response $C(j\omega)G_1(j\omega)$ and $C(j\omega)G_2(j\omega)$, from which we can estimate that the closed-loop control systems have the minimum gain margin of 2 and phase margin $\pi/4$.

In the closed-loop simulation, the PID controller is discretized and a derivative filter with time constant $\tau_f = 0.1\tau_D$ is added to the derivative term to avoid amplification of measurement noise. The discretized control signal ready for implementation is calculated using (4.40) in Chapter 4. Additionally, the control signal is computed with quantization for the possible implementation of the control system by a plant operator. The control signal with quantization is chosen to be a multiple of 0.01, which corresponds to 1 percent change in the control signal as the basis unit. Also, if the calculated control signal change $|\Delta u| = |u(t_i) - u(t_{i-1})|$ is less than 1 percent, then the control signal remains constant.

The control objective is to maintain a constant output $y(t)$, and due to the filtration operation, it drops with respect to time. The closed-loop control system is simulated with an output disturbance added to the system while maintaining a constant reference response. The typical case of the disturbance mimics the situation where the $y(t)$ reduces in a series of step changes. Because of the nonlinearity, the same PID controller is used to control both $G_1(s)$ and $G_2(s)$ in the simulation studies. Figure 8.9 shows the control signal response and output response to the output disturbance in a series of steps. It is seen that the closed-loop PID control has maintained the constant output value despite of the disturbance. Note that with the same filter, but at different operational time, $G_2(s)$ has a smaller steady-state gain, corresponding to the filter condition deteriorating. As a

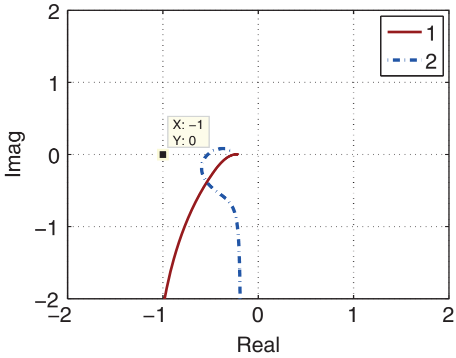
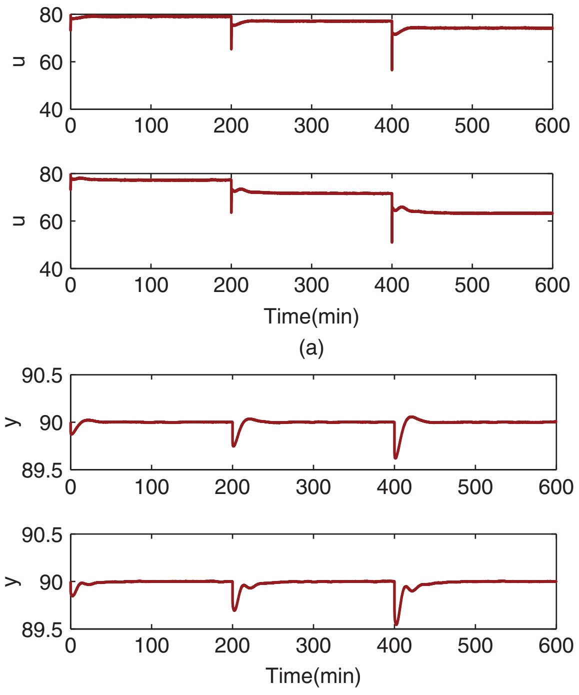


Figure 8.8 Nyquist plot. Key: line (1) $C(j\omega)G_1(j\omega)$; line (2) $C(j\omega)G_2(j\omega)$.



(a)

(b)

Figure 8.9 Closed-loop control simulation for output stair case disturbance rejection. (a) Control response (top figure: results from using $G_1(s)$, bottom figure: results from using $G_2(s)$). (b) Output response (top figure: results from using $G_1(s)$, bottom figure: results from using $G_2(s)$).

result, a larger steady-state control signal is required to maintain the same operational conditions. This is evident from comparing the control signals in Figure 8.9 (a).

8.3.6 Food for Thought

1. In the PID controller parameter solutions (see (8.14)–(8.16)), is it correct to say that the proportional controller gain K_c only uses the information from the first frequency point ω_1, however, τ_I and τ_D use the information from both first and second frequency points?
2. Upon finding the controller parameters, K_c, τ_I and τ_D, using the information from the two frequency points, we have the options to test different combinations of controllers without changing their parameters. First instance, we can use the proportional controller with K_c, PI controller with K_c and τ_I, PID controller with K_c, τ_I and τ_D or PD controller with K_c and τ_D. Why do you think that the frequency domain design can lead to such a result?
3. We have chosen ω_1 and ω_2 for easy implementation. Can we choose other frequency points as ω_1 and ω_2 as long as they are in the medium frequency range with cross-over frequency contained? Why is that?
4. In the specification of desired closed-loop transfer function (see (8.19)), if the parameter $\beta = 1$, the closed-loop dominant time constant is specified to be equal to the open-loop dominant time constant but with unity steady-state gain. Would you consider this as a default choice?

8.4 PID Controller Design for Integrating Systems

PID control of integrating systems has become increasingly important in control engineering applications. A large number of electro-mechanical systems can be classified as integrating plus time delay systems. For instance, the angular position control of a robot is the control of integrating system, and the quadrotor control is also related to control of integrating systems.

The most widely encountered integrating systems have time delay in addition to first order or higher order dynamics. Because the integral action is expressed as a pole on the origin of the complex plane, which essentially is the dominant dynamics for an integrating system, it may not be necessarily to capture the first order or higher dynamics in the design of PID controllers. Instead, these stable dynamics are approximated using an equivalent time delay to describe the effect of their phase lag in the PID control system design.

8.4.1 The Approximate Model

The approximate model of an integrating system is assumed to be of the following form:

$$G(s) = \frac{K_p e^{-ds}}{s} \tag{8.22}$$

where K_p is the gain of the integrating system and d is its time delay. For most physical systems, there are more or less approximations involved in obtaining the integrating plus time delay model. An easy way to find the parameters in (8.22) is through frequency response analysis.

Assume that the frequency response $G(j\omega_1)$ is available at the frequency ω_1. This frequency information $G(j\omega_1)$ is estimated using the relay experiments in many applications as shown in the next chapter.

Now, letting the frequency response of the integrating plus delay model (8.22) be equal to the measured $G(j\omega_1)$ leads to

$$\frac{K_p e^{-jd\omega_1}}{j\omega_1} = G(j\omega_1) \tag{8.23}$$

Equating the magnitudes on both side of (8.23) gives

$$K_p = \omega_1 |G(j\omega_1)| \tag{8.24}$$

where $|e^{-jd\omega_1}| = 1$. Additionally, from (8.23), the following relationship holds:

$$e^{-jd\omega_1} = \frac{j\omega_1 G(j\omega_1)}{K_p}$$

This gives the estimate of time delay as

$$d = -\frac{1}{\omega_1} tan^{-1} \frac{Imag(jG(j\omega_1))}{Real(jG(j\omega_1))}$$

It is seen here that if the system is truly integrating with time delay, the plant information at a single frequency is sufficient to determine the plant gain and time delay.

8.4.2 Selection of Desired Closed-loop Performance

Because the transfer function for the time-delay e^{-ds} is irrational, approximation is often needed when using the model based designs (see Chapter 3). An effective way to avoid the approximation is to derive the PID controller parameters using the frequency response analysis.

Similar to the PID controller design introduced in the previous section, we will first introduce the specification of desired closed-loop performance. Considering the PID controller structure

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s}$$

together with the integrating plus delay model

$$G(s) = \frac{K_p e^{-ds}}{s}$$

it is clear that the loop transfer function

$$L_p(s) = \frac{K_p e^{-ds}}{s} \frac{c_2 s^2 + c_1 s + c_0}{s} \tag{8.25}$$

contains a double integrator. Therefore, this characteristic should be reflected in the selection of the desired closed-loop performance. Additionally, the four characteristics of the desired complementary sensitivity function specified in Section 8.3.2 should be satisfied. It is simpler to choose the desired control sensitivity function to this effect.

A candidate for such a choice is the control sensitivity to have the following form:

$$S_u(s) = \frac{1}{K_p} \frac{s((2\xi\tau_{cl} + d)s + 1)}{\tau_{cl}^2 s^2 + 2\xi\tau_{cl} s + 1} \tag{8.26}$$

where $\tau_{cl} > 0$ is the desired closed-loop time constant and ξ is the damping coefficient typically chosen as 0.707 or 1. A larger τ_{cl} corresponds to a slower closed-loop response speed.

The desired complementary sensitivity function $T_d(s)$ is composed of the control sensitivity $S_u(s)$ and the model $G(s)$ given by

$$\begin{aligned} T_d(s) = G(s)S_u(s) &= \frac{K_p e^{-ds}}{s} \frac{1}{K_p} \frac{s((2\xi\tau_{cl} + d)s + 1)}{\tau_{cl}^2 s^2 + 2\xi\tau_{cl} s + 1} \\ &= \frac{((2\xi\tau_{cl} + d)s + 1)e^{-ds}}{\tau_{cl}^2 s^2 + 2\xi\tau_{cl} s + 1} \end{aligned} \tag{8.27}$$

where the steady-state gain K_p and the factor s have been cancelled to obtain (8.27).

It is seen from (8.27) that the desired complementary sensitivity function $T_d(s)$ has all poles on the left-hand side of the complex plane. Additionally, the complementary sensitivity $T_d(s)$ is equal to unity at $s = 0$ and the plant time-delay e^{-ds} appears in the numerator of $T_d(s)$. Therefore, all the characteristic requirements discussed in Section 8.3.2 are satisfied for the integrating with time delay system by PID control.

Furthermore, a stable zero at $s = -\frac{1}{2\xi\tau_{cl}+d}$ is introduced in the desired complementary sensitivity. The introduction of this stable zero is to ensure that at $s = 0$ the desired loop transfer function $L_d(s)$ will have the structure of a double integrator, which matches that of the actual loop transfer function $L_p(s)$ in (8.25). This claim can be verified through the following calculation:

$$\begin{aligned}L_d(s) &= \frac{T_d(s)}{1 - T_d(s)}\\ &= \frac{((2\xi\tau_{cl} + d)s + 1)e^{-ds}}{\tau_{cl}^2 s^2 + 2\xi\tau_{cl}s + 1 - ((2\xi\tau_{cl} + d)s + 1)e^{-ds}}\end{aligned} \tag{8.28}$$

Writing the irrational transfer function e^{-ds} in Taylor series expansion gives

$$e^{-ds} = 1 - ds + \frac{1}{2!}(ds)^2 - \frac{1}{3!}(ds)^3 + \dots \approx 1 - ds + O((ds)^2) \tag{8.29}$$

where $O((ds)^2)$ denotes the higher order terms in the Taylor series. Then, the denominator of $L_d(s)$ is expressed as

$$\begin{aligned}&\tau_{cl}^2 s^2 + 2\xi\tau_{cl}s + 1 - ((2\xi\tau_{cl} + d)s + 1)e^{-ds}\\ &\quad = \tau_{cl}^2 s^2 + 2\xi\tau_{cl}s + 1 - (1 + 2\xi\tau_{cl}s - d(2\xi\tau_{cl} + d)s^2)\\ &\quad\quad + O((ds)^2)((2\xi\tau_{cl} + d)s + 1)\\ &\quad = \tau_{cl}^2 s^2 + d(2\xi\tau_{cl} + d)s^2 - O((ds)^2)((2\xi\tau_{cl} + d)s + 1)\end{aligned} \tag{8.30}$$

Since from (8.30) the higher order $O((ds)^2)$ contains a factor s^2, then it is clearly seen that the denominator $L_d(s)$ contains the factor s^2.

It is emphasized that the choice of $T_d(s)$ given by (8.27) ensures the desired loop transfer function $L_d(s)$ has the feature of a double integrator at $s = 0$ matching that of the actual loop transfer function $L_p(s)$. This means that at the lower frequency region, the PID controller parameters will automatically lead to the low frequency requirement of the sensitivity functions. This choice of desired complementary sensitivity function reduces the errors in the frequency curve fitting for computation of the PID controller parameters.

8.4.3 Normalization of the Parameters and Empirical Rules

To normalize the process parameters for derivation of the PID controller parameters in empirical rules, the actual loop transfer function $L_p(s)$ from (8.25) is re-written as

$$\begin{aligned}L_p(s) &= \frac{K_p e^{-ds}}{s} K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s\right)\\ &= \frac{\hat{K}_c e^{-\hat{s}}}{\hat{s}} \left(1 + \frac{1}{\hat{\tau}_I \hat{s}} + \hat{\tau}_D \hat{s}\right)\end{aligned} \tag{8.31}$$

where $\hat{s} = ds$, $\hat{K}_c = dK_pK_c$, $\hat{\tau}_I = \frac{\tau_I}{d}$ and $\hat{\tau}_D = \frac{\tau_D}{d}$. For convenience in computation, (8.31) is expressed as

$$L_p(s) = \frac{e^{-\hat{s}}}{\hat{s}} \frac{\hat{c}_2\hat{s}^2 + \hat{c}_1\hat{s} + \hat{c}_0}{\hat{s}} \tag{8.32}$$

where the parameters are defined as

$$\hat{K}_c = \hat{c}_1; \quad \hat{\tau}_I = \frac{\hat{c}_1}{\hat{c}_0}; \quad \hat{\tau}_D = \frac{\hat{c}_2}{\hat{c}_1}$$

Note that the loop transfer function $L_p(s)$ is free of the process gain K_p and the time delay d. Similarly, the desired loop transfer function $L_d(s)$ given by (8.28) is required to be normalized. To this end, the desired closed-loop time constant τ_{cl} is selected as the function of the time delay d:

$$\tau_{cl} = \beta d \tag{8.33}$$

where $\beta > 0$ is the desired closed-loop performance parameter used in the design. This leads to the re-writing of (8.28) in the following form:

$$\begin{aligned} L_d(s) &= \frac{((2\xi\beta d + d)s + 1)e^{-ds}}{\beta^2 d^2 s^2 + 2\xi\beta ds + 1 - ((2\xi\beta d + d)s + 1)e^{-ds}} \\ &= \frac{((2\xi\beta + 1)\hat{s} + 1)e^{-\hat{s}}}{\beta^2\hat{s}^2 + 2\xi\beta\hat{s} + 1 - ((2\xi\beta + 1)\hat{s} + 1)e^{-\hat{s}}} \end{aligned} \tag{8.34}$$

Note that the desired loop transfer function $L_d(s)$ is also free of the time delay parameter d.

The solution of the PID controller parameters follows from the frequency domain solution proposed in Section 8.3, but with different choices of the frequency points $\hat{\omega}_1$ and $\hat{\omega}_2$ (see Wang and Cluett (2000)).

Because the PID controller parameters are normalized, there are only the desired closed-loop time constant $\tau_{cl} = \beta d$ and the damping coefficient adjustable. Thus, we can find the normalized PID controller parameters numerically with respect to the parameter β and form empirical rules. There are two sets of empirical rules obtained below through polynomial fitting of the normalized PID controller parameters, together with gain and phase margins.

Selecting 100 β values from $\beta = 1.0$ to $\beta = 11$ with increment of 0.1, together with a damping coefficient ξ, there are 100 sets of normalized PID controller parameters calculated. By using the polynomial fitting tool in MATLAB to find the calculated PID parameters, the following empirical rules for the normalized parameters are obtained as shown in Tables 8.1–8.2. With the normalized PID controller parameters calculated, the actual PID controller parameters are then obtained with the scaling parameters K_p and d, as

$$K_c = \frac{\hat{K}_c}{dK_p}; \tau_I = \hat{\tau}_I d; \tau_D = \hat{\tau}_D d \tag{8.35}$$

The polynomial functions in both Tables have provided quite accurate descriptions to the original data (see Figure 8.10 as an illustration). Therefore, when an integrator with delay is given, the PID controller parameters will be calculated simply using the polynomial equations presented in the tables.

Table 8.1 Normalized PID controller parameters ($\xi = 0.707$, $1 \leq \beta \leq 11$).

$\hat{K}_c$	$\frac{1}{0.7184\beta + 0.3661}$
$\hat{\tau}_I$	$1.3970\beta + 1.2271$
$\hat{\tau}_D$	$\frac{1}{1.4275\beta + 1.6450}$

Table 8.2 Normalized PID controller parameters ($\xi = 1, 1 \le \beta \le 11$).

$\hat{K}_c$	$\dfrac{1}{0.5138\beta + 0.5909}$
$\hat{\tau}_I$	$1.9886\beta + 1.2118$
$\hat{\tau}_D$	$\dfrac{1}{1.0156\beta + 1.7550}$

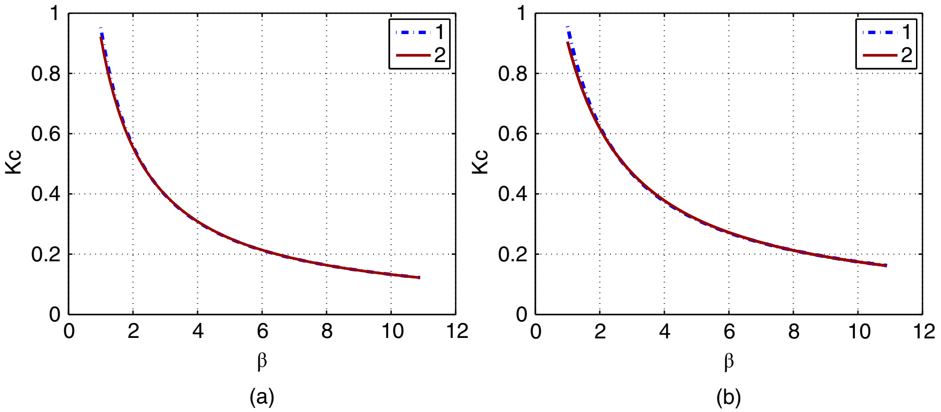


Figure 8.10 Calculated normalized proportional controller gain. (a) $\xi = 0.707$, $1 \le \beta \le 11$. (b) $\xi = 1$, $1 < \beta \le 11$. Key: line (1) data; line (2) using Tables 8.1 and 8.2.

8.4.4 Gain and Phase Margins

Because the original PID controller parameters are calculated using two frequency response data points, the PID controller parameters can be used in combination to obtain PID controller, PI controller and PD controller.

The gain and phase margins for the PID controllers are calculated using the empirical forms, which are also function of the parameter β and are shown in Figure 8.11. Additionally, the gain and phase margins for PI and PD controllers are calculated shown in Figures 8.12 and 8.13. These gain and phase margins are useful in measuring closed-loop performance and quantify robustness of the PID control system designed. It also provides some guidance on the choice of controller structures.

8.4.5 Simulation Examples

Example 8.5 *Consider the integrator plus time delay system described by the transfer function*

$$G(s) = \frac{e^{-5s}}{s}$$

In the first part of the example, we will evaluate the performance of the PID, PI, and PD controllers designed. In the second part of the example, we will show that the closed-loop performance is similar to a PID controller using the pole assignment design method.

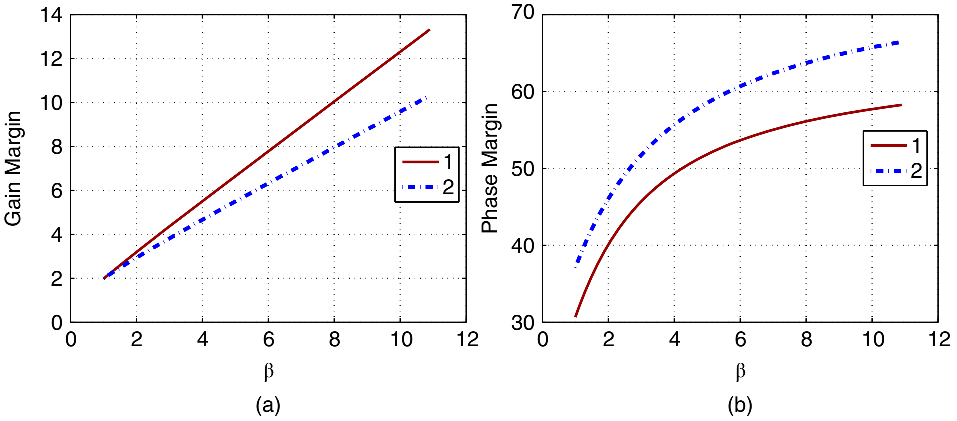


Figure 8.11 Calculated gain and phase margins for PID controllers. (a) Gain margin. (b) Phase margin. Key: line (1) using Table 8.1 ($\xi = 0.707$); line (2) using Table 8.2 ($\xi = 1$).

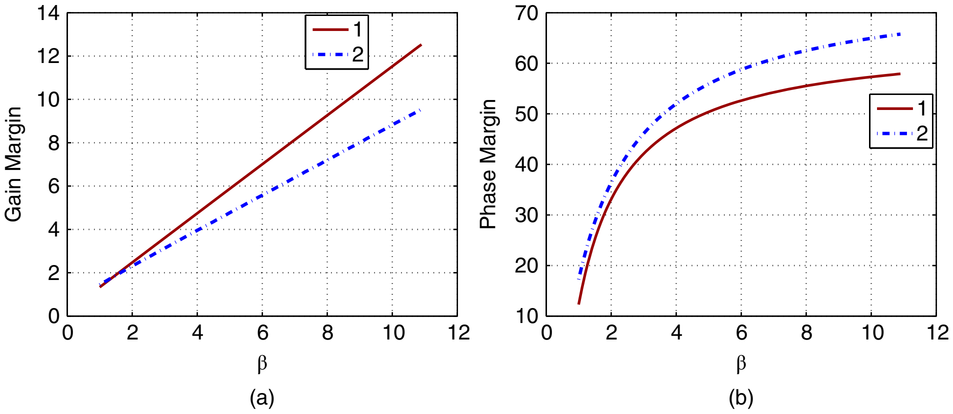


Figure 8.12 Calculated gain and phase margins for PI controllers. (a) Gain margin. (b) Phase margin. Key: line (1) using Table 8.1 ($\xi = 0.707$); line (2) using Table 8.2 ($\xi = 1$).

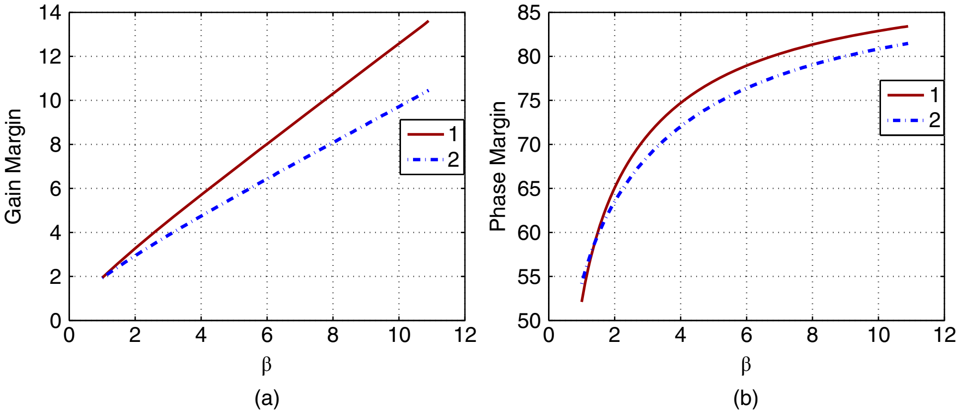


Figure 8.13 Calculated gain and phase margins for PD controllers. (a) Gain margin. (b) Phase margin. Key: line (1) using Table 8.1 ($\xi = 0.707$); line (2) using Table 8.2 ($\xi = 1$).

However, there is no comparable PI or PD controller when using model based design. The damping coefficient ξ is chosen to be 0.707.

Solution. *From Figures 8.11–8.13, when $\beta = 1$, we can see that the phase margin for the PID control system is about 45°, for the PI control system about 20°, and for the PD control system about 55°. All gain margins exceeded 1.8. Thus, it is expected that the closed-loop systems are stable for all three types of controller. There is a closed-loop oscillation when using the PI controller because its phase margin is too small. We usually need to have a phase margin larger than 40° to avoid closed-loop oscillation.*

From Table 8.1, the normalized PID controller parameters are calculated, and from (8.35) the actual PID controller parameters are calculated to give

$$K_c = 0.1844; \quad \tau_I = 13.1205; \quad \tau_D = 1.6297.$$

The closed-loop systems are simulated for a step reference change at $t = 0$ and a step input disturbance entering system at $t = 150$ (s). The PI controller takes the form $C(s) = K_c(1 + \frac{1}{\tau_I s})$ and PD controller takes the form $C(s) = K_c(1 + \frac{\tau_D s}{0.1\tau_D s+1})$. All derivative terms are implemented on output only. Figure 8.14 compares the control signals and the output signals for the three control systems. It is seen that the PID control system shows satisfactory performance, the PI control system has oscillatory responses due to the small phase margin, and PD controller has a better performance in reference tracking; however, it is not capable of rejecting the step input disturbance, leading to steady-state errors in the disturbance rejection.

The second part of this example is to design a PID controller for the integrator plus delay model using the pole assignment controller design method. Here the Padé approximation is used to approximate the time delay to yield

$$G_p(s) = \frac{e^{-5s}}{s} \approx \frac{(-s + 0.4)}{s(s + 0.4)}. \tag{8.36}$$

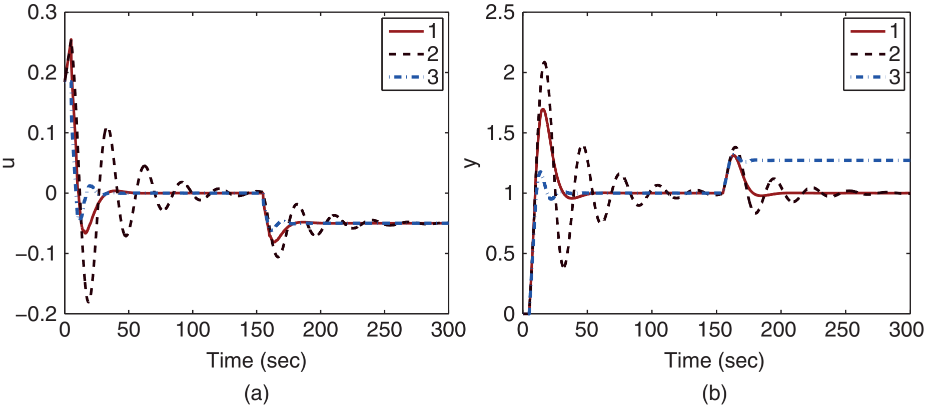


Figure 8.14 Comparison of closed-loop performance for three types of controllers (Example 8.5). (a) Control signal. (b) Output. Key: line (1) PID control response; line (2) PI control response; line (3) PD control response.

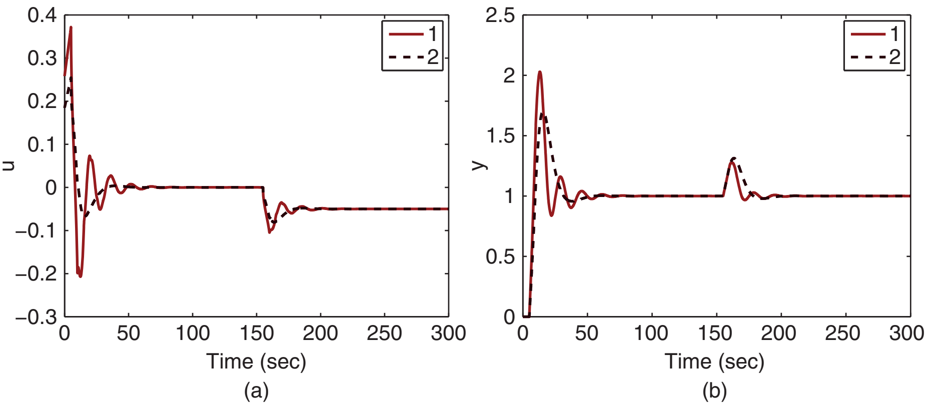


Figure 8.15 Comparison of closed-loop PID control performance between the model based design and the tuning rules (Example 8.5). (a) Control signal. (b) Output signal. Key: line (1) model based design; line (2) using tuning rules.

Choose the desired closed-loop polynomial as

$$A_{cl} = (s^2 + 0.707w_n s + w_n^2)(s + 5w_n)^2$$

where $w_n = 1/d = 0.2$, which is equivalent to the selection of the dominant closed-loop time constant d. With this selection, the dominant closed-loop time constant is equal to the previous case. Using the pole-assignment design method, the PID controller parameters are found as

$$K_c = 0.2581; \quad \tau_I = 11.3425; \quad \tau_D = 1.9870; \quad \tau_f = 0.2275$$

Although the two design approaches have selected an equivalent desired closed-loop time constant, the model based design method leads to the closed-loop control system with oscillation [see Figure 8.15]. This is likely caused by the modeling error between the approximated transfer function and the true integrator with time delay model [see (8.36)].

Interestingly, for the model-based design, the direct combination of proportional control and integral control is unstable for this design method.

8.4.6 Food for Thought

1. In the derivation of the PID controller rules for integrating with delay system, why is the desired closed-loop control sensitivity function specified with a stable zero?
2. Do you expect an overshoot in closed-loop response to the step reference signal by observing the desired complementary sensitivity function $T_d(s)$ in (8.27)? If you wish to eliminate such an overshoot, which reference filter should you choose in a two-degrees of freedom controller implementation?
3. To obtain an approximate integrating with time delay model, a frequency ω_1 is required. Which region on the Nyquist curve of a system is a good candidate? Why?

8.5 Summary

This chapter has discussed several approaches to PID controller design using frequency domain information. PID controllers can be designed using gain margin and phase margin as their performance specifications in the frequency domain. One drawback with respect to the gain margin and phase margin specification is that these parameters are not related to the closed-loop response speed in a simple and intuitive way.

The other important aspects in the chapter are summarized as follows.

- PID controller parameters can be found analytically in a manner related to curve fitting of the open-loop frequency response in two frequency points. In this approach, the closed-loop performance specification is the desired dominant time constant via complementary sensitivity function.
- In order to produce the best fit possible with the limited number of controller parameters, the complementary sensitivity function contains the zeros of the plant as well as the time delay of the plant.
- A special case of the approach is the PID controller for integrator with delay system. With a normalized delay parameter, the PID controller parameters are simply expressed in empirical forms analytically with achieved gain margin and phase margin.

8.6 Further Reading

1. The PID controller design methods using two points of frequency response data were originally introduced in Wang et al. (1995), Wang and Cluett (1997) and Wang and Cluett (2000).
2. More PID controller design techniques using gain margin and phase margin specifications can be found in Ho et al. (1995), Ho et al. (1996), Ho and Xu (1998), Ho et al. (1998), Ho et al. (2000).
3. Second order with delay model was obtained for PID controller design by using two frequency response points together with nonlinear optimization in Wang et al. (1999).

Problems

8.1 *The transfer function of a complex system is given by*

$$G(s) = \frac{-3(-s^2+s+1)}{(10s+1)(8s+1)(6s+1)(5s+1)}e^{-6s} \tag{8.37}$$

1. *Design a PI controller for this system by specifying the desired gain margin of* 2.
2. *Determine the phase margin and delay margin using Nyquist diagram.*

8.2 *The transfer function of a complex, underdamped system is given by*

$$G(s) = \frac{-3(-s^2+s+1)}{(10s^2+20\xi s+1)(6s+1)(5s+1)}e^{-6s} \tag{8.38}$$

where $\xi = 0.3$.

1. *Design a PI controller for this system by specifying the desired phase margin of* $\theta = \frac{\pi}{3}$.
2. *Determine the gain margin and delay margin using Nyquist diagram.*

3. *Show using simulation studies that the closed-loop response to a step input disturbance is oscillatory no matter what you do.*
4. *Explain your results by examining the sensitivity function between the step input disturbance and output.*

8.3 *The mathematical model for the eighth reactor in a copolymerization reactor train is described by the following transfer function model (Madhuranthakam and Penlidis (2016)):*

$$Y(s) = \left[\frac{Ks + 1}{\tau_1^2 s^2 + 2\tau_1\tau_2 s + 1}\right]^8 U(s) \tag{8.39}$$

where the input is the flow rate of the Chain Transfer Agent (CTA) to the first reactor in the reactor train and the output is the weight-based average molecular weight (MW_w). The parameters in the transfer function for the reactor are given as $K = 361.54$, $\tau_1 = 106.84$ and $\tau_2 = 1.72$. Design PID controller for this polymer reactor using two frequency response points (see Section 8.3).

1. *Choosing $\beta = 1$, which is case that the desired closed-loop transfer function equal to the open-loop transfer function, find the PID controller parameters.*
2. *Choosing $\beta = 2$ and $\tau_{cl} = 200$, find the PID controller parameters.*
3. *Choosing $\beta = 0.5$ and $\tau_{cl} = 50$, find the PID controller parameters.*
4. *Compare the Nyquist plots for the PID control systems and find their gain margin, phase margin and delay margin.*
5. *Simulate the closed-loop PID control systems with unit step reference signal and a unit step disturbance entering the systems at half of the simulation time. The sampling interval Δt is 1 (min), and the simulation time is 3000 (min) because it is a very slow process. In the simulation, both proportional control and derivative control are implemented on the output only to reduce overshoot in the reference response. The derivative filter time constant is selected as $0.1\tau_D$. For the simulations studies, use the MATLAB real-time function PIDV.slx created in Tutorial 4.1.*
6. *What are the observations when we compare the three PID control systems? What are the observations when we compare the three PID control systems with the results obtained in Example 2.7?*
7. *What are the reasons behind the closed-loop performance improvement when using the frequency domain based design technique for this particular system?*

9

Automatic Tuning of PID Controllers

9.1 Introduction

Relay feedback control has been one of the key instruments used in the automatic tuning of PID controllers. Its application is associated with the identification of process frequency response information via the self-generated excitation signals in a closed-loop operation. Because it is in a closed-loop operation, the feedback control effect maintains the system around its operating condition while conducting the experiment.

This chapter will discuss the automatic tuning of PID controllers where the process frequency response information is obtained from relay feedback experiments and the PID controllers are designed using the design methods introduced in Chapter 8.

9.2 Relay Feedback Control

This section will discuss relay feedback control systems in combination with hysteresis or with an integrator. Simulink tutorials are presented for both cases, which is convenient for simulation studies and experimental validations.

9.2.1 Relay Control with Hysteresis

In applications, because of measurement noise, a hysteresis element is incorporated within the relay feedback control mechanism to avoid possible random switches due to the effect of noise. Figure 9.1 illustrates a block diagram of relay feedback control.

For the purpose of relay feedback control, there are a few parameters that need to be defined.

1. A plant steady-state operating condition with input signal u_{ss} and output signal y_{ss} is chosen based on which relay control experiment will be carried out. For some applications, the steady-state input and output signals are zero.
2. The amplitude of the relay control signal is chosen as parameter a, which represents the variations of the input signal at the steady-state operating condition.
3. The hysteresis parameter is chosen to be ϵ, which is used to prevent the triggering of relay switching from random noise. If the standard deviation of the measurement noise is σ, then approximately the hysteresis parameter ϵ is chosen to be 3σ.

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

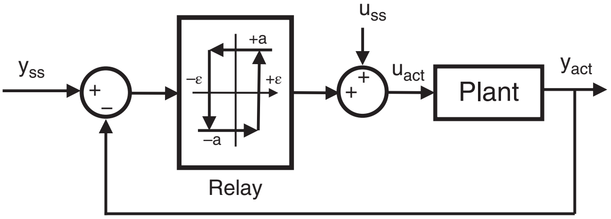


Figure 9.1 Block diagram of relay feedback control.

We define the deviation variable of the control signal as

$$u(t) = u_{\text{act}}(t) - u_{\text{ss}}$$

and the initial condition of the deviation control signal as $u(t_0) = a$. We assume that the system is already operating at the steady state. With the actual measurement of the output $y_{\text{act}}(t_i)$ at the sampling time t_i, the closed-loop control signal $u_{\text{act}}(t_i)$ is calculated using the following relay switching rules:

$$u(t_i) = \begin{cases} u(t_{i-1}) & \text{if } |e(t_i)| \leq \epsilon \\ a \times \text{sign}(e(t_i)) & \text{if } |e(t_i)| > \epsilon \end{cases} \tag{9.1}$$

and

$$u_{\text{act}}(t_i) = u(t_i) + u_{\text{ss}} \tag{9.2}$$

where $e(t_i) = y_{\text{ss}} - y_{\text{act}}(t_i)$ is the feedback error at the sampling time t_i.

It is worthwhile emphasizing that the relay switching rule given in (9.1) is for the systems that have a positive steady-state gain. For the systems having a negative steady-state gain, the deviation variable $u(t_i)$ is calculated as

$$u(t_i) = \begin{cases} u(t_{i-1}) & \text{if } |e(t_i)| \leq \epsilon \\ -a \times \text{sign}(e(t_i)) & \text{if } |e(t_i)| > \epsilon \end{cases} \tag{9.3}$$

$$u_{\text{act}}(t_i) = u(t_i) + u_{\text{ss}}. \tag{9.4}$$

Clearly, the relay feedback control is a nonlinear control law with a small amount of a priori information required and it is very easy to implement.

It is known (Astrom and Hagglund (1984), Astrom and Hagglund (1988)) that this relay controlled system will generate a sustained periodic oscillation that contains the fundamental frequency ω_1 at the point of the Nyquist curve of the process, which approximately has the imaginary part as $-\frac{\pi\epsilon}{4a}$, as shown in Figure 9.2. One immediately pays attention to the location of the frequency ω_1 on the Nyquist curve. If the hysteresis value ϵ is chosen to be zero, then the frequency ω_1 becomes the cross-over frequency on the Nyquist curve if a proportional controller is used. This was indeed the key link between the classical Ziegler–Nichols tuning rules and the first generation of auto-tuners (Astrom and Hagglund (1984), Astrom and Hagglund (1988)). The second point is that if the hysteresis ϵ is increased, while maintaining the same amplitude a, the frequency ω_1 is reduced, meaning that the oscillation period $T = \frac{2\pi}{\omega_1}$ is increased.

Under relay feedback control, there are several classes of systems that will have a sustained periodic oscillation. In general, these systems should be stable to ensure safety

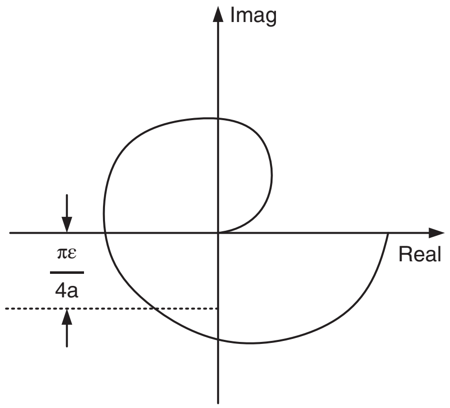


Figure 9.2 Location of ω_1 on a Nyquist curve.

of the operation during experiments. They include the first order plus delay systems, systems with higher order dynamics, systems with non-minimum phase behavior and higher order under-damped systems. The common feature of these systems is that their Nyquist curve will expand to the second and third quadratures on the complex plane, as shown in Figure 9.2. For a first order system, relay feedback control will not generate sustained periodic oscillation unless there are additional dynamics from actuation and measurement involved leading to a higher order system as a result.

The following tutorial is written to produce a MATLAB function that can be used for Simulink simulations and for real-time implementations.

Tutorial 9.1 *This tutorial is to illustrate how to implement the relay feedback control algorithm in real time. The core of this activity is to produce a MATLAB embedded function that can be used in a Simulink simulation as well as in xPC Target implementation. The entire embedded MATLAB function completes one cycle of computation for the relay feedback control signal. For every sampling period, it will repeat the same computation procedure.*

Step by Step

1. *Create a new Simulink file called RelayH.slx*
2. *In Simulink's directory of User-Defined Functions, find the icon of embedded MATLAB function and copy it to the RelayH model.*
3. *Click on the icon of the embedded function, and define the input and output variables to the RelayH model so that the embedded function has the following form:*

   ```
   function uCurrent=RelayH(e,Ra,epsilon)
   ```

   where uCurrent is the calculated relay control signal at the sampling time t_i, the first element(e) is the feedback error, the second element (Ra) is the relay amplitude, and epsilon is the hysteresis used to avoid random switches caused by noise.
4. *At the top of the embedded function, find Model Explorer among the Tools. When opening the Model Explorer, select discrete for the update method and input Deltat into the sample time; select Support variable-size arrays; select Saturate on integer overflow; select Fixed point. Click Apply to save the changes.*

5. *We need to edit the input and output data ports in order to let the embedded function know which input ports are the real time variables and which are the parameters. This editing task is performed using Model Explorer.*
   - *click on e, on Scope, select input, assign port 1 and size -1, complexity Inherited, type Inherit: Same as Simulink.*
   - *The remaining two inputs to the embedded function are the parameters required in the computation. Click on Ra, on Scope, select Parameter and click Tunable and click Apply to save the changes. Repeat the same editing procedure for the epsilon.*
   - *To edit the output port from the embedded function, click on uCurrent, on Scope, select Output, Port 1, Size -1, Sampling Model Sample based, Type Inherit: Same as Simulink, and click on Apply to save the changes.*
6. *In the following, the program will declare those variables that are stored in the embedded function during each iteration for their dimensions and initial values. uPast is the past control signal ($u(t_{i-1})$). Enter the following program into the file:*

```
persistent uPast
if isempty(uPast)
    uPast=Ra;
end
```

7. *Check to see if the feedback error is within the specified hysteresis level ($|e(t)| \leq \epsilon$) and if so, the control signal remains unchanged. Otherwise, the relay control signal equals to the sign of the feedback error multiplying the amplitude of the relay, followed by updating the past control signal $u(t_{i-1})$. Continue entering the following codes into the program.*

```
if abs(e)<=epsilon
    uCurrent=uPast;
else
   uCurrent=sign(e)*Ra;
    uPast=uCurrent;
 end
```

8. *Test this program using Simulink simulator you build for Example 9.1.*

Example 9.1 *Use relay feedback control to generate sustained oscillation for the following system:*

$$G(s) = \frac{(-2s+1)e^{-6s}}{(3s+1)(5s+1)} \tag{9.5}$$

where the measurement noise is a band-limited white noise with noise power 1 and sampling interval Δt and a gain 0.1. *In the closed-loop relay feedback control, the relay amplitude is chosen to be* 1.75, $y_{ss} = u_{ss} = 0$, *and the sampling interval* $\Delta t = 0.3$.

Solution. *Choosing the relay amplitude $a = 1.75$, we build the Simulink simulator as shown in Figure 9.3 where the band limited white noise is used with a gain 0.1 to simulate the measurement noise. We also used the RelayH.slx function in the closed-loop simulation as the relay feedback controller. Note that the noise is added in the closed-loop system, which will affect the switching frequency if the parameter ϵ is too small.*

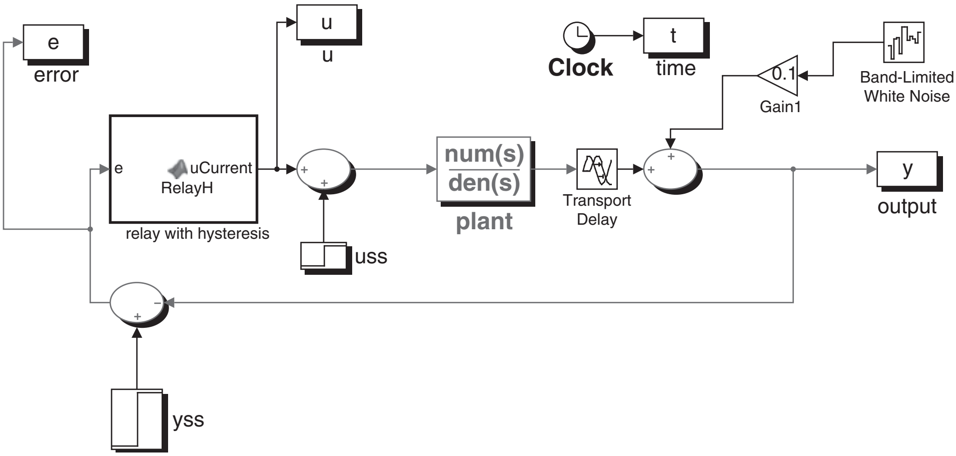


Figure 9.3 Simulink diagram for the relay feedback control.

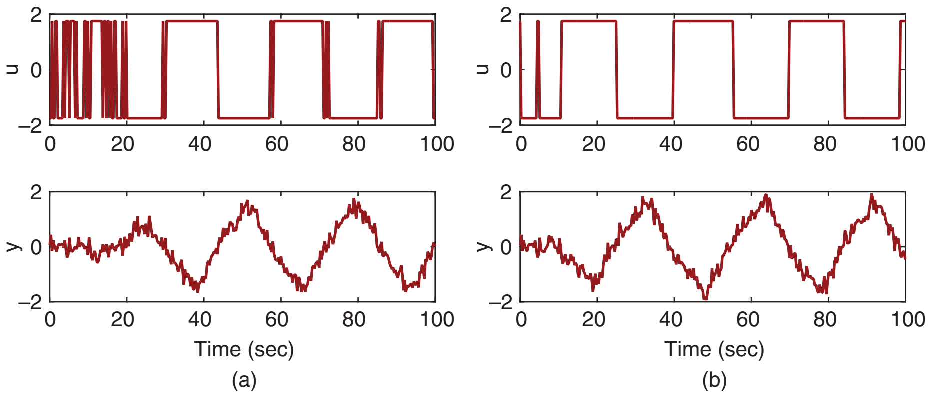


Figure 9.4 Relay feedback control signals with hysteresis (Example 9.1). (a) $\epsilon = 0.01$. (b) $\epsilon = 0.3$.

We first use a small hysteresis value $\epsilon = 0.01$, *and the relay feedback control signals are shown in Figure 9.4(a). It is seen that, because of the noise, it takes a long time to establish a stable oscillation, and additionally, the control signal responds to the noise, generating fast and random switches. In contrast, when we increase the hysteresis value* ϵ *to* 0.3, *as shown in Figure 9.4(b), the stable oscillation is established rather quickly and the control signal seldom has the fast and random switches triggered by noise once the sustained oscillation is reached.*

9.2.2 Relay Control with Integrator

Assuming that the plant is stable with all poles on the left half of the complex plane, the relay feedback controller incorporates an integrator in the closed-loop system, as illustrated in Figure 9.5. It is known (Astrom and Hagglund (1984), Astrom and Hagglund (1988)) that this relay controlled system will generate a sustained periodic oscillation that contains the fundamental frequency ω_1 at the point where the Nyquist curve of the process intersects the imaginary axis as shown in Figure 9.6.

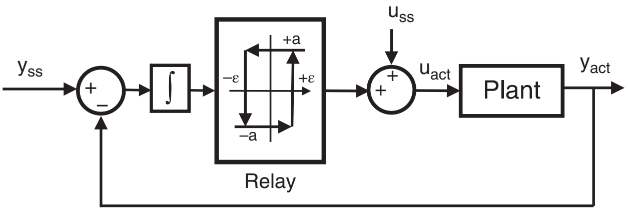


Figure 9.5 Block diagram of integrated relay feedback control.

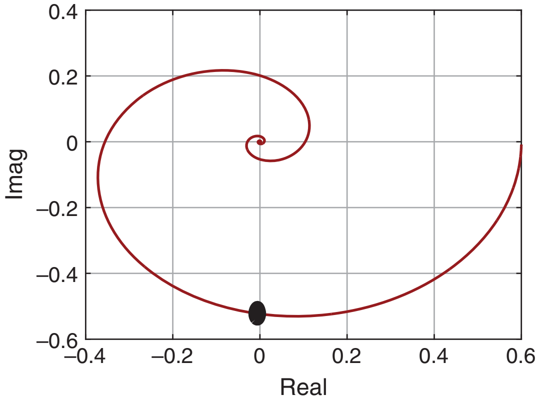


Figure 9.6 Location of ω_1 on a Nyquist curve when using an integrator with relay.

With the integrator incorporated in the relay feedback control, the frequency ω_1 is smaller than the case without the integrator, meaning that the oscillation period $T = \frac{2\pi}{\omega_1}$ is larger. The main purpose of choosing this configuration instead of the basic relay feedback control is to ensure that the plant information obtained is contained at the lower and medium frequency range, as illustrated in Figure 9.6. Additionally, because of the integrator, the effect of high frequency measurement noise is minimized, and the need to use a hysteresis for the noise is reduced. It will be shown in Section 9.6 that the proposed configuration is essential in providing the key information of steady-state gain and dominant time constant in the auto-tuner design.

The relay feedback control with integrator is easy to implement. We assume that the system is operating at the steady state with given values of u_{ss} and y_{ss}. The integral error $e_I(t)$ is defined as

$$e_I(t) = \int_0^t (y_{ss} - y_{act}(\tau))d\tau.$$

With the first order approximation of the derivative $\dot{e}_I(t)$, we obtain at the sampling time t_i,

$$\dot{e}_I(t_i) \approx \frac{e_I(t_i) - e_I(t_{i-1})}{\Delta t} = y_{ss} - y_{act}(t_i) = e(t_i).$$

Thus, the integral error is calculated recursively with

$$e_I(t_i) = e_I(t_{i-1}) + e(t_i)\Delta t.$$

The control law for the relay with integrator is summarized as follows. Here, we assume that the steady-state gain of the system is positive.

$$e_{\mathrm{I}}(t_i) = e_{\mathrm{I}}(t_{i-1}) + e(t_i)\Delta t \tag{9.6}$$

$$u(t_i) = \begin{cases} u(t_{i-1}) & \text{if } |e_{\mathrm{I}}(t_i)| \leq \epsilon \\ a \times \mathrm{sign}(e_{\mathrm{I}}(t_i)) & \text{if } |e_{\mathrm{I}}(t_i)| > \epsilon \end{cases} \tag{9.7}$$

$$u_{\mathrm{act}}(t_i) = u(t_i) + u_{\mathrm{ss}}. \tag{9.8}$$

The following Tutorial is to present a Simulink function for the relay with integrator control law. It is very similar to Tutorial 9.1. With these real time functions, one can convert them into C-code for the real time implementation of the relay feedback control systems.

Tutorial 9.2 *We assume that one has already written the RelayH.slx codes.*

Step by Step

1. *Save the RelayH.slx as RelayI.slx*
2. *The embedded function for the relay with integrator has the following form:*

   ```
   function uCurrent=RelayI(e,Ra,Deltat)
   ```

   where uCurrent is the calculated relay control signal at the sampling time t_i, the first element(e) is the feedback error, and the second element (Ra) is the relay amplitude, and epsilon is the hysteresis used to avoid random switches caused by noise and Deltat is the sampling interval. As before, we will edit the input and output ports. The new parameter is Deltat, which will be added as a parameter in the editing process.
3. *In the following steps, the program will declare those variables that are stored in the embedded function during each iteration for their dimensions and initial values. 'uPast' is the past control signal ($u(t_{i-1})$), and 'eIPast' is the past integrated error signal ($e_I(t_{i-1})$). Enter the following program into the file:*

   ```
   persistent uPast
   if isempty(uPast)
       uPast=Ra;
   end
   persistent eIPast
   if isempty(eIPast)
       eIPast=0;
   end
   ```

4. *Calculate the integrated error. Enter the following program into the file:*

   ```
   eI=eIPast+e*Deltat;
   eIPast=eI;
   ```

5. *Check to see if the integrated feedback error is within the specified hysteresis level ($|e_I(t)| \leq \epsilon$) and if so, the control signal remains unchanged. Otherwise, the relay control signal equals to the sign of the integrated feedback error multiplying the amplitude of the relay, followed by updating the past control signal $u(t_{i-1})$. Here, we set the default value of $\epsilon = 0.001$. Continue entering the following codes into the program.*

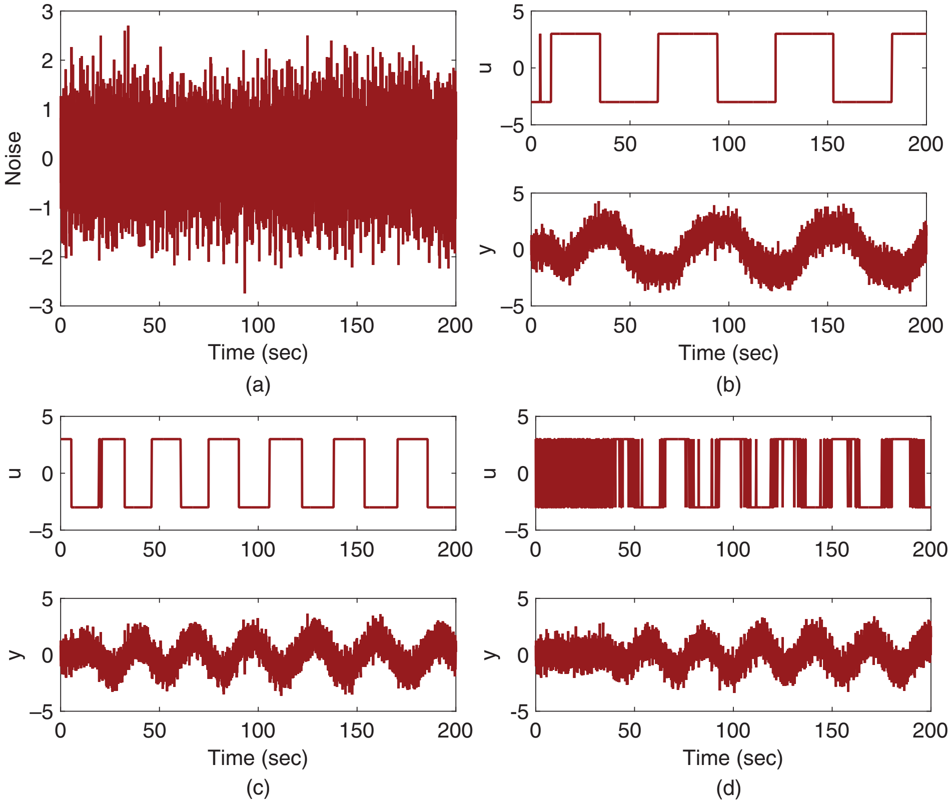
```
if abs(eI)<=0.001
    uCurrent=uPast;
else
   uCurrent=sign(eI)*Ra;
    uPast=uCurrent;
 end
```

6. *Test this program using Simulink simulator built for Example 9.2.*

Example 9.2 *Use of a relay with an integrator controller to generate a sustained oscillation for the following transfer function:*

$$G(s) = \frac{0.6e^{-s}}{(3s+1)^3}.$$

The sampling interval is selected as $\Delta t = 0.02$*, and the amplitude of the relay control is* $a = 3$*. In the closed-loop control simulation, a band-limited white noise with power* 1 *and sampling interval* Δt *is used as shown in Figure 9.7(a). Compare the relay feedback closed-loop responses with and without the integrator.*



(a)

(b)

(c)

(d)

Figure 9.7 Relay feedback control signals with and without integrator (Example 9.2). (a) Noise used in simulation. (b) Relay with integrator. (c) Relay with hysteresis ($\epsilon = 2$). (d) Relay with hysteresis ($\epsilon = 1$).

Solution. *We use the same Simulink configuration as illustrated in Figure 9.3, except replacing the RelayH function by the RelayI function when deploying the relay with integrator controller. Figure 9.7(b) shows the input and output signal for the relay with integrator control system. It is seen that although there is much noise in the system, with hysteresis value almost zero (*$\epsilon = 0.001$*), the input signal does not respond to the measurement noise.*

From Figure 9.7(a) it is seen that the noise amplitude is about 2. Thus, without integrator in the relay control, we choose the hysteresis $\epsilon = 2$*. Figure 9.7(c) shows the closed-loop responses. With* $\epsilon = 2$*, the relay control does not respond to the measurement noise. However, if the parameter* ϵ *is reduced to 1, then the relay control signal is severely affected by the noise, as seen in Figure 9.7(d). Clearly,* $\epsilon = 1$ *is too small for the noise in the system.*

From this example, it is seen that by incorporating a hysteresis element with the correct ϵ *into the relay feedback control, or using the relay with integrator, the correlation of noises between the control input signal and the output signal existing in a general feedback control system is broken, rendering the control input signal noise free in a closed-loop operation. Consequently, system identification under relay feedback control becomes a much simpler task without the noise correlations and complex issues such as identifiability in a closed-loop operation are avoided (Soderstrom et al. (2013), Soderstrom (2018)).*

9.2.3 Food for Thought

1. How do you determine the noise level in the system in order to choose the hysteresis level ϵ?
2. If the noise is severe and you are not sure about the noise level, would you try to use the relay with integrator approach?
3. If the system is first order, relay feedback control will not generate sustained oscillation. Would you consider to incorporate a delay element to the relay feedback control in order to generate a sustained oscillation? How will you define the input and output data in this situation?
4. Instead of using an integrator, will you be able to use a stable transfer function to achieve the same effect? In comparison, what is the key advantage of using an integrator?
5. Is it possible to use relay feedback control to stabilize an unstable system?

9.3 Estimation of Frequency Response using the Fast Fourier Transform (FFT)

It is well known that under a stable relay feedback control, both control input and process output signals are periodic in nature (Astrom and Hagglund (1984), Astrom and Hagglund (1988), Astrom and Hagglund (1995), Astrom and Hagglund (2006), Hagglund and Astrom (1985)). If a single relay experiment is conducted, a standard Fourier analysis (Kreyszig (2006)) shows that the periodic signals contain the frequencies in multiples of the fundamental frequency ω_1 as ω_1, $3\omega_1$, $5\omega_1$, …, $k\omega_1$, where k is an odd number. Because of this, many approaches have been proposed to extract meaningful process information from the set of relay feedback control data, including the describing function analysis (Atherton (1975), Astrom and Hagglund (1984)) and fast Fourier analysis (Wang et al. (2001)).

9.3.1 FFT Estimation

For a period T, the Fourier series expansion of the periodic input signal $u(t)$, is expressed as (see Kreyszig (2006)),

$$u(t) = \frac{4a}{\pi}\left(\sin\frac{2\pi}{T}t + \frac{1}{3}\sin\frac{6\pi}{T}t + \frac{1}{5}\sin\frac{10\pi}{T}t + \dots\right). \tag{9.9}$$

Here, we emphasize that the continuous time fundamental frequency is $\omega_1 = \frac{2\pi}{T}$. From (9.9), it is seen that the construction of the periodic signal $u(t)$ using the sinusoidal components is predominantly dependent on the first few terms because their contributions decay as the frequency increases.

By choosing sampling interval Δt, the discretized input signal $u(t)$ at sampling instant $t_i = i\Delta t$ becomes

$$u(t_i) = \frac{4a}{\pi}\left(\sin\frac{2\pi i}{N} + \frac{1}{3}\sin\frac{6\pi i}{N} + \frac{1}{5}\sin\frac{10\pi i}{N} + \dots\right) \tag{9.10}$$

where $N = \frac{T}{\Delta t}$ is the number of samples within one period. The expression of the discrete time signal $u(t_i)$ reveals the discrete time fundamental frequency as $\omega_\mathrm{d} = \frac{2\pi}{N}$ when we compare (9.10) with (9.9). Therefore, the relationship between the continuous time and discrete time frequencies exists as,

$$\omega_1 = \frac{\omega_\mathrm{d}}{\Delta t}. \tag{9.11}$$

The simplest way to estimate the frequency response of the system under relay feedback is to use the FFT. Assuming that the data length is L, the Fourier transform of an input signal $u(k)$, $k = 1, 2, \dots, L$, is

$$U(n) = \frac{1}{L}\sum_{k=1}^{L} u(k)\mathrm{e}^{-\mathrm{j}\frac{2\pi(k-1)(n-1)}{L}} \tag{9.12}$$

and the corresponding Fourier transform of the output is

$$Y(n) = \frac{1}{L}\sum_{k=1}^{L} y(k)\mathrm{e}^{-\mathrm{j}\frac{2\pi(k-1)(n-1)}{L}} \tag{9.13}$$

where $n = 1, 2, 3, \dots, L$. With the computation of the Fourier transformation of both input and output signals, the estimation of the frequency response of the plant $G(n)$ is simply (Ljung (1999))

$$G(n) = \frac{Y(n)}{U(n)}. \tag{9.14}$$

From both (9.12) and (9.13), with the definition of the Fourier transform, it is seen that the corresponding discrete frequency is defined from 0 to $\frac{2\pi(L-1)}{L}$ with an incremental of $\frac{2\pi}{L}$, where L is the data length.

The remaining tasks are to find which index n corresponds to the fundamental frequency ω_d in discrete time, followed by converting the discrete time frequency to the continuous time frequency ω_1.

An intuitive way to find the fundamental frequency ω_{d} is to find the averaged number of samples of the input signal $u(k)$ within one period [see Figures 9.7(b)–(c)]. However, this could potentially lead to an inaccurate result because of the possible random switches in the input signal due to the measurement noise [see Figure 9.7(d)].

Since the input signal exhibits the feature of a periodic signal, the amplitude of the discrete time Fourier transform of the input signal will yield a maximum value at the fundamental frequency ω_{d}. Thus, we will use the maximum value of the amplitude of the Fourier transform as the signature to identify the discrete time fundamental frequency ω_{d} while taking into account of the incremental frequency $\frac{2\pi}{L}$ in a general Fourier transformation.

9.3.2 MATLAB Tutorial using the FFT for Estimation

The following MATLAB tutorial shows how to estimate $G(\mathrm{j}\omega_1)$, $G(\mathrm{j}\omega_3)$, $G(\mathrm{j}\omega_5)$ and $G(0)$ through Fourier analysis.

Tutorial 9.3 *In this tutorial, we will estimate the plant frequency response at ω_1, ω_3, ω_5 and 0.*

Step by Step

1. *Create a new file called FFTRelay.m for the MATLAB function.*
2. *Define the input and output variables for the MATLAB function. Enter the following program into the file:*

```
function [G1,G3,G5,G0]=FFTRelay(u,y,Deltat)
```

where u and y are the input and output data from the relay feedback control. Deltat is the sampling interval.

3. *Remove the steady-state values of the input and output data to obtain zero mean data. Enter the following program into the file:*

```
L=length(u);
u=u-mean(u);
y=y-mean(y);
```

4. *Calculate Fourier transform of the input and output signal. Enter the following program into the file:*

```
Ujw=fft(u);
Yjw=fft(y);
```

5. *Estimate the frequency response of the plant for all frequencies. Enter the following program into the file:*

```
fftEst=Yjw./Ujw;
```

6. *We find the index that corresponds to the maximum value of the amplitude of the Fourier transform of the input signal, pointing to the fundamental frequency ω_1 and the estimated plant frequency response $G(j\omega_1)$. Enter the following program into the file:*

```
[n1,m1]=max(abs(Ujw));
P1=m1;
w1=2*pi*(P1-1)/(L*Deltat);
G1=fftEst(P1);
```

7. *We will find the second peak in the amplitude of the Fourier transform of the input signal, pointing to the frequency ω_3. Because of the noise, in many cases, $\omega_3 \neq 3\omega_1$. Enter the following program into the file:*

```
[n2,m2]=max(abs(Ujw(P1+3:P1+2*m1+3)));
P3=P1+2+m2;
w3=2*pi*(P3-1)/(L*Deltat);
G3=fftEst(P3);
```

8. *We will locate the third peak in the amplitude of the Fourier transform of the input signal, pointing to the frequency ω_5. Enter the following program into the file:*

```
[n3,m3]=max(abs(Ujw(P3+3:P3+3+2*m1)));
P5=P3+2+m3;
G5=fftEst(P5);
w5=2*pi*(P5-1)/(L*Deltat);
```

9. *We can also estimate the steady-state gain. Enter the following program into the file:*

```
G0=fftEst(1);
```

10. *The estimated frequency responses are exported together with their corresponding frequencies in the following format. Enter the following program into the file:*

```
G1=[G1 w1];
G3=[G3 w3];
G5=[G5 w5];
G0=[G0 0];
```

11. *We can test this program in the Examples 9.3 and 9.4 presented in the next section.*

9.3.3 Monte-Carlo Simulation Studies

In the Monte-Carlo simulation, the measurement noise is generated with a random initial seed that changes with each simulation to reflect the randomness of the noise and its effect on the estimation results. It means that the measurement noise sequence is unique for each simulation and different from others. Consequently, the estimated parameters for each simulation are unique due to the noise. From the total number of simulations, the mean and variance of the estimated parameters are assessed. This assessment can be performed graphically by plotting the estimated parameters for each run against the original parameters or by calculating the mean and variance of the parameters from the total number of simulations. We choose to use a graphic display of the Monte-Carlo simulation results.

Two examples are presented in this section to show the estimation of frequency response of the system using the FFT with relay feedback control data in a noisy environment. The first case is to investigate the estimation when a relay is used with hysteresis to control the system and the second case is to show the estimation when a relay with an integrator is used to control the system.

The system under study has the transfer function

$$G(s) = \frac{0.6e^{-6s}}{(3s+1)^2(2s+1)}.$$

The noise used in the simulation is band limited noise with sampling interval $\Delta t = 0.05$ (s) and power strength 1 together with a gain 0.1. The simulation time is 800 s. The initial seed used to generate the noise sequence varies from 0 to 68 to form the Monte-Carlo simulation studies. In total, we will conduct 69 simulation studies for the estimation.

Example 9.3 *The Simulink simulator is set up using the diagram illustrated in Figure 9.3. In the simulation studies, the hysteresis level ϵ is chosen to be 2 in order to prevent the relay from random switching triggered by the noise. The relay amplitude a is chosen to be 3 . This combination will give the value $\frac{\pi\epsilon}{4a} = 0.5236$ (see Figure 9.2), which will result in sustained oscillation for this system. Because the measurement noise is used in the closed-loop relay feedback control, it affects the switching pattern of the input signal when the relay element is triggered occasionally by random noise. Figure 9.8(a) shows a section of the input and output data generated using seed 68 for the band*

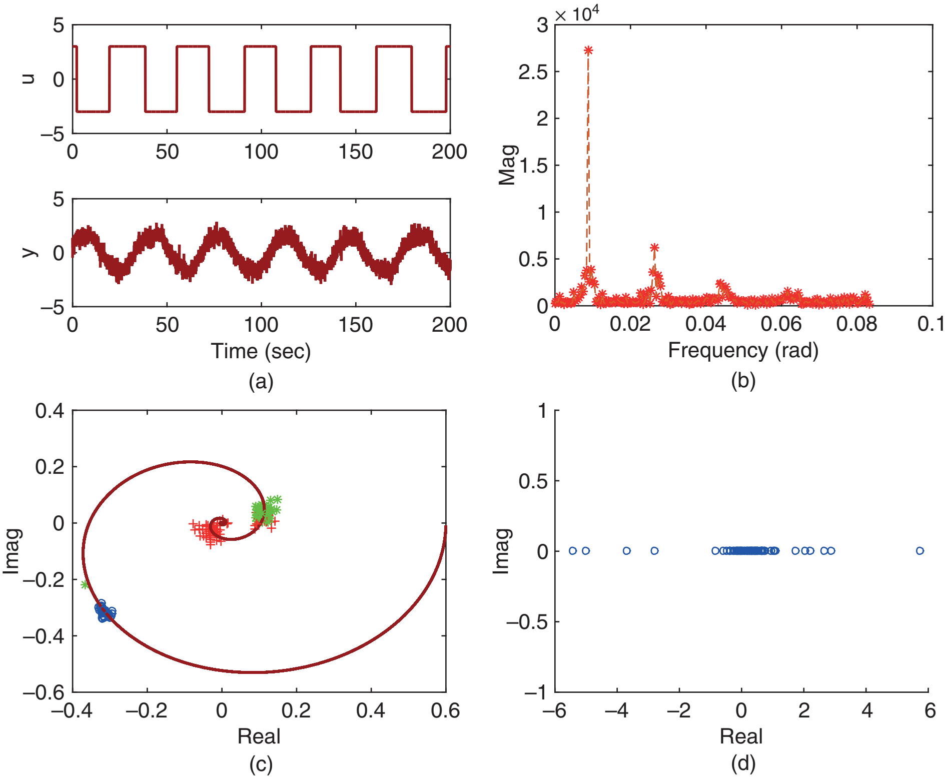


Figure 9.8 Frequency response estimation using FFT (Example 9.3). (a) Input and output data using seed 68. (b) Fourier transform of the input data. (c) Estimated frequency responses (ω_1 (o), ω_3 ($*$), ω_5 (+)). (d) Estimated steady-state gain.

limited noise, and Figure 9.8(b) shows the Fourier transform of the input data. The first 1000 data points were removed from the estimation. Using the program FFTRelay.m, the estimation is performed for the 69 *sets of data generated from the relay feedback control. Figure 9.8(c) shows the estimated frequency points in clusters for* ω_1, ω_3 *and* ω_5 *from the* 69 *estimations. From this figure, it is seen that the estimation for* $G(\omega_1)$ *is reliable because all* 69 *estimates are close to the true frequency response at* ω_1. *For the estimation of* $G(j\omega_3)$, *there is one failure out of the* 69 *experiments, which is quite good. For the estimation of* $G(j\omega_5)$, *there are a few more failed estimations. The estimation for the steady-state gain is not reliable, as shown in Figure 9.8(d).*

It is worthwhile emphasizing that the selection of hysteresis level in relation to amplitude of the relay in a noisy environment is an important task. Without a priori knowledge about the noise level and the gain of the system, their combination is to be revealed often through trial and error experiments.

Example 9.4 *For the relay with integrator control studies, the Simulink simulator is set up using the diagram illustrated in Figure 9.3 with the MATLAB embedded function RelayI.slx produced in Tutorial 9.2.*

For the relay with integrator control, the relay amplitude a is selected to be one and a very small hysteresis level $\epsilon = 0.001$ *is used to avoid the possible numerical problem during the relay feedback control. With the exactly same noise sequences used in Example 9.3, the closed-loop relay feedback control system is simulated for* 69 *different noise sequences. Sustained oscillations are generated for all the experiments. Figure 9.9(a) shows the first segment of the input and output data generated using seed 68 and Figure 9.9(b) shows the Fourier transform of the input data. Figure 9.9(c) presents the estimated results for* $G(j\omega_1)$, $G(j\omega_3)$ *and* $G(j\omega_5)$. *From this figure, it is seen that the estimated* $G(j\omega_1)$ *is very good, which is centered at the true frequency response with a small cluster (i.e. a small variance). There is one failure for the estimation of* $G(j\omega_3)$ *with a larger cluster, also a larger cluster for* $G(j\omega_5)$. *There is no reliable estimation for the steady-state gain, as shown in Figure 9.9(d).*

It is not essential to use hysteresis in the relay with integrator feedback control because the integral action acting as a low-pass filter has prevented the possible random switches caused by the noise. In comparison, the relay with integrator control is much more robust in the presence of noise, as demonstrated in this example. If the system has severe noises, relay with integrator gives a viable solution to generate a sustained oscillation without having to choose a hysteresis level.

9.3.4 Food for Thought

1. Is it correct to say that the periodicity of the input and output signals with relay feedback control plays the key role at obtaining accurate estimated frequency response parameters?
2. Will you still get estimated frequency parameters if the estimated period T is wrong?
3. Why do we use the peak amplitude of the Fourier transform of input signal to identify the period of the oscillation?
4. Why can not we reliably estimate the steady-state gain of the system from the periodic input and output data generated by the relay control?
5. If the input and output data are not zero mean, will you be able to correctly obtain the frequency response estimation using the Fourier analysis?

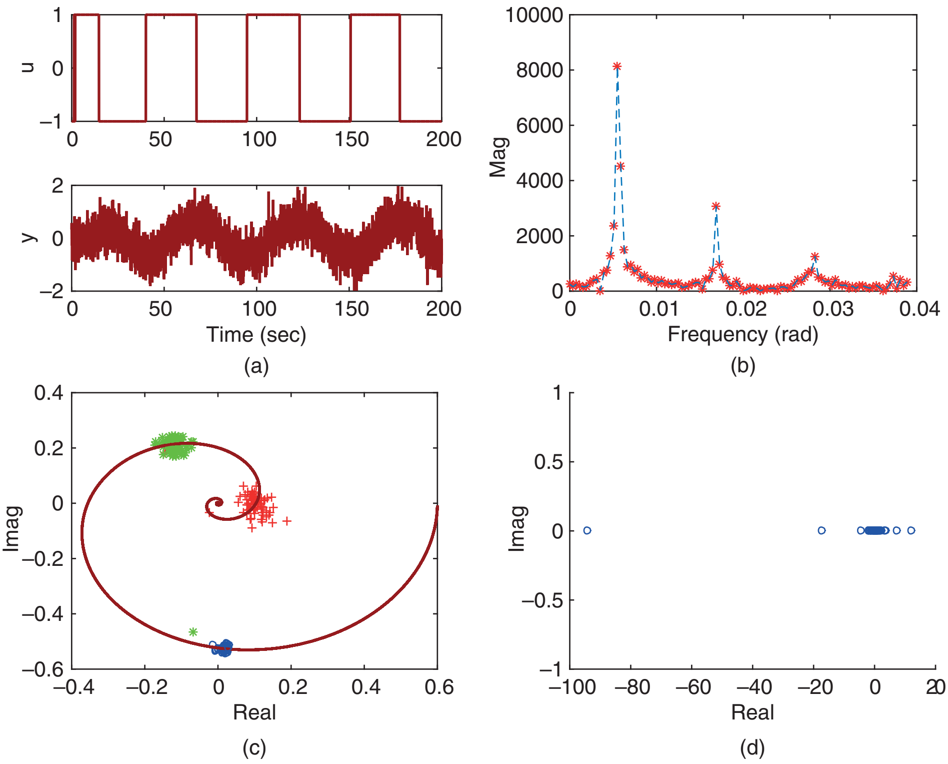


Figure 9.9 Frequency response estimation using FFT (Example 9.4). (a) Input and output data using seed 68. (b) Fourier transform of the input data. (c) Estimated frequency responses ($\omega_1, \omega_3, \omega_5$). (d) Estimated steady-state gain.

9.4 Estimation of Frequency Response Using the frequency sampling filter (FSF)

This section introduces a model based approach for the estimation of process frequency response based on the input and output data collected from relay feedback control. The model is called the FSF model (Bitmead and Anderson (1981),Wang and Cluett (2000)). The advantages of using such a model based approach in comparison with the FFT analysis include that the computation can be performed recursively in real time and the estimated frequency responses have better accuracy because of the model optimization.

9.4.1 Frequency Sampling Filter Model

Assuming that a relay experiment is performed, a set of input and output signals $u(k)$ and $y(k)$ is obtained. Because the input and output signals are periodic signals, the period in number of samples, denoted by N, is calculated and the fundamental discrete frequency is denoted by ω_d, where $\omega_d = \frac{2\pi}{N}$. Also, the backward shift operator q^{-1} is defined as $q^{-1}x(k) = x(k-1)$ where $x(k)$ is a discrete time signal.

Then, associated with the plant frequency response, $G(0)$ and $G(e^{jl\omega_d})$, where $l = 1, 2, 3, \dots, N$, the output $y(k)$ is expressed using the frequency sampling filter model in the relationship to the input signal $u(k)$ as (Bitmead and Anderson (1981),Wang and Cluett (2000)),

$$y(k) = G(0)\ f^0(k) + \sum_{l=-\frac{N-1}{2}}^{\frac{N-1}{2}} G(e^{jl\omega_d})\ f^l(k) + v(k) \tag{9.15}$$

where$f^0(k)$ is the FSF output for the zero frequency, defined as

$$f^0(k) = \frac{1}{N}\frac{1-q^{-N}}{1-q^{-1}}u(k)$$

and$f^l(k)$ is the lth FSF filter output defined as

$$f^l(k) = \frac{1}{N}\frac{1-q^{-N}}{1-e^{jl\omega_d}q^{-1}}u(k)$$

and $v(k)$ is the output measurement noise assumed to be Gaussian distributed with zero mean and variance σ^2. If the input signal $u(k)$ is a perfect period signal with period N, then, from Fourier analysis (Kreyszig (2006)), it only contains the odd number of frequencies and the magnitude decays as the number increases (see (9.10)). As a result, the outputs from the frequency sampling filters with even numbers are zero in response to the relay feedback control signal $u(k)$, and they could be removed from the sum in (9.15). Figure 9.10 shows a block diagram of the frequency sampling filter model with a reduced number of filters. In practice, because of nonlinearities and other imperfect conditions, the relay control signals may not be perfectly periodic signals. The outputs of the zero frequency sampling filter and the even number of frequency filters may have signals with small magnitudes. To avoid bias errors in estimation, the expression of the output signal will take into consideration the effect of the near zero terms, which yields,

$$\begin{aligned} y(k) \approx{} & G(e^{j0})\ f^0(k) + G(e^{j\omega_d})\ f^1(k) + G(e^{-j\omega_d})\ f^{-1}(k) + G(e^{j2\omega_d})\ f^2(k) \\ & + G(e^{-j2\omega_d})\ f^{-2}(k) + G(e^{j3\omega_d})\ f^3(k) + G(e^{-j3\omega_d})\ f^{-3}(k) + v(k). \end{aligned} \tag{9.16}$$

The model in (9.16) with seven complex parameters proves to be adequate for the majority of the applications even in the situation when the relay control does not produce perfect periodic signals.

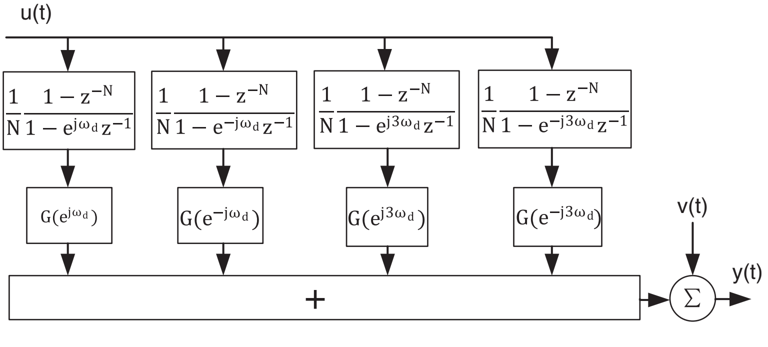

u(t)
$\frac{1}{N}\frac{1-z^{-N}}{1-e^{j\omega_d}z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-e^{-j\omega_d}z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-e^{j3\omega_d}z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-e^{-j3\omega_d}z^{-1}}$
$G(e^{j\omega_d})$
$G(e^{-j\omega_d})$
$G(e^{j3\omega_d})$
$G(e^{-j3\omega_d})$
v(t)
+
Σ
y(t)

Figure 9.10 Block diagram of the frequency sampling filter model with reduced order.

With the description of the output signal in terms of the frequency parameters, the next step is to estimate these parameters using the relay feedback control data. Define the complex parameter vector to be estimated as

$$\theta = [G(0)G(e^{j\omega_d})G(e^{-j\omega_d})G(e^{j2\omega_d})G(e^{-j2\omega_d})G(e^{j3\omega_d})G(e^{-j3\omega_d})]^*$$

and its corresponding regressor vector as

$$\phi(k) = [f^0(k)\ f^1(k)\ f^{-1}(k)\ f^2(k)\ f^{-2}(k)\ f^3(k)\ f^{-3}(k)]^*$$

where A^* denotes the complex conjugate transpose of A.

Assuming that the number of data samples is M, then the least squares estimation of θ has the following analytical solution (Soderstrom and Stoica (1989), Ljung (1999), Soderstrom (2018), Young (2012), Goodwin and Sin (1984)):

$$\hat{\theta} = \left(\sum_{k=1}^{M} \phi(k)\phi(k)^* \right)^{-1} \sum_{k=1}^{M} \phi(k)y(k) \tag{9.17}$$

For real-time computation, we can use a recursive least squares algorithm to compute the frequency parameter vector $\hat{\theta}(k)$ at sampling instant k (see Goodwin and Sin (1984),Young (2012),Ljung (1999)). Here, a standard recursive least squares algorithm is written in the following steps, where the initial conditions of $P(0)$ and $\hat{\theta}(0)$ are selected by the user or they can also be calculated using the relay testing data based on the least squares algorithm (Wang and Cluett (2000)). The following computation steps are repeated in real time, beginning with the sampling instant $k = 1$.

1. Calculate the estimated parameter vector $\hat{\theta}(k)$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k-1)\phi(k)(y(k) - \phi(k)^*\hat{\theta}(k-1)). \tag{9.18}$$

2. Update the covariance matrix $P(k)$ using

$$P(k) = P(k-1) - \frac{P(k-1)^*\phi(k)\phi(k)^*P(k-1)}{1 + \phi(k)^*P(k-1)\phi(k)} \tag{9.19}$$

   where $\hat{\theta}(k)$ contains the estimated frequency response parameters at the sampling instant k.
3. Go back to step one as the next sampling period arrives.

There are two comments given below.

1. It is worthwhile emphasizing that the estimated zero frequency and even frequency parameters are not reliable due to the near zero magnitudes of their filter outputs due to the periodicity of the input signal. This is illustrated by the Monte-Carlo simulation in the examples.
2. It is important to note that in an ideal situation, the parameter N can be determined from the averaged period of the relay control signal. However, this calculation is no longer correct when the relay feedback control does not produce perfectly periodic oscillations. An effective way to calculate N is to find the first maximum magnitude of the Fourier transformation of the input signal and the corresponding frequency ω_d from which $\frac{2\pi}{\omega_d}$ is rounded to the nearest integer to yield the parameter N.

9.4.2 MATLAB Tutorial on Estimation Using the FSF Model

Tutorial 9.4 *In this tutorial, we will estimate the parameter vector $\hat{\theta}(k)$ using recursive least squares. We assume that the entire input and output data set is available. One can adapt the program for real-time data analysis.*

Step by Step

1. *Create a new file called FSFRelay.m for the MATLAB function.*
2. *Define the input and output variables for the MATLAB function with u for the input and y for the output variables. Enter the following program into the file:*

```
function [thetaCurrent,n,N]=FSFRelay(u,y)
```

3. *Remove the steady-state values of the input and output variables. Enter the following program into the file:*

```
L=length(u);
u=u-mean(u);
y=y-mean(y);
```

4. *Determine the averaged period N using FFT. Enter the following program into the file:*

```
Ujw=fft(u);
[n1,m1]=max(abs(Ujw));
 P1=m1;
 N=round(L/(P1-1));
```

5. *Specify the number of frequencies we wish to estimate. Here we select $n = 7$. Enter the following program into the file:*

```
n=7;
```

6. *Set up Fourier coefficient matrix. Enter the following programs into file:*

```
alpha=exp(j*2*pi/N);
beta=1;
for i=1:N-1
beta=[beta alpha^i];
end
zeta=beta.^0;
for i=1:(n-1)/2
zeta=[zeta; beta.^i];
end
```

7. *Use the first N data points to estimate the initial recursive parameters. Enter the following program into file:*

```
U_input=zeros(N,1);
for kk=1:N
Pt=zeta*U_input/N;
Pn=Pt(1);
 for i=2:(n+1)/2
```

```
Pn=[Pn;Pt(i);conj(Pt(i))];
end
Ps(:,kk)=Pn;
U_input=[u(kk);U_input(1:N-1,1)];
end
```

8. *Least squares estimation for the initial $\hat{\theta}(0)$ and covariance matrix $P(0)$. Continue entering the following program into the file:*

```
M=inv(Ps(:,1:N)*Ps(:,1:N)');
thetaPast=M*Ps(:,1:N)*y(1:N);
P_ls=M;
```

9. *Set up forgetting factor for time varying system.*[1] *Enter the following program into the file:*

```
lam=1;
```

10. *Computation of recursive least squares estimation begins. This computation can be adapted for micro-controller applications. Enter the following program into the file:*

```
for k=N+1:length(y)-1;
```

11. *Update FSF outputs at the sampling instant k. Enter the following program into the file:*

```
U_input=[u(k);U_input(1:N-1,1)];
Pt=zeta*U_input/N;
Pn=Pt(1);
 for i=2:(n+1)/2
Pn=[Pn;Pt(i);conj(Pt(i))];
 end
```

12. *Calculate the error $e(k)$ at the current time. Enter the following program into the file:*

```
eCu=y(k)-Pn’*thetaPast;
```

13. *Update the estimated parameter vector $\hat{\theta}(k)$. Enter the following program into the file:*

```
thetaCurrent=thetaPast+P_ls*Pn*eCu/(lam+Pn'*P_ls*Pn);
```

14. *Update the covariance matrix $P(k)$. Enter the following program into the file:*

```
 P_ls=(1/lam)*(P_ls-P_ls*Pn*Pn'*P_ls/(lam+Pn'*P_ls*Pn));
```

15. *Prepare for the next cycle of computation. Enter the following program into the file:*

```
 thetaPast=thetaCurrent;
 end
```

16. *Test this program using the Monte-Carlo simulation in Section 9.4.3.*

[1] One can choose the parameter lam to be less than 1 for time varying systems (see Goodwin and Sin (1984), Young (2012), Ljung (1999)).

9.4.3 Monte-Carlo Simulation using the FSF Estimation

In this section, we will illustrate the estimation of frequency response using the frequency sampling filter model based on a relay with integrator control. The same Monte-Carlo simulation studies as in Example 9.4 will be used to generate the input and output data under relay feedback control.

Example 9.5 *Use the input and output data from the Monte-Carlo simulation studies presented in Example 9.4 to estimate the plant frequency response* $G(j\omega)$ *at* $\omega = 0, \omega_1, 2\omega_1, 3\omega_1$.

Solution. *We will use the program FSFRelay.m produced in Tutorial 9.4 to calculate the estimated frequency response at* $\omega = 0, \omega_1, 2\omega_1, 3\omega_1$, *where* $\omega_1 = \frac{2\pi}{N\Delta t}$. *Because of the noise, the fundamental period N may vary from one experiment to another. Thus, this parameter is determined automatically using FFT as illustrated in the tutorial.*

Figure 9.11 (a) compares the estimated $G(j\omega_1)$ *and* $G(j\omega_3)$ *against the true frequency response* $G(j\omega)$. *It is seen that for these two estimated frequency response parameters, they are indeed very close to the true underlying value. In particular, the 69 estimated values of* $G(j\omega_1)$ *are almost identical to the true value evident by the location and the size of the cluster. The quality of the estimation for the third frequency parameter* $G(j\omega_3)$ *is lower as the size of the cluster is larger, indicating a larger variance associated with the estimation. The estimation for the frequency parameters* $G(0)$ *and* $G(j2\omega_1)$ *is not reliable as the size of the clusters are far too large due to the periodicity of the input and output signals, which inevitably leads to little information contained in the input signals for these two frequency parameters.*

In general, the frequency sampling filter model based estimation leads to an improved estimation compared with FFT based approaches when using the relay feedback control data because it is a model based approach that uses the principle of optimization in the parameter solutions. Additionally, the FSF model based approach can be implemented using a recursive method, as shown in Tutorial 9.4, which is suitable for real time computation on a micro-controller. If necessary, a noise model can be incorporated in the estimation (Wang and Cluett (2000)). The main advantage of using the FFT for the estimation is its simplicity of implementation, as shown in Tutorial 9.3.

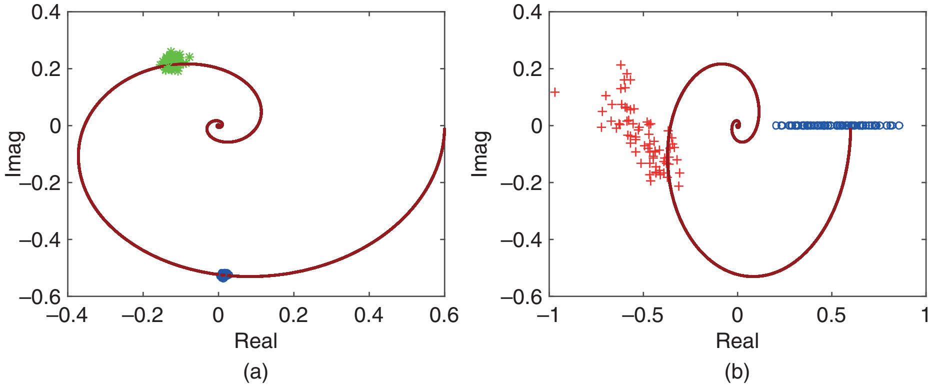


Figure 9.11 Frequency response estimation using FSF (Example 9.5).

9.4.4 Food for Thought

1. In the frequency estimation when using the FSF model and the relay feedback control data, we need to estimate the period in number of samples, N. Why do you think that we identify the parameter N using the maximum peak value of the FFT of the input signal? Can you propose an alternative approach to identify the parameter N?
2. What would happen to the frequency response estimation using FSF if the estimated parameter N is wrong? Even so, would you still be able to get some estimation results?
3. Where are the poles of the frequency sampling filters? Can you express the frequency sampling filter model in terms of real coefficients instead of the complex coefficients?
4. Do you think that the estimated results from the recursive least squares estimation are identical to those obtained from the least squares estimation (see 9.17) with appropriately selected initial conditions?
5. What are the advantages when using the recursive least squares estimation?

9.5 Monte-Carlo Simulation Studies

In this section, we will further evaluate the estimation of the plant frequency response using the frequency sampling filter model in comparison with the estimation using Fourier analysis, where Monte-Carlo simulation studies are used.

The transfer function used in the Monte-Carlo simulation is

$$G(s) = \frac{0.6e^{-6s}}{(3s+1)^2(2s+1)}.$$

The relay amplitude is selected to be 1 and the sampling interval $\Delta t = 0.05$ (s). The relay testing time is selected to be 800 (s). The noise used in the simulation is band limited noise with sampling interval $\Delta t = 0.05$ (s) and power strength 1 together with a gain 0.02. In the Monte-Carlo simulation, the measurement noise is generated with a random seed that changes with each simulation to reflect the randomness of noise and its effect on the estimation results. The seeds used in the simulations are 0, 2, … ,60. Thus, there are 31 random seeds used in the Monte-Carlo simulation to generate the measurement noise. There is a small hysteresis in the proposed relay feedback control because the integral action acting as a low-pass filter has reduced the possible random switches caused by the noise.

9.5.1 Effect of Unknown Constant Disturbance

In many applications, during the relay experiments, there is an unknown constant disturbance. This type of disturbance often enters the system at the input variable, which is called the input disturbance. A typical example is the electrical load in an AC motor (Wang et al. (2015)). This type of disturbances will cause periodic oscillations in the relay feedback control to become unbalanced.

A constant input disturbance with magnitude of 0.3 is added to the relay feedback control experiments. Figure 9.12(a) shows the closed-loop relay feedback control responses in the presence of measurement noise and the input disturbance. From this figure it is seen that with the disturbance the oscillations are no longer in symmetry. Fourier analysis reveals [see Figure 9.12(b)] that the input signal has the fundamental frequency

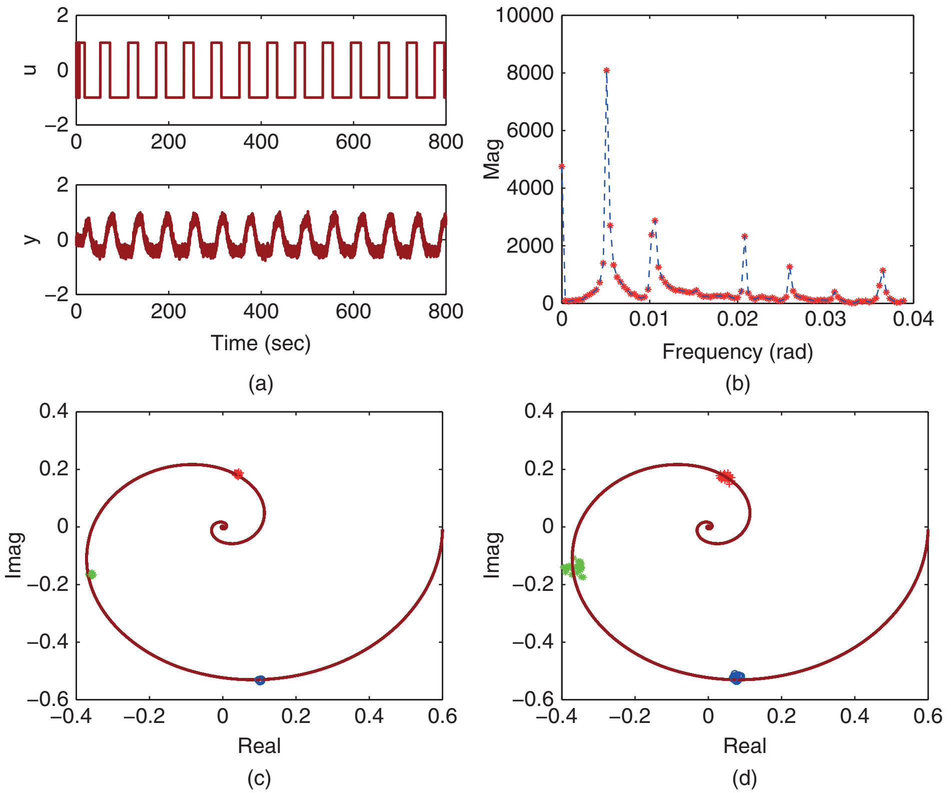


Figure 9.12 Monte-Carlo simulation results with 31 random seeds in the presence of constant load disturbance. (a) Input and output signal. (b) Magnitude of DFT. (c) FSF estimation. (d) FFT estimation. Key: $G(j\omega)$ (solid line), o is the estimated values at $\omega_1 = \frac{2\pi}{N\Delta t}$, $*$ is the estimated values at $\omega_2 = 2\omega_1$, + is the estimated values at $\omega_4 = 4\omega_1$.

$\omega_\mathrm{d} = \frac{2\pi}{N} = 0.0051$ (rad), where $N = 1231$ and the next significant frequency in the periodic signal is $2\omega_\mathrm{d}$ followed by $4\omega_\mathrm{d}$. By choosing the nine frequency parameters ($n = 9$) and $N = 1231$ in the frequency sampling filter model, we obtain the estimated frequency parameters as shown in Figure 9.12(c). It is seen that the estimated parameters are unbiased with small variances, as shown through the Monte-Carlo simulations. In comparison, Figure 9.12(d) shows the estimated parameters using Fourier analysis, which are seen to have larger variances. The estimation results for frequency parameters at 0 and $3\omega_\mathrm{d}$ are not consistent when the oscillations are not symmetric caused by the constant disturbance.

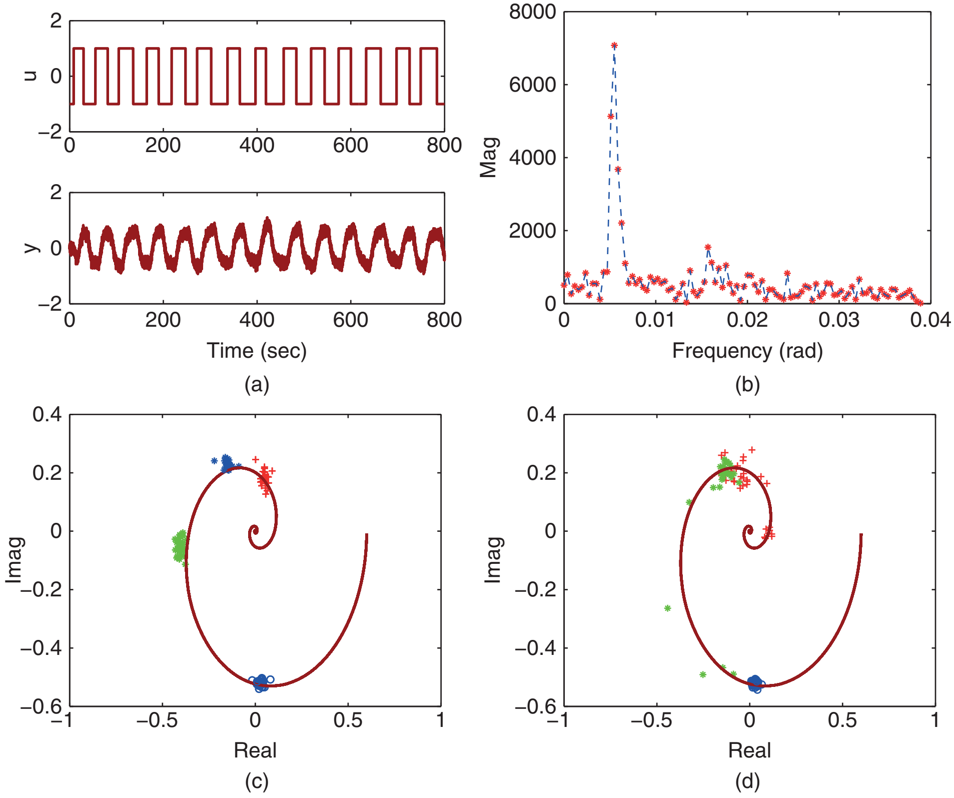
9.5.2 Effect of Unknown Low Frequency Disturbance

In the application of process control, low frequency disturbances are frequently encountered. This group of studies will investigate what would happen to the relay feedback control and to the frequency response estimation in the presence of unknown low frequency disturbance.

In the Monte-Carlo simulation studies, the low frequency disturbance, entering into the system via the input signal, is generated by passing the same band limited measurement noise through a first order filter with transfer function:

$$G_{\mathrm{d}}(s) = \frac{0.05}{s + 0.05}$$

which is then sampled using the same sampling interval $\Delta t = 0.05$ (s). Note that the disturbance model has a time constant of 20 (s), which is much larger than the time constants of the system. Figure 9.13(a) shows the input and output data generated by the relay feedback control system. From this figure, it is interesting to see that the relay feedback control system no longer generates periodic oscillations in the presence of low frequency disturbances. This is confirmed through the Fourier analysis [see Figure 9.13(b)], from which only one spike in the magnitude of the Fourier transformation is observed. It is important to note that the relay feedback control system produces the oscillations that have the largest amplitude at $\omega_{\mathrm{d}} = 0.0055$, where $N = 1143 \approx \frac{2\pi}{\omega_{\mathrm{d}}}$. Figures 9.13(c) and (d) compare the estimation results obtained using the frequency sampling filter model and the Fourier analysis of the input and output



(a) (b) (c) (d)

Figure 9.13 Monte-Carlo simulation results with 31 random seeds in the presence of unknown low frequency disturbance. (a) Input and output signal. (b) Magnitude of DFT. (c) FSF estimation. (d) FFT estimation. Key: $G(j\omega)$ (solid line), o is the estimated values at $\omega_1 = \frac{2\pi}{N\Delta t}$, $*$ is the estimated values at $\omega_2 = 2\omega_1$, + is the estimated values at $\omega_4 = 4\omega_1$.

data. It is seen from Figure 9.13(c) that the variances of the estimated parameters using the frequency sampling filter model are relatively small; however, there are small biases in the estimated parameters. In contrast, because the input signal is not periodic, the frequency response estimation using Fourier analysis produces poor results except the estimated parameters at $\omega_d = 0.0055$, as shown in Monte-Carlo simulation studies, which is evident by the scatters on the complex plane [see Figure 9.13(d)].

9.5.3 Estimation of the Steady-state Value

The final question to be answered in the Monte-Carlo simulation studies is whether the steady-state estimation will be valid using the frequency sampling filter model. Figure 9.14 shows that the estimation of steady-state information is not reliable when using relay feedback control. Depending on the particular seed of the noise generator, all the estimated steady-state gains vary significantly from the true value (0.6). The worst case is shown in Figure 9.14(b), where with the constant disturbance the steady-state gain estimated is 0.

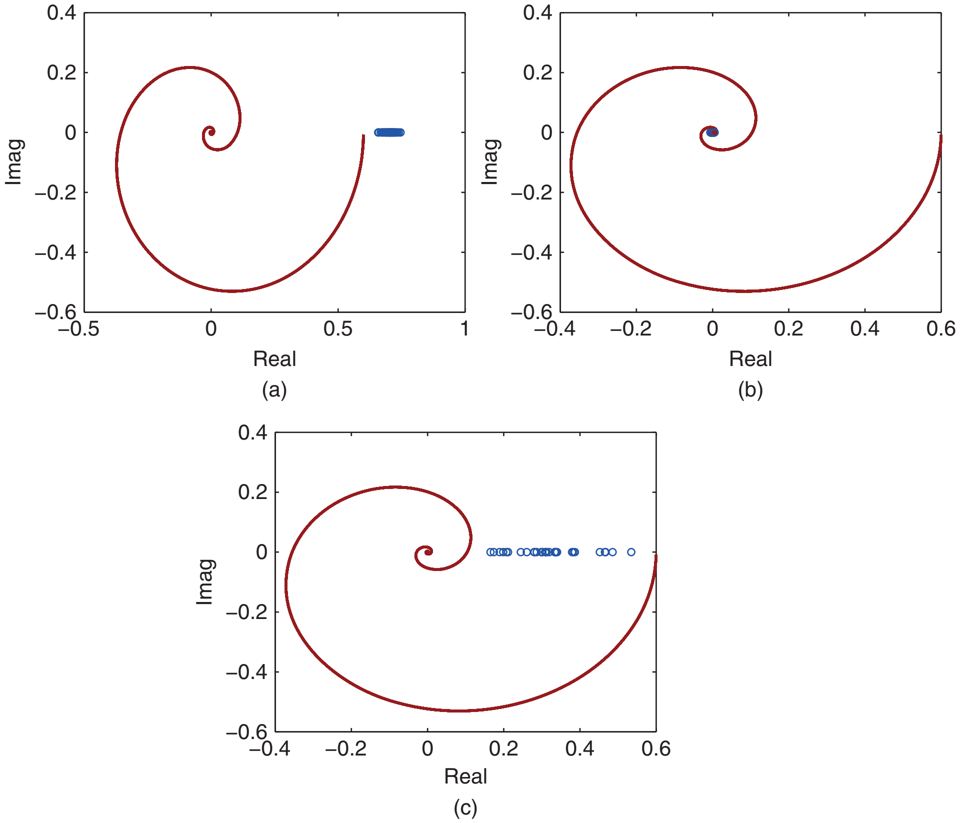


Figure 9.14 Monte-Carlo simulation results for estimation of steady-state gain with 31 random seeds. (a) Steady-state estimation (measurement noise). (b) Steady-state estimation (constant disturbance). (c) Steady-state estimation (low frequency disturbance). Key: $G(j\omega)$ (solid line), o is the estimated values at $\omega = 0$.

9.5.4 Food for Thought

1. The Monte-Carlo simulation studies presented in this section are based on integrator with relay feedback control. In the presence of constant disturbance, what is the role of the integrator played in the relay feedback control system? What are your observations in terms of the behavior of input and output signals?
2. In the presence of low frequency disturbance, has the integrator with relay feedback control tried to compensate the disturbance? Does this cause the breakdown of the periodicity of the input and output signals?
3. When there is a constant disturbance in the system, what kind of behavior would you expect from the input and output data for a relay with hysteresis? Do you expect accurate estimation results for this case when using FFT and FSF algorithms?
4. Why can not we consistently estimate the steady-state gain of the system using the data generated from the relay feedback control? Even for the case with low frequency disturbance?

9.6 Auto-tuner Design for Stable Plant

To design the auto-tuner for a stable plant, the integrator with relay feedback control is used to generate the input and output data for the frequency response identification because this configuration handles the measurement noise and disturbances well and also provides the valuable low frequency information for the PID controller design. We assume that the two estimated frequency response points from the relay testing data are the fundamental frequency $G_p(j\omega_1)$, $\omega_1 = \frac{2\pi}{N\Delta t}$ and $G_p(j\omega_2)$, where N is the number of samples within one period. Because of noise, the oscillation generated from the relay feedback control may not be perfectly periodic. Thus, the parameter N is estimated using FFT analysis as illustrated in the previous sections (see Section 9.4). In the majority of the cases, the second frequency ω_2 is selected as $3\omega_1$.

We assume that the plant transfer function $G_p(s)$ is stable with all poles strictly on the left-half complex plane. Recall the desired closed-loop performance for the PID controller design using frequency response data in Chapter 8, which is specified through the control sensitivity function (see (8.18)), as

$$S_u(s) = \frac{1}{K_p}\frac{\frac{1}{\beta}\tau_{cl}s + 1}{\tau_{cl}s + 1} \tag{9.20}$$

where $\tau_{cl} > 0$ is the desired closed-loop time constant, the parameter β is selected so that $\frac{\tau_{cl}}{\beta} = \tau_{op}$ is approximately equal to the dominant time constant of the system, K_p is the steady-state gain of the system.

To link the PID controller design method to the auto-tuner, the question remains as how to choose the design parameters in (9.20). One possibility that works well is to derive the approximate dominant time constant of the system using the period of relay control, N. For the majority of the system, it can be readily verified through simulation studies that the settling time of the system is roughly $0.5N\Delta t$, namely half of the period under integrated relay control. With this knowledge, the dominant time constant of the process is selected about one-fifth of the settling time as

$$\tau_{op} = \frac{0.5N\Delta t}{5} = 0.1N\Delta t \tag{9.21}$$

Now, one can choose the desired closed-loop time constant τ_{cl} in relation to the dominant time constant of the process via the scaling parameter $\beta > 0$ by defining

$$\tau_{cl} = \beta\tau_{op} = 0.1\beta N\Delta t \tag{9.22}$$

Substituting (9.21) and (9.22) into (9.20) leads to the desired control sensitivity function as,

$$S_u(s) = \frac{1}{K_p}\frac{0.1N\Delta ts + 1}{0.1\beta N\Delta ts + 1} \tag{9.23}$$

hence, the desired closed-loop transfer function as

$$T_d(s) = \frac{1}{K_p}\frac{0.1N\Delta ts + 1}{0.1\beta N\Delta ts + 1}G_p(s) \tag{9.24}$$

Note that in (9.24) the parameters N and Δt are known from the relay experiment and the user selected performance parameter is β, which is the scaling factor between the estimated open-loop and the desired closed-loop time constants. This modification of the closed-loop performance specification leads to the automatic calculation of the PID controller parameters with minimum effort from the user.

Because the direct estimation of the steady-state gain is not consistent using the relay control experiment, an approximation is taken to yield a rough estimation of the gain as

$$K_p \approx sign(K_{ss})|G_p(j\omega_1)|$$

where K_{ss} is the unknown steady-state gain of the system.

9.6.1 MATLAB Tutorial on Auto-tuner for Stable Plant

Tutorial 9.5 *The objective of this tutorial is to implement the auto-tuner in a simulation environment. The system to be controlled has the following transfer function*

$$G_p(s) = \frac{0.6(-6s+1)e^{-s}}{(2s+1)(3s+1)}$$

Step by Step

1. *Build a Simulink simulation program Tuner4StableSys.slx as shown in Figure 9.15, where the output data ports are the relay feedback error e, the control signal u and the output signal y. The steady-state values u_{ss} and y_{ss} are set to zero. Inside the Configuration Parameters, the start time is selected as* 0.0*, stop time as Tsim, Solver option: fixed-step and ode4 (Runge-Kutta), Fixed-step size: Deltat. In the Simulink simulator, we have used the relay with integrator controller built in Tutorial 9.2. We can add noise and disturbance to the simulator for the different cases.*
2. *Create a new MATLAB file called Test4tuner.m and save it in the same directory as the Simulink simulator.*
3. *Define the system parameters using numerator, denominator and time delay d. Enter the following program into the MATLAB file:*

```
num=0.6*[-6 1];
den=conv([3 1],[2 1]);
d=1;
```

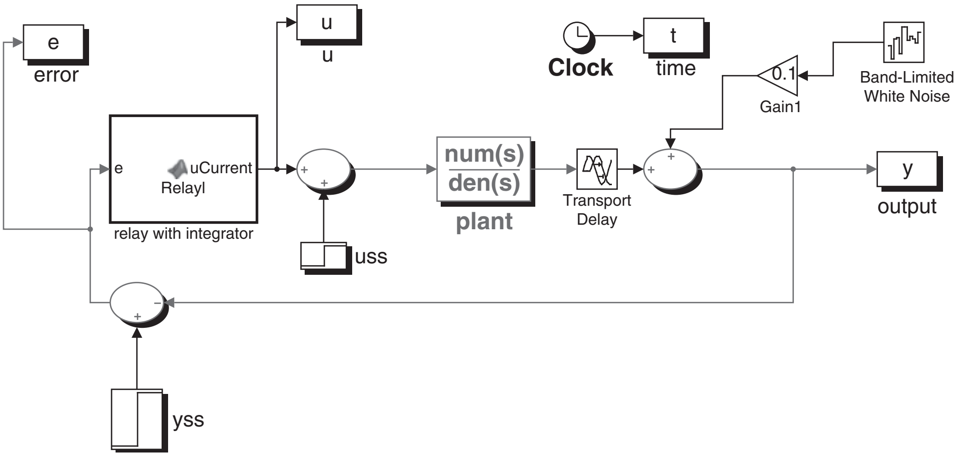


Figure 9.15 Simulink diagram of auto-tuner for stable system.

4. *Define the relay amplitude and hysteresis level, simulation time T_{sim} and sampling interval Δt. Continue entering the following program into the MATLAB file:*

```
Ra=1;  %relay amplitude
epsilon=0.001;  % hysteresis
Deltat=0.05;
Tsim=400;
Nsim=Tsim/Deltat;
```

5. *Call the Simulink program to generate the three data output sets u,y and e. Continue entering the following program into the MATLAB file:*

```
sim('Tuner4StableSys')
```

6. *Call the FSF estimation program created in Tutorial 9.4. Continue entering the following program into the file:*

```
[thetaCurrent,n,N]=FSFRelay(u,-e);
```

7. *Define the estimated frequency parameters. Continue entering the following program into the file:*

```
w1=2*pi/(N*Deltat);
w2=6*pi/(N*Deltat);
Gjw1=thetaCurrent(3);
Gjw2=thetaCurrent(7);
```

8. *Find the estimated steady-state gain in approximation. Continue entering the following program into the file:*

```
K=abs(Gjw1);
```

9. *Define the ratio β ($\beta = 1$ as an example) between desired closed-loop time constant in relation to the open-loop time constant, and the desired closed-loop time constant. Continue entering the following program into the file:*

```
beta=1;
taucl=beta*0.1*N*Deltat;
```

10. *Call the FR4PID.m from Tutorial 8.1 to calculate the PID controller parameters. Continue entering the following program into the file:*

```
[Kc,tauI,tauD]=FR4PID(beta,taucl,w1,w2,Gjw1,Gjw2,K);
```

11. *Test this program with the closed-loop simulation as demonstrated in the next section and compare with results from Case B in the next section.*

9.6.2 Evaluation of the Auto-tuner for a Stable Plant

In this section, we will firstly evaluate the closed-loop PID control performance of the proposed auto-tuner using four classes of systems commonly encountered in process control, and secondly perform comparative studies. For simplicity of the presentations, only measurement noise is added to the relay experiments. It has been verified that the auto-tuner will provide similar closed-loop performance when the constant disturbance or the low frequency disturbance are introduced during the relay experiments as long as we make an adjustment on the choice of the second frequency estimate for the PID controller design (see Sections 9.5.1 and 9.5.2).

There are four classes of systems used to evaluate the auto-tuner. Time delays are added for all testing cases to reflect the cases in process control. The same steady-state gain is used in all four cases for consistency in signal-to-noise ratio and the scaling of disturbance rejection in the closed-loop simulation. The sampling interval is $\Delta t = 0.05$, the relay amplitude is selected to be 1, and a measurement noise with standard deviation $\sigma = 0.02$ was added to the output. After the relay experiment, the number of samples for the period of the relay signal is calculated automatically to give the parameter N, and the frequency responses at $\omega_1 = \frac{2\pi}{N\Delta t}$ and $3\omega_1$ are estimated. Then, by choosing $\beta = 0.5$, 1, and 2, three sets of PID controller parameters are calculated using the program FR4PID.m.

Case A. This is a testing case for high order systems. The transfer function of this system is given as

$$G_p(s) = \frac{0.6e^{-s}}{(3s+1)^6}.$$

Case B. This is the testing case for systems with non-minimum phase behavior. The transfer function is given by

$$G_p(s) = \frac{0.6(-6s+1)e^{-s}}{(2s+1)(3s+1)}.$$

Case C. This is the testing case for systems with dominant time delay. The transfer function is given as

$$G_p(s) = \frac{0.6e^{-10s}}{3s+1}.$$

Case D. The final testing case uses an underdamped system, which has the transfer function

$$G_p(s) = \frac{0.6e^{-s}}{(9s^2+2.4s+1)(3s+1)}.$$

This transfer function has the damping coefficient $\xi = \frac{2.4}{6} = 0.4$. Thus, the system has oscillatory response in open-loop operation.

Relay feedback control is used to generate the input and output data for all four processes in the presence of measurement noise. Because of the integral action used in the relay feedback control, the measurement noise has not affected the input signal without using hysteresis in the design, and there is no correlation between the measurement noise and the input signal. This adds to the benefit of using an integrator in series with a relay control. The estimation of the frequency responses at $\omega_1 = \frac{2\pi}{N\Delta t}$ and $3\omega_1$ is performed using the frequency sampling filter model and the estimated frequency responses are accurate for all four cases.

9.6.2.1 PID Controller Parameters

By choosing $\beta = 0.5$, 1, and 2, three sets of PID controller parameters are calculated, which corresponds to the selection of three different desired closed-loop response speed. For instance, when $\beta = 0.5$, the desired closed-loop time constant is selected equivalent to half of the estimated dominant open-loop time constant. Table 9.1 shows the PID controller parameters of the four cases for the three β values together with the mean squared errors, where the mean squared error is defined as $\frac{1}{M}\sum_1^M e(t_i)^2$, M being the number of samples and $e(t_i)$ being the error between the reference and output signals.

It is noticed that as β increases, the proportional controller gain K_c reduces, and the derivative gain reduces. In contrast, the variation of integral gain is smaller when β changes.

9.6.2.2 Nyquist Plots

Figures 9.16(a)–(d) show the Nyquist plots of the four cases for the three control systems designed using $\beta = 0.5$, 1 and 2, where the frequency response of the system, $G(j\omega)$, is used in the computation. It is seen from the Nyquist plots that the PID controllers with all three choices of β lead to stable closed-loop systems. However, for $\beta = 0.5$, the

Table 9.1 PID controller parameters for different β values.

Case	β	K_c	τ_I	τ_D	$\frac{1}{M}\sum_1^M e(t_i)^2$
A	0.5	1.2849	9.0098	5.0778	0.0894
	1	0.9992	9.5168	3.0527	0.0971
	2	0.6829	9.7083	1.1979	0.1171
B	0.5	0.9479	6.2732	1.1641	0.1285
	1	0.7265	5.9704	0.9729	0.1033
	2	0.4911	5.6412	0.7095	0.1072
C	0.5	1.1366	6.0222	3.8558	0.0805
	1	0.8841	5.9116	3.2651	0.0793
	2	0.6044	6.0983	1.8358	0.0902
D	0.5	0.9885	3.2625	2.1823	0.0417
	1	0.7693	3.7356	0.9428	0.0450
	2	0.5261	3.8386	0.2066	0.0543

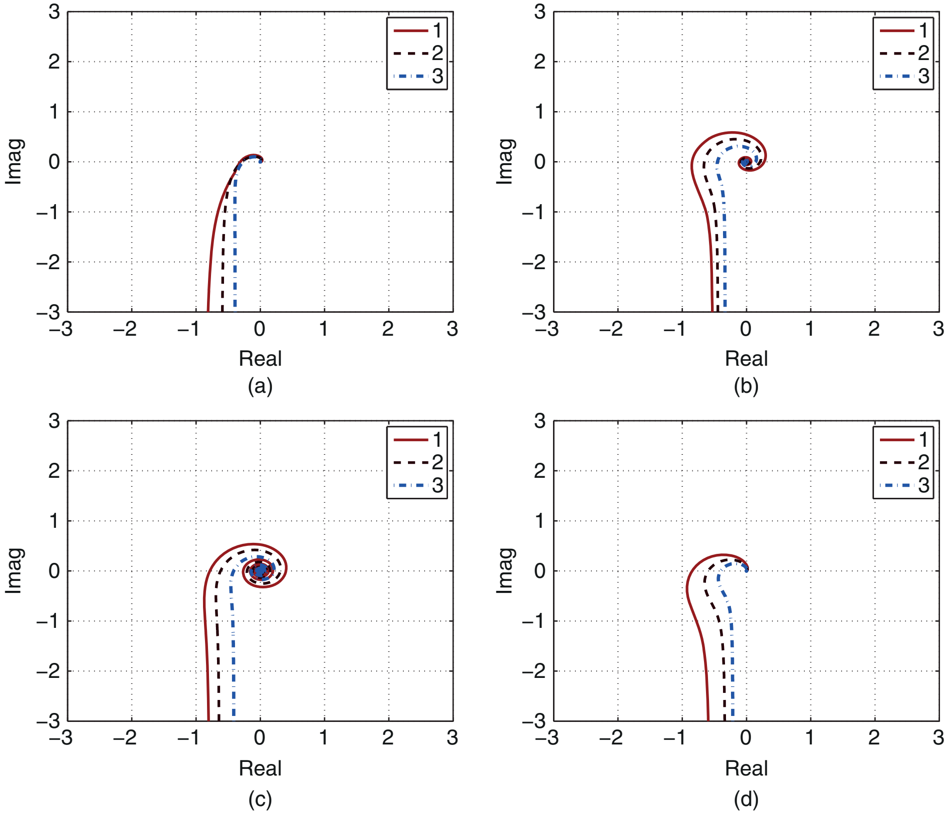


Figure 9.16 Nyquist plots using the PID controller parameters in Table 9.1. (a) Case A. (b) Case B. (c) Case C. (d) Case D. Key: line (1) $\beta = 0.5$; line (2) $\beta = 1$; line (3) $\beta = 2$.

closed-loop control systems for case B to case D had gain margins less than 2. As β increases, both gain and phase margins for all four systems have increased. In particular, for the default choice of $\beta = 1$, the gain margins for all four case are greater than 2 and phase margins are greater than 45°.

9.6.2.3 Closed-loop Simulation Results

Closed-loop simulation is conducted for all four cases with $\beta = 0.5$, 1, and 2. In the closed-loop simulation, the derivative term is implemented on the output only and a derivative filter with time constant $\frac{1}{0.1\tau_D s+1}$ is used. A unit step signal is used as the reference signal at $t = 0$ (s), and a unit step input disturbance enters at time $t = 100$ (s). Figures 9.17(a)–(d) show the closed-loop simulation results for reference following and disturbance rejection. It is seen that all closed-loop systems are stable. The simulation results demonstrate that for a faster response to the reference signal, the PID control system will also have a faster response to disturbance rejection. The user can select the scaling parameter β to achieve the desired closed-loop response as required. As β increases, the closed-loop response speed reduces to both reference following and disturbance rejection. The overshoot in the reference response when β is small could be overcome by using a two degree of freedom PID controller implementation as shown in Chapters 1 and 2.

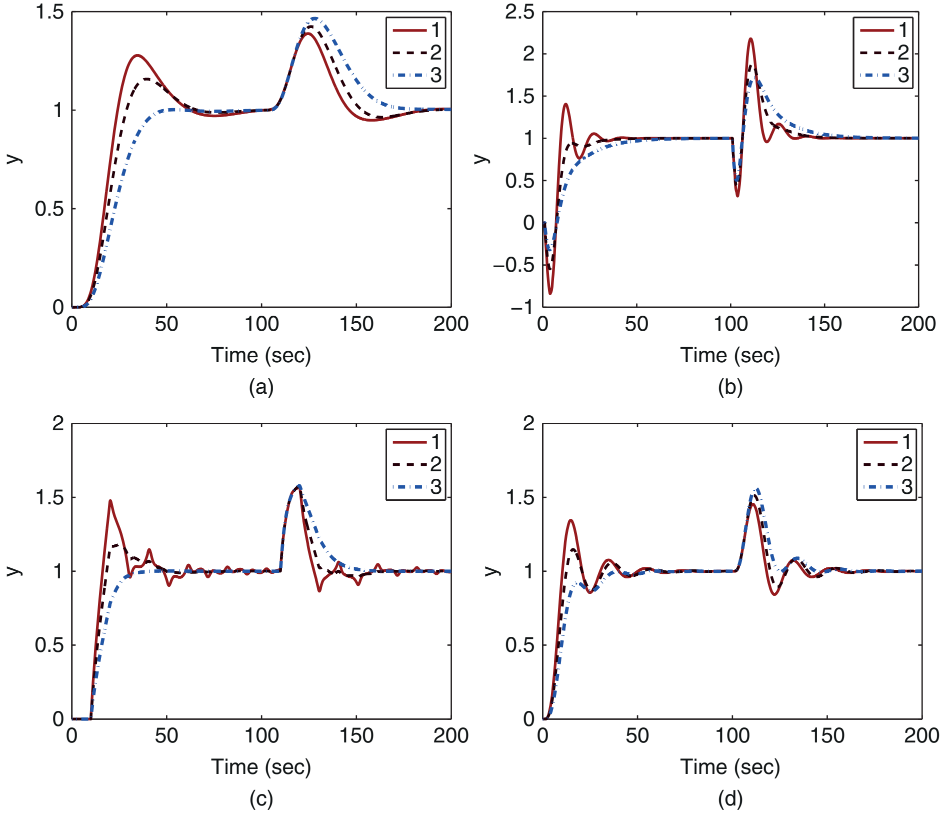


Figure 9.17 Closed-loop simulation results using the PID controller parameters in Table 9.1. (a) Case A. (b) Case B. (c) Case C. (d) Case D. Key: line (1) $\beta = 0.5$; line (2) $\beta = 1$; line (3) $\beta = 2$.

9.6.3 Comparative Studies

In this section, we will benchmark the closed-loop performance of the PID controller found by the auto-tuner against the performance of the several other well known PID controllers. We will consider the following first order plus time delay model,

$$G(s) = \frac{e^{-50s}}{10s + 1}. \tag{9.25}$$

In the comparative studies, we will calculate the PID controller parameters using the IMC-PID controller design proposed by Rivera et al. (1986) (see Section 1.4.1), and the PID controller design by Padula and Visioli (2011) (see Section 1.4.2).

As in Chapter 1, the PID controller parameters using IMC-PID controller and Padula and Visioli tuning rules are calculated using the transfer function model (9.25). The PID controller parameters from the auto-tuner are calculated from the relay experiment where measurement noise with standard deviation of 0.02 was added.

In Table 9.2, we will choose two β parameters to yield two sets of PID controller parameters from the auto-tuner. With this selection, it seems that all three PID controllers have a similar gain for the proportional control. However, the integral time constant varies between 26 and 35, and the derivative gain varies between 7 and 17. Figures 9.18(a)–(b) compare the three control signals and output signals in

Table 9.2 PID controller parameters and mean squared errors from the control simulation studies. Cases A and B use an auto-tuner, case C uses IMC-PID, cases D and E use Padula and Visioli PID.

Case	Spec.	K_c	τ_I	τ_D	$\frac{1}{M}\sum_1^M e(t_i)^2$
A	$\beta = 1$	0.5137	27.9241	14.4603	0.1326
B	$\beta = 2.5$	0.2982	28.3523	7.4440	0.1671
C		0.538	35.00	7.14	0.1358
D	$M_s = 1.4$	0.1664	17.2991	11.9938	0.1923
E	$M_s = 2$	0.3205	22.5160	13.5770	0.1504

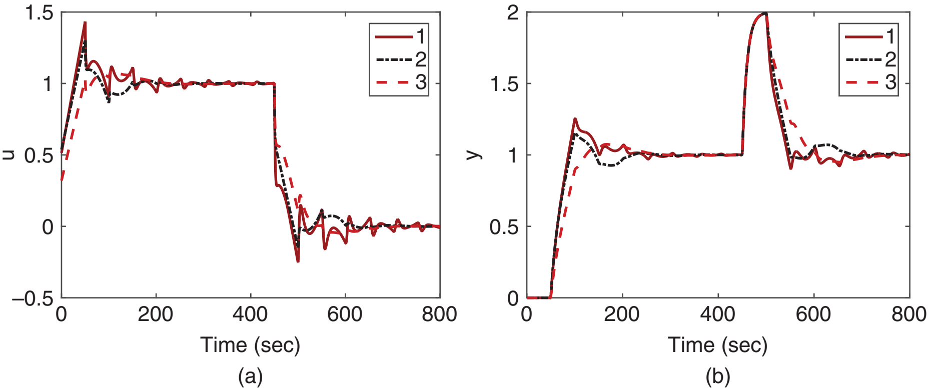


Figure 9.18 Closed-loop simulation results using the PID controller parameters in Table 9.2. (a) Control signal. (b) Output signal. Key: line (1) auto-tuner ($\beta = 1$); line (2) IMC-PID controller; line (3) Padula–Visioli design ($M_s = 2$).

both reference following and disturbance rejection, which show that all three control systems have similar closed-loop performances in reference following and disturbance rejection.

9.6.4 Food for Thought

1. What are the key advantages when using the estimated frequency information directly for the auto-tuner design?
2. Can you propose an approach to convert the estimated frequency points to first order plus delay model, then design PID controller using the Padula and Visioli tuning rules given in Chapter 1?
3. If you wish to have a faster closed-loop response speed for disturbance rejection and reference following, should you increase or decrease the parameter β?
4. If the system has a large measurement noise, and you wish not to amplify it, should you increase or decrease the parameter β?
5. If the auto-tuner finds the derivative time constant to be negative, should we simply neglect the derivative term and implement the PI controller?

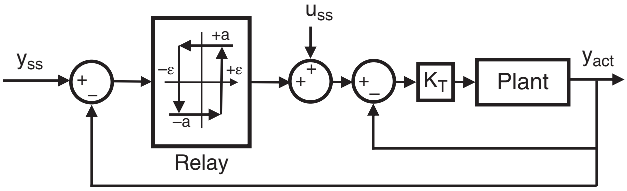


Figure 9.19 Block diagram of relay feedback control for an integrating system.

9.7 Auto-tuner Design for a Plant with an Integrator

For systems containing an integrator, in the design of auto-tuner for PID controllers, a proportional controller K_T is first used to stabilize the plant. A relay with hysteresis element is utilized to generate the sustained oscillation for the closed-loop control system. The block diagram of the relay feedback control for an integrating system is illustrated in Figure 9.19.

9.7.1 Estimation of an Integrating Plus Delay Model

The approximate model of an integrating system is assumed to be of the following form:

$$G(s) = \frac{K_p e^{-ds}}{s}. \tag{9.26}$$

For most physical systems, there are more or less approximations involved in obtaining the integrating plus time delay model. For an integrating plus time delay system, a single frequency is sufficient to determine its gain K_p and time delay d. Therefore, it is exceedingly simple to obtain an integrating plus delay model using the relay feedback control experimental data.

As shown in Section 9.4, the estimation of the closed-loop frequency response using either FFT analysis or an FSF model will yield the information $\hat{T}(j\omega_1)$ where $\hat{T}(j\omega_1)$ is the estimated closed-loop frequency response and ω_1 is the fundamental frequency of the relay feedback control.

With the knowledge of the proportional controller K_T, the frequency response of the plant $G(e^{j\omega_1})$ is calculated from the closed-loop frequency response relationship,

$$\hat{T}(e^{j\omega_1}) = \frac{G(e^{j\omega_1})K_T}{1 + G(e^{j\omega_1})K_T}$$

leading to

$$G(e^{j\omega_1}) = \frac{1}{K_T} \frac{\hat{T}(e^{j\omega_1})}{1 - \hat{T}(e^{j\omega_1})}. \tag{9.27}$$

Now, letting the frequency response of the integrating plus delay model (9.26) be equal to the estimated $G(j\omega_1)$ leads to

$$\frac{K_p e^{-jd\omega_1}}{j\omega_1} = G(j\omega_1). \tag{9.28}$$

Equating the magnitudes on both side of (9.28) gives

$$K_p = \omega_1 |G(j\omega_1)| \tag{9.29}$$

where $|e^{-jd\omega_1}| = 1$. Additionally, from (9.28), the following relationship holds:

$$e^{-jd\omega_1} = \frac{j\omega_1 G(j\omega_1)}{K_p}.$$

This gives the estimate of time delay as

$$d = -\frac{1}{\omega_1}\tan^{-1}\frac{\text{Imag}(jG(j\omega_1))}{\text{Real}(jG(j\omega_1))}. \tag{9.30}$$

In the event that parameter d is negative, its absolute value is taken as the estimated time delay.

It is seen here that if the system is approximated by integrating with the time delay, the plant information at a single frequency is sufficient to determine the plant gain and time delay.

9.7.2 Auto-tuner for Integrating Systems

On obtaining the estimated integrating plus time delay model (9.26), one can find the PID controller using the empirical rules such as the modified IMC-PI controller in Skogestad (2003), which was discussed in Section 1.4.1, or tuning rules in Tyreus and Luyben (1992). We will use the empirical rules presented in Section 8.4.3, which gives the flexibility of PI, PID and PD controllers together with the gain margin and phase margin for the selection of performance parameter β.

One is encouraged to follow Tutorial 9.1 for the relay feedback control and Tutorial 9.4 for the estimation of the frequency response using an FSF so as to validate the following simulation example.

The following example is to show the behaviour of the auto-tuner when it is applied to a complex integrating system.

Example 9.6 *Consider the design of PID controller using auto-tuner for the integrating system described by the following transfer function:*

$$G(s) = \frac{(-3s+1)e^{-6s}}{s(10s+1)^2}. \tag{9.31}$$

The proportional feedback control gain is selected as $K_T = 0.01$ *to produce a stable system for the relay experiments. The relay amplitude of* 1.75 *and hysteresis of* 0.2 *are chosen. Find the PID controller parameters for* $\beta = 1.5$ *and* $\xi = 0.707$. *In the relay feedback control, band limited white noise is added to the output, where the signal power is selected to be* 1 *with gain* 0.02 *and seed* 23341.

Solution. *Figure 9.20 shows the input and output data with the relay feedback control. Using the frequency sampling filter model, the closed-loop frequency response is estimated as*

$$\hat{T}(e^{j\omega_1}) = -0.1924 - j0.0486$$

and the plant frequency response is estimated using (9.27) as $G(j\omega_1) = -16.2741 - j3.4158$ *where* $\omega_1 = 0.0503$. *Figure 9.21 compares the estimated frequency response at*

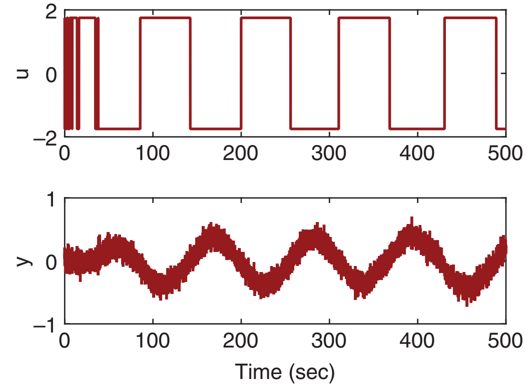


Figure 9.20 Input and output relay feedback control data (Example 9.6).

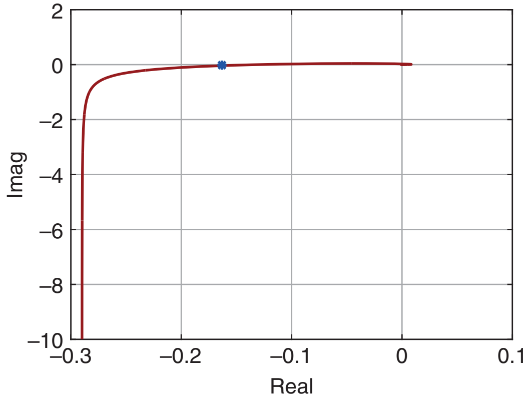


Figure 9.21 Comparison between the estimated frequency point with the original frequency response (Example 9.6). Key: solid line, plant frequency response; dot, estimated frequency response.

ω_1 against the frequency response of the plant transfer function (9.31). It is seen that the estimated result is quite accurate.

With the single frequency response point, the integrating time delay model is found using (9.29) and (9.30):

$$G(s) = \frac{0.8359e^{-27.1341s}}{s}. \tag{9.32}$$

This model leads to the PID controller parameter using the empirical rules in Section 8.4.3 as

$$K_c = 0.0305; \quad \tau_I = 90.1559; \quad \tau_D = 7.1750.$$

Because the original system actually contains an integrator, the Nyquist diagrams (see Figure 9.22) show their frequency responses are very similar at the cross over frequency region as well as low and medium frequency regions, comparing the PID controller applying to the integrating plus delay model (9.32) with that applying to the original system (9.31). Figure 9.22(b) compares the magnitudes of their sensitivity functions. It is seen that the discrepancies are very small between the sensitivity functions in all the frequency regions. The PID control system performance is evaluated with a step reference signal and a step disturbance signal entering the simulation at $t = 300$ (s). The derivative control is implemented on the output only with the filter time constant equal to $0.1\tau_D$. Three

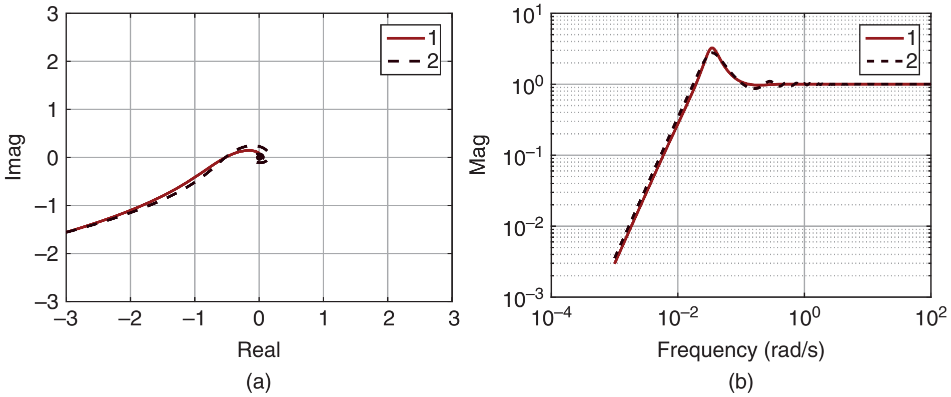


Figure 9.22 Frequency response comparison (Example 9.6). (a) Nyquist diagram. (b) Sensitivity function. Key: line (1) frequency response calculated with integrating plus delay model; line (2) frequency response calculated with the actual plant.

control configurations are considered (see Figures 9.23(a) and (b)). Overall, the PID controller produces the best performance in terms of disturbance rejection. The PD controller could not reject the step disturbance; however, it produces the best reference following performance. There is an oscillation in the closed-loop control if a PI controller is used.

Although it is derived for an integrator plus delay system, the auto-tuner provides satisfactory closed-loop performance for many classes of systems. This is illustrated by the following example.

Example 9.7 *Assume that a second order system with time delay is described by the transfer function:*

$$G(s) = \frac{e^{-3s}}{(8s+1)(6s+1)}. \tag{9.33}$$

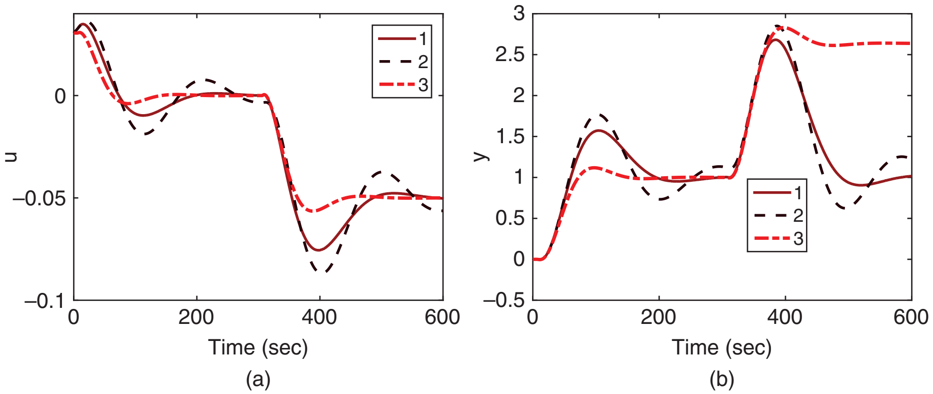
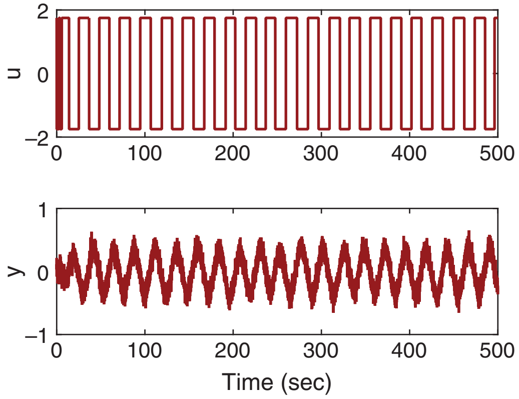


Figure 9.23 Comparison of closed-loop performance for three types of controllers ($\beta = 1.5$, $\xi = 0.707$). (a) Control signal. (b) Output. Key: line (1) PID control response; line (2) PI control response; line (3) PD control response.

Figure 9.24 Input and output relay feedback control data (Example 9.7).



Choosing the proportional feedback control gain $K_T = 0.6$, *and relay amplitude of* 1.75 *and hysteresis of* 0.2, *find the PID controller parameters for* $\beta = 1.5$ *and* $\xi = 0.707$. *In the relay feedback control, band limited white noise is added to the output, where the signal power is selected to be* 1 *with gain* 0.02 *and seed* 23341.

Solution. *Following Tutorial 9.1, we obtain the relay control data for the proportional controlled system as shown in Figure 9.24. Then, following Tutorial 9.4, the closed-loop frequency response with proportional control is estimated using the FSF model as*

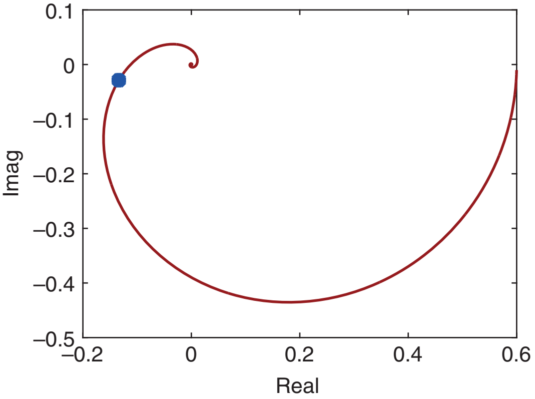
$$\hat{T}(e^{j\omega_1}) = -0.1539 - j0.0380$$

and the plant frequency response is estimated using (9.27) as $G(j\omega_1) = -0.2238 - j0.0475$ *where* $\omega_1 = 0.264$. *Figure 9.25 compares* $G(j\omega_1)$ *against the plant frequency response produced by the original system (9.33). It is seen that despite of the measurement noise, the estimation result is very accurate. With the frequency response information, the integrator plus time delay model becomes*

$$G(s) = \frac{0.0604e^{-5.1577s}}{s} \tag{9.34}$$

based on (9.29) and (9.30).

Figure 9.25 Comparison between the estimated frequency point with the original frequency response (Example 9.7). Key: solid line, plant frequency response; dot, estimated frequency response.



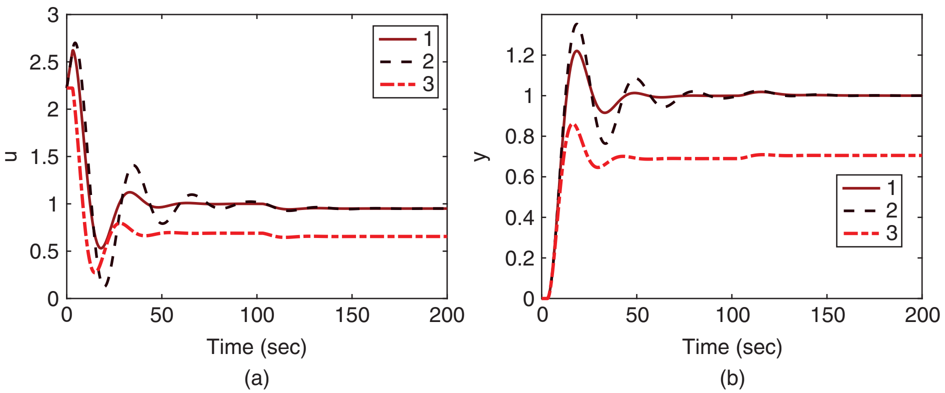


Figure 9.26 Comparison of closed-loop performance for three types of controllers ($\beta = 1.5$, $\xi = 0.707$). (a) Control signal. (b) Output. Key: line (1) PID control response; line (2) PI control response; line (3) PD control response.

By choosing $\beta = 1.5$, the PID controller parameters are found using the empirical rules in Section 8.4.3:

$$K_c = 2.2232; \quad \tau_I = 17.1371; \quad \tau_D = 1.3638.$$

This PID control system is simulated using a unit step input reference signal with sampling interval $\Delta t = 0.05$ s. A step input disturbance with amplitude 0.05 enters the simulation at half of the simulation time. Figure 9.26 shows the evaluation results for the configurations of PID, PI, and PD control systems. In the implementation, the derivative control is implemented on output only with a filter time constant $0.1\tau_D$, both proportional and integral control functions are implemented on the feedback error. The PID controller provides much improved performance over the PI and PD control systems for this choice of β parameter. The oscillation for the PI control system reduces when β is increased. In general, if a PI controller is used, the parameter β should be selected to be larger than 2 to reduce the oscillation due to the small phase margin, which can be verified for this example.

To understand why the auto-tuner designed for the integrating system worked for a stable system, a frequency response analysis is performed. Figure 9.27(a) makes a comparion between the Nyquist diagrams of the controller with the estimated integrating plus delay model (9.34) and the controller with the stable plant transfer function (9.33). The Nyquist diagrams show that their frequency responses are quite similar at the cross over frequency region, which produces the closed-loop stability with sufficient gain and phase margins when the controller found by the auto-tuner is applied to the original system. Figure 9.27(b) compares the magnitudes of their sensitivity functions. It is seen that at the high frequency region, they are very similar. However, at the lower and medium frequency regions, there are some discrepancies. In particular, in the lower frequency region, the magnitude of the sensitivity is much smaller when the controller is applied to the original system, which indicates that the controller may have a better disturbance rejection when it is found using the auto-tuner designed for the integrating systems.

This auto-tuner will be used for finding the PID controllers for the unmanned aerial vehicles in Chapter 10.

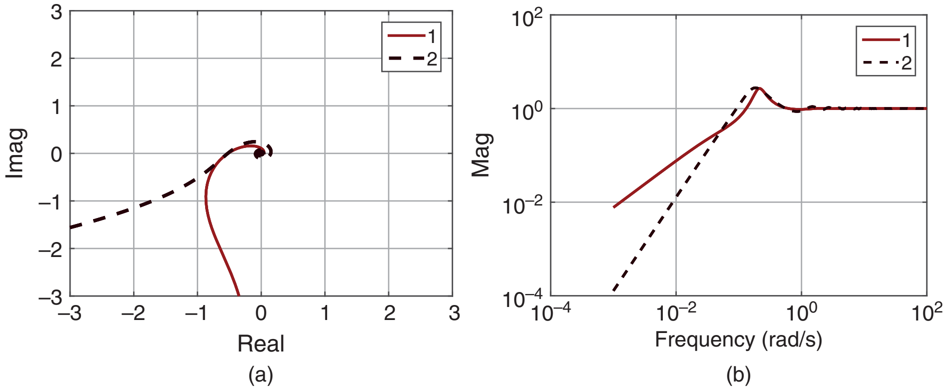


Figure 9.27 Frequency response comparison (Example 9.7). (a) Nyquist diagram. (b) Sensitivity function. Key: line (1) frequency response calculated with integrating plus delay model; line (2) frequency response calculated with the actual plant.

9.7.3 Auto-tuning of Cascade Control Systems

Auto-tuners are very convenient for tuning cascade control systems. The inner-loop control system will be auto-tuned first to find the appropriate controller, followed by the implementation of this secondary controller. An outer-loop control system will be auto-tuned with the closed-loop secondary system considered. As an illustration, we consider the following example.

Example 9.8 *The transfer function for the secondary system is assumed to have the transfer function:*

$$G_1(s) = \frac{2e^{-3s}}{s(s+1)} \tag{9.35}$$

The transfer function for the primary system is described by

$$G_2(s) = \frac{0.1e^{-s}}{s(s+1)}$$

Find the PID controllers using the auto-tuner for the cascade PID control system.

Solution.

Auto-tuning inner-loop controller

The proportional controller used to stabilize the inner-loop system is selected to be $K_{T1} = 0.04$*. In the simulation, a zero mean white noise with standard deviation of*0.025 *is added to the measured output. The relay amplitude is selected to be* 1.75 *and hysteresis is* 0.2 *to prevent the relay from the switching caused by the random noise. Figure 9.28 shows the input and output data for the closed-loop system. It is seen from this figure that the measurement noise has caused some random switches of the relay, particularly in the beginning of the simulation. From the relay testing data, the number of samples within one period is estimated to yield* $N = 355$*. The estimated fundamental frequency response for the closed-loop system with* $K_{T1} = 0.04$ *is* $-0.2472 - j0.0673$*. From this value, the frequency response of the integrating system is found as* $G(j\omega_1) = -5.0134 - j1.0791$*. With*

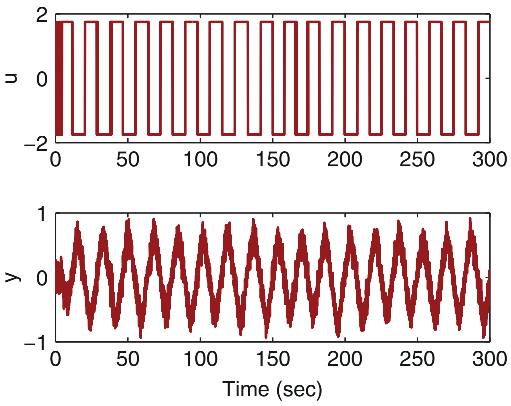


Figure 9.28 Relay feedback control signals from inner closed-loop system (Example 9.8). top figure input signal; bottom figure output signal.

the frequency response value of the inner-loop system, the following integrator with delay model is calculated based on (9.29)–(9.30):

$$G_s(s) = \frac{1.7852e^{-4.0547s}}{s}$$

Choosing $\beta = 2$*, which gives the desired closed-loop time constant about* 8*, with PID controller tuning rules presented in Section 8.4.3, the PID controller parameters for the inner-loop control system are calculated as*

$$K_c = 0.0844; \quad \tau_I = 21.0864; \quad \tau_D = 1.0449$$

Considering that the τ_D *value is quite small, also noise in the system, the inner-loop controller is selected as a PI controller. The Nyquist curve for the PI controller against the original system described by (9.35) is shown in Figure 9.29(a). Roughly it can be read from the figure that the control system has gain margin about* 2 *and phase margin about* 45 *degree, which are quite close to the specification in the tuning rules. The inner-loop PI control system is evaluated via a step change at time* $t = 0$ *and a step input disturbance with amplitude 0.05 entering the system at* $t = 60$ *(sec). Figure 9.29(b) shows the control signal response and the output response to the reference change and the disturbance.*

Auto-tuning outer-loop controller

The automatic tuning process is repeated for the outer-loop control system, however, the inner-loop system is controlled with the PI controller auto-tuned previously. It is important for a cascade control system in the relay experiment that the inner-loop dynamics are considered. In the outer-loop relay experiment, the proportional controller $K_{T2} = 0.1$ *is used to stabilize the primary system, which has an integrator. The relay feedback control data with amplitude of* 1.75 *and hysteresis of* 0.2 *for the outer-loop system is shown in Figure 9.30 where the same measurement noise is added. It is seen that there are many more random switches due to the measurement noise. The estimation of the closed-loop frequency response using the frequency sampling filter model leads to the value* $-0.1133 - j0.0011$ *at the fundamental frequency. Then, the calculated frequency response of the primary plant together with the inner-closed-loop system at the fundamental frequency of the relay signal is* $G_p(j\omega_1) = -1.0177 - j0.0091$.

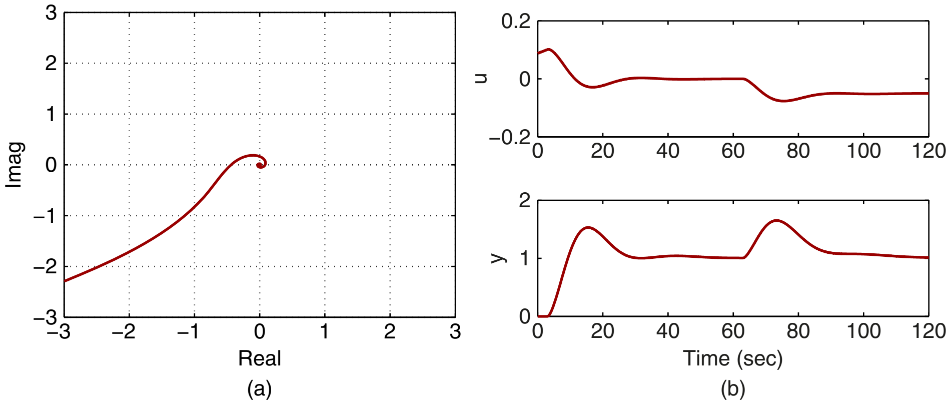
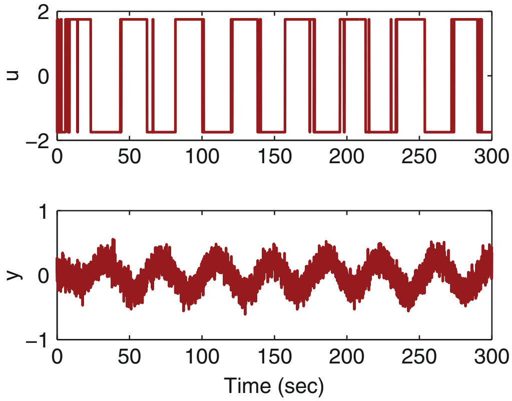


Figure 9.29 Auto-tuning inner-loop control system (Example 9.8). (a) Nyquist curve with $C_s(j\omega)$. (b) Closed-loop response.

Figure 9.30 Relay feedback control signals from outer- closed-loop system (Example 9.8). top figure input signal; bottom figure output signal.



The integrator with time delay model for the design of primary controller is derived using (9.29)–(9.30) to give:

$$G_p(s) = \frac{0.1752e^{-9.0729s}}{s}$$

Note that the original time delay for the primary plant is 1*, however, because of the inner-loop dynamics, the estimated time delay is over* 9*. By choosing* $\beta = 2$*, which gives the desired closed-loop time constant about* 18*, the PID controller parameters are calculated using the tuning rules to give*

$$K_c = 0.3844; \quad \tau_I = 47.1835; \quad \tau_D = 2.3701$$

By neglecting the derivative control term, the PI controller is determined for the outer-loop control system. Figure 9.31(a) shows the Nyquist curve for the cascade control system with the primary controller auto-tuned. It is seen that the closed-loop system has gain margin approximately greater than 2 *and phase margin greater than* 45 *degree. The cascade control system is finally evaluated by using a step reference signal at* $t = 0$ *and a step disturbance with magnitude of* 0.05 *entering at* $t = 150$ *as the input disturbance to the*

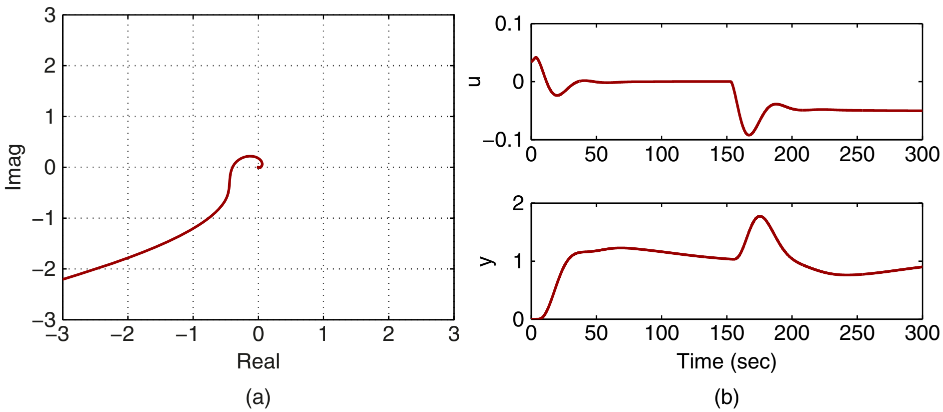


Figure 9.31 Auto-tuning outer-loop control system (Example 9.8). (a) Nyquist curve with $C_p(j\omega)$. (b) Closed-loop response.

secondary plant. Figure 9.31(b) shows the cascade closed-loop control system to the reference signal and the disturbance signal.

9.7.4 Food for Thought

1. Why do you think that the auto-tuner designed for integrating system can be applied to a stable system?
2. Do you expect that this class of auto-tuners would produce a faster disturbance rejection based on the observation of the sensitivity function in Example 9.7?
3. We use relay with hysteresis in the identification experiments. Do you think that it would work better if we use the integrator with relay control apparatus in the identification experiment? Why?
4. If you wish to have a faster closed-loop response for disturbance rejection, should you increase or decrease the parameter β?
5. If you wish to reduce the effect of measurement noise, should you increase or decrease the parameter β?

9.8 Summary

We have discussed automatic tuning of PID controllers in this chapter. The auto-tuners are designed for stable systems and integrating systems. Both involve relay feedback control to generate the input and output data for estimation of plant frequency response. Then, the PID controller design methods discussed in Chapter 8 are used to automatically find the controller parameters.

The other important aspects of the chapter are summarized as follows.

- Relay feedback control is utilized in generating the input and output data for estimation of process frequency response. Following the MATLAB tutorials, we can create our own Simulink functions for simulations of relay feedback control system and those programs can also be translated into C-program for micro-controller implementation.

- The input and output data collected from the relay feedback control experiments in simulation or in real-time implementation are used for the estimation either using Fourier analysis or using the frequency sampling filter model. The frequency sampling filter based estimation is performed in a recursive least squares algorithm with the feature for real-time applications.
- Monte-Carlo simulation studies have been used to investigate the accuracy of the estimated models in the presence of measurement noise and low frequency disturbances.
- Once the process frequency response is estimated, the PID controller design methods either using two frequency points or using an integrating with delay model are applied to obtain the PID controller parameters.

9.9 Further Reading

1. Relay feedback control has been one of the key instruments used in the automatic tuning of PID controllers (Astrom and Hagglund (1984), Astrom and Hagglund (1988), Astrom and Hagglund (1995), Astrom and Hagglund (2006), Hagglund and Astrom (1985), Yu (2006), Johnson and Moradi (2005)).
2. The auto-tuner for stable systems was originally presented in Wang (2017).
3. Auto-tuners are experimentally compared in laboratory test beds (Berner et al. (2018)).
4. Books for automatic tuning of PID controllers include Yu (2006), Sung et al. (2009)
5. Software package introduced for tuning of PID controllers is discussed in Oviedo et al. (2006). Use of phase-locked-loop idea is used to auto-tune PID controllers in Crowe and Johnson (2002).
6. Auto-tuners in closed-loop operations include Lee et al. (1990), Schei (1992), Tan et al. (2000). Automatic tuning of cascade PID control system was discussed in Jeng and Lee (2012) and Jeng (2014). Automatic tuning of PID controller for nonlinear systems was presented in Cetin and Iplikci (2015). Multi-loop PID controller tuning was proposed using an optimization based approach (Dittmar et al. (2012)). A simplified approach to auto-tuner design with respect to disturbance rejection was proposed in Romero et al. (2011). An auto-tuner was designed for fractional order plus delay model in Jin.
7. Estimation of transfer function model by changing between two relay feedback controllers to obtain multi-frequency signals (Schei (1994)), and estimation of step response model (Wang and Cluett (1997)).
8. Frequency sampling filters was introduced in Bitmead and Anderson (1981) and Wang and Cluett (2000).
9. The auto-tuning algorithm presented in this chapter has been successfully used to attitude control of fixed-wing unmanned aerial vehicle with experimental validations (Poksawat et al. (2016), Poksawat et al. (2017) and Poksawat (2018)). It has been used to find a PID control system for an electro-mechanical system with two inputs and two outputs (see Wang et al. (2017)). In Chen (2017), the auto-tuning algorithm has been successfully used to find flight controllers for quadrotor UAVs.
10. There are many books for the topics of system identification (see Ljung (1999), Soderstrom and Stoica (1989), Goodwin and Sin (1984), Young (2012), Soderstrom (2018)).

Problems

9.1 *Consider the systems with the following transfer functions:*

$$G(s) = \frac{e^{-3s}}{5s+1}$$
$$G(s) = \frac{(-10s+1)}{(10s+1)^6}$$
$$G(s) = \frac{s+1}{(5s^2+s+1)^3}$$

1. *Suppose that the measurement is corrupted by a noise source that has zero mean with variance 0.1. Following Tutorial 9.1, write the RelayH.slx real-time function and build a Simulink relay feedback control simulation program. Sampling interval* $\Delta t = 0.1$.
2. *Choose the relay amplitude to be* 3 *and adjust the hysteresis level* ϵ *to prevent the random switches caused by the noise. Apply the relay feedback control program to the above systems. What are your observations on the choice of* ϵ *with regard to relay amplitude and the magnitude of the measurement noise? If you use Simulink noise generator in the continuous-time, will the sampling interval* Δt *affect the choice of* ϵ*?*
3. *Add a constant input disturbance with amplitude of* 0.3 *to the relay feedback control. What are the effects of this disturbance on the relay with hysteresis control system?*

9.2 *Consider the three systems given in Problem 9.1.*

1. *Following Tutorial 9.2, write the MATLAB real-time function RelayI.slx for relay with integrator control. Choose the same relay amplitude with very small hysteresis (*$\epsilon = 0.001$*).*
2. *Apply the relay with integrator control to the three systems given in Problem 9.2. What are your observations on the effect of noise with this type of relay feedback control system?*
3. *Add a constant input disturbance with amplitude of* 0.3 *to the relay feedback control. What are the effects of this disturbance on the relay with integrator control system?*
4. *Will the sampling interval* Δt *affect the relay with integrator control system?*

9.3 *Assume that we have generated relay feedback control data by solving Problem 9.1.*

1. *Following Tutorial 9.3, write the MATLAB program FFTRelay.m for estimation of frequency response using relay feedback data.*
2. *Apply the program to the six sets of data generated from Problem 9.1.*
3. *Evaluate the estimation results against the Nyquist curves obtained from the transfer functions.*
4. *What are your observations on the effect of measurement noise on the accuracy? The effect of constant input disturbance?*

5. *Alternatively, following Tutorial 9.4, write the MATLAB program FSFRelay.m for the estimation of frequency response using relay feedback data and repeat the exercises outlined in the previous steps.*

9.4 *Apply the MATLAB program FFTRelay.m (or FSFRelay.m) to the six sets of data generated from Problem 9.2, where relay with integrator control has been used. What are your observations on the effect of measurement noise and disturbance on the accuracy of the estimation?*

9.5 *Consider the systems with the following transfer functions:*

$$G(s) = \frac{10(s+1)e^{-s}}{(s+0.1)^5}$$

$$G(s) = \frac{(s-2)}{(s^2+0.5s+3)^2}$$

$$G(s) = \frac{1}{(s+1)^2(s+5)^2(s+10)}$$

1. *Following Tutorial 9.5, write the auto-tuner for stable systems.*
2. *Apply the auto-tuner to the above systems with appropriate sampling interval Δt, where the performance parameter $\beta = 1, 2, 3$.*
3. *Evaluate the closed-loop performance with a unit step reference signal and a step input disturbance with magnitude of* -0.5.
4. *What are your observations on the closed-loop responses to reference following and disturbance rejection in terms of the parameter β?*
5. *Add measurement noise and the input disturbance in the relay with integrator control system. Will the noise and disturbance significantly affect the performance of the closed-loop control system produced by the auto-tuner?*

9.6 *Consider the integrating systems with the following transfer functions:*

$$G(s) = \frac{0.1e^{-3s}}{s(s+1)}$$

$$G(s) = \frac{2e^{-s}(-s+1)}{s(s+3)^2}$$

1. *Following the computational steps outlined in Section 9.7, write the auto-tuner for integrating systems where the relay with hysteresis is applied to a proportionally controlled system (see Figure 9.19).*
2. *Choose a proportional controller K_T that will produce a stable closed-loop system and apply the auto-tuner to the above systems, where sampling interval Δt, relay amplitude and hysteresis are selected appropriately.*
3. *Evaluate the closed-loop control system performance with unit step reference signal and a step input disturbance with amplitude* -0.1 *for the performance parameter* $\beta = 1, 2, 3$.

9.7 *The auto-tuner derived for integrating system can also be applied to stable systems in a closed-loop controlled environment. Considering the systems with the following transfer functions*

$$G(s) = \frac{3e^{-3s}}{(2s+1)^4}$$
$$G(s) = \frac{(-s+1)e^{-s}}{(3s+1)(2s+1)}$$

Use the auto-tuner with integrator model to find the PID controller parameters for these systems. In the relay control experiments, feedback control gain $K_T = 0.2$*, and relay amplitude of* 1.75 *and hysteresis of* 0.2 *are used. Sampling interval* $\Delta t = 0.1$ *is selected. The performance parameters* $\beta = 1$ *and* $\xi = 0.707$ *are selected for fast disturbance rejection. Evaluate the closed-loop control system performance for step input disturbance rejection with a disturbance amplitude of* 1.

9.8 *The auto-tuners are very convenient for tuning cascade control systems. Consider that a cascade control system has the inner-loop transfer function*

$$G_1(s) = \frac{e^{-s}}{s+1}$$

and the outer-loop transfer function

$$G_2(s) = \frac{2}{s(s+10)}$$

1. *Construct an auto-tuner for this cascade control system from tuning the inner-loop control system first followed by tuning the outer-loop control system.*
2. *By choosing the closed-loop performance parameter* β*, the closed-loop response speed can be specified for the inner-loop and outer-loop control systems. In order for the cascade control system to work properly, the inner-loop response speed needs to be much faster than the outer-loop's response speed. How do you choose* β *for this auto-tuned cascade control system?*

9.9 *We can design an auto-tuner for integrating systems using the modified IMC-PI controller introduced in Skogestad (2003), which was discussed in Section 1.4.1. For the estimated integrating with delay model,* $G(s) = \frac{K_p e^{-ds}}{s}$*, the PI controller parameters are calculated using the following expressions:*

$$K_c = \frac{1}{K_p(\tau_{cl} + d)}$$
$$\tau_I = 4(\tau_{cl} + d).$$

Repeat the exercises given in Problem 9.7 with the closed-loop performance parameter $\tau_{cl} = d$*. What are your observations when you compare these two PI control systems?*

10

PID Control of Multi-rotor Unmanned Aerial Vehicles

10.1 Introduction

This chapter will present the PID control of multi-rotor unmanned aerial vehicles as case studies. Because of nonlinearities and physical parameter uncertainties, the attitude control systems for an unmanned aerial vehicle are proposed to be PID control systems in a cascade structure and the PID controller parameters are found using the auto-tuners designed in Chapter 9. The auto-tuners are implemented on laboratory test rigs specifically designed for multi-rotor unmanned aerial vehicles and operated on the ground for the safety of the equipment. Experiments are conducted to evaluate the attitude control systems.

10.2 Multi-rotor Dynamics

This section will discuss the dynamics of a quadrotor and a hexacopter for the PID control system design. We will pay attention to their shared common features and their differences.

To ensure stable flight of a multi-rotor unmanned aerial vehicle (UAV), its attitude is required to be under feedback control. The attitude of a multi-rotor is captured by the variations of the three Euler angles: roll angle ϕ, pitch angle θ and yaw angle ψ. More specifically, the roll angle ϕ defines the rotation about the x body axis, the pitch angle θ about the y body axis, and the yaw angle about the z body axis. To maintain stable flight with a multi-rotor UAV, the three Euler angles are required to follow desired reference signals in closed-loop control. The attitude control forms the common ground of PID control methodologies for multi-rotor UAVs. However, the difference comes from the details of the control signal realizations through actuators that are the rotors associated with the UAVs.

10.2.1 Dynamic Models for Attitude Control

From the control system design point of view, the outputs of an attitude control system are clearly the three Euler angles: roll angle ϕ, pitch angle θ and yaw angle ψ. One important reason why these three angles are selected as the outputs is that the reference signals to the angles are readily available. For instance, in order to maintain stable flight, the reference signals to the roll angle ϕ and pitch angle θ are commonly selected

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

to be zero whilst the yaw angle ψ is determined by the position of the UAV in the horizontal plane. The questions remain as what the control input signals are and how they are related to the output signals for the attitude control problem.

In order to derive the dynamic models for attitude control, a reference frame is defined. The mathematical models for the multi-rotor unmanned aerial vehicles share the same reference frame leading to the same dynamic models at this common ground. As an example, we examine the derivation of the dynamic models for a quadrotor UAV.

Figure 10.1 shows the framework used to derive a quadrotor dynamics model (Bouabdallah et al. (2004), Corke (2011), Derafa et al. (2006)), which illustrates that the origin of the body frame is in the mass center of the quadrotor UAV and the z-axis is upwards. M_1, M_2, M_3, and M_4 are the four rotors and $\omega_1, \omega_2, \omega_3$, and ω_4 represent the rotors' angular velocities.

In order to obtain an unique solution for the mathematical models, the transformation sequence is assumed to be $\psi \rightarrow \theta \rightarrow \phi$. The input signals for the quadrotor are the torques τ_x, τ_y, and τ_z in the x-, y-, and z-axes, respectively. In the same three dimensional space, we define p, q, and r as their angular velocities and I_{xx}, I_{yy}, and I_{zz} as the moments of inertia for the three axes in the x, y, and z directions. The quadrotor UAV is assumed to have a symmetric structure with four arms aligned with the x-axis and y-axis, and as a result there is no interaction between the torques along the three axes. From Euler's equation of motion (Bouabdallah et al. (2004), Corke (2011), Derafa et al. (2006)), the following dynamic equations in the x, y, and z axes are obtained:

$$\begin{aligned} I_{xx}\dot{p} &= (I_{yy} - I_{zz})qr + \tau_x \\ I_{yy}\dot{q} &= (I_{zz} - I_{xx})pr + \tau_y \\ I_{zz}\dot{r} &= (I_{xx} - I_{yy})pq + \tau_z. \end{aligned} \tag{10.1}$$

From the control system design point of view, if the multi-rotor UAV carries a payload, the load torque can be projected onto the x-, y- and z-axes, denoted by τ_x^{d}, τ_y^{d} and τ_z^{d}. These quantities are unknown in general, and are considered as constant disturbances in control system design. With the consideration of load disturbances, we modify the motion equations as

$$\begin{aligned} I_{xx}\dot{p} &= (I_{yy} - I_{zz})qr + \tau_x - \tau_x^{\mathrm{d}} \\ I_{yy}\dot{q} &= (I_{zz} - I_{xx})pr + \tau_y - \tau_y^{\mathrm{d}} \\ I_{zz}\dot{r} &= (I_{xx} - I_{yy})pq + \tau_z - \tau_z^{\mathrm{d}}. \end{aligned} \tag{10.2}$$

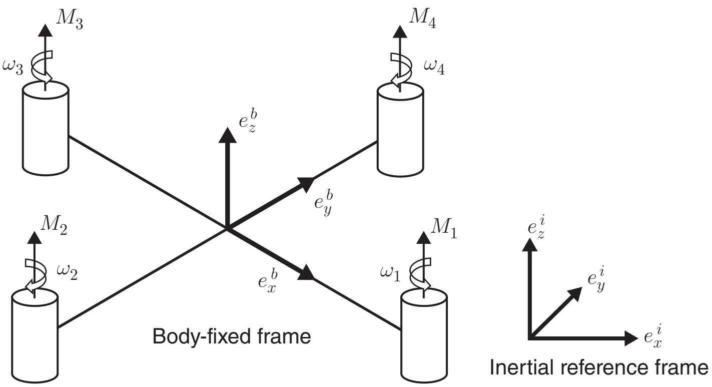


Figure 10.1 Inertial frame and body frame of the quadrotor.

If the payload is aligned with the mass center and is symmetric, the projections of the load torque on the x- and y-axes are small.

Now, the relationships between Euler angular velocities and the body frame angular velocities (p, q, and r) are described in the following differential equations (Corke (2011)):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{10.3}$$

The dynamic models (10.2) and (10.3) present mathematical descriptions for the attitude control system design for the quadrotor. Clearly, there are three outputs in the control system, and the manipulated variables or the control signals are the three torques, τ_x, τ_y, and τ_z, along the x, y, and z directions.

The dynamic models for attitude control of a hexacopter are also described by the differential equations (10.2) and (10.3). Because a hexacopter uses six rotors attached to the end of each arm with equal distance from the vehicle's center of gravity, which is illustrated in Figure 10.2, it has better fault-tolerant properties and capability of carrying a larger payload than the quadrotor UAVs.

10.2.2 Actuator Dynamics for Quadrotor UAVs

It is worthwhile emphasizing that the dynamic models (10.1)–(10.3) derived for the quadrotor have not taken the actuators into consideration. The control signals, the torques τ_x, τ_y, and τ_z in the body frame, will be implemented using DC motors. Thus, there will be additional first order or second order models used to capture the DC motor dynamics for the control system design.

In quadrotor control, the torques τ_x, τ_y, and τ_z in the body frame are generated by the differences in rotor thrusts. The upward thrust produced by each rotor is

$$T_i = b_t\omega_i^2, \quad i = 1, 2, 3, 4.$$

The total thrust is, hence,

$$T = T_1 + T_2 + T_3 + T_4 = b_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{10.4}$$

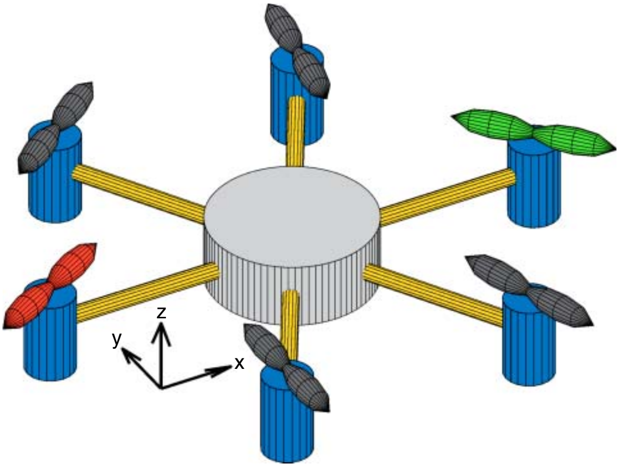


Figure 10.2 Representation of a hexacopter (Ligthart et al. (2017)).

where b_t is the thrust constant determined by air density, the length of the blade, and the blade radius, and ω_i is the ith rotor's angular speed.

When we only consider the attitude control, the altitude of the quadrotor UAV is not controlled and the total thrust T is manually set by the operator. Therefore, there will be an independent reference signal to the total thrust T.

The torques about quadrotor's x-axis and y-axis are

$$\tau_x = d_\text{mm}(T_4 - T_2) = d_\text{mm}b_\text{t}(\omega_4^2 - \omega_2^2) \tag{10.5}$$

$$\tau_y = d_\text{mm}(T_3 - T_1) = d_\text{mm}b_\text{t}(\omega_3^2 - \omega_1^2) \tag{10.6}$$

where d_mm is the distance from the motor to the mass center. The torque applied to each propeller by the motor is opposed by aerodynamic drag, and the total reaction torque about the z-axis is

$$\tau_z = k_\text{d}({\omega_1}^2 + {\omega_3}^2 - {\omega_2}^2 - {\omega_4}^2) \tag{10.7}$$

where k_d is a drag constant determined by the same factors as b_t.

To determine the angular velocities for the four DC motors with regard to the control signals τ_x, τ_y, τ_z, and T, the linear equations (10.4)–(10.7) are solved to give the following algebraic equations in the matrix form:

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b_\text{t}} & 0 & -\frac{1}{2d_\text{mm}b_\text{t}} & -\frac{1}{4k_\text{d}} \\ \frac{1}{4b_\text{t}} & -\frac{1}{2d_\text{mm}b_\text{t}} & 0 & \frac{1}{4k_\text{d}} \\ \frac{1}{4b_\text{t}} & 0 & \frac{1}{2d_\text{mm}b_\text{t}} & -\frac{1}{4k_\text{d}} \\ \frac{1}{4b_\text{t}} & \frac{1}{2d_\text{mm}b_\text{t}} & 0 & \frac{1}{4k_\text{d}} \end{bmatrix} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \tag{10.8}$$

From (10.8), once the manipulated variables T, τ_x, τ_y, and τ_z are decided by the feedback controllers, the squared velocities, ω_1^2, ω_2^2, ω_3^2, and ω_4^2, of the motors will be uniquely determined.

From the context of feedback control, the squared velocities are translated into the velocity reference signals ω_1^*, ω_2^*, ω_3^*, and ω_4^* that will be implemented in a typical DC motor drive, which are equal to the square roots of the components calculated using (10.8).

The DC motor dynamics will affect the closed-loop control performance, which should be included in the quadrotor model and they are approximated by a first-order transfer function:

$$\frac{\Omega_i(s)}{V_i(s)} = \frac{r_\text{wv}}{\epsilon_\text{m}s + 1}, \tag{10.9}$$

where $V_i(s)$ is the Laplace transform of the armature voltage v_i to the ith motor, $\Omega_i(s)$ is the Laplace transform of the motor velocity, ϵ_m is the time constant, and r_wv is the steady-state gain for the motor. The armature voltage v_i is changed by manipulating the duty cycle of the pulse width modulation (PWM) signal of each motor drive, where the relationship between the motor armature voltage and the PWM duty cycle is

$$v_i = d_i^\text{c} V_\text{bat}, \tag{10.10}$$

where d_i^c is the PWM signal duty cycle of the ith DC motor drive and V_{bat} is the battery voltage assumed to be constant. Substituting equation (10.10) into equation (10.9) yields:

$$\frac{\Omega_i(s)}{D_i^c(s)} = \frac{V_{\text{bat}} r_{\text{wv}}}{\epsilon_{\text{m}} s + 1} \tag{10.11}$$

which describes the ith DC motor dynamics. To achieve the desired speed ω_i^* without steady-state error, a PI controller is required, which is designed by following the pole-assignment PI controller design method introduced in Chapter 3 with the parameters V_{bat}, r_{wv} and ϵ_{m}. If we select two identical closed-loop poles at $-\frac{1}{\tau_{\text{cl}}}$, then the closed-loop control system for the ith DC motor is approximately a second order system with unit gain with the transfer function:

$$\frac{\Omega_i(s)}{\Omega_i(s)^*} = \frac{1}{(\tau_{\text{cl}} s + 1)^2} \tag{10.12}$$

where $\Omega_i(s)^*$ is the Laplace transform of the velocity reference signal to the DC motor. Because the DC motor control system has a very small time constant, the transfer function (10.12) is approximated using a time delay e^{-ds} with the parameter d determined through actual experiments.

The DC motor control systems are most commonly purchased together with the motors. Thus, for the implementation of the quadrotor control system, the control signals are the desired speed reference signals, ω_1^*, ω_2^*, ω_3^*, and ω_4^* to the DC motors and their closed-loop dynamics are modeled by time delay components assuming the PI controllers are used in DC motor control.

10.2.3 Actuator Dynamics of Hexacopters

Similar to quadrotor control, the torques τ_x, τ_y, and τ_z in the body frame of a hexacopter are generated by the rotor thrusts. The upward thrust produced by each rotor is

$$T_i = b_{\text{t}} \omega_i^2, \quad i = 1, 2, 3, 4, 5, 6.$$

The total upward thrust is used to control the translational motion along the z-axis and is defined as follows:

$$T = \sum_{i=1}^{6} T_i = b_{\text{t}} \sum_{i=1}^{6} \omega_i^2 \tag{10.13}$$

where b_{t} is the thrust constant determined by air density, the length of the blade, and the blade radius, and ω_i is the ith rotor's angular speed.

Let d_{mm} be the distance from the center of gravity to the rotor and k_{d} be the drag constant. For the hexacopter, the roll, pitch, and yaw control objectives are achieved by controlling the difference in thrust generated by each rotor, which is defined as:

$$\tau_x = d_{\text{mm}} \sin(60°)(-T_2 - T_3 + T_5 + T_6) \tag{10.14}$$

$$\tau_y = -d_{\text{mm}} T_1 + d_{\text{mm}} T_4 + d_{\text{mm}} \sin(30°)(-T_2 - T_6 + T_3 + T_5) \tag{10.15}$$

$$\tau_z = k_{\text{d}}(-T_1 + T_2 - T_3 + T_4 - T_5 + T_6). \tag{10.16}$$

Equations (10.13), (10.14), (10.15), and (10.16) can be arranged in matrix form as:

$$\overbrace{\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}}^{Z} = \overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -\frac{\sqrt{3}d_{\mathrm{mm}}}{2} & -\frac{\sqrt{3}d_{\mathrm{mm}}}{2} & 0 & \frac{\sqrt{3}d_{\mathrm{mm}}}{2} & \frac{\sqrt{3}d_{\mathrm{mm}}}{2} \\ -d_{\mathrm{mm}} & -\frac{d_{\mathrm{mm}}}{2} & \frac{d_{\mathrm{mm}}}{2} & d_{\mathrm{mm}} & \frac{d_{\mathrm{mm}}}{2} & -\frac{d_{\mathrm{mm}}}{2} \\ -k_{\mathrm{d}} & k_{\mathrm{d}} & -k_{\mathrm{d}} & k_{\mathrm{d}} & -k_{\mathrm{d}} & k_{\mathrm{d}} \end{bmatrix}}^{\Phi} \overbrace{\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}}^{X}$$

which has the simplified expression:

$$Z = \Phi X. \tag{10.17}$$

To implement the attitude control system, we need to determine the values of upward thrust T_i, $i = 1, 2, \dots, 6$. Unlike the quadrotor control case, from (10.17), there is no explicit one-to-one relationship between the actuators and the τ_x, τ_y, τ_z, and T variables.

In the literature, this was often determined using the pseudo inverse of the matrix Φ, leading to

$$X = \Phi^{+}Z \tag{10.18}$$

where Φ^{+} denotes the pseudo inverse of matrix Φ.

An interesting approach is to borrow the idea from the model predictive control of hexacopter (Ligthart et al. (2017)) to formulate the inversion problem in terms of optimization. We define the following objective function:

$$J = (Z - \Phi X)^{\mathrm{T}}(Z - \Phi X) + X^{\mathrm{T}}WX \tag{10.19}$$

where A^{T} denotes the transpose of A matrix, W is a positive definite matrix and for most cases, it is selected as a diagonal matrix with all positive elements. The first term in the objective function says that we would like to find the best X vector such that the vector Z is matched as close as possible while the second term indicates that we wish the upward thrust vector to be limited with a weighting matrix W. In most cases, we wish all the upward thrusts to have the same consideration, and W is chosen as $W = \epsilon I$, $\epsilon > 0$, where I is a diagonal matrix with dimension 6 × 6.

The minimization of objective function (10.19) leads to the following analytical solution:

$$X_{\mathrm{opt}} = (\Phi^{\mathrm{T}}\Phi + W)^{-1}\Phi^{\mathrm{T}}Z. \tag{10.20}$$

Now, the matrix $\Phi^{\mathrm{T}}\Phi + W$ is invertible because of the existence of the weighting matrix W, which is positive definite.

From the X_{opt}, the six upward thrust values T_i, $i = 1, 2, 3, 4, 5, 6$ are determined as well as the angular velocities of the six DC motors

$$\omega_i = \sqrt{\frac{T_i}{b_{\mathrm{t}}}}.$$

These ω_i values, $i = 1, 2, 3, 4, 5, 6$, will be used as reference signals ω_i^* for the motor control systems.

10.2.4 Food for Thought

1. Which variables are used to define the attitude of a multi-rotor UAV?
2. Neglecting the actuator dynamics, what are the input and output variables for the attitude control of a multi-rotor UAV? Taking the actuators into consideration, what are the input and output variables?
3. For the DC motor control problem, if the battery voltage is less than expected, will the motor drive increase or decrease the duty cycle to compensate the discrepancy?
4. From the mathematical modelling, do you think that it is important to use a PI controller to control the velocity of each motor?
5. Which constants do you need to determine in order to implement the control system? Which constants do you need to determine for the attitude control system design?
6. Are there any redundancies in the actuators for the hexacopter?

10.3 Cascade Attitude Control of Multi-rotor UAVs

In order to fly a UAV, the closed-loop control of the three Euler angles ϕ, θ, and ψ is necessary. Because of the products of angular velocities in (10.2) and the sinusoidal functions in (10.3), a multi-rotor UAV is a nonlinear system.

In theory, combining (10.2) with (10.3) will lead to three second order nonlinear systems. Therefore, three PID controllers could be adequate for the attitude control applications. However, in practice, a cascade PI or PID controller structure offers a better solution for the following reasons.

1. If the multi-rotor UAV carries a payload [see (10.2)], the load disturbance is much more effectively rejected in a cascade control system structure because it occurs at the secondary plant [see Chapter 7].
2. Looking at Euler's equations of motion (10.1), if the multi-rotor UAV is well designed with a balanced load, the moment of inertia at the x-axis and y-axis equal each other: $I_{xx} = I_{yy}$. However, $I_{zz} \neq I_{xx}$ in general. There are interactions between the variables p and q. The interactions in a PID controlled system would be translated as disturbances. We could compensate their effects using a feedforward control action as shown in Section 3.6 if the parameters I_{xx}, I_{yy}, and I_{zz} are available with reasonable accuracy. Alternatively, we simply neglect them in the PID controller design, and they are automatically compensated in the feedback control using a high gain feedback control. Clearly the bilinear terms in (10.1) act on the torques τ_x, τ_y, and τ_z, and they are regarded as input disturbances to the multi-rotor UAV system. It was shown in Section 7.3 that the cascade control system has a much improved performance in disturbance rejection. The existence of bilinear terms in (10.1) is one consideration for the choice of cascade control system.
3. Because the torques τ_x, τ_y, and τ_z will be realized by the electrical motors installed on the multi-rotor UAV and the motor dynamics are not captured in the motion equations, there will be model uncertainties in the mathematical models having an impact on the closed-loop control. These uncertainties occurring in the secondary plant are better dealt with in a cascade control structure.
4. Above all, the cascade control structure offers a simpler controller design framework because for each stage only first order models or first order plus time delay models are involved.

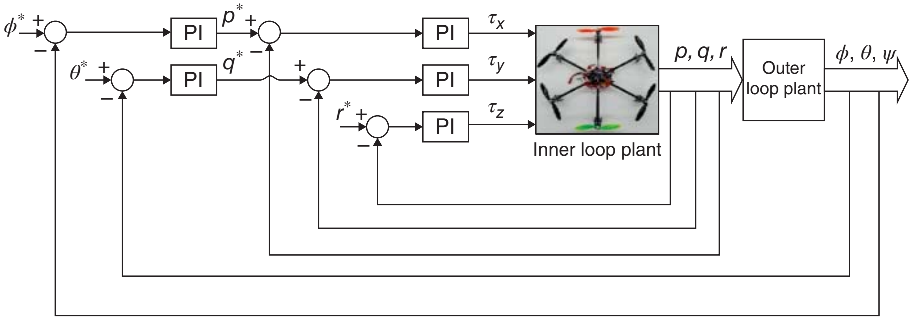


Figure 10.3 Attitude control system structure.

Figure 10.3 shows the cascade control system configuration for the attitude control of a multi-rotor UAV where the body frame angular velocities p, q, and r along the x, y, and z directions are the secondary variables. With this configuration, the secondary system is described by the differential equations given in (10.1), and the primary system is described by (10.3). To control the angular velocities of the multi-rotor UAV, three PI controllers are used to calculate the control signals based on their respective reference signals p^*, q^*, and r^*. Additionally, there are two PI controllers to control the roll and pitch angles, where their reference signals are ϕ^* and θ^*.

10.3.1 Linearized Model for the Secondary Plant

The dynamic models in (10.1) used for the design of secondary controllers are expressed as

$$\begin{aligned}\dot{p} &= \tfrac{1}{I_{xx}}((I_{yy} - I_{zz})qr + \tau_x)\\ \dot{q} &= \tfrac{1}{I_{yy}}((I_{zz} - I_{xx})pr + \tau_y)\\ \dot{r} &= \tfrac{1}{I_{zz}}((I_{xx} - I_{yy})pq + \tau_z)\end{aligned} \tag{10.21}$$

where the load disturbances are neglected. Clearly, the secondary plant is integrating systems with gain inversely proportional to their moment of inertia constant.

If one wishes to use feedforward compensation as in Section 3.6 of Chapter 3, the intermediate variables are defined as

$$\tilde{\tau}_x = (I_{yy} - I_{zz})qr + \tau_x; \quad \tilde{\tau}_y = (I_{zz} - I_{xx})pr + \tau_y; \quad \tilde{\tau}_z = (I_{xx} - I_{yy})pq + \tau_z.$$

With these variables defined, the dynamic models (10.21) for the secondary plant become:

$$\begin{aligned}\dot{p} &= \frac{1}{I_{xx}}\tilde{\tau}_x\\ \dot{q} &= \frac{1}{I_{yy}}\tilde{\tau}_y\\ \dot{r} &= \frac{1}{I_{zz}}\tilde{\tau}_z.\end{aligned} \tag{10.22}$$

The dynamics from actuators, i.e. the rotors, can be modeled as a time delay d with a gain γ_{m} because their time constants are relatively small in comparison with the dynamics from the secondary plant. In short, the secondary plant in the multi-rotor UAV system is approximated by three integrator with time delay models.

10.3.2 Linearized Model for the Primary Plant

For the problem of attitude control, the nonlinear plant for the three Euler angles is required to be linearized around their operating conditions. To maintain stable flight, the reference signals to the roll and pitch angles (ϕ and θ) are chosen to be zero at the steady-state operating conditions while the reference signal to the yaw angle ψ may change according the position reference signals of the multi-rotor UAV. Thus, the linearization of the nonlinear equations (10.3) at the steady-state operating conditions ($\phi^0 = \theta^0 = 0$) gives:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{10.23}$$

With consideration of time delay from the secondary closed-loop system, the primary plant is approximately modeled using an integrator with the time delay model.

Depending on a time delay existing in the system, one may wish to use PID controllers for the secondary or primary plant if closed-loop control performance can be improved with the derivative action.

10.3.3 Food for Thought

1. In the cascade control structure, what are the reference signals for ϕ and θ to ensure a stable flight?
2. How do we generate a reference signal for yaw rate r if the position of a multi-rotor UAV is controlled by an operator?
3. Do we need the total thrust T in the cascade attitude control system?
4. Is it correct to say that we can not change τ_x, τ_y and τ_z directly although they are the manipulated variables? If it is correct, which variables can be changed that will result in the changes in τ_x, τ_y and τ_z?
5. Is it correct to say that the dynamic models for the secondary plant are not accurate because the actuators are not considered?
6. With the cascade control structure, we still face the choice of P, PI, or PID controllers for each loop. Which factors and criteria do you consider for the selection of the controller structures for each individual loop?

10.4 Automatic Tuning of Attitude Control Systems

The dynamic models for the attitude control system are relatively simple with the parameters from the moments of inertia, I_{xx}, I_{yy}, and I_{zz}. However, the parameters associated with the actuators are more complicated if one wishes to measure them accurately. The purpose of using an auto-tuner for the attitude control system is to avoid the time consuming tasks of finding the physical parameters for the secondary and primary plants as well as the actuators. In addition, the dynamics of the closed-loop secondary system are taken into consideration in the primary control system through the application of the cascade auto-tuner.

In order to implement the cascade PI control system with auto-tuner for a multi-rotor UAV, the coefficients b_t, k_d and d_mm are pre-determined so that the reference signals to the rotors in the example of quadrotor UAV are calculated using the equation below:

$$\begin{bmatrix} (\omega_1^*)^2 \\ (\omega_2^*)^2 \\ (\omega_3^*)^2 \\ (\omega_4^*)^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b_\mathrm{t}} & 0 & -\frac{1}{2d_\mathrm{mm}b_\mathrm{t}} & -\frac{1}{4k_\mathrm{d}} \\ \frac{1}{4b_\mathrm{t}} & -\frac{1}{2d_\mathrm{mm}b_\mathrm{t}} & 0 & \frac{1}{4k_\mathrm{d}} \\ \frac{1}{4b_\mathrm{t}} & 0 & \frac{1}{2d_\mathrm{mm}b_\mathrm{t}} & -\frac{1}{4k_\mathrm{d}} \\ \frac{1}{4b_\mathrm{t}} & \frac{1}{2d_\mathrm{mm}b_\mathrm{t}} & 0 & \frac{1}{4k_\mathrm{d}} \end{bmatrix} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{10.24}$$

where T, τ_x, τ_y, and τ_z are the attitude controller outputs, which are used to generate the desired velocities for the rotors. For a DC motor with a commercial drive, a PI controller is often used for controlling its velocity where the desired the velocity reference signal ω_i^*, $i = 1, 2, 3, 4$, is used for each motor.

A similar approach is adopted for the hexacopter by calculating the desired rotor velocities using (10.18).

The auto-tuning algorithms for an integrator with delay systems were developed in Section 9.7. They are used for tuning both primary and secondary controllers in the multi-rotor UAV applications without modifications. More detailed discussions on auto-tuning of attitude control for multi-rotor UAVs can be found in Chen and Wang (2016) and Poksawat and Wang (2017).

10.4.1 Test Rigs for Auto-tuning Cascade PI Controllers of Multi-rotor UAVs

In order to conduct identification experiments in a controlled environment, the quadrotor is fixed on a mechanical stance for the testing on ground in order to ensure safety of the electronics during the testing process. The test rig is shown in Figure 10.4, which was built and used to conduct the relay experiments for identification of the quadrotor's two-axis dynamics. For instance, with the objective to identify the integrator with a delay model for the roll angle, two quadrotor arms along the x-axis are fixed on the stand and the quadrotor can only rotate about the x-axis. Furthermore, the test rig is carefully adjusted to make rotating axis aligned with the quadrotor's body frame axis, so that the torque due to weight force is minimized. As the quadrotor platform is very light and the two rotating pivots are very smooth, the friction is negligible in the experiments.

A similar test rig is also built for the hexacopter as shown in Figure 10.5.

10.4.2 Experimental Results for Quadrotor UAV

The quadrotor consists of five main components: RC transmitter/receiver, IMU sensor board, data logger, microprocessor, and actuators. The RC transmitter/receiver is to send and receive reference signals. The IMU sensor board is to measure the Euler angles

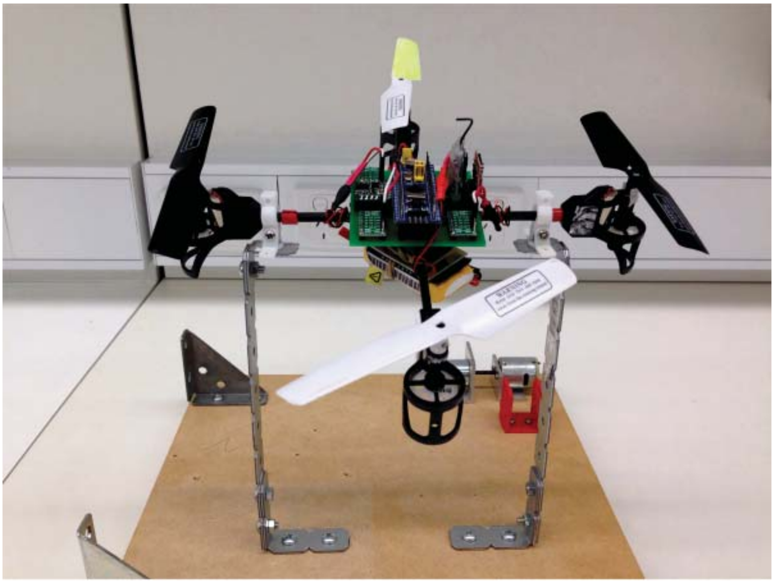


Figure 10.4 Quadrotor test-bed.

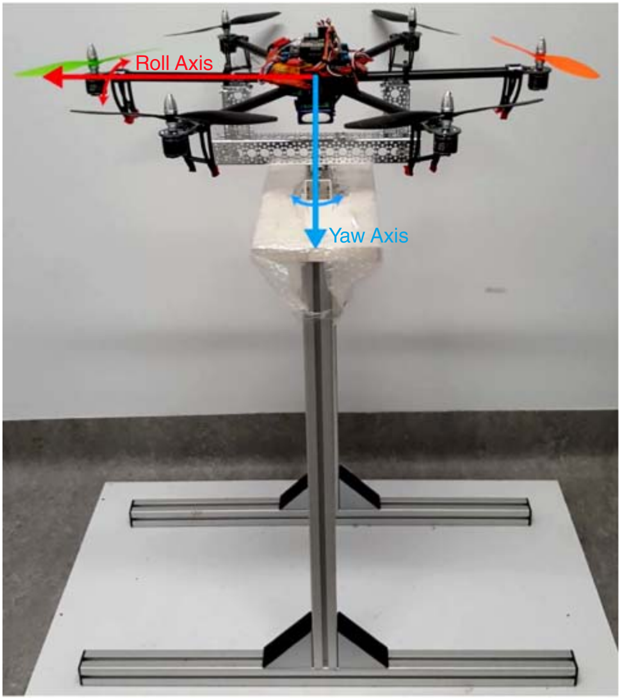


Figure 10.5 Experimental rig for a hexacopter.

Table 10.1 Quadrotor hardware list.

Function	Model
DC motor drive	DRV8833 Dual Motor Driver Carrier
Sensor board	MPU6050
Micro processor	STM32F103C8T6
RC receiver	WFLY065
DC motor	820 Coreless Motor
RC transmitter	WFT06X-A
Data logger	SparkFun OpenLog

and angular velocities. The data logger is to record flight data such as Euler angles and reference signals. The micro-processor is to generate control signals to stabilize the UAV's attitude. The actuators are to generate thrust and torques, which consist of motor drives, DC motors, gearboxes, and blades. The quadrotor's hardware used in the experimental tests is listed in Table 10.1.

The sampling interval Δt for the secondary, primary controllers and relay test are all set to be 0.01 s, which is the IMU sensor's maximum updating rate. The other physical parameters of the quadrotor are shown in Table 10.2.

The auto-tuning of the cascade PI control system begins at the inner loop. Before the relay control experiment, a proportional controller $K_{\mathrm{T}_1} = 0.06$ is selected to stabilize the secondary system. The sensor measurement of velocity, $\dot{\theta}(t)$, contains noise. Thus, a relay amplitude of 0.8 together with a hysteresis level of 0.1 is selected to reflect the measurement noise level. Figure 10.6 shows a segment of the relay feedback control data. From the input signal to the inner-loop, closed-loop control system, the period of the sustained oscillations is identified as $N = 28$ samples, leading to the fundamental frequency

Table 10.2 Quadrotor parameters.

Parameters	Description	Value	Unit
I_{xx}	Moment of inertia about x–axis	3.2×10^{-4}	kg m^2
I_{yy}	Moment of inertia about y–axis	3.2×10^{-4}	kg m^2
I_{zz}	Moment of inertia about z–axis	4.6×10^{-4}	kg m^2
b_{t}	Thrust constant	9.7×10^{-7}	N/A
k_{d}	Drag constant	2.5×10^{-9}	N/A
m	Quadrotor total mass	0.145	kg
d_{mm}	Motor to mass center distance	0.110	m
V_{bat}	Battery voltage	8.28	V
r_{wv}	Motor's DC gain	137.6571	rad V^{-1} s^{-1}
w_0	Rotor normal speed	606.2469	rad s^{-1}
d_{m}	Motor delay	0.032	s
ϵ_{m}	Motor time constant	0.072	s

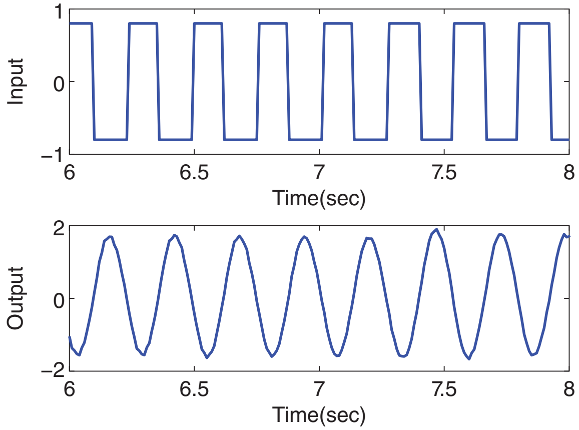


Figure 10.6 Relay feedback control signals from the inner-loop system: top figure, input signal; bottom figure, output signal.

in the frequency sampling filter as $\frac{2\pi}{N}$ (rad). The use of the frequency sampling filter based estimation algorithm gives the inner-loop, closed-loop frequency response as

$$T(e^{j\frac{2\pi}{N}}) = -1.225 - j0.5137.$$

Converting this discrete-time frequency to continuous time frequency, which is $\omega_1 = \frac{2\pi}{N\Delta t} = 22.44$ (rad s^{-1}), together with the knowledge of the proportional controller used in the relay experiment ($K_{T_1} = 0.06$), the continuous-time frequency response of the inner-loop plant is calculated as

$$G_1(j\omega_1) = \frac{1}{K_{T_1}} \frac{T(e^{j\frac{2\pi}{N}})}{1 - T(e^{j\frac{2\pi}{N}})} = -9.551 - j1.6421.$$

From this frequency information, an integrator plus delay model is identified as

$$G_1(s) = \frac{217.5589e^{-0.0624s}}{s}.$$

By choosing the desired closed-loop time constant as three times the estimated delay: $\tau_{cl} = \beta d = 3d = 0.1872$ (s) and damping coefficient of 1, from the empirical rules shown in Section 8.4.3, the PI controller parameters are found for the inner-loop control system as $K_c = 0.0343$ and $\tau_I = 0.4487$. This set of PI controller parameters approximately gives a gain margin of 3 and phase margin of 48° for the closed-loop system with the integrator plus delay model.

Figure 10.7 shows the closed-loop step response of $\dot{\theta}(t)$ where the reference signal has a magnitude of 0.5 (rad s^{-1}). It is seen from this figure that the closed-loop velocity response follows the reference signal without steady-state error, and there is a large overshoot and a slight oscillation. Additionally, there are disturbances and measurement noise in the inner-loop system. For a cascade control system, the inner-loop control system is required to have a fast response speed, which is achieved in the design here.

The second step in auto-tuning the cascade control system is to find the outer-loop controller. For the outer-loop experiment, the proportional controller $K_{T_2} = 2$ is used to stabilize the integrator with delay system. The amplitude of the relay is selected to be 0.4 and the hysteresis level ϵ is 0.05 to prevent the relay from random switching. Figure 10.8 shows a segment of the input and output data generated from this relay

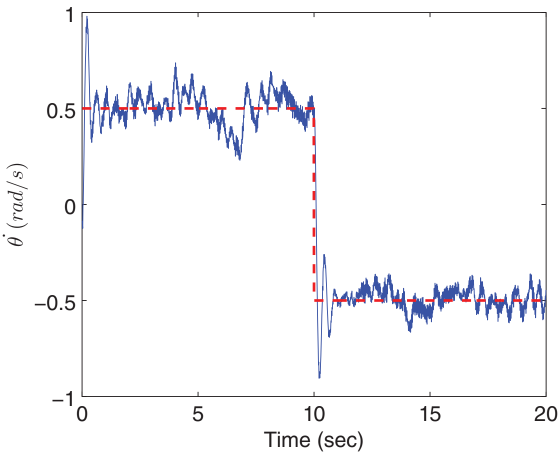


Figure 10.7 Inner-loop step response in closed-loop control. Dashed line, reference signal; solid line, output.

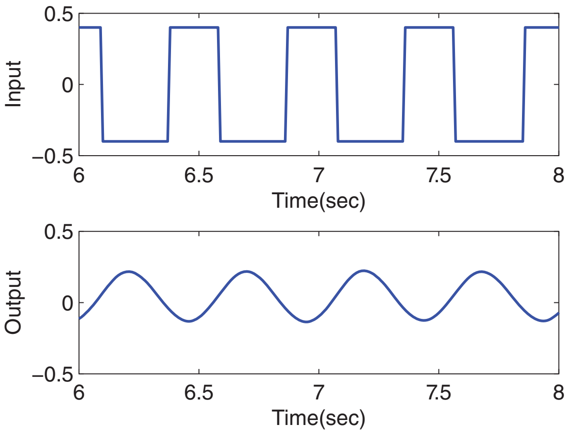


Figure 10.8 Relay feedback control signals from outer-loop system: top figure, input signal; bottom figure, output signal.

feedback control of the primary plant under proportional control. The averaged period of the sustained oscillation is $N = 59$ in number of samples, which gives the fundamental frequency in discrete time as $\frac{2\pi}{59}$ (rad). A frequency sampling filter model is used to estimate the closed-loop frequency response based on the set of input and output data shown in Figure 10.8, yielding to

$$T(e^{j\frac{2\pi}{N}}) = -0.1317 - j0.3130.$$

With the proportional controller $K_{T_2} = 2$, the frequency response of the outer-loop system is found at $\omega_1 = \frac{2\pi}{N\Delta t} = 10.649$ (rad s^{-1}) as

$$G_2(j\omega_1) = \frac{1}{K_{T_2}} \frac{T(e^{j\frac{2\pi}{N}})}{1 - T(e^{e\frac{2\pi}{N}})} = -0.0896 - j0.1135.$$

From this frequency information, the integrator plus delay model for the primary system is calculated as

$$G_2(s) = \frac{1.54e^{-0.0627s}}{s}. \tag{10.25}$$

For a typical cascade control system design, the outer-loop control system should have a slower desired closed-loop response than that of the inner-loop control system. By selecting the desired closed-loop time constant as eight times the delay value: $\tau_{\mathrm{cl}} = \beta d = 8.5d$ with damping coefficient $\xi = 1$ which gives gain margin ≈ 7, and phase margin $\approx 63°$ (see Section 8.4.3, the PI controller parameters are calculated using the empirical rules:

$$K_{\mathrm{c}} = 2.095; \quad \tau_{\mathrm{I}} = 1.137.$$

For comparison purposes, with a faster desired closed-loop time constant $\tau_{\mathrm{cl}} = 5d$ and a slower desired closed-loop time constant $\tau_{\mathrm{cl}} = 10d$, two additional sets of PI controller parameters are calculated as $K_{\mathrm{c}} = 3.27$, $\tau_{\mathrm{I}} = 0.7006$, and $K_{\mathrm{c}} = 1.8155$, $\tau_{\mathrm{I}} = 1.3244$.

Figure 10.9 shows the comparative closed-loop responses for the three cases experimentally. It is seen from the comparative results that all three PI controllers lead to stable closed-loop systems. Clearly when $\beta = 5$, the fastest closed-loop response is obtained. A sequence of step reference changes is applied to the roll angle for a further experimental test, as shown in Figure 10.10, which shows a fast response with an overshoot.

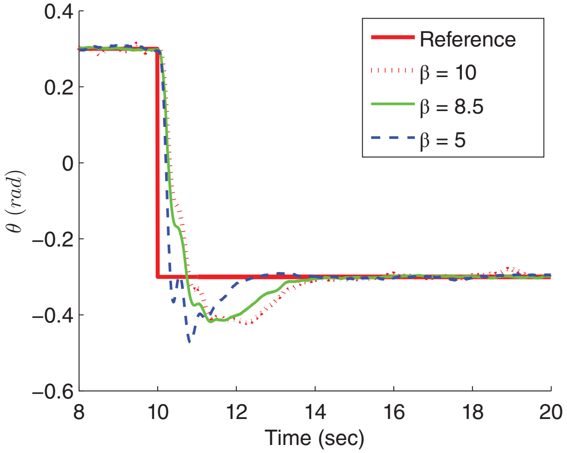


Figure 10.9 Comparative roll angle step response in closed-loop control.

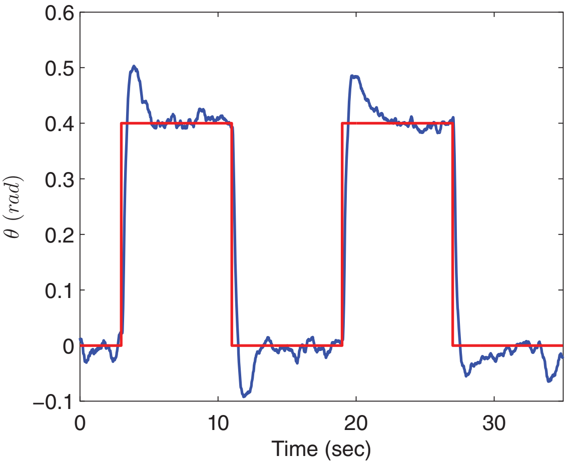


Figure 10.10 Roll angle step response of quadrotor using test rig.

We note that the estimated integrator with delay model for the roll angle given in (10.25) has a gain of 1.54, which is much larger than the expected value of 1 because the secondary closed-loop system should have unit gain under PI control. This could be caused by the existence of the nonlinearity or how the IMU sensor behaves when the roll angle swings with a large amplitude.

10.4.3 Experimental Results for Hexacopter

A hexacopter is built for experimental testing of the auto-tuner and the cascade attitude control system (Poksawat and Wang (2017)). The flight controller specifications and avionic components are presented in Table 10.3. The physical parameters are presented in Table 10.4.

To validate the proposed strategy, an automatic tuning experiment is performed on the hexacopter and the results are presented here. Firstly, the roll angular rate controller parameters are tuned with the relay test. It is assumed that the airframe is symmetrical, thus the pitch controller parameters are selected to be identical to those obtained in the experiments from the roll axis. For the yaw angular rate loop, the tuning procedure will follow the same approach. The sampling interval Δt is chosen to be 0.006 s.

Table 10.3 Flight controller and avionic components.

Components	Descriptions
Airframe	Turnigy Talon Hexacopter
Microprocessor	ATMega2560
Inertial measurement unit	MPU6050
Electronic speed controllers	Turnigy 25A Speed Controller
Brushless DC motors	NTM Prop Drive 28-26 235W
Propellers	10x4.5 SF Props
RC receiver	OrangeRX R815X 2.4Ghz receiver
RC transmitter	Turnigy 9XR PRO transmitter
Datalogger	CleanFlight Blackbox Datalogger

Table 10.4 Physical specifications of the hexacopter.

Parameters	Details
Mass (m)	1.61 kg
Arm length (ℓ)	0.3125 m
Blade radius (r)	0.127 m
Moment of inertia (I_{xx})	0.2503 kg m^2
Moment of inertia (I_{yy})	0.2914 kg m^2
Moment of inertia (I_{zz})	0.6177 kg m^2
Thrust constant (b_t)	1.562e^{-5}
Torque constant (k_d)	0.0209

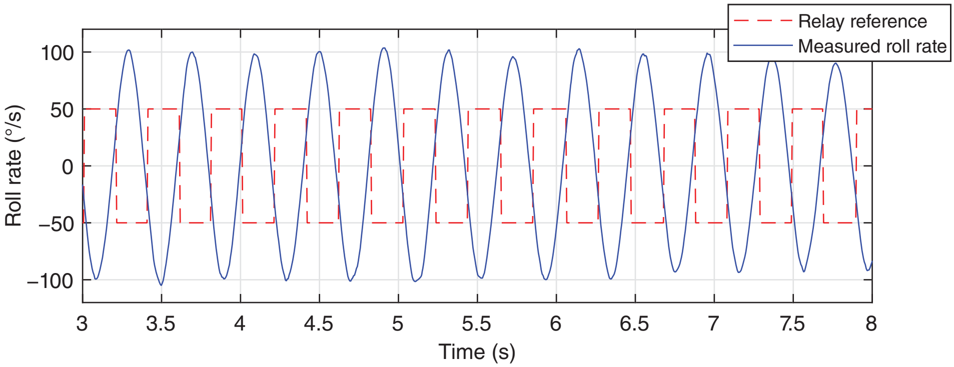


Figure 10.11 Inner loop relay test result.

For the automatic tuning of the secondary controller, which is the roll angular rate, the proportional gain used in the relay test is chosen as $K_\mathrm{t} = 0.3$ for stabilization of the integral system. The amplitude of the relay reference signal has to be within the operating condition of the hexacopter UAV, thus it is selected as 50° s^{-1}. The hysteresis is chosen to be $\epsilon = 30°\ \mathrm{s}^{-1}$ to prevent the relay switching from measurement noise.

A section of the relay feedback experimental data is presented in Figure 10.11. The period of the sustained oscillations is calculated to be $N = 70$, leading to the fundamental frequency in the sampling filter of $\frac{2\pi}{N} = 0.0898$ rad.

The inner loop frequency response is then estimated using frequency sampling filter estimation algorithm as

$$T(\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}}) = -1.382 - \mathrm{j}0.6641.$$

The continuous time frequency can be calculated as $\omega_1 = \frac{2\pi}{N\Delta t} = 14.96$ (rad s^{-1}). From this, the continuous time frequency response of the secondary plant is

$$G(j\omega_1) = \frac{1}{K_\mathrm{t}} \frac{T(\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}})}{1 - T(\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}})} = -2.035 - \mathrm{j}0.362.$$

The integrator plus time delay transfer function is obtained as

$$G(s) \approx \frac{K_p \mathrm{e}^{-ds}}{s} = \frac{30.92\mathrm{e}^{-0.09s}}{s}.$$

For the hexacopter, we decided to use a PID controller instead of a PI controller because the delay is quite large and the derivative term improves the closed-loop response. In order to achieve fast closed-loop system response, the desired closed-loop time constant is selected to be relatively small. Here, β is selected as 1, which leads to an approximate gain margin of 2 and phase margin of 37° respectively. The controller parameters for the roll angle rate system are then obtained as $K_\mathrm{c} = 0.33$, $\tau_\mathrm{I} = 0.26$, and $\tau_\mathrm{d} = 0.03$ from using the empirical rules presented in Section 8.4.3.

Once the inner loop controller is tuned, the outer loop controller's parameters can be selected with the following procedures. Instead of automatic tuning of the primary controller, the mathematical model for the primary plant is approximated by the following integrator with delay system:

$$G(s) = \frac{\mathrm{e}^{-(\tau_\mathrm{cl}+d)}s}{s}$$

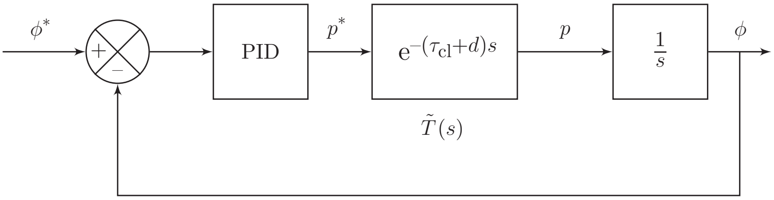


Figure 10.12 Attitude control system with approximated inner loop.

where the integrator is from the primary plant (roll angular rate to roll angle) and the time delay is from the approximation of the closed-loop secondary system, where τ_{cl} is the closed-loop time constant and d is the time delay from the inner-loop. Figure 10.12 shows the primary control system.

Generally, for a control system with a cascaded loop configuration, the outer loop response time needs to be slower than the inner loop. Hence, the outer loop time constant is chosen to be twice of the time delay, leading to the controller parameters for the angular position loop as $K_c = 3.3$, $\tau_I = 0.63$, and $\tau_d = 0.013$.

An outdoor flight test was conducted to validate the stability of the UAV in real flight against external disturbances such as turbulence (see Figure 10.13). The roll, pitch, and yaw data obtained from the flight test are presented in Figures 10.14, 10.15, and 10.16 respectively. The dashed lines represent the reference signals and the solid lines represent the measured flight data.

It is clearly seen that the hexcopter is able to follow the pilot's commands, due to the fact that the outputs are regulated closely to their references. Furthermore, it is able to hover, roll, pitch, and yaw while maintaining stability.

Figure 10.13 Outdoor flight test.

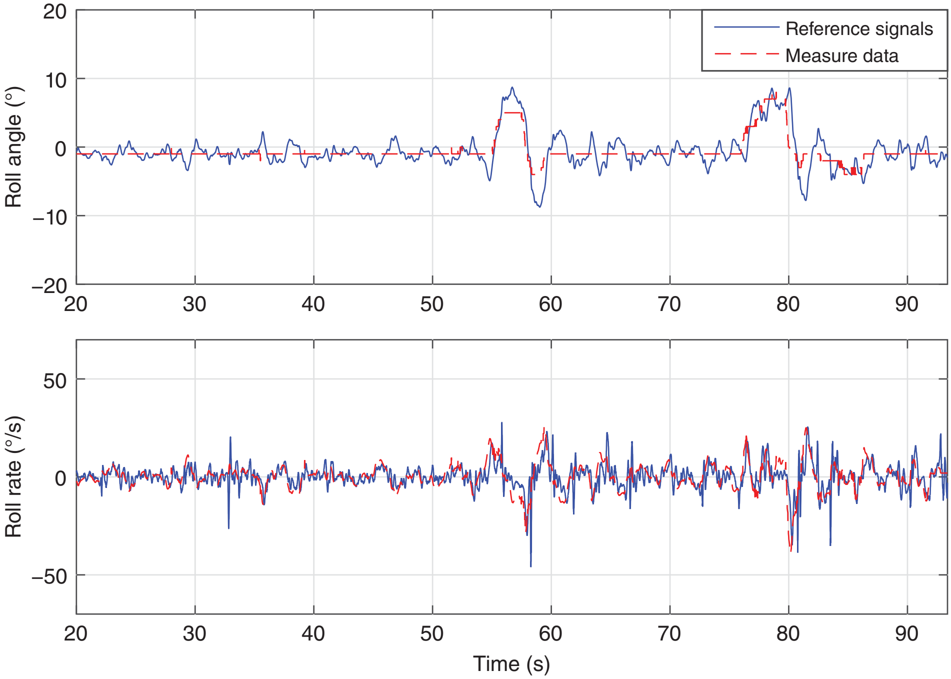


Figure 10.14 Flight data for roll axis.

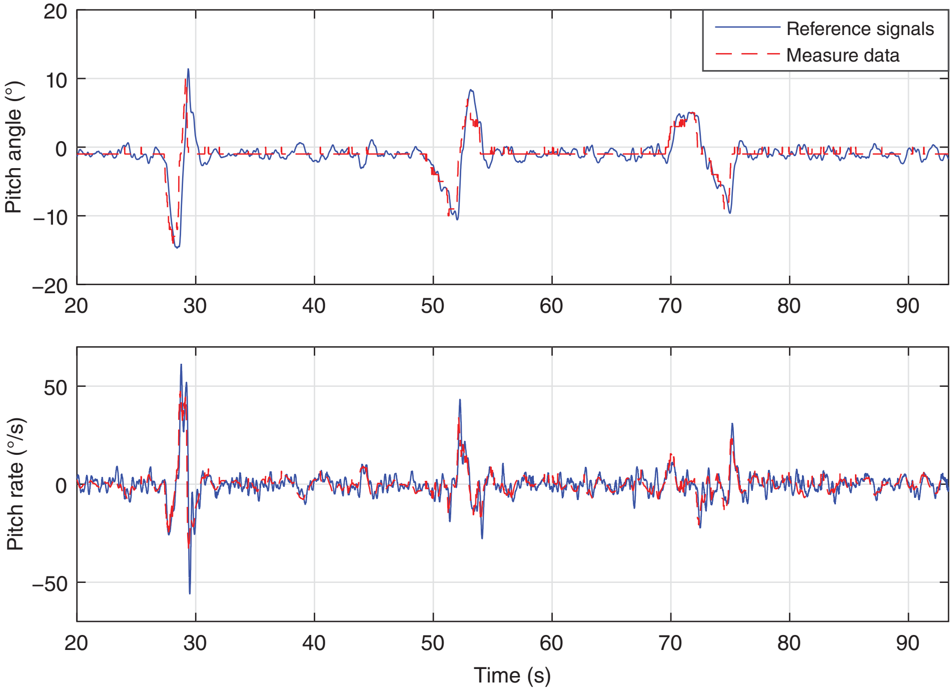


Figure 10.15 Flight data for pitch axis.

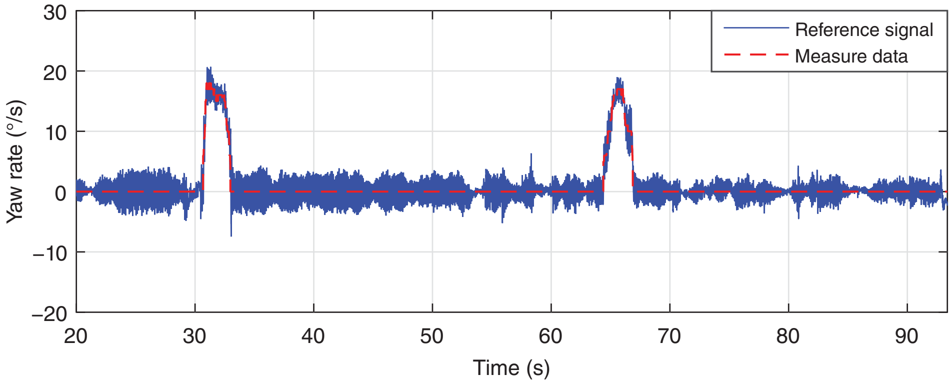


Figure 10.16 Flight data for yaw axis.

10.4.4 Food for Thought

1. What are the steps required to implement the auto-tuners?
2. Is it correct to say that the relay experiments because they are conducted in closed-loop with a feedback controller K_T, are relatively simple to implement using the existing cascade control structure of a multi-rotor UAV?
3. If the parameters k_d, k_t and d_{mm} are not precisely measured, will the auto-tuner compensate the errors because of the experiments are carried out on the actual physical system?
4. Have the integrator with delay models for the secondary plant included the actuator dynamics?
5. If an integrator is included in the controller for the secondary plant, what is the steady-state gain of the primary plant? What is the minimum value of the estimated time delay for the primary plant?
6. Will PD controllers be adequate for controlling both secondary and primary plants of the multi-rotor UAV? What problems would you envisage that might occur?

10.5 Summary

We have discussed PID control of multi-rotor unmanned aerial vehicles in this chapter. Dynamic models for both quadrotor UAV and hexacopter UAV are discussed from the control system design point of view. Cascade control system structures are proposed for both quadrotor and hexacopter UAVs. The auto-tuning algorithms introduced in Chapter 9 are used to find the PID controller parameters for the UAVs on test rigs. The cascade PID control systems are experimentally evaluated through outdoor flight tests.

The other important aspects of the chapter are summarized as follows.

- Both inner-loop and outer-loop systems for the unmanned aerial vehicles are modelled using integrator with time delay systems.
- A proportional controller is used to produce a stable closed-loop system prior to the implementation of auto-tuner.
- Both PID controller and auto-tuner implementations on the UAVs have used micro-controllers. The programs for PID implementations have included

anti-windup mechanisms and are based on Tutorial 4.1. The auto-tuner implementations are based on Tutorial 9.1 for the relay feedback control.

- The estimation of frequency response of the UAVs is performed based on Tutorial 9.4.
- Upon obtaining the estimation of the frequency response, we can follow the computational examples presented in Section 9.7 to calculate the PID controller parameters.

10.6 Further Reading

1. Books for unmanned aerial vehicles include Beard and McLain (2012), Fahlstrom and Gleason (2012), Austin (2011) and Gundlach (2012).
2. Mathematical modelling and control of hexacopter were presented in Alaimo et al. (2013). Detailed nonlinear modelling together with control was presented in Bangura and Mahony (2012). Nonlinear control of quadrotor UAV was presented in Goodarzi et al. (2013). A survey of control methods was presented in Li and Song (2012).
3. Automatic tuning of the PID attitude control systems for the quadrotor UAV presented in this chapter was designed and experimentally validated in Chen and Wang (2016) and in Chen (2017). The closed-loop performance was assessed on ground using the same test rig based on system identification of the closed-loop transfer function (Chen and Wang (2015)). Automatic tuning of the PID attitude control systems for the hexacopter presented in this chapter was designed and experiementally validated in Poksawat and Wang (2017). Automatic tuning of PID attitude control systems for a micro fixed-wing UAV can be found in Poksawat et al. (2016), Poksawat et al. (2017) and Poksawat (2018).
4. Optimization based tuning method was used to find controllers for an auto-pilot of fixed-wing UAV (Ahsan et al. (2013)).
5. Model predictive control system was designed and experimentally validated for the hexacopter presented in this chapter (Ligthart et al. (2017)).

Problems

10.1 *The dynamic models for a multi-rotor UAV without considering the actuator dynamics are described by the following differential equations (see (10.2)–(10.3)):*

$$\begin{aligned} I_{xx}\dot{p} =&(I_{yy}-I_{zz})qr+\tau_x-\tau_x^d \\ I_{yy}\dot{q} =&(I_{zz}-I_{xx})pr+\tau_y-\tau_y^d \\ I_{zz}\dot{r} =&(I_{xx}-I_{yy})pq+\tau_z-\tau_z^d \end{aligned} \tag{10.26}$$

and

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{10.27}$$

where the moments of inertial constants are $I_{xx} = 3.6 \times 10^{-2}$ *(kg.m^2),* $I_{yy} = 3.8 \times 10^{-2}$ *(kg.m^2) and* $I_{zz} = 4.6 \times 10^{-2}$ *(kg.m^2).*

1. *Build a Simulink simulation model for the multi-rotor UAV without actuator dynamics.*
2. *With zero initial conditions for all the state variables θ, ϕ, ψ and p, q, r, compute the open-loop responses of the states to the load disturbances $\tau_x^d = 0.01$, $\tau_y^d = 0.02$ and $\tau_z^d = 0.2$. Here, we assume that the reference signals to the open-loop system are $\theta^* = \phi^* = \psi^* = 0$ and the sampling interval is $\Delta t = 0.0001$ (sec).*
3. *What are your observations on the system dynamics from this open-loop simulation exercise?*

10.2 *Continue from Problem 10.1. Design a cascade PI control system for the multi-rotor UAV. The desired closed-loop poles for the secondary plant are all chosen to be -100 and for the primary plant are all chosen to be -20. Use the model based PI controller design method in Chapter 3 to find the PI controller parameters.*

1. *Choosing the reference signals $\theta^* = \phi^* = 0$ and $\psi^* = \frac{\pi}{3}$, and simulate the cascade closed-loop control system's responses to the reference following and disturbance rejection. You may use a smaller sampling interval to improve the numerical stability in the simulation.*
2. *Increase the load torque τ_z^d to 0.4 and observe how the control signals τ_x, τ_y and τ_z change in comparison to the responses from the original load of 0.2.*
3. *Vary the desired closed-loop poles for both secondary and primary control systems and observe the changes on the control signals τ_x, τ_y and τ_z, and output signals θ, ϕ, ψ.*

10.3 *Continue from Problem 10.1. Instead of the cascade control structure, design three PID controllers using the linearized models from (10.26) and (10.27). For simplicity, we may choose all closed-loop poles at $-\lambda$ and tune the parameter λ to achieve closed-loop stability from the nonlinear control system simulation. The pole-assignment controller design introduced in Chapter 3 is used to find the parameters of the PID controller with filter.*

1. *What are the λ values found to achieve closed-loop stability without oscillations?*
2. *What are your observations on the disturbance rejection when comparing this control system with the previous cascade control system?*

10.4 *Modify the Simulink simulator built from Problem 10.1 to consider the actuator dynamics. For simplicity, the actuator dynamics for each axis are modeled using the delay model ke^{-ds}. Choosing $d = 0.007$ second, through the nonlinear system simulation determine the range of k for the closed-loop stability of the cascade control system.*

10.5 *Using the physical parameters from Table 10.2, we build a Simulink simulation model for the quadrotor UAV with actuators. Perform automatic tuning of the PID controllers for the quadrotor using simulation studies. The closed-loop time constants for the secondary controllers are selected as $2d$ where d is the estimated delay and for the primary controllers are selected as $4d$.*

Suggestions to Food for Thought Questions

Chapter 1
Section 1.2.6. 1 K_c could be both positive and negative, τ_I positive, and τ_D positive. 3 Its main role is to compensate steady-state errors, however, it also increases dynamic response speed. 5 Yes. This is a quite simple and effective approach to reduce overshoot in the reference response.
Section 1.3.3. 2 No, because it is not safe to conduct open-loop experiments for unstable systems and the tuning rules are derived for stable systems. 4 System could easily become unstable. 6 Time delay, time constant and steady-state gain. 8 Cohen-Coon tuning rules produce larger K_c, smaller τ_I and τ_D.
Section 1.4.4. 1 Decrease. 3 No, decrease.

Chapter 2
Section 2.2.3. 1 Not oscillatory. 3 Because the coefficients of the denominator are real. 5 No.
Section 2.3.3. 2 Yes. 4 Combination of K_c, τ_I and τ_D.
Section 2.4.4. 1 Yes. 3 It is simpler for implementation. 5 $\frac{1}{\beta\tau_D s+1}$.
Section 2.5.4. 2 Yes. 4 Because it can achieve a faster disturbance rejection without causing a larger overshoot in reference following.
Section 2.6.3. 1 No. 3 No.
Section 2.7.4. 2 Yes. 2 Yes.

Chapter 3
Section 3.3.1. 2 No. 4 Current control in AC drives; liquid level control; temperature control.
Section 3.4.6. 1 No. 3 Yes. 5 No. 7 Sylvester matrix is invertible.
Section 3.5.4. 2 Yes. $4s(s^2 + \omega_0^2)$.
Section 3.7.1. 1 Yes. 3 Open-loop compensation technique; faster and better disturbance rejection through measurement.

Chapter 4
Section 4.3.3. 2 Yes. 4 Reduce the sampling interval Δt. 6 Yes.
Section 4.4.4. 1 At the initial step when the closed-loop control starts. 3 Constant plant operational condition.
Section 4.5.3. 2 No. 4 Before the saturation block.

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

Section 4.6.3. 1 The control signal computed is the actual variable to be implemented; the integral action needs to be stopped when the control signal reaches its limits. 3 Control signal behaves like a ramp signal.

Chapter 5

Section 5.2.5. 2 Yes. 4 Yes.
Section 5.3.6. 1 See Tutorial 5.2. 3 Add the control signal steady-state before implementing the saturation limits and the disturbance observer (see Figure 5.2). The output steady-state should be considered together with the steady-state of the reference signal.
Section 5.4.6. 2 Yes. 4 The implementation uses a stable structure and the calculated control signal equals the actual control signal when its steady state is added.
Section 5.5.3. 1 Yes. 3 Yes.

Chapter 6

Section 6.2.5. 2 No. 4 Yes. 6 Yes.
Section 6.3.5. 1 Yes. 3 The actuator dynamics, the sensor errors, and errors from the physical parameters.
Section 6.4.4. 1 Yes, as long as they are measurable. 3 Yes.

Chapter 7

Section 7.2.4. 2 For simplicity of the primary controller design. 4 Yes.
Section 7.3.3. 2 Yes. 4 Because the original PI controller implementation will lead to a faster inner-loop dynamic response for a better performance in the cascade control system.
Section 7.4.4. 1 Yes. 3 Yes.

Chapter 8

Section 8.2.3. 2 Yes. 4 Yes.
Section 8.3.6. 1 Yes. 3 Yes, however the two frequency points should not be too close to each other. The approach is based on matching the frequency response of the Nyquist curve at the two selected points.
Section 8.4.6. 2 $\frac{1}{(2\xi\tau_{cl}+d)s+1}$.

Chapter 9

Section 9.2.3. 1 Use steady-state experimental data. 3 Yes. The data can be used for estimation of the first order model. A simple approach is to incorporate a delay in the feedback path, where the input and output signals are only related to the original first order dynamics. 5 Yes, but can not guarantee to produce a stable closed-loop system in general.
Section 9.3.4. 2 Yes. 4 Because the input and output data do not contain the frequency information at $\omega = 0$.
Section 9.4.4. 1 Because the maximum peak value of the FFT of the input signal reliably carries information about the fundamental frequency of the periodic signal. The alternative approach is to take an average of the periods of the input signal, which is not as reliable because of the possible random switches due to noise and disturbances. 3 On the unit circle of the complex plane. Yes. 5 It offers a real-time implementation platform and can adapt to parameter changes in the system.

Section 9.5.4. 2 Yes. Yes. 4 Because the input and output signals do not contain the frequency information at $\omega = 0$. Even with the low frequency disturbance, the estimation at $\omega = 0$ is not reliable as shown in the Monte-Carlo simulation studies.
Section 9.6.4. 1 High accuracy with the estimated frequency parameters, and no need for conversion to a transfer function model. 3 Decrease. 5 Yes.
Section 9.7.4. 2 Yes. 4 Decrease.

Chapter 10

Section 10.2.4. 1 roll angle ϕ, pitch angle θ and yaw angle ψ. 3 From (10.11), when the battery voltage is less than expected, in order to maintain the same motor speed, the duty cycle needs to increase. 5 To implement the control system, the constants for the actuator dynamics listed in Sections **10.2.2** and **10.2.3** are required together with the controller parameters. To design the control system, the moments of inertia, I_{xx}, I_{yy} and I_{zz} are required, although it would be ideal to include the actuator dynamics into the attitude control system design.
Section 10.3.3. 2 Through remote control transmitter/receiver for sending and receiving position reference signals. 4 Yes. The velocities for the DC motors. 6 The secondary control systems could use P, PI and PD depending on the actuator dynamics and sensor noise. The primary control systems could use PI or PID. Integrator is needed in the primary controller to compensate for the steady-state errors. The selection of the controller structure could be performed with flight experiments.
Section 10.4.4. 1 Setup ground test apparatus for the relay experiments, setup flight controller in cascade control structure that contains the estimated constants for the actuators. 3 Yes. 5 Unity. Minimum time delay is the delay from the secondary system.

Bibliography

Ahsan, M., Rafique, K., and Mazhar, F. (2013). Optimization based tuning of autopilot gains for a fixed wing UAV. *World Academy of Science, Engineering and Technology* 77: 781–786.

Alaimo, A., Artale, V., Milazzo, C. et al. (2013). Mathematical modeling and control of a hexacopter. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1043–1050. IEEE.

Alcántara, S., Vilanova, R., and Pedret, C. (2013). Control in terms of robustness/performance and servo/regulator trade-offs: a unifying approach to balanced autotuning. *Journal of Process Control* 230 (4): 527–542.

Alfaro, V.M. and Vilanova, R. (2016). *Model-Reference Robust Tuning of PID Controllers*. Springer.

Alfaro, V.M., Vilanova, R., and Arrieta, O. (2009). Robust tuning of two-degree-of-freedom (2-DoF) PI/PID based cascade control systems. *Journal of Process Control* 190 (10): 1658–1670.

Ang, K.H., Chong, G., and Li, Y. (2005). Control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology* 130 (4): 559–576.

Araki, M. and Taguchi, H. (2003). Two-degree-of-freedom PID controllers. *International Journal of Control, Automation and Systems* 10 (4): 401–411.

Astrom, K.J. and Hagglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica* 20: 645–651.

Astrom, K.J. and Hagglund, T. (1988). *Automatic Tuning of PID Controllers*. Research Triangle Park, NC: Instrument Society of America.

Astrom, K.J. and Hagglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, NC: Instrument Society of America.

Astrom, K.J. and Hagglund, T. (2006). *Advanced PID Control*. Research Triangle Park, NC: Instrument Society of America.

Astrom, K.J. and Murray, R.M. (2008). *Feedback Systems: An Introducton for Scientistis and Engineers*. Princeton University PRESS.

Åström, K.J. and Hägglund, T. (2001). The future of PID control. *Control Engineering Practice* 90 (11): 1163–1175.

Åström, K.J. and Hägglund, T. (2004). Revisiting the Ziegler–Nichols step response method for PID control. *Journal of Process Control* 140 (6): 635–650.

Atherton, D.P. (1975). *Nonlinear Control Engineering-Describing Function Analysis and Design*. London: Van Nostrand Reinhold Co.

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.
© 2020 John Wiley & Sons Ltd. Published 2020 by John Wiley & Sons Ltd.
Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

Austin, R. (2011). *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*, vol. 54. Wiley.

Bangura, M. and Mahony, R. (2012). Nonlinear dynamic modeling for high performance control of a quadrotor. In: *Australasian Conference on Robotics and Automation*, 1–10. Australian Robotics and Automation Association.

Bay, J.S. (1999). *Fundamentals of Linear State Space Systems*. McGraw-Hill.

Beard, R.W. and McLain, T.W. (2012). *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press.

Berner, J., Soltesz, K., Hägglund, T., and Åström, K.J. (2018). An experimental comparison of PID autotuners. *Control Engineering Practice* 73: 124–133.

Bitmead, R.R. and Anderson, B.D.O. (1981). Adaptive frequency sampling filters. *IEEE Transactions on Circuits and Systems* 28: 524–533.

Blevins, T.L. (2012). Advances in industrial control. *IFAC Proceedings Volumes* 450 (3): 23–28.

Böling, J.M., Seborg, D.E., and Hespanha, J.P. (2007). Multi-model adaptive control of a simulated pH neutralization process. *Control Engineering Practice* 150 (6): 663–672.

Bouabdallah, S., Noth, A., and Siegwart, R. (2004). *Proceedings.2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004)*, vol. 3, 2451–2456. IEEE.

Burden, R.L. and Faires, J.D. (1989). *Numerical Analysis*. Boston, MA: PWS-KENT Publishing Company.

Cetin, M. and Iplikci, S. (2015). A novel auto-tuning PID control mechanism for nonlinear systems. *ISA Transactions* 58: 292–308.

Chen, D. and Seborg, D.E. (2002). PID controller design based on direct synthesis and disturbance rejection. *Industrial and Engineering Chemistry Research* 410 (19): 4807–4822.

Chen, W.-H., Ballance, D.J., Gawthrop, P.J., and O'Reilly, J. (2000). A nonlinear disturbance observer for robotic manipulators. *IEEE Transactions on Industrial Electronics* 470 (4): 932–938.

Chen, W.-H., Yang, J., Guo, L., and Li, S. (2016). Disturbance-observer-based control and related methods: an overview. *IEEE Transactions on Industrial Electronics* 630 (2): 1083–1095.

Chen, X. (2017). Flight controller design, automatic tuning and performance evaluation of quadrotor UAVs. PhD thesis. RMIT University.

Chen, X. and Wang, L. (2015). Step response identification of a quadcopter UAV using frequency-sampling filters. *IFAC-PapersOnLine* 480 (28): 122–127.

Xi, C. and Wang, L. (2016). Quadrotor cascade PID controller automatic tuning. In: *2016 Australian Control Conference (AuCC)*, 311–316. IEEE.

Cominos, P. and Munro, N. (2002). Controllers: recent tuning methods and design to specification. *IEE Proceedings - Control Theory and Applications* 1490 (1): 46–53.

Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, vol. 73. Springer. ISBN: 3642201431.

Crowe, J. and Johnson, M.A. (2002). Towards autonomous PI control satisfying classical robustness specifications. *IEE Proceedings - Control Theory and Applications* 1490 (1): 26–31.

Cunha, M.A.B., Guenther, R., De Pieri, E.R., and De Negri, V.J. (2002). Design of cascade controllers for a hydraulic actuator. *International Journal of Fluid Power* 30 (2): 35–46.

De Paor, A.M. and O'Malley, M. (1989). Controllers of Ziegler–Nichols type for unstable process with time delay. *International Journal of Control* 490 (4): 1273–1284.

Derafa, L., Madani, T., and Benallegue, A. (2006). *IEEE International Conference on Industrial Technology, 2006. ICIT 2006*, 1834–1839. IEEE. ISBN: 1424407265.In

Dittmar, R., Gill, S., Singh, H., and Darby, M. (2012). Robust optimization-based multi-loop PID controller tuning: A new tool and its industrial application. *Control Engineering Practice* 200 (4): 355–370.

Fahlstrom, P. and Gleason, T. (2012). *Introduction to UAV Systems*. Wiley.

Fertik, H.A. (1975). Tuning controllers for noisy processes. *ISA Transactions* 14: 292–304.

Franklin, G.F., Powell, J.D., and Emami-Naeini, A. (1991). *Feedback Control of Dynamic Systems*, 2e. Addison-Welsley Publishing Company.

Gene, F., Franklin, J.D.P., and Workman, M.L. (1998). *Digital Control of Dynamic Systems*, vol. 3. Menlo Park, CA: Addison-Wesley.

Franks, R.G. and Worley, C.W. (1956). Quantitative analysis of cascade control. *Industrial and Engineering Chemistry* 480 (6): 1074–1079.

Galeani, S., Tarbouriech, S., Turner, M., and Zaccarian, L. (2009). A tutorial on modern anti-windup design. In: *2009 European Control Conference (ECC)*, 306–323. IEEE.

Garpinger, O., Hägglund, T., and Åström, K.J. (2014). Performance and robustness trade-offs in PID control. *Journal of Process Control* 240 (5): 568–577.

Golnaraghi, F. and Kuo, B.C. (2010). *Automatic Control Systems*. Wiley.

Goodarzi, F., Lee, D., and Lee, T. (2013). Geometric nonlinear PID control of a quadrotor UAV on SE (3). In: *2013 European Control Conference (ECC)*, 3845–3850. IEEE.

Goodwin, G.C. and Sin, K.S. (1984). *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall.

Goodwin, G.C., Graebe, S., and Salgado, M. (2000). *Control System Design*. Englewood Cliffs, NJ: Prentice-Hall.

Gorez, R. (2003). New design relations for 2-DOF PID-like control systems. *Automatica* 390 (5): 901–908.

Gundlach, J. (2012). *Designing Unmanned Aircraft Systems: A Comprehensive Approach*. American Institute of Aeronautics and Astronautics.

Hagglund, T. and Astrom, K.J. (1985). Method and an apparatus in tuning a PID-regulator. US Patent 4, 549, 123.

Hägglund, T. (2012). Signal filtering in PID control. *IFAC Proceedings Volumes* 450 (3): 1–10.

Hägglund, T. (2013). A unified discussion on signal filtering in PID control. *Control Engineering Practice* 210 (8): 994–1006.

Han, J. (2009). From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics* 560 (3): 900–906.

Hang, C.C., Astrom, K.J., and Ho, W.K. (1991). Refinements of the Ziegler–Nichols tuning formula. *IEE Proceedings - Control Theory and Applications* 138: 111–118.

Henson, M.A. and Seborg, D.E. (1994). Adaptive nonlinear control of a pH neutralization process. *IEEE Transactions on Control Systems Technology* 20 (3): 169–182.

Henson, M.A. and Seborg, D.E. (1997). *Nonlinear Process Control*. Upper Saddle River, NJ: Prentice Hall PTR.

Ho, W.K. and W, X. (1998). Tuning for unstable processes based on gain and phase-margin specifications. *IEE Proceedings - Control Theory and Applications* 1450 (5): 392–396.

Ho, W.K., Hang, C.C., and Cao, L.S. (1995). Tuning of PID controllers based on gain and phase margin specifications. *Automatica* 31: 497–502.

Ho, W.K., Lee, T.H., and W, X. (2000). *IEEE Transactions on Industrial Electronics* 470 (1): 175–185.

Ho, W.K., Gan, O.P., Tay, E.B., and Ang, E.L. (1996). Performance and gain and phase margins of well-known PID tuning formulas. *IEEE Transactions on Control Systems Technology* 40 (4): 473–477.

Ho, W.K., Lim, K.W., and Xu, W. (1998). Optimal gain and phase margin tuning for PID controllers. *Automatica* 340 (8): 1009–1014.

Isaksson, A.J. and Graebe, S.F. (2002). Derivative filter is an integral part of PID design. *IEE Proceedings - Control Theory and Applications* 1490 (1): 41–45.

Isidori, A. (2013). *Nonlinear Control Systems*. Springer Science & Business Media.

Jayawardhana, B., Logemann, H., and Ryan, E.P. (2008). Control of second-order systems with hysteresis. *International Journal of Control* 810 (8): 1331–1342.

Jeng, J.-C. (2014). Simultaneous closed-loop tuning of cascade controllers based directly on set-point step-response data. *Journal of Process Control* 240 (5): 652–662.

Jeng, J.-C. and Lee, M.-W. (2012). Simultaneous automatic tuning of cascade control systems from closed-loop step response data. *Journal of Process Control* 220 (6): 1020–1033.

Jia, Q.W. (2009). Disturbance rejection through disturbance observer with adaptive frequency estimation. *IEEE Transactions on Magnetics* 450 (6): 2675–2678.

Johnson, M.A. and Moradi, M.H. (2005). *PID Control: New Identification and Design Methods*. New York: Springer-Verlag.

Kalafatis, A., Arifin, N., Wang, L., and Cluett, W.R. (1995). A new approach to the identification of pH processes based on the wiener model. *Chemical Engineering Science* 500 (23): 3693–3701.

Kalafatis, A.D., Wang, L., and Cluett, W.R. (1997). Identification of Wiener-type nonlinear systems in a noisy environment. *International Journal of Control* 660 (6): 923–941.

Kalafatis, A.D., Wang, L., and Cluett, W.R. (2005). Linearizing feedforward–feedback control of pH processes based on the wiener model. *Journal of Process Control* 150 (1): 103–112.

Khalil, H.K. (2002). *Noninear Systems*. Upper Saddle River, NJ: Prentice-Hall.

Khare, Y.B. and Singh, Y. (2010). Control of heat exchanger system. *International Journal of Computer Applications* 80 (6): 24–28.

Knospe, C. (2006). PID Control. *IEEE Control Systems* 260 (1): 30–31.

Ko, C.C., Chen, B.M., Chen, J. et al. (2001). Development of a web-based laboratory for control experiments on a coupled tank apparatus. *IEEE Transactions on Education* 440 (1): 76–86.

Komada, S., Ishida, M., Ohnishi, K., and Hori, T. (1991). Disturbance observer-based motion control of direct drive motors. *IEEE Transactions on Energy Conversion* 60 (3): 553–559.

Kothare, M.V., Campo, P.J., Morari, M., and Nett, C.N. (1994). A unified framework for the study of anti-windup designs. *Automatica* 300 (12): 1869–1883.

Kreyszig, E. (2006). *Advanced Engineering Mathematics*. New York: Wiley.

Krishnaswamy, P.R., Rangaiah, G.P., Jha, R.K., and Deshpande, P.B. (1990). When to use cascade control. *Industrial and Engineering Chemistry Research* 290 (10): 2163–2166.

Langer, J. and Landau, I.D. (1999). Combined pole placement/sensitivity function shaping method using convex optimization criteria. *Automatica* 350 (6): 1111–1120.

Larsson, P.-O. and Hägglund, T. (2011). Control signal constraints and filter order selection for PI and PID controllers. In: *Proceedings of the 2011 American Control Conference (ACC)*, 4994–4999. IEEE.

Lee, J., Cho, W., and Edgar, T.F. (1990). An improved technique for PID controller tuning from closed-loop tests. *AICHE Journal* 360 (12): 1891–1895.

Lee, J. (2013). Trajectory tracking control of ball and plate system. Undergraduate thesis. RMIT University.

Lee, Y., Park, S., and Lee, M. (1998). Controller tuning to obtain desired closed-loop responses for cascade control systems. *Industrial and Engineering Chemistry Research* 370 (5): 1859–1865.

Yongho Lee, S.O. and Park, S. (2002). Enhanced control with a general cascade control structure. *Industrial and Engineering Chemistry Research* 410 (11): 2679–2688.

Lees, M. and Wang, L. (2015). Controller design for industrial beer filtration. In: *2015 5th Australian Control Conference (AUCC)*, 306–311. IEEE.

Li, S., Yang, J., Chen, W.H., and Chen, X. (2012). Generalized state observer based control for systems with mismatched uncertainties. *IEEE Transactions on Industrial Electronics* 59: 4792–4802.

Li, S., Yang, J., Chen, W.H., and Chen, X. (2014). *Disturbance Observer-Based Control: Methods and Applications*. CRC Press.

Li, Y. and Song, S. (2012). *2012 IEEE 5th International Conference on Advanced Computational Intelligence, ICACI 2012*, 365–369.

Li, Y., Ang, K.H., and Chong, G.C.Y. (2006). Control system analysis and design. *IEEE Control Systems* 260 (1): 32–41.

Ligthart, J.A.J., Poksawat, P., Wang, L., and Nijmeijer, H. (2017). Experimentally validated model predictive controller for a hexacopter. *IFAC-PapersOnLine* 500 (1): 4076–4081.

Ljung, L. (1999). *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall.

Luyben, W.L. (1996). Tuning proportional- integral- derivative controllers for integrator/deadtime processes. *Industrial and Engineering Chemistry Research* 350 (10): 3480–3483.

Luyben, W.L. (2001). Effect of derivative algorithm and tuning selection on the PID control of dead-time processes. *Industrial and Engineering Chemistry Research* 400 (16): 3605–3611.

Madhuranthakam, C.M.R. and Penlidis, A. (2016). Surrogate models for online monitoring and process troubleshooting of NBR emulsion copolymerization. *Processes* 40 (1): 6.

Marlin, T.E. (1995). *Process Control: Designing Processes and Control Systems for Dynamic Performance*, vol. 2. New York: McGraw-Hill.

McNabb, L., Wang, L., and McGrath, B. (2017). Intrinsically stable realization of a resonant current regulator for a single phase inverter. In: *2017 11th Asian Control Conference (ASCC)*, 2256–2261. IEEE.

Middleton, R.H. and Goodwin, G.C. (1990). *Digital Control and Estimation a Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall.

Miller, R.K., Mousa, M.S., and Michel, A.N. (1988). Quantization and overflow effects in digital implementations of linear dynamic controllers. *IEEE Transactions on Automatic Control* 330 (7): 698–704.

Mishra, J.P., Wang, L., Zhu, Y. et al. (2018). A novel mixed cascade finite-time switching control design for induction motor. *IEEE Transactions on Industrial Electronics* 66 (2): 1172–1181.

Mita, T., Hirata, M., Murata, K., and Zhang, H. (1998). H_∞ control versus disturbance-observer-based control. *IEEE Transactions on Industrial Electronics* 450 (3): 488–495.

Moradi, H. and Salarieh, H. (2012). Analysis of nonlinear oscillations in spur gear pairs with approximated modelling of backlash nonlinearity. *Mechanism and Machine Theory* 51: 14–31.

Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall.

Nijmeijer, H. and Van der Schaft, A. (1990). *Nonlinear Dynamical Control Systems*, vol. 175. Springer.

Nurges, Ü. (2006). Robust pole assignment via reflection coefficients of polynomials. *Automatica* 420 (7): 1223–1230.

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.

Ogata, K. (2002). *Modern Control Engineering*. Prentice-Hall.

Ogunnaike, B.A. and Ray, W.H. (1994). *Process Dynamics, Modeling, and Control*. New York: Oxford University Press.

Espinosa Oviedo, J.J., Boelen, T., and Van Overschee, P. (2006). Robust advanced PID control (RaPID): PID tuning based on engineering specifications. *IEEE Control Systems* 260 (1): 15–19.

Padula, F. and Visioli, A. (2012). Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes. *IET Control Theory and Applications* 60 (6): 776–786.

Padula, F. and Visioli, A. (2011). Tuning rules for optimal PID and fractional-order PID controllers. *Journal of Process Control* 210 (1): 69–81.

Panagopoulos, H., Astrom, K.J., and Hagglund, T. (2002). Design of PID controllers based on constrained optimisation. *IEE Proceedings - Control Theory and Applications* 1490 (1): 32–40.

Patikirikorala, T., Wang, L., Colman, A., and Han, J. (2012). Hammerstein–Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems. *Control Engineering Practice* 200 (1): 49–61.

Peng, Y., Vrancic, D., and Hanus, R. (1996). Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. *IEEE Control Systems* 160 (4): 48–57.

Poksawat, P. (2018). Control system for fixed-wing unmanned aerial vehicles: automatic tuning, gain scheduling and turbulence mitigation. PhD thesis. RMIT University.

Poksawat, P. and Wang, L. (2017). Automatic tuning of hexacopter attitude control systems with experimental validation. In: *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, 753–758. IEEE.

Poksawat, P., Wang, L., and Mohamed, A. (2016). Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles. *IET Control Theory and Applications* 100 (17): 2233–2242.

Poksawat, P., Wang, L., and Mohamed, A. (2017). Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning. *IEEE Transactions on Control Systems Technology* 26 (4): 1192–1203.

Quang, N.P. and Dittrich, J.-A. (2008). *Vector Control of Three-Phase AC Machines*, 1e. Springer.

Ralhan, S. and Badgwell, T.A. (2000). Robust disturbance rejection for FIR systems with bounded parameters. *IFAC Proceedings Volumes* (10): 330, 815–820.

Rivera, D.E., Morari, M., and S.S. (1986). Internal model control. 4. PID controller design. *Industrial and Engineering Chemistry Process Design and Development* 25: 252–265.

Romero, J.A., Sanchis, R., Balaguer, P. et al. (2011). *Journal of Process Control* 210 (6): 840–851.

Saeed, J., Mishra, J.P., Wang, L., and Xinghuo, Y. (2018). A second-order sliding mode voltage oriented control of three-phase active front end rectifier. In: *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, 175–180. IEEE.

Sariyildiz, E. and Ohnishi, K. (2015). Stability and robustness of disturbance-observer-based motion control systems. *IEEE Transactions on Industrial Electronics* 620 (1): 414–422.

Schei, T.S. et al. (1992). *Automatica* 280 (3): 587–591.

Schei, T.S. (1994). Automatic tuning of PID controllers based on transfer function estimation. *Automatica* 30: 1983–1989.

Schrijver, E. and van Dijk, J. (2002). Disturbance observers for rigid mechanical systems:equivalence, stability and design. *Journal of Dynamic Systems, Measurement, and Control* 124 (0): 539–548.

Seborg, D.E., Mellichamp, D.A., Edgar, T.F., and Doyle, F.J. III, (2010). *Process Dynamics and Control*. Wiley.

She, J.H., Xin, X., and Pan, Y. (2011). Equivalent-input-disturbance approach: analysis and application to disturbance rejection in dual-stage feed drive control system. *IEEE/ASME Transactions on Mechatronics* 16: 330–340.

Shi, Z. and Zuo, Z. (2015). Backstepping control for gear transmission servo systems with backlash nonlinearity. *IEEE Transactions on Automation Science and Engineering* 120 (2): 752–757.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 130 (4): 291–309.

Skogestad, S. (2006). Tuning for smooth PID control with acceptable disturbance rejection. *Industrial and Engineering Chemistry Research* 450 (23): 7817–7822.

Slaughter, J. (1964). Quantization errors in digital control systems. *IEEE Transactions on Automatic Control* 90 (1): 70–74.

Soderstrom, T. (2018). *Errors-in-Variables Methods in System Identification*. Springer.

Soderstrom, T. and Stoica, P. (1989). *System Identification*. Englewood Cliffs, NJ: Prentice-Hall.

Soderstrom, T., Wang, L., Pintlon, R., and Schoukens, J. (2013). Can errors-in-variables systems be identified from closed-loop experiments. *Automatica* 49: 681–684.

Soh, Y.C., Evans, R.J., Petersen, I.R., and Betz, R.E. (1987). Robust pole assignment. *Automatica* 230 (5): 601–610.

Sung, S.W., Lee, J., and Lee, I.-B. (2009). *Process Identification and PID Control*. Wiley.

Tan, K.K., Lee, T.H., and Jiang, X. (2000). Robust on-line relay automatic tuning of PID control systems. *ISA Transactions* 390 (2): 219–232.

Tan, K.K., Wang, Q.-G., and Hang, C.C. (2012). *Advances in PID Control*. Springer Science & Business Media.

Tao, G. and Kokotovic, P.V. (1993). Adaptive control of systems with backlash. *Automatica* 290 (2): 323–335.

Tarbouriech, S. and Turner, M. (2009). Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory and Applications* 30 (1): 1–19.

Tongue, B.H. (2002). *Principles of Vibration*, 2e. New York: Oxford University Press.

Tyreus, B.D. and Luyben, W.L. (1992). Tuning PI controllers for integrator/dead time processes. *Industrial and Engineering Chemistry Research* 31: 2625–2628.

Vilanova, R. (2008). Based robust PID design: tuning guidelines and automatic tuning. *Journal of Process Control* 180 (1): 61–70.

Visioli, A. (2003). Modified anti-windup scheme for PID controllers. *IEE Proceedings - Control Theory and Applications* 1500 (1): 49–54.

Visioli, A. (2006). *Practical PID Control*. Springer Science & Business Media.

Visioli, A. (2012). Research trends for PID controllers. *Acta Polytechnica* 520 (5): 144–150.

Visioli, A. and Piazzi, A. (2003). Improving set-point-following performance of industrial controllers with a fast dynamic inversion algorithm. *Industrial and Engineering Chemistry Research* 420 (7): 1357–1362.

Visioli, A. and Piazzi, A. (2006). An automatic tuning method for cascade control systems. In: *2006 IEEE International Conference on Control Applications on Computer Aided Control System Design, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2968–2973. IEEE.

Wang, F.-S., Juang, W.-S., and Chan, C.-T. (1995). Optimal tuning of PID controllers for single and cascade control loops. *Chemical Engineering Communications* 1320 (1): 15–34.

Wang, L. and Cluett, W.R. (1997). Frequency-sampling filters: an improved model structure for step-response identification. *Automatica* 33: 939–944.

Wang, L. and Cluett, W.R. (1997). Tuning PID controllers for integrating processes. *IEE Proceedings - Control Theory and Applications* 142: 385–392.

Wang, L. and Cluett, W.R. (2000). *From Plant Data to Process Control: Ideas for Process Identification and PID Design*. London: Taylor and Francis.

Wang, L., Barnes, T.J.D., and Cluett, W.R. (1995). New frequency-domain design method for PID controllers. *IEE Proceedings - Control Theory and Applications* 142: 265–271.

Wang, L., Chai, S., Yoo, D. et al. (2015). *PID and Predictive Control of Electrical Drives and Power Converters*. Wiley-IEEE Press.

Wang, L. (2017). Automatic tuning of PID controllers using frequency sampling filters. *IET Control Theory and Applications* 110 (7): 985–995.

Wang, L., Freeman, C., and Rogers, E. (2017). Experimental evaluation of automatic tuning of PID controllers for an electro-mechanical system. *IFAC-PapersOnLine* 500 (1): 3063–3068.

Wang, Q.-G., Lee, T.-H., Fung, H.-W. et al. (1999). Tuning for improved performance. *IEEE Transactions on Control Systems Technology* 70 (4): 457–465.

Wang, Q.-G., Zhang, Y., and Guo, X. (2001). *Journal of Process Control* 11: 519–530.

Wang, Q.-G., Ye, Z., Cai, W.-J., and Hang, C.-C. (2008). *PID Control for Multivariable Processes*, vol. 373. Springer Science & Business Media.

Wang, Q.-G., Zhang, Z., Astrom, K.J., and Chek, L.S. (2009). Guaranteed dominant pole placement with PID controllers. *Journal of Process Control* 190 (2): 349–352.

Wellstead, P.E., Prager, D., and Zanker, P. (1979). Pole assignment self-tuning regulator. In: *Proceedings of the Institution of Electrical Engineers*, vol. 126, 781–787. IET.

Yang, J., Li, S., and Chen, W.H. (2012). Nonlinear disturbance observer-based control for multi-input and multi-output nonlinear systems subject to mismatching conditions. *International Journal of Control* 85: 1071–1082.

Yang, J., Chen, W.-H., Li, S. et al. (2017). Disturbance/uncertainty estimation and attenuation techniques in PMSM drives: a survey. *IEEE Transactions on Industrial Electronics* 640 (4): 3273–3285.

Yeung, K. and Huang, J. (2003). Development of a remote-access laboratory: a DC motor control experiment. *Computers in Industry* 520 (3): 305–311.

Young, P.C. (2012). *Recursive Estimation and Time-Series Analysis: An Introduction.* Springer Science & Business Media.

Cheng-Ching, Y. (2006). *Autotuning of PID Controllers*. Springer Science & Business Media.

Yukitomo, M., Shigemasa, T., Baba, Y., and Kojima, F. (2004). A two degrees of freedom PID control system, its features and applications. In: *Control Conference, 2004. 5th Asian*, vol. 1, 456–459. IEEE.

Zaccarian, L. and Teel, A.R. (2011). *Modern Anti-windup Synthesis: Control Augmentation for Actuator Saturation*. Princeton University Press.

Zheng, C., Su, Y., Mercorelli, P. et al. (2018). *IET Control Theory and Applications*.

Zhou, J. and Shen, X.Z. (2007). Robust adaptive control of nonlinear uncertain plants with unknown dead-zone. *IET Control Theory and Applications* 10 (1): 25–32.

Zítek, P., Fišer, J., and Vyhlídal, T. (2013). Dimensional analysis approach to dominant three-pole placement in delayed PID control loops. *Journal of Process Control* 230 (8): 1063–1074.

Index

a
anti-windup
  amplitude constraint 126
  disturbance observer 144, 153
  example 147
  implementation 126
  position form 124
  rate of change constraint 129
  resonant controller 158, 166
  tutorial 130
  velocity form 126
anti-windup implementation 132
auto-tuner 283
  attitude control 314, 321
  cascade control 297
  evaluation 286
  example 286, 292, 294, 297
  integrating plant 291
  stable plant 283
  tutorial 284

b
backlash
  example 223
bandwidth 48
  closed-loop 48

c
cascade control 203
  auto-tuner 297
  closed-loop poles 209
  closed-loop polynomial 208
  design steps 203
  disturbance rejection 207
  example 204, 206, 211, 215, 220, 223
  input disturbance 209
  nonlinearity 214
  performance 208
  quantization 219
  UAV 311
case studies
  ball and plate 187
  beer filtration 245
  double tank 182
  fired heater 25, 55
  induction motor 184
  polymer reactor 62
  three springs and double mass 103
  UAV 305, 309
comparative studies 289, 319
complementary sensitivity 47
  bandwidth 48
  cascade control 210
  desired 240
  noise attenuation 54
  Padula and Visioli controller 51
  reference following 48
  robust stability 61
  sinusoidal reference 53
  specified 241, 250
control system configuration
  one-degree-of-freedom 43
  two degrees of freedom 44

d
derivative filter 4, 84
  discretization 119
  time constant 58
  time constant $\beta\tau_D$ 4, 10, 150
  time constant τ_f 153, 155

PID Control System Design and Automatic Tuning using MATLAB/Simulink, First Edition. Liuping Wang.

Companion website: www.wiley.com/go/Wang/PIDcontrolsystems

discretization
  derivative filter 119
  higher order 120
  PI controller 117
  PID controller 119
  PID position form 115
  velocity form 117
disturbance estimator 140
disturbance observer
  anti-windup 153
  example 145, 148, 155, 162
  implementation 142, 151, 153
  PI controller equivalence 143
  PID control 150
  PID control equivalence 151
  resonant control 157
  resonant controller equivalence 159
disturbance rejection 54
  ball and plate 192
  beer filtration 247
  cascade control 209, 212
  example 55
  PID control 91
  resonant control 98
  sensitivity 55
double integrator system 4, 189

e

equilibrium point 181

f

feedback control
  block diagram 44
feedback error 1
FFT estimation
  example 271, 272
FFTRelay.m 269
final value theorem 2, 7, 90, 98
Fourier series
  fundamental frequency 268
  periodic signal 268
Fourier transform 268
FR4PID.m 243, 286
frequency sampling filter 274, 317
FSF 274
  estimation 276
  example 278
  UAV case studies 317
FSFRelay.m 276
fundamental frequency
  continuous time 268
  discrete time 268
  UAV case studies 321

g

gain margin
  PI controller design 234
gain scheduled control 194
  identifiers 195

i

implementation
  ball and plate 191
  disturbance observer 197
  gain scheduled control 196
  quantization 247
inner-loop
  auto-tuner 297
integrating plus delay 249
  example 253
  UAV model 317, 321
integrating with delay
  model 291
integrator windup 122
  anti-windup 124

l

least squares 275
linearization 179
  ball and plate 188
  double tank 182
  example 182
  induction motor 184
  UAV 312, 313

m

MATLAB tutorial
  auto-tuner 284
  FFT 269
  FSF 276
  relay control 261
  relay with integrator 265
MATLAB tutorial 2.1 39, 144
MATLAB tutorial 3.1 79
MATLAB tutorial 3.2 94

MATLAB Tutorial 4.1 130
MATLAB tutorial 5.2 153
MATLAB tutorial 5.3 160
MATLAB tutorial 8.1 243
MATLAB tutorial 8.2 244
MATLAB tutorial 9.1 261
MATLAB tutorial 9.2 265
MATLAB tutorial 9.3 269
MATLAB tutorial 9.4 276
MATLAB tutorial 9.5 284
measurement noise 267
modelling error 59
  additive 60
  multiplicative 60
Monte-Carlo simulation 270, 278
  comparative studies 279
  constant disturbance 279
  low frequency disturbance 280
  steady state 282

n

noise attenuation 54
  example 55
nonlinearity
  backlash 223
  deadzone 214
  inverse 214
  oscillation 223
  quantization 218
  UAV model 306
Nyquist diagram 287, 298
Nyquist stability criterion 36
  delay margin 38
  example 40
  gain margin 38
  phase margin 38
  PID control 37
NyquistPlot.m 39

o

outer-loop
  auto-tuner 298
  overshoot 131

p

Padula and Visioli tuning rule 289
PD controller
  parameter solutions 78
PD.m 79
phase margin
  PI controller design 234
PI controller 5
  K_c, τ_I 5
  block diagram 5, 45
  disturbance observer 143
  example 234
  IP configuration 7, 45
  overshoot 8
  parameter solutions 73
  steady-state error 7
PID control
  ball and plate 189
  feedforward 102
  fired heater 25
  UAV 305
PID controller 9, 82
  K_c, τ_I, τ_D 10
  block diagram 10
  empirical rules 252
  gain margin 253
  IPD configuration 10
  overshoot 10
  parameter solutions 82, 87, 92, 239, 252
  phase margin 253
  pole–zero cancellation 82
  qantization error 135
  start-up 134
PID controller design
  frequency domain 239
  two frequency points 242, 243
PID tuning rules 13
  Cohen and Coon 17
  first order plus delay 15, 18, 19
  IMC-PID 18, 62
  Padula and Visioli 20, 50
  rework 40
  Skogestad 19, 32
  Tyreus and Luyben 14
  UAV case studies 317, 322
  Wang and Cluett 20
  Ziegler-Nichols oscillation 13
  Ziegler-Nichols step response 16
PIDEstim.slx 153

PIDplace.m 94
PIDV.slx 130
PIEstim.slx 144
pole-assignment 73
poles 3, 6, 32
  cascade control 209
  characteristic equation 6
  closed-loop disturbance observer 141
  determining 32
  example 32
proportional control 1, 140
  gain K_1 140
  gain K_c 1
proportional plus derivative control 3, 149
  gain K_1 and K_2 149
  gain K_c and τ_D 3
  proportional plus integral controller 5

q
quantization
  example 219
  simulation 221

r
recursive least squares 275
relay control 259
  example 262, 266
  tutorial 261
  UAV case studies 316, 321
  with hysteresis 259
  with integrator 263
RelayH.slx 261
RelayI.slx 265
ResEstim.slx 160
resonant controller 96
  anti-windup 158
  disturbance observer 157
  example 162
  implementation 158
  parameter solutions 96, 101
  pole–zero cancellation 99
robust stability 60
  condition 61
Routh-Hurwitz stability criterion 33
  example 35

s
sensitivity
  cascade control 210
  disturbance observer 166
  disturbance rejection 50, 54, 166
  sinusoidal disturbance 53
sensitivity functions 45
  complementary sensitivity 46
  control sensitivity 46
  input disturbance sensitivity 46
  sensitivity 46
  steady-state information 114
  step response test 15

t
Taylor series expansion 180
test4FR2PID.m 244
Tuner4StableSys.slx 284