Introduction to Modern Controls, with Illustrations in MATLAB and Python

Introduction to Modern Controls with Illustrations in MATLAB and Python

Xu Chen and Masayoshi Tomizuka

June 24, 2024

University of Washington University of California, Berkeley Introduction to Modern Controls, with Illustrations in MATLAB and Python

Copyright

Copyright ©Xu Chen and Masayoshi Tomizuka, 2023-

License for Code

The code/software in this book is contained under the © MIT License. To view a copy of the CC0 code, visit: http://creativecommons.org/publicdomain/zero/1.0/

Publisher

To cite this material: Xu Chen and Masayoshi Tomizuka, "Introduction to Modern Controls – with Illustrations in MATLAB and Python," 2023, ISBN: 9798860587496.

Preface

This book introduces the theory and practice of modern control systems. The emphasis is on using state-space methods to model, analyze, and control dynamic systems. Topics include state-space modeling and solutions, stability, controllability and observability, state-feedback control, observers, observer state feedback controls, least square estimation, Kalman filter, and Linear Quadratic Gaussian optimal control. These topics are discussed in both continuous- and discrete-time settings throughout the book.

The material in this book is based on many years of teaching experience at the University of Washington and the University of California, Berkeley. The main sources of the material are:

- ▶ ME 232 and ME 233 at the University of California, Berkeley, and
- ▶ ME 547 at the University of Washington, Seattle.

This book consists of four parts. Part I introduces the basics of dynamic systems modeling, such as Laplace and Z transforms, state-space descriptions and realization theory, and how to solve the state equation. Part II examines the properties of dynamic systems, such as classic and Lyapunov stability theories, controllability, observability, and the decomposition of an uncontrollable and/or unobservable system. After understanding these system properties, in Part III, we cover estimation and controls for state-space systems. Chapter 11 centers on the power of state feedback. Then in Chapter 12, we discuss state observers and observer-state feedback. As a powerful state-feedback control method, Chapter 13 covers the linear quadratic optimal control algorithm. Part IV is dedicated to estimation and control of stochastic systems, where the state-space system equations are subject to input and output stochastic noises. We review first relevant results in probability theory in Chapter 14, building on which we derive the least square estimation in Chapter 15 and then the discrete- and continuous-time Kalman filters in Chapter 16. Chapter 17 integrates linear quadratic optimal control with the Kalman filter, to provide the celebrated Linear Quadratic Gaussian (LQG) Optimal control. At the end of the book, we provide a review of related linear algebra for controls.

Over three hundred examples, figures, table summaries, and exercises distilled from physical systems supplement the learning. MATLAB and Python are the primary tools for our numerical demonstrations. When MATLAB examples appear, complementary Python codes will follow to provide results as equivalent as possible in the more nascent and open-source computation environment.

All the main codes are available for download on the book website https://mcimp-book.github.io/mcimp/. We have also provided accompanying slides and lecture recordings – accessible from the same book website.

All the MATLAB demonstrations were performed in MATLAB 2022b, and the Python demonstrations in Python 3.9.13, using toolboxes SymPy v1.11.1 and python-control v0.9.2. For simple calculations and graphical illustrations, we use gnuplot, a light-weight command-line driven graphing utility across different operation systems. The coding commands and results are all provided in an "in-line" fashion, directly embedded in the text materials. Appendix "How to Install and Run Python" provides a summary of ways to configure Python in different operation systems.

We are grateful to the many teaching assistants and students who helped typeset problems and proofread course contents over the years. In particular, we would like to thank Liting Sun for creating some of the drawings in LaTeX and Lingfeng Sun for typesetting the Berkeley ME 233 course reader. Jonas Beachy, Xiaohai Hu, and Marina Ruediger helped with a few case studies. Their contributions have greatly enhanced the quality of this book.

We hope this book will serve as a useful reference for students and researchers interested in the field of dynamics and control.

June 24, 2024 Seattle, Washington Berkeley, California

About the Authors

Xu Chen is an Associate Professor and holds the Bryan T. McMinn Endowed Research Professorship of Mechanical Engineering at the University of Washington (UW), Seattle. He obtained his Ph.D. degree in mechanical engineering from the University of California, Berkeley in 2013, and his bachelor's degree in mechanical engineering from Tsinghua University, China in 2008. He researches into dynamic systems, controls, and robotics, to better understand and engineer smart manufacturing (e.g., with feedback controls, lasers, machine vision, and nondestructive inspection). He also serves as Director of the Boeing Advanced Research Collaboration at the UW – an interdisciplinary Boeing-UW partnership for the future of flight. Xu Chen is an alumnus of the National Academy of Engineering's 2023 Frontiers of Engineering Symposium, a recipient of the National Science Foundation CAREER Award, the SME Sandra L. Bouckley Outstanding Young Manufacturing Engineer Award, the Mechatronic Systems Outstanding Young Researcher Award from the International Federation of Automatic Control (IFAC) Technical Committee on Mechatronic Systems, the Young Investigator Award from ISCIE / ASME International Symposium on Flexible Automation, and the inaugural UTC Institute for Advanced Systems Engineering Breakthrough Award.

Masayoshi Tomizuka received his Ph. D. degree in Mechanical Engineering from the Massachusetts Institute of Technology in February 1974. In 1974, he joined the faculty of the Department of Mechanical Engineering at the University of California at Berkeley, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair and serves as Associate Dean for the Faculty in the College of Engineering. His current research interests are optimal and adaptive control, digital control, motion control, and control problems related to robotics and manufacturing, vehicles and mechatronic systems. He served as Program Director of the Dynamic Systems and Control Program of the National Science Foundation (2002-2004). He has supervised about 130 Ph. D. students to completion. He served as President of the American Automatic Control Council (AACC) (1998-99). He is Honorary Member of the ASME, Life Fellow IEEE, and Fellow of IFAC and the Society of Manufacturing Engineers (SME). He is the recipient of the J-DSMC Best Paper Award (1995, 2010), the DSCD Outstanding Investigator Award (1996), the Charles Russ Richards Memorial Award (ASME, 1997), the Rufus Oldenburger Medal (ASME, 2002), the John R. Ragazzini Award (AACC, 2006), the Richard Bellman Control Heritage Award (AACC, 2018), the Honda Medal (ASME, 2019) and the Nichols Medal (IFAC, 2020). He is a member of the U.S. National Academy of Engineering.

Contents

Pı	reface		iii
A	bout	the Authors	v
C	onten	its	vii
1	Intr	oduction	1
	1.1	The Power of Controls	1
	1.2	Relevant Terminologies	1
	1.3	The Objectives and The Means of Controls	3
	1.4	Societies to Learn More about Controls	5
S	YSTE	M Description	7
2	Mo	deling	9
	2.1	Methods of Modeling	9
	2.2	Continuous-Time Systems	10
	2.3	Discrete-Time Systems	11
	2.4	Example: Atomic Force Microscopy	11
	2.5	Example: Hard Disk Drive and Information Storage	14
	2.6	Model Properties	19
	2.7	271 Example: Magnetically Grammed ad Ball	20
		2.7.1 Example: Magnetically Suspended Ball	20
		2.7.2 Example: Water Talk	20
		27.4 Example: Vehicle Steering	21
	2.8	"All Models are Wrong, but Some are Useful"	23
	2.9	Exercise	27
2	Lan	lace and 7 Transforms	20
5	21	The Lanlace Transform	29
	5.1	311 The Laplace Approach to ODEs	29
		3.1.2 Relevant Properties of the Laplace Transofrm	36
	3.2	Inverse Laplace Transform and Partial Fraction Expansion	39
	3.3	From Laplace Transform to Transfer Functions	41
	3.4	The Z Transform	45
		3.4.1 Definition	45
		3.4.2 Relevant Properties	47
		3.4.3 Applications to Dynamic Systems	52
	3.5	From Difference Equation to Discrete-Time Transfer Functions	53
	3.6	Recap	56
	3.7	Exercise	57

4	Stat	e-Space Description of a Dynamic System	61			
	4.1	The Concept of States	61			
	4.2	General State-Space Descriptions	62			
	4.3	From the State Space to Transfer Functions	63			
	4.4	Linearization and State-Space Representation of Nonlinear Systems	69			
		4.4.1 State-Space Representation of General Nonlinear Systems	69			
		4.4.2 Equilibrium Point and Linearization around an Equilibrium Point	70			
		4.4.2 Equilibrium Font and Encontration around an Equilibrium Font	70			
		4.4.4 Example: Water Tank	74			
		4.4.5 Example: Water faits	74			
	4 5	4.4.5 Example: vehicle Steering	20			
	4.5		84			
	4.6	Exercise	84			
5	Stat	e-space Realizations: The Canonical Forms	87			
	5.1	Controllable Canonical Form	87			
	5.2	Observable Canonical Form	90			
	53	Diagonal and Jordan Canonical Forms	94			
	5.0	Diagonal and Jordan Canonical Tornis	06			
	5.4	Cincilar Desligations	90			
	5.5		90			
	5.6	Recap	99			
	5.7	Exercise	101			
6	Solu	ution of Time-Invariant State-Space Equations	103			
	6.1	Continuous-Time State-Space Solutions	103			
		6.1.1 The Solution to $\dot{x} = ax + bu$	103			
		6.1.2 The Irrational number <i>e</i>	105			
		6.1.3 Fundamental Theorem of Differential Equations	107			
		614 The Solution to n^{th} -order LTI Systems	107			
	62	Discrete_Time_ITI State_Space Solutions	111			
	63	Explicit Computation of the State Transition Matrix e^{At}	112			
	0.5	6.2.1 The Case with Distinct Eigenvalues (Diagonalization)	11.0			
		6.3.1 The Case with Distinct Eigenvalues (Diagonalization)	114			
			110			
		6.3.3 The Case with Complex Eigenvalues	116			
		6.3.4 The Case with Repeated Eigenvalues, via Generalized Eigenvectors	117			
		6.3.5 Physical Interpretation	120			
	6.4	Explicit Computation of the State Transition Matrix A^*	123			
	6.5	Transition Matrix via Inverse Transformation	123			
	6.6	Solutions of Time-Varying State Equations	125			
		6.6.1 Continuous-Time Case	125			
		6.6.2 Discrete-Time Case	125			
	6.7	Recap	126			
	6.8	Exercise	127			
7	Discusts Time Madels of Continuous Contenas					
1		Crete-Time Would's Of Continuous Systems	129			
	/.I		129			
	1.2		130			
	7.3	Iranster-Function Models	132			
	7.4	Exercise	138			

System Properties

8	Stab	oility	141
	8.1	Definitions	141
		8.1.1 Review of Relevant Functional Analysis	141
		8.1.2 Lyapunov's Definition of Stability	142
	8.2	Stability of LTI Systems	144
		8.2.1 Method of Eigenvalue Locations	144
		8.2.2 Routh-Hurwitz Criterion for Continuous-Time LTI Systems	146
		8.2.3 Routh-Hurwitz Criterion for Discrete-Time LTI Systems	147
	8.3	Lyapunov's Approach to Stability	149
		8.3.1 Stability from an Energy Viewpoint	149
		8.3.2 Relevant Mathematical Tools	150
	8.4	Lyapunov Stability Theorems	159
		8.4.1 Lyapunov Stability for Continuous-Time Systems	159
		8.4.2 Lyapunov Stability for Discrete-Time Systems	166
	8.5	Recap	169
	8.6	Exercise	169
9	Con	trollability and Observability	171
	9.1	Basic Concepts	171
	9.2	The Case for Discrete-Time Systems	172
		9.2.1 Controllability	172
		9.2.2 Observability	177
	9.3	The Case for Continuous-Time Systems	181
		9.3.1 Controllability	181
		9.3.2 Observability	184
	9.4	Transforming Single-Input Controllable Systems Into the Controllable Canonical Form	189
	9.5	Transforming Single-Output Observable Systems Into the Observable Canonical Form	191
	9.6	Recap	194
	9.7	Exercise	194
10	Kalr	nan Decomposition	197
	10.1	Basic Concepts	197
	10.2	Kalman Decomposition of Uncontrollable Systems	198
	10.3	Kalman Decomposition of Unobservable Systems	205
	10.4	General Kalman Decomposition, Stabilizability, and Detectability	207
	10.5	Recap	209
	10.6	Exercise	210
Es	TIM	ation and Control	211
11	State	e Feedback	212
11	11 1	State Feedback and Figenvalue Assignments	213
	11.1	11.1.1 Dynamic Properties of the Closed Loop System	213 214
		min Dynamic roperties of the Closed-Loop System	414

	11.1.2	Eigenvalue Assignment for Single-Input Systems	215
	11.1.3	Multi-Input Systems	218
11.2	Nume	rical Tools	218

	11.3	Output Feedback	219
	11.4	Recap	220
	11.5	Exercise	220
12	Obs	ervers and Observer-State Feedback	223
	12.1	Open-Loop Observer	223
	12.2	Continuous-Time Luenberger Observer	224
	12.3	Observer State Feedback Control	233
	12.4	Discrete-Time Observers	234
		12.4.1 Discrete-Time Full State Observer	234
		12.4.1 Discrete-Time Full State Observer with Predictor	234
	12 5	Reduced-Order Observer	234
	12.5		237
	12.0	Recap	242
	12.7		242
13	Line	ear Quadratic Optimal Control	245
	13.1	Problem Formulation	245
	13.2	Solution of the Continuous-Time LQ Problem	246
	13.3	Stationary Continuous-Time LQ Problem	254
	13.4	Application and Practice	259
		13.4.1 Choosing <i>Q</i> and <i>R</i>	259
		13.4.2 MATLAB and Python Commands	260
		13.4.3 Obtaining P_+	260
		13.4.4 Example: Inverted Pendulum on a Cart	262
		13.4.5 Robustness of Stationary LO Regulators	275
	13.5	Further Development of the Continuous-Time LO Regulator	275
	10.0	13.5.1 Return Difference Equality	275
		13.5.2 Stability Margins of Scalar I OR	276
		13.5.3 The Closed-Loop Figenvalues	277
		13.5.4 Symmetric Root Locus	278
		12.5.5 Stoady State Property	225
	12.6	Discrete Time I O Optimel Control	200
	13.0	12.6.1 Introduction of Dynamic Programming	207
		12.6.2 Demanuia Dragonania for Consul Ontined Control Duckland	20/
		13.6.2 Dynamic Programming for General Optimal Control Problems	200
		13.6.3 Solving Discrete-Time Finite-Horizon LQ Problems via Dynamic Programming	290
	10 5	13.6.4 Discrete-Time Stationary LQ Optimal Control	293
	13.7	Further Development of the Discrete-Time LQ Regulator	297
		13.7.1 Obtaining P_+	297
		13.7.2 Return Difference Equality	297
		13.7.3 The Closed-Loop Eigenvalues	299
	13.8	Recap	301
	13.9	Exercise	305
S	госн	astic Estimation and Control	309
51	. e en		207
14	Rev	iew of Probability Theory	311
	14.1	Sample Space, Events, and Probability Axioms	311

	14.2	Random Variables, Probability Density, and Moments of Distributions	312
	14.3		313
	14.4	Random Vector, Joint Probability and Distribution, Conditional Probability	316
	14.5		320
		14.5.1 Filtering a Random Process	322
		14.5.2 Filtering a Random Process in the State Space	323
	14.6	Continuous-Time Random Process	325
	14.7	Recap	328
	14.8	Exercise	331
15	Leas	st Square Estimation	333
	15.1	General Solution	333
	15.2	Solution in the Gaussian Case	334
	15.3	Properties of Least Square Estimate (Gaussian Case)	336
	15.4	Example Application of Least Square Estimation	339
	15.5	Recap	344
	15.6	Exercise	345
16	Stoc	hastic State Estimation and Kalman Filter	347
	16.1	Review of State Observers	347
	16.2	Discrete-Time Stochastic State Estimation	347
		16.2.1 Problem Definition	347
		16.2.2 Solution	349
		16.2.3 Discrete-Time Kalman Filter	351
		16.2.4 Steady-State Kalman Filter	352
		16.2.5 Return Difference Equality for Equation 16.29	361
	16.3	Continuous-Time Stochastic State Estimation	364
	10.0	16.3.1 General Continuous-Time Kalman Filter	365
		16.3.2 Steady-State Kalman Filter	367
		16.3.3 Return Difference Equality	367
	16.4	Kalman Filter Application: Kinematic Kalman Filter	369
	16.5	The Kalman Filter Equations using Other Notation Systems	371
	16.6	Recan	373
	16.7	Exercise	375
17	Line	par Quadratic Gaussian (LOG) Optimal Control	377
17	17 1	Stochastic Control with Evactly Known State	377
	17.1	17.1.1. Stochastic Control with Inovactly Known State	221
		17.1.1 Stochastic Control with mexacity Known State	201
	17.0	Continuous Time LOC Broblem	205
	17.2		303
	17.3		380
	17.4	Exercise	388
18	Furt	her Readings	391
Aŗ	penc	lix: Review of Relevant Linear Algebra	393
1	1	Basic Concepts of Matrices and Vectors	393
		1.1 Matrix Addition and Multiplication	394
		1.2 Matrix Transposition	395

2	Linear Systems of Equations	395
3	Vector Space, Linear Independence, Basis, and Span	398
4	Matrix Properties	398
	4.1 Rank	398
	4.2 Range and Null Spaces	399
	4.3 Determinants	399
	4.4 Matrix and Linear Equations	400
5	Eigenvalues and Eigenvectors	400
	5.1 Matrix, Mappings, and Eigenvectors	400
	5.2 Computation of Eigenvalue and Eigenvectors	401
	5.3 Eigenbase and Diagonalization	403
	5.4 Similarity Transformation	405
6	Matrix Inversion	406
	6.1 Block Matrix Decomposition and Inversion	408
	6.2 LU and Cholesky Decomposition	409
	6.3 Determinant and Matrix Inverse Identity	410
7	Spectral Mapping Theorem	412
8	Matrix Exponential	413
9	Inner Product	414
	9.1 Inner Product Spaces	414
	9.2 Trace (Standard Matrix Inner Product)	415
10	Vector Norms	415
11	Symmetric and Orthogonal Matrices	416
12	Positive-Definite Matrices	417
	12.1 Definitions and Basic Properties	417
	12.2 General Positive-Definite Matrices	418
	12.3 Positive-Definite Functions	419
13	Singular Value Decomposition (SVD)	419
	13.1 Motivation	419
	13.2 Main Result	420
	13.3 Properties of Singular Values	422
14	Induced Matrix Norm	422
Append	dix: How to Install and Run Python	425

Alphabetical Index

List of Figures

1.1	A general block diagram.	2
1.2	An open-loop control system.	2
1.3	A closed-loop heating control example.	3
1.4	Control system with feedforward control.	3
1.5	Overview of book topics from system description to estimation and controls.	4
1.6	Structure of digital control systems.	4
1.7	Design approaches of digital control systems.	5
2.1	Maior components in a hard disk drive	9
2.2	Example frequency response of the voice coil motor in a hard disk drive.	10
2.3	Isaac Newton.	11
2.4	Schematic of the AFM.	12
2.5	Mechanical models of an AFM.	13
2.6	Control-related components in a hard disk drive.	14
2.7	Baseline model of the VCM component in an HDD.	15
2.8	Schematic structure of dual-stage HDDs.	17
2.9	Baseline model of the PZT component in an HDD.	17
2.10	Free-body diagram of a magnetically suspended ball.	20
2.11	Water flow in a tank.	20
2.12	Free-body diagram of a pendulum system	21
2.13	Kinematic modelling of the four-wheel vehicle steering system.	21
2.14	The bicycle model of the vehicle steering system.	22
2.15	Example frequency responses of the voice coil motor stage in a batch of hard disk drives	24
3.1	The Laplace approach to ODEs.	30
3.2	Pierre-Simon Laplace	30
3.3	Illustration of piecewise continuous functions.	31
3.4	Illustration of exponential order	31
3.5	Illustration of the Dirac impulse function.	33
3.6	Approximation of the unit-step function.	34
3.7	Approximation of the Dirac-delta function.	35
3.8	Toward smoother approximations of the unit-step function.	36
5.1	Transforming between transfer functions and state-space realizations	87
5.2	Block diagram of the controllable canonical form for third-order systems.	89
5.3	Block diagram of the observable canonical form for third-order systems	91
5.4	Block diagram of the diagonal realization for third-order systems with distinct poles.	94
5.5	Block diagram of the Jordan form realization for third-order systems with repeated poles	95
5.6	Block diagram of the modified canonical form for third-order systems with complex poles. $k_2 =$	
	$(\beta + \alpha \sigma)/\omega$ and $k_3 = \alpha$	96
6.1	The concept of the time constant.	105
6.2	The unit-step response of a first-order system	105
6.3	Convergence of the infinite series $\sum_{n=0}^{\infty} \frac{1}{n!}$ to e	106

6.4	Example state trajectory for a second-order system under different initial conditions	116
6.5	Map of eigenvalue and associated response mode (continuous-time case)	121
6.6	Map of eigenvalue and associated response mode (discrete-time case)	122
7.1	Illustration of a sampler.	129
7.2	Illustration of a ZOH	130
7.3	Illustration of a fast ZOH.	130
7.4	A continuous-time state-space system preceded by zero order hold.	130
7.5	Continuous-time transfer function preceded by a zero order hold.	133
7.6	Discretization of the voice coil motor models in HDDs under different sampling time	136
8.1	Example continuous functions.	142
8.2	A uniformly continuous function.	142
8.3	A continuous but not uniformly continuous function.	142
8.4	Stability concepts in the sense of Lyapunov.	143
8.5	Time-domain plot of function $te^{\lambda t}$.	145
8.6	The bilinear transform maps the closed left half plane to the closed unit disk.	148
8.7	A positive-definite function $W(x_1, x_2) = x^2 + x^2$.	158
8.8	A locally positive-definite function $W(x_1, x_2) = x_1^2 + \sin^2 x_2$.	159
9.1	Arthur Cayley	173
9.2	William Hamilton.	174
11.1	Illustration of state-feedback control.	213
12.1	Concept of a closed-loop observer.	224
12.2	Realization of the Luenberger observer.	224
12.3	Block diagram of observer state feedback control.	233
12.4	Block diagram of the continuous-time reduced-order observer.	239
12.5	Block diagram of reduced-order observer: implementation form.	240
13.1	LQ example with small penalty on control. $P^*(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 0.0001 \dots \dots \dots \dots \dots$	251
13.2	LQ example with medium penalty on control. $P^*(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 1 \dots \dots \dots \dots \dots \dots$	252
13.3	LQ example with large penalty on control. $P^*(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 100$	253
13.4	LQ example with large penalty on control. $P^*(0) = \begin{bmatrix} 20 & 0 \\ 0 & 2 \end{bmatrix}$, $R = 100 \dots \dots \dots \dots \dots \dots \dots$	253
13.5	State trajectories of an inverted pendulum on a cart under full-state LQ optimal control	266
13.6	State trajectories of a linearized inverted pendulum on a cart under full-state LQ optimal control	268
13.7	State trajectories of a linearized inverted pendulum on a cart under observer-state LQ control.	271
13.8	State estimation errors of a linearized inverted pendulum on a cart under observer-state LQ control.	272
13.9	State estimation errors of a linearized inverted pendulum on a cart under observer-state LO control.	274
13.10	Block diagram of the optimal LQ system.	275
13.11	Frequency response of the continuous-time LO loop transfer function.	277
13.12	Frequency response of the discrete-time LQ loop transfer function.	299
14.1	Probability density function of a uniform distribution.	313

14.2 14.3 14.4 14.5	Probability density function of a Gaussian distribution	314 315 316 320
15.1	Geometric interpretation of the least square estimator $E[x y]$	336
15.2	Geometric interpretation of the least square estimator $E[x y, z]$ where y and z are uncorrelated	337
15.3	Geometric interpretation of the least square estimator $E[x y, z]$ where y and z are correlated	339
16.1	Rudolf Kalman.	347
16.2	Block-diagram representation of the corrector and predictor form of the Kalman filter	351
16.3	State estimation by Kalman filter and finite settling time observer (FSTO) with $r = 2$	355
16.4	Performance of the Kalman filter and finite settling time observer (FSTO).	356
16.5	A closed-loop return-difference representation of the Kalman filter	362
16.6	Block diagram of the return difference in a Kalman filter ($u(t) = 0$ or the forced response has been	
	subtracted)	368
16.7	Example eigenvalues of a second-order Kalman filter	369
17.1	Structure of the LQG control scheme.	384
1	Example relationship between x and Ax	401
2	Decomposition of <i>x</i>	401
3	Construction of Ax	401
4	Example points in vector space 1.	405
5	The same points in vector space 2	405

List of Tables

3.1	Common Laplace transforms.	39
3.2	Common properties of the Z transform.	50
3.3	Table of Laplace and Z transforms [5]. $x(t) = 0$ for $t < 0$. $x(kT) = x(k) = 0$ for $k < 0$. Unless otherwise	
	noted, $k = 0, 1, 2, 3, \dots$	51
3.4	Comparison of continuous- and discrete-time transfer functions.	54
6.1	Common pairs of J and J^k .	123
6.2	Summary of solutions to state equations.	123
6.3	Common state-transition matrices.	127
7.1	Zero order hold equivalents of continuous-time transfer functions	135
8.1	Stability of the origin for $\dot{x} = Ax$.	145
8.2	Stability of the origin for $x (k + 1) = Ax (k)$.	146
8.3	Analogy between symmetric, skew-symmetric, and orthogonal matrices and the complex plane	154
8.4	Analogy between symmetric, skew-symmetric, orthogonal, and positive-definite matrices and the	
	complex plane.	157
8.5	Summary of Lyapunov equations in continuous- and discrete-time dynamics.	169

9.1 Controllability and observability gramians and their relation to Lyapunov Equations	194
13.1 Summary of continuous- and discrete-time LQ optimal-control properties	304
16.1 Comparison of the estimation error covariances (diagonal entries) between a Kalman filter and a finite settling time observer.	355

Introduction | .

1.1 The Power of Controls

Our internal body temperature is regulated around the normal value of about 37° C or 98.6° F. A part of our brain called the hypothalamus checks our current temperature and compares it with the normal value. In a sauna room where the temperature is too high, sweat is produced to cool the skin,¹ and the blood vessels under our skin get wider to increase the blood flow to the skin.² On the other hand, when building a snowman outside, the blood vessels under our skin become narrower to decrease blood flow to the skin, retaining heat near the warm inner body; muscles, organs, and brain produce heat (e.g., muscles can produce heat by shivering); and our thyroid gland releases hormones to increase our metabolism.^{3 4} That is the power of *feedback controls*: it allows us to make a precision device out of a crude one that works well even in changing environments.

We also use *prediction and feedforward controls* in our regulation of body temperature: as kids, we had learned to wear T-shirts in summer, long sleeves and coats in winter. With such predictive and feedforward controls, the burden of feedback control is greatly lifted.

Using these temperature-control activities, fine-tuned naturally as we grow, our body can respond to internal and external stimuli and make adjustments to keep the body within one or two degrees of the normal temperature, whether in summer or winter, at the north pole or in the Sahara Desert!

1.2 Relevant Terminologies

More formally, **feedback** is the use of information of the past or the present to influence behaviors of a system. A **system** is an interconnection of elements and devices for a desired purpose. In your undergraduate control course, you have obtained basic understandings of a control system with concepts such as transfer functions, proportional integral derivative (PID) controllers, and frequency responses in classic control. Below, we provide a brief review of the key concepts and terminologies.

We use a **block diagram** such as the one in Figure 1.1 to visualize the system structure and the interconnection of system components. Here, the **plant** consists of: (i) a process whose output is to be controlled and (ii) an actuator – a physical device capable to influence the controlled variable of the process. The sensor measures the output of the plant and feeds it back to be compared to a reference signal. The error after the comparison then drives the controller to generate the command for the actuator. It is not uncommon to have an input filter that shapes the reference signal: for example, in controlling the body temperature, we wear clothes of

1.1 The Power of Controls	1
1.2 Relevant Terminologies	1
1.3 The Objectives and The	
Means of Controls	3
1.4 Societies to Learn More	
about Controls	5

1: The middle layer of the skin, or dermis, stores most of the body's water. When the temperature is too high, that water, along with the body's salt, are brought to the surface of the skin as sweat. On the skin surface, the water evaporates and cools the body.

2: This process is called vasodilatation.

3: The three mechanisms are called vasoconstriction, thermogenesis, and hormonal thermogenesis, respectively.

4: We are continuously losing heat: The Basal Metabolic Rate (BMR) is the number of calories we burn as our body performs basic (basal) life-sustaining function. An average man has a BMR of around 7,100 kJ per day, while an average woman has a BMR of around 5,900 kJ per day. See more at https: //www.betterhealth.vic.gov.au/health/ conditionsandtreatments/metabolism.



Figure 1.1: A general block diagram.

different thickness and warmth that build a buffer between our skin and the environment.

In our introductory example, the body cannot immediately adjust the temperature. Instead, it is an intricate **dynamic process** – blood vessels expand and contract to move blood and heat closer to or further away from the skin, thus releasing or conserving warmth. Dynamic systems do not show the full effect of the input immediately but after some **transient** and/or **delays**.

The inputs and outputs in a dynamic system are *signals*, i.e., they are functions of time, e.g., speed of a car, temperature in a room, voltage applied to a motor, price of a stock, and electrical-cardiograph. In practice, there will be disturbances to the system and noises in the sensor measurement signals. These are also signals and will negatively influence the performance of the control system.

An **open-loop** control system (cf. Figure 1.2) is one where the output of the plant does not influence the input to the controller.



Figure 1.2: An open-loop control system.

A **closed-loop** system is one where multiple components (plant, controller, etc) have a closed interconnection. For exampel, in the heating control system for a house, the thermostat will measure the room temperature, compare it to the set value, and turn on or off the furnace to keep the house warm in a closed loop. There is always feedback in a closed-loop system.

Closed-loop (feedback) controllers provide a more robust performance than open-loop controllers in the presence of disturbances and plant uncertainties. If the reference signal varies fast and its variation is known in advance, then **feedforward** control based on information about the reference is useful. This is often the case in machine tool control and robot control. Feedforward control is also effective if the disturbance signal can

If the mechanism of transient and delay is not correctly taken into account, the feedback controller may over compensate or pump in an excessive amount of control energy, resulting in system instability.



Figure 1.3: A closed-loop heating control example.

be measured or estimated. Figure 1.4 shows a control system that integrates these benefits of feedback and feedforward controls.



Figure 1.4: Control system with feedforward control.

1.3 The Objectives and The Means of Controls

A few aspects of control objectives are universal. For example, we would always want our control system to result in closed-loop dynamics that are stable and insensitive to disturbances. These form the **stabilization** problem and the **disturbance rejection** problem, respectively. When the reference is a fixed point such as the normal temperature of our body, the control objective is the **regulation** of output in the presence of disturbances and noises. When the reference changes and is time-varying, a **tracking** problem is formed.

To achieve the control objectives, the control engineer must *model* the controlled plant, *analyze* the characteristics of the plant, *design* control algorithms (controllers), *analyze* performance and robustness of the control system, and *implement* the controller.

To *model*, *analyze*, and *design* control systems, we must be able to describe systems formally, understand their key properties, and then design estimation and control algorithms. Figure 1.5 provides the essential workflow of this book covering these topics. In describing a system, we cover topics from state-space description and system realization theory to linearization and discretization of system models. We will provide solutions to control systems from the transfer-function domain to the state-space domain. Build on the system description and solution concepts, we analyze system properties and exploit the importance of stability, controllability, and

observability, along with foundational properties such as causality and linearity. In estimation and controls, we integrate all topics and discuss the power of control design when certain system properties are met. We cover state estimation in both the deterministic and the stochastic cases. We discuss the power of feedback control from arbitrary pole placement of closed-loop eigenvalues to LQR and LQG optimal feedback control.



Figure 1.5: Overview of book topics from system description to estimation and controls.

For *implementation* of controls, control engineers use computers extensively in both (off-line) analysis/design and (real-time) implementation. Computers are inherently discrete devices. If a computer is used in real-time control, it receives the output of the controlled plant as a sensor signal intermittently often at a fixed sampling frequency after an analog to digital (A/D) converter, computes the right control input and send it out to the controlled plant, as depicted in Figure 1.6. The discrete control sequence is then sent to a digital to analog (D/A) converter to form the continuoustime control command to the plant. From the viewpoint of the computer, the plant is a **discrete-time** device that produces a discrete-time output sequence in response to a control sequence provided by the computer.



Figure 1.6: Structure of digital control systems.

While computers are used in almost all implementation of control, the controller design, however, may be carried out in the **continuous-time**



Figure 1.7: Design approaches of digital control systems.

domain or in the **discrete-time domain**. The choice of the time domain may depend on many factors such as target systems, control methodologies, and personal taste. It is important that the control engineers have a broad knowledge of the analysis and design tools for control systems. For example, there are many design approaches to digital control systems. Figure 1.7 summarizes typical design approaches that we may follow given a physical plant. The continuous-time linear control theory and the discrete-time control theory that we will study will provide us an important and useful set of tools.

1.4 Societies to Learn More about Controls

Founded in Paris in 1957, the International Federation of Automatic Control (IFAC, website: https://www.ifac-control.org) is the worldwide organization tasked with promoting the science and technology of automatic control in both theory and application. IFAC also disseminate the impact of control technology on society through its conferences, publications, technical committees, and journals. IFAC is well known through the editorship of eight archival journals:

- ► Automatica,
- Control Engineering Practice,
- Annual Reviews of Control,
- ► Engineering Applications of Artificial Intelligence,
- ► Journal of Process Control,
- Mechatronics,

1 Introduction

- Nonlinear Analysis: Hybrid Systems, and
- ► IFAC Journal of Systems and Control.

These are published in partnership with the official IFAC publisher, Elsevier. Control Engineering Practice and Annual Reviews of Control, for example, are good starting points that are rich in examples and reviews of recent advances in controls.

The American Automatic Control Council (AACC, website: www.a2c2.org) represents the United States to the global control community and is the US National Member Organization (NMO) of IFAC. AACC helps arrange for IFAC events in the U.S. and is an association of nine professional societies:

- American Institute of Aeronautics and Astronautics (AIAA)
- American Institute of Chemical Engineers (AIChE)
- American Society of Civil Engineers (ASCE)
- ► American Society of Mechanical Engineers (ASME)
- ► Institute of Electrical and Electronics Engineers (IEEE)
- Institute for Operations Research and the Management Sciences Applied Probability Society (INFORMS APS)
- International Society of Automation (ISA)
- Society for Industrial and Applied Mathematics (SIAM)
- Society for Modeling and Simulation International (SCS)

AIAA, ASME, and IEEE, for example, publish journals such as

- IEEE Control Systems Magazine
- IEEE Transactions on Control Systems Technology
- ▶ IEEE Transactions on Automatic Control
- ► AIAA Journal of Guidance, Control and Navigation
- ► ASME Journal of Dynamic Systems, Measurement and Control

These are good starting good points at AACC to learn more about controls in both theory and applications.

System Description



2 Modeling

Modeling of physical systems is a vital component of modern engineering. After we understand the governing dynamics of a system, we can simulate and predict system responses, design model-based controllers, and evaluate system properties.

2.1 Methods of Modeling

The dynamics of many systems often consist of complex coupled differential or difference equations. Two general approaches exist to extract these system models. The first and more physics-based approach capitalizes on principles of physics such as Newton's laws and energy conservation. The second and more data-centric approach integrates input-output responses to extract the system dynamics.¹

The most successful modeling often integrates both physics- and data-based modeling and analysis techniques. We provide an example below.

In a hard disk drive storage system, the main dynamics of the voice coil motor that rotates the read/write head are governed by Newton's second law for rotation:



Let the angular position θ be the output and τ be the input. Then the input-output dynamics follow the formula:

$$\ddot{\theta} = \alpha = \frac{1}{J}\tau.$$

2.1 Methods of Modeling . . . 9 2.2 Continuous-Time Systems 10 Discrete-Time Systems . . 11 2.3 **Example: Atomic Force** 2.4 Microscopy 11 **Example: Hard Disk Drive** 2.5 and Information Storage . 14 Model Properties 19 2.6 Nonlinear Systems 20 2.7 "All Models are Wrong, but 2.8 Some are Useful" 23 2.9

1: The field of system identification and adaptive control is dedicated to such databased approach to model and control dynamic systems.



Figure 2.1: Major components in a hard disk drive. https://en.wikipedia.org/wiki/ Hard_disk_drive#/media/File: Hard_drive-en.svg

However, hard disk drives are high-speed high-precision (nanometerscale!) mechatronic systems. In addition to the above fundamental mode, at high angular speeds and frequencies, the rotating disks and actuator arms become no longer rigid, but instead will bend and exhibit highorder vibration modes. Figure 2.2 shows the frequency response of a typical voice coil motor in modern hard disk drives. Many vibration modes appear at high frequencies. The parameters of these modes are not as easy to obtain analytically. Finite element methods and system identification become useful here. However, the analysis from physics is still critical to understand the shapes of these vibration modes.





Figure 2.2: Example frequency response of the voice coil motor in a hard disk drive.

2.2 Continuous-Time Systems

Mathematical models of continuous dynamic systems are differential equations. Here, inputs and outputs of the continuous-time systems are defined for all t, i.e. u(t) and y(t). Continuous linear dynamic systems are described by linear differential equations in the form of

$$\frac{d^{n}y(t)}{dt^{n}} + a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \dots + a_{0}y(t) = b_{m}\frac{d^{m}u(t)}{dt^{m}} + b_{m-1}\frac{d^{m-1}u(t)}{dt^{m-1}} + \dots + b_{0}u(t),$$
(2.1)

with the initial conditions $y(0) = y_0, \ldots, y^{(n)}(0) = y_0^{(n)}$, where $y^{(n)}$ is a shorthand of $\frac{d^n y}{dt^n}$.

Example 2.2.1 (Mass spring damper) Consider a mass spring damper system:



Newton's second law gives

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t), \ y(0) = y_0, \ \dot{y}(0) = \dot{y}_0.$$
 (2.2)

The system is modeled as a second-order ordinary differential equation (ODE) with input u(t) and output y(t).

2.3 Discrete-Time Systems

Inputs and outputs of discrete-time systems are defined at discrete-time points, i.e. u(k) and y(k), where k = 0, 1, 2, ... Models of discrete dynamic systems are difference equations in the form of:

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-n), u(k), u(k-1), \dots, u(k-n)),$$

where the output at k + 1 depends on the input and output at $k, k - 1, \ldots, k - n$.

Example 2.3.1 (Banking and Interest Rate) In a bank account, let x(k) denote the beginning balance at the *k*-th month, and let u(k) denote the accumulated deposit/credit or payment/debit during the *k*-th month). A discrete dynamic model to describe the balance at the beginning of every month is

$$x(k+1) = (1 + a(k))x(k) + u(k),$$

where a(k) is the interest rate of the *k*-th month. This model is used for a variety of purposes, for example, to predict the balance after 12 months assuming the interest rate and the pattern of deposit during the period.

2.4 Example: Atomic Force Microscopy

One of the most powerful techniques for imaging nanoscale objects is Atomic Force Microscopy (AFM). It can reveal the fine details of surfaces ranging from single molecules to the uneven texture of a glass pane. AFM has many applications in various fields of research, such as cell biology, semiconductors, thin film and coatings, tribology (surface and friction interactions), molecular biology, and energy storage and energy generation (photovoltaic) materials. The key to the high performance of AFM is the



Figure 2.3: Isaac Newton (1642-1726) developed Newton's laws in 1686. He is an extremely brilliant scientist and in the meantime often known to be eccentric. He was described as "...so absorbed in his studies that he forgot to eat". https://en.wikipedia.org/wiki/ Isaac_Newton#/media/File: Portrait_of_Sir_Isaac_Newton, 1689.jpg

feedback control system that enables precise scanning at the nanometer level.

The AFM system has two working modes: the tapping mode and the contact mode. A contact mode AFM system that images a sample surface is shown in Figure 2.4. The system consists of a cantilever with an atomic-point needle that scans the sample's surface. The contact point follows the surface topology by moving up and down. A laser beam is directed at the cantilever and is reflected onto a photodiode that measures the beam's exact location. Based on this information, a control system can adjust the position of the cantilever (or the sample under it). The height of the point is recorded as the surface height at that location. A map of the surface can be created by combining the heights from the whole scan.



Figure 2.4: Schematic of the AFM.

The control system aims to maintain a constant force on the sample surface by the needle tip. The cantilever deflection changes with the force on the needle, and the photodiode can sense the laser movement. This information is fed into the control system, which adjusts the sample height to keep the cantilever deflection at the desired level.

The sample height is controlled by a piezoelectric stack actuator element (piezo) under the sample. Piezo elements are crystals that deform according to the electric charge they receive. This deformation is used to move the sample up or down in the z-direction, to regulate the force on the needle. Usually, piezoelectric stack actuators are also used to move the sample in the *x* and *y* directions, but since scanning is done at a constant speed and pattern, these piezo elements do not require the same precision and bandwidth in their feedback control as the z-direction piezo.

The needle encounters forces on the nanoscale that are not obvious to us. Forces such as the attractive van der Waals force, which pulls molecules together, and the repulsive Pauli force, which pushes molecules away, influence the needle as it moves across the surface, in addition to the reaction force from the surface. These forces are nonlinear. However, the major mode of AFM is a spring-mass-damper system. Reference [1] created two models, one second-order and one fourth-order, to describe the dynamics of the AFM system simply, and then more accurately.

Figure 2.5 shows the second-order system. Here, M_1 models the sample and the upper part of the piezo element, and has a combined mass of m_1 . M_2 models the lower part of the piezo element and the mass of the supporting element beneath, with a combined mass of m_2 . The piezo

[1]: Schitter et al. (2007), Design and Modeling of a High-Speed AFM-Scanner

element resizes due to force F, which affects the two masses at the center of the piezo element. The supporting element has a spring constant of k_2 and a damping coefficient of b_2 . The input to the system is force F, which is generated by a voltage signal that causes the piezo element to expand and contract.



Figure 2.5: Mechanical models of an AFM.

Based on Newton's law, the governing equation of the second-order model of the AFM here is

$$m_1 \frac{d^2 x_1}{dt^2} = F,$$

$$m_2 \frac{d^2 x_2}{dt^2} = -b_2 \frac{dx_2}{dt} - k_2 x_2 - F,$$

$$l = x_1 - x_2,$$
(2.3)

where *l* is the distance between x_1 and x_2 as shown in Figure 2.5.

To account for the piezo element dynamics, we need to modify the system to a fourth-order model. This is done by adding another spring-damper component. The final schematic model in Figure 2.5 shows k_1 as the effective spring constant and b_1 as the effective damping coefficient of the piezo element. Force F still affects both masses. The fourth-order model includes the dynamics of the second-order model and also captures high-order dynamics observed in the frequency response of a real-world AFM system [2].

Using first principles in Figure 2.5, when spring and damping effects are considered between the two masses, the governing equations become:

$$m_1 \frac{d^2 x_1}{dt^2} = -b_1 \left(\frac{dx_1}{dt} - \frac{dx_2}{dt} \right) - k_1 (x_1 - x_2) + F,$$

$$m_2 \frac{d^2 x_2}{dt^2} = b_1 \left(\frac{dx_1}{dt} - \frac{dx_2}{dt} \right) + k_1 (x_1 - x_2) - b_2 \left(\frac{dx_2}{dt} - 0 \right) - k_2 x_2 - F.$$
(2.4)

Fourth order is known to be the highestorder model for an AFM system that provides benefits in control design. Models with higher orders do not enhance the precision much, but instead will increase the computational cost considerably.

[2]: Schitter et al. (2001), High performance feedback for fast scanning atomic force microscopes



Figure 2.6: Control-related components in a hard disk drive.

Read more about the HDD mechatronics at, e.g.,

- Hard disk drive Wikipedia. https://en.wikipedia.org/wiki/ Hard_disk_drive.
- 2. Anatomy of a Storage Drive: Hard Disk Drives | TechSpot. https: //www.techspot.com/article/ 1984-anatomy-hard-drive/ Accessed 6/7/2023.
- 3. Hard disk | Definition & Facts | Britannica. https://www.britannica.com/ technology/hard-disk Accessed 6/7/2023.

The example is based on: Takenori (2023), Atsumi Magnetic-head positioning control system in HDDs (https://www.mathworks. com/matlabcentral/fileexchange/ 111515-magnetic-head-positioning-MATLAB control-system-in-hdds), Central File Exchange. Retrieved June 9, 2023

2.5 Example: Hard Disk Drive and Information Storage

Hard disk drives (HDDs) are amazing mechanical systems that store and retrieve digital data using magnetic storage. They consist of one or more rigid rapidly rotating platters coated with magnetic material, and a readwrite head that moves across the platter surface to access the data. HDDs were the standard storage system for personal computers for over 30 years and have been the main storage element in data centers. Their history goes back to the 1950s when IBM invented the first HDD. Since then, HDDs have undergone tremendous improvements in terms of capacity, speed, size, power consumption and reliability.

HDDs are examples of high-precision engineering, as they operate at nanometer scales and millisecond speeds, while being mass-produced and affordable. HDDs are also versatile, as they can store any kind of digital data, from text and images to audio and video. They are truly remarkable devices that have revolutionized the field of data storage and enabled the development of modern computing.

In a modern HDD, data is stored in circular patterns of magnetization known as data tracks or simply, tracks (Figure 2.6). During reading and writing of the data, the disk spins and the read/write head is controlled to follow the circular tracks. This creates the track-following problem, where the servo system performs regulation control to position the read/write head at the desired track, with as low variance as possible. During track following, the position errors are measured periodically at servo sectors that are embedded uniformly over one period of rotation of the disk. Suppose a disk has a rotational speed of 7200 revolutions per minute (rpm) and the number of servo sectors are 220. Then at every revolution of the disks, 220 measurements are obtained, at a sampling frequency of $220 \times 7200/60$ (= 26, 400) Hz.

The actuator in a single-stage HDD is powered by a voice coil motor (VCM). The dynamics between the input current to the voice coil motor and the output position error signal of the read/write head include the effects of inertia, damping, spring constant, and resonant modes of the head assembly. In Section 2.1, we have seen that by Newton's law, the dynamics again has a nominal response of a double integrator. Sometimes, the nominal model also considers friction effects and is written as a second-order damped system instead of a pure double integrator. In high precision control, multiple high-frequency modes are typically present due to structural resonances (Figure 2.7) and the full-order model of the VCM system is in the form of:

$$G(s) = K \sum_{i=1}^{n} \frac{\kappa_i}{s^2 + 2\zeta_i \omega_i s + \omega_i^2}.$$
 (2.5)

The following codes establish a VCM model of a 7200 rpm HDD with 420 servo sectors.

```
% modeling/hddvcm.m
```

```
% MATLAB code to generate a single-stage HDD model
num_sector=420; % Number of sector
```



The Python version of the code is as follows:

Figure 2.7: Baseline model of the VCM component in an HDD.

16 | 2 Modeling

```
# modeling/hddvcm.py
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
import control as ct
num_sector = 420 # Number of sector
num_rpm = 7200 # Number of RPM
Ts = 1 / (num_rpm / 60 * num_sector) # Sampling time
# VCM
Kp_vcm = 3.7976e+07 # VCM gain
omega_vcm = np.array([0, 5300, 6100, 6500, 8050, 9600, 14800, 17400,
                     21000, 26000, 26600, 29000, 32200, 38300, 43300,
                     → 44800]) * 2 * np.pi
kappa_vcm = np.array([1, -1.0, +0.1, -0.1, 0.04, -0.7, -
                    0.2, -1.0, +3.0, -3.2, 2.1, -1.5, +2.0, -0.2, +0.3,
                     \rightarrow -0.5])
zeta_vcm = np.array([0, 0.02, 0.04, 0.02, 0.01, 0.03, 0.01,
                    0.02, 0.02, 0.012, 0.007, 0.01, 0.03, 0.01, 0.01,
                    \rightarrow 0.01])
Sys Pc vcm c1 = ct.TransferFunction(
   [], [1]) # Create an empty transfer function
for i in range(len(omega vcm)):
    Sys_Pc_vcm_c1 = Sys_Pc_vcm_c1 + ct.TransferFunction(np.array(
        [0, 0, kappa_vcm[i]]) * Kp_vcm, np.array([1, 2 * zeta_vcm[i] *
         \rightarrow omega_vcm[i], (omega_vcm[i]) ** 2]))
# Frequency response
f = np.logspace(1, np.log10(60e3), 3000)
w = f * 2 * np.pi
magPc_vcm, phase_Pc_vcm, omega_Pc_vcm = ct.freqresp(
    Sys_Pc_vcm_c1, w) # Get the frequency response
plt.figure()
plt.subplot(211)
plt.semilogx(f, 20*np.log10(magPc_vcm))
plt.title('$P_{cv}$')
plt.xlabel('Frequency [Hz]')
plt.ylabel('Gain [dB]')
plt.grid()
plt.axis([10, f[-1], -90, 100])
plt.subplot(212)
plt.semilogx(f, np.mod(phase_Pc_vcm*180/np.pi+360, 360)-360)
plt.xlabel('Frequency [Hz]')
plt.ylabel('Phase [deg.]')
plt.grid()
plt.axis([10, f[-1], -360, 0])
plt.yticks(np.arange(-360, 90, 90))
```

With the ever increasing demand of larger capacity in HDDs, dual-stage actuation using both a VCM and a piezoelectric actuator has become an essential technique to break the bottleneck of the servo performance in single-actuator HDDs. A dual-stage HDD applies an additional piezoelectric microactuator (MA) at the end of the VCM actuator, as shown in Figure 2.8. We have discussed how piezoelectric microactuators are useful for high-precision control in AFMs. The MA stage has much smaller moving range but greatly improved positioning speed and accuracy. Its dynamical response is also much easier to control, with the low-frequency dynamics governed simply by a DC gain (Figure 2.9). Compared to the VCM actuator,

the MA has enhanced mechanical performance in the high-frequency region, providing the capacity to greatly increase the servo bandwidth and disturbance-attenuation capacity.

Only the position error of the read/write head is measurable in practice. The plant is hence a dual-input-single-output system.



The following codes establish a normalized MA model of a 7200 rpm HDD with 420 servo sectors.

```
% modeling/hddpzt.m
% MATLAB code to generate the pzt-stage HDD model
                                 % Number of sector
num sector=420:
num rpm=7200;
                                  % Number of RPM
Ts = 1/(num_rpm/60*num_sector);
                                % Sampling time
% PZT
omega_pzt=[14800 ,21500 ,28000 ,40200 ,42050,44400,46500
→ ,100000]*2*pi;
kappa_pzt=[-0.005,-0.01,-0.1,+0.8,0.3,-0.25,0.3,10.0];
zeta_pzt =[0.025 ,0.03 ,0.05 ,0.008 ,0.008 ,0.01 ,0.02 ,0.3 ];
Sys_Pc_pzt_c1=0;
for i=1:length(omega_pzt)
       Sys_Pc_pzt_c1=Sys_Pc_pzt_c1+tf([0,0,kappa_pzt(i)],[1,
        \hookrightarrow 2*zeta_pzt(i)*omega_pzt(i), (omega_pzt(i))^2]);
end
Sys_Pc_pzt_c1=Sys_Pc_pzt_c1/abs(freqresp(Sys_Pc_pzt_c1,0));
%% Frequency response
f=logspace(1,log10(60e3),3000);
Fr_Pc_pzt_c1=squeeze(freqresp(Sys_Pc_pzt_c1,f*2*pi)).';
figure
subplot(211)
semilogx(f,20*log10(abs(Fr_Pc_pzt_c1)))
```

17

```
title('P_{cp}');xlabel('Frequency [Hz]');ylabel('Gain

→ [dB]');grid;axis([1e3 f(end) -10 30])

subplot(212)

semilogx(f,angle(Fr_Pc_pzt_c1)*180/pi)

xlabel('Frequency [Hz]');ylabel('Phase [deg.]');grid;axis([1e3 f(end)

→ -180 180]);yticks(-180:90:180)
```

Here is the model construction in Python:

```
# modeling/hddpzt.py
import numpy as np
import matplotlib.pyplot as plt
import control as ct
num_sector = 420 # Number of sector
num_rpm = 7200 # Number of RPM
Ts = 1 / (num_rpm / 60 * num_sector) # Sampling time
# PZT
omega_pzt = np.array([14800, 21500, 28000, 40200, 42050,
                     44400, 46500, 100000]) * 2 * np.pi
kappa_pzt = np.array([-0.005, -0.01, -0.1, +0.8, 0.3, -0.25, 0.3,
\rightarrow 10.0])
zeta_pzt = np.array([0.025, 0.03, 0.05, 0.008, 0.008, 0.01, 0.02,
\rightarrow 0.31)
s = ct.TransferFunction.s # Create a variable for the differentiation
\hookrightarrow operator
Sys_Pc_pzt_c1 = 0 # Create an empty transfer function
for i in range(len(omega_pzt)):
   Sys_Pc_pzt_c1 += kappa_pzt[i] / (s**2 + 2 * zeta_pzt[i] *
    \hookrightarrow omega_pzt[i]
                                      * s + (omega_pzt[i]) ** 2) # Add
                                      \rightarrow the transfer functions
Sys_Pc_pzt_c1 /= Sys_Pc_pzt_c1(0) # Normalize the gain at zero
\hookrightarrow frequency
# Frequency response
f = np.logspace(1, np.log10(60e3), 3000)
w = f * 2 * np.pi
magPc_pzt, phase_Pc_pzt, omega_Pc_pzt = ct.freqresp(
    Sys Pc pzt c1, w) # Get the frequency response
plt.figure()
plt.subplot(211)
plt.semilogx(f, 20*np.log10(magPc_pzt))
plt.title('$P_{cp}$')
plt.xlabel('Frequency [Hz]')
plt.ylabel('Gain [dB]')
plt.grid()
plt.axis([1000, f[-1], - 10, 30])
plt.subplot(212)
plt.semilogx(f, phase_Pc_pzt*180/np.pi)
plt.xlabel('Frequency [Hz]')
plt.ylabel('Phase [deg.]')
plt.grid()
plt.axis([1000, f[-1], - 180, 180])
plt.yticks(np.arange(-180, 270, 90))
```

With the VCM and PZT models, a dual-stage HDD model can be formed as follows.

```
% MATLAB:
Sys_Pc = [Sys_Pc_vcm_c1; Sys_Pc_pzt_c1];
```

Python
Sys_Pc = ct.append(Sys_Pc_vcm_c1, Sys_Pc_pzt_c1)

2.6 Model Properties

We now formalize key properties of system models for control purpose. Consider a general system \mathcal{M} with input $u(\sigma)$ ($\sigma = t$ or k depending on whether the signal is in the continuous- or discrete-time domain) and output $y(\sigma)$:

$$u \longrightarrow \mathcal{M} \longrightarrow y$$

 \mathcal{M} is said to be

- *memoryless* or *static* if $y(\sigma)$ depends only on $u(\sigma)$,
- *dynamic* (has memory) if y at the current time depends on input values at other times.

For example, $y(t) = \mathcal{M}(u(t)) = \gamma u(t)$ is memoryless; $y(t) = \int_0^t u(\tau) d\tau$ and $y(k) = \sum_{i=0}^k u(i)$ are dynamic.

The system \mathcal{M} is *linear* if it satisfies the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(\sigma) + \alpha_2 u_2(\sigma)) = \alpha_1 \mathcal{M}(u_1(\sigma)) + \alpha_2 \mathcal{M}(u_2(\sigma))$$

for any input signals $u_1(\sigma)$ and $u_2(\sigma)$, and any real numbers α_1 and α_2 . If not, the system is nonlinear.

 \mathcal{M} is *time-invariant* if its properties do not change with respect to time. Assuming the same initial conditions, if we shift $u(\sigma)$ by a constant time interval, then \mathcal{M} is time-invariant if the output $\mathcal{M}(u(\sigma + \tau_0)) = y(\sigma + \tau_0)$. For example, $\dot{y}(t) = Ay(t) + Bu(t)$ is linear and time-invariant; $\dot{y}(t) = 2y(t) - \sin(y(t))u(t)$ is nonlinear, yet time-invariant; $\dot{y}(t) = 2y(t) - t \sin(y(t))u(t)$ is time-varying. Often, we abbreviate linear time-invariant systems as LTI systems.

In mechanical systems, torque limits of motors, hardening spring, Coulomb friction forces, etc. make systems nonlinear. However, even if physical systems are nonlinear, they can often be linearized or be well represented by linear systems under specific conditions, making linear analysis and design tools powerful.

The system \mathcal{M} is called

- *causal* if y(t) or y(k) depends on $u(\tau)$ for $\tau \le t$ or k,
- *strictly causal* if the inequality is strict.

For example, y(t) = u(t - 10) is strictly causal.

A system that is not causal is said to be acausal.

For an LTI continuous-time system to be causal, the order of the denominator must be greater than or equal to the order of the numerator in its transfer

function. For example, for the model in Equation 2.1 to be causal, it must be that $n \ge m$. (Check, e.g., the case with n = 0 and m = 1.)

2.7 Nonlinear Systems

Although we will be mostly focusing on linear systems, in practice, many control systems are nonlinear. Several examples are presented next to show how the nonlinearity plays a role in modeling.

2.7.1 Example: Magnetically Suspended Ball

A magnetically suspended ball consists of a ball of mass *m* suspended by an electromagnet as shown in Figure. 2.10. Let y be the position of the ball, measured down from the base of the electromagnet. If a current *u* is injected into the coils of the electromagnet, it will produce a magnetic field which in turn exerts an upward force on the ball given by $F_{up} = -\frac{cu^2}{u^2}$. Note that the force decreases as y^2 increases because the effect of the magnet weakens when the ball is further away, and that the force is proportional to u^2 which is representative of the power supplied to the magnet.

Let us assume that we can measure the position y of the ball. This can be arranged optically using light-emitting diodes with photocells.

We can then write a simple model for the motion of the ball from Newton's second law as

$$m\ddot{y} = mg - \frac{cu^2}{y^2}.$$
 (2.6)

Note that this is a single-input, single-output nonlinear model. The input and output of this model are:

> the current injected into the magnet coils U

the position of the ball Y

2.7.2 Example: Water Tank

Consider the water tank in Figure. 2.11. Denote by $q_{in}(t)$ the water flow entering the tank and $q_{out}(t)$ the water flow leaving the tank from a hole of area a. Out goal is to obtain a model that describes the evolution of the tank height h(t) as a function of the external input $u(t) = q_{in}(t)$. Let *A* be the area of the tank. By using the conservation law we can state that

$$A\dot{h}(t) = q_{\rm in}(t) - q_{\rm out}(t).$$

Let $v_{out}(t)$ be the speed of the water at the outlet. Then, $q_{out}(t) = av_{out}(t)$.

Torricelli's law states that the speed of a fluid through a sharp-edged hole under the force of gravity is the same as the speed that a body would acquire in falling freely from a height *h*, i.e. $v_{out}(t) = \sqrt{2gh(t)}$, where *g* is



Figure 2.11: Water flow in a tank.



Figure 2.10: Free-body diagram of a mag-

netically suspended ball.
the acceleration due to gravity. Considering all the above, our *nonlinear* model is

$$\dot{h} = \frac{1}{A}(u(t) - a\sqrt{2gh(t)}).$$

2.7.3 Example: Pendulum

Consider the pendulum shown in Figure. 2.12. We assume that the pendulum has mass m = 0.333 kg which is concentrated at the end point and length l = 0.75 meters. The angle θ is measured, and an actuator can supply a torque $u(t) = T_c(t)$ to the pivot point. The moment of inertia about the pivot point is $I = ml^2$. By analyzing the rigid body dynamics and writing Euler's equation for the pendulum, we can readily arrive at a differential equation model for this system:

$$I\ddot{\theta}(t) = T_c - mgl\sin(\theta(t)),$$

or,

$$\ddot{\theta}(t) = \frac{1}{ml^2}T_c - \frac{g}{l}\sin(\theta(t)),$$

which is a second-order nonlinear differential equation.

2.7.4 Example: Vehicle Steering





Figure 2.12: Free-body diagram of a pendulum system.

Figure 2.13: Kinematic modelling of the four-wheel vehicle steering system.

Figure 2.13 shows the kinematics of a four-wheeled vehicle. If we assume the front and rear pairs of wheels act the same as a single wheel on each axle at the centerline of the vehicle, we arrive at a two-wheel abstraction known as the bicycle model. The bicycle model does not take into account tilting of the vehicle. A four-wheeled vehicle would experience very different tilting mechanics than those of a two-wheeled vehicle. However, when only the steering of the vehicle is of concern with the assumption that it cannot tilt, such as in vehicles with a low center of gravity and operating at slow enough speeds, the bicycle model is appropriate.

When the vehicle turns, the inner and outer tires exhibit different steering angles due to their varying distances from point *O* in the bicycle model in

Read more about the bicycle model at, e.g., https://theflclan.com/2020/09/21/ vehicle-dynamics-the-kinematic-bicycle-model/.



Figure 2.14: The bicycle model of the vehicle steering system.

Figure 2.14. Here, δ is the steering angle of the front wheels, b is the wheel base, a is the distance between the rear axle and the center of mass, x and y are the positions of the center of mass, θ is the heading, and α is the angle between the velocity v and the centerline of the vehicle. The point O is the intersection of the centerlines of the of the front and rear wheels. By letting the distance from the center of rotation O to the rear wheel contact point be r_r , we can deduce that $b = r_r \tan \delta$ and $a = r_r \tan \alpha$. This leads to the relationship between α and the steering angle δ :

$$\alpha = \arctan\left(\frac{a\,\tan\delta}{b}\right).\tag{2.7}$$

Given a vehicle speed v at its center of mass, the motion of the center of mass is expressed as:

$$\frac{dx}{dt} = v \cos(\alpha + \theta),$$

$$\frac{dy}{dt} = v \sin(\alpha + \theta).$$
(2.8)

To determine the influence of the steering angle on the heading angle θ , we note that the distance from the center of mass to the center of rotation *O* is $r_c = a/\sin \alpha$. As the vehicle rotates around point *O*, the angular velocity is given by $v/r_c = (v/a) \sin \alpha$. Therefore,

$$\dot{\theta} = \frac{v}{r_c} = \frac{v \sin \alpha}{a} = \frac{v}{a} \sin \left(\arctan \left(\frac{a \tan \delta}{b} \right) \right).$$
(2.9)

When the steering angle δ and the angle α (known as the slip angle) is small, the above equation becomes

$$\dot{\theta} \approx \frac{v}{b}\delta.$$
 (2.10)

Let the input u be the steering angle δ . The full set of nonlinear equations of motion for the vehicle steering problem is now:

$$\frac{d}{dt} \underbrace{ \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} }_{x} = f(\mathbf{x}, u) = \begin{bmatrix} v \cos(\alpha(u) + \theta) \\ v \sin(\alpha(u) + \theta) \\ \frac{v \sin\alpha(u)}{a} \end{bmatrix}, \ \alpha(u) = \arctan\frac{a \tan u}{b}.$$
(2.11)

2.8 "All Models are Wrong, but Some are Useful"

A perfect model is very hard to be obtained in reality. In the Journal of the American Statistical Association [3], British statistician George Box famously wrote in 1976, that "all models are wrong, but some are useful." The aphorism acknowledges that statistical models always fall short of the complexities of reality but can still be useful nonetheless. A dynamic system may simply be too complex (consider the neural system of human brains), or there are inevitable hardware uncertainties such as the fatigue of gears or bearings in a car. Modeling thus involves varying degrees of approximation. For example in modeling a car, we may chose to ignore rolling resistance, aero-dynamic effects, road-tire interactions, etc.

However, feedback control can empower a system to tolerate model uncertainties. In a 5-year-old car, the same engine control unit (ECU) can maintain smooth performance of the internal combustion engine despite the accumulated mileage and wear, whether they come from highways or mountains. In the hard disk drive example in Section 2.1, the highfrequency vibration modes can change their shapes and locations in a batch of products and when working under different temperatures, leading to the frequency responses in Figure 2.15 [4]. However, thanks to the robustness from feedback controls, when making thousands of hard disk drives per day, the manufactures do not need to tune the controller in each individual drive. Instead, each batch can robustly read and write data using the same servo controller.

The following MATLAB codes generate the perturbed HDD model in Figure 2.15.

```
% modeling/hdddsa.m
% Dual-stage HDD model
num sector=420:
                                 % Number of sector
                                % Number of RPM
num rpm=7200:
Ts = 1/(num_rpm/60*num_sector); % Sampling time
% VCM
Kp vcm=3.7976e+07;
omega_vcm=[0, 5300,6100,6500,8050,9600,14800,17400,21000,26000
→ ,26600 ,29000 ,32200 ,38300 ,43300 ,44800]*2*pi;
kappa_vcm=[1, -1.0 ,+0.1 ,-0.1 ,0.04 ,-0.7 ,-0.2 ,-1.0 ,+3.0 ,-3.2
\hookrightarrow ,2.1 ,-1.5 ,+2.0 ,-0.2 ,+0.3 ,-0.5 ];
zeta_vcm =[0, 0.02 ,0.04 ,0.02 ,0.01 ,0.03 ,0.01 ,0.02 ,0.02 ,0.012
\rightarrow ,0.007 ,0.01 ,0.03 ,0.01 ,0.01 ,0.01 ];
% PZT
omega_pzt=[14800 ,21500 ,28000 ,40200 ,42050,44400,46500
\rightarrow ,100000]*2*pi;
```

[3]: Box (1976), Science and statistics

Robust control is not tied to complex control. A PID controller can create robust stability too.

[4]: Atsumi (2022), Magnetic-head Positioning Control System in HDDs

The subfield of robust control covers details of why and how to handle the model uncertainties.

24 2 Modeling



Figure 2.15: Example frequency responses of the voice coil motor stage in a batch of hard disk drives.

```
kappa_pzt=[-0.005,-0.01 ,-0.1 ,+0.8 ,0.3 ,-0.25 ,0.3 ,10.0 ];
zeta_pzt =[0.025 ,0.03 ,0.05 ,0.008 ,0.008 ,0.01 ,0.02 ,0.3 ];
%% LT(Low temp.) model: VCM: +4 % resonance shift from nominal
\, \hookrightarrow \, model, PZT actuator: +6 % resonance shift from nominal
→ model
% VCM
Sys_Pc_vcm_c1=0;
for i=1:length(omega_vcm)
        Sys_Pc_vcm_c1=Sys_Pc_vcm_c1+ss(tf([0,0,kappa_vcm(i)]*Kp_-
        \rightarrow vcm, [1, 2*zeta_vcm(i)*0.8*omega_vcm(i)*1.04,
         \hookrightarrow (omega_vcm(i)*1.04)^2]));
        Sys_Pc_vcm_c1=ssbal(Sys_Pc_vcm_c1);
end
% PZT
Sys_Pc_pzt_c1=0;
for i=1:length(omega_pzt)
        Sys_Pc_pzt_c1=Sys_Pc_pzt_c1+ss(tf([0,0,kappa_pzt(i)],[1,
         \hookrightarrow 2*zeta_pzt(i)*0.8*omega_pzt(i)*1.06,
         \hookrightarrow (omega_pzt(i)*1.06)^2]));
        Sys_Pc_pzt_c1=ssbal(Sys_Pc_pzt_c1);
end
Sys_Pc_pzt_c1=Sys_Pc_pzt_c1/abs(freqresp(Sys_Pc_pzt_c1,0));
%% RT(Room temp.) model: Same as nominal models
% VCM
Sys_Pc_vcm_c2=0;
for i=1:length(omega_vcm)
        Sys_Pc_vcm_c2=Sys_Pc_vcm_c2+ss(tf([0,0,kappa_vcm(i)]*Kp_-
         \rightarrow vcm, [1, 2*zeta_vcm(i)*omega_vcm(i),
         \hookrightarrow omega_vcm(i)^2]));
        Sys_Pc_vcm_c2=ssbal(Sys_Pc_vcm_c2);
end
% PZT
Sys_Pc_pzt_c2=0;
```

```
for i=1:length(omega pzt)
                  Sys_Pc_pzt_c2=Sys_Pc_pzt_c2+ss(tf([0,0,kappa_pzt(i)],[1,
                   2*zeta_pzt(i)*omega_pzt(i), omega_pzt(i)^2]));
                  Sys_Pc_pzt_c2=ssbal(Sys_Pc_pzt_c2);
end
Sys_Pc_pzt_c2=Sys_Pc_pzt_c2/abs(freqresp(Sys_Pc_pzt_c2,0));
%% HT(High temp.) model: VCM: -4 % resonance shift from nominal
\rightarrow model, PZT actuator: -6 % resonance shift from nominal
        model
% VCM
Sys_Pc_vcm_c3=0;
for i=1:length(omega vcm)
                  Sys_Pc_vcm_c3=Sys_Pc_vcm_c3+ss(tf([0,0,kappa_vcm(i)]*Kp_-
                   \hookrightarrow vcm, [1, 2*zeta_vcm(i)*1.2*omega_vcm(i)*0.96,
                  \hookrightarrow (omega_vcm(i)*0.96)^2]));
                 Sys Pc vcm c3=ssbal(Sys Pc vcm c3);
end
% P7T
Sys Pc pzt c3=0;
for i=1:length(omega_pzt)
                  Sys_Pc_pzt_c3=Sys_Pc_pzt_c3+ss(tf([0,0,kappa_pzt(i)],[1,
                  \hookrightarrow 2*zeta_pzt(i)*1.2*omega_pzt(i)*0.94,
                  \hookrightarrow (omega pzt(i)*0.94)^2]);
                 Sys_Pc_pzt_c3=ssbal(Sys_Pc_pzt_c3);
end
Sys_Pc_pzt_c3=Sys_Pc_pzt_c3/abs(freqresp(Sys_Pc_pzt_c3,0));
%% LT / PZT gain +5% (Case 4)
Sys_Pc_vcm_c4=Sys_Pc_vcm_c1;
Sys_Pc_pzt_c4=Sys_Pc_pzt_c1*1.05;
%% RT / PZT gain +5% (Case 5)
Sys_Pc_vcm_c5=Sys_Pc_vcm_c2;
Sys_Pc_pzt_c5=Sys_Pc_pzt_c2*1.05;
%% HT / PZT gain +5% (Case 6)
Sys_Pc_vcm_c6=Sys_Pc_vcm_c3;
Sys_Pc_pzt_c6=Sys_Pc_pzt_c3*1.05;
%% LT / PZT gain -5% (Case 7)
Sys_Pc_vcm_c7=Sys_Pc_vcm_c1;
Sys_Pc_pzt_c7=Sys_Pc_pzt_c1*0.95;
%% RT / PZT gain -5% (Case 8)
Sys_Pc_vcm_c8=Sys_Pc_vcm_c2;
Sys_Pc_pzt_c8=Sys_Pc_pzt_c2*0.95;
%% HT / PZT gain -5% (Case 9)
Sys_Pc_vcm_c9=Sys_Pc_vcm_c3;
Sys_Pc_pzt_c9=Sys_Pc_pzt_c3*0.95;
%% All plant
Sys_Pc_vcm_all=[Sys_Pc_vcm_c1;Sys_Pc_vcm_c2;Sys_Pc_vcm_c3;Sys_Pc_vcm_-
\hookrightarrow \quad c4; Sys\_Pc\_vcm\_c5; Sys\_Pc\_vcm\_c6; Sys\_Pc\_vcm\_c7; Sys\_Pc\_vcm\_c8; Sys\_-c_vcm\_c7; Sys\_Pc\_vcm\_c8; Sys\_-c_vcm\_c8; Sys\_-c_vcm\_
\rightarrow Pc vcm c9];
Sys_Pc_pzt_all=[Sys_Pc_pzt_c1;Sys_Pc_pzt_c2;Sys_Pc_pzt_c3;Sys_Pc_pzt_-
 → c4;Sys_Pc_pzt_c5;Sys_Pc_pzt_c6;Sys_Pc_pzt_c7;Sys_Pc_pzt_c8;Sys_-
\rightarrow Pc_pzt_c9];
%% Cotrolled object (Discrete-time system)
% Case 1
```

```
Sys_Pd_vcm_c1=c2d(Sys_Pc_vcm_c1,Ts,'ZOH');
```

26 2 Modeling

Sys_Pd_pzt_c1=c2d(Sys_Pc_pzt_c1,Ts,'ZOH'); % Case 2 Sys_Pd_vcm_c2=c2d(Sys_Pc_vcm_c2,Ts,'ZOH'); Sys_Pd_pzt_c2=c2d(Sys_Pc_pzt_c2,Ts,'ZOH'); % Case 3 Sys_Pd_vcm_c3=c2d(Sys_Pc_vcm_c3,Ts,'ZOH'); Sys_Pd_pzt_c3=c2d(Sys_Pc_pzt_c3,Ts,'ZOH'); % Case4 Sys_Pd_vcm_c4=c2d(Sys_Pc_vcm_c4,Ts,'ZOH'); Sys_Pd_pzt_c4=c2d(Sys_Pc_pzt_c4,Ts,'ZOH'); % Case 5 Sys_Pd_vcm_c5=c2d(Sys_Pc_vcm_c5,Ts,'ZOH'); Sys_Pd_pzt_c5=c2d(Sys_Pc_pzt_c5,Ts,'ZOH'); % Case 6 Sys_Pd_vcm_c6=c2d(Sys_Pc_vcm_c6,Ts,'ZOH'); Sys_Pd_pzt_c6=c2d(Sys_Pc_pzt_c6,Ts,'ZOH'); % Case 7 Sys_Pd_vcm_c7=c2d(Sys_Pc_vcm_c7,Ts,'ZOH'); Sys_Pd_pzt_c7=c2d(Sys_Pc_pzt_c7,Ts,'ZOH'); % Case 8 Sys_Pd_vcm_c8=c2d(Sys_Pc_vcm_c8,Ts,'ZOH'); Sys_Pd_pzt_c8=c2d(Sys_Pc_pzt_c8,Ts,'ZOH'); % Case 9 Sys_Pd_vcm_c9=c2d(Sys_Pc_vcm_c9,Ts,'ZOH'); Sys_Pd_pzt_c9=c2d(Sys_Pc_pzt_c9,Ts,'ZOH'); % A11 Sys_Pd_vcm_all=[Sys_Pd_vcm_c1;Sys_Pd_vcm_c2;Sys_Pd_vcm_c3;Sys_Pd_vcm_- \leftrightarrow c4;Sys_Pd_vcm_c5;Sys_Pd_vcm_c6;Sys_Pd_vcm_c7;Sys_Pd_vcm_c8;Sys_- \hookrightarrow Pd_vcm_c9]; Sys_Pd_pzt_all=[Sys_Pd_pzt_c1;Sys_Pd_pzt_c2;Sys_Pd_pzt_c3;Sys_Pd_pzt_- $\hookrightarrow \quad c4; Sys_Pd_pzt_c5; Sys_Pd_pzt_c6; Sys_Pd_pzt_c7; Sys_Pd_pzt_c8; Sys_edpzt_c8; Sys_c8; Sys_$ \hookrightarrow Pd_pzt_c9]; %% Frequency response f=logspace(1,log10(60e3),3000); Fr_Pc_vcm_all=squeeze(freqresp(Sys_Pc_vcm_all,f*2*pi)).'; Fr_Pc_pzt_all=squeeze(freqresp(Sys_Pc_pzt_all,f*2*pi)).'; Fr_Pd_vcm_all=squeeze(freqresp(Sys_Pd_vcm_all,f*2*pi)).'; Fr_Pd_pzt_all=squeeze(freqresp(Sys_Pd_pzt_all,f*2*pi)).'; figure subplot(211) semilogx(f,20*log10(abs(Fr_Pc_vcm_all(:,1:7)))) hold on semilogx(f,20*log10(abs(Fr_Pc_vcm_al1(:,8:9))),'--') hold off title('P_{cv}');xlabel('Frequency [Hz]');ylabel('Gain → [dB]');grid;axis([1e3 f(end) -90 10]) subplot(212) semilogx(f,mod(angle(Fr_Pc_vcm_all(:,1:7))*180/pi+360,360)-360) hold on

semilogx(f,mod(angle(Fr_Pc_vcm_all(:,8:9))*180/pi+360,360)-360,'--')
hold off

```
legend('Case 1', 'Case 2', 'Case 3', 'Case 4', 'Case 5', 'Case 6', 'Case
→ 7', 'Case 8', 'Case 9', 'Location', 'NorthWest')
figure
subplot(211)
semilogx(f,20*log10(abs(Fr_Pc_pzt_all(:,1:7))))
hold on
semilogx(f,20*log10(abs(Fr_Pc_pzt_all(:,8:9))),'--')
hold off
title('P_{cp}');xlabel('Frequency [Hz]');ylabel('Gain
→ [dB]');grid;axis([1e3 f(end) -10 30])
subplot(212)
semilogx(f,angle(Fr_Pc_pzt_all(:,1:7))*180/pi)
hold on
semilogx(f,angle(Fr_Pc_pzt_all(:,8:9))*180/pi,'--')
hold off
xlabel('Frequency [Hz]');ylabel('Phase [deg.]');grid;axis([1e3 f(end)
→ -180 180]);yticks(-180:90:180)
legend('Case 1','Case 2','Case 3','Case 4','Case 5','Case 6','Case
→ 7', 'Case 8', 'Case 9', 'Location', 'NorthWest')
```

2.9 Exercise

- 1. What is the difference between a causal system and an acausal system?
- 2. Can a static system be acausal?
- 3. Can a linear system be time-varying?
- 4. Solve the following ODEs:
 - a) $\ddot{y} + \dot{y} 2y = 0$, y(0) = 4, $\dot{y}(0) = -5$,
 - b) $\ddot{y} + \dot{y} + 0.25y = 0$, y(0) = 3, $\dot{y}(0) = -3.5$,
 - c) $\ddot{y} + 0.4\dot{y} + 9.04y = 0.$
- 5. A quadcopter is a type of unmanned aerial vehicle (UAV) that is controlled by adjusting the speed of its four rotors. Read relevant literature and develop a mathematical model of a quadcopter.
- 6. A robotic arm is used to move objects in a manufacturing facility. The arm consists of several joints that can be controlled to move the end-effector to a desired position. Read relevant literature. Develop a dynamic model between the torque input to the motors and the joint angles.
- 7. A power system consists of generators, transformers, transmission lines, and loads. Read relevant literature and develop a simplified model of a power system.
- 8. A cruise control system in a car maintains a constant speed by adjusting the throttle. The system must be able to handle changes in terrain and driving conditions. Read relevant literature and develop a simplified model of a cruise control system.
- 9. A temperature control system in a chemical plant consists of a heater, a temperature sensor, and a controller. The controller receives input from the temperature sensor and sends a signal to the heater to adjust the temperature. Develop a block diagram of the control system and a mathematical model of the plant.



3 Laplace and Z Transforms

3.1 The Laplace Transform

Consider the summation $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}$. You may recall from calculus that the series converges to a finite value. You may also find that the infinite sum equals ln 2, and that the solution steps can be overwhelming if you do not deal with infinite integrals regularly. It turns out that Laplace transform can help reduce the problem to basic and finite integrals!

Beyond infinite series, for control engineering, the Laplace transform is a foundational tool for modeling and system analysis. In the previous chapter, we have seen various continuous-time system models in the form of Ordinary Differential Equations (ODEs). In this chapter, we review how such ODEs and models can be reformed and simplified for controls with the Laplace transform.

3.1.1 The Laplace Approach to ODEs

The Laplace transform is a powerful tool to solve a wide variety of ODEs. To directly solve an ODE in calculus, we often need to solve a characteristic equation, find the time-domain modes of the solution, and use the method of undetermined coefficients to determine the specific parameters of each mode – a lot of work! With the Laplace transform, the calculus operators in the time domain are replaced by algebraic operations. Algebraic solutions are much easier to obtain, and time-domain ODE solutions can be subsequently obtained via the inverse Laplace transform. Moreover, we will be able to easily manipulate interconnected ODEs and conduct closed-loop analyses – all very useful for dynamic systems and controls.

We start with reviewing a number of relevant mathematical definitions and notations.

3.1	The Laplace Transform 29
3.2	Inverse Laplace Trans-
	form and Partial Fraction
	Expansion
3.3	From Laplace Transform to
	Transfer Functions 41
3.4	The Z Transform 45
3.5	From Difference Equation
	to Discrete-Time Transfer
	Functions 53
3.6	Recap 56
3.7	Exercise



Figure 3.1: The Laplace approach to ODEs.



A set is a well-defined collection of distinct objects, e.g., $\{1, 2, 3\}$. The most relevant sets for us are:

- \blacktriangleright \mathbb{R} : the set of real numbers,
- ▶ C: the set of complex numbers, and
- \mathbb{R}_+ : the set of positive real numbers.

We write $x \in S$ to indicate that x is a member of, or belongs to, set S. For example, $1 \in \mathbb{R}$, the notation \triangleq reads "is defined as": for example, $y(t) \triangleq 3x(t) + 1.$

 $f : \mathbb{R}_+ \to \mathbb{R}$ indicates that f is a continuous function the maps a value from its domain in \mathbb{R}_+ to a value in its codomain \mathbb{R} . We will mostly use f(t) to denote a continuous-time function, whose domain is time. Unless stated otherwise, we assume that f(t) = 0 for all t < 0.

Definition 3.1.1 *For a continuous-time function* $f : \mathbb{R}_+ \to \mathbb{R}$ *, the Laplace* Transform is defined as:

$$F(s) = \mathcal{L}\{f(t)\} \triangleq \int_0^\infty f(t)e^{-st}dt,$$
(3.1)

where $s \in \mathbb{C}$.

Notice that the integration from 0 to ∞ eliminates *t* and the Laplace transform is a function of the complex variable *s*.

For any integration over an infinite horizon, we must pay attention to the condition for the integral to converge. The Laplace transform exists if

- f(t) is piecewise continuous (see, for example, Figure 3.3), and
- f(t) does not grow faster than an exponential as $t \to \infty$:

$$|f(t)| < ke^{\alpha t}$$
, for all $t \ge t_0$,

for some constants k, α , $t_0 \in \mathbb{R}_+$, as illustrated in Figure 3.4. An f(t)satisfying such a property is called to be of exponential order.

From undergraduate controls course(s), you have learnt the following basic Laplace transforms:



Figure 3.2: Pierre-Simon Laplace (1749-1827) is often referred to as the French Newton or Newton of France. Thirteen years more junior than Lagrange, Laplace developped and pioneered astronomical stability, mechanics based on calculus, Bayesian interpretation of probability, mathematical physics, just to name a few. He studied under Jean le Rond d'Alembert (co-discovered fundamental theorem of algebra, aka d'Alembert/Gauss theorem).

https://en.wikipedia.org/wiki/ Pierre-Simon_Laplace#/media/File: Laplace,_Pierre-Simon,_marquis_de.jpg

30



Figure 3.3: Illustration of piecewise continuous functions.

Figure 3.4: Illustration of exponential order.

1. Exponential functions: If $f(t) = e^{-at}$, $a \in \mathbb{C}$, then

$$F(s) = \int_0^\infty e^{-at} e^{-st} dt = \int_0^\infty e^{-(a+s)t} dt$$

= $\int_0^\infty \frac{1}{-(a+s)} de^{-(a+s)t} = \frac{1}{-(a+s)} e^{-(a+s)t} \Big|_0^\infty$ (3.2)
= $\frac{1}{s+a}$.

2. Step functions: As a special case of the exponential function (a = 0), when

$$f(t) = 1(t) = \begin{cases} 1, & t \ge 0\\ 0, & t < 0 \end{cases}$$
(3.3)

then

$$F(s) = \frac{1}{s}$$

3. Ramp function: if

$$f(t) = \begin{cases} t, & t \ge 0\\ 0, & t < 0 \end{cases}$$
(3.4)

- then $\mathscr{L}{t} = \frac{1}{s^2}$. 4. Sine function: If $f(t) = \sin(\omega t)$, then $F(s) = \frac{\omega}{s^2 + \omega^2}$. This is again an application of Equation 3.2. From the Euler formula, we have $\sin(\omega t) = \frac{1}{s^2 + \omega^2}$. $\frac{e^{j\omega t}-e^{-j\omega t}}{2j}$ and hence $\mathscr{L}\{\sin(\omega t)\} = \frac{1}{2j} (\mathscr{L}\{e^{j\omega t}\} - \mathscr{L}\{e^{-j\omega t}\})$, where the Laplace transforms of the two exponential functions are readily available by applying Equation 3.2: $\mathscr{L}\{e^{j\omega t}\} = \frac{1}{s-j\omega}$ and $\mathscr{L}\{e^{-j\omega t}\} = \frac{1}{s-j\omega}$
- $\frac{1}{s+j\omega}$. 5. Cosine function: If $f(t) = \cos(\omega t)$ then $F(s) = \frac{s}{s^2 + \omega^2}$. This is again an application of the Euler formula, which gives $\cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2}$.

The Euler formula establishes the fundamental relationship between the trigonometric functions and the complex exponential function. Euler's formula states that for any real number *a*:

$$e^{ja} = \cos a + j \sin a.$$

Leonhard Euler (04/15/1707-09/18/1783) is a Swiss mathematician, physicist, astronomer, geographer, logician and engineer. He studied under Johann Bernoulli, and taught Lagrange. He wrote 380 articles within 25 years at Berlin, and produced on average one paper per week at age 67, when he was almost blind!

6. Impulse/Dirac Delta function: If $f(t) = \delta(t) \triangleq \lim_{\epsilon \to 0} \frac{\mu(t) - \mu(t-\epsilon)}{\epsilon}$, then F(s) = 1.

Laplace Transform in MATLAB

In MATLAB, the command laplace performs the Laplace transform as illustrated in the following codes.¹

1: The Symbolic Math Toolbox is required here.

```
% laplaceZtransforms/simplelaplace.m
syms a t
% exponential functions
f = exp(-a*t);
F = laplace(f)
F =
1/(a + s)
g = \exp(-2*t);
G = laplace(g)
G =
1/(s + 2)
% ramp function
h = 2*t;
H = laplace(h)
H =
2/s^2
% impulse function
d = dirac(t);
D = laplace(d)
D =
1
```

Laplace Transform in Python

In Python, the Laplace transform can be realized using the sympy package as shown next.²

```
# laplaceZtransforms/simplelaplace.py
import sympy
t, s = sympy.symbols('t, s')
a = sympy.symbols('a', real=True, positive=True)
f = sympy.exp(-a*t)
F = sympy.laplace_transform(f, t, s, noconds=True)
print(F)
g = sympy.exp(-2*t)
G = sympy.laplace_transform(g, t, s, noconds=True)
print(G)
h = 2*t
H = sympy.laplace_transform(h, t, s, noconds=True)
print(H)
d = sympy.DiracDelta(t)
D = sympy.laplace_transform(d, t, s, noconds=True)
print(D)
```

2: You can install the package via: pip install sympy, if not done already. If you use Anaconda, run conda install sympy instead. The results are shown below.³

1/(a + s)1/(s + 2)2/s**2 1

Let us now revisit the calculus problem at the beginning of the chapter, $\sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{n}$. Note that $\mathcal{L} \{1\} = \frac{1}{s}$, we have $\sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{n} = \sum_{n=1}^{\infty} (-1)^{n+1} \int_{0}^{\infty} 1 \cdot e^{-nt} dt$. The integration is over *t* instead of *n*. Hence,

$$\sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{n} = \int_0^{\infty} \sum_{n=1}^{\infty} (-1)^{n+1} e^{-nt} dt$$
$$= \int_0^{\infty} (-1) \sum_{n=1}^{\infty} (-1)^n e^{-nt} dt$$
$$= \int_0^{\infty} (-1) \sum_{n=1}^{\infty} (-e^{-t})^n dt$$
$$= \int_0^{\infty} (-1) \frac{-e^{-t}}{1+e^{-t}} dt$$
$$= -\int_2^1 \frac{du}{u} \qquad (u \triangleq 1+e^{-t})$$
$$= \ln 2$$

With the Laplace transform, only basic calculus tools were used in the solution approach!

The Dirac Delta Function and Distributions

Figure 3.5 illustrates the Dirac impulse function $\delta(t)$. It satisfies the following properties:

- $\int_0^\infty \delta(t-T)dt = 1$, and $\int_0^\infty \delta(t-T)f(t)dt = f(T)$.

Using $\int_0^\infty \delta(t) f(t) dt = f(0)$, we can compute the Laplace transform:

$$\mathcal{L}\{\delta(t)\} = \int_0^\infty e^{-st} \delta(t) dt = e^{-s0} = 1.$$

The impulse function is not a normal function but a generalized function, more formally known as a distribution. To develop an intuition, consider solving a differential equation

$$\dot{y} - ay = \dot{u} + bu$$

where the input u is a unit step function 1(t). On the right-hand side of the equation, we have a differentiation of a unit step function, which is not feasible directly, because the function is not continuous everywhere but has a step jump at time zero.



Figure 3.5: Illustration of the Dirac impulse function.

3: Note that s * * 2 denotes s^2 in Python.

Instead, consider the following limits that approach a unit step:

$$\mu_{\epsilon}(t) := \begin{cases} 0 & \text{for } t < 0\\ \frac{1}{\epsilon}t & \text{for } 0 \le t < \epsilon\\ 1 & \text{for } \epsilon \le t \end{cases}$$
(3.5)

where $\epsilon > 0$. A picture of μ_{ϵ} , with different values of ϵ , is shown in Figure 3.6. We see that as $\epsilon \to 0$, the function indeed approaches to the original unit step.



Figure 3.6: Approximation of the unit-step function.

Note that μ_{ϵ} is continuous, and piecewise differentiable, with derivative

$$\dot{\mu}_{\epsilon}(t) = \begin{cases} 0 & \text{for } t < 0\\ \frac{1}{\epsilon} & \text{for } 0 < t < \epsilon\\ 0 & \text{for } \epsilon < t \end{cases}$$
(3.6)

You can have a plot of $\dot{\mu}_{\epsilon}$ and will see that it is zero for the majority of time, except at time near t = 0. Between time t = 0 and $t = \epsilon$, $\dot{\mu}_{\epsilon}$ has a rectangular shape that is $1/\epsilon$ tall and ϵ wide.

Call the function in Equation 3.6 δ_{ϵ} . Note that for all values of $\epsilon > 0$,

$$\int_{-\infty}^{\infty} \delta_{\epsilon}(t) dt = \int_{-\infty}^{0} 0 dt + \int_{0}^{\epsilon} \frac{1}{\epsilon} dt + \int_{\epsilon}^{\infty} 0 dt$$

$$= 1,$$
(3.7)

and that for t < 0 and $t > \epsilon$, $\delta_{\epsilon}(t) = 0$.

Moreover, for any continuous function f, we have

$$\lim_{\epsilon \to 0} \int_{-\infty}^{\infty} f(t) \delta_{\epsilon}(t) dt = \lim_{\epsilon \to 0} \left[\int_{-\infty}^{0} f(t) 0 dt + \int_{0}^{\epsilon} f(t) \frac{1}{\epsilon} dt + \int_{0}^{\epsilon} f(t) 0 dt \right]$$
$$= \lim_{\epsilon \to 0} \int_{0}^{\epsilon} f(t) \frac{1}{\epsilon} dt$$
$$= f(0),$$
(3.8)

and for any $t \neq 0$, $\lim_{\epsilon \to 0} \delta_{\epsilon}(t) = 0$.

Hence, in the limit we can imagine a "function" δ whose value at nonzero t is 0, whose value at t = 0 is undefined, but whose integral is finite, namely 1.

We cannot differentiate $\dot{\mu}_{\epsilon}(t)$ in Equation 3.6 further. However, we can use another more smoother approximation. Let δ_{ϵ} be defined as

$$\delta_{\epsilon}(t) := \begin{cases} 0 & \text{for } t < 0, \\ \frac{t}{\epsilon^2} & \text{for } 0 < t < \epsilon, \\ \frac{2\epsilon - t}{\epsilon^2} & \text{for } \epsilon < t < 2\epsilon, \\ 0 & \text{for } 2\epsilon < t. \end{cases}$$
(3.9)

Plotted out with different values of ϵ , this looks like the following in Figure 3.7.



Figure 3.7: Approximation of the Diracdelta function.

Note that independent of ϵ , we have

$$\int_{-\infty}^{\infty} \delta_{\epsilon}(t) dt = 1$$

Plotted out with different values of ϵ , the integral of $\delta_{\epsilon}(t)$ has the shape in Figure 3.8.

Compared to Figure 3.6, we have a new, smoother approximation of the unit step function. This new $\mu_{\epsilon}(t)$, however, is twice differentiable with $\dot{\mu}_{\epsilon}(t) = \delta_{\epsilon}(t)$ and the derivative of $\delta_{\epsilon}(t)$ is well defined, satisfying

$$\frac{d\delta_{\epsilon}}{dt} = \begin{cases} 0 & \text{for } t < 0, \\ \frac{1}{\epsilon^2} & \text{for } 0 < t < \epsilon, \\ -\frac{1}{\epsilon^2} & \text{for } \epsilon < t < 2\epsilon, \\ 0 & \text{for } t > 2\epsilon. \end{cases}$$

You can verify that for any continuous function f, we also have

$$\lim_{\epsilon \to 0} \int_{-\infty}^{\infty} f(t) \delta_{\epsilon}(t) dt = f(0),$$



Figure 3.8: Toward smoother approximations of the unit-step function.

and for any differentiable function f, we have

$$\lim_{\epsilon \to 0} \int_{-\infty}^{\infty} f(t) \frac{d\delta_{\epsilon}}{dt} dt = -\dot{f}(0)$$

In the limit, we get an even more singular function, $\dot{\delta}$. Beyond Equation 3.6 and Equation 3.9, we can have more smoother approximations to the impulse, which can have valid higher-order derivatives $\ddot{\delta}$, $\delta^{(3)}$, $\delta^{(4)}$,

3.1.2 Relevant Properties of the Laplace Transofrm

In this section, we review a few important properties of the Laplace transform.

The Laplace operator is a linear operator and satisfies the superposition principle. For any $\alpha, \beta \in \mathbb{C}$ and functions f(t), g(t), let $F(s) = \mathcal{L}{f(t)}$, $G(s) = \mathcal{L}{g(t)}$. Then

$$\mathcal{L}\{\alpha f(t)+\beta g(t)\}=\alpha F(s)+\beta G(s).$$

A pivotal property of the Laplace transform is that the *derivative* operator (d/dt) in the time domain corresponds to multiplication by *s* in the Laplace *s* domain. Let $\dot{f}(t) = \frac{df(t)}{dt}$, then

$$\mathcal{L}\{\dot{f}(t)\}=sF(s)-f(0).$$

This can be verified via integration by parts:

$$\begin{aligned} \mathscr{L}\{\dot{f}(t)\} &= \int_{0}^{\infty} e^{-st} \dot{f}(t) dt \\ &= -\int_{0}^{\infty} \frac{de^{-st}}{dt} f(t) dt + \left\{ e^{-st} f(t) \right\}_{t=0}^{t \to \infty} \\ &= s \int_{0}^{\infty} e^{-st} f(t) dt - f(0) = sF(s) - f(0). \end{aligned}$$
(3.10)

Also, the *integral* operator in the time domain corresponds to division by *s* or multiplication by $\frac{1}{s}$:

$$\mathscr{L}\left\{\int_{0}^{t} f(\tau)d\tau\right\} = \int_{0}^{\infty} \int_{0}^{t} f(\tau)d\tau e^{-st}dt$$
$$= \frac{1}{-s} \int_{0}^{t} f(\tau)d\tau e^{-st} \bigg|_{0}^{\infty} - \frac{1}{-s} \int_{0}^{\infty} f(t)e^{-st}dt \qquad (3.11)$$
$$= \frac{1}{s}F(s).$$

Using the definition, we have the following additional useful properties.

Multiplication by e^{-at}

Multiplication in the time domain by an exponential function, in other words, adding a time-domain decay, corresponds simply to a translation in the *s* domain:

$$\mathscr{L}\left\{e^{-at}f(t)\right\} = F(s+a).$$

Example 3.1.1 With $\mathscr{L}{1(t)} = \frac{1}{s}$, we can use the time-domain multiplication property to verify that $\mathscr{L}{e^{-at}} = \frac{1}{s+a}$. Also, given $\mathscr{L}{\sin(\omega t)} = \frac{\omega}{s^2+\omega^2}$, it is immediate that $\mathscr{L}{e^{-at}\sin(\omega t)} = \frac{\omega}{(s+a)^2+\omega^2}$.

Multiplication by t

Notice that $\mathscr{L}{1(t)} = \frac{1}{s}$ and $\mathscr{L}{t} = \frac{1}{s^2}$. Multiplication by *t* in the time domain corresponds to derivative in the *s* domain:

$$\mathcal{L}\left\{tf(t)\right\} = -\frac{dF(s)}{ds}.$$

Time delay

A time delay, on the other hand, corresponds to multiplication by the exponential function in the *s* domain: $\mathscr{L}\left\{f(t-\tau)\right\} = e^{-s\tau}F(s)$. Hence, for a shifted impulse, we have the following Laplace transform:

```
% laplaceZtransforms/laplaceTimeDelay.m
syms t
d = dirac(t-4);
D = laplace(d)
D =
exp(-4*s)
```

In Python, the same result can be realized by:

```
# laplaceZtransforms/laplaceTimeDelay.py
import sympy
t, s = sympy.symbols('t, s')
d = sympy.DiracDelta(t-4)
D = sympy.laplace_transform(d, t, s, noconds=True)
print(D)
```

Convolution

Given
$$f_1(t)$$
, $f_2(t)$, and $(f_1 \star f_2)(t) = \int_0^t f_1(t-\tau)f_2(\tau)d\tau = (f_1 \star f_2)(t)$, then
 $\mathscr{L} \{f_1(t) \star f_2(t)\} = F_1(s)f_2(s).$

To see this, notice that the step function $1(t - \tau)$ is 1 if $0 < \tau < t$ and zero if $\tau > t$. Hence

$$\begin{aligned} \int_{t=0}^{\infty} \int_{\tau=0}^{t} e^{-st} f_{1}(t-\tau) f_{2}(\tau) d\tau dt \\ &= \int_{t=0}^{\infty} \int_{\tau=0}^{\infty} e^{-st} f_{1}(t-\tau) 1(t-\tau) f_{2}(\tau) d\tau dt \\ &= \int_{t=0}^{\infty} \int_{\tau=0}^{\infty} e^{-s(t-\tau)} e^{-s\tau} f_{1}(t-\tau) 1(t-\tau) f_{2}(\tau) d\tau dt \\ &= \int_{\tau=0}^{\infty} \int_{t=0}^{\infty} e^{-s(t-\tau)} f_{1}(t-\tau) 1(t-\tau) e^{-s\tau} f_{2}(\tau) d\tau dt \\ &= \int_{\tau=0}^{\infty} e^{-s\tau} f_{2}(\tau) \int_{t=0}^{\infty} e^{-s(t-\tau)} f_{1}(t-\tau) 1(t-\tau) dt d\tau \end{aligned}$$
(3.12)
$$&= \int_{\tau=0}^{\infty} e^{-s\tau} f_{2}(\tau) \left[\int_{t=0}^{\infty} e^{-s(t-\tau)} f_{1}(t-\tau) 1(t-\tau) dt \right] d\tau \\ &= \int_{\tau=0}^{\infty} e^{-s\tau} f_{2}(\tau) F_{1}(s) d\tau \\ &= F_{1}(s) \int_{\tau=0}^{\infty} e^{-s\tau} f_{2}(\tau) d\tau \end{aligned}$$

Initial Value Theorem

If
$$f(0_+) = \lim_{t \to 0_+} f(t)$$
 exists, then $f(0_+) = \lim_{s \to \infty} sF(s)$.

Final Value Theorem

If $\lim_{t\to\infty} f(t)$ exists, then $\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$. Using the above properties, along with the Laplace-transform results for fundamental signals such as the exponential functions, we have the common Laplace transforms in Table 3.1.

f(t)	F(s)	f(t)	F(s)
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$	e^{-at}	$\frac{1}{s_1 + a}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$	t	$\frac{1}{s^2}$
tx(t)	$-\frac{dX(s)}{ds}$	t^2	$\frac{2}{s^3}$
$\frac{x(t)}{t}$	$\int_{s}^{\infty} X(s) ds$	te ^{-at}	$\frac{1}{\left(s+a\right)^2}$
$\delta\left(t\right)$	1	$e^{-at}\sin(\omega t)$	$\frac{(c + a)^2}{(c + a)^2 + \omega^2}$
1(t)	$\frac{1}{s}$	$e^{-at}\cos(\omega t)$	$\frac{(s+a)^2 + \omega^2}{(s+a)^2 + \omega^2}$

Table 3.1: Common Laplace transforms.

3.2 Inverse Laplace Transform and Partial Fraction Expansion

The inverse Laplace transform of F(s) is defined by the contour integral

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \int_{c-j\infty}^{c+j\infty} F(s)e^{st}ds, \ t \ge 0.$$
(3.13)

This integral is challenging to calculate. Usually, we do not use the definition directly. Instead, we break a large Laplace transform into small blocks by partial fraction expansion, also known as partial fraction decomposition:

$$F(s) = \frac{B(s)}{A(s)} = \frac{B_1(s)}{A_1(s)} + \frac{B_2(s)}{A_2(s)} + \dots$$

and then identify the individual inverse Laplace transforms on the righthand side by observation or a lookup table such as Table 3.1.

We use a few examples to demonstrate strategies for common partial fraction expansions.

Example 3.2.1 (Real and Distinct Roots in the Denominator) Consider

$$F(s) = \frac{B(s)}{A(s)} = \frac{32}{s(s+4)(s+8)}$$

A(s) has three distinct roots: 0, -4, and -8. The partial fraction expansion is

$$\frac{32}{s(s+4)(s+8)} = \frac{K_1}{s} + \frac{K_2}{s+4} + \frac{K_3}{s+8},$$

where the residues can be calculated by:

- $K_1 = \lim_{s \to 0} sF(s) = 1$,
- $K_2 = \lim_{s \to -4} (s+4)F(s) = -2$,
- $K_3 = \lim_{s \to -8} (s+8)F(s) = 1.$

The results can be verified in MATLAB and Python as follows:

```
% laplaceZtransforms/partial_fraction_expansion.m
syms s
G = 32/s/(s+4)/(s+8)
partfrac(G)

# laplaceZtransforms/partial_fraction_expansion.py
import sympy
s = sympy.symbols('s')
G = 32/s/(s+4)/(s+8)
print(sympy.apart(G))
```

1/(s + 8) - 2/(s + 4) + 1/s

Example 3.2.2 (Real and Repeated Roots in A(s)) Consider

$$F(s) = \frac{2}{(s+1)(s+2)^2}$$

The partial fraction expansion is

$$\frac{2}{(s+1)(s+2)^2} = \frac{K_1}{s+1} + \frac{K_2}{s+2} + \frac{K_3}{(s+2)^2}.$$
 (3.14)

Here, the residues for the single roots are calculated by

- $K_3 = \lim_{s \to -2} (s+2)^2 F(s) = -2$, and
- $K_1 = \lim_{s \to -1} (s+1)F(s) = 2.$

For K_2 , we multiply both sides of Equation 3.14 with $(s + 2)^2$ and differentiate once with respect to *s*, to get

$$K_2 = \lim_{s \to -2} \frac{d}{ds} (s+2)^2 F(s) = -2.$$

With the Laplace transform, the calculus operators in the time domain are replaced by algebraic operations. For example, consider the following first-order system

$$\dot{y}(t) = -ay(t) + b1(t),$$

where $a > 0, b > 0, y(0) = y_0 \in \mathbb{R}$. Laplace transform gives $\mathcal{L}{\dot{y}(t)} = sY(s) - y(0)$. Thus,

$$Y(s) = \frac{1}{s+a}y(0) + \frac{b}{s(s+a)}$$

and by partial fraction expansion,

$$Y(s) = \frac{1}{s+a}y(0) + \frac{b}{a}\left(\frac{1}{s} - \frac{1}{s+a}\right).$$

The first-order terms on the right-hand side are readily relatable to their time-domain functions. By the inverse transform, we have

$$y(t) = e^{-at}y(0) + \frac{b}{a}(1(t) - e^{-at}).$$

From the ODE, we observe that $y(\infty) = \frac{b}{a}$. So in the end, the system has scaled the step input by a factor of b/a.⁴

Example 3.2.3 Let $a > 0, b > 0, y(0) = y_0 \in \mathbb{R}$. Obtain the solution to the ODE: $\dot{y}(t) = -ay(t) + b\delta(t)$. Solution: Applying Laplace transform yields $\mathcal{L}{\dot{y}(t)} = sY(s) - y_0 = -aY(s) + b$. Solving for Y(s), we have $Y(s) = \frac{1}{s+a}(y_0 + b)$. Hence, $y(t) = \mathcal{L}^{-1}{Y(s)} = e^{-at}(y_0 + b)$.⁵ 4: On the other hand, knowing that a final value exists and using the Final Value theorem, we have: $\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \frac{b}{a}$, which matches the result from the direct ODE solution.

5: What's the initial value from initial value theorem? What does the impulse do to the initial condition?

3.3 From Laplace Transform to Transfer Functions

Consider an N-th order differential equation

$$\frac{d^{n}y}{dt^{n}} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \dots + a_{1}\dot{y} + a_{0}y = b_{m}\frac{d^{m}u}{dt^{m}} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \dots + b_{1}\dot{u} + b_{o}u,$$
(3.15)

where y(0) = 0, $\frac{dy}{dt}\Big|_{t=0} = 0$, ..., $\frac{d^{n-1}y}{dt^{n-1}}\Big|_{t=0} = 0$. Applying Laplace transform yields

$$(s^{n} + a_{n-1}s^{n-1} + \dots + a_{0})Y(s) = (b_{m}s^{m} + b_{m-1}s^{m-1} + \dots + b_{0})U(s),$$

and hence

$$Y(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_0} U(s).$$

 $G(s) \triangleq \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_0}$ relates the input u(t) to the output y(t) in the Laplace domain and is known as the transfer function. The Laplace transform of the output signal is the product of the transfer function and the Laplace transform of the input signal.

A(s) = 0 is called the characteristic equation, and its roots are the *poles* of G(s). Notice that $|G(s)| = \infty$ at every pole. Roots of B(s) = 0 are the *zeros* of G(s), and |G(s)| = 0 at every zero. G(s) is called *proper* if the order of the denominator n is no less than the order of the numerator m, and is called *strictly proper* if n > m. The condition that $m \le n$ is called the realizability condition.

Example 3.3.1 $G_1(s) = K$ is proper. $G_2(s) = \frac{k}{s+a}$ is strictly proper.

The following codes demonstrate the construction and basic analysis of a transfer function in MATLAB and Python.

```
% laplaceZtransforms/transfer_fun.m
num = [1 2];
den = [1 2 3];
```

```
sys_tf = tf(num,den)
poles = pole(sys_tf);
zeros = zero(sys_tf);
disp(['System Poles = ',num2str(poles')])
disp(['System Zeros = ',num2str(zeros')])
[yout, T] = step(sys_tf);
figure, plot(T, yout)
figure, impulse(sys_tf)
u1 = 2*ones(length(T),1);
u2 = sin(T);
figure, lsim(sys_tf,u1,T)
figure, lsim(sys_tf,u2,T)
```

```
# laplaceZtransforms/transfer_fun.py
import control as ct
import matplotlib.pyplot as plt
import numpy as np
# Creating a transfer function system
num = [1,2] # Numerator co-efficients
den = [1,2,3] # Denominator co-efficients
sys tf = ct.tf(num,den)
print(sys_tf)
# Poles and zeros
poles = ct.pole(sys_tf)
zeros = ct.zero(sys_tf)
print('\nSystem Poles = ', poles, '\nSystem Zeros = ', zeros)
T,yout = ct.step_response(sys_tf)
# Plot the response
plt.figure(1,figsize = (6,4))
plt.plot(T,yout)
plt.grid(True)
plt.ylabel("y")
plt.xlabel("Time (sec)")
plt.show()
T,yout_i = ct.impulse_response(sys_tf)
# Plot the response
plt.figure(1,figsize = (6,4))
plt.plot(T,yout_i)
plt.grid(True)
plt.ylabel("y")
plt.xlabel("Time (sec)")
plt.show()
u1 = np.full((1,len(T)),2) #Create an array of 2's, equal to 2*step
u^2 = np.sin(T)
T,yout_u1 = ct.forced_response(sys_tf,T,u1) # Response to input 1
T,yout_u2 = ct.forced_response(sys_tf,T,u2) # Response to input 2
plt.figure(2,figsize = (6,4))
plt.plot(T,yout_u1)
plt.plot(T,yout_u2)
```

plt.grid()
plt.xlabel("Time (sec)")
plt.ylabel("y")
plt.legend(["Input 1","Input 2 (sin)"])
plt.show()

Example 3.3.2 (AFM System) For the AFM system in Section 2.4, assuming zero initial conditions and taking the Laplace transform of Equation 2.3 yield

$$X_{1}(s) = \frac{1}{m_{1}s^{2}}F(s),$$

$$X_{2}(s) = \frac{-1}{m_{2}s^{2} + b_{2}s + k_{2}}F(s),$$

$$L(s) = x_{1}(s) - x_{2}(s) = \frac{(m_{2} + m_{1})s^{2} + b_{2}s + k_{2}}{m_{1}s^{2}(m_{2}s^{2} + b_{2}s + k_{2})}.$$
(3.16)

Eliminating F(s) and $X_2(s)$ from the above equations, we can get the transfer function from L(s) to $X_1(s)$ in the second-order model:

$$G_1(s) = \frac{X_1(s)}{L(s)} = \frac{m_2 s^2 + b_2 s + k_2}{(m_2 + m_1)s^2 + b_2 s + k_2}.$$
(3.17)

Let $\omega_0^2 = k_2/(m_1 + m_2)$, $\zeta = b_2/(2\sqrt{k_2(m_1 + m_2)})$, and $\alpha = m_2/(m_1 + m_2)$. We can write the above transfer function in terms of the resonant frequency ω_0 , damping ratio ζ , and mass ratio α :

$$G_1(s) = \frac{\alpha s^2 + 2\zeta \omega_0 s + \omega_0^2}{s^2 + 2\zeta \omega_0 s + \omega_0^2}.$$
(3.18)

For the fourth-order system model, taking the Laplace transform of Equation 2.4 and assuming zero initial condition yield

$$X_1(s)(m_1s^2 + b_1s + k_1) - X_2(b_1s + k_1) = F(s),$$

$$X_2(s)(m_2s^2 + (b_1 + b_2)s + (k_1 + k_2)) - X_1(s)(b_1s + k_1) = -F(s).$$
(3.19)

Eliminating $X_2(s)$ and simplifying, we can get

$$G_2(s) = \frac{X_1(s)}{F(s)} = \frac{s^2 + 2\zeta_2\omega_2 s + \omega_2^2}{m_1(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)(s^2 + 2\zeta_3\omega_3 s + \omega_3^2)},$$
 (3.20)

where [1]

$$2(\zeta_1\omega_1 + \zeta_3\omega_3) = \frac{m_1(b_1 + b_2) + m_2b_1}{m_1m_2},$$

$$\omega_1^2 + 4\zeta_1\zeta_3\omega_1\omega_3 + \omega_3^2 = \frac{m_1(k_1 + k_2) + m_2k_1 + b_1b_2}{m_1m_2},$$

$$2(\zeta_1\omega_1\omega_3^2 + \omega_1^2\zeta_3\omega_3) = \frac{b_1k_2 + b_2k_1}{m_1m_2},$$

$$\omega_1^2\omega_3^2 = \frac{k_1k_2}{m_1m_2}.$$

[1]: Schitter et al. (2007), *Design and Modeling of a High-Speed AFM-Scanner* Usually, the mass m_1 is smaller than m_2 and the spring constant k_1 is much larger than k_2 , which give the following frequencies:

$$_{1} \qquad \approx \sqrt{\frac{k_{2}}{m_{1}+m_{2}}}, \tag{3.21}$$

$$\omega_2 \qquad = \sqrt{\frac{k_2}{m_2}},\tag{3.22}$$

$$\approx \sqrt{\frac{(m_1+m_2)k_1}{m_1m_2}},$$
 (3.23)

where ω_1 is approximately the same as ω_0 .

ω

 ω_3

When the poles of a transfer function G(s) are on the left half of the complex plane, the system is stable. Then we can evaluate the steady-state gain, also known as the DC gain – the ratio of the output to a constant input after all transients have decayed. Two methods exist. First, by definition, u and yare all constant at the steady state; all the derivative terms in Equation 3.15 are thus zero. ⁶ Let the steady-state values of u and y be, respectively, u_{ss} and y_{ss} . Then Equation 3.15 gives,

$$a_0 y_{ss} = b_0 u_{ss} \Longrightarrow y_{ss} = \frac{b_0}{a_0} u_{ss}.$$

The DC gain is thus b_0/a_0 . We can also use the Final Value Theorem to find the DC gain of a system. Let the input be a unit step, namely, 1/s in the Laplace domain. Then by definition, the final value of a stable system's output will equal the DC gain. We have

DC gain of
$$G(s) = \lim_{s \to 0} sY(s) = \lim_{s \to 0} sG(s)\frac{1}{s} = \lim_{s \to 0} G(s) = \frac{b_0}{a_0}$$
.

In MATLAB and Python, the following codes demonstrate the computation of the DC gain:

```
% laplaceZtransforms/CTDCgain.m
s = tf('s');
G = (2*s+3)/(4*s^2+3*s+1);
dcgain(G)
# laplaceZtransforms/CTDCgain.py
import control as ct
s = ct.tf('s')
G = (2*s+3)/(4*s**2+3*s+1);
print(ct.dcgain(G))
```

The result is as follows, which is precisely the value of $G(s)|_{s=0}$.

3.0

Note that the DC gain is well defined for stable systems. The following codes will tell that the DC gain of $\frac{3}{s-2}$ is -1.5. However, when plotting the step response, it is observed that the output does not even converge!

```
% laplaceZtransforms/CTDCgain_caution.m
H = tf([0 3],[1 -2])
dcgain(H)
figure, step(H)
```

6: Equation 3.15 restated for reference:

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \dot{y} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \dots + b_1 \dot{u} + b_o u,$$

```
# laplaceZtransforms/CTDCgain_caution.py
import control as ct
H = ct.tf([0, 3],[1, -2])
print(ct.dcgain(H))
T, yout = ct.step_response(H)
print(yout)
```

3.4 The Z Transform

The Z transformation is a powerful tool to solve a wide variety of Ordinary difference Equations (OdEs). In control engineering, the Z transformation is used for discrete-time sequences and is analogous to Laplace transform for continuous-time signals.

3.4.1 Definition

Let f(k) be a real discrete-time sequence that is zero if k < 0. The (one-sided) Z transform of f(k) is

$$F(z) \triangleq \mathscr{Z}{f(k)} = \sum_{k=0}^{\infty} f(k)z^{-k}$$

= $f(0) + f(1)z^{-1} + f(2)z^{-2} + \dots,$ (3.24)

where z is a complex variable and must be such that the summation on the right-hand side converges. ⁷

Recall from calculus that:

- ► the infinite series $1 + \gamma + \gamma^2 + ...$ converge to $\frac{1}{1-\gamma}$ for $|\gamma| < 1$, ⁸ and
- the finite series $1 + \gamma + \gamma^2 + \dots + \gamma^N$ equals $\sum_{k=0}^N \gamma^k = \frac{1 \gamma^{N+1}}{1 \gamma}$ if $\gamma \neq 1$.

From the above, we can evaluate the Z transform of many discrete-time sequences.

Example 3.4.1 Consider the Z transform of the geometric sequence $\{a^k\}_{k=0}^{\infty}$. We have

$$\mathfrak{L}\{a^k\} = \sum_{k=0}^{\infty} a^k z^{-k} = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$$

for $|az^{-1}| < 1$ (|z| > |a|).

In MATLAB, the Z transform can be obtained via the following codes:

```
% laplaceZtransforms/simpleZtransform.m
syms a k;
f = a^k;
F = ztrans(f)
```

The result is as follows:

7: The Z transform is also a linear operator. By using the definition, you should be able to show that given real numbers α and β , $\mathcal{X} \{ \alpha f(k) + \beta g(k) \} = \alpha \mathcal{X} \{ f(k) \} + \beta \mathcal{X} \{ g(k) \}.$

8: |x| < 1 is called the region of convergence (ROC).

F = -z/(a - z) An equivalent Python implementation is: # laplaceZtransforms/simpleZtransform.py import lcapy as lc from lcapy.discretetime import n import sympy a = sympy.symbols('a') f = a**n F = f.ZT() print(F)

which yields the same Z transform:

z/(-a + z)

Example 3.4.2 Consider the step sequence (discrete-time unit step function):

$$1(k) = \begin{cases} 1, & \forall k = 1, 2, \dots \\ 0, & \forall k = \dots, -1, 0 \end{cases}$$

Its Z transform is $\mathfrak{X}{1(k)} = \mathfrak{X}{a^k}\Big|_{a=1} = \frac{1}{1-z^{-1}} = \frac{z}{z-1}$ for |z| > 1.

Example 3.4.3 The Z transform of the discrete-time impulse

$$\delta(k) = \begin{cases} 1, & k = 0, \\ 0, & \text{otherwise.} \end{cases}$$

is $\mathscr{Z}{\delta(k)} = 1 + 0 \cdot z^{-1} + 0 \cdot z^{-2} + \dots = 1.$

Example 3.4.4 For $f(k) = e^{j\omega k}$, the Z transform is

$$F(z) = \sum_{k=0}^{\infty} e^{j\omega k} z^{-k} = \frac{1}{1 - e^{j\omega} z^{-1}}$$
$$= \frac{1}{1 - (\cos \omega z^{-1}) - j (\sin \omega z^{-1})}$$
$$= \frac{1 - (\cos \omega z^{-1}) + j (\sin \omega z^{-1})}{1 - 2(\cos \omega) z^{-1} + z^{-2}}$$
$$= \frac{z(z - \cos \omega + j \sin \omega)}{z^2 - 2(\cos \omega) z + 1}.$$

Noting that $e^{j\omega k} = \cos \omega k + j \sin \omega k$, from the Z transform of $e^{j\omega k}$, we have

f(k)	F(z)
$\sin \omega k$	$\frac{z\sin\omega}{z^2-2(\cos\omega)z+1}$
$\cos \omega k$	$\frac{z(z-\cos\omega)}{z^2-2(\cos\omega)z+1}$

Example 3.4.5 For a periodic sequence, f(k + N) = f(k), where *N* is the period, we have

$$\begin{split} F(z) &= f(0) + f(1)z^{-1} \dots f(N-1)z^{-(N-1)} \\ &+ f(N)z^{-N} + f(N+1)z^{-(N+1)} + \dots \\ &= f(0) + f(1)z^{-1} \dots f(N-1)z^{-(N-1)} \\ &+ f(0)z^{-N} + f(1)z^{-(N+1)} + \dots \\ &= f(0)(1+z^{-N}+z^{-2N}+\dots) \\ &+ f(1)z^{-1}(1+z^{-N}+z^{-2N}+\dots) \\ &+ f(N-1)z^{-(N-1)}(1+z^{-N}+z^{-2N}+\dots) \\ &= \frac{1}{1-z^{-N}} \left[f(0) + f(1)z^{-1} + \dots + f(N-1)z^{-(N-1)} \right]. \end{split}$$

3.4.2 Relevant Properties

Similar to the Laplace transform, the Z transform has nice properties that provide conveniences in system analysis. Let $\mathfrak{L}{f(k)} = F(z)$ and f(k) = 0 $\forall k < 0$. We have

1. Time-Shift Theorem: When the signal has a one-step delay, we have

$$\begin{split} &\mathcal{L}\{f(k-1)\} \\ &= \sum_{k=0}^{\infty} f(k-1) z^{-k} = \left(\sum_{k=1}^{\infty} f(k-1) z^{-k}\right) + f(-1) \\ &= \left(\sum_{k=1}^{\infty} f(k-1) z^{-(k-1)} z^{-1}\right) + f(-1) \\ &= z^{-1} F(z) + f(-1) = \boxed{z^{-1} F(z)}, \end{split}$$

where f(-1) = 0 as f(k) was assumed causal. Analogously, if the signal has a one-step advance, then

$$\underbrace{\mathfrak{Z}\{f(k+1)\}}_{k=0} = \sum_{k=0}^{\infty} f(k+1)z^{-k} = zF(z) - zf(0)$$

More generally, you should be able to derive that

$$\begin{aligned} & \mathcal{Z}\{f(k-i)\} = z^{-i}F(z), \quad i > 0, \\ & \mathcal{Z}\{f(k+i)\} = z^{i}F(z) - \sum_{j=1}^{i} z^{j}f(i-j), \quad i > 0. \end{aligned} \tag{3.25}$$

2. Convolution: Let $F_1(z)$ and $F_2(z)$ be the Z transforms of $f_1(k)$ and

48 3 Laplace and Z Transforms

 $f_2(k)$, respectively. Then

$$F_{1}(z)F_{2}(z) = \left\{\sum_{i=0}^{\infty} f_{1}(i)z^{-i}\right\} \left\{\sum_{j=0}^{\infty} f_{2}(j)z^{-j}\right\}$$
(3.26)
$$= \sum_{k=0}^{\infty} \left\{\sum_{i=0}^{k} f_{1}(k-i)f_{2}(i)\right\} z^{-k} = \mathcal{E}\left\{f_{1}(k) * f_{2}(k)\right\},$$

where * denotes the convolution sum in the discrete-time domain. 3. Initial Value Theorem:

$$f(0) = \lim_{z \to \infty} F(z). \tag{3.27}$$

4. Final Value Theorem: If $\lim_{k\to\infty} f(k)$ exists, then

$$\lim_{k \to \infty} f(k) = \lim_{z \to 1} (z - 1)F(z).$$
(3.28)

To see the result, notice that if $\lim_{k\to\infty} f(k) = f_{\infty}$ exists, then

$$\sum_{k=0}^{\infty} \left[f(k+1) - f(k) \right] = f_{\infty} - f(0).$$

We have, on the one hand,

$$\lim_{z \to 1} \sum_{k=0}^{\infty} z^{-k} \left[f\left(k+1\right) - f\left(k\right) \right] = \sum_{k=0}^{\infty} \left[f\left(k+1\right) - f\left(k\right) \right],$$

and on the other hand,

$$\sum_{k=0}^{\infty} z^{-k} \left[f(k+1) - f(k) \right] = zF(z) - zf(0) - F(z).$$

Thus $f_{\infty} - f(0) = \lim_{z \to 1} [zF(z) - zf(0) - F(z)]$, which gives Equation 3.28.

5. Z-domain scaling:

$$\mathcal{\mathcal{X}}\left\{a^{k}f\left(k\right)\right\}=F\left(a^{-1}z\right).$$

You should be able to prove the fact by definition.

6. Differentiation:

$$\mathscr{Z}\left\{kf\left(k\right)\right\} = -z\frac{dF\left(z\right)}{dz}.$$

For example, the time-scaled geometric sequence $k(0.5)^k$ can be obtained by $-z \frac{d\mathcal{I}\{0.5^k\}}{dz} = -z \frac{d}{dz} \frac{z}{z-0.5} = \frac{0.5z}{(z-0.5)^2}$, as verified below:

```
% laplaceZtransforms/timescaledGeometricSeqZ.m
syms k;
f = 0.5^k;
F = ztrans(f)
f1 = k*f;
F1 = ztrans(f1)
```

which yield

F = z/(z - 1/2)F1 = $(2*z)/(2*z - 1)^2$

Equivalent results can be obtained in Python:

```
# laplaceZtransforms/timescaledGeometricSeqZ.py
import lcapy as lc
from lcapy.discretetime import n
f=0.5**n
print(f)
F = f.ZT()
print(F)
f1 = n*f
print(f1)
F1 = f1.ZT()
print(F1)
```

which give

(1/2)**n 2*z/(2*z - 1) n/2**n 2*z/(2*z - 1)**2

7. Time reversal:

 $\mathcal{Z}\left\{f\left(-k\right)\right\}=F\left(z^{-1}\right).$

Table 3.2 below summarizes the common Z transform properties and example applications. As we observe, Laplace and Z transforms share many styles of properties such as initial and final value theorems. Table 3.3 provides the common Z and Laplace transforms in a unified view.

 Table 3.2: Common properties of the Z transform.

(1)	\sim ((1))
x(k)	$\mathcal{I}\left\{x\left(k\right)\right\}$
ax(k)	aX(z)
$ax_{1}\left(k\right)+bx_{2}\left(k\right)$	$aX_{1}(z) + bX_{2}(z)$
x(k+1)	$zX\left(z\right)-zx\left(0\right)$
x(k+2)	$z^{2}X(z) - z^{2}(0) - zx(1)$
x(n+k)	$z^{k}X(z) - z^{k}x(0) - z^{k-1}x(1) - \dots$
	-zx(k-1)
x(n-k)	$z^{-k}X(z)$
kx(k)	$-z\frac{d}{dz}X(z)$
$e^{-ak}x(k)$	$X(\tilde{ze^a})$
$a^{k}x(k)$	$X\left(\frac{z}{a}\right)$
$ka^{k}x(k)$	$-z \frac{d}{dz} X \left(\frac{z}{a}\right)$
$x\left(0 ight)$	$\lim_{z\to\infty} X(z)$ if the limit exists
$x(\infty)$	$\lim_{z \to 1} \left[(1 - z^{-1}) X(z) \right]$
	if $(1 - z^{-1}) X(z)$ is analytic
	on and outside the unit circle
$x\left(k\right)-x\left(k-1\right)$	$(1-z^{-1})X(z)$
$x\left(k+1\right)-x\left(k\right)$	(z-1)X(z) - zx(0)
$\sum_{k=0}^{n} x(k)$	$\frac{1}{1-z^{-1}}X(z)$
$k^m x(k)$	$\left(-z\frac{d}{dz}\right)^m X(z)$
$\sum_{k=0}^{n} x(k) y(n-k)$	X(z)Y(z)
$\sum_{k=0}^{\infty} x(k)$	$X\left(1 ight)$

X(s)	x(t)	x(kT) or $x(k)$	X(z)
		$\delta(k)$	1
		$\delta(k-k_0)$	z^{-k_0}
$\frac{1}{s}$	1(t)	1(k)	$\frac{1}{1-z^{-1}}$
$\frac{1}{s+a}$	e^{-at}	e^{-akT}	$\frac{1}{1-e^{-aT}z^{-1}}$
$\frac{1}{s^2}$	t	kT	$\frac{Tz^{-1}}{(1-z^{-1})^2}$
$\frac{2}{s^3}$	t^2	$(kT)^2$	$\frac{\dot{T}^2 z^{-1} (1+z^{-1})}{(1-z^{-1})^3}$
$\frac{6}{s^4}$	t^3	$(kT)^{3}$	$\frac{T^{3}z^{-1}(1+4z^{-1}+z^{-2})}{(1-z^{-1})^{4}}$
$\frac{a}{s(s+a)}$	$1 - e^{-at}$	$1 - e^{-akT}$	$\frac{(1-e^{-aT})'z^{-1}}{(1-z^{-1})(1-e^{-aT}z^{-1})}$
$\frac{b-a}{(s+a)(s+b)}$	$e^{-at} - e^{-bt}$	$e^{-akT} - e^{-bkT}$	$\frac{(1-2)(1-e^{-bT})z^{-1}}{(1-e^{-aT}z^{-1})(1-e^{-bT}z^{-1})}$
$\frac{1}{(s+a)^2}$	te ^{-at}	kTe^{-akT}	$\frac{\frac{(1-e^{-aT}z^{-1})}{(1-e^{-aT}z^{-1})^2}}{(1-e^{-aT}z^{-1})^2}$
$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$	$(1-akT)e^{-akT}$	$\frac{1 - (1 + aT)e^{-aT}z^{-1}}{\left(1 - e^{-aT}z^{-1}\right)^2}$
$\frac{2}{(s+a)^3}$	$t^2 e^{-at}$	$(kT)^2 e^{-akT}$	$\frac{T^2 e^{-aT} (1 + e^{-aT} z^{-1}) z^{-1}}{(1 - e^{-aT} z^{-1})^3}$
$\frac{a^2}{s^2(s+a)}$	$at - 1 + e^{-at}$	$akT - 1 + e^{-akT}$	$\frac{\left[\left(aT-1+e^{-aT}\right)+\left(1-e^{-aT}-aTe^{-aT}\right)z^{-1}\right]z^{-1}}{(1-z^{-1})^2(1-e^{-aT}z^{-1})}$
$\frac{\omega}{s^2 + \omega^2}$	$\sin \omega t$	$\sin \omega kT$	$\frac{z^{-1}\sin\omega T}{1-2z^{-1}\cos\omega T+z^{-2}}$
$\frac{s}{s^2 + c^2}$	$\cos \omega t$	$\cos \omega kT$	$\frac{1-22}{1-2z^{-1}\cos\omega T}$
$\frac{\omega}{(c+a)^2+\omega^2}$	$e^{-at}\sin\omega t$	$e^{-akT}\sin\omega kT$	$\frac{1-2z}{e^{-aT}z^{-1}\sin\omega T}$ $\frac{1-2e^{-aT}z^{-1}\cos\omega T}{1-2e^{-aT}z^{-1}\cos\omega T+e^{-2aT}z^{-2}}$
$\frac{(s+a)^2}{(s+a)^2+c^2}$	$e^{-at}\cos\omega t$	$e^{-akT}\cos\omega kT$	$\frac{1 - e^{-aT} z^{-1} \cos \omega T}{1 - 2e^{-aT} z^{-1} \cos \omega T + e^{-2aT} z^{-2}}$
(5+u) +w		a ^k	$\frac{1}{1-\frac{1}{2}}$
		ka^{k-1}	$\frac{1-az^{-1}}{z^{-1}}$
			$(1-az^{-1})^2$
		$k^2 a^{k-1}$	$\frac{2(1+u^2)}{(1-az^{-1})^3}$
		$k^{3}a^{k-1}$	$\frac{z^{-1}(1+4az^{-1}+a^2z^{-2})}{(1-az^{-1})^4}$
		k^4a^{k-1}	$\frac{z^{-1}(1+11az^{-1}+11a^2z^{-2}+a^3z^{-3})}{(1-az^{-1})^5}$
		$a^k \cos k\pi$	$\frac{1}{1+az^{-1}}$ (1 uz)

Table 3.3: Table of Laplace and Z transforms [5]. x(t) = 0 for t < 0. x(kT) = x(k) = 0 for k < 0. Unless otherwise noted, k = 0, 1, 2, 3, ...

3.4.3 Applications to Dynamic Systems

With the Z transform, the calculus operators in discrete-time domain are replaced by the algebraic operation in the Z domain. For example, consider the following first-order system,

$$x(k+1) - ax(k) = bu(k), \ x(0) = 0.$$
(3.29)

Applying the Z transformation yields

$$zX(z) - zx(0) - aX(z) = bU(z) \Longrightarrow X(z) = \frac{b}{z - a}U(z), \qquad (3.30)$$

where $X(z) = \mathcal{X}(x(k))$ and $U(z) = \mathcal{X}(u(k))$. If u(k) is a unit step function u(k) = 0 for k < 0 and u(k) = 1 for $k \ge 0$, then

$$X(z) = \frac{b}{z-a} \frac{z}{z-1} = \frac{b}{1-a} \left\{ \frac{z}{z-1} - \frac{z}{z-a} \right\}.$$
 (3.31)

Now applying the inverse Z transformation yields

$$x(k) = \frac{b}{1-a} \left(1 - a^k \right).$$
 (3.32)

The development above shows that one way to obtain the solution of linear difference equations is to use the Z transformation. This is analogous to the continuous-time case where the solution of linear differential equation can be obtained by using the Laplace transformation.

Example 3.4.6 Let us use the Z transform to solve the difference equation

$$y(k) + 3y(k-1) + 2y(k-2) = u(k-2),$$
(3.33)

where y(-2) = y(-1) = 0 and u(k) = 1(k). From the Time-Shift Theorem:

$$\mathscr{Z}\{y(k-2)\} = \mathscr{Z}\{y(k-1-1)\} = z^{-1}\mathscr{Z}\{y(k-1)\} + y(-2)$$

= $z^{-1}(z^{-1}Y(z) + y(-1)) + y(-2).$ (3.34)

Taking Z transforms on both sides of Equation 3.33 and applying the initial conditions yield

$$(1+3z^{-1}+2z^{-2})Y(z) = z^{-2}U(z)$$

$$\Rightarrow Y(z) = \frac{1}{z^2+3z+2}U(z) = \frac{1}{(z+2)(z+1)}U(z).$$
 (3.35)

The step input u(k) = 1(k) has a Z transform of $U(z) = 1/(1 - z^{-1})$. Hence,

$$Y(z) = \frac{z}{(z-1)(z+2)(z+1)} = \frac{1}{6}\frac{z}{z-1} + \frac{1}{3}\frac{z}{z+2} - \frac{1}{2}\frac{z}{z+1}.$$

Inverse Z transform then gives

$$y(k) = \frac{1}{6}1(k) + \frac{1}{3}(-2)^k - \frac{1}{2}(-1)^k, \ k \ge 0.$$

Be careful with the partial fraction expansion.

Example 3.4.7 (Finance and Mortgage) As another example, difference equations rise naturally in a mortgage payment. Consider borrowing \$100,000 for a mortgage with an 4 percent annual percent interest rate (APR). If we plan to pay off in 30 years with fixed monthly payments, what will be the monthly payment?

Let *k* be the number of months in debt, and y(k) be the remaining debt to be paid. The initial debt of \$100,000 yields y(0) = 100,000. Translating the 4 percent APR to a monthly percent rate (MPR), we have $MPR = \frac{4.0\%}{12} = 0.0033$. The governing equation for the remaining debt is

$$y(k+1) = \underbrace{(1+MPR)}_{a} y(k) - \underbrace{b}_{\text{monthly payment}} 1(k).$$
(3.36)

Applying the Z transform yields

$$Y(z) = \frac{z}{z-a}y(0) - \frac{1}{z-a}\frac{b}{1-z^{-1}}$$

= $\frac{1}{1-az^{-1}}y(0) + \frac{b}{1-a}\left(\frac{1}{1-az^{-1}} - \frac{1}{1-z^{-1}}\right).$ (3.37)

Inverse Z transform now gives

$$y(k) = a^{k}y(0) + \frac{b}{1-a}\left(a^{k} - 1\right).$$
(3.38)

To pay off in 30 years, the terminal condition is y(N) = 0 where $N = 30 \times 12 = 360$ months, yielding $a^N y(0) = -\frac{b}{1-a} (a^N - 1) \Rightarrow b = \frac{a^N y(0)(a-1)}{a^N - 1} =$ \$477.42.

3.5 From Difference Equation to Discrete-Time Transfer Functions

Consider the difference equation:

$$y(k) + a_{n-1}y(k-1) + \dots + a_0y(k-n) = b_m u(k+m-n) + \dots + b_0u(k-n), \quad (3.39)$$

where u(k) is a known input sequence. Assume that $y(k) = 0 \forall k < 0$. We now generalize the concept at the start of Section 3.4.3. Applying the Z transform to each term yields

$$Y(z) + a_{n-1}z^{-1}Y(z) + \dots + a_0z^{-n}Y(z) = b_m z^{-n+m}U(z) + \dots + b_0z^{-n}U(z),$$
(3.40)

and thus,

$$Y(z) = \frac{b_m z^{-n+m} + \dots + b_0 z^{-n}}{1 + a_{n-1} z^{-1} + \dots + a_0 z^{-n}} U(z),$$
(3.41)

or equivalently,

$$Y(z) = \frac{b_m z^m + \dots + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_0} U(z).$$
 (3.42)

53

Notice that two forms (one in z^{-1} and the other in z) of the discrete-time transfer function have been given. Poles, zeros and the realizability condition $m \le n$ are similarly defined as the continuous-time case. Realizability implies that the present output depends on past and present inputs but not on future inputs.

When applying a constant input to a discrete-time system, if the output reaches a constant steady state after the transient response, the ratio between the output and the input is the DC gain of the system. At the steady state, $y(k) = y(k-1) = \cdots = y(k-n) \triangleq y_{ss}$ and $u(k+m-n) = u(k+m-n-1) = \cdots = u(k-n) \triangleq u_{ss}$. Equation 3.39 becomes

$$y_{ss} + a_{n-1}y_{ss} + \dots + a_0y_{ss} = b_m u_{ss} + \dots + b_0 u_{ss}.$$

Thus,

DC gain of
$$G_{yu}(z) = \frac{b_m + b_{m-1} + \dots + b_0}{1 + a_{n-1} + \dots + a_0}$$

which is nothing but $G_{yu}(z)|_{z=1}$.

We have the following comparison of continuous- and discrete-time transfer functions in Table 3.4.

Properties $G_{yu}(s) = \frac{B(s)}{A(s)}$ $G_{yu}(z) = \frac{B(z)}{A(z)}$ polesroots of A(s)roots of A(z)zerosroots of B(s)roots of B(z)causality condition $n \ge m$ $n \ge m$ DC gain (if exists) $G_{yu}(0)$ $G_{yu}(1)$

The following codes demonstrate the construction and basic analysis of a discrete-time transfer function in MATLAB and Python. Different from the continuous-time system description, a sampling time must now be specified to indicate the actual time difference between the discrete-time indices k and k + 1.

```
% laplaceZtransforms/transfer_fun_dt.m
num = [0.09952, -0.08144];
den = [1, -1.792, 0.8187];
Ts = 0.1; % sampling time
sys_tf = tf(num,den,Ts)
poles = pole(sys_tf);
zeros = zero(sys_tf);
disp(['System Poles = ',num2str(poles')])
disp(['System Zeros = ',num2str(zeros')])
[yout, T] = step(sys_tf);
figure, stairs(T, yout)
figure, impulse(sys_tf)
u1 = 2*ones(length(T), 1);
u2 = sin(T);
figure, lsim(sys_tf,u1,T)
figure, lsim(sys_tf,u2,T)
# laplaceZtransforms/transfer_fun_dt.py
```

```
import control as ct
import matplotlib.pyplot as plt
```

Table 3.4: Comparison of continuous- anddiscrete-time transfer functions.

```
import numpy as np
Ts = 0.1 \# sampling time
num = [0.09952, -0.08144] # Numerator co-efficients
den = [1, -1.792, 0.8187] # Denominator co-efficients
sys_tf = ct.tf(num,den, Ts)
print(sys_tf)
poles = ct.pole(sys_tf)
zeros = ct.zero(sys_tf)
print('\nSystem Poles = ', poles, '\nSystem Zeros = ', zeros)
T,yout = ct.step_response(sys_tf)
# Plot the response
plt.figure(1, figsize = (6, 4))
# instead of plt.plot(T,yout), we use step to show the discrete nature
\hookrightarrow of the response
# plt.step(T,yout)
# at the time of writing this code, there is a difference between how
\hookrightarrow MATLAB and Python make stair plots
# to correctly show the initial one-step delay, we need to append a
\, \hookrightarrow \, zero at the beginning of the output
plt.step(T,np.append(0,yout[0:-1]))
plt.grid(True)
plt.ylabel("y")
plt.xlabel("Time (sec)")
plt.show()
T,yout_i = ct.impulse_response(sys_tf)
# Plot the response
plt.figure(1, figsize = (6, 4))
plt.step(T,np.append(0,yout_i[0:-1]))
plt.grid(True)
plt.ylabel("y")
plt.xlabel("Time (sec)")
plt.show()
u1 = np.full((1,len(T)),2) #Create an array of 2's, equal to 2*step
u^2 = np.sin(T)
T,yout_u1 = ct.forced_response(sys_tf,T,u1) # Response to input 1
T,yout_u2 = ct.forced_response(sys_tf,T,u2) # Response to input 2
plt.figure(2,figsize = (6,4))
plt.step(T,np.append(0,yout_u1[0:-1]))
plt.step(T,np.append(0,yout_u2[0:-1]))
plt.grid()
plt.xlabel("Time (sec)")
plt.ylabel("y")
plt.legend(["Input 1","Input 2 (sin)"])
plt.show()
```

The first part of the results from Python are as follows. You should be able to implement the code and observe the different response figures.

```
0.09952 z - 0.08144

z^2 - 1.792 z + 0.8187

dt = 0.1

System Poles = [0.896+0.12603174j 0.896-0.12603174j]

System Zeros = [0.81832797+0.j]
```

3.6 Recap

In this chapter, we reviewed Laplace and Z transforms, along with their applications to control systems.

The Laplace transform is defined by $\mathscr{L}\left\{f\left(t\right)\right\} = \int_{0}^{\infty} e^{-st} f\left(t\right) dt$, subject to existence conditions. Based on the definition, we derived that the elemental Laplace transform pair of the exponential function: $\mathscr{L}\left\{e^{at}\right\} = \frac{1}{s-a}$. From there, using Euler formula, we can obtain the Laplace transforms of the sine and cosine functions: $\mathscr{L}\left\{\cos \omega t\right\} = \frac{s}{s^{2}+\omega^{2}}$ and $\mathscr{L}\left\{\sin \omega t\right\} = \frac{\omega}{s^{2}+\omega^{2}}$. These functions have second-order Laplace transforms because they consists of two exponential functions: e.g., $\cos \omega t = \frac{1}{2}\left(e^{j\omega t} + e^{-j\omega t}\right)$, hence the second-order nature in the *s* domain. Laplace transform has excellent properties such as the differential property:

$$\mathscr{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0),$$

the integration property:

$$\mathscr{L}\left\{\int_{0}^{t}f(\tau)\,d\tau\right\} = \mathscr{L}\left\{\mathbf{1}\left(t\right)*f\left(t\right)\right\} = \frac{F\left(s\right)}{s},$$

frequency shifting:

$$\mathscr{L}\left\{e^{at}f\left(t\right)\right\}=F\left(s-a\right),$$

time shifting:

$$\mathscr{L}\left\{f\left(t-a\right)\mathbf{1}\left(t-a\right)\right\} = e^{-as}F\left(s\right),$$

and frequency differentiation:

$$\mathscr{L}\left\{t^{n}f\left(t\right)\right\} = (-1)^{n} \, \frac{d^{n}X\left(s\right)}{ds^{n}}.$$

Now that we have learnt the convolution property, the integration property can be proven by using properties of the unit step function:

$$\int_0^t f(\tau) d\tau = \int_{-\infty}^\infty \mathbf{1} (t - \tau) f(\tau) d\tau$$

=
$$\int_0^t \mathbf{1} (t - \tau) f(\tau) d\tau,$$
 (3.43)
where we used the facts that: (i) $1(t - \tau) = 0$ if $\tau > t$, (ii) $1(t - \tau) = 1$ if $\tau \le t$, and (iii) $f(\tau) = 0$ if $\tau < 0$. The quantity on the right side of the equation is nothing but convolution of the unit step function and f(t).

The Z transform is defined by $\mathfrak{Z} \{x(k)\} = \sum_{k=0}^{\infty} x(k) z^{-k}$. By definition, it is easy to compute the Z transform of the unit step and for the geometric sequence: $\mathfrak{Z} \{1(k)\} = \frac{1}{1-z^{-1}} = \frac{z}{z-1}$, |z| < 1 and $\mathfrak{Z} \{a^k\} = \frac{1}{1-az^{-1}} = \frac{z}{z-a}$, $|az^{-1}| < 1$. The Z transform has excellent properties such as: time shifting:

$$\mathfrak{X}\left\{x\left(k-n_{d}\right)\right\}=z^{-n_{d}}X\left(z\right),$$

z-domain scaling:

$$\mathcal{\mathcal{X}}\left\{a^{k}x\left(k\right)\right\}=X\left(a^{-1}z\right),$$

and differentiation:

$$\mathscr{Z}\left\{kx\left(k\right)\right\} = -z\frac{dX\left(z\right)}{dz}.$$

With Laplace and Z transforms, we can easily obtain solutions to ordinary differential and difference equations, as well as obtain the transfer function of an LTI system. Subsequent transfer function analyses can reveal powerful characteristics of the system dynamics, such as poles and zeros, stability, and DC/steady-state gains.

3.7 Exercise

1. Consider the following two by two matrix:

$$A = \left[\begin{array}{rrr} -5 & 2\\ 2 & -2 \end{array} \right].$$

- a) Obtain the eigenvalues and eigenvectors of the matrix.
- b) Obtain the rank of the matrix analytically, and confirm your solution by using MATLAB or Python.

2. Obtain the inverses of the following matrices

$$M = \begin{bmatrix} s & -1 \\ \frac{k}{m} & s + \frac{b}{m} \end{bmatrix}, N = \begin{bmatrix} s & 6 & 0 \\ 2 & s - 1 & 0 \\ 0 & 0 & s + 1 \end{bmatrix}.$$

- 3. Find the Laplace transform of the function $f(t) = 2te^{-3t}$.
- 4. Find the inverse Laplace transform of the function $F(s) = (s+3)/(s^2+4s+5)$.
- 5. Find the Laplace transform of the function $f(t) = \sin(2t)\mathbf{1}(t)$.
- 6. Find the Laplace transform of the function $f(t) = te^{-2t} \cos(3t)$.
- 7. Find the Laplace transform of the function $f(t) = t^2$.
- 8. Find the Z-transform of the function $f(k) = 2^k 1(k)$.
- 9. Find the inverse Z-transform of the function $F(z) = (z+2)/(z^2-2z+3)$.
- 10. Find the Z-transform of the function $f(k) = k^2 (0.5)^k$.
- 11. Find the Z-transform of the function $f(k) = \cos(0.2\pi)\mathbf{1}(k)$.
- 12. Find the inverse Z-transform of the function $F(z) = (z^2 z + 1)/(z 1)^3$.

58 3 Laplace and Z Transforms

13. Let $\mathscr{L}{f(t)} = F(s)$. Show that

$$\mathcal{L}\left\{f^{(n-1)}(t)\right\} = s^{n-1}F(s) - \left[\begin{array}{ccc}s^{n-2} & s^{n-3} & \dots & 1\end{array}\right] \left[\begin{array}{c}f(0)\\\dot{f}(0)\\\vdots\\f^{(n-2)}(0)\end{array}\right].$$

- 14. Solve the ordinary differential equation $\dot{x} + 3x = 1(t)$ where x(0) = 0. 15. Show that if G(s) is strictly proper, then $G(\infty) = 0$.
- 16. Obtain the time-domain expression of the following Laplace- and Z-domain functions:

$$\frac{1}{\left(s+a\right)^{2}}, \ \frac{Tz^{-1}}{\left(a-z^{-1}\right)^{2}}.$$

- 17. Obtain f(t) if $F(s) = (s + 10) / (s^2 + 2s + 5)$.
- 18. Show that for systems with relative degree greater than or equal to zero, the impulse response always converges to 0.
- 19. Solve the ordinary differential equation $\ddot{x} + a_1\dot{x} + a_2x = 0$ where $a_1 = 2$ and $a_2 = 1$.
- 20. Obtain the Laplace or Z transforms of the following time functions.
 - a) Continuous-time periodic function: f(t) = f(t T), assuming $\{f(\tau) \mid 0 \le \tau < T\}$ is given.
 - b) Discrete-time periodic function: f(k) = f(k N), assuming $\{f(j) \mid 0 \le j < N 1\}$ is given.
- 21. The Laplace transform of f(t) is expressed as

$$F(s) = \frac{(K_1 - K_2\tau)s + (K_1 - K_2)}{s(\tau s + 1)(s + 1)}.$$

Assume $\tau > 0$ in this problem (think about what happens if this is not satisfied).

- a) Use the initial value and final value theorems to obtain the conditions so that f(t) possesses a negative initial slope (derivative) and a positive final value.
- b) Note that f(t) is regarded as the unit step response of the system described by

$$G(s) = \frac{(K_1 - K_2\tau)s + (K_1 - K_2)}{(\tau s + 1)(s + 1)}.$$

Assume the following values of the system parameters:

$$K_1 = 2, K_2 = 1, \tau = 4.$$

Obtain the time plot for f(t) by MATLAB or Python or Julia.

22. Given a Z transform

$$X(z) = \frac{z^{-1}}{(1 - z^{-1})(1 - 1.4z^{-1} + 0.48z^{-2})}$$

determine the initial and final values of x(k), the inverse Z-transform of X(z). Find x(k) in a closed form.

- 23. Let f(t) be a function that has convergent Laplace transform and $f(0^+) \neq f(0^-)$.⁹ Based on the definition $F(s) = \int_{0^-}^{\infty} f(t) e^{-st} dt$,

 - a) show that $\lim_{s\to\infty} sF(s) = f(0^+) \neq f(0^-)$ b) which one is true: $\mathscr{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) f(0^-)$ or $\mathscr{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) f(0^+)$? Justify your choice.

9: i.e., f(t) has a first-kind discontinuity at t = 0.