

# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xv</b>
<b>Preface</b>	<b>xvii</b>
<b>1 Introduction to Simulink</b>	<b>1</b>
1.1 Using a Picture to Write a Program . . . . .	1
1.2 Example 1 — Galileo Drops Two Objects from the Leaning Tower of Pisa . . . . .	5
1.3 Example 2 — Modeling a Pendulum and the Escapement of a Clock . . . . .	18
1.3.1 History of Pendulum Clocks . . . . .	18
1.3.2 A Simulation Model for the Clock . . . . .	20
1.4 Example 3: Complex Rotations – The Foucault pendulum . . . . .	24
1.4.1 Forces from Rotations . . . . .	25
1.4.2 Foucault pendulum Dynamics . . . . .	26
1.5 Further Reading . . . . .	31
<b>2 Linear Differential Equations, Matrix Algebra and Control Systems</b>	<b>33</b>
2.1 Linear Differential Equations – Linear Algebra . . . . .	33
2.1.1 Solving a Differential Equation at Discrete Time Steps . . . . .	36
2.1.2 Linear Differential Equations in Simulink . . . . .	38
2.2 Laplace Transforms for Linear Differential Equations . . . . .	40
2.3 Linear Feedback Control . . . . .	44
2.3.1 What is a Control System? . . . . .	44
2.3.2 Control Systems and Linear Differential Equations . . . . .	49
2.4 Linearization and the Control of Linear Systems . . . . .	49
2.4.1 Linearization . . . . .	50
2.4.2 Eigenvalues and the Response of a Linear System . . . . .	51
2.5 Poles and the Roots of the Characteristic Polynomial . . . . .	55
2.5.1 Feedback of the Position of the Mass in the Spring-Mass Model . . . . .	56
2.5.2 Feedback of the Velocity of the Mass in the Spring Mass Model . . . . .	57
2.5.3 Comparing Position and Rate Feedback . . . . .	61
2.5.4 The Structure of a Control System – Transfer Functions . . . . .	62
2.6 Transfer Functions – Bode Plots . . . . .	64

---

2.6.1	The Bode Plot for Continuous Time Systems . . . . .	65
2.6.2	Calculating the Bode Plot for Continuous Time Systems . . . . .	66
2.7	PD Control, PID Control and Full State Feedback . . . . .	70
2.7.1	PD Control . . . . .	70
2.7.2	PID Control . . . . .	72
2.7.3	Full State Feedback . . . . .	74
2.7.4	Getting Derivatives for PID Control or Full State Feedback . . . . .	75
2.8	Further Reading . . . . .	80
<b>3</b>	<b>Nonlinear Differential Equations</b>	<b>83</b>
3.1	The Lorenz Attractor . . . . .	83
3.1.1	Linear Operating Points – Why the Lorenz Attractor is Chaotic . . . . .	85
3.2	Differential Equation Solvers in MATLAB and Simulink . . . . .	88
3.3	Tables, Interpolation and Curve Fitting in Simulink . . . . .	90
3.3.1	The Simple Lookup Table . . . . .	90
3.3.2	Interpolation – Fitting a Polynomial to the Data and Using the Result in Simulink . . . . .	93
3.3.3	Using Real Data in the Model – From Workspace and File . . . . .	95
3.4	Rotations in 3 Dimensions – Euler Rotations, Axis-angle Representations, Direction Cosines and the Quaternion . . . . .	97
3.4.1	Euler Angles . . . . .	97
3.4.2	Direction Cosines . . . . .	100
3.4.3	Axis-Angle Rotations . . . . .	101
3.4.4	The Quaternion Representation . . . . .	102
3.5	Modeling the Motion of a Satellite in Orbit . . . . .	108
3.5.1	Creating an Attitude Error when Using Direction Cosines . . . . .	110
3.5.2	Creating an Attitude Error Using Quaternion Representations . . . . .	111
3.5.3	The Complete Spacecraft Model . . . . .	112
3.6	Further Reading . . . . .	114
<b>4</b>	<b>Digital Signal Processing in Simulink</b>	<b>117</b>
4.1	Difference Equations, Fibonacci Numbers and z-Transforms . . . . .	118
4.1.1	The z-Transform . . . . .	121
4.1.2	Fibonacci (again) using z-Transforms . . . . .	122
4.2	Digital Sequences, Digital Filters and Signal Processing . . . . .	123
4.2.1	Digital Filters – Using z-Transforms – Discrete Transfer Functions . . . . .	123
4.2.2	Simulink Experiments –Filtering a Sinusoidal Signal and Aliasing . . . . .	126
4.2.3	The Simulink Digital Library . . . . .	131
4.3	Matrix Algebra and Discrete Systems . . . . .	132
4.4	The Bode Plot for Discrete Time Systems . . . . .	137
4.5	Digital Filter Design – Sampling Analog Signals, the Sampling Theorem and Filters . . . . .	139
4.5.1	Sampling and Reconstructing Analog Signals . . . . .	140
4.5.2	Analog Prototypes of Digital Filters – The Butterworth Filter . . . . .	144
4.6	The Signal Processing Blockset . . . . .	148
4.6.1	Fundamentals of the Signal Processing Blockset – Analog Filters . . . . .	149

---

4.6.2	Creating Digital Filters from Analog Filters . . . . .	151
4.6.3	Digital Signal Processing . . . . .	152
4.6.4	Implementing Digital Filters – Structures and Limited Precision . . . . .	157
4.6.5	Batch Filtering Operations, Buffers and Frames . . . . .	163
4.7	The Phase Lock Loop . . . . .	171
4.8	Further Reading . . . . .	175
<b>5</b>	<b>Random Numbers, White Noise and Stochastic Processes</b>	<b>179</b>
5.1	Modeling with Random Variables in Simulink – Monte Carlo Simulations	179
5.1.1	Monte Carlo Analysis and the Central Limit Theorem . . . . .	180
5.1.2	Simulating a Rayleigh Distributed Random Variable . . . . .	182
5.2	Stochastic Processes and White Noise . . . . .	184
5.2.1	The Random Walk Process . . . . .	185
5.2.2	Brownian motion and white noise . . . . .	187
5.3	Simulating System with White Noise Inputs Using the Weiner Process . . . . .	190
5.3.1	White Noise and a Spring Mass Damper System . . . . .	190
5.3.2	Noisy Continuous and Discrete Time Systems – The Covariance Matrix . . . . .	193
5.3.3	Discrete Time Equivalent of a Continuous Stochastic Process . . . . .	195
5.3.4	Modeling a Specified Power Spectral Density – One-Over-f Noise . . . . .	200
5.4	Further Reading . . . . .	206
<b>6</b>	<b>Modeling a Partial Differential Equation in Simulink</b>	<b>209</b>
6.1	The Heat Equation – Partial Differential Equations in Simulink . . . . .	210
6.1.1	Finite Dimensional Models . . . . .	210
6.1.2	An Electrical Analogy of the Heat Equation . . . . .	211
6.2	Converting the Finite Model into Equations for Simulation with Simulink	213
6.2.1	Using Kirchoff’s Law to Get the Equations . . . . .	214
6.2.2	The State-Space Model . . . . .	216
6.3	Partial Differential Equations for Vibration . . . . .	220
6.4	Further Reading . . . . .	220
<b>7</b>	<b>Stateflow – A Tool for Creating and Coding State Diagrams, Complex Logic, Event Driven Actions and Finite State Machines</b>	<b>223</b>
7.1	Properties of Stateflow – Building a Simple Model . . . . .	224
7.1.1	Stateflow Semantics . . . . .	226
7.1.2	Making the Simple Stateflow Chart Do Something . . . . .	229
7.1.3	Following Stateflow’s Semantics Using the Debugger . . . . .	232
7.2	Using Stateflow – A Controller for Home Heating . . . . .	232
7.2.1	Creating a Model of the System and an Executable Specification . . . . .	234
7.2.2	Stateflow’s Action Language Types . . . . .	237
7.2.3	The Heating Controller Layout . . . . .	238
7.2.4	Adding the User Actions, the Digital Clock, and the Stateflow chart to the Simulink Model of the Home Heating System . . . . .	239
7.2.5	Some Comments on Creating the GUI . . . . .	247
7.3	Further Reading . . . . .	249

---

<b>8</b>	<b>Physical Modeling – SimPowerSystems &amp; SimMechanics</b>	<b>251</b>
8.1	SimPowerSystems . . . . .	252
8.1.1	How the SimPowerSystems Blockset Works – Modeling a Nonlinear Resistor . . . . .	256
8.1.2	Using the Nonlinear Resistor Block . . . . .	259
8.2	Modeling an Electric Train Moving on a Rail . . . . .	262
8.3	SimMechanics – A Tool for Modeling Mechanical Linkages and Mechanical Systems . . . . .	266
8.3.1	Modeling a Pendulum with SimMechanics . . . . .	268
8.3.2	Modeling the Clock – Simulink and SimMechanics Together . . . . .	271
8.4	More Complex Models in SimMechanics and SimPowerSystems . . . . .	272
8.5	Further Reading . . . . .	277
<b>9</b>	<b>Putting Everything Together; Using Simulink in a System Design Process</b>	<b>279</b>
9.1	Specifications Development and Capture . . . . .	280
9.1.1	Modeling and Analysis – Converting the Specifications into an “Executable Specification” . . . . .	281
9.2	Modeling the System to Incorporate the Specifications – Lunar Module Rotation Using Time Optimal Control . . . . .	282
9.2.1	From Specification to Control Algorithm . . . . .	283
9.3	Design of System Components to Meet Specifications – Modify the Design to Accommodate Computer Limitations . . . . .	287
9.3.1	Final LM Control System Executable Specification . . . . .	289
9.3.2	The Control System Logic – Using Stateflow . . . . .	293
9.4	Verification and Validation of the Design . . . . .	296
9.5	The Final Step – Creating Embedded Code . . . . .	296
9.6	Further Reading . . . . .	298
<b>10</b>	<b>Conclusion – Thoughts about Broad Based Knowledge</b>	<b>299</b>
	<b>Bibliography</b>	<b>301</b>
	<b>Index</b>	<b>305</b>