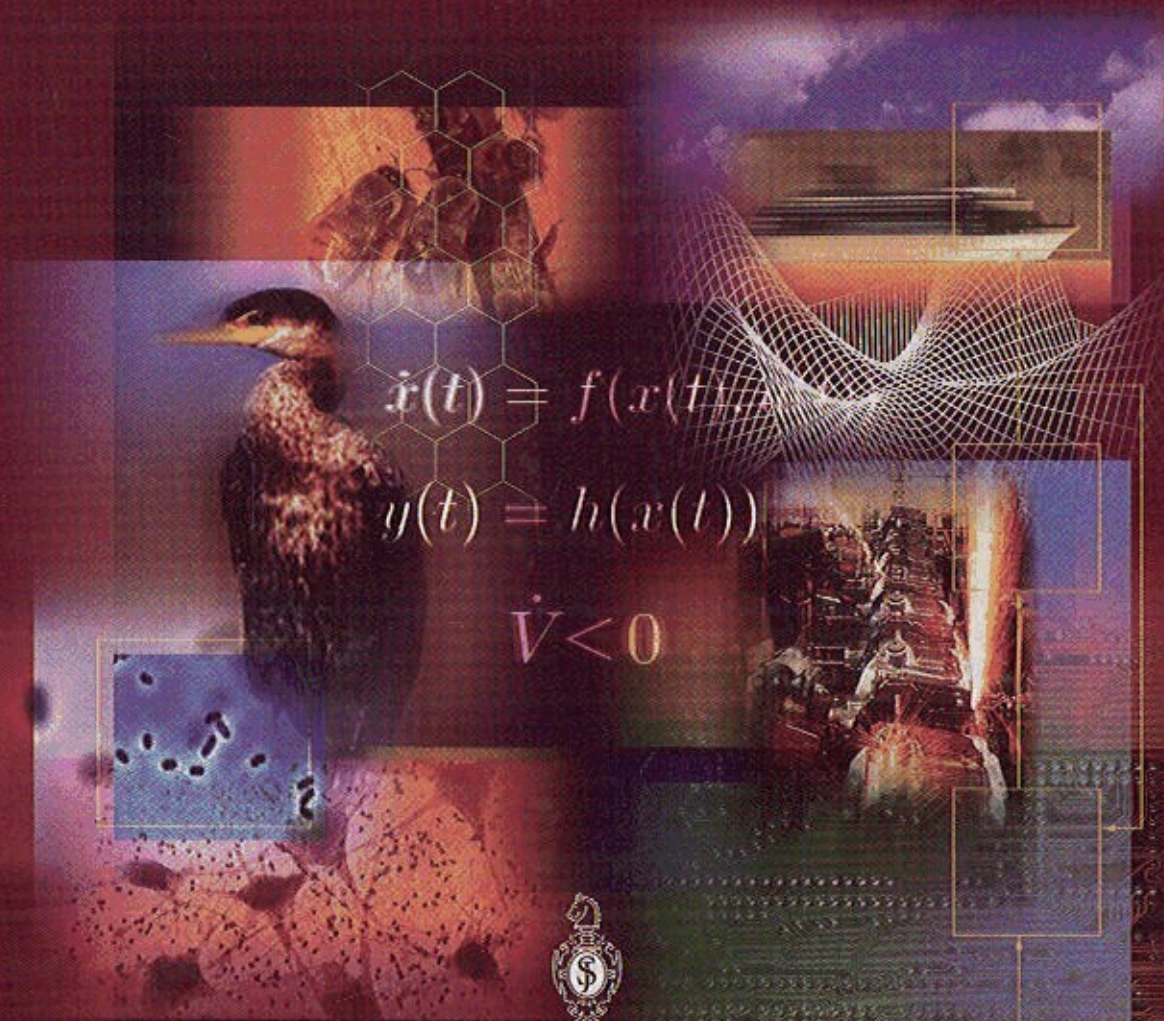


Kevin M. Passino

Biomimicry for Optimization, Control, and Automation



Springer

Contents

I	INTRODUCTION	1
----------	---------------------	----------

1	Challenges in Computer Control and Automation	7
1.1	The Role of Traditional Feedback Control Systems in Automation	9
1.2	Design Objectives for Control Systems	11
1.2.1	Tracking	12
1.2.2	Reducing Effects of Adverse Conditions	13
1.2.3	Behavior in Terms of Time Responses	14
1.2.4	Engineering Goals for Control Systems	16
1.3	Control System Design Methodology	18
1.3.1	Understand Plant and Specify Design Objectives	18
1.3.2	Construct Models and Uncertainty Representations	19
1.3.3	Analyze Model Accuracy and System Properties	21
1.3.4	Construct and Evaluate the Control System	23
1.3.5	Summary: The Iterative Design Procedure When Using Mathematical Models	26
1.3.6	Methodology Without Mathematical Models: The Use of Heuristics	28
1.4	Complex Hierarchical Control Systems for Automation	31
1.4.1	Functional Architectures	33
1.4.2	Organizing the Controller Functional Architecture	35
1.4.3	Fundamental Operational Characteristics	36
1.4.4	Example: Building Temperature Control	37
1.4.5	Example: Intelligent Transportation Systems	39
1.5	Design Objectives for Automation	41
1.6	Software Engineering for Complex Control Systems	43
1.6.1	Software Engineering Methods	43
1.6.2	Software Vs. Control Engineering Methodologies	47
1.6.3	Complex Control System Design Methodology	47
1.7	Implementing Complex Control Systems	50
1.8	Hybrid System Theory and Analysis	51
1.9	Exercises	52

2	Scientific Foundations for Biomimicry	57
2.1	Control Systems in Biology	60
2.2	Nervous Systems	61
2.2.1	Sensory, Motor, and Brain Processes	61
2.2.2	Functional Operation of the Brain	62
2.2.3	The Neurophysiological Level	64
2.2.4	Hierarchical Neural Organization	66
2.2.5	Brain Science: An Expanding Frontier	67
2.2.6	Biomimicry for Automation: Cognition	67
2.3	Organisms	69
2.3.1	Organisms Vs. Computers for Control	69
2.3.2	Example: Skinner's Pigeons for Missile Guidance	70
2.3.3	Human Control Expertise: "Human-Mimicry"	72
2.3.4	Human Operator Control Expertise: Quantity and Quality	73
2.4	Groups of Organisms	75
2.4.1	Social Foraging and Emergent Swarm Behavior	75
2.4.2	Example: Bacterial Chemotaxis	77
2.4.3	Emulation of Coordinated Behavior of Humans	78
2.4.4	Biomimicry for Automation	79
2.5	Evolution	80
2.5.1	The Evolutionary Process	80
2.5.2	Evolution of Behavior	82
2.5.3	Evolution of Intelligence	83
2.5.4	Example: Selective Breeding for Intelligence?	84
2.5.5	Biomimicry for Automation: Evolution	85
2.5.6	Evolution of Control Technology: Global Perspective	86
2.6	A Control Engineering Viewpoint	87
2.6.1	Why All the Biology? Do I Need It?	87
2.6.2	The Focus Is Engineering and Not the Foundational Sciences	88
2.6.3	Close Relationships To Conventional Control	89
2.6.4	Themes: Optimization, Adaptation, and Decision-Making	89
2.6.5	Intelligent Vs. Conventional Control? No Actual Conflict!	91
2.6.6	The Goal is Not "Intelligence," It Is Autonomy	91
2.7	Exercises	92
3	For Further Study	95

II ELEMENTS OF DECISION MAKING 99

4	Neural Network Substrates for Control Instincts	105
4.1	Biological Neural Networks and Their Role in Control	107
4.1.1	Neurons and Neural Networks	107
4.1.2	Example: Instinctual Neural Control Functions in Simple Organisms	109

4.2	Multilayer Perceptrons	111
4.2.1	The Neuron	112
4.2.2	Feedforward Network of Neurons	114
4.3	Design Example: Multilayer Perceptron for Tanker Ship Steering	116
4.3.1	Tanker Ship Model and Heading Regulation	116
4.3.2	Construction of a Multilayer Perceptron for Ship Steering	121
4.3.3	Multilayer Perceptron Stimulus-Response Characteristics	125
4.3.4	Behavior of the Ship Controlled by the Multilayer Perceptron	126
4.4	Radial Basis Function Neural Networks	131
4.5	Design Example: Radial Basis Function Neural Network for Ship Steering	133
4.5.1	A Radial Basis Function Neural Network for Ship Steering	133
4.5.2	Stimulus-Response Characteristics	138
4.5.3	Behavior of the Ship Controlled by the Radial Basis Function Neural Network	139
4.6	Stability Analysis	141
4.6.1	Differential Equations and Equilibria	141
4.6.2	Stability Definitions	144
4.6.3	Lyapunov's Direct Method for Stability Analysis	145
4.6.4	Stability of Discrete Time Systems	146
4.6.5	Example: Stable Instinctual Neural Control	147
4.7	Hierarchical Neural Networks	148
4.7.1	Example: Marine Mollusc	149
4.7.2	Hierarchical Neural Structures	149
4.8	Exercises and Design Problems	149
5	Rule-Based Control	153
5.1	Fuzzy Control	155
5.1.1	Choosing Fuzzy Controller Inputs and Outputs	155
5.1.2	Putting Control Knowledge into Rule Bases	157
5.1.3	Fuzzy Quantification of Knowledge	163
5.1.4	Matching: Determining Which Rules to Use	171
5.1.5	Inference Step: Determining Conclusions	175
5.1.6	Converting Decisions into Actions	178
5.2	General Fuzzy Systems	184
5.2.1	Multiple Input Multiple Output Fuzzy Systems	184
5.2.2	Takagi-Sugeno Fuzzy Systems	186
5.2.3	Mathematical Representations of Fuzzy Systems	187
5.2.4	Relationships Between Neural and Fuzzy Systems	193
5.3	Design Example: Fuzzy Control for Tanker Ship Steering	194
5.3.1	Simulation of a Fuzzy Controller	194
5.3.2	Fuzzy Controller Tuning for the Tanker Ship	201
5.3.3	Design Concerns	205
5.4	Stability Analysis	212
5.4.1	Example: Stability and Limit Cycles in Ship Steering	213

5.4.2	Discussion: Lyapunov Stability Analysis of Fuzzy Control Systems	214
5.5	Expert Control	214
5.5.1	General Knowledge Representations	215
5.5.2	General Inference Mechanisms	216
5.5.3	Stability Analysis of Expert Control Systems	216
5.6	Hierarchical Rule-Based Control Systems	217
5.7	Exercises and Design Problems	217
6	Planning Systems	225
6.1	Psychology of Planning	227
6.1.1	Essential Features of Planning	227
6.1.2	Generic Planning Steps	229
6.2	Design Example: Vehicle Guidance	231
6.2.1	Obstacle Course and Vehicle Characteristics	232
6.2.2	Path Planning Strategy	234
6.2.3	Simulation of the Guidance Strategy	238
6.2.4	More Challenges: Complex Mazes, Mobile Obstacles, and Uncertainty	238
6.3	Planning Strategy Design	241
6.3.1	Closed-Loop Planning Configuration	241
6.3.2	Models and Projecting into the Future	242
6.3.3	Optimization Criterion and Method for Plan Selection	243
6.3.4	Planning Using Preset Controllers and Model Learning	246
6.3.5	Hierarchical Planning Systems	247
6.3.6	Discussion: Concepts for Stable Planning	248
6.4	Design Example: Planning for a Process Control Problem	250
6.4.1	Level Control in a Surge Tank	250
6.4.2	Planner Design	251
6.4.3	Closed-Loop Performance	254
6.4.4	Effects of Planning Horizon Length	256
6.5	Exercises and Design Problems	257
7	Attentional Systems	263
7.1	Neuroscience and Psychology of Attention	265
7.1.1	Dynamically Changing Focus	266
7.1.2	Multistage Processing: Filtering, Selection, and Resource Allocation	267
7.2	Dynamics of Attention: Search and Optimization Perspective	268
7.2.1	Attentional Map	270
7.2.2	Optimization/Search Process for Focusing	271
7.3	Attentional Strategies for Multiple Predators and Prey	272
7.3.1	Cognitive Resource Allocation Model	272
7.3.2	Focus on a Predator/Prey Ignored for the Longest Time	276
7.3.3	Additional Attentional Strategies	279
7.3.4	Attentional Strategies Based on Predator/Prey Priority	281

7.3.5	Viewpoint of Attention Scheduling as Online Optimization	283
7.4	Design Example: Attentional Strategies	284
7.4.1	Simulation Approach and Performance Measures	284
7.4.2	Attentional Strategy Behavior: Focus on Longest Ignored	285
7.4.3	Effect of Focusing on Higher Priority Predators/Prey	287
7.4.4	Tuning Attentional Strategy Parameters	288
7.5	Stability Analysis of Attentional Strategies	290
7.5.1	Stability Properties of Attentional Strategies	290
7.5.2	Stabilizing Mechanism for Attentional Strategies	296
7.5.3	Planning and Attention	298
7.6	Attentional Systems in Control and Automation	302
7.6.1	Attentional Strategies for Control	302
7.6.2	Filtering and Focusing: Multisensor Integration	303
7.7	Exercises and Design Problems	305
8	For Further Study	309

III LEARNING 313

9	Learning and Control	321
9.1	Psychology and Neuroscience of Learning	323
9.1.1	Habituation and Sensitization	325
9.1.2	Classical Conditioning: Learning Associations Between Stimuli	325
9.1.3	Hebbian/Gradient Model of Neural-Level Classical Conditioning	328
9.1.4	Operant Conditioning: Learning to Predict Consequences of Actions	335
9.1.5	Control System Model of Behavioral-Level Operant Conditioning	339
9.2	Function Approximation as Learning	342
9.2.1	Using Functions to Represent Mappings in Data	343
9.2.2	Choosing the Training Data Set	345
9.2.3	Example: Collecting Data for Function Approximation	347
9.2.4	Measuring Approximation Accuracy: Using a Test Set	350
9.3	Approximator Structures as Substrates for Learning	351
9.3.1	Linear and Polynomial Approximator Structures	352
9.3.2	Neural and Fuzzy System Approximator Structures	354
9.3.3	Universal Approximation Property and Substrate Capabilities	364
9.3.4	Approximator Complexity Vs. Substrate Flexibility	366
9.3.5	Linear Vs. Nonlinear in the Parameter Approximators: Substrate Tunability	367
9.3.6	Online Function Approximation: Dynamic Learning	369

9.4	Biomimicry for Heuristic Adaptive Control	371
9.4.1	Reinforcement Learning for Neural Control	372
9.4.2	Design Example: Neural Control for the Tanker Ship . . .	376
9.4.3	Adaptive Fuzzy Control: Emulating Adaptation Expertise	388
9.4.4	Design Example: Rule-Tuning for the Tanker Ship	402
9.4.5	Expert, Planning, and Attentional Systems for Adaptive Control	407
9.4.6	Development, Plasticity, and Control	412
9.5	Exercises and Design Problems	415
10	Linear Least Squares Methods	421
10.1	Batch Least Squares	423
10.1.1	Batch Least Squares Derivation	423
10.1.2	Numerical Issues in Computing the Estimate	425
10.1.3	Example: Fitting a Line to Data	428
10.2	Example: Offline Tuning of Approximators	429
10.2.1	Multilayer Perceptrons	429
10.2.2	Takagi-Sugeno Fuzzy Systems	433
10.3	Design Example: Rule Synthesis Using Operator Data	437
10.3.1	Data Analysis, Correlation Analysis, and Controller Input Selection	438
10.3.2	Determine if a Linear Controller Is Sufficient	440
10.3.3	Study the Effects of Removing Input Variables	441
10.3.4	Construct a Fuzzy Controller from Operator Data	444
10.3.5	Methods to Test Generalization/Extrapolation and Con- troller Validity	450
10.4	Recursive Least Squares	451
10.4.1	Recursive Least Squares Derivation	451
10.4.2	Weighted Recursive Least Squares: Using a Forgetting Factor	453
10.4.3	Numerical Issues and Covariance Modifications	454
10.4.4	Example: Fitting a Line to Data	456
10.5	Example: Online Tuning of Approximators	457
10.5.1	Multilayer Perceptrons	457
10.5.2	Takagi-Sugeno Fuzzy Systems	462
10.6	Exercises and Design Problems	464
11	Gradient Methods	471
11.1	The Steepest Descent Method	475
11.1.1	Steepest Descent Parameter Updates	475
11.1.2	Example: Convergence, Step Size, and Termination Issues	476
11.1.3	Descent Direction Possibilities and Momentum	478
11.1.4	The Linear in the Parameter Case	479
11.1.5	Step Size Choice	481
11.1.6	Parameter Initialization, Constraints, and Update Termi- nation	485

11.1.7	Offline and Online, Serial and Parallel Data Processing	489
11.1.8	Stochastic Gradient Optimization Basics	490
11.2	Levenberg-Marquardt and Conjugate Gradient Methods	491
11.2.1	Newton's Method	491
11.2.2	Conjugate Gradient and Quasi-Newton Methods	493
11.2.3	Gauss-Newton and Levenberg-Marquardt Methods	496
11.3	Matlab for Training Neural Networks	499
11.3.1	Motivation to Use Software Packages	499
11.3.2	Example: Matlab Neural Networks Toolbox	500
11.4	Example: Levenberg-Marquardt Training of a Fuzzy System	503
11.4.1	Update Formulas	505
11.4.2	Parameter Constraint Set and Initialization	507
11.4.3	Approximator Tuning Results: Effects on the Nonlinear Part	508
11.4.4	Approximator Tuning Results: Effects of Initialization	509
11.4.5	Overtraining, Overfitting, and Generalization	511
11.4.6	Approximator Reparameterization for Flexibility and Com- plexity Reduction	512
11.4.7	Approximation Error Measures: Using a Test Set	513
11.5	Example: Online Steepest Descent Training of a Neural Network	514
11.5.1	Update Formulas	516
11.5.2	Parameter Constraints and Initialization	520
11.5.3	Approximator Tuning Results: Effects of Step Size	521
11.5.4	Approximator Tuning Results: Effects of Initialization	523
11.5.5	Can We Improve Approximation Accuracy?	524
11.5.6	Local Vs. Global Tuning/Learning	526
11.6	Clustering for Classifiers and Approximators	528
11.6.1	Using Approximators to Solve Classification Problems	530
11.6.2	Clustering Methods: Gradient Approaches	531
11.6.3	Fuzzy C-Means Clustering and Function Approximation	536
11.7	Neural or Fuzzy: Which is Better? Bad Question!	542
11.8	Exercises and Design Problems	543
12	Adaptive Control	547
12.1	Strategies for Adaptive Control	549
12.1.1	Indirect Adaptive Control	549
12.1.2	Direct Adaptive Control	550
12.2	Classes of Nonlinear Discrete-Time Systems	551
12.2.1	Nonlinear Discrete-Time Systems	551
12.2.2	Example: Linear Systems with Unknown Constant Coef- ficients	552
12.3	Indirect Adaptive Neural/Fuzzy Control	553
12.3.1	Estimators and Certainty Equivalence Controller	553
12.3.2	Error Equations and Representation Error Bounds	555
12.3.3	Adaptation Methods	557
12.3.4	Discussion: Multiple Model Adaptation Strategies	561

12.4	Design Example: Indirect Neural Control for a Process Control	
	Problem	562
12.4.1	The Neural Controller Development	562
12.4.2	Indirect Neural Control Results	565
12.5	Direct Adaptive Neural/Fuzzy Control	567
12.5.1	Development of Controller Estimator and Error Equations	567
12.5.2	Adaptation Method	572
12.6	Design Example: Direct Neural Control for a Process Control	
	Problem	573
12.6.1	Direct Neural Controller Results	573
12.6.2	Tuned and Ideal Controller Mapping Shapes	575
12.7	Stable Adaptive Fuzzy/Neural Control	576
12.7.1	Class of Nonlinear Systems	577
12.7.2	Indirect Adaptive Control	579
12.7.3	Direct Adaptive Control	589
12.7.4	Design Example: Aircraft Wing Rock Regulation	591
12.8	Discussion: Tuning Structure and Nonlinear in the Parameter	
	Approximators	594
12.9	Exercises and Design Problems	597
13	For Further Study	601

IV	EVOLUTION	607
-----------	------------------	------------

14	The Genetic Algorithm	613
14.1	Biological Evolution	615
14.2	Representing the Population of Individuals	616
14.2.1	Strings, Chromosomes, Genes, Alleles, and Encodings	616
14.2.2	Encoding Examples	617
14.2.3	The Population of Individuals	619
14.3	Genetic Operations	620
14.3.1	Selection	621
14.3.2	Reproduction Phase, Crossover	622
14.3.3	Reproduction Phase, Mutation	625
14.4	Programming the Genetic Algorithm	626
14.4.1	Pseudocode for a Simple Genetic Algorithm	626
14.4.2	Alternative Sequencing of Operations	627
14.4.3	Representations, Complexity, Termination, and Initializa- tion	628
14.5	Example: Solving an Optimization Problem	630
14.5.1	Genetic Algorithm Design	631
14.5.2	Algorithm Performance and Tuning	631
14.6	Approximations to Reduce Algorithm Complexity	636
14.6.1	Reducing Algorithm Complexity	636

14.6.2	Crossover Option 1	637
14.6.3	Crossover Option 2	638
14.7	Exercises and Design Problems	639
15	Stochastic and Nongradient Optimization for Design	647
15.1	Design of Robust Organisms and Systems	649
15.2	Response Surface Methodology for Design	652
15.2.1	Controller Design for a Tanker Ship	653
15.2.2	Response Surface and “Optimal Gains”	653
15.2.3	Response Surface and “Optimal Gains” for Off-Nominal Conditions	654
15.2.4	Design Optimization Over Multiple Response Surfaces: Robustness Trade-Offs	655
15.2.5	Choosing Design Points, Approximating Response Sur- faces, and Optimizing Designs	658
15.3	Nongradient Optimization	661
15.3.1	Pattern and Coordinate Search	661
15.3.2	Simultaneous Perturbation Stochastic Approximation Al- gorithm	668
15.3.3	Nelder-Mead Simplex Method	677
15.3.4	The Multidirectional Search Method	686
15.4	SPSA for Decision-Making System Design: Examples	692
15.4.1	Design Example: SPSA for Tanker Ship PD Controller Design	692
15.4.2	Design Example: SPSA for Attentional Strategy Design	697
15.5	Parallel, Interleaved, and Hierarchical Nongradient Methods	699
15.5.1	Pattern Search	701
15.5.2	Evolutionary Strategies	702
15.6	Set-Based Stochastic Optimization for Design	703
15.6.1	A Set-Based Stochastic Optimization Method	704
15.6.2	Design Example: Tanker Controller Design	705
15.7	Discussion: Evolutionary Control System Design	706
15.7.1	Genetic Algorithms for Computer-Aided Control System Design	706
15.7.2	Darwinian Design for Physical Control Systems	710
15.8	Exercises and Design Problems	712
16	Evolution and Learning: Synergistic Effects	719
16.1	Relevant Theories of Biological Evolution	721
16.1.1	Genetics of Learning	722
16.1.2	The Evolution of Learning	722
16.1.3	The Baldwin Effect: Learning Can Accelerate Evolution of Instincts	724
16.1.4	Evolving an Instinct-Learning Balance	725
16.1.5	Cultural Influences on Learning and Evolution	726
16.2	Robust Approximator Size Design	727

16.2.1	Approximator Size Design Problem	727
16.2.2	Average Mean-Squared Error Response Surface for Approximator Size	728
16.3	Instinct-Learning Balance in an Uncertain Environment	730
16.3.1	Estimator Design Problem and Analogies with Instincts and Learning	731
16.3.2	Response Surfaces for Optimal Instinct-Learning Balance	733
16.3.3	Evolving Instinct-Learning Balance Via Set-Based Optimization	735
16.4	Discussion: Instinct-Learning Balance for Adaptive Control	737
16.5	Genetic Adaptive Control	738
16.5.1	Indirect Genetic Adaptive Control	740
16.5.2	Direct Genetic Adaptive Control	744
16.5.3	Design Example: Process Control Problem	747
16.6	Exercises and Design Problems	750
17	For Further Study	755

V FORAGING 759

18	Cooperative Foraging and Search	765
18.1	Foraging Theory	768
18.1.1	Elements of Foraging Theory	768
18.1.2	Behavioral/Sensory Ecology of Search	769
18.1.3	Cooperative/Social Foraging	772
18.1.4	Nongradient Optimization Models	775
18.2	Bacterial Foraging: <i>E. coli</i>	776
18.2.1	Swimming and Tumbling via Flagella	777
18.2.2	Bacterial Motile Behavior: Climbing Nutrient Gradients	780
18.2.3	Underlying Sensing and Decision-Making Mechanisms	782
18.2.4	Elimination and Dispersal Events	783
18.2.5	Evolution of Bacteria	783
18.2.6	Taxes in Other Swimming Bacteria	784
18.2.7	Other Group Phenomena in Bacteria	785
18.3	<i>E. coli</i> Bacterial Swarm Foraging for Optimization	787
18.3.1	An Optimization Model for <i>E. coli</i> Bacterial Foraging	788
18.3.2	Bacterial Foraging Optimization Algorithm	792
18.3.3	Guidelines for Algorithm Parameter Choices	794
18.3.4	Relations to Other Nongradient Optimization Methods	795
18.3.5	Example: Function Optimization via <i>E. coli</i> Foraging	796
18.4	Stable Social Foraging Swarms	798
18.4.1	Example: The Biology of Honey Bee Swarms	799
18.4.2	Swarm and Environment Models	800
18.4.3	Stability Analysis of Swarm Cohesion Properties	804

18.4.4 Cohesion Characteristics and Swarm Dynamics	814
18.5 Design Example: Robot Swarms	817
18.5.1 Robot Swarm Formulation	817
18.5.2 Performance in Obstacle Avoidance and Noise Effects . .	817
18.5.3 Additional Robot Swarm Design Challenges	818
18.6 Exercises and Design Problems	820
19 Competitive and Intelligent Foraging	829
19.1 Competition and Fighting in Nature	831
19.1.1 Foraging Games	832
19.1.2 Intelligent Foragers	833
19.1.3 Evolution and Foraging	834
19.2 Introduction to Game Theory	834
19.2.1 Strategies and Information for Decisions	835
19.2.2 Nash, Minimax, and Stackelberg Strategies	840
19.2.3 Cooperation and Pareto-Optimal Strategies	847
19.3 Design Example: Static Foraging Games	855
19.3.1 Static Foraging Game Model	855
19.3.2 Competition and Cooperation for a Resource	858
19.3.3 Energy Constraints and Multiple Resources	862
19.4 Dynamic Games	863
19.4.1 Modeling the Game Arena and Observations	865
19.4.2 Information Space and Strategies	866
19.4.3 Decision and Action Timing	868
19.5 Example: Dynamic Foraging Games	868
19.5.1 Dynamic Foraging Game Model	868
19.5.2 Biomimicry for Foraging Strategies	874
19.6 Challenge Problems: Intelligent Social Foraging	877
19.6.1 Intelligent Foraging	878
19.6.2 Intelligent Social Foraging	881
19.7 Exercises and Design Problems	892
20 For Further Study	895
BIBLIOGRAPHY	899
INDEX	922