

Contents

<i>Preface</i>	vii
1. Artificial Evolution Based Autonomous Robot Navigation	1
1.1 Introduction	1
1.2 Evolutionary Robotics	4
1.3 Adaptive Autonomous Robot Navigation	8
1.4 Artificial Evolution in Robot Navigation	11
1.4.1 Neural Networks	12
1.4.2 Evolutionary Algorithms	13
1.4.3 Fuzzy Logic	16
1.4.4 Other Methods	17
1.5 Open Issues and Future Prospects	18
1.5.1 SAGA	18
1.5.2 Combination of Evolution and Learning	19
1.5.3 Inherent Fault Tolerance	20
1.5.4 Hardware Evolution	21
1.5.5 On-Line Evolution	22
1.5.6 Ubiquitous and Collective Robots	23
1.6 Summary	24
<i>Bibliography</i>	25
2. Evolvable Hardware in Evolutionary Robotics	33
2.1 Introduction	33
2.2 Evolvable Hardware	34
2.2.1 Basic Concept of EHW	34

2.2.2	Classification of EHW	35
2.2.2.1	Artificial evolution and hardware device	35
2.2.2.2	Evolution process	37
2.2.2.3	Adaptation methods	39
2.2.2.4	Application areas	39
2.2.3	Related Works	40
2.3	Evolutionary Robotics	40
2.4	Evolvable Hardware in Evolutionary Robotics	42
2.5	Promises and Challenges	50
2.5.1	Promises	50
2.5.2	Challenges	52
2.6	Summary	53
	<i>Bibliography</i>	55
3.	FPGA-Based Autonomous Robot Navigation via Intrinsic Evolution	63
3.1	Introduction	63
3.2	Classifying Evolvable Hardware	66
3.2.1	Evolutionary Algorithm	66
3.2.2	Evaluation Implementation	66
3.2.3	Genotype Representation	68
3.3	Advantages of Evolvable Hardware	69
3.3.1	Novel Hardware Designs	69
3.3.2	Low Cost	71
3.3.3	Speed of Execution	71
3.3.4	Economy of Resources	72
3.4	EHW-Based Robotic Controller Design	72
3.4.1	Evolvable Hardware	73
3.4.2	Function Unit	75
3.4.3	EHW-Based Robotic Controller Design	76
3.4.3.1	Boolean function controller	78
3.4.3.2	Chromosome representation	79
3.4.3.3	Evolution and adaptation methodology	80
3.4.3.4	Robot navigation tasks	82
3.5	Hardware and Development Platform	82
3.5.1	Sensor Information	82
3.5.2	FPGA Turret	84

3.5.2.1	Description	85
3.5.2.2	Architecture	85
3.5.2.3	Configuration bits	86
3.5.3	Hardware Configuration	87
3.5.4	Development Platform	89
3.6	Implementation Of EHW-Based Robot Navigation	91
3.6.1	Preliminary Investigation	91
3.6.2	Light Source Following Task	97
3.6.2.1	Software structure of light following task	97
3.6.2.2	Program settings of light following task	98
3.6.2.3	Implementation of light source following task	101
3.6.3	Obstacle Avoidance using Robot with a Traction Fault	103
3.6.3.1	Software structure of anti-collision task	106
3.6.3.2	Implementation of obstacle avoidance task	106
3.7	Summary	109

Bibliography 113

4.	Intelligent Sensor Fusion and Learning for Autonomous Robot Navigation	117
4.1	Introduction	117
4.2	Development Platform and Controller Architecture	120
4.2.1	The Khepera Robot and Webots Software	120
4.2.2	Hybrid Architecture of the Controller	123
4.2.3	Function Modules on Khepera Robot	125
4.3	Multi-Stage Fuzzy Logic (MSFL) Sensor Fusion System	127
4.3.1	Feature-Based Object Recognition	127
4.3.2	MSFL Inference System	128
4.4	Grid Map Oriented Reinforcement Path Learning (GMRPL)	131
4.4.1	The World Model	131
4.4.2	GMRPL	132
4.5	Real-Time Implementation	134
4.6	Summary	135

Bibliography 143

5.	Task-Oriented Developmental Learning for Humanoid Robots	145
5.1	Introduction	145

5.2	Task-Oriented Developmental Learning System	148
5.2.1	Task Representation	148
5.2.2	The AA-Learning	148
5.2.3	Task Partition	149
5.3	Self-Organized Knowledgebase	150
5.3.1	PHDR Algorithm	150
5.3.2	Classification Tree and HDR	150
5.3.3	PHDR	152
5.3.4	Amnesic Average	154
5.4	A Prototype TODL System	155
5.4.1	Robot Agent	155
5.4.1.1	Khepera robot	155
5.4.1.2	Interface	156
5.4.2	TODL Mapping Engine	157
5.4.3	Knowledge Database	158
5.4.4	Sample Task	158
5.4.4.1	Goods distribution problem	158
5.4.4.2	Objects identification	158
5.4.4.3	PHDR representation of the sample task	159
5.5	Discussions	161
5.6	Summary	162
	<i>Bibliography</i>	165
6.	Bipedal Walking Through Reinforcement Learning	167
6.1	Introduction	167
6.2	The Bipedal Systems	169
6.3	Control Architecture	170
6.4	Key Implementation Tools	172
6.4.1	Virtual Model Control	172
6.4.2	Q-Learning	173
6.4.3	Q-Learning Algorithm Using Function Approximator for Q-Factors	174
6.5	Implementations	176
6.5.1	Virtual Model Control Implementation	176
6.5.2	Reinforcement Learning to Learn Key Swing Leg's Parameter	177
6.5.2.1	State variables	178

6.5.2.2	Reward function and reinforcement learning algorithm	179
6.6	Simulation Studies and Discussion	180
6.6.1	Effect of Local Speed Control on Learning Rate . . .	180
6.6.2	Generality of Proposed Algorithm	184
6.7	Summary	187
<i>Bibliography</i>		191
7.	Swing Time Generation for Bipedal Walking Control Using GA tuned Fuzzy Logic Controller	195
7.1	Introduction	195
7.2	Fuzzy Logic Control and Genetic Algorithm	197
7.2.1	Fuzzy Logic Control (FLC)	197
7.2.2	Genetic Algorithms (GAs)	198
7.2.3	GA Tuned FLC	199
7.3	Linear Inverted Pendulum Model	200
7.4	Proposed Bipedal Walking Control Architecture	202
7.4.1	Bipedal Walking Algorithm	203
7.4.2	Intuitions of Bipedal Walking Control from Linear Inverted Pendulum Model	203
7.4.3	Fuzzy Logic Controller (FLC) Structure	204
7.4.4	Genetic Algorithm Implementation	206
7.4.4.1	Coding the information	206
7.4.4.2	Evaluation	207
7.4.4.3	Evolutionary operators	208
7.5	Simulation Result	210
7.6	Summary	212
<i>Bibliography</i>		217
8.	Bipedal Walking: Stance Ankle Behavior Optimization Using Genetic Algorithm	219
8.1	Introduction	219
8.2	Virtual Model Control	220
8.3	Genetic Algorithm (GA)	224
8.3.1	GA's Operations	224
8.3.2	GA's Parameters	225

8.3.3	Fitness Function	226
8.4	Simulation Results and Discussion	227
8.4.1	Convergence to Optimal Solution	227
8.4.2	A Comparison with Solution Produced by Enumerative Method of Optimization	229
8.4.3	The Effects of GA's Parameters	230
8.5	Summary	232
	<i>Bibliography</i>	235
9.	Concluding Statements	237
9.1	Summary	237
9.2	Future Research Emphases	239
9.2.1	On-Line Evolution	239
9.2.2	Inherent Fault Tolerance	240
9.2.3	Swarm Robotics	241
	<i>Index</i>	243