Springer Tracts in Advanced Robotics

Volume 11

Editors: Bruno Siciliano · Oussama Khatib · Frans Groen

Springer

Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

J.-H. Kim · D.-H. Kim · Y.-J. Kim · K.-T. Seow

Soccer Robotics

With 205 Figures and 12 Tables

Springer

Professor Bruno Siciliano, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy, email: siciliano@unina.it

Professor Oussama Khatib, Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA, email: khatib@cs.stanford.edu

Professor Frans Groen, Department of Computer Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, email: groen@science.uva.nl

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

Authors

Dr. Jong-Hwan Kim
Dept. of Electrical Engineering
and Computer Science
Korea Advanced Institute of
Science and Technology (KAIST)
373-1 Gusong-dong, Yusong-gu
Daejeon 305-701
Republic of Korea

Dr. Dong-Han Kim
Dept. of Electrical Engineering
and Computer Science
Korea Advanced Institute of
Science and Technology (KAIST)
373-1 Gusong-dong, Yusong-gu
Daejeon 305-701
Republic of Korea

Dr. Yong-Jae Kim
Intelligent Robot Lab.
Inst. of Intel. Syst. Mechatronic Center
Samsung Electronics Co.
Maeton 3-dong
Paldal-gu, Suwon-si
Gyeonggi-do 442-742
Republic of Korea

Dr. Kiam-Tian Seow
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue
Singapore 639798
Singapore

ISSN 1610-7438

ISBN 3-540-21859-9 Springer-Verlag Berlin Heidelberg New York

Library of Congress Control Number: 2004104485

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Digital data supplied by authors.
Data-conversion and production: PTP-Berlin Protago-TeX-Production GmbH, Germany
Cover-Design: design & production GmbH, Heidelberg
Printed on acid-free paper 62/3020Yu - 5 4 3 2 1 0

Foreword

At the dawn of the new millennium, robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into the challenges of unstructured environments. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

The goal of the new series of Springer Tracts in Advanced Robotics (STAR) is to bring, in a timely fashion, the latest advances and developments in robotics on the basis of their significance and quality. It is our hope that the wider dissemination of research developments will stimulate more exchanges and collaborations among the research community and contribute to further advancement of this rapidly growing field.

This monograph written by Jong-Hwan Kim, Dong-Han Kim, Yong-Jae Kim and Kiam-Tian Seow forms an introduction to the field of Soccer Robotics. Soccer Robotics has become an important research area with different competing initiatives. It integrates mechatronics, computer science and artificial intelligence techniques to create real-world autonomous systems, which are not only fun to see. Soccer Robotics forms also a test bed for system integration of autonomous systems comparing different approaches in various competitions with different levels of distributed perception and collaboration. Soccer Robotics opens the route towards collaborating autonomous robot systems in a real-world adversarial setting.

The focus of this monograph is on the **FIRA** framework of Soccer Robotics, in particular MiroSot, which uses a central overhead camera to overview the whole soccer field, and a central control of the robots. The monograph gives a complete description of the different aspects needed to create a soccer team. It describes the hardware aspects, the computer vision needed, navigation, action selection, basic skills and game strategy. These aspects are described at an undergraduate level, and up to a junior graduate level, showing its use of as text book but also a must for everyone who wants to enter MiroSot robotics. A fine addition to the series!

Amsterdam — *Frans Groen*

February 2004 — STAR Editor

Preface

Autonomous robots which are adaptable, communicative and objective-oriented, and intelligent multi-agent robotic systems in general, are so evidently complex that it has become increasingly necessary to find a domain that can serve as an integrated framework for the complementary purposes of research and education. Robot soccer is one such suitable domain that is representative of intelligent multi-agent robotic systems, in which multiple robotic agents (or simply, multiple robots) need to cooperate in an adversarial environment to achieve specific objectives. It is a game based on the modified rules of human soccer and is played in a scaled down soccer field, in which two soccer robot teams compete by attempting to move a ball into the opponent team's goal. The team with a higher score at the end of regulation time wins. Technically, robot soccer is a competitive game that makes heavy demands in all the key areas of robot technology, namely, mechanics, control, sensors, communication, and intelligence. On the one hand, it spurs wide-ranging multidisciplinary research work by providing a comprehensive test bed that facilitates the concrete demonstration and performance evaluation of new ideas and concepts. On the other hand, it captivates as an educational tool that helps students better understand and appreciate the scientific knowledge and technological developments in an inherently multidisciplinary setting of intelligent multi-agent robotic systems.

Since its inception in 1995, robot soccer has evolved into a recognized area of its own. This area, called *Soccer Robotics*, is a subfield of *AI Robotics* that offers a challenging domain for research and education in a large spectrum of issues integrating the problems of *sensing*, *deciding* and *acting* that are of relevance to the development of complete autonomous agents in general. The hope in *Soccer Robotics*, of course, is that by discovering how to get a team of robots to sense with acuity, decide collaboratively and act in coordination within the limited context of a soccer game, it will be possible to use the same techniques and technologies to build robots that carry out other more useful tasks. The development of this subfield is actively supported through the Micro-Robot Soccer Tournament (MiroSoT) and Simulated-Robot Soccer Tournament (SimuroSoT) Categories of the **FIRA Cup**, an international event organized by the Federation of International Robot-soccer Association (**FIRA**, *http://www.fira.net*). **FIRA Cup**, held annually since 1996, has

been the 'examination' ground for the testing of new techniques and technologies integrated in the game of robot soccer, and has provided much excitement and entertainment for all those who participated.

This new book *Soccer Robotics* is intended to be a comprehensive introduction to the field of soccer robotics, emphasizing breadth of coverage and accessibility of the material to readers with possibly different backgrounds. Its key feature is the emphasis placed on a robot soccer-programming framework that integrates all the key areas of robot technology. Until now, these areas had been treated mainly in separate books or in research literature only, outside the arena of soccer robotics.

A substantial portion of this book is based on the first author's lectures **EE006 Robot Soccer System** at the Korea Advanced Institute of Science and Technology in the period July 13 - August 14, 1998. The material on robot soccer originated with the KAIST postgraduate theses of Dong-Han Kim (2003,1998), Yong-Jae Kim (2003), Hyun-Sik Shim (1998), Mun-Soo Lee (2000), Heung-Soo Kim (1997) and others, together with joint publications with the first author. The experimental robot soccer system program for Small League MiroSoT that supplements this book has been developed with the help of many students in the first author's *Robot Intelligence Technology* (RIT) Lab at KAIST, while the simulator package for Large League SimuroSoT has been developed by Bing-Rong Hong's research team at Harbin Institute of Technology, P.R. China. Both are available for free download from the **FIRA** website *http://www.fira.net*.

Soccer Robotics is written as a textbook for practical courses at the undergraduate level, and up to the first-year graduate level. It is useful for researchers and practising engineers interested in trying out new techniques in the domain of robot soccer. This book is also suitable for anyone interested in learning and developing robot soccer systems for edutainment purposes. For those interested in participating in either the MiroSoT or SimuroSoT categories of the annual **FIRA Cup** and other robot-soccer championship events, the material in this book will provide a firm foundation for the development of robot soccer systems to competitive standards. The book will be of interest to scientists, engineers and students in a variety of disciplines besides *AI Robotics*, where the use of robot soccer as a test bed is relevant: sensors (including computer vision), control, communication, multiagent systems and artificial intelligence.

To review the chapters briefly:

Chapter 1 defines the multi-agent framework of soccer robotics in terms of the three commonly accepted primitives of AI robotics, namely, **SENSE**, **DECIDE** and **ACT**. The goals of soccer robotics in research and education of intelligent multi-agent robotic systems are explained. The various categories of robot soccer created by **FIRA**, an international regulating body for robot soccer, are described. The classification of robot soccer systems for MiroSoT is also examined.

Chapter 2 presents the basic theoretical background on the mechanical motion of mobile robots, with emphasis on the kinematics of a two-wheel MiroSoT robot. The essentials of hardware and firmware needed to build a two-wheel MiroSoT robot with IR or RF communication are covered in sufficient detail.

Chapter 3 focusses on the (visual) **SENSE** primitive; in particular, it presents how the postures of target objects in robot soccer can be computed using centralized vision techniques. The basics of computer vision are first introduced. Real examples are then provided to highlight the practical considerations in building a good vision system for a MiroSoT team.

Chapter 4 focusses on the **DECIDE** and **ACT** primitives. A hybrid control architecture is introduced that integrates the three primitives of **SENSE**, **DECIDE** and **ACT** in a hierarchy of four interacting levels, namely, role, action, behaviour and execution. To expose the technical challenges involved, example strategies are given at the role level and action level. Action designs for robot soccer, to be implemented at the behavioral level, are classified and explained. An overview of classical PID control, applicable at the behavioral level and execution level, follows. Finally, two different navigation methods, applicable at the behavioral level, are presented.

Chapter 5 motivates the importance of the various aspects of intelligence, namely, *search and evolution*, *knowledge representation and inference* and *learning and adaptation*, as needed by the **DECIDE** and **ACT** primitives. Following, it demonstrates, by examples, the applicability of Petri nets, Q-learning, neural networks, evolutionary programming and fuzzy logic to robot soccer under the MiroSoT category. These soft-computing paradigms make concrete (at least one of) the abstract aspects of intelligence. For each paradigm, one or two examples are provided that address some key issues at specific hierarchical levels of the hybrid control architecture introduced in Chapter 4.

Chapter 6 introduces a host software model for MiroSoT robot soccer system. An overview of the programming framework for robot soccer is then presented, in which a number of the robot soccer concepts described in earlier chapters are illustrated through example 'C' programs which are the key functions of a robot soccer system for Small League (3-a-side) MiroSoT.

Chapter 7 complements the *real-system* programming framework presented in the previous chapter with a *computer-simulated system* programming framework. It presents the core simulator system and programming framework for Large League (11-a-side) SimuroSoT. Example 'C^{++}' codes are provided for illustration.

As do all authors of technical work, we wish to acknowledge the many contributors on whose work our own presentation is partly based. The list of references gives some indication of those to whom we are in debt. On a more

personal level, we would expressly like to thank Hyun-Sik Shim, Myung-Jin Jung, Heung-Soo Kim, Kuk-Hyun Han, Kui-Hong Park, Ming Yu-Chi, Jun-Su Jang, Kang-Hee Lee, Jayyati Ghoshal and many other students in the RIT Lab who have contributed to this book in a wide range of invaluable ways.

This book would never have been possible without the funding that came from a variety of sources to support the research and development work in soccer robotics, and the writing of this book. The first author would like to acknowledge each of these agencies: LG, Samsung, POSCO, KOSEF and MRDEC. The last author would like to acknowledge the award of a 'Brain Korea 21' Institute Fellowship in 2002 that supported his joint research and authorship at KAIST.

Finally, the authors are indebted to Dr. Thomas Ditzinger, Engineering Editor at Springer Verlag, for editorial assistance, and quality production of this book.

KAIST, Daejeon, Korea, Jan 2003 *Jong-Hwan Kim*

KAIST, Daejeon, Korea *Dong-Han Kim*

Samsung Electronics, Suwon, Korea *Yong-Jae Kim*

NTU, Singapore *Kiam-Tian Seow*

Contents

Foreword VIII

Preface IX

List of Figures XIX

List of Tables XXV

1. Soccer Robotics 1
1.1 Introduction 1
1.1.1 Agents, Multi-agent Systems, and AI Robotics 1
1.1.2 Cooperative Robot Teams 2
1.1.3 Domain Characteristics 6
1.1.4 Robot Soccer 6
1.2 The Goals of Soccer Robotics 7
1.2.1 Test Bed for Robotics Research and Development (R&D) 8
1.2.2 Educational Tool for AI Robotics 9
1.2.3 FIRA Robot World Cup 10
1.2.4 Technology Transfer to New Useful Tasks 10
1.3 Fundamental Motion Benchmarks for Robot Soccer 11
1.3.1 Striking the Ball 11
1.3.2 Passing the Ball to Another Robot 12
1.3.3 Striking a Moving Ball 12
1.3.4 Passing a Moving Ball to a Moving Robot 12
1.3.5 Dribbling the Ball Past Obstacles 12
1.4 Categories of Robot Soccer 13
1.4.1 MiroSoT and NaroSoT 14
1.4.2 SimuroSoT 15
1.4.3 RoboSoT and KheperaSoT 16
1.4.4 HuroSoT 18
1.5 The MiroSoT Robot Soccer System 19
1.6 Classification of MiroSoT Robot Soccer Systems 22
1.6.1 Command-Based Robots 23
1.6.2 Action-Based Robots 24

1.6.3 Intelligence-Based Robots 25
1.7 Purpose of This Book 25
Notes on Selected References 26

2. Robot Soccer System: Hardware and Firmware Components 27
2.1 Introduction 27
2.2 Mobile Robots 27
2.2.1 Mechanical Movement Mechanisms 27
2.2.2 Kinematics of a Two-Wheel Robot 31
2.2.3 Basic Motion Control: A Circular Path Analysis 34
2.3 A Two-Wheel Command-Based Soccer Robot 36
2.3.1 Microcontroller 37
2.3.2 DC Motors and Auxiliary Components 45
2.3.3 Motor Driving and Circuits 52
2.3.4 Velocity and Duty Cycle 54
2.3.5 Communication 57
2.3.6 Power System 66
2.3.7 Other Considerations 69
Notes on Selected References 69

3. How to Sense? Use Computer Vision Techniques 71
3.1 Introduction 71
3.2 Vision Basics 72
3.2.1 Computer Vision 72
3.2.2 Vision System Operations 73
3.2.3 Sampling, Pixel, and Quantization 74
3.2.4 Gray Scale, Binary, and Colour Images 74
3.2.5 Colour Models 75
3.3 Binary Image Processing 77
3.3.1 Thresholding 78
3.3.2 Computing Geometric Properties 80
3.3.3 Labelling 84
3.3.4 Labelling Algorithm 1: Recursive 87
3.3.5 Labelling Algorithm 2: Sequential, 4-Connectivity 87
3.3.6 Size Filtering 89
3.4 Vision System For MiroSoT Robot Soccer 90
3.4.1 System Processing 90
3.4.2 Image and Physical Coordinates on MiroSoT Playground 91
3.4.3 Example 1: System Hardware 93
3.4.4 Example 2: Vision Processing 94
3.4.5 Example 3: Information Extraction 97
3.4.6 Example 4: Window Tracking for Fast Vision Processing 100
Notes on Selected References 101

4. How to Decide and Act?
Use Intelligent Systems and Control Techniques 103
4.1 Introduction 103
4.2 Hybrid Control Architecture 104
4.2.1 Role Level: The Who Issue 105
4.2.2 Action Level: The What Issue 105
4.2.3 Behaviour Level: The How Issue 106
4.2.4 Execution Level: The Motion Issue 107
4.3 Example Strategies 107
4.3.1 Action-Level Strategy 108
4.3.2 Role-Level Strategy 109
4.4 Design of Robot Soccer Actions 109
4.4.1 Base Class: Primitive 110
4.4.2 Attacker Class: Shoot 111
4.4.3 Defender Class: Push 114
4.4.4 Goalkeeper Class: Block 115
4.5 Control Basics 117
4.5.1 PID Control 117
4.6 Unified Navigation Control 120
4.6.1 Control of a Two-Wheel Robot 121
4.6.2 Univector Field Method 126
4.6.3 Limit-Cycle Method 130
Notes on Selected References 140

5. How to Improve Intelligence?
Use Soft Computing Techniques 141
5.1 Introduction 141
5.2 Intelligence Basics 141
5.2.1 Search and Evolution 142
5.2.2 Knowledge Representation and Inference 143
5.2.3 Learning and Adaptation 144
5.3 Petri Nets 146
5.3.1 Petri Net Structure and Graph 146
5.3.2 Petri Net Markings 147
5.3.3 Rules for Petri Net Execution 149
5.3.4 Example 1: Role Level 149
5.3.5 Example 2: Action Level 153
5.4 Q-Learning: A Model-Free Reinforcement Learning Method . . 159
5.4.1 Standard Reinforcement Learning 161
5.4.2 Q-Learning 162
5.4.3 Example 1: Role Level 164
5.5 Neural Networks 167
5.5.1 A Simple Neuron 167
5.5.2 Neural Network Structure 169
5.5.3 Learning 170

5.5.4 Example 2: Action Level 170
5.6 Evolutionary Programming 178
5.6.1 The EP Process 179
5.6.2 EP and GAs 182
5.6.3 EP and ES 182
5.6.4 Example: Behaviour Level 183
5.7 Fuzzy Logic and Control 188
5.7.1 Fuzzy Logic 189
5.7.2 A Fuzzy Controller 189
5.7.3 Fuzzy Inference 191
5.7.4 Defuzzification 193
5.7.5 Example: Behaviour Level 194
Notes on Selected References 204

6. Robot Soccer System: Software Components and Programming 205
6.1 Introduction 205
6.2 MiroSoT Host Software Model 206
6.2.1 Modular Software 206
6.2.2 Modular Design Rules 206
6.3 Programming Framework: An Overview 208
6.4 Basic Skill Functions 211
6.4.1 `Velocity()` 211
6.4.2 `Angle()` 213
6.4.3 `Position()` 215
6.4.4 `Shoot()` 221
6.5 Applied Skill Functions 227
6.5.1 `Kick()` 227
6.5.2 `Goalie()` 236
6.5.3 `AvoidBound()` 242
6.6 Game Strategy Development 247
6.6.1 Zone-Defence Strategy 247
6.6.2 Univector Field Navigation 250
6.6.3 Limit-Cycle Navigation 254
Notes on Selected References 256

7. Simulated Robot Soccer 257
7.1 Introduction 257
7.2 Client-Server Architecture 257
7.2.1 Server Side 258
7.2.2 Client Side 259
7.3 Kinematics Models 260
7.3.1 For the Ball 260
7.3.2 For the Robot 260
7.4 How To Run the Simulator 261

7.4.1 System Requirements 261
7.4.2 Server Program 262
7.4.3 Client Program 263
7.5 Client Programming 264
7.5.1 Basic Program Structure 264
7.5.2 Attack Direction 265
7.5.3 System-Defined Variables and Constants 265
7.5.4 `Velocity()` and `Position()` Functions 266
7.5.5 Example Game Strategy Programs: `NormalGame()` 269
Notes on Selected References 271

Appendix 273

A. Programming the PIC16C73/73A Microcontroller 273
A.1 On-chip PWM Programming for Robot Motion Control 273
A.2 On-chip USART Programming for Robot Communication . . . 276

B. Reference Manual for an Experimental MiroSoT System . . 283
B.1 Vision System: Set-up and Initialization 283
B.1.1 Build Program Executable Code 283
B.1.2 Run the Program 285
B.1.3 Set Camera Image 285
B.1.4 Set Ball Colour 288
B.1.5 Set Robot Team ID Colour 291
B.1.6 Set Robot ID Colour 293
B.1.7 Set Playground Boundary 293
B.1.8 Set Robot Size 294
B.1.9 Set Pixel Size 294
B.1.10 Save Vision Settings 295
B.1.11 Open Vision Settings File 295
B.1.12 Set Auto Colours 295
B.1.13 Change Colour 300
B.2 System Library 302
B.2.1 The **SENSE** Category 302
B.2.2 The Communication Category 312
B.2.3 The **DECIDE**-and-**ACT** Category 314

References 321

Index 325

List of Figures

1.1 Hardware entities of a robot soccer team 8
1.2 Basic set-up for the 'dribbling the ball past obstacles' test 13
1.3 FIRA robot soccer: off-board/centralized vision categories 15
1.4 FIRA robot soccer: simulation category 16
1.5 FIRA robot soccer: onboard/distributed vision categories 17
1.6 FIRA robot soccer: humanoid category 19
1.7 General set-up for robot soccer (MiroSoT Category) 20
1.8 A general **SENSE-DECIDE-ACT** block diagram for robot soccer 22
1.9 Command-based robot soccer system 23
1.10 Action-based robot soccer system 24
1.11 Intelligence-based robot soccer system 25

2.1 Different types of 'move by rolling' mechanism for a mobile robot . 28
2.2 The robot's posture 29
2.3 Different wheel assemblies resulting in different ICR-axes 30
2.4 Kinematics of a robot 32
2.5 Computation of the robot's unique ICR 33
2.6 Circular path and angle of turning 34
2.7 Rotational velocity profile of the two wheels 35
2.8 Hardware architecture of a soccer robot 36
2.9 Harware of a soccer robot 37
2.10 Normal operational sequence of a microcontroller 38
2.11 Handling an interrupt request 39
2.12 Supporting architecture for a microcontroller 39
2.13 Scheme separating address and data signals 40
2.14 Input-output of a chip selector 41
2.15 An example CS equation 41
2.16 A system address map and corresponding CS equations 42
2.17 Selecting the clock frequency 42
2.18 Device interfacing 43
2.19 Pin Configuration of PIC16C73/73A microcontroller 44
2.20 A DC motor 'package' and its exploded view 45
2.21 Working principle of a DC motor 46
2.22 Rotational velocity and current versus torque 47
2.23 Torque about the wheel axis 47

2.24 Two mechanical designs of motor-wheel assembly 50
2.25 H-Bridge circuit for motor driving 53
2.26 Amplified PWM signals with different duty cycles 54
2.27 PWM-based operations of H-bridge circuit 54
2.28 Graph showing the relationship between the wheel rotational velocity $\bar{\omega}_G$ and the PWM data W^+ required to attain it 58
2.29 IR communication using ASK and IrDA1.0 methods 59
2.30 Module block diagrams implementing the base band method for IR communication 60
2.31 Generic module for IR base band communication 61
2.32 A game set-up for teams using IR base band communication 61
2.33 IR transmission coverage 62
2.34 Circuit block diagram of a transceiver 63
2.35 Communication message format 63
2.36 RF communication using the FSK method 64
2.37 An ALLINTEK ARFM-424 RF communication module 66
2.38 An ALLINTEK RF transceiver circuit for use by the host computer 67
2.39 An RF transceiver circuit for use by each team robot (within dotted lines) 67
2.40 Communication message format 68
2.41 Examples of power regulation IC chips 70

3.1 Basic architecture of a computer vision system 73
3.2 An $n \times m$ image grid 75
3.3 The RGB colour cube 76
3.4 A gray level image and its resulting binary images using different thresholds 79
3.5 An image and its histogram 80
3.6 An example showing the centre position $(\bar{x}, \bar{y})$ of an image 81
3.7 Finding the orientation of the object 82
3.8 A binary image and its labelled connected components 85
3.9 The 4- and 8-neighbourhoods of a pixel at square position $[i, j]$ . . . 86
3.10 Examples of a 4-path and an 8-path 86
3.11 An example illustrating the workings of the sequential connected components algorithm on an image 89
3.12 A noisy binary image and its resulting image after application of a size filter ($A^f = 8$) 90
3.13 A high-level flow chart showing vision processing as a software component of a robot soccer host-system program 92
3.14 On mapping the image and physical coordinate points in the playground 93
3.15 Frame grabber (Media camp 7 plus) 94
3.16 Example 1: Labelled components image 96
3.17 A robot's colour patch layout 97
3.18 Computing a robot's posture 98

3.19 Basic window tracking 101

4.1 A hybrid control architecture for robot soccer (MiroSoT Category) 105
4.2 Situational problems encountered by role-level assigner 107
4.3 Basis areas of manoeuvre for zone defence strategy (Small League MiroSoT Category) 108
4.4 Formations for Middle League MiroSoT 110
4.5 Wandering action 111
4.6 SweepBall action 112
4.7 Shoot action 112
4.8 Cannon_Shoot action 113
4.9 Position_To_Shoot action 113
4.10 PushBall action 114
4.11 Position_To_PushBall action 115
4.12 BlockBall action 116
4.13 Block diagram representation of a PID controller 118
4.14 Robot soccer situation: A robot (in white) should kick the ball (round) avoiding a opponent robot (in grey) 120
4.15 Robot modeling 122
4.16 Available velocity region 124
4.17 Component forces for generating a potential field 127
4.18 A potential field 127
4.19 A univector field 129
4.20 Univector field for obstacle avoidance by a point object 129
4.21 Modified univector field for obstacle avoidance by a robot while moving towards a target point g 130
4.22 Phase portrait of a limit-cycle 132
4.23 Navigation using the limit-cycle method 133
4.24 Multiple obstacle situation 134
4.25 Decision of rotational direction 135
4.26 Navigation example 137
4.27 Local minima with two overlapped obstacles 138
4.28 Extended navigation method 138
4.29 Robot soccer example 139

5.1 A Petri net structure and its graph 148
5.2 A Petri-net graph for role assignment supervision 151
5.3 A Petri-net supervisor for role assignment: transition firings and token redistributions 151
5.4 Role selection: who should attack? 152
5.5 A role-action structure for Petri net supervision and control 153
5.6 Four key situations for the defending robot 154
5.7 A Petri-net graph for defending robot control 156
5.8 Predictor of target point of the ball. 157
5.9 A Petri-net graph for goalkeeping robot control 159

5.10 A Petri-net control for robot goalkeeping: a transition firing and token redistributions 160
5.11 The standard reinforcement-learning model 161
5.12 States of the attacking robot 164
5.13 States of the defending robot 165
5.14 States of the ball 165
5.15 A Q-learning state $(r_{a3}, \theta_1, r_{d2}, b_5)$ of a role-level strategy for robot soccer 166
5.16 Schematic diagram of a simple neuron 168
5.17 Two sigmoid functions, with $\lambda_1 = 0, \lambda_2 = 1$ 168
5.18 A feedforward neural network 169
5.19 Structure of the proposed ASM 170
5.20 The process of selecting $a^j_{i\_fs}$ 174
5.21 A one-a-side MiroSoT game 175
5.22 Four situation variables characterizing ball possession 176
5.23 Four situation variables representing the team (or home) robot's winning score against the opponent robot and the risk level of conceding a goal 176
5.24 Structure of the feedforward neural network 177
5.25 Pseudocode of algorithm EP 180
5.26 Grid net of the function approximator 184
5.27 Simple example of grid net 185
5.28 Illustration of membership functions for fuzzy values. The upper diagram shows the membership functions of *cold*, *moderate*, and *hot*. The middle diagram shows the membership functions for *cold and moderate*; the lower diagram shows the membership functions for *cold or moderate*. 190
5.29 A fuzzy PD controller 190
5.30 Illustration of fuzzy inference with two rules using the min-max rule. 193
5.31 Shooting from the left side when the line connecting the ball to the opponent goal is on the right 194
5.32 Variables for relative posture characterization 195
5.33 Overall fuzzy control structure for the Shoot action 196
5.34 Desired output 197
5.35 Membership functions 198
5.36 θ_d sampled at each input region in the vicinity of the ball at $(0, 0)$ 199
5.37 Obstacle avoidance scheme 199
5.38 Membership functions 200
5.39 Membership functions 201
5.40 FLC for obstacle block 202
5.41 Membership functions 203

6.1 Host software model for a MiroSoT team 207
6.2 Overall program structure 208

6.3 `My_Strategy()` 209
6.4 Angle or turning control 214
6.5 Position control 216
6.6 Problem of oscillation about $\theta_e = \pm 90°$ with `Position()` 218
6.7 Graph of V_c against d_e 219
6.8 Different paths of the robot for shooting the ball in different desired directions 222
6.9 Geometric relationships for calculating the robot's desired heading angle θ_d at its online position (x, y) 223
6.10 Problem of chattering with `Shoot()` 225
6.11 A modified univector field for solving the problem of chattering .. 226
6.12 A strategy for the `Kick()` function 228
6.13 Mapping playground subareas to desired directions of ball kick ... 229
6.14 State S0: the desired point (`pos[0], pos[1]`) behind the ball the specified robot should move to. 231
6.15 In state S0: to switch to state S1 if the robot is at less than a distance of D_c from the desired point (`pos[0], pos[1]`). Note that in this figure, the illustrated C0-condition '$\sqrt{dx' \times dx' + dy' \times dy'} < D_c$' is true. 231
6.16 In state S1: to switch to state S2 if the angle error is less than a specified value A_d. Note that in this figure, the illustrated C1-condition '$|\theta_e| < A_d$' is false. 233
6.17 State S2: the desired point (`pos[0], pos[1]`) the specified robot should reach in order to move through the ball's position at a certain positive velocity. 234
6.18 In state S2: to switch to state S0 if the robot is at less than a distance of D_f from the ball, or is behind the ball. 235
6.19 Areas of surveillance for the `Goalie()` function 236
6.20 Desired position (`estimate_x, estimate_y`) of goalkeeping robot when the ball is in far-distance area, with $\triangle y = \frac{dy}{dx} \cdot \triangle x$, where $dy =$ `PositionOfBall`$[1] - 130/2$ and $dx =$ `PositionOfBall`$[0] - 15$. 237
6.21 Desired position (`estimate_x, estimate_y`) of goalkeeping robot when the ball is in middle-distance area 237
6.22 Desired position (`estimate_x, estimate_y`) of goalkeeping robot when the ball is in near-distance area 238
6.23 The two Y-coordinate boundaries of the goalkeeping robot 239
6.24 Numbered exception-situations and the desired turning directions for the goalkeeping robot 240
6.25 Labelled exception-situations characterized by specified bounds (`DISTANCE_BOUND, ANGLE_BOUND`) and wall locations, namely top and bottom walls, left and right walls 243
6.26 Team robots' assigned areas according to the zone-defence strategy 248

7.1 Client-server platform for SimuroSoT 258

7.2 Internal architectue of client-server based simulator for robot soccer programming . . . 258
7.3 SimuroSoT simulation display . . . 262
7.4 SimuroSoT client interface . . . 263
7.5 SimuroSoT client connection window . . . 264

A.1 Using CCP in PWM mode for motor driving and USART for data reception . . . 273
A.2 Switch for assigning Team ID . . . 281

B.1 Visual C++ window environment . . . 284
B.2 MiroSoT system user-interface . . . 284
B.3 Colour-setting buttons . . . 285
B.4 Default colour setting . . . 285
B.5 Colour parameter-setting window . . . 286
B.6 Default camera image . . . 286
B.7 Adjusting Brightness . . . 287
B.8 Adjusting Hue . . . 287
B.9 Adjusting Saturation . . . 288
B.10 Setting ball colour . . . 289
B.11 Ball colour control Window . . . 290
B.12 Well-set ball colour . . . 290
B.13 Selecting a robot on the real-image screen . . . 292
B.14 Setting a robot team ID colour . . . 292
B.15 Setting robot ID colour . . . 293
B.16 Setting boundary of playround . . . 294
B.17 Setting robot size . . . 295
B.18 Setting the pixel size of colours with the **Set Pixel Size** button . . . 296
B.19 Saving settings with [**Save Vision File**] . . . 296
B.20 Importing settings with [**Open Vision File**] . . . 297
B.21 Zooming in after clicking on **Auto Set Colour** button . . . 297
B.22 Setting the team ID colour with auto colour setting . . . 298
B.23 Setting robot ID colour with **Auto Colour** Setting . . . 299
B.24 Setting the ID number for a robot . . . 299
B.25 Robot colour 2 . . . 300
B.26 Menu for changing colours . . . 301
B.27 Change colour parameter . . . 301
B.28 Display function . . . 302
B.29 Set-colour function . . . 304
B.30 Vision function . . . 307
B.31 Communication function . . . 313

List of Tables

1.1 Robot primitives defined in terms of inputs and outputs 3
1.2 Robot primitives and system entities for robot soccer (MiroSoT Category) 21

2.1 Some motor and gear characteristics (source: datasheets) 51
2.2 Pin Configuration of ALLINTEK ARFM-424 module 65
2.3 Some batteries and their characteristics 68

4.1 Robot primitives and hierarchy levels defined for robot soccer (MiroSoT Category) 106

5.1 Algorithm EP for offline training of univector fields 187
5.2 Representation of the fuzzy PD controller as a table 191
5.3 Rules for the destination block 197
5.4 Rules for the obstacle block, FLC1 (left) and FLC2 (right) 200
5.5 Rules for right wheel 202

6.1 Program functions for robot soccer system (MiroSoT category) . . . 210