Evolutionary Learning of Control and Strategies in Robot Soccer

Peter James Thomas, B.Eng(Hons)

A thesis submitted for the Degree of Doctor of Philosophy

Central Queensland University

James Goldston Faculty of Engineering and Physical Systems

School of Advanced Technology and Processes

25^{th} July 2003

Abstract

Robot soccer provides a fertile environment for the development of artificial intelligence techniques. Robot controls require high speed lower level reactive layers as well as higher level deliberative functions.

This thesis focuses on a number of aspects in the robot soccer arena. Topics covered include boundary avoidance strategies, vision detection and the application of evolutionary learning to find fuzzy controllers for the control of mobile robot.

A three input, two output controller using two angles and a distance as the input and producing two wheel velocity outputs, was developed using evolutionary learning. Current wheel velocities were excluded from the input. The controller produced was a coarse control permitting only either forward or reverse facing impact with the ball. A five input controller was developed which expanded upon the three input model by including the current wheel velocities as inputs. The controller allowed both forward and reverse facing impacts with the ball.

A five input hierarchical three layer model was developed to reduce the number of rules to be learnt by an evolutionary algorithm. Its performance was the same as the five input model.

Fuzzy clustering of evolved paths was limited by the information available from the paths. The information was sparse in many areas and did not produce a controller that could be used to control the robots.

Research was also conducted on the derivation of simple obstacle avoidance strategies for robot soccer. A new decision region method for colour detection in the UV colour map to enable better detection of the robots using an overhead vision system. Experimental observations are given.

Publications

1. P. J. Thomas & R. J. Stonier, Evolutionary Learning of a 5I2O Fuzzy Controller Including Wheel Lift Constraint, *IEEE The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems* (CIRAS2003), Singapore, 2003.

2. P. J. Thomas & R. J. Stonier, Hierarchical Fuzzy Control in Robot Soccer using Evolving Algorithms, *IEEE Congress on Evolutionary Computation* (CEC2003), Canberra, Australia, pp. 2434–2440, 2003.

3. P. J. Thomas & R. J. Stonier, Using Evolved Paths for Control in Robot-Soccer, *Computational Intelligence in Robotics and Automation* (CIRA-2003), Kobe, Japan, pp. 735–740, 2003.

4. P. J. Thomas & R. J. Stonier, Fuzzy Control in Robot-Soccer, Evolutionary Learning in the First Layer of Control, Journal of Systemics, Cybernetics and Informatics, Vol. 1, No. 1, 2003.

5. P. J. Thomas & R. J. Stonier, Evolutionary Learning of Fuzzy Control in Robot-Soccer, Book Chapter in M. Mohammadian (ed) *Computational Intelligence in Control*, Idea Publishing, USA, pp. 88-103, 2003.

6. P. J. Thomas & R. J. Stonier & P. J. Wolfs, Robustness of Colour Detection for Robot Soccer, *The Seventh International Conference on Control,*

Automation, Robotics and Vision (ICARCV'2002), Singapore, 2002.

7. P. J. Thomas & R. J. Stonier, Fuzzy Control in Robot-Soccer, Evolutionary Learning in the First Layer of Control, *Proceedings of the 6th World Multi-conference on Systemics, Cybernetics and Informatics* (SCI'2002), Florida, July, Vol. XII, pp. 181-186, 2002.

8. P. J. Thomas & R. J. Stonier, Evolutionary Learning of Fuzzy Control in Robot-Soccer, *Conference on Computational Intelligence for Modelling, Control and Automation* (CIMCA'2001), ISBN 0-85889847-0 on CD, M. Mohammadian (Ed), Las Vegas, pp 494-503, 2001.

9. X. Mao & R. J. Stonier & P. J. Thomas, An Effective Fuzzy Image Filter, *International Conference on Computational Intelligence for Modelling, Control and Automation* (CIMCA'2001), On CD, ISBN 0-85889847-0, M. Mohammadian (Ed), Las Vegas, pp 280-285, 2001.

10. P. J. Thomas & G. Springer & R. J. Stonier & P. J. Wolfs, Boundary-Avoidance and Attack Strategy in Robot-Soccer, *FIRA Robot World Cup France '98 Proceedings*, ISBN 89-7163-102-3, La Cité des Sciences et de l'Industrié Paris, France, pp 27-32, 1998.

Contents

1 Introduction **1**

1.1 Artificial Intelligence (AI) 2

1.2 Sensory Perception 3

1.3 Decision Making Techniques 4

1.4 Robot-Soccer Regulations 6

1.4.1 Communication 9

1.5 Software 9

1.6 Curvilinear Trajectory Control 9

1.7 Identification of Team Robot Colours 16

1.8 Towards More Intelligent Control Systems 19

1.9 Fuzzy Logic 21

1.9.1 Self-Organising Fuzzy Logic Controls 25

1.9.2 Hierarchical Fuzzy Logic Controllers 26

1.10 Evolutionary Learning of Fuzzy Control 28

1.10.1 Key Objectives 28

1.10.2 Evolutionary Algorithms 29

1.10.3 Learning the Knowledge Base 30

1.11 Scope 34

2 Fuzzy Logic **36**

2.1 Introduction 36

2.2 Review of Classical Set Theory 37

2.3 Fuzzy Sets 40

2.3.1 Zadeh Notation 43

2.3.2 Probability versus Fuzzy Sets 45

2.3.3 Some Basic Concepts 45

2.3.4 Basic Operations on Fuzzy Sets 47

2.3.5 Fuzzy Complement 49

2.3.6 Fuzzy Union: S-norm 50

2.3.7 Fuzzy Intersection: T-norm 52

2.4 Relations 53

2.5 Fuzzy relations 56

2.6 Projections and Cylindrical Extensions 58

2.7 Composition of Fuzzy Relations 64

2.8 Extension Principle 66

2.8.1 Linguistic Hedges 68

2.9 Fuzzy If-Then Rules 70

2.9.1 Fuzzy Propositions 73

2.9.2 Interpretation of Fuzzy If-Then Rules 77

2.9.3 Local and Global Implication 80

2.9.4 Mamdani Implications 81

2.10 Inference 83

2.10.1 Compositional Rule of Inference 85

2.11 Generalised Inference Rules 87

2.11.1 Generalised Modus Ponens 87

2.11.2 Generalised Modus Tollens 88

2.11.3 Generalised Hypothetical Syllogism 89

2.12 Fuzzy Rule Base and Fuzzy Inference Engine 91

2.13 Fuzzy Rule Base Structure 92

2.13.1 Properties of the Fuzzy Rule Base 93

2.14 Fuzzy Inference Engine 95

2.14.1 Compositional Based Inference 95

2.14.2 Individual Rule Based Inference 98

2.14.3 Some Inference Engines 99

2.15 Fuzzifiers and Defuzzifiers 104

2.16 Fuzzifiers 105

2.16.1 Singleton Fuzzifier 105

2.16.2 Gaussian Fuzzifier 107

2.16.3 Triangular Fuzzifier 111

2.17 Defuzzifiers 113

2.17.1 Centre Average Defuzzifier 114

2.17.2 Defuzzifier Comparison 117

3 Genetic/Evolutionary Algorithms 118

3.1 Introduction 118

3.2 Genetic Algorithms 121

3.2.1 Nomenclature 121

3.2.2 Data Structures 124

3.2.3 Selection 125

3.2.4 Crossover 126

3.2.5 Mutation 127

3.2.6 Elitism 128

3.2.7 Mapping Objective Functions to Fitness Coding 128

3.2.8 Genetic Algorithm Convergence 130

3.3 De Jong and Function Optimisation 133

3.4 Genetic Algorithms and Parallel Processors 136

3.5 Evolutionary Algorithms 140

3.6 Constraints 143

4 Design of Fuzzy Systems **146**

4.1 Introduction 146

4.2 Table Look-Up Scheme 146

4.3 Gradient Descent Training 147

4.4 Nearest Neighbourhood Clustering 151

4.5 Evolutionary Learning of Fuzzy Systems 152

4.5.1 Genetic Fuzzy Systems 152

4.6 Fuzzy Amalgamation 154

5 Robot Modelling **157**

5.1 Introduction 157

5.2 Robot Parameters 158

5.3 Kinematics 162

5.4 Constraints 164

5.4.1 Time Constraint 165

5.4.2 Boundary Constraint 165

5.4.3 Robot to Ball Impact Constraint 166

5.4.4 Wheel Lift Constraint 169

5.5 Cartesian Coordinate System 172

5.5.1 Fuzzy Control using Cartesian Coordinates 172

5.6 Relative Coordinate System 174

6 Boundary Avoidance **178**

6.1 Introduction 178

6.2 Basic Strategy Definitions 179

6.3 Boundary Avoidance 180

6.3.1 Stop and Turn 180

6.3.2 Artificial Potential Fields (APF) 181

6.3.3 Angle Limiting Functions 181

6.3.4 Proportional Cosine Control 184

6.4 Modified Attack Strategy 187

6.4.1 Strategy Implementation 189

6.5 Multi-Agent Strategies 190

6.6 Comments 191

7 Three Input Two Output (3I2O) Control 193

7.1 Introduction 193

7.2 Fuzzy Control System Design 194

7.3 Evolutionary Learning 197

7.4 Discussion 200

7.4.1 Results: Fuzzy Control at a Distance 201

7.4.2 Learning of Fuzzy Control Close to the Ball 204

7.5 Comments 206

8 Five Input Two Output (5I2O) 224

8.1 Introduction 224

8.2 Fuzzy Control System Design 226

8.3 Evolutionary Learning 229

8.4 Discussion 232

8.5 Comments 240

9 Hierarchical Modelling **249**

9.1 Introduction 249

9.2 Fuzzy Control System Design 251

9.2.1 First Fuzzy System 251

9.2.2 Second Fuzzy System 253

9.2.3 Third Fuzzy System 255

9.2.4 System Design 256

9.3 Evolutionary Learning 258

9.4 Discussion 259

9.5 Comments 268

10 Evolving Velocity Profiles **277**

10.1 Introduction 277

10.2 Evolutionary Learning of Paths 279

10.3 Discussion 284

10.4 Fuzzy Clustering 286

10.5 Comments 294

11 Colour Space **296**

11.1 Introduction 296

11.2 YUV 298

11.3 Colour Detection of the Ball 300

11.4 Pie Slice Decision Regions 301

11.5 Experimental Observations 305

11.6 Comments 305

12 Conclusions and future directions **307**

List of Figures

1.1 Fields of Artificial Intelligence [2] 3

1.2 Definition Categories of AI [2] 3

1.3 MIROSOT Configuration 1.3 7

1.4 MIROSOT Field Detail 1.3 8

1.5 Robot and Ball 8

1.6 Ball Attacking—Case 1 11

1.7 Position Error Caused by Vision Acquisition and Processing Delays 12

1.8 Robot Arc Ability 13

1.9 Ball Attack Strategy with Lead Transfer Function 14

1.10 Lead Transfer Function in Time Domain 14

1.11 Prediction Error caused by Position Errors 15

1.12 Boundary Avoidance by Piece-Wise Linear Profile 16

1.13 Basic Structure of a FLC 23

1.14 Basic Structure of a SOFLC 26

1.15 GA Fuzzy Rule Generator Architecture 26

2.1 Characteristic Functions for A', $A \cap B$ and $A \cup B$ given A and B 39

2.2 Membership Function for "Middle Age" 40

2.3 Possible Membership Functions "Close to or Equal to Zero" . . . 42

2.4 Centres of Fuzzy Sets A, B, C, and D 46

2.5 Fuzzy Membership of $A \cap B$, $A \cup B$ and A' 48

2.6 S-norms: (a) $s_{\max}(a, b) = \max(a, b)$, and (b) $s_{ab}(a, b) = a + b - ab$ 52

2.7 T-norms: $t_{\min}(a, b) = \min(a, b)$, and $t_{ap}(a, b) = ab$ 53

2.8 Unit Disc Projections and Extensions 59

2.9 Extension Principle: Ambiguity in Defining Membership $\mu_B(v)$ . . 68

2.10 Simulated Robot in Workspace 74

2.11 Truth Table for $p \wedge (p \rightarrow q) \rightarrow q$ 83

2.12 Truth Table for $\neg q \wedge (p \rightarrow q) \rightarrow \neg p$ 84

2.13 Truth Table for $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ 85

2.14 Components of a Pure Fuzzy System 91

2.15 A MIMO System Decomposed into MISO Systems 92

2.16 Membership Functions with $\mu_{A_p}(u_p^*) = 0.7$ 102

2.17 Membership Functions with $\mu_{A_p}(u_p^*) = 0.3$ 103

2.18 Components of Fuzzy System with Fuzzifier and Defuzzifier 104

2.19 Graphical Illustration of the Centre of Gravity Defuzzifier 114

2.20 Graphical Illustration of the Centre Average Defuzzifier 115

2.21 Illustration of Continuity Difficulties with the Maximum Defuzzifier116

2.22 Example Illustrating Problem with the Maximum Defuzzifier . . . 117

3.1 Basic Genetic Algorithm Structure 123

3.2 One-Point Crossover 126

3.3 Two-Point Crossover 127

3.4 GA with Full Replacement Policy 130

3.5 GA using Parallel Section for Fitness Evaluation 138

3.6 Parallel GA's Solving Velocity Profiles 139

5.1 Curvilinear Formulae Symbols 163

5.2 Ball Impact Symbols 166

5.3 Wheel Lift Diagram 169

5.4 Wheel Lift Diagram with $L = 68.5mm$ and $h = 37mm$ 171

5.5 Wheel Lift Diagram with $L = 75mm$ and $h = 12mm$ 171

5.6 Membership Sets for Cartesian Fuzzy System 173

5.7 Relative Coordinate Parameters 175

5.8 Robot with relative coordinate system at centre 175

6.1 Modules and dependencies 179

6.2 Piece-Wise Linear Profile 182

6.3 Boundary Avoidance 183

6.4 Cosine Control 185

6.5 Cosine Control Path 185

6.6 Ball Attack Strategy 188

6.7 Quadrant Division of Field 191

7.1 Three Input Two Output Model 194

7.2 Fuzzy Input Sets 196

7.3 All Medium Distance Paths 202

7.4 Evolutionary Statistics Average of Minimums over 11 runs 205

7.5 Evolutionary Statistics Average of Averages over 11 runs 206

7.6 All Short Distance Paths 207

7.7 Medium Distance Paths: c077–c104 210

7.8 Medium Distance Paths: c105–c125 211

7.9 Medium Distance Velocity Profiles: c077–c083 212

7.10 Medium Distance Velocity Profiles: c084–c090 213

7.11 Medium Distance Velocity Profiles: c091–c097 214

7.12 Medium Distance Velocity Profiles: c098–c104 215

7.13 Medium Distance Velocity Profiles: c105–c111 216

7.14 Medium Distance Velocity Profiles: c112–c118 217

7.15 Medium Distance Velocity Profiles: c119–c125 218

7.16 Individual short distance paths 219

7.17 Short Distance Velocity Profiles: c000–c006 220

7.18 Short Distance Velocity Profiles: c007–c013 221

7.19 Short Distance Velocity Profiles: c014–c020 222

7.20 Short Distance Velocity Profiles: c021–c027 223

8.1 Five Input Two Output Model 225

8.2 Fuzzy Input Sets 226

8.3 Average of averages over 11 runs for 50 000 generations 233

8.4 End point statistics of $\alpha_1 T_1$ term 234

8.5 End point statistics of $\alpha_2 T_2$ term 235

8.6 Long Distance Path from Left 236

8.7 Medium Distance Path from Above 237

8.8 Medium Distance Path from Bottom Right 237

8.9 Long Distance Path from Right 238

8.10 Table of Results c0056–c0083 242

8.11 Table of Results c0560–c0587 243

8.12 Table of Results c1148–c1175 244

8.13 Table of Results c2072–c2099 245

8.14 Table of Results c2324–c2351 246

8.15 Table of Results c2968–c2995 247

8.16 Table of Results c3864–c3891 248

9.1 Hierarchical Model 250

9.2 Membership Sets Defined for the First Hierarchical Layer 252

9.3 Membership Sets Defined for the Second Hierarchical Layer 254

9.4 Membership Sets Defined for the Third Hierarchical Layer 256

9.5 Average of averages over 11 runs for 50 000 generations 260

9.6 End point statistics of $\alpha_1 T_1$ term 261

9.7 End point statistics of $\alpha_2 T_2$ term 262

9.8 Long Distance Path from Left 263

9.9 Medium Distance Path from Above 264

9.10 Long Distance Path from Bottom Right 265

9.11 Long Distance Path from Right 265

9.12 Table of Results c0056–c0083 270

9.13 Table of Results c0560–c0587 271

9.14 Table of Results c1148–c1175 272

9.15 Table of Results c2072–c2099 273

9.16 Table of Results c2324–c2351 274

9.17 Table of Results c2968–c2995 275

9.18 Table of Results c3864–c3891 276

10.1 Path Plots from Configuration c08400(550, 550, 7π/9, 0, 0) 287

10.2 Path Plots from Configuration c08490(550, 650, 0, 54, 54) 287

10.3 Path Plots from Configuration c08900(550, 750, 4π/9, 27, 0) . . . 288

10.4 Path Plots from Configuration c09187(550, 850, 2π/9, -27, 0) . . . 289

10.5 Path Plots from Configuration c11587(650, 550, 16π/9, 78, 78) . . 290

10.6 Path Plots from Configuration c14587(750, 550, 4π/3, 103, -103) . 291

10.7 Path Plots from Configuration c14687(750, 750, 0, 56, 78) 292

10.8 Path Plots from Configuration c16887(850, 450, 2π/3, -78, 56) . . 293

10.9 Path Plots from Configuration c17087(850, 550, 0, -27, 54) 294

10.10Path Plots from Configuration c17287(850, 550, 10π/9, 54, 0) . . 295

10.11Path Plots from Configuration c18099(850, 850, $2\pi/9$, -54, -27) . . 295

11.1 RGB Colours Mapped on UV Coordinates 299

11.2 Averaged Pixels caused by Red-Yellow Boundary 300

11.3 Averaged Colours on UV Map 301

11.4 RGB Rectangle in YUV Space 302

11.5 UV Rectangle over YUV 303

11.6 Pie Segment 304

11.7 Screen Capture of Window using Pie Segment 306

12.1 Three Input Two Output Model 310

12.2 Five Input Two Output Model 311

12.3 Hierarchical Model 312

List of Tables

2.1 Fuzzy Identities [3] 49

2.2 Truth Table for $p \rightarrow q$ 77

2.3 Truth Table for $\neg p \wedge q$ and $(p \wedge q) \vee \neg p$ 78

2.4 Intuitive Reasoning in Generalised Modus Ponens 87

2.5 Intuitive Reasoning in Generalised Modus Tollens 89

2.6 Intuitive Reasoning in Generalised Hypothetical Syllogism 90

2.7 Comparing Defuzzifiers to their Criteria 117

3.1 Biological and GA Nomenclature Comparison [4] 122

3.2 Local Maxima of $f_1(x) = 1 - \sin(\pi x)\cos(11\pi x), x \in (0, 1)$ 124

5.1 Experimental Results of Robot Parameter Test 159

5.2 Angular robot error using $\overline{BG}$ and acceptable ball to goal trajectory angle 177

7.1 Medium Path Final Robot Statistics 208

7.2 Medium Path Final Robot Statistics (continued) 209

7.3 Short Paths Robot Statistics 209

8.1 Statistics of 5I2O fuzzy controller 234

9.1 Statistics of Hierarchal fuzzy controller 260

11.1 RGB Colour Mapping onto YUV Coordinates 299

List of Symbols

$\emptyset$	empty set
$\subset$	subset
$\supset$	superset
$\subseteq$	subset or equal
$\supseteq$	superset or equal
$\cap$	intersection
$\cup$	union
$\mid$	such that
$\prime$	compliment
$\forall$	for all
$\exists$	there exists
ℓ	length
$\in$	belongs to or is a member of
$\notin$	is **not** an element of
μ	degree of membership
$\overline{x}$	mean
σ	standard deviation
$\wedge$	and
$\vee$	or
$\neg$	not
Z	set of Natural numbers
$\rightarrow$	implies or mapping
$\leftrightarrow$	equivalent
b	bit
B	byte
ϕ	angle
x	x-coordinate or independent variable
y	y-coordinate
m	gradient
r	radius
P	population
t	time
U	universe

$\sum$	summation or union in Zadeh notation
$\int$	integral or continuous counterpart of $\sum$ in Zadeh notation
$\prod$	product
$\oplus$	bounded sum
$\ominus$	bounded difference
$\overline{A}$	not A
t	t-norm
s	s-norm
$\times$	cross product
$\circ$	composition
T	true
F	false
$\underset{\sim}{x}$	vector x
$\overline{x}$	average of x
∂	partial differential
i, j, k	index variables
v	linear or tangential velocity
ω	angular velocity
d	distance
m	mass
g	gravational constant
K_p	proportional gain
K_a	angular gain
Ω	angle
Δ	change in
F	fuzzifier
DF	defuzzifier

Acknowledgements

I would like to acknowledge the support of my supervisors: A/Professor P. J. Wolfs, School of Advanced Technology and Processes, and A/Professor R. J. Stonier, School of Mathematics and Decision Sciences; Central Queensland University. I would like to thank my wife, Sandra, for her support.

Declaration

This thesis contains no material that has been accepted for the award of another degree. Furthermore, to the best of my knowledge, this thesis does not contain any material previously published or written by another person, except where due reference is made in the text.

.....

P J Thomas

Chapter 1

Introduction

Robot soccer has been promoted as an environment that offers rich opportunities for the development of a broad spectrum of artificial intelligence and control system techniques. An international competition, based loosely on the concept of the Soccer World Cup, has been promoted for several years to encourage postgraduate study in these areas [1].

This thesis focuses on the author's experiences within this challenging field. While the thesis primarily focuses on evolutionary fuzzy controls, some useful contributions have also been made in robot vision. The author has also been responsible for the development of a competitive team, and fielding that team at international competitions. Some details relating the robot hardware systems will be included to give a holistic overview of the entire development.

The remainder of the chapter will briefly introduce artificial intelligence concepts, robot soccer, and outline the thesis. Chapter 2 gives a detailed overview of fuzzy logic systems. An overview of genetic and evolutionary algorithms are shown in Chapter 3. The use of genetic/evolutionary algorithms to learn a fuzzy rule base

is discussed in Chapter 4. Modelling of the soccer-robot characteristics is covered in Chapter 5. Chapters 6 to 10 are based on papers presented at conferences. A benchmark that describes the earliest work is presented in Chapter 6. Evolving a fuzzy rule base for a three input, two output (3I2O) robot path controller is shown in Chapter 7. A five input, two output (5I2O) fuzzy robot path controller is developed in Chapter 8. A five input, two output, three layer hierarchical fuzzy controller is developed in Chapter 9. The developing of a fuzzy rule base from evolutionary robot velocity profiles is presented in Chapter 10. The classification of robot colour patches from video composite input is shown in Chapter 11. Concluding remarks are given in Chapter 12.

The remainder of this chapter will give a compact overview of artificial intelligence, the issues pertinent to robot soccer and an overview of some of the concepts of fuzzy and evolutionary control.

1.1 Artificial Intelligence (AI)

Over the past 2000 years, philosophers have tried to understand and model truths or principles governing knowledge and being. Psychologists have provided tools to investigate the human mind and a scientific language to express the results. Mathematicians have provided formal theories of logic, probability, decision making and computation over the past 400 years. Linguists have provided theories of language structure and meaning. Computing technology has provided the means to apply AI concepts. Figure 1.1 shows some of the many sub-fields of AI.

AI [2] is an attempt to emulate human "intelligence", as illustrated in Figure 1.2, by thought processes or behaviour. These categories can also be broken into a

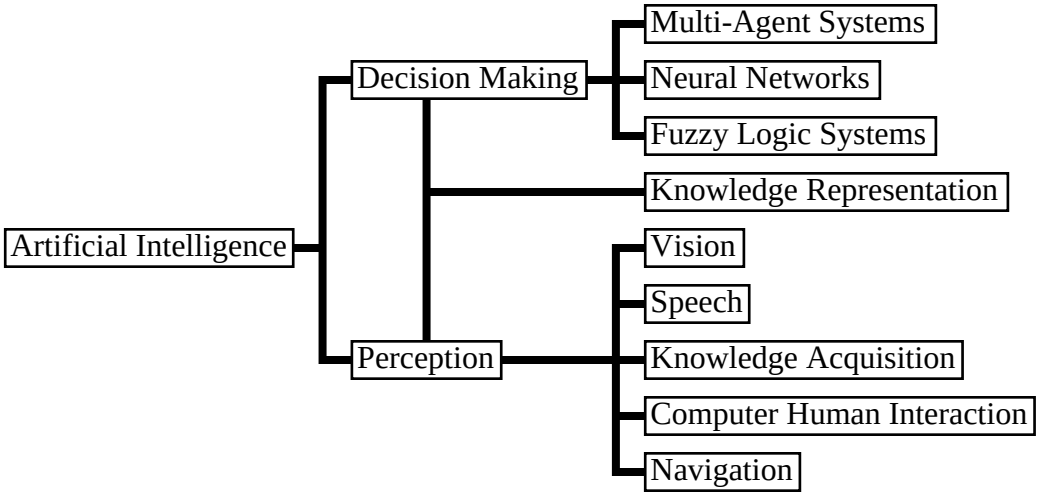

Figure 1.1: Fields of Artificial Intelligence [2]

measure of human performance and rationality. It is estimated that over 100 000 rules are required to simulate simple intelligent behaviour [5].

Systems that think like humans	Systems that think rationally
Systems that act like humans	Systems that act rationally

Figure 1.2: Definition Categories of AI [2]

AI often relies on perception of the environment through sensors. An overview of the available sensor technologies is now presented.

1.2 Sensory Perception

Lately, there is a growing trend of AI techniques used in real applications. In order for AI techniques to operate in real applications, measurements must be acquired from the environment. Some of the available sensors used to measure environmental quantities are:

- switches
- strain gauges
- optical sensors
- accelerometers
- gyroscopes
- proximity sensors
  - inductive devices
  - capacitive devices
  - resistance devices
  - magnetic sensors
- pressure sensors
  - piezoelectric
  - barometric devices
  - resistance devices
  - capacitive devices
- radar devices
- sonar devices
  - sound
  - light
- vision sensors.

A discussion of sensor devices is beyond the scope of this thesis and is covered extensively in a large number of books in AI, robotics, engineering texts and manufacturer data sheets.

1.3 Decision Making Techniques

Several AI programmes have been designed to solve specific problems. Game playing for example has received much attention because the rules are well defined and the environment of the game is well constrained. Simple AI [2] problems such as these can be solved using decision trees. Decision trees provide a hierarchy structure with all possible solutions from each node. Othello, tic-tac-toe and

draughts have been solved by decision trees. Another solution is to compute all possible outcomes from the current state of play.

Chess has a high combination of moves that prohibits the use of decision trees, but has simple calculations for moves. In 1997, Deep Blue became the first computer to beat a grand master level chess player, Kasparov. Chess playing programmes have been improving since then, and so have the grand master's abilities. Grand master's have developed methods that cause the search path for calculating the next move to be very long. This prevents the computer from being the outright winner to date. Computer chess programmes have their weakness and strengths as do human players. One of the methods used in developing intelligent systems such as playing chess is the expert system.

Expert systems [2] were developed to simulate an expert's knowledge of a problem. They consist of three major components: a knowledge base, a data base and a rule interpreter. The knowledge base contains the inference rules used during the reasoning process usually by if-then-else (Production Systems) coding. Reasoning is aided by propositional and predicate calculus. The data base contains the context or facts available at any given time, such as a mammal: has hair, gives milk. Hierarchical tree structures are commonly used to represent the problem context. The rule interpreter (inference engine) guides the reasoning process through the knowledge base by attempting to match the facts in the data base to the rule conditions, including resolving rule conflict.

Almost all real world problems are extremely non-linear. Linearisation has helped to simplify problems for modelling purposes. This technique simplifies the mathematics of the problem and in general allows only a small part of the problem to be solved.

Artificial Neural networks (ANNs) use structures that mimic the functioning of the brain [3, 6]. These are powerful tools, but issues of training are complex. Recent advancements in extracting rules from ANNs are shown in [7]. A similar and comparative non-linear system is fuzzy systems.

Fuzzy Logic Systems (FLSs) are based on the linguistic language that people use [5, 3, 8]. The spoken language is vague, but contains enough information for another person to interpret. Terms are used that cannot be given an exact number for a computer programme to use. An expression has different meaning in dissimilar environments.

This thesis will concentrate on the application of evolving a fuzzy system to control a soccer playing robot. The key features of fuzzy control will be developed in depth in subsequent chapters. An overview of the target hardware is now considered.

1.4 Robot-Soccer Regulations

A MIROSOT team was purchased from Micro Adventure to facilitate testing of AI techniques on a physical system in a competitive environment. The game is played by two opposing teams configured as Figure 1.3.

The CQU robot-soccer team has participated at competitions in Korea - joint team (1996), Korea - joint team (1997), Alaska (1998), Paris (1998) and Rockhampton (2000).

Details of up-to-date rules and regulations for the Mirosot competition may be

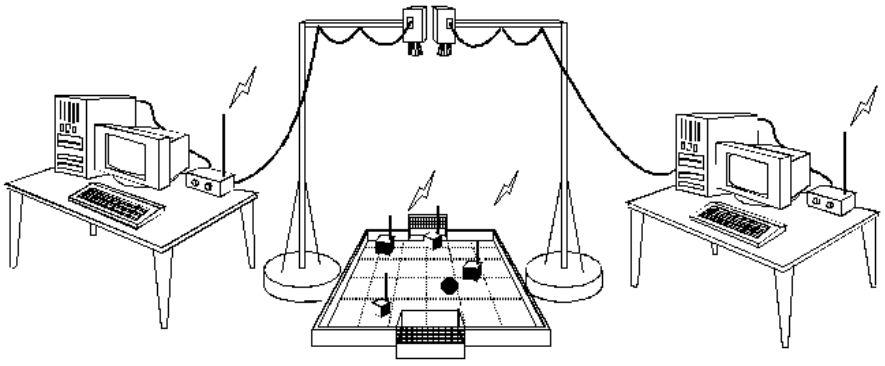

Figure 1.3: MIROSOT Configuration 1.3

found on the Federation of International Robot Soccer Association (FIRA) [1]. Some basic principles of the robot soccer game that were defined at the time the Central Queensland University team played in competition are now presented.

Robot soccer is played on a rectangular field with dimensions and marking as shown in Figure 1.4.

The basic control software purchased with the robots used simple strategies producing travel along straight line segments. The software would calculate a desired point to move to, rotate the robot to face that point, move off to that point in a straight line and stop. "Stop and Turn" algorithms such as this do not take momentum into consideration. As a consequence, the robot passes the stop position and often collides with another robots or the perimeter wall. Another disadvantage of "Stop and Turn" is loss of ball control by the robot. This motion control was replaced by a curvilinear trajectory control in the early development of the CQU team as described briefly in a following section.

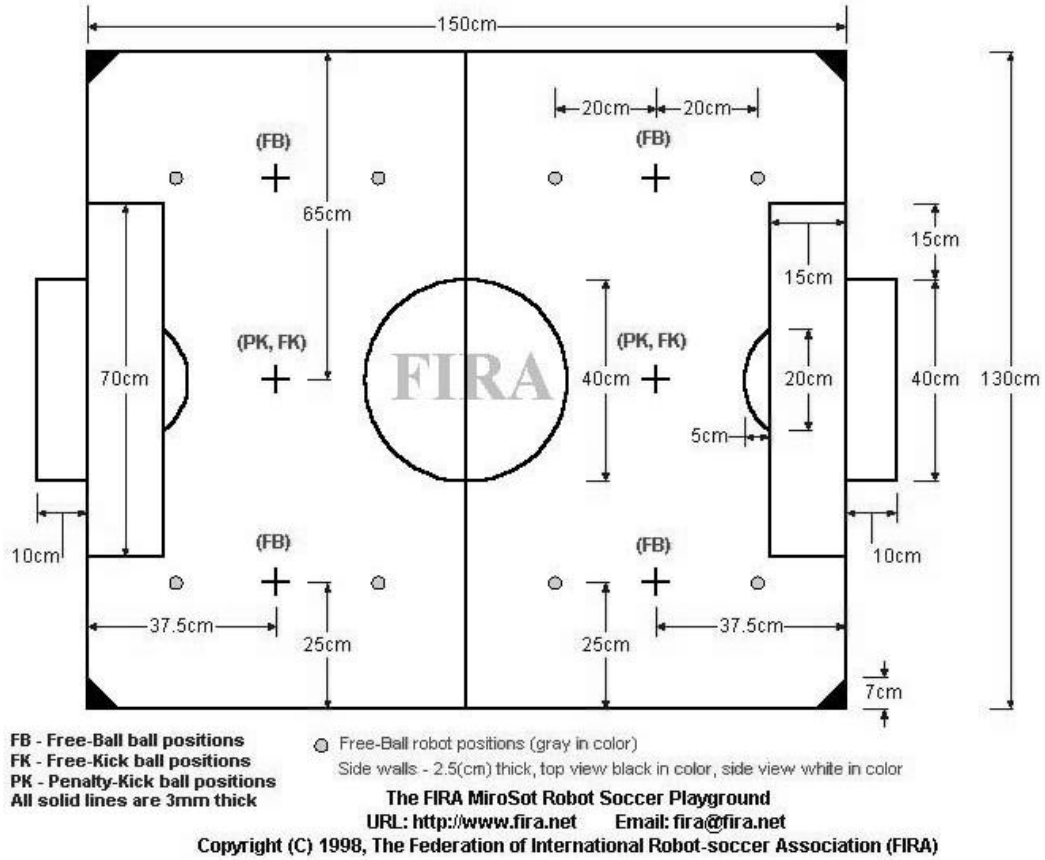

Figure 1.4: MIROSOT Field Detail 1.3

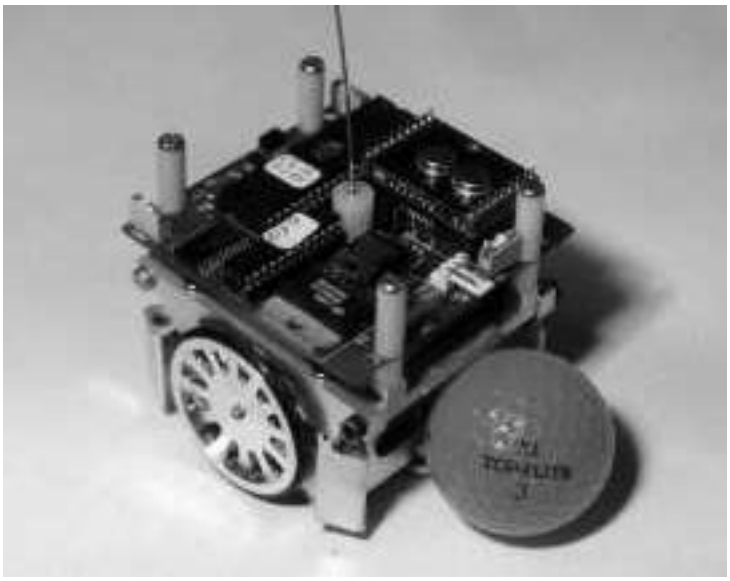

Figure 1.5: Robot and Ball

1.4.1 Communication

The current radio communications is by a commercially available Radiometrix module. The module is capable of transmitting 20kbps with Manchester encoding. Two bytes (velocity left and velocity right) are transmitted for each of three robots. Transmitting six bytes at 9600bps takes 6.25ms. This corresponds to the ball traversing 31mm at the peek speed of $5ms^{-1}$.

To reduce positional error at $5ms^{-1}$, the Radiometrix communication should be upgraded to a module capable of 60kbps.

1.5 Software

Control of the robots is performed by a centralised programme hosted on a computer. The basic control software purchased with the robots was written in Microsoft Visual C++ version 1.52. The programme was written using the Microsoft Foundation Class library (MFC) [15]. A full description is found in the Microsoft MFC library reference manuals [16, 17, 18, 19]. The use of the MFC library assumes a background knowledge in Application Programming Interface (API) programming [20].

1.6 Curvilinear Trajectory Control

At the Robot Soccer Competition at KAIST in Korea, 1996, [21], the Central Queensland University (CQU) team presented a paper which described a circular

path planning algorithm to attack the ball. The algorithm modified the actual direction vector of the robot to a desired heading vector by simple geometric calculations. Figure 1.6 illustrates this simple method of generating a circular trajectory. Except in the parallel case, every pair of tangent lines on a circle intersect at some point H. An isosceles triangle is formed with the tangent lines BH and RH. An important property of an isosceles triangle is that two angles $\angle HBR$ and $\angle HRB$ are equal. Let G represent the goal and B represent the ball. A line passing through GB is the desired direction for the robot R to move the ball towards the goal. The angle of the GB line (ϕ_1) is calculated by the arctangent of ($\Delta y_{BG}/\Delta x_{BG}$). The angle of the BR line (ϕ_2) is calculated by the arctangent of ($\Delta y_{RB}/\Delta x_{RB}$). Using the property of an isosceles triangle, the angle $\angle BRH$ is $\phi_1 + \phi_2$. The line BR is at angle ϕ_2 to the horizontal. Equation 1.1 gives the desired angle for the robot to follow a curved path to the ball and be lined up to run the ball straight into the goal. The centre of the curve can be calculated by the intersection of the perpendicular of the GB line at B and the perpendicular of the HR line at R. The centre of curvature is not required in the curvilinear calculation.

$$\phi_3 = 180^\circ - \phi_1 - 2\phi_2 \tag{1.1}$$

For the case of the robot being on the goal side of the ball, Equation 1.1 directs the robot on a curve to push the ball away from the goal. Equation 1.2 prevents this by adding a constant angle to the line connecting the ball and robot until Equation 1.1 takes effect. Equation 1.2 always brings the robot around the right side of the ball before Equation 1.1 directs the robot into a curved path.

$$\phi_3 = \phi_2 + 180^\circ + \alpha \tag{1.2}$$

This scheme worked well in simulation [21], and in real application provided the

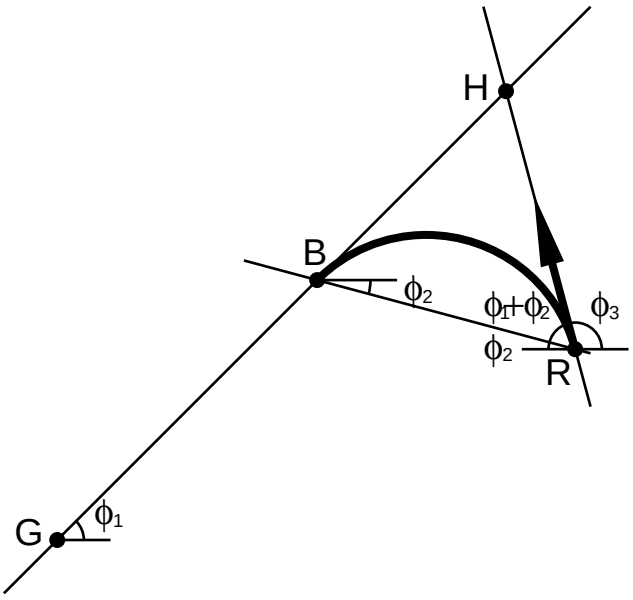

Figure 1.6: Ball Attacking—Case 1

speed of the robot was low. In the simulation small time steps were used to calculate the path of the robots. The robot soccer environment has large delays caused by vision acquisition and processing so when the robot is moving slowly, the step between vision acquisition and processing is small and the robot traverses as predicted. However, high robot speed results in a long travel distances and large positional errors. The robot either hits the ball on the corner of the robot, propelling the ball away, or the robot misses the desired position behind the ball and hunts around trying to get back to the required position. Figure 1.7 shows the effect of the robot travelling large line segments due to time delay caused by vision acquisition and processing.

The robot wheels are placed in a "wheel chair" arrangement. The "wheel chair" configuration gives the robot the ability to scribe any arc. Figure 1.8 shows the robot arcing capability including the capacity to pirouette. This inherent behaviour is best taken advantage of by developing equations that allow the robot to use the arcing ability.

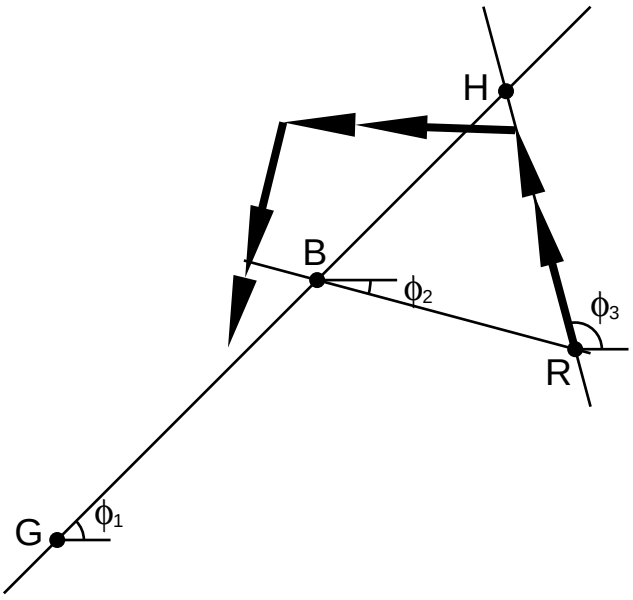

Figure 1.7: Position Error Caused by Vision Acquisition and Processing Delays

Circular paths around the ball are at times very large. Equation 1.1 makes a curved path to the ball depending on the tangent circle. Figure 1.9 shows some of the circular paths produced by Equation 1.1.

The attack the ball strategy was further modified for the Paris competition in 1998. The details of this research are more fully discussed in Chapter 6. A term coined as "lead transfer function" [23] (LTF) was used to reduce the arc that the robot traversed when attacking the ball. This term was used as the function resembled a lead transfer function, but is actuality is not.

Figure 1.9 shows the effect of applying the LTF to the ball attack strategy. The transfer function is used to add a proportion of the angle α to the heading angle to guide the robot to transverse a least distance such that the robot curves around behind the ball. To make the robot have a better chance of getting behind the ball, the point P_1 is placed a constant distance behind the ball. The fixed point P_1 caused a problem with different speeds of the approaching robot. The point

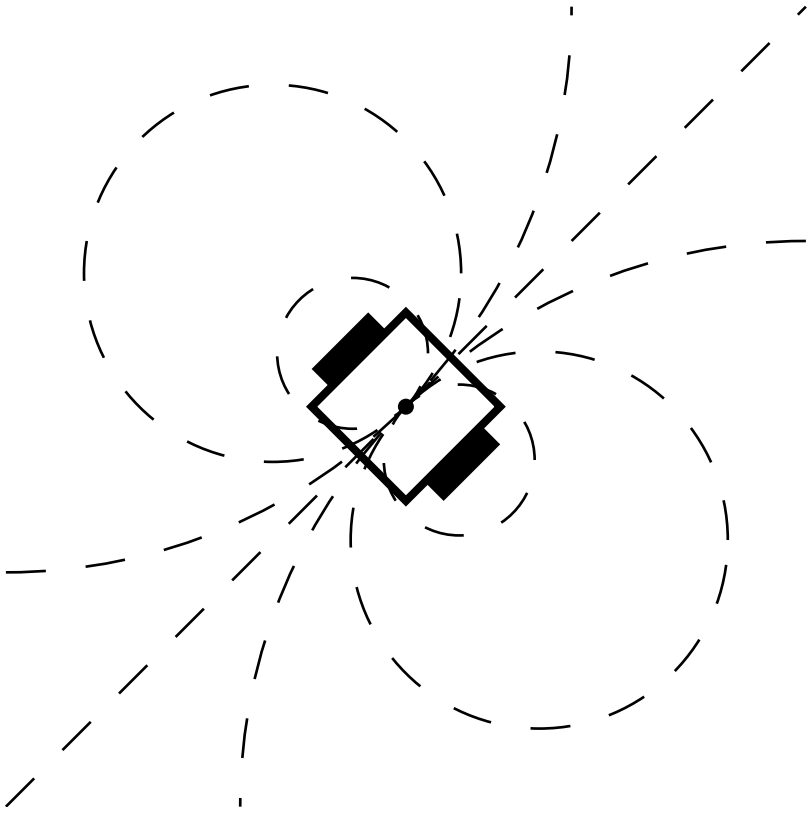

Figure 1.8: Robot Arc Ability

needed to be at the ball for low robot speed and a large distance away for high speed.

The lead transfer function is a high pass filter. A high pass filter in the s-domain is: $sT/(1+sT)$. Figure 1.10 shows the function in the spatial rather than the usual time domain.

One possible solution to large errors caused by long time delays in the image capture and control would be to use prediction to find the position of the robot after a finite time delay. Prediction was used in the CQU team's robot soccer programme intended for competition in Paris. The problem with prediction is that it is highly sensitive to position error. If the object is moving fast, the position error causes a very large prediction error. If the object is moving slow,

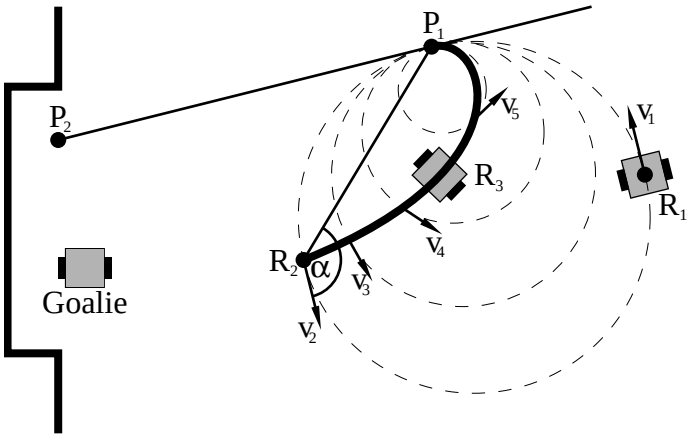

Figure 1.9: Ball Attack Strategy with Lead Transfer Function

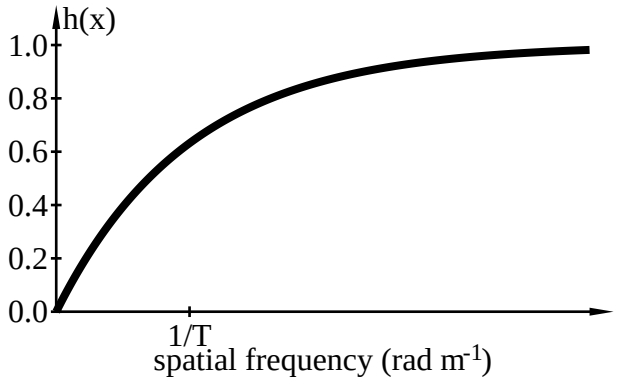

Figure 1.10: Lead Transfer Function in Time Domain

the prediction is more accurate.

However, other problems occur at high speed. Figure 1.11 shows the prediction area and range of prediction angles caused by position error. Prediction is almost impossible without precise position data.

The current status of robot control in multi-agent terms is in its infancy. All teams observed in top level MIROSOT competitions use a centralised agent playing several roles. Observations also indicate that all robot soccer teams in competition level are a reactive agent. A reactive agent produces reflex actions to environmental changes—such as follow the ball. Even the winner of the Paris

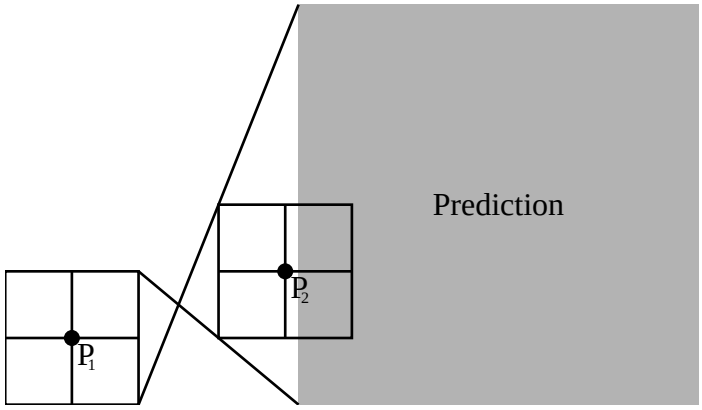

Figure 1.11: Prediction Error caused by Position Errors

competition was observed to be purely reactive. No teams showed deliberative control of their robots at the FIRA Robot World Cup in Paris.

The second strategy implemented for the Paris competition was boundary avoidance strategy. The robots had a habit of colliding with the wall due to the attack the ball strategy. To overcome this problem a potential field was placed near the boundary. The potential field theory was tested and found unsatisfactory due to the robot tending to meander down the wall. The meandering was caused by interactions between the potential field forcing the robot away from the wall and the robot controls directing the robot towards the wall. A piece-wise linear profile was used to limit the robot meandering. Figure 1.12 shows the piece-wise linear profile designed for the Paris competition.

The discussion above showed that there was a need for research into intelligent control algorithms which would deliver more precise control of the robot, especially in regions close to the ball. It has led to the fuzzy control research presented in this thesis. A related area is robot vision. Competition experience has shown this area has been a source of difficulty. Some identification issues are now considered.

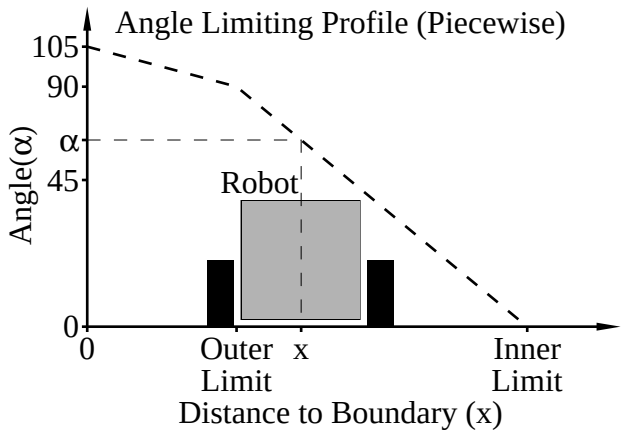

Figure 1.12: Boundary Avoidance by Piece-Wise Linear Profile

1.7 Identification of Team Robot Colours

The main loop for the robot system requires the following actions:

- Image capture,
- Image processing,
- Robot control, and
- RF Transmission.

A commercial video camera is generally used to capture the image. While these cameras are low cost, they are designed for television and lack resolution and image transfer speed. NTSC cameras are generally used as they have a higher frame rate of 30fps compared to the PAL 25fps.

There are 525 lines/frame at a rate of 29.97 frames/s for NTSC. Of the 63.5μs/line, $\approx$ 10.5μs is required for horizontal blanking, leaving 53.33μs/line of displayable

image. With NTSC video composite bandwidth of 6MHz, 53.33μs/line corresponds to 320 black and white cycles, giving $2 \times 320 = 640$ distinguishable horizontal black/white picture elements. Vertical blanking time is around 1.3ms, leaving approximately 480 lines/frame for the image. The DOOIN image capture card used provides YUV 4:1:1 video composite format, giving 160×240 UV chroma elements per field. Projecting the camera resolution onto the playing area equates to each chroma element being 11×5.5mm in size. YUV 4:2:2 can be obtained from a Philips video processing chip [24], giving chroma resolution 320×240 per field.

Using the chroma information at a rate of 60fps gives an effective resolution of 320×240 for odd and even fields. Projecting this resolution on the field requires pixel size to be 5.5mm giving image area of 1760×1320mm, provided that the system is capable of classifying all pixels covering a coloured object. A minimum sized robot team identification patch positioned square to pixels will contain a possible 36 chroma elements. The same colour patch rotated 45° will contain at most a possible $1+3+5+7+7+5+3+1 = 32$ chroma elements. Thirty-seven chroma elements cover the ball. The practical limit to identifying an object is a measure of how well the system can classify pixels correctly.

Image transfer speed of an NTSC commercial video camera is sixty fields per second (16.6ms). Assuming classification of objects, calculation of control and transmission to the robot takes 6.3ms, the ball moving at 5ms^{-1} traverses 115mm in 23ms. In camera terms, a ball moving in the horizontal frame of view, will traverse 15 chroma elements. Some video cameras allow integration times of 1ms. The ball will traverse less than one chroma element for a video camera designed for capturing high speed sports mode. Otherwise, the object in the image is a translucent streak which is very hard to classify when travelling at high speed.

Depending on the design of the video camera, the integration time could be as high as 1/30 seconds. Within a translucent streak of 30 chroma elements is not only hard to classify an object of length 6 chroma elements, but also difficult to determine position.

Delay times of 20ms make it impossible to control a robot travelling at 2.6ms^{-1}. A camera available from Dalsa (*htp://www.dalsa.com*), model DALSTAR 1M75-SA, is suitable for playing robot-soccer. This video camera transfers full frames at 75fps of 1 024 $\times$ 1 024 pixels (transfer data rate 2×40MHz). It also is capable of transferring sub-fields at 100 000fps. The data sheet does not specify the smallest sub-field, but an estimation is $\sqrt{80MHz/100000fps} \approx 28$; rounding to 2^n gives 32×32 pixels. Pixel size projected onto field is 1.5mm giving coverage of 1 536 $\times$ 1 536mm. Assuming integration time of 1ms , image transfer of seven sub-fields taking 1ms, transmission to the robots taking 6.25ms, a 5ms^{-1} object traverses 41.25mm. If the camera has a separate transfer area, the integration time and transfer time is 1ms. Improvement of transmission to the robots will give better system performance with this camera. If the capturing images to transferring to robot time is reduced to 2ms, there will be no need to estimate the position of the robot.

A good reference on image analysis is Gonzalez [25]. An excellent reference on the storage of images in computers is found in Levine [26].

CQU's team first actual competition was in the North American Championship, Anchorage Alaska during the World Automation Congress (WAC) in May 1998. This first competition showed some key problem areas to be solved. The most significant problem related to the data communication link which was prone to failure. This problem was rectified easily and was related to a coding issue.

The second problem was loss of identity of the robots. Each robot is identified by a team colour patch and a personal colour patch. For discussion sake, let the team colour be yellow, the personal colour of robot 1 be red, the personal colour of robot 2 be green and the personal colour of robot 3 be blue. The software constructs regions of colour blobs and calculates the geometric centroid of each colour. Robot centre is calculated by averaging the coordinates of the centres of the two colours. Loss of identification is caused by the software connecting a team colour, say yellow of robot 1, with the personal green colour of robot 2. The heading is calculated by adding 45° to the line joining the centres of the two colours. As a result, the position and heading of the robot is incorrectly identified. Attempts to rectify loss of identification by scanning methods consumed too much computation time and were discarded. The vision issues lead to the development of improved methods discussed in Chapter 11.

1.8 Towards More Intelligent Control Systems

In this thesis we are interested in the application of such modern artificial intelligence techniques to multi-robot systems, path planning and collision-avoidance problems.

Most teams now entering in the FIRA world cup and in the RoboCup competitions are using intelligent systems techniques ranging from neural networks [6], fuzzy logic [8], evolutionary learning [27], Q-learning [30], to multi-agent strategies [31].

The following papers show attempt to design more highly intelligent control systems [32], with: problem solving, planning, adaptation, self-organisation, cog-

nition, and other human operator characteristics. They begin with the more classical approaches of adaptive control and advance to control by modern artificial intelligence techniques which include neural networks, fuzzy logic control (FLC) and genetic or evolutionary algorithms (GA/EA):

- hierarchical coordination of dynamic control has been used for both problems [33, 34]
- artificial potential fields for avoidance in [35]
- potential functions for attraction to targets and avoidance of obstacles [36]
- growing polyhedral obstacles for planning collision-free paths [37]
- differential game theory (precursor to multi-agent strategy) applications in [38]
- Liapunov techniques for point mass and articulated robot arms in [39, 40]
- neural network optimisation [41]
- genetic algorithm application [42]
- evolutionary algorithms [43, 44]
- integration of fuzzy logic and genetic algorithm for adaptive learning in [45, 46, 47, 48, 49, 50, 51]
- and fuzzy logic controllers for motion planning of mobile robots [53].

The usual definition that an adaptive system is one that is capable of accommodating unpredictable environmental changes or disturbances (whether these arise within the system or external to it). The concept of adaptation is a fundamental

characteristic of all living organisms as they attempt to maintain physiological equilibrium in the midst of changing environmental conditions. Indeed, numerous definitions of adaptive control systems exist, (see Ogata [54] for a formal definition in control systems). Vagueness surrounds most definitions and classifications of adaptive systems due to the large variety of mechanisms by which adaptation may be achieved. Definitions also fail to differentiate between external manifestations of adaptive behaviour and the internal mechanisms used to achieve it. However, inclusion of adaptability into systems has great appeal to designers for versatility by accommodating environmental changes, to compensate for minor engineering errors and show robustness against minor system failures. The ultimate goal of high system reliability with minimum design appears achievable with the new control techniques.

Intelligent control systems self-organise, learn and adapt control laws or control rules to environmental changes. As such, these systems are said to be less deterministic than their classical control counterparts which are modelled by exact algebraic and differential equations. Complex systems do not have an exact modelling process, input and output data have a degree of uncertainty. This thesis will focus on fuzzy logic systems to the soccer robot control problem.

1.9 Fuzzy Logic

The fuzzy logic based approach to solving problems in control has been found to excel in those systems which are very complex, highly non-linear and with parameter uncertainty. This is certainly the case for the game of robot soccer. We will now present a very compact overview of fuzzy logic systems and the evolutionary methods that may be used to drive their adaptation to the robot

problem. This very compact overview will be significantly expanded in Chapters 2 and 3. A reader who finds this section rather too compact may refer to the following chapters.

We may view a fuzzy logic controller as a real-time expert system that employs fuzzy logic to analyse input to output performance. Fuzzy logic provides a method of conversion from/to crisp values to fuzzy values by fuzzification/defuzzification. This allows the main reasoning unit to use linguistic control strategy derived directly from expert knowledge. Inherent in the fuzzy system is a simple method to interrogate the system to answer the "why?" question in linguistic quantities. This is the main advantage over potential theory approaches and the application of neural networks. It is impossible to obtain a knowledge of what is happening in the system of the alternate methods without making inferences between all possible input and output combinations.

A Fuzzy logic controller (FLC), see Figure 1.13, usually consists of:

(i) A fuzzification unit which maps measured inputs of crisp value into fuzzy linguistic values to be used by a fuzzy reasoning mechanism.

(ii) A knowledge base (KB) which is the collection of expert control knowledge required to achieve the control objective.

(iii) A fuzzy reasoning mechanism that performs various fuzzy logic operations to infer the control action for the given fuzzy inputs.

(iv) A defuzzification unit which converts the inferred fuzzy control action into the required crisp control values to be entered into the system process.

Knowledge in fuzzy reasoning (the decision making unit), is usually expressed as

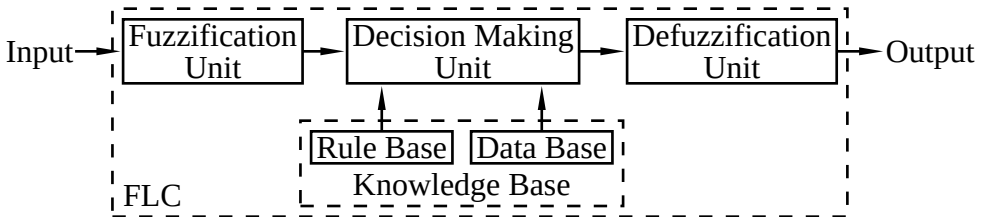

Figure 1.13: Basic Structure of a FLC

rules with sentence conjunctives AND, OR and ALSO, for example:

> **If** Distance x to second player is **SMALL** OR Distance y to obstacle is **CLOSE** AND speed v is **HIGH Then** Perform **LARGE** Correction to steering angle θ ALSO make **MEDIUM** reduction in speed v.

Three antecedents in this example give rise to two outputs. For Multi-Input Single-Output (MISO) system, letting the *linguistic* variable **MEDIUM** be represented by B_k for example, in general can be written:

> **If** x_{k1} is A_{k1} AND x_{k2} is A_{k2} AND $\cdots$ AND x_{kn} is A_{kn} **Then** y is B_k.

In control, language statements such as this can be expressed easily as fuzzy control rules stored in set called the fuzzy KB. Reasoning is usually based on the general modus ponens or direct reasoning law. Each rule in the KB corresponds to a fuzzy relation. For the k^{th} rule defined above, the fuzzy relation can be expressed as $R_k = A_k \rightarrow B_k$. For N rules in the KB, the overall fuzzy relation $R = \cup_{k=1}^{N} R_k$, where R may take either point valued fuzzy implication form or inter-valued fuzzy implication form. Point Value Composition Rule of Inference (PVCRI) involves the following operators:

- Identify 'AND' and 'OR' in the antecedent by intersection and union operators respectively (Intersection is always performed first)
- Find the firing strength α_k of the k^{th} rule,
- Apply the chosen compositional operator to infer control actions in the consequent part of the rule.

A number of compositional operators are in the literature. We illustrate PVCRI; using the sup-min operator. Assume the KB has two rules.

If x is A_1 AND y is B_1 **Then** z is C_1 **If** x is A_2 AND y is B_2 **Then** z is C_2

The membership of the inferred consequence C is point-wise given by $\mu_C(z) = (\alpha_1 \wedge \mu_{C1}(z)) \vee (\alpha_2 \wedge \mu_{C2}(z))$.

A defuzzification strategy produces a non fuzzy control action that best represents the possibility distribution of the inferred control action. The *mean of maximum defuzzification* yields the mean value of all local control actions whose membership functions reach the maximum. The crisp value of z is given in Equation 1.3.

$$z = \sum_{k=1}^{2} \alpha_k H_k W_k / \sum_{k=1}^{2} \alpha_k H_k, \tag{1.3}$$

with W_k being the crisp support value at which the membership function reaches maximum H_k (most usually 1 for normalised membership functions). A symmetrical function is required for each rules consequent fuzzy set.

In general a 'rule of inference' in fuzzy logic reasoning is a mathematical statement describing how linguistic variables are to be manipulated and employed to control

the problem environment. The first step in any application is to determine which variables will be important in choosing an effective control action. Any number of decision variables may appear, but the more that are used, clearly the larger the rule set that must be written. Several approaches to FLC development are possible.

1.9.1 Self-Organising Fuzzy Logic Controls

Typically the KB is generated by an expert but a fundamental weakness with the static fuzzy logic is the acquisition of the rule base which is frequently incomplete and its control strategies are conservative. One approach to overcome this is the construction of *self-organising fuzzy logic controllers* [55] (SOFLC) which have been found useful in developing controllers for systems subject to time varying changes and unknown environmental disturbances. Figure 1.14 outlines the typical structure of a SOFLC. These controllers consist of a two-level hierarchical rule based controller in which the fuzzy control rule base is created and monitored by a self-organising learning algorithm in the upper level and used by a fundamental controller in the lower level. The learning module may contain a performance index table and a rule generation and modification algorithm for creating new rules or modifying existing ones. This module can be developed indirectly through on-line identification of a control system model which can then be used to formulate the control strategies or directly from actual knowledge of the control system itself.

Such self-organising fuzzy logic controllers are used mainly for the creation and modification of the rule base. An adaptive FLC is characterised by a self-tuning ability to generate new rules and modify existing rules. Tuning is also done by

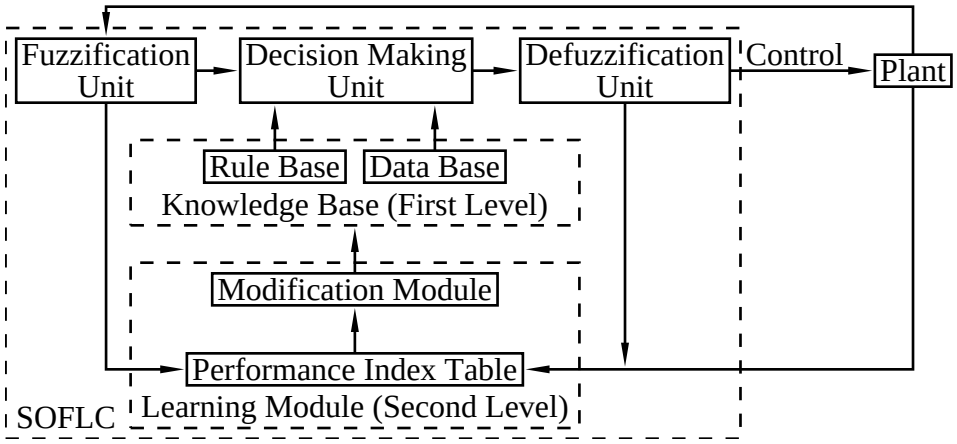

Figure 1.14: Basic Structure of a SOFLC

adjusting their membership functions, the universes of discourse, scaling factors or control resolutions. Of interest is the question of how this self-organisation and adaptation can be carried out in an automated fashion.

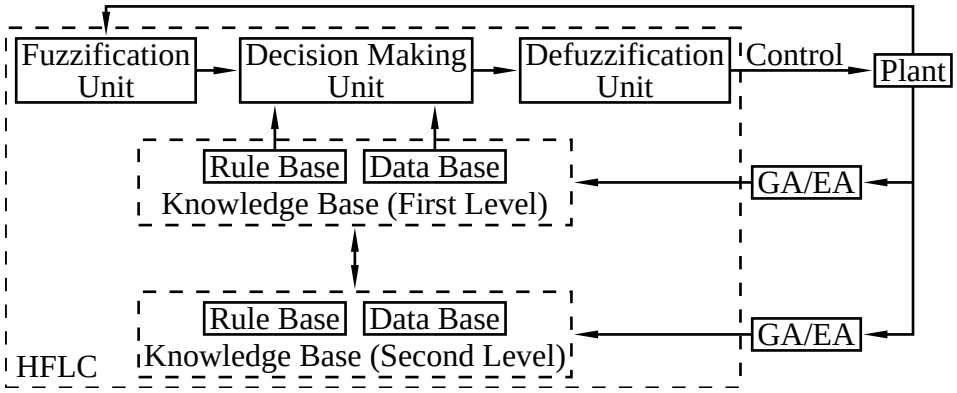

Figure 1.15: GA Fuzzy Rule Generator Architecture

1.9.2 Hierarchical Fuzzy Logic Controllers

As mentioned previously the first step is to determine which variables will be important in choosing an effective control action. Any number of these decision variables may appear, but the more that are used, the larger the rule set that

must be written. It is known [56], that the total number of rules in a system is an exponential function of the number of system variables. This is referred to as the "curse of dimensionality". For a multi-dimensional system, it may become unproductive to realise the fuzzy rule-based controller. To overcome this problem, a hierarchical fuzzy control structure (HFLC) can be used where the most influential parameters are chosen as the system variables in the first level, the next most important parameters are chosen as the system variables in the second level, and so on, [57]. In this hierarchy, the first level gives an approximate output which is then modified by the second level rule set. This procedure can be repeated in succeeding levels of hierarchy. The number of rules in a complete rule set is reduced towards a linear function of the number of variables by the hierarchy. Methods for transforming human knowledge or experience into the rule base and database of a fuzzy inference system seek to generate the rule-base of a system by learning all parameters of the system at once. Their success is limited and if there is a need to add or remove one of the parameters of the rule-base the whole rule-base must be rebuilt.

Problems encountered in this application area of fuzzy logic particularly in multi-robot systems such as the robot-soccer test bed are:

- How to develop reactive and deliberative rules in the knowledge, this involves an overlay of multi-agent strategy theory.
- How to reduce the number of rules for effective fuzzy control—the curse of dimensionality. This involves understanding of how to design fuzzy knowledge based systems using hierarchical and multi-layered fuzzy systems.
- How to build the fuzzy knowledge base for each robot so that it is adaptive to a dynamic changing environment. This involves incorporating of a self-

organising structure that changes/modifies, adds to, or deletes rules from the knowledge base.

- How to evolve the knowledge base directly without expert knowledge, directly from input-output data via simulation or from real-time operation within a robot-soccer game.

1.10 Evolutionary Learning of Fuzzy Control

1.10.1 Key Objectives

Our key objective is to examine techniques for evolving fuzzy logic controllers for each robot in a team, that incorporates multi-agent strategies between team robots. We believe this can be achieved in two ways:

1. Evolve fuzzy controllers via simulation for the robots-soccer scenario similar to that done by Mohammadian and Stonier, see [45, 46, 52, 67, 68, 69, 70, 71]. This will require the use of a simulation package with sufficient kinematic/dynamic structure of each robot's behaviour such that the end result will have acceptable performance in real hardware. Learning of the KB could be done using input output data from actual real-time robot soccer games. FIRA is now to implement codes within international competition such all teams are required to keep logged data from competition matches to be used by all world teams as simulation data input-output.

2. Evolve/modify/adapt a fuzzy control knowledge base for each robot during a real-time robot soccer game between two teams.

To undertake these objectives, research is required in:

- The design of fuzzy logic control systems and their application to control systems.
- The application of evolutionary algorithms to evolve fuzzy logic systems and evolve fuzzy logic controllers in mobile robot systems.

1.10.2 Evolutionary Algorithms

Since their appearance genetic algorithms and variants thereof now commonly lumped under the heading of evolutionary algorithms, have been used in a variety of applications to solve many real world problems in such areas as: ecology, machine learning, engineering and management. As one of a number of evolutionary computation techniques, including evolution strategies, genetic programming and evolutionary programming, it is based on principles of evolution associated with inheritance and the fight for survival of the fittest. It is a heuristic search technique that transforms a population of individuals $P(t) = \{x_1^t, \ldots x_n^t\}$ at iteration t to the next $t + 1$. Each individual can be considered to represent a potential solution to a given problem. Although in their original formulation the structure of each individual was binary encoded [4, 59] modern formulation of structure includes for example, integer and real encoding [27]. To each individual is an associated measure of its fitness for survival. The new population $P(t+1)$ is obtained from the old by the use of genetic operators such as crossover—creating children by recombining parts of usually more than one parent through a process of selection for mating of the parents, and mutation—creating new individuals by perturbing the individual's structure. The form of the algorithm may be written as:

```
begin
   t = 0
   Create random P(0)
   Evaluate Fitness of P(0)
   while (not Terminated) do
   begin
      Create P(t+1) from P(t)
      Evaluate Fitness of P(t+1)
      t = t+1
   end
end
```

In creating the new population, there are many different variants of operators that may be found in the literature. These include proportional selection where the probability of selection for mating is proportional the individual's fitness, ranking methods and tournament selection, one point and many point crossover as well as arithmetic crossover. A full coverage of evolutionary computational algorithms and the applications can be found in the following two surveys [60, 61] and papers [62, 63].

1.10.3 Learning the Knowledge Base

An important aspect of fuzzy logic application is the determination of a fuzzy logic knowledge base to satisfactorily control the specified system. This may be derivable from an appropriate mathematical model or just from system input-output data. Inherent in this are two main problems. The first is to obtain an adequate knowledge base structure for the controller, usually obtained from

expert knowledge, and second is that of selection of key parameters defined in the method.

The KB is typically generated by an expert but a fundamental weakness is that it is frequently incomplete, and its control strategies are conservative. To overcome this one approach is to construct *self-organising fuzzy logic controllers* [55]. These self-organising fuzzy logic controllers are used mainly for the creation and modification of the rule base.

Of interest is the question of how this self-organisation and adaptation can be carried out in an automated fashion. One way is to incorporate genetic/evolutionary algorithms and a substantial amount of literature is now available on these so called *genetic fuzzy systems*. Initial research appears to have commenced with Karr [64] who applied genetic algorithms for learning efficient membership functions and also Thrift [65] who used genetic algorithms for learning the control rules for the cart pole balancing problem. Details of the many interesting approaches can be found in the survey by Cordón and Herrera [66].

As an illustration, consider the following collision-avoidance problem in a two robot system. A three-level hierarchical, fuzzy logic system is proposed to solve the problem, full details can be found in [67]. In the first layer, two knowledge bases, one for each robot, are developed to find the steering angle to control each robot to its target. In the second layer two new knowledge bases are developed using the knowledge in the first layer to control the speed of each robot so that each robot approaches its target with near zero speed. Finally in the third layer, a single knowledge base is developed to modify the controls of each robot to avoid collision in a restricted common workspace.

The important issue is that of learning knowledge in a given layer sufficient for use in higher layers. Consider the knowledge base of a single robot in layer one. It is not sufficient to learn a fuzzy knowledge base from an initial configuration and use this knowledge base for information on the steering angle of the robot to learn fuzzy controllers in the second layer. Quite clearly this knowledge base is only guaranteed to be effective from this initial configuration as not all the fuzzy rules will have fired in taking the robot to its target. We have to find a KB that is effective to some acceptable measure, in controlling the robot to its target from 'any' initial configuration. One way is to first learn a set of local fuzzy controllers, each KB learnt by an evolutionary algorithm from a given initial configuration within a set of initial configurations spread uniformly over the configuration space. These KBs can then be *fused* through a *fuzzy amalgamation* process [49] into the global (final), fuzzy control knowledge base. An alternative approach [68, 69], is to develop an evolutionary algorithm to learn directly the 'final' KB by itself over the region of initial configurations. This can be achieved by appropriately evaluating the fitness of a string at different configurations within the chosen set of initial configurations in each generation and the use of appropriate operators which transfers this local knowledge from generation to generation.

This work differs from other learning methods of evolving rules in fuzzy systems in that it examines how evolutionary learning of the fuzzy knowledge base for a given system can be achieved directly over a broad region of initial configurations for single layered structures with many rules, to multi-layered and hierarchical structures. It is particularly important for the multi-layered and hierarchical structures as pointed out in the previous discussion. This learning evolves an initial *arbitrary* knowledge base using evolutionary algorithms, tuning of membership functions can be easily incorporated but has not been presented

in this report. This work has been reported and fully analysed in the paper by Mohammadian and Stonier [70] in 1998.

Fuzzy logic controllers have been successfully used to replace conventional controllers. However, a subjective choice of the FLC's parameters has a great influence on the overall performance of the system. One of the main issues in recent research has been the improvement of the system performance through automatic design of the FLC without human intervention. Evolution is an effective method for forming FLC's since it does not suffer from the local optimum problem with the use of arbitrary cost function that can not be expressed in exact mathematical formulas, such as the convergence time, the number of rules used, etc. This is why the evolutionary fuzzy approach is recently drawing significant attention as an alternative to the neuro-fuzzy approach.

Evolutionary fuzzy systems, EA's and especially GA's in the majority of cases are used to tune the membership functions and/or select the optimal rule set. Two main approaches are used to develop these systems:

1. FLC-level evolution (Pittsburgh approach)

   Each individual encodes the whole rule set of an FLC. Thus an individual by itself represents a possible solution.

2. Rule-level evolution (Michigan approach)

   The individual is a single rule and an FLC is constructed by collecting some individuals. Thus an individual is only a partial solution and a special credit assignment scheme must be introduced for the appropriate evaluation of each rule.

While in FLC-level evolution each individual evolves independently of the other ones, in Rule-level evolution, the individuals must coevolve by evaluating how well they perform when combined with other individuals. This is essentially the essence to strategy tasking amongst agents (mobile robots) in a multi-agent system. In general, this coevolutionary approach decomposes a high-level composite into sub-components, which get specialised in interaction with the other ones. As a result of collaboration of these sub-components, the high-level goal is effectively achieved. According to a study on the high-dimensional fuzzy classification application, the Michigan approach is reported to be superior to the Pittsburgh approach.

Mohammadian and Stonier have used the Pittsburgh approach in evolving a full knowledge base for each robot in their robot simulation system [71].

1.11 Scope

Present robot soccer competitions appear to ignore the basics of control and concentrate of the more elaborate higher level controls. However, observations at competitions have highlighted a lack or control over the robots. The higher level AI techniques, such as multi-agent, cannot compensate for poor robot control in the lower levels.

Most teams seem to be having difficulty with misalignment errors and do not account for robot dynamics. The “Stop-Turn-Run” algorithms introduce large momentum forces as average robot speed increases. Robots have been increasing in speed every competition and are continuing to show loss of control caused by limitations on: image capture rates, image resolution, soft real time operating

systems (processing) and smoothness of robot control. As robot speed increases, the need for smooth control becomes dominant. A good foundation of mobile robot control is needed for higher level AI techniques to be effective.

Our early work in mobile robot control highlighted the difficulty of using linear techniques in a highly non-linear, time varying, multi-dimensional, system—robot soccer. A decision to use fuzzy systems seemed appropriate to control this system, for they deal with problems that are non-linear and consume little execution time for small systems. The high dimensionality of the system leads to development of the fuzzy knowledge base by using evolutionary algorithms.

The next chapter introduces fuzzy logic concepts used in this thesis.