A Novel Dual Surface Type-2 Fuzzy Logic Controller for a Micro Robot

By
Philip Birkin, BA, BSc

Thesis submitted to The University of Nottingham
for the Degree of Doctor of Philosophy

Intelligent Modelling & Analysis Research Group
School of Computer Science
The University of Nottingham
Nottingham, United Kingdom

July 2010

A Novel Dual Surface Type-2 Fuzzy Logic Controller for a Micro Robot

Philip Birkin

Submitted for the Degree of Doctor of Philosophy
July 2010

Abstract

Over the last few years there has been an increasing interest in the area of type-2 fuzzy logic sets and systems in academic and industrial circles. Within robotic research the majority of type-2 fuzzy logic investigations has been centred on large autonomous mobile robots, where resource availability (memory and computing power) is not an issue. These large robots usually have a variation of a Unix operating system on board. This allows the implementation of complex fuzzy logic systems to control the motors. Specifically the implementation of interval and geometric type-2 fuzzy logic controllers is of interest as they are shown to outperform type-1 fuzzy logic controllers in uncertain environments. However when it comes to using micro robots it is not practical to use type-1 and type-2 fuzzy logic controllers, due to the lack of memory and the processor time needed to calculate a control output value. The choice of motor controller is usually either fixed pre-set values, a variable scaled value or a PID controller to generate wheel velocities.

In this research novel ways of implementing type-1 and interval type-2 fuzzy logic controllers on micro robots with limited resources are investigated. The solution that we are proposing is the use of pre-calculated 3D surfaces generated by an off-line Fuzzy Logic System covering the expected ranges of the input and output variables. The surfaces are then loaded into the memory of the micro robots and can be accessed by the motor controller. The aim of the research is to test if there is an advantage of using type-2 fuzzy logic controllers implemented as surfaces over type-1 and PID controllers on a micro robot with limited resources.

Control surfaces were generated for both type-1 and average interval type-2 fuzzy

logic controllers. Each control surface was then accessed using bilinear interpolation to provide the crisp output value that was used to control the motor. Previously when this method has been used a single surface was employed to hold the information. This thesis presents the novel approach of the dual surface type-2 fuzzy logic controller on micro robots. The lower and upper values that are averaged for the classic interval type-2 controller are generated as surfaces and installed on the micro robots. The advantage is that nuances and features of both the lower and upper surfaces are available to be exploited, rather than being lost due to the averaging process.

Having conducted the experiments it is concluded that the best approach to controlling micro robots is to use fuzzy logic controllers over the classical PID controllers where ever possible. When fuzzy controllers are used then type-2 fuzzy controllers (dual or single surface) should be used over type-1 fuzzy controllers when applied as surfaces on micro robots. When a type-2 fuzzy controller is used then the novel dual surface type-2 fuzzy logic controller should be used over the classic average surface. The novel dual surface controller offers a dynamic, weighted, adaptive and superior response over all the other fuzzy controllers examined.

Declaration

The work in this thesis is based on research carried out at the *Intelligent Modelling and Analysis* (and, originally, the *Automated Scheduling, Optimisation and Planning*) *Research Group*, The School of Computer Science, The University of Nottingham, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Dr. Jon Garibaldi, for providing support, guidance and opportunities throughout my PhD and who has made it possible to complete this work. I am extremely grateful for his valuable comments and the help he has afforded throughout my research.

In particular, I would like to thank Dr. Peer-Olaf Siebers for the help, encouragement and advice that he provided during my research. I would also especially like to thank Dr. Amanda Whitbrook for her help and advice, together with Dr. Joe Baxter, Dr. William O. Wilson, Dr. Robert Foster Oates (boB) and Dr. Julie Greensmith.

I would also like to thank the staff and all students of the IMA/ASAP Groups at the University of Nottingham. Thank you for creating a warm and friendly environment for academic research.

Finally, I would like to express my sincerest thanks and deepest gratitude to my wife, Margaret Mary Kenney, and to my parents Stanley and Teresa Birkin and families, who have provided their deep love and unfailing encouragement throughout my studies. They had a lot to put up with.

Contents

Abstract ii

Declaration iv

Acknowledgements v

1 Introduction 1

1.1 Fuzzy Logic Systems 1

1.1.1 Applications of Fuzzy Logic 5

1.1.2 Fuzzy Logic Controller Development 5

1.1.3 Other Robot Controllers 5

1.1.4 Robot Football 6

1.2 Motivation 6

1.3 Research Aims and Objectives 7

1.4 Organisation of the Thesis 9

2 Literature Review 11

2.1 Theory of Fuzzy Sets and Systems 11

2.1.1 Classical Set Theory 12

2.1.2 Boolean Logic 15

2.1.3 Boolean Tautologies 16

2.1.4 Fuzzy Sets 18

2.1.5 Fuzzy Operators 20

2.1.6 Fuzzy Logic 24

2.1.7 Defuzzification 27

2.1.8 Fuzzy Inference and Reasoning 28
2.1.9 Fuzzy Inferencing Systems (FIS) 29
2.2 Type-2 Fuzzy Logic Systems 30
2.2.1 General Type-2 Fuzzy Sets 33
2.2.2 Operations on General Type-2 Fuzzy Sets 36
2.2.3 General Type-2 Fuzzy Logic Systems 37
2.2.4 Consequent Sets Combination 40
2.2.5 The Generalised Centroid Calculation 44
2.2.6 General Type-2 Centroid Type-Reduction 45
2.3 Interval Type-2 Fuzzy Logic 46
2.3.1 The Footprint of Uncertainty 47
2.3.2 Interval Type-2 Lower and Upper Membership Functions 48
2.3.3 Meet and Join of Interval Type-2 Fuzzy Sets 50
2.3.4 The Interval Type-2 Inference Process 51
2.3.5 Interval Type-2 Type Reduction and Defuzzification 52
2.4 Architectures of Robotic Systems 53
2.4.1 Subsumption Architecture 53
2.4.2 Motor Schema Based Systems 57
2.4.3 Other Architectures 61
2.5 Robotic Strategy Methods 62
2.5.1 Potential Fields (PF) 63
2.5.2 Neural Networks (NN) 63
2.5.3 Fuzzy Systems (FS) 64
2.5.4 Rule-Based Systems (RB) 64
2.5.5 Reactive Systems (RS) 64
2.5.6 Other Methods 65
2.6 PID Control 68
2.7 Robot Football 69
2.8 Potential Areas of Research Interest 70
2.9 Recent Work on Type-2 Fuzzy Sets and Systems 73
2.9.1 Mobile Robot and Control Applications 74

2.10 Summary 78

3 Experimental Setup 79

3.1 Laboratory Environment 79

3.2 Miabot Robots 80

3.3 Robot Soccer Engine 81

3.4 Fuzzy Inference System 86

3.4.1 DC Motor Model Setup 88

3.5 Independent PID Controller 89

3.6 The Strategies 89

3.6.1 Case Based Strategy 90

3.7 Problems Discovered During Experimentation 90

3.7.1 Lighting 90

3.7.2 Camera Frame Speeds 90

3.7.3 Physical Damage 91

3.7.4 Robot Control Parameters 91

3.7.5 Impact 91

3.8 Robot Framework and PID Controller 92

3.8.1 Initialization 92

3.8.2 Command Support 93

3.8.3 Communication 93

3.8.4 Motor Control 95

3.8.5 Control Algorithm 95

3.9 Summary 97

4 Development of Baseline PID Controller 98

4.1 Introduction 98

4.2 No Load Encoder Clix to Power Curve Experiment 98

4.2.1 Setup 99

4.2.2 Discussion 100

4.2.3 Conclusion 102

4.3 With Load Encoder Clix to Power Curve Experiment 102

4.3.1 Setup . . . 103
4.3.2 Discussion . . . 103
4.3.3 Conclusion . . . 106
4.4 Robot PID Tuning Experiment . . . 107
4.4.1 Setup . . . 108
4.4.2 Discussion . . . 108
4.4.3 Conclusion . . . 111
4.5 Summary . . . 111

5 Evaluation of Membership Functions for Type-1 and Type-2 Fuzzy Logic Controllers 113

5.1 Introduction . . . 113
5.2 Evaluation of Alternatives for Fuzzy Logic Controllers . . . 114
5.2.1 Running the Simulator for Seven Term Controllers . . . 117
5.3 Selection of Type-1 and Type-2 controllers . . . 119
5.3.1 Five and Three Term Type-2 Membership Functions . . . 121
5.4 Running the Simulator for Seven, Five and Three Term Controllers . . . 122
5.4.1 FLC Step Change RMSE Results . . . 122
5.4.2 FLC Inertia Change RMSE Results . . . 124
5.4.3 FLC Statistical Analysis . . . 124
5.4.4 PID Controller Step and Inertia Results . . . 125
5.4.5 Comparison of Type-2 Controllers with Type-1 and PID . . . 126
5.5 Discussion . . . 128
5.6 Summary . . . 129

6 The Investigation of Single/Dual Surfaces for Type-2 Fuzzy Logic Controllers 131

6.1 Setup . . . 132
6.1.1 Seven Term and Three Term Fuzzy Logic Rules . . . 132
6.1.2 Seven Term Membership Function Parameters and Shapes . . . 133
6.1.3 Three Term Membership Function Parameters and Shapes . . . 133
6.2 Controller Surfaces . . . 134
6.2.1 Seven Term Membership Function Surfaces . . . 134

6.2.2 Three Term Membership Function Surfaces 135
6.3 Simulator Experiments 135
6.3.1 Simulation Inputs 135
6.4 Type-1 and Type-2 Seven and Three Term FLC Simulation Experiments . 136
6.4.1 Experiment Comparing Type-2 and Type-1 FLCs without Noise . 136
6.4.2 Simulation Results 136
6.4.3 Experiment Comparing Type-2 and Type-1 FLCs with Noise . . . 138
6.4.4 Simulation Results 139
6.4.5 Experiment Comparing Type-2 FLCs with Varying Membership Thresholds and Noise Levels 142
6.4.6 Simulation Results 143
6.5 Discussion 145
6.6 Dual Surface Seven Term and Three Term FLC Simulation Experiments . 145
6.6.1 Introducing the Dual Surface Type-2 Fuzzy Logic Controller . . . 145
6.6.2 The Dual Surface Average Threshold Mechanism 146
6.6.3 The Dual Surface Weighted Average Mechanism 146
6.6.4 Experimental Setup 147
6.6.5 Seven Term Dual Surface Fuzzy Logic Controller 147
6.6.6 Simulation Results for Seven Term Dual Surface FLC 148
6.6.7 Three Term Trapezoidal Dual Surface Fuzzy Logic Controller . . 150
6.6.8 Three Term Trapezoidal Triangular Dual Surface FLC with Minimum Implication 151
6.6.9 Three Term Trapezoidal Triangular Dual Surface FLC with Product Implication 152
6.7 Discussion 155
6.8 Summary 157

7 Evaluation of Single and Dual Surfaces in the Real World 159
7.1 Introduction 159
7.2 How surfaces are generated and how they hold information 161
7.3 Bilinear Access of Surface Arrays in the Micro Robot 164
7.4 Robot Response Time to a Command 167

7.4.1 Background . . . 167
7.4.2 Method . . . 167
7.4.3 Results of the LED Detection Test . . . 168
7.4.4 Conclusion . . . 168
7.5 Method for Comparing the Controllers . . . 168
7.5.1 Results for Comparing the Controllers . . . 171
7.6 Introduction to Dual Surface Type-2 Controller . . . 173
7.6.1 Method for The Threshold Dual Surface Controller . . . 173
7.6.2 Results for the Threshold Dual Surface Controller . . . 174
7.6.3 Method for the Minimum and Maximum Dual Surface Controllers 175
7.6.4 Results for the Minimum and Maximum Dual Surface Controllers 176
7.7 Comparison of Three Controllers . . . 176
7.8 Method . . . 176
7.8.1 Observations . . . 177
7.9 Results . . . 177
7.10 Statistical Analysis of the Real World Robot Controllers . . . 178
7.10.1 First Group Analysis - PID Controllers . . . 178
7.10.2 Second Group Analysis - All Controllers . . . 181
7.10.3 Second Group Analysis - All robots and controllers . . . 184
7.10.4 Third Group Analysis - Type-2 Fuzzy Logic Controllers . . . 185
7.11 Discussion . . . 190
7.12 Summary . . . 190

8 Conclusions 192

8.1 Contributions . . . 192
8.2 Limitations . . . 195
8.3 Future Work . . . 196
8.3.1 Expectation . . . 197
8.4 Dissemination . . . 197
8.4.1 Refereed Conference Papers . . . 197
8.4.2 Presentations . . . 197

Appendix **213**

A Chapter 3 Tables **213**

A.1 Maibot Hardware Specification 213

B Chapter 4 Tables **215**

C Chapter 5 Membership Function Shapes, Surfaces and Tables **226**

D Chapter 6 Membership Functions, Surfaces and Tables **257**

E Chapter 7 Tables and Results **307**

List of Figures

1.1 a) the Crisp Set Tall. b)The Type-1 Fuzzy Set Tall. 3

2.1 a) The Type-2 Interval Fuzzy Set $\tilde{A}$. b) The Membership Grade of a Point x in $\tilde{A}$. 47

2.2 Footprint of Uncertainty for Gaussian primary membership function with uncertain standard deviation. 48

2.3 a) The FOU for a Type-2 Fuzzy Set $\tilde{A}$. b) Secondary Membership Functions for the points x_1 and x_2 in $\tilde{A}$. 50

3.1 Maibot Pro BT' Robot 80

3.2 Robot Soccer Engine Overview 82

3.3 Robot Player Caps 84

3.4 Type-2 FLC Schema 87

3.5 DC-Motor Model 89

4.1 Histograms of No Load PWM Motor Average Clixs 101

4.2 Comparisons of Load and No Load Motor Responses 104

4.3 Histograms of Loaded PWM Motor Average Clixs 105

4.4 Pwm Overview for PID Parameters 110

7.1 Lower and Upper Dual Surfaces 162

7.2 Average Type-2 and Type-1 Surfaces 163

7.3 Difference Between Upper Surface and Lower Surface 163

7.4 Difference between Average Type-2 Surface and Type-1 Surface 164

7.5 Bilinear Interpolation Sample Space 165

7.6 Box Plots of PID Controller Response without Ramp 179

7.7 Box Plots of PID Controller Response with Ramp 180
7.8 Robot 1 Controller Response without Ramp 182
7.9 Robot 1 Controller Response with Ramp 183
7.10 All Controllers Response at Speeds 10,20,40 and 80 without Ramp 186
7.11 Dual Surface and Average Response for Robot 1 only with Ramp 187
7.12 Dual Surface and Average Response for 3 Robots with Ramp 188

C.1 Type-1 Seven Term Gaussian Membership Shapes and Surface 227
C.2 Type-1 Seven Term Trapezoidal Membership Shapes and Surface 228
C.3 Type-1 Seven Term Triangular Membership Shapes and Surface 229
C.4 Type-1 Seven Term Trapezoidal Triangular Membership Shapes and Surface 230
C.5 Type-2 Seven Term Gaussian Membership Shapes and Surface 233
C.6 Type-2 Seven Term Trapezoidal Membership Shapes and Surface 234
C.7 Type-2 Seven Term Triangular Membership Shapes and Surface 235
C.8 Type-2 Seven Term Trapezoidal Triangular Membership Shapes and Surface 236
C.9 Type-1 Five Term Trapezoidal Triangular Membership Shapes and Surface 238
C.10 Type-1 Three Term Trapezoidal Triangular Membership Shapes and Surface 242
C.11 Type-1 Three Term Trapezoidal Membership Shapes and Surface 243
C.12 Type-2 Five Term Trapezoidal Triangular Membership Shapes and Surface 244
C.13 Type-2 Three Term Trapezoidal Triangular Membership Shapes and Surface 246
C.14 Type-2 Three Term Trapezoidal Membership Shapes and Surface 248
C.15 Type-1 and Type-2 Overview Responses with PID 249
C.16 Type-1 Response Overview 250
C.17 Type-2 Response Overview 250
C.18 Fuzzy Logic Controllers Step Response Overview 251
C.19 Fuzzy Logic Controllers Inertia Response Overview 251
C.20 Type-1 Step Response Overview 252
C.21 Type-1 Inertia Response Overview 252
C.22 Type-2 Step Response Overview 253
C.23 Type-2 Inertia Response Overview 254
C.24 PID and Type-2 with PID Overview Responses with Noise 254
C.25 Type-1 and Type-2 Overview Step Responses with PID and Noise 254

C.26 Type-1 and Type-2 Overview Inertia Responses with PID and Noise . . . 255

D.1 Type-2 Seven Term Trapezoidal with Threshold - 1 259
D.2 Type-2 Seven Term Trapezoidal with Threshold - 0.9 260
D.3 Type-2 Seven Term Trapezoidal with Threshold - 0.8 261
D.4 Type-2 Three Term Trapezoidal with Threshold - 1 263
D.5 Type-2 Three Term Trapezoidal with Threshold - 0.9 264
D.6 Type-2 Three Term Trapezoidal with Threshold - 0.8 265
D.7 Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 1 267
D.8 Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 0.9 . . . 268
D.9 Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 0.8 . . . 269
D.10 Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 1 271
D.11 Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 0.9 . . . 272
D.12 Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 0.8 . . . 273
D.13 Type-2 Seven Term Trapezoidal Min Surfaces 274
D.14 Type-2 Seven Term Trapezoidal Product Surfaces 275
D.15 Type-2 Three Term Trapezoidal Min Surfaces 276
D.16 Type-2 Three Term Trapezoidal Product Surfaces 277
D.17 Type-2 Three Term Trapezoidal Triangular 300 Min Surfaces 278
D.18 Type-2 Three Term Trapezoidal Triangular 300 Product Surfaces 279
D.19 Type-2 Three Term Trapezoidal Triangular 305 Min Surfaces 280
D.20 Type-2 Three Term Trapezoidal Triangular 305 Product Surfaces 281
D.21 Simulator Input 281

List of Tables

2.1 Summary of classical set theory operator properties 15
2.2 An example Boolean logic truth table. 16
2.3 Number of Hits in Google for Robot Soccer Strategies 67

3.1 Player Identification and Role 83
3.2 Colour Definition at Intensity of 120 83

4.1 No Load Mean Clix for Selected Motor PWM Demands 99
4.2 Loaded Mean Clix for Selected Motor PWM Demands 100

5.1 Seven Term Fuzzy Logic Controller 114
5.2 Results for Seven Term Controllers 118
5.3 Five Term Fuzzy Logic Controller 119
5.4 Three Term Fuzzy Logic Controller 120
5.5 Step RMSE for 7, 5 and 3 Term Controllers 123
5.6 Step Change RMSE Means and SDs 123
5.7 Inertia RMSE for 7, 5 and 3 Term Controllers 125
5.8 Inertia RMSE 7, 5 and 3 Terms Means and SDs 125
5.9 Mean and SD of All FLCs 126

6.1 Seven Term Fuzzy Logic Controller 132
6.2 Three Term Fuzzy Logic Controller 132
6.3 MF Run Numbers for Type-1 and Type-2 FLCs 136
6.4 Type-1 FLCs against Type-2 137
6.5 Type1 against Type-2 within Inertia 138
6.6 RMSE for Type-1 and Type-2 FLCs with Noise 140

6.7 RMSE for Type-1 and Type-2 FLCs with Noise 141
6.8 T2Trap7 RMSE for Dual Surface Average Thresholds within Membership Thresholds 149
6.9 T2Trap3 RMSE for Dual Surface Average Thresholds within Membership Thresholds 150
6.10 Minimum T2Tri300 for Dual Surface Average Thresholds within Membership Thresholds 152
6.11 Type1 and Average Interval Type2 RMSE Comparison 152
6.12 RMSE for the Dual Surface Controller without Noise 153
6.13 RMSEs for the Dual Surface Controller with Noise 154

7.1 Simulation Results for Dual Surface Selection 161
7.2 Hypothesis Tests for PID Controllers 178
7.3 Hypothesis Tests for PID,AveT2,T1 and DST2 Controllers on Robot 1 . . 184
7.4 Hypothesis Tests for PID,AveT2,T1 and DST2 Controllers across Robots 185
7.5 Hypothesis Tests for Dual Surface and Average Response for Robot 1 only with Ramp 189
7.6 Hypothesis Tests Dual Surface and Average Response for 3 Robots with Ramp 189

A.1 Robot Specification 214

B.1 No Load Mean and SD Clix for Left and Right Motor PWM Demands . . 216
B.2 Loaded Mean and SD Clix for Left and Right Motor PWM Demands . . . 216
B.3 Robot Response to Proportional Action 217
B.4 Robot Response to Proportional and Integral Action 218
B.5 Robot Response to Proportional(40),Integral and Differential Action . . . 219
B.6 Robot Response to Proportional(50),Integral and Differential Action . . . 220
B.7 Robot Response to Proportional(60),Integral and Differential Action . . . 221
B.8 Robot Response to Proportional(70),Integral and Differential Action . . . 222
B.9 Robot Response to Proportional(80),Integral and Differential Action . . . 223
B.10 Robot Response to Proportional(100),Integral and Differential Action . . 224
B.11 Robot Parameter Definition File 225

C.1 Seven Term Type-1 and Type-2 Gauss Parameters 231
C.2 Type-1 Seven Term Trapezoidal MF Parameters 231
C.3 Type-1 Seven Term Triangular MF Parameters 232
C.4 Type-1 Seven Term Trapezoidal Triangular1 MF Parameters 232
C.5 Type-2 Seven Term Trapezoidal MF Parameters 237
C.6 Type-2 Seven Term Triangular MF Parameters 239
C.7 Type-2 Seven Term Trapezoidal Triangular1 MF Parameters 240
C.8 Type-1 Five Term Trapezoidal Triangular1 MF Parameters 240
C.9 Type-1 Three Term Trapezoidal Triangular1 MF Parameters 241
C.10 Type-1 Three Term Trapezoidal MF Parameters 241
C.11 Type-2 Five Term Trapezoidal Triangular1 MF Parameters 245
C.12 Type-2 Three Term Trapezoidal Triangular1 MF Parameters 247
C.13 Type-2 Three Term Trapezoidal MF Parameters 249
C.14 Key to Type-1 and Type-2 FLC Response Graphs 253
C.15 Step Change Response Results 255
C.16 Inertia Change Response Results 256
C.17 PID Response Results 256

D.1 Type-2 Seven Term Trapezoidal MF Parameters 258
D.2 Type-2 Three Term Trapezoidal MF Parameters 262
D.3 Type-2 Three Term Trapezoidal Triangular 300 MF Parameters 266
D.4 Type-2 Three Term Trapezoidal Triangular 305 MF Parameters 270
D.5 Simulator Input 276
D.6 MF Run Numbers for Type-1 and Type-2 FLCs with Noise 282
D.7 MF Run Numbers for Type-2 Thresholds 282
D.8 Varying Thresholds with Inertia and No Noise for Type-2 Membership Functions 283
D.9 Noise within Threshold within MF 284
D.10 MF within Threshold within Noise 285
D.11 Noise within MF within Threshold 286
D.12 Threshold within MF within Noise 287
D.13 Type-1 FLCs against Type-2 FLCs (Full) 288

D.14 Type1 against Type-2 within Inertia (Full) . . . 289
D.15 Type-1 against Type-2 with Noise (Full) . . . 290
D.16 Varying Thresholds with Inertia for Type-2 Membership Functions (Full) 291
D.17 Noise within Threshold within MF (Full) . . . 292
D.18 Noise within MF within Threshold (Full) . . . 293
D.19 Threshold within MF within Noise (Full) . . . 294
D.20 MF within Threshold within Noise (Full) . . . 295
D.21 Results Tri3 Test6 Product Rule, MF thresholds . . . 296
D.22 Results Tri3 Test 6 Min rule MF Th=1, DS Th Varied . . . 296
D.23 Results Tri3 Test 6 Min rule MF Th=0.9, DS Th Varied . . . 297
D.24 Results Trap3 Test 6 Min rule MF Th=1,0.9,0.8, DS Th Varied . . . 298
D.25 Results Trap7 Test 6 Min rule MF Th=1,0.9,0.8, DS Th Varied . . . 299
D.26 Results Tri3 Test 6 Min rule MF Th=1,0.8, DS Th Varied . . . 300
D.27 Type-1 Trap Ramp and Step . . . 301
D.28 Type-1 Tri Ramp and Step . . . 301
D.29 Type-2 Trap Ramp,Step and Threshold . . . 302
D.30 Type-2 Tri Ramp,Step and Threshold . . . 303
D.31 Type-1 Seven MF Trap Ramp and Step . . . 304
D.32 Type-2 Seven MF Trap Ramp and Step . . . 305
D.33 Type-2 Seven MF Trapezoidal Threshold . . . 306

E.1 Simulation Results for Dual Surface Selection . . . 308
E.2 Lower Surface for Dual Type-2 Controller . . . 309
E.3 Upper Surface for Dual Type-2 Controller . . . 310
E.4 Average Type-2 Surface for Miabot Controller . . . 311
E.5 Type-1 Surface for Miabot Controller . . . 312
E.6 Difference Between Upper Surface and Lower Surface . . . 313
E.7 Difference between Average Type-2 Surface and Type-1 Surface . . . 314
E.8 Key to Tests for Three Controller Comparison . . . 315
E.9 Results for Three Controller Comparison Tests . . . 316
E.10 Means and SDs by Controller . . . 317
E.11 Means and SDs by Speed Demand . . . 317

E.12 Results for Threshold = 70, No Ramp 318
E.13 Results for Threshold = 70, Ramp Applied 319
E.14 Mean and SD for the Dual Surface Type-2 Controller - Threshold=70 . . 320
E.15 Mean and SD for the Average Type-2 Controller 320
E.16 Results for Minimum Dual Controller with Ramp 320
E.17 Results for Maximum Dual Controller with Ramp 321
E.18 Threshold Results - for Experiment 109 321
E.19 Robot mean speed and standard deviation for each controller type 322
E.20 Average Speed for Controller Types 322

Chapter 1

Introduction

This thesis reports on the research carried out and the results obtained by applying type-2 fuzzy logic systems to micro robots with limited on-board memory and CPU power. In particular it is concerned with developing a type-2 fuzzy logic controller which can successfully control a micro robot with limited on-board memory and CPU power in the real world. This research proposes a novel dual surface controller type 2 fuzzy logic controller which outperforms the standard interval type-2 fuzzy logic controller, and as such, is a significant contribution to the sphere of fuzzy logic controllers.

In this chapter the main issues that the research addresses are set out. The reasons why generally type-2 fuzzy logic systems produce better models than type-1 fuzzy logic systems is discussed. The application of fuzzy logic and fuzzy logic controller development, together with other robot controllers are identified. The motivation for this research is given together with its aims and objectives. The final part of the chapter is the organisation of the rest of the thesis.

1.1 Fuzzy Logic Systems

Since the publishing of the seminal work "Fuzzy Sets" by Lotfi Zadeh in 1965 [1], there has been considerable progress in the development of fuzzy logic systems over the past 45 years [2]. However the ability of type-1 fuzzy logic sets to model uncertain concepts has increasingly been questioned. Researchers are frequently proposing the use of type-2 fuzzy sets as a richer model of uncertainty.

Type-1 Fuzzy Logic Systems

A crisp set reports assertions to be binary true or false. Either you are a member of the set or not. The crisp set Tall is depicted in Figure 1.1a) by a function, called the membership function that describes the crisp set Tall. The set Tall is defined by its membership function to be a height measurement $\geq$ 174cm and $\leq$ 195cm. So heights of 175cm, 180cm and 188cm are all members of Tall. However the set does not report the fact that some members are taller than others, so this information cannot be modelled by a crisp set. There are many examples where degrees of category are modelled as crisp sets, such as warm, long and fast. Fuzzy sets have the property that their members can have continuous membership values [0,...,1]. The boundaries are graduated and membership of the sets can be partial. In our crisp set Tall, a height of 175cm would be classified as Tall. A casual observer might be uncertain if the person as tall, but be tempted to put them into the set Tall. Fuzzy assertions however are not binary true or false, but have a degree of truth. Due to the property that fuzzy sets have a graduated boundary, Aristotle's Law of the excluded middle, $A \cup \neg A = X$, is broken. This allows all heights to be members or non-members of the fuzzy set Tall. Figure 1.1b) describes the fuzzy set Tall, by its membership function. Now there is a way to capture and model this type of data representation. In our example a person of 180cm could be a member of Tall with a membership grade of 0.4. The person of height 188cm could have a membership grade of 0.85 and the 175cm person could be a member with membership grade of 0.1. If we have another type of height called Medium then the height of 175 could have a membership grade of 0.8, 180cm a membership grade of 0.3 and 188cm a membership grade of 0. The height information that could not be modelled by a crisp set can now be modelled by a fuzzy set. This can be applied to any vague concept. This leads to an understanding of the concept of vagueness to be something that cannot be adequately modelled by a crisp set. Fuzzy logic systems are based upon knowledge that is captured in fuzzy sets and rules. Using actual facts and observations, the process of reasoning and making decisions is provided. Due to being able to model the vagueness of a concept with a fuzzy set, an improvement in the decision making process occurs.

In order to use fuzzy sets type-1 fuzzy logic systems have extended the crisp logical operators 'and', 'or' and 'implies'. This allows the processing of fuzzy production

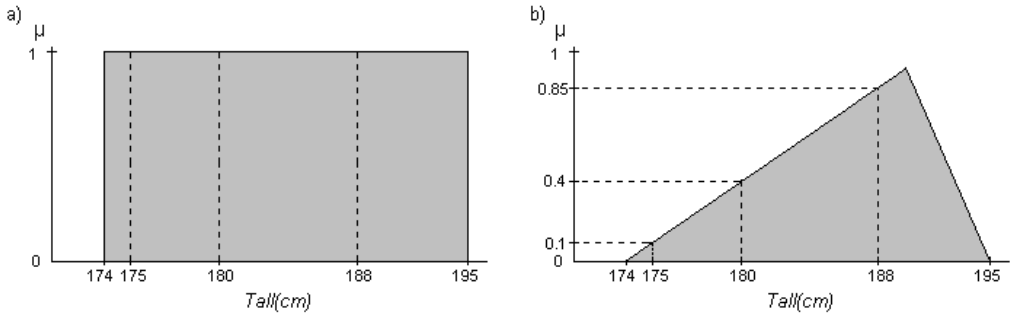


Figure 1.1: a) the Crisp Set Tall. b)The Type-1 Fuzzy Set Tall.

rules to occur. During the inference processing, when a rule is fired then it is fired to a degree. This means that the vagueness of the variables is passed on, in keeping with the original meaning. In 'Computing with Words' [3] Zadeh expected that fuzzy systems and computing in general would be able to use natural language to describe and execute requirements. However the reality is that fuzzy logic systems are dominated by control system applications.

One of the drawbacks of type-1 fuzzy logic, is concerned with the concept of uncertainty. Uncertainty is understood to be when the boundaries of a concept are vague. A type-1 fuzzy logic cannot capture uncertainty only vagueness. In our example a height of 188cm is 0.85 in Tall requires that everything about the concept Tall is known and an accurate model exists. In the Baltic states the average height is higher than the UK, so the environment where the measurement takes place introduces uncertainty. Also a group of experts might disagree as to the value of 0.85 membership grade for 188cm, and also their opinions can alter over time.

Type-2 Fuzzy Logic Systems

In order to capture uncertainty the type-1 fuzzy logic system has to be extended. In our example - 188cm is 0.85 membership grade in the set Tall, becomes hedged with the word 'about'. This gives the statement - 188cm is about 0.85 membership grade in the set Tall. This means that the membership function of the fuzzy set has to capture the vagueness of the membership grade. This is achieved by modelling the secondary membership functions as type-1 fuzzy sets. So in a type-2 fuzzy set the boundaries are graduated allowing

the modelling of vagueness. The boundaries themselves are also vague about the knowledge of the concept, that is they are uncertain. The class of type-2 fuzzy set described here is a Generalised type-2 fuzzy set and is usually referred to as a type-2 fuzzy set.

The price of having uncertainty in the type-2 Fuzzy Logic System model is increased computation complexity. The type-1 model is based on a 2-D representation which maps the domain elements of the set to values [0,..,1]. By having the secondary membership grades as type-1 fuzzy sets, the model becomes 3-D. Now the process is to map the domain elements of the boundary graduated sets to a function which than maps onto the membership grade sets. The Cartesian product of this is potentially massive and seriously impacts and increases the complexity of type-2 systems.

In order to reduce the complexity, another method is to represent type-2 fuzzy logic systems using interval type-2 fuzzy sets. All points in each of the secondary membership functions are set at unity, so each secondary membership can be represented as an interval set. An interval set $\tilde{A}$ contains two elements, a lower bound which is denoted $\underline{\tilde{A}}$ and an upper bound denoted $\overline{\tilde{A}}$, such that $\tilde{A} = [\underline{\tilde{A}}, \overline{\tilde{A}}]$. All points between these bounds are implicit elements of the interval set. By using interval type-2 fuzzy sets the processing is reduced, due to the third dimension in the interval type-2 fuzzy sets being unity and thus ignored in the computation. However this causes a loss of accuracy in the final result.

Fuzzy Logic Rule Base

Whether a type-1 or a type-2 fuzzy logic system is used the fuzzy logic rule base is the same. Fuzzy rules are of the form 'IF antecedents THEN consequents'. The antecedents are combinations of one or more input variables, connected by the fuzzy operators 'AND' or 'OR'. The consequents are types of the output variable. For example using the Mamdani rules base:

IF Room is Hot AND Weather is Warm THEN Heating is Off
IF Room is Warm AND Weather is Cold THEN Heating is Medium

The values for the types are described by membership functions which are usually defined by expert opinion. Usually this is sufficient to set up a fuzzy logic system.

1.1.1 Applications of Fuzzy Logic

With the boom in consumer goods and the investments made by Japanese manufactures fuzzy logic controllers can be found in such diverse equipment as washing machines and auto focus cameras, refrigerators, photocopiers and televisions. With the development of fuzzy programmable logic controllers by companies like Moeller, the industrial application of fuzzy logic is ever growing. In fact wherever decisions have to be made such as data analysis or signal processing, fuzzy logic can be used to make them. In the industrial world fuzzy logic has been used to solve many problems, these include Power Generation, Turbine Engine Control, Heating, Pumping, Air Conditioning, Electric Motor and Voltage Regulation and Motion Control.

1.1.2 Fuzzy Logic Controller Development

The first practical use of fuzzy logic was by Mamdani and Assilian in 1974/5 [4]. They successfully used fuzzy logic to control a steam engine. In 1977 Siemens used a fuzzy set concept to control traffic lights [5]. The first industrial application was in Denmark. The purpose was to control a cement kiln. The project was started in 1976 and was completed in 1982 [6]. Yasunobu working for Hitachi in Japan started work in 1979 on a project to control the subway trains of Sendai [7]. The system was successfully commissioned in 1987. Following a development program, Fuji Electric offered the first commercial fuzzy logic controller in 1985.

1.1.3 Other Robot Controllers

In non-reactive architectures robot controllers provide the reasoning on what to do. Too close then move away, or nothing near then go faster. Many different controllers have been proposed over the years, including Neural Networks [8], Potential Field [9], Fuzzy Logic [10], Sensor Fusion [11], and more recently Artificial Immune Systems [12, 13] and Idiotypic Networks [14, 15] carried out at the University of Nottingham. However the reason for selecting fuzzy logic systems as the controllers of choice for the research was the wealth of departmental expertise in the field of fuzzy logic systems.

1.1.4 Robot Football

There are two organisations that are concerned with robot football, The RoboCup Federation[1] and FIRA (Federation of International Robot-soccer Association)[2]. The aim of the RoboCup project is 'By 2050, develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer'. Robot football is played in many different configurations including humanoid robots. Each league is governed by a comprehensive set of rules that ensure the physical robots are matched. However the implementation of architectures and strategies are left entirely to the team designers. As demonstrated in the Dapra challenges, competition is a massive driver of progress [16]. Robot football games are fast and dynamic, especially in the FIRA mirosot leagues. The games are unpredictable and random which mirrors the real world. Due to the robots moving so quickly and colliding there are lots of failures, which the teams have to handle.

1.2 Motivation

The theme that runs through the previous discussion is that the robots are medium sized and the motors used are large. Indeed, the engine used by Hagras [17] is a marine diesel engine capable of getting a boat across the Atlantic.

In contrast, the motivation of this research is to facilitate and enable micro robots to successfully operate in a robot football team environment. In order to play robot football, the robots have to be able to move accurately at speeds of over $1 msec^{-1}$ and intercept the golf ball football. They must be able to control the ball and change direction quickly. They are subject to rapid accelerations and decelerations. The environment is very harsh and very noisy, with many high speed collisions occurring with other team members and opposition robots. This real world problem is the driving force behind this research.

A major issue of the micro robots is the limited on-board memory and CPU power and the space available to hold boards and batteries within the allowed shape and size of the robots. This was solved by the manufacturers by supplying a PID controller to control the robot motors, which required very little code and a very small amount of data storage.

[1] www.robocup.org
[2] www.fira.net

The Robot Soccer Engine (RSE) supplied to control the robots during a football match configures all the robots with the same PID parameters. However the characteristics of each robot player in the football team are different. The goalkeeper has to be able to accurately intercept or block the ball usually from a standing start, and so requires very fast reaction times. Where as a striker has to be able to control the ball and direct it towards the opponents goal, avoiding any defenders trying to block its path. The control requirements for this demand accuracy of speed and position.

The need to have different PID controllers for each player in the robot football team led to the use of PID controllers being questioned. Would a fuzzy logic controller provide an alternative solution to the problem? Having posed this question, the next question is what type of fuzzy controller to use. The possibilities were either a type-1 or a type-2 controller. This leads to the question and challenge of how to implement a fuzzy logic controller on the micro robot given the major issue of limited on-board memory and CPU power. Indeed this issue channelled the direction of the research aims and objectives and the solution that was consequently arrived at.

Generally throughout the research the motivation and aim is to produce a fuzzy logic controller that operates successfully in the real world. The major issue of having limited on-board memory and CPU power on the micro robot has a large impact by constraining the scope of the research to a solution that can be implemented under these conditions. Having shown that it is possible to develop a fuzzy logic controller for a micro robot, then it expected that the controller can be scaled up and implemented on medium and large robots successfully. The aim is ultimately to be able to implement and abstract the fuzzy logic controller onto any fuzzy logic system.

1.3 Research Aims and Objectives

The central aim for this research is to discover whether alternatives to PID control, particularly fuzzy controllers of either type-1 and type-2, can make substantial, observable difference to the micro robots within a robot football environment. More specifically, could alternative controllers actually result in a better game being played by the robots. This central aim led to three specific research objectives:

1. *Compare PID, type-1 and type-2 controllers on a micro robot with limited on-board memory and CPU power.* Various alternative controllers were designed and then implemented in software in a form suitable to be deployed on the robot football micro robots. This meant that the computational resources of the robots were a hard constraint on the implementation process. As a result, the fuzzy controllers were implemented as static look-up tables of the control surface which was produced by the various fuzzy controllers. This led to the observation that there was no essential difference between a control surface look-up table produced from a type-1 fuzzy controller and that produced by a type-2 controller.

2. *Investigate a novel dual surface type-2 controller.* Following the above observation, a novel form of type-2 controller was investigated, in which the upper control surface and lower control surface (see Chapter 6 for full explanation of these terms) were both stored within the micro robot. This allowed more of the information produced by the type-2 controller to be represented within the robot, potentially providing more sophisticated type-2 control. The results showed that in simulation, generally the type-2 fuzzy logic controller outperformed the equivalent type-1 controller both with and without Gaussian noise being applied to the system. Both fuzzy controllers out-performed the PID reference controller. This is consistent with the theoretical and practical observations made for simulated experiments.

3. *Measure the effects of these alternative controllers on real world performance.* While much literature has examined the differences between type-1 fuzzy control and PID control, and more recently (but to a lesser extent) the additional benefits of type-2 control, often these experiments have been in terms of easily measurable parameters such as the RMSE (root-mean-squared-error) between the controller output and the required set-point. Studies of the effect of these controller differences on the observable real world behaviour have been less common — does, for example, a 1% reduction in controller RMSE lead to any observable difference in behaviour? The last objective of this work was to examine the controller differences in terms of whether they lead to any substantial, observable differences in high-level behaviour of the micro robots — essentially, are any differences in the

controllers enough to lead to observable improvements in the ability of the robots to play football?

1.4 Organisation of the Thesis

The remainder of the thesis is organised as follows.

- Chapter 2 reviews the relevant literature covering the research domain of this thesis. The first part is concerned with common problems associated with robots. There is a detailed literature review of the architectures and systems used in multi-robot teams. Significant approaches to the solution of these problems are discussed and critically analysed. The taxonomies that are available to describe the teams are analysed. The control methodologies used in micro-robot football are explained and critically analysed. The second part of the literature review is a discussion about fuzzy logic. It origins and development as a type-1 logic and the development and advancement of type-2 fuzzy logic.

- Chapter 3 describes the environment within which the robots operate. This includes a detailed description of the robot hardware, the firmware of the robot and the software that is used to control the robots. The system specifications and configurations are detailed. The performance attributes of the robots are discussed along with problems that were encountered whilst running the robots. There is a discussion about the difficulty of getting meaningful data from the robot whilst it was running.

- Chapter 4 investigates the performance of the micro robots using an independent PID controller.

- Chapter 5 describes the investigations into various alternative type-1 and type-2 fuzzy controllers, with particular emphasis on the design of membership functions.

- Chapter 6 is concerned with the design of the novel dual-surface type-2 fuzzy controller, its associated membership functions and the simulation experiments that were carried out to evaluate its performance.

- Chapter 7 describes the real world experiments that were performed to test the results obtained from the simulation. These include the real world PID experiments that were carried out to ascertain the best tuning parameter set for the robot when in PID mode. The effect of using a ramp to control the power input to the robot when in PID mode. A smoothing algorithm was trialled. A series of experiments comparing all different controllers at different power values. Also tested a voting system that picked from the upper, lower or averaged surface value, depending on the state of an error metric. A statistical breakdown of each test is given. This demonstrated that in the real world that there was no significant difference between the robot controllers.

- Chapter 8 concludes the thesis with a discussion on the contribution made and the limitations of what was done. The future work that is available and dissemination of the research.

Chapter 2

Literature Review

The previous chapter defined the goals and scope of this thesis. This chapter presents the fields of classical set theory and the theory of fuzzy sets and systems. In addition, the chapter introduces the notion of type-1 and type-2 fuzzy sets and explains their differences together with the theory underpinning type-2 fuzzy sets. Architectures of robotic systems are introduced together with strategies that are available to control the robots. The PID controller is briefly reviewed as this the most common form of motor control in micro robots. Robot football, the motivation for this research is touched upon, and is then followed by an examination of potential areas of research interest. Finally recent work on type-2 fuzzy sets and systems including new methods of defuzzification are reported.

2.1 Theory of Fuzzy Sets and Systems

A fuzzy set is a generalization of a classical set, but differs in that it allows objects to take partial membership of its sets, with the real-valued degree of membership ranging from 0 to 1. Since much of the theory of fuzzy sets is rooted in classical set theory, this chapter begins with a introduction to the fundamental concepts of classical set theory and Boolean logic. It goes on to define fuzzy sets and show how they differ from classical sets. The application of Boolean logic to fuzzy sets and systems, which is known as fuzzy reasoning, is also discussed along with a description of fuzzy membership functions, fuzzy operators and defuzzification.

2.1.1 Classical Set Theory

A set is defined as a collection of unique objects termed *elements* or *members* of the set. In set theory, as Cantor [18] defined and Zermelo [19] and Fraenkel [20] axiomatised, an object is either a member of a particular set or not. For example, a pedigree Bengal cat is in the set of all pedigree cats, but a pedigree Pekingese dog is not. In dealing with sets it is also necessary to define a *universe of discourse*, which provides the set of allowable values for objects, i.e. the relevant set of entities that are being dealt with. For example, the universe of discourse could be the set of all animals, the set of real numbers or the set of natural numbers.

In representing sets mathematically, the convention is to use capital letters for sets, and lower case letters for objects and set members. In addition, the universal set, defined as the set of all sets, is represented as $\mathbb{U}$, and a set with no elements, called a *null*, or *empty* set and is denoted as $\emptyset$. A set may be defined by listing all its members (the list method), for example:

$$A = \{a_1,\ a_2,\ ...,\ a_n\},$$

(where n is the number of members of the set), or by defining the criteria that each member must meet in order to be considered a member of the set (the rule method):

$$A = \{a \mid a \text{ meets some condition(s)}\},$$

for example,

$$A = \{a \mid 0 \leq a \leq 10\}.$$

The relationship between a set A in a universe of discourse X and an object x is expressed mathematically as:

$$x \in A,$$

if x is a member of A and by:

$$x \notin A,$$

if x is not a member of A. Alternatively, a zero-one *membership function* for A (also known as a characteristic function, discrimination function or indicator function) may be

introduced, which encapsulates the notion that an object $x \in \mathbb{U}$ is either a full member of A or not a member of A at all. The membership function is denoted by $\mu_A(x)$ and is such that:

$$\mu_A(x) = \begin{cases} 1 & \text{if and only if} \quad x \in A, \\ 0 & \text{if and only if} \quad x \notin A. \end{cases}$$

Therefore, μ_A maps all elements of the universal set into the set A with values 0 and 1, which can be written in functional notation as:

$$\mu_A : \mathbb{U} \rightarrow \{0,1\}.$$

When every element in the set A is also a member of set B, then A is said to be a *subset* of B, and this is written as:

$$A \subseteq B.$$

For example, $\{1, 2\}$ is a subset of $\{1, 2, 3\}$, but $\{1, 4\}$ is not. It thus follows that if every element in A is also in B and every element in B is also in A, i.e. $A \subseteq B$ and $B \subseteq A$, then the sets A and B are equal:

$$A = B.$$

Alternatively, if at least one element in set A is not in set B, or at least one element in B is not in A, then A and B are not equal:

$$A \neq B.$$

Set A is a *proper subset* of B if A is a subset of B but A and B are not equal, i.e. $A \subseteq B$ and $A \neq B$. This is represented as:

$$A \subset B.$$

In this case, every element of A is in B, but B has additional elements not found in A.

Just as arithmetic is concerned with operations on numbers, set theory is concerned with operations on sets. Some of the more common operators are summarized below:

- The *union* of sets A and B, denoted by $A \cup B$, is the set of all distinct elements in both sets. For example, the union of $\{1, 2, 3\}$ and $\{2, 3, 4\}$ is the set $\{1, 2, 3, 4\}$.

This is similar to the logical "OR" operator, see Section 2.1.2.

- The intersection of the sets A and B, denoted by $A \cap B$, is the set that contains all elements of A that also belong to B (or equivalently, all elements of B that also belong to A), but no other elements. The intersection of $\{1, 2, 3\}$ and $\{2, 3, 4\}$ is the set $\{2, 3\}$. This is similar to the logical "AND" operator.

- The complement of set A relative to set X, denoted A^c, $\bar{A}$ or A' is the set of all members of X that are not members of A. This terminology is most commonly employed when X is the universal set $\mathbb{U}$, as in the study of Venn diagrams. The complement of $\{1, 2, 3\}$ relative to $\{2, 3, 4\}$ is $\{4\}$, while, conversely, the complement of $\{2, 3, 4\}$ relative to $\{1, 2, 3\}$ is $\{1\}$. This is similar to the logical "NOT" operator and can also be written as "not A" or $\neg A$. In classical set theory, the complement is reflexive, i. e., $\neg(\neg A) \equiv A$.

- The symmetric difference between sets A and B is the set with members that only belong to A or B. For instance, for the sets $\{1, 2, 3\}$ and $\{2, 3, 4\}$, the symmetric difference is the set $\{1, 4\}$.

- The set difference between sets A and B, expressed as $A - B$, is the set of elements that are in A, with those that are in B subtracted out. Therefore, $A - B$ is the set of elements that are in A and not in B, i. e. $A - B \equiv A \cap \neg B$. For instance, for the sets $\{1, 2, 3\}$ and $\{2, 3, 4\}$, the set difference is the set $\{1\}$.

- The Cartesian product of A and B, denoted $A \times B$, is the set with members that represent all possible ordered pairs (a, b) where a is a member of A and b is a member of B. For the sets $\{1, 2, 3\}$ and $\{2, 3, 4\}$, the Cartesian product produces the set $\{(1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$. The operation can be expressed mathematically as:

$$A \times B = \{(a,b) | a \in A, b \in B\}. \tag{2.1}$$

The examples quoted above use only two sets, but it is important to note that these operators can be used on any number of sets and that they may be combined to produce

more complex operators. The mathematical properties of the union, intersection and complement operators are summarized in Table 2.1.

Table 2.1: Summary of classical set theory operator properties

Property	Example
Involution	$\neg\neg A \equiv A$
Commutativity	$A \cup B \equiv B \cup A$ $A \cap B \equiv B \cap A$
Associativity	$(A \cup B) \cup C \equiv A \cup (B \cup C)$ $(A \cap B) \cap C \equiv A \cap (B \cap C)$
Idempotence	$A \cup A \equiv A$ $A \cap A \equiv A$
Distributivity	$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$
Absorption	$A \cup (A \cap B) \equiv A$ $A \cap (A \cup B) \equiv A$

Classical sets are often referred to as *crisp* sets to distinguish them from fuzzy sets, see Section 2.1.4.

2.1.2 Boolean Logic

Boolean logic is named after George Boole, who first defined an algebraic system of logic in the mid 19^{th} century. His system defines a number of logical operators, but includes only two possible truth values, true (1) and false (0). These values can be combined using the defined logical operators to produce a *vocabulary* of Boolean logic, which can be expressed as a *truth table*. An example table, which demonstrates the results of applying the logical operators AND, OR, NOT, EQUIVALENCE and IMPLICATION to variables A and B is shown in Table 2.2 below. The operators are symbolized by standard notation, i.e. $\wedge$, $\vee$, $\neg$, $\Leftrightarrow$ and $\Rightarrow$ respectively.

The most simple and most commonly used operators are AND and OR. The AND operation returns true if and only if all inputs are true; i.e. the output is false if any of the

Table 2.2: An example Boolean logic truth table.

A	B	$A \wedge B$	$A \vee B$	$\neg A$	$\neg B$	$A \Leftrightarrow B$	$A \Rightarrow B$
0	0	0	0	1	1	1	1
0	1	0	1	1	0	0	1
1	0	0	1	0	1	0	0
1	1	1	1	0	0	1	1

inputs are false. Alternatively, the OR operation returns true if any of the inputs are true. The NOT operator is applied to a single input and returns its opposite, i.e. false if the input is true and vice versa. The EQUIVALENCE operator returns true if two inputs are the same and false if they are different. The IMPLICATION operator is more complex and most relevant to the field of fuzzy logic, since it deals with antecedent and consequent parts of complex logical statements. When applied to two simple logical statements A (the antecedent part of a more complex statement or the *premise*) and B (the consequent part or *conclusion*), if B is a logical consequence of A then the value of $A \Rightarrow B$ is true. In fact, it is worth noting that the operator returns false only in the case where the premise is true and the conclusion is false. The logical reasoning underlying this is interpreted as:

- A true premise implies a true conclusion, therefore if T represents true, $T \Rightarrow T \equiv T$.
- A true premise cannot imply a false conclusion, therefore $T \Rightarrow F \equiv F$.
- Anything may be concluded from a false assumption, so $F \Rightarrow F \equiv T$ and $F \Rightarrow T \equiv T$.

Note that the IMPLICATION operator is not commutative, i.e., $A \Rightarrow B \neq B \Rightarrow A$.

2.1.3 Boolean Tautologies

In any particular logic system, reasoning procedure relies on *tautologies*, or rules that remain true in that logic system regardless of the values assigned to the variables involved. Four examples of Boolean tautologies are discussed below.

The first tautology is called *modus ponens*, sometimes referred to as "affirming the antecedent" or "the law of detachment". Modus ponens states that if $P \Rightarrow Q = 1$ and $P = 1$, where 1 represents true, then it can be concluded that Q, the consequent of the conditional claim, must be true as well. As an example, suppose P is the logical statement "Today is Tuesday" and Q is the logical statement "I have to go to work today". If P implies Q, i.e., if the complex logical statement "If today is Tuesday then I have to go to work today" holds true and P is true, i.e. it is Tuesday, then it is also true that I have to go to work today, i.e. Q is also true. This can be expressed mathematically as:

$$((A \Rightarrow B) \wedge A) \Rightarrow B \equiv 1. \tag{2.2}$$

Equation 2.2 holds true whatever value the inputs for A and B take. In Artificial Intelligence, modus ponens is often called forward chaining.

The second tautology is called *modus tollens*. This can be expressed mathematically as:

$$((A \Rightarrow B) \wedge \neg B) \Rightarrow \neg A \equiv 1, \tag{2.3}$$

and can also be referred to as "denying the consequent". Modus tollens states that if $P \Rightarrow Q = 1$ and $Q = 0$, where 1 represents true and 0 represents false, then it can be concluded that P, the premise of the conditional claim, must be false as well. In the example above, if the complex logical statement "If today is Tuesday then I have to go to work today" holds true and Q is false, i.e. I do not have to go to work today, then one can conclude that it cannot be a Tuesday, i.e. P is false.

The third tautology is known as *hypothetical syllogism* and can be represented as:

$$((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C) \equiv 1. \tag{2.4}$$

This can be interpreted as "If A implies B and B implies C then A also implies C". The final tautology is called *contraposition* and is written mathematically as:

$$(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A) \equiv 1. \tag{2.5}$$

The interpretation of this tautology is that "If A implies B then $\neg B$ implies $\neg A$".

A *contradiction* is the opposite of a tautology, i.e., the statement made is always false. For more information on classical logic see Klir & Folger [21].

2.1.4 Fuzzy Sets

Fuzzy sets were first introduced by Zadeh in 1965 [1] to model the imprecision and uncertainty inherent in assigning membership of elements to real-world sets, for example the set of *short* people or the set of *young* people. The theory provided formalized tools for dealing with such imprecision or vagueness in real world applications, allowing many complex decision-making problems to be simplified.

The fuzzy set is a generalization of a crisp or classical set and is characterized by a membership function $\mu(x)$, that takes on values in the interval [0,1], as opposed to the membership functions of crisp sets that can take only the values 0 or 1, see [22]. This means that elements are permitted to take partial membership of fuzzy sets. The *membership grade* of an element x with respect to a set A is denoted by $\mu_A(x)$, with μ_A mapping all elements of the universal set into the set A with values in the continuous interval 0 to 1. This can be expressed in functional notation as:

$$\mu_A : \mathbb{U} \rightarrow [0,1].$$

As with crisp sets, a fuzzy set may be defined formally using the list method. When applied to fuzzy sets, this becomes a list of the strength of membership of each element of a discrete, countable universe of discourse to the set in question, which is represented mathematically as:

$$A = \sum_{i=1}^{n} \mu_i / x_i, \tag{2.6}$$

or

$$A = \{\mu_1/x_1 + \cdots + \mu_n/x_n\}, \tag{2.7}$$

where x_i denotes the i^{th} member of the universe of discourse, μ_i denotes the strength

of membership of element x_i and A is the set of interest. Note that the summation and addition symbols in Equation 2.6 and Equation 2.7 do not represent algebraic summation; they indicate the collection or aggregation of each element. The description of a fuzzy set on a continuous universe is written as:

$$A = \int_U \mu_A(x)/x, \tag{2.8}$$

where U is the universe of discourse and the integral symbol represents a continuous, function-theoretic aggregation operator for continuous variables, not algebraic integration [23].

An example membership function, used to assign membership values to the fuzzy set A, which represents temperatures close to 30 °C, could be:

$$\mu_A(x) = \frac{1}{1 + \frac{(x-30)^2}{30}}. \tag{2.9}$$

Equation 2.9 maps every real number in a continuous universe of discourse into the set of temperatures close to 30 °C, for example 15°C would be assigned a membership value of 0.12, 20 °C would be assigned a membership value of 0.23 and 30 °C would have a membership value of 1.0.

Alternatively, if one is dealing with a discrete universe of discourse, for example element values between 15 and 45 in steps of five, one may express the fuzzy set A as:

$$A = \left\{ \frac{0.12}{15} + \frac{0.23}{20} + \frac{0.55}{25} + \frac{1.0}{30} + \frac{0.55}{35} + \frac{0.23}{40} + \frac{0.12}{45} \right\}, \tag{2.10}$$

for example. Equation 2.10 uses the discrete values obtained from the continuous function of Equation 2.9.

In general, the membership function can assume any shape, but those most commonly used are triangular, Gaussian, Sigmoid, and S-shaped functions.

2.1.5 Fuzzy Operators

Since fuzzy set theory is a generalization of classical set theory, its axiomatic foundation has some important differences [24]. In particular, it violates two fundamental laws of Boolean algebra; the *law of excluded middle* $A \cup \neg A = U$, where U is the universe of discourse, and the *law of contradiction* $A \cap \neg A = \emptyset$. This is because it is possible for an element to have degrees of belonging both to a fuzzy set and its complement. This means that logically equivalent formula from classical set theory are not necessarily equivalent in fuzzy logic.

As discussed in Section 2.1.1, the main operators on any set, whether crisp or fuzzy are intersection, union and complement. In classical set theory these operations are uniquely defined, see [25], but in fuzzy set theory there are many different ways to define them as the operations are based on membership values, which are no longer restricted to $\{0,1\}$. However, it is important to note that any definition of these operations on fuzzy sets must include the limiting case of crisp sets. Example definitions for the above three operators are those given by Zadeh [1]:

- Intersection $A \cap B$: $\mu_{A\cap B}(x) = min[\mu_A(x), \mu_B(x)]$,
- Union $A \cup B$: $\mu_{A\cup B}(x) = max[\mu_A(x), \mu_B(x)]$,
- Complement A': $\mu_{A'}(x) = 1 - \mu_A(x)$.

Other widely-used definitions include the algebraic product for intersection, see Section 2.1.5 and the algebraic sum for union, see Section 2.1.5, but there are an infinite number of other choices. Also, it is worth noting that the selection of an intersection operator may influence the choice of union operator due to the *principle of duality* between them [24]. A fuzzy intersection operator $t(x,y)$ and a fuzzy union operator $s(x,y)$ form a dual pair if they satisfy the following condition:

$$1 - t(x,y) = s(1-x, 1-y). \tag{2.11}$$

The above duality condition ensures that:

$$\neg(A \cap B) \equiv \neg A \cup \neg B, \tag{2.12}$$

(which is always true in classical set theory), still holds in fuzzy set theory. The application of the intersection, union, complement and implication operators to fuzzy sets is discussed in more detail in Sections 2.1.5, 2.1.5, 2.1.5 and 2.1.5 respectively.

Fuzzy Intersection

The set of candidate fuzzy intersection operators is termed *triangular norms* or *t-norms* and is defined by a set of axioms. These are set out below:
A t-norm operator $t(x,y)$ must satisfy the following conditions for any w, x, y, $z \in [0,1]$:

1. $t(0,0) = 0, t(x,1) = t(1,x) = x$ (boundary condition),
2. $t(x,y) \leq t(z,w)$ if $x \leq z$ and $y \leq w$ (monotonicity condition),
3. $t(x,y) = t(y,x)$ (commutativity condition),
4. $t(x,t(y,z)) = t(t(x,y),z)$ (associativity condition).

The boundary condition imposes a generalization to crisp sets and the monotonicity condition implies that a decrease in the membership values of sets X or Y cannot produce an increase in the membership value of $X \cap Y$. The commutativity condition ensures that $X \cap Y \equiv Y \cap X$, and the associativity condition permits the intersection of any number of sets in any order of pairwise groupings.

It is often useful to limit the cluster of fuzzy intersections by taking into account additional conditions as follows:

- t is a continuous function (continuity condition),
- $t(x,x) \leq x$ (subidempotency condition).

Continuity avoids the situation where a small change in the degree of membership to set X or set Y leads to a large change in the degree of membership to $X \cap Y$. The subidempotency

condition deals with degrees of membership of X or Y having the same value x. This axiom expresses that the degree of membership to $X \cap Y$ should not exceed the value of x.

The most common t-norms (besides minimum) are listed below:

- Algebraic product: $t(x,y) = xy$,
- Limited difference: [26] $t(x,y) = \max(0, x+y-1)$,
- Drastic intersection: $t(x,y) = \begin{cases} x & \text{if } y = 1 \text{ ;} \\ y & \text{if } x = 1 \text{ ;} \\ 0 & \text{otherwise.} \end{cases}$

Fuzzy Union

The set of candidate fuzzy union operators is termed *triangular conorms*, *t-conorms* or *s-norms* and is defined by the following set of axioms:
An s-norm operator $s(x,y)$ must satisfy the following conditions for any w, x, y, $z \in [0,1]$:

1. $s(1,1) = 1, s(x,0) = s(0,x) = x$ (boundary condition),
2. $s(x,y) \leq s(z,w)$ if $x \leq z$ and $y \leq w$ (monotonicity condition),
3. $s(x,y) = s(y,x)$ (commutativity condition),
4. $s(x,s(y,z)) = s(s(x,y),z)$ (associativity condition).

As with fuzzy intersection, it is often useful to limit the choice of fuzzy union operators by imposing additional conditions as follows:

- s is a continuous function (continuity condition),
- $s(x,x) \leq x$ (subidempotency condition).

The most common s-norms (besides maximum) are listed below:

- Algebraic sum: $s(x,y) = x+y-(xy)$,
- Boundary sum: $s(x,y) = \min(1, x+y)$, see [26],
- Drastic union: $s(x,y) = \begin{cases} x & \text{if } y = 0, \\ y & \text{if } x = 0, \\ 1 & \text{otherwise.} \end{cases}$

Fuzzy Complement

Zadeh [1] defined membership of a fuzzy complement set as one minus the degree of membership of the original set, but there are an infinite number of other definitions for fuzzy complement operators $c(x)$. Each, however, is characterized by the following axioms:

1. $c(0) = 1, c(1) = 0$,
2. If $x < y$ then $c(x) \geq c(y)$; $x, y \in [0,1]$.

The first axiom defines the boundary conditions and the second defines the fuzzy complement as monotonic increasing, which mimics the crisp complement. This is intuitive since, as the degree of membership of an element in set X increases, its membership in the complement set should decrease. Two further conditions are also very useful when assigning fuzzy complement operators. These are:

- c is a continuous function (continuity condition),
- $c(c(x)) = x$ for all $x \in [0,1]$ (involution condition).

The involution constraint is imposed when the complement of a fuzzy set must be reversible.

Fuzzy Implication

Classical set theory has only one definition of the implication operator, but an infinite number of implication operators are possible with fuzzy sets and systems. Zadeh [27] defined fuzzy implication as:

$$A \Rightarrow B = A \times B, \tag{2.13}$$

where $\times$ indicates the Cartesian product of the two fuzzy sets A and B. If A is a subset of the universe of discourse U and B is a subset of the universe of discourse V, this can be expressed as:

$$A \times B = \int_{U \times V} t(\mu_A(u), \mu_B(v))/(u,v), \tag{2.14}$$

where:

$$U \times V = \{(u,v)|u \in U, v \in V\}, \tag{2.15}$$

from Equation 2.1.

Equations 2.13 and 2.14 show that, under this particular definition of fuzzy implication, the result of $A \Rightarrow B$ is a fuzzy set of ordered pairs $(u,v), u \in U, v \in V$, and that the membership values of (u,v) are given by $t(\mu_A(u), \mu_B(v))$, where t represents some t-norm operator. As an example, suppose that:

$$U = 4+5, \tag{2.16}$$

$$V = 5+6+7, \tag{2.17}$$

$$A = 0.5/4 + 0.7/5, \tag{2.18}$$

$$B = 0.4/5 + 0.8/6 + 0.3/7, \tag{2.19}$$

and that the t-norm is the minimum. Then

$$\begin{aligned} A \times B = {} & 0.4/(4,5) + 0.5/(4,6) + 0.3/(4,7) + 0.4/(5,5) \\ & + 0.7/(5,6) + 0.3/(5,7). \end{aligned} \tag{2.20}$$

There are many other commonly-used implication operators, e.g. the S-implication, QL-implication, and R-implication operators. A thorough review of fuzzy implications is presented in Dubois & Prade [28], and the use of the implication operator within fuzzy logic is discussed further in Section 2.1.6.

2.1.6 Fuzzy Logic

In fuzzy set theory, elements have degrees of membership to sets. Fuzzy logic builds upon this by introducing degrees of truth for statements. In order to facilitate this, the theory defines a concept known as the *linguistic variable*. A variable in the classic sense is a placeholder that can take on any value defined over its universe of discourse. So for example, to describe the temperature of a room, the variable would be *temperature* and the value would be a number, such as 30°C. A linguistic variable differs in that, in addition

to accepting a crisp number as input, it also has any number of fuzzy terms defined over its universe of discourse, for example the linguistic variable for temperature would still be *temperature*, but would incorporate fuzzy *terms* or *linguistic labels* such as *low*, *medium* and *high*. These linguistic labels represent fuzzy sets to which variable values have partial membership and each linguistic label is a subset of the parent linguistic variable, which may be expressed as:

$$\{term_1, term_2, \ldots, term_n\} \subset \text{parent linguistic variable}, \tag{2.21}$$

where n represents the number of linguistic labels. A fuzzy set is constructed in order to capture the meaning of a particular linguistic label; this maps the linguistic variable values to membership values of the fuzzy set using a membership function.

There are two ways to create the terms of a linguistic variable. The first is simply to define them from the beginning, and the second is to modify an existing term. A fuzzy modifier is generally referred to as a *hedge* or *linguistic modifier*. For instance, if the term *low* is already defined and the term *very low* is required, it is not necessary to create the definition from scratch; the term *low* can be altered mathematically by the hedge *very* to create the new term. Hedges are a useful tool since a wide range of terms can be created in a standard way, which permits an expansion of available terms with little extra effort.

For example, to define *very A* one might square the membership values of the linguistic term or fuzzy set A. If A is the fuzzy set *close to 30 °C* and the membership function in Equation 2.9 is used, then the term *very close to 30 °C* would have the membership function defined below:

$$\mu_{very(A)}(x) = \left(\frac{1}{1 + \frac{(x-30)^2}{30}} \right)^2. \tag{2.22}$$

There are many different classes of hedge, for example the powered hedge [29] and the shifted hedge [30], each with their own benefits and drawbacks. In general, the type of hedge chosen relates to the problem that needs to be solved.

Once linguistic variables and terms have been defined it is possible to create fuzzy statements termed *fuzzy propositions* by associating linguistic terms with their parent vari-

ables, for example *temperature is low*. Fuzzy rules and reasoning are then used to evaluate fuzzy propositions, see Sections 2.1.6, 2.1.7 and 2.1.8.

Fuzzy Relations and Rules

Fuzzy propositions can be amalgamated to form more complex statements. When the propositions are based upon different universes of discourse, a *fuzzy relation* is formed. For example, consider the proposition P:

$$P: \ x \text{ is } A \text{ AND } y \text{ is } B, \tag{2.23}$$

where A and B are fuzzy sets with membership functions μ_A and μ_B respectively. The membership function μ_P of the fuzzy relation P is represented as:

$$\mu_P(x,y) = t(\mu_A(x), \mu_B(y)), \tag{2.24}$$

where t is any general t-norm representing the intersection (AND) operation.

Fuzzy rules are used to link an antecedent fuzzy proposition or fuzzy premise to a consequent fuzzy proposition or fuzzy conclusion. For example:

if A then B,

where A is the antecedent and B is the consequent. This is a short-hand way of writing:

if x is A then y is B,

which can also be expressed as:

$A \Rightarrow B$.

It is also possible to create complex fuzzy rules by amalgamating single ones, i.e., by linking antecedent fuzzy relations with consequent fuzzy relations. This results in the

form:

if x_1 is A_1 AND $\ldots x_n$ is A_n then y_1 is B_1 AND $\ldots y_m$ is B_m.

Here, the operator used is intersection (AND) but it could be any valid connective operator, see Zadeh [27]. The amalgamation may also be expressed using the following notation:

$$R = I(t(A_1 \ldots A_n), B_1 \ldots B_m), \tag{2.25}$$

where t is any t-norm, I is the implication function, n is the number of antecedents, m is the number of consequents, and R represents the relation.

If more than one rule is defined then the collection of rules is referred to as a *rule base*. For example, a possible rule base with n rules and m premises per rule may be expressed as:

$$r_1 : \text{if } x_1 \text{ is } A_{1,1} \text{ AND } \ldots x_m \text{ is } A_{m,1} \text{ then } y \text{ is } B_1, \tag{2.26}$$

$$r_k : \text{if } x_1 \text{ is } A_{1,k} \text{ AND } \ldots x_m \text{ is } A_{m,k} \text{ then } y \text{ is } B_k, \tag{2.27}$$

$$r_n : \text{if } x_1 \text{ is } A_{1,n} \text{ AND } \ldots x_m \text{ is } A_{m,n} \text{ then } y \text{ is } B_n. \tag{2.28}$$

This may also be written as:

$$R = \bigcup_n R_n. \tag{2.29}$$

Here, the connective between the premises is an intersection (AND operator). If it is a union (OR operator), the relation is written as:

$$R = \bigcap_n R_n. \tag{2.30}$$

2.1.7 Defuzzification

A fuzzy system transforms a number of variables into a fuzzy result described in terms of degrees of membership of fuzzy sets. Defuzzification is necessary in order to pro-

duce a quantifiable result. For example, rules designed to decide on a change in medication dosage might result in "Decrease Dosage (10%), Maintain Dosage (42%), Increase Dosage (48%)". The defuzzification process transforms these membership values into a single number representing the required dosage change.

There are many possible defuzzification techniques; the simplest is to choose the set with the highest membership value, and then convert it into a single value. However, such an approach is generally ineffective as the other sets are not taken into consideration; a useful defuzzification process should combine the results of the rules together in some way. Graphical defuzzification methods are a popular choice, for example, for triangular-shaped membership functions, part of the triangle can be cut away to form a trapezoid for each fuzzy set. In the dosage example, the triangle for "Decrease Dosage" would be cut 10% of the way up from the bottom, the triangle for "Maintain Dosage" would be cut 42% of the way up from the bottom and the triangle for "Increase Dosage" would be cut 48% of the way up from the bottom. There are then several different ways to proceeed. One of the most common techniques is the *centroid* method, which takes the centre of the area under the graph derived from combining the trapezoids of each fuzzy set. This is also known as the *centroid* of the resulting shape and the method uses its x coordinate as the defuzzified value. Alternatively, the *bisector* method takes the vertical line that divides the area under the graph into two sub-regions of equal area. There are also the *Middle of Maximum* (MOM), *Smallest of Maximum* (SOM), or *Largest of Maximum* (LOM) methods, which take the maximum value assumed by the aggregate membership function shape and assign the defuzzified value as the median, minimum and maximum corresponding x value respectively.

2.1.8 Fuzzy Inference and Reasoning

Fuzzy reasoning or inference is the process of deriving conclusions based on a set of fuzzy rules and given facts in order to make decisions. It consists of the subprocesses of constructing the linguistic terms and their membership functions (*fuzzification stage*), creating fuzzy rules, i.e. fuzzy implications (*rule stage*), selecting suitable operators to combine the rules and extract the truth of given propositions (*inference stage*) and defuzzifying the result to produce a quantifiable result (*defuzzification stage*), see Section 2.1.7.

An example of a type of fuzzy inference is *generalized modus ponens*. In Boolean logic, the rule "If X is A then Y is B" together with the observation that "X is A", mean that "Y is B" holds true. In other words, given an if-then rule and a premise, the outcome can be determined. In fuzzy logic, a proposition "X is A'", close to the premise "X is A" can be observed to derive the conclusion that "Y is B'", which is close to the conclusion "Y is B". Generalized modus tollens may also be employed to derive valid conclusions from valid premises in fuzzy logic, and there are many other valid inferential procedures. For further information see Li [31], which provides an analysis of the various types of fuzzy reasoning and their suitability towards different applications and *Fuzzy Logic and Fuzzy Reasoning* by [32]. Fuzzy inference systems (FIS) are also known as fuzzy models, fuzzy rule-based systems and perhaps the more well known fuzzy controllers.

2.1.9 Fuzzy Inferencing Systems (FIS)

Algorithms that implement fuzzy reasoning are known as Fuzzy Inference Systems (FISs). They are also known as fuzzy models, fuzzy rule-based systems and perhaps the more well known fuzzy logic controllers. There are two main types; *Mamdani* controllers [4] and *Takagi, Sugeno and Kang*, universally known as TSK controllers [33], [34].

Both the Mamdani and TSK systems are characterised by IF-THEN rules and have the same antecedent structures. The difference is in the structure of their consequents. In the Mamdani method the output membership functions are fuzzy sets, which means that a defuzzification stage is required. With the Takagi, Sugeno & Kang(TSK) method the consequent of a rule is a function, and so defuzzification is not required. The TSK method represents the output as a constant or as linear functions of the input.

A typical rule in a TSK fuzzy logic system has the form:

If Input1 $= x$ and Input2 $= y$, then Output is $z = ax + by + c$.

For a zero-order model, the output level z is a constant ($a = b = 0$).

The output level z_i of each rule is weighted by the firing strength w_i of the rule, and the final output of the system is the weighted average of all the rule outputs:

$$\text{Final Output} = \frac{\sum_{i+1}^{n} w_i z_i}{\sum_{i+1}^{n} w_i}, \tag{2.31}$$

where n is the number of rules.

In an interval Type-2 TSK fuzzy logic system the final output is an interval type-1 fuzzy set. This can be defuzzified using the same process as discussed in Section 2.3.5.

2.2 Type-2 Fuzzy Logic Systems

As discussed, Lotfi Zadeh introduced fuzzy sets in 1965 [1]. The concept of fuzzy sets allowed vagueness, partial truths and partial membership sets to be expressed mathematically. Furthermore it allowed formalised tools to be generated which handled the impreciseness of many real world problems. These tools became known as (type-1) fuzzy logic systems. Type-1 fuzzy logic is able to model human reasoning as it is based upon the use of rules and levels of belief to describe the problem. From this many aspects of human reasoning can be modelled, incorporating vagueness and approximate information, to generate decisions based upon incomplete data. Being rule based type-1 fuzzy logic was able to encapsulate knowledge and successfully simplify and solve many decision problems. Although many successful applications have been made using type-1 fuzzy logic systems on rule based systems they have a major draw back. Type-1 fuzzy logic systems are not able to completely handle or address the linguistic and numerical uncertainties that are found in unstructured and dynamic environments. They are not able to model uncertainty. This is due to using fuzzy sets which are precise. The issue of handling uncertainty in fuzzy sets was addressed by Zadeh in 1975. Zadeh proposed 'fuzzy sets with fuzzy membership functions' as an extension to the concept of an ordinary, i.e. type-1, fuzzy set. The new sets were defined to be of type n, for n = 2, 3, ..., for which the fuzzy membership functions range over fuzzy sets of type n-1 [35]. With the introduction of general fuzzy sets, the original fuzzy set became known as a type-1 fuzzy set. The most important general fuzzy set to date is the type-2 fuzzy set and the corresponding type-2 fuzzy logic systems. Type-2 fuzzy sets and logic have a methodology that is mathematically rigorous to allow reasoning with terms that are uncertain.

Type-2 fuzzy sets provide a method for defining the membership grades of fuzzy sets

with linguistic terms. For example the membership grades of a fuzzy sets such as *warm* can be labelled with linguistic terms such as *slightly, fairly and quite*. Zadeh suggests that using linguistic terms in this way allows for better understanding and measurement of the fuzzy membership grades. Mendel [36, 37] however, demonstrates in his work on survey based sets that expertise can be applied to arbitrary scales within a specific context and set of terms. Each membership grade gives a probabilistic distribution of a given point in a domain of a set regardless of the label. So if the membership grade is a label with a context specific term such as *slightly*, or a scale based term such as *close to* 0.7, then it can be modelled using type-2 fuzzy sets. Thus the vagueness and uncertainty surrounding the knowledge modelled by the fuzzy set is captured by the fuzzy membership functions of the fuzzy set. This is the mechanism used by type-2 fuzzy sets to describe and capture any inherent uncertainty within a system. Of course if there is no uncertainty within the fuzzy sets then type-2 fuzzy sets revert to type-1 fuzzy sets [38].

Type-2 fuzzy systems can be used to overcome the issue of inter-expert vagueness. This is when a group of experts disagree on specific points, to a greater or lesser extent. The membership of the sets may be modelled with fuzzy numbers describing the degree of membership of the expert to the set. Type-2 fuzzy sets cannot however model intra-expert variation. This is when an expert alters their view or conclusion over time. This is addressed by non-stationary fuzzy system proposed by Garibaldi *et al* [39,40]. Essentially these are type-1 fuzzy systems where the membership functions are subject to variation over time and hence non-deterministic. The variation is system dependent comprising of normal distributions, random distributions or variation within a given time interval. This results in a non-stationary fuzzy system with the fine detail of the decision making process varying over the time period. Due to type-2 fuzzy systems being deterministic they cannot model intra-expert variation.

Mendel in his book [22] identified and listed the following four sources of uncertainty in a fuzzy logic system:

- Uncertainty about the meaning of words that are used in the rules
- Uncertainty about the consequent that is used in a rule
- Uncertainty about the measurements that activate the fuzzy logic system

- Uncertainty about the parameter tuning data used in the fuzzy logic system

The first two are due to the uncertainty of the semantics of the fuzzy sets and membership functions. The last two are concerned with measurement and noise from sensors when collecting real and training data. Mechanical vibration, electrical noise, flow turbulence all create uncertainty as to the correct value of the signal. A type-2 fuzzy logic system is capable of overcoming the uncertainty problem and provides an attractive low cost solution. Type-1 and type-2 fuzzy logic systems both allow small measurement changes to have a small or zero change on the output. This provides a smoother control response when compared to a crisp set system with a similar number of sets. Due to the fact that type-2 fuzzy systems use type-1 fuzzy sets to model membership values, theoretically type-2 controllers should handle noisy conditions better.

Further work has been carried out on type-2 fuzzy sets. The set theoretic operations of type-2 sets and properties of membership degrees of such sets were studied by Mizumoto and Tanaka [41]. They examined type-2 sets under the operations of algebraic product and algebraic sum [42]. Dubois and Prade gave a formula for the composition of type-2 relations as an extension of the type-1 sup-star composition for the minimum t-norm [43]. Due to the significant increase in the computational complexity and required computational power need to implement them, general type-2 fuzzy logic systems have been limited in numbers and applications.

Mendel established and introduced a set of terms when working with type-2 fuzzy sets. The best known is the concept of the *footprint of uncertainty* which provides a useful verbal and graphical description of the uncertainty captured by any given type-2 set. Mendel has particularly investigated a restricted class of general type-2 fuzzy sets called *interval type-2 fuzzy sets* [22]. Interval type-2 fuzzy sets differ from general type-2 fuzzy sets by having secondary membership functions which only take values of either 0 or 1. This massively reduces the computational calculation in the inference with type-2 sets. The derivation of union, intersection, and complement operations, and computational algorithms for type reduction was simplified by Mendel and John [44]. The development of interval type-2 fuzzy logic allowed the implementation of type-2 fuzzy logic systems to occur and advance the technology.

Simon Coupland in 2006 introduced the concept of geometrical fuzzy logic systems [45]. Fuzzy logic sets were defined as geometric objects or geometric fuzzy sets. The logical operations on these sets were then defined to be geometric manipulations of the geometric sets. This approach reduced the computational complexity of general type-2 fuzzy logic to a point, where for the first time, general type-2 fuzzy logic was successfully applied to a control problem - navigation of a mobile robot. The results showed that the general type-2 fuzzy logic controller out performed type-1 and interval type-2 controllers that it was compared with.

2.2.1 General Type-2 Fuzzy Sets

Type-2 fuzzy sets were initially defined by Zadeh in a series of three papers [46, 47, 48]. The papers defined the basic operations of type-2 fuzzy logic sets and were further expanded by Mizumoto and Tanaka [41, 42], Dubois and Prade [49, 43, 50], Karnik and Mendel [51, 52, 53] and Mendel and John [44]. A type-2 fuzzy set is characterized by a fuzzy membership function, i.e. a membership value or grade for each element of this set is a fuzzy set in [0,1], whereas the membership grade of type-1 fuzzy set is a crisp value in [0,1]. The definition of a type-2 fuzzy set, widely used across the literature, was given by Mendel and John.

Definition 2.2.1 A type-2 fuzzy set, denoted $\tilde{A}$, is characterised by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq U = [0,1]$, such that:

$$\tilde{A} = \{((x,u),\mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq U = [0,1]\} \tag{2.32}$$

where $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$, X *is the domain of the fuzzy set and* J_x *is the domain of the secondary function at x. An alternative expression for* $\tilde{A}$ *is given by:*

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq U = [0,1] \tag{2.33}$$

where $\int \int$ *denotes union over all admissible x and u. For discrete universes of discourse, use* $\sum$ *instead of* $\int$*. Taken from Mendel and John [44]*

Vertical Slice Definition

In the vertical slice definition, the membership grade at each point in a type-2 fuzzy set is seen as a type-1 fuzzy number bounded in the interval [0,1].

Definition 2.2.2 At each value of x, say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{A}}(x,u)$ is called a vertical slice of $\mu_{\tilde{A}}(x,u)$. A secondary membership function is a vertical slice of $\mu_{\tilde{A}}(x,u)$. It is $\mu_{\tilde{A}}(x = x',u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq U = [0,1]$, i.e.

$$\mu_{\tilde{A}}(x = x',u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u \quad J_x \subseteq U = [0,1] \tag{2.34}$$

Taken from [44]

Using Equation 2.34, we can also re-express $\tilde{A}$ as a vertical slice manner, i.e.

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x,u) | \forall x \in X\} \tag{2.35}$$

or,

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_{x'}} f_x(u)/u \right] /x \quad J_x \subseteq U = [0,1] \tag{2.36}$$

where $\int \int$ denotes union over all admissible x and u. Use $\sum$ instead of $\int$ for discrete universes of discourse, as:

$$\tilde{A} = \sum_{x \in X} \mu_{\tilde{A}}(x)/x = \sum_{x \in X} \left[\sum_{u \in J_x} f_x(u)/u \right] /x \quad J_x \subseteq U = [0,1] \tag{2.37}$$

Definition 2.2.3 The domain of a secondary membership function is called the primary membership grade of x. In Equation 2.37, J_x is the primary membership function of x. where $J_x \subseteq [0,1]$ for $\forall x \in x$.

Definition 2.2.4 The amplitude of the secondary membership function is called a secondary grade

A secondary grade of Equation 2.37 is $f_x(u)$, the amplitude of the secondary membership function.

The Representation Theorem

The membership function of a type-2 fuzzy set has be defined in two equivalent models, the Mendel and John model, and the Vertical Slice model. The Representation Theorem model introduced by [44] approaches the issue using embedded sets. The Representation Theorem states that 'A type-2 fuzzy set can be represented as the union of its type-2 embedded sets'. The Representation Theorem model does not use the extension principle to derive operations. Instead the operations can be derived directly from type-1 fuzzy sets and operations. The drawback to using the representation theorem model is that a large number of embedded type-2 sets are needed to model the type-2 fuzzy set. This makes the operations extremely inefficient due to the massive redundancy generated in the unions when only one is required. In the continuous domain the number of embedded sets is countably infinite and therefore not very useful.

Definition 2.2.5 For discrete universes of discourse X and U, *an embedded type-2 set* $\tilde{A}_e$ has N elements, where $\tilde{A}_e$ contains exactly one element from $J_{x1}, J_{x2}, ..., J_{xN}$, namely $u_1, u_2, ..., u_N$, each with its associated secondary grade, namely $f_{x1}(u_1), f_{x2}(u_2), ..., f_{xN}(u_N)$, i.e.,

$\tilde{A}_e = \sum_{i=1}^{N} [f_{x_i}(u_i)/x_i]/x_i u_i \in J_{x_i} \subseteq U = [0,1]$

Where $\tilde{A}_e^j$ is the j^{th} embedded set in $\tilde{A}$, M_i is the number of points in the domain of the i^{th} secondary membership function of $\tilde{A}$. There are a total of $\prod_{i=1}^{N} M_i \tilde{A}_e^j$ embedded $\tilde{A}_e^j$ sets in $\tilde{A}$.

Adapted from [44]

So $\tilde{A}$, a discrete type-2 fuzzy set, can be represented as the union of its type-2 embedded sets, i.e.

$$\tilde{A}_e = \sum_{j=1}^{n} \tilde{A}_e^j \tag{2.38}$$

where

$$n = \prod_{i=1}^{N} M_i \tag{2.39}$$

2.2.2 Operations on General Type-2 Fuzzy Sets

Using the vertical slice definition of a type-2 fuzzy set in Equation 2.36, let $\tilde{A}$ and $\tilde{B}$ be:

$$\tilde{A} = \int_{x\in X} \mu_{\tilde{A}}(x)/x = \int_{x\in X} \left[\int_{u\in J_{x'}} f_x(u)/u \right] /x \quad J_x \subseteq U = [0,1] \tag{2.40}$$

and

$$\tilde{B} = \int_{x\in X} \mu_{\tilde{B}}(x)/x = \int_{x\in X} \left[\int_{u\in J_{x'}} g_x(u)/u \right] /x \quad J_x \subseteq U = [0,1] \tag{2.41}$$

Then the intersection of the secondary membership functions of $\tilde{A}$ and $\tilde{B}$ is given by:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \int_{u\in J_x^u} \int_{v\in J_x^v} f_x(u) \bigstar g_x(v)/(u\wedge v) = \mu_{\tilde{A}}(x) \prod \mu_{\tilde{B}}(x), \quad x \in X \tag{2.42}$$

where $\wedge$ means minimum or product, and $\bigstar$ means minimum or product t-norm. $\prod$ denotes the meet operation [41]. In order to perform the meet operation between two secondary membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $m = u \wedge v$ has to be performed on every possible pairing of u and v, such that $u \in J_x^u$ and $v \in J_x^v$. The secondary grade of $\mu_{\tilde{A}\cap\tilde{B}}(x)$ must be calculated as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $f_x(u)$ and $g_x(v)$ respectively. To obtain $\mu_{\tilde{A}\cap\tilde{B}}(x)(x,m)$ this must be done for $\forall x \in X$. If two or more combinations of u and v give the same point $u \wedge v$ then the largest membership grade one is put into the meet.

The union of secondary membership functions of $\tilde{A}$ and $\tilde{B}$ is given by:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \int_{u\in J_x^u} \int_{v\in J_x^v} f_x(u) \bigstar g_x(v)/(u\vee v) = \mu_{\tilde{A}}(x) \coprod \mu_{\tilde{B}}(x), \quad x \in X \tag{2.43}$$

where $\vee$ means maximum, and $\bigstar$ means minimum or product t-norm. $\coprod$ denotes the join operation [41]. In order to perform the join operation between two secondary membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $j = u \wedge v$ has to be performed on every possible pairing of u and v, such that $u \in J_x^u$ and $v \in J_x^v$. The secondary grade of $\mu_{\tilde{A}\cup\tilde{B}}(x)$ must be calculated as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$,

$f_x(u)$ and $g_x(v)$ respectively. To obtain $\mu_{\tilde{A}\cap\tilde{B}}(x)(x,j)$ this must be done for $\forall x \in X$. If two or more combinations of u and v give the same point $u \vee v$ then the largest membership grade one is put into the join.

2.2.3 General Type-2 Fuzzy Logic Systems

General type-2 fuzzy logic systems can be constructed from type-1 fuzzy logic system. The commonality is in the rule base of both fuzzy systems. The difference between the two systems is in the way that the membership functions are defined and processed. There is no need to alter the rule base if the variables and the terms are not altered. The rule base is a set of IF THEN rules, based on expert knowledge which is represented by linguistic terms and for type-2 systems, implied uncertainty. For type-2 systems the rules have uncertain antecedent parts and/or consequent parts. These then translate into uncertain antecedent and/or consequent membership functions. The inference engine combines the rules and generates a mapping between the input and output type-2 fuzzy sets. It has to discover or infer the intersections and unions of the type-2 sets together with the compositions of the type-2 relationships. All this results in an output that is another type-2 fuzzy set. Nearly all applications that use type-2 fuzzy logic require a crisp value to be derived from the output set. To achieve this the output fuzzy set then has to be type reduced from a type-2 set into a set of type-1 fuzzy sets. The defuzzification process is then required to process each of these type-1 fuzzy sets according to the required defuzzification action. For example if the centroid of gravity is required then the centre of gravity method is applied to all the type-1 sets and the centroid calculated. The complete type-2 fuzzy logic theory including the handling of uncertainty, the operations on type-2 sets, aggregation, type reduction and defuzzification methods such as the centroid of gravity are given in [54, 44, 52, 55, 53, 38, 56, 57, 58].

The set of rules used in a fuzzy logic system have the same structure for either a type-1 or a type-2 implementation. They are distinguished by the type of the membership function sets used in the antecedent and consequent. A type-2 fuzzy rule R^i maps p inputs of the input space $X_1 \times X_2 \times ... \times X_p$ to the output space Y of the fuzzy logic system. i.e.

$$R^i = IF\ x_1\ is\ \tilde{F}_1^i\ and\ x_2\ is\ \tilde{F}_2^i\ and\ ...\ x_p\ is\ \tilde{F}_p^i\ THEN\ y\ is\ \tilde{G}^i \tag{2.44}$$

From Karnik and Mendel [38]

where $x_1 ... x_p$ are the crisp inputs, and y is the crisp output of the system. $\tilde{F}_1^i to \tilde{F}_p^i$ are the antecedent and $\tilde{G}^i$ the consequent type-2 fuzzy sets respectively of the rule R^i. The operator 'and' can interchanged with the operator 'or', also 'not' can be used as well. The 'and' operation is implemented using the *meet* ($\sqcap$) operation and the 'or' operation is implemented by the *join* ($\sqcup$) operation. The implication operation is normally the meet operation although other operations can be used. A scaling operation was suggested by Karnik and Mendel [38] as an alternative. Prior to discussing the operations used in inference, the extension principle is given. This provides the method to define these operations.

In the same set of papers that Zadeh [35] introduced type-2 fuzzy logic, he also presented the extension principle. This allowed all the logical operations to be defined for type-2 fuzzy sets by extending the type-1 operations. As previously seen the representation theorem is now available to do this from type-1 fuzzy operations as well.

Definition 2.2.6 The extension principle for fuzzy sets is a basic identity which permits the the domain of the definition of a mapping or a relation to be extended from point in U to fuzzy subsets of U. Let f be a mapping from U to V, and A is a fuzzy subset of U such that

$$A = \mu_1 \mu_1 + ... + \mu_n \mu_n \tag{2.45}$$

Then the extension principle asserts that

$$f(A) = f(\mu_1 \mu_1 + ... + \mu_n \mu_n) \equiv (\mu_1 f(\mu_1) + ... + \mu_n f(\mu_n)) \tag{2.46}$$

Thus, knowing the images of $\mu_1, ..., \mu_n$ under f, the image of A under f can be deduced.

The extension principle allows any function on a know set to be extended to the next set. So a function on a type-0 set (crisp set) can be extended to a type-1 set, and that function having been defined on a type-1 set then can be extended to a type-2 set. In this way Zadeh, extended the meet and join operations on type-1 sets to type-2 sets. The stages of inferencing for a type-2 fuzzy logic system are now discussed.

The process of identifying the membership grade of the input on a fuzzy set is known as fuzzification. For discrete systems the process simply takes the crisp input value and returns a secondary membership function. If a continuous input has to be fuzzified then function application or linear interpolation is used to calculate the membership grade value. The membership grade for every type-2 fuzzy set in the antecedents of every rule in the rule base must be found. Then all these secondary membership functions have to be logically connected to produce the rule antecedents values. The logical connections can be in the rules or be defined globally for the fuzzy logic system.

The 'or' and 'and' logical connectors for type-2 fuzzy sets were provide by Zadeh [35]. They were then renamed to the 'join' and 'meet' operators respectively by Mizumoto and Tanaka [41], who were the first to investigate the properties of these operators. Along with Dubois and Prade [43], Mizumoto and Tanaka [42] discussed the use of t-norm and t-conorm (s-norm) operators. By limiting the secondary membership functions to normal - at least one point has a value of one, and convex - only a single peak value, Karnik and Mendel [52] defined more computationally efficient methods to calculate the join and meet for them.

The join ($\sqcup$) operation operates on two secondary membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ and generates their conjunction. Let $\mu_{\tilde{A}}(x) = \sum_{i=1}^{M} \alpha_i / v_i$ and let $\mu_{\tilde{B}}(x) = \sum_{j=1}^{N} \beta_j / w_j$. The conjunction of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ is given by

$$\mu_{\tilde{A} \sqcup \tilde{B}}(x) = \sum_{i=1}^{M} \sum_{j=1}^{N} (\alpha_i \star \beta_j) / (v_i \vee w_j) \tag{2.47}$$

where $\vee$ is the t-conorm, which is generally taken to be the maximum and $\star$ which is a t-norm usually either a product or a minimum.

The meet ($\sqcap$) operation operates on two secondary membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ and generates their disjunction. The disjunction of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ is given by

$$\mu_{\tilde{A} \sqcap \tilde{B}}(x) = \sum_{i=1}^{M} \sum_{j=1}^{N} (\alpha_i \star \beta_j) / (v_i \star w_j) \tag{2.48}$$

taken from [42] where $\star$ is a t-norm.

The operators join and meet form the basic operations that are extensively used in

the inference process of type-2 fuzzy logic systems. They are used in conjunction with each other to logically combine the antecedent values used to calculate the value of a rule consequent during a type-2 implication operation. Having obtained these antecedent values they are used in the calculation of the rule consequent in the implication operation of the type-2 fuzzy logic system.

The implication operation requires that each antecedent of the rule consequent for each rule is calculated. The *meet* of the antecedent with every point in the consequent type-2 fuzzy set is found. Expressing the type-2 fuzzy relation R^i in equation 2.44 in terms of membership functions gives:

$$\mu_{R^i}(\mathbf{x},y) = \mu_{\tilde{F}_1^i \times \tilde{F}_2^i \times \cdots \tilde{F}_p^i \to \tilde{G}^i(\mathbf{x},y)} \tag{2.49}$$

$$\mu_{R^i}(\mathbf{x},y) = \mu_{\tilde{F}_1^i}(x_1) \prod \cdots \prod \mu_{\tilde{F}_p^i}(x_p) \prod \mu_{\tilde{G}^i}(y) \tag{2.50}$$

$$\mu_{R^i}(\mathbf{x},y) = [\prod_{k=1}^{p} \mu_{\tilde{F}_k^i}(x_k)] \prod \mu_{\tilde{G}^i}(y) \tag{2.51}$$

where $\prod$ denotes the *meet* operation.

The implication of a rule generates a consequent type-2 fuzzy set. As an illustration, let the antecedent value be $\mu_{\tilde{A}}(x_1) \prod \mu_{\tilde{B}}(x_2)$ and the consequent be $\tilde{G}$ over the domain Y. Then the value of $\mu_{\tilde{A}}(x_1) \prod \mu_{\tilde{B}}(x_2) \to \tilde{G}$ is

$$\mu_{\tilde{A}}(x_1) \prod \mu_{\tilde{B}}(x_2) \to \tilde{G} = \int_{y \in Y} \big(\mu_{\tilde{A}}(x_1) \prod \mu_{\tilde{B}}(x_2)\big) \prod \mu_{\tilde{G}}(y) \tag{2.52}$$

The meet operation is normally the minimum or product t-norms. Having given the type-2 fuzzy implication operation under the meet operator all the consequent sets have to be combined.

2.2.4 Consequent Sets Combination

The combination of the consequent set is achieved by using the 'or' operation i.e. the join operation. The join operation is applied to every point within the domains of the

consequent sets. Let $\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_p$ be the consequent sets and $\tilde{F}$ be the final combined set, all over domain X, then the value of $\tilde{F}$ is given by:

$$\tilde{F} = \int_{x \in X} \coprod_{i=1}^{p} \tilde{G}_i(x) \tag{2.53}$$

The result of the combination operation is a single type-2 fuzzy set. This is the decision of the fuzzy logic system to the input variables. From this set is derived the crisp output which is acted on by the external system.

Output Operations - Type Reduction

In order to have a practical application nearly all fuzzy logic systems have to produce a crisp output in response to crisp inputs. This means that the single combined type-2 fuzzy set has to be processed. In type-2 fuzzy logic systems this is usually achieved by the type reducer and the defuzzifier. Type reduction was introduced by Karnik and Mendel [38,53] and is an extension of type-1 defuzzification using the extension principle. Many of the type-1 defuzzification methods compute the centroid of a type-1 fuzzy set, so the generalised centroid of a type-2 fuzzy set needs to be calculated. The generalised centroid represents a mapping of the type-2 fuzzy set onto a type-1 fuzzy set. It finds every possible embedded type-2 fuzzy set within the initial type-2 fuzzy set. The centroid of each possible embedded set is then paired with the uncertainty of that set. By aggregating the pairings a type-1 set is generated which provides a distribution of the possible centroids of the initial type-2 fuzzy set and their associated uncertainty. The type-1 fuzzy set is then defuzzified to generate a crisp value. The type reduced set can be thought of as a measure of the variation in the crisp output, and a confidence interval of the linguistic variables in the fuzzy logic system. There are many methods that can be used for type reduction, three of the most common are centroid type reduction, height type reduction and centre of sums type reduction. In an interval type-2 fuzzy logic system the final output is an interval type-1 fuzzy set, and is defuzzified as discussed in Section 2.3.5.

Centroids of Fuzzy Sets

The centroids of type-1 and type-2 fuzzy set are derived as follows:

For a type-1 fuzzy set A with domain $x \in X$ discretised into N points, the centroid C_A is given by:

$$C_A = \frac{\sum_{i=1}^{N} x_i u_A(x_i)}{\sum_{i=1}^{N} u_A(x_i)} \tag{2.54}$$

For a type-2 fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x,u) | \forall x \in X\}$ with the x domain discretised into N points, as $x_1, x_2, \ldots, x_N$ such that

$$\tilde{A} = \sum_{i=1}^{N} \left[\int_{u \in J_{x_i}} f_{x_i}(u)/u \right] / x_i \tag{2.55}$$

the centroid $C_{\tilde{A}}$, using the extension principle, is given by:

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \ldots \int_{\theta_N \in J_{x_N}} [f_{x_1}(\theta_1) \star \ldots \star f_{x_N}(\theta_N)] / \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} x_i} \tag{2.56}$$

Where $C_{\tilde{A}}$ is a type-1 fuzzy set. Using the definition of an embedded type-2 set, every combination of $\theta_1, \cdots, \theta_n$ and its associated secondary grade $f_{x_1}(\theta_1) \star \cdots \star f_{x_1}(\theta_n)$ forms an embedded type-2 set $\tilde{A}_e$ in Equation 2.55. Each element of $C_{\tilde{A}}$ is generated by calculating the centroid of the embedded type-1 set A_e, associated with $\tilde{A}_e$ using:

$$\frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i}^{N} \theta_i} \tag{2.57}$$

and calculating the t-norm of the associated secondary grades of $\theta_1, \ldots, \theta_n$, $f_{x_1}(\theta_1) \star \ldots \star f_{x_1}(\theta_n)$.

To obtain the complete centroid $C_{\tilde{A}}$, this has to be done for all the embedded type-2 sets in $\tilde{A}$.

Letting $\theta = [\theta_1, \ldots, \theta_N]^T$,

$$a(\theta) = \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \tag{2.58}$$

and

$$b(\theta) = f_{x_1}(\theta_1) \star \ldots \star f_{x_1}(\theta_n) \tag{2.59}$$

then the complete centroid $C_{\tilde{A}}$ can be expressed as

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \ldots \int_{\theta_N \in J_{x_N}} b(\theta)/a(\theta) \tag{2.60}$$

There is a possibility that two or more combinations of the vector θ generate the same point in $a(\theta)$. If this occurs then the point with the largest value of $b(\theta)$ is kept in the centroid set. From Equation 2.60 the domain of $C_{\tilde{A}}$ is an interval $[a_l(\theta), a_u(\theta)]$ where

$$a_l(\theta) = min_{\theta} a(\theta) labeleq : al \tag{2.61}$$

and

$$a_u(\theta) = max_{\theta} a(\theta) labeleq : au \tag{2.62}$$

A suggested work flow for the calculation of the type-2 centroid $C_{\tilde{A}}$ is as follows:

1. Discretise the x-domain into N points $x_1, \cdots, x_N$.
2. Discretise the primary memberships of x_j, each J_{x_j}, into a manageable number of points, M_j, where $j = 1, \cdots, N$.
3. Enumerate all the embedded type-1 sets; there will be $\prod_{j=1}^{N} M_j$ of them needed to compute $C_{\tilde{A}}$. $\prod_{j=1}^{N} M_j$ can be very large, so the values of M_j or N need to be considered.
4. Compute $C_{\tilde{A}}$ using Equation 2.60. This involves the α tuples (a_k, b_k), for $k = 1, 2, \ldots, \prod_{j=1}^{N} M_j$ where a_k and b_k are given in Equations 2.58 and 2.59. $\alpha = \prod_{j=1}^{N} M_j$ in this case.

Due to a problem with the product form of the t-norm when calculating the centroid in a continuous domain, it must not be used. This is explained in Karnik and Mendel [38,53]. They recommend that the minimum t-norm is used in the calculation of the centroid for a type-2 fuzzy set with a continuous domain.

2.2.5 The Generalised Centroid Calculation

The centroid of a type-1 fuzzy set given in Equation 2.54 can be considered as a weighted average. This can be expressed in a general form where $z_i \in \Re$ (real numbers), and $v_i \in [0,1]$ for $i = 1, \cdots, N$.

$$y(z_1, \cdots, z_N\, v_1, \cdots, v_N) = \frac{\sum_{i=1}^{N} z_i v_i}{\sum_{i}^{N} v_i} \tag{2.63}$$

The extension of Equation 2.54 to Equation 2.55 only considers v_i to be a type-1 set. The generalisation of Equation 2.63 considers the case where z_i is also a type-1 set. Replace each z_i by the type-1 fuzzy set $Z_i \subset \Re$ with associated membership function $\mu_{Z_i}(z_i)$, and, each v_i by the type-1 fuzzy set $V_i \subseteq [0,1]$ with associated membership function $\mu_{V_i}(v_i)$. Then the extension of Equation 2.63, the generalised centroid $GC_{\tilde{A}}$ is given by:

$$GC_{\tilde{A}} = \int_{z_1 \in Z_1} \cdots \int_{z_N \in Z_N} \int_{v_1 \in V_1} \cdots \int_{v_N \in V_N} \left[T_{i=1}^{N} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{N} \mu_{V_i}(v_i) \right] / \frac{\sum_{i=1}^{N} z_i v_i}{\sum_{i}^{N} v_i} \tag{2.64}$$

where T and $\bigstar$ are the t-norm operators used i.e. minimum or product. It is seen that $GC_{\tilde{A}}$ is a type-1 fuzzy set.

Letting $\Theta = [z_1, \ldots, z_N, v_1, \ldots, v_N]^T$ and re-expressing $a(\theta)$ and $b(\theta)$ in Equations 2.58 and 2.59 as

$$\alpha(\theta) = \frac{\sum_{i=1}^{N} z_i v_i}{\sum_{i=1}^{N} v_i} \tag{2.65}$$

and

$$\beta(\theta) = \left[T_{i=1}^{N} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{N} \mu_{V_i}(v_i) \right] \tag{2.66}$$

then the generalised centroid $GC_{\tilde{A}}$ can be expressed as

$$GC_{\tilde{A}} = \int_{z_1 \in Z_1} \cdots \int_{z_N \in Z_N} \int_{v_1 \in V_1} \cdots \int_{v_N \in V_N} \beta(\theta)/\alpha(\theta) \tag{2.67}$$

The requirement for the generalised centroid is due extending a centre of sets type-1

defuzzification to the type reduction of a centre of sets. This type reduction requires both z_i and v_i to be type-1 sets. There is a huge increase in the computational effort required to calculate the generalised centroid. Using the centroid derivation scheme, both z_i and v_i have to be discretised to a reasonable number of points, M_i and N_i. Then the total number of point computations will be $\prod_{i=1}^{N} M_i N_i$.

2.2.6 General Type-2 Centroid Type-Reduction

For type-1 defuzzification, the output type-1 fuzzy sets are combined using the maximum or union operator and then the centroid of this set is found. So for the rule-output type-2 sets, $\tilde{B}^l$, the centroid type-reducer uses the union operator to combine them. The union operation requires that the join of their secondary membership functions is calculated, see Equation 2.47. So $\tilde{B} \equiv \bigcup_{l=1}^{M} \tilde{B}^l$, has a secondary membership function $\mu_{\tilde{B}(y)}$ given by

$$\mu_{\tilde{B}}(y) = \coprod_{l=1}^{M} \mu_{\tilde{B}^l}(y), \quad \forall y \in Y \tag{2.68}$$

where $\mu_{\tilde{B}^l}(y)$ is the secondary membership function for the l^{th} rule. The characteristics of $\tilde{B}(y)$ is governed by factors such as meet and join operations performed on the embedded sets that make up the secondary membership functions.

The centroid of the type-2 output set $\tilde{B}$, is calculated by the centroid type reducer. The expression for the centroid type-reduced set $Y_c(\mathbf{x})$, extends the following type-1 centroid defuzzifier

$$y_c(\mathbf{x}) = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)} \tag{2.69}$$

The extension can be expressed as:

$$Y_c(\mathbf{x}) = \int_{\theta_i \in J_{y_1}} \cdots \int_{\theta_N \in J_{y_N}} [f_{y_1}(\theta_1) \bigstar \ldots \bigstar f_{y_N}(\theta_N)] \Big/ \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} \tag{2.70}$$

where $i = 1, \cdots, N$. For different fuzzy logic system inputs, different values of $y_c($**x** are obtained. A similar sequence to that used to calculate $C_{\tilde{A}}$, the centroid of a type-2 set is given.

1. Compute $\mu_{\tilde{B}()y}$ as in Equation 2.68
2. Discretise the y-domain into N points $y_1, \cdots, y_N$.
3. Discretise the primary memberships of y_i, each J_{y_i}, into a manageable number of points, M_i, where $i = 1, \cdots, N$.
4. Enumerate all the embedded type-1 sets in $\tilde{B}$; The total number of enumerations will be $\prod_{i=1}^{N} M_i$.
5. Compute Y_c, the centroid type-reduced set using Equation 2.70 As noted previously the minimum t-norm must be used in the calculation.

In all the membership and centroid calculations have to be repeated $\prod_{i=1}^{N} M_i$ times since all the points for all the embedded sets have to be included in the centroid. For practical purposes this is a prohibitive number of calculations. Fortunately using interval sets the number of calculations is considerably reduced.

2.3 Interval Type-2 Fuzzy Logic

The interval type-2 fuzzy logic system were introduced by Zadeh [47] and relied on the concept of an α cut. In the definition of a type-2 fuzzy set given in Definition 2.2.1, if $\mu_{\tilde{A}}(x,u) = 1, \forall u \in J_x \subseteq U = [0,1]$, then the secondary membership functions are interval sets. Then, if this is true for $\forall x \in X$, they are *interval type-2 membership functions*. Interval secondary membership functions have the property that the membership grade at each point in the domain of x is uniform and is understood to be unity [22]. So the definition of an interval type-2 fuzzy logic set is given by:

Definition 2.3.1 An interval type-2 fuzzy set, denoted $\tilde{A}$, is characterised by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq U = [0,1]$, such that:

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq U = [0,1]\} \tag{2.71}$$

where $\mu_{\tilde{A}}(x,u) \in \{0,1\}$, X is the domain of the fuzzy set and J_x is the domain of the secondary function at x. *An alternative expression for* $\tilde{A}$ *is given by:*

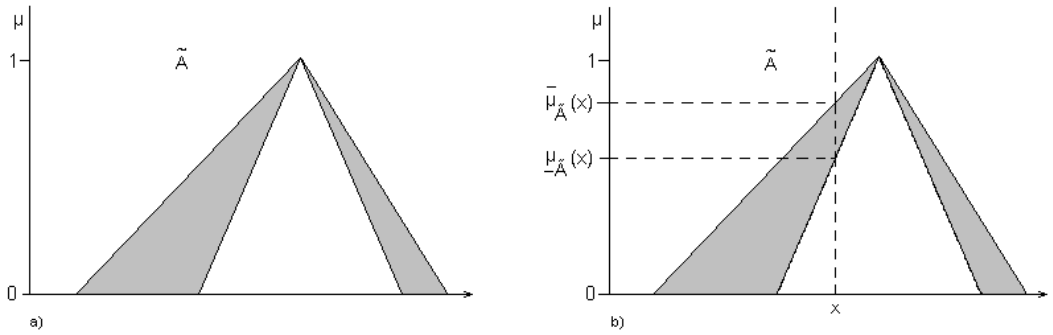


Figure 2.1: a) The Type-2 Interval Fuzzy Set $\tilde{A}$. b) The Membership Grade of a Point x in $\tilde{A}$.

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x} 1/u \right] /x \quad J_x \subseteq U = [0,1] \tag{2.72}$$

Figure 2.1a) shows an interval type-2 fuzzy set. Due to all the points in each of the secondary membership functions being at unity, every secondary membership can be represented as an interval set. An interval set $\tilde{A}$ contains two elements, a lower bound which is denoted $\underline{\tilde{A}}$ and an upper bound denoted $\overline{\tilde{A}}$, such that $\tilde{A} = [\underline{\tilde{A}}, \overline{\tilde{A}}]$. All points between these bounds are implicit elements of the interval set. So an interval set can be represented by its domain interval, namely the left and right end points $[l,r]$, or by its centre and spread as $[c-s, c+s]$ where

$$c = (l+r)/2 \; and \; s = (l-r)/2 \tag{2.73}$$

2.3.1 The Footprint of Uncertainty

The concept of 'footprint of uncertainty' (FOU) was invented by Mendel. The FOU is an extremely important term as it provides a convenient and succinct verbal description of the entire domain of support for all the secondary grades of a type-2 membership function. It was first published in Karnik *et al* in 1999 [51] and appeared in Karnik and Mendel [52]. The FOU together with the associated type-1 upper and lower bound membership functions, introduced by Liang and Mendel [59], allow the characterisation

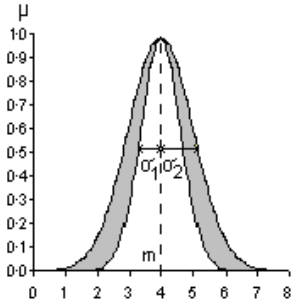


Figure 2.2: Footprint of Uncertainty for Gaussian primary membership function with uncertain standard deviation.

of type-2 fuzzy sets.

Definition 2.3.2 Uncertainty in the primary memberships of a type-2 fuzzy set, $\tilde{A}$, consists of a bounded region that is termed the footprint of uncertainty (FOU) and is the union of all primary memberships.

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \tag{2.74}$$

where J_x is the domain of the secondary membership function at x.

Taken from [22]

An example of a FOU is the shaded region in Figure 2.2. The FOU is shaded uniformly to indicate that it is for an interval type-2 fuzzy set; thus, a uniformly shaded FOU also represents the entire interval type-2 fuzzy set $\mu_{\tilde{A}}(x,u)$. Figure 2.2 represents the FOU for Gaussian primary membership function with uncertain standard deviation.

Mendel commented that the term FOU provides a 'very convenient verbal description of the entire domain of support for all the secondary grades of a type-2 membership function' [22].

2.3.2 Interval Type-2 Lower and Upper Membership Functions

As the FOU represents a complete interval type-2 set, then the FOU can be extended by representing the region covered with two type-1 membership functions, a lower bound and an upper bound [59, 51]. Let the lower bound membership function be $\underline{\mu}_{\tilde{A}}(x)$ and the

upper bound membership function be $\overline{\mu}_{\tilde{A}}(x)$, then Equation 2.72 is given as:

$$\tilde{A} = \int_{x \in X} \left[\int_{u \in \left[\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x) \right]} 1/u \right] /x \tag{2.75}$$

The lower membership function is a subset that contains the minimum membership grade of the FOU.

$$\underline{\mu}_{\tilde{A}}(x) = \int_{X} \left[\underline{\mu}_{\tilde{A}}(x) \star \underline{\mu}_{\tilde{A}}(x) \right] /x \tag{2.76}$$

Where $\star$ is the minimum t-norm.

Likewise the upper membership function is a subset that contains the maximum membership grade of the FOU.

$$\overline{\mu}_{\tilde{A}}(x) = \int_{X} \left[\overline{\mu}_{\tilde{A}}(x) \star \overline{\mu}_{\tilde{A}}(x) \right] /x \tag{2.77}$$

Where $\star$ is the maximum t-conorm.

Figure 2.1b) shows the upper and lower membership grades of a set $\tilde{A}$ and a point x.

Generally an interval type-2 membership function can be expressed in terms of its lower and upper bound type-1 membership functions given as:

For the membership function $\mu_{\tilde{L}_k^r}(x_k)$

$$\mu_{\tilde{L}_k^r}(x_k) = \int_{w^r \in [\underline{\mu}_{\tilde{L}_k^r}(x_k), \overline{\mu}_{L_k^r}(x_k)]} 1/w^r \tag{2.78}$$

Where $k = 1, \ldots, p$ (the number of antecedents) and $r = 1, \ldots, M$ (the number of rules).

Also the lower and upper membership functions can be expressed as:

$$\underline{\mu}_{\tilde{L}_k^r}(x_k) = \int_{X_k} \left[\underline{\mu}_{\tilde{X}_k}(x_k) \star \underline{\mu}_{\tilde{F}_k}(x_k) \right] /x_k \tag{2.79}$$

$$\overline{\mu}_{\tilde{L}_k^r}(x_k) = \int_{X_k} \left[\overline{\mu}_{\tilde{X}_k}(x_k) \star \overline{\mu}_{\tilde{F}_k}(x_k) \right] /x_k \tag{2.80}$$

Another example of a FOU is given in Figure 2.3a). The primary memberships J_{x_1} and

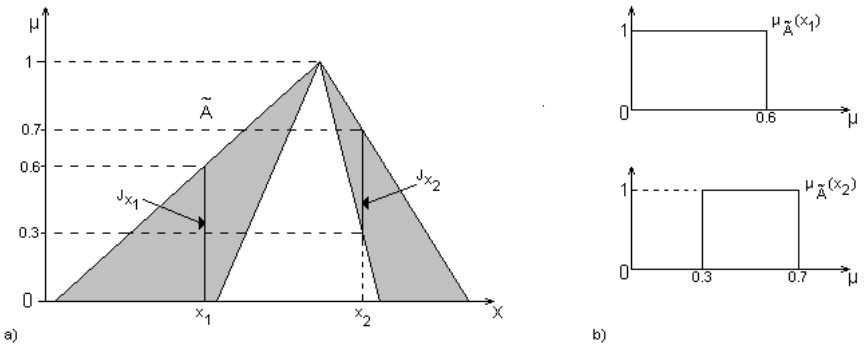


Figure 2.3: a) The FOU for a Type-2 Fuzzy Set $\tilde{A}$. b) Secondary Membership Functions for the points x_1 and x_2 in $\tilde{A}$.

J_{x_2} and their associated secondary membership functions $\mu_{\tilde{A}}(x_1)$ and $\mu_{\tilde{A}}(x_2)$ are shown at the points x_1 and x_2. The secondary membership functions are interval sets shown in Figure 2.3b).

2.3.3 Meet and Join of Interval Type-2 Fuzzy Sets

Lower and upper bound membership functions $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ respectively, can also be used to express the logical operations for inferencing. For two interval type-2 fuzzy sets $\tilde{A}$ and $\tilde{B}$, the meet operation is given by:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \int_{u\in[\underline{\mu}_{\tilde{A}}(x)\star\underline{\mu}_{\tilde{B}}(x),\overline{\mu}_{\tilde{A}}(x)\star\overline{\mu}_{\tilde{B}}(x)]} 1/u \tag{2.81}$$

From [60] Where $\star$ is the minimum or product t-norm.

Similarly the join is given by:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \int_{u\in[\underline{\mu}_{\tilde{A}}(x)\vee\underline{\mu}_{\tilde{B}}(x),\overline{\mu}_{\tilde{A}}(x)\vee\overline{\mu}_{\tilde{B}}(x)]} 1/u \tag{2.82}$$

From [60] Where $\vee$ is generally taken to be the maximum t-conorm.

Widening the argument to a general case using the general type-2 fuzzy sets, let $\tilde{A}$ and $\tilde{B}$ be two interval sets $F = \int_{u\in F} 1/u$ and $G = \int_{w\in G} 1/w$, respectively, with domains $u \in [l_f, r_f] \subseteq [0,1]$, and $w \in [l_g, r_g] \subseteq [0,1]$.

The meet operation between F and G is given by $Q = F \prod G = \int_{q\in Q} 1/q$. Re-expressing

equation 2.42 for an interval type-2 fuzzy set gives:

$$Q = F \prod G = \int_{q \in \left[l_f \star l_g, rf \star r_g\right]} 1/q \tag{2.83}$$

where $\star$ denotes a t-norm.

The *join* operation between F and G is given by $Q = F \coprod G = \int_{q \in Q} 1/q$. Re-stating equation 2.43for an interval type-2 fuzzy set gives:

$$Q = F \coprod G = \int_{q \in \left[l_f \star l_g, rf \star r_g\right]} 1/q \tag{2.84}$$

The *meet* and *join* operations given in equations 2.83 and 2.84, are determined just by the two end-points of each interval set in the operation, i.e. $[l_f, r_f]$ and $[l_g, r_g]$.

2.3.4 The Interval Type-2 Inference Process

For an interval type-2 fuzzy logic system, the meet operation becomes the t-norm operation between two lower and upper membership functions. It was shown by Liang and Mendel [60] that for an interval type-2 non-singleton type-2 FLS, with a minimum or product t-norm meet operation, the result of antecedent and input operations in the firing set $F^i(X')$,is an interval type-1 set. i.e.

$$F^i(X') = \left[\underline{f}^i(X'), \overline{f}^i(X')\right] = \left[\underline{f}^i, \overline{f}^i\right] \tag{2.85}$$

where

$$\underline{f}^i(X') = sup_X \int_{x_1 \in X_1} \cdots \int_{X_n} \left[\underline{\mu}_{\tilde{X}_1}(x_1) \star \underline{\mu}_{\tilde{F}_1^i}(x_1)\right] \star \cdots \star \left[\underline{\mu}_{\tilde{X}_n}(x_n) \star \underline{\mu}_{\tilde{F}_n^i}(x_n)\right] / x \tag{2.86}$$

and

$$\overline{f}^i(X') = sup_X \int_{x_1 \in X_1} \cdots \int_{X_n} \left[\overline{\mu}_{\tilde{X}_1}(x_1) \star \overline{\mu}_{\tilde{F}_i^i}(x_1)\right] \star \cdots \star \left[\overline{\mu}_{\tilde{X}_n}(x_n) \star \overline{\mu}_{\tilde{F}_n^i}(x_n)\right] / x \tag{2.87}$$

Secondly they showed that the output consequent set, $\mu_{\tilde{B}^i}(y)$ of the interval type-2 FLS

fired rules R^i is a type-1 fuzzy set, given by

$$\mu_{\tilde{B}^i}(y) = \int_{b^i \in \left[\underline{f}^i \star \underline{\mu}_{\tilde{G}^i}(y), \overline{f}^i \star \overline{\mu}_{\tilde{G}^i}(y)\right]} \tag{2.88}$$

where $\underline{\mu}_{\tilde{G}^i}(y)$ and $\overline{\mu}_{\tilde{G}^i}(y)$ are the lower and upper membership grades of $\mu_{\tilde{G}^i}(y)$, the consequent set in Equation 2.52.

Final they showed that if N of the M rules are fired in the FLS, the *join* of these type-1 output sets is given by applying Equation 2.53 such that

$$\mu_{\tilde{B}^i}(y) = \int_{b \in \left[\left[\underline{f}^i \star \underline{\mu}_{\tilde{G}^i}(y)\right] \vee \cdots \vee \left[\underline{f}^N \star \underline{\mu}_{\tilde{G}^N}(y)\right], \left[\overline{f}^i \star \overline{\mu}_{\tilde{G}^i}(y)\right] \vee \cdots \vee \left[\overline{f}^N \star \overline{\mu}_{\tilde{G}^N}(y)\right]\right]} 1/b, y \in Y. \tag{2.89}$$

2.3.5 Interval Type-2 Type Reduction and Defuzzification

For the type-2 centroid in Equation 2.56, an interval type-2 fuzzy set reduces it to

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} 1 \Big/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} x_i} = [c_l, c_r] \tag{2.90}$$

Letting $J_{x_i} \equiv [L_i, R_i]$, then J_{x_N} is the secondary membership grade at N in the secondary membership function J_x. Type reducing $C_{\tilde{A}}$ generates an interval set C, which is the distribution of all the possible centroids of $C_{\tilde{A}}$. The end points of the set $[C_l, C_r]$ provide the range of the points where the centroid of C, subject to system uncertainty, could lie. The final defuzzified value of the set C is given by its mid point. So for the centroid type-reduction of an interval type-2 fuzzy logic system the defuzzification method is to find the lower and upper membership functions of the centroid membership function. The lower membership function, L_i, is the subset that contains the minimum membership grade and the upper membership function R_i, is the subset that contains the maximum membership grade. It is then a simple matter of calculating the the final defuzzified value as $y = (y_l + y_r)/2$, the average of the two defuzzified lower and upper type-1 membership sets. Since the output of an interval type-2 TSK FLS is an interval type-1 fuzzy set then the crisp TSK output is calculated in the same way.

2.4 Architectures of Robotic Systems

It is sometimes said that the artificial intelligence (AI) world is split into two camps. The first camp were concerned with AI challenging humans in competitive games such as chess, draughts, cards games and trying to pass the Turing Test. This aspect is not discussed in this literature review. The second camp were interested in the robotic side of AI, the ability to move and reason. This section discusses the major architectures that have been used in robotics. Examples of each are given together with a measure of strengths and weaknesses.

2.4.1 Subsumption Architecture

Subsumption Architecture was first proposed by Rodney Brooks in the seminal paper 'A Robust Layered Control System for a Mobile Robot' [61]. Before this paper the problem of designing and building control systems was to functionally decompose them into a series of vertical slices. Brooks' idea was to describe the functionality in terms of 'levels of competence' of a desired class of behaviours. In this paper, Brooks defined what attributes a behavioural robot should have. He described a vertical decomposition based on task achieving behaviours and provided a description for how to build the controller. He described the physical robot to which he was applying the controller. However, at the time of writing the paper, the testing had only been carried out on a simulated robot.

Definitions

Architecture The process of constructing robotic control systems from common software modules to allow them to perform situated specific tasks. See Hayes-Roth 1995 pg. 330 [62].

Subsume Consider (an instance of something) as part of a general rule or principle, contain or include (Oxford University Press).

To take up into or under, as individual under species, species under genus, to include under something else (Webster).

Features

- Multiple goals - Each layer is able to work on its goals concurrently. The results can be used for the final decision on what to do.
- Multiple sensors - No need to centrally process sensor information. Preception processing determines what information is to be centrally represented. However each level may process them in a way that achieves its own objectives.
- Robustness - having multiple sensors increases the robustness of the system. Also the subsumptive architecture allows higher levels to suppress the lower levels only when their output is timely and valid. If the higher levels fail then the robot still functions, but at a lower level of competence.
- Additivity - Each level has its own processor, which are loosely coupled. Bandwidth requirements are low on comms channels between the layers. So it is easy to engineer.

Assumptions

Brooks proposed the following dogmatic principles

- Complex and useful behaviour need not necessarily be the product on an extremely complex control system.
- Things should be simple.
- No complexity in interfaces.
- Solving ill conditioned problems is bad and to be avoided, from the view point of robustness.
- Map making is of crucial importance. Build cheap robots.
- Must have a 3D capability.
- Must have relational maps - alters design space for perceptive systems.
- Build no artificial environments, must be real.

- Visual data for intelligent interaction. Sonar maybe useful for obstacle avoidance (low level interactions).
- Must have built in self-calibration for robustness.
- Must be self-sustaining for long-life without human assistance in a complex environment.

A Vertical Decomposition based on Task Achieving Behaviours

The architecture is rule based and has a priority-based arbitrator. Coordination is achieved by Suppression which overrides lower signals, and Inhibition which prevents signals from reaching motor actuators. The principle design criteria for subsumptive robots is situatedness and embodiment. Situatedness is the ability to sense its surroundings, with no representations. Embodiment means that the robots experience the real world directly, not through a simulation. The control system was not demonstrated on a real robot but on a simulator, which goes against what Brooks was advocating. Interestingly the level of competence obtained was 'Wander' with some work done on 'Explore', but Brooks expected that over time others would be built and implemented.

Each layer is a Finite State Machine (FSM)- Referring back to Turing Machines. The paper talks about algorithms, these by Turing's definitions do not allow input or output until halted. This is further discussed by Brooks in 'Intelligence without Reason' [63]. Brooks defined a Robot Control Specification Language for the FSM. This consisted of four states, output, side effect, conditional dispatch, event dispatch plus the distinguished state of NIL.

Brooks' Message

The message that Brooks put forward is that the robot must reuse components, it must be robust, it must operate in the real world, it must be cheap to build, it must have a subsumptive architecture which is distributed, incremental and parallel.

Application of Subsumptive Architecture on Real Robots

Subsumptive architecture has been used on the following early robots:

- Allen the first subsumptive robot, based on the ideas of Brooks [61].
- Attila and Genghis hexapods which walked autonomously [64].
- Polly, a tour guide for MIT AI lab [65].
- Seymour, a visual tracker [66]
- Squirt, a photosensitive miniature robot [67]
- Toto, the first map builder and the first to use Behaviour Language [68] plus others.

This proved that the concept and its application was fundamentally sound, and gave confidence to others to follow similar lines.

Evaluation

The following analysis of the strengths and weaknesses of subsumption architecture is taken from Arkin's book 'Behaviour Based Robotics' [69].

- Strengths Subsumption architectures can be compiled down directly onto programmable-array logic circuitry providing hardware retargetability. They provide support for parallel processing due to each behaviour level being able to run independently and asynchronously. They are easily adapted to allow niche targetability whereby custom behaviours can be created for specific task-environment pairs.
- Neither Strengths or Weaknesses Robustness can be successfully engineered into these systems but is often hard-wired and hence hard to implement. Timeliness of development depends on the existence of support tools for these systems, but a significant learning curve is still associated with custom built behavioural design. Experimental design, involving trial and error development, can slow development. Also consistent with Brooks' philosophy simulators are not used to pre-test behavioural efficiency.
- Weaknesses The priority-based coordination mechanism, the ad hoc flavour of behaviour generation, and the architecture's hard-wired aspects limit the ways the system can be adapted during execution and reduce runtime flexibility. Support for

modularity is low, although behavioural reuse is possible, it is not widely evidenced in constructed robots. Subsumption has also been criticised on the basis that since upper layers interfere with lower ones, they cannot be designed completely independently [70]. Also behaviours cannot always be prioritised (not should they be), leading to artificial arbitration schemes [70]. Commitment to subsumption as the sole coordination mechanism is restrictive.

In summary, Brooks' paper was a seminal paper that advanced the cause of robotics by twenty years. Previously it had become moribund, stuck in a loop of planning and action, without reference to the real world. Brooks started with the world as the operating environment and put the robot into it. The robot had to survive using the information that the world supplied. The direction was behavioural, not cognition.

2.4.2 Motor Schema Based Systems

Following closely after the introduction of Subsumption Architecture, are motor schema based systems. These systems are behaviour-based on natural and biological science. The ideas on motor schemas are based on work by Arbib [71] and Khatib [72].

Schema Theory — What does it Provide

Schema Theory was proposed by Arbib [73] and provides capabilities for specification and design of behaviour-based systems. It is a distributed computation model, providing a language to connect preception with action. The theory explains motor behaviour in terms of concurrent control of many different activities, with a schema holding how to respond and how to achieve the response. Schema Theory provides activation levels for schemas to determine their appropriateness for acting, and provides a learning mechanism through schema acquisition and schema tuning.

Motor Schema Development

The primary developer of Motor Schema Based Systems is Ronald A Arkin. The reason it was selected is that it attempted to account for the commonalities in both neurobiological and artificial behaviour [69]. Arkin wrote three papers addressing the implications

of scheme theory for autonomous robots. Neuroscience in Motion:The Application of Schema theory to Mobile Robots, [74], The Impact of Cybernetics on the Design of a Mobile Robot System:A Case Study [75] and Modelling Neural Function at the Schema Level:Implications and Results for Robotic Control [76].

Motor Schema Provision

Motor Schemas provide large grained modularity in contrast with neural network models, for expressing the relationship between motor control and preception. They are concurrent, parallel and object orientated agents which cooperate and compete and respond to external stimuli. They are easily onto distributed computer architectures. They have additive properties from which complex behaviours can be generated. They are grounded in neuro and cognitive science and easily adapted to later generation models.

'The overall aim of schema-based robotics is to provide behavioural primitives that can act in a distributed, parallel manner to yield intelligent robotic action in response to environmental stimuli' [69].

Differences between Motor Schemas and Subsumption Architectures

Arkin states [69], that the motor schema approach differs from other behavioural methods in the following ways. Behavioral responses are represented in a single uniform format as vectors in a potential field approach, genreating a continuous response encoding. Coordination is achieved through cooperative means by vector addition, however there is no predefined hierarchy for coordination. Behaviours are configured at runtime based on the robots aims, capabilities and environmental constraints. The schemas can be instantiated or destroyed at any time based on preceptual events giving a structure more like a changing network than a layered architecture. Pure arbitration is not used, rather weighted behaviours contribute to the final vector. The weights can be altered within the runtime to allow adaption and learning to occur. Perceptual uncertainty can be reflected in the behaviour's response by allowing it to be an input within the behavioural computation. It is possible to inject noise into the behaviour in order to generate randomness via a noise schema. This is a common method to avoid local minima in gradient decent methods such as annealing and mutation.

Design in Motor Schema Based Systems

Arkin [69] recommends the following approach to designing motor schema based systems. The problem domain should be characterised in terms of the motor behaviours necessary to do the task. The motor behaviours are then decomposed to their most primitive level, using biological studies wherever feasible for guidelines. Co-ordination is achieved through cooperative means by vector addition, however there is no predefined hierarchy for coordination. Behaviours should be configured at runtime based on the robots aims, capabilities and environmental constraints.

Schemas can be instantiated or destroyed at any time based on preceptual events giving a structure more like a changing network than a layered architecture. Pure arbitration is not used, rather weighted behaviours contribute to the final vector. The weights can be altered within the runtime to allow adaption and learning to occur. Perceptual uncertainty can be reflected in the behaviour's response by allowing it to be an input within the behavioural computation. Noise can be injected in to the system and formulae an be developed to express the robot's reaction to perceived environmental events. Simple simulation studies can be conducted assessing the desired behaviours' approximate performance in the proposed environment. The perceptual requirements needed to satisfy the inputs for each motor schema should be determined.

Design specific perceptual algorithms that extract the required data for each behaviour, utilising action-orientated perception, expectations and focus of attention techniques to ensure computational efficiency. The resulting control system is then integrated onto the target robot. Finally the systems performance is tested and evaluated and the process iterated to expand the behavioural repertoire as necessary.

Application of Motor Schema Architecture on Real Robots

Examples of early robots using motor schema architecture are as follows:-

- HARV - A Denning robot that exhibited a wide range of behaviours
- George - Showed behaviour based docking capability [77], teleautonomy [78], and avoid past behaviours [79].
- Ren and Stimpy - Multiagent behavioural research

Evaluation

The following analysis of the strengths and weaknesses of schema theory is taken from Arkin's book, 'Behaviour Based Robotics' [69].

- Strengths Support for parallelism: Scheme theory is a distributed theory of computation involving multiple parallel processes. Motor schemas are naturally parallelised. Run-time flexibility: As schemas are software agents, instantiated at run time as processes and are not hard wired into the architecture it is simple to reconfigure the behavioural control system at any time. Timeliness for development and support for modularity: Schemas are essentially software object and are by definition modular. They can be stored in behavioural libraries and are easily reused.

- Neither Strengths or Weaknesses As with any reactive system, schemas are robust and can cope well with change in the environment. One deficiency lies in the use of potential field analogues for behavioural combination, which has several well known problems (stalling). Specific methods, however, such as the introduction of noise and the avoid-past behaviour, have been developed to circumvent this difficulty.

- Weaknesses Niche targetability is not easily achieved. Although feasible to design niche robots, the generic modular nature of the primitive schemas somewhat discourages the design of very narrowly focused components. Hardware targetability is also a problem. Schema based systems are essentially software architectures mappable onto hardware multiprocessor systems. They do not provide the hardware compilers that either subsumption or Gapps does. Hardware mappings are feasible, however, just not as convenient as with some other systems [80].

Motor Schema architecture offers a flexible, programmable way of implementing behaviour-based control. The ability to be able to store schemas and implement them at runtime means that the robots are extremely flexible in the way that they can deal with real world problems. In fact it should be possible to implement subsumptive architecture using motor schemas, without anyone being able to detect the difference.

2.4.3 Other Architectures

Circuit Architecture

Strength provides/involves the use of abstraction through the bundling of reactive behaviours into assemblages and by allowing arbitration to occur within each level of abstraction, what the designers called hierarchical mediation [81]. It provides modularity, awareness and robustness. It was manifested in a robot called Flakey, using the programming methods Rex and Gapps.

Action-Selection

This architecture has only been used in simulation. However, its designer, Pattie Maes of MIT, is widely known in Robotics [82] and [83]. Individual behaviours have associated activation levels that provide the basis of runtime arbitration. Additionally an activation level has to be above a threshold to be considered applicable for a task. The activation level for any behaviour is affected by the current situation, higher level goals, spreading activation from other behaviours and inhibition from conflicting modules. The idea is to pick the highest activation level from all behaviours that meet the pre-conditions. It has to run as quickly as possible. It provides greater emergent behaviours since there is no predefining of levels as in subsumption. Its advantage is that it is flexible and open. There is no indication as to how any behaviours have to be generated and what pre-conditions have to be met. This architecture also requires an arbitrator to be implemented.

Colony Architecture

This architecture differs from subsumption by having a treelike ordering of behaviours. This gives a more flexible response to situations. It is implemented using a priority based arbitration with a suppression only, coordination strategy. It was implemented on a robot called Herbert, which collected cans for re-cycling at MIT by Connell [84].

Animate Agent Architecture

This architecture is based upon Reaction Action Packages (RAPs) with two extra components, a skills system and a problem solving module [85] and [86]. This architecture is

task based rather that behaviour based. They provide multiple methods of acting within a given situation. A sequence of steps is provided to achieve a method. This however looks like planning to me, although a weak plan. Animate Agent was implemented in a robot called CHIP which carried out rubbish removal.

Distributed Architecture for Mobile Navigation (DAMN)

This is a fine grained subsumptive architecture. The behaviours are a collection of asynchronous each generating outputs as a collection of votes cast over a range of responses. There are many ways in which the votes can be cast including statistical options over a range of responses. A winner takes all strategy for the response with the highest vote is enacted [87] and [88]. This architecture was used by DARPA for battlefield scouting tests. These was a suite of behaviours which were resolved by DAMN, for both Turn and Steer. The results were sent to the actuators for implementation.

Summary

Every robotic application requires an architecture, however there is no best one. It is a case of deciding, out of all the available ones, to select that which provides the closest match to the problem. A guide to this process is to try to implement a layered architecture, due to their flexibility and ability to operate in parallel and at multiple levels of extraction.

2.5 Robotic Strategy Methods

This section describes the major strategies used in the development of robotics. The numbers reported are searches used in Google Scholar that combine Robot and the name of the strategy. This is a first pass heuristic giving a measure as to the research carried out. Further searching could be carried out around FIRA, Robocup and Robosoccer, to give a measure of research applied to each strategy. The use of robot football implies multi robot system (MRS) coordination. A summary of the results of the search are give in Table 2.3

2.5.1 Potential Fields (PF)

The potential field strategy is based on ideas by [89] and [90]. It provides a method by which robots can avoid objects or be attracted to them. It is based on the inverse square law relationship between force and distance and/or Coulomb's Law of charged particles. It was investigated by Arkin and described in his book 'Behavior Based Robots' amongst others [69]. In the Coulomb's law version of the strategy the force between two objects is calculated as

$$F = \frac{Q_1 Q_2}{d^2}$$

where Q_1 is the signed charge on the first object, Q_2 is the signed charge on the second object, and d is the distance between them. The advantage of this implementation is that the force can be attractive or repulsive, according to the charge signs allocated to the objects. Potential field strategies have the advantage of being easy to program and because they encode a continuous navigational space through the sensed environment, they provide an infinite set of reactive possibilities. However they also have problems as outlined in [90]. Particularly they are vulnerable to getting stuck in local minima, and cyclic-oscillatory behaviour. Another disadvantage is the time required to calculate the entire potential field. This is overcome in reactive systems by only calculating each fields force contribution at the instantaneous position at which the robot is located. This also allows for potential fields to be used in highly dynamic environments and in a highly parallel implementation.

2.5.2 Neural Networks (NN)

Neural Networks are essentially learning devices, which given our robots might yield interesting strategies. We would need to do this in simulation and then try out the strategy with the robots. There are many neural networks available to use, the following might be of interest to investigate on the Maibot micro robots :- Perceptrons and Multilayer perceptrons, which incorporate feed forward neural networks with hidden layers and pack propagation [91]; Hopfield networks [92], Bidirectional associative memory(BAM) and self-organising neural networks such as Hebbian learning and competitive learning [93].

2.5.3 Fuzzy Systems (FS)

A fuzzy system is a system which handles imprecise data using the principles of fuzzy sets and their associated methods as proposed by Zadeh [1]. Fuzzy inference is a process of mapping from a given input to an output, using fuzzy set theory. Four steps are involved: fuzzification of the input variables, evaluation of the rules, aggregation of the rule outputs and defuzzification of the result. Two fuzzy inference techniques are commonly used.

- Mamdani - good ability to capture expert knowledge in fuzzy rules, but with a high computational overhead.
- Sugeno - a singleton membership function is used. Works well with optimisation and adaptive control, good for non-linear dynamics.

2.5.4 Rule-Based Systems (RB)

A rule based system has five basic components

1. Knowledge base - contains domain knowledge represented as rules
2. Database - set of facts that are used to match against the IF part of rules
3. Inference engine - links rules with facts and provides reasoning for the solution
4. Explanation - Allows queries as to how a conclusion was arrived at, and why a specific fact is needed.
5. User interface.

They allow natural knowledge representation, uniform structures, separation of knowledge from its processing and ability to handle incomplete or uncertain knowledge. The disadvantages include no ability to learn, opaque relationships between rules and inefficient searching.

2.5.5 Reactive Systems (RS)

Reactive systems are tightly coupled perception and action systems, to provide timely responses in dynamic and unstructured environments. Reactive systems can also have planners added to them, the Planner-Reactor Architecture for example [94] and [95].

2.5.6 Other Methods

Genetic Algorithms (GA)

Genetic algorithms (GA) are a class of stochastic search algorithms based on biological evolution. Given a clearly defined problem to be solved and a binary string representation for candidate solutions a basic GA can be represented.

Genetic Programming (GP)

Genetic Programming represents an application of the genetic learning model to generate a computer program that solves the problem. This area of investigation took off in the early 1990's stimulated by John Kosa [96] and [97].

Hybrib Neuro Fuzzy (HNF)

Neuro-fuzzy systems are neural networks which are functionally equivalent to a fuzzy inference model. The NF system can be trained to generate IF-THEN fuzzy rules and determine the membership functions for the input and output variables. It is easy to incorporate expert knowledge into the neuro-fuzzy structure. A benefit is that the connectionist structure avoids fuzzy interference and its attendant computational overhead.

Hybrid Neural Expert (HNE)

Neural Expert systems combine a neural network with a rule-based expert system, using domain knowledge in IF-THEN rules as well as sets of numerical data. They are limited by Boolean logic, and attempts to represent continuous input variables can lead to infinite increases in rules.

Adaptive Neuro Fuzzy Inference Systems (ANFIS)

Adaptive Neuro Fuzzy Inference Systems was first proposed by Roger Jang of the Tsing Hua University, in Taiwan. It is very good at generalising and rapidly converging, so can be used in on-line learning. Applications include adaptive control.

Frame-Based Expert Systems (FB)

These are essentially object orientated expert systems, each frame in a class hierarchy being more specialised. Rules often use pattern matching to locate conditions amongst the instance-frames.

Immune Systems (IS)

Detection of foreign agents by pattern recognition. Can neutralise with antidote and isolation.

Virus/viral Systems (VS)

Invasion systems which mimic the real agents but carry a different message. Normally malevolent.

Path Planning (PP)

These include both deliberative(planning) and reactive strategies. These can also include potential field strategies

Deliberative Systems (DS)

Deliberative or Planning Systems use symbolic knowledge and reasoning to generate a strategy. Extremely well adapted for manufacturing tasks.

Table 2.3: Number of Hits in Google for Robot Soccer Strategies

Method	PF	NN	GA	GP	HNF	HNE	ANFIS	FS	RB	FB	IS	VS	PP	DS	RS
Title	62	584	125	85	52	0	1	230	32	3	27	0	650	0	1
Body	2900	18300	5540	2460	304	18	244	7240	1570	287	859	9	15700	124	872
Soccer	235	626	0	288	8	0	17	726	70	16	28	0	602	21	78
FIRA	18	50	20	15	3	0	0	90	4	3	16	0	62	0	2
robocup	171	466	224	249	5	0	14	588	69	9	16	0	447	23	74
robosoccer	3	14	7	12	0	0	0	26	2	0	1	0	24	0	3
mirosot	29	24	10	14	0	0	0	69	3	1	0	0	65	0	4

Where:- PF - Potential Fields, NN - Neural Networks, GA - Genetic Algorithms, GP - Genetic Programming HNF - Hybrib Neuro Fuzzy HNE - Hybrid Neural Expert, ANFIS - Adaptive Neuro Fuzzy Inference Systems FS - Fuzzy Systems, RB - Rule-Based Systems FB - Frame-Based Expert Systems, IS - Immune Systems, VS - Virus/viral Systems, PP - Path Planning, DS - Deliberative Systems, RS - Reactive Systems

2.6 PID Control

The most used feed back controller design in current use, is probably the PID controller. PID is the commonly used acronym for this controller standing for Proportional-Integral-Derivative. Alternative names included a three term controller referring to the number of components that make up the controller. The controller operates in a sense-think-act cycle.

```
Repeat
     sense the current state
     reduce difference between current state and required state
Until
     current state = required state
```

At time t, If y(t) is the current state and r(t) is the required state, then the error e(t) = r(t) - y(t). At time t, the current error is the Proportional component, the sum of the errors up to time t is the Integral component and the rate of change of the error is the Differential component.

The control action at time t CA(t), is then simply the weighted sum of the three components.

$CA(t) = K_P e(t) + K_I \int e(t)\mathrm{d}t + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t)$

The required closed loop dynamics of the controller is then achieved by adjusting the three parameters K_P, K_I and K_D. This is often done iteratively by the tuning process and without specific knowledge of a plant model. There are many instances where all three parameters are not needed to control a process. The appropriate parameters can be set to zero in order to create PI, PD, P or I controllers. The PI controller is very common since the I component removes steady state error and the D component is sensitive to noisy measurements. The effects of the parameters within the controller are as follows. The Proportional term $K_P e(t)$ is usually the dominant term of the controller. A high proportional gain results in a large change in the output of a given error. If the gain is too high then the system can become unstable. Too low and the response time to changes become unacceptably slow. Stability can often be achieved by using the proportional term only. However the system will not settle at the setpoint. To achieve the setpoint

the Integral term $K_I \int e(t)\mathrm{d}t$ is required. The integral term contributes proportionally both to the magnitude and the duration of the error, being the sum of errors over time. The integral term accelerates the process towards the setpoint and eliminates any steady state errors. However the drawback is that it can cause overshoot to occur generating opposite error and drive the process away from the setpoint. If the system is too dynamic then the output oscillates about the setpoint and hits the control limits of the device. Reset is often used in controllers to prevent Integral windup. This is when the Integral term exceeds a given value, then the Integral term is reset to zero. The contribution of the Derivative term, $K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t)$, is proportional to the rate of change of the error over time. It is calculated as $K_D(e(t_0) - e(t_{-1}))$ for unit time. The effect of the Derivative term is to slow the rate of change of the error, known as damping. It acts to reduce any overshoot effects caused by the Integral term and is most effective when the process is close to the setpoint. However it amplifies noisy signals and can cause instability to occur in the system. This problem is usually solved by having a filter on the input and adjusting the controller parameters accordingly. PID controllers are the most established class of controllers however they cannot properly control non-linear systems or systems that have multiple inputs and outputs. There are many reference books describing PID controllers and methods to tune them, as given in [98], [99], [100] and [101].

2.7 Robot Football

As stated previously in Chapter 1, one of the major motivations in getting involved with this research was the presence of a robot football facility in the University of Nottingham's robot laboratory. There are two major organisations that govern the playing of competition robot football. The first is FIRA [102], to which the University of Nottingham (UoN) is affiliated, and the other is The RoboCup Federation [103]. Each organisation has different rules governing the configuration of the game, but share a common goal in developing, promoting and advertising the use of robots. The UoN is unable to compete under RoboCup Federation rules due to use of BlueTooth wireless communications, which is banned by that organisation. There are no restrictions on the software deployed on the robots. Except where stated this literature review refers to the FIRA organisation

and mirosot middle league rules.

Robot football is a problem that combines strategy, cooperation and control among a team of robots. The highly dynamic nature of robot football makes it so difficult to solve, especially as the robots are capable of speeds in excess of 3msec^{-1}. During a robot football game robots frequently collide with each other, often at high speed. Although in the rules this is a foul, it is usually only called when it occurs in the penalty area. Also it is permitted to push another team member subject to occupation rules. The UoN robot football team is described in [104].

2.8 Potential Areas of Research Interest

Since the introduction of subsumption architecture [61], robot development has tended to be behaviour based. However, there is a vast amount of work available on the mechanical attributes of robots and the control paradigms used by them. This section identifies projects and experiments that encompass low level robot control, high level strategies and a mix of both.

The following sections identify work that is concerned with the physical robot as a standalone entity. The first two sections are designed to capture physical and system characteristics of the robot. The third section is a completely new development whereby the system runs a miabot simulator in parallel and in real time.

Odometry

Apart from the manufacturers specification there is no published work on the characteristics of Merlin miabots. Before any major research is carried out, a comprehensive study is required on the performance characteristics of these robots. This work is mainly odometry and tuning of the PID control algorithm supplied by the manufacturer. A list of experiments is given that would provide this information.

- PID control parameter optimisation
- Velocity and acceleration
- Straight line errors

- Circular errors
- Braking distances and deceleration
- Forces — stiction, friction and slippage

An enhancement to the robot controller would be traction control and power smoothing, which currently does not exist.

Error Measurement by Path Tracing

Measurement of the effects of system noise, camera abberation, timing delays and other unknowns needs to be investigated. The robots would have to follow a pre-determined path through an environment. The error generated by the robot from the path would be measured at various speeds. It is known that positional error is greater in the corners of the football pitch for example. This would provide a form of calibration.

Develop a Miabot Simulator Model

This is a framework project to simulate the miabot at every stage of its application loop. The simulator is integrated into the RSE and run in real time. Initially its purpose would be as a tuning tool for the robot controller. It would provide a facility where the real world and simulation can be compared instantaneously. Multiple parallel simulations can be carried out on upto five robots, and the best selected for actioning on the robots.

The Effect of Low Level Controllers on High Level Strategies using Robot Football

These experiments are designed to investigate if PID, Type I or Type II fuzzy controllers provide the best control platform for playing robot football. Games of robot football are played using the same strategy only varying the robot controllers. Then a further set of experiments are run where the teams are made heterogeneous in terms of their controllers. Further variations can be made by changing the strategies that each team plays.

Attack and escape behaviour

This series of experiments would be designed to discover strategies which could be employed to attack and escape. The generation of the strategies would be via a learning

system that used punishments and rewards. The outline is a follows. There is a single gazelle grazing. Near by are a pride of lions. The aim is that the cats have to kill the gazelle. However only three cats can hunt a single gazelle A kill is defined as containing/holding the gazelle in an inscribed circle of the triangle of the cats, for a specified amount of time. The time in the circle and the radius of the circle can be varied with each experiment. Hitting the gazelle does not count as a kill. The cats have a time dependent top speed that declines with time from an initial charge. Similarly for the gazelle. The top speed of the gazelle is greater than that of the cats. The change of direction of the gazelle is greater than that of the cats. The gazelle can charge a cat and not be killed, however it suffers a speed penalty proportional to the time in contact with the cat. Similarly if two cats collide they lose energy. The length of the attack is determined by the cat speed profile or if the gazelle can get to the edge furthest away from its current position. Each cat has a different profile, some have a higher top speed but worse cornering, some can accelerate faster but have a slower top speed. others have better cornering but lower top speed. The gazelles are similarly varied, some better detection range, better cornering, different top speeds. The longer the pride go without a kill the worse is their performance. The time of day effects the system. This is down to the vision system being effected by different light levels. There is a breeding program whereby lionesses can generate off-spring by mating with lions. Similarly for the gazelles. Further variations can be introduced, by altering the number of cats, and the number of gazelles. The type of cat can be changed (for example, lion $\rightarrow$ cheetah), as can the prey (gazelle $\rightarrow$ gnu, which can kill a lion). Also have scavengers i.e. a pack of hyenas vs. a cheetah for the kill. As the cats age they have more skills but are slower. Young cats are more impetuous etc. These situations all require cooperation and collaboration to succeed. However unlike football the target is also reactive. The speed that the robots move is relative and so wheels/tyre problems should not occur. The research would require a simulator to be built constrained by the soccer pitch dimensions. After each generation of cats and prey, the simulations would be run to generate new strategies. These would them be run on the real robots. Each cat has a genetic and kill history. All the cats in the pride have to survive which means feeding themselves and any cubs. the variations are endless. The gazelles have different weights and so become more tempting to attack.

Collaboration and Cooperation Behaviours

In these experiments an egg/golf ball has to be transported from one place to another. The problem is to discover how many robots would be needed to do this. However each robot has a handicap which effects its ability to do the task. Need to workout what the problem is and come up with a solution on how to do it. Need to do it with 2,3,4 robots. Apply time limits, speed limits, non linear paths. Avoidance techniques, different strategies. Could strategies be learnt to do the task in simulation/learning algorithms and then be applied to a set of miabots and do it for real using the developed technique.

Ball Passing Experiments

Get three robots to play passing with different ball types. Measure the effectiveness of strategies against the ball type. Again vary time and speed, strategies, accuracy of passing. Pass balls in a circle holding the radius to limits, with up to five robots involved.

Use of Immune and Viral Techniques Situated in a Robot Football Team

This project would be designed to discover if immunity and infection could be applied to a football team. Learning methods previously described would be used to develop strategies in simulation. These would then be tested out on real robot football teams for effectiveness.

2.9 Recent Work on Type-2 Fuzzy Sets and Systems

In 1975 Zadeh proposed 'fuzzy sets with fuzzy membership functions' as an extension of the fuzzy set. This produced fuzzy sets of $type-n, n = 2,3,...,$; the membership function ranging over fuzzy sets of type n-1 [46]. Type-n fuzzy sets are necessary because although type-1 fuzzy sets, as they became known, address a lot of the real world uncertainty they do not address all uncertainty. Examples are the variability of expert opinion on a fuzzy set, and their self-referencing variability over time; opinions do change. Noise of the system and errors of measurement also have an effect. Each type-n fuzzy set is then a measure of the variability of the preceding types.

The use of type-n fuzzy logic has been taken up and extended by many researchers. Type-2 fuzzy sets are shown to allow the modelling and minimization of the uncertainty of type-1 fuzzy sets [22]. However, some of the major problems with type-2 fuzzy sets are difficulty of understanding, envisaging how they look due to their 3-D nature, and the computational complexity needed to generate solutions. The derivations of union, intersection and complement all rely on the use of Zadeh's Extension Principle. [35].

Mendel and John [44] comprehensively addressed these issues by 'presenting a new representation for type-2 fuzzy sets' and 'using this representation to derive formulas for union, intersection and complement of type-2 fuzzy sets without having to use the Extension Principle'. By using interval type-2 fuzzy sets, characterised by secondary membership functions taking values of either 0 or 1, the type reduction necessary for defuzzification of type-2 fuzzy sets becomes simplified [60]. Although interval type-2 fuzzy logic controllers are a lot less computationally intensive than the general type-2 FLC, there is a computational overhead that is directly proportional to the number of rules [22], [60].

2.9.1 Mobile Robot and Control Applications

Many researchers have used type-2 fuzzy logic in control applications. Type-2 interval systems have been used by Melin and Castillo [105, 106] and Castillo et al. [107] in industrial plant control systems. Wu and Tan [108] applied type-2 interval systems to the control of a complex multi-variable liquid level process. In [109] they simplified the application of type-2 fuzzy logic control to real-time control applications. A type-2 interval system was used by Doctor et al [110] and Hagras et al. [111, 112] to model and adapt to the behaviour of people in an intelligent dormitory room. Type-2 fuzzy logic control applications were investigated by Hagras [113] including industrial, mobile robots, and ambient intelligent environments control. With regard to hardware implimentations of type-2 fuzzy logic controllers, Wu [114] designed and implemented type-2 fuzzy logic control on Motorola 68HC11 8-bit micro-controllers. The micro controllers were used to navigate a miniature robot in an unknown maze without touching the walls. Melgarejo et al. [115] developed a limited hardware implementation of a type-2 interval controller, and Lin et al. [116, 117] designed a type-2 fuzzy controller for buck DC-DC converters.

The first application of a type-2 fuzzy logic controller to an autonomous mobile robot was implemented by Hagras [118], who demonstrated that it outperformed a type-1 FLC. The architecture of the controller was based upon interval type-2 fuzzy logic controllers which were used to implement the basic navigation behaviours, and also the coordination of them to produce a type-2 hierarchical FLC. Experiments were carried out in a laboratory environment and also outdoors. The environments were challenging, dynamic and unstructured in nature. Numerous experiments were carried out including night time operation. It was shown that the type-2 controllers dealt in real-time with the uncertainties of the environments. The results obtained showed very good real-time control responses, which had outperformed the equivalent type-1 FLCs and HFLCs. There was about a 64% reduction in the number of rules for the type-2 FLCs and HFLCs to those used in the equivalent type-1 configuration system. The first instance of an industrial DSP embedded platform, with a real time type-2 FLC, used to control a marine diesel engine was by Lynch *et al* [17], [119]. They found that the type-2 FLCs dealt with the uncertainties in real-time and produced a robust control response. This demonstrated that the embedded type-2 FLCs outperformed the PID and type-1 FLCs previously used to control the marine engine whilst using smaller rule bases. Coupland has shown that the use of geometric methods can resolve the computational overhead required in general type-2 fuzzy logic, and so allow it to be applied to time critical control problems [45]. This was demonstrated in [120], where a general type-2 FLC outperformed both an interval type-2 and a type-1 FLC, all executing the same tasks. Studies comparing type-2 and type-1 FLC performance have shown that the best results are given by the type-2 controllers [121], [122]. In the realm of robot soccer games Figueroa et al. [123] explored how the type-2 fuzzy logic controller overcome the uncertainty in the control loop without increasing the computational cost of the application. Hagras recently described a method to develop a type-2 FLC through embedded type-1 FLCs demonstrating that the type-2 FLC outperforms the type-1 FLCs that it was based on [124].

Further theoretical and practical work on defuzzification has been carried out by Coupland and John [125]. Generalised type-2 fuzzy logic systems have not been used to solve practical problems due to the amount of computational effort required to defuzzify a generalised fuzzy set. Their method involved the use of geometrical representations

and operations. The analysis of the results showed that the new representation was not detrimental to the accuracy of the set. The use of the geometrical technique was shown to be over two hundred thousand times faster than the traditional type reductions when compared in real world experiments.

Wagner and Hagras in 2009 carried out work using the recently introduction concept of zSlices based general type-2 fuzzy sets to implement a zSlices based general type-2 FLC (zFLC). [126]. They investigated this new approach using the implementation and operations of the zFLC for a two-wheel mobile robot navigating in real world outdoor environments. A performance analysis showed that the results for the zSlices based general type-2 FLC were very promising, when compared with type-1 and interval type2 FLCs in certain circumstances.

In 2009 Hamwari and Coupland used interval analysis to explore type-2 fuzzy arithmetic, with an approach involving alpha-planes and alpha-cuts [127]. The paper also discusses the use of quasi type-2 fuzzy sets proposed by Mendel and Liu [128]. They defined quasi type-2 fuzzy numbers and arithmetic operations for use on them.

In 2009 Greenfield et. al. proposed the collapsing method of defuzzification of interval type-2 fuzzy sets [129]. The collapsing method converts an interval type-2 fuzzy set into a type-1 representative embedded set (RES). The defuzzified values of the type-1 set closely approximate those of the type-2 set. The RES being a type-1 set can then be easily defuzzified. The representative embedded set approximation (RESA), to which the method is inextricably linked, is expounded, stated and proved. It has two forms: Simple RESA where this approximation deals with the most simple interval FOU, in which a vertical slice is discretised into 2 points. Interval RESA: where this approximation concerns the case in which a vertical slice is discretised into 2 or more points. The collapsing method (simple RESA version) gave excellent results, when tested for accuracy and speed. The collapsing method proved more accurate than the Karnik-Mendel iterative procedure (KMIP) [53] for an asymmetric test set. For both a symmetric and an asymmetric test set, the collapsing method was shown to outperformed the KMIP in relation to speed.

The researchers then investigated if the direction of collapse in the collapsing method affect the accuracy [130]. The collapsing defuzzifier comes in many variants which de-

pend on the direction of collapse. The accuracy of the fundamental variants of forward, backward, inward and outward, and the composite variants of forward-backward and outward right-left were compared experimentally for the discretised interval type-2 fuzzy set. The results showed that for the three tests sets used, horizontal, triangular and Gaussian, that the best performing variant was outward, followed by inward, then forward and backward. For the symmetrical sets (horizontal and triangular), the errors of collapsing forward were equal and opposite to those of collapsing backward. Therefore in these cases it was expected that collapsing forward-backward would give exact results, which was confirmed by experiment. For the Gaussian test set, backward performed more poorly than forward. In this case the composite of forward-backward performed worse than forward, though better than backward. The reason given for the outward collapse to be the most accurate for the symmetrical sets, was due to the initial selection of the central slice. The greatest inaccuracy is associated with the first collapsed slice. To achieve maximum accuracy, the ideal place for this first slice to be positioned is centrally, as the effect on the defuzzified value obtained is then minimal.

Finally a review paper was generated which surveyed some of the current type-reduction methods for interval type-2 fuzzy sets [131]. The paper reported on an investigation on how discretisation affects performance. The collapsing method was compared in detail with the Nie-Tan methods for type-reduction [132]. In the usual case where the set to be type-reduced is asymmetrical, the collapsing method produces more accurate results than Nie-Tan, and as the discretisation increases the performance of the tow methods converge.

In 2009 and 2010, Miller, Gongora and John, building on previous work [133, 134], introduced a interval type-2 fuzzy logic model of a multiple echelon supply chain [135]. The use of a type-2 FLS allowed the uncertainty and vagueness present in supply chain operation and resource planning models to be better represented. A Genetic Algorithm used the supply chain model to search for near-optimal plans of the scenario. It was shown that the GA was able to find good multiple echelon resource plans that were both cost-effective and sensible. In this case the GA was solely guided by cost and service level. The fact that the GA found these solutions, was taken to indicate that the model was valid.

Following on in 2010 Miller et al, used a series of evolutionary algorithms to find good

solutions for the resource plans of the supply chain model [136]. Using these resource plans the same evolutionary algorithm techniques, inventory optimisation was carried out using the model [137].

2.10 Summary

In this chapter was given the elements around which the majority of the research was based. This provides a firm practical and theoretical basis on which to proceed with the investigation on how to meet the research aims and objectives of the thesis. In the literature review investigation there were no papers found on applying type-2 fuzzy logic controllers to micro robots with limited resources, except those by myself. This gives added confidence that the research would make a contribution to knowledge. This thesis seeks to discover a method by which the advantages of type-2 fuzzy logic can be successfully applied as a micro robot controller, and then consequently as a novel generic type-2 fuzzy logic controller.

Chapter 3

Experimental Setup

In this chapter the robot used in all the practical experiments is introduced, the Merlin Miabot Pro robot. A detailed description of the environment that the robots were operated in is provided. Technical descriptions of the robot are given. The software that I developed to enhance the supplied software is described. The implementation of a type-2 Fuzzy Inference System programmed in Java is described. Descriptions of how the robots are configured and controlled are detailed.

3.1 Laboratory Environment

All the practical experiments were performed in the University of Nottingham's robot laboratory. This is a multi purpose laboratory consisting of three sections. The first section is a twin robot enclosure area surrounded on three sides by tables carrying computers. This is used primarily for undergraduate teaching and experimentation. The second area is the robot soccer section. This is where the practical experiments of this thesis were performed. The third section is the robot village were both large robot experiments and multi micro robot experiments are carried out. The robot soccer section contains a standard 5-a-side robot soccer pitch with a playing area of 2.2m by 1.8m. The pitch is painted black with white lines marking out the playing area. It is lit by seven strip lights suspended 2.4m above the pitch which provide an approximately even 750 LUX at the surface. The laboratory is equipped with blackout curtains so that there is no variability due to external light. Two cameras are attached to the lighting rig and positioned approximately

Figure 3.1: Maibot Pro BT' Robot

centrally over the pitch. The first camera which was used in the practical experiments is a MicroPix Firewire C640 low resolution camera operating at 30 frames per second. The second camera is a MicroPix Firewire C1024 high resolution running at 15 frames per second. The MicroPix C640 camera delivers a 640 x 480 pixel image with a pixel covering an area of approximately 12.89mm^{2}at the pitch surface. The MicroPix C1024 has a 1024 x 768 pixel image giving a pixel coverage of approximately 5.03mm^2. Both cameras provide a 24 bit RGB colour data output. The cameras are controlled through a Unibrain Fire-i API to the IEEE-1394 digital camera standard. Each camera is connected to a laptop computer. The laptop computers used in the practical experiments were DELL Notebooks, with a Pentium4 processor running at 2.80GHz and 512Mb RAM. Each of DELL Notebook's operating system was the Microsoft Windows 2002 XP with Service Pack 2.

3.2 Miabot Robots

The robots used in the practical experiments were Merlin Systems Corp. Ltd. 'Maibot Pro BT' robots as shown in Figure 3.1. These are micro robots and were designed to meet the rules of the FIRA MiroSot Middle League competition. The rules for MiroSot can be found at http://www.fira.net. The detailed specification describing the features of the robots is given in Table A.1.

The Maibot Pro BT robot is a two wheeled micro robot with independent motors driving each wheel. This allows a differential steering drive for the robot. Each motor has an optical encoder giving a linear resolution of 0.04mm. The maximum speed is

reported to be 3.5 m/s with a linear straight line speed of upwards of 1 m/s. The main processor board is an Atmel ATMega64 with 64k of flash memory and supports up to eight digital IO devices. The communications board is a Bluetooth Wireless bidirectional board transmitting in the 2.4GHz range over a distance of up to 100m and at a rate of 115kb/s.

3.3 Robot Soccer Engine

The Robot Soccer Engine (RSE) is a software package that was supplied by Merlin Systems Corp. Ltd. and provides all the interfaces necessary to set up and control up to five individual Miabot Pro BT robots simultaneously. These include robot identification, pitch mapping and allocation of COM ports for the Bluetooth Comms package. The RSE starts and stops the robots and displays the image received from the attached camera whilst the robots are on the robot soccer pitch. As the name suggests it was primarily designed to enable micro robots to play robot soccer. The process that is used by the Robot Soccer Engine is as follows. The Robot Soccer Engine receives frame pictures from the camera via the Fire-i API. Each frame is processed by the **ProcessImage** routine, and current robot and ball positions are calculated. The **ProcessImage** routine passes *RobotPositions* to the **User Defined Strategy** routine. The **User Defined Strategy** is where *MoveTo* commands are calculated for each of the robots, according to the strategy loaded at the start of the game. When completed, the **User Defined Strategy** calls the **Dispatcher** routine. The **Dispatcher** routine packages the commands into messages for each robot. The messages are then sent via the **Bluetooth Comms** package to the corresponding robots. The *MoveTo* messages are processed and actioned on the robots, so that the time spent in RSE processing is a minimum. This is shown in Figure 3.2. However in order to enable the Robot Soccer Engine to control and report data from the practical experiments modifications and enhancements had to be made, over and above those made to get the Robot Soccer Engine to the FIRA competition standard. These included introducing stop watch facilities for timing events, a Bluetooth channel read utility, and writing data to a file for subsequent processing. Identification of individual robots was achieved by placing coloured caps on the top surface of the micro robots. In MiroSot matches the two teams

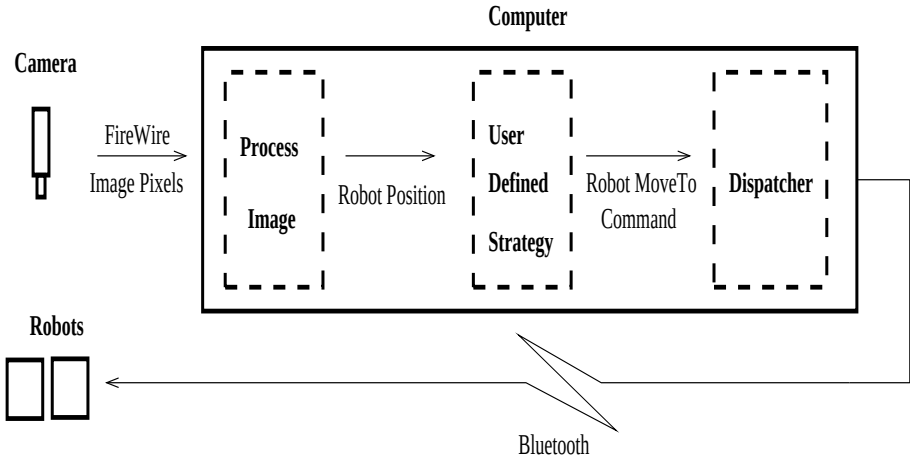


Figure 3.2: Robot Soccer Engine Overview

are identified by either a blue or a yellow colour on a cap. In order to run the practical experiments under the control of the Robot Soccer Engine all the robots had the same team colour caps. The actual team colour used was not important as the Robot Soccer Engine has a team colour definition option. Usually the team colour cap selected to run the practical experiments was the blue cap set. In the FIRA rules for robot soccer the ball has to be a standard golf ball coloured orange. However it is not a requirement that an individual player is required to be identified. If the players are identified, the identifying colours must not be blue, yellow or orange. The Robot Soccer Engine uses a set of caps which are labelled P1 - P5. These are colour coded so that the Robot Soccer Engine can identify each robot. A three colour Hue Saturation Intensity (HSI) system is used with two different sized patch squares for each player. The large patch is $33mm^2$ and the small patch $23mm^2$. The team patch is the same size as the large patch. The Robot Soccer Engine requires the team colours blue and yellow; and the player colours magenta, green and cyan, and the ball colour orange to be defined before the game starts. The actual hues are entered as ranges against the required colours. To complete the identification the minimum and maximum blob size in screen pixels for each patch must be entered into the Robot Soccer Engine. This is used to ensure that a colour patch has been correctly identified. However the greater the range in the blob size, the slower the identification process. The reason for having a three blob identification system over a two blob system is that when the robots get close or collide they are still identifiable. In our practical experiments

Player	Large	Small	Role
P1	Green	Green	Goalkeeper
P2	Cyan	Cyan	Defender
P3	Magenta	Magenta	Left
P4	Magenta	Cyan	Right
P5	Cyan	Magenta	Attacker

Table 3.1: Player Identification and Role

Colour	Hue	Sat
Blue	144	120
Yellow	40	240
Magenta	200	240
Green	96	120
Cyan	120	120

Table 3.2: Colour Definition at Intensity of 120

the loss or swapping of identity rarely occurred and was quickly corrected if it did.

During a game the Robot Soccer Engine provides information about each player on the camera image screen. This is a confidence measure of the identity of a player. The range is 0-99 where 99 is the most confident prediction of identity. Whilst a game is being played the values reported are about 80. The player identification is given in Table 3.1.

In order to determine the most recognisable blob colours for the Robot Soccer Engine image processing system, a series of experiments were carried out. A set of colour charts of small patch (23mm x 23mm) squares were printed out on a colour printer. The colours were generated for Hue within Saturation within Intensity. Only two Intensity values of 120 and 136 where used. The Saturations used were 60 - 360 in steps of 60. The Hues were 8 - 360 in steps of 8. The colour chart blobs were labelled on the back with their HSI values. The charts were then cut up into blobs and then put under the cameras and the Hue values read off using the camera image screen. Various camera aperture settings were tested, and the most robust colour values are given in Table 3.2. Two sets of five player identification caps were then generated according to the colour combinations in Table 3.1, one set for the Blue team and the other set for the Yellow team. The player cap for the Blue team goalkeeper, Player P1, is shown in Figure 3.3. The RSE requires the pitch to be defined on the camera image screen. The user clicks on the corners of the pitch

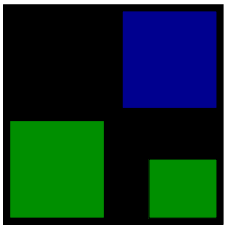


Figure 3.3: Robot Player Caps

image from which the corner pixel values are obtained. This frames the pitch in terms of the camera image. The position of the robots within the pitch can then be calculated by the Robot Soccer Engine image processor. Within the **User Defined Strategy**, the pitch is held in a pitch array of dimension 80 x 100, giving a coverage of 22.5mm x 22mm per pitch element. Whilst working on pitch definition it was discovered that the cameras did not have lens spherical aberration correction. Further investigation showed that there was no parallax correction for the micro robots. This had minimal effect at the centre of the pitch, but in the goal area the error was about 42mm, or over half the micro robots body. I modified the Robot Soccer Engine to handle both these issues.

The Robot Soccer Engine image processor sub system scans the latest camera frame and detects the location of the coloured blobs in the image. The centre of each blob is calculated and then grouped according to the blob colours defined for each player. If the centres lie within a given distance and correspond to a known player colour configuration the three blob image is given a player number. The centre of the player is then calculated. The surface containing the charging port is considered to be the back of the micro robot and the small blob is always placed above it on the cap. The angle that the centre of the small blob makes to the centre of the robot in pitch axes determines the orientation of the micro robot. The whole image is scanned and all the home team robots are located and their orientations found within the previously defined pitch. The x,y positions of the robots are reported to the other subsystems in units of millimetre and the orientation is in degrees. If the Robot Soccer Engine is being used in a competitive match, the position of the ball is calculated as is the position of the opposing team. If a non Robot Soccer Engine cap system is used then only the mandatory team blob is available and this is taken to be the centre of the opposition robot.

With the Robot Soccer Engine application running on the computer, by turning the micro robot on a Bluetooth COMMs channel is allocated to the robot. For each active robot, the COM port number of the micro robot is entered against the player number. There is a facility on the Robot Soccer Engine to test that the allocation of the robot player is correct, by rotating the robot that the COM port is connected to.

The final feature of the Robot Soccer Engine is a facility to alter the motor characteristics of the micro robots by changing the internal PID controller parameters. The PID controller operates on the error between the Robot Soccer Engines demanded wheel velocity and the actual wheel velocity of the robot. The PID values can only be changed when the robots are not being controlled by the Robot Soccer Engine. Also the PID parameters are applied to all of the robots. This reduces the facility as the *Goalkeeper* is required to have a very fast response time when under attack, due to the small distances between it and the ball. This is usually achieved by having a very high Integral action parameter and a different Proportion value. Consequently the *Goalkeeper* robot has to have these values loaded separately.

The Robot Soccer Engine has a predefined hook to access a file called strategy.dll. The strategy.dll is generated from a C++ Strategy project file that contains strategy.cpp. The strategy.cpp file contains all the commands that the user has programmed to control the robots. Within the Strategy project file is the movement.cpp file that contains all the primitive commands that were programmed to move the robot. By using the primitives from the strategy.cpp the user can generate an infinite number of strategies using from one to five robots simultaneously.

However during the experiments various problems were encounted and solved. Initially the laboratory was equipped with venetian blinds. We found that the Robot Soccer Engine could not locate the robots and they went out of control, characterised by a high-speed spin. The problem was resolved by using blackout curtains to exclude external light, and installing strip lighting above the pitch to provide an even and consistent light source at the surface of the pitch. After this was done the problem did not manifest itself. After running the Robot Soccer Engine for a series of ten soccer games each lasting for 10 minutes, it was observed that the team which was using the MicroPix Firewire C640 camera won more games than the team using the MicroPix Firewire C1024 camera by

six to two with two drawn. The teams were swapped over in a second series of games again the team using the MicroPix Firewire C640 camera won more games, by seven to one with two drawn. The conclusion that was drawn is that the camera with the faster frame rate enabled a better performance to be obtained from the team it was controlling. So it was decided that the MicroPix Firewire C640 camera would be used in preference to the MicroPix Firewire C1024 camera when conducting experiments and competitions involving a single team. Over the series of games the robots sustained a high level of damage due to collisions. Tyres were thrown off or got jammed between the wheel and the chassis. The main axles became loose and wheels fell off. This lead to improvements being made in the design and manufacture process, such as stub axles being introduced and tighter tyres being used. There was no damage sustained by the robot electronics, however some of the battery connections became loose and turned off one of the motors so the robot went round in a circle.

3.4 Fuzzy Inference System

The Fuzzy Inference System used in this thesis was originally sourced from SourceForge as jFuzzyLogic. This is a Type-1 fuzzy logic system written in Java. I then enhanced and upgraded the source to incorporate a Type-2 fuzzy logic system in the same application. Now the Fuzzy Inference System application can support either a type-1 or type-2 fuzzy logic. However type-1 and type-2 fuzzy logics cannot be run together. Consistent with the theory of type-2 fuzzy logic when all uncertainty is removed then the type-2 fuzzy logic becomes a type-1 fuzzy logic. So by making the type-2 membership functions have no degree of uncertainty, that is making their upper and lower bounds equal, it is possible to coerce the Fuzzy Inference System to support both fuzzy logic systems.

Although there are type-2 Fuzzy Inference Systems already available there are none freely available written in Java, which is one of the major programming languages. By writing a type-2 Fuzzy Inference System I was able to learn and understand how a Fuzzy Inference System worked, having never previously worked on one. By selecting jFuzzy-Logic as a base I was confident that the code had been tested by many different people and projects, and could be considered stable. However during during testing and code walks

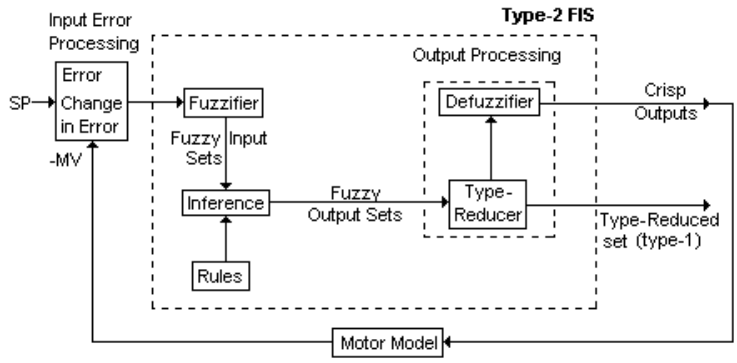


Figure 3.4: Type-2 FLC Schema

instances a few instances of incorrect and inefficient code were discovered and corrected.

The schematic of the Fuzzy Logic Controller using the Fuzzy Inference System is given in Figure 3.4 in the type-2 configuration. The figure shows the Fuzzy Inference System in a simulation control configuration with a Motor Model, which is discussed later. The two crisp inputs to the Type-2 FLS are firstly the error between the motor speed demand set point(SP) and the actual motor speed(MV) - $(Error = SP - MV)$, and secondly the change in error - $(ChangeinError = error_n - error_{n-1})$. The Fuzzy Inference System itself is made up from five subsystems. The purpose of each subsystem is as follows.

The Fuzzifier The fuzzifier is the first subsystem and handles the crisp Error and Change in Error inputs to the Fuzzy Inference System. The inputs are mapped into fuzzy input sets, using the systems membership functions, for further processing.

Rules The rules subsystem holds the rules that have been supplied to the system. The fuzzy input sets are weighted according to the degree of support for each rule.

Inference The Inference subsystem is where all the main processing of the fuzzy input sets and the rules are processed.

Type Reducer The type reducer subsystem processes the type-2 inference results into

type-1 sets.

Defuzzifier The defuzzifier subsystem produces the final output of the system by converting the output type-1 fuzzy set into one or more crisp values, using one of a variety of defuzzification methods.

The crisp output of the defuzzifier subsystem is a fractional percentage and is input to the DC Motor Model where it is scaled according to the range of the Motor Model. After processing the motor model demand, the motor output measured variable(MV) is scaled to the FLC input range and the next set of inputs can then be calculated. The type-2 fuzzy logic controller is designed to match the motor output with the system demand such that the Error and Change in Error variables are both zero.

3.4.1 DC Motor Model Setup

A schematic of the DC motor model is given in Figure 3.5. The state-space equations of the DC motor model are as follows:-

$$\frac{d}{dt}\begin{bmatrix} i \\ \omega \end{bmatrix} = \mathbf{M}\begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_{app}(t) \tag{3.1}$$

where:

$$\mathbf{M} = \begin{bmatrix} \frac{-R}{L} & \frac{-K_b}{L} \\ \frac{K_m}{J} & \frac{-K_f}{J} \end{bmatrix}$$

and

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} . \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} . V_{app}(t) \tag{3.2}$$

The model uses the following nominal values for all the experiments:- R = 0.2Ω, L = 0.5 H, K_m = 0.015 TorqueConstant, K_b = 0.015 EMFConstant, K_f = 0.2 Nms, J = 0.01 Kgm^2s^{-2}.

3.5 Independent PID Controller

A PID controller was used to control the DC motor model and to provide a reference. The PID controller used in the simulation takes the independent form without integral wind-up to avoid saturation:

$$u(t) = K_p\, e(t) + K_i \int_0^T e(t)\, dt + K_d \frac{de}{dt}, \tag{3.3}$$

where u is the control signal or process output and e is the process error. The controller parameters are the proportional gain K_p, integral gain K_i and derivative gain K_d.
The range of the controller was -2 to 2 V, the same as the DC motor model input voltage. The process output of the model was fed back into the PID controller as the process variable and subtracted from the set point to generate the error term. In a series of benchmark experiments to determine the control constants, the PID controller was used to control the motor model. The following values were found to give the best control response, $K_p = 100$, $K_i = 200$ and $K_d = 10$.

3.6 The Strategies

Various strategies have been investigated and developed. They are written in C++ and were tested on the FIRA simulator to eliminate errors such as direction of play.

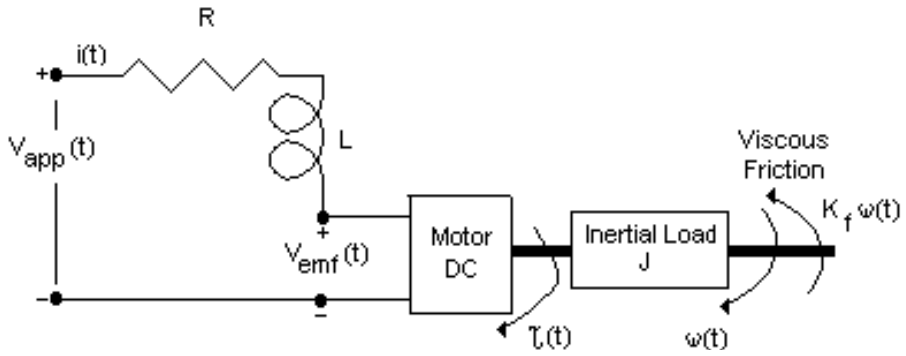


Figure 3.5: DC-Motor Model

3.6.1 Case Based Strategy

The case based strategy uses the position of the ball, the position of the robots and the position of a target to generate a set of vectors. The vectors will be used to direct the robot through the ball towards the target. This is repeated on a frame by frame basis. The robots are arranged in a 1-2-1 formation with one *Defender*, two midfielders (*Left* and *Right*), and one *Attacker*. The *Defender* always plays in his own half of the pitch, and the *Attacker* only plays in the oppositions half of the pitch. *Left* plays on the left side of the pitch, and *Right* on the right side. Both these players can play in either half. Attacking is considered to be chasing the ball towards the opponents goal, and defending is when chasing the ball the other way. If the players are moving towards their own goal, they try to run round the ball, so as to get between the ball and the goal, without hitting the ball towards their own goal. If that is not possible, then the players attempt to drive the ball into the corners, and then start attacking. Tackling is a behaviour whereby a player moves towards the ball which is effectively controlled by an opponent, attempts to intercept the ball and start attacking. Players shoot when they are inside the opponents penalty area and within one ball diameter of the ball. The player spins to effect the shot. Free kicks and penalties use a move to the ball and a spin to hit it. The *Goalkeeper* attempts to move and stay in line with the ball and hit it away from the goal when ever possible. The *Goalkeeper* stays within the penalty area.

3.7 Problems Discovered During Experimentation

3.7.1 Lighting

Initially the laboratory was equipped with venetian blinds. It was found that the RSE could not locate the robots and they went out of control, characterised by a high-speed spin. After installing strip lighting and blackout curtains the problem has not re-occurred.

3.7.2 Camera Frame Speeds

After setting up the cameras for focus and aperture in the RSE, a series of games were played. Analysing the results showed that there was a problem with the high resolution

camera running at 15 fps. The team it was controlling was not responding quickly enough and was unable to win any games. This requires further investigation to determine the cause.

3.7.3 Physical Damage

Over the series of games the robots sustained a high level of damage due to collisions. Tyres were thrown off or got jammed between the wheel and the chassis. The main axles became loose and wheels fell off. This lead to improvements being made in the design and manufacture process, such as stub axles being introduced, however our robots were not updated. There was no damage sustained by the robot electronics.

3.7.4 Robot Control Parameters

Each robot has a basic Proportional Integral Differential (PID) control algorithm. The PID is given the distance between the robot and the ball as an error value, and moves the robot to reduce the distance to zero by altering how the motors are driven. The RSE only provides one set of PID parameters for all the robots. However we require the *Goalkeeper* to have a very fast response time when under attack, due to the small distances between it and the ball. This is usually achieved by having a very high Integral action parameter and a different Proportion value. Consequently the *Goalkeeper* robot has to have these values loaded separately.

3.7.5 Impact

The result of these problems had a direct bearing on the experimental runs that are described in Chapter 6. The alternative camera runs at 30 fps. However the calculation of the position of the robot is not as accurate due to the image being smudged. The problem with physical damage proved to be the most critical. Lots of individual runs of the robots had to be ignored due to their effects when detected.

3.8 Robot Framework and PID Controller

The robot has an ATMega64 processor with 64K of flash memory and 2K of EEPROM running a compiled 'C' program. There is a slave ATMega8 processor which provides motor input/output and power control. There are two UARTs, one external and one internal providing communication facilities. The software is a single executable program, logically composed of four processing blocks

Initialization Initialization sets up the robot to a point where it can be commanded.

Command Support Command Support handles the external commands received by the robot.

Communication Communication sets up and handles interrupts from the wireless module (UART0) and the motor IO module (UART1).

Motor Control Motor Control holds the control algorithm for the motors.

3.8.1 Initialization

On power-up of the robot, the current EEPROM held data values are loaded and the downloaded executable is run. The initialization process is as follows:

- Set up Ports C, E and G for the UARTS
- Initialise robot variables from values held in the EEPROM
- Set up the command table
- Perform default communication port setup
- Add user commands
- Initialise the motors
- Enable interrupts
- Run the program main in a forever loop

At the end of the initialization the program is in the main loop checking for a command to execute.

3.8.2 Command Support

This provides the instructions that the robot has to perform. There are over thirty user commands supported, for which only four of are interest in this context.
The routine `dispatchCommand` is called in the main loop, when `commandBufferMode` is set to `COMMAND_READY` by Radio Comms. `DispatchCommand` matches the received command against the list of commands, and when a match is found, executes the corresponding command routine.

- SetSpeed command sets the required speed of the robot. This is the command received from the Robot Soccer Engine to move the robot. It is executed by entering the command `[-{leftByte}{rightByte}]`. The routine `setSpeed` scales the each byte value by multiplying by param `V_ByteScaleSpeed` (=1024) and shifting right by 8, e.g. -128 x $1024/256 = -512$. The result is stored in `nReqSpeedLeft` and `nReqSpeedRight` respectively.

- Stop command sets the speed of the robot to zero. It is executed by entering the command [s].

- Test command runs the user defined routine test in the downloaded executable. It is executed by entering the command [t].

- Param command allows the viewing and setting of parameters. To view a parameter enter [.id], to set a parameter enter [.id=nnn] and to show all parameters enter [.].

3.8.3 Communication

This provides the mechanism to transfer commands and data between the modules of the robot under interrupt. The two uarts on the robot are controlled by two communication sub processes, radio comms (UART0) and IO comms (UART1).

Radio Comms

External commands and replies are processed through the radio comms. The receive signal is handled by `processReceiveByte0` and the transmit signal by `processTransmitByte0`. The routine `processReceiveByte0` stores the incoming command in `command-`

Buffer, and when completed sets the commandBufferMode to COMMAND_READY for execution in the main loop by the Command Support process. The routine processTransmitByte0 sends the contents of sendBuffer to the radio comms port. This mechanism as well as altering the required speed, allows parameter changes to be made whilst the robot is running.

IO Comms

This subprocess controls the exchange of data between the robot program and the motor IO under interrupt. There are two data exchanges to be handled. Firstly, the request for left and right motor clix data from motor IO, and secondly the transmission of left and right motor power values to motor IO.

The motor IO sends a timeout interrupt every millisecond. The interrupt handler for the timeout counts down the parameter variable nControlRateTicks to zero from param V_PidTickRate (=5). When the countdown reaches zero, the IoState of the robot is changed from IOSTATE_IDLE to IOSTATE_GetLeftHi and UDR1 is set to the command char IOCHAR_GETPOS.

This initiates the exchange of left and right motor clix data through the receive signal routine processReceiveByte1. On receipt of each motor clix component the IoState is set to the next expected byte. A confirm is set in UDR1 and sent to the motor IO by the transmit signal routine processTransmitByte1. This causes the motor IO to signal the next byte is ready to be handled by processReceiveByte1. After the last confirm is sent, the IoState of the robot is set to IOSTATE_GotNewPos by the transmit signal routine processTransmitByte1, so that the motor control algorithm can be actioned. The current motor clix values are stored in the variables nNewClixLeft and nNewClixRight.

Motor power data transfer is initiated by motor control, triggering the motor IO to call the receive signal routine processReceiveByte1. The left then right motor power values are loaded in UDR1 for transfer. The IoState of the robot is set back to IOSTATE_Idle when completed.

3.8.4 Motor Control

The motor control process provides the control algorithm for the robot. The routine `controlMotors` is called in the main loop and is actioned when `IoState` is set to `IOSTATE_GotNewPos` by the IO Comms sub process. The `IoState` is set to `IOSTATE_Idle` and the motor clix are validated. The control algorithm is then called to calculate the required motor power for each motor.

Finally the transfer of motor power data is initiated. The `IoState` of the robot is set to `IOSTATE_SendLeft` and `UDR1` loaded with the command char `IOCHAR_SETPWM`, the receive is handled by IO comms.

3.8.5 Control Algorithm

The suffixes `Left` and `Right` are implied in the variable names. The control algorithm currently implemented is a three term PID controller.

The algorithm starts by calculating the rate of change of the motor clix, `iClixRate` as `nNewClix - nOldClix`. If this rate is below a threshold of `RATE_MAX` then `iClixRate` is stored in `nClixRate` otherwise the previous `nClixRate` is used in the control algorithm. This is important since the motors can report spurious clix values, and there is no indicator if the Motor Clix counter goes through the 65535 integer boundary and resets. Using the previous value minimises the controller bumping. The `nNewClix` are then stored in `nOldClix` for next time. The `nTargetSpeed` is then loaded. This can be the scaled demand `nReqSpeed` from the `setSpeed` command, or some function of `nReqSpeed`.

Control Algorithm Calculation Function Call

For each motor, the function `calcSpeed_nPwm` is called with input parameters `nTargetSpeed` and `nClixRate`, and output parameters `OldSpeed` and `ErrorDist`. The function returns a motor power value, `nPwm`.

Speed and Motor Power Calculations

The following calculations are performed to calculate the power to be put onto the motor

- Current speed of the robot

```
nSpeed = nClixRate * param(V_PidSpeedgain(=4))
```

- Change in speed — negative for increasing speed

  ```
  nSpeedChange = OldSpeed - nSpeed
  ```

- Error in speed — positive for too slow

  ```
  nSpeedError = nTargetSpeed - nSpeed
  ```

- Error in distance - positive when behind the target

  ```
  nErrorDist = ErrorDist + nSpeedError
  ```

  In order to prevent integral windup `nErrorDistance` is limited between $\pm$ `param V_PidMaxIerr(=2048)`, which is half a wheel turn. `nSpeed` and `nErrorDistance` are stored in `OldSpeed` and `ErrorDist` for use next time.

- Open loop power value

  ```
  nOpenLoopCalc = param(V_PidOpOffs(=0)) +
    param(V_PidOpGain(=3)) * nTargetSpeed
  ```

- Speed error correction

  ```
  nSpeedErrorCalc = param( V_PidP) * nSpeedError.
  ```

- Accumulative error correction

  ```
  nErrorDistCalc = param(V_PidI) * nErrorDist.
  ```

- Differential correction

  ```
  nSpeedChangeCalc = param(V_PidD) * nSpeedChange.
  ```

- Total calculated correction

  ```
  nCalc = nOpenLoopCalc + nSpeedErrorCalc +
    nErrorDistCalc + nSpeedChangeCalc.
  ```

  The correction is then shifted right by `param(V_PidDownshift(=7))`, limited to lie between -127 and 127 and then stored in the return variable `nPwm`.

3.9 Summary

In this chapter the environment that the real world experiments was discussed. Descriptions of the pitch, cameras and the robot used in the real world experiments was provided. The software and hardware required to operate the robot was reviewed. The mechanism to identify and separate the robots using coloured caps was given. A description of how the robot operated internally was given. Details of the real world problems encountered in controlling and operating the robots was provided together with the lack of spherical aberration correction and the corrective action applied. An overview of the Fuzzy Logic system used to develop the type-1 and dual surface and average type-2 controllers was described. The DC motor model used in the simulation was defined and subsequently coded.

Chapter 4

Development of Baseline PID Controller

4.1 Introduction

This chapter is concerned with the development of a Proportional, Integral and Differential controller, commonly known as a PID controller. The controller used is the independent PID controller as described in Chapter 3. The purpose of developing the controller was to establish a baseline controller against which the type-1 and type-2 fuzzy logic controllers could be compared. In order to remove as much variation as possible from the investigation the same robot was used throughout the following experiments. A series of experiments were carried out with the objective of tuning the independent PID controller. The tuning experiments were to establish a set of P, I and D parameters which would allow the robot to perform across the range of wheel speed demand inputs.

4.2 No Load Encoder Clix to Power Curve Experiment

The aim of the experiment was to measure the micro robot motors response to a range of Pulse Width Modulation (PWM) demand values that spanned the input to the motors. The motors were run without a load being applied to them. By not loading the motors then effects like body friction and wheel slippage do not occur, and so do not distort the result. The purpose for doing this was to discover if the pair of motors on the robot were different, or that they could be considered to be similar for operating purposes.

	LEFT	MOTOR			RIGHT	MOTOR			
Pwm	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Diff L-R Mean
-110	-416.51	1.78	22	-5.41	-400.35	1.86	21	-7.69	-16.17
110	411.11	1.36	21		392.65	1.83	20		18.46
-70	-399.95	1.88	40	-3.06	-367.72	2.10	37	-2.70	-32.23
70	396.89	1.36	41		365.02	2.16	38		31.87
-30	-314.64	4.82	114	4.22	-245.38	3.13	99	3.66	-69.27
30	318.86	6.36	117		249.04	3.59	100		69.82

Table 4.1: No Load Mean Clix for Selected Motor PWM Demands

4.2.1 Setup

The micro robot was set up using the parameters in the Robot Definition File given in Appendix Table B.11. The robot was operated under the control of the Microsoft Windows hyper terminal utility. The robot was placed on its side so that the wheels had no load placed on them. The command [t] was transmitted to the robot and the previously loaded parameters were action by the robot controller. When the sample was completed the experimental run was halted under program control and the memory transmitted to the hyper terminal utility of the controlling computer. The Capture Text option of the hyper terminal was used to store the clix data information that was transmitted by the robot.

The robot operates between -127 and 127 PWM. The test was started at an initial power demand level PWM of -120. The PWM values were directly loaded into the interrupt transmission variables to be sent to the motor controller. This acted as a step demand on the motors. At the demanded power level the number of wheel encoder interrupts per control loop call was recorded for both wheels for 250 control loop interrupts. A control loop interrupt, known as a tick, occurs every 5ms (”pT” - V_PIDTickrate) giving a total sample time period of one second. After 250 samples had been collected the demand PWM was set to zero and the data transmitted by the robot to the hyper terminal and stored in a text file for further processing.

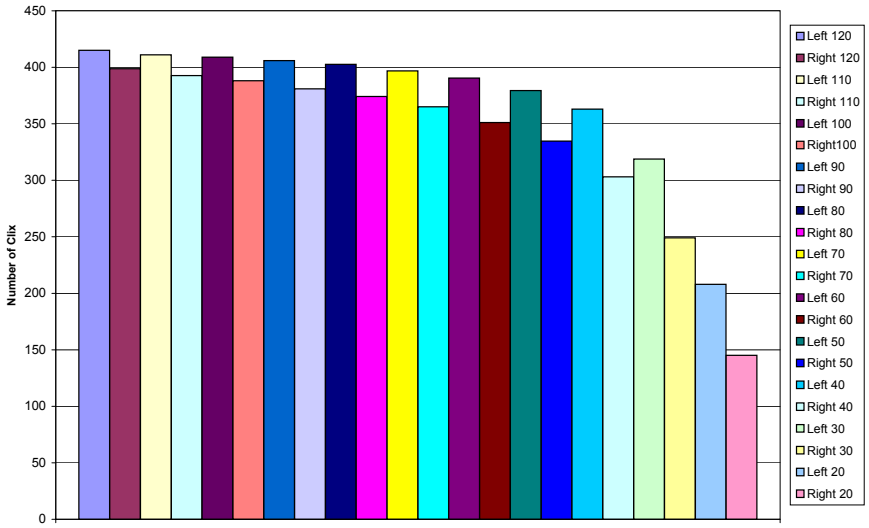
Restarting the test using the command [t], incremented the PWM level by 10. The tests were run for PWM level from -120 to 120, with the levels between -10 to 10 inclusive being skipped. In pre-testing it was found that the power levels were too low to cause the robot to move, and so were considered unnecessary to be tested in the no-load experiment.

	LEFT	MOTOR			RIGHT	MOTOR			
Pwm	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Diff L-R Mean
-110	-326.48	50.19	90	-1.56	-193.87	25.93	8	32.40	-132.60
110	324.91	33.54	126		226.27	38.44	38		98.64
-70	-254.12	29.99	159	-19.26	-194.92	30.19	93	43.01	-59.20
70	234.86	21.15	162		237.93	19.77	155		-3.07
-30	-56.11	7.35	172	-2.18	-62.12	8.39	177	-5.52	6.01
30	53.93	7.21	174		56.59	7.16	169		-2.66

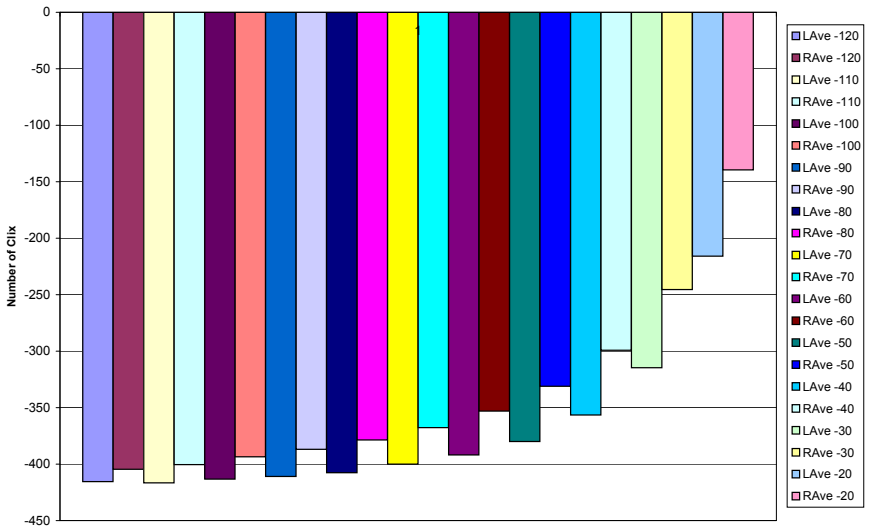
Table 4.2: Loaded Mean Clix for Selected Motor PWM Demands

4.2.2 Discussion

The results of the No-Load PWM motor response for PWM values of +-110, +-70 and +-30 are given in Table 4.1. The results for all the PWM values are given in the Appendix Table B.1. In Table 4.1 the mean and SD values reported are for the plateau response values which are calculated for the last 100 data points of the sample. The mean clix for the left motor are greater than those of the right motor for the same PWM values. This bias between motors is not unexpected as it would be very fortunate to get matched motors, and is not considered to be critical for the performance of the robot. The SDs are also small showing that the clix values are close to each other. 90% Clix# is the number of 5ms ticks that the response requires to achieve 90% of the mean value. For the high values PWMs the ticks to 90% are very close. The right motor reaches 90% of the mean value slightly faster than the left motor, however the mean value is lower in the right motor so this is to be expected. Diff -+ Mean is the difference between the means for the negative and positive PWMs for each motor. The table shows that the difference within the motors is less than 8 clix. This shows that the motors do not have a bias within themselves as to the polarity of the PWM input. The values in Diff L-R Mean report the difference between the means of the left and right motor for the same PWM value. The table shows that the difference between the motors is nearly the same for each PWM value. It is noted that the higher the PWM values the difference between the means decreases as the motors approach saturation. Histograms of the right and left motor average clix values are given in Figure 4.1(a) for the positive PWM values, and in Figure 4.1(b) for the negative PWM values. These show that for values of PWM greater than $|50|$ the difference between each PWM level is less than 10 clix and that the power curve is asymptotic as the PWM



(a) Positive PWM



(b) Negative PWM

Figure 4.1: Histograms of No Load PWM Motor Average Clixs

increases. In the Average Clix histograms it can be seen that the left motor consistently generates more clix for the same PWM value than the right motor when no load is applied. This suggests that when the micro robot is loaded that it might not travel in a straight line due to the difference. However since 1 clix is equal to 0.04mm linearly the difference is not large and is not expected to be significant. The histograms also show that the overall PWM demand to average clix motor response to be simular for both motors.

Statistical Analysis of No Load Data

A statistical analysis was carried out on the all the no load data given in Appendix Table B.1 and the following results were obtained.

Correlation Coeff between the two motors 0.997

Correlation Coeff between left -ve right -ve 0.992

Correlation Coeff between left +ve right +ve 0.988

An hypothesis test was carried out using a two tailed paired t-test. Null Hypothesis : There is no significant difference between the mean values of the two motors when there is no load on them. Alternative Hypothesis : These is a significant difference between the two means.

The t-test p value = 0.98 at 21 degrees of freedom, this is greater than 0.05 and is not significant. Hence the Null Hypothesis cannot be rejected.

4.2.3 Conclusion

In this experiment the number of motor encoder clix per controller tick were captured for both motors of the robot. Analysis of the results showed that there was no significant difference between the mean values of clixs, the motor response, for the range of PWMs applied to the motors. This shows that the micro robots motors can be considered to be the same in terms of performance and characteristics when there is no load on the motors.

4.3 With Load Encoder Clix to Power Curve Experiment

Following on from the experiment to investigate motor performance without a load on the motors, the aim of this experiment was to measure the motor response to a range of PWM

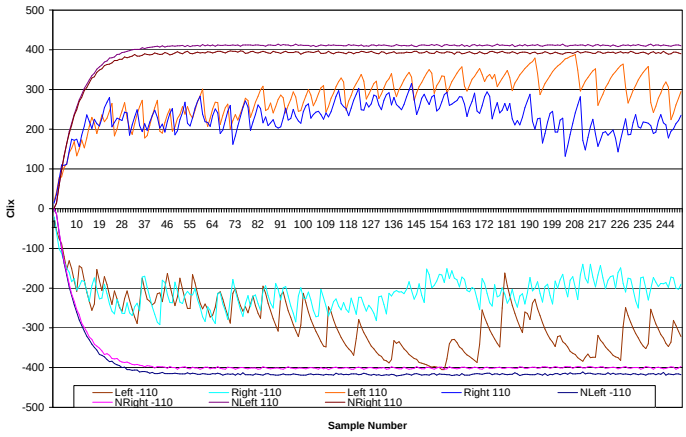
inputs with a load on the motors. Having shown that the motors are statistically the same without a load on them, the expectation is that they will perform equally when a load is applied to them. However the effects of friction and wheel slippage will probably cause a wider variation in the results.

4.3.1 Setup

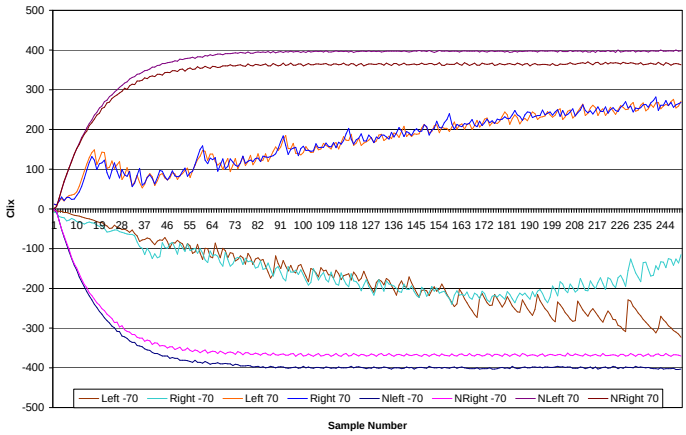
The robot was reset by turning the power off and on. All parameters are held in eeprom and were reloaded on power up. The robot was positioned with the wheels on the robot soccer pitch. At the end of each PWM run the robot was returned to the same starting position on the robot soccer pitch. The runs were repeated for each PWM level, as per the previous experiment, and the data captured for analysis. Prior to starting the experiment the batteries were recharged to eliminate voltage variations, which might bias the results of the experiments. The only difference in the two experiments was that the motors were loaded by driving the robot.

4.3.2 Discussion

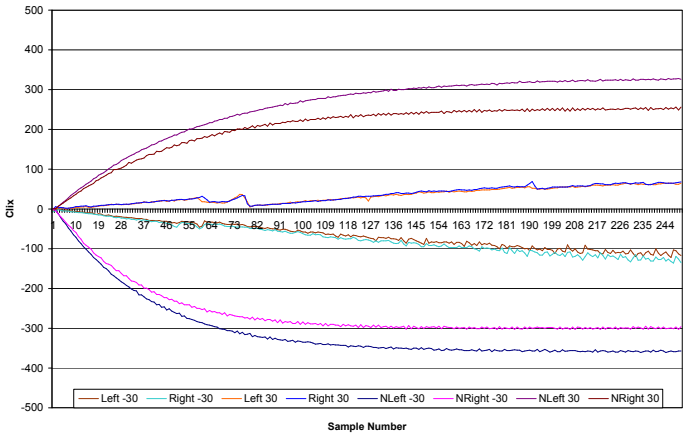
The results of the Load PWM motor response for PWM values of +-110, +-70 and +-30 are given in Table 4.2. The results for all the PWM values are given in the Appendix Table B. In Table 4.2 the mean clix value at PWM values of +-110 and -70 are greater for the left motor than the right motor. For the other three PWM values the difference in the means are very close. The difference between the means for each pair of PWM values in the left motor is much lower than the corresponding values for the right motor. As expected the SDs for each PWM value are high compared to the corresponding no-load values. Expect for the -110 PWM value, the SDs of each pair of PWMs are close. The right motor reaches its 90% Clix# much faster than the left motor for the +-110 and -70 PWM values, otherwise they are close. Histograms of the right and left motor average clix values are given in Figure 4.3(a) for the positive PWM values, and in Figure 4.3(b) for the negative PWM values. Again the asymptotic shape of the power curve is seen as the PWMs are increased. However at the higher PWM values especially in the right motor it breaks down.



(a) PWM 110 Motors Response

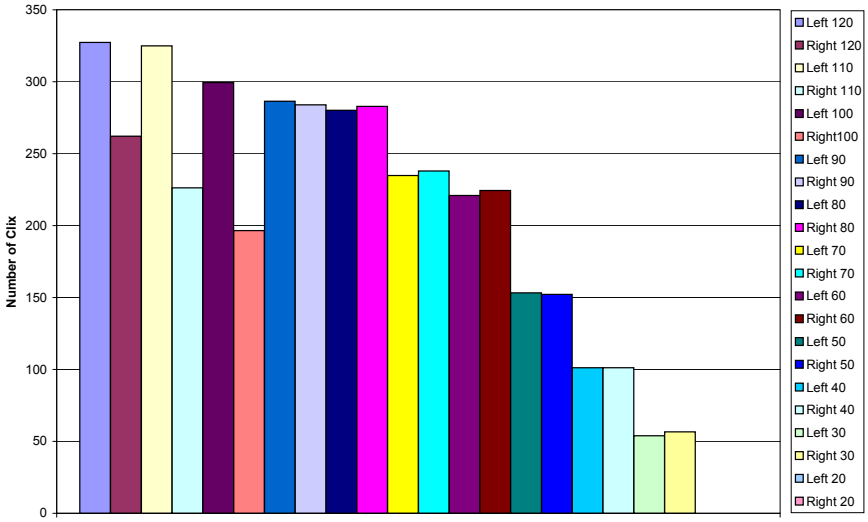


(b) PWM 70 Motors Response

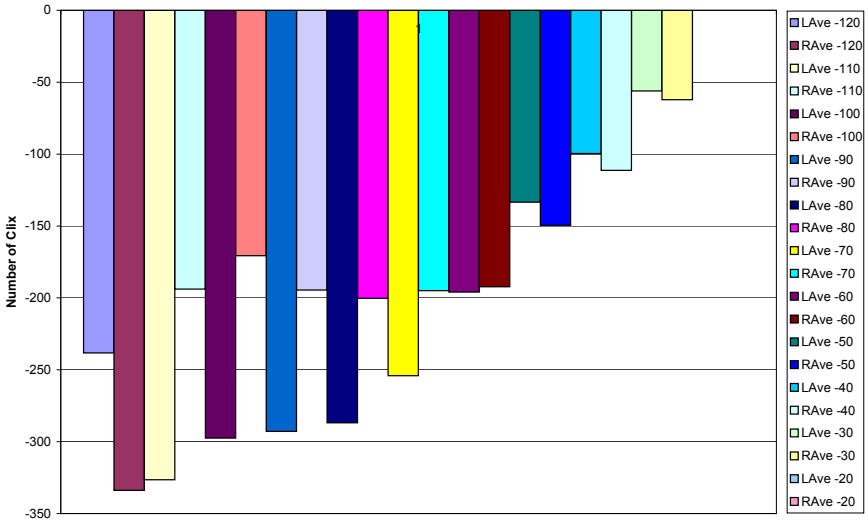


(c) PWM 30 Motors Response

Figure 4.2: Comparisons of Load and No Load Motor Responses



(a) Positive PWM



(b) Negative PWM

Figure 4.3: Histograms of Loaded PWM Motor Average Clixs

In Figure 4.2 the clix traces for each pair of PWM values are plotted. It is seen that in Figure 4.2(a) for the PWM 110 response that the left motor response trace approaches the no-load traces and is varying by large amounts as reflected in the table. The most probable reason for this behaviour is that the left wheel is slipping at these values and so the encoders are generating high clix values. The right motor by comparison maintains a much more stable trace profile. In Figure 4.2(b) for the PWM 70 response the motors are able to drive the robot much more smoothly and do not approach saturation. However at about sample number 154 the left motor starts to deviate from the right motor at PWM value of -70. The left motor clix increases as the wheel slips and the right motor clix decrease as the weight of the robot is loaded onto the right wheel. For PWM values of +70 the pair of motors drive the robot equally. Finally in Figure 4.2(c) the pair of motors drive the robot equally for both PWM values of +-30.

Statistical Analysis of Loaded Data

A statistical analysis was carried out on all the loaded data given in Appendix Table B and the following results were obtained.

Correlation Coeff between the two motors 0.969

Correlation Coeff between left -ve and right -ve motor 0.618

Correlation Coeff between left +ve and right +ve motor 0.889

This shows that the two motors are highly matched however there when negative PWM is applied the correlation between the two motors is not so strong.

An hypothesis test was carried out on the left and right motor clix data using a two tailed paired t-test. Null Hypothesis : There is no significant difference between the mean values of the two motors when there is a load on them. Alternative Hypothesis : These is a significant difference between the two means. The t-test p value = 0.698 at 21 degrees of freedom, this is greater than 0.05 and is not significant. Hence the Null Hypothesis cannot be rejected.

4.3.3 Conclusion

Analysis of the micro robot motor clix data shows that under load there is no significant difference between the motors. This confirms the results obtained in the previous

experiment that the motors can be considered to be the same in terms of performance and characteristics. Although the experiment was only run for one second, there was still enough time for the wheels to slip as the power was applied.

4.4 Robot PID Tuning Experiment

The software that is installed and run in the robot is collectively known as the robot controller. This is made up of various subsystems including communications, motor control, command interpretation, internal interrupt handling and processing. The motor controller used in this experiment is a Proportional, Integral and Differential (PID) controller that uses the independent configuration, also see Chapter 3. The purpose of a PID controller is to reduce the error between the measured variable (MV) of the process and the setpoint (SP) or target of the process to zero. So if the required speed of a robot has been set then the PID controller will act so that the measured speed of the robot equals the required speed. Three terms make up the PID controller. The first term is the Proportional term which is the error between the measured variable and the setpoint multiplied by the Proportional parameter (P). The second term is the Integral term which is the sum of errors multiplied by the Integral parameter (I). The third term is the Differential term which is the change in error between this measurement and the previous measurement multiplied by the Differential parameter (D). By altering the values of P, I and D the characteristics of the PID controller are altered. Having investigated the motors and shown that there is no significant difference between them, then it is considered valid to use the same PID controller on both motors. The purpose of these experiments was to determine the best set of P,I and D parameters, out of those tested, to use in the PID motor controller. However it is not intended to find the optimal set of parameters of the PID controller.

Three sets of experiments were carried out. The first experiment was to determine the working range for the proportional parameter pP. Then having obtained a range the second experiment investigated varying the integral parameter pI, within each value of pP. The third experiment was to vary the value of the differential parameter pD, within each value of pI, within each value of pP. As stated before the intention was not to develop an optimal PID controller for the micro robot. It was to obtain a set of PID parameters

that would operated across the range of velocity inputs to the robot, giving an acceptable performance to the user.

4.4.1 Setup

As previously described in the load and unload experiments the robot was controlled using the hyper terminal to start the runs. All the runs took place on the robot soccer pitch. The micro robot was set up according to the standard settings. Throughout the runs the only parameters that were varied were the three PID parameters, pP, pI and pD. In experiment 1, pP was varied from 5 to 150 in steps of 5 with pI = 0 and pD = 0. The wheel velocity demand was set to -120 throughout all three experiments. This is the largest allowed value allowed by the Robot Soccer Engine, although not the largest wheel velocity demand value, which is -127. By tuning the PID controller to be able to control the robot at high demands, then the performance at lower demands is deemed to be acceptable. The variables reported back by the robot to the hyper terminal were the PWM value sent to the motors and the number of clix per 5ms tick, with two hundred readings per run being reported.

4.4.2 Discussion

The results for the Proportional set of tests are given in Appendix Table B.3. The PWM and Clix averages and SDs were calculated for the last one hundred samples reported. If the parameter sets were going to be acceptable, then the PWM and Clix should be stable in this sample range. Ramp90 is the clix value that is 90% of the clix average value, and 90%# is the number of sample points required to reach Ramp90 for the controller under test. The Ramp Rate is the Ramp90 value divided by 90%#. The selection criteria are the sets of parameters that give the highest ramp rates. It can be seen that the Proportional only controller was operating at a ramp rate of about 41 to 42 for values of pP from 35 to 140. The robot started to suffer instability above pP = 140. Below pP = 35 the ramp rate was considered to be too low. From this result the range of the Proportional parameter values in experiment 2 were set between 40 and 120. The value of pP was increased in steps of 10. Within each pP the value of pI was varied in five equal steps from 10% to

50% of the pP value. The reason for selecting these Proportional parameter values was that they covered most of the best rate values and were well away from the instability above pP=140.

The results for the experiment 2 set of tests are given in Appendix Table B.4. It can be seen that the PI controller was operating with a ramp rate about 43.2 for most of the first two pI steps within the pP values for pP ranging from 40 to 100. The increased ramp rate showed that the PI controller was an improvement on the Proportional only controller.

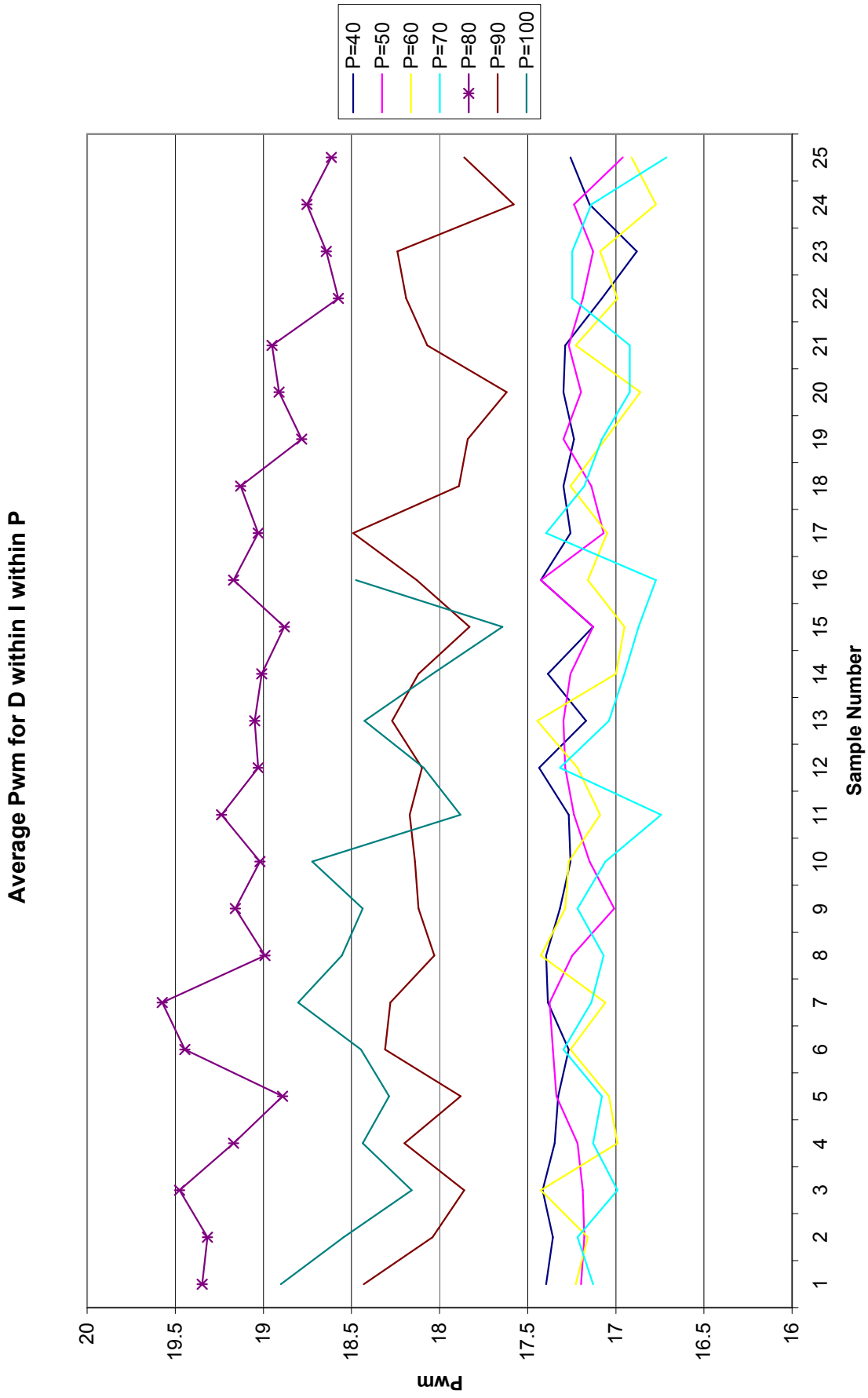


Figure 4.4: Pwm Overview for PID Parameters

In experiment 3 the Differential parameter(D) was introduced to create the PID controller. For the experiment 3 tests the range of pP values was set from 40 to 100 with pI being varied from 10% to 50% of pP within pP, and pD was varied within pI from 10% to 50% of pP. By basing the pI and pD values on pP, it allowed the possibility of pD being greater than pI. The results of the experiment 3 set of tests are given in Appendix Tables B.5 to B.10. The ramp rates fall into two groups, less than or equal to pP=70 with values about 43 clix per tick, and the rest with values about 54 clix per tick. In Figure 4.4 a graph of the PWM averages for each PID controller is given. The controllers are linked by their pP values and are in sets of 5 for pD within pI. e.g. for the pP=80 trace, sample 3 is pI=8, pD=24, sample 17 is pI=32, pD=16. It is seen that the controllers with the highest PWM averages are the set with pP=80. No other set exceeds these PWM average values. The highest individual controller has parameters of pP=80, pI=16, pD=16 with a PWM average value of 19.57 and a ramp rate of 54.00 to 2d.p.

4.4.3 Conclusion

These experiments to tune a micro robot PID controller demonstrated that the best controller was given by PID parameter values of pP=80, pI=16 and pD=16. It was identified that for values of pP above 140 that the robot was becoming unstable. By using the maximum allowed wheel velocity demand of -120 it was shown that the controller can handle that level of input and remain stable. A safe and stable PID controller has been developed and can be used as the baseline PID controller.

4.5 Summary

This chapter has presented a series of experiments designed to develop and assess a baseline PID controller. In the No Load Encoder Clix to Power Curve Experiment it was shown that there was no significant difference between the motors on the test micro robot. Using the same robot in the With Load Encoder Clix to Power Curve Experiment again it was shown that there was no significant difference between the motors. These results showed that the motors of the micro robot could be considered to be effectively the same in terms of performance and characteristics. This meant that the robots could be

considered interchangeable if the experimental robot failed.

In the Robot PID Tuning Experiment the development of a baseline PID controller was described and demonstrated. The reason for performing these experiments was to have a definitive set of PID parameters which have been shown to give the best controller for the micro robots. This baseline controller was also shown to be safe and stable. Having developed this PID controller, then any comparisons made against it when using fuzzy logic controllers will be valid and not subject to doubt or discussion of the validity of the PID controller's performance.

The next chapter investigates and evaluates membership functions for type-1 and type-2 fuzzy logic controllers in order to establish a best performing set for the robots.

Chapter 5

Evaluation of Membership Functions for Type-1 and Type-2 Fuzzy Logic Controllers

5.1 Introduction

Having investigated in detail the selection and tuning of Proportional Integral and Differential controllers in the previous chapter, the issue was how to implement a Fuzzy Logic Controller. One of the aims of this thesis is to work in the real world as far as possible. However being pragmatic it was decided that the best approach to developing a Fuzzy Logic Controller was to use a simulation of the micro robot motor. Due to the limited resources of the micro robots, it would not be possible to properly distinguish the differences between controllers that are broadly similar. Simulation provides the capabilities and facilities to make the necessary distinctions. However simulation is just a tool which provides modelling facilities and contributes to the decision making process. The researcher has to keep in mind the end goal of implementing the Fuzzy Logic Controllers on real robots which are resource constrained. This chapter describes the process which was employed to develop Type-1 and Type-2 Fuzzy Logic Controllers. The main thrust of the process was to reduce the number of Fuzzy Logic rules in the rule base without seriously compromising the effectiveness of the controller. The Fuzzy Inference System with the DC Motor model described in Chapter Three was used in all the simulations.

Control Change	Error Change						
	nb	nm	ns	zr	ps	pm	pb
nb	nb	nb	nb	nb	nm	ns	zr
nm	nb	nb	nb	nm	ns	zr	ps
ns	nb	nb	nm	ns	zr	ps	pm
Error zr	nb	nm	ns	zr	ps	pm	pb
ps	nm	ns	zr	ps	pm	pb	pb
pm	ns	zr	ps	pm	pb	pb	pb
pb	zr	ps	pm	pb	pb	pb	pb

Where:- nb - negative big; nm - negative medium; ns - negative small; ze - zero pb - positive big; pm - positive medium; ps - positive small;

Table 5.1: Seven Term Fuzzy Logic Controller

Fuzzy Logic Controllers of Type-1 and Type-2 were simulated and compared to each other and to a PID controller, described in Chapter 3, which was used as a control. All the experiments carried out are described and their results shown. The chapter ends with a summary.

5.2 Evaluation of Alternatives for Fuzzy Logic Controllers

The starting point of the evaluation of Fuzzy Logic Controllers was a seven term controller rule base that was proposed by Mamdani and Assilian [4] and is given in Table 5.1. These rules were selected as they have been used in many fuzzy logic controller applications and are well understood. The reason for selecting a Mamdani FLS over a Takagi, Sugeno and Kang (TSK) FLS [138] and [34] is that the Mamdani FLS can compensate for uncertain measurements. The TSK FLS is limited to situations where there is no uncertainty (as in the design of deterministic TSK FL controllers) or when all the uncertainty can be accounted for just in the antecedent membership functions. This allows the Mamdani FLS to be more applicable to a broader range of applications and problems than a TSK FLS [22]. By using a TSK FLS the Dual Surface Type-2 FLC would be unnecessarily constrained and less generic and adaptable.

The rule bases for the fuzzy logic controllers used in this thesis are the same whether a type-1 fuzzy logic controller or a type-2 fuzzy logic controller is used. The membership functions used decide the type of the fuzzy logic controller.

The following generic assumptions were made in the simulation for the Mamdani controllers:

1. Double-input/single-output process is a non-linear input/output relationship.

2. The control goal is to obtain an output variable at steady state by altering the input variable. This may be noisy.

3. A non-idealized response is expected due to the inertia of the process being controlled.

The Fuzzy Inference System was set up to process the inference rules as follows. For fuzzification the range of the inputs and output was between -2.00 and 2.00 Volts. The rule connection method used was minimum, the inference action method was minimum and the aggregation method was maximum. There are many defuzzification methods available, the most common being the Centre of Gravity method for type-1 and the equivalent Centroid of Gravity interval generator for type-2 FLS. It was for this reason that the C of G methods were used in the Fuzzy Logic Systems.

The output of both controllers was scaled to the DC motor model input range of {-2,2} Volts. The output of the DC motor model was then scaled to the fuzzy Controller input range of {-2,2} Volts and fed back into the Fuzzy Inference System. The scaled DC Motor output was then subtracted from the input to create a new error value. Noise was not added to the motor input voltage signal $V_{app} \in$ [-2,2].

In order to investigate the effects of different membership function shapes on the type-1 seven term FLC, four sets of membership functions were used.

1. Gaussian Type-1 Fuzzy Logic Controller. The shapes of the seven term type-1 Gaussian membership functions and the surface are given in Figure C.1. The parameters are given in Table C.1.

2. Trapezoidal Fuzzy Logic Controller, where all the membership function terms were trapezoids. The shapes of the type-1 seven term trapezoidal membership functions and the surface are given in Figure C.2. The parameters for the membership functions are given in Table C.2.

3. A Triangular Fuzzy Logic Controller, where all the membership function terms were triangles. The shapes of the type-1 seven term triangular membership functions and the surface are given in Figure C.3. The parameters for the membership functions are given in Table C.3.

4. A Trapezoidal Triangular Fuzzy Logic Controller, where the two outer membership function terms were trapezoids and the five internal membership function terms were triangles. The shapes of the type-1 seven term triangular membership functions and the surface are given in Figure C.4. The parameters for the membership functions are given in Table C.4.

The positions of the membership function terms were equally distributed across the Universe of Discourse and symmetrically about the Universe of Discourse's zero. The intersection of the membership functions is in the region of a membership grade of 0.5. In order to generate and explore uncertainty in the definition of the membership functions, some of the crossovers were set below the 0.5 value. The Gaussian Type-1 seven term fuzzy logic controller was used as a guide to constructing the other three fuzzy logic controllers consisting of trapezoidal, triangular and trapezoidal triangular membership function sets. For each controller the complete Universe of Discourse was covered. By using four different fuzzy logic controllers, all similar in the membership function positions on the Universe of Discourse allowed the effects of variations in the membership function shapes to be explored.

The membership functions for the interval type-2 Fuzzy Logic Controllers were generated from the type-1 Fuzzy Logic Controllers. The method used is known as blurring. The technique is to create Footprints of Uncertainty (FOUs) using "uncertainty intervals". The points where the type-1 MFs met the Universe of Discourse were the centre points of the FOUs for each type-2 MF. For the three non Gaussian FLCs, the intervals were set nominally at +/- 0.2, for the terms nm, ns, ps and pm and +/- 0.5 for the nb, ze and pb terms about these centre points. The apex of the type-2 membership functions was as close to the type-1 membership function apexes, so that valid comparisons could be made in the experimental results. The reason for using uncertainty intervals is that it is the most common method of generating type-2 fuzzy sets. Four seven term interval type-2 Fuzzy

Logic Controllers were used in the experiments. In the following membership function figures, the uniformly shaded regions are the FOUs for the interval type-2 fuzzy sets.

1. Gaussian Type-2 Fuzzy Logic Controller. The shapes of the seven term type-2 Gaussian membership functions and the surface are given in Figure C.5. The parameters are given in Table C.1.

2. Trapezoidal Fuzzy Logic Controller, where all the MF terms were trapezoids. The shapes of the type-2 seven term trapezoidal MFs and the surface are given in Figure C.6. The parameters for the MFs are given in Table C.5.

3. A Triangular Fuzzy Logic Controller, where all the membership function terms were triangles. The shapes of the type-2 seven term triangular membership functions and the surface are given in Figure C.7. The parameters for the membership functions are given in Table C.6.

4. A Trapezoidal Triangular Fuzzy Logic Controller, where the two outer membership function terms were trapezoids and the five internal membership function terms were triangles. The shapes of the type-2 seven term trapezoidal triangular membership functions and the surface are given in Figure C.8. The parameters for the membership functions are given in Table C.7.

5.2.1 Running the Simulator for Seven Term Controllers

Having generated the membership function terms the controller under simulation was run for 200 timesteps of 1ms with a step change from 0 to 100 Vx10^{-2} at time $t = 1$. The results were recorded for each seven term controller for later analysis. Runs 1-4 were the type-1 controllers using Gaussian, Trapezoidal, Triangular and Trapezoidal-Triangular shaped membership functions respectively. Runs 5-8 were the corresponding type-2 controllers. Runs 9-16 are repeats of the first eight runs with Gaussian noise N(0,0.04), which gives 1% of the motor input voltage signal $V_{app} \in$ [-2,2], added to the input signal to generate process noise.

The statistic used to report the results was the Root Mean Square Error (RMSE) between the Motor Demand and the Process Variable of the DC motor simulator. The mea-

Run	MFShape	TypeNo	RMSE
7-1	Gaussian	T1	0.00486
7-2	Trapezoidal	T1	0.00398
7-3	Triangular	T1	0.00355
7-4	Trapezoidal Triangular	T1	0.00369
7-5	Gaussian	T2	0.00491
7-6	Trapezoidal	T2	0.00458
7-7	Triangular	T2	0.00390
7-8	Trapezoidal Triangular	T2	0.00387
7-9	GaussianN	T1	0.0410
7-10	TrapezoidalN	T1	0.0439
7-11	TriangularN	T1	0.0393
7-12	Trapezoidal TriangularN	T1	0.0384
7-13	GaussianN	T2	0.0411
7-14	TrapezoidalN	T2	0.0400
7-15	TriangularN	T2	0.0396
7-16	Trapezoidal TriangularN	T2	0.0390
RMSE	Mean	SD	
All w/o Noise	0.004168	0.000535	
All with Noise	0.040288	0.001731	
Type-1 w/o Noise	0.00402	0.000588	
Type-2 w/o Noise	0.004315	0.000515	
Type-1 with Noise	0.04065	0.00242	
Type-2 with Noise	0.039925	0.000885	

Table 5.2: Results for Seven Term Controllers

surement is made over 200 values from when the Motor Demand is increased from 0 to 1.00 V for each of the sixteen runs. The results are given in Table 5.2. The results of the experiment show that having a stable seven term controller the difference between the different shaped membership functions within the controller type is small. The worst performing controllers were the two Gaussian Controllers. When comparing type-1 controllers with type-2 controllers without noise, the type-2 controllers do not improve the RMSE, if anything they are worse. This is not what was expected, since type-2 controllers in the absence of uncertainty revert back to type-1 controllers [55, 38]. The expectation was that the type-2 controllers would be equivalent to the corresponding type-1 controller. When noise is added then the type-2 controllers improved over the type-1 controllers. This is in line with the literature, that type-2 controllers are better at handling uncertainty over type-1 controllers [22].

	Control Change	Error Change				
		nb	nms	zr	pms	pb
Error	nb	nb	nb	nb	nms	zr
	nms	nb	nb	nms	zr	pms
	zr	nb	nms	zr	pms	pb
	pms	nms	zr	pms	pb	pb
	pb	zr	pms	pb	pb	pb

Table 5.3: Five Term Fuzzy Logic Controller

Following on from the investigation into the seven term Fuzzy Logic Controllers the emphasis was on discovering if the number of membership function terms could be reduced. Intuitively it is expected that the more rules and terms that can be utilised to describe the problem the better the system can be modelled. In the limit the system would be fully described. However the amount of work required to achieve this would be prohibitive, and maybe impossible. Conversely then the question becomes what is the minimum number of rules and membership function terms that would be able to successfully control the micro robot's DC motors without a significant loss of control.

5.3 Selection of Type-1 and Type-2 controllers

This investigation was to compare the differences between a seven, a five and two three term controllers of type-1 and interval type-2. From the results of the previous investigation the two Gaussian Fuzzy Logic Controllers had the highest RMSE values of the eight controllers. For this reason it was decided to remove the Gaussian membership function shapes completely from the investigation. In order to provide a reference controller an independent term PID controller was included in the investigation. This is described in Chapter 3. The rule base for the seven term controllers was that used in the first investigation given in Table 5.1.

The five term rule base was generated from the seven term rule base, by combining the *NegMed* and *NegSmall* terms give a *NegMS* term and combining the *PosMed* and *PosSmall* terms to give a *PosMS* term. The rule base for the five term controllers is given in Table 5.3 where nb - *NegBig*, nms - *NegMS*, zr - *Zero*, pms - *PosMS* and pb - *PosBig*.

Finally the three term base was generated from the five term rule base, by combining

	Control Change	Error Change nb	cen	pb
Error	nb	nb	nb	cen
	cen	nb	cen	pb
	pb	cen	pb	pb

Table 5.4: Three Term Fuzzy Logic Controller

the *NegMS*, *Zero* and *PosMS* terms into the *Cen* term. The rule base for the three term controllers is given in Table 5.4, where nb - *NegBig*, cen - *Cen* and pb - *PosBig*.

The type-1 seven term controller was a Trapezoidal Triangular Fuzzy Logic Controller, where the two outer membership function terms were trapezoids and the five internal membership function terms were triangles, as shown in Figure C.2. The parameters for the type-1 seven term trapezoidal triangular controller are given Table C.4. This was decided upon due to the performance of the controller in the previous investigation being the best in the noise run. The *NegBig* and *PosBig* MFs were the trapezoids and the remainder triangles. This was due to the controller only infrequently operating at the top and bottom of the input and output ranges. The inputs and output linguistic variables used the same types and values for the membership functions in the controller. The membership function shapes for the type-1 five term trapezoidal triangular controller were generated so that the new triangular membership functions had an apex at the average of the original triangular apexes with a base of 1.5V and the *Zero* term having a base of 1V about the zero. The shapes of the type-1 five term trapezoidal triangular membership functions and the surface are given in Figure C.9. The parameters used in the type-1 five term trapezoidal triangular controller are given in Table C.8.

There were two type-1 three term controllers generated. One was a Trapezoidal Triangular Fuzzy Logic Controller, where the single internal membership function term was a triangle. The other type-1 three term controller was a Trapezoidal Fuzzy Logic Controller, where all the membership function terms were trapezoids. The membership function shapes for the three term type-1 triangular controller were generated from the type-1 five term controller using the seven to five generation method described previously. The membership function shapes for the three term trapezoidal type-1 controller had the *Cen* from -20 to +20 Vx10^{-2} equally about the zero point. The shapes of the type-1 three term

trapezoidal triangular membership functions and the surface are given in Figure C.10. The parameters used in the type-1 three term trapezoidal triangular controller are given in Table C.9. For the type-1 three term trapezoidal triangular membership functions, the shapes and the surface are given in Figure C.11. The parameters used in the type-1 three term trapezoidal controller are given in Table C.10.

5.3.1 Five and Three Term Type-2 Membership Functions

For the five and three type-2 controllers, the uncertainty interval method used to generate the seven term type-2 controllers was used. The uncertainty intervals were generated at approximately +/-14% about the type-1 membership function base end points. The choice of the FOU interval values was arbitrarily made since any value was uncertain. In the following membership function figures, the uniformly shaded regions are the FOUs for the interval type-2 fuzzy sets.

1. Five term type-2 trapezoidal triangular Fuzzy Logic Controller, where the two outer membership function terms were trapezoids and the three internal membership function terms were triangles. The shapes of the type-2 five term trapezoidal triangular membership functions and the surface are given in Figure C.12. The parameters for the membership functions are given in Table C.11.

2. Three term type-2 trapezoidal triangular Fuzzy Logic Controller, where the two outer membership function terms were trapezoids and the single internal membership function term was a triangle. The shapes of the type-2 five term trapezoidal triangular membership functions and the surface are given in Figure C.13. The parameters for the membership functions are given in Table C.12.

3. Three term type-2 trapezoidal Fuzzy Logic Controller, where all the membership function terms were trapezoids. The shapes of the type-2 three term trapezoidal membership functions and the surface are given in Figure C.14. The parameters for the membership functions are given in Table C.13.

5.4 Running the Simulator for Seven, Five and Three Term Controllers

The Fuzzy Inference System set up to process the inference rules was that used in the seven term controller investigation. That is the MIN rule connection method, the MIN inference action method and the MAX aggregation method. For defuzzification, the type-1 FLC used a centre of gravity method to generate a crisp value. The interval type-2 FLC used a centroid of gravity method, and a crisp value was obtained by averaging the upper and lower interval values.

There were two generic methods used in running the controller simulations. The first was to run the controller under simulation for 600 timesteps of 1ms with a step change from 0 to 100 Vx10^{-2} at time $t = 1$. The second was to make the step change and then increase the inertia parameter J of the DC motor model by a factor of 10 at time $t = 300$. This represents a load on the wheels, such as occurs when the robot hits an obstacle. Gaussian noise N(0,0.04), which gives 1% of the motor voltage $V_{app} \in$ [-2,2], was added to $Vapp$ to generate process noise. The added noise runs were repeated fifty times to obtain an average measurement. In total fifty six simulation runs were made.

The FLC and PID controllers were run for 600 timesteps of 1 ms with a step change from 0V to 1V at time $t = 1$. Then a second run was made increasing the inertia parameter J of the DC motor model by a factor of 10 at $t = 300$. This represents a load on the wheels, such as occurs when the robot hits an obstacle. Gaussian noise N(0,0.04), which gives 1% of the motor voltage $V_{app} \in$ [-2,2], was added to $Vapp$ to generate process noise. The runs were repeated 50 times to obtain an average measurement.

5.4.1 FLC Step Change RMSE Results

The Root Mean Square Error (RMSE) between the Motor Demand and the Process Variable of the DC motor was calculated for each controller. In the step change without noise set of runs the results are given in Table 5.5. The results show that the best performing controller in the without noise set of runs was the type-1 seven term trapezoidal triangular controller. However, the other type-2 controllers out performed their equivalent type-1 controllers. The worst performing controller is the five term type-1, with a RMSE value

MF Shape	RMSE Type-1 (Rank)	RMSE Type-2 (Rank)
	Without Noise	
7MF Trapezoidal Triangular	0.000730(1)	0.000923(2)
5MF Trapezoidal Triangular	0.002818(8)	0.001565(7)
3MF Trapezoidal Triangular	0.001019(6)	0.001017(5)
3MF Trapezoidal	0.001016(4)	0.001013(3)
	With Noise	
7MF Trapezoidal Triangular	0.040038(5)	0.039046(2)
5MF Trapezoidal Triangular	0.041480(7)	0.039112(3)
3MF Trapezoidal Triangular	0.041629(8)	0.041113(6)
3MF Trapezoidal	0.038871(1)	0.039222(4)

Table 5.5: Step RMSE for 7, 5 and 3 Term Controllers

	WITHOUT		NOISE	
	Mean	SD	Mean	SD
Type-1	0.00402	0.000588	0.04065	0.00242
Type-2	0.004315	0.000515	0.039925	0.000885

Table 5.6: Step Change RMSE Means and SDs

four times that of the best controller, the type-1 seven term trapezoidal triangular controller. The five term type-1 controller has an intersection value of 0.3 of membership grade, which is the probable reason for the poor performance. The three term controllers are very close, differing only in the sixth decimal place for both type-1 and type-2. Comparing the means of all type-1 against type-2 controllers in Table 5.6, shows that the type-2 controllers have a lower RMSE average and standard deviation. The difference between the means of type-1 and type-2 is $2.66E^{-4}$.

In the with noise set of runs the type-2 controllers outperform their type-1 counterparts except for the three term trapezoidal controller which performed the best. Again the mean and standard deviation of the type-2 controllers is lower but the difference between them is much closer. The type-2 five term controller improved by four ranking places. The controller uses a maximum aggregation method and under noise the effective membership grade intersection point is probably higher. Again the type-2 controllers have a lower RMSE average and standard deviation. The difference between the means of type-1 and type-2 is $7.91E^{-3}$.

5.4.2 FLC Inertia Change RMSE Results

In the inertia change without noise set of runs the results are given in Table 5.7. The results show that the best performing controller is the type-1 seven term controller. The worst performing controller is the type-1 five term controller. The rank order of the type-1 controllers is the same as the step only runs. The rank order of the type-2 controllers is the same as the step only runs, but overall the type-2 three term trapezoidal triangular controller improved its ranking to equal that of the type-1 three term trapezoidal controller. With no noise present the type-2 controllers outperformed their type-1 equivalents except for the seven term case. The worst performing controller is the type-1 five term controller, with a RMSE value three times that of the best controller, the type-1 seven term controller. The RMSE of the three term controllers are very close only differing by 4×10^{-6}. Comparing the means of all type-1 against type-2 controllers in Table 5.8, shows that the type-2 controllers have a lower RMSE average and standard deviation. The difference between the means of type-1 and type-2 is $2.61E^{-4}$.

In the with noise runs the results show that again the best performing controller was the type-1 seven term controller. The most dramatic change occurred in the type-1 five term controller which improved from last to second. Contrary to expectation, when noise was present the type-1 controllers were the best, except for the three term trapezoidal triangular case. Comparing the means in Table 5.8, shows that the type-1 controllers had a lower RMSE average and standard deviation when noise was present. The difference between the means of type-1 and type-2 is $6.22E^{-3}$. This is the only occasion that the type-1 controller's average is better than the type-2 controllers. A possible explanation is that the level of noise at 1% was insufficient to generate a difference.

5.4.3 FLC Statistical Analysis

A statistical analysis of the results was carried out. The means and standard deviations for the type-1 and type-2 RMSEs for the step and inertia responses are given in Table 5.9. In both the step response cases the type-2 controllers had a lower overall RMSE. In the inertia response case without noise the type-2 controllers had a lower overall RMSE. However with noise the type-1 controllers had the lower overall RMSE. A paired t test

MF Shape	RMSE Type-1 (Rank)	RMSE Type-2 (Rank)
	Without Noise	
7MF Trapezoidal Triangular	0.000996(1)	0.001171(2)
5MF Trapezoidal Triangular	0.002959(8)	0.001742(7)
3MF Trapezoidal Triangular	0.001359(6)	0.001357(4)
3MF Trapezoidal	0.001357(4)	0.001354(3)
	With Noise	
7MF Trapezoidal Triangular	0.039015(1)	0.039353(3)
5MF Trapezoidal Triangular	0.039153(2)	0.040627(7)
3MF Trapezoidal Triangular	0.040225(5)	0.039898(4)
3MF Trapezoidal	0.040274(6)	0.041277(8)

Table 5.7: Inertia RMSE for 7, 5 and 3 Term Controllers

	WITHOUT		NOISE	
	Mean	SD	Mean	SD
PID	0.207588	-	0.214935	-
Type-1	0.001667	0.000878	0.039667	0.000675
Type-2	0.001406	0.000241	0.040289	0.000840

Table 5.8: Inertia RMSE 7, 5 and 3 Terms Means and SDs

were performed to compare the type-1 and type-2 controllers for a step change without noise. No statistical difference was found at the 95% significance level. This was repeated for the other three cases with the same result.

5.4.4 PID Controller Step and Inertia Results

The RMSE of the PID controller in the case of the step change is 44 times greater than the mean of the RMSE for all the type-1 controllers without noise and 55 times greater than the mean of the RMSE for all the type-2 controllers without noise, see Table 5.9. The ratios of PID to type-1 and for PID to type-2 for the mean of all the RMSE FLCs with noise are 1.86 and 1.89 respectively. For the inertia change case the RMSE of the PID controller is 16 times greater that the mean of the RMSE for all the type-1 controllers without noise and 19 times greater than the mean of the RMSE for all the type-2 controllers without noise. The ratios of PID to type-1 and for PID to type-2 for the mean of all the RMSE FLCs with noise are 5.8 and 5.3 respectively. These results show that the PID controller is outperformed by both type-1 and type-2 Fuzzy Logic Controllers.

Step Response	Without Noise		With Noise	
	Mean	SD	Mean	SD
PID	0.062436	-	0.075369	-
Mean Type-1	0.001396	0.000958	0.040504	0.001304
Mean Type-2	0.001130	0.000293	0.039713	0.000941
Inertia Response	Without Noise		With Noise	
	Mean	SD	Mean	SD
PID	0.207588	-	0.214935	-
Mean Type-1	0.001667	0.000878	0.039667	0.000675
Mean Type-2	0.001406	0.000241	0.040289	0.000840

Table 5.9: Mean and SD of All FLCs

5.4.5 Comparison of Type-2 Controllers with Type-1 and PID

An overview of the PID and Fuzzy Logic Controllers' responses to the step and inertia changes without noise is given in Fig C.15. In the large scale graphs the FLCs are all flat lined compared to the PID controller response to them. This visually shows that the PID controller is outperformed by the Fuzzy Logic Controllers. Expanding the scale and removing the PID controller's response shows the overall type-1 and type-2 response over six hundred timesteps in Figures C.16 and C.17 respectively. The type-1 response graph shows that the five term type-1 FLC T1-5, has the largest overshoot to the step demand. The smallest overshoot is given by the two three-term FLCs T1-trap and T1-tri. All these three controllers settle to the offset position of -0.999, with T1-5 controller being the slowest. The seven term controller T1-7, has a larger overshoot over the three term controllers but settled to the best offset position of -0.9998. At step 300 the inertia load was increased. The seven term controller reached the set point the quickest with a slight over shoot and the best offset. The five and three term controllers followed each other returning to their pre inertia position. The type-2 response graph shows that the largest overshoot came again from the five term FLC controller. The two three term controllers had the smallest overshoot but had a larger offset position. The seven term controller settled quicker than the five term controller, with both closest to the set point of one. At step 300 the seven term controller reached the set point the quickest with a slight overshoot. The five term controller was slightly slower without an overshoot. Both had a very small offset. The two three term controllers performed with the same characteristics and finished with a slightly larger offset. In Fig C.18 the without noise

responses for all eight fuzzy logic controllers to a step change are plotted over one hundred and twenty time steps, to show the difference between the type-1 and type-2 controllers. This shows that the three term controllers responded equally to the step change. The best response came from the type-1 seven term followed by the type-2 seven term, this was followed by the type-2 five term. The worst response came from the type-1 five term controller. In Fig C.19 the without noise responses for all eight fuzzy logic controllers to an inertia change are plotted over one hundred and twenty time steps showing the difference between the type-1 and type-2 controllers. The type-1 seven term controller T1-7, responded the quickest with a slight overshoot. This was followed by the seven term type-2 controller T2-7 with no overshoot, and then the type-2 five term controller T5-2 with a small offset. All the other controllers followed the same response curve. The responses are further separated and expanded into type-1 responses for step and inertia changes in Figures C.20 and C.21. The corresponding type-2 responses for step and inertia changes are given in Figures C.22 and C.23. The keys used to denote the controller responses in the figures are given in Table C.14.

For the with noise case the PID and type-2 controller responses are plotted in Fig C.24. Fig C.25 shows the noisy response to the step change for each controller type with the PID response added. The corresponding response to the inertia change is given in Fig C.26.

Controller Response Times and Offset Positions

The response results for the step change are given in Table C.15. In response to the step change the lowest overshoot was given by the three term type-2 trapezoidal controller — 0.11% at $t = 1$. This controller crossed back over the setpoint (SP) at $t = 7$, and reached a minimum at $t = 40$. It had reached a steady state by $t = 158$. The best time to steady state was from the seven term type-1 controller — 99.98% in 104 timesteps. The type-1 and type-2 three term controller responses are extremely close together as shown in Fig C.18.

The results of the inertia change response are given in Table C.16. When responding to the inertia change all the controllers dipped to 99.616% of the SP. The seven term type-1 crossed back at $t = 357$ overshooting by 0.006% of the SP. It returned to the setpoint at $t = 445$ and reached a steady state of 99.97% of SP. The type-2 controller crossed back at $t = 383$ overshooting by 0.0002% of SP, and returned to the setpoint at $t = 461$, reaching

a steady state of 99.96% of SP. Both types of five and three term controllers behave in an overdamped way having maximum values of 99.89% of SP at $t = 600$. Both the type-1 and type-2 three term controllers together with the type-1 five term are indistinguishable, as shown in Fig C.19.

The results for the PID response are given in Table C.17. The PID controller reached 90% of SP at $t = 9$, and crossed the SP at $t = 37$. It reached a maximum of 100.079% of SP at $t = 54$. The PID controller reached a steady state of 100.0016% of SP at $t = 139$. When the inertia change occurred the controller went to 0% of SP at $t = 301$ and reached 90% of SP at $t = 331$. It crossed the SP at $t = 334$ and reached a maximum of 100.479% at $t = 367$. It had not reached a steady state by $t = 600$.

In Fig C.26 the noisy response to the inertia change of the controllers is given. When the inertia change occurred no observable difference was detected in the controllers response.

5.5 Discussion

It can be seen in Tables 5.5 and 5.7 that the overall performance of the two five term controllers were the worst. The intersection of their membership functions was much lower than the MFs of the seven term controllers and is a possible cause for the poor performance. However the improvement of the type-2 five term controller with noise over its without noise equivalent is noted, especially when it is not repeated for the inertia change results. Generally like for like, the three term controllers RMSEs were between the RMSEs of the seven term and five term controllers. When noise was applied, as expected the type-2 controllers were usually better than the type-1 controllers for the step change response. Surprisingly in the inertia change response with noise, the type-1 controllers were generally better. A possible cause for this is that the level of noise was too low. The results of increasing the load on the motor correspond with those obtained by Hagras [124], with the type-2 FLC's being more damped than the type-1 controllers. When noise was applied the overall means of the type-1 and type-2 FLC RMSEs increased by over twenty five times, where as the PID RMSE increased by 21% in the step response and by 3.5% for the inertia response. However the fuzzy controllers completely outperformed the

PID controller. For the type-2 controllers with triangular MFs, the created FOU decreases towards the peak of the triangles reducing the uncertainty to zero. This was kept so as to be consistent with the original seven term controllers. However, for the three term type-2 controller with the triangular *Cen* MF, there is less uncertainty around zero. Statistically there was no significant difference between the type-1 and type-2 controllers at the 95% level for both the step change and the inertia change. This suggests that either type-1 or type-2 controllers can be used. The study demonstrates that, in simulation of a micro robot DC Motor, seven and five term controllers can be nominally replaced by ideally, a three term controller of type-1 without appreciable loss of control. The three term membership functions can either be all trapezoidal or the central membership function triangular. The results show that the three term controllers are as stable as the seven and five term controllers when a step change or load is applied. When noise is applied the three term controllers perform equally with the seven and five term controllers.

By starting with seven term type-1 and type-2 controllers and then reducing the number of membership functions down to five then three, it was possible to show that it was valid to use three term controllers as alternatives to the seven term controllers. This meant that the effort needed to create and test membership functions could be significantly reduced.

5.6 Summary

In this Chapter a range of alternative designs of fuzzy logic controllers were evaluated. A seven term rule base was set up as the starting point of the fuzzy controllers. Eight controllers were developed, four of type-1 and four of type-2. Four sets of membership functions were designed for each set of fuzzy logic controllers. These consisted of seven term gauss, trapezoidal, triangular and trapezoidal triangular membership functions. The DC motor simulation was run in the fuzzy logic system for the eight controllers and for the comparison PID controller. The simulations were run with and without noise applying a step change to the input and an inertia change to the motor. It was shown that both the type-1 and type-2 Gaussian fuzzy controllers were performing the worst and consequently were eliminated from the evaluation. A five term rule base was created

by combining rules of the seven term rule base. Trapezoidal, triangular and trapezoidal triangular membership functions for type-1 and type-2 controllers were designed and implemented. The simulation experiments were then repeated for these controllers. Finally a three term rule base was created again by combining the five term rules and the experiments were repeated. It was demonstrated that the fuzzy logic controllers outperformed the PID controller in terms of response and resilience. It was shown that the seven term and five term controllers can be successfully replaced by three term controllers without a significant loss of control. It was shown that the trapezoidal, triangular and trapezoidal triangular membership functions in the three term controllers are effectively equivalent.

The results demonstrated that the fuzzy logic controllers completely outperformed the PID controller and that a fuzzy controller should be used in preference to a PID controller where ever possible.

In the next chapter the average thresholds mechanism is explored for type-2 fuzzy logic controllers in conjunction with the membership functions thresholds.

Chapter 6

The Investigation of Single/Dual Surfaces for Type-2 Fuzzy Logic Controllers

Having completed the evaluation of membership functions for the type-1 and type-2 controllers, the membership function selection of the controllers was addressed. The previous chapter showed that three term rule based controllers could be used in place of seven and five term rule based controllers, without a significant loss of control capability. It was also shown that the shapes of the membership functions used within the three term controllers were equivalent in terms of response performance. The membership function parameters were initially selected to give evenly distributed shapes with their intersections about 0.5 of the membership grade. This however was not a hard and fast rule and any other intersections were broadly acceptable. The results from Chapter 5, showed that the performance of the five term controllers were generally the worst performing of the controllers, however there was no statistical difference between any of them. It was decided to concentrate on the three term controllers, as they were the simplest to generate. The seven term controllers were used as a comparison, since they performed very slightly better. This scenario then eliminated the need to study the five term controllers, and they were consequently dropped from the tuning investigation. In all seven controller types were selected, three type-1 and four type-2 with a mix of trapezoidal and trapezoidal triangular membership functions.

	Control Change	Error Change						
		nb	nm	ns	zr	ps	pm	pb
Error	nb	nb	nb	nb	nb	nm	ns	zr
	nm	nb	nb	nb	nm	ns	zr	ps
	ns	nb	nb	nm	ns	zr	ps	pm
	zr	nb	nm	ns	zr	ps	pm	pb
	ps	nm	ns	zr	ps	pm	pb	pb
	pm	ns	zr	ps	pm	pb	pb	pb
	pb	zr	ps	pm	pb	pb	pb	pb

Table 6.1: Seven Term Fuzzy Logic Controller

	Control Change	Error Change		
		nb	cen	pb
Error	nb	nb	nb	cen
	cen	nb	cen	pb
	pb	cen	pb	pb

Table 6.2: Three Term Fuzzy Logic Controller

6.1 Setup

In order to maintain compatibility across experiments, the setup that was used in Chapter 5, to investigate the selection of the fuzzy logic controllers, was used for the membership function selection process experiments for the dual surface controllers. The fuzzy base rules, membership function parameters and fuzzy logic system options are contained in .fcl ASCII files. The implementation of the tuning parameters was simply accomplished by editing the .fcl file and running the type-1 or type-2 fuzzy logic system appropriately. The rule bases and membership function parameters used in the membership function process are given as follows.

6.1.1 Seven Term and Three Term Fuzzy Logic Rules

The seven term and three term fuzzy logic rules and assumptions used in the tuning simulation were those used in Chapter 5. These are repeated in Table 6.1 for the seven term controller and in Table 6.2 for the three term controller for ease of reference.

6.1.2 Seven Term Membership Function Parameters and Shapes

The membership functions shapes of the seven term type-2 fuzzy logic controller are all trapezoids and their parameters are given in Table D.1 - Seven Term Type-2 Trapezoidal MF Parameters. For each term in the three variables there is an outer and inner parameter value. They define the outer and inner bounding edge of the footprint of uncertainty for each term. The outer and inner bounding edge pairs are parallel with a linear separation of 10 units. The membership functions are regularly spaced with a separation of 60 units between the five inner membership functions. This gave an equal coverage across the Universe of Discourse for the three controller variables.

Three threshold values were applied to the membership functions. The first threshold was applied at a membership grade of 1.0. This allowed the full membership function action to be applied across both inputs and the output variables - Figure D.1 The second threshold was set at 0.9 of the membership grade for the three variables - Figure D.2. So any variable value that generated a membership grade above the threshold was given the threshold value. The third threshold was set to 0.8 of the membership grade - Figure D.3.

The type-1 fuzzy logic controller used in the experiments was T1Trap7. This used the membership function parameters and shapes as given in Chapter 5 Table C.2 and shown in Chapter 5 Figure C.2.

6.1.3 Three Term Membership Function Parameters and Shapes

For the type-2 fuzzy logic controllers, three sets of three term membership functions were used in the simulations. The first set consisted of all trapezoidal membership functions. The first controller was identified as T2Trap3. The membership function parameters are given in Table D.2. Three threshold levels were applied to the membership functions, as in the seven term membership functions, with values of 1, 0.9 and 0.8. The membership function shapes are shown in Figures D.4 - D.6. The second set consisted of trapezoidal triangular membership functions. The membership function parameters are given in Table D.3. Three threshold levels were applied to the membership functions, with values of 1, 0.9 and 0.8. The membership function shapes are shown in Figures D.7 - D.9.

The defining feature of this set is that the inner membership function of the 'cen'

term tapers to zero. The set of membership functions is referred to as Type-2 Trapezoidal Triangular Three Term 300, and the controller as T2Tri300.

The third set also consisted of trapezoidal triangular membership functions, with the same three thresholds. The difference is that the outer and inner membership functions are parallel and the inner is grounded at +-5. This set is referred to as Type-2 Trapezoidal Triangular Three Term 305, and the controller as T2Tri305. The membership function parameters are given in Table D.4 and the membership function shapes are shown in Figures D.10 - D.12.

The trapezoidal type-1 fuzzy logic controller, T1Trap3, used the type-1 membership function parameters and shapes as given in Chapter 5 Table C.10 and FigureT1Trape3MF. The trapezoidal triangular type-1 fuzzy logic controller, T1Tri3, used the type-1 membership function parameters and shapes as given in Chapter 5 Table C.9 and Figure C.10.

6.2 Controller Surfaces

6.2.1 Seven Term Membership Function Surfaces

The method described in Chapter 4 for generating surfaces was used. For each seven term controller the surfaces for the implication action of minimum and the implication action of product were generated, since both of these implications are utilised in the simulations. The surfaces for the three different thresholds are given for each of the two implication actions. The type-2 seven term trapezoidal minimum surfaces are shown in Figure D.13. The surfaces for the product implication are shown in Figure D.14.

On initial study of the surfaces they appear to be the same, but closer inspection shows that there are subtle differences in the structures.

For the type-1 seven term controller, T1Trap7, the surface generated was for the minimum implication and is given in Chapter 5 Figure C.2. Comparing the type-2 and type-1 surfaces, it is seen that the both the type-2 surfaces are much smoother than the type-1 surface. This is expected to be repeated in the simulations with the type-2 controllers producing lower RMS error values than generated by the type-1 controllers.

6.2.2 Three Term Membership Function Surfaces

For each of the three term type-2 controllers again the implication action of minimum and product were generated for the three controllers. The surfaces for the T2Trap3 controllers are shown in Figures D.15 and D.16 for the minimum and product implications respectively.

The surfaces for the two sets of type-2 three term trapezoidal triangular 300 membership functions controllers, T2Tri300, are given in Figures D.17 and D.18 for the minimum and product implications respectively. Finally the surfaces for the T2Tri305 controllers are given in Figures D.19 and D.20 for the minimum and product implications respectively.

Again the differences in the surfaces when compared with each other were very hard to discern, suggesting that the controllers will perform equally against each other.

For the type-1 three term controller, T1Trap3, the surface generated was for the minimum implication and is given in Chapter 5 Figure C.11. The surface for T1Tri3, type-1 three term controller is shown in Chapter 5 Figure C.10.

Comparing the type-2 and type-1 surfaces, it is seen that the both the type-2 surfaces are less precipitant than the type-1 surfaces. Again a smoother response is expected from the type-2 controllers together with lower RMS error values than generated by the type-1 controllers in the simulations.

6.3 Simulator Experiments

6.3.1 Simulation Inputs

The input used in the simulation is given in Table D.5. The input setpoints are changed every 91 steps for a run of 1000 steps. The setpoint is held at the current value between the setpoint changes. This forces the controllers to react to setpoint step changes which reflect the setpoint changes experienced say, in robot football. Also any instabilities in the controllers could be exposed, however this is unlikely in the micro robot simulation environment. The graph of the input is given in Figure D.21

Number of Steps	1000	IMP - Min
Noise%	0	
Input	Step Ramp	
MFs	Without Inertia	With Inertia
T1trap3	1	2
T1tri3	3	4
T1trap7	5	6
T2trap3	7	8
T2tri305	9	10
T2tri300	11	12
T2trap7	13	14

Table 6.3: MF Run Numbers for Type-1 and Type-2 FLCs

6.4 Type-1 and Type-2 Seven and Three Term FLC Simulation Experiments

The purpose of carrying out these experiments was to determine a fuzzy logic controller configuration that could be used in the real world micro robot experiments. These experiments provide a measure of confidence that the fuzzy logic controllers selected performed well in simulation experiments compared to the others tested.

6.4.1 Experiment Comparing Type-2 and Type-1 FLCs without Noise

The input used in the experiment was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, for both no inertia change and inertia change in the DC motor model. There was no added noise added to the input signal. The simulation used the minimum implication method. The membership functions used and their run number references, together with the other details of the experiment are given in Table 6.3

6.4.2 Simulation Results

The simulation RMSE results for the tests carried out in Table 6.3 are detailed as follows. The comparison between type-1 and type-2 fuzzy logic controllers is given in Table 6.4. The results in Table 6.4 show that the all the controllers handle the change in inertia

Run	MF	File	RMS_FZC	Inertia	Diff	RMS_PID
		Type-1				
1	T1trap3	4979.3	-		1.7E14	
2	T1trap3	5880.4	I	901.1	1.7E14	
3	T1tri3	4705.2	-		-	
4	T1tri3	5648.5	I	943.3	1.7E14	
5	T1trap7	6254.1	-		-	
6	T1trap7	6873.5	I	619.4	1.7E14	
		Type-2				
7	T2trap3	4928.7	-		-	
8	T2trap3	5842.7	I	914	-	
9	T2tri305	4665.4	-		-	
10	T2tri305	5617.6	I	952.2	-	
11	T2tri300	4690.5	-		-	
12	T2tri300	5640.3	I	949.8	-	
13	T2trap7	6186.2	-		-	
14	T2trap7	6821.7	I	635.5	-	

Table 6.4: Type-1 FLCs against Type-2

successfully. The differences between the RMSEs for with and without inertia are given under Diff. The smallest difference is given by the type-1 seven term controller T1Trap7, the largest difference is given by the type-2 three term controller T2Tri305. The results for the PID controller show that it performed very badly, and is not commented on.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if the type-1 and type-2 membership functions could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of membership functions.

Alternative Hypothesis H_1: samples are not drawn from the same population of membership functions.

The value of the statistic was p = 0.69 $>$ 0.05, so the Null Hypothesis cannot be rejected.

The same results with the controllers grouped by an inertia change are given in Table 6.5. The Diff column shows that the all the type-2 controllers outperformed their corresponding type-1 controllers. The ranking of the controllers for both the inertia cases was the same. The best controller was the type-2 three term controller T2Tri305 with the

Run	MF	File	RMS_FZC	Inertia	Diff	Rank
	Without Inertia					
1	T1trap3	4979.3	-		5	
7	T2trap3	4928.7	-	50.6	4	
3	T1tri3	4705.2	-		3	
9	T2tri305	4665.4	-	39.8	1	
11	TtTri300	4690.5	-	14.7	2	
5	T1trap7	6254.1	-		7	
13	T2trap7	6186.2	-	67.9	6	
	With Inertia					
2	T1tri3	5880.4	I		5	
8	T2trap3	5842.7	I	37.7	4	
4	T1tri3	5648.5	I		3	
10	T2tri305	5617.6	I	30.9	1	
12	T2tri300	5640.3	I	8.2	2	
6	T1trap7	6873.5	I		7	
14	T2trap7	6821.7	I	51.8	6	

Table 6.5: Type1 against Type-2 within Inertia

lowest RMSE error for both inertia cases. All the three term controllers outperformed all the seven term controllers.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if the without inertia and with inertia data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of inertia types. Alternative Hypothesis H_1: samples are not drawn from the same population of inertia types.

The value of the statistic was p = 5.71881E-06 $<$ 0.05, so the Null Hypothesis will have to be rejected.

The results show that the RMSE error was consistently lower for the type-2 controllers compared with the type-1 controllers. Within the type-2 controllers the best performing controller was the T2Tri305 controller.

6.4.3 Experiment Comparing Type-2 and Type-1 FLCs with Noise

The input used in the experiment was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, for both no inertia change

and inertia change in the DC motor model. Randomly generated noise, as described in Chapter 5, was added to the input signal as a percentage of the input. The percentage values used were 1% and 16%. Simulations using intermediate noise levels were carried out using values of 2%, 4%, 8% and 12%, however these were not reported since they only increased the RMSE proportionally and the results lay between the 1% and 16% values. The simulation used the minimum implication method. The membership functions used and their run number references, together with the other details of the experiment are given in Table D.6

6.4.4 Simulation Results

The simulation RMSE results for the tests carried out in Table D.6 are detailed as follows. In Table 6.6 the results are grouped by noise within inertia case within controller type. Ranking the results shows that for type-1 controllers, the T1trap3 controller has the lowest rank sum. For the type-2 controllers, the T2tri305 has the lowest rank sum. The results for the PID controller show that it performed very badly, and is not commented on.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if all the type-1 MF data and all the type-2 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of MF types.
Alternative Hypothesis H_1: samples are not drawn from the same population of MF types.

The value of the statistic was p = 0.89 $>$ 0.05, so the Null Hypothesis cannot be rejected.

In Table 6.7 the results are group by controller type within noise within inertia case. For no inertia and noise of 1% the best controller was T2tri305, and for 16% noise it was T1trap7. The with inertia change simulation the best controllers were T2tri300 for 1% noise and T1trap3 for 16% noise. This result is rather surprising as from the no noise experiments the best controllers were type-2. Also this goes against the perceived belief that type-2 controllers outperform type-1 controllers. It is seen that for the inertia change with noise of 16% all the type-2 controllers were outperformed by their type-1 equivalents.

Student's t-test for two samples with unequal variances for a two tailed option were

Run	MF	RMS_FZC	Inertia	Noise%	RMS_PID	Rank
			Type-1			
101	T1trap3	5212.6	-	1	2E14	2
105	T1tri3	4890.4	-	1	-	1
109	T1trap7	6368.7	-	1	-	3
102	T1trap3	20659.4	-	16	2E15	2
106	T1tri3	21464.7	-	16	-	3
110	T1trap7	17955.9	-	16	-	1
103	T1trap3	5960.0	I	1	2E14	2
107	T1tri3	5925.5	I	1	2E14	1
111	T1trap7	6993.1	I	1	2E14	3
104	T1trap3	18718.5	I	16	2E15	1
108	T1tri3	19964.1	I	16	2E15	3
112	T1trap7	19066.4	I	16	2E15	2
		Type-2				
113	T2trap3	5092.5	-	1	-	3
117	T2tri305	4734.0	-	1	-	1
119	T2tri300	4835.2	-	1	-	2
125	T2trap7	6382.8	-	1	-	4
114	T2trap3	20477.8	-	16	-	2
118	T2tri305	19044.7	-	16	-	1
120	T2tri300	22346.1	-	16	-	4
126	T2trap7	20957.7	-	16	-	3
115	T2trap3	5841.6	I	1	-	2
121	T2tri305	5882.4	I	1	-	3
123	T2tri300	5753.7	I	1	-	1
127	T2trap7	7022.3	I	1	-	4
116	T2trap3	19482.2	I	16	-	1
122	T2tri305	20045.8	I	16	-	2
124	T2tri300	20903.9	I	16	-	3
128	T2trap7	22064.6	I	16	-	4

Table 6.6: RMSE for Type-1 and Type-2 FLCs with Noise

Run	MF	RMS_FZC	Inertia	Noise%	Diff	Rank
			No Inertia Noise 1%			
101	T1trap3	5212.6	-	1		5
113	T2trap3	5092.5	-	1	120.1	4
105	T1tri3	4890.4	-	1		3
117	T2tri305	4734.0	-	1	156.4	1
119	T2tri300	4835.2	-	1	35.2	2
109	T1trap7	6368.7	-	1		6
125	T2trap7	6382.8	-	1	-85.9	7
			No Inertia Noise 16%			
102	T1trap3	20659.4	-	16		4
114	T2trap3	20477.8	-	16	181.6	3
106	T1tri3	21464.7	-	16		6
118	T2tri305	19044.7	-	16	2420	2
120	T2tri300	22346.1	-	16	-881.4	7
110	T1trap7	17955.9	-	16		1
126	T2trap7	20957.7	-	16	-3001.8	5
			Inertia Noise 1%			
103	T1trap3	5960.0	I	1		5
115	T2trap3	5841.6	I	1	119.4	2
107	T1tri3	5925.5	I	1		4
121	T2tri305	5882.4	I	1	43.1	3
123	T2tri300	5753.7	I	1	171.8	1
111	T1trap7	6993.1	I	1		6
127	T2trap7	7022.3	I	1	-29.2	7
			Inertia Noise 16%			
104	T1trap3	18718.5	I	16		1
116	T2trap3	19482.2	I	16	-763.7	3
108	T1tri3	19964.1	I	16		4
122	T2tri305	20045.8	I	16	-81.7	5
124	T2tri300	20903.9	I	16	-939.8	6
112	T1trap7	19066.4	I	16		2
128	T2trap7	22064.6	I	16	-2998.2	7

Table 6.7: RMSE for Type-1 and Type-2 FLCs with Noise

carried out on the data to test if all No Inertia and all Inertia data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of Inertia types. Alternative Hypothesis H_1: samples are not drawn from the same population of Inertia types.

The value of the statistic was p = 0.94 $>$ 0.05, so the Null Hypothesis cannot be rejected.

For the with noise experiment the expectation that the type-2 controllers would be better suited to handling the input noise. The initial reaction was that the type-2 membership functions were not blurred enough and did not cover the range of the noisy signal. However this argument does not hold for the 16% noisy signal. All the type-1 controllers performed better, using single membership functions, than the type-2 controllers.

6.4.5 Experiment Comparing Type-2 FLCs with Varying Membership Thresholds and Noise Levels

This experiment is concerned with the performance of the type-2 controller when membership threshold were applied to the membership functions. Two levels of membership threshold were applied to the membership functions. The membership threshold was a value above which the inner surface of the membership function's degree of membership could not exceed. The first membership threshold value was 0.9 and the second membership threshold value was 0.8. The input used in the experiment was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, for both no inertia change and inertia change in the DC motor model. Randomly generated Gaussian noise, as described in Chapter 5, was added to the input signal as a percentage of the input. The percentage values used were 1% and 16%. The simulation used the minimum implication method. The membership functions used and their run number references, together with the other details of the experiment are given in Table D.7

6.4.6 Simulation Results

The RMSE results from running the simulation tests given in Table D.7 are given as follows.

Table D.8 groups the results by the membership functions of the controller, and by the thresholds applied to each controller with inertia applied and no noise. Results from the previously described simulations are included for comparison purposes, indicated by the run number against the table entry.

The results show that for within membership thresholds the controller with the lowest RMSE is the T2Tri305 type-2 controller with a value of 5617.6. The next best performing controller is the T2Tri300 with a value of 5640. The RMSE for this controller does not alter for the two membership threshold values. Across the experiment the change in the threshold level does not significantly alter the reported RMSE.

The results of running the simulation without a change in inertia are given in Table D.9. The table groups the results by the noise level within the applied membership threshold for the membership function used in the controller.

Three Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if T2trap3, T2tri3 and T2trap7 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of MF types.
Alternative Hypothesis H_1: samples are not drawn from the same population of MF types.

For T2trap3 and T2tri3, the value of the statistic was p = 0.92 $>$ 0.05, so the Null Hypothesis cannot be rejected. For T2trap3 and T2trap7, the value of the statistic was p = 0.82 $>$ 0.05, so the Null Hypothesis cannot be rejected. For T2trap7 and T2tri3, the value of the statistic was p = 0.73 $>$ 0.05, so the Null Hypothesis cannot be rejected.

The best performing controller is the T2Tri305 controller for a no noise input with a RMSE value of 4665.4. For the three term trapezoidal controllers the 0.8 threshold version has the lowest rank sum of 12. The triangular 300 controller has joint lowest rank sums for the 0.9 and 0.8 thresholds, and the seven term controller 0.8 threshold is lowest with a rank sum of 10.

The results for varying the membership function used for the controller within the required thresholds for the applied noise are given in Table D.10.

Table D.10 groups the simulation results by the MFs within the membership threshold levels within the noise levels.

Three Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if 0%, 1% and 16% noise data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of noise level types. Alternative Hypothesis H_1: samples are not drawn from the same population of noise level types.

For 0% and 1% noise, the value of the statistic was p = 0.72 $>$ 0.05, so the Null Hypothesis cannot be rejected. For 1% and 16% noise, the value of the statistic was p = 1.32115E-17 $<$ 0.05, so the Null Hypothesis has to be rejected. For 0% and 16% noise, the value of the statistic was p = 8.64582E-18 $<$ 0.05, so the Null Hypothesis has to be rejected.

The best controller for the no noise inputs is T2Tri305, with a RMSE value of 4665.4. For the 1% added noise the T2tri305 is the best with a RMSE value of 4734, and for the 16% added noise input it is again the T2tri305 controller with a value of 19044.7.

The results for varying the applied noise within the membership function used in the controller for the required thresholds are presented in Table D.11.

Three Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if Th1 MF, Th0.9 MF and Th0.8 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of Threshold MF types.

Alternative Hypothesis H_1: samples are not drawn from the same population of Threshold MF types.

For Th1 MF and Th0.9 MF, the value of the statistic was p = 0.98 $>$ 0.05, so the Null Hypothesis cannot be rejected. For Th1 MF and Th0.8 MF, the value of the statistic was p = 0.98 $>$ 0.05, so the Null Hypothesis cannot be rejected. For Th1 MF, Th0.9 MF and Th0.8 MF, the value of the statistic was p = 0.97 $>$ 0.05, so the Null Hypothesis cannot be rejected.

Table D.12 presents the results for varying the threshold within the membership func-

tion used in the controller for each level of applied noise.

6.5 Discussion

The results of the type-2 simulations varying the thresholds and noise levels showed that the best overall controller was the T2Tri305 type-2 trapezoidal triangular controller. The evidence from the simulation shows that varying the threshold levels for the membership functions has little or no effect. In the real world the small difference in the simulation RMSE would be swamped by global noise. Variation in the added noise levels caused the largest variation in the response of the controllers as shown in Table D.10. At all noise levels the T2Tri305 controller with a membership function threshold level performed the best.

6.6 Dual Surface Seven Term and Three Term FLC Simulation Experiments

Having run simulations on the dual surface controller acting as an average single surface type-2 controller a series of simulations was performed to investigate the performance of the dual surface type-2 controller.

6.6.1 Introducing the Dual Surface Type-2 Fuzzy Logic Controller

In conventional type-2 controllers two control surfaces are obtained, one from the lower bounds of interval type-2 defuzzification and one from the upper bounds. The average of the two control surface values is then calculated to generate the crisp controller output [139]. The novel formulation of the dual surface controller is one which switches dynamically between the two surfaces. If the average of the lower and upper control surfaces is used, this novel dual surface controller reduces to a 'conventional' interval type-2 controller. However, it is also possible to implement a more sophisticated controller that makes use of the two surfaces by dynamically switching according to context. These include Average Threshold and Weighted Average mechanisms, other dynamic combina-

tions are also possible. In this way, some of the additional information available from an interval type-2 system is maintained in order to be utilised by the controller. This novel process is termed a 'Dual Surface' (interval) Type-2 Fuzzy Logic Controller.

6.6.2 The Dual Surface Average Threshold Mechanism

The experiments used an average threshold mechanism to determine the controller action. If the error between the measured variable and the set point of the controller was within the average threshold then the averaged dual surface value was returned. If the error was greater than the average threshold the lower surface value was returned otherwise the upper surface was returned. The average threshold level is considered to be a tuning parameter. The algorithm for the Dual Surface Average Threshold mechanism is given below.

```
error = mv - sp
diff = Math.abs(error)
if (diff <= averageThreshold)
  retVal = (LowerSurfaceValue + UpperSurfaceValue) / 2
  //The classic average interval type-2 calculation
else
//The Dual Surface Variation
  if (error > 0)
    retVal = LowerSurfaceValue
    // Too fast - Use lower surface (error > 0)
  else
    retVal = UpperSurfaceValue
    // Too slow - Use upper surface (error <= 0)
```

6.6.3 The Dual Surface Weighted Average Mechanism

Although not used in experiments, a weighted average mechanism is proposed as an alternative to determine the controller action.

```
retVal = WA * LowerSurfaceValue + (1 - WA) * UpperSurfaceValue
```

By dynamically setting the Weighted Average (WA) parameter to 0, 1 and 0.5 the average threshold mechanism can be modelled. However all values of WA between 0 and 1 can be applied $(0 <= WA <= 1)$, providing a more subtle controller action.

6.6.4 Experimental Setup

In this section the idea of the dual surface type-2 fuzzy logic controller is explored through simulation. The average threshold mechanism configuration was used to investigate different membership function sets each having membership function thresholds of 1, 0.9 and 0.8. Within each of these controllers the average threshold parameter was then varied.

6.6.5 Seven Term Dual Surface Fuzzy Logic Controller

The input used in the simulation was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, with no inertia change in the DC motor model. There was no Gaussian noise added to the input signal, so that the performance of the dual surface controller could be accurately monitored. The use of noise at this point of the investigation was obviously considered. Indeed experiments using noise at 1% and 16% were carried out for the various membership functions and average threshold levels for the Dual Surface controller. However these were not reported as the results were just an increase on the reported noiseless RMSEs of the simulations, as the seen in the results of the previous experiment. This was hardly surprising as the dual surface controller uses the same components as the classic average interval type-2 controller. A comparison of the two types of controllers using noise in simulation would effectively come down to comparing the noise generation signals.

The simulation used the minimum implication method. The membership functions were the T2Trap7 set given in Table D.1. Two membership function thresholds were used. The first membership threshold was applied at a membership grade of 1.0. This allowed the full membership function action to be applied across both inputs and the output variables. The second membership threshold was set at 0.8 of the membership grade for

the three variables. So any variable value that generated a membership grade above the threshold was given the membership threshold value. The membership function shapes for the second set of parameters with a threshold setting of 1 are shown in Figure D.1, and for the threshold setting of 0.8 in Figure D.3. For the membership function setting of 1, three runs were made to calculate the RMSE for only the Lower, Upper and the Average surfaces. The simulator was then run in dual surface mode from average threshold values ranging from 0 to 25 plus 1000. At the average threshold value of zero no average interval type-2 calculation is made, unless the error is zero. Only the upper and lower surfaces are used. At an average threshold value of 1000 the average interval type-2 calculation is made, which should be the same as the no dual surface result. The runs were repeated for a membership threshold of 0.8.

6.6.6 Simulation Results for Seven Term Dual Surface FLC

The results of the simulation are given in Table 6.8.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if T2Trap7 Th1 MF and T2TrapTh0.8 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of T2Trap7 Threshold types.
Alternative Hypothesis H_1: samples are not drawn from the same population of T2Trap7 Threshold types.

The value of the statistic was p = 0.94 $>$ 0.05, so the Null Hypothesis cannot be rejected.

The results show that the lower the average threshold used the lower the RMSE. As the average threshold increases the RMSE increases and at a very large average threshold the dual surface controller acts as the equivalent interval type2 controller as expected. Comparing the RMSE for the lower and upper surfaces of the two membership thresholds shows that there is an increase in the lower RMSE for the 0.8 membership threshold. However the upper RMSE shows a decrease. The effect on the average interval RMSE is negligible.

Number of Steps	1000	IMP - Min		
Input	Step Ramp			
MF	T2Trap7			
MF Th	Ave Th	LowerRMS	AveRMS	UpperRMS
1	None	7608.0	6186.2	5076.4
MF Th	Ave Th		DS RMS	
1	0		5111.2	
1	0.5		5136.7	
1	1		5170.9	
1	2		5234.0	
1	4		5318.9	
1	5		5344.8	
1	10		5450.7	
1	15		5508.9	
1	20		5573.1	
1	25		5673.7	
1	1000		6186.2	
MF Th	Ave Th	LowerRMS	AveRMS	UpperRMS
0.8	None	7671.4	6185.1	4999.1
MF Th	Ave Th		DS RMS	
0.8	0		5140.5	
0.8	0.5		5109.9	
0.8	1		5148.6	
0.8	2		5202.3	
0.8	4		5290.8	
0.8	5		5327.9	
0.8	10		5435.3	
0.8	15		5502.7	
0.8	20		5572.1	
0.8	25		5672.8	
0.8	1000		6185.1	

Table 6.8: T2Trap7 RMSE for Dual Surface Average Thresholds within Membership Thresholds

Number of Steps	1000	IMP - Min			
Input	Step Ramp				
MF	T2Trap3				
Run	MF Th	Ave Th	LowerRMS	AveRMS	UpperRMS
510	1	None	6087.0	4928.7	4089.4
MF Th	Ave Th		DS RMS		
511	1	0.5		4449.1	
512	1	1		4384.8	
513	1	2		4406.6	
514	1	4		4419.3	
513	0.9	4		4415.1	
514	0.8	0.5		4450.7	
515	0.8	1		4368.9	
516	0.8	2		4382.4	
517	0.8	4		4404.3	

Table 6.9: T2Trap3 RMSE for Dual Surface Average Thresholds within Membership Thresholds

6.6.7 Three Term Trapezoidal Dual Surface Fuzzy Logic Controller

The input used in the simulation was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, with no inertia change in the DC motor model. There was no added noise added to the input signal. The simulation used the minimum implication method. The membership function parameters are given in Table D.2. Three threshold levels were applied to the membership functions, with values of 1, 0.9 and 0.8. The membership function shapes are shown in Figures D.4 - D.6. For the membership function threshold of 1 the average thresholds were 0.5, 1, 2 and 4. For the 0.9 membership function variables the average threshold was set to 4. For the membership function threshold of 0.8 values of 0.5, 1, 2 and 4 were used. The simulation was again run against all the thresholds.

The results of the simulation are given in Table 6.9.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if T2Trap3 Th1 MF and T2Trap Th0.8 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of T2Trap3 Threshold types.

Alternative Hypothesis H_1: samples are not drawn from the same population of T2Trap3

Threshold types.

The value of the statistic was p = 0.57 $>$ 0.05, so the Null Hypothesis cannot be rejected.

The lower surface RMSE increased with the membership function threshold. The upper surface followed the average threshold except for the 0.5 value.

6.6.8 Three Term Trapezoidal Triangular Dual Surface FLC with Minimum Implication

The input used in the simulation was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 steps, with no inertia change in the DC motor model. There was no added noise added to the input signal. The simulation used the minimum implication method. Three threshold levels were applied to the membership functions, with values of 1, 0.9 and 0.8. The membership function shapes are shown in Figures D.7 - D.9. For the membership function threshold of 1 the average thresholds were 0.5, 1, 2 and 4. For the 0.9 membership function variables the average threshold was set to 0. For the membership function threshold of 0.8 values of 0 and 4 were used. The simulation was again run against all the thresholds.

The results of the simulation are given in Table 6.10.

Student's t-test for two samples with unequal variances for a two tailed option were carried out on the data to test if T2Tri300 Th1 MF and T2Tri300 Th0.8 MF data could be considered to be equivalent to each other.

Null Hypothesis H_0: samples are drawn from the same population of T2Tri300 Threshold types.

Alternative Hypothesis H_1: samples are not drawn from the same population of T2Tri300 Threshold types.

The value of the statistic was p = 0.93 $>$ 0.05, so the Null Hypothesis cannot be rejected.

Again the lower surface RMSE increases as the membership function threshold increases. The upper surface for the membership function thresholds 1 and 0.8 are fairly close together and follow each other.

Number of Steps	1000	IMP - Min			
Input	Step Ramp				
MF	T2Tri300				
Run	MF Th	Ave Th	LowerRMS	AveRMS	UpperRMS
520	1	None	5695.1	4690.5	3994.4
MF Th	Ave Th		DS RMS		
521	1	0.5		4343.4	
522	1	1		4301.8	
523	1	2		4260.3	
524	1	4		4271.3	
Run	MF Th	Ave Th	LowerRMS	AveRMS	UpperRMS
525	0.9	None	5695.1	4690.5	3992.7
526	0.8	None	5701.6	4690.5	3984.7
MF Th	Ave Th		DS RMS		
527	0.8	0.5		4347.4	
528	0.8	1		4311.8	
529	0.8	2		4256.7	
530	0.8	4		4270.9	

Table 6.10: Minimum T2Tri300 for Dual Surface Average Thresholds within Membership Thresholds

	T1TriProd	AveT2tri305
Noise	Ave(SD)	Ave(SD)
0%	4482.6	2840.1
1%	4614.53(128.42)	3025.17(55.10)
16%	20742.46(1621.03)	19324.12(738.77)

Table 6.11: Type1 and Average Interval Type2 RMSE Comparison

6.6.9 Three Term Trapezoidal Triangular Dual Surface FLC with Product Implication

Finally with the dual surface T2tri305 Product controller showing the best results, it was decided to investigate this controller further. As the best RMSE values were given by the version with a membership threshold level of 0.9, so that one was selected for the investigation. As before, the input used in the experiment was the Step Ramp input sequence as given in Table D.5 and Figure D.21. The simulation was run for 1000 sets, with no inertia change applied to the DC motor model. Simulations without noise and with Gaussian noise added at values of 1% and 16% of the controller range input signal were carried out. Also the type-1 T1TriProd controller was investigated for comparison purposes.

AveTh	AveIT2	Lower	Upper	
None	2840.1	4396.8	1893.1	
AveTh	DS RMSE	numAve	numLower	numHigher
0	2754.1	0	460	540
0.5	2754.8	40	439	521
1	2755.5	53	429	518
2	2762.8	73	413	514
4	2757.4	85	408	507
5	2755.4	92	404	504
10	2754.1	134	389	477
15	2619.1	233	343	424
20	2568.5	337	297	366
25	2840.1	419	263	318
1000	2840.1	1000	0	0

Table 6.12: RMSE for the Dual Surface Controller without Noise

The results in Table 6.11 show that the average interval type-2 T2tri305 Product has lower averaged RMSE values and SDs for the three levels of noise investigated. This showing that the average interval type-2 is better than the equivalent type-1 controller at handling uncertainty generated by Gaussian noise.

In Table 6.12 the results of varying the dual surface average threshold from zero to 1000 is reported for the without noise option. Two further simulations were carried to obtain the RMSEs when only the lower and only the upper surfaces were accessed. The RMSE for only the lower surface is 4396.8 and for only the upper surface it is 1893.1. The averaged value is 3144.95 which is higher than the average interval value of 2840.1. The reason that the upper surface has a lower RMSE is due to the input increasing. A decreasing input gives the same values in the opposite variables.

When using the dual surface controller with a range of average thresholds the reported RMSE is consistently below the average interval type-2 controller. The other columns in the table show the number of times the controller spends in the three parts of the controller at each average threshold level.

An average threshold value of 0 uses either the upper or lower surface. This option is better than the average interval type-2 controller giving a RMSE of 2754.1. The best average threshold value is 20 giving a RMSE of 2568.5. It is seen that the number of times spent in each part of the controller is the closest to being equal for this average threshold.

1% Noise		
AveTh	Ave	SD
0	3142.08	180.68
0.5	3131.55	176.34
1	3112.74	111.83
2	3109.66	158.92
4	3116.98	149.82
5	3144.23	117.33
10	3016.54	148.98
15	2944.94	118.77
20	2925.81	120.72
25	3021.03	86.54
1000	3086.45	111.94
16% Noise		
AveTh	Ave	SD
0	19739.62	1124.89
0.5	19965.43	1943.25
1	19317.41	1191.28
2	19838.03	1696.72
4	18989.65	1557.95
5	18910.87	1453.4
10	18707.55	947.28
15	18352.2	1895.91
20	19402.32	2017.05
25	18528.54	2015.03
1000	19356.46	1281.53

Table 6.13: RMSEs for the Dual Surface Controller with Noise

As the average threshold increases the dual surface controller becomes equivalent to the average interval type-2 controller.

The simulation was repeated for noise level of 1% and 16% and the results are given in Table 6.13. At the 1% noise level the average interval controller gave a mean value of 3025.17 with a SD of 55.10. The dual surface controller gave a higher RMSE for average threshold values below 10 and lower RMSEs for values 10 and above.

However the SD of the results was consistently higher than that of the average interval type-2 controller. At the 16% level the average interval controller gave a mean value of 19324.12 with a SD of 738.77. The dual surface controller gave a higher RMSE for average threshold values below 2 and at 20, and lower for the rest. Again though the SD of the dual surface controller was consistently higher than the average interval controller.

6.7 Discussion

From the statistical tests it was observed that there were no statistical differences between the different number of membership function, their structure types and any membership function thresholding that might be applied. This result indicates that any one of them could be used in the real world experiments. However other factors should be considered, such as ease of designing the structure of the membership functions to be used and the number of rules to use. This would indicate that selecting a three term controller would be the better option.

Comparing the type-2 against type-1 controllers without noise and with and without inertia, all the type-2 controllers outperform their type-1 equivalents. The controller rankings in both cases was the same. The worst performing controller was the T1trap7 controller. The best performing controller was T2tri305. The seven term trapezoidal controllers showed the small increase in RMSE when the inertia change was applied.

For 1% noise and with and without inertia change the type-2 controllers outperform their type-1 equivalents except for the seven term controllers. The best performing controllers were T2tri300 and T2tri305. For 16% noise without an inertia change the T2trap3 and T2tri305 out performed their type-1 equivalents, however for the T2tri300 and T2trap7 controllers they were outperformed by their type-1 equivalents. For the with inertia change case all the type-2 controllers were outperformed by their type-1 equivalents, the expectation was that the type-2 controllers would outperform the type-1 controllers.

The results of the type-2 simulations varying the membership function thresholds and noise levels showed that the best overall controller was the T2Tri305 type-2 trapezoidal triangular controller. The evidence from the simulation shows that varying the membership threshold levels for the membership functions has little or no effect. In the real world the small difference in the simulation RMSE would be swamped by global noise. Variation in the added noise levels caused the largest variation in the response of the controllers as shown in Table D.10.

At all three noise levels the T2tri305 controller with a membership function threshold level of 1 performed the best. The RMSE values returned by the T2tri305 controller at a membership function threshold of 1 were the minimum values at all three noise levels.

These occurred in runs 9, 117 and 118. However the values recorded by the T2tri300 controller were extremely close and so the two controllers were considered to be equivalent in terms of expected performance in the real world.

For the dual surface simulations the T2Trap7 controller was the worst performing with the highest RMSEs, with values between 7608 and 7672 for the lower surface and between 5076 and 5319 for the upper surface. Following that the T2Trap3 had the next highest RMSE values. The values ranged between 6087 and 6164 for the lower surface, and between 4089 and 4451 for the upper surface.

Two simulations were carried out to compare the T2Tri300 controller using the implication method Minimum and the T2Tri305 controller using the Product implication method. The T2Tri300 Minimum controller had the next highest RMSE values. The lower surface had a values between 5695 and 5702, and the upper surface had values between 3992 and 4347. The best performing controller was the T2Tri305 Product controller with a lower surface RMSE range between 5575 and 5800 and an upper surface RMSE range between 3699 and 4249.

For the T2trap7 controller with a membership threshold of 1 and average threshold of 0, the upper RMSE was the lowest and increased with increasing average threshold. Increasing the membership threshold to 0.8 and with an average threshold of 4, the lower RMSE increases and the upper RMSE decreases compared to the performance at a membership threshold of 1 and an average threshold of 4.

The T2trap3 controller followed the same pattern. Comparing the membership function thresholds at an average threshold value of 4 shows that the lower RMSE increases and the upper RMSE decreases very slightly. For the T2tri300 minimum controller again the pattern was repeated with the differences between the membership thresholds very small for the comparable average thresholds. Finally for the T2tri305 product controller the previous pattern was again observed. The differences between the membership thresholds was very small for the comparable average thresholds.

Overall it was shown that the best controllers in terms of RMSEs, were the T2tri300 and T2tri305 variations. For T2tri305 the option of using the product implication method slightly outperformed the T2tri300 minimum implication method. The T2tri305 product controller with the lowest average RMSE had a membership threshold of 0.9. This con-

troller was then tested at a range of average threshold values at noise levels of 0% 1%and 16% and compared with the average interval type-2 controller.

This showed that in the no noise case that the dual surface controller outperformed the average interval controller. At 1% noise the dual surface controller was outperformed by the average controller at average threshold levels less than 10. At 16% noise the same result occured below 4. In both cases though, the SD values for the dual surface controller were over twice those of the average interval type-2 controller.

This indicates that on average that the dual surface controller is able to outperform the average interval type controller. However the SDs of the controller cause concern due to the wider variability of the results. With the encouraging results from the simulation testing of the dual surface type-2 T2tri305 Product controller it was decided to used it in the real world experiments.

6.8 Summary

In this chapter two trapezoidal and two trapezoidal triangular type-2 controllers basic were designed and investigated. The first trapezoidal controllers was a seven term controller, T2trap7 and the second was a three term controller T2trap3. The two trapezoidal triangular controllers were three term controllers called T2tri300 and T2tri305. These controllers varied slightly in the structure of the central triangular membership function. T2tri300 had an edge structure which tapered to the apex, T2tri305 had a parallel edged based structure. All the controllers each had three variations, using membership thresholds of 1, 0.9 and 0.8.

Control surfaces were generated for each of the controllers to show the differences between them. Simulations were run for 1000 steps using ramp inputs to compare the type-1 and derived type-2 controllers. The simulations explored the effects of different levels of noise and the effect of changing the inertia of the DC motor model. The simulations were carried out using the three different membership thresholds. This showed that the worst controller was T2trap7, and the best was T2tri305 closely followed by T2tri300.

The average threshold was introduced which investigated the RMSE generated by the lower and upper defuzzified crisp values as well as the classic averaged type-2 crisp

value. Four average threshold values were used 0.5, 1, 2 and 4. The controllers that were investigated using the average threshold mechanism were T2trap7, T2trap3, T2Tri300 and T2Tri305. T2tri300 used the implication rule of Minimum and T2Tri305 used the implication rule of Product.

Simulations were run for the three membership thresholds with the average threshold values for all the controllers. The worst controller was again the T2trap7 and the best was T2tri305 with the Product implication rule. Having determined that the T2tri305 Product controller was the best candidate to test in the real world, it was then tested at a range of average threshold values at noise levels of 1%and 16% and compared with the average interval type-2 controller.

In the next chapter, the novel Dual Surface Type-2 fuzzy logic controller is introduced. The Dual Surface is used to implement the T2tri305 Product controller on a micro robot with limited resources. The Dual Surface controller is compared with the classic average type-2 controller, the equivalent type-1 controller and the PID controller developed in Chapter 4.

Chapter 7

Evaluation of Single and Dual Surfaces in the Real World

7.1 Introduction

In Chapter 6 the effect of using four different basic type-2 controllers and variants on a DC motor was simulated. The simulation allowed a specific controller to be selected for implementation into a micro robot. The main feature of the controller was that it was configured to generated three type-2 defuzzified parameters with which to control the DC motor in the simulation. These parameters were the lower surface defuzzified value, the upper surface defuzzified value and the classic average type-2 defuzzified value. The simulation was run for each parameter under investigation to generate its RMSE for the DC motor model.

The problem that was being investigated was to be able to develop a fire and forget strategy for a micro robot playing robot soccer. The micro robot is required to travel in a straight line as fast as possible for a distance of at least 127cms. During this strategy there are no trajectory correction commands generated or transmitted to the robot. A simple requirement that proved to be very difficult to achieve consistently.

This chapter describes the real world performance of the micro robots in fuzzy and PID controller configurations. The chapter breaks down into three sections. The first section describes how the generation and installing of the dual surface controller was achieved. The method used for accessing the surfaces is described in [140]. The second

section is concerned with experiments that investigate the performance of a type-1, a dual surface type-2 and a classic average type-2 fuzzy logic controller, together with a PID controller in a micro robot. The third section is concerned with the performance of the dual surface type-2 fuzzy logic controller in various internal configurations.

All the experiments were carried out on the robot football pitch, see Chapter 3 The Environment. This provides a consistent environment with a camera to capture movement and enable measurements to be made. The experiments were controlled through the Robot Soccer Engine. (Chapter 3).

A strategy was written to be implemented by the Robot Soccer Engine to control the movement of the robots and to capture their time and position during the run. The user interface for editing and compiling the strategy.cpp file is MS Visual Studio 2005. The set of robots used in the experiments were the blue robots. By compiling the strategy.cpp file a strategy.dll file is generated and stored in a predefined directory (C:\strategy\blue). The Robot Soccer Engine was modified to store the time and position data into an ascii text file. The name of the file was the datetime that the file was generated i.e. MMDDHRMN.txt, so as to prevent any data from being accidentally overwritten. Each robot had the controller .hex image downloaded prior to the start of the experiments using the Pony Prog download program. The robot controllers are completely independent of the Robot Soccer Engine, only obeying bluetooth transmitted command codes received via their comms port. The Robot Soccer Engine allows the user to set up the comms ports on the robots through the bluetooth communications package. Also the comms ports can be setup via the Microsoft Hyper terminal communications package. Which ever method is used then the alternative had to be configured to the same comms port number. The robots were set up by sending parameter configuration commands and code values via the Microsoft Hyper terminal communications package to their comms port. In the experiments the Robot Soccer Engine was used to start, monitor and record the positions of the robots. There were no command signals sent to the robots during the experiments. This was decided upon so that there were no effects due to the controllers having their setpoints altered during the run. Also it was not possible to accurately record the time that the command signal was actioned, or know if any motor clix interrupts were lost during the robot run. Each robot was identified by a coloured cap which has a team colour, blue in these

Run	IMP	MF Th	Ave Th	LowerRMS	T2AveRMS	UpperRMS
520	Min	1	0	5695.1	4690.5	3994.4
525	Min	0.9	0	5695.1	4690.5	3992.7
526	Min	0.8	0	5701.6	4690.5	3984.7
541	Prod	1	0	5575.9	4487.5	3803.0
546	Prod	0.9	0	5654.5	4505.1	3775.7
551	Prod	0.8	0	5799.2	4518.9	3699.1

Table 7.1: Simulation Results for Dual Surface Selection

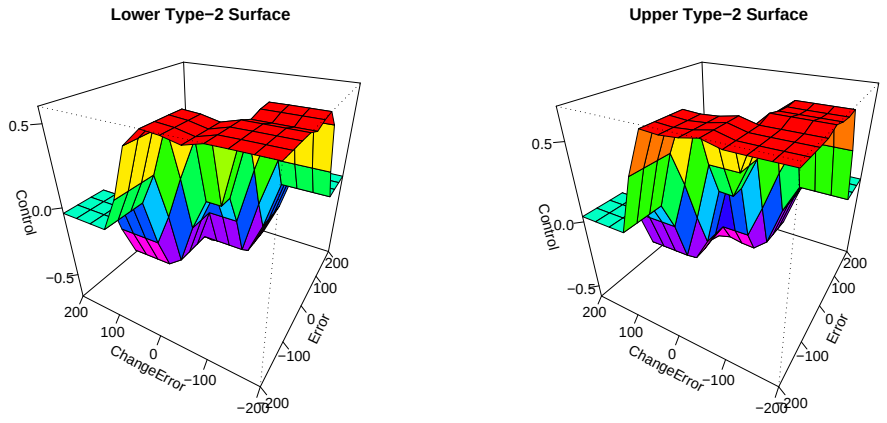
experiments, and a unique player colour combination as defined in Chapter 3. In order to reduce the experimental error, the robots were positioned at the side of the pitch rather than the end and run across the width. This was necessary due to the pitch being made up of two panels with a tape covering the width ways join. When playing robot soccer, for example at slow speeds, the robots could become stuck on the ridge. This would have caused the experimental results to be invalidated had the robots become stuck during the runs.

7.2 How surfaces are generated and how they hold information

The selection of the type-2 fuzzy logic controller to be used in the live micro robot experiments was based upon the RMSE results of the simulations that were performed. The controllers considered are given in Table 7.1.

From Table 7.1 it is seen that the PROD parameter gave a better RMSE and the MIN parameter for the three term trapezoidal triangular membership function shapes. Within the PROD results the membership function with a threshold of one was the best. However it was considered that having no uncertainty in the membership functions at their peaks was unrealistic so the three term trapezoidal triangular membership function shapes with a threshold of 0.9 were selected to produce the surface.

The process to generate the dual surface was to vary the input variable Error of the simulator between -120 and 120 in steps of 20 and within each step vary the input variable ChangeError between -120 and 120 in steps of 20. This gives a 13x13 output for the lower and the upper surface. Also the average type-2 controller surface was generated by the



(a) Lower Surface (b) Upper Surface

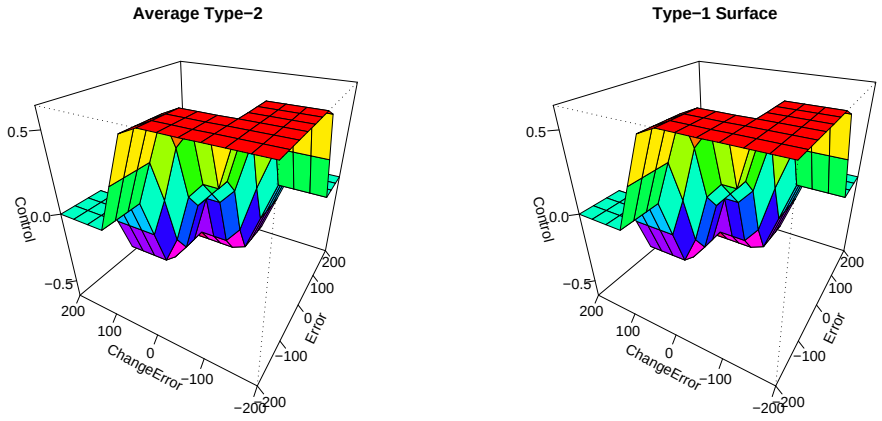
Figure 7.1: Lower and Upper Dual Surfaces

FIS. The upper and lower ranges of the tables are the PWM range values in the micro robot. The values reported in the surface tables are normalised between -1 and 1 so that they can be directly applied in the micro robot software. The lower and upper surface tables that were used are given in Figure 7.1 and Tables E.2 and E.3 respectively.

Initially a step size of 10 was used to generate the tables, however when they were compiled and loaded into the micro robots, the robots failed to respond to commands and had to be reprogrammed. There was no indication as to the cause of the problem except that dumping out the micro robot command table showed that it had been corrupted. The most likely explanation was that the memory had been corrupted by the size of the tables. The values returned in the tables were of type double in order to hold as much information as possible. The surface values for the average type-2 controller and the equivalent type-1 trapezoidal triangular controller are shown in Figure 7.2 and given in Tables E.4 and E.5 respectively.

The difference between the dual type-2 upper and lower surfaces is given in Figure 7.3 and in Table E.6. The table shows that the greatest differences occur between the -40 to 40 range for both the input variables Error and ChangeError. This is mainly where the *cen* membership function shape interacts with the *pb* and nb membership functions, as expected.

In Figure 7.4 and in Table E.7 the difference between the average type-2 surface and



(a) Average Type-2 (b) Type-1

Figure 7.2: Average Type-2 and Type-1 Surfaces

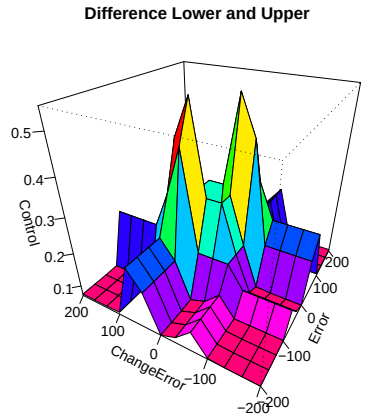


Figure 7.3: Difference Between Upper Surface and Lower Surface

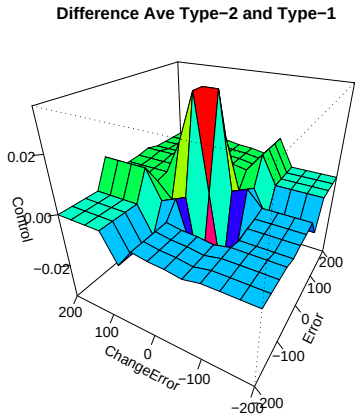


Figure 7.4: Difference between Average Type-2 Surface and Type-1 Surface

the type-1 surface is given. The table shows that the maximum difference between the two normalised surfaces is 0.033719.

7.3 Bilinear Access of Surface Arrays in the Micro Robot

Due to the ease of programming and the popularity of the method, a bilinear interpolation method was programmed into the micro robot to access the 2-D surface arrays of the controllers. Bilinear interpolation is an extension of linear interpolation for interpolating functions of two variables on a grid or 2-D surface. Bilinear interpolation performs linear interpolation in one direction then in the remaining direction. Despite the name and the fact that two linear interpolations occur in the sample values, the interpolation as a whole is not linear but quadratic in the sample location Figure 7.5.

The two input variables used in the micro robot controller were the Error (e) and ChangeError (c) variables. The values for the four surface points closest to the input variables are shown in red, and the Control output variable point P, is shown in green. Knowing the value of the surface, f, at the four points $S_{11} = (e1, c1), S_{12} = (e1, c2), S_{21} = (e2, c1), S_{22} = (e2, c2)$, then the value of the surface f(e,c) at the point P = (e,c) is calculated as follows.

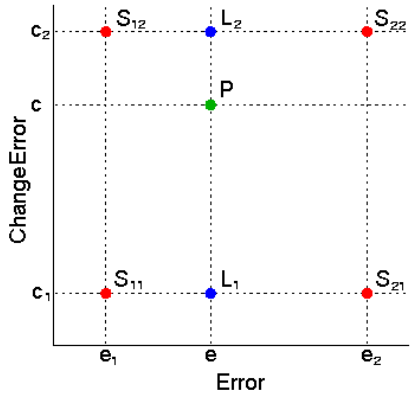


Figure 7.5: Bilinear Interpolation Sample Space

Applying linear interpolation in the e-direction gives

$$f(L_1) \approx \frac{e_2 - e}{e_2 - e_1} f(S_{11}) + \frac{e - e_1}{e_2 - e_1} f(S_{21}) \quad \text{where} \quad L_1 = (e, c_1), \tag{7.1}$$

$$f(L_2) \approx \frac{e_2 - e}{e_2 - e_1} f(S_{12}) + \frac{e - e_1}{e_2 - e_1} f(S_{22}) \quad \text{where} \quad L_2 = (e, c_2) \tag{7.2}$$

Then applying linear interpolation in the c-direction gives

$$f(P) \approx \frac{c_2 - c}{c_2 - c_1} f(L_1) + \frac{c - c_1}{c_2 - c_1} f(L_2) \tag{7.3}$$

The estimate of f(e,c) is given by

$$f(e,c) \approx \frac{f(S_{11})}{(e_2 - e_1)(c_2 - c_1)} (e_2 - e)(c_2 - c) \tag{7.4}$$

$$+ \frac{f(S_{21})}{(e_2 - e_1)(c_2 - c_1)} (e - e_1)(c_2 - c) \tag{7.5}$$

$$+ \frac{f(S_{12})}{(e_2 - e_1)(c_2 - c_1)} (e_2 - e)(c - c_1) \tag{7.6}$$

$$+ \frac{f(S_{22})}{(e_2 - e_1)(c_2 - c_1)} (e - e_1)(c - c_1). \tag{7.7}$$

Choosing a coordinate system in which the four points, where f is known, are (0, 0),

(0, 1), (1, 0), and (1, 1), then the interpolation formula simplifies to

$$f(e,c) \approx f(0,0)\,(1-e)(1-c) + f(1,0)\,e(1-c) + f(0,1)\,(1-e)c + f(1,1)ec. \quad (7.8)$$

Or equivalently, in matrix operations:

$$f(e,c) \approx \begin{bmatrix} 1-e & e \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-c \\ c \end{bmatrix}. \quad (7.9)$$

The interpolant is not linear, it is a product of the two linear interpolations and has the form

$$(a_1 e + a_2)(a_3 c + a_4), \quad (7.10)$$

An alternative form is given by

$$b_1 + b_2 e + b_3 c + b_4 ec \quad (7.11)$$

where

$$b_1 = f(0,0) \quad (7.12)$$

$$b_2 = f(1,0) - f(0,0) \quad (7.13)$$

$$b_3 = f(0,1) - f(0,0) \quad (7.14)$$

$$b_4 = f(0,0) - f(1,0) - f(0,1) + f(1,1) \quad (7.15)$$

In both cases, there are four constants corresponding to the number of data points where f is given. The interpolant is quadratic along all straight lines, except for lines parallel to either the e or the c axes, equivalently if e or c is set constant. Along these lines it is linear. The result of bilinear interpolation is independent of the order of interpolation.

In order to minimise the reprogramming of the micro robot controller from accessing two tables in the dual surface case to only accessing one table for the average type-2 and type-1 controllers, the non dual surfaces were put into both the surface arrays of the controller. This meant that the cycle access time of the micro robot controller was the same whether the dual surface was being used or not. In the non dual configuration a

simple average of the two identical tables is returned as the output variable Control.

7.4 Robot Response Time to a Command

This experiment was designed to verify that the time delay in receiving a Bluetooth transmission is of the order of 40ms for a Miabot Robot.

7.4.1 Background

In a potentially high speed situation, such as Robot Soccer, the time delay to obey a command can make a difference as to the desired outcome. Possibly the ball is missed, or a collision occurs between the robots. This experiment used the robots Light Emitting Diodes (LEDs) and the Robot Soccer Engine (RSE) to determine the response to a command. The advised time delay for a Bluetooth transmission is of the order of 40ms. Provided that the transmission delay is of this order then the effect is considered to containable.

7.4.2 Method

The following modifications were made to the robot system.

- Robot Flash LED Command The command to cause the LEDs to flash is [!]. The robot driver code was modified to light the red LED on receipt of a [!] command. The red LED then remains lit for upto 4 seconds, and is then turned off.

- Robot Soccer Engine Command The Robot Soccer Engine was modified to transmit the [!] command. A timer Class call StopWatch was written which contained methods to capture system time in start and end variables and calculate the difference.

- Robot Soccer Engine Vision Subsystem The Robot Soccer Engine vision subsystem was modified to scan the left gaol area of the soccer pitch to detect the red LED.

The Robot Soccer Engine transmitted the [!] command, recording the start time in StopWatch. The pitch image was scanned by the vision subsystem for the LED being

turned on by the robot. This time was recorded and the difference calculated and displayed on the output window. When the vision subsystem detects that the LED has been turned off by the robot, another [!] command is sent and the process repeated.

7.4.3 Results of the LED Detection Test

The Robot Soccer Engine reported the following data for difference between sending the command and detecting the LED being turned on.

Run No	Diff ms
1	63
2	62
3	63
4	78
5	78
6	63

The camera model used runs at 30 fps or performs an image data capture every 33ms. The exact time difference is also subject to error due to the system time being updated every 10ms.

7.4.4 Conclusion

If the advised time delay of the Bluetooth system is of the order of 40ms then the expected detection of the LED would be in the second or third frame with a time difference between 56ms and 83ms. The results reported by the Robot Soccer Engine are consistent with the Bluetooth transmission delay being in the order of 40ms

7.5 Method for Comparing the Controllers

In this set of experiments three controllers were compared with each other. They were a PID controller, a type-1 fuzzy logic controller and the novel dual surface type-2 fuzzy logic controller. The purpose of the experiment was to establish a baseline set of results and to discover if there were any differences between the controllers when they were used to control a micro robot.

The experiments were carried out on the robot soccer pitch. For each robot controller five strategy files were created. The strategy files only differed in the speed demand setting that was used. The speed demand settings were 10, 20, 40, 80 and 120. Four robots were used in the experiment. This was a precaution, due to reliability problems that had been experienced previously with the robots. It was expected that at least one set of results would be obtained for each set of speed demands and controllers across the set of robots in any of the test runs of the experiment.

The blue bodied robot team was used in the experiment. The robot soccer engine was configured in a left to right playing mode for the purposes of the experiments. The robots were positioned against the bottom length edge of the robot soccer pitch, at approximately 40, 80, 130 and 170 cm from the left goal facing forward with the pitch in front of the viewer. To reduce experimental error the robots were positioned against a maker at these points on the pitch. The actual position of the robots was recorded by the RSE for each test run, so the variation could be investigated if required. The repeatability of the start position was not considered to be important to the results since the travel time was being measured. Selection of the type of controller to use in a robot, was by a transmitted parameter.

Under control of the Robot Soccer Engine (RSE), the robots were command to move forward from their start positions, across the width of robot soccer pitch, at the speed demand setting of the strategy command file. The start command was sent serially to all the robots in the same command cycle within the Robot Soccer Engine. This allowed the response of each of the robots to be compared in case of internal problems within the robots causing systematic errors. The expectation was that the robots would move in a straight line across the width of the pitch, travelling parallel to each other. No other commands were sent and no mid course corrections were allowed. This was to ensure that the experiments were not biased by external commands.

In this set of experiments the dual surface fuzzy logic controller was set to operate as an Average dual surface controller. This is the classic type-2 crisp output, and was used to provide a baseline set of results for the dual surface type-2 controller in a configuration that was well understood.

The surfaces used for the dual surface type-2 fuzzy logic controller was that generated

by the three term trapezoidal triangular membership function shapes with ACT PROD and threshold 0.9 given in Tables E.2 and E.3 respectively. The surface used in the type-1 fuzzy logic controller was the corresponding three term type-1 trapezoidal triangular function shapes, given in Table E.5. The PID parameters used in the PID controller tests were pP = 80, pI = 16, pD = 16. These are the parameters that were established in Chapter 4 - Development of a Baseline PID controller.

In total thirty test runs were made. These consisted of five speed demand strategies for each of the three controller types. The test runs were made with and with out a ramp function being applied. The purpose of the ramp function was to smooth the controller output especially at the higher speed demand settings. This was necessary due to the robots careering off line when high speed demands were directly applied. The number of runs in each test run was nominally 10. If an obvious failure occurred during a run then that run was repeated. For the tests of the type-1 and type-2 fuzzy logic controllers, robot number 4 always used the PID controller with the above parameters of pP = 80, pI = 16, pD = 16. The purpose of having a micro robot with a PID controller was to provide a test run comparison for each run within the test speed setting.

Robot Responses during Runs

Generally the robots travelled across the width of the robot soccer pitch in a straight line without colliding with an other robot, or hitting the sides before the minimum acceptable run distance was achieved. However during the runs individual robots failed to start, or at low speeds stalled. At high speeds the robots became unstable and went round in circles or crashed into the sides of the arena. During the runs robots would collide with each other. There was no supervisory positioning commands given by the RSE, once a robot started it continued along the trajectory. Periodically the wheels became loose and disengage from the motor drive resulting in the robot driving differentially and travel in a circle. After each run the robots were checked for wheel problems and if found the runs were repeated. Genuine control failures often exhibited the characteristics of loose wheels so care had to be taken on this issue.

7.5.1 Results for Comparing the Controllers

The necessary distance travelled to be included in the results was to pass the 127 cm point on the robot soccer pitch. However if the robot passed the 150 cm point then this point was used in the calculation of the average speed of the robot. Between 127 and 150 cm the stop distance was used. A Pythagorean adjustment was made to give a straight line speed for the robots from their start position. The justification for this was that sometimes the robot would slew at the start and then travel in a straight line past the 127 cm criteria. In some of the runs robot number 3 just failed to respond at all to any of the commands. This also happened with robot number 4 but not as frequently. This is designated by ** in the averages tables. The initial expectation of the robots was that they would not usually travel in straight lines along their intended trajectories and that some correction would be needed to be programmed into the strategy to ensure that deviations were corrected or compensated for. However there were runs where the robots travelled along the intended trajectories, or carried on in a straight trajectory after an initial slew.

An investigation as to why the failures were occurring was carried out. The robots were all reloaded with the control image and set up using a hyper terminal file. This was to prevent errors from being introduced. Each robot was given fully charged batteries and then moved under hyper terminal control to check that it was receiving signals via the bluetooth communications system. They were then put under RSE control and the tests were executed. However the problem still remained. Each robot was then driven singularly and moved as command. The only explanation was that there was a problem with bluetooth communicating with all the robots. However it was intermittent. When running the robots in the football configuration, bluetooth commands are sent continually to all the robots. So if a single command was missed, the effect would have been negligible. However in the experiment the start command is only sent once to each robot. If it was not sent then the robot would not move. To my knowledge there were never instances of a robot not obeying a received command, in all the time that I used them.

The tests carried out for the comparison of the three controllers, PID, type-1 and average dual surface type-2 fuzzy logic controllers are given in Table E.8. The results obtained for the comparison of the three controllers are given in Table E.9. The means and standard deviations for the three controllers are given in Table E.10 and for the speed

demands in Table E.11.

Within the results it is expected that for the same speed demand from each controller the actual speed obtained would be the same. The speeds achieved for the ramped option would be expected to be lower than that of the no ramp option, especially at the higher speed demand value. The results show that the PID controller delivers a higher actual speed for the same speed demand when compared against the two fuzzy logic controllers. This is not considered to be a problem since the fuzzy logic controllers were using the same internal scaling parameters that the PID controller was set up to use. In the three controller version of the micro robot used in these experiments, it was decided not to change the parameters from the PID settings. This was done so that the baseline results can be used across all the experiments if required. Obviously if the same actual speed was required from the fuzzy controllers as that delivered by the PID controller then a calibration exercise would be required. A t-test was carried out on the mean speeds of the controllers to test the null hypothesis that there was no difference between the means against the alternative hypothesis that there is a difference between them. This showed that there was no significant difference between the means of the fuzzy logic controllers but there was a significant difference between the PID controller and the two fuzzy logic controllers. This was partly expected due to the scaling issue previously discussed. The differences between the means of the two fuzzy logic controllers is very close. Again this is expected since the type-2 controller was generated from the type-1 controller by blurring the membership functions. The only anomaly is in tests 21 and 27 for the no ramp option of the type-2 controller. At speed demands of 80 and 120 there is virtually no difference between the mean values of the actual speed. However the standard deviations reported show a much larger spread of data at these values. At the start of the runs the controllers would be operating at the edges of the controller surface which was limited between -120 and 120 for both the error and change error input variables. An explanation of this anomaly is that the 120 speed demand has saturated the controller and so it delivers the best performance which is at the 80 speed demand. In the corresponding ramp tests 22 and 28, a small difference in the actual speeds is seen, where the change in speed demand is only small in the ramp when compared to the 120 speed demand step change. Having completed the three controller comparison experiment and established that there were no

differences between the fuzzy logic controllers the next experiment was to investigate different options in the novel dual surface type-2 fuzzy logic controller.

7.6 Introduction to Dual Surface Type-2 Controller

In this set of experiments three variations of the dual surface type-2 twin surface controller were investigated and compared with each other. The three controllers were the threshold dual surface, the minimum dual surface and the maximum dual surface type-2 controllers. The purpose of the experiment was to discover if there were any differences between the controllers when they were used to control a micro robot.

7.6.1 Method for The Threshold Dual Surface Controller

The setup used in the dual surface type-2 controller investigation was the same as that used for the comparison of the three controllers. Prior to starting the experimental runs, the common robot control image was download to all the robots using the PonyProg download utility. Again four robots were used in the experiment. A threshold configuration file was created with the parameters set as follows, .pF=2 .pZ=2. This configured the robots to use the threshold option. The threshold parameter was set to seventy (.rT=70) in the configuration file. The configuration file was downloaded to three of the robots using the Microsoft Hyper terminal communications package. An average configuration file was created with the parameters set as follows, .pF=1 .pZ=2. This file was downloaded to the fourth robot. Two tests of the threshold dual surface controller were ran. The first test experiment was run without ramping being applied by setting the parameter .rA=0. So the robot controllers had to handle a straight step change from a standing start. Three robots were used to try and counter the reliability problems that had previously been observed. The aim was to get at least ten valid runs across the three robots for every test. By setting the fourth robot as an average dual surface type-2 controller it provided a control and it was possible to directly compare the threshold results with the results for a baseline robot as soon as the run was completed. The second test experiment was to run with ramping being applied by setting .rA=1. This was done directly through the Microsoft Hyper terminal for each of the four robots.

Having set up the robots the Robot Soccer Engine was used to control the runs. Again five sets of test runs were made one for each speed demand of 10, 20, 40, 80 and 120. If any obvious failures occurred, such as collisions, then the run was repeated. There was no supervisory commands made during the runs other than the start and stop commands.

7.6.2 Results for the Threshold Dual Surface Controller

The same criteria that was used in the three controller comparison experiments was used to determine if a run was valid. Also the Pythagorean adjustment was made to the results if the robot had initially slewed and then travelled on a straight trajectory. For the tests of the threshold dual surface controller it was found that the controller could not control a step speed demand of 120 on any of the threshold robots. Many extra runs were made when a step was applied, with robots colliding with each other and hitting the sides before the 127 cm minimum distance was achieved. When a ramp was applied for the speed demand of 10, the controller could not move the micro robots for the minimum distance of 127 cm without the micro robots stalling at some point in the journey making the speed calculations inaccurate. The results obtained from the test runs are given in Tables E.12 and E.13. In the threshold dual surface controller test number 102 for speed demand 20, the average dual surface control robot four failed to record any valid runs. So the results of the three controller comparisons from Table E.10, for the average dual surface type-2 controller, test 3, were used as the control.

The mean and standard deviations for the dual surface controller with a threshold value of 70 across the three robots running without a ramp are given in Table E.14.

The corresponding results for the average type-2 controller are given in Table E.15.

The results show that the average type-2 controller produces a higher speed for the speed demand in three of the four speed tests. This result was rather disappointing given that in simulation the dual surface type-2 controller outperformed the average type-2 controller. Examining the results show that micro robot number three was consistently under performing when compared with the average speed of the other three robots. Studying the position data of all the robots, it was observed that there were times when the reported position had not altered. These mini stalls were seen more frequently in robot three compared to the others and would explain why a lower speed was obtained. This difference

had not been observed during the baseline comparisons. However the robot had experienced problems due to loose wheels and that they might have been overtightened when attached to prevent them from falling off. During the tests it was not noticed that robot three was usually behind the others, as would be expected from the results. For the threshold test with no ramp applied, the global means are lower except for the speed demand 20 runs in test number 102. The global means are all within one standard deviation of the mean of the average dual surface control robot four. Removing the poor performing robot three from the comparison does not change the number of global means that are lower, but moves all the means close to the mean of the control. For the ramp test the global means are lower except for the speed demand 40 runs in test number 111. Again removing robot three from the comparison does not alter the result. With the results so close this allows the claim that the threshold dual surface controller is a safe and creditable alternative to the classic type-2 fuzzy logic controller when used to control micro robots.

7.6.3 Method for the Minimum and Maximum Dual Surface Controllers

Having completed the threshold tests the robot's configuration was changed to operate as a minimum dual surface controller by setting .pF=3, using the Microsoft Hyper terminal communications package. Only the ramp option was used in these tests with speed demands of 20, 40, 80 and 120. This was due to the problems experienced during the threshold tests, with the response to the step change. When completed the robot's configuration was altered to operate as a maximum dual surface controller by setting .pF=4, again using the Microsoft Hyper terminal communications package. The results of the two tests are given in Tables E.16 and E.17. In the minimum dual surface controller test for speed demand 20, the average dual surface control robot four failed to record any valid runs. So the results from the equivalent maximum test for robot four were used.

7.6.4 Results for the Minimum and Maximum Dual Surface Controllers

The results show that there is a distinct difference between the means of the minimum and maximum surfaces, however they are not significantly different when a t-test was applied. The global means of the minimum dual surface controllers are all less than the global means of the maximum dual surface controllers. Comparing the global means of the minimum dual surface against the mean of the average dual surface control robot shows that the minimum global means are less than the means of the average dual surface control. This is also true for the three individual robots except for the speed demand of 80 runs for robot one. In this case the difference is $1.93 cmsec^{-1}$ with a standard difference of 1.844 for robot one and 2.76 for the control. The global means of the maximum dual surface controllers are not greater than the mean of the control. The differences are all within one standard deviation. If the poor performing robot three is removed from the comparison, then individually the means of the maximum dual surface controllers do exceed the control mean, except for the speed demand of 20 runs for robot two.

7.7 Comparison of Three Controllers

In this experiment three controllers were compared with each other. They were a PID controller, a Type-1 controller and the dual surface average Type-2 controller. The purpose of the experiment was to discover if there were any differences between the controllers when they were used to control a micro robot.

7.8 Method

The experiments were carried out on the robot soccer pitch. Three test runs were performed in the experiment. Each test run had four robots running the strategy. The first test run consisted of two robots running the PID controller and two robots running the Type-1 controller. The second test run consisted of two robots running the dual surface average Type-2 controller and the other two robots the Type-1 controller. In the third test run two robots ran the PID controller and the other two robots the dual surface average

Type-2 controller. This was a precaution, due to reliability problems that had been experienced previously with the robots. it was expected that at least one set of results would be obtained for each set of speed demands and controllers across the set of robots in any of the test runs of the experiment.

The blue robot team was used in the experiment. The robots were positioned against the bottom edge of the robot soccer pitch, at approximately 40, 80, 130 and 170 cm from the left goal facing forward. The robot soccer engine was configured in a left to right playing mode for the purposes of the experiment. The actual position of the robots was recorded by the RSE for each test run.

Selection of the type of controller to use in a robot, was by a parameter which could be altered through a hyper terminal connection using bluetooth. When the robots were using the PID controller the PID parameters were set to pP = 80, pI = 16, pD = 16.

Under control of the Robot Soccer Engine (RSE), the robots were command to move forward from their start positions, across the width of the football pitch, at the initial speed demand setting of 60 in the strategy command file.

The number of runs in each test run was nominally 10. If an obvious failure occurred during a run then that run was repeated.

7.8.1 Observations

During the runs individual robots failed to start, or at low speeds stalled. There was no supervisory positioning commands given by the RSE, once a robot started it continued along the trajectory.

7.9 Results

The necessary distance travelled to be included in the results was to pass the 150 cm point on the robot soccer pitch. However if the robot passed the 150 cm point then this point was used in the calculation of the average speed of the robot. A Pythagorean adjustment was made to give a straight line speed for the robots from their start position. Table E.19 gives the results for each robot running the three different controller types.

Speed	K-W Chi-Squared Statistic	df	p-value	p>0.05 Reject H_0
10 NR	27.0639	2	1.328e-06	Yes
20 NR	4.4015	2	0.1107	No
40 NR	8.2149	2	0.01645	Yes
80 NR	0	1	1	No
120NR	5.1429	1	0.02334	Yes
10 R	4.0039	2	0.1351	No
20 R	19.6276	2	5.469e-05	Yes
40 R	8.2149	2	0.01645	Yes
80 R	6.7276	1	0.009493	Yes
120R	8.25	1	0.004075	Yes

Table 7.2: Hypothesis Tests for PID Controllers

The average speed and standard deviation for each controller across all the test runs of the robots is given in Table E.20

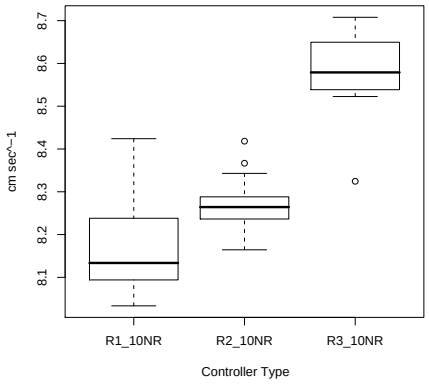
The results show that the controllers all achieve an average speed that is very close to each other. This indicates that for the micro robots that any of the three controllers can be used to successfully control the robots.

7.10 Statistical Analysis of the Real World Robot Controllers

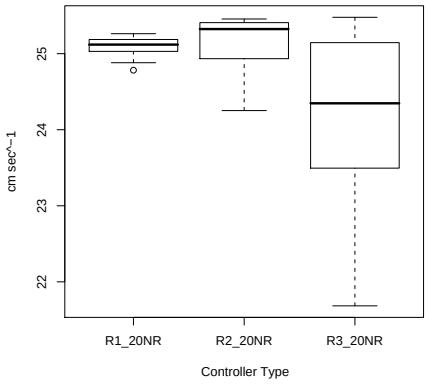
A statistical analysis of the results obtained from running the controllers was carried out using using the statistical analysis package R Version 2.8.1(2008-12-22). The results are presented in three groups. The first group is the results from running the robots using the PID controllers to control them. The second group was the performance of the different controllers against each other. The third group of results is the comparison of the dual surface interval type two controller against the classic average interval type-2 controller.

7.10.1 First Group Analysis - PID Controllers

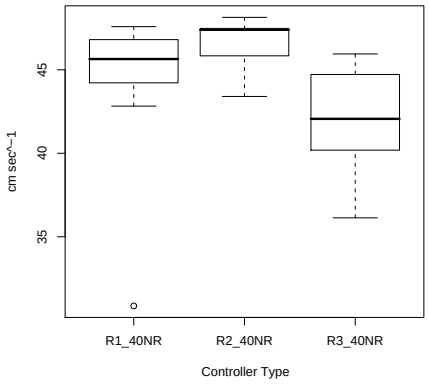
The statistical analysis of the first group, the PID controllers for three robots with and without a ramp being applied to the input, was performed. The results are graphically illustrated as box-and-whisker plots in Figures 7.6 and 7.7.



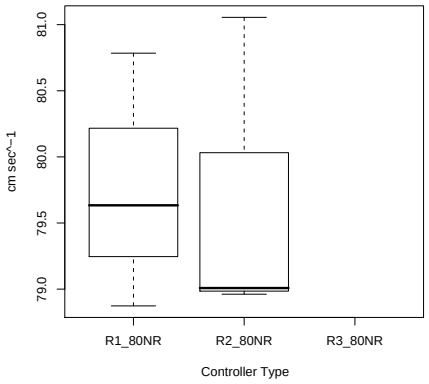
(a) Speed 10



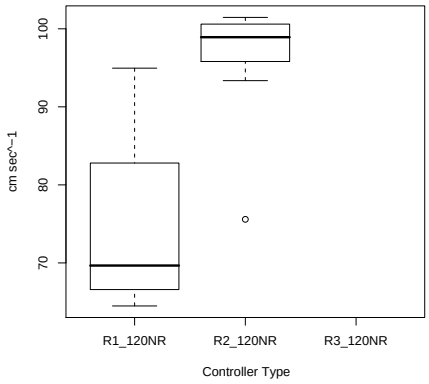
(b) Speed 20



(c) Speed 40



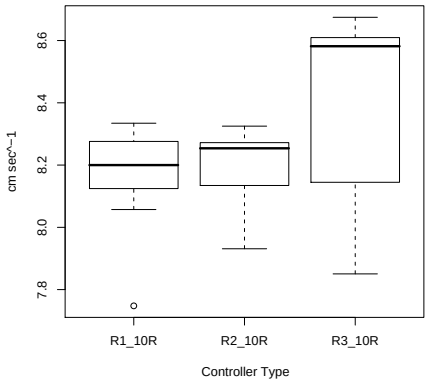
(d) Speed 80



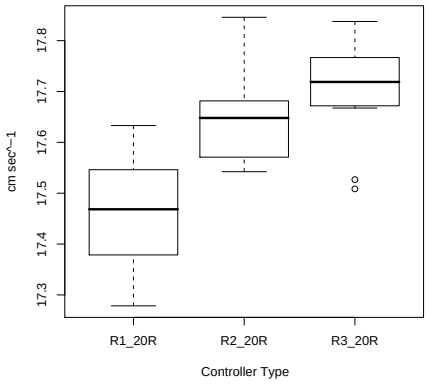
(e) Speed 120

Where Rn_sssNR is robot x at speed sss with No Ramp applied

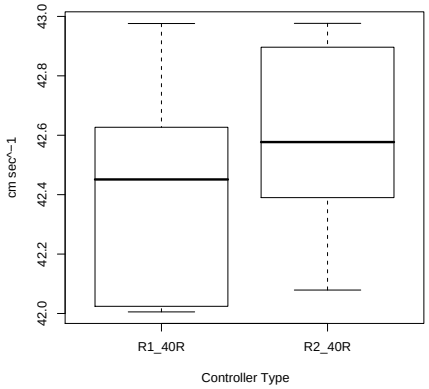
Figure 7.6: Box Plots of PID Controller Response without Ramp



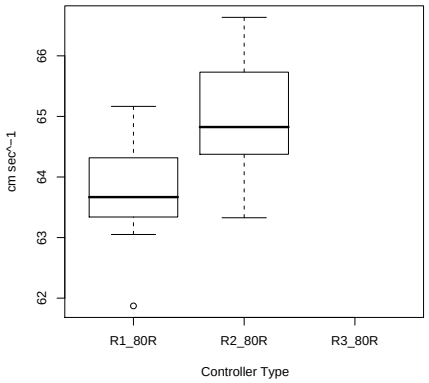
(a) Speed 10



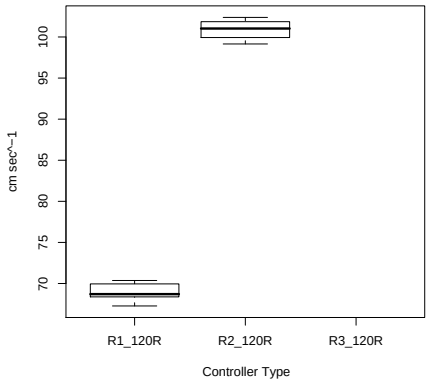
(b) Speed 20



(c) Speed 40



(d) Speed 80



(e) Speed 120

Where Rn_sssR is robot x at speed sss with Ramp applied

Figure 7.7: Box Plots of PID Controller Response with Ramp

Kruskal-Wallis rank sum tests were carried out on the data to test if the three robots could be considered to be equivalent to each other across a range of speed demands.

null Hypothesis H_0: samples are drawn from the same population of robots.
alternative Hypothesis H_1: samples are not drawn from the same population of robots.

The statistical analysis given in Table 7.2 for the first group, shows that for a value of $p > 0.05$, the non ramped inputs for speeds demands of 20 and 80 the robots are not statistically significantly different, for the others three speeds they are. The ramped inputs for speed demand of 10 the robots are not statistically significantly different, for the other four speed demands they are. These results suggest that at low speed demands using a PID controller the robots can be considered to be equivalent. At the higher speed demands the behaviour of the robots diverge. There is also a large variation within each robot's set of runs indicating that the PID controller struggled to maintain its demanded speed, especially at high speed. The difficulty of obtaining results over the qualifying distance at higher speed demands indicates that the PID robot controllers were becoming unstable at the higher speed demands.

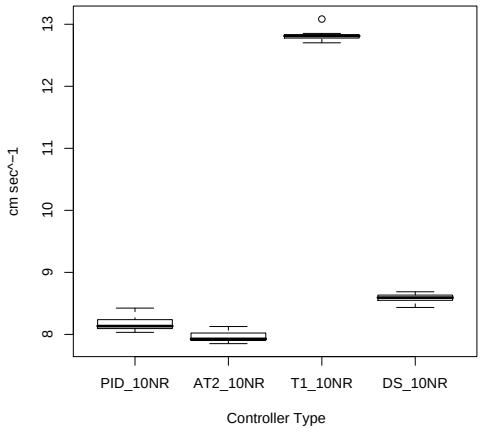
7.10.2 Second Group Analysis - All Controllers

The second group results are those for comparing four different controllers. The controllers were PID, type-1, average interval type-2 and the dual surface controller. The results initially are reported for robot R1. This is due to the previous result that showed at the higher speed demands the performance of the robots diverged. Figures 7.8 and 7.9 show the box-and-whisker plots for the data.

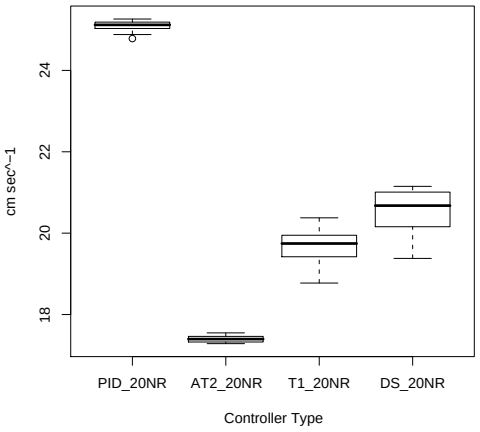
Kruskal-Wallis rank sum tests were carried out on the data to test if the four controllers could be considered to be equivalent to each other across a range of speed demands on the same robot.

null Hypothesis H_0: samples are drawn from the same population of controllers.
alternative Hypothesis H_1: samples are not drawn from the same population of controllers.

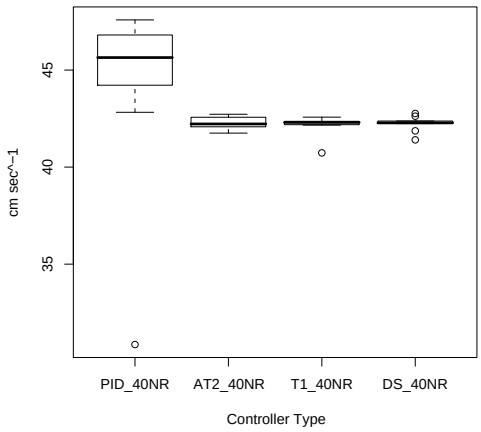
The statistical analysis given in Table 7.3 for the second group, shows that for a value of $p > 0.05$, all the controllers are statistically significantly different from each other at



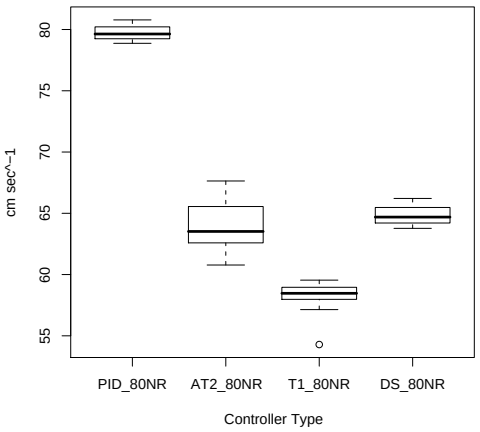
(a) Speed 10



(b) Speed 20



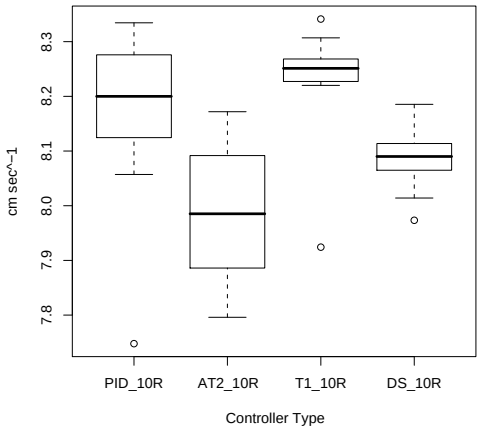
(c) Speed 40



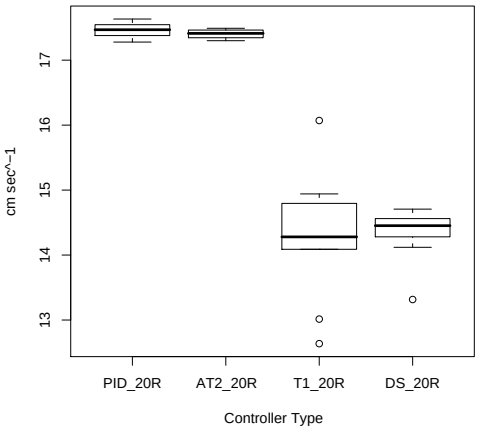
(d) Speed 80

Where PID_ssNR is the PID controller at speed ss with No Ramp applied.
AT2_ssNR is the Average type-2 controller at speed ss with No Ramp applied.
T1_ssNR is the Type-1 controller at speed ss with No Ramp applied.
DS_ssNR is the Dual Surface controller at speed ss with No Ramp applied

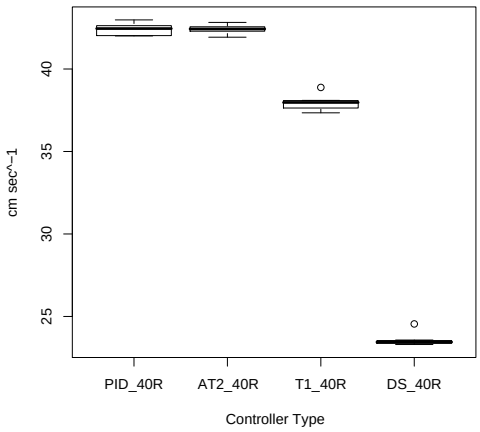
Figure 7.8: Robot 1 Controller Response without Ramp



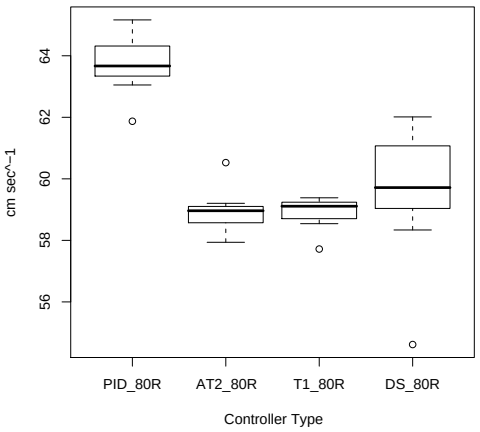
(a) Speed 10



(b) Speed 20



(c) Speed 40



(d) Speed 80

Where PID_ssR is the PID controller at speed ss with Ramp applied.
AT2_ssR is the Average type-2 controller at speed ss with Ramp applied.
T1_ssR is the Type-1 controller at speed ss with No Ramp applied.
DS_ssR is the Dual Surface controller at speed ss with Ramp applied

Figure 7.9: Robot 1 Controller Response with Ramp

Speed	K-W Chi-Squared Statistic	df	p-value	p>0.05 Reject H_0
10 NR	39.4325	3	1.405e-08	Yes
20 NR	39.2513	3	1.535e-08	Yes
40 NR	13.9649	3	0.002953	Yes
80 NR	20.4769	3	0.0001352	Yes
10 R	19.7302	3	0.0001931	Yes
20 R	30.3551	3	1.162e-06	Yes
40 R	29.961	3	1.406e-06	Yes
80 R	25.3166	3	1.326e-05	Yes

Table 7.3: Hypothesis Tests for PID,AveT2,T1 and DST2 Controllers on Robot 1

each of the demand speed inputs. This is in line with the expectation that they would be different.

The box-and-whisker plots show a large spread of mean observed speeds. For the no ramp case in Figure 7.8 the best grouping is for the 40 Speed demand. Here the PID has the largest variation in observed speed. At the 80 Speed demand the Fuzzy controllers still operate close to each other. The PID controller has started to go out of control. It is seen that in all four plots the the dual surface type-2 controller is equal to or faster than the classical average type-2 controller, in line with expectation. For the with an acceleration ramp case in Figure 7.9 the results are more varied. The effect of slowly applying power to the motors obviously distorts how quickly the robots are able to overcome stiction, thus effecting the average speed. At 80 Speed demand the means are closely grouped again with the dual surface type-2 controller out performing the two others. The PID controller again going out of control. As an aside using a step input rather than a ramp produces a better response for a given speed demand.

7.10.3 Second Group Analysis - All robots and controllers

A further analysis was made on the results of all the robots used in the four controller comparison. An equal total sample size of six results was randomly selected from all three robots involved in the tests for each controller type at demand speed inputs of 10,20,40 and 80. The purpose of using the data from all three robots to check if there was any effects that were common to all three robots. Figure 7.10 shows the box-and-whisker plot

Speed	K-W Chi-Squared Statistic	df	p-value	p>0.05 Reject H_0
10 NR	16.1667	3	0.001048	Yes
20 NR	15.5	3	0.001436	Yes
40 NR	16.3667	3	0.0009536	Yes
80 NR	16.2667	3	0.0009998	Yes

Table 7.4: Hypothesis Tests for PID,AveT2,T1 and DST2 Controllers across Robots

of the data.

Kruskal-Wallis rank sum tests were carried out on the data to test if the four controllers could be considered to be equivalent to each other across a range of speed demands.

Null Hypothesis H_0: samples are drawn from the same population of controllers. Alternative Hypothesis H_1: samples are not drawn from the same population of controllers.

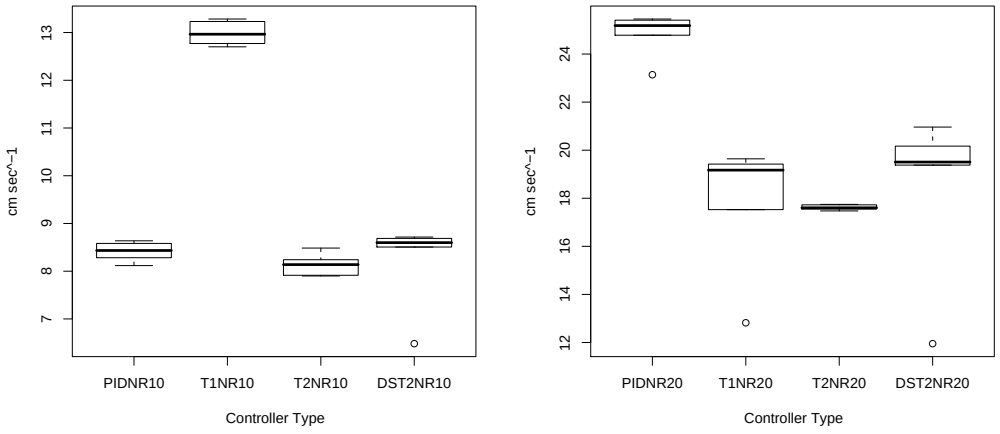
The statistical analysis given in Table 7.4 shows that for a value of $p > 0.05$, all the controllers are statistically significantly different from each other at each of the demand speed inputs. This result confirms the results of the experiments that were performed on the single robot. In Figure 7.10 across all the robots, there appears to be no discernible pattern in the Speed demand responses, except that the PID controller is going faster. Except for the 10 Speed demand the means of the Fuzzy Logic Controllers are close but significantly different.

7.10.4 Third Group Analysis - Type-2 Fuzzy Logic Controllers

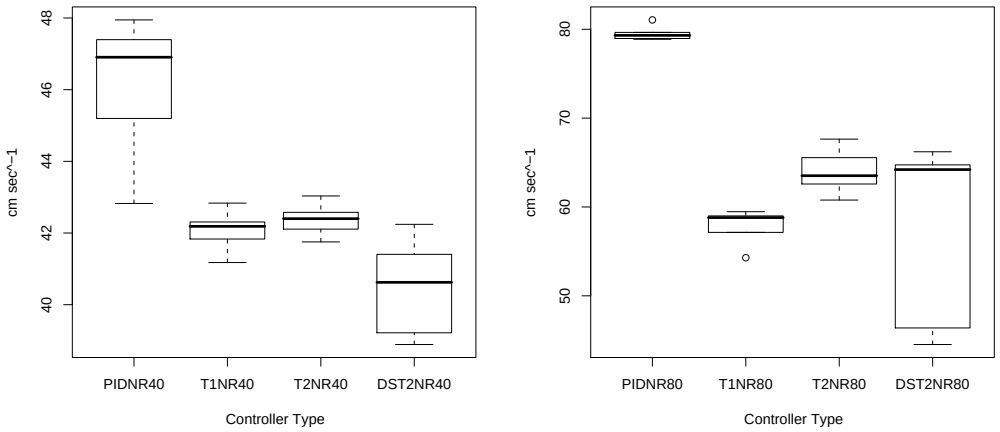
The third group of results are those obtained from running the robots with the dual surface type two controller with a threshold of 70 and the minimum and maximum configurations. These were compared with the corresponding classic average interval type-2 controller. The results are shown in Figures 7.11 and 7.12 as box-and-whisker plots of the data.

Kruskal-Wallis rank sum tests were carried out on the data to test if the four controllers could be considered to be equivalent to each other across a range of speed demands.

Null Hypothesis H_0: samples are drawn from the same population of controllers. Alternative Hypothesis H_1: samples are not drawn from the same population of con-



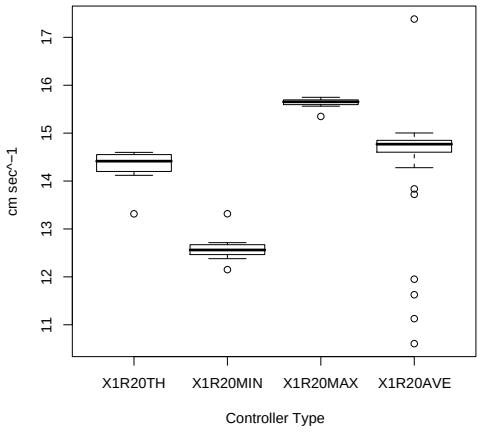
(a) Speed 10 (b) Speed 20



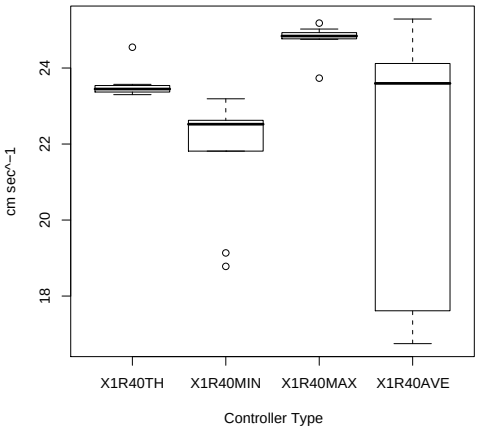
(c) Speed 40 (d) Speed 80

Where PIDNRss is the PID controller with No Ramp applied at speed ss.
T1NRss is the Type-1 controller with No Ramp applied at speed ss.
T2NRss is the average Type-2 controller with No Ramp applied at speed ss.
DST2NRss is the Dual Surface Type-2 controller with No Ramp applied at speed ss.

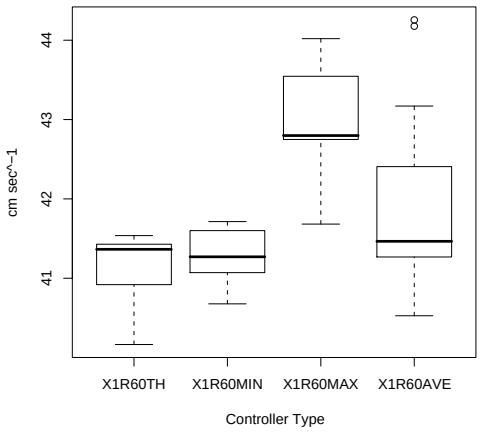
Figure 7.10: All Controllers Response at Speeds 10,20,40 and 80 without Ramp



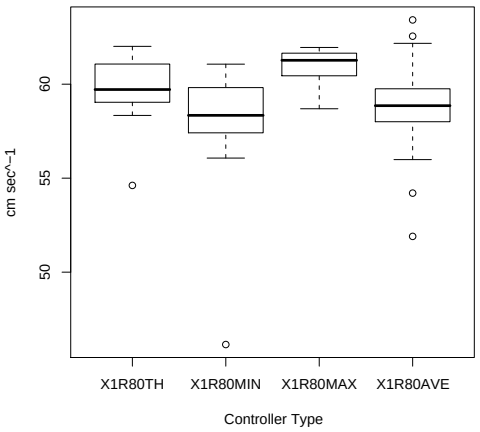
(a) Speed 20



(b) Speed 40



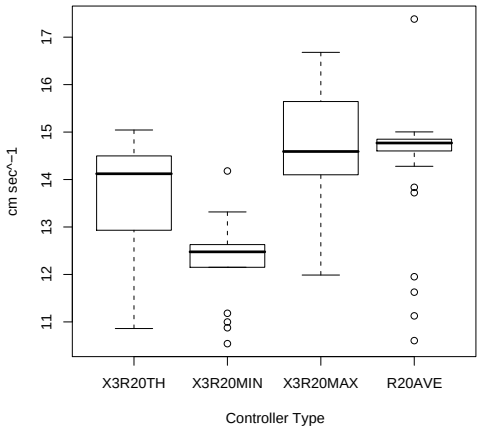
(c) Speed 60



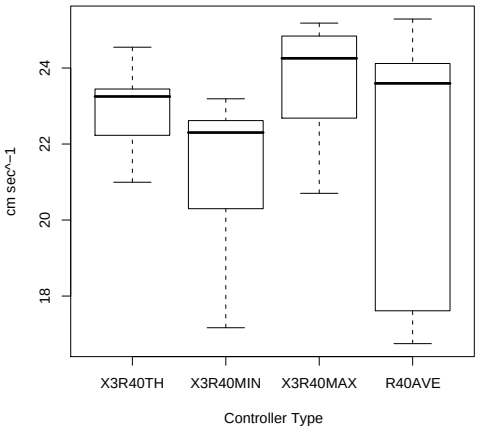
(d) Speed 80

Where X1RssTH is the Robot 1 Dual Surface Threshold controller at speed ss.
X1RssMIN is the Robot 1 Dual Surface Minimum controller at speed ss.
X1RssMAX is the Robot 1 Dual Surface Maximum controller at speed ss.
X1RssAVE is the Robot 1 Average Type-2 controller at speed ss.

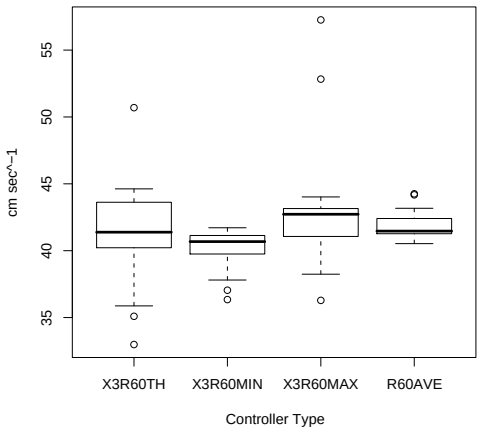
Figure 7.11: Dual Surface and Average Response for Robot 1 only with Ramp



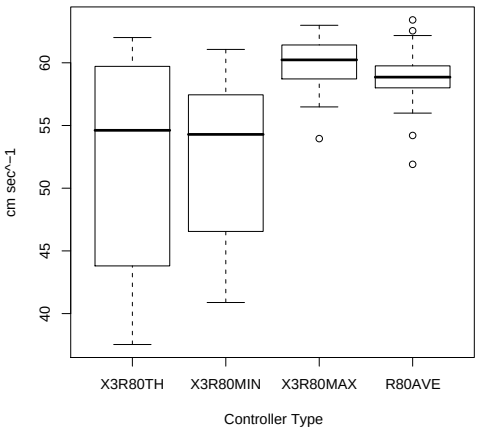
(a) Speed 20



(b) Speed 40



(c) Speed 60



(d) Speed 80

Where X3RssTH is the 3 Robots Dual Surface Threshold controller at speed ss.
X1RssMIN is the 3 Robots Dual Surface Minimum controller at speed ss.
X1RssMAX is the 3 Robots Dual Surface Maximum controller at speed ss.
RssAVE is the Average Type-2 controller at speed ss.

Figure 7.12: Dual Surface and Average Response for 3 Robots with Ramp

Speed	K-W Chi-Squared Statistic	df	p-value	p>0.05 Reject H_0
20 R	32.3282	3	4.463e-07	Yes
40 R	18.972	3	0.0002771	Yes
60 R	21.5151	3	8.228e-05	Yes
80 R	10.3967	3	0.01548	Yes

Table 7.5: Hypothesis Tests for Dual Surface and Average Response for Robot 1 only with Ramp

Speed	K-W Chi-Squared Statistic	df	p-value	p>0.05 Reject H_0
20 R	31.6089	3	6.327e-07	Yes
40 R	18.1194	3	0.0004156	Yes
60 R	16.5708	3	0.000866	Yes
80 R	23.5926	3	3.038e-05	Yes

Table 7.6: Hypothesis Tests Dual Surface and Average Response for 3 Robots with Ramp

trollers.

The statistical analysis given in Tables 7.5 and 7.6 for the two type-2 fuzzy logic controllers, shows that for a value of $p > 0.05$, all the controllers are statistically significantly different from each other at each of the demand speed inputs. This was the expected result.

In Figure 7.11 for the single Robot 1 the behaviour of the robot is seen to be consistent. For each speed demand the lowest mean response is given by the minimum dual surface controller.The highest mean response is given by the maximum dual surface controller. The threshold dual surface controller lies between the two and is very close to the classic average type-2 controller. This result demonstrates the extra properties of the dual surface controller when compared to the classic average type-2 controller. Considering all the noise that is in the experimental environment, the range of the results are very small showing that there is no loss of characteristics when the dual surface controller is compared to the classic average type-2 controller. In Figure 7.6 when comparing the controllers across three robots, the same properties are seen as for the single robot. As expected the variation is greater, however the mean values are close especially for the 60 Speed demand, demonstrating the stability and repeatability of the dual surface controller.

7.11 Discussion

In order to test the dual surface type-2 controller, in all over eight hundred runs were made. The dual surface type-2 controller was configured and tested in four options. The first option was the average dual surface controller. This is the equivalent of the classic type-2 fuzzy logic controller where the crisp control value is generated internally in a fuzzy logic controller as an average of the defuzzification process of the footprint of uncertainty. The second option was the threshold dual surface controller. This option allows the characteristics of the controller to be altered within the dual surface frame work. Exceeding the threshold value caused the controller to use the maximum or minimum dual surface value dependent on the required action. Within the threshold the average dual surface value was used. The third and fourth options were to use the minimum and maximum values of the dual surfaces. Generally the average, minimum and maximum dual surface crisp values are weighted values from the lower and upper surfaces of the controller.

With the results so close this allows the claim that the dual surface type-2 controller is a safe and creditable alternative to the average type-2 fuzzy logic controller when used to control micro robots. The results also show that by using the novel dual surface controller in a micro robot that a classic type-2 fuzzy logic controller can be implemented using the average dual surface option.

7.12 Summary

The T2tri305 Product controller identified in the previous chapter was used as the basis of the dual surface controller. Input variables Error and ChangeError were varied against each other across their ranges. The resulting Control output variable array was stored into its constituent lower and upper surface arrays for the entire output range. A type-1 surface was also created from a type-1 controller, using the same method. A bilinear interpolation method was implemented which allowed the lower and upper surface arrays to be accessed in situ on the micro robot. An experiment was conducted to check the response time of the robot to a command issued from the RSE or Hyper Terminal which controlled the robot. It was found out to be in the order of 40msec, in line with expectation. Robot control hexadecimal images were created for the type-1 and type-2 dual surface controllers. In

order to maintain compatibility between the fuzzy logic controllers the type-1 surface was loaded into both the lower and upper array and an average taken. Experiments were carried out on the robots using PID, type-1 and dual surface type-2 controllers. The dual surface type-2 controller was set up to be the classic average type-2 controller. This was to generate a base line against which variations in the dual surface controller could be measured. The configuration of the dual surface controller was altered to enable it to act in the threshold mode. Again experiments were carried out to obtain performance results of the controller. The dual surface controller was then configured to operate firstly as a minimum controller just using the lower surface, and then as a maximum controller just using the upper surface. Finally a statistical analysis was carried out on the results. This showed that there were significant performance differences between the PID, dual surface type-1 and dual surface type-2 controllers at different speed and configurations with the PID controller performing the worst. However not one single fuzzy controller was shown to be consistently better than any other fuzzy controller. Between the dual surface and classic average type-2 controllers, the dual surface controllers demonstrated consistent extra properties over the classic average controller. These extra attributes show that the dual surface controller should be used in preference to the classic average controller where ever possible. The advantage being in the extra flexibility and the ease of providing it.

The only reason for the performance inconsistencies can only be attributed to the excessive noise that is present in the real world and experienced by the micro robots. All the experiments were designed to ensure that each robot controller was observed and measured under the same conditions and subject to the same minimised systematic process errors e.g. one camera and no communication signals during run operations. The fact that the fuzzy controllers were able to operate in the noisy environment is testament to the inherent robustness of fuzzy logic controllers.

Chapter 8

Conclusions

In this chapter, the contribution made by this research to the field of type-2 fuzzy logic controllers will be discussed. The known limitations of the proposed dual surface type-2 fuzzy logic controllers proposed will be explored, and areas of future work to improve the controller will be suggested. The chapter will be concluded with a list of the papers published from this research to date.

8.1 Contributions

The central aim for this research was to discover whether alternatives to PID control, particularly fuzzy controllers of either type-1 and type-2, can make substantial, observable difference to the micro robots within a robot football environment. More specifically, could alternative controllers actually result in a better game being played by the robots.

In conclusion to the investigation carried out in this thesis, it can be stated that the use of alternative controllers particularly type-1 or type-2 fuzzy logic controller has shown a substantial difference from the performance of a PID controller.

Consequently the performance of a micro robot football team is expected to produce a better game when using the fuzzy logic controllers.

More specifically the investigation of the research objectives came to the following conclusions.

1. *Compare PID, type-1 and type-2 controllers on a micro robot with limited computational resources.* In simulation all the fuzzy logic controllers outperformed the

PID controller. In the real world the result was that the fuzzy controllers were able to control the robots better especially at the higher speed demands.

The simulation experiments carried out in Chapter 5 showed that the PID controller was the worst performing controller. The cases tested were with and with out Gaussian noise added to the input and with and without a change of inertia midway through the simulation. In the simulation experiments using the type-1 and type-2 fuzzy logic controllers under the same conditions, the interval type-2 controller was generally the best controller across a set of rule bases and membership functions. However there were instances where the corresponding type-1 controller was marginally better.

In the real world experiments reported in Chapter 7, the situation was that the fuzzy controllers out performed the highly tuned PID controller. Statistical analysis of the time taken to cover a fixed distance at a given speed demand showed that there were significant differences between all the controllers. Within the fuzzy controllers their means were mainly closely grouped around the speed demand. With all the evidence that type-2 controllers out perform type-1 then they should be selected. Within the type-2 controllers it was shown that the dual surface type-2 controller provided a flexible and adaptable alternative to the classic average type-2 controller. As the dual surface controller is a super set of the classic average type-2 controller it should be used in preference to the classic average controller whenever possible.

2. *Investigate a novel dual surface type-2 controller.* It was found that in simulation that the dual surface fuzzy logic controller outperformed the interval classic average type-2 controller and the type-1 controller.

   Simulations were carried out to identify the performance of the controllers. A series of tests were carried out to calculate the RMSE for a set of seven and three term rule base type-1 and average type-2 controllers with and without noise. The best controller was identified to be a T2tri305 controller with an implementation method of Product. This controller was selected to be used in the real world testing.

   The contribution made by the introduction of the Dual Surface allows robot football micro robots, and micro robots in general, to run type-2 fuzzy logic based

controllers that do not generate fixed crisp outputs for a given pair of inputs. Each of the lower and upper surfaces that make up the dual surface controller provide a crisp value for the input pair. By averaging the two crisp values, then an Interval type-2 controller value is obtained. However the advantage of the Dual Surface type-2 controller is that various controller configurations can be exploited.

A Dual Surface controller was generated in the Threshold Configuration. This allowed the controller to select the upper crisp control value when the error of the controller was below threshold value, and the lower crisp value when the error exceeded the threshold value. When the error was within the threshold range then the average of the two surfaces was taken and the controller acted as the average interval type-2 controller. In the situation where the threshold is breached then the motor output demand from the Dual Surface controller will always exceed that of the average interval type-2 controller, allowing the robot to reach its setpoint faster.

Another variation is the Weighted Average Configuration where the input error signal selects a pre-determined weighting of the sum of the two Dual Surface values to give the final crisp value. A variation would be to have a self tuning mechanism to determine the weighting, although the memory constraints need to be considered. The Dual Surface controller can also be configured to act as a type-1 controller. By configuring the Dual Surface controller to be a Maximum or Minimum Controller then only the upper or lower surface crisp value is returned respectively. These are effectively the corresponding type-1 embedded sets of the Dual Surface controller.

3. *Measure the effects of these alternative controllers on real world performance.* The finding of the investigation was that it inconclusive as to which fuzzy logic controller was the most effective in the real world. With the systematic design of the experiments to reduce variation to a minimum, the only explanation is the amount of real world noise experienced by the micro robots. Over 800 real world robot runs were made during this investigation. Unfortunately some of the runs failed to meet the requirements of the experiments and the data discarded. However the data that was accepted was valid if not repeatable, though the experiments are repeatable.

   Analysis of the data showed that the Dual Surface type-2 fuzzy logic controller was

a creditable alternative to the average interval type-2 fuzzy logic controller when used to control a robot football micro robot. All its configurations performed as expected when compared to the classic average type-2 controller. This flexibility and adaptability, together with the property of being able to be run as a classic type-2, recommends that the dual surface type-2 controller has to be the controller of choice.

8.2 Limitations

A major limitation of the Dual Surface controller is that it is dependent upon the memory resource of the robot football micro robots which in this case was limited. This affected the density of the two surfaces that could be stored. The density was half that of the average interval surface. In order to compare like with like the average interval surface was loaded into the upper and lower arrays at the same density. This also ensured that the time to generate the crisp value was the same for both controllers. However the calculation was within the tick time of the robot controller and so was not an issue.

The perennial and open problem of which membership functions to choose for the controllers was overcome by using the same set of membership functions to generate the Interval and Dual Surface controllers. The controllers were then consistent within themselves. A simulation selection process was carried out to identify the best performing set, but it was not globally optimal. A second issue with the robot football micro robots was that there was no possibility of storing any internal values within the Dual Surface controller and reporting them. This was a major source of frustration in understanding how the controller was performing.

From the laboratory experiments the major issue was the performance of the robots themselves. There was hardly a consistent set of experiments that showed how each of the controllers performed across the range of speed demands. Wheels on the micro robots often came off the axles or were sloppy on the axle. The latter problem was the most difficult to detect and adversely effected the performance of the robots. Tightening the wheel screw locks had the effect of causing the wheel to run slowly due to the increased friction.

There were problems with robots not starting altogether and then colliding with each other and invalidating results. At the higher speed demands robots would slip on one wheel and then go into a spin or head off in the direction faced when the slip stopped. The noise generated in the laboratory experiments was not Gaussian as modelled in the simulator experiments. Indeed it was not possible to model in the simulator, the failures experienced in the laboratory.

8.3 Future Work

Future work includes implementing the Dual Surface type-2 fuzzy logic controller on a large robot, such as a MobileRobots Inc Pioneer 3 class. This would allow the Dual Surface Controller to operate on a high specification robot, not subject to the type of slippage and friction problems experienced by the micro robots.

By using a Pioneer the Dual Surface controller would be able to be run in a Fuzzy Inference System. This would allow the dual surfaces crisp values to be calculated every step rather than having to be stored and accessed via a surface array. Any internal variables and parameters can then be recorded, replayed and inspected. The Pioneer implication would also address any scalability issues within the Dual Surface controller, however this is not expected to cause any problems. By installing the Dual Surface controller on Pioneer Robots it will be possible to directly compare its performance against other type-2 fuzzy logic controllers.

Many more studies should be conducted into how the Dual Surface Controller performs under different operating conditions. The studies that were carried out ran the micro robots in a straight line from a standing start without course correction being applied. The performance of the robots when playing robot football against a term using a PID controller could then be investigated.

The algorithms used in the Dual Surface Type-2 Controller can be easily incorporated into other type-2 controllers such as the Geometric and General Type-2 Controllers and should also be investigated within these controllers.

8.3.1 Expectation

With the introduction of the Dual Surface Interval Type-2 Controller as a super set of the Classic Average Interval Type-2 controller it is expected that it will be universally accepted as the de facto interval type-2 controller of choice.

8.4 Dissemination

The following articles have been published as a result of the research presented in this thesis.

8.4.1 Refereed Conference Papers

- P.A.S. Birkin & J.M. Garibaldi, "ASAP Nottingham Team Description", *In the Proceedings of the FIRA World Congress 2005*, Singapore, 14th-16th December, CD-ONLY, 2005.
- P.A.S. Birkin & J.M. Garibaldi, "Comparison of Tuned Membership Functions of Type-1 and Type-2 for a Seven Term Fuzzy Logic Controller", *In the Proceedings of the UK Workshop on Computational Intelligence (UKCI 2008)*, pp 201-206, De Montfort University, Leicester, UK, 10th - 12th Sept 2008.
- P.A.S. Birkin & J.M. Garibaldi, "A comparison of Type-1 and Type-2 fuzzy controllers in a micro-robot context", *In the Proceedings of the 2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009)*, pp 1857-1862, Jeju Island, S.Korea, 20th - 24th August 2009.
- P.A.S. Birkin & J.M. Garibaldi, "A Novel Dual-Surface Type-2 Controller for Micro Robots",*In the Proceedings of the 2010 IEEE International Conference on Fuzzy Systems (Fuzz-IEEE 2010)*, Barcelona, Spain, 18th - 22nd July 2010

8.4.2 Presentations

The following talks have focused on the work presented in this thesis.

- "ASAP RoboSoccer", in Automated Scheduling Optimisation and Planning Research Group seminar, University of Nottingham, Nottingham, 13th July 2005. Given by P.A.S. Birkin.

- "ASAP Nottingham Team Description", *In the Proceedings of the FIRA World Congress 2005*, Singapore, 14th -16th December 2005. Given by P.A.S. Birkin.

- "ASAP RoboSoccer 2006", in Automated Scheduling Optimisation and Planning Research Group seminar,University of Nottingham, Nottingham, 17th May 2006. Given by P.A.S. Birkin

- "ASAP RoboSoccer 2007 A Review of the Development Program", in Automated Scheduling Optimisation and Planning Research Group seminar, University of Nottingham, Nottingham, 17th May 2007. Given by P.A.S. Birkin.

- "A Feasibility Study on a Fire and Forget Control Strategy for a Miabot Robot", in Automated Scheduling Optimisation and Planning Research Group seminar, University of Nottingham, Nottingham, 19th Feb 2008. Given by P.A.S. Birkin.

- "Comparison of Tuned Membership Functions of Type-1 and Type-2 for a Seven Term Fuzzy Logic Controller", *In the Proceedings of the UK Workshop on Computational Intelligence (UKCI 2008)*, De Montfort University, Leicester, UK, 10th - 12th Sept 2008. Given by P.A.S. Birkin.

- "A comparison of Type-1 and Type-2 fuzzy controllers in a micro-robot context", *In the Proceedings of the 2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009)*, Jeju Island, S.Korea, 20th - 24th Aug 2009. Given by J.M. Garibaldi.

- "A Novel Dual-Surface Type-2 Controller for Micro Robots",*In the Proceedings of the 2010 IEEE International Conference on Fuzzy Systems (Fuzz-IEEE 2010)*, Barcelona, Spain, 18th - 22nd July 2010. Given by J.M. Garibaldi.

Bibliography

[1] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[2] R. Seising. 1965 'fuzzy sets' appear - a contribution to the 40th anniversary. In *FUZZ '05 - Proceedings of the 14th IEEE International Conference on Fuzzy Systems*, pages 5–10, Reno, AZ, USA, May 2005.

[3] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.

[4] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

[5] H. Hellendoorn and R. Palm. Fuzzy system technologies at Siemens R & D. *Fuzzy Sets and Systems*, 63(3):245–269, May 1994.

[6] L.P. Holmblad and J-J. Ostergaard. Control of a cement kiln by fuzzy logic. *Fuzzy Sets and Systems*, 70(2-3):135–146, March 1995.

[7] S. Yasunobu, S. Miyamoto, and L. Ihara. Fuzzy control for automatic train operation system. In *IFAC 4th Int. Congress on Control in Transportation Systems*, pages 33–39, 1983.

[8] Zhao-Hui. Jiang and Taiki. Ishita. A neural network controller for trajectory control of industrial robot manipulators. *Journal of Computers*, 3:1–8, Aug 2008.

[9] B. Damas, P. Lima, and L. Custdio. A modified potential fields method for robot navigation applied to dribbling in robotic soccer. In *Proceedings of RoboCup-2002*, 2002.

[10] S. Coupland. Type-2 fuzzy control of robot. De Montford University, Nov 2003. MPhil to PhD transfer report.

[11] M. Veloso, P. Stone, and M. Bowling. Anticipation as a key for collaboration in a team of agents: A case study in robotic soccer. In *Sensor Fusion and Decentralized Control in Robotic Systems II*, volume 3839, Boston, MA, September 1999.

[12] Robert Oates, Julie Greensmith, Uwe Aickelin, Jonathan M. Garibaldi, and Graham Kendall. The application of a dendritic cell algorithm to a robotic classifier. In Leandro Nunes de Castro, Fernando J. Von Zuben, and Helder Knidel, editors, *ICARIS*, volume 4628 of *Lecture Notes in Computer Science*, pages 204–215. Springer, 2007.

[13] Robert Oates, Julie Greensmith, Uwe Aickelin, Jonathan M. Garibaldi, and Graham Kendall. The application of a dendritic cell algorithm to a robotic classifier. *CoRR*, abs/1004.5222, 2010.

[14] Amanda M. Whitbrook, Uwe Aickelin, and Jonathan M. Garibaldi. Idiotypic immune networks in mobile-robot control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(6):1581–1598, 2007.

[15] Amanda M. Whitbrook, Uwe Aickelin, and Jonathan M. Garibaldi. Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Appl. Soft Comput.*, 10(3):876–887, 2010.

[16] Unknown. Darpa grand challenge(2005), 2005.

[17] C. Lynch, H. Hargas, and V. Callaghan. Embedded type-2 flc for real-time speed control of marine and traction diesel engines. In *FUZZ '05. The 14th IEEE International Conference on Fuzzy Systems*, pages 347–352, Reno, NV, USA, May 2005.

[18] G. Cantor. Aber eine eigenschaft des inbegriffes aller reellen algebraischen zahlen. *Crelles Journal f. Mathematik*, 77:258–262, 1874.

[19] E. Zermelo. Neuer beweis fur die moglichkeit einer wohlordnung. *Mathematische Annalen*, 65:107–128, 1908.

[20] A. A. Fraenkel. Der begriff “definit” und die unabhngigkeit des auswahlsaxioms. In *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pages 253–257, 1922.

[21] G.J. Klir and T.A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall International Editions, London, 1988.

[22] J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, Upper Saddle River, NJ, 2001.

[23] T. J. Ross. *Fuzzy Logic with Engineering Applications*. John Wiley and Sons, August 2004.

[24] J. Yen. Fuzzy logic - a modern perspective. *IEEE Transactions on Knowledge and Data Engineering*, 11, Issue 1:153–165, 1999.

[25] J.L. Bell. *Boolean-valued models and independence proofs in set theory*. Clarendon Press, Oxford, 1977.

[26] J. Lukasiewicz. *Philosophical remarks on many-valued systems of propositional logic*. Oxford University Press: Reprinted (1967) in Sorrs McCall (Ed.), Oxford, 1930.

[27] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:28–44, 1973.

[28] D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions. *Fuzzy Sets and Systems*, 40:143–202, 1991.

[29] H.J. Zimmermann. *Fuzzy Set Theory and Its Applications - 3/ed*. Kluwer Academic Publishers, 1996.

[30] G. Lakoff. Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2:458–508, 1973.

[31] Z. Li. Suitability of fuzzy reasoning methods. *Fuzzy Sets and Systems*, 108:299–311, 1999.

[32] J.F. Baldwin. *Fuzzy logic and fuzzy reasoning, in Fuzzy Reasoning and Its Applications, E.H. Mamdani and B.R. Gaines (eds.)*. London Academic Press, 1981.

[33] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.

[34] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.

[35] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - I,II,III. *Information Sciences*, 8;8;9:199–249;301–357;43–80, 1975.

[36] J. M. Mendel. Computing with words, when words can mean different things to different people. In *Proceedings of Third International ICSC Congress on Computational Intelligence Methods and Applications*, 1999.

[37] N.N. Karnik, J.M. Mendel, and Q.L. Liang. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643–658, 1999.

[38] N.N. Karnik and J.M. Mendel. Introduction to type-2 fuzzy logic systems. In *Proceedings of 7th International Conference on Fuzzy Systems FUZZ-IEEE'98*, pages 915–920, May 1998.

[39] T. Ozen and J. M. Garibaldi. Effect of type-2 fuzzy membership function shape on modelling variation in human decision making. In *IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, July 2004.

[40] J.M. Garibaldi, S. Musikasuwan, and T. Ozen. The association between non-stationary and interval type-2 fuzzy sets: A case study. In *Proceedings 2005 IEEE International Conference on Fuzzy Systems*, pages 224 – 229, Reno, NV, May 2005.

[41] M. Mizumoto and K. Tanaka. Some properties of fuzzy sets of type-2. *Information and Control*, 31:312 – 340, 1976.

[42] M. Mizumoto and K. Tanaka. Fuzzy sets of type-2 under algebraic product and algebraic sum. *Fuzzy Sets and Systems*, 5(3):277 – 290, May 1981.

[43] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic, New York, 1980.

[44] J. M. Mendel and R. I. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, April 2002.

[45] S. C. Coupland. *Geometric Fuzzy Logic Systems*. PhD thesis, De Montfort University, Leicester, UK, June 2006.

[46] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - I. *Information Sciences*, 8:199–249, 1975.

[47] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - II. *Information Sciences*, 8:301–357, 1975.

[48] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - III. *Information Sciences*, 9:43–80, 1975.

[49] D. Dubois and H. Prade. Operations on fuzzy numbers. *International Journal of Systems Sciences*, 9:613–626, 1978.

[50] D. Dubois and H. Prade. ”a class of fuzzy measures based on triangular norms”. *”International Journal of General Systems”*, ”1982”.

[51] N.N. Karnik and J.M. Mendel. Application of type-2 fuzzy logic system to forecasting of time-series. *Information Science*, 120:89–111, 1999.

[52] N.N. Karnik and J.M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*, 122:327–348, 2001.

[53] N.N. Karnik and J.M. Mendel. Centroid of a type-2 fuzzy set. *Information Sciences*, 132:195 – 220, 2001.

[54] Q Liang and JM Mendel. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Transactions Fuzzy Systems*, 8:535 – 550, 2000.

[55] N.N. Karnik and J.M. Mendel. Type-2 fuzzy logic systems: Type-reduction. In *Proceedings IEEE Systems, Man and Cybernetics*, pages 2046–2051, 1998.

[56] R.I. John. Type 2 fuzzy sets for knowledge representation and inferencing. *Proceedings of 7th International Conference on Fuzzy Systems FUZZ-IEEE'98*, pages 1003–1008, 1998.

[57] R.I. John. Embedded interval valued fuzzy sets. *Proceedings of FUZZ-IEEE 2002*, pages 1316–1321, 2002.

[58] R.I. John and C. Czarnecki. A type 2 adaptive fuzzy inferencing system. In *Proceedings IEEE Systems, Man and Cybernetics*, pages 2068–2073, 1998.

[59] Q. Liang and J.M. Mendel. An introduction to type-2 tsk fuzzy logic systems. In *Proceedings of 8th International Conference on Fuzzy Systems FUZZ-IEEE'99*, pages 1534–1539, 1999.

[60] Q. Liang and J.M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE. Transactions Fuzzy Systems*, 6(5):535 – 550, October 2000.

[61] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986. A.I.Memo 864 MIT AIL Sept 1985.

[62] B. Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72(1-2):329–365, January 1995.

[63] R. A. Brooks. Intelligence without reason. *IJCAI, Computers and Thought*, 1991.

[64] R. A. Brooks. A robot that walks: Emergent behaviors from a carefully evolved network. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 692–694, May 1989.

[65] I. Horswill. Polly, a vision-based artificial agent. In *Proceedings of the AAAI-93*, pages 824–829, Washington DC, 1993.

[66] R.A. Brooks and A. Flynn. Robot beings. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems(IROS-89)*, pages 2–10, Tsukuba Japan, 1989.

[67] A. Flynn, R. A. Brooks, W. Wells, and D Barrett. Squirt: The prototypical mobile robot for autonomous graduate students. A.I. Memo 1120, MIT AI Lab, Cambridge MA, July 1989.

[68] M. J. Mataric. Integration of representation into goal-driven behaviour based robots. *IEEE Translations on Robotics and Automation*, 8(3):304–312, June 1992.

[69] R. C. Arkin. *Behaviour-based robotics*. MIT Press, Cambridge, Mass 02142, 1998.

[70] R. Hartley and F. Pipitone. Experiments with the subsumption architecture. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1652–1658, Sacramento, CA, 1991.

[71] M. A. Arbib. Perceptual structures and distributed motor control. In V. B. Brooks, editor, *Handbook of Physiology - The Nervous System II:Motor Control*, pages 1449–1480, Bethesda, MD, 1981. American Physiological Society.

[72] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500–505, St Louis, MO, 1985.

[73] M. A. Arbib. Schema theory. In S. Shapiro, editor, *The Encyclopaedia of Artificial Intelligence*, pages 1427–1443, New York, NY, 1992. Wiley-Interscience.

[74] R. C. Arkin. Neuroscience in motion: The application of schema theory to mobile robots. In J-P. Ewert and M.A. Arbib, editors, *Visuomoter Coordination:Amphibians, Comparisons, Models and Robots*, pages 649–672, New York, 1989. Plenum Press.

[75] R. C. Arkin. The impact of cybernetics on the design of a mobile robot system: A case study. *IEEE Transactions on Systems, Man and Cybernetics*, 20(6):1245–1257, November/December 1990.

[76] R. C. Arkin. Modelling neural function at the schema level: Implications and results for robotic control. In R Ritzmann T McKenna and R Beer, editors, *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, pages 473–478, San Diego, CA, 1993.

[77] R. C. Arkin and R. R. Murphy. Autonomous navigation in a manufacturing environment. *IEEE Transactions on Robotics and Automation*, 6(4):445–454, August 1990.

[78] R. C. Arkin. Reactive control as a substrate for telerobotic systems. *IEEE Aerospace and Electronic Systems Magazine*, 6(6):24–31, June 1991.

[79] T. Balch and R. C. Arkin. Avoiding the past: A simple but effective strategy for reactive navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 678–685, Atlanta GA, May 1993.

[80] T. R. Collins, R. C. Arkin, and A. M. Henshaw. Integration of reactive navigation with a flexible parallel hardware architecture. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 271–276, Atlanta GA, May 1993.

[81] S. Rosenschein and L. Kaelbling. The synthesis of digital machines with provable epistemic properties. Technical Note 412, SRI International, Menlo Park, CA, April 1987.

[82] P. Maes. The dynamics of action selection. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence'(IJCAI-89)*, pages 991–997, Detroit, MI, 1989.

[83] P. Maes. Situated robots can have goals. *Robotics and Automated Systems*, 6:49–70, 1990.

[84] J. Connell. A colony architecture for an artificial creature. Technical Report 1151, MIT AI Lab, Cambridge MA, August 1989.

[85] R. J. Firby. *Adaptive execution in complex dynamic worlds*. Technical report yaleu/csd/rrno.672, Yale University, New Haven CT, 1989.

[86] R. J. Firby. Lessons learnt from the animate agent project(so far). In *Working Notes, AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, pages 92–96, Palo Alto CA, March 1995.

[87] J. Rosenblatt and D. Payton. A fine grained alternative to the subsumption architecture for mobile robot control. In *Proceedings of the International Joint Conference on Neural Networks*, pages 317–323, June 1989.

[88] J. Rosenblatt. Damn: A distributed architecture for mobile navigation. In *Working Notes, AAAI Spring Symposium on Lessons Learned for Implemented Software Architectures for Physical Agents*, pages 167–178, Palo Alto CA, March 1995.

[89] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[90] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. *Proc. IEEE conf. Robotics and Automation*, pages 1398–1404, April 1991.

[91] Bassam Daya. A multilayer perceptrons model for the stability of a bipedal robot. *Neural Process. Lett.*, 9(3):221–227, 1999.

[92] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. U.S.A.*, 79(8):2554–2558, 1982.

[93] J. Nielsen and H. H. Lund. Spiking neural building block robot with hebbian learning. In *Proceedings of IROS03. IEEE.* Press, 2003.

[94] D. Lyons and A. Hendriks. Planning for reactive robot behaviour. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2675–2680, Nice France, 1992.

[95] D. Lyons and A. Hendriks. Planning as incremental adaptation of a reactive system. *Robotics and Autonomous Systems*, 14(4):255–288, 1995.

[96] John R. Koza and James P. Rice. Automatic programming of robots using genetic programming. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 194–201. The MIT Press, 1992.

[97] John R. Koza. Evolution of subsumption using genetic programming. In *Proceedings of the first European Conference on Artificial Life*, pages 110–119. MIT Press, 1993.

[98] Karl Astrom and Tore Hagglund. *PID Controllers: Theory, Design and Tuning - Third Edition*. ISA - The Instrumentation, Systems, and Automation Society, 2005.

[99] Karl Astrom and Tore Hagglund. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, 2005.

[100] Aidan O'Dwyer. *Handbook of PI and PID controller tuning rules - Third Edition*. Imperial College Press, 2009.

[101] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems - Sixth Edition*. Prentice-Hall, Oct 2009.

[102] FIRA. Federation of international robot-soccer association, 2009.

[103] RoboCup. The robocup federation. *http://www.robocup.org*, 2007.

[104] P.A.S. Birkin and J.M. Garibaldi. Asap nottingham team description. In *Proceedings of the 10th FIRA RoboWorld Congress*, Singapore, 14th - 16th Dec 2005.

[105] P. Melin and O. Castillo. Intelligent control of non-linear dynamic plants using type-2 fuzzy logic and neural networks. In *FUZZ-IEEE 2004*, Budapest,Hungary, July 2004.

[106] Patricia Melin and Oscar Castillo. A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks. *International Journal of General Systems*, 33(2-3):289 – 304, 2004.

[107] Oscar Castillo, Gabriel Huesca, and Fevrier Valdez. Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants. In *NAFIPS 2005*, pages 247 – 251, Detroit, MI, United States, June 2005.

[108] Dongrui Wu and Woei Wan Tan. A type-2 fuzzy logic controller for the liquid-level process. In *IEEE International Conference on Fuzzy Systems*, volume 2, pages 953 – 958, Budapest, Hungary, July 2004.

[109] Dongrui Wu and Woei Wan Tan. A simplified type-2 fuzzy logic controller for real-time control. *ISA Transactions*, 45(4):503 – 516, 2006.

[110] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments. *Journal of Information Sciences*, 171(4):309 – 334, May 2005.

[111] Hani Hagras, Faiyaz Doctor, Antonio Lopez, and Victor Callaghan. Evolving type-2 fuzzy agents for ambient intelligent environments. In *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS'06*, pages 208 – 213, Lake District, United Kingdom, September 2006.

[112] H. Hagras, F. Doctor, and A. Lopez. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. *IEEE Transactions on Fuzzy Systems*, 15(1):45 – 55, February 2007.

[113] H. Hagras. Type-2 flcs: A new generation of fuzzy controllers. *IEEE Computational Intelligence Magazine*, 2(1):30–43, February 2007.

[114] Kung Chris Wu. Fuzzy interval control of mobile robots. *Computers & Electrical Engineering*, 22(3):211 – 229, 1996.

[115] R. Melgarejo, Miguel A., and Carlos A. Pena-Reyes. Hardware architecture and fpga implementation of a type-2 fuzzy system. In *Proceedings of the ACM Great Lakes Symposium on VLSI*, pages 458 – 461, Boston, MA, United States, April 2004.

[116] Ping-Zong Lin, Chun-Fei Hsu, and Tsu-Tian Lee. Type-2 fuzzy logic controller design for buck dc-dc converters. In *IEEE International Conference on Fuzzy Systems*, pages 365 – 370, Reno, NV, United States, May 2005.

[117] P.-Z. Lin, C.-M. Lin, C.-F. Hsu, and T.-T. Lee. Type-2 fuzzy controller design using a sliding-mode approach for application to dc-dc converters. *IEE Proceedings: Electric Power Applications*, 152(6):1482 – 1488, November 2005.

[118] Hani Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems*, 12(4):524–539, 2004.

[119] Christopher Lynch, Hani Hagras, and Victor Callaghan. Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for

marine diesel engines. In *IEEE International Conference on Fuzzy Systems*, pages 1446 – 1453, Vancouver, BC, Canada, July 2006.

[120] S. Coupland, M. Gongora, R.I. John, and K. Wills. A comparative study of fuzzy logic controllers for autonomous robots. In *IPMU 2006*, pages 1332 – 1339, Paris, France, 2006.

[121] Roberto Sepúlveda, Oscar Castillo, Patricia Melin, Antonio Rodríguez-Díaz, and Oscar Montiel. Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Inf. Sci.*, 177(10):2023–2048, 2007.

[122] Dongrui Wu and Woei Wan Tan. Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers. *Eng. Appl. Artif. Intell.*, 19(8):829–841, 2006.

[123] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas. A type-2 fuzzy logic controller for tracking mobile objects in the context of robotic soccer games. In *Proceedings 2005 IEEE International Conference on Fuzzy Systems*, pages 359–364, Reno, NV, May 2005.

[124] H. Hagras. Developing a type-2 flc through embedded type-1 flcs. In *FUZZ-IEEE 2008*, Hong Kong, June 2008.

[125] S. Coupland and R. John. A fast geometric method for defuzzification of type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(4):929–941, 2008.

[126] C. Wagner and H. Hagras. zSlices based general type-2 flc for the control of autonomous mobile robots in real world environments. In *FUZZ-IEEE'09: Proceedings of the 18th international conference on Fuzzy Systems*, pages 718–725, Piscataway, NJ, USA, 2009. IEEE Press.

[127] H. Hamrawi and S. Coupland. Type-2 fuzzy arithmetic using alpha-planes. In *Proceedings of IFSA/EUSFLAT 2009*, Lisbon, Portugal, 2009. Accepted for publication.

[128] J. Mendel and L. Liu. On new quasi-type-2 FLS. In *IEEE International Conference on Fuzzy Systems(FUZZ-IEEE 2008)*, pages 354–360, Hong Kong, China, 1-6 June 2008.

[129] S. Greenfield, F. Chiclana, S. Coupland, and R. John. The collapsing method of defuzzification for discretised interval type-2 fuzzy sets. *Information Sciences*, 179(13):2055–2069, 2009.

[130] S. Greenfield, F. Chiclana, and R.I. John. The collapsing method: Does the direction of collapse affect accuracy? In *IFSA-EUSFLAT 2009 Conference*, 2009. Accepted for oral presentation.

[131] S. Greenfield, F. Chiclana, and R.I. John. Type-reduction of the discretised interval type-2 fuzzy set. In *2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009)*, pages 738–743, Jeju Island, South Korea, August 2009.

[132] N. Nie and W.W. Tan. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *Proceedings of Fuzz-IEEE 2008*, pages 1425–1432, Hong Kong, China, 2008.

[133] S. Miller, V. Popova, R.I. John, and M. Gongora. Improving resource planning with soft computing techniques. In *Proceedings of UKCI 2008*, pages 37–42, De Montfort University, Leicester, UK., 2008.

[134] S. Miller, V. Popova, R.I. John, and M. Gongora. An interval type-2 fuzzy distribution network. In *IFSA-EUSFLAT 2009 Conference*, pages 697–702, Lisbon, Portugal, 2009.

[135] Simon Miller and Robert John. An interval type-2 fuzzy multiple echelon supply chain model. In Max Bramer, Richard Ellis, and Miltos Petridis, editors, *SGAI Conf.*, pages 407–420. Springer, 2009.

[136] S. Miller, M. Gongora, and R. John. Optimising resource plans using an interval type-2 fuzzy model. In *Proceedings of Fourth International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS 2010)*, Mieres, Asturias, Spain, 2010. European Centre for Soft Computing, IEEE. Accepted for oral presentation.

[137] S. Miller, M. Gongora, and R. John. Inventory optimisation with an interval type-2 fuzzy model. In *Proceedings of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010)*, Barcelona, Spain., 18 - 23 July 2010. Accepted for oral presentation.

[138] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on System, Man, Cybernetics*, 15(1):116–132, 1985.

[139] Miguel Melgarejo and Carlos A Pena-Reyes. Implementing interval type-2 fuzzy processors. *IEEE Computational Intelligence Magazine*, 2(1):63–71, Feb 2007.

[140] C. Wagner and J.M. Hargas. Employing interpolation to enable higher order fuzzy logic controllers on resource constrained embedded devices. In *UKCI 2008*, Leicester, UK, 10th - 12th Sept 2008.

Appendix A

Chapter 3 Tables

A.1 Maibot Hardware Specification

This appendix contains the hardware specification of the Merlin Maibot micro robot used in the real world experiments.

Processors	
Main	Atmel ATMega64
Slave	Atmel ATMega8
Flash Memory	64K
E2ROM	2k Bytes
SRAM	4K
Speed	Up to 16 MIPS
HBbridge	2 x 5A MOSFET H-Bridge
Drive Train	
Dual DC Motors	6v
Encoders	0.04mm Resolution
Wheel Diameter	52mm
Gearbox Ratio	8:1
Stall Torque	175 g/cm @2.85A
Speed	3.5 m/s
Batteries	
NIMH	6 x 1.2v AA NIMH Cells
Operation	Approx. 2Hr
Development	
Simulator	FIRA 3D
'C' Compiler	GCC with development kit
Firmware Download	PonyProg
Expansion	
I/O Port 1	8 x 8 A/D inputs 10 bit or 8 Digital I/O
I/O Port 2	In System Programming Socket Serial Hardlink Comms
Multiplexed Functions	I2C,SPI,5 x PWM, 6 x Interrupt Capable I/O
Communications	
Wireless	Bluetooth
Range	Up to 100m
Speed	Bidirectional @ 115kb/s
Other	
LEDS	2
DIP Switches	4
Power Switch	1
Firmware	
Control	PID Control system
Communications	ASCII drive protocol.
Extensible	Many extendible features
Bluetooth firmware	
Chassis	
Construction	All aluminum
Case Dimensions	$7.5cm^3$

Table A.1: Robot Specification

Appendix B

Chapter 4 Tables

This appendix contains the full data tables for the chapter.

	LEFT	MOTOR			RIGHT	MOTOR			
Pwm	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Diff L-R Mean
-120	-415.44	1.52	14	-0.31	-404.57	1.83	14	-5.86	-10.86
-110	-416.51	1.78	22	-5.41	-400.35	1.86	21	-7.69	-16.17
-100	-413.29	1.73	24	-4.36	-393.52	2.01	23	-5.42	-19.76
-90	-410.98	1.62	29	-5.02	-386.84	2.09	29	-5.82	-24.14
-80	-407.62	1.66	34	-4.98	-378.58	2.05	32	-4.37	-29.04
-70	-399.95	1.88	40	-3.06	-367.72	2.10	37	-2.70	-32.23
-60	-391.87	1.62	51	-1.32	-352.99	2.10	45	-1.88	-38.88
-50	-380.02	1.71	63	-0.56	-331.07	2.21	55	3.68	-48.95
-40	-356.44	2.42	80	6.58	-299.17	2.16	74	3.88	-57.27
-30	-314.64	4.82	114	4.22	-245.38	3.13	99	3.66	-69.27
-20	-215.89	10.45	147	-7.90	-139.49	4.61	134	5.60	-76.41
20	207.99	11.12	152		145.09	6.11	139		62.90
30	318.86	6.36	117		249.04	3.59	100		69.82
40	363.02	1.82	87		303.05	2.27	77		59.97
50	379.46	1.49	63		334.75	2.20	62		44.70
60	390.55	1.58	50		351.11	2	45		39.45
70	396.89	1.36	41		365.02	2.16	38		31.87
80	402.64	1.35	33		374.22	2	30		28.43
90	405.96	1.28	28		381.02	1.85	26		24.94
100	408.93	1.48	24		388.11	1.82	23		20.82
110	411.11	1.36	21		392.65	1.83	20		18.46
120	415.13	1.40	17		398.71	1.71	16		16.42

Table B.1: No Load Mean and SD Clix for Left and Right Motor PWM Demands

	LEFT	MOTOR			RIGHT	MOTOR			
Pwm	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Mean Clix	SD Clix	90% Clix#	Diff +-Mean	Diff L-R Mean
-120	-238.24	37.21	9	89.07	-333.86	47.15	30	-71.71	95.62
-110	-326.48	50.19	90	-1.56	-193.87	25.93	8	32.40	-132.60
-100	-297.46	50.52	65	2.02	-170.70	25.47	15	25.84	-126.75
-90	-292.80	59.69	130	-6.34	-194.49	41.96	29	89.48	-98.32
-80	-286.81	27.10	146	-6.64	-200.25	47.79	74	82.63	-86.56
-70	-254.12	29.99	159	-19.26	-194.92	30.19	93	43.01	-59.20
-60	-195.96	26.40	155	25	-192.30	11.83	133	32.10	-3.66
-50	-133.49	15.36	145	19.72	-149.49	16.19	160	2.72	16
-40	-99.58	10.90	164	1.58	-111.32	12.40	169	-10.09	11.73
-30	-56.11	7.35	172	-2.18	-62.12	8.39	177	-5.52	6.01
-20	0	0	1	0	0	0	2	0	0
20	0	0	249		0	0	249		0
30	53.93	7.21	174		56.59	7.16	169		-2.66
40	101.17	14.38	183		101.23	13.36	174		-0.06
50	153.21	15.54	162		152.21	14.17	162		1
60	220.96	19.43	168		224.40	19.17	161		-3.44
70	234.86	21.15	162		237.93	19.77	155		-3.07
80	280.17	22.70	167		282.88	21.57	153		-2.71
90	286.47	24.93	169		283.96	25.17	160		2.50
100	299.48	34.87	113		196.54	38.34	46		102.93
110	324.91	33.54	126		226.27	38.44	38		98.64
120	327.31	34.98	84		262.15	31.31	9		65.16

Table B.2: Loaded Mean and SD Clix for Left and Right Motor PWM Demands

P	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
5	14.03960396	0.195999596	147.1287129	1.33164084	132.4158416	54	2.452145215
10	15.65346535	0.537320083	188.1089109	1.121614819	169.2980198	23	7.36078347
15	16.1980198	0.583434692	204.1683168	1.140783125	183.7514851	14	13.12510608
20	16.56435644	0.805181242	212.6435644	1.100764811	191.3792079	8	23.92240099
25	16.96039604	0.720011001	217.6138614	1.076757141	195.8524752	6	32.64207921
30	17.12871287	1.101484147	220.8811881	1.125052254	198.7930693	5	39.75861386
35	17.04950495	1.023486567	223.950495	1.023486567	201.5554455	5	40.31108911
40	17.21782178	1.213292713	225.8712871	1.02628813	203.2841584	5	40.65683168
45	17.3960396	1.720925379	227.2574257	1.1717804	204.5316832	5	40.90633663
50	17.25742574	1.735819491	228.6534653	1.135214901	205.7881188	5	41.15762376
55	17.27722772	1.78392159	229.6138614	1.095173932	206.6524752	4	51.66311881
60	17.23762376	1.99072105	230.4752475	1.1277771	207.4277228	5	41.48554455
65	17.32673267	2.159207776	231.3366337	1.079603888	208.2029703	5	41.64059406
70	17.3960396	2.349805132	231.8019802	1.174902566	208.6217822	5	41.72435644
75	17.16831683	2.530096073	232.4752475	1.07325728	209.2277228	5	41.84554455
80	17.13861386	2.828532139	233.039604	1.112841337	209.7356436	5	41.94712871
85	17.20792079	2.940465377	233.2871287	1.089372605	209.9584158	5	41.99168317
90	17.26732673	2.962738899	233.6435644	1.082443148	210.2792079	5	42.05584158
95	17.11881188	3.392011582	233.960396	1.130670527	210.5643564	5	42.11287129
100	17.36633663	3.065037593	234.2178218	1.006021475	210.7960396	5	42.15920792
105	17.42574257	3.207324538	234.5247525	1.035316951	211.0722772	5	42.21445545
110	17.32673267	3.455745682	234.8316832	0.990649352	211.3485149	5	42.26970297
115	17.45544554	3.835426319	234.970297	1.062595366	211.4732673	5	42.29465347
120	17.30693069	4.115197624	235.2673267	1.130407795	211.7405941	5	42.34811881
125	17.0990099	4.439605727	235.4752475	1.109901432	211.9277228	5	42.38554455
130	17.26732673	4.076496263	235.6831683	1.019124066	212.1148515	5	42.4229703
135	16.77227723	5.057432527	235.9108911	1.167039073	212.319802	5	42.4639604
140	16.92079208	5.059018024	236.019802	1.183048588	212.4178218	5	42.48356436
145	23.43564356	44.95206688	234.7326733	9.920575678	211.2594059	8	26.40742574
150	52.87128713	94.98417377	228.6336634	20.26164987	205.770297	13	15.82848439

Table B.3: Robot Response to Proportional Action

P,I	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
40,4	17.22772277	1.567681014	240.019802	1.174565435	216.0178218	4	54.00445545
8	17.1980198	1.697172955	239.9306931	1.142430967	215.9376238	5	43.18752475
12	16.99009901	1.763491137	240.019802	1.104356808	216.0178218	5	43.20356436
16	17.08910891	1.944731395	240	1.148912529	216	5	43.2
20	16.81188119	2.138751371	240.029703	1.187058933	216.0267327	14	15.43048091
50,5	17.04950495	1.878170587	239.960396	1.130670527	215.9643564	5	43.19287129
10	17.01980198	2.083171611	239.950495	1.160829338	215.9554455	5	43.19108911
15	16.72277228	2.117162308	240.009901	1.126898838	216.0089109	5	43.20178218
20	16.62376238	2.395209741	240.009901	1.126898838	216.0089109	5	43.20178218
25	16.77227723	2.576358625	240.019802	1.166020566	216.0178218	13	16.61675552
60,6	16.77227723	2.297307938	240.039604	1.165510979	216.0356436	5	43.20712871
12	16.71287129	2.628066337	240.029703	1.220290503	216.0267327	5	43.20534653
18	16.9009901	2.574897864	240.009901	1.178940622	216.0089109	5	43.20178218
24	16.68316832	2.9527299	240.009901	1.244950196	216.0089109	5	43.20178218
30	16.66336634	3.066846027	240.039604	1.174059556	216.0356436	19	11.37029703
70,7	16.84158416	2.674818399	240	1.140175425	216	5	43.2
14	17.10891089	2.852721473	239.960396	1.148222906	215.9643564	5	43.19287129
21	16.54455446	2.703792716	240.019802	1.039039922	216.0178218	5	43.20356436
28	16.53465347	3.413398179	239.980198	1.264754506	215.9821782	13	16.61401371
35	16.97029703	3.232508146	240	1.113552873	216	13	16.61538462
80,8	16.85148515	2.829792002	240.019802	1.067522347	216.0178218	5	43.20356436
16	16.88118812	2.906155979	240.039604	1.028793391	216.0356436	5	43.20712871
24	16.73267327	3.560592898	240.019802	1.199834972	216.0178218	26	8.308377761
32	16.51485149	3.785799417	239.990099	1.212394734	215.9910891	5	43.19821782
40	16.79207921	4.203134144	240.019802	1.224583178	216.0178218	19	11.36935904
90,9	16.84158416	3.028969043	239.980198	1.01960971	215.9821782	5	43.19643564
18	16.64356436	3.84859496	239.990099	1.212394734	215.9910891	5	43.19821782
27	16.53465347	3.882175566	240.009901	1.170427695	216.0089109	5	43.20178218
36	16.64356436	3.801536948	239.990099	1.090825829	215.9910891	5	43.19821782
45	150.3366337	194.8380495	239.3069307	52.58150674	215.3762376	8	26.9220297
100,10	16.73267327	3.911242997	240.009901	1.178940622	216.0089109	5	43.20178218
20	16.87128713	3.767395297	240	1.058300524	216	5	43.2
30	16.18811881	4.621066698	240	1.224744871	216	5	43.2
40	16.11881188	4.290191438	240.019802	1.104356808	216.0178218	26	8.308377761
50	129.4752475	182.2025573	239.8118812	44.0153866	215.8306931	133	1.622787166
110,11	16.88118812	3.459153448	240	0.959166305	216	5	43.2
22	16.8019802	4.374973833	239.990099	1.135738082	215.9910891	5	43.19821782
33	16.1980198	4.787525043	240.009901	1.204118346	216.0089109	5	43.20178218
44	83.51485149	149.9124154	240.3960396	33.0363676	216.3564356	157	1.378066469
55	165.2970297	220.2648653	238.8514851	48.61571477	214.9663366	13	16.53587205
120,12	16.43564356	4.622587677	240.029703	1.170089275	216.0267327	4	54.00668317
24	16.37623762	4.867959501	239.980198	1.157412615	215.9821782	5	43.19643564
36	132.2376238	189.0796736	239.8514851	43.12456055	215.8663366	5	43.17326733
48	135.8217822	197.5897971	239.8415842	42.71878572	215.8574257	26	8.302208682
60	173.6336634	221.8964498	238.9306931	44.87053764	215.0376238	176	1.22180468

Table B.4: Robot Response to Proportional and Integral Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
40:4,4	17.3960396	1.66180148	239.960396	1.076297283	215.9643564	5	43.19287129
8	17.35643564	1.723857061	240.019802	1.029370662	216.0178218	5	43.20356436
12	17.41584158	2.12728619	239.990099	1.144509061	215.9910891	5	43.19821782
16	17.34653465	1.956709705	240.029703	0.953472029	216.0267327	5	43.20534653
20	17.32673267	2.454012677	240.009901	1.109009013	216.0089109	5	43.20178218
40:8,4	17.26732673	1.593054231	240.009901	0.984835514	216.0089109	5	43.20178218
8	17.38613861	1.912957381	239.970297	1.071964977	215.9732673	5	43.19465347
12	17.3960396	1.970173637	239.9306931	0.99254648	215.9376238	5	43.18752475
16	17.31683168	2.424585297	240.019802	1.104356808	216.0178218	5	43.20356436
20	17.25742574	2.648220026	240.009901	1.109009013	216.0089109	7	30.85841584
40:12,4	17.26732673	2.13021637	239.980198	1.264754506	215.9821782	5	43.19643564
8	17.43564356	2.075648533	240.049505	1.089736093	216.0445545	5	43.20891089
12	17.16831683	2.172875086	239.980198	1.113375031	215.9821782	5	43.19643564
16	17.38613861	2.720184909	239.990099	1.228780286	215.9910891	4	53.99777228
20	17.12871287	2.536388639	240	1.077032961	216	5	43.2
40:16,4	17.42574257	1.981648479	239.980198	1.104356808	215.9821782	5	43.19643564
8	17.25742574	2.198424278	239.990099	1.135738082	215.9910891	5	43.19821782
12	17.2970297	2.269557465	240.009901	1.109009013	216.0089109	5	43.20178218
16	17.23762376	2.510571707	240.019802	1.104356808	216.0178218	5	43.20356436
20	17.2970297	2.791216776	239.950495	1.152182604	215.9554455	5	43.19108911
40:20,4	17.28712871	2.155628139	239.980198	1.113375031	215.9821782	13	16.61401371
8	17.07920792	2.72280432	240.019802	1.318940469	216.0178218	5	43.20356436
12	16.88118812	2.811715237	239.980198	1.256822963	215.9821782	5	43.19643564
16	17.14851485	2.961034071	239.990099	1.268818738	215.9910891	5	43.19821782
20	17.25742574	2.995508188	240.019802	1.199834972	216.0178218	5	43.20356436

Table B.5: Robot Response to Proportional(40),Integral and Differential Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
50:5,5	17.1980198	2.181833183	240	1.13137085	216	5	43.2
10	17.17821782	2.036644493	240	0.989949494	216	5	43.2
15	17.18811881	2.331149379	240.049505	1.023486567	216.0445545	6	36.00742574
20	17.21782178	2.755372789	240.019802	1.067522347	216.0178218	5	43.20356436
25	17.33663366	2.912995804	239.970297	1.071964977	215.9732673	5	43.19465347
50:10,5	17.35643564	2.347697418	240.029703	1.144162974	216.0267327	5	43.20534653
10	17.37623762	2.125330493	239.970297	0.942925719	215.9732673	5	43.19465347
15	17.24752475	2.295238291	239.990099	0.953887305	215.9910891	5	43.19821782
20	17.00990099	3.014946267	240.039604	1.148222906	216.0356436	5	43.20712871
25	17.14851485	2.971148393	240.019802	1.048620027	216.0178218	5	43.20356436
50:15,5	17.23762376	2.083979438	240.039604	0.999207607	216.0356436	5	43.20712871
10	17.28712871	2.829617054	240.019802	1.208140704	216.0178218	5	43.20356436
15	17.2970297	2.964943691	239.970297	1.178604646	215.9732673	5	43.19465347
20	17.25742574	3.045171474	239.990099	1.135738082	215.9910891	5	43.19821782
25	17.12871287	3.279217487	240	1.13137085	216	5	43.2
50:20,5	17.42574257	2.503383849	239.990099	1.135738082	215.9910891	5	43.19821782
10	17.06930693	2.593289131	240.019802	1.058113397	216.0178218	6	36.0029703
15	17.13861386	3.075807871	240	1.166190379	216	5	43.2
20	17.2970297	3.864051124	240.019802	1.334017976	216.0178218	5	43.20356436
25	17.1980198	3.171812737	240.019802	1.058113397	216.0178218	5	43.20356436
50:25,5	17.26732673	2.969481736	239.990099	1.244950196	215.9910891	5	43.19821782
10	17.18811881	2.999042752	239.990099	1.187392517	215.9910891	5	43.19821782
15	17.12871287	3.01550449	240.019802	1.086095742	216.0178218	5	43.20356436
20	17.23762376	3.47893235	239.990099	1.204118346	215.9910891	5	43.19821782
25	16.96039604	4.147097279	240	1.349073756	216	5	43.2

Table B.6: Robot Response to Proportional(50),Integral and Differential Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
60:6,6	17.22772277	2.549043696	240.039604	1.139480514	216.0356436	4	54.00891089
12	17.15841584	2.583535071	240.009901	1.024646764	216.0089109	5	43.20178218
18	17.42574257	2.854282868	239.990099	1.004938302	215.9910891	5	43.19821782
24	16.99009901	3.482800739	240.019802	1.140001737	216.0178218	5	43.20356436
30	17.03960396	3.597000951	240.009901	1.06296801	216.0089109	5	43.20178218
60:12,6	17.25742574	2.659524263	240	1.13137085	216	5	43.2
12	17.05940594	3.126729225	240.019802	1.216389724	216.0178218	5	43.20356436
18	17.42574257	3.007811612	239.9405941	1.037514165	215.9465347	5	43.18930693
24	17.28712871	3.688730496	239.960396	1.139480514	215.9643564	5	43.19287129
30	17.26732673	3.719922282	239.990099	1.090825829	215.9910891	5	43.19821782
60;18,6	17.08910891	2.404574848	240.009901	0.953887305	216.0089109	5	43.20178218
12	17.21782178	3.008667348	240.029703	1.081253398	216.0267327	5	43.20534653
18	17.44554455	3.485613999	239.970297	1.170089275	215.9732673	5	43.19465347
24	17	3.635931793	240	1.095445115	216	5	43.2
30	16.95049505	4.255293733	239.990099	1.244950196	215.9910891	5	43.19821782
60:24,6	17.15841584	2.938478087	239.990099	1.117989709	215.9910891	5	43.19821782
12	17.04950495	3.499646375	239.990099	1.212394734	215.9910891	5	43.19821782
18	17.25742574	3.618434649	240.019802	1.174565435	216.0178218	5	43.20356436
24	17.05940594	4.211464786	240.009901	1.228780286	216.0089109	5	43.20178218
30	16.86138614	4.16180178	240	1.13137085	216	5	43.2
60:30,6	17.22772277	2.925341649	239.990099	1.043983233	215.9910891	13	16.61469916
12	16.99009901	3.214016333	240.019802	1.095264334	216.0178218	13	16.61675552
18	17.08910891	3.914330108	240	1.224744871	216	13	16.61538462
24	6.77227723	4.61493486	240.029703	1.345031193	216.0267327	5	43.20534653
30	16.91089109	4.13545405	240	1.191637529	216	5	43.2

Table B.7: Robot Response to Proportional(60),Integral and Differential Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
70:7,7	17.12871287	2.587134965	239.990099	0.994937682	215.9910891	5	43.19821782
14	17.21782178	2.869160018	240	0.989949494	216	5	43.2
21	16.99009901	3.694577241	240.029703	1.135389321	216.0267327	5	43.20534653
28	17.12871287	3.817494902	239.990099	1.090825829	215.9910891	5	43.19821782
35	17.07920792	3.996706565	240	1.048808848	216	5	43.2
70:14,7	17.2970297	3.09368568	240.009901	1.135738082	216.0089109	5	43.20178218
14	17.13861386	3.101708249	239.980198	1.009754406	215.9821782	5	43.19643564
21	17.06930693	3.060253015	240.019802	0.999801961	216.0178218	5	43.20356436
28	17.21782178	3.531583102	239.990099	1.043983233	215.9910891	5	43.19821782
35	17.05940594	4.009543072	240.009901	1.109009013	216.0089109	5	43.20178218
70:21,7	16.74257426	4.853150452	239.990099	1.212394734	215.9910891	5	43.19821782
14	17.31683168	3.133466748	239.960396	1.076297283	215.9643564	5	43.19287129
21	17.03960396	3.307327598	239.990099	1.043983233	215.9910891	5	43.19821782
28	16.95049505	3.782528883	239.990099	1.081619614	215.9910891	5	43.19821782
35	16.87128713	4.263011533	239.990099	1.144509061	215.9910891	5	43.19821782
70:28,7	16.77227723	4.925202916	240.019802	1.208140704	216.0178218	5	43.20356436
14	17.3960396	3.46721562	240.009901	1.109009013	216.0089109	5	43.20178218
21	17.17821782	3.77464181	239.980198	1.13119581	215.9821782	5	43.19643564
28	17.07920792	4.204005633	240	1.191637529	216	5	43.2
35	16.92079208	4.305074142	240.009901	1.117989709	216.0089109	5	43.20178218
70:35,7	16.92079208	4.464713134	239.980198	1.113375031	215.9821782	5	43.19643564
14	17.24752475	3.235447235	240.019802	0.989749443	216.0178218	13	16.61675552
21	17.24752475	3.468157841	240.009901	0.994937682	216.0089109	5	43.20178218
28	17.13861386	3.891091628	239.980198	1.076849089	215.9821782	5	43.19643564
35	16.71287129	5.103599972	240.029703	1.299657228	216.0267327	5	43.20534653

Table B.8: Robot Response to Proportional(70),Integral and Differential Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
80:8,8	19.34653465	3.247878211	240	1.104536102	216	4	54
16	19.31683168	3.957096645	240.009901	1.195784675	216.0089109	4	54.00222772
24	19.47524752	4.389975078	239.980198	1.191471343	215.9821782	4	53.99554455
32	19.16831683	5.278388593	239.990099	1.268818738	215.9910891	4	53.99777228
40	18.89108911	6.006498131	240.029703	1.330078536	216.0267327	4	54.00668317
80:16,8	19.44554455	3.606869134	240	1.15758369	216	4	54
16	19.57425743	4.368859198	239.990099	1.252956899	215.9910891	4	53.99777228
24	18.99009901	4.401124969	240.019802	1.174565435	216.0178218	4	54.00445545
32	19.15841584	5.328663385	239.950495	1.252008288	215.9554455	4	53.98886139
40	19.01980198	5.368389326	240.009901	1.220615005	216.0089109	4	54.00222772
80:24,8	19.23762376	4.022806271	239.980198	1.208140704	215.9821782	4	53.99554455
16	19.02970297	4.464203054	240	1.256980509	216	4	54
24	19.04950495	4.315961625	240.009901	1.117989709	216.0089109	4	54.00222772
32	19.00990099	5.117606959	240.019802	1.140001737	216.0178218	4	54.00445545
40	18.88118812	5.657361803	239.970297	1.220290503	215.9732673	4	53.99331683
80:32,8	19.16831683	3.797550018	240.009901	1.126898838	216.0089109	4	54.00222772
16	19.02970297	4.497678169	240	1.183215957	216	4	54
24	19.12871287	4.778416822	239.990099	1.178940622	215.9910891	4	53.99777228
32	18.78217822	6.014322839	240	1.349073756	216	4	54
40	18.91089109	5.889140871	239.990099	1.212394734	215.9910891	4	53.99777228
80:40,8	18.95049505	4.131286089	240	1.166190379	216	4	54
16	18.57425743	4.742038664	240	1.208304597	216	4	54
24	18.64356436	5.021123696	239.990099	1.195784675	215.9910891	4	53.99777228
32	18.75247525	5.779975676	239.990099	1.260912761	215.9910891	4	53.99777228
40	18.61386139	6.187035311	240	1.208304597	216	4	54

Table B.9: Robot Response to Proportional(80),Integral and Differential Action

P:I,D	Pwm Average	S.D.	Clix Average	S.D.	Ramp90%	90%#	RampRate
100:10,10	18.9009901	4.048468724	239.9405941	1.08463618	215.9465347	4	53.98663366
20	18.54455446	4.676590109	240.009901	1.117989709	216.0089109	5	43.20178218
30	18.15841584	5.833922648	240.039604	1.248365268	216.0356436	5	43.20712871
40	18.43564356	5.590019395	239.980198	1.086095742	215.9821782	5	43.19643564
50	18.28712871	6.403649949	240	1.140175425	216	5	43.2
100:20,10	18.44554455	5.062559131	240	1.29614814	216	5	43.2
20	18.8019802	5.142022563	239.970297	1.178604646	215.9732673	5	43.19465347
30	18.55445545	5.620454159	240.019802	1.174565435	216.0178218	5	43.20356436
40	18.43564356	5.28093901	239.990099	1.024646764	215.9910891	5	43.19821782
50	18.72277228	6.991593255	240	1.191637529	216	5	43.2
100:30,10	17.88118812	5.269320884	240.019802	1.295995355	216.0178218	5	43.20356436
20	18.08910891	5.278444865	240.009901	1.178940622	216.0089109	5	43.20178218
30	18.42574257	4.682620067	240.019802	0.958959833	216.0178218	5	43.20356436
40	18.03960396	6.572550178	240	1.208304597	216	5	43.2
50	17.64356436	7.33291778	240	1.288409873	216	5	43.2
100:40,10	18.47524752	4.920557	239.990099	1.153213332	215.9910891	5	43.19821782
20	18.2970297	5.714095824	240.029703	1.203789396	216.0267327	5	43.20534653
30	18.24752475	6.018979881	240	1.166190379	216	5	43.2
40	18.20792079	6.030450782	239.990099	1.072334365	215.9910891	5	43.19821782
50	17.30693069	7.852060334	240.029703	1.330078536	216.0267327	5	43.20534653
100:50,10	18.44554455	5.299953297	239.990099	1.187392517	215.9910891	5	43.19821782
20	18.03960396	5.979834098	240.019802	1.232722175	216.0178218	5	43.20356436
30	17.83168317	6.354635012	240	1.2	216	26	8.307692308
40	17.84158416	7.382049408	239.980198	1.280470211	215.9821782	5	43.19643564
50	17.53465347	7.528033417	240.009901	1.212394734	216.0089109	5	43.20178218

Table B.10: Robot Response to Proportional(100),Integral and Differential Action

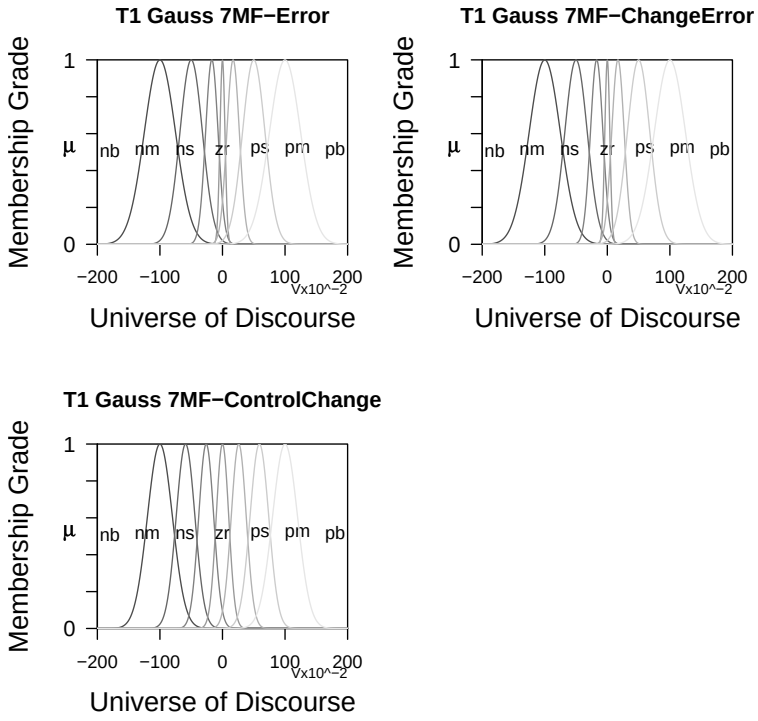
Ident	Size	Name	Value	Comment
Ramp Parameters				
V_RampTicks	4	”rT”	8	num pidTicks to get to max value
V_RampIncrement	4	”RI”	60	Power increments per step
V_RampApplied	4	”rA”	0	Use acceleration ramp(=1)
PID calc controls				
V_PidTickrate	4	”pT”	5	num interrupts per pidTick
V_PidSpeedgain	4	”pVg”	4	speed-gain, ticks to speed-value
V_PidOpOffs	4	”poK”	0	open-loop speed offset
V_PidOpGain	4	”poG”	0	open-loop speed gain term
V_PidDownshift	4	”p>>”	7	pid-calc scaling
PID Controller				
V_PidP	4	”pP”	80	P-term
V_PidI	4	”pI”	16	I-term
V_PidD	4	”pD”	16	D-term
V_PidMaxIerr	4	”pMI”	2048L	anti-windup Imax : half a turn
Fuzzy Controller				
V_FZCPID	4	”pF”	0	Select FZC(=1) or PID(=0)
V_FZCType	4	”pZ”	2	Select FZCType1(=1) or FZCType2(=2)
V_Vote	4	”pV”	0	Program Action Number Selected
Position/Decel controls				
V_PosSpeedScale	4	”xS”	2	dist_error-to-speed scaling
V_PosTolerance	4	”xT”	0	sets up Traction Factor
V_PosMaxdist	4	”xM”	0	sets up Smooth Factor
Wheel Velocity Setup				
V_StopMinDist	4	”qX”	60	sets up Left Motor
V_StopTimeout	4	”qT”	60	sets up Right Motor
Miabot Scaling				
V_ByteScaleSpeed	4	”bV”	0x0800	1024 max speed
V_ByteScaleDistMove	4	”b^”	0	sets up nPwmFZCOffset
V_ByteScaleSteps	4	”bQ”	60	sets up nPwmFZCScale
Debug Variables				
V_DbgPosLeft	4	”Xl”	0	
V_DbgPosRight	4	”Xr”	0	
V_DbgClixrateLeft	2	”Vl”	0	Num left clix per pidTick
V_DbgClixrateRight	2	”Vr”	0	Numright clix per pidTick
i2c retry controls				
V_i2cRetries	2	”iN”	0	
V_i2cRetryTime	2	”iT”	0	
Echo control				
V_bEcho	2	”eE”	0	

Table B.11: Robot Parameter Definition File

Appendix C

Chapter 5 Membership Function Shapes, Surfaces and Tables

This appendix contains the membership function shapes, surfaces and tables for the chapter.



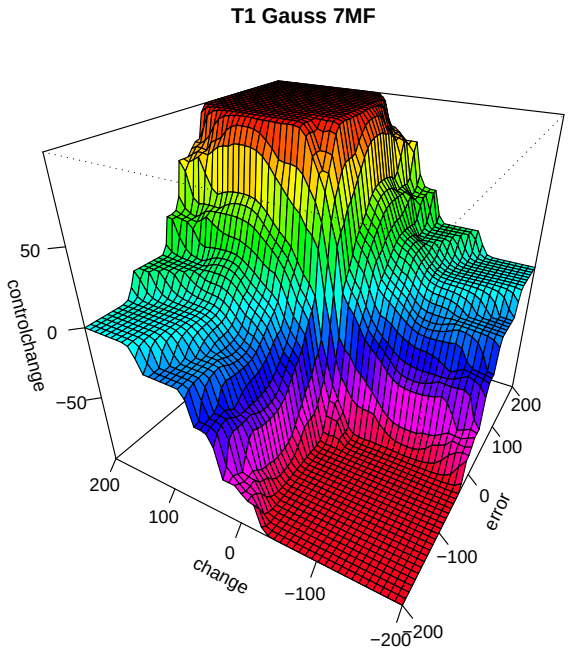
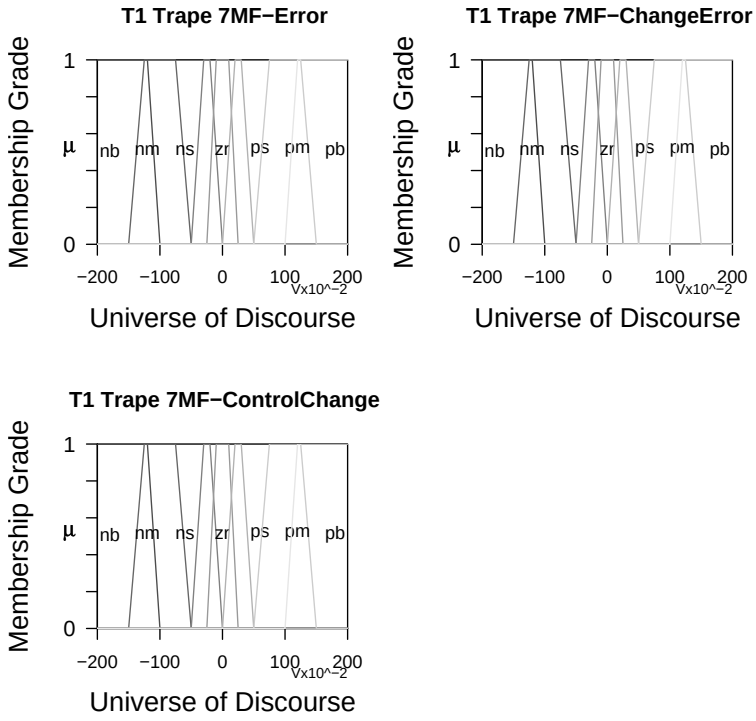


Figure C.1: Type-1 Seven Term Gaussian Membership Shapes and Surface



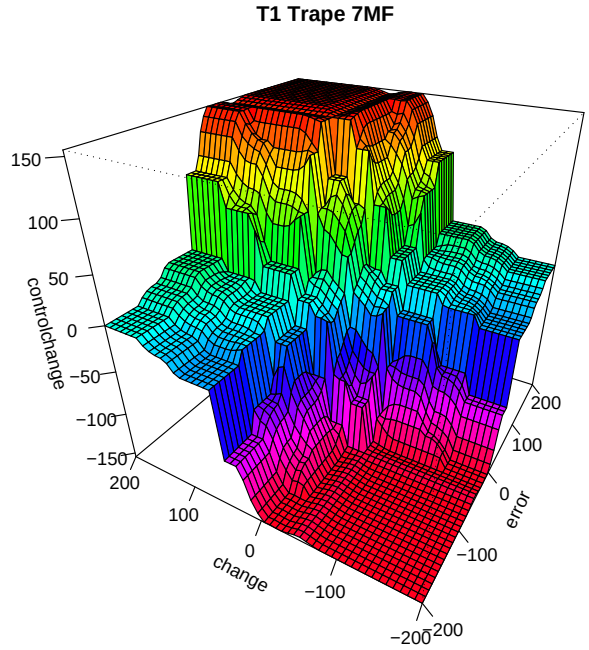
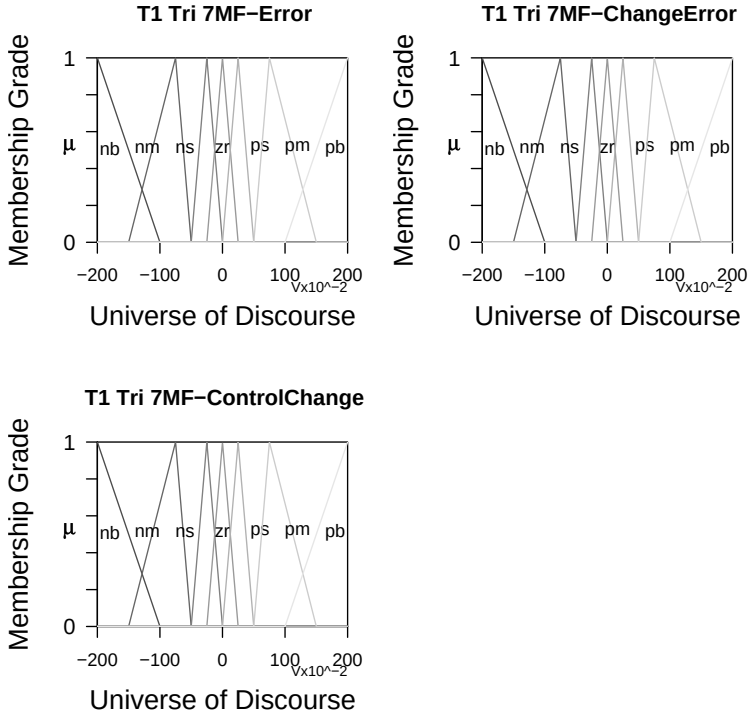


Figure C.2: Type-1 Seven Term Trapezoidal Membership Shapes and Surface



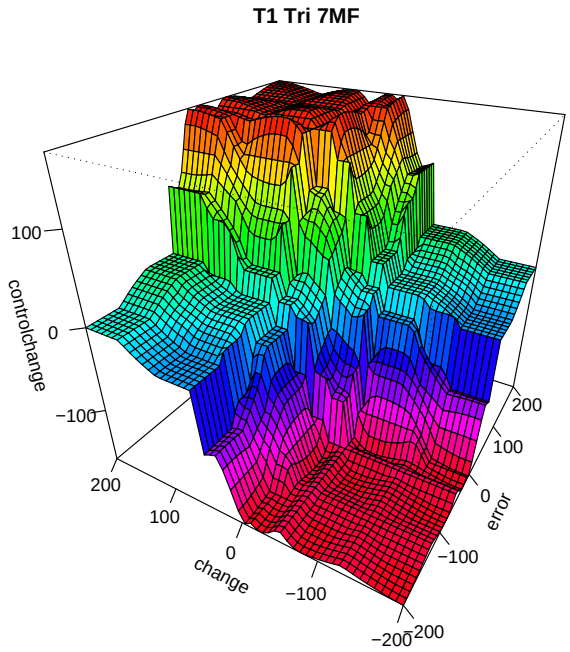
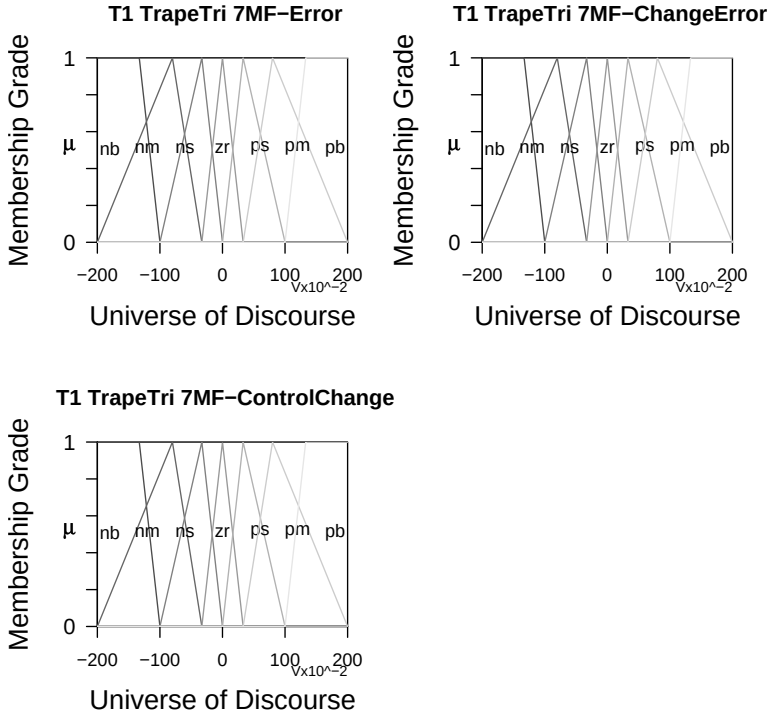


Figure C.3: Type-1 Seven Term Triangular Membership Shapes and Surface



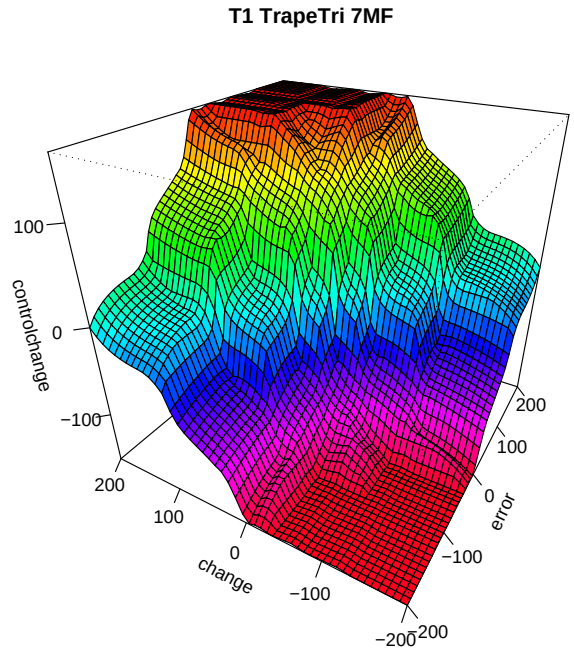


Figure C.4: Type-1 Seven Term Trapezoidal Triangular Membership Shapes and Surface

TYPE-1							
Variables	nb	nm	ns	ze	ps	pm	pb
Change μ	-100	-50	-17	0	17	50	100
Error σ	25	18	10	4	10	18	25
Error μ	-100	-50	-17	0	17	50	100
σ	25	18	10	4	10	18	25
Control μ	-100	-59	-26	0	26	59	100
Change σ	20	15	12	10	12	15	20
TYPE-2							
Variables	nb	nm	ns	ze	ps	pm	pb
Change μ1	-100	-50.0	-17	-2.0	17	50.0	100
Error μ2	-90	-45.0	-15	2.0	15	45.0	90
σ1	25	18.0	10	4.0	10	18.0	25
σ2	23	16.5	9	3.5	9	16.5	23
Error μ1	-100	-50.0	-17	-2.0	17	50.0	100
μ2	-90	-45.0	-15	2.0	15	45.0	90
σ1	25	18.0	10	4.0	10	18.0	25
σ2	23	16.5	9	3.5	9	16.5	23
Control μ1	-100	-59.0	-26	-5.0	26	59.0	100
Change μ2	-90	-54.0	-24	5.0	24	54.0	90
σ1	20	15.0	12	10.0	12	15.0	20
σ2	18	13.5	11	9.5	11	13.5	18

Table C.1: Seven Term Type-1 and Type-2 Gauss Parameters

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200	-150	-50	-25	0	50	100
Error	-200	-125	-30	-10	20	75	120
	-120	-75	-20	10	30	125	200
	-100	-50	0	25	50	150	200
Error	-200	-150	-50	-25	0	50	100
	-200	-125	-30	-10	20	75	120
	-120	-75	-20	10	30	125	200
	-100	-50	0	25	50	150	200
Control	-200	-150	-50	-25	0	50	100
Change	-200	-125	-30	-10	20	75	120
	-120	-75	-20	10	30	125	200
	-100	-50	0	25	50	150	200

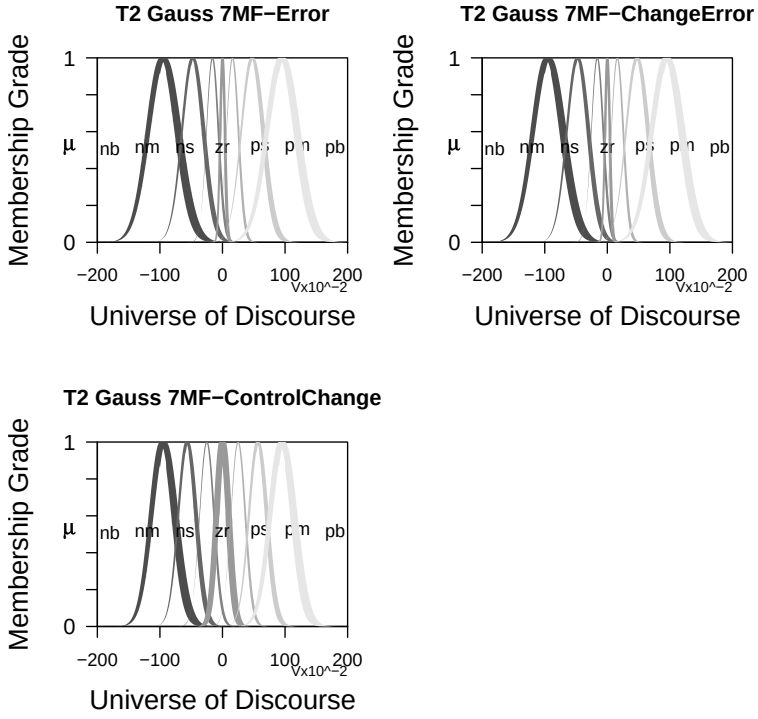
Table C.2: Type-1 Seven Term Trapezoidal MF Parameters

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200	-150	-50	-25	0	50	100
Error	-200	-75	-25	0	25	75	200
	-100	-50	0	25	50	150	200
Error	-200	-150	-50	-25	0	50	100
	-200	-75	-25	0	25	75	200
	-100	-50	0	25	50	150	200
Control	-200	-150	-50	-25	0	50	100
Change	-200	-75	-25	0	25	75	200
	-100	-50	0	25	50	150	200

Table C.3: Type-1 Seven Term Triangular MF Parameters

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200	-200	-100	-33	0	33	100
Error	-200	-80	-33	0	33	80	133
	-133	-33	0	33	100	200	200
	-100						200
Error	-200	-200	-100	-33	0	33	100
	-200	-80	-33	0	33	80	133
	-133	-33	0	33	100	200	200
	-100						200
Control	-200	-200	-100	-33	0	33	100
Change	-200	-80	-33	0	33	80	133
	-133	-33	0	33	100	200	200
	-100						200

Table C.4: Type-1 Seven Term Trapezoidal Triangular1 MF Parameters



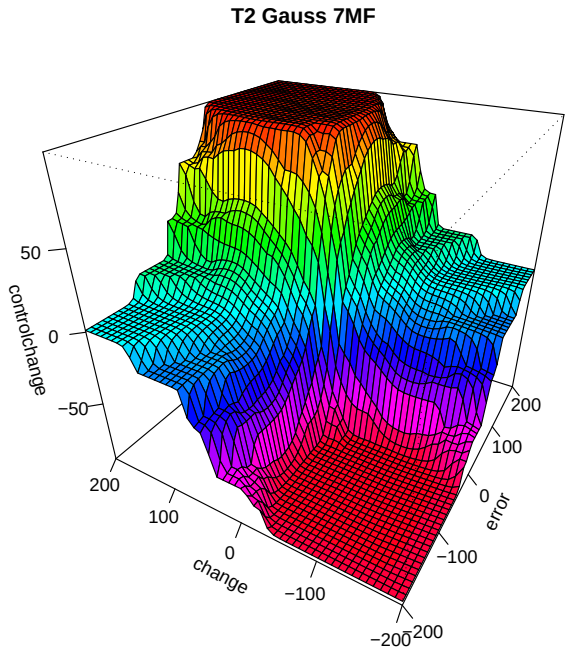
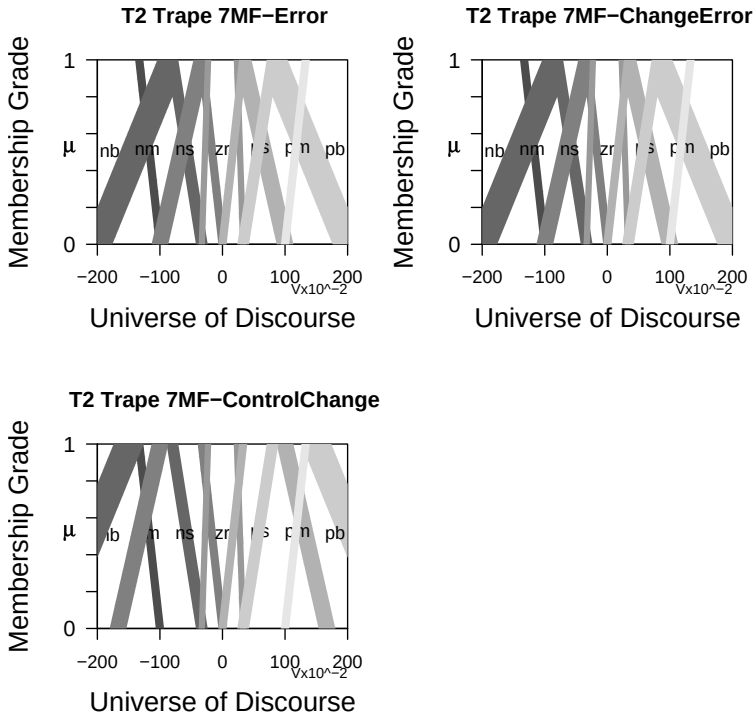


Figure C.5: Type-2 Seven Term Gaussian Membership Shapes and Surface



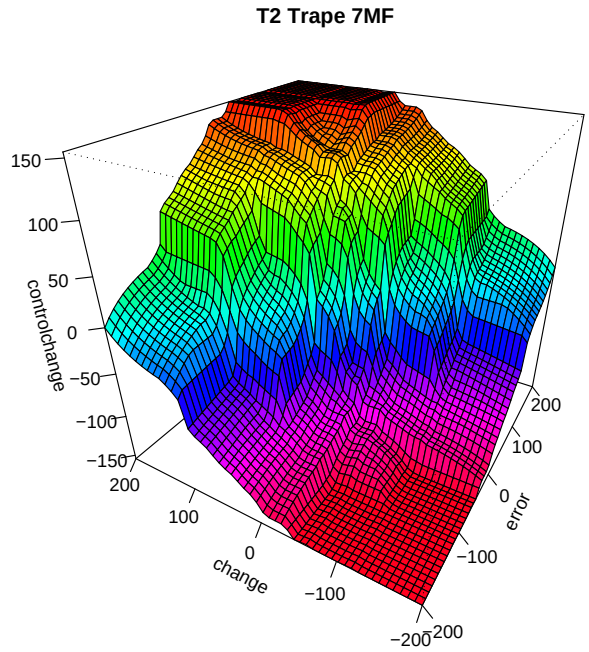
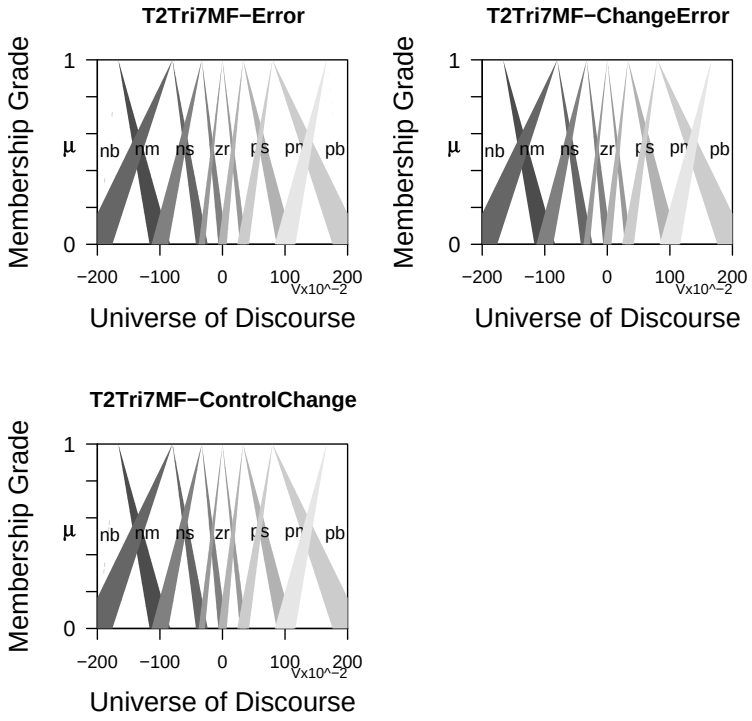


Figure C.6: Type-2 Seven Term Trapezoidal Membership Shapes and Surface



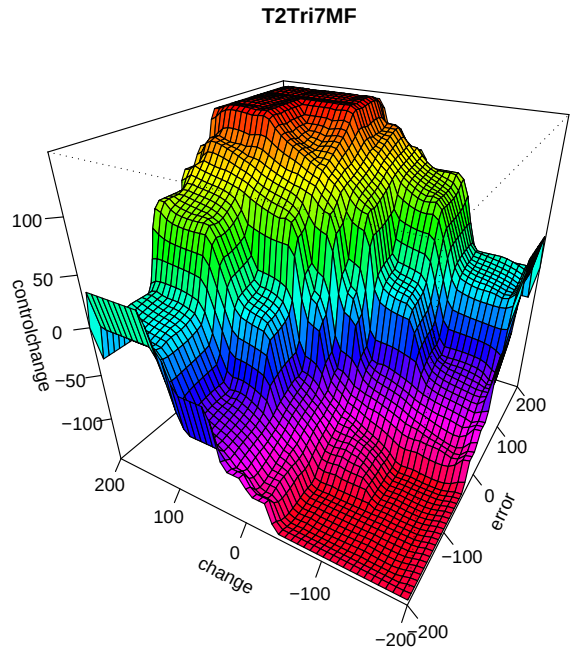
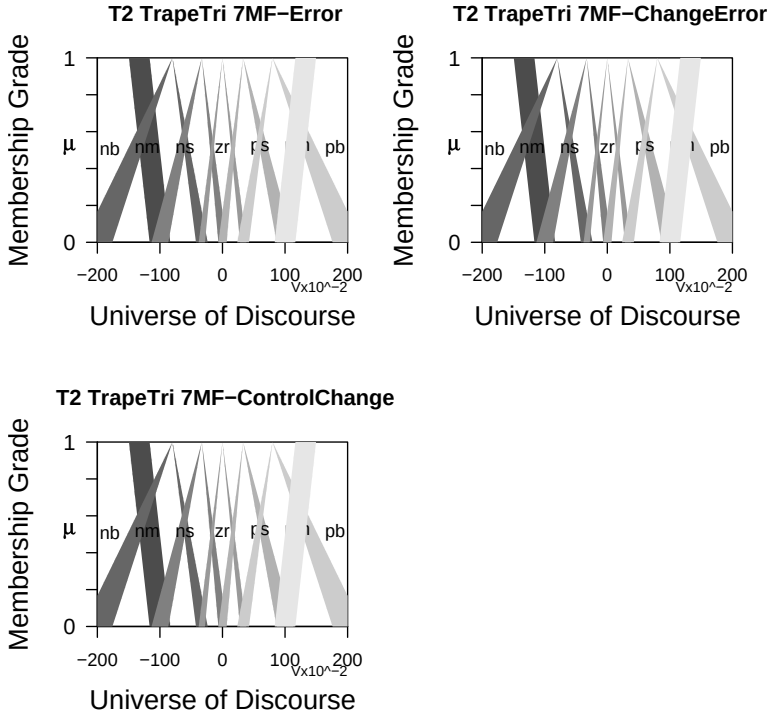


Figure C.7: Type-2 Seven Term Triangular Membership Shapes and Surface



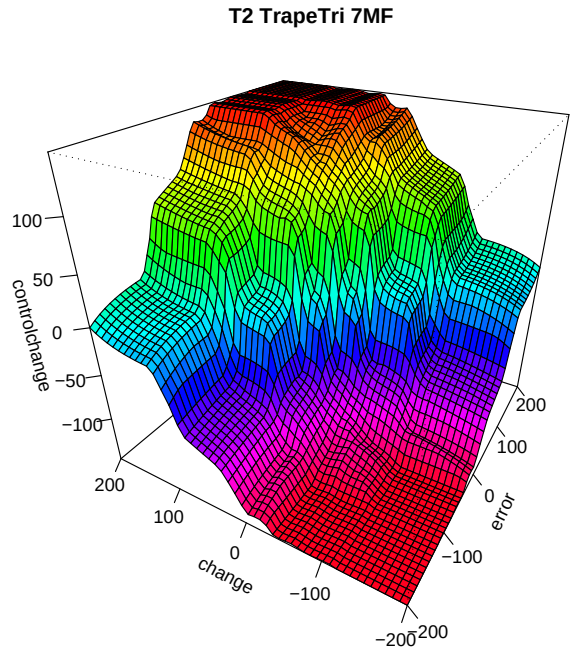
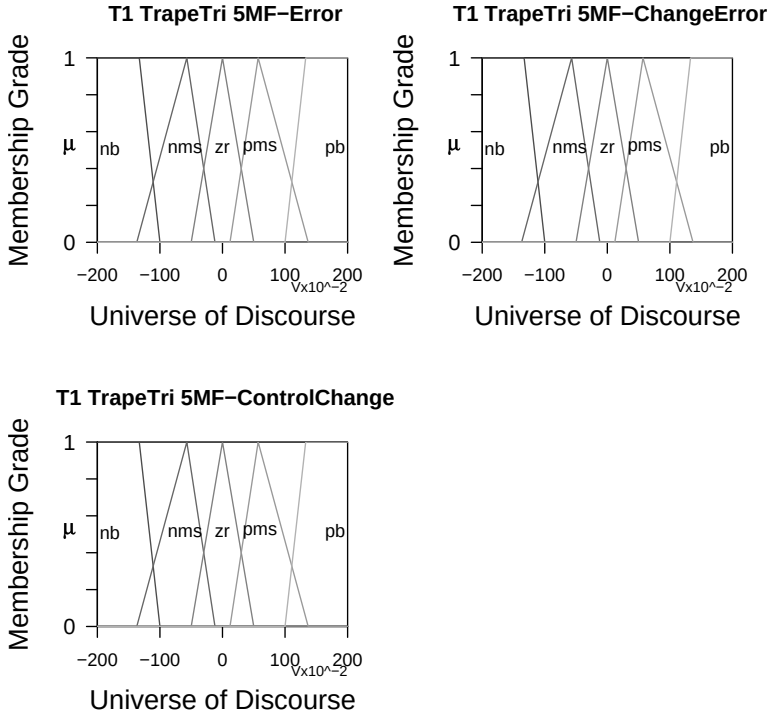


Figure C.8: Type-2 Seven Term Trapezoidal Triangular Membership Shapes and Surface

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200	-294	-180.4	-38	-6.6	23.6	83.5
Error 1	-200	-174	-113.4	-28	26.4	70.6	116.5
	-116.5	-70.6	-26.4	28	113.4	174	200
	-83.5	-23.6	6.6	38	180.4	294	200
Change	-200	-246	-153.6	-28	6.5	42.4	116.5
Error 2	-200	-126	-86.6	-18	39.6	89.4	149.5
	-149.5	-89.4	-39.6	18	86.6	126	200
	-116.5	-42.4	-6.5	28	153.6	246	200
Error 1	-200	-294	-180.4	-38	-6.6	23.6	83.5
	-200	-174	-113.4	-28	26.4	70.6	116.5
	-116.5	-70.6	-26.4	28	113.4	174	200
	-83.5	-23.6	6.6	38	180.4	294	200
Error 2	-200	-246	-153.6	-28	6.5	42.4	116.5
	-200	-126	-86.6	-18	39.6	89.4	149.5
	-149.5	-89.4	-39.6	18	86.6	126	200
	-116.5	-42.4	-6.5	28	153.6	246	200
Control	-200	-294	-180.4	-38	-6.6	23.6	83.5
Change1	-200	-174	-113.4	-28	26.4	70.6	116.5
	-116.5	-70.6	-26.4	28	113.4	174	200
	-83.5	-23.6	6.6	38	180.4	294	200
Control	-200	-246	-153.6	-28	6.5	42.4	116.5
Change2	-200	-126	-86.6	-18	39.6	89.4	149.5
	-149.5	-89.4	-39.6	18	86.6	126	200
	-116.5	-42.4	-6.5	28	153.6	246	200

Table C.5: Type-2 Seven Term Trapezoidal MF Parameters



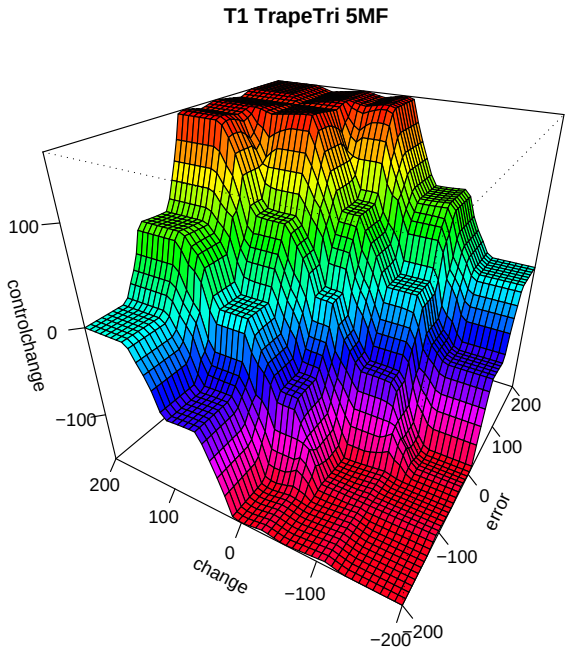


Figure C.9: Type-1 Five Term Trapezoidal Triangular Membership Shapes and Surface

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200	-224	-113.4	-38	-6.6	23.6	83.5
Error 1	-166.5	-80	-33	0	33	80	166.5
	-83.5	-23.6	6.6	38	113.4	224	200
Change	-200	-176	-86.6	-28	6.6	42.4	116.5
Error 2	-166.5	-80	-33	0	25	80	166.5
	-116.5	-42.4	-6.6	28	86.6	176	200
Error 1	-200	-224	-113.4	-38	-6.6	23.6	83.5
	-166.5	-80	-33	0	33	80	166.5
	-83.5	-23.6	6.6	38	113.4	224	200
Error 2	-200	-176	-86.6	-28	6.6	42.4	116.5
	-166.5	-80	-33	0	25	80	166.5
	-116.5	-42.4	-6.6	28	86.6	176	200
Control	-200	-224	-113.4	-38	-6.6	23.6	83.5
Change 1	-166.5	-80	-33	0	33	80	166.5
	-83.5	-23.6	6.6	38	113.4	224	200
Control	-200	-176	-86.6	-28	6.6	42.4	116.5
Change 2	-166.5	-80	-33	0	25	80	166.5
	-116.5	-42.4	-6.6	28	86.6	176	200

Table C.6: Type-2 Seven Term Triangular MF Parameters

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-200						83.5
Error 1	-200	-224	-113.4	-38	-6.6	23.6	116.5
	-116.5	-80	-33.	0	33	80	200
	-83.5	-23.6	6.6	38	113.4	224	200
Change	-200	-176	-86.6	-28	6.6	42.4	116.5
Error 2	-200	-80	-33	0	33	80	149.5
	-149.5	-42.4	-6.6	28	86.6	176	200
	-116.5						200
Error 1	-200						83.5
	-200	-224	-113.4	-38	-6.6	23.6	116.5
	-116.5	-80	-33.	0	33	80	200
	-83.5	-23.6	6.6	38	113.4	224	200
Error 2	-200	-176	-86.6	-28	6.6	42.4	116.5
	-200	-80	-33	0	33	80	149.5
	-149.5	-42.4	-6.6	28	86.6	176	200
	-116.5						200
Control	-200						83.5
Change 1	-200	-224	-113.4	-38	-6.6	23.6	116.5
	-116.5	-80	-33.	0	33	80	200
	-83.5	-23.6	6.6	38	113.4	224	200
Control	-200	-176	-86.6	-28	6.6	42.4	116.5
Change 2	-200	-80	-33	0	33	80	149.5
	-149.5	-42.4	-6.6	28	86.6	176	200
	-116.5						200

Table C.7: Type-2 Seven Term Trapezoidal Triangular1 MF Parameters

Variables	nb	nms	ze	pms	pb
Change	-200	-137	-50	12	100
Error	-200	-57	0	57	133
	-133	-12	50	137	200
	-100				200
Error	-200	-137	-50	12	100
	-200	-57	0	57	133
	-133	-12	50	137	200
	-100				200
Control	-200	-137	-50	12	100
Change	-200	-57	0	57	133
	-133	-12	50	137	200
	-100				200

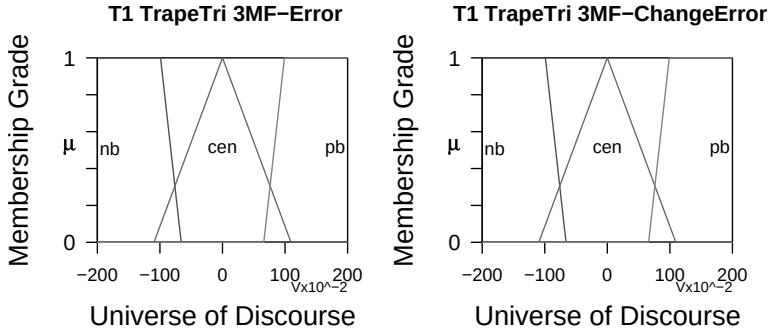
Table C.8: Type-1 Five Term Trapezoidal Triangular1 MF Parameters

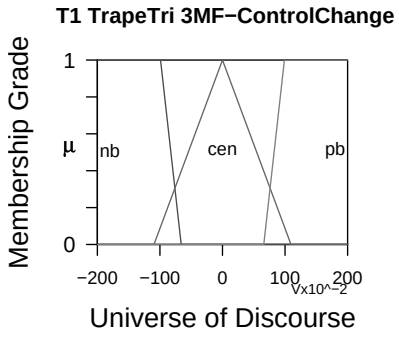
Variables	nb	cen	pb
Change	-200	-109.5	66
Error	-200	0	99
	-99	109.5	200
	-66		200
Error	-200	-109.5	66
	-200	0	99
	-99	109.5	200
	-66		200
Control	-200	-109.5	66
Change	-200	0	99
	-99	109.5	200
	-66		200

Table C.9: Type-1 Three Term Trapezoidal Triangular1 MF Parameters

Variables	nb	cen	pb
Change	-200	-109.5	66
Error	-200	-31.5	99
	-99	31.5	200
	-66	109.5	200
Error	-200	-109.5	66
	-200	-31.5	99
	-99	31.5	200
	-66	109.5	200
Control	-200	-109.5	66
Change	-200	-31.5	99
	-99	31.5	200
	-66	109.5	200

Table C.10: Type-1 Three Term Trapezoidal MF Parameters





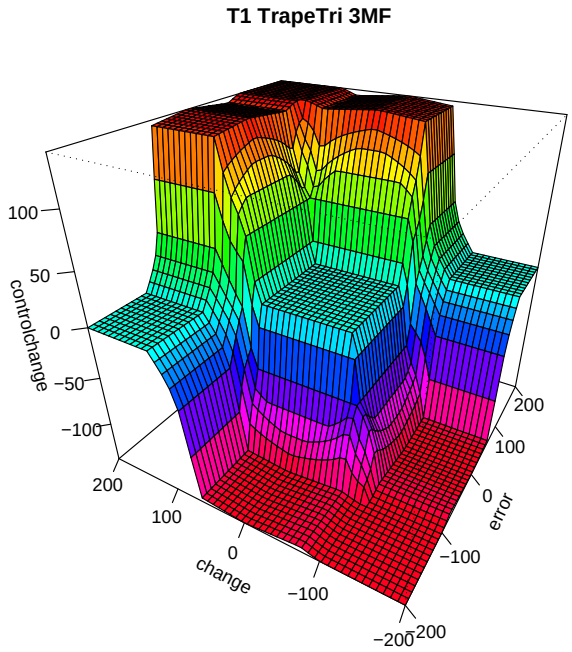
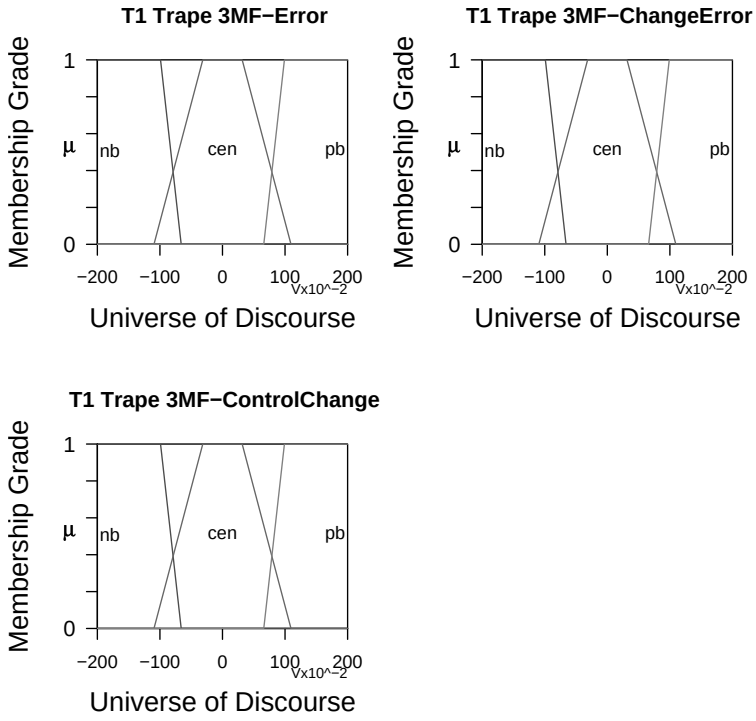


Figure C.10: Type-1 Three Term Trapezoidal Triangular Membership Shapes and Surface



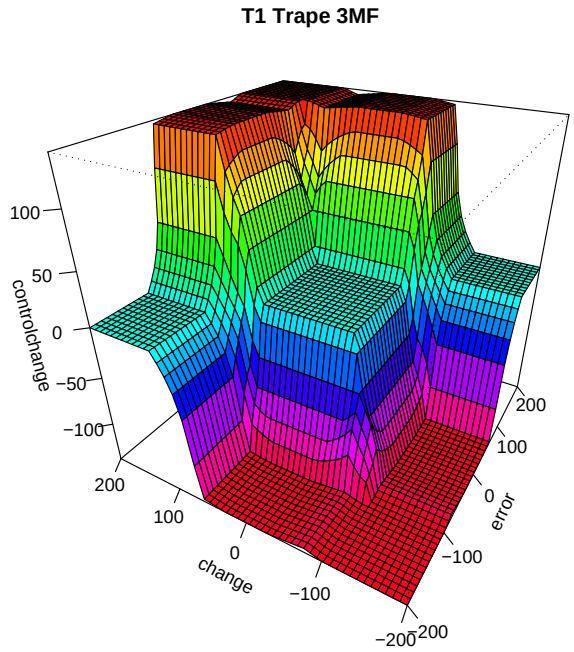
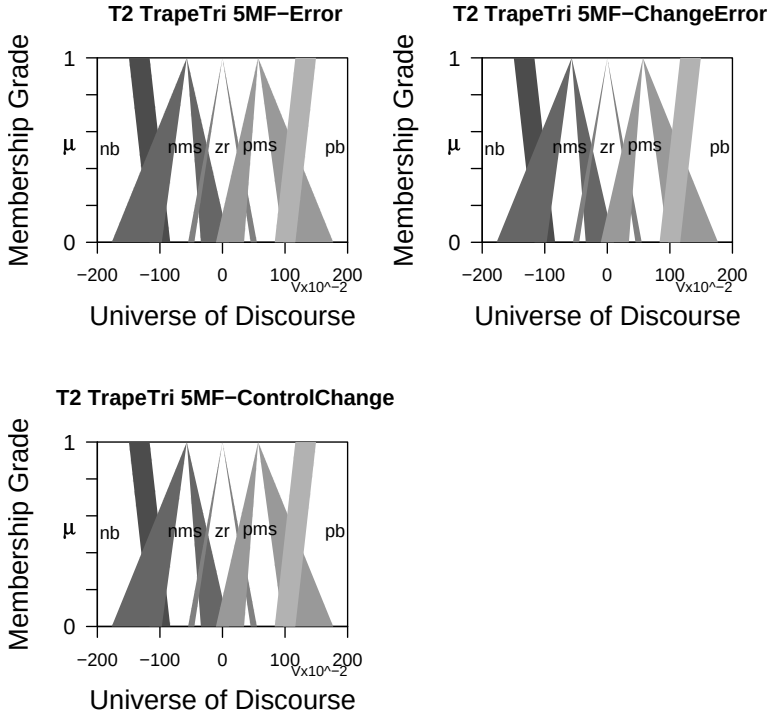


Figure C.11: Type-1 Three Term Trapezoidal Membership Shapes and Surface



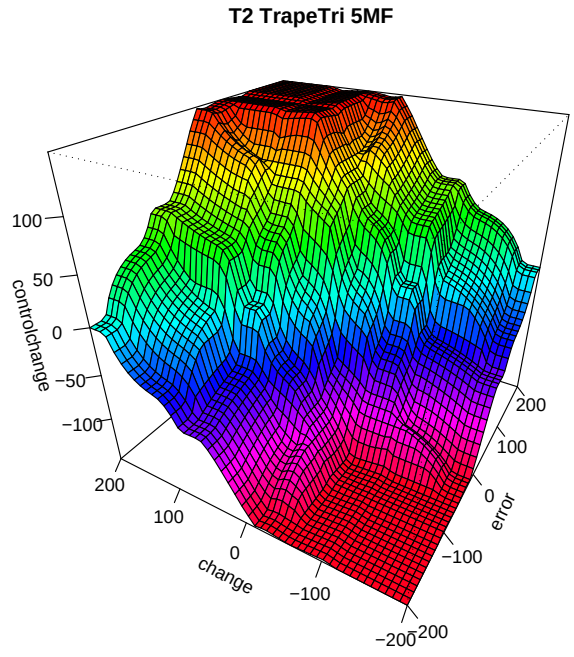
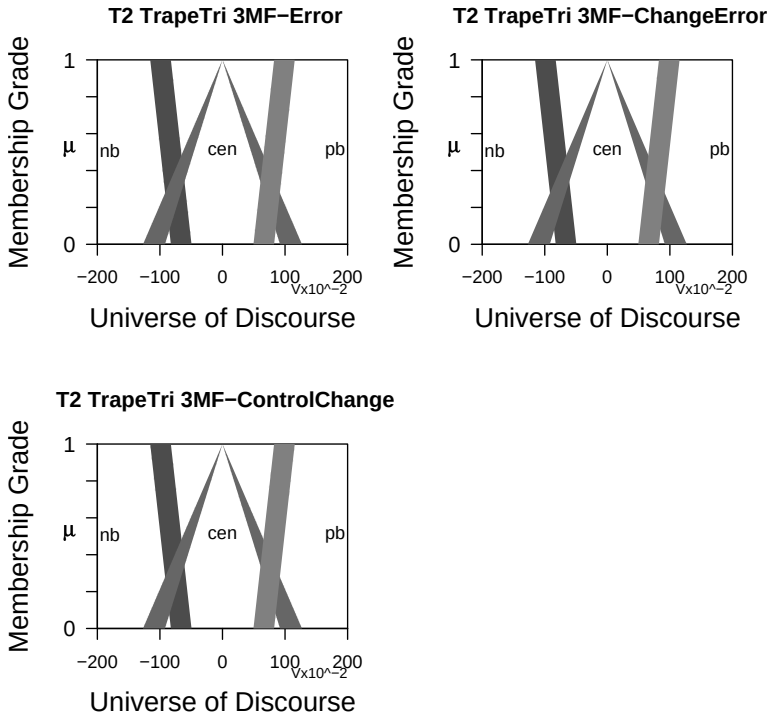


Figure C.12: Type-2 Five Term Trapezoidal Triangular Membership Shapes and Surface

Variables	nb	nms	ze	pms	pb
Change	-200	-177	-55	-10.5	83.5
Error 1	-200	-57	0	57	116.5
	-116.5	10.5	55	177	200
	-83.5				200
Change	-200	-97	-45	34.5	116.5
Error 2	-200	-57	0	57	149.5
	-149.5	-34.5	45	97	200
	-116.5				200
Error 1	-200	-177	-55	-10.5	83.5
	-200	-57	0	57	116.5
	-116.5	10.5	55	177	200
	-83.5				200
Error 2	-200	-97	-45	34.5	116.5
	-200	-57	0	57	149.5
	-149.5	-34.5	45	97	200
	-116.5				200
Control	-200	-177	-55	-10.5	83.5
Change 1	-200	-57	0	57	116.5
	-116.5	10.5	55	177	200
	-83.5				200
Control	-200	-97	-45	34.5	116.5
Change 2	-200	-57	0	57	149.5
	-149.5	-34.5	45	97	200
	-116.5				200

Table C.11: Type-2 Five Term Trapezoidal Triangular1 MF Parameters



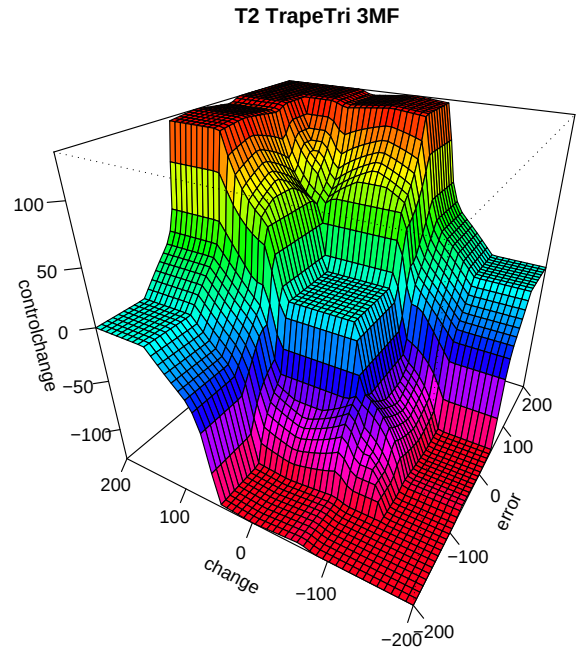
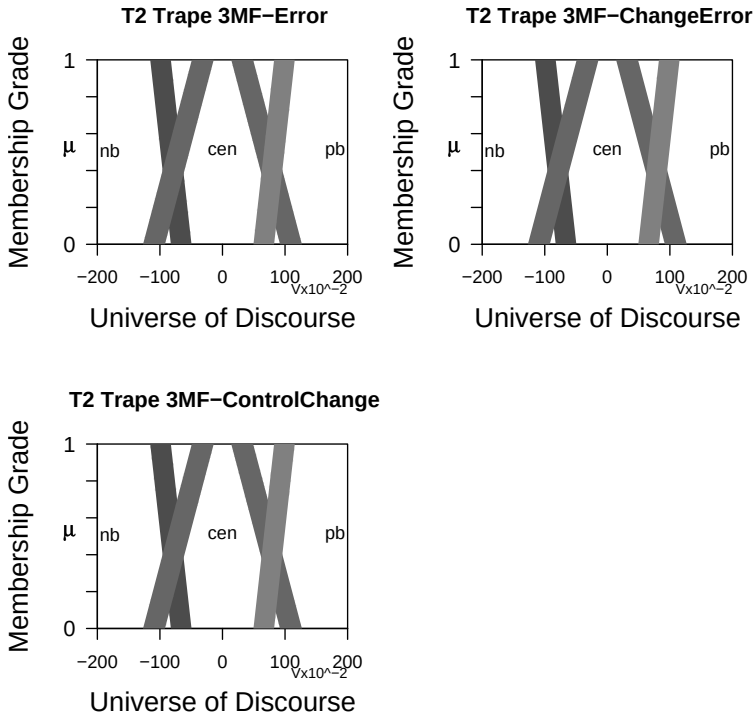


Figure C.13: Type-2 Three Term Trapezoidal Triangular Membership Shapes and Surface

Variables	nb	cen	pb
Change	-200	-127.5	49.5
Error 1	-200	0	82.5
	-82.5	127	200
	-49.5		200
Change	-200	-92	82.5
Error 2	-200	0	115.5
	-115.5	92	200
	-82.5		200
Error 1	-200	-127.5	49.5
	-200	0	82.5
	-82.5	127	200
	-49.5		200
Error 2	-200	-92	82.5
	-200	0	115.5
	-115.5	92	200
	-82.5		200
Control	-200	-127.5	49.5
Change 1	-200	0	82.5
	-82.5	127	200
	-49.5		200
Control	-200	-92	82.5
Change 2	-200	0	115.5
	-115.5	92	200
	-82.5		200

Table C.12: Type-2 Three Term Trapezoidal Triangular1 MF Parameters



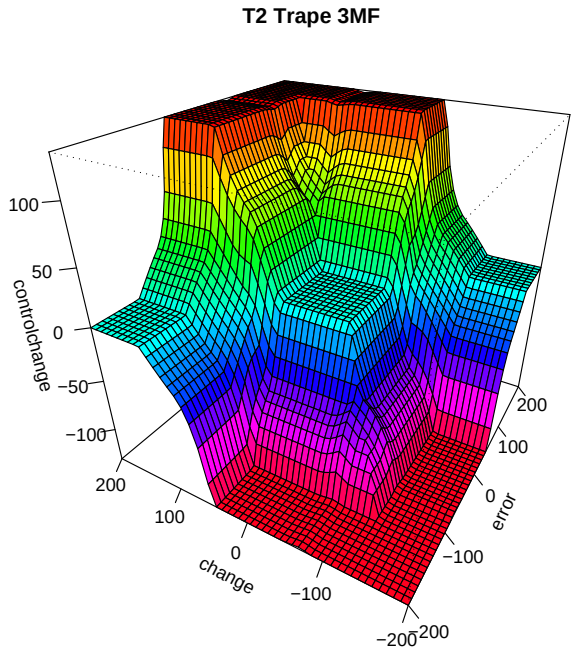


Figure C.14: Type-2 Three Term Trapezoidal Membership Shapes and Surface

Variables	nb	cen	pb
Change	-200	-127.5	49.5
Error 1	-200	-49	82.5
	-82.5	49	200
	-49.5	127	200
Change	-200	-92	82.5
Error 2	-200	-14	115.5
	-115.5	14	200
	-82.5	92	200
Error 1	-200	-127.5	49.5
	-200	-49	82.5
	-82.5	49	200
	-49.5	127	200
Error 2	-200	-92	82.5
	-200	-14	115.5
	-115.5	14	200
	-82.5	92	200
Control	-200	-127.5	49.5
Change 1	-200	-49	82.5
	-82.5	49	200
	-49.5	127	200
Control	-200	-92	82.5
Change 2	-200	-14	115.5
	-115.5	14	200
	-82.5	92	200

Table C.13: Type-2 Three Term Trapezoidal MF Parameters

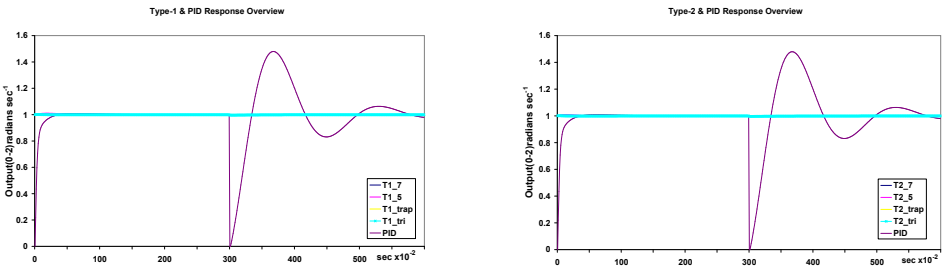


Figure C.15: Type-1 and Type-2 Overview Responses with PID

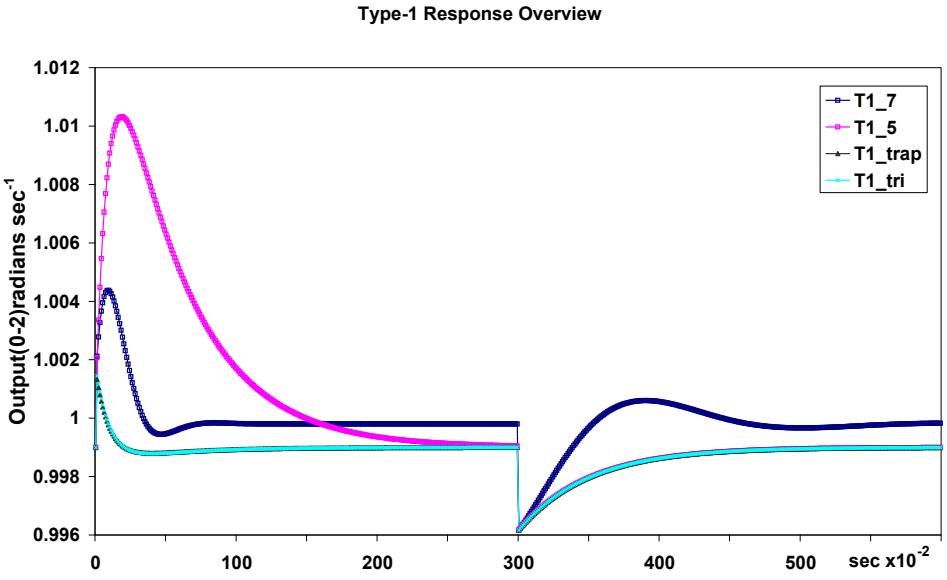


Figure C.16: Type-1 Response Overview

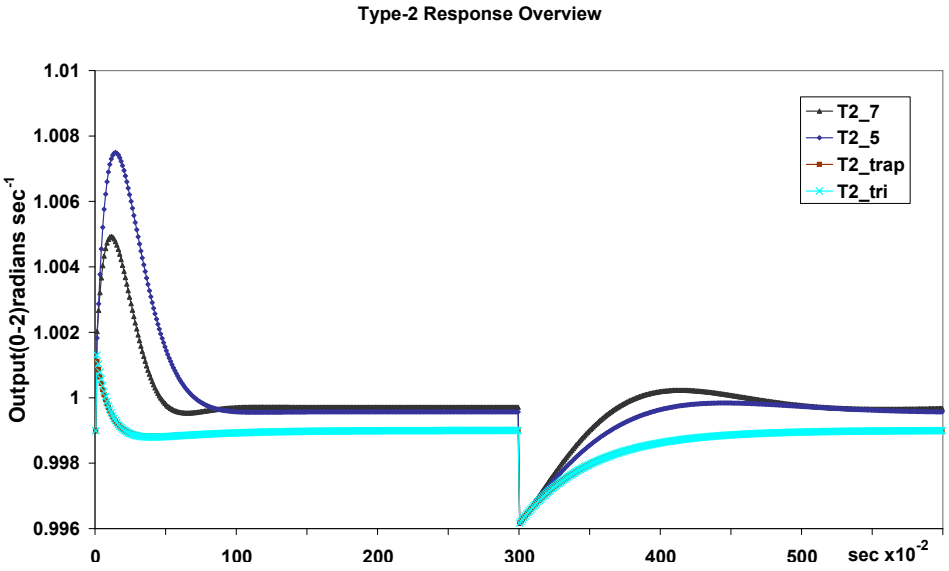


Figure C.17: Type-2 Response Overview

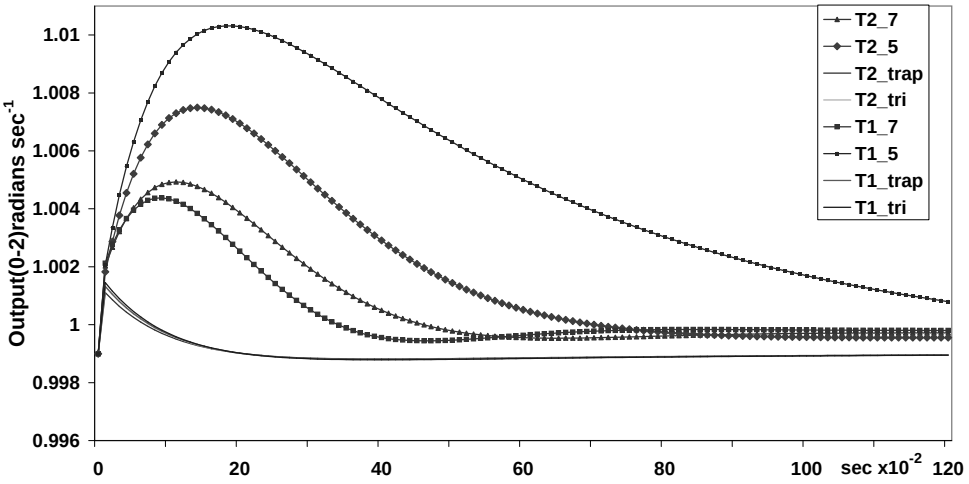


Figure C.18: Fuzzy Logic Controllers Step Response Overview

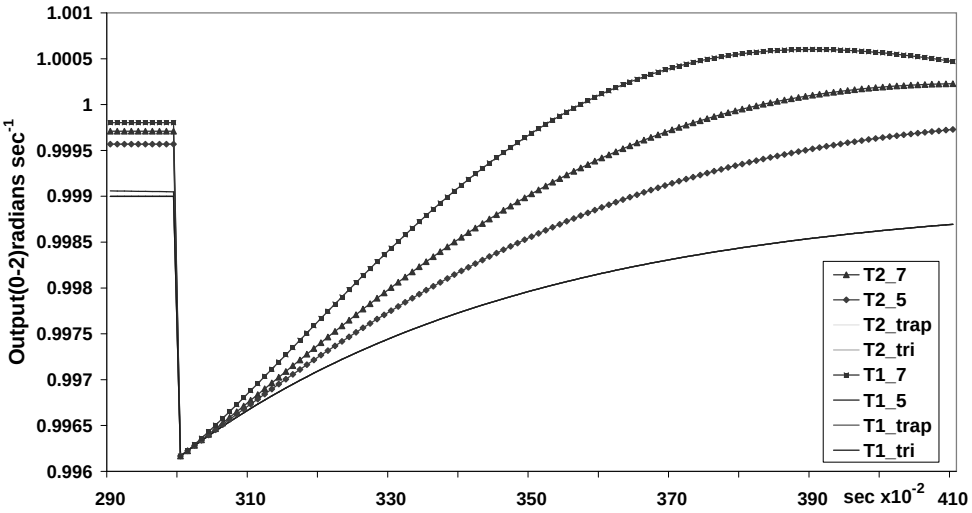


Figure C.19: Fuzzy Logic Controllers Inertia Response Overview

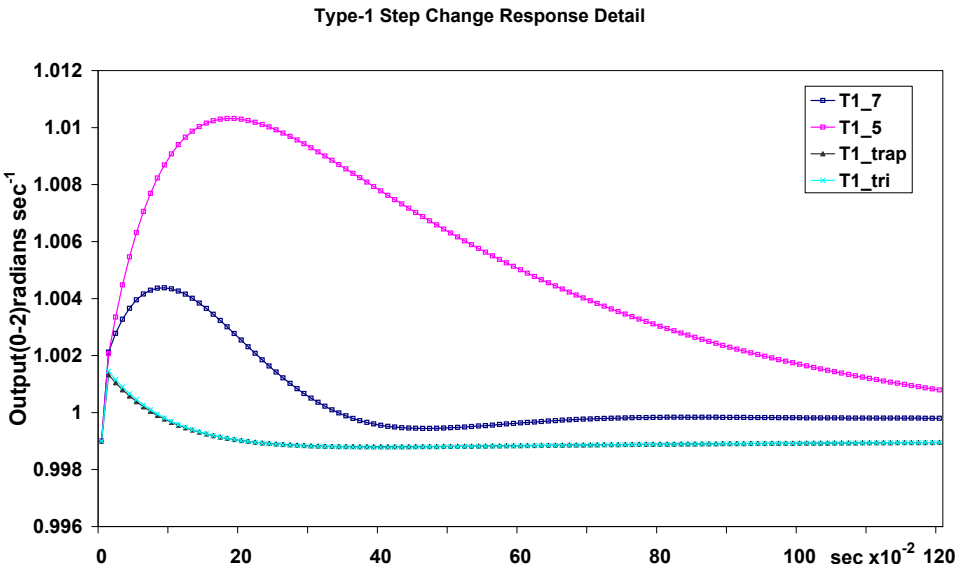


Figure C.20: Type-1 Step Response Overview

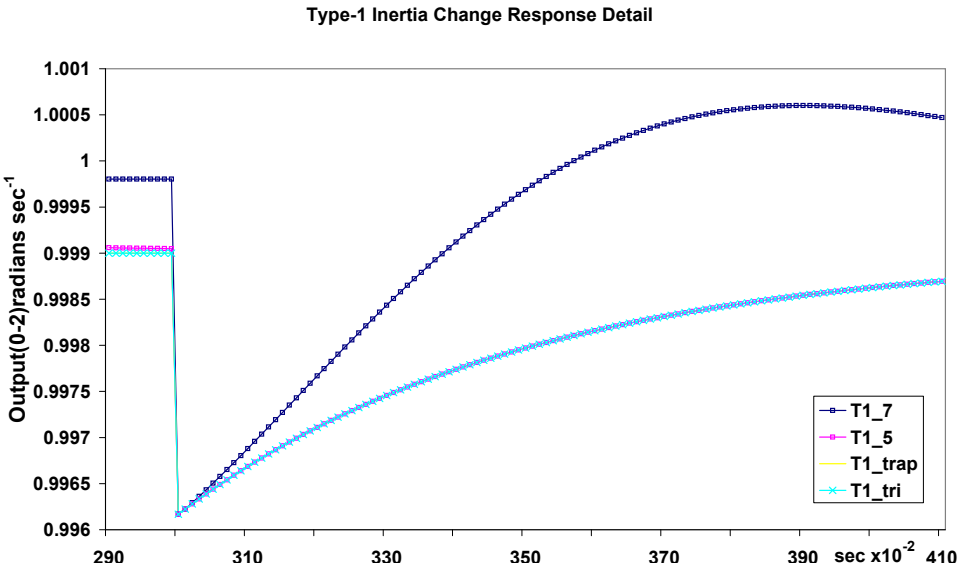


Figure C.21: Type-1 Inertia Response Overview

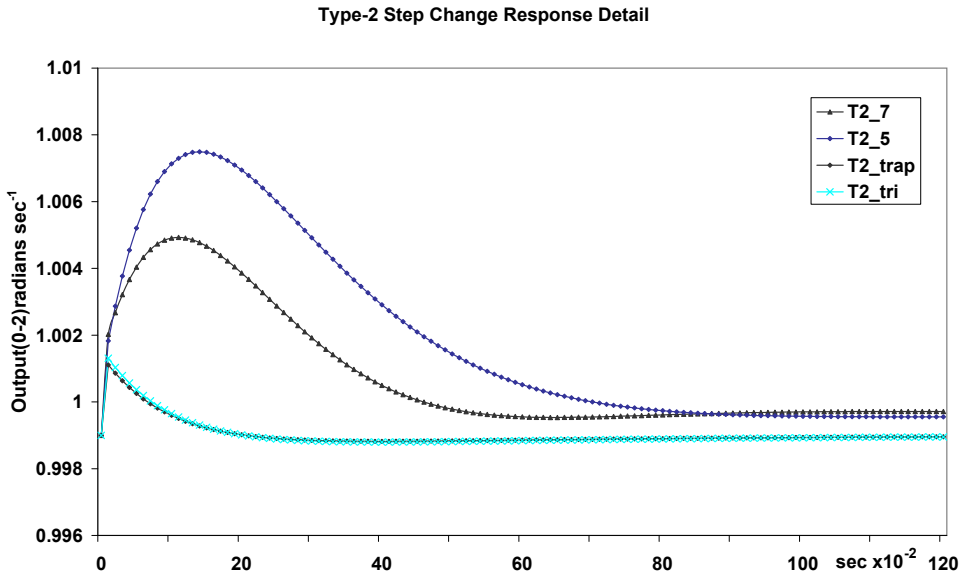


Figure C.22: Type-2 Step Response Overview

Key	Series Without Noise
T2_7	Type-2 Seven Term Trapezoidal Triangular MF
T2_5	Type-2 Five Term Trapezoidal Triangular MF
T2_trap	Type-2 Three Term Trapezoidal MF
T2_tri	Type-2 Three Term Trapezoidal Triangular MF
T1_7	Type-1 Seven Term Trapezoidal Triangular MF
T1_5	Type-1 Five Term Trapezoidal Triangular MF
T1_trap	Type-1 Three Term Trapezoidal MF
T1_tri	Type-1 Three Term Trapezoidal Triangular MF

Table C.14: Key to Type-1 and Type-2 FLC Response Graphs

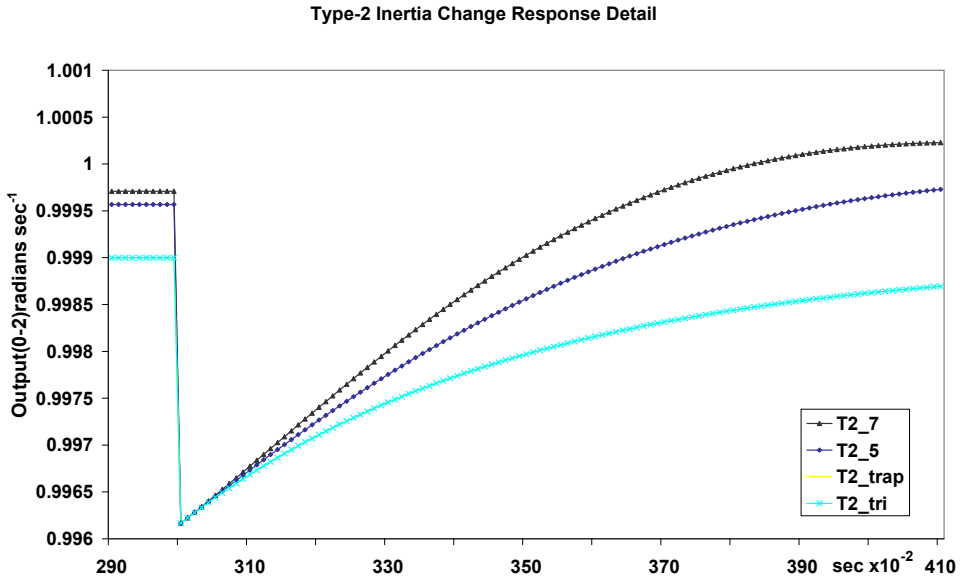


Figure C.23: Type-2 Inertia Response Overview

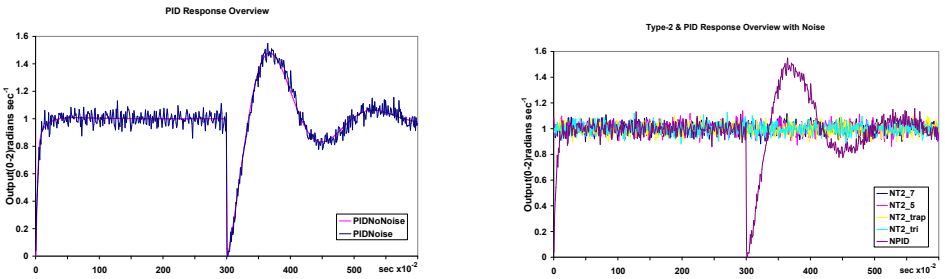


Figure C.24: PID and Type-2 with PID Overview Responses with Noise

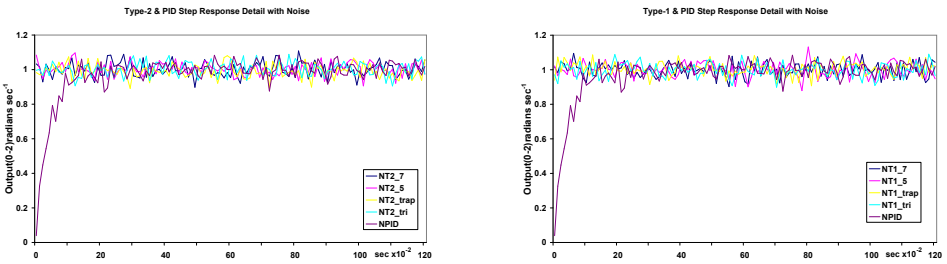


Figure C.25: Type-1 and Type-2 Overview Step Responses with PID and Noise

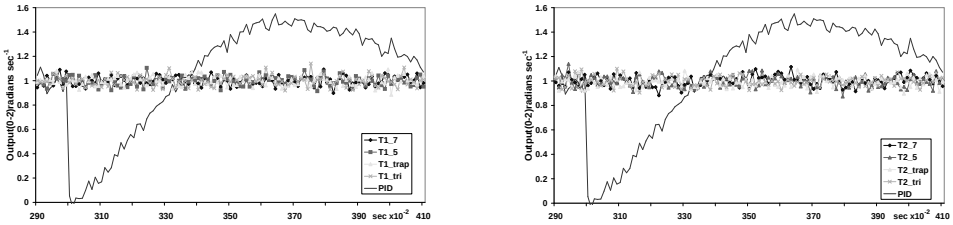


Figure C.26: Type-1 and Type-2 Overview Inertia Responses with PID and Noise

	Type-1	Type-2
7 Term		
90%	Pt1	Pt1
Max	1.004377/Pt9	1.004928/Pt 11
Across SP	Pt34	Pt46
Min	0.999444/Pt46	0.999536/Pt63
Settle	0.999811/Pt104	0.999701/Pt101
5 Term		
90%	Pt1	Pt1
Max	1.010292/Pt17	1.007474/Pt 13
Across SP	Pt150	Pt70
Min	0.999051/Pt300	0.999555/Pt111
Settle	0.999051/Pt300	0.999578/Pt123
3 TermTrap		
90%	Pt1	Pt1
Max	1.001325/Pt1	1.001108/Pt1
Across SP	Pt8	Pt7
Min	0.998803/Pt40	0.998803/Pt40
Settle	0.998980/161	0.998980/158
3 TermTrapTri		
90%	Pt1	Pt1
Max	1.001450/Pt1	1.001307/Pt1
Across SP	Pt8	Pt8
Min	0.998789/Pt40	0.998789/Pt40
Settle	0.998980/Pt165	0.998980/Pt163

Table C.15: Step Change Response Results

	Type-1	Type-2
7 Term		
Dip	0.996167/Pt301	0.996167/Pt301
Across SP	Pt357	Pt383
Max	1.000600/Pt390	1.000215/Pt415
Return SP	Pt445	Pt461
Min	0.999697/Pt495	0.999657/Pt543
5 Term		
Dip	0.996167/Pt301	0.996167/Pt301
Across SP	N/A	N/A
Max	0.998994/Pt600	0.999839/Pt435
Return SP	N/A	N/A
Min	N/A	N/A
3 TermTrap		
Dip	0.996167/Pt301	0.996167/Pt301
Across SP	N/A	N/A
Max	0.998994/Pt600	0.998994/Pt600
Return SP	N/A	N/A
Min	N/A	N/A
3 TermTrapTri		
Dip	0.996167/Pt301	0.996167/Pt301
Across SP	N/A	N/A
Max	0.998994/Pt600	0.998994/Pt600
Return SP	N/A	N/A
Min	N/A	N/A

Table C.16: Inertia Change Response Results

Step	90%	Across SP	Max	Settle
	Pt9	Pt37	1.007978/Pt54	1.001652/Pt139
Inertia	Dip	90%	Across SP	Max
	0/Pt301	Pt331	Pt334	1.479419/Pt367

Table C.17: PID Response Results

Appendix D

Chapter 6 Membership Functions, Surfaces and Tables

This appendix contains the membership function shapes, parameters, surfaces and tables for the chapter. The inputs and results, both summarised and full, of the simulations are reported.

Variables	nb	nm	ns	ze	ps	pm	pb
Change	-201	-175	-115	-55	15	75	135
Error	-201	-145	-85	-25	45	105	165
Outer	-165	-105	-45	25	85	145	201
	-135	-75	-15	55	115	175	201
Change	-201	-165	-105	-45	25	85	145
Error	-201	-135	-75	-15	55	115	175
Inner	-175	-115	-55	15	75	135	201
	-145	-85	-25	45	105	165	201
Error	-201	-175	-115	-55	15	75	135
Outer	-201	-145	-85	-25	45	105	165
	-165	-105	-45	25	85	145	201
	-135	-75	-15	55	115	175	201
Error	-201	-165	-105	-45	25	85	145
Inner	-201	-135	-75	-15	55	115	175
	-175	-115	-55	15	75	135	201
	-145	-85	-25	45	105	165	201
Control	-201	-175	-115	-55	15	75	135
Change	-201	-145	-85	-25	45	105	165
Outer	-165	-105	-45	25	85	145	201
	-135	-75	-15	55	115	175	201
Control	-201	-165	-105	-45	25	85	145
Change	-201	-135	-75	-15	55	115	175
Inner	-175	-115	-55	15	75	135	201
	-145	-85	-25	45	105	165	201

Table D.1: Type-2 Seven Term Trapezoidal MF Parameters

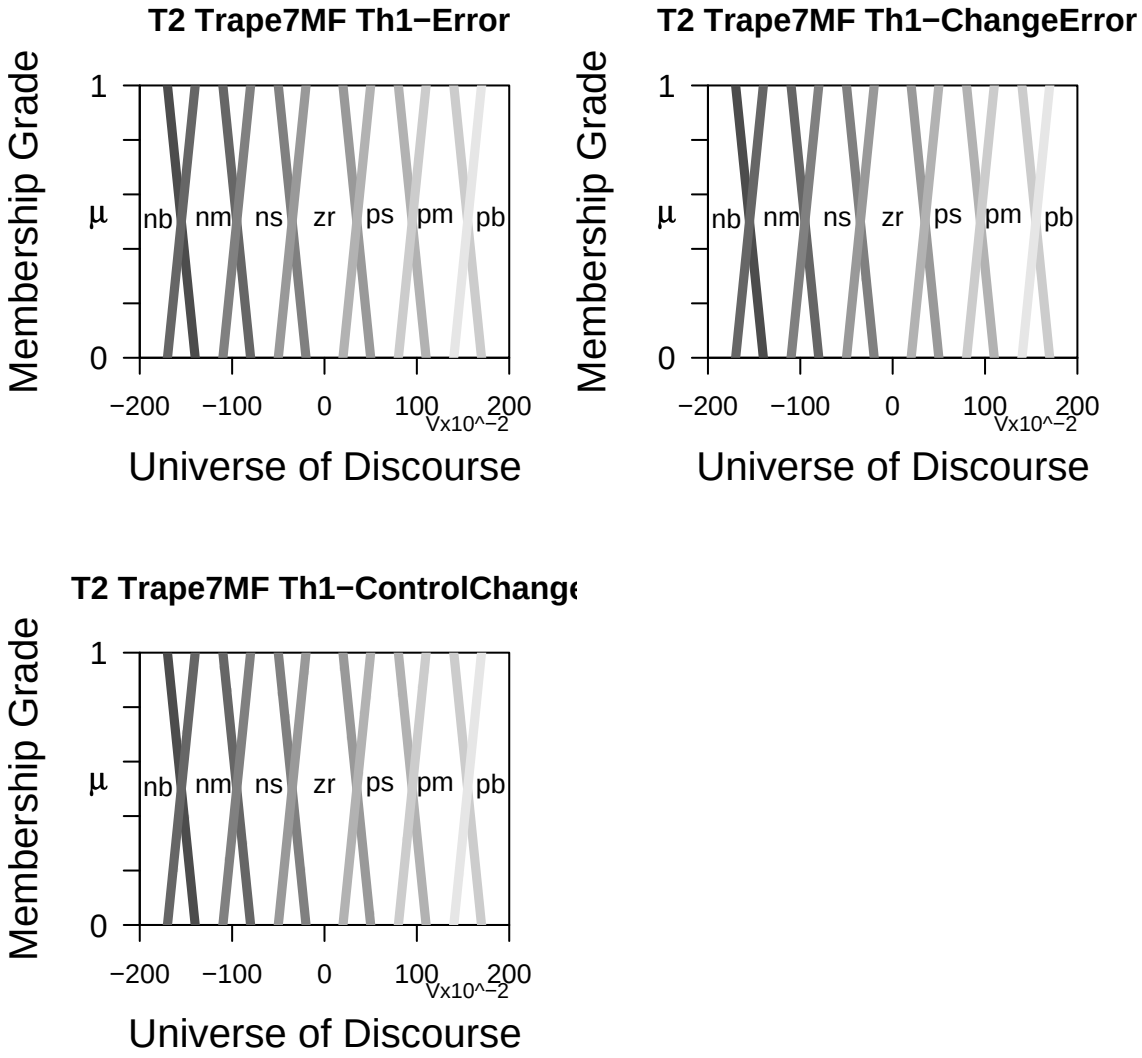


Figure D.1: Type-2 Seven Term Trapezoidal with Threshold - 1

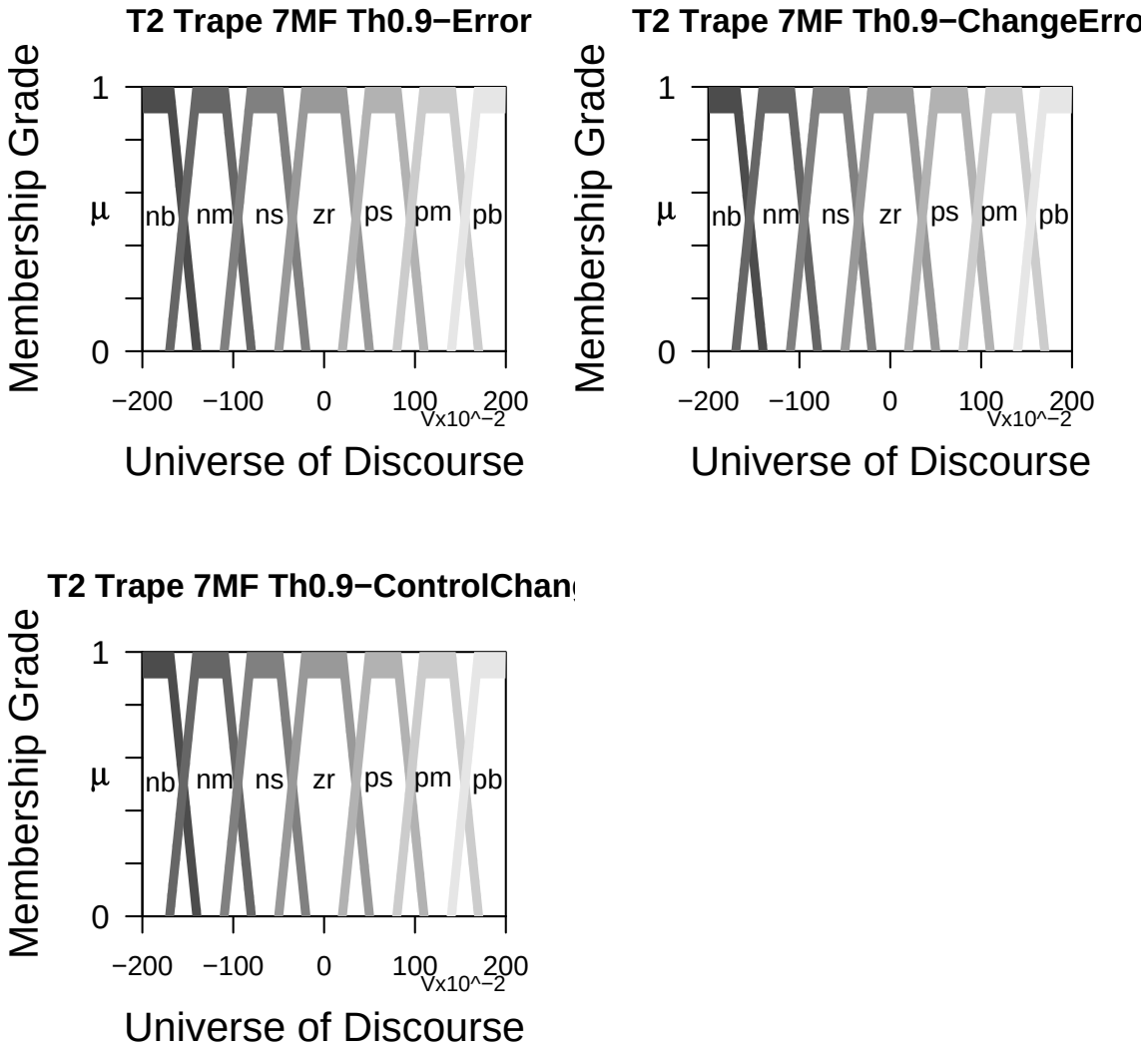


Figure D.2: Type-2 Seven Term Trapezoidal with Threshold - 0.9

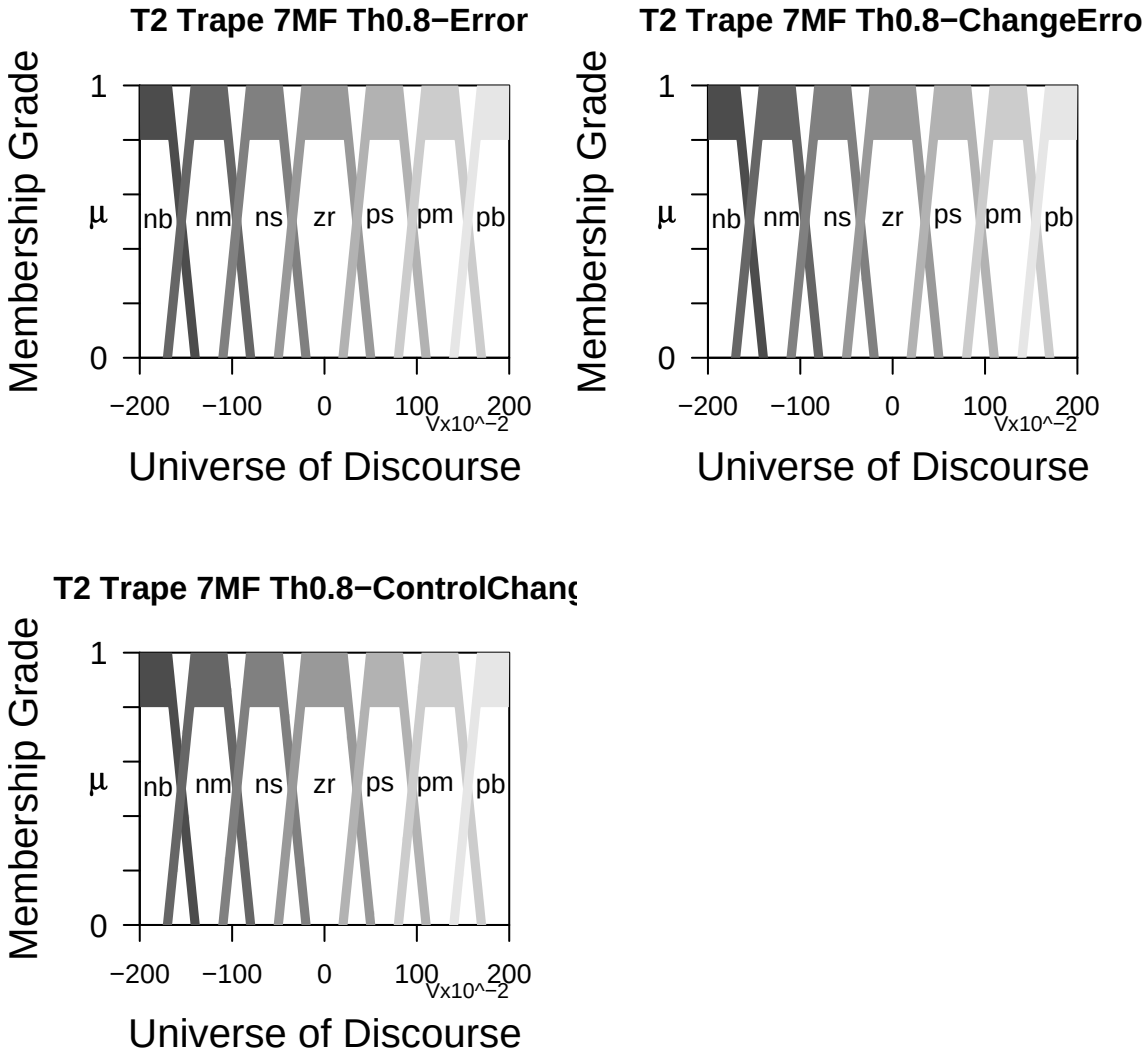


Figure D.3: Type-2 Seven Term Trapezoidal with Threshold - 0.8

Variables	nb	cen	pb
Change	-201	-60	10
Error	-201	-25	55
Outer	-55	25	201
	-10	60	201
Change	-201	-50	20
Error	-201	-15	65
Inner	-65	15	201
	-20	50	201
Error	-201	-60	10
Outer	-201	-25	55
	-55	25	201
	-10	60	201
Error	-201	-50	20
Inner	-201	-15	65
	-65	15	201
	-20	50	201
Control	-201	-60	10
Change	-201	-25	55
Outer	-55	25	201
	-10	60	201
Control	-201	-50	20
Change	-201	-15	65
Inner	-65	15	201
	-20	50	201

Table D.2: Type-2 Three Term Trapezoidal MF Parameters

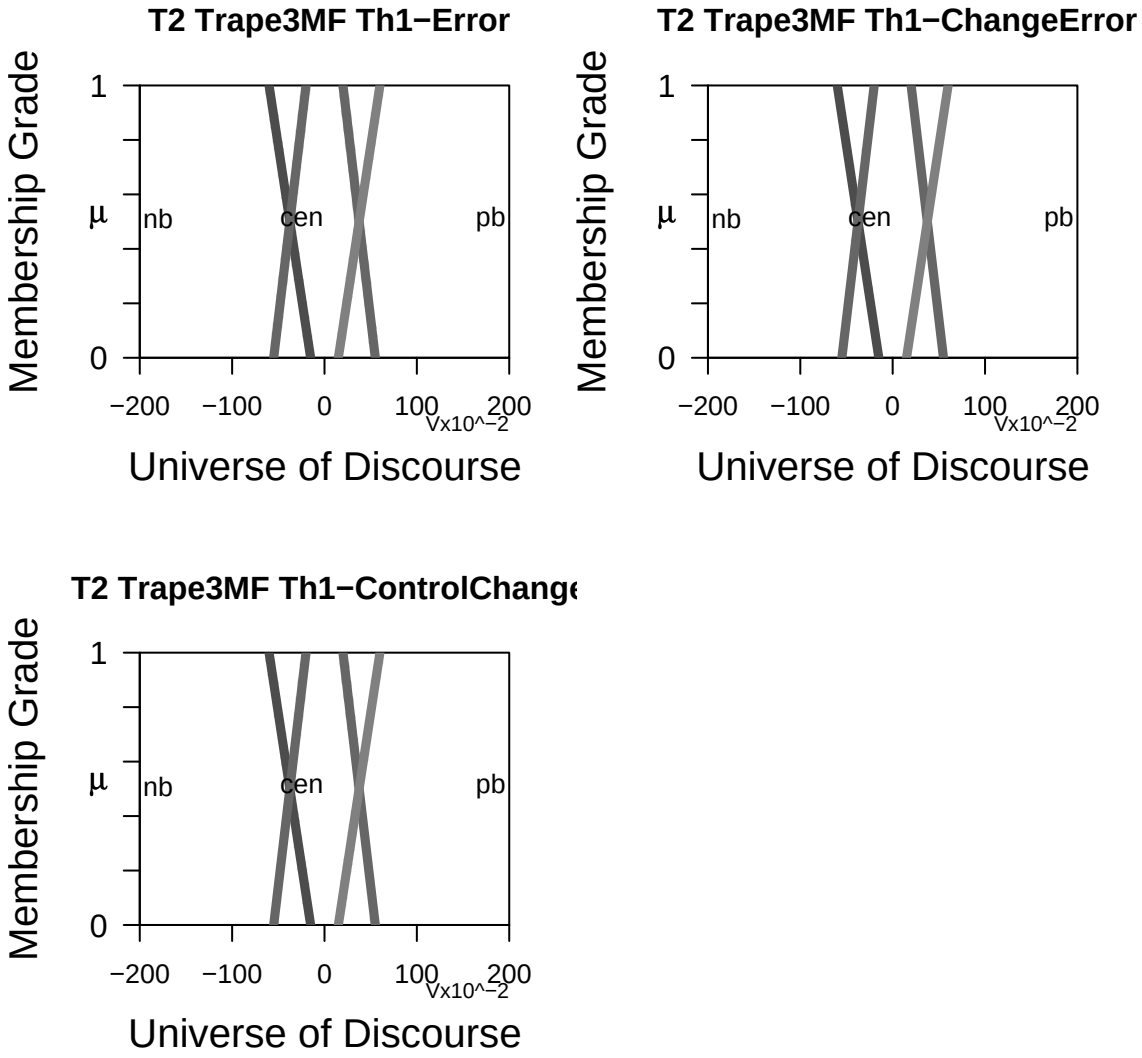


Figure D.4: Type-2 Three Term Trapezoidal with Threshold - 1

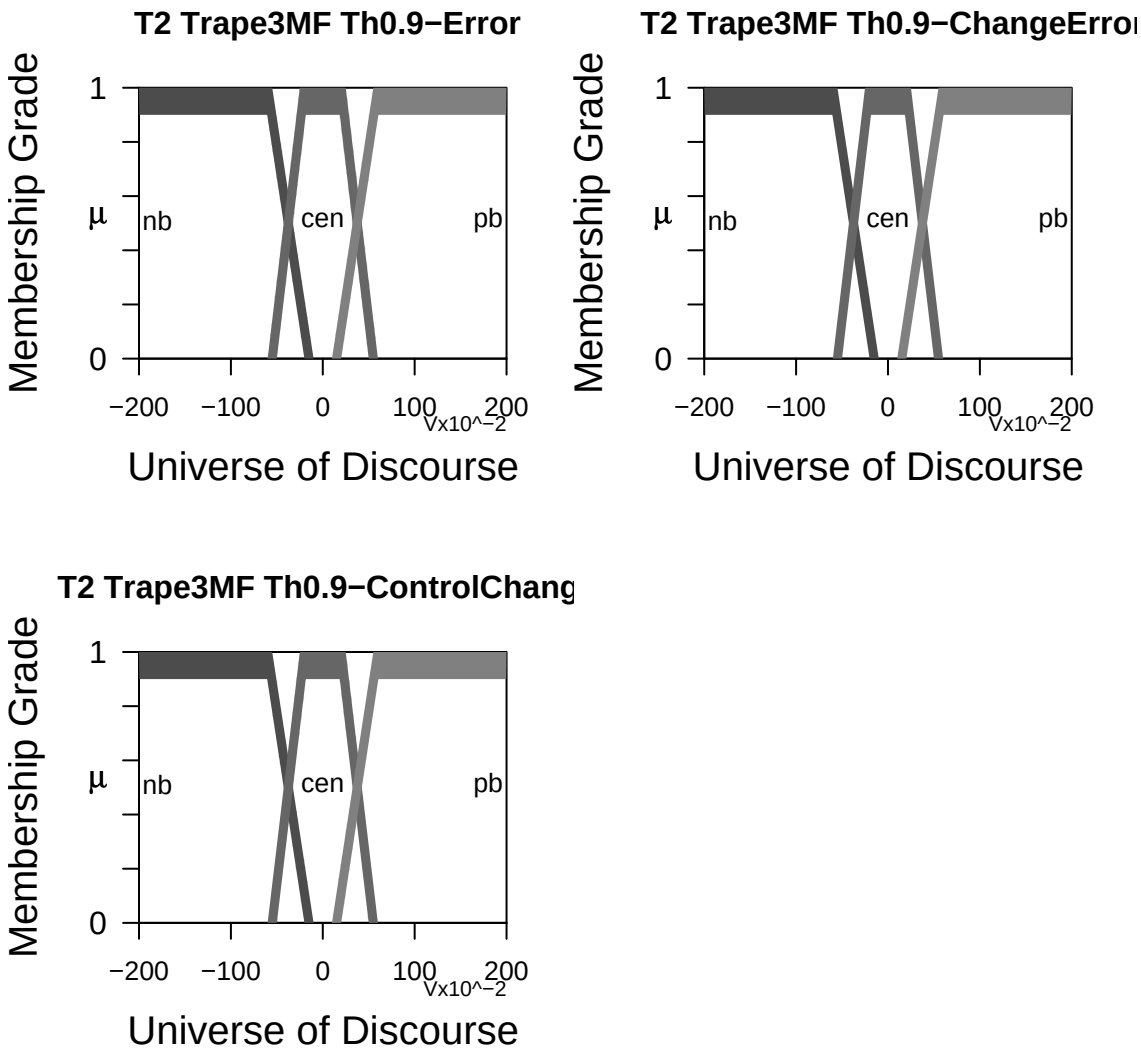


Figure D.5: Type-2 Three Term Trapezoidal with Threshold - 0.9

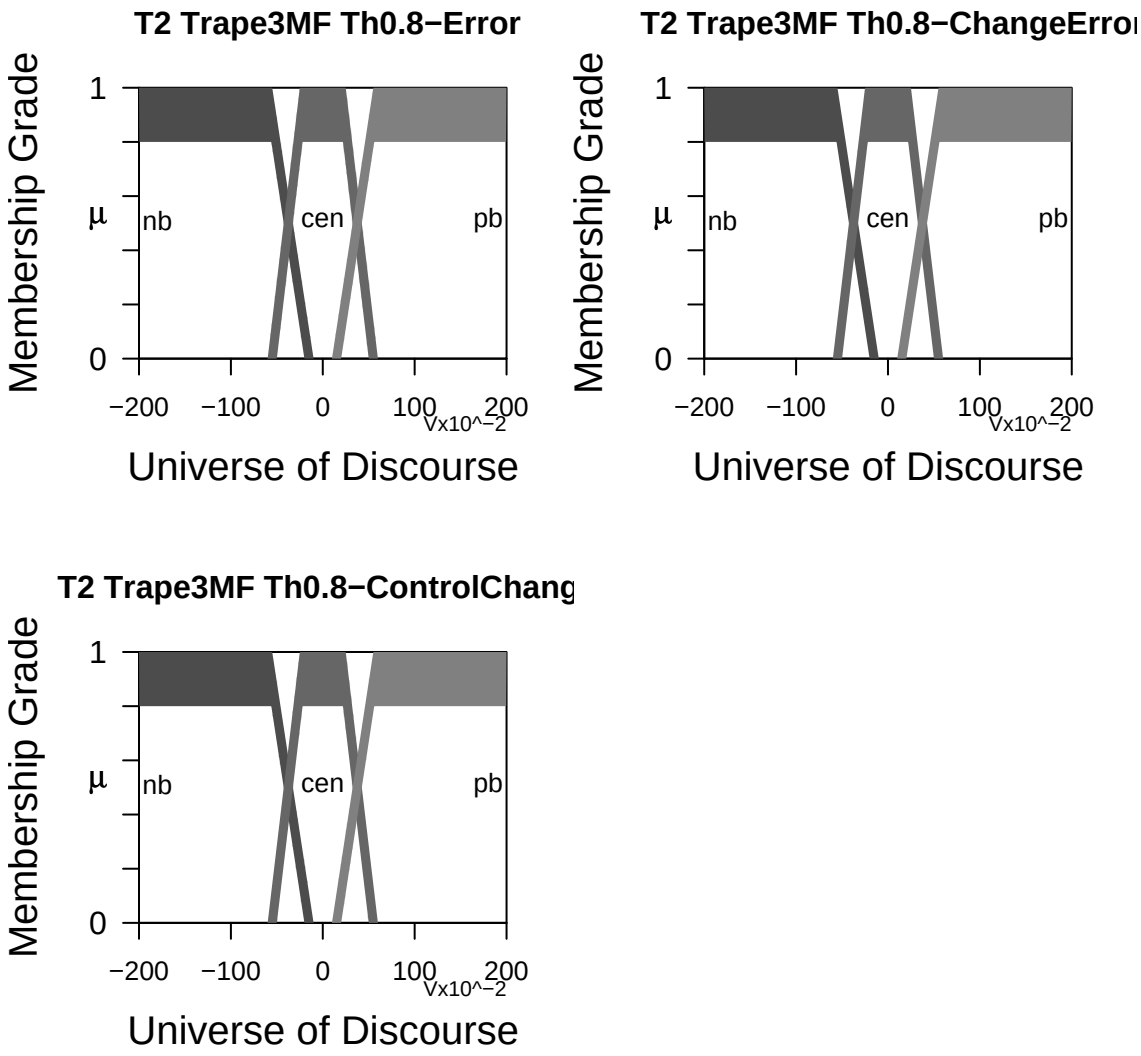


Figure D.6: Type-2 Three Term Trapezoidal with Threshold - 0.8

Variables	nb	cen	pb
Change	-201	-60	10
Error	-201	-5	55
Outer	-55	5	201
	-10	60	201
Change	-201	-50	20
Error	-201	0	65
Inner	-65	0	201
	-20	50	201
Error	-201	-60	10
Outer	-201	-5	55
	-55	5	201
	-10	60	201
Error	-201	-50	20
Inner	-201	0	65
	-65	0	201
	-20	50	201
Control	-201	-60	10
Change	-201	-5	55
Outer	-55	5	201
	-10	60	201
Control	-201	-50	20
Change	-201	0	65
Inner	-65	0	201
	-20	50	201

Table D.3: Type-2 Three Term Trapezoidal Triangular 300 MF Parameters

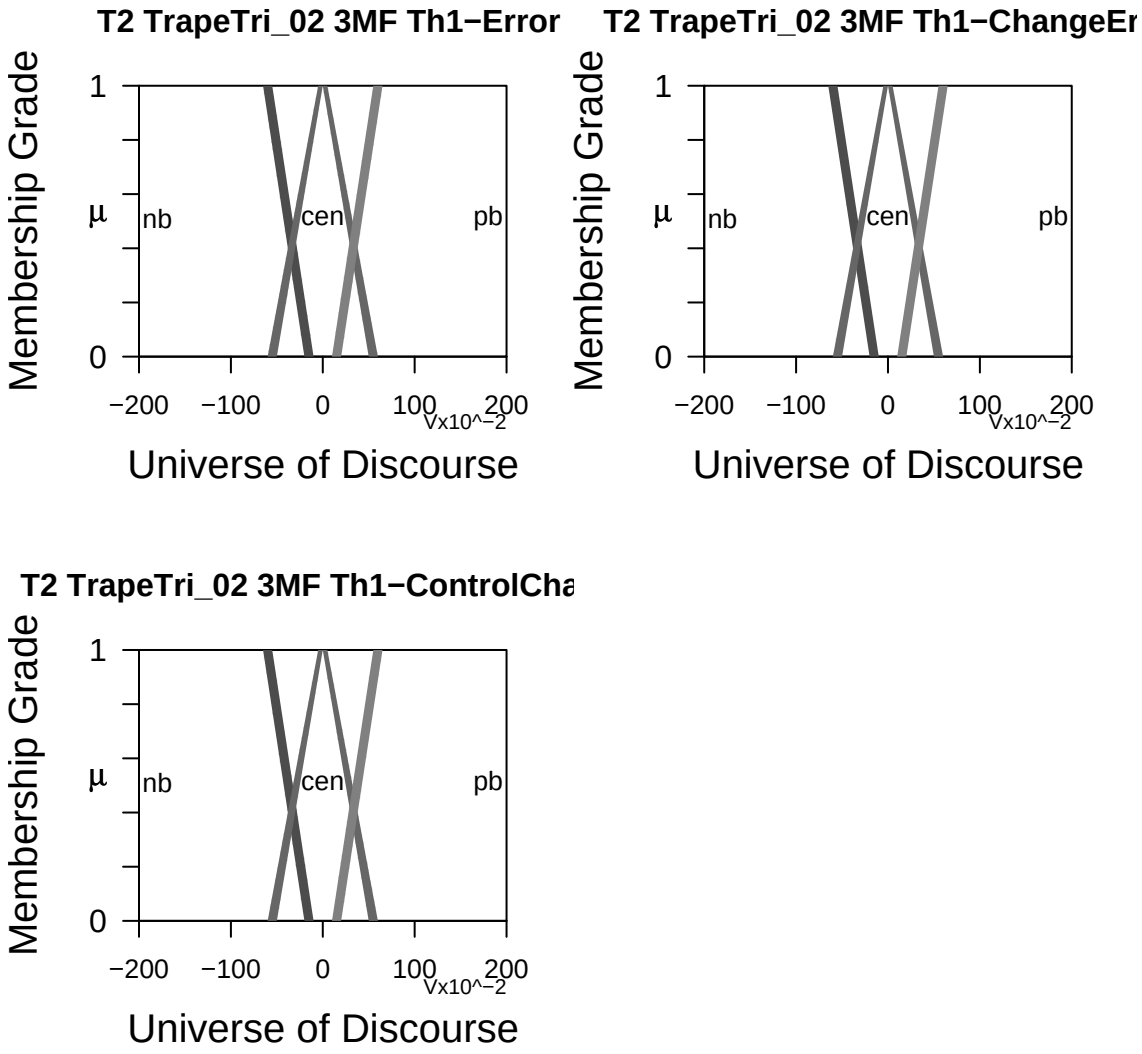


Figure D.7: Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 1

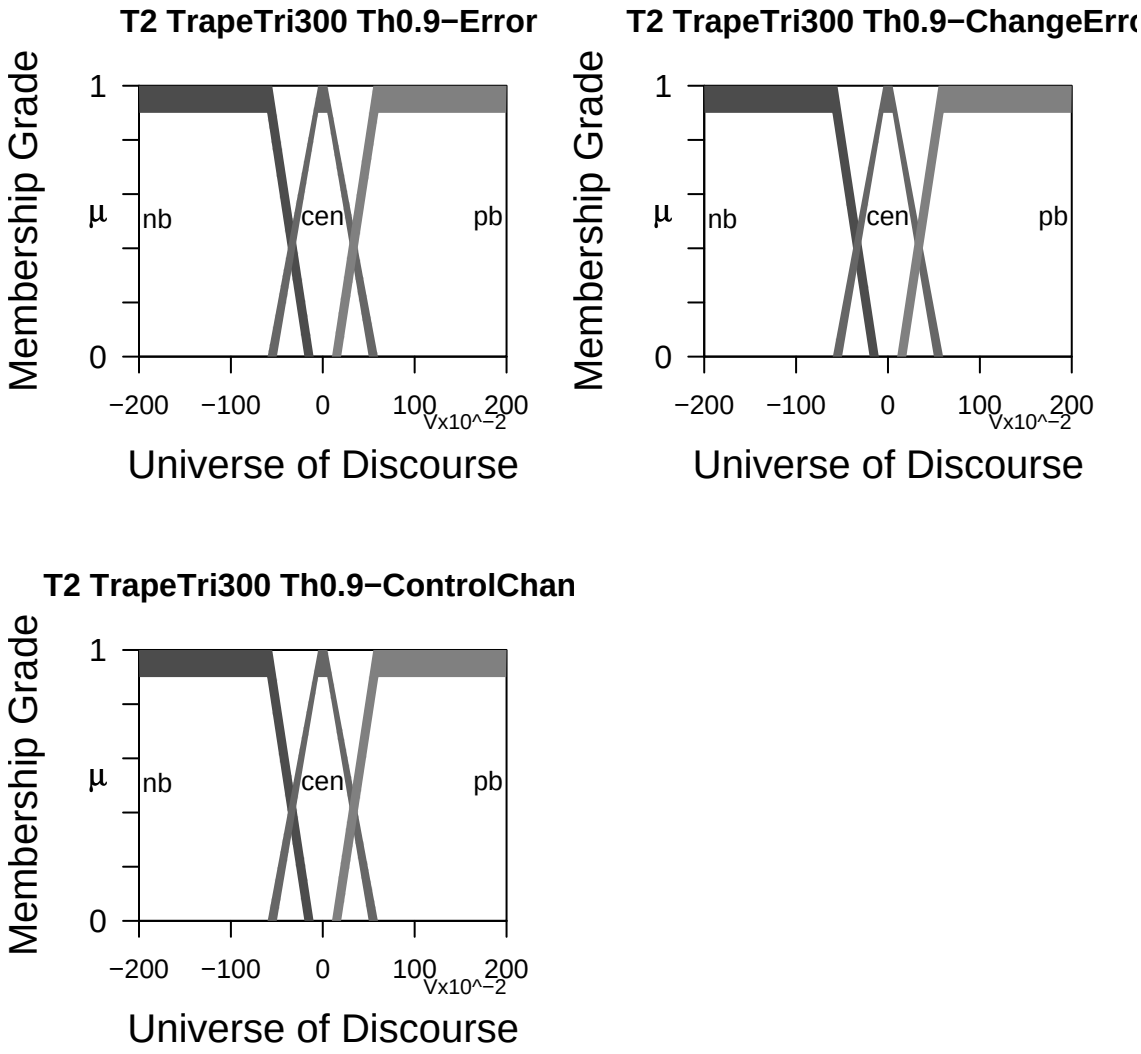


Figure D.8: Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 0.9

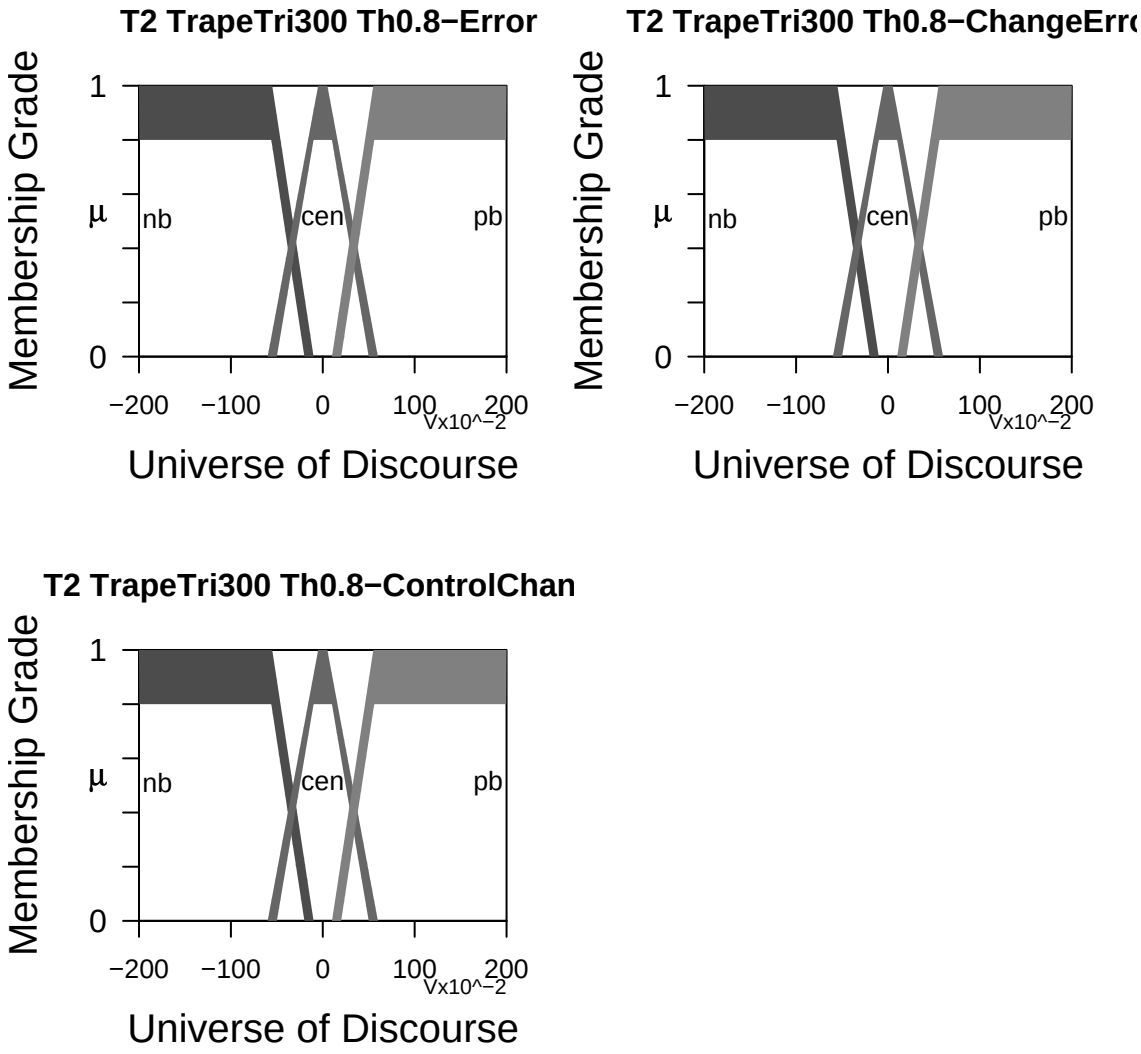


Figure D.9: Type-2 Three Term Trapezoidal Triangular 300 with Threshold - 0.8

Variables	nb	cen	pb
Change	-201	-60	10
Error	-201	-5	55
Outer	-55	5	201
	-10	60	201
Change	-201	-50	20
Error	-201	5	65
Inner	-65	-5	201
	-20	50	201
Error	-201	-60	10
Outer	-201	-5	55
	-55	5	201
	-10	60	201
Error	-201	-50	20
Inner	-201	5	65
	-65	-5	201
	-20	50	201
Control	-201	-60	10
Change	-201	-5	55
Outer	-55	5	201
	-10	60	201
Control	-201	-50	20
Change	-201	5	65
Inner	-65	-5	201
	-20	50	201

Table D.4: Type-2 Three Term Trapezoidal Triangular 305 MF Parameters

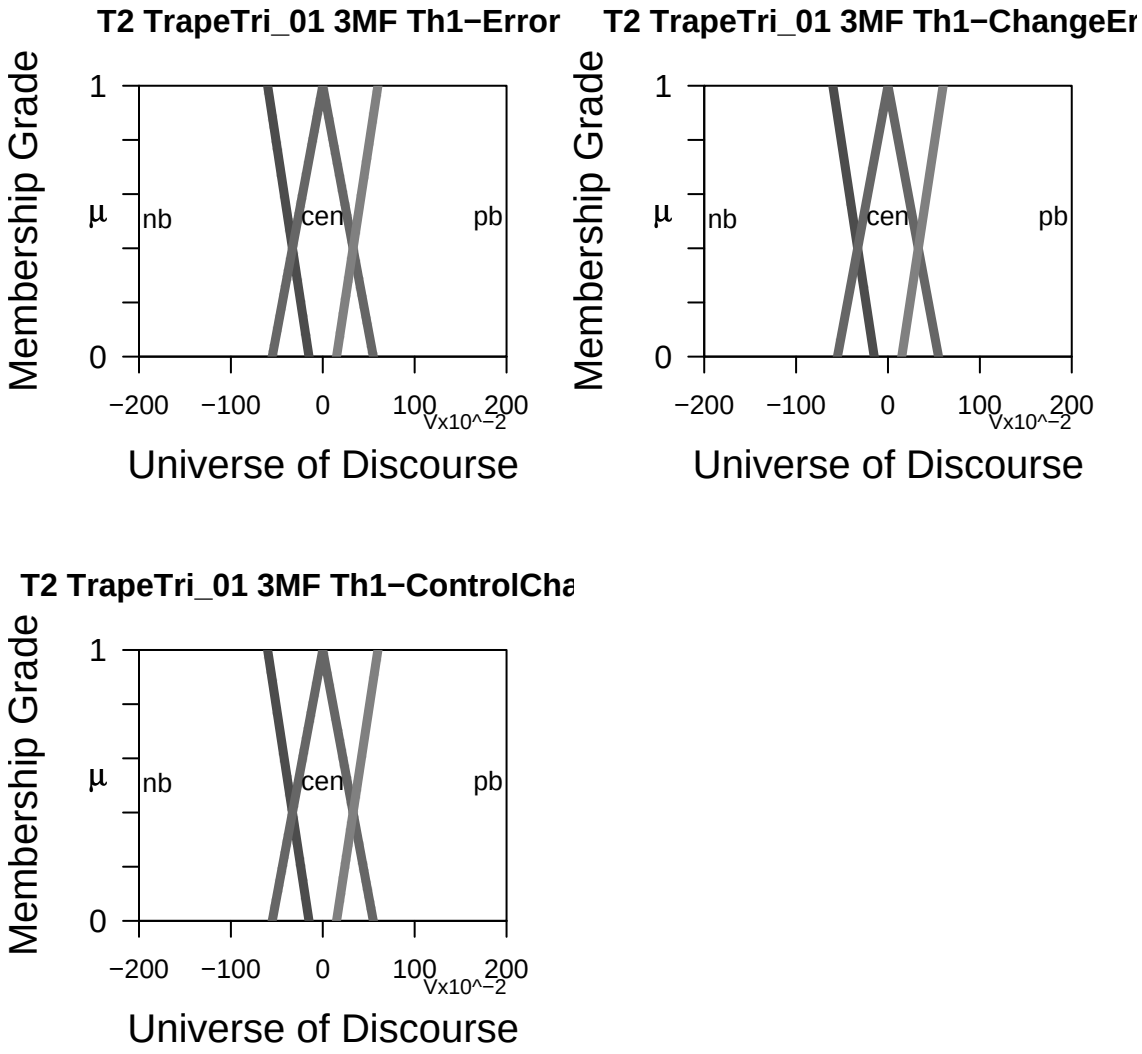


Figure D.10: Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 1

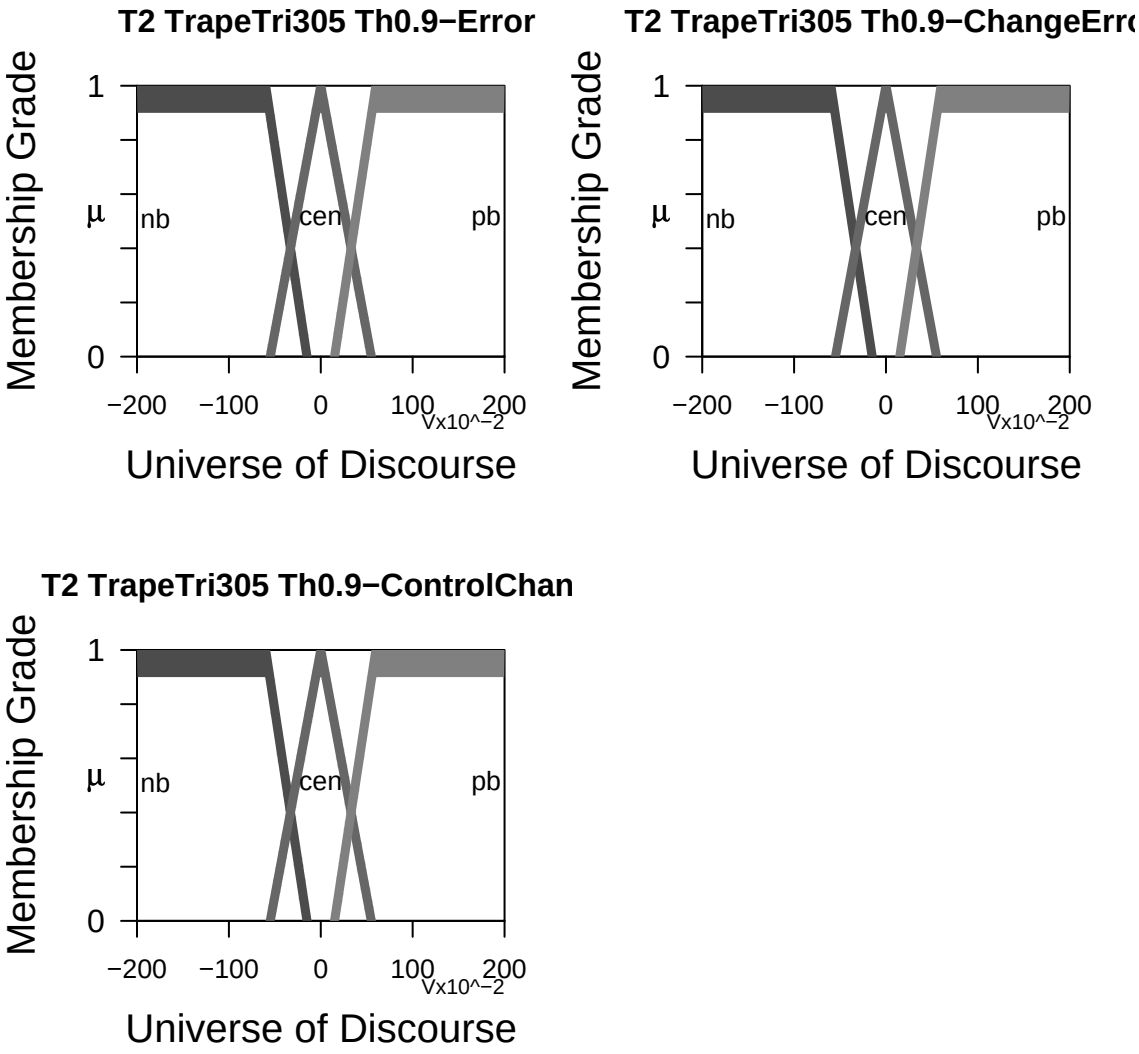


Figure D.11: Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 0.9

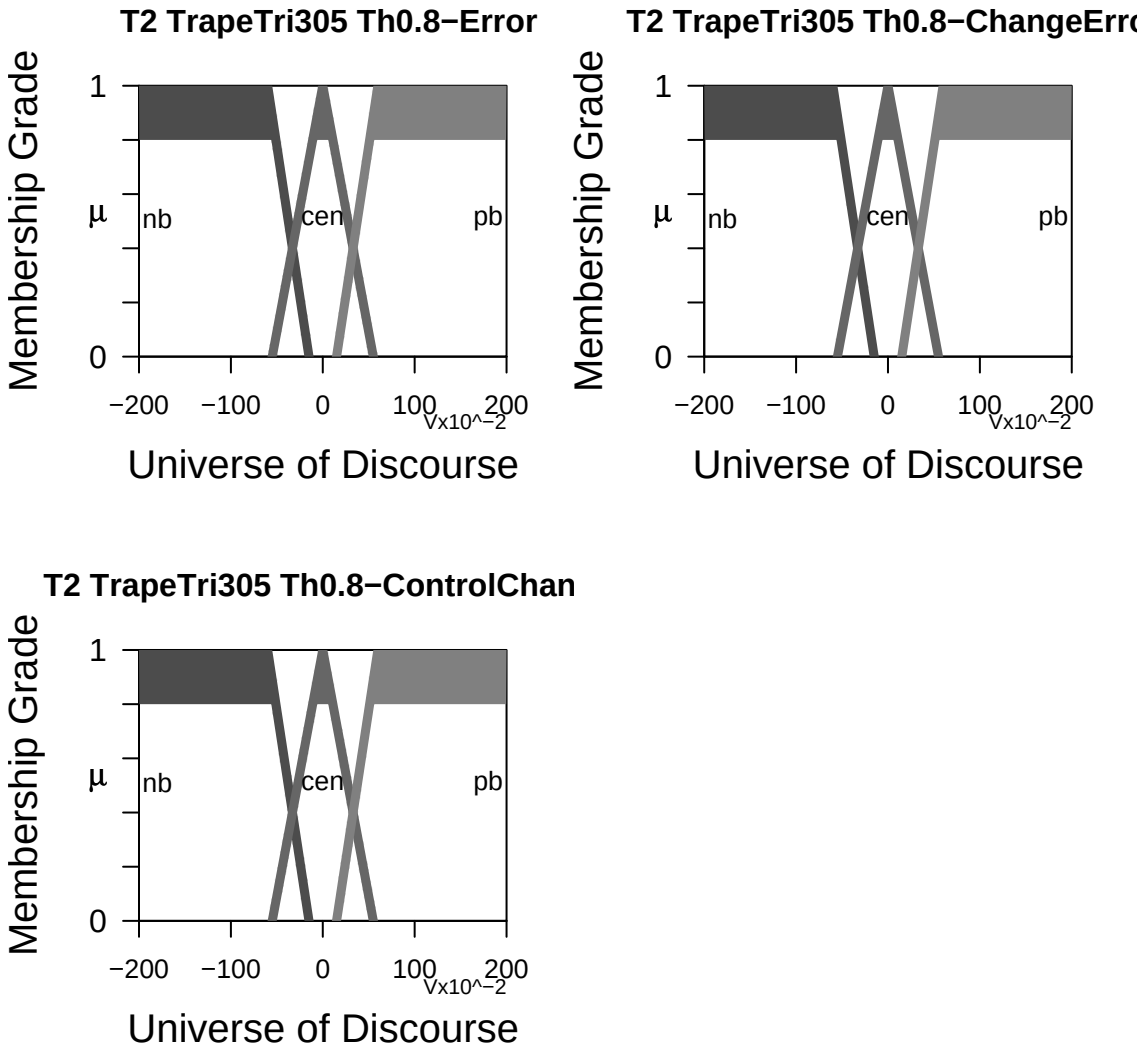
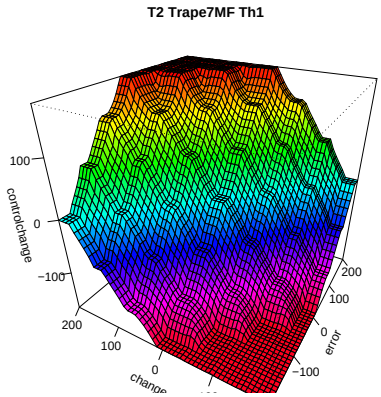
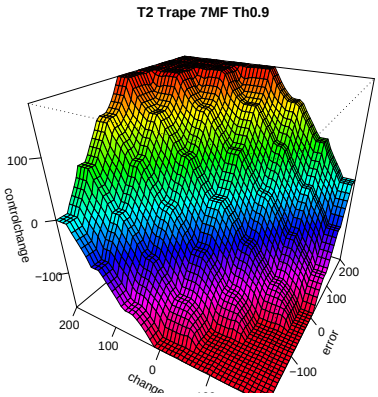


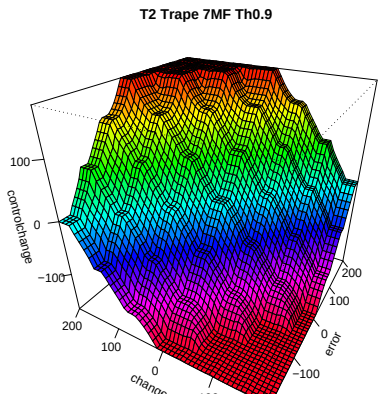
Figure D.12: Type-2 Three Term Trapezoidal Triangular 305 with Threshold - 0.8



(a) Threshold 1.0

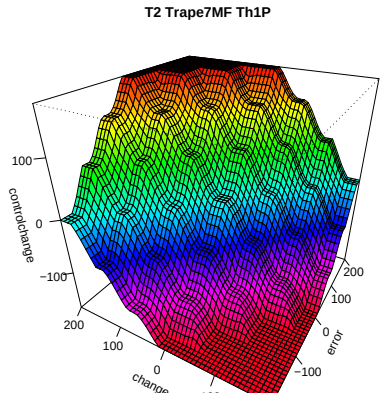


(b) Threshold 0.9

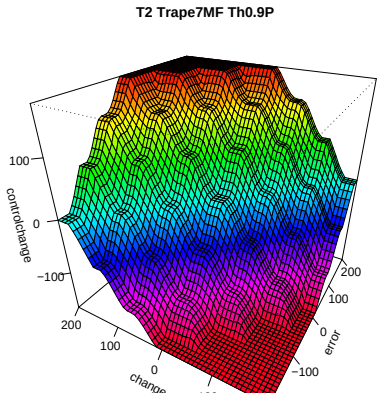


(c) Threshold 0.8

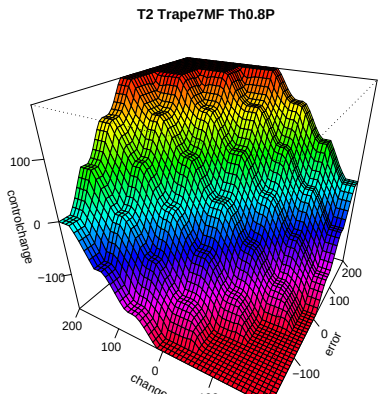
Figure D.13: Type-2 Seven Term Trapezoidal Min Surfaces



(a) Threshold 1.0

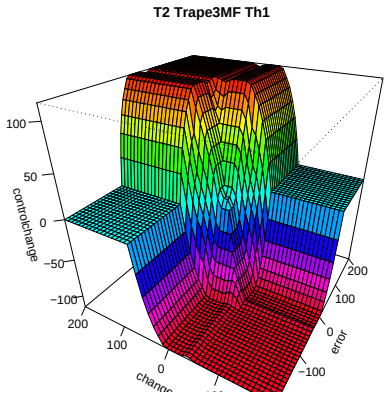


(b) Threshold 0.9

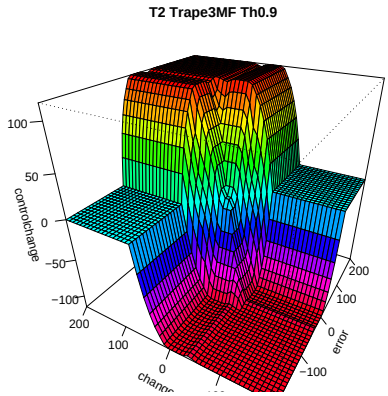


(c) Threshold 0.8

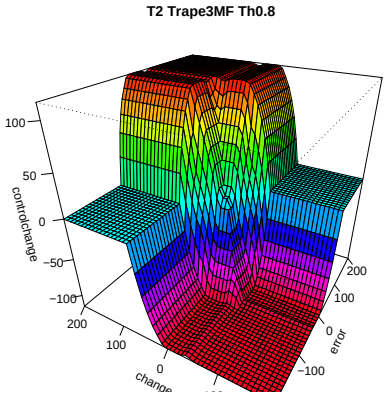
Figure D.14: Type-2 Seven Term Trapezoidal Product Surfaces



(a) Threshold 1.0



(b) Threshold 0.9

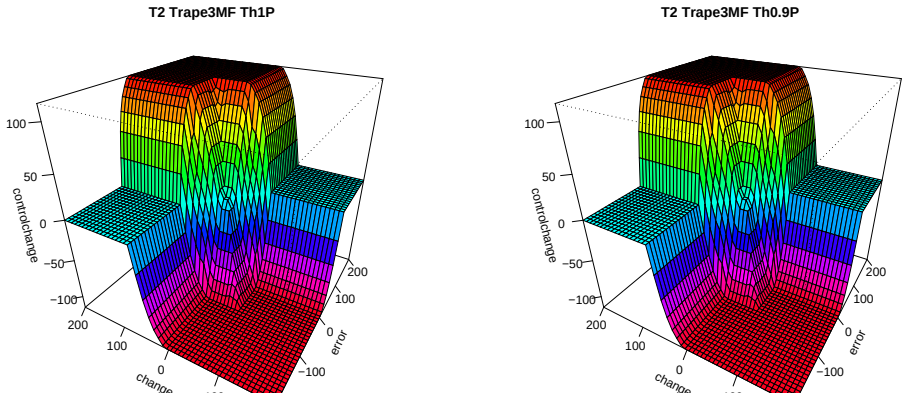


(c) Threshold 0.8

Figure D.15: Type-2 Three Term Trapezoidal Min Surfaces

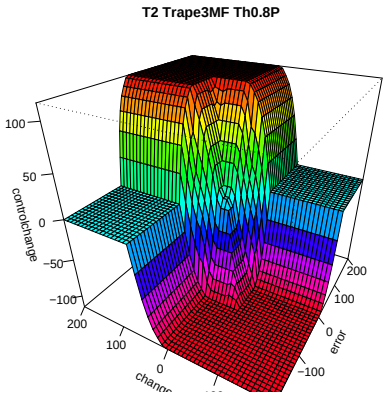
Input StepNum	StepRamp Setpoints
0	-200
90	-160
181	-120
272	-80
363	-40
454	0.001
545	40
636	80
727	120
818	160
909	200

Table D.5: Simulator Input



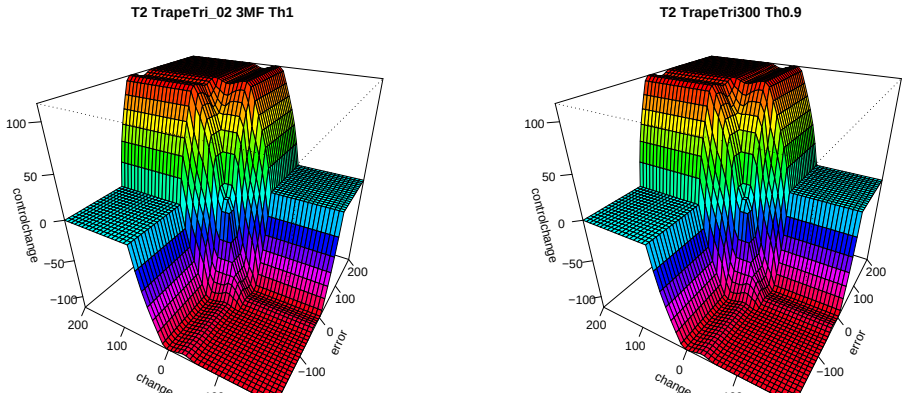
(a) Threshold 1.0

(b) Threshold 0.9



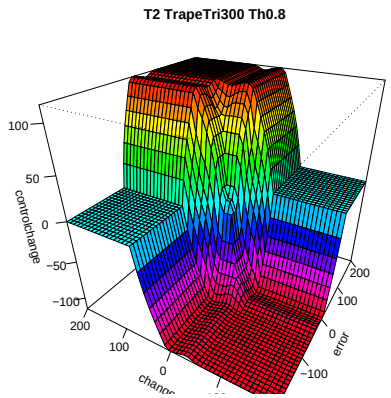
(c) Threshold 0.8

Figure D.16: Type-2 Three Term Trapezoidal Product Surfaces



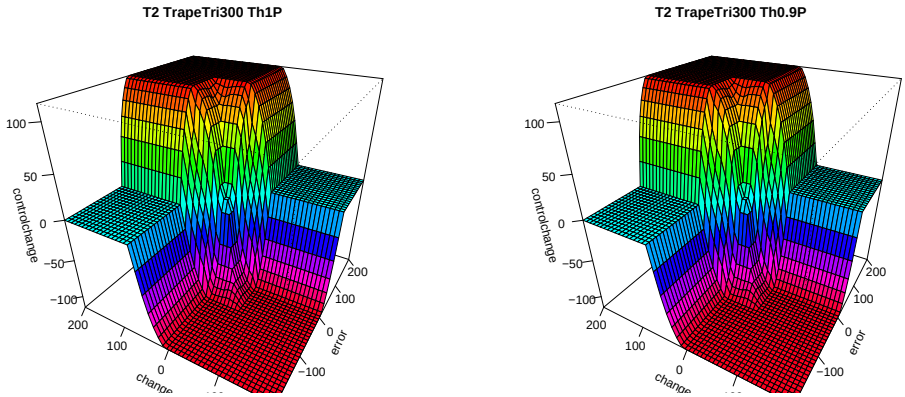
(a) Threshold 1.0

(b) Threshold 0.9



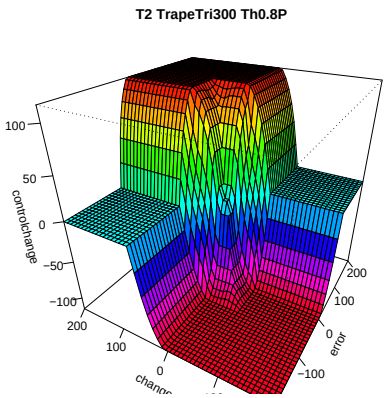
(c) Threshold 0.8

Figure D.17: Type-2 Three Term Trapezoidal Triangular 300 Min Surfaces



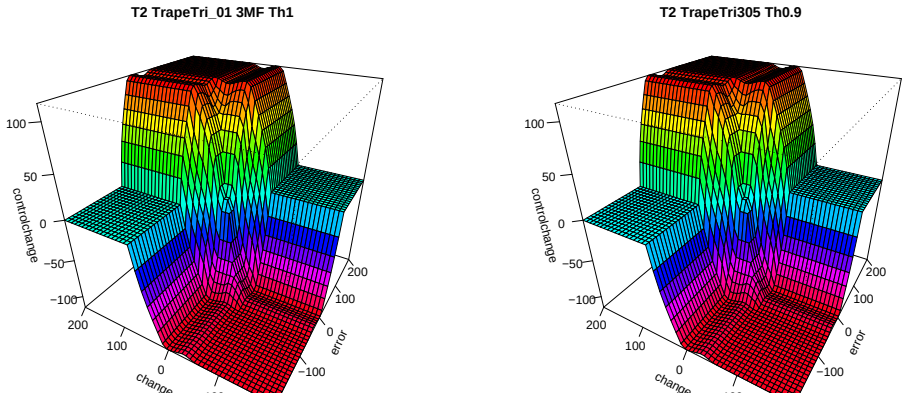
(a) Threshold 1.0

(b) Threshold 0.9

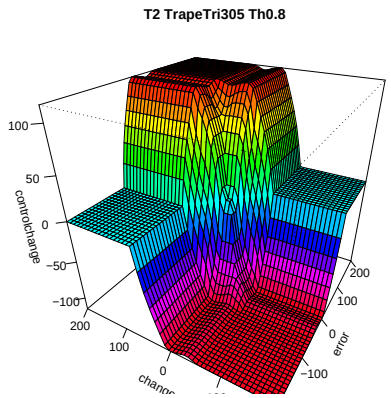


(c) Threshold 0.8

Figure D.18: Type-2 Three Term Trapezoidal Triangular 300 Product Surfaces

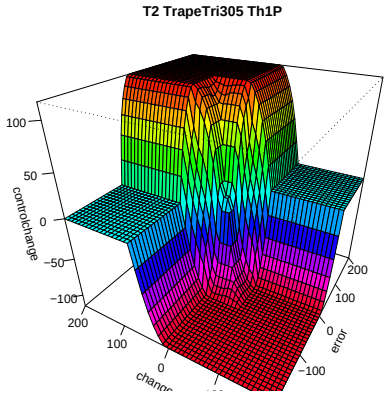


(a) Threshold 1.0 (b) Threshold 0.9

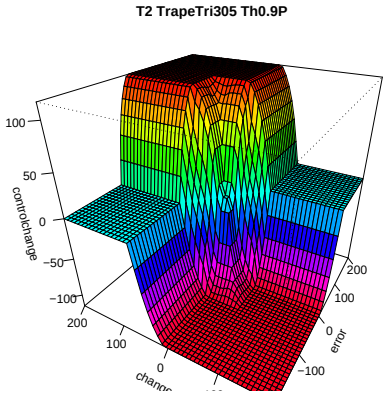


(c) Threshold 0.8

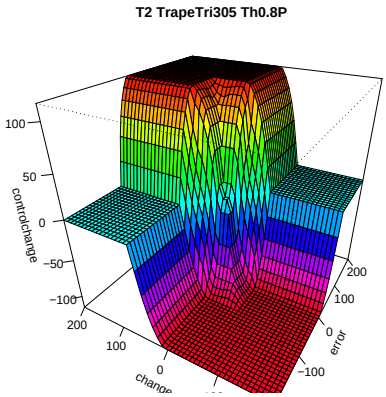
Figure D.19: Type-2 Three Term Trapezoidal Triangular 305 Min Surfaces



(a) Threshold 1.0



(b) Threshold 0.9



(c) Threshold 0.8

Figure D.20: Type-2 Three Term Trapezoidal Triangular 305 Product Surfaces

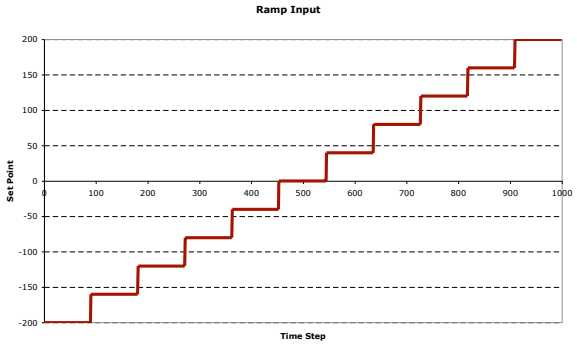


Figure D.21: Simulator Input

Number of Steps Input MFs	1000 Step Ramp Noise%	IMP - Min Without Inertia	Noise With Inertia
T1trap3	1	101	102
T1trap3	16	103	104
T1tri3	1	105	106
T1tri3	16	107	108
T1trap7	1	109	110
T1trap7	16	111	112
T2trap3	1	113	114
T2trap3	16	115	116
T2tri305	1	117	118
T2tri305	16	119	120
T2tri300	1	121	122
T2tri300	16	123	124
T2trap7	1	125	126
T2trap7	16	127	128

Table D.6: MF Run Numbers for Type-1 and Type-2 FLCs with Noise

Number of Steps Input MFs	1000 Step Ramp Noise%	IMP - Min Threshold Level	Without Inertia	With Inertia
T2trap3	0	0.9	201	251
T2trap3	1	0.9	202	-
T2trap3	16	0.9	203	-
T2trap3	0	0.8	204	252
T2trap3	1	0.8	205	-
T2tri300	16	0.8	206	-
T2tri300	0	0.9	207	253
T2tri300	1	0.9	208	-
T2tri300	16	0.9	209	-
T2tri300	0	0.8	210	254
T2tri300	1	0.8	211	-
T2tri300	16	0.8	212	-
T2trap7	0	0.9	213	255
T2trap7	1	0.9	214	-
T2trap7	16	0.9	215	-
T2trap7	0	0.8	216	256
T2trap7	1	0.8	217	-
T2trap7	16	0.8	218	-

Table D.7: MF Run Numbers for Type-2 Thresholds

Run	MF	Thresh	RMS_FZC	Rank
8	T2trap3	1	5842.7	3
251	T2trap3	0.9	5841.0	2
252	T2trap3	0.8	5836.5	1
10	T2tri305	1	5617.6	1
12	T2tri300	1	5640.3	2
253	T2tri300	0.9	5640.3	2
254	T2tri300	0.8	5640.3	2
14	T2trap7	1	6821.7	3
255	T2trap7	0.9	6821.6	2
256	T2trap7	0.8	6820.9	1
8	T2trap3	1	5842.7	3
10	T2tri305	1	5617.6	1
12	T2tri300	1	5640.3	2
14	T2trap7	1	6821.7	4
251	T2trap3	0.9	5841.0	2
253	T2tri300	0.9	5640.3	1
255	T2trap7	0.9	6821.6	3
252	T2trap3	0.8	5836.5	2
254	T2tri300	0.8	5640.3	1
256	T2trap7	0.8	6820.9	3

Table D.8: Varying Thresholds with Inertia and No Noise for Type-2 Membership Functions

Run	MF	Thresh	RMS_FZC	Noise%	Rank
7	T2trap3	1	4928.7	0	3
113	T2trap3	1	5092.5	1	5
114	T2trap3	1	20477.8	16	8
201	T2trap3	0.9	4926.6	0	2
202	T2trap3	0.9	5122.2	1	6
203	T2trap3	0.9	21701.9	16	9
204	T2trap3	0.8	4921.9	0	1
205	T2trap3	0.8	5025.3	1	4
206	T2trap3	0.8	20280.3	16	7
9	T2tri305	1	4665.4	0	1
117	T2tri305	1	4734.0	1	5
118	T2tri305	1	19044.7	16	9
11	T2tri300	1	4690.5	0	2
119	T2tri300	1	4835.2	1	8
120	T2tri300	1	22346.1	16	12
207	T2tri300	0.9	4690.5	0	2
208	T2tri300	0.9	4819.7	1	7
209	T2tri300	0.9	19453.5	16	10
210	T2tri300	0.8	4690.5	0	2
211	T2tri300	0.8	4761.6	1	6
212	T2tri300	0.8	20395.3	16	11
13	T2trap7	1	6186.2	0	4
125	T2trap7	1	6382.8	1	6
126	T2trap7	1	20957.7	16	9
213	T2trap7	0.9	6186.1	0	3
214	T2trap7	0.9	6226.5	1	5
215	T2trap7	0.9	20907.9	16	8
216	T2trap7	0.8	6185.1	0	2
217	T2trap7	0.8	6161.2	1	1
218	T2trap7	0.8	20342.1	16	7

Table D.9: Noise within Threshold within MF

Run	MF	Thresh	RMS_FZC	Noise%	Rank
7	T2trap3	1	4928.7	0	7
9	T2tri305	1	4665.4	0	1
11	T2tri300	1	4690.5	0	2
13	T2trap7	1	6186.2	0	10
201	T2trap3	0.9	4926.6	0	6
207	T2tri300	0.9	4690.5	0	2
213	T2trap7	0.9	6186.1	0	9
204	T2trap3	0.8	4921.9	0	5
210	T2tri300	0.8	4690.5	0	2
216	T2trap7	0.8	6185.1	0	8
113	T2trap3	1	5092.5	1	6
117	T2tri305	1	4734.0	1	1
119	T2tri300	1	4835.2	1	4
125	T2trap7	1	6382.8	1	10
202	T2trap3	0.9	5122.3	1	7
208	T2tri300	0.9	4819.7	1	3
214	T2trap7	0.9	6226.5	1	9
205	T2trap3	0.8	5025.3	1	5
211	T2tri300	0.8	4761.6	1	2
217	T2trap7	0.8	6161.2	1	8
114	T2trap3	1	20477.8	16	6
118	T2tri305	1	19044.7	16	1
120	T2tri300	1	22346.1	16	10
126	T2trap7	1	20957.7	16	8
203	T2trap3	0.9	21701.9	16	9
209	T2tri300	0.9	19453.4	16	2
215	T2trap7	0.9	20907.9	16	7
206	T2trap3	0.8	20280.3	16	3
212	T2tri300	0.8	20395.3	16	5
218	T2trap7	0.8	20342.1	16	4

Table D.10: MF within Threshold within Noise

Run	MF	Thresh	RMS_FZC	Noise%
7	T2trap3	1	4928.7	0
113	T2trap3	1	5092.5	1
114	T2trap3	1	20477.8	16
11	T2tri300	1	4690.5	0
119	T2tri300	1	4835.2	1
120	T2tri300	1	22346.1	16
9	T2tri305	1	4665.4	0
117	T2tri305	1	4734.0	1
118	T2tri305	1	19044.7	16
13	T2trap7	1	6186.2	0
125	T2trap7	1	6382.8	1
126	T2trap7	1	20957.7	16
201	T2trap3	0.9	4926.6	0
202	T2trap3	0.9	5122.3	1
203	T2trap3	0.9	21701.9	16
207	T2tri300	0.9	4690.5	0
208	T2tri300	0.9	4819.7	1
209	T2tri300	0.9	19453.4	16
213	T2trap7	0.9	6186.1	0
214	T2trap7	0.9	6226.5	1
215	T2trap7	0.9	20907.9	16
204	T2trap3	0.8	4921.9	0
205	T2trap3	0.8	5025.3	1
206	T2trap3	0.8	20280.3	16
210	T2tri300	0.8	4690.5	0
211	T2tri300	0.8	4761.6	1
212	T2tri300	0.8	20395.3	16
216	T2trap7	0.8	6185.1	0
217	T2trap7	0.8	6161.2	1
218	T2trap7	0.8	20342.1	16

Table D.11: Noise within MF within Threshold

Run	MF	Thresh	RMS_FZC	Noise%
7	T2trap3	1	4928.7	0
201	T2trap3	0.9	4926.6	0
204	T2trap3	0.8	4921.9	0
9	T2tri305	1	4665.4	0
11	T2tri300	1	4690.5	0
207	T2tri300	0.9	4690.5	0
210	T2tri300	0.8	4690.5	0
13	T2trap7	1	6186.2	0
213	T2trap7	0.9	6186.1	0
216	T2trap7	0.8	6185.1	0
113	T2trap3	1	5092.5	1
202	T2trap3	0.9	5122.3	1
205	T2trap3	0.8	5025.3	1
117	T2tri305	1	4734.0	1
119	T2tri300	1	4835.2	1
208	T2tri300	0.9	4819.7	1
211	T2tri300	0.8	4761.6	1
125	T2trap7	1	6382.8	1
214	T2trap7	0.9	6226.5	1
217	T2trap7	0.8	6161.2	1
114	T2trap3	1	20477.8	16
203	T2trap3	0.9	21701.9	16
206	T2trap3	0.8	20280.3	16
118	T2tri305	1	19044.7	16
120	T2tri300	1	22346.1	16
209	T2tri300	0.9	19453.4	16
212	T2tri300	0.8	20395.3	16
126	T2trap7	1	20957.7	16
215	T2trap7	0.9	20907.9	16
218	T2trap7	0.8	20342.1	16

Table D.12: Threshold within MF within Noise

Run	MF	File	RMS_FZC	Inertia	Noise%	RMS_PID
		Type-1				
1	T1Trap3	09Oct/Test3_1013_164006.txt	4979.322366748708	-	0	1.724041093639547E14
2	T1Trap3	09Oct/Test5_1013_164105.txt	5880.3508018661505	I	0	1.723494885743913 8E14
3	T1Tri3	09Oct/Test1_1013_164447.txt	4705.1812345916 26	-	0	-
4	T1Tri3	09Oct/Test5_1013_164512.txt	5648.446445334611	I	0	1.723494885743913 8E14
5	T1Trap7	09Oct/Test1_1013_164240.txt	6254.1088108142 68	-	0	-
6	T1Trap7	09Oct/Test5_1013_164322.txt	6873.4647842108 48	I	0	1.723494885743913 8E14
		Type-2				
7	T2Trap3	09Oct/Test6_1013_164730.txt	4928.709049179 16	-	0	-
8	T2Trap3	09Oct/Test7_1013_164811.txt	5842.6826715207 83	I	0	-
9	T2Tri305	09Oct/Test6_1013_165232.txt(5,-5)	4665.3671742289 33	-	0	-
10	T2Tri305	09Oct/Test7_1013_165316.txt(5,-5)	5617.6179408296 41	I	0	-
11	T2Tri300	09Oct/Test6_1014_101051.txt(0, 0)	4690.505420476 33	-	0	-
12	T2Tri300	09Oct/Test7_1014_100058.txt(0, 0)	5640.332467793 9745	I	0	-
13	T2Trap7	09Oct/Test6_1013_174406.txt	6186.2260625147 32	-	0	-
14	T2Trap7	09Oct/Test7_1013_175146.txt	6821.6868626785 6	I	0	-

Table D.13: Type-1 FLCs against Type-2 FLCs (Full)

Run	MF	File	RMS_FZC	Inertia	Noise%	RMS_PID
	Without Inertia					
1	T1Trap3	09Oct/Test3_1013_164006.txt	4979.322366748708	-	0	1.7240410936395 47E14
7	T2Trap3	09Oct/Test6_1013_164730.txt	4928.70904917916	-	0	-
3	T1Tri3	09Oct/Test1_1013_164447.txt	4705.181234591626	-	0	-
9	T2Tri305	09Oct/Test6_1013_165232.txt(5,-5)	4665.367174228933	-	0	-
11	T2Tri300	09Oct/Test6_1014_101051.txt(0, 0)	4690.50542047633	-	0	-
5	T1Trap7	09Oct/Test1_1013_164240.txt	6254.108810814268	-	0	-
13	T2Trap7	09Oct/Test6_1013_174406.txt	6186.226062514732	-	0	-
	With Inertia					
2	T1Tri3	09Oct/Test5_1013_164105.txt	5880.3508018661505	I	0	1.7234948857439138E14
8	T2Trap3	09Oct/Test7_1013_164811.txt	5842.682671520783	I	0	-
4	T1Tri3	09Oct/Test5_1013_164512.txt	5648.446445334611	I	0	1.7234948857439138E14
10	T2Tri305	09/Oct/Test7_1013_165316.txt(5,-5)	5617.617940829641	I	0	-
12	T2Tri300	09Oct/Test7_1014_100058.txt(0, 0)	5640.3324677939745	I	0	-
6	T1Trap7	09Oct/Test5_1013_164322.txt	6873.464784210848	I	0	1.7234948857439138E14
14	T2Trap7	09Oct/Test7_1013_175146.txt	6821.68686267856	I	0	-

Table D.14: Type1 against Type-2 within Inertia (Full)

Run	MF	File	RMS_FZC	Inertia	Noise%	RMS_PID
			Type-1			
101	T1trap3	09Oct/Test3_1013_165738.txt	5212.598759742	-	1	2.1138177227409606E14
102	T1trap3	09Oct/Test3_1013_170354.txt	20659.444200403806	-	16	1.7727146794917115E15
103	T1trap3	09Oct/Test5_1013_170250.txt	5960.025499043392	I	1	1.985443712209436E14
104	T1trap3	09Oct/Test5_1013_170423.txt	18718.486127693217	I	16	1.6267870384682035E15
105	T1tri3	09Oct/Test1_1013_171153.txt	4890.415783245636	-	1	-
106	T1tri3	09Oct/Test1_1013_171005.txt	21464.7080099066	-	16	-
107	T1tri3	09Oct/Test5_1013_171219.txt	5925.517432351581	I	1	2.0447995152081725E14
108	T1tri3	09Oct/Test5_1013_171043.txt	19964.148352048203	I	16	1.7737181876650958E15
109	T1trap7	09Oct/Test1_1013_170610.txt	6368.685402443599	-	1	-
110	T1trap7	09Oct/Test1_1013_170752.txt	17955.948955554606	-	16	-
111	T1trap7	09Oct/Test5_1013_170651.txt	6993.056175876338	I	1	2.2171581772568944E14
112	T1trap7	09Oct/Test5_1013_170840.txt	19066.41118599032	I	16	1.903794471583325E15
		Type-2				
113	T2trap3	09Oct/Test6_1013_171440.txt	5092.52816718091	-	1	-
114	T2trap3	09Oct/Test6_1013_171700.txt	20477.816256161575	-	16	-
115	T2trap3	09Oct/Test7_1013_171522.txt	5841.643984365342	I	1	-
116	T2trap3	09Oct/Test7_1013_171741.txt	19482.201390283008	I	16	-
117	T2tri305	09Oct/Test6_1013_172441.txt	4733.970234941016	-	1	-(5,-5)
118	T2tri305	09Oct/Test6_1013_172623.txt	19044.718434810937	-	16	-(5,-5)
119	T2tri300	09Oct/Test6_1014_101405.txt	4835.234635957925	-	1	-(0,0)
120	T2tri300	09Oct/Test6_1014_101456.txt	22346.061184401682	-	16	-(0,0)
121	T2tri305	09Oct/Test7_1013_172526.txt	5882.35502448692	I	1	-(5,-5)
122	T2tri305	09Oct/Test7_1013_172705.txt	20045.78854427974	I	16	-(5,-5)
123	T2tri300	09Oct/Test7_1014_101557.txt	5753.713280872572	I	1	-(0,0)
124	T2tri300	09Oct/Test7_1014_101738.txt	20903.88351814896	I	16	-(0,0)
125	T2trap7	09Oct/Test6_1013_172140.txt	6382.821165082555	-	1	-
126	T2trap7	09Oct/Test6_1013_171912.txt	20957.721521671505	-	16	-
127	T2trap7	09Oct/Test7_1013_172243.txt	7022.321511676284	I	1	-
128	T2trap7	09Oct/Test7_1013_172017.txt	22064.612240888895	I	16	-

Table D.15: Type-1 against Type-2 with Noise (Full)

Run	MF	File	Thresh	RMS_FZC	Inertia	Noise%
		Threshold within		Membership Function		
8	T2Trap3	09Oct/Test7_1013_164811.txt	1	5842.682671520783	I	0
251	T2trap3	09Oct/Test7_1014_104656.txt	0.9	5840.984309612984	I	0
252	T2trap3	09Oct/Test7_1014_105826.txt	0.8	5836.523615884737	I	0
10	T2Tri305	09Oct/Test7_1013_165316.txt(5,-5)	1	5617.617940829641	I	0
12	T2Tri300	09Oct/Test7_1014_100058.txt(0, 0)	1	5640.3324677939745	I	0
253	T2tri3	09Oct/Test7_1014_115324.txt	0.9	5640.33246598459	I	0
254	T2tri3	09Oct/Test7_1014_120300.txt	0.8	5640.332456747383	I	0
14	T2Trap7	09Oct/Test7_1013_175146.txt	1	6821.68686267856	I	0
255	T2trap7	09Oct/Test7_1014_111606.txt	0.9	6821.597557815642	I	0
256	T2trap7	09Oct/Test7_1014_185550.txt	0.8	6820.912391494358	I	0
		Membership Function		within Threshold		
8	T2Trap3	09Oct/Test7_1013_164811.txt	1	5842.682671520783	I	0
10	T2Tri305	09Oct/Test7_1013_165316.txt	1	5617.617940829641	I	0
12	T2Tri300	09Oct/Test7_1014_100058.txt	1	5640.3324677939745	I	0
14	T2Trap7	09Oct/Test7_1013_175146.txt	1	6821.68686267856	I	0
251	T2trap3	09Oct/Test7_1014_104656.txt	0.9	5840.984309612984	I	0
253	T2tri3	09Oct/Test7_1014_115324.txt	0.9	5640.33246598459	I	0
255	T2trap7	09Oct/Test7_1014_111606.txt	0.9	6821.597557815642	I	0
252	T2trap3	09Oct/Test7_1014_105826.txt	0.8	5836.523615884737	I	0
254	T2tri3	09Oct/Test7_1014_120300.txt	0.8	5640.332456747383	I	0
256	T2trap7	09Oct/Test7_1014_185550.txt	0.8	6820.912391494358	I	0

Table D.16: Varying Thresholds with Inertia for Type-2 Membership Functions (Full)

Run	MF	File	Thresh	RMS_FZC	Inertia	Noise%
		Noise within Threshold		within MF		
7	T2Trap3	09Oct/Test6_1013_164730.txt	1	4928.70904917916	-	0
113	T2trap3	09Oct/Test6_1013_171440.txt	1	5092.52816718091	-	1
114	T2trap3	09Oct/Test6_1013_171700.txt	1	20477.816256161575	-	16
201	T2trap3	09Oct/Test6_1014_104241.txt	0.9	4926.644407181475	-	0
202	T2trap3	09Oct/Test6_1014_104143.txt	0.9	5122.283696273073	-	1
203	T2trap3	09Oct/Test6_1014_103925.txt	0.9	21701.90929193883	-	16
204	T2trap3	09Oct/Test6_1014_105352.txt	0.8	4921.912429626856	-	0
205	T2trap3	09Oct/Test6_1014_105506.txt	0.8	5025.296683423538 5	-	1
206	T2trap3	09Oct/Test6_1014_105556.txt	0.8	20280.314421558473	-	16
9	T2Tri305	09Oct/Test6_1013_165232.txt	1	4665.367174228933	-	0
11	T2Tri300	09Oct/Test6_1014_101051.txt	1	4690.50542047633	-	0
117	T2tri305	09Oct/Test6_1013_172441.txt	1	4733.970234941016	-	1
119	T2tri300	09Oct/Test6_1014_101405.txt	1	4835.234635957925	-	1
118	T2tri305	09Oct/Test6_1013_172623.txt	1	19044.718434810937	-	16
120	T2tri300	09Oct/Test6_1014_101456.txt	1	22346.061184401682	-	16
207	T2tri3	09Oct/Test6_1014_115116.txt	0.9	4690.505418331806	-	0
208	T2tri3	09Oct/Test6_1014_115031.txt	0.9	4819.688764986232	-	1
209	T2tri3	09Oct/Test6_1014_114936.txt	0.9	19453.377236646036	-	16
210	T2tri3	09Oct/Test6_1014_120053.txt	0.8	4690.505407383603	-	0
211	T2tri3	09Oct/Test6_1014_115834.txt	0.8	4761.575267068617	-	1
212	T2tri3	09Oct/Test6_1014_184709.txt	0.8	20395.293012397757	-	16
13	T2Trap7	09Oct/Test6_1013_174406.txt	1	6186.226062514732	-	0
125	T2trap7	09Oct/Test6_1013_172140.txt	1	6382.821165082555	-	1
126	T2trap7	09Oct/Test6_1013_171912.txt	1	20957.721521671505	-	16
213	T2trap7	09Oct/Test6_1014_110923.txt	0.9	6186.08275049523 85	-	0
214	T2trap7	09Oct/Test6_1014_111056.txt	0.9	6226.498709979921	-	1
215	T2trap7	09Oct/Test6_1014_111157.txt	0.9	20907.87620110287	-	16
216	T2trap7	09Oct/Test6_1014_185941.txt	0.8	6185.078115404573 5	-	0
217	T2trap7	09Oct/Test6_1014_190126.txt	0.8	6161.187450103015	-	1
218	T2trap7	09Oct/Test6_1014_190307.txt	0.8	20342.09660575451	-	16

Table D.17: Noise within Threshold within MF (Full)

Run	MF	File	Thresh	RMS_FZC	Inertia	Noise%
7	T2Trap3	09Oct/Test6_1013_164730.txt	1	4928.70904917916	-	0
113	T2trap3	09Oct/Test6_1013_171440.txt	1	5092.52816718091	-	1
114	T2trap3	09Oct/Test6_1013_171700.txt	1	20477.816256161575	-	16
9	T2Tri305	09Oct/Test6_1013_165232.txt	1	4665.367174228933	-	0
11	T2Tri300	09Oct/Test6_1014_101051.txt	1	4690.50542047633	-	0
117	T2tri305	09Oct/Test6_1013_172441.txt	1	4733.970234941016	-	1
119	T2tri300	09Oct/Test6_1014_101405.txt	1	4835.234635957925	-	1
118	T2tri305	09Oct/Test6_1013_172623.txt	1	19044.718434810937	-	16
120	T2tri300	09Oct/Test6_1014_101456.txt	1	22346.061184401682	-	16
13	T2Trap7	09Oct/Test6_1013_174406.txt	1	6186.226062514732	-	0
125	T2trap7	09Oct/Test6_1013_172140.txt	1	6382.821165082555	-	1
126	T2trap7	09Oct/Test6_1013_171912.txt	1	20957.721521671505	-	16
201	T2trap3	09Oct/Test6_1014_104241.txt	0.9	4926.644407181475	-	0
202	T2trap3	09Oct/Test6_1014_104143.txt	0.9	5122.283696273073	-	1
203	T2trap3	09Oct/Test6_1014_103925.txt	0.9	21701.90929193883	-	16
207	T2tri3	09Oct/Test6_1014_115116.txt	0.9	4690.505418331806	-	0
208	T2tri3	09Oct/Test6_1014_115031.txt	0.9	4819.688764986232	-	1
209	T2tri3	09Oct/Test6_1014_114936.txt	0.9	19453.377236646036	-	16
213	T2trap7	09Oct/Test6_1014_110923.txt	0.9	6186.0827504952385	-	0
214	T2trap7	09Oct/Test6_1014_111056.txt	0.9	6226.498709979921	-	1
215	T2trap7	09Oct/Test6_1014_111157.txt	0.9	20907.87620110287	-	16
204	T2trap3	09Oct/Test6_1014_105352.txt	0.8	4921.912429626856	-	0
205	T2trap3	09Oct/Test6_1014_105506.txt	0.8	5025.2966834235385	-	1
206	T2trap3	09Oct/Test6_1014_105556.txt	0.8	20280.314421558473	-	16
210	T2tri3	09Oct/Test6_1014_120053.txt	0.8	4690.505407383603	-	0
211	T2tri3	09Oct/Test6_1014_115834.txt	0.8	4761.575267068617	-	1
212	T2tri3	09Oct/Test6_1014_184709.txt	0.8	20395.293012397757	-	16
216	T2trap7	09Oct/Test6_1014_185941.txt	0.8	6185.0781154045735	-	0
217	T2trap7	09Oct/Test6_1014_190126.txt	0.8	6161.187450103015	-	1
218	T2trap7	09Oct/Test6_1014_190307.txt	0.8	20342.09660575451	-	16

Table D.18: Noise within MF within Threshold (Full)

Run	MF	File	Thresh	RMS_FZC	Inertia	Noise%
7	T2Trap3	09Oct/Test6_1013_164730.txt	1	4928.70904917916	-	0
201	T2trap3	09Oct/Test6_1014_104241.txt	0.9	4926.644407181475	-	0
204	T2trap3	09Oct/Test6_1014_105352.txt	0.8	4921.912429626856	-	0
9	T2Tri305	09Oct/Test6_1013_165232.txt	1	4665.367174228933	-	0
11	T2Tri300	09Oct/Test6_1014_101051.txt	1	4690.50542047633	-	0
207	T2tri3	09Oct/Test6_1014_115116.txt	0.9	4690.505418331806	-	0
210	T2tri3	09Oct/Test6_1014_120053.txt	0.8	4690.505407383603	-	0
13	T2Trap7	09Oct/Test6_1013_174406.txt	1	6186.226062514732	-	0
213	T2trap7	09Oct/Test6_1014_110923.txt	0.9	6186.0827504952385	-	0
216	T2trap7	09Oct/Test6_1014_185941.txt	0.8	6185.0781154045735	-	0
113	T2trap3	09Oct/Test6_1013_171440.txt	1	5092.52816718091	-	1
202	T2trap3	09Oct/Test6_1014_104143.txt	0.9	5122.283696273073	-	1
205	T2trap3	09Oct/Test6_1014_105506.txt	0.8	5025.2966834235385	-	1
117	T2tri305	09Oct/Test6_1013_172441.txt	1	4733.970234941016	-	1
119	T2tri300	09Oct/Test6_1014_101405.txt	1	4835.234635957925	-	1
208	T2tri3	09Oct/Test6_1014_115031.txt	0.9	4819.688764986232	-	1
211	T2tri3	09Oct/Test6_1014_115834.txt	0.8	4761.575267068617	-	1
125	T2trap7	09Oct/Test6_1013_172140.txt	1	6382.821165082555	-	1
214	T2trap7	09Oct/Test6_1014_111056.txt	0.9	6226.498709979921	-	1
217	T2trap7	09Oct/Test6_1014_190126.txt	0.8	6161.187450103015	-	1
114	T2trap3	09Oct/Test6_1013_171700.txt	1	20477.816256161575	-	16
203	T2trap3	09Oct/Test6_1014_103925.txt	0.9	21701.90929193883	-	16
206	T2trap3	09Oct/Test6_1014_105556.txt	0.8	20280.314421558473	-	16
118	T2tri305	09Oct/Test6_1013_172623.txt	1	19044.718434810937	-	16
120	T2tri300	09Oct/Test6_1014_101456.txt	1	22346.061184401682	-	16
209	T2tri3	09Oct/Test6_1014_114936.txt	0.9	19453.377236646036	-	16
212	T2tri3	09Oct/Test6_1014_184709.txt	0.8	20395.293012397757	-	16
126	T2trap7	09Oct/Test6_1013_171912.txt	1	20957.721521671505	-	16
215	T2trap7	09Oct/Test6_1014_111157.txt	0.9	20907.87620110287	-	16
218	T2trap7	09Oct/Test6_1014_190307.txt	0.8	20342.09660575451	-	16

Table D.19: Threshold within MF within Noise (Full)

Run	MF	File	Thresh	RMS_FZC	Inertia	Noise%
7	T2Trap3	09Oct/Test6_1013_164730.txt	1	4928.70904917916	-	0
9	T2Tri305	09Oct/Test6_1013_165232.txt	1	4665.367174228933	-	0
11	T2Tri300	09Oct/Test6_1014_101051.txt	1	4690.50542047633	-	0
13	T2Trap7	09Oct/Test6_1013_174406.txt	1	6186.226062514732	-	0
201	T2trap3	09Oct/Test6_1014_104241.txt	0.9	4926.644407181475	-	0
207	T2tri3	09Oct/Test6_1014_115116.txt	0.9	4690.505418331806	-	0
213	T2trap7	09Oct/Test6_1014_110923.txt	0.9	6186.0827504952385	-	0
204	T2trap3	09Oct/Test6_1014_105352.txt	0.8	4921.912429626856	-	0
210	T2tri3	09Oct/Test6_1014_120053.txt	0.8	4690.505407383603	-	0
216	T2trap7	09Oct/Test6_1014_185941.txt	0.8	6185.0781154045735	-	0
113	T2trap3	09Oct/Test6_1013_171440.txt	1	5092.52816718091	-	1
117	T2tri305	09Oct/Test6_1013_172441.txt	1	4733.970234941016	-	1
119	T2tri300	09Oct/Test6_1014_101405.txt	1	4835.234635957925	-	1
125	T2trap7	09Oct/Test6_1013_172140.txt	1	6382.821165082555	-	1
202	T2trap3	09Oct/Test6_1014_104143.txt	0.9	5122.283696273073	-	1
208	T2tri3	09Oct/Test6_1014_115031.txt	0.9	4819.688764986232	-	1
214	T2trap7	09Oct/Test6_1014_111056.txt	0.9	6226.498709979921	-	1
205	T2trap3	09Oct/Test6_1014_105506.txt	0.8	5025.2966834235385	-	1
211	T2tri3	09Oct/Test6_1014_115834.txt	0.8	4761.575267068617	-	1
217	T2trap7	09Oct/Test6_1014_190126.txt	0.8	6161.187450103015	-	1
114	T2trap3	09Oct/Test6_1013_171700.txt	1	20477.816256161575	-	16
118	T2tri305	09Oct/Test6_1013_172623.txt	1	19044.718434810937	-	16
120	T2tri300	09Oct/Test6_1014_101456.txt	1	22346.061184401682	-	16
126	T2trap7	09Oct/Test6_1013_171912.txt	1	20957.721521671505	-	16
203	T2trap3	09Oct/Test6_1014_103925.txt	0.9	21701.90929193883	-	16
209	T2tri3	09Oct/Test6_1014_114936.txt	0.9	19453.377236646036	-	16
215	T2trap7	09Oct/Test6_1014_111157.txt	0.9	20907.87620110287	-	16
206	T2trap3	09Oct/Test6_1014_105556.txt	0.8	20280.314421558473	-	16
212	T2tri3	09Oct/Test6_1014_184709.txt	0.8	20395.293012397757	-	16
218	T2trap7	09Oct/Test6_1014_190307.txt	0.8	20342.09660575451	-	16

Table D.20: MF within Threshold within Noise (Full)

Using PROD Rule		
Tri3 test6 ramp no noise	Using PROD Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1015_140414.txt	Ave	4487.476092914041
09Oct/Test6_1015_140456.txt	ldv[0]	5575.952484848338
09Oct/Test6_1015_140538.txt	ldv[1]	3803.034082089118
Tri3 test6 ramp no noise	Using PROD Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_123828.txt	Ave	4505.051077049851
09Oct/Test6_1015_123916.txt	ldv[0]	5654.497553684084
09Oct/Test6_1015_123946.txt	ldv[1]	3775.6647508631436
Tri3 test6 ramp no noise	Using PROD Th = 0.8	
Results File	Test Type	RMSE
09Oct/Test6_1015_123441.txt	Ave	4518.904393677202
09Oct/Test6_1015_123520.txt	ldv[0]	5799.247537531058
09Oct/Test6_1015_123550.txt	ldv[1]	3699.0913629809625

Table D.21: Results Tri3 Test6 Product Rule, MF thresholds

Tri3 test6 ramp no noise	calcAul aveTh = 0.5 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1015_100915.txt	mv-sp = ldv0	5575.936744486354
09Oct/Test6_1015_101118.txt	mv-sp = ldv1	4237.779672404135
Tri3 test6 ramp no noise	calcAul aveTh = 1 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1015_101516.txt	mv-sp = ldv0	5575.936744486354
09Oct/Test6_1015_101646.txt	mv-sp = ldv1	4204.396397022508
Tri3 test6 ramp no noise	calcAul aveTh = 2 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1015_101919.txt	mv-sp = ldv0	5575.936744486354
09Oct/Test6_1015_101808.txt	mv-sp = ldv1	4126.480036463399
Tri3 test6 ramp no noise	calcAul aveTh = 4 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1015_102059.txt	mv-sp = ldv0	5575.937562377387
09Oct/Test6_1015_102215.txt	mv-sp = ldv1	4102.423661506847

Table D.22: Results Tri3 Test 6 Min rule MF Th=1, DS Th Varied

Tri3 test6 ramp no noise	calcAul aveTh = 0.5 Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_111723.txt	mv-sp = ldv0	5654.488397458394
09Oct/Test6_1015_111835.txt	mv-sp = ldv1	4239.325797518904
Tri3 test6 ramp no noise	calcAul aveTh = 1 Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_111419.txt	mv-sp = ldv0	5654.488397458394
09Oct/Test6_1015_111243.txt	mv-sp = ldv1	4203.301199148781
Tri3 test6 ramp no noise	calcAul aveTh = 2 Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_110645.txt	mv-sp = ldv0	5654.488397458394
09Oct/Test6_1015_110511.txt	mv-sp = ldv1	4126.095913174164
Tri3 test6 ramp no noise	calcAul aveTh = 4 Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_110921.txt	mv-sp = ldv0	5654.488579564006
09Oct/Test6_1015_111038.txt	mv-sp = ldv1	4097.2541740416045

Table D.23: Results Tri3 Test 6 Min rule MF Th=0.9, DS Th Varied

Using MIN		
Trap3 test 6	step no noise	
Results File	Test Type	RMSE
Trap3 test 6	step no noise	
09Oct/Test6_1014_155020.txt	Ave	0.12706520044323755
09Oct/Test6_1014_155150.txt	Ldv[0]	0.1376927229130902
09Oct/Test6_1014_155245.txt	Ldv[1]	0.1345553047202415
Trap3 test 6	ramp no noise	
Results File	Test Type	RMSE
09Oct/Test6_1014_155856.txt	Ave	4928.70904917916
09Oct/Test6_1014_160002.txt	Ldv[0]	6086.949799617645
09Oct/Test6_1014_160050.txt	Ldv[1]	4089.400593244523
Trap3 test 6	ramp no noise	calcAul aveTh = 1 Th = 1
Results File	Test Type	RMSE
09Oct/Test6_1014_160532.txt	mv-sp = ldv0	6086.945930788766
09Oct/Test6_1014_161016.txt	mv-sp = ldv1	4384.7993284445765
Trap3 test 6	ramp no noise	calcAul aveTh = 0.5 Th = 1
Results File	Test Type	RMSE
09Oct/Test6_1014_161456.txt	mv-sp = ldv0	6086.945930788766
09Oct/Test6_1014_161322.txt	mv-sp = ldv1	4449.13609147451
Trap3 test 6	ramp no noise	calcAul aveTh = 0.5 Th = 0.8
Results File	Test Type	RMSE
09Oct/Test6_1014_162055.txt	mv-sp = ldv0	6163.548954476571
09Oct/Test6_1014_162213.txt	mv-sp = ldv1	4450.737242832209
Trap3 test 6	ramp no noise	calcAul aveTh = 1 Th = 0.8
Results File	Test Type	RMSE
09Oct/Test6_1014_162700.txt	mv-sp = ldv0	6163.548954476571
09Oct/Test6_1014_162503.txt	mv-sp = ldv1	4368.927979752041
Trap3 test 6	ramp no noise	calcAul aveTh = 2 Th = 0.8
Results File	Test Type	RMSE
09Oct/Test6_1014_163203.txt	mv-sp = ldv0	6163.548954476571
09Oct/Test6_1014_163417.txt	mv-sp = ldv1	4382.381330928145
Trap3 test 6	ramp no noise	calcAul aveTh = 4 Th = 0.8
Results File	Test Type	RMSE
09Oct/Test6_1014_163909.txt	mv-sp = ldv0	6163.548954476571
09Oct/Test6_1014_163715.txt	mv-sp = ldv1	4404.288576736168
Trap3 test 6	ramp no noise	calcAul aveTh = 4 Th = 0.9
Results File	Test Type	RMSE
09Oct/Test6_1014_164146.txt	mv-sp = ldv0	6117.160977981133
09Oct/Test6_1014_164307.txt	mv-sp = ldv1	4415.064955137956

Table D.24: Results Trap3 Test 6 Min rule MF Th=1,0.9,0.8, DS Th Varied

Using MIN		
Trap7 test6	ramp no noise	
Results File	Test Type	RMSE
09Oct/Test6_1014_194734.txt	Ave	6186.226062514732
09Oct/Test6_1014_194257.txt	ldv[0]	7607.971523311311
09Oct/Test6_1014_194548.txt	ldv[1]	5076.366521250794
Trap7 test6 ramp no noise	calcAul aveTh = 0.5 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_195410.txt	mv-sp = ldv0	7607.971577706904
09Oct/Test6_1014_195136.txt	mv-sp = ldv1	5136.724073717586
Trap7 test6 ramp no noise	calcAul aveTh = 1 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_195741.txt	mv-sp = ldv0	7607.971577706904
09Oct/Test6_1014_195929.txt	mv-sp = ldv1	5170.627494202966
Trap7 test6 ramp no noise	calcAul aveTh = 4 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_200314.txt	mv-sp = ldv0	7607.971577706904
09Oct/Test6_1014_200504.txt	mv-sp = ldv1	5318.882784061826
Trap7 test6 ramp no noise	calcAul aveTh = 4 Th = 0.8	
Results File	Test Type	RMSE
09Oct/Test6_1014_200752.txt	mv-sp = ldv0	7671.4443662245885
09Oct/Test6_1014_201059.txt	mv-sp = ldv1	5290.819747400814

Table D.25: Results Trap7 Test 6 Min rule MF Th=1,0.9,0.8, DS Th Varied

Using MIN		
Tri3 test6	ramp no noise Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_201739.txt	Ave	4690.50542047633
09Oct/Test6_1014_201849.txt	ldv[0]	5695.11522564894
09Oct/Test6_1014_202003.txt	ldv[1]	3994.4429936109077
Tri3 test6	ramp no noise Th = 0.9	
Results File	Test Type	RMSE
09Oct/Test6_1015_122435.txt	Ave	4690.505418331806
09Oct/Test6_1015_122611.txt	ldv[0]	5695.115235879456
09Oct/Test6_1015_122724.txt	ldv[1]	3992.675509380171
Tri3 test6 ramp no noise Th = 0.8		
Results File	Test Type	RMSE
09Oct/Test6_1015_122911.txt	Ave	4690.505407383603
09Oct/Test6_1015_123012.txt	ldv[0]	5701.578873506942
09Oct/Test6_1015_123101.txt	ldv[1]	3984.691196782237
Tri3 test6 ramp no noise	calcAul aveTh = 0.5 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_202356.txt	mv-sp = ldv0	5695.108147932632
09Oct/Test6_1014_202222.txt	mv-sp = ldv1	4343.402362200754
Tri3 test6 ramp no noise	calcAul aveTh = 1 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_202549.txt	mv-sp = ldv0	5695.108147932632
09Oct/Test6_1014_202720.txt	mv-sp = ldv1	4301.7644580347505
Tri3 test6 ramp no noise	calcAul aveTh = 2 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_203346.txt	mv-sp = ldv0	5695.108147932632
09Oct/Test6_1014_202926.txt	mv-sp = ldv1	4260.274789755921
Tri3 test6 ramp no noise	calcAul aveTh = 4 Th = 1	
Results File	Test Type	RMSE
09Oct/Test6_1014_203630.txt	mv-sp = ldv0	5695.108187650267
09Oct/Test6_1014_203859.txt	mv-sp = ldv1	4271.283587778327
Tri3 test6 ramp no noise	calcAul aveTh = 4 Th = 0.8	
Results File	Test Type	RMSE
09Oct/Test6_1014_204206.txt	mv-sp = ldv0	5701.572235374108
09Oct/Test6_1014_204047.txt	mv-sp = ldv1	4270.850813400014

Table D.26: Results Tri3 Test 6 Min rule MF Th=1,0.8, DS Th Varied

TestFile	RMS_FZC	RMS_PID
	Ramp Results	
Test3_1013_164006.txt	4979.322366748708	1.724041093639547E14
Test3_1013_165738.txt	5212.598759742	2.1138177227409606E14
Test3_1013_170354.txt	20659.444200403806	1.7727146794917115E15
Test5_1013_164105.txt	5880.3508018661505	1.7234948857439138E14
Test5_1013_170250.txt	5960.025499043392	1.985443712209436E14
Test5_1013_170423.txt	18718.486127693217	1.6267870384682035E15
	Step Results	
Test1_1013_152315.txt	0.12706231173677582	
Test1_1013_160059.txt	0.13293384142269804	
Test1_1013_160323.txt	0.6464137786675844	
Test3_1013_160146.txt	0.13446267418970737	0.06508788508394939
Test3_1013_160355.txt	0.6610152976874848	0.6791094818154001
Test5_1013_152456.txt	0.14349678574442584	0.16085856481734423
Test5_1013_160220.txt	0.1450100332319987	0.16253502105199102
Test5_1013_160429.txt	0.3552517056477986	0.3721596888106293

Table D.27: Type-1 Trap Ramp and Step

TestFile	RMS_FZC	RMS_PID
	Ramp Results	
Test1_1013_164447.txt	4705.181234591626	
Test1_1013_171005.txt	21464.7080099066	
Test1_1013_171153.txt	4890.415783245636	
Test5_1013_164512.txt	5648.446445334611	1.7234948857439138E14
Test5_1013_171043.txt	19964.148352048203	1.7737181876650958E15
Test5_1013_171219.txt	5925.517432351581	2.0447995152081725E14
	Step Results	
Test1_1013_153048.txt	0.12687162797950133	
Test1_1013_161713.txt	0.13448660195587722	
Test1_1013_161900.txt	0.6517869685977625	
Test5_1013_153240.txt	0.14355969835107626	0.16085856481734423
Test5_1013_161757.txt	0.14555075150176158	0.1612047898567448
Test5_1013_162026.txt	0.35454394460847677	0.35785552046820035

Table D.28: Type-1 Tri Ramp and Step

TestFile	RMS_FZC	RMS_PID
	Ramp Results	
Test6_1013_164730.txt	4928.70904917916	
Test6_1013_171440.txt	5092.52816718091	
Test6_1013_171700.txt	20477.816256161575	
Test7_1013_164811.txt	5842.682671520783	
Test7_1013_171522.txt	5841.643984365342	
Test7_1013_171741.txt	19482.201390283008	
	Step Results	
Test6_1013_153409.txt	0.12706520044323755	
Test6_1013_162808.txt	0.13409954352714895	
Test6_1013_162955.txt	0.6580162819186266	
Test7_1013_153447.txt	0.12706522008588153	
Test7_1013_162853.txt	0.1334153901784213	
Test7_1013_163040.txt	0.6596848615109742	
	Threshold Results	
Test6_1014_103925.txt	21701.90929193883	
Test6_1014_104143.txt	5122.283696273073	
Test6_1014_104241.txt	4926.644407181475	
Test6_1014_104342.txt	0.12717459956678606	
Test6_1014_104443.txt	0.13243740180986685	
Test6_1014_104527.txt	0.6723357726340708	
Test6_1014_105043.txt	0.1272843206004966	
Test6_1014_105159.txt	0.13269804359263634	
Test6_1014_105242.txt	0.652263722693561	
Test6_1014_105352.txt	4921.912429626856	
Test6_1014_105506.txt	5025.2966834235385	
Test6_1014_105556.txt	20280.314421558473	
Test7_1014_104656.txt	5840.984309612984	
Test7_1014_104806.txt	0.12717461919640788	
Test7_1014_105725.txt	0.12728434022833013	
Test7_1014_105826.txt	5836.523615884737	

Table D.29: Type-2 Trap Ramp,Step and Threshold

TestFile	RMS_FZC	RMS_PID
	Ramp Results (* = Also in aul.txt)	
Test6_1013_165232.txt	4665.367174228933	
Test6_1013_172441.txt	4733.970234941016	
Test6_1013_172623.txt	19044.718434810937	
Test6_1014_101051.txt	4690.50542047633 *	
Test6_1014_101405.txt	4835.234635957925 *	
Test6_1014_101456.txt	22346.061184401682 *	
Test7_1013_165316.txt	5617.617940829641	
Test7_1013_172526.txt	5882.35502448692	
Test7_1013_172705.txt	20045.78854427974	
Test7_1014_100851.txt	0.6438532538198586 *	
Test7_1014_101557.txt	5753.713280872572 *	
Test7_1014_101738.txt	20903.88351814896 *	
	Step Results	
Test6_1013_153909.txt	0.12689009876269952	
Test6_1013_154226.txt	0.13338347190364339	
Test6_1013_154855.txt	0.6817511887357052	
Test6_1014_95946.txt	0.12687972047153626 *	
Test6_1014_100536.txt	0.13322276065760894 *	
Test6_1014_100807.txt	0.6516664865711289 *	
Test7_1013_153951.txt	0.1268899511857723	
Test7_1013_154632.txt	0.13261642137117052	
Test7_1013_154938.txt	0.6854057238504982	
Test7_1014_100404.txt	0.1268797408843304 *	
Test7_1014_100659.txt	0.1342453213653109 *	
Test7_1014_100851.txt	0.6438532538198586 *	

Table D.30: Type-2 Tri Ramp,Step and Threshold

TestFile	RMS_FZC	RMS_PID
	Ramp Results	
Test1_1013_164240.txt	6254.108810814268	
Test1_1013_170610.txt	6368.685402443599	
Test1_1013_170752.txt	17955.948955554606	
Test5_1013_164322.txt	6873.464784210848	1.7234948857439138E14
Test5_1013_170651.txt	6993.056175876338	2.2171581772568944E14
Test5_1013_170840.txt	19066.41118599032	1.903794471583325E15
	Step Results	
Test1_1013_152839.txt	0.1368759130985531	
Test1_1013_160700.txt	0.14393363052928057	
Test1_1013_161305.txt	0.6644613693207286	
Test3_1013_160911.txt	0.14334641545070992	0.06325624654972502
Test5_1013_152921.txt	0.13802119469739438	0.16085856481734423
Test5_1013_161103.txt	0.13887901931576446	0.1626165509798357
Test5_1013_161500.txt	0.35070640106997636	0.3861312838832343

Table D.31: Type-1 Seven MF Trap Ramp and Step

TestFile	RMS_FZC
	Ramp Results
Test6_1013_171912.txt	20957.721521671505
Test6_1013_172140.txt	6382.821165082555
Test6_1013_174406.txt	6186.226062514732
Test7_1013_172017.txt	22064.612240888895
Test7_1013_172243.txt	7022.321511676284
Test7_1013_175146.txt	6821.68686267856
	Step Results
Test6_305_123717.txt	0.13673363844713102
Test6_305_130257.txt	0.14135210687798333
Test6_305_131354.txt	0.6746376488056046
Test6_1013_153629.txt	0.13674059734288593
Test6_1013_155229.txt	0.1427775741213351
Test6_1013_155532.txt	0.6517598358484246
Test6_1014_110348.txt	0.1367360750450983
Test6_1014_110545.txt	0.1416110970132653
Test6_1014_110657.txt	0.6678716576511934
Test7_305_131728.txt	0.13673365204384724
Test7_1013_153729.txt	0.13674061093388382
Test7_1013_155401.txt	0.1436613218841645
Test7_1013_155639.txt	0.6820065767882226
Test7_1014_111431.txt	0.13673608863731834
Test7_1014_111801.txt	0.13673365204384724
Redundent not sure what used for	
Test7_1014_111801.txt	0.13673365204384724
Test7_1014_112134.txt	0.13673365204384724
Test7_1014_112314.txt	0.13673365204384724
Test7_1014_112433.txt	0.13673365204384724
Test7_1014_112816.txt	0.13673365204384724
Test7_1014_112959.txt	0.13673365204384724
Test7_1014_113223.txt	0.13673365204384724

Table D.32: Type-2 Seven MF Trap Ramp and Step

TestFile	RMS
	Threshold Results
Test6_1014_110923.txt	6186.0827504952385
Test6_1014_111056.txt	6226.498709979921
Test6_1014_111157.txt	20907.87620110287
Test6_1014_185941.txt	6185.0781154045735
Test6_1014_190126.txt	6161.187450103015
Test6_1014_190307.txt	20342.09660575451
Test7_1014_111606.txt	6821.597557815642
Test7_1014_111930.txt	6820.912391494358
Test7_1014_185550.txt	6820.912391494358

Table D.33: Type-2 Seven MF Trapezoidal Threshold

Appendix E

Chapter 7 Tables and Results

This appendix contains the tables and results for Chapter 7.

Results File			Values	Average	
Tri3 test6 ramp no noise	MIN Th 1				Max/Min
09Oct/Test6_1014_201739.txt	Ave	RMS_FZC:	4690.50542		
09Oct/Test6_1014_201849.txt	ldv[0]	RMS_FZC:	5695.115226	4844.77911	5799.247538
09Oct/Test6_1014_202003.txt	ldv[1]	RMS_FZC:	3994.442994		3699.091363
Tri3 test6 ramp no noise	MIN Th = 0.9				
09Oct/Test6_1015_122435.txt	Ave	RMS_FZC:	4690.505418		
09Oct/Test6_1015_122611.txt	ldv[0]	RMS_FZC:	5695.115236	4843.895373	
09Oct/Test6_1015_122724.txt	ldv[1]	RMS_FZC:	3992.675509		
Tri3 test6 ramp no noise	MIN Th = 0.8				
09Oct/Test6_1015_122911.txt	Ave	RMS_FZC:	4690.505407		
09Oct/Test6_1015_123012.txt	ldv[0]	RMS_FZC:	5701.578874	4843.135035	
09Oct/Test6_1015_123101.txt	ldv[1]	RMS_FZC:	3984.691197		
Tri3 test6 ramp no noise	PROD Th = 1				
09Oct/Test6_1015_140414.txt	Ave	RMS_FZC:	4487.476093		
09Oct/Test6_1015_140456.txt	ldv[0]	RMS_FZC:	5575.952485	4689.493283	
09Oct/Test6_1015_140538.txt	ldv[1]	RMS_FZC:	3803.034082		
Tri3 test6 ramp no noise	PROD Th = 0.9				
09Oct/Test6_1015_123828.txt	Ave	RMS_FZC:	4505.051077		
09Oct/Test6_1015_123916.txt	ldv[0]	RMS_FZC:	5654.497554	4715.081152	Picked this one
09Oct/Test6_1015_123946.txt	ldv[1]	RMS_FZC:	3775.664751		
Tri3 test6 ramp no noise	PROD Th = 0.8				
09Oct/Test6_1015_123441.txt	Ave	RMS_FZC:	4518.904394		
09Oct/Test6_1015_123520.txt	ldv[0]	RMS_FZC:	5799.247538	4749.16945	
09Oct/Test6_1015_123550.txt	ldv[1]	RMS_FZC:	3699.091363		

Table E.1: Simulation Results for Dual Surface Selection

	$\Delta Error$	-120	-100	-80	-60	-40	-20	
	Error							
0	-120	0.59011172	0.59011172	0.59011172	0.588882077	0.560496411	0.581503712	
0	-100	0.59011172	0.59011172	0.59011172	0.588882077	0.560496411	0.581503712	
0	-80	0.59011172	0.59011172	0.59011172	0.588882077	0.560496411	0.581503712	
0	-60	0.588882077	0.588882077	0.588882077	0.588882077	0.560496411	0.581503712	
0	-40	0.560496411	0.560496411	0.560496411	0.560496411	0.398937475	0.398937475	
0	-20	0.581503712	0.581503712	0.581503712	0.581503712	0.398937475	-0.042562596	
0	0	0.59011172	0.59011172	0.59011172	0.588882077	0.398937475	-0.042562596	
0	20	0.491105835	0.491105835	0.491105835	0.491105835	0.134715578	-0.27488257	
0	40	0.178256162	0.178256162	0.178256162	0.178256162	-0.16323528	-0.585928421	
0	60	-0.037606375	-0.037606375	-0.037606375	-0.037606375	-0.406131254	-0.66654689	
0	80	-0.036781557	-0.036781557	-0.036781557	-0.037606375	-0.406131254	-0.66654689	
0	100	-0.036781557	-0.036781557	-0.036781557	-0.037606375	-0.406131254	-0.66654689	
0	120	-0.036781557	-0.036781557	-0.036781557	-0.037606375	-0.406131254	-0.66654689	
	$\Delta Error$	0	20	40	60	80	100	120
	Error							
0	-120	0.59011172	0.491105835	0.178256162	-0.037606375	-0.036781557	-0.036781557	-0.036781557
0	-100	0.59011172	0.491105835	0.178256162	-0.037606375	-0.036781557	-0.036781557	-0.036781557
0	-80	0.59011172	0.491105835	0.178256162	-0.037606375	-0.036781557	-0.036781557	-0.036781557
0	-60	0.588882077	0.491105835	0.178256162	-0.037606375	-0.037606375	-0.037606375	-0.037606375
0	-40	0.398937475	0.134715578	-0.16323528	-0.406131254	-0.406131254	-0.406131254	-0.406131254
0	-20	-0.042562596	-0.27488257	-0.585928421	-0.66654689	-0.66654689	-0.66654689	-0.66654689
0	0	-0.036781557	-0.27488257	-0.585928421	-0.659762008	-0.658630029	-0.658630029	-0.658630029
0	20	-0.27488257	-0.27488257	-0.585928421	-0.66654689	-0.66654689	-0.66654689	-0.66654689
0	40	-0.585928421	-0.585928421	-0.585928421	-0.685781943	-0.685781943	-0.685781943	-0.685781943
0	60	-0.659762008	5468	-0.685781943	-0.659762008	-0.659762008	-0.659762008	-0.659762008
0	80	-0.658630029	-0.66654689	-0.685781943	-0.659762008	-0.658630029	-0.658630029	-0.658630029
0	100	-0.658630029	-0.66654689	-0.685781943	-0.659762008	-0.658630029	-0.658630029	-0.658630029
0	120	-0.658630029	-0.66654689	-0.685781943	-0.659762008	-0.658630029	-0.658630029	-0.658630029

Table E.2: Lower Surface for Dual Type-2 Controller

	$\Delta Error$ Error	-120	-100	-80	-60	-40	-20	
1	-120	0.658630029	0.658630029	0.658630029	0.659762008	0.685781943	0.66654689	
1	-100	0.658630029	0.658630029	0.658630029	0.659762008	0.685781943	0.66654689	
1	-80	0.658630029	0.658630029	0.658630029	0.659762008	0.685781943	0.66654689	
1	-60	0.659762008	0.659762008	0.659762008	0.659762008	0.685781943	0.66654689	
1	-40	0.685781943	0.685781943	0.685781943	0.685781943	0.585928421	0.585928421	
1	-20	0.66654689	0.66654689	0.66654689	0.66654689	0.585928421	0.27488257	
1	0	0.658630029	0.658630029	0.658630029	0.659762008	0.585928421	0.27488257	
1	20	0.66654689	0.66654689	0.66654689	0.66654689	0.585928421	0.27488257	
1	40	0.406131254	0.406131254	0.406131254	0.406131254	0.16323528	-0.134715578	
1	60	0.037606375	0.037606375	0.037606375	0.037606375	-0.178256162	-0.491105835	
1	80	0.036781557	0.036781557	0.036781557	0.037606375	-0.178256162	-0.491105835	
1	100	0.036781557	0.036781557	0.036781557	0.037606375	-0.178256162	-0.491105835	
1	120	0.036781557	0.036781557	0.036781557	0.037606375	-0.178256162	-0.491105835	
	$\Delta Error$ Error	0	20	40	60	80	100	120
1	-120	0.658630029	0.66654689	0.406131254	0.037606375	0.036781557	0.036781557	0.036781557
1	-100	0.658630029	0.66654689	0.406131254	0.037606375	0.036781557	0.036781557	0.036781557
1	-80	0.658630029	0.66654689	0.406131254	0.037606375	0.036781557	0.036781557	0.036781557
1	-60	0.659762008	0.66654689	0.406131254	0.037606375	0.037606375	0.037606375	0.037606375
1	-40	0.585928421	0.585928421	0.16323528	-0.178256162	-0.178256162	-0.178256162	-0.178256162
1	-20	0.27488257	0.27488257	-0.134715578	-0.491105835	-0.491105835	-0.491105835	-0.491105835
1	0	0.036781557	0.042562596	-0.398937475	-0.588882077	-0.59011172	-0.59011172	-0.59011172
1	20	0.042562596	0.042562596	-0.398937475	-0.581503712	-0.581503712	-0.581503712	-0.581503712
1	40	-0.398937475	-0.398937475	-0.398937475	-0.560496411	-0.560496411	-0.560496411	-0.560496411
1	60	-0.588882077	-0.581503712	-0.560496411	-0.588882077	-0.588882077	-0.588882077	-0.588882077
1	80	-0.59011172	-0.581503712	-0.560496411	-0.588882077	-0.59011172	-0.59011172	-0.59011172
1	100	-0.59011172	-0.581503712	-0.560496411	-0.588882077	-0.59011172	-0.59011172	-0.59011172
1	120	-0.59011172	-0.581503712	-0.560496411	-0.588882077	-0.59011172	-0.59011172	-0.59011172

Table E.3: Upper Surface for Dual Type-2 Controller

	$\Delta Error$ Error	-120	-100	-80	-60	-40	-20	
A	-120	0.624370874	0.624370874	0.624370874	0.624322042	0.623139176	0.624025301	
A	-100	0.624370874	0.624370874	0.624370874	0.624322042	0.623139176	0.624025301	
A	-80	0.624370874	0.624370874	0.624370874	0.624322042	0.623139176	0.624025301	
A	-60	0.624322042	0.624322042	0.624322042	0.624322042	0.623139176	0.624025301	
A	-40	0.623139176	0.623139176	0.623139176	0.623139176	0.492432948	0.492432948	
A	-20	0.624025301	0.624025301	0.624025301	0.624025301	0.492432948	0.116159987	
A	0	0.624370874	0.624370874	0.624370874	0.624322042	0.492432948	0.116159987	
A	20	0.578826362	0.578826362	0.578826362	0.578826362	0.360321999	2.7755E-17	
A	40	0.292193708	0.292193708	0.292193708	0.292193708	-2.8865E-15	-0.360321999	
A	60	-1.0408E-17	-1.0408E-17	-1.0408E-17	-1.0408E-17	-0.292193708	-0.578826362	
A	80	-6.9388E-18	-6.9388E-18	-6.9388E-18	-1.0408E-17	-0.292193708	-0.578826362	
A	100	-6.9388E-18	-6.9388E-18	-6.9388E-18	-1.0408E-17	-0.292193708	-0.578826362	
A	120	-6.9388E-18	-6.9388E-18	-6.9388E-18	-1.0408E-17	-0.292193708	-0.578826362	

	$\Delta Error$ Error	0	20	40	60	80	100	120
A	-120	0.624370874	0.578826362	0.292193708	-1.0408E-17	-6.9388E-18	-6.9388E-18	-6.9388E-18
A	-100	0.624370874	0.578826362	0.292193708	-1.0408E-17	-6.9388E-18	-6.9388E-18	-6.9388E-18
A	-80	0.624370874	0.578826362	0.292193708	-1.0408E-17	-6.9388E-18	-6.9388E-18	-6.9388E-18
A	-60	0.624322042	0.578826362	0.292193708	-1.0408E-17	-1.0408E-17	-1.0408E-17	-1.0408E-17
A	-40	0.492432948	0.360321999	-2.8865E-15	-0.292193708	-0.292193708	-0.292193708	-0.292193708
A	-20	0.116159987	2.77554E-17	-0.360321999	-0.578826362	-0.578826362	-0.578826362	-0.578826362
A	0	-6.9388E-18	-0.116159987	-0.492432948	-0.624322042	-0.624370874	-0.624370874	-0.624370874
A	20	-0.116159987	-0.116159987	-0.492432948	-0.624025301	-0.624025301	-0.624025301	-0.624025301
A	40	-0.492432948	-0.492432948	-0.492432948	-0.623139176	-0.623139176	-0.623139176	-0.623139176
A	60	-0.624322042	-0.624025301	-0.623139176	-0.624322042	-0.624322042	-0.624322042	-0.624322042
A	80	-0.624370874	-0.624025301	-0.623139176	-0.624322042	-0.624370874	-0.624370874	-0.624370874
A	100	-0.624370874	-0.624025301	-0.623139176	-0.624322042	-0.624370874	-0.624370874	-0.624370874
A	120	-0.624370874	-0.624025301	-0.623139176	-0.624322042	-0.624370874	-0.624370874	-0.624370874

Table E.4: Average Type-2 Surface for Miabot Controller

	$\Delta Error$ Error	-120	-100	-80	-60	-40	-20	
0	-120	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	
0	-100	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	
0	-80	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	
0	-60	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	0.628556290	
0	-40	0.628556290	0.628556290	0.628556290	0.628556290	0.508607888	0.508607888	
0	-20	0.628556290	0.628556290	0.628556290	0.628556290	0.508607888	0.149879192	
0	0	0.628556290	0.628556290	0.628556290	0.628556290	0.508607888	0.149879192	
0	20	0.582059980	0.582059980	0.582059980	0.582059980	0.357906586	0.000000000	
0	40	0.302628082	0.302628082	0.302628082	0.302628082	0.000000000	-0.357906586	
0	60	0.000000000	0.000000000	0.000000000	0.000000000	-0.302628082	-0.582059980	
0	80	0.000000000	0.000000000	0.000000000	0.000000000	-0.302628082	-0.582059980	
0	100	0.000000000	0.000000000	0.000000000	0.000000000	-0.302628082	-0.582059980	
0	120	0.000000000	0.000000000	0.000000000	0.000000000	-0.302628082	-0.582059980	
	$\Delta Error$ Error	0	20	40	60	80	100	120
0	-120	0.628556290	0.582059980	0.302628082	0.000000000	0.000000000	0.000000000	0.000000000
0	-100	0.628556290	0.582059980	0.302628082	0.000000000	0.000000000	0.000000000	0.000000000
0	-80	0.628556290	0.582059980	0.302628082	0.000000000	0.000000000	0.000000000	0.000000000
0	-60	0.628556290	0.582059980	0.302628082	0.000000000	0.000000000	0.000000000	0.000000000
0	-40	0.508607888	0.357906586	0.000000000	-0.302628082	-0.302628082	-0.302628082	-0.302628082
0	-20	0.149879192	0.000000000	-0.357906586	-0.582059980	-0.582059980	-0.582059980	-0.582059980
0	0	0.000000000	-0.149879192	-0.508607888	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	20	-0.149879192	-0.149879192	-0.508607888	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	40	-0.508607888	-0.508607888	-0.508607888	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	60	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	80	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	100	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290
0	120	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290	-0.628556290

Table E.5: Type-1 Surface for Miabot Controller

$\Delta Error$ Error	-120	-100	-80	-60	-40	-20	
-120	0.068518309	0.068518309	0.068518309	0.070879932	0.125285533	0.085043179	
-100	0.068518309	0.068518309	0.068518309	0.070879932	0.125285533	0.085043179	
-80	0.068518309	0.068518309	0.068518309	0.070879932	0.125285533	0.085043179	
-60	0.070879932	0.070879932	0.070879932	0.070879932	0.125285533	0.085043179	
-40	0.125285533	0.125285533	0.125285533	0.125285533	0.186990946	0.186990946	
-20	0.085043179	0.085043179	0.085043179	0.085043179	0.186990946	0.317445166	
0	0.068518309	0.068518309	0.068518309	0.070879932	0.186990946	0.317445166	
20	0.175441056	0.175441056	0.175441056	0.175441056	0.451212843	0.549765140	
40	0.227875092	0.227875092	0.227875092	0.227875092	0.326470560	0.451212843	
60	0.075212749	0.075212749	0.075212749	0.075212749	0.227875092	0.175441056	
80	0.073563114	0.073563114	0.073563114	0.075212749	0.227875092	0.175441056	
100	0.073563114	0.073563114	0.073563114	0.075212749	0.227875092	0.175441056	
120	0.073563114	0.073563114	0.073563114	0.075212749	0.227875092	0.175441056	
$\Delta Error$ Error	0	20	40	60	80	100	120
-120	0.068518309	0.175441056	0.227875092	0.075212749	0.073563114	0.073563114	0.073563114
-100	0.068518309	0.175441056	0.227875092	0.075212749	0.073563114	0.073563114	0.073563114
-80	0.068518309	0.175441056	0.227875092	0.075212749	0.073563114	0.073563114	0.073563114
-60	0.070879932	0.175441056	0.227875092	0.075212749	0.075212749	0.075212749	0.075212749
-40	0.186990946	0.451212843	0.326470560	0.227875092	0.227875092	0.227875092	0.227875092
-20	0.317445166	0.549765140	0.451212843	0.175441056	0.175441056	0.175441056	0.175441056
0	0.073563114	0.317445166	0.186990946	0.070879932	0.068518309	0.068518309	0.068518309
20	0.317445166	0.317445166	0.186990946	0.085043179	0.085043179	0.085043179	0.085043179
40	0.186990946	0.186990946	0.186990946	0.125285533	0.125285533	0.125285533	0.125285533
60	0.070879932	0.085043179	0.125285533	0.070879932	0.070879932	0.070879932	0.070879932
80	0.068518309	0.085043179	0.125285533	0.070879932	0.068518309	0.068518309	0.068518309
100	0.068518309	0.085043179	0.125285533	0.070879932	0.068518309	0.068518309	0.068518309
120	0.068518309	0.085043179	0.125285533	0.070879932	0.068518309	0.068518309	0.068518309

Table E.6: Difference Between Upper Surface and Lower Surface

$\Delta Error$ Error	-120	-100	-80	-60	-40	-20	
-120	-0.004185415	-0.004185415	-0.004185415	-0.004234247	-0.005417113	-0.004530989	
-100	-0.004185415	-0.004185415	-0.004185415	-0.004234247	-0.005417113	-0.004530989	
-80	-0.004185415	-0.004185415	-0.004185415	-0.004234247	-0.005417113	-0.004530989	
-60	-0.004234247	-0.004234247	-0.004234247	-0.004234247	-0.005417113	-0.004530989	
-40	-0.005417113	-0.005417113	-0.005417113	-0.005417113	-0.016174940	-0.016174940	
-20	-0.004530989	-0.004530989	-0.004530989	-0.004530989	-0.016174940	-0.033719204	
0	-0.004185415	-0.004185415	-0.004185415	-0.004234247	-0.016174940	-0.033719204	
20	-0.003233617	-0.003233617	-0.003233617	-0.003233617	0.002415414	0.000000000	
40	-0.010434374	-0.010434374	-0.010434374	-0.010434374	0.000000000	-0.002415414	
60	0.000000000	0.000000000	0.000000000	0.000000000	0.010434374	0.003233617	
80	0.000000000	0.000000000	0.000000000	0.000000000	0.010434374	0.003233617	
100	0.000000000	0.000000000	0.000000000	0.000000000	0.010434374	0.003233617	
120	0.000000000	0.000000000	0.000000000	0.000000000	0.010434374	0.003233617	

$\Delta Error$ Error	0	20	40	60	80	100	120
-120	-0.004185415	-0.003233617	-0.010434374	0.000000000	0.000000000	0.000000000	0.000000000
-100	-0.004185415	-0.003233617	-0.010434374	0.000000000	0.000000000	0.000000000	0.000000000
-80	-0.004185415	-0.003233617	-0.010434374	0.000000000	0.000000000	0.000000000	0.000000000
-60	-0.004234247	-0.003233617	-0.010434374	0.000000000	0.000000000	0.000000000	0.000000000
-40	-0.016174940	0.002415414	0.000000000	0.010434374	0.010434374	0.010434374	0.010434374
-20	-0.033719204	0.000000000	-0.002415414	0.003233617	0.003233617	0.003233617	0.003233617
0	0.000000000	0.033719204	0.016174940	0.004234247	0.004185415	0.004185415	0.004185415
20	0.033719204	0.033719204	0.016174940	0.004530989	0.004530989	0.004530989	0.004530989
40	0.016174940	0.016174940	0.016174940	0.005417113	0.005417113	0.005417113	0.005417113
60	0.004234247	0.004530989	0.005417113	0.004234247	0.004234247	0.004234247	0.004234247
80	0.004185415	0.004530989	0.005417113	0.004234247	0.004185415	0.004185415	0.004185415
100	0.004185415	0.004530989	0.005417113	0.004234247	0.004185415	0.004185415	0.004185415
120	0.004185415	0.004530989	0.005417113	0.004234247	0.004185415	0.004185415	0.004185415

Table E.7: Difference between Average Type-2 Surface and Type-1 Surface

Test No	Speed	Controller	Type2 Action	Ramp
Test01	20	All PID 80,16,16	-	No Ramp .rA=0
Test02	20	All PID 80,16,16	-	Ramp .rA=1
Test03	20	FZC Type2 .pZ=2	Average .pF=1	No Ramp .rA=0
Test04	20	FZC Type2 .pZ=2	Average .pF=1	Ramp .rA=1
Test05	20	FZC Type1 .pZ=1	-	No Ramp .rA=0
Test06	20	FZC Type1 .pZ=1	-	Ramp .rA=1
Test07	10	All PID 80,16,16	-	No Ramp .rA=0
Test08	10	All PID 80,16,16	-	Ramp .rA=1
Test09	10	FZC Type2 .pZ=2	Average .pF=1	No Ramp .rA=0
Test10	10	FZC Type2 .pZ=2	Average .pF=1	Ramp .rA=1
Test11	10	FZC Type1 .pZ=1	-	No Ramp .rA=0
Test12	10	FZC Type1 .pZ=1	-	Ramp .rA=1
Test13	40	All PID 80,16,16	-	No Ramp .rA=0
Test14	40	All PID 80,16,16	-	Ramp .rA=1
Test15	40	FZC Type2 .pZ=2	Average .pF=1	No Ramp .rA=0
Test16	40	FZC Type2 .pZ=2	Average .pF=1	Ramp .rA=1
Test17	40	FZC Type1 .pZ=1	-	No Ramp .rA=0
Test18	40	FZC Type1 .pZ=1	-	Ramp .rA=1
Test19	80	All PID 80,16,16	-	No Ramp .rA=0
Test20	80	All PID 80,16,16	-	Ramp .rA=1
Test21	80	FZC Type2 .pZ=2	Average .pF=1	No Ramp .rA=0
Test22	80	FZC Type2 .pZ=2	Average .pF=1	Ramp .rA=1
Test23	80	FZC Type1 .pZ=1	-	No Ramp .rA=0
Test24	80	FZC Type1 .pZ=1	-	Ramp .rA=1
Test25	120	All PID 80,16,16	-	No Ramp .rA=0
Test26	120	All PID 80,16,16	-	Ramp .rA=1
Test27	120	FZC Type2 .pZ=2	Average .pF=1	No Ramp .rA=0
Test28	120	FZC Type2 .pZ=2	Average .pF=1	Ramp .rA=1
Test29	120	FZC Type1 .pZ=1	-	No Ramp .rA=0
Test30	120	FZC Type1 .pZ=1	-	Ramp .rA=1

Table E.8: Key to Tests for Three Controller Comparison

Test Number	Mean R1	Mean R2	Mean R3	Mean R4	SD R1	SD R2	SD R3	SD R4	Global Mean	Global SD
1	25.09000272	25.09290203	24.14207481	29.24598542	0.147843893	0.444900459	1.261833991	0.478718392	24.85102855	0.800764332
2	17.45974878	17.64959836	17.70343733	29.83270173	0.100870192	0.089279501	0.099231564	0.248433666	17.60426149	0.141544568
” 3	17.402191	17.59758701	17.67887257	**	0.081629133	0.095376954	0.088564225	**	17.55955019	0.145850508
4	17.40473531	17.59168495	17.72447482	29.86135059	0.070119982	0.142776572	0.214775064	0.114737154	17.5736317	0.199907581
5	19.67558359	18.66102101	16.7067144	25.21107161	0.474372381	0.941413083	2.826020281	0.720883963	18.347773	2.102554947
6	14.28268319	13.22348508	12.58021071	17.66732539	0.96060956	1.370095881	1.240537857	1.14542014	13.59226554	1.308622404
7	8.16885226	8.273226326	8.579779005	**	0.109681312	0.069526767	0.098632548	**	8.340619196	0.199028451
8	8.164887833	8.19310257	8.398054652	**	0.162209221	0.139233862	0.301551398	**	8.25581582	0.236391872
9	7.957087952	8.082665608	8.274217891	**	0.082580709	0.175289891	0.137559964	**	8.104657151	0.187389615
10	7.982604408	7.960289284	8.083053831	**	0.125865232	0.196071752	0.094166959	**	8.008649174	0.150584251
11	8.275204595	8.193124907	8.254653468	**	0.069204703	0.173353484	0.219955648	**	8.23731686	0.145879871
12	8.230531075	8.255929381	8.317096896	**	0.107952304	0.136814283	0.962325139	**	8.267852451	0.547950039
13	44.1641471	46.50201832	41.99341213	49.41650528	4.927415809	1.619256949	3.536202351	3.3323262	44.38898158	3.976834124
14	42.39186889	42.59654727	**	41.8185966	0.371427703	0.289791811	**	4.549522775	42.49420808	0.342159105
15	42.2741549	42.25881751	**	44.0068068	0.304099001	0.45460729	**	2.834076267	42.26648621	0.37832556
16	42.41333427	42.52266614	**	45.37765105	0.239608916	0.846417784	**	2.794103772	42.4680002	0.61091205
17	42.12539208	42.26124487	**	43.81854005	0.576207865	0.767140714	**	1.739998054	42.19331846	0.659164447
18	37.94474443	38.35522042	**	37.86507314	0.434967238	0.356613623	**	0.37150991	38.14998242	0.439863995
19	79.73131003	79.67493463	**	**	0.788362157	1.195485181	**	**	79.70714915	0.887727681
20	63.78283309	65.03387999	**	64.91227259	0.899909805	1.047650589	**	0.230393016	64.40835653	1.148129848
21	67.63766954	63.19073752	**	**	**	2.40262836	**	**	63.93189286	2.40262836
22	58.94807542	59.6581579	**	52.97458458	0.668574769	1.119168287	**	3.162512816	59.24046234	0.922440512
23	58.13598434	58.45253379	**	55.08323548	1.445808317	1.298684299	**	3.143487871	58.28672216	1.352917607
24	58.90444361	60.26617835	**	53.5893468	0.499751896	1.111974266	**	2.730397762	59.58531098	1.091781647
25	74.69217403	95.54192621	**	81.60271942	13.75153443	9.205830481	**	17.37599811	87.96019815	14.77273856
26	69.02972133	100.8977062	**	70.9787504	1.028291035	1.35207425	**	1.483250325	77.52785063	14.62644122
27	**	64.61243367	**	60.83690987	**	3.40514829	**	6.371328949	64.61243367	3.40514829
28	62.40456216	60.37515494	**	60.0812936	0.979269814	2.164770817	**	5.027655754	61.38985855	1.938519331
29	70.23815188	64.60588085	**	60.02921002	2.071300993	3.411859267	**	6.378316174	67.16600405	4.023442753
30	62.89577566	**	**	58.53130791	0.87369174	**	**	4.368422259	62.89577566	0.87369174

Table E.9: Results for Three Controller Comparison Tests

PID Test Number	Speed	No Ramp Mean	SD	Ramp Mean	SD
07/08	10	8.340619196	0.199028451	8.25581582	0.236391872
01/02	20	24.85102855	0.800764332	17.60426149	0.141544568
13/14	40	44.38898158	3.976834124	42.49420808	0.342159105
19/20	80	79.70714915	0.887727681	64.40835653	1.148129848
25/26	120	87.96019815	14.77273856	77.52785063	14.62644122
Type1 Test Number	Speed	No Ramp Mean	SD	Ramp Mean	SD
11/12	10	8.23731686	0.145879871	8.267852451	0.547950039
5/6	20	18.347773	2.102554947	13.59226554	1.308622404
17/18	40	42.19331846	0.659164447	38.14998242	0.439863995
23/24	80	58.28672216	1.352917607	59.58531098	1.091781647
29/30	120	67.16600405	4.023442753	62.89577566	0.87369174
Type2 Test Number	Speed	No Ramp Mean	SD	Ramp Mean	SD
9/10	10	8.104657151	0.187389615	8.008649174	0.150584251
3/4	20	17.55955019	0.145850508	17.5736317	0.199907581
15/16	40	42.26648621	0.37832556	42.4680002	0.61091205
21/22	80	64.56259506	2.057277181	59.24046234	0.922440512
27/28	120	64.61243367	3.40514829	61.38985855	1.938519331

Table E.10: Means and SDs by Controller

Speed	Test Number	Type	No Ramp Mean	SD	Ramp Mean	SD
10	7/8	PID	8.340619196	0.199028451	8.25581582	0.236391872
	9/10	FZC2	8.104657151	0.187389615	8.008649174	0.15058423
	11/12	FZC1	8.23731686	0.145879871	8.267852451	0.547950039
20	1/2	PID	24.85102855	0.800764332	17.60426149	0.141544568
	3/4	FZC2	17.55955019	0.145850508	17.5736317	0.199907581
	5/6	FZC1	18.347773	2.102554947	13.59226554	1.308622404
40	13/14	PID	44.38898158	3.976834124	42.49420808	0.342159105
	15/16	FZC2	42.26648621	0.37832556	42.4680002	0.61091205
	17/18	FZC1	42.19331846	0.659164447	38.14998242	0.439863995
80	19/20	PID	79.70714915	0.887727681	64.40835653	1.148129848
	21/22	FZC2	64.56259506	2.057277181	59.24046234	0.922440512
	23/24	FZC1	58.28672216	1.352917607	59.58531098	1.091781647
120	25/26	PID	87.96019815	14.77273856	77.52785063	14.62644122
	27/28	FZC2	64.61243367	3.40514829	61.38985855	1.938519331
	29/30	FZC1	67.16600405	4.023442753	62.89577566	0.87369174

Table E.11: Means and SDs by Speed Demand

Test No	Speed	R1	R2	R3	R4
101	10				
	Ave Speed	8.585317594	8.072607307	6.921556726	8.840696182
	StdDev	0.074667153	0.366106095	1.204702539	0.088217416
	Global Ave	8.079683787	Global SD	0.945116078	
102	20				
	Ave Speed	20.51609887	19.21661448	14.92100136	17.55955019
	StdDev	0.582299097	0.980353833	2.938341111	0.145850508
	Global Ave	18.39176509	Global SD	2.964568539	
103	40				
	Ave Speed	42.23650176	40.45706717	36.60162405	42.18234271
	StdDev	0.400443901	0.387956588	3.227373276	1.321979739
	Global Ave	40.49096083	Global SD	2.92816849	
104	80				
	Ave Speed	64.84106001	61.37621969	42.77125655	68.18365718
	StdDev	1.012581485	1.393763103	4.317916058	6.157102342
	Global Ave	59.33596555	Global SD	12.31814255	

Table E.12: Results for Threshold = 70, No Ramp

Test No	Speed	R1	R2	R3	R4
106	20				
	Ave Speed	14.32848622	13.41113684	13.14083982	15.75833258
	StdDev	0.4191932	0.937223416	1.738897719	2.616979427
	Global Ave	13.6875686	Global SD	1.136927498	
111	40				
	Ave Speed	23.58714455	23.35223285	22.30465082	17.11296733
	StdDev	0.433415013	0.517357916	0.841275605	0.272113512
	Global Ave	22.94589769	Global SD	0.925299325	
116	80				
	Ave Speed	59.57815735	53.79528414	43.70016309	58.91022201
	StdDev	2.199355248	2.289521588	2.140543481	0.702900748
	Global Ave	52.63153487	Global SD	8.401883737	
121	120				
	Ave Speed	41.01614416	43.27848639	36.77363964	42.80953087
	StdDev	0.469963292	3.999452715	4.127234016	0.855536287
	Global Ave	41.16358166	Global SD	3.684205156	

Table E.13: Results for Threshold = 70, Ramp Applied

TestNo	Speed	Mean	SD
101	10	8.079683787	0.945116078
102	20	18.39176509	2.964568539
103	40	39.88673511	2.92816849
104	80	53.80615828	12.1488483
104	120	No results	

Table E.14: Mean and SD for the Dual Surface Type-2 Controller - Threshold=70

TestNo	Speed	Mean	SD
9	10	8.104657151	0.187389615
3	20	17.55955019	0.145850508
15	40	42.26648621	0.37832556
21	80	63.93189286	2.4026283

Table E.15: Mean and SD for the Average Type-2 Controller

Test No	Speed	R1	R2	R3	R4
107	20				
	Ave Speed	12.60792964	13.32824815	11.57721419	14.71408656
	StdDev	0.334633316	1.204804588	0.887500808	0.353146157
	Global Ave	12.26826075	Global SD	0.917684588	
112	40				
	Ave Speed	21.73632347	22.03279059	19.6637413	23.81926001
	StdDev	1.618282654	1.25790724	1.795594783	0.270637348
	Global Ave	21.401771	Global SD	1.734682749	
117	80				
	Ave Speed	59.0129949	52.261165	45.83153341	57.08068249
	StdDev	1.844675185	3.73126969	3.310387705	2.761092006
	Global Ave	53.26974134	Global SD	6.17790629	
122	120				
	Ave Speed	41.26708147	40.27816115	38.45690846	41.63988009
	StdDev	0.382801947	0.653646173	2.05308105	0.678089167
	Global Ave	40.09152929	Global SD	1.612257215	

Table E.16: Results for Minimum Dual Controller with Ramp

Test No	Speed	R1	R2	R3	R4
108	20				
	Ave Speed	15.62849164	14.18760923	13.73691755	14.71408656
	StdDev	0.115344697	0.294191568	1.640591418	0.353146157
	Global Ave	14.63776852	Global SD	1.190098367	
113	40				
	Ave Speed	24.77895399	24.20131524	22.02739704	23.86573946
	StdDev	0.390105649	0.64380586	1.308458277	1.490862187
	Global Ave	23.61010063	Global SD	1.524461526	
118	80				
	Ave Speed	60.97584475	58.44483289	59.2541979	60.12335664
	StdDev	0.97124424	2.107782734	4.30414986	2.459840082
	Global Ave	59.65084219	Global SD	2.277399255	
123	120				
	Ave Speed	43.06648654	44.89851226	39.55896475	41.19206854
	StdDev	0.68432219	4.850603799	1.644053959	0.305849795
	Global Ave	42.66979233	Global SD	3.733007086	

Table E.17: Results for Maximum Dual Controller with Ramp

109	Speed = 20	Min	Ramp	
	R1	R2	R3	R4/RO
Ave	15.35195936	0	12.80569231	14.67052802
StdDev	0.145470171	0	0.738574385	0.356156517
Global Ave	14.39710922		Global SD	1.347366368

Table E.18: Threshold Results - for Experiment 109

Speed = 60	R1	R2	R3	R4
Controller	Type-1	Type-1	PID	PID
Ave	23.74986351	22.5187092	21.42226295	19.54816069
StdDev	1.563089597	1.964573942	1.16512163	1.448481398
R1,R2 Ave	23.17905997	R3,R4 Ave	20.44447047	
R1,R2 SD	1.615833957	R3,R4 SD	1.60696609	
Controller	Type-2	Type-2	Type-1	Type-1
Ave	23.13250688	22.95321163	21.07140634	19.41651569
StdDev	0.739488158	1.327334036	0.720703115	0.898061765
R1,R2 Ave	23.04285926	R3,R4 Ave	20.24396102	
R1,R2 SD	1.046392334	R3,R4 SD	1.161424256	
Controller	Type-1	Type-1	Type-2	Type-2
Ave	23.21436214	23.46693344	19.95352904	18.70332725
StdDev	0.614807089	0.93325015	2.020850185	1.134704825
R1,R2 Ave	23.34064779	R3,R4 Ave	19.30342411	
R1,R2 SD	0.784906949	R3,R4 SD	1.709361463	

Table E.19: Robot mean speed and standard deviation for each controller type

Controller	Type-1	Type-2	PID
Average	21.68912369	21.1856437	21.9129298
SD	1.869151553	2.205642317	1.820397421

Table E.20: Average Speed for Controller Types