2002

INTELLIGENT FAULT TOLERANT CONTROL SCHEMES FOR AUTONOMOUS UNDERWATER VEHICLES

PEARSON, ANDREW RAYMOND

Intelligent Fault Tolerant Control Schemes for Autonomous Underwater Vehicles

Andrew Raymond Pearson

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

Doctor of Philosophy

Department of Mechanical and
Marine Engineering

Faculty of Technology

February 2002

UNIVERSITY OF PLYMOUTH
Faculty of Technology
Department of Mechanical and Marine Engineering

Intelligent Fault Tolerant Control Schemes for Autonomous Underwater Vehicles

Andrew Raymond Pearson

Abstract

The area of autonomous underwater vehicles (AUVs) is an increasingly important area of research, with AUVs being capable of handling a far wider range of missions than either an inhabited underwater vehicle or a remotely operated vehicle (ROV). One of the major drawbacks of such vehicles is the inability of their control systems to handle faults occurring within the vehicle during a mission. This study aims to develop enhancements to an existing control system in order to increase its fault tolerance to both sensor and actuator faults.

Faults occurring within the sensors for both the yaw and roll channels of the AUV are considered. Novel fuzzy inference systems (FISs) are developed and tuned using both the adaptive neuro-fuzzy inference system (ANFIS) and simulated annealing tuning methods. These FISs allow the AUV to continue operating after a fault has occurred within the sensors.

Faults occurring within the actuators which control the canards of the AUV and hence the yaw channel are also examined. Actuator recovery FISs capable of handling faults occurring within the actuators are developed using both the simulated annealing and tabu search methods of tuning FISs. The fault tolerance of the AUV is then further enhanced by the development of an error estimation FIS that is used to replace an error sensor.

It concludes that the novel FISs designed and developed within the thesis provide an improved performance to both sensor and actuator faults in comparison to benchmark control systems. Therefore having these FISs embedded within the overall control scheme ensure the AUV is fault tolerant to a range of selected failures.

Acknowledgments

The work detailed in this thesis would not have been possible without the encouragement of my supervisory team. I would sincerely like to thank Professor Robert Sutton (Director of Studies) for his technical input, advice and patience which made this Ph.D. programme a smooth and enjoyable one. I would also like to express my thanks to Professor Roland Burns for his advice and technical input and Paul Robinson for his advice. They are both acknowledged for their careful reading of this thesis.

Thanks are extended to the Defence Evaluation and Research Agency, Winfrith for the provision of their underwater vehicle model. The financial support of the Engineering and Physical Sciences Research Council is also acknowledged.

I am extremely grateful to my family without whom this thesis would not have been possible. Their financial support has been gratefully appreciated throughout my academic career.

Finally I would like to thank my friends for their support, advice, encouragement and meaningful discussions throughout this study, especially Dawn.

Declaration

- The work presented within this thesis is original; no part of this work has been used for any other award or degree at anytime.

- During the candidature the author has not been registered for any other award at any other institution.

Andrew Raymond Pearson

15/2/2002

PhD Thesis Chapter Headings

Chapter 1 Introduction 1

1.1. Introduction 1
1.2. Objectives of the Thesis 1
1.3. Contributions of the Thesis 2
1.4. Publications 3
1.5. Outline of the Thesis 4

Chapter 2 A Review of Uninhabited Underwater Vehicles (UUVs) Fault Tolerant Control 6

2.1. Introduction 6
2.2. Fault Tolerant Control 8
2.2.1. Fault Tolerance 8
2.2.2. Reconfigurable Control 10
2.2.3. Restructurable Control 10
2.3. Fault Tolerant Control Systems for Non-UUVs 11
2.3.1. Fault Tolerant Control 11
2.3.2. Reconfigurable Control 13
2.3.3. Restructurable Control 18
2.4. Uninhabited Underwater Vehicles 20
2.4.1. Fault Tolerant Control 20
2.4.2. Reconfigurable Control 24
2.5. Concluding Remarks 31

Chapter 3 Uninhabited Underwater Vehicle Dynamics 32

3.1. Introduction 32
3.2. UUV Matlab Model Dynamics 32
3.2.1. Equations of Motion 33
3.2.2. PD Controller 36

3.2.3. Adaptive Neuro-Fuzzy Inference System (ANFIS) Controller 38
3.3. Identification of AUV Dynamics 43
3.3.1. Training and Testing Data 44
3.3.2. Elman Networks 47
3.3.3. Adaptive Neuro-Fuzzy Inference System (ANFIS) 48
3.3.4. Linear Models 48
3.3.5. Results and Comparisons 49
3.3.6. Summary of Models 55
3.3.7. Model Selected 60
3.3.8. Closed Loop Responses of Models 60
3.3.9. Roll Model 61
3.4. Fault Set-up 63
3.4.1. Sensor Faults 63
3.4.2. Actuator Faults 65
3.5. Fuzzy Inference System (FIS) Tuning Methods 66
3.5.1. ANFIS 66
3.5.2. Simulated Annealing 66
3.5.3. Tabu 69
3.6. Conclusions 71

Chapter 4 The Kalman Filter Approach to Fault Tolerant Control 72

4.1. Introduction 72
4.2. Kalman Filter Theory 72
4.3. Kalman Responses to Sensor Faults 76
4.3.1. Yaw Sensor Failures 78
4.3.2. Yaw Rate Sensor Failures 84
4.3.3. Roll Sensor Failures 88
4.3.4. Roll Rate Sensor Failures 93
4.4. Kalman Responses to Actuator Faults 98
4.4.1. Yaw Step Inputs of 10 Degrees 99
4.4.2. Yaw Step Inputs of 20 Degrees 104
4.4.3. Yaw Step Inputs of 30 Degrees 109

4.5. Conclusions 114

Chapter 5 Sensor Recovery System 117

5.1. Introduction 117
5.2. Fault Tolerant System 117
5.3. Q Statistic 120
5.4. Fuzzy Tuning 121
5.5. Results 137
5.5.1. Yaw Sensor FISs 137
5.5.2. Yaw Rate Sensor FISs 146
5.5.3. Roll Sensor FISs 154
5.5.4. Roll Rate Sensor FISs 162
5.5.5. Yaw Sensor Failures 170
5.5.6. Yaw Rate Sensor Failures 176
5.5.7. Roll Sensor Failures 183
5.5.8. Roll Rate Sensor Failures 188
5.6. Discussion and Concluding Remarks 194
5.6.1. Yaw Sensor 194
5.5.2. Yaw Rate Sensor 195
5.5.3. Roll Sensor 196
5.5.4. Roll Rate Sensor 197

Chapter 6 Actuator Recovery System With Error Sensor 198

6.1. Introduction 198
6.2. Fault Recovery System 198
6.3. Fuzzy Tuning 201
6.3.1. Identity FIS 201
6.3.2. Heuristic FIS 204
6.3.3. ANFIS 207
6.3.4. Simulated Annealing 208
6.3.5. Tabu Search 208

6.4. Results 209
6.4.1. Simulated Annealing Tuned FIS 209
6.4.2. Tabu Search Tuned FIS 214
6.4.3. Yaw Step Inputs of 10 Degrees 218
6.4.4. Yaw Step Inputs of 20 Degrees 227
6.4.5. Yaw Step Inputs of 30 Degrees 237
6.5. Conclusions 246

Chapter 7 Actuator Recovery System Without Error Sensor 248

7.1. Introduction 248
7.2. Replacing The Error Sensor 248
7.3. Fuzzy Tuning 252
7.3.1. ANFIS 255
7.4. Results 256
7.4.1. The Three Rules FIS 256
7.4.2. The Five Rules FIS 258
7.4.3. The Seven Rules FIS 260
7.4.4. The Nine Rules FIS 262
7.4.5. Yaw Step Inputs of 10 Degrees 264
7.4.6. Yaw Step Inputs of 20 Degrees 269
7.4.7. Yaw Step Inputs of 30 Degrees 275
7.5. Conclusions 281

Chapter 8 Concluding Remarks 283

8.1. Conclusions 283
8.2. Research Objectives 285
8.3. Recommendations for Future Research 286

References 288

Appendices

A	Publications	294
B	Open Loop Model Results	325
C	Elman ANN Yaw Channel Model	326
D	Yaw Channel Sensor Failure Results	332
E	Roll Channel Sensor Failure Results	342
F	Yaw Actuator Failure Results	352
G	Yaw Channel Sensor Recovery FISs Results	357
H	Roll Channel Sensor Recovery FISs Results	367
I	Actuator Recovery FISs Canard Responses	377
J	Actuator Recovery Without Error Sensor Results	383
K	Sensorless Actuator Recovery FISs Canard Responses	389

List of Figures

Figure No.		Page No.
2.1	A Reconfigurable Control System	9
2.2	Integrated-System Components for Advanced Plant Control [After Eryurek and Upadhyaya (1995)]	12
2.3	SWATH Control System Set-up [After Kim et *al* (1997)]	16
2.4	The Feedforward Multilayer Perceptron	17
2.5	Error Backpropagation [After Waldock (1996)]	17
2.6	Block Diagram of Adaptive Controller [After Rauch (1995)]	19
2.7	AUV Control Scheme [After Katebi and Grimble (1999)]	25
3.1	Diagram of AUV Control Surfaces	37
3.2	The ANFIS Architecture	39
3.3	Open Loop AUV Set-up	45
3.4	Random Input Used for Training	45
3.5	AUV Yaw Rate Response to Training Data	45
3.6	AUV Yaw Response to Training Data	45
3.7	Step Input Signals for Testing Models	46
3.8	Corresponding Yaw Rates used for Testing Models	47
3.9	Open Loop Set-up for Testing of Models	49
3.10	Elman Model Yaw Response to Testing Path	51
3.11	Elman Model Yaw Rate Response to Testing Path	51
3.12	Yaw Error for Elman Model on Testing Path	51
3.13	Yaw Rate Error for Elman Model on Testing Path	51
3.14	Fuzzy Model Yaw Response to Testing Path	52
3.15	Fuzzy Model Yaw Rate Response to Testing Path	52
3.16	Yaw Error for Fuzzy Model on Testing Path	52
3.17	Yaw Rate Error for Fuzzy Model on Testing Path	52
3.18	Linear Model Yaw Response to Testing Path	53
3.19	Linear Model Yaw Rate Response to Testing Path	53
3.20	Yaw Error for Linear Model on Testing Path	54
3.21	Yaw Rate Error for Linear Model on Testing Path	54

3.22	DERA Model Yaw Response to Testing Path	54
3.23	DERA Model Yaw Rate Response to Testing Path	54
3.24	Yaw Error for DERA Model on Testing Path	55
3.25	Yaw Rate Error for DERA Model on Testing Path	55
3.26	Yaw Responses for a Closed Loop 60 Degrees Step Input	61
3.27	Roll Responses for a Closed Loop Initial Roll Angle of 5 Degrees	62
3.28	Gain on Sensor Feedback for Yaw Channel	64
3.29	Flip Switch Between Total and Zero Failure on Yaw Sensor	64
3.30	Noise Added to Yaw Sensor Feedback	65
3.31	Actuator Simulator Within Matlab AUV Model	65
3.32	The Boltzman Probability Distribution	67
4.1	Kalman Filter Flow Chart	75
4.2	The Closed Loop Kalman Filter Control System	76
4.3	Yaw Responses of Normal and Kalman Filter Controllers	77
4.4	Yaw Rate Responses of Normal and Kalman Filters Controllers	77
4.5	Yaw Angle Responses to Total Yaw Sensor Failure Over a Step Demand of 50 Degrees	79
4.6(a)	Kalman Filter Gain Value for Matrix Position (1,1)	80
4.6(b)	Kalman Filter Gain Value for Matrix Position (1,2)	80
4.6(c)	Kalman Filter Gain Value for Matrix Position (2,1)	80
4.6(d)	Kalman Filter Gain Value for Matrix Position (2,2)	80
4.7	Yaw Angle Responses to Intermittent Signal Failure Over a Step Demand of 50 Degrees	82
4.8	Yaw Angle Responses to SNR Faults Over a Step Demand of 50 Degrees for The ANFIS Controller	83
4.9	Yaw Angle Responses to SNR Faults Over a Step Demand of 50 Degrees for The Kalman Filter Enhanced Controller	84
4.10	Yaw Angle Responses to Total Yaw Rate Signal Failure over a Step Demand of 90 Degrees	85
4.11	Yaw Angle Responses to Intermittent Yaw Rate Signal Failure of 10% Over a Step Demand of 90 Degrees	86
4.12	Yaw Angle Responses to SNR Faults Over a Step Demand	

	of 90 Degrees for The ANFIS Controller	87
4.13	Yaw Angle Responses to SNR Faults Over a Step Demand of 90 Degrees for The Kalman Filter Enhanced Controller	88
4.14	Roll Angle Responses to Percentage Signal Loss Tests Roll Signal Failure Over an Initial Angle of 25 Degrees for The ANFIS Controller	90
4.15	Roll Angle Responses to Percentage Signal Loss Tests Roll Signal Failure Over an Initial Angle of 25 Degrees for The Kalman Filter Controller	90
4.16	Roll Angle Responses to Intermittent Signal Failure for an Initial Angle of 25 Degrees	91
4.17	Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees for The ANFIS Controller	92
4.18	Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees for The Kalman Filter Controller	93
4.19	Roll Angle Responses to Percentage Signal Loss Tests Roll Rate Signal Failure Over an Initial Angle of 25 Degrees for The ANFIS Controller	94
4.20	Roll Angle Responses to Percentage Signal Loss Tests Roll Rate Signal Failure Over an Initial Angle of 25 Degrees for The Kalman Filter Controller	95
4.21	Roll Angle Responses to Intermittent Roll Rate Signal Failure Over an Initial Angle of 25 Degrees	96
4.22	Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees for The ANFIS Controller	97
4.23	Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees for The Kalman Filter Controller	98
4.24	PD Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 10 Degrees Step Input	100
4.25	ANFIS Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 10 Degrees Step Input	101
4.26	Kalman Filter Enhanced Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 10 Degrees Step Input	102

4.27	Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 10 Degrees Step Input Demand	103
4.28	PD Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 20 Degrees Step Input	105
4.29	ANFIS Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 20 Degrees Step Input	106
4.30	Kalman Filter Enhanced Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 20 Degrees Step Input	107
4.31	Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 20 Degrees Step Input Demand	108
4.32	PD Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 30 Degrees Step Input	110
4.33	ANFIS Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 30 Degrees Step Input	111
4.34	Kalman Filter Enhanced Controller Yaw Angle Responses for Both Actuator Percentage LOE Faults on 30 Degrees Step Input	112
4.35	Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 30 Degrees Step Input Demand	113
5.1	The Sensor Fault Tolerant Control System	118
5.2	The Input Output Structure of The Sensor Recovery FIS	119
5.3	Heuristic Membership Functions for Q Input	123
5.4	Heuristic Membership Functions for Sensor Input	124
5.5	Heuristic Membership Functions for Linear Sensor Input	125
5.6	AUV Yaw Response to Training Path	126
5.7	Complete Yaw Q Statistic Input Data Set	127
5.8	Complete Yaw Sensor Input Data Set	127
5.9	Complete Linear Model Yaw Input Data Set	128
5.10	Complete Yaw Sensor Output Data Set	128
5.11	Complete Yaw Rate Q Statistic Input Data Set	129
5.12	Complete Yaw Rate Sensor Input Data Set	130
5.13	Complete Linear Model Yaw Rate Input Data Set	130
5.14	Complete Yaw Rate Sensor Output Data Set	131
5.15	AUV Roll Response to Training Path	132

5.16	Complete Roll Q Statistic Input Data Set	133
5.17	Complete Roll Sensor Input Data Set	133
5.18	Complete Linear Model Roll Input Data Set	134
5.19	Complete Roll Sensor Output Data Set	134
5.20	Complete Roll Rate Q Statistic Input Data Set	135
5.21	Complete Roll Rate Sensor Input Data Set	135
5.22	Complete Linear Model Roll Rate Input Data Set	136
5.23	Complete Roll Rate Sensor Output Data Set	136
5.24	ANFIS Tuned Membership Functions for Q Statistic Input	138
5.25	ANFIS Tuned Membership Functions for Yaw Sensor Input	139
5.26	ANFIS Tuned Membership Functions for Linear Yaw Input	139
5.27	Simulated Annealing Tuned Membership Functions for Q Statistic Input	142
5.28	Simulated Annealing Tuned Membership Functions for Yaw Sensor Input	143
5.29	Simulated Annealing Tuned Membership Functions for Linear Yaw Input	143
5.30	ANFIS Tuned Membership Functions for Q Statistic Input	146
5.31	ANFIS Tuned Membership Functions for Yaw Rate Sensor Input	147
5.32	ANFIS Tuned Membership Functions for Linear Yaw Rate Input	147
5.33	Simulated Annealing Tuned Membership Functions for Q Statistic Input	150
5.34	Simulated Annealing Tuned Membership Functions for Yaw Rate Sensor Input	151
5.35	Simulated Annealing Tuned Membership Functions for Linear Yaw Rate Input	151
5.36	ANFIS Tuned Membership Functions for Q Statistic Input	154
5.37	ANFIS Tuned Membership Functions for Roll Sensor Input	155
5.38	ANFIS Tuned Membership Functions for Linear Roll Input	155
5.39	Simulated Annealing Tuned Membership Functions for Q Statistic Input	158
5.40	Simulated Annealing Tuned Membership Functions for Roll Sensor Input	159

5.41	Simulated Annealing Tuned Membership Functions for Linear Roll Input	159
5.42	ANFIS Tuned Membership Functions for Q Statistic Input	162
5.43	ANFIS Tuned Membership Functions for Roll Rate Sensor Input	163
5.44	ANFIS Tuned Membership Functions for Linear Roll Rate Input	163
5.45	Simulated Annealing Tuned Membership Functions for Q Statistic Input	166
5.46	Simulated Annealing Tuned Membership Functions for Roll Rate Sensor Input	167
5.47	Simulated Annealing Tuned Membership Functions for Linear Roll Rate Input	167
5.48	The Yaw Responses to a 75% Yaw Signal Loss for a 10 Degrees Step Yaw Demand	172
5.49	The Yaw Responses to a 75% Yaw Signal Loss for a 50 Degrees Step Yaw Demand	174
5.50	The Yaw Responses to a 75% Yaw Signal Loss for a 90 Degrees Step Yaw Demand	176
5.51	The Yaw Responses to a 75% Yaw Rate Signal Loss for a 10 Degrees Step Yaw Demand	178
5.52	The Yaw Responses to a 75% Yaw Rate Signal Loss for a 50 Degrees Step Yaw Demand	180
5.53	The Yaw Responses to a 75% Yaw Rate Signal Loss for a 90 Degrees Step Yaw Demand	182
5.54	The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 5 Degrees	184
5.55	The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 15 Degrees	186
5.56	The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 25 Degrees	188
5.57	The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 5 Degrees	190
5.58	The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 15 Degrees	192

5.59	The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 25 Degrees	194
6.1	The Actuator Fault Tolerant System	199
6.2	Basic Input Output Structure	200
6.3	Membership Functions for Control Input	201
6.4	Membership Functions for Demand Input	202
6.5	Membership Functions for Error Input	203
6.6	AUV Tuning Path for Actuator Faults	207
6.7	Membership Functions for Control Input of SA FIS	210
6.8	Membership Functions for Demand Input of SA FIS	211
6.9	Membership Functions for Error Input of SA FIS	211
6.10	Values for The Fuzzy Singletons Before and After Tuning by The Simulated Annealing Algorithm	213
6.11	Membership Functions for Control Input of TABU FIS	215
6.12	Membership Functions for Demand Input of TABU FIS	215
6.13	Membership Functions for Error Input of TABU FIS	216
6.14	Values for The Fuzzy Singletons Before and After Tuning by The Tabu Search Algorithm	218
6.15	Simulated Annealing Results for The Saturation Block LOEs for a 10 degrees Demand	220
6.16	Tabu Search Results for The Saturation Block LOEs for a 10 Degrees Demanded Yaw Angle	220
6.17	Simulated Annealing Results for The Rate Limiter Block LOEs for a 10 Degrees Demanded Yaw Angle	222
6.18	Tabu Search Results for The Rate Limiter Block LOEs for a 10 Degrees Demanded Yaw Angle	223
6.19	Simulated Annealing Results for Both Block LOEs for a 10 Degrees Demanded Yaw Angle	224
6.20	Tabu Search Results for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle	225
6.21	Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle	226

6.22 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle 226
6.23 Simulated Annealing Results for The Saturation Block LOEs for a 20 Degrees Demanded Yaw Angle 229
6.24 Tabu Search Results for The Saturation Block LOEs for a 20 Degrees Demanded Yaw Angle 229
6.25 Simulated Annealing Results for The Rate Limiter Block LOEs for a 20 Degrees Demanded Yaw Angle 231
6.26 Tabu Search Results for The Rate Limiter Block LOEs for a 20 Degrees Demanded Yaw Angle 232
6.27 Simulated Annealing Results for Both Block LOEs for a 20 Degrees Demanded Yaw Angle 233
6.28 Tabu Search Results for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle 234
6.29 Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle 236
6.30 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle 236
6.31 Simulated Annealing Results for The Saturation Block LOEs for a 30 Degrees Demanded Yaw Angle 239
6.32 Tabu Search Results for The Saturation Block LOEs for a 30 Degrees Demanded Yaw Angle 239
6.33 Simulated Annealing Results for The Rate Limiter Block LOEs for a 30 Degrees Demanded Yaw Angle 240
6.34 Tabu Search Results for The Rate Limiter Block LOEs for a 30 Degrees Demanded Yaw Angle 241
6.35 Simulated Annealing Results for Both Block LOEs for a 30 Degrees Demanded Yaw Angle 242
6.36 Tabu Search Results for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle 243
6.37 Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle 245
6.38 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle 245

7.1	The Actuator Fault Tolerant System	249
7.2	Heuristic Error Estimation FIS Input Membership Functions	250
7.3	Heuristic Error Estimation FIS Output Line	251
7.4	Error Estimation FIS Random Input Signal	253
7.5	AUV Yaw Rate Response	253
7.6	The Modular Difference Between The Linear Model and AUV Yaw Rates	254
7.7	The Modular Error in The Actuator Position	255
7.8	The Three Rules FIS Input Membership Functions	256
7.9	The Three Rules FIS Output Line	257
7.10	The Five Rules FIS Input Membership Functions	258
7.11	The Five Rules FIS Output Line	259
7.12	The Seven Rules FIS Input Membership Functions	260
7.13	The Seven Rules FIS Output Line	261
7.14	The Nine Rules FIS Input Membership Functions	262
7.15	The Nine Rules FIS Output Line	263
7.16	The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 10 degrees	266
7.17	The Error Traces for The 50% LOE in Both Blocks for a Yaw Step Input of 10 degrees	267
7.18	Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 10 degrees	268
7.19	Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 10 degrees	268
7.20	The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 20 degrees	272
7.21	The Error Traces for The 50% LOE in Both Blocks 72 for a Yaw Step Input of 20 degrees	272
7.22	Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 20 degrees	273
7.23	Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 20 degrees	274
7.24	The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 30 degrees	278

7.25 The Error Traces for The 50% LOE in Both Blocks for a Yaw Step Input of 30 degrees 278
7.26 Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 30 degrees 279
7.27 Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 30 degrees 280

List of Tables

Table No.		Page No.
3.1	The Simulated Annealing Algorithm	68
3.2	The Tabu Search Algorithm	70
4.1	The Rise Times For A Yaw Step Input of 10 Degrees	104
4.2	The Rise Times For A Yaw Step Input of 20 Degrees	109
4.3	The Rise Times For A Yaw Step Input of 30 Degrees	114
5.1	The 10 Degrees Step Input RMSEs for Yaw Sensor Faults	170
5.2	The 10 Degrees Step Input Percentage Increases for Yaw Sensor Faults	171
5.3	The 50 Degrees Step Input RMSEs for Yaw Sensor Faults	173
5.4	The 50 Degrees Step Input Percentage Increases for Yaw Sensor Faults	173
5.5	The 90 Degrees Step Input RMSEs for Yaw Sensor Faults	175
5.6	The 90 Degrees Step Input Percentage Increases for Yaw Sensor Faults	175
5.7	The 10 Degrees Step Input RMSEs for Yaw Rate Sensor Faults	177
5.8	The 10 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults	177
5.9	The 50 Degrees Step Input RMSEs for Yaw Rate Sensor Faults	179
5.10	The 50 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults	179
5.11	The 90 Degrees Step Input RMSEs for Yaw Rate Sensor Faults	181
5.12	The 90 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults	181
5.13	The Initial Angle of 5 Degrees RMSEs for Roll Sensor Faults	183
5.14	The Initial Angle of 5 Degrees Percentage Increases for Roll Sensor Faults	184
5.15	The Initial Angle of 15 Degrees RMSEs for Roll Sensor Faults	185

5.16	The Initial Angle of 15 Degrees Percentage Increases for Roll Sensor Faults	185
5.17	The Initial Angle of 25 Degrees RMSEs for Roll Sensor Faults	187
5.18	The Initial Angle of 25 Degrees Percentage Increases for Roll Sensor Faults	187
5.19	The Initial Angle of 5 Degrees RMSEs for Roll Rate Sensor Faults	189
5.20	The Initial Angle of 5 Degrees Percentage Increases for Roll Rate Sensor Faults	189
5.21	The Initial Angle of 15 Degrees RMSEs for Roll Rate Sensor Faults	191
5.22	The Initial Angle of 15 Degrees Percentage Increases for Roll Rate Sensor Faults	191
5.23	The Initial Angle of 25 Degrees RMSEs for Roll Rate Sensor Faults	193
5.24	The Initial Angle of 25 Degrees Percentage Increases for Roll Rate Sensor Faults	193
6.1	The 10 Degrees Step Input RMSEs	219
6.2	The 10 Degrees Step Input Rise Times	219
6.3	The 10 Degrees Step Input Percentage Improvements	227
6.4	The 20 Degrees Step Input RMSEs	227
6.5	The 20 Degrees Step Input Rise Times	228
6.6	The 20 Degrees Step Input Percentage Improvements	237
6.7	The 30 Degrees Step Input RMSEs	238
6.8	The 30 Degrees Step Input Rise Times	238
6.9	The 30 Degrees Step Input Percentage Improvements	246
7.1	The 10 Degrees Step Input RMSEs	265
7.2	The 10 Degrees Step Input Rise Times	265
7.3	The 10 Degrees Step Input Percentage Decrease	269
7.4	The 20 Degrees Step Input RMSEs	270
7.5	The 20 Degrees Step Input Rise Times	271
7.6	The 20 Degrees Step Input Percentage Decrease	275

7.7 The 30 Degrees Step Input RMSEs 276
7.8 The 30 Degrees Step Input Rise Times 277
7.9 The 30 Degrees Step Input Percentage Decrease 281

Nomenclature

(In alphabetical order)

Symbol	Definition
A	Constant which governs the rate of decay
$\mathbf{A}$	Linear model state vector's coefficients matrix
$\mathbf{A}_{dm}$	Linear model state vector's coefficients matrix
$a_i \ldots c_i$	The ith parameter set
$A_i \ldots P_i$	Premise membership functions
$\mathbf{A}_{kl}$,	State space matrix of the linear model
$\mathbf{A}_m$	Reference model coefficients
$\mathbf{A}_p$	Plant coefficients
AUV_i	The ith information received from the vehicle
B	Vehicle buoyancy in air
$\mathbf{B}$	Linear model control vector's coefficients matrix
$\mathbf{B}_{dm}$	Linear model control vector's coefficients matrix
$\mathbf{B}_{kl}$	The control matrix of the linear model
BM	Fuzzy set representing big medium
$\mathbf{B}_m$	The reference model coefficients
$\mathbf{B}_p$	Plant coefficients
$\mathbf{B}_p^+$	The pseudo-inverse of $\mathbf{B}_p$
BS	Fuzzy set representing big small
c	Control signal
$\mathbf{C}$	Linear model state vector's coefficients matrix
$\mathbf{ck}(t)$	The function which represents the hydrostatic and hydrodynamic forces and moments acting on the vehicle
$\mathbf{C}_{kl}$	State space matrix of the linear model
$\mathbf{D}$	Linear model control vector's coefficients matrix

Dif	Fuzzy set representing the difference between the AUV's and the linear model's yaw rates
E	RMSE error value
$E(\theta)$	The energy associated with the state θ
$\mathrm{E}\langle\ \rangle$	The expected value
$\mathbf{E}_{dm}$	Matrix describing vehicle dynamics
Error	The error estimate produced by the error estimation FIS
fk	Mappings of the vehicle motions into forces
f_i	The ith fuzzy rule output
Fox	The force in the x direction
Foy	The force in the y direction
Foz	The force in z direction
fR	State transition function
$\mathbf{G}$	Linear model disturbance vector's coefficients matrix
$G_{c,\psi_e}(s)$	PD controller
gk	Mapping of the vehicle motions into forces
G_K	Hydrostatic moment in roll angle
G_M	Hydrostatic moment in pitch angle
G_N	Hydrostatic moment in yaw angle
gR	The function of $\mathbf{x}^*$ and $\mathbf{q}^*$ used to calculate $\mathbf{uR}$
G_X	Hydrostatic force in x-direction
G_Y	Hydrostatic force in y-direction
G_Z	Hydrostatic force in z-direction
hk	The function relating to the kinematic relations
$\mathbf{H}_{kl}$	State space matrix of the linear model
$HorTh(\mathrm{x})$	The force of the horizontal thruster (where x = 1,2,3,4)
$\mathbf{I}$	The identity matrix

$IDEAL_i$	Information recorded in a fault free situation
I_x	Moment of inertia about body fixed x-direction
I_y	Moment of inertia about body fixed y-direction
I_z	Moment of inertia about body fixed z-direction
K'_{uu}... $K'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in roll angle
k	A constant used by simulated annealing method
K	Hydrodynamic moment of force in roll angle
K	Matrix obtained by solving a time-invariant Riccati equation
KG	Kalman gain matrix
KT	Function for calculating the external moment in the pitch angular direction
kt	The value at the ktth time
K_{wave}	Wave moment in roll angle
l	Length of vehicle
L	Vector form the linear model
L	Fuzzy set representing large
LS	The linear model input to the sensor recovery FIS
M'_{uu}... $M'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in pitch angle
m	Vehicle mass in air
M	Hydrodynamic moment of force in pitch angle
M	Fuzzy set representing medium
MB	Fuzzy set representing medium big
$\mathbf{MK}(t)$	The mass matrix
MOD_i	The ith information received from the model
Mox	The moment in the x direction
Moy	The moment in the y direction
Moz	The moment in the z direction
MS	Fuzzy set representing medium small
MT	Function for calculating the external moment in the roll angular direction

M_{wave}	Wave moment pitch angle
N'_{uu}... $N'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in yaw angle
N	Hydrodynamic moment of force in yaw angle
N	Fuzzy set representing negative
n_e	Training epoch
Neg	Fuzzy set representing negative
nrmse	Number of points measured
nt	The propulsion revolutions per minute
NT	Function for calculating the external moment in the yaw angular direction
N_{wave}	Wave moment in yaw angle
$\mathbf{O}_{k,I}$	The output of the *i*th node within layer *k*
p	The pitch angular velocity
P	Error covariance matrix
P	Fuzzy set representing positive
p_i... v_i	Linear coefficients within the *i*th fuzzy consequent function
Pos	Fuzzy set representing positive
q*	The estimated vector parameters
q	The roll angular velocity
Q	The Q value input to the sensor recovery FIS
QT	Weighting matrix
r	The yaw angular velocity
R	Initial error covariance matrix
RMSE	Error value
RT	Weighting matrix
Ry	The distance from ODIN's centre to the centre of the vertical thruster
Rz	The distance from ODIN's centre to the centre of the horizontal thruster

S	Vector from the sensors
S	Fuzzy set representing small
SB	Fuzzy set representing small big
SE	The signal estimate produced by the sensor recovery FIS
SEN	The sensor signal input to the sensor recovery FIS
SM	Fuzzy set representing small medium
ss	The constant 0.707
t	Time
T	A temperature parameter which decays training
T	Forces arising from thrusters
T_0	Initial temperature
u	The velocity component along the x-direction
u	The control input vector
$\mathbf{u}_c$	The constant ocean current
uk	The control input vector
ukl(t)	The vector of the actuator inputs for the linear model
$\mathbf{u}_m$	The input to the reference model
$\mathbf{u}_p$	The input to the plant
uR	Control vector
ut	Control vector
v	The velocity component along the y-direction
VerTh(x)	The force of the vertical thruster (where x = 1,2,3,4)
w	The velocity component along the z-direction
W	Vehicle weight in air
w_i	Weight of the ith fuzzy rule
$\overline{w}_i$	Normalized weight of the ith fuzzy rule
x*	The estimated state
X'_{uu}... $X'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in x-direction

$\dot{\mathbf{x}}$	New state vector
X_0	The x component of the fixed reference system of the vehicles position
$\mathbf{x}$	Current state vector
X	hydrodynamic forces acting in the x-direction
$\mathbf{X}$	Best estimate output
x_B	X co-ordinate for vehicles centre of buoyancy
$\mathbf{X}_{Current}$	Vector of parameters defining the current FIS
x_G	X co-ordinate for vehicles centre of gravity
$\mathbf{xG}_p$	The plant state
$\mathbf{xk}$	The vector of surge, sway, heave, roll, pitch and yaw velocities
$\mathbf{xkl}(t)$	The vector representing the positions and attitude and the translational and rotational with respect to the body axis co-ordinates of the linear model
$\mathbf{x}_m$	Vector describing the state of vehicle model
$\mathbf{xR}$	State vector
$\mathbf{xt}$	Augmented state vector
$\hat{\mathbf{x}}\mathbf{t}(t)$	Optimal estimate of the state vector $\mathbf{xtk}(t)$
XT	Function for calculating the external force in the x direction
$\mathbf{xtk}(t)$	State vector supplied by a Kalman filter
$\mathbf{X}_{Trial}$	Vector of parameters defining the altered FIS
X_{wave}	Wave force in x-direction
$Y'_{uu} \ldots Y'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in y-direction
Y_0	The y component of the fixed reference system of the vehicles position
$\mathbf{y}$	Output vector
Y	hydrodynamic forces acting in the y-direction
y_B	Y co-ordinate for vehicles centre of buoyancy
y_G	Y co-ordinate for vehicles centre of gravity
$\mathbf{ykl}(t)$	The variables to be controlled in the linear model
$\mathbf{y}_m$	The state space linear models output
YT	Function for calculating the external force in the y direction
Y_{wave}	Wave force in y-direction

$Z'_{uu}\ldots Z'_{r\|r\|}$	Velocity dependent hydrodynamic coefficients in z-direction
Z_0	The z component of the fixed reference system of the vehicles position
Z	hydrodynamic forces acting in the z-direction
Z	Fuzzy set representing zero
z_B	Z co-ordinate for vehicles centre of buoyancy
z_G	Z co-ordinate for vehicles centre of gravity
zk	The vector of x, y and z positions and roll, pitch and yaw angles
$\mathbf{zkl}(t)$	The measured variables of the linear model
ZT	Function for calculating the external force in the z direction
Z_{wave}	Wave force in z-direction
α	Control signal
α_1	Output form the ANFIS controller
α_2	Output form the ANFIS controller delayed by one second
α_3	Output from the ANFIS controller delayed by two seconds
β	Control signal multiplier
δ	Desired canard angle
$\delta_{bp}\ldots\delta_{sp}$	Forces arising from control surfaces
δ_{br}	The actuator input controlling the yaw motion
δ_{cr}	The actuator input controlling the sway motion
δ_p	Desired plane angle
δt_1	Upper rudder angle
δt_2	Lower rudder angle
δt_3	Port elevator
δt_4	Starboard elevator
δ_{sr}	The actuator input controlling the roll motion
Δ	A constant used by the tabu search method
Φ	Transition matrix
ζ_f	Fuzzy logic models yaw rate estimate
η	Learning rate during gradient transition
He	Error in the actuator position

θ	The pitch angle
θ	Matrix of parameters
μ_{Ai}	The ith premise membership function
ξ	Zero-mean white noise sequence
Ξ	Vector of previous changes made to the parameters in the last iteration
ς	Zero-mean white noise sequence
ρ	Density of water
ϕ	The roll angle
ϕ_e	Roll angle error
ψ	The yaw angle
ψ_e	Yaw angle error
ψ_ε	Demanded yaw angle
ω	Noise
ωt	Vector to model external disturbances
Ω	Vector of random changes made to the parameters

List of Acronyms

Acronym	Definition
ANFIS	- Adaptive Neuro-Fuzzy Inference System
ANN	- Artificial Neural Network
AUV	- Autonomous Underwater Vehicle
DERA	- Defence Evaluation and Research Agency
EVS	- Error Vector Suppression
FIS	- Fuzzy Inference System
IFCS	- Intelligent Fault Control System
LOE	- Loss of Effectiveness
LQG	- Linear Quadratic Gaussian
ODIN	- Omi-Directional Intelligent Navigator
PC	- Predictive Controller
PD	- Proportion Derivative
PID	- Proportion Integral Derivative
RCS	- Reconfigurable Control System
ROV	- Remotely Operated Vehicle
RMSE	- Root Mean Squared Error
SA FIS	- Simulated Annealing tuned Fuzzy Inference System
SNR	- Signal to Noise Ratio
SSE	- Sum Squared Error
SWATH	- Small Waterplane Area Twin Hull
TCM	- Thruster Control Matrix
UUV	- Uninhabited Underwater Vehicle

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

Autonomous underwater vehicles (AUVs) are widely used for conducting research, commercial and navel operations, often in harsh and dangerous environments. By definition a truly autonomous vehicle needs to have a certain level of fault tolerance in-built to allow the successful completion of its mission. Any method used to increase this level of fault tolerance will enhance the vehicle's range of operational capabilities.

The overall aim of this research project is to produce fault tolerant control systems for a given AUV model. Ideally these are capable of handling faults in both the sensors and actuators being considered in this study. The methods and techniques employed should be applicable to other underwater vehicles.

Firstly an investigation into the application of artificial intelligence techniques to the design of fault tolerant control systems for AUVs is undertaken. The study concentrates primarily on the design of fault tolerant control subsystems for both the yaw and roll channels of an AUV with particular attention being paid to the use of neurofuzzy techniques in their development. This constitutes a natural progression of the work of Craven (1999), which developed intelligent control strategies for an AUV.

1.2. OBJECTIVES OF THE THESIS

To achieve the aim of this research programme the following objectives are defined:

(a) Critically review the current fault tolerant control literature.

(b) Investigate and then develop identification models of the AUV dynamics using linear modelling, neurofuzzy and artificial neural network (ANN) approaches.

(c) Devise sensor/actuator failure scenarios within the AUV model to assess the model responses.

(d) Investigate and then develop Kalman filter based fault tolerant control systems to be used as benchmarks. Test the robustness of the control system to various levels of faults.

(e) Develop intelligent fault control systems (IFCSs) for the yaw channel sensors based on neurofuzzy approaches such as the adaptive neuro-fuzzy inference system (ANFIS).

(f) Develop IFCSs for the roll channel sensors based on fuzzy approaches similar to those for yaw channel.

(g) Develop IFCSs for the yaw channel actuator faults using suitable fuzzy approaches.

(h) Remove and replace error sensor used in IFCS for actuator faults.

1.3. CONTRIBUTIONS OF THE THESIS

The main contributions of this thesis are considered to be:

- Investigation into the use of ANFIS [Jang (1991)], simulated annealing [Kirkpatrick et *al* (1983)] and tabu search [Denna et *al* (1999)] to tune fault tolerant fuzzy inference systems (FISs) for underwater vehicles.

- Provide an alternative to Kalman filters for sensor faults. This is achieved by creating several sensor fault recovery FISs which are shown to improve on the

performance of the Kalman filter enhanced control system which is used as a benchmark.

- Consider the effects of actuator faults in the actuators controlling the canards of AUVs and ways of applying fuzzy logic methods to compensate for the actuator faults. To achieve this FISs were produced capable of handling the considered faults occurring within the AUV. This then enables the given AUV to complete missions that would have otherwise been impeded after the occurrence of the given actuator faults.

1.4. PUBLICATIONS

To date the following papers have been published as a result of this research programme:

(a) Pearson A. R., Sutton. R., Burns. R. S., Robinson. P. and Tiano A. (2000). Fault tolerant control strategies for uninhabited underwater vehicles. Underwater Technology, Vol. 24, No 2: 61-72.

(b) Pearson A. R., Sutton. R., Burns. R. S. and Robinson. P. (2000). A Kalman filter approach to fault tolerant control in autonomous underwater vehicles. Proceedings of Fourteenth International Conference on Systems Engineering, Coventry, U.K. September 12-14. Vol. 2: 456-461.

(c) Pearson A. R., Sutton. R., Burns. R. S. and Robinson. P. (2001). A fuzzy fault tolerant control scheme for an autonomous underwater vehicle. Proceedings of IFAC Control Applications in Marine Systems 2001 Conference, Glasgow, U.K. July 17-20.

(d) Sutton. R., Pearson. A. R. and Tiano. A. (2001). A fuzzy fault tolerant control scheme for an autonomous underwater vehicle. IEEE Proceedings of Methods and Models in Automation and Robotics 2001 Conference, Międzyzdroje, Poland, August 28-31. Vol. 2: 595-600.

See Appendix A for further details.

1.5. OUTLINE OF THE THESIS

Chapter 2 of this thesis provides the reader with a critical review of fault tolerant control systems for plants in general as well as specifically for AUVs. This review is split into two sub-sections the first being fault tolerant controllers for non underwater vehicles and then the second considering fault tolerant control systems which do operate in underwater vehicles. The need for a fault tolerant control system is then discussed, clarifying the motivation behind the work submitted within this thesis and highlighting the current lack of AUV fault tolerant control systems.

The dynamic characteristics of the underwater vehicle are detailed in Chapter 3, along with dynamic models of the vehicle, faults to be considered, and the methods used to tune the FISs. Various dynamic models are presented which model the yaw and roll channels of the AUV. The types and levels of faults being considered are also defined and explained within this Chapter.

Benchmark fault tolerant controllers are developed in Chapter 4. A Kalman filter approach to fault tolerant control is considered along with the original controllers specifically developed for the given AUV. A Kalman filter enhanced controller is defined and then compared to the ideal system and the original controllers for both sensor and actuator faults being considered. The Chapter concludes by suggesting which of the controllers should be used for a comparative basis with the results of the following Chapters.

Several sensor recovery FISs are designed in Chapter 5. Using the Kalman filter as a benchmark FISs are developed using the Q statistic and linear models of the AUV, derived in Chapter 3, along with the faulty sensor information as inputs to estimate the actual sensor information required. The ANFIS [Jang (1991)] and simulated annealing [Kirkpatrick et *al* (1983)] tuning methods are used to produce sensor recovery FISs. The results for two FISs for each of the sensors being considered are presented. One of each type of tuning method considered.

Chapter 6 presents the actuator fault tolerant FIS. Only the yaw channel is considered for the actuator faults. The simulated annealing [Kirkpatrick et *al* (1983)] and the tabu search [Denna et *al* (1999)] methods are used to tune two types of FISs. The first being the identity FIS and the second a heuristic FIS. The actuator fault tolerant FIS

uses three inputs, the demand placed on the ANFIS controller, the control signal produced by the ANFIS controller and error in the actuator to produce a value by which the ANFIS control signal is enhanced to correct for the fault. The results are presented and compared to those of the benchmark results from Chapter 4.

Consequently, Chapter 7 discusses a natural progression of the actuator fault tolerant FIS developed in Chapter 6. The need for a sensor to feed back the position of the effected actuator is removed. Instead of the sensor a FIS estimator is introduced to further enhance the fault tolerance of the control system.

Concluding remarks, ideas for future work and a summary of the effectiveness of certain fault tolerant FISs are presented in Chapter 8.

CHAPTER 2

A REVIEW OF UNINHABITED UNDERWATER VEHICLE FAULT TOLERANT CONTROL SYSTEMS

2.1. INTRODUCTION

In recent years, considerable interest has been shown into the commercial, scientific and naval use of uninhabited underwater vehicles (UUVs). UUV being a generic expression to describe both an autonomous underwater vehicle (AUV) and a remotely operated vehicle (ROV). An AUV being a marine craft which fulfils a mission or task without being constantly monitored and supervised by a human operator, whilst a ROV is a marine vessel that requires instruction from an operator via a tethered cable. Commercial, naval and scientific operational specifications for UUVs continue to become more challenging in line with the advances being made in control engineering. In order to survive actuator and/or sensor failure during a mission, such vehicles need to possess a reconfigurable or fault tolerant control system. This Chapter explains the basic principles of fault tolerant control systems. It then reviews their application in the design of UUVs and other systems where it is considered a technology transfer is possible.

By steering an aircraft via differential engine thrust, the captain of a crippled American Airlines DC10 landed safely at Windsor, Ontario, under circumstances similar to those which claimed the lives of three hundred and forty-six passengers and crew of a Turkish Airlines aeroplane in the forest of Ermenouville, France on 3 March 1974 [Eddy et *al* (1976)]. The successful survival of the DC10 at Windsor can be attributed to the fast adaptation ability of the pilot to control what had become, in effect, a different vehicle.

It is interesting to note that the DC10 pilot is claimed to have the maxim "he who hesitates will probably survive" insofar as hasty action may make a situation irrecoverable, but calm experimentation in a high stress environment may well lead to success. The event itself was "recoverable" because it was not totally unexpected, a

similar problem having luckily been identified and solved on a training simulator some months earlier by the particular pilot concerned. Once he recognised the symptoms of the flight failure, the pilot was relatively well equipped to land the aircraft.

As will be seen from the material to be presented herein, the incident recounted above provides an excellent example of a human reconfigurable control scheme. Given the ongoing advances being made in control engineering and artificial intelligence techniques, serious consideration is now being devoted to the development of automated reconfigurable control systems (RCSs) that will operate autonomously whether or not there is a human in the loop.

Demands are growing for the requirement of AUVs to be able to operate at extreme depths and/or in confined areas such as under packed ice. Unfortunately, owing to the nature of these vehicles, data transmissions to and from the craft to the mother station through the sea water medium are poor. Thus, by definition an AUV has to be totally self-sufficient for the duration of a mission. Hence, in order for an AUV to survive sensor and/or actuator failure in this environment, it is paramount to have on board a RCS. Such RCS's could also be beneficially installed in ROVs. Their employment within ROVs would lighten the work load of the human operators, whilst at the same time allowing them to maintain overall supervisory control.

Thus, the following two questions may be posed:

(i) What are the essential elements in a non-human RCS?, and

(ii) How does such a system function?

This Chapter attempts to answer these questions and also to review applications of this technology in the underwater vehicle field and other areas. For the interested reader some other excellent reviews on this subject area can be found in Antsaklis et *al* (1991), Rauch (1994), Bodson and Groszkiewicz (1997) and Fossen and Fjellstad (1995).

Throughout this text it is assumed the reader has a basic understanding of control engineering principles. If this not the case reference should be made to Kuo (1982) for an introduction to the subject. Control strategies, which have been applied to UUVs, are reviewed by Craven et *al* (1998). Some of the control schemes described were developed using artificial intelligence techniques. Background reading on artificial intelligence approaches can be found in Kosko (1994) and Hassoun (1995).

2.2. FAULT TOLERANT CONTROL

If a vehicle is damaged, or has a sensor or actuator failure during operation it may not be possible to repair the damage or fault immediately. There may, however, be a requirement for it to continue operating until it is possible to carry out the necessary repairs. Under such circumstances, the overall system is called an impaired system. When such a scenario as this occurs it is very unlikely that a standard control scheme will be of much use and therefore a different kind of control system, which can handle the anomalies would be more appropriate. A controller with such capabilities is considered to have a fault tolerant, a reconfigurable or a restructurable structure. Collectively, such systems are generically called 'fault tolerant' systems and the controllers called 'fault tolerant' controllers. However each approach operates in a different manner. As well as being the generic term for this type of system it is also possible to have a fault tolerant controller that is neither reconfigurable or restructurable.

All controllers have authority over active parts, such as surfaces or motors of some kind, which they use in a certain configuration to perform given tasks. These controllers can be simple, or very complex, depending on the performance requirements of the system and will be designed accordingly.

Having dealt with the basic ideas concerned with a fault tolerant controller, it would now be helpful to describe the objectives of this approach. The first priority of such a controller is to stabilise the system. Having stabilised the system, the next step is to try to return the system as close as possible to normal operating conditions which will necessitate the use of different control laws.

2.2.1. Fault Tolerance

A fault tolerant controller is capable of maintaining a system at a given level after it has been damaged. The controller does not necessarily return the system to a perfect condition after the damage, but, obviously, this would be an advantage. Such a device has three subsystems or components that constitute the control system.

The first subsystem deals with detecting if a problem has arisen in the system, the second subsystem copes with identifying the location of the fault and its seriousness

and the third is the actual reconfiguration subsystem itself. A simple block diagram of these subsystems and how they fit into a system is shown in Figure 2.1.

The fault tolerant controller is only called into use when there is a fault in the system. Hence, the first part of the controller, which deals with detecting if a fault has occurred is continually monitoring the plant. This subsystem is not concerned with what the problem is or how to fix it but only whether the system is performing as required. This can be achieved by having an analytical method for checking out the system, this method compares what should be happening to what is actually happening in the system. When a fault has occurred this subsystem informs the next subsystem that there is some problem and continues to do so until the fault had been corrected.

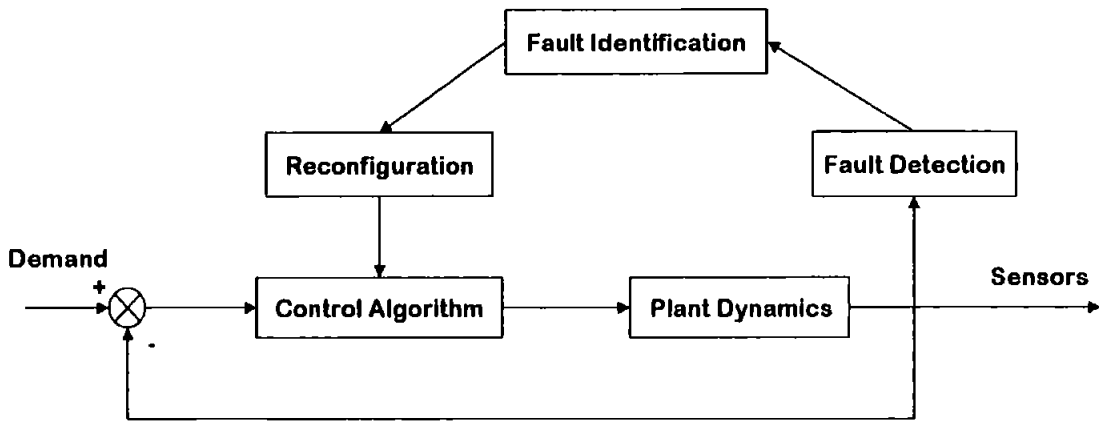


Figure 2.1 A Reconfigurable Control System

The function of the second subsystem is to identify what fault has occurred in the system and the seriousness of the fault. There are many different kinds of analytical methods that have been developed for this purpose, such as the multiple model method [Rauch (1995)] or the generalised likelihood ratio method and will be discussed in more detail later in this section. It is important to know the seriousness of the problem as this will help the controller decide how to deal with the problem. Basseville (1988) has investigated many methods to identify faults and Wang (1995) has investigated fault diagnosis in some detail.

A considerable amount of research has been undertaken using model-based methods to detect and identify faults in systems. Patton et *al* (1995) describe several methods based on more traditional approaches and gives some general guidelines for the

implementation of these types of subsystems. In addition Artificial Neural Networks (ANNs) have also been used by Naidu et *al* (1990) for sensor failure detection.

Alessandri et *al* (1999) considered actuator failures in a ROV using approximate non-linear models of its dynamics. The output of the model is compared to the actual output to detect and diagnose the fault. This is performed by a bank of estimators, which are extended Kalman filters. Extended Kalman filters were used because of the non-linearities of the model and the inability of ordinary Kalman filters to work with non-linear models. Results are reported to show the effectiveness of this approach compared to the unfiltered model output.

It is the final component in which the type of controller is defined. A fault tolerant controller may use many different methods to handle faults in the system. One way is for the fault tolerant controller to make use of a second backup component. This allows the system to continue operating until the first component can be repaired or replaced.

2.2.2. Reconfigurable Control

A reconfigurable controller [Gao and Antsaklis (1992)] deals with a fault by reconfiguring the control laws of the system. The reconfigurable controller has a basic set-up very similar to that of the fault tolerant controller described in Section 2.2.1 and again consists of three subsystems.

The fault detection and identification subsystems are as explained in section 2.2.1. Once it has been established that there is a problem and identification of it has taken place, the only remaining operation is the reconfiguration of the controller. In this subsystem one of a set of predefined reconfigurations, with different control laws, is used to accommodate the fault. This is achieved by using components of the system for purposes other than their designed task. The predefined reconfigurations being those predicted by the designer. Thus the system will be only able to handle those problems envisaged by the designer.

2.2.3. Restructurable Control

The modus operandi of a restructurable controller [Looze et *al* (1985)] is again similar to that in Section 2.2.1 where the first two subsystems are used for detection and

identification. Once again the difference lies in the third subsystem. This time the controller tries to restructure the control laws to accommodate the fault. This is achieved by using every available component of the vehicle, for purposes other than their designed task, in the same fashion as the reconfigurable controller. However this method does not need any predefined sets of control laws, unlike the reconfigurable controller, and therefore may be very flexible at handling unanticipated faults. Such an approach lends itself to solution by artificial intelligent techniques.

2.3. FAULT TOLERANT CONTROL SYSTEMS FOR NON-UUVs

The vast majority of research in the area of fault tolerant control systems has been concerned with plants other than UUVs and therefore this section has been included. It contains explanations of fault tolerant, reconfigurable and restructurable controllers used by other related systems. The actual ideas discussed in Section 2.3.1, however, can be modified and applied to underwater vehicles.

2.3.1. Fault Tolerant Control

The fault tolerant controller, used in the regulation of the feedwater system in a four-loop pressurised water reactor power plant, presented by Eryurek and Upadhyaya (1995) is capable of handling both sensor faults and controller failure. However this controller cannot handle equipment malfunctions (actuator faults) or multiple simultaneous faults. The controller is made up of the following five major components:

1. Parallel control module
2. Signal validation module
3. Command validation module
4. Decision making module
5. System executive module

These five modules and the method by which they are connected can be seen in Figure 2.2. The control system has three different controllers working in parallel with each other. The system uses a method called horizontal redundancy to decide which of the controllers to use at any given time. Horizontal redundancy feeds different subsets of measurements to each of the controllers and then compares the outputs.

When all the outputs are the same the horizontal redundancy procedure has no effect. When a fault has occurred one of the outputs changes and the horizontal redundancy procedure overrules the controller with a different output. This allows any fault to be overruled by the other controllers that have fault free information. The three controllers used in the example are a reconstructive inverse dynamics controller, a fuzzy logic controller and a conventional proportional integral derivative controller. These three different controllers provide different methods to reach the same result when controlling the plant. As they use different inputs a single fault will only affect one controller. This will then be out-voted by the other two unaffected controllers. The same result would be achieved if one of the controllers was to develop a failure during normal operations.

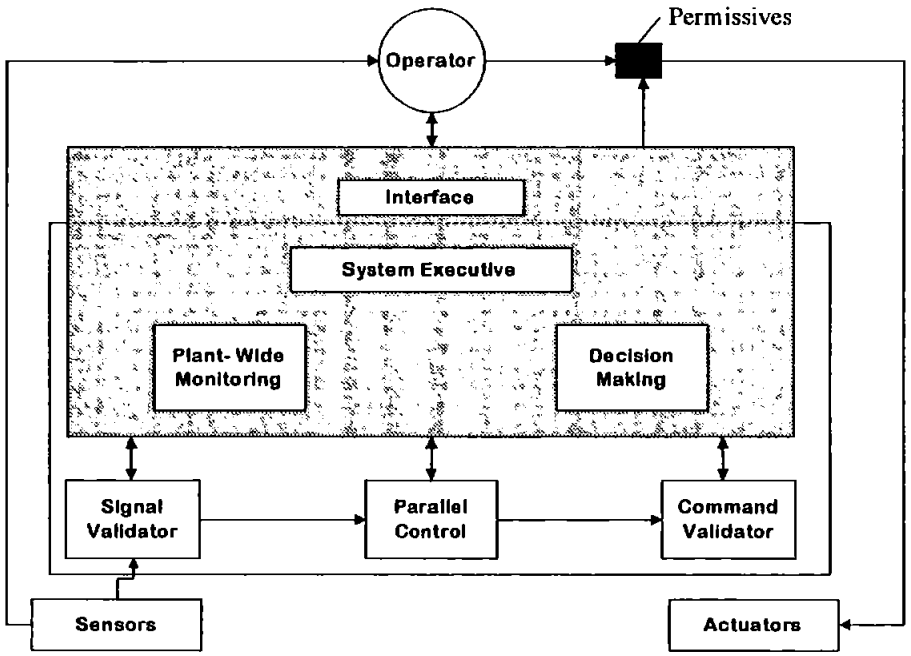


Figure 2.2 Integrated-System Components for Advanced Plant Control [After Eryurek and Upadhyaya (1995)]

This is the main fault tolerant feature of the system. A second is the signal validation module which uses two different routines, namely process empirical modelling and an ANN technique, to check against the actual output of the plant before going to the controllers. In a similar method the command validation module checks the outputs of each controller before the decision making module chooses the best response.

Tests were performed where faults were introduced into one of the controllers. The results show that the decision module is capable of handling these faults without the system becoming unstable. It was also shown that the decision module changed from one controller to another without a fault occurring. This took place because the controller's approach to the situation was judged to be an improvement on the original.

Despite this fault tolerant controller being designed for use in large scale systems, the idea of a parallel controller, as presented here, could be easily applied in the area of marine underwater technology. In addition, the ability to change from one controller to another to improve the performance could provide an UUV with a much more flexible control structure.

Lopez-Toribio et *al* (2000) presents a fault tolerant control scheme based on a Takagi-Sugeno fuzzy model that is used to control a rail vehicle traction system.

The system is split into two major sections. The first is the fault detection and isolation scheme based on five fuzzy observers. This scheme uses residual information gained from fuzzy observers to detect sensor faults within the system. The system then decides which observer is sending out the correct information before sending it to the fuzzy controller.

This fault tolerant control scheme is then tested with sensor faults being simulated within the torque and flux sensors. The results show firstly how the system copes without the scheme, i.e. large oscillating errors are shown in the torque error, and then how much better the torque error is controlled with the scheme in place.

A similar approach could be used to compensate for sensor faults occurring within an AUV.

Other papers in this area are Stengel (1991) which gives a general review of fault tolerant controllers, some of which use ANNs, and McLean et *al* (1997) which deals with sensor faults in a helicopter, also using ANNs.

2.3.2. Reconfigurable Control

Rauch (1995) considers autonomous control reconfiguration in relation to fault accommodation and learning systems. One of the approaches considered is multiple models, another is a single model with adaptive techniques for updating system characteristics. This may be considered a restructural technique and discussed herein.

When considering the multiple model approach the general form for a non-linear system is shown in Equation (2.1) and the appropriate assumptions are stated.

$$\mathbf{xR}_{kt+1} = fR[\mathbf{xR}_{kt}, \mathbf{uR}_{kt}, \mathbf{qR}] + \text{plant noise} \qquad (2.1)$$

Where q is the parameter values vector, **xR** is the state vector, **uR** is the control vector, $\mathbf{uR}_{kt} = gR[\mathbf{x}_{kt}^*, \mathbf{q}^*]$, gR is a function of $\mathbf{x}^*$ (estimated state) and $\mathbf{q}^*$ (estimated vector parameters), and also fR is the state transition function. The subscript kt is the value at the ktth time.

An example is then given for terminal guidance of an interceptor missile. The target that the missile is attempting to hit can make unknown manoeuvres. The multiple models represent the sets of possible manoeuvres the missile can perform.

The multiple models were run off-line by the author and a single extended Kalman filter is used on-line to measure the target state and compare with stored estimates from the off-line multiple models. The multiple models are generated using a general regression neural network.

When first considering this system it may not appear to be a RCS, but it does fit the definition given above. This example however is not concerned with handling a fault within the system, but is concerned with a changing variable outside of its control. It does have many models each with its own control configuration and the system does change between them depending on the situation.

A simulation was performed by Rauch (1995) using an unmodified Kalman filter, the multiple model approach discussed above and a perfect guidance approach where the target trajectory is known exactly. The results showed that the multiple model had a hit probability of 50% which, as one would expect, is better than the original unmodified Kalman filter (15%), but worse than the perfect guidance (84%). The results presented show that an increase in performance is possible with minimal increase in on-line computation. It is suggested in the paper that fuzzy logic and artificial neural network techniques could be used in this approach to improve the given method and produce even better results.

Gao and Antsaklis (1992) develop a different approach to RCS design called perfect model following. This is a development of standard linear model following methods that are designed to make the output of the plant match the output of a model system

with the desired behaviour. There is an explanation of standard linear model following methods, which states that they need both a feedforward and feedback controller in order to fulfil the task. The difference between a standard linear model following method and the perfect model following method is that for the latter the state variables of the model match the state variables of the plant.

For perfect model following to be achieved Equation (2.2) must hold and the solution is of the form Equation (2.3).

$$\left(\mathbf{A}_m - \mathbf{A}_p\right) \mathbf{xG}_p + \mathbf{B}_m \mathbf{u}_m - \mathbf{B}_p \mathbf{u}_p \equiv 0 \tag{2.2}$$

$$\mathbf{u}_p = \mathbf{B}_p^+ \left(\mathbf{A}_m - \mathbf{A}_p\right) \mathbf{xG}_p + \mathbf{B}_p^+ \mathbf{B}_m \mathbf{u}_m \tag{2.3}$$

Where $\mathbf{xG}_p \in \Re^n; \mathbf{u}_m, \mathbf{u}_p \in \Re^m; \mathbf{A}_m, \mathbf{A}_p \in \Re^{n \times n}; \mathbf{B}_m, \mathbf{B}_p \in \Re^{n \times m}$. $\mathbf{A}_m$ and $\mathbf{B}_m$ are the reference model coefficients. $\mathbf{A}_p$ and $\mathbf{B}_p$ are the plant coefficients and $\mathbf{B}_p^+$ represents the pseudo-inverse of the matrix $\mathbf{B}_p$. $\mathbf{u}_m$ and $\mathbf{u}_p$ are the inputs to the reference model and plant respectively and $\mathbf{xG}_p$ is the plant state.

These equations lead to Erzberger's conditions (Equation (2.4) and (2.5)).

$$\left(\mathbf{I} - \mathbf{B}_p \mathbf{B}_p^+\right) \left(\mathbf{A}_m - \mathbf{A}_p\right) = \mathbf{0} \tag{2.4}$$

$$\left(\mathbf{I} - \mathbf{B}_p \mathbf{B}_p^+\right) \mathbf{B}_m = \mathbf{0} \tag{2.5}$$

Where $\mathbf{I}$ is the identity matrix.

In order to satisfy Erzberger's conditions the system must obviously have the same number of inputs as states. This is very rare and so it is difficult to find an appropriate reference model that represents the desired dynamics and satisfies the conditions. However, even if the conditions are not satisfied then Equation (2.3) can still be used to find $\mathbf{u}_p$.

There are two drawbacks with this method. If the conditions are not met then the system can become unstable and there is no control over the location of the poles of the system. The poles are the key to determining if a controller is stable in that they are the roots of the characteristic equation associated with the system transfer

function. If any of these poles are located on the right hand side of the s-plane the system will become unstable [Kuo (1982)]. One advantage is that this type of control system is not very complex and the method does not use $\mathbf{xG}_p$.

Kim et *al* (1997) uses fuzzy logic and an ANN for the detection and isolation section of the reconfigurable control system before the remaining section reconfigures the control laws of the system to handle failures. The block diagram of this system can be seen in Figure 2.3, with the neural network using the control signals and the measurable system outputs as its inputs. The ANN is originally trained off-line to detect faults but is then further trained on-line to update the network.

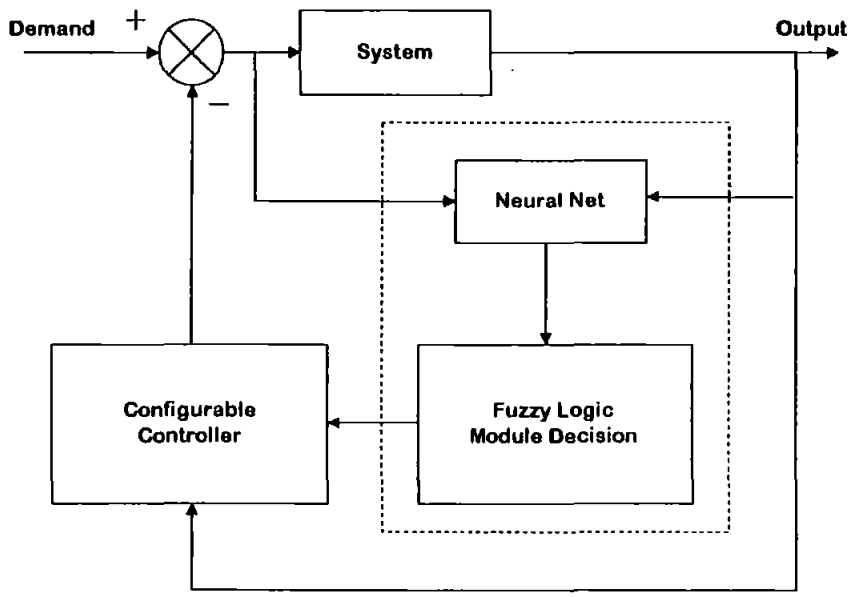


Figure 2.3 SWATH Control System Set-up [After Kim et *al* (1997)]

These ideas were then used in conjunction with a small waterplane area twin hull (SWATH) vessel. A back propagation ANN is used which has sixteen inputs, two hidden layers with sixteen processing elements in each layer, and one output. A fuzzy logic block takes the output from the ANN and decides if a failure has occurred.

ANNs are made up of a collection of neurons that are arranged in layers. All ANNs have an input and output layer, however they also have a varying number of hidden layers depending on their function. For fault detection and isolation a single hidden layer is sufficient. An example of one type of network is the feedforward multilayer

perceptron, which can be seen in Figure 2.4. Back propagation uses information that is fed back from the output layer to the input layer as a further input. This method is described by Waldock (1996) and an example of error backpropagation can be seen in Figure 2.5.

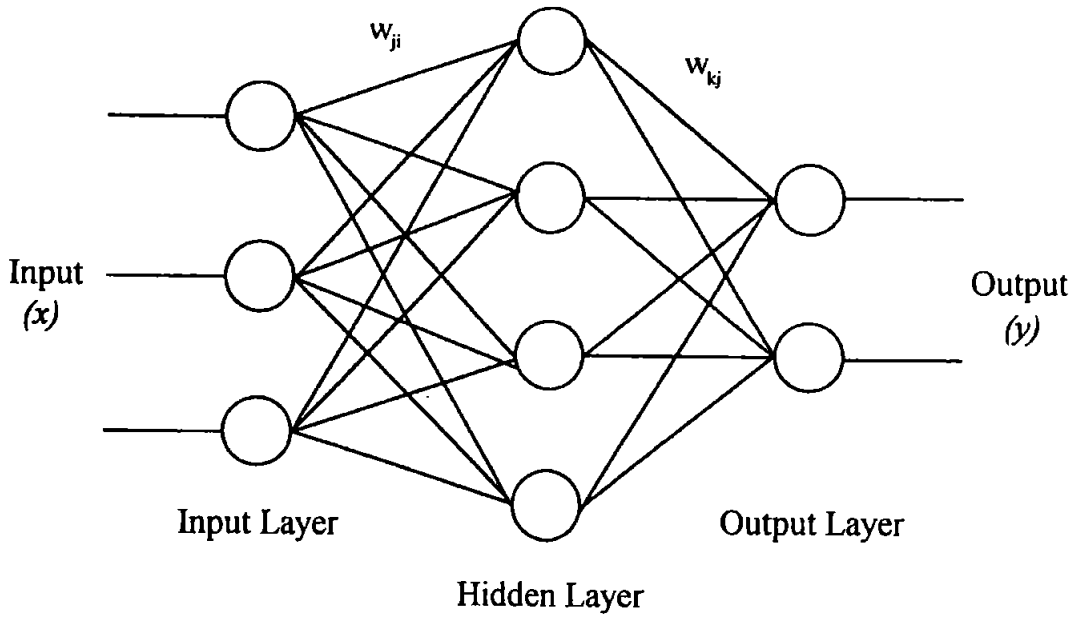


Figure 2.4 The Feedforward Multilayer Perceptron

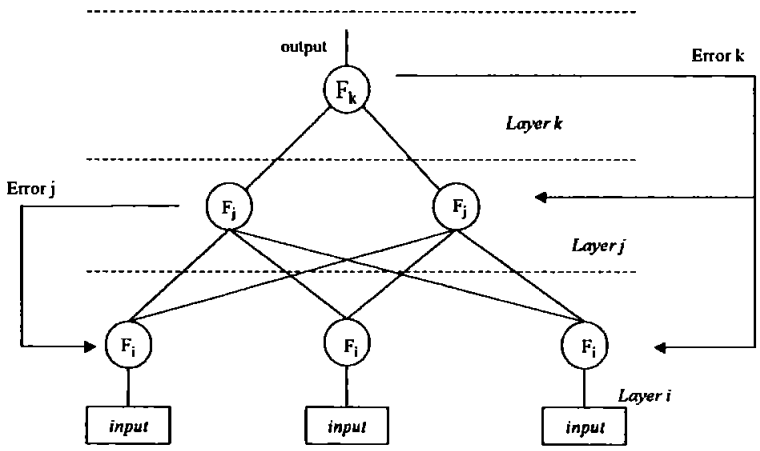


Figure 2.5 Error Backpropagation [After Waldock (1996)]

Two examples of failure detection using a SWATH vessel are then performed and explained. The first example shows how the failure has been detected but no reconfiguration occurs. For this example, the ANN detection system was trained on

four sets of data with one control surface failing. Then a further ten tests with varying levels of failure were performed. It took the system between 8 and 44 seconds to detect the failure. No false alarms were recorded during any of these tests.

In the second example, reconfiguration does take place after the failure has been detected but there was no explanation of how the system is reconfigured to handle the failure. The same ANN is used for this example and one of the previous tests is repeated. The ANN and fuzzy logic subsystems take 19 seconds to detect the failure. After this, the reconfiguration subsystem makes the necessary modifications and the system returns to a stable state.

This system could be improved by training the ANN with further examples. This should speed up the process of identifying failures, but may increase the risk of a false alarm if the ANN was over trained. Over-training an ANN occurs when the network has been trained on the training data to such a degree that it will match it very well, but will lose its generalisation and thus, when checking other data it would incur large errors. The fuzzy logic decision module could also be optimised after further testing. The paper does not explain the actual reconfiguration but does provide a good example of the other two subsystems used, i.e. failure detection and identification.

Other work in this area presents several different failure scenarios with respect to aircraft control reconfiguration during flight [Bodson and Groskiewicz (1997)]. The aim was to control an aircraft after it had suffered surface damage and/or actuator failures by using some form of adaptive controller. Three algorithms were presented and compared. The result being that the direct input error algorithm was deemed the most applicable to the problem.

2.3.3. Restructurable Control

Rauch (1995) also considers a method that fits the definition of restructurable control. This is illustrated for a non-linear system using an adaptive controller in a SWATH ship. Three proportional-integral-derivative controllers control heading, pitch, and roll. It then has an adaptive controller consisting of two blocks, computation element and decision element, which chooses the appropriate control laws for the conditions and adapts the control parameters continuously as the mission progresses. This can be seen in block diagram form in Figure 2.6.

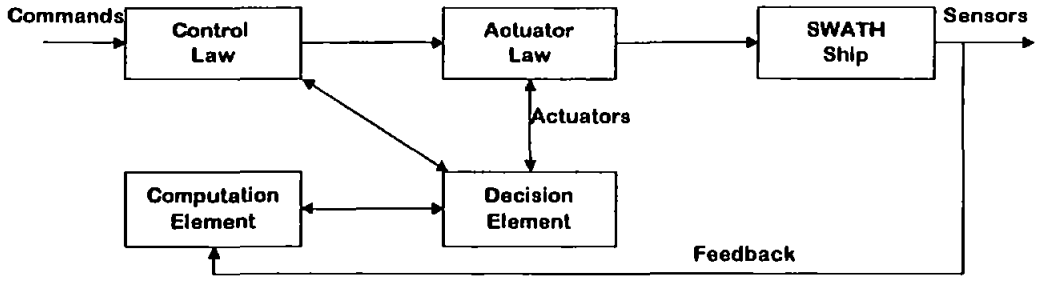


Figure 2.6 Block Diagram of Adaptive Controller [After Rauch (1995)]

The algorithm to generate the non-linear function used in the adaptive non-linear model is given but there are no examples of it in use. The model could be of any system that requires a restructurable controller. The basic idea of this algorithm is to take a set of training data and find a function that fits the data. This forms the basic starting model and then new information is input, as it becomes available. The model is then updated with the new information being the most important, but the old information is also taken into account. This could be used in some kind of fault tolerant system, as the new information would be from the damaged system. It is suggested that fuzzy logic and artificial neural network techniques could be used to improve this approach.

Other papers in this field of research are Looze et *al* (1985) which uses linear quadratic design techniques to produce a control system for an aircraft suffering one or more control element failures. Ochi and Kanai (1991) which presents a restructurable flight control system based on a linearization method, and Diao and Passino (2001) which presents an adaptive Takagi-Sugeno fuzzy controller for a turbine engine.

2.4. UNINHABITED UNDERWATER VEHICLES

This section considers the ideas from the previous sections and new ideas with respect to UUVs. Many of the concepts use a basic controller and then append a fault handling portion or redesign a standard controller. Eleven control architectures for underwater vehicles are given in Valavanis et *al* (1997), whilst Leonard (1995) shows that the notion of an UUV being able to operate after a fault has occurred is a reasonable one. This is achieved by considering the six degrees of freedom that most UUVs have as a mathematical set of dynamic equations and showing that losing motion in one direction does not effect the domain of the set and hence is a recoverable fault. This is then demonstrated using a model that performed a yaw movement by using only roll and pitch movements. It was shown that by a positive roll and then pitch motion followed by a negative roll and then reverse pitch motion the net motion of the underwater vehicle will be in the yaw direction.

2.4.1. Fault Tolerant Control

In the area of underwater vehicles, fault tolerant controllers can be considered the most general type of controller and use some of the simplest methods. Yang et *al* (1999) describe two fault tolerant systems for the Omni-directional intelligent navigator (ODIN) vehicle. ODIN is an AUV with six degrees of freedom. These two fault tolerant systems focus on thruster and sensor failures in the vehicle. Results are then presented to show the effectiveness of the systems.

The first fault tolerant system is for ODIN's thrusters. This system uses the thruster control matrix (TCM), Equation (2.6). This matrix represents the thruster output force to input force relationship.

$$\begin{bmatrix} Fox \\ Foy \\ Foz \\ Mox \\ Moy \\ Moz \end{bmatrix} = \begin{bmatrix} ss & -ss & -ss & ss & 0 & 0 & 0 & 0 \\ ss & ss & -ss & -ss & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & Ry \times ss & Ry \times ss & -Ry \times ss & -Ry \times ss \\ 0 & 0 & 0 & 0 & Ry \times ss & -Ry \times ss & -Ry \times ss & Ry \times ss \\ Rz & -Rz & Rz & -Rz & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} HorTh1 \\ HorTh2 \\ HorTh3 \\ HorTh4 \\ VerTh1 \\ VerTh2 \\ VerTh3 \\ VerTh4 \end{bmatrix} \quad (2.6)$$

Where Fox, Foy, Foz are forces in the x, y, z directions; Mox, Moy, Moz are moments in the x, y, z directions; ss is 0.707; Ry is the distance from ODIN's centre to the centre of the vertical thruster; Rz is the distance from ODIN's centre to the centre of the horizontal thruster; $HorTh$(x) is the force of the horizontal thruster (where x = 1,2,3,4); and $VerTh$(x) is the force of the vertical thruster (where x = 1,2,3,4).

This matrix is used to calculate the required thruster force for each thruster as shown in Equation (2.7).

$$[Thruster \quad Force] = [TCM]^{-1}[Force \quad Input] \quad (2.7)$$

This can then be used to find the correct input voltage to produce the required output thruster force and hence the correct movement by ODIN.

There were two constraints placed on the fault detection and isolation subsystems. The first limited the number of thrusters that could fail during a mission, one vertical and one horizontal thruster. The second constraint was that once a thruster fault is detected then it is out of operation throughout the mission.

The fault detection and isolation processes were implemented as one process due to each thruster being fitted with its own Hall effect sensor [Hatton and Fennell (1999)]. The desired voltage is then compared, to the voltage measured by the sensor, using the conditional algorithm shown on the next page:

IF (Input Signal-Output Signal)/Input Signal > TOLERANCE
THEN count number of times TOLERANCE is continuously exceeded
IF number of times > TOLERANCE TIME
THEN send signal that Thruster is Faulty
ELSE reset counter and repeat monitoring routine

The thruster fault accommodation subsystem, having determined where the fault is located, then eliminates the corresponding column in the TCM and recalculates the required input voltage for the remaining thrusters. This effectively reconfigures the TCM to permit ODIN to continue with its mission.

Two tests where two of the thrusters failed during a simple mission, were performed and showed that the system could handle this fault by doubling the voltage to the remaining thrusters. This allowed the vessel to finish the mission and remain at the desired depth. These two tests do show that the system can handle some simple faults, but it is not shown if it could handle a fault where it is not possible to double the voltage to the remaining thrusters. This would be the case if the thrusters were already operating at their maximum when the fault occurred.

The second fault tolerant system presented is concerned with sensor faults. The sensor fault considered is in the heave direction, for which ODIN has two different sensors and one virtual sensor obtained by an analytical model of itself. For this system three assumptions are made about the fault that may occur. The fault is permanent, only one fault may occur and if a sensor is faulty, it is completely inactive and outputs zero.

Once again the fault detection and isolation processes were implemented as one process. This process is a series of IF-THEN logic rules that compare the outputs of the two sensors and one virtual sensor to determine which one has the fault.

The fault accommodation subsystem for this system could not be simpler. The algorithm simply switches over to the good sensor and ignores the faulty sensor output. This shows why only one fault is allowed to occur, as a second fault would leave no good sensors left for the controller to use.

This subsystem was then tested for a fault occurring in each sensor. Both tests showed that the subsystem worked well with the only noticeable effect being the change in oscillation of ODIN. This was caused by the different characteristics of the sensors and not the fault tolerant system. The oscillation was greater when the sonar sensor

was used. This was due to its relatively low resolution when compared to the pressure sensor.

The fault tolerant systems presented were reasonably effective at detecting, isolating and accommodating faults for the ODIN vehicle. These systems are unique to the ODIN vehicle, but the approach and concept can be extended for use in other underwater vehicles in order to deal with similar types of faults.

Podder and Sarker (2001) present and demonstrate a fault tolerant control system for general use in AUVs. The novel approach given to the allocation of thruster forces depends on an excessive number of thrusters being available, and operational, on the given AUV.

A mathematical formulation of thruster forces is then given to prove that the vehicle can be controlled in this way is then presented. This work leads to the derivation of a relationship between the task space acceleration and the thruster forces. Then reasons for the occurrence of thruster faults are given. These faults are detected by monitoring the change in current drawn by motors used by the thrusters. It is then shown mathematically how these faults can be accommodated by redistributing the missing thruster force amongst remaining functioning thrusters, provided there is at least six in fully operational state.

Then results are presented where the ODIN [as seen in Yang et *al* (1999)] vehicle is used in simulations to test the fault tolerant control system. Two cases are given, case one where no fault occurs and case two where two thruster faults occur after 12 seconds (one vertical and one horizontal). In both cases the ODIN vehicle tries to track a circular path of diameter two meters in the horizontal plane in 30 seconds. In both cases this is achieved and when the faults do occur the system compensates with such speed that ODIN appears unaffected.

This proves the effectiveness of the presented method for making an AUV controlled only by thrusters fault tolerant to a limited number of failures within the said thrusters. A similar approach could be used for an AUV controlled by actuators only if there is sufficient over actuation.

Perrault and Nahon (1999) demonstrates the ability of a standard AUV to be fault tolerant to actuator failures. They explain the method used to control the AUV. Two different approaches are used, a standard method when the actuators are available and a method based on work by Leonard (1994) when the actuators are unavailable. For

both methods the actuators are not permitted to reach saturation levels. This will decrease the level of non-linearity of the AUV and make control simpler.

The theory used for the second control system was that motion in one channel of an AUV could be accomplished by using motion from other channels. Two examples are then displayed showing a five degree yaw angle being achieved by two different methods. The first method used the standard yaw controlling actuators and reached and maintained the demanded angle after approximately 40 seconds. The second method used the actuators controlling the roll and pitch motions to create a yaw motion, this method did manage to achieve its aim of a five degree yaw angle, but took over 100 seconds.

Finally, their control system was presented with total failure of the yaw controlling actuators and showed how a yaw demand of ten degrees was achieved by using motions in the roll and pitch channels using the unaffected control actuators.

2.4.2. Reconfigurable Control

This section examines methods used in reconfigurable controllers and shows how they can be used with respect to UUVs in order to handle faults and, in some cases, also improve the general performance.

In the paper by Katebi and Grimble (1999), a whole control scheme is proposed for an AUV, which is composed of three fully integrated layers. These can be seen in Figure 2.7. The top layer is the navigation layer and this is where all of the reconfiguration will take place. The middle layer is the guidance subsystem and the bottom layer is the AUV autopilot.

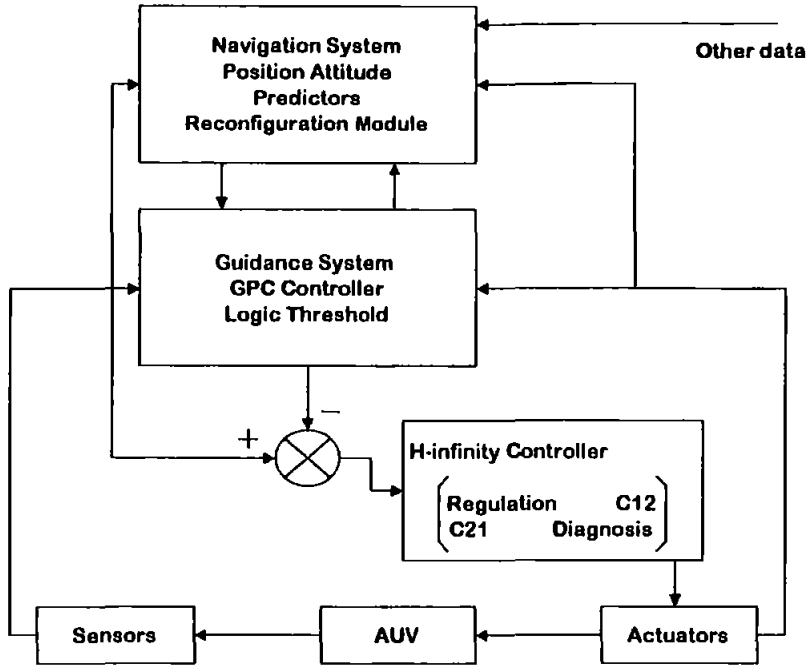


Figure 2.7 AUV Control Scheme [After Katebi and Grimble (1999)]

The AUV model used for this work is described by a set of non-linear differential equations (Equation (2.8)).

$$\begin{aligned} \mathbf{MK}(t)\frac{d\mathbf{xk}(t)}{dt} &= fk[\mathbf{xk}(t),\mathbf{zk}(t),\mathbf{ck}(t)] + gk[\mathbf{xk}(t),\mathbf{zk}(t)]\ \mathbf{uk}(t) \\ \frac{d\mathbf{zk}(t)}{dt} &= hk[\mathbf{zk}(t),\mathbf{xk}(t),\mathbf{u}_c] \end{aligned} \tag{2.8}$$

Where $\mathbf{MK}(t)$ is the mass matrix, the functions *fk* and *gk* are mappings of the vehicle motions into forces, the function $\mathbf{ck}(t)$ represents the hydrostatic and hydrodynamic forces and moments acting on the vehicle. The function *hk* includes the kinematic relations, $\mathbf{u}_c$ is the ocean current constant, **xk** is the vector of surge, sway, heave, roll, pitch and yaw velocities, **uk** is the control input vector and **zk** is the vector of x, y and z positions and roll, pitch and yaw angles.

A linear state-space model of the system (as shown in Equation (2.9)) is then presented for use in the local controller and diagnostic subsystems.

$$\begin{aligned}\dot{\mathbf{x}}\mathbf{kl}(t) &= \mathbf{A}_{kl}\mathbf{x}kl(t) + \mathbf{B}_{kl}\mathbf{u}\mathbf{kl}(t) + \varsigma(t)\\ \mathbf{y}\mathbf{kl}(t) &= \mathbf{C}_{kl}\mathbf{x}\mathbf{kl}(t)\\ \mathbf{z}\mathbf{kl}(t) &= \mathbf{H}_{kl}\mathbf{x}\mathbf{kl}(t) + \xi(t)\end{aligned} \tag{2.9}$$

Where the vector $\mathbf{xkl}(t)$ represents positions and attitude and the translational and rotational with respect to the body axis co-ordinates, $\mathbf{ykl}(t)$ denotes the variables to be controlled, $\mathbf{zkl}(t)$ represents the measured variables, the vector $\mathbf{ukl}(t)$ is the actuator inputs, the signals ς and ξ are zero-mean white noise sequences, $\mathbf{A}_{kl}$, $\mathbf{C}_{kl}$ and $\mathbf{H}_{kl}$ are state space matrices and $\mathbf{B}_{kl}$ is the control matrix.

The local controller used is a H-infinity controller [Thompson (1993)] and is a trade-off between the plant controller and the diagnostic controller. Also shown in the H-infinity controller in Figure 2.7 are the interactions between these two controllers, C12 and C21. This controller is designed particularly for the model used, but the same H-infinity approach could be used to design controllers for other UUV models.

The guidance system of this AUV is a predictive controller (PC) which does not suffer from the problems associated with a line of sight [Healey and Lienard (1993)] algorithm. This is because the PC predicts where the AUV will need to be and so has less overshoot when waypoints (target points for the AUV) are close together. The robustness of the H-infinity and PC are discussed with the outcome being that optimisation can be obtained by appropriate choice of weighting functions for the H-infinity controller and then suitable tuning of the PC.

The main area of interest, reconfigurable control, is then considered. Two approaches to reconfigurable control are examined, which are multiple models and single models with adaptive techniques. For the multiple model approach a number of system models, each with their own corresponding control law, are first obtained and then a decision element chooses which is the most appropriate model and associated control law. For fault diagnosis a model must be included for each particular fault condition and a normal no-fault condition model.

The single non-linear model that is continuously adapting is then presented as the second approach to reconfigurable control. In general the initial model is based on prior information and then continuously adjusted as new information is received. An example is given which uses a gain scheduling system.

Simulations are carried out to demonstrate the improved performance of the new controller compared to a simple H_2 controller [Thompson (1993)]. The H_2 controller works in a similar manner to a H-infinity controller but is less complicated in its structure. A further simulation was then performed where the steering system develops a fault and simultaneously the gyro fails (sensor and actuator faults). The reconfiguration controller is activated when a set level has been passed in the heading. A new controller stabilises the system after 30 seconds of activity.

Tacconi and Tiano (1989) explore reconfigurable control techniques applied to an AUV where they consider both sensor and actuator faults.

A mathematical model (Equation (2.10)) of an AUV using a mobile reference system is presented for use. This model of the vehicle uses six non-linear coupled equations.

$$\begin{aligned} m\dot{u} &= \mathrm{XT} - mg\sin\theta + mvr - mwq \\ m\dot{v} &= \mathrm{YT} + mg\sin\phi\cos\theta + mwp - mur \\ m\dot{w} &= \mathrm{ZT} + mg\cos\phi\cos\theta + muq - mvp \\ \mathrm{I}_x\dot{p} &= \mathrm{KT} \\ \mathrm{I}_y\dot{q} &= \mathrm{MT} \\ \mathrm{I}_z\dot{r} &= \mathrm{NT} \end{aligned} \tag{2.10}$$

Where u, v, w are velocity components along the three axes, p, q, r are the three components of pitch, roll and yaw angular velocities, θ, ϕ, φ the corresponding Euler angles, XT, YT, ZT are the external forces and KT, MT, NT are the external moments.

This leads to a linear state space model of the type shown in Equation (2.11).

$$\dot{\mathbf{x}}\mathbf{t} = \mathbf{A}\mathbf{x}\mathbf{t} + \mathbf{B}\mathbf{u}\mathbf{t} + \mathbf{G}\boldsymbol{\omega}\,\mathbf{t} \tag{2.11}$$

Where $\mathbf{xt} = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \varphi \quad X_0 \quad Y_0 \quad Z_0]^{\mathrm{T}}$ is an augmented state vector, which contains X_0, Y_0, Z_0 the fixed reference system of the vehicles position; $\mathbf{ut} = [nt \quad \delta t_1 \quad \delta t_2 \quad \delta t_3 \quad \delta t_4]^{\mathrm{T}}$ is the control vector, which contains nt the propulsion revolutions per minute, δt_1 and δt_2 the upper and lower rudder angles, δt_3 and δt_4 the port and starboard elevators. $\boldsymbol{\omega}\mathbf{t}$ is a vector that takes external disturbances into

account. The three matrices **A**, **B**, and **G** are determined by both the hydrodynamic derivatives and the vehicle's speed.

A Linear Quadratic Gaussian (LQG) method is discussed which produces a robust method and can handle small faults. This removes the need to consider such faults later and allows the focus to be on larger faults due to system failures.

A LQG control system design requires the control vector be chosen during each transition from one way point of the AUV mission task to the subsequent one in such a way that the expected value of a quadratic cost function J of state and control vector $\mathbf{ut}(t)$ is given by:

$$J = \mathrm{E}\int_{t_0}^{t_f} \left(\mathbf{xt}^{\mathrm{T}}(t)\ \mathbf{QTxt}(t) + \mathbf{ut}^{\mathrm{T}}(t)\mathbf{RTut}(t)\right) dt \tag{2.12}$$

Different types of missions can be easily managed by a proper choice of the weighting matrices **QT** and **RT**.

If it is assumed that $t_f >> t_o$ then a computationally simpler problem can be solved, which supplies a linear feedback of the type shown in Equation (2.13).

$$\mathbf{ut}(t) = -\mathbf{K}\hat{\mathbf{x}}\mathbf{t}(t) \tag{2.13}$$

Where the matrix **K** is obtained by solving a time-invariant Riccati equation, while $\hat{\mathbf{x}}\mathbf{t}(t)$ is an optimal estimate of the state vector $\mathbf{xtk}(t)$ supplied by a Kalman filter.

The linear quadratic method for controlling plants is a robust form of controller and therefore very good at compensating for noise in the system. For further details of linear-quadratic control theory see Dorato et *al* (1995).

Brief consideration is then given to integrated navigation systems, on-line monitoring and fault detection. The model uses Kalman filters in the navigation system and a statistical decision test for the fault detection and identification module.

When faults occur in the AUV structural changes may occur in the mathematical model, which will cause the vehicles performance to decrease to an irreparable level. The robust LQG design can be used to provide a reconfigurable controller. This is achieved by using a previously computed mathematical model to handle the new

system. It computes the new model as required however it would be computationally advantageous to have the solutions stored into a look-up table.

These methods were originally proposed for use in the aerospace field and have easily been adapted for implementation in an AUV control system. Unfortunately, despite discussing the method in detail, no results are presented within the paper.

Derradji and Mort (1996) describe and test two methods of reconfigurable control for a submersible vehicle using an ANN approach. The traditional algorithm for an ANN controller used for such vehicles has problems if any form of control failure occurs. In order to deal with this, two new methods are presented both of which are capable of handling faults.

The first of these methods is the linear model following approach, which uses a state space model of the normal plant as the ideal model and the plant as a state space model of the impaired model. Whilst in operation this approach modifies the signal from the neural network controller to make the impaired system act the same as the ideal system.

The second method is the Error Vector Suppression (EVS) method. This approach simply disregards the error vector element in the adjustment algorithm and thus forces the impaired closed loop system output to be the same as the ideal systems output.

These two methods were tested using the state variable model in Equation (2.14) which is a linear multivariable representation of a large submarine vehicle.

$$\begin{aligned} \mathbf{x}_m(i+1) &= \mathbf{A}_{dm}\mathbf{x}_m(i) + \mathbf{B}_{dm}\mathbf{u}_m(i) \\ \mathbf{y}_m(i) &= \mathbf{E}_{dm}\mathbf{x}_m(i) \end{aligned} \tag{2.14}$$

Where $\mathbf{A}_{dm} \in \Re^{n\times n}, \mathbf{B}_{dm} \in \Re^{n\times r}, \mathbf{E}_{dm} \in \Re^{m\times n}$ are matrixes related to the vehicle dynamics, $\mathbf{x}_m(i) \in \Re^{n\times 1}$ the state of the vehicle model, $\mathbf{u}_m(i) \in \Re^{r\times 1}$ are the inputs to the model and $\mathbf{y}_m(i) \in \Re^{m\times 1}$ are the model outputs, all at time i.

For the tests, a three layer neural network with an input layer containing 8 linear neurons, an output layer containing 4 linear neurons and 30 non-linear neurons in the hidden layer, was used as the controller. The tests involved simulating several levels of control surface failures, first in the rudder and then the starboard stern plane. Both methods provided satisfactory performance over a variety of conditions. The EVS and linear model following both managed to reconfigure the remaining control surfaces to

accommodate the failures. There was very little difference between the methods in performance, but the EVS used less computer memory and so makes it a much better option for real-time applications.

Ishii et *al* (1995) control an AUV using an adaptive ANN which is continuously updated as the AUV operates. For the method described, an ANN is first trained off-line to be the controller for the given AUV and then a second ANN is used to model the output of the system, known as the identification network. These two ANNs act together almost as a single ANN to control the AUV. In the improved method the identification network is regularly updated as the mission develops. This simply improves the performance of the controller, but could, with a little work, be adapted to become some form of reconfigurable controller or even a restructurable controller where the ANN adapts to the new input of the now impaired system to restructure the controller and regain total control of the AUV.

Caccia and Veruggio (2000) use a proportional plus integral-type guidance algorithm to control a prototype ROV's depth and motion. This controller has a three level hierarchical architecture. The motion, during operation, is estimated by a set of sensors all with different capabilities. This information is compared with the mission tasks in order to keep a theoretical track of what the ROV has accomplished and is still required to undertake. Once the controller is informed of the intentions of the ROV, the information is used to compute the force and torque, which must be applied to complete the given tasks. The next step performed by the controller is to take the required force and torque information and translate it into actuator outputs. This is where reconfiguration takes place. In theory, when an actuator fails the translation will simply ignore that actuator as a possible output. As the same force and torque are still required, the workload will need to be redistributed between the actuators that remain functioning, hence reconfiguring the controller to cope with the loss of an actuator. Their reconfigurable controller is capable of handling total failure in one or more thrusters as was shown in the presented test results.

2.5. CONCLUDING REMARKS

This Chapter has provided background knowledge of intelligent control and other strategies for AUVs. It started by explaining why a RCS is required in an AUV. Then an explanation of the three types of fault tolerant controller was given. The next section gave examples of fault tolerant controllers being used in non-AUV systems. Finally, there are examples given of the work undertaken thus far on fault tolerant controllers when used in AUVs.

Throughout the last two sections of this Chapter many of the examples of fault tolerant control systems used artificial intelligence techniques such as fuzzy logic or ANNs to achieve their objectives of creating a level of tolerance to given failures within the systems presented.

There has been much work carried out on non-AUV fault tolerant control, for both sensor and actuator failures. However, limited attention has been given to applying these ideas in the important area of underwater vehicles. In work where failures have been considered within an AUV, the main focus has been on sensor and thruster failures. Clearly from the aforementioned dearth of research into neuro-fuzzy approaches to sensor and canard controlling actuator faults, it is clear that these areas have not been considered in any depth. Therefore it is felt that research in these areas could offer significant advances in the development of fault tolerant control schemes for AUVs and thus would provide an excellent area of research.

The research to be undertaken in this thesis will look at fuzzy logic approaches to both sensor and actuator faults, with the actuators being considered being those which control the AUVs canards. The research presented will lead to an increase in knowledge in the areas being considered.

The next Chapter will give details of the dynamics of the UUV used in this study, the proportion-derivative controller and adaptive neuro-fuzzy inference system (ANFIS) controller developed by Craven (1999) for the given AUV model are also presented. In addition the identification models of the AUV that will be used later in the thesis will also be introduced. The faults to be considered for this work are also displayed along with the methods used to tune the fuzzy inference systems used within the fault tolerant controller.

CHAPTER 3

UNINHABITED UNDERWATER VEHICLE DYNAMICS, IDENTIFICATION MODELS, VEHICLE FAILURES AND TUNING METHODS

3.1. INTRODUCTION

The aim of this Chapter is to explain the dynamics of the supplied uninhabited underwater vehicle (UUV) model and to give relevant information required for the remainder of the thesis.

The information given within this Chapter concerns the identification models and the methods used in their creation, the types faults to be considered, and the methods used to train the fuzzy inference systems (FISs). First, background information on the UUV model dynamics and the benchmark standard proportional derivative (PD) and the adaptive neuro-fuzzy inference system (ANFIS) controllers to be used is introduced. Then the development of the identification models is described and it is shown which type of model is most appropriate for use in the fault tolerant controller. Attention is also given to the sensor and actuator failures to be considered in this work. Finally the three methods which are to be used to tune the FISs in later Chapters of the thesis will be presented.

3.2. UUV MATLAB MODEL DYNAMICS

The model employed throughout this thesis was purposely designed to provide a common design framework within the United Kingdom UUV research community in the navigation, guidance and control field and supplied by the Defence Evaluation and Research Agency (DERA), Sea Systems Sector, Winfrith. If required, the cable dynamics pertaining to a remotely operated vehicle (ROV) can be included during a simulation. The vehicle is considered herein as an autonomous underwater vehicle (AUV). An in-depth description of the dynamics can be found in Craven (1999).

Obviously, the UUV model alleviates the requirement for an in-depth study of the UUV background modelling work. The previous work by Craven (1999) also removes the requirement for an in-depth study of control strategies for UUVs.

3.2.1. Equations of Motion

To implement the vehicle equations of motion use is made of a MATLAB/Simulink simulation model termed *Release Version 1.0/UUVmod-GEN*. This model has been validated against standard DERA hydrodynamic code using tank test data and an experimentally derived set of hydrodynamic coefficients from the Southampton Oceanography Centre's AUTOSUB vehicle.

The inertial terms within the AUV equations of motion are given by Equation (3.1):

$$
\begin{aligned}
&m\left[\dot{u} - vr + wq - x_G\left(q^2 + r^2\right) + y_G\left(pq - \dot{r}\right) + z_G\left(pr + \dot{q}\right)\right] = X \\
&m\left[\dot{v} + ur - wp + x_G\left(pq + \dot{r}\right) - y_G\left(p^2 + r^2\right) + z_G\left(qr - \dot{p}\right)\right] = Y \\
&m\left[\dot{w} - uq + vp + x_G\left(pr - \dot{q}\right) + y_G\left(qr + \dot{p}\right) - z_G\left(p^2 + q^2\right)\right] = Z \\
&I_x\dot{p} - \left(I_y - I_z\right)qr + I_{xy}\left(pr - \dot{q}\right) - I_{yz}\left(q^2 - r^2\right) - I_{xz}\left(pq + \dot{r}\right) \\
&+ m\left[y_G\left(\dot{w} - uq + vp\right) - z_G\left(\dot{v} + ur - wp\right)\right] = K \\
&I_y\dot{q} + \left(I_x - I_z\right)pr - I_{xy}\left(qr + \dot{p}\right) + I_{yz}\left(pq - \dot{r}\right) + I_{xz}\left(p^2 - r^2\right) \\
&- m\left[x_G\left(\dot{w} - uq + vp\right) - z_G\left(\dot{u} - vr + wq\right)\right] = M \\
&I_z\dot{r} + \left(I_y - I_x\right)pq - I_{xy}\left(p^2 - q^2\right) - I_{yz}\left(pr + \dot{q}\right) + I_{xz}\left(qr - \dot{p}\right) \\
&+ m\left[x_G\left(\dot{v} + ur - wp\right) - y_G\left(\dot{u} - vr + wq\right)\right] = N
\end{aligned}
\tag{3.1}
$$

The hydrodynamic force model provided by the DERA is as shown in Equation (3.2).

$$
\begin{aligned}
X &= \frac{1}{2}\rho l^2\left[X'_{uu}u^2 + X'_{vv}v^2 + X'_{ww}w^2\right] \\
&+ \frac{1}{2}\rho l^3\left[X'_{\dot{u}}\dot{u} + X'_{vr}vr + X'_{wq}wq\right] \\
&+ \frac{1}{2}\rho l^2u^2\left[X'_{uu\delta bp}\delta_{bp} + X'_{uu\delta bs}\delta_{bs} + X'_{uu\delta sp}\delta_{sp} + X'_{uu\delta ss}\delta_{ss}\right] \\
&+ \frac{1}{2}\rho l^2u^2\left[X'_{uu\delta bru}\delta_{bru} + X'_{uu\delta brl}\delta_{brl} + X'_{uu\delta sru}\delta_{sru} + X'_{uu\delta srl}\delta_{srl}\right] \\
&+ \frac{1}{2}\rho l^4\left[X'_{qq}q^2 + X'_{rr}r^2 + X'_{pr}pr\right] + X_{wave} + G_X
\end{aligned}
$$

$$
\begin{aligned}
Y &= \frac{1}{2}\rho l^2\left[Y'_{uu}u^2 + Y'_{uv}uv + Y'_{vw}vw + Y'_{v}vT_p + Y'_{v}vT_s\right] \\
&+ \frac{1}{2}\rho l^2u^2\left[Y'_{uu\delta bru}\delta_{bru} + Y'_{uu\delta brl}\delta_{brl} + Y'_{uu\delta sru}\delta_{sru} + Y'_{uu\delta srl}\delta_{srl}\right] \\
&+ \frac{1}{2}\rho l^3\left[Y'_{\dot{v}}\dot{v} + Y'_{up}up + Y'_{ur}ur + Y'_{vq}vq + Y'_{wp}wp + Y'_{wr}wr\right] \\
&+ \frac{1}{2}\rho l^3\left[Y'_{u|r|\delta bru}u|r|\delta_{bru} + Y'_{u|r|\delta brl}u|r|\delta_{brl} + Y'_{u|r|\delta sru}u|r|\delta_{sru} + Y'_{u|r|\delta srl}u|r|\delta_{srl}\right] \\
&+ \frac{1}{2}\rho l^4\left[Y'_{\dot{p}}\dot{p} + Y'_{\dot{r}}\dot{r} + Y'_{p|p|}p|p| + Y'_{pq}pq + Y'_{qr}qr\right] + Y_{wave} + G_Y
\end{aligned}
$$

$$
\begin{aligned}
Z &= \frac{1}{2}\rho l^2\left[Z'_{uu}u^2 + Z'_{uw}uw + Z'_{vv}v^2\right] \\
&+ \frac{1}{2}\rho l^2u^2\left[Z'_{uu\delta bp}\delta_{bp} + Z'_{uu\delta bs}\delta_{bs} + Z'_{uu\delta sp}\delta_{sp} + Z'_{uu\delta ss}\delta_{ss}\right] \\
&+ \frac{1}{2}\rho l^2\left[Z'_{u|w|}u|w| + Z'_{uv}uv + Z'_{w}wT_b + Z'_{w}wT_s\right] \\
&+ \frac{1}{2}\rho l^3\left[Z'_{\dot{w}}\dot{w} + Z'_{uq}uq + Z'_{vp}vp + Z'_{vr}vr\right] \\
&+ \frac{1}{2}\rho l^3\left[+ Z'_{u|q|\delta sp}u|q|\delta_{sp} + Z'_{u|q|\delta ss}u|q|\delta_{ss} + Z'_{w|q/w|}w|q/w|T_b + Z'_{w|q/w|}w|q/w|T_s\right] \\
&+ \frac{1}{2}\rho l^4\left[Z'_{\dot{q}}\dot{q} + Z'_{pp}p^2 + Z'_{rr}r^2 + Z'_{pr}pr\right] + Z_{wave} + G_Z
\end{aligned}
\tag{3.2}
$$

and the hydrodynamic moment model is as shown in Equation (3.3):

$$
\begin{aligned}
K &= \frac{1}{2}\rho l^3\left[K'_{uu}u^2 + K'_{uv}uv + K'_{vw}vw\right] \\
&+ \frac{1}{2}\rho l^3u^2\left[K'_{uu\delta bru}\delta_{bru} + K'_{uu\delta brl}\delta_{brl} + K'_{uu\delta sru}\delta_{sru} + K'_{uu\delta srl}\delta_{srl}\right] \\
&+ \frac{1}{2}\rho l^4\left[K'_{\dot{v}}\dot{v} + K'_{up}up + K'_{ur}ur + K'_{vq}vq + K'_{wp}wp + K'_{wr}wr\right] \\
&+ \frac{1}{2}\rho l^5\left[K'_{\dot{p}}\dot{p} + K'_{\dot{r}}\dot{r} + K'_{qr}qr + K'_{pq}pq + K'_{p|p|}p|p|\right] + K_{wave} + G_K
\end{aligned}
$$

$$
\begin{aligned}
M &= \frac{1}{2}\rho l^3\left[M'_{uu}u^2 + M'_{vv}v^2 + M'_{uw}uw\right] \\
&+ \frac{1}{2}\rho l^3u^2\left[M'_{uu\delta bp}\delta_{bp} + M'_{uu\delta bs}\delta_{bs} + M'_{uu\delta sp}\delta_{sp} + M'_{uu\delta ss}\delta_{ss}\right] \\
&+ \frac{1}{2}\rho l^3\left[M'_{u|w|}u|w| + M'_{uv}uv + M'_{w}wT_b + M'_{w}wT_s\right] \\
&+ \frac{1}{2}\rho l^4\left[M'_{\dot{w}}\dot{w} + M'_{uq}uq + M'_{vr}vr + M'_{vp}vp\right] \\
&+ \frac{1}{2}\rho l^4\left[M'_{q}qT_b + M'_{q}qT_s + M'_{u|q|\delta sp}\delta_{sp}u|q| + M'_{u|q|\delta ss}\delta_{ss}u|q|\right] \\
&+ \frac{1}{2}\rho l^5\left[M'_{\dot{q}}\dot{q} + M'_{pp}p^2 + M'_{rr}r^2 + M'_{pr}pr + M'_{q|q|}q|q|\right] + M_{wave} + G_M
\end{aligned}
$$

$$
\begin{aligned}
N &= \frac{1}{2}\rho l^3\left[N'_{uu}u^2 + N'_{uv}uv + N'_{vw}vw +\right] \\
&+ \frac{1}{2}\rho l^3u^2\left[N'_{uu\delta bru}\delta_{bru} + N'_{uu\delta brl}\delta_{brl} + N'_{uu\delta sru}\delta_{sru} + N'_{uu\delta srl}\delta_{srl}\right] \\
&+ \frac{1}{2}\rho l^4\left[N'_{\dot{v}}\dot{v} + N'_{up}up + N'_{ur}ur + N'_{wp}wp + N'_{wr}wr + N'_{vq}vq\right] \\
&+ \frac{1}{2}\rho l^4\left[N'_{r}rT_b + N'_{r}rT_s\right] \\
&+ \frac{1}{2}\rho l^4\left[N'_{u|r|\delta bru}\delta_{bru}u|r| + N'_{u|r|\delta brl}\delta_{brl}u|r| + N'_{u|r|\delta sru}\delta_{sru}u|r| + N'_{u|r|\delta srl}\delta_{srl}u|r|\right] \\
&+ \frac{1}{2}\rho l^5\left[M'_{\dot{r}}\dot{r} + M'_{\dot{p}}\dot{p} + N'_{pq}pq + N'_{qr}qr + N'_{r|r|}r|r|\right] + N_{wave} + G_N
\end{aligned}
\tag{3.3}
$$

where the hydrostatic terms ($G_{X,\ldots,N}$) acting upon the vehicle are commonly referred to as restoring forces and moments.

Naturally, gravitational forces act down through the vehicle's centre of gravity, whereas the force provided by the buoyancy of the vehicle acts through the vehicle's centre of buoyancy. With respect to the underwater vehicle used within this study, the

Euler angle representations of these restoring forces and moments are thus given by Equation (3.4):

$$\begin{aligned} G_X &= -(W - B)\sin(\theta) \\ G_Y &= (W - B)\cos(\theta)\sin(\phi) \\ G_Z &= (W - B)\cos(\theta)\cos(\phi) \\ G_K &= (y_G W - y_B B)\cos(\theta)\cos(\phi) - (z_G W - z_B B)\cos(\theta)\sin(\phi) \\ G_M &= -(x_G W - x_B B)\cos(\theta)\cos(\phi) - (z_G W - z_B B)\sin(\theta) \\ G_N &= (x_G W - x_B B)\cos(\theta)\sin(\phi) + (y_G W - y_B B)\sin(\theta) \end{aligned} \tag{3.4}$$

With respect to Equation (3.1), Equation (3.2), Equation (3.3) and Equation (3.4) the following parameters describe the AUV model used herein:

$W = 35316$ N	$B = 35316$ N	$l = 7.0$ m	$m = 3600$ kg
$\rho = 1025.2$ kgm^{-3}	$I_x = 320$ kgm^2	$I_y = 8304$ kgm^2	$I_z = 8304$ kgm^2
$I_{xy} = 0$ kgm^2	$I_{xz} = 0$ kgm^2	$I_{yz} = 0$ kgm^2	$x_G = 0.34$ m
$y_G = 0$ m	$z_G = 0.02$ m	$x_B = 0.34$ m	$y_B = 0$ m
$z_B = 0$ m			

It should be noted that whilst the full six degree of freedom equations of motion are reproduced here, the hydrodynamic coefficients of the model remain the property of the DERA.

3.2.2. PD Controller

The classical method to control a vehicle such as this AUV is to use a proportional, integral and derivative (PID) controller or one of its variants. The PID controller uses information from the feedback sensors and then multiplies them by set values to produce a control signal. These set values would be derived from the plant with which the controller is to operate. This method does however have limitations as all of the values are invariant.

For this thesis a linear proportional and derivative (PD) controller, designed by Craven (1999) for this exact AUV model, will be used to demonstrate how a classical approach handles faults being considered. This PD controller was produced from the

open loop to canard rudder transfer function. Thus, the standard PD controller used in this thesis is the same as was used in previous work [Craven (1999)] and is shown in Equation (3.5).

$$G_{c,\psi_e}(s) = \frac{1+1.204s}{1+0.708s} \tag{3.5}$$

Where c is the control signal and ψ_e is the yaw angle error.

The AUV is controlled by many rudders attached to the hull in the configuration shown in Figure 3.1. This project is concerned only with motions in the yaw and roll channels. For this project the yaw channel is primarily controlled by the upper and lower canards, with the roll channel being controlled by the two stern planes. It is possible to control both the yaw and roll channels by using other control surfaces.

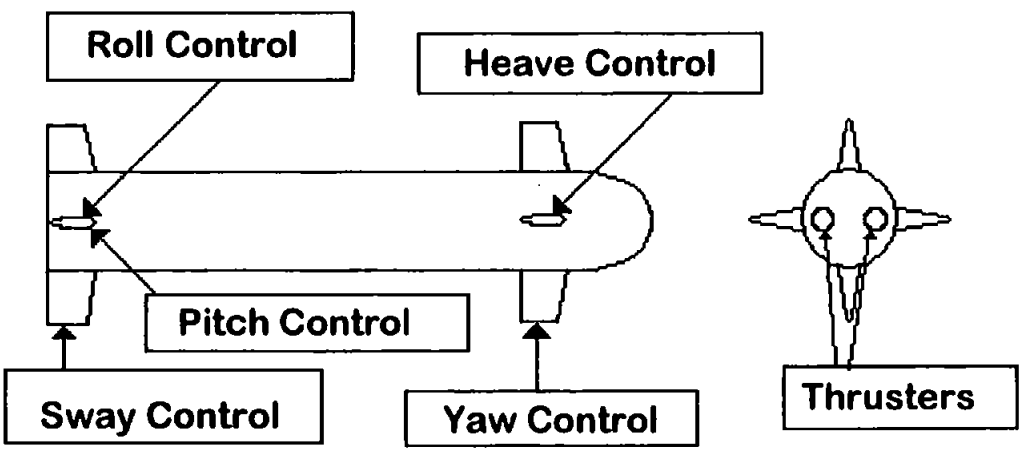


Figure 3.1 Diagram of AUV Control Surfaces

3.2.3. ANFIS Controller

There are several ways of improving on the basic PD controller design, one of which is to use a fuzzy logic controller. For this thesis a fuzzy logic controller, developed by Craven (1999) using the ANFIS approach, will be used as the starting point for the fault tolerant control system. This ANFIS tuned fuzzy logic controller is more robust to perturbations in the hydrodynamic coefficients as shown by Craven (1999). For the definitive guide to this controller it would be best to consult the thesis referenced above, however there now follows a short review.

Functionally, there are almost no constraints on the membership functions of an adaptive network except piecewise differentiability. The only structural limitation on network configuration is that it should be of the feed-forward type. Due to these minimal restrictions, the adaptive network's applications are immediate and immense in various areas.

If it is assumed that the fuzzy inference system (FIS) under consideration has multiple inputs and one functional output (f) (as is the case for this work) then the fuzzy rule-based algorithm may be represented in the first order Sugeno form as shown in Equation (3.6):

$$\begin{aligned}
&\text{Rule 1 : If x is } A_1 \text{ and y is } B_1 \text{ then } f_1 = p_1 x + q_1 y + r_1 \\
&\text{Rule 2 : If x is } A_2 \text{ and y is } B_2 \text{ then } f_2 = p_2 x + q_2 y + r_2 \\
&\quad : \quad : \quad : \quad : \quad : \quad : \quad : \\
&\quad : \quad : \quad : \quad : \quad : \quad : \quad : \\
&\text{Rule n : If x is } A_n \text{ and y is } B_n \text{ then } f_n = p_n x + q_n y + r_n
\end{aligned} \tag{3.6}$$

The corresponding ANFIS architecture being shown in Figure 3.2.

The node functions in the same layer are of the same function family as described by the following:

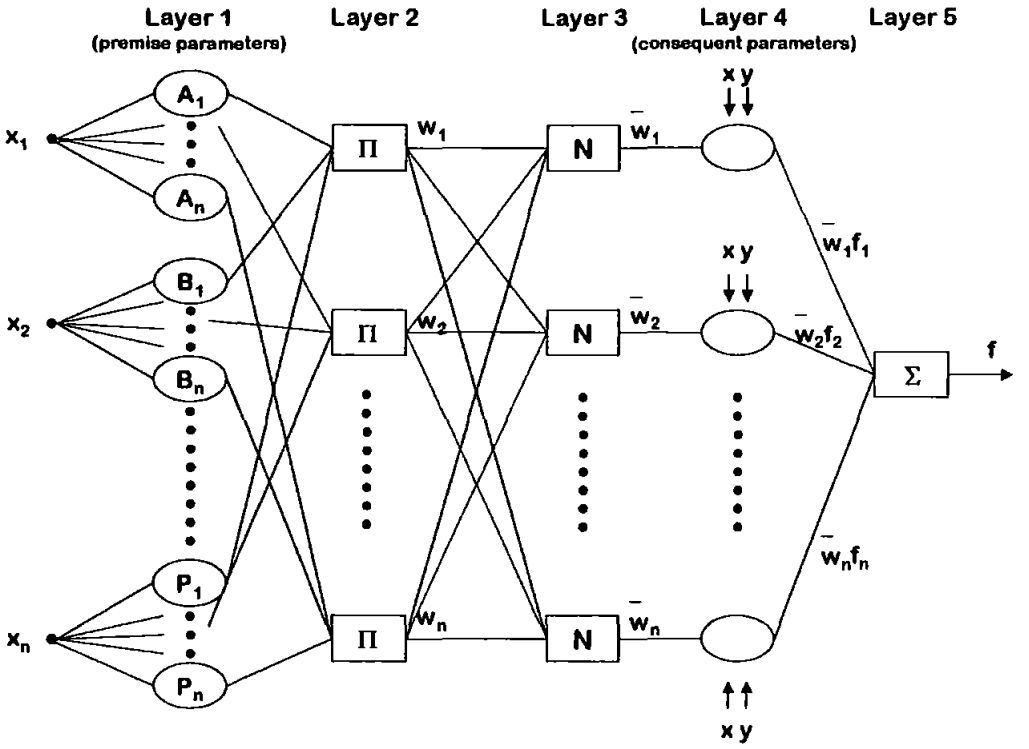


Figure 3.2 The ANFIS Architecture

Layer 1: Every *ith* node in this layer is an adaptive node with a node output defined by :

$$O_{1,i} = \mu_{A_i}(x) \tag{3.7}$$

where x is the input to the general node and A_i is the fuzzy set associated with this node. One possibility is that A_i is characterised by the generalised bell function:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}} \tag{3.8}$$

where $\{a_i, b_i, c_i\}$ is the parameter set. Parameters in this layer are referred to as *premise parameters*.

Layer 2: Every node in this layer is a fixed node labelled Π, which multiplies the incoming signals and outputs the product or T-norm operator result, for example:

$$O_{2,i} = w_i = \mu_{A_i}(x) \text{ x } \mu_{B_i}(y), \quad i = 1, 2 \tag{3.9}$$

Each node output represents the *firing strength* of a rule.

Layer 3: Every node in this layer is a fixed node labelled N. The *ith* node calculates the ratio of the *ith* rules' firing strength to the sum of all rules' firing strengths:

$$O_{3,i} = \bar{w}_1 = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{3.10}$$

For convenience, outputs of this layer are called *normalised firing strengths*.

Layer 4: Every *ith* node in this layer is an adaptive node with a node function:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \tag{3.11}$$

where $\bar{w}_i$ is the output of Layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are referred to as *consequent parameters*.

Layer 5: The single node in this layer is labelled $\sum$, which computes the overall output as the summation of incoming signals:

$$O_{5,i} = \text{overall output} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{3.12}$$

This has explained the basic structure used by the ANFIS. Now follows some information on the learning method used.

The training method used by the ANFIS is now described.

Rewriting the premise membership function Equation (3.8) as:

$$\mu_{ij}(x) = \frac{1}{1 + \left[\left(\frac{x - c_{ij}}{a_{ij}}\right)^2\right]^{b_{ij}}} \tag{3.13}$$

then Equation (3.13) now represents the *jth* membership function on the *ith* input universe of discourse. Therefore the learning rule for a general parameter may be described as follows:

$$\begin{aligned}\Delta\alpha_{ij} &= -\eta.\sum_{n=1}^{P}\frac{\partial E_n}{\partial O_{1n}^{i}}.\frac{\partial O_{1n}^{i}}{\partial \alpha_{ij}} \\ &= -\eta.\sum_{n=1}^{P}\frac{\partial E_n}{\partial O_{2n}^{ij}}.\frac{\partial O_{2n}^{ij}}{\partial O_{1n}^{ij}}.\frac{\partial O_{1n}^{ij}}{\partial \alpha_{ij}}.\end{aligned} \tag{3.14}$$

Hence, the learning rules for each individual parameter are:

$$\Delta a_{ij} = -\eta.\sum_{n=1}^{P}\frac{\partial E_n}{\partial O_{2n}^{ij}}.\frac{\partial O_{2n}^{ij}}{\partial O_{1n}^{ij}}.\left[\frac{2b_{ij}a_{ij}^{(2b_{ij}-1)}\left(x-c_{ij}\right)a_{ij}^{2b_{ij}}\left(x-c_{ij}\right)^{2b_{ij}}}{\left\{a_{ij}^{2b_{ij}}\left(x-c_{ij}\right)^{2b_{ij}}+\left(x-c_{ij}\right)^{2b_{ij}}a_{ij}^{2b_{ij}}\right\}^2}\right] \tag{3.15}$$

$$\Delta b_{ij} = -\eta.\sum_{n=1}^{P}\frac{\partial E_n}{\partial O_{2n}^{ij}}.\frac{\partial O_{2n}^{ij}}{\partial O_{1n}^{ij}}.\left[\frac{2a_{ij}^{2b_{ij}}\left(x-c_{ij}\right)^{2b_{ij}}a_{ij}^{2b_{ij}}\left(x-c_{ij}\right)^{2b_{ij}}\ln\left[\left|\frac{a_{ij}}{x-c_{ij}}\right|\right]}{\left\{a_{ij}^{2b_{ij}}\left(x-c_{ij}\right)^{2b_{ij}}+\left(x-c_{ij}\right)^{2b_{ij}}a_{ij}^{2b_{ij}}\right\}^2}\right] \tag{3.16}$$

$$\Delta c_{ij} = -\eta.\sum_{n=1}^{P}\frac{\partial E_n}{\partial O_{2n}^{ij}}.\frac{\partial O_{2n}^{ij}}{\partial O_{1n}^{ij}}.\left[\frac{-2b_{ij}a_{ij}^{2b_{ij}}\left(c_{ij}-x\right)^{(2b_{ij}-1)}a_{ij}^{2b_{ij}}\left(-\left(x-c_{ij}\right)\right)^{2b_{ij}}}{\left\{a_{ij}^{2b_{ij}}\left(-\left(x-c_{ij}\right)\right)^{2b_{ij}}+\left(c_{ij}-x\right)^{2b_{ij}}a_{ij}^{2b_{ij}}\right\}^2}\right] \tag{3.17}$$

The fuzzy consequent parameters being updated using a recursive least squares method. The main advantage of the ANFIS approach is its adaptive capability, which is advantageous when developing a fault tolerant control system. This approach has been used to tune FISs in later Chapters due to its successful use by Craven (1999) in developing the intelligent control system which is used within this work.

A fuzzy logic controller permits the set values to be altered depending on the size of the inputs. A non-linear plant, such as the AUV, may behave differently in extreme conditions. A fuzzy logic controller can effectively have a different set of control variables for each extreme case. The fuzzy logic control system used within this work is of this kind and it is called an ANFIS controller because this is the method used to tune the controller. ANFIS uses a set of input output data along with a neural network approach to tune the fuzzy rule base in-order create the best possible controller.

Two controllers were developed in this way for this AUV, one for controlling just the yaw channel, and a second for just the roll. As this thesis deals with work in both the yaw and roll channels the ANFIS controllers for both are given in Equations (3.18) and (3.19):

The yaw channel ANFIS controller,

$$
\begin{aligned}
&\text{If } \psi_\varepsilon \text{ is negative and } \dot{\psi} \text{ is negative} && \text{then } \delta = -0.4863\psi_\varepsilon - 0.8791\dot{\psi} - 0.02926\\
&\text{If } \psi_\varepsilon \text{ is negative and } \dot{\psi} \text{ is zero} && \text{then } \delta = -0.4890\psi_\varepsilon - 0.9021\dot{\psi} + 0.001381\\
&\text{If } \psi_\varepsilon \text{ is negative and } \dot{\psi} \text{ is positive} && \text{then } \delta = -0.4858\psi_\varepsilon - 0.8962\dot{\psi} + 0.003143\\
&\text{If } \psi_\varepsilon \text{ is zero and } \dot{\psi} \text{ is negative} && \text{then } \delta = -0.2994\psi_\varepsilon - 0.7034\dot{\psi} - 0.1227\\
&\text{If } \psi_\varepsilon \text{ is zero and } \dot{\psi} \text{ is zero} && \text{then } \delta = -0.4879\psi_\varepsilon - 0.8910\dot{\psi} + 0.003723 \quad (3.18)\\
&\text{If } \psi_\varepsilon \text{ is zero and } \dot{\psi} \text{ is positive} && \text{then } \delta = -0.3053\psi_\varepsilon - 0.3055\dot{\psi} - 0.03744\\
&\text{If } \psi_\varepsilon \text{ is positive and } \dot{\psi} \text{ is negative} && \text{then } \delta = -0.5902\psi_\varepsilon - 0.8387\dot{\psi} - 0.1172\\
&\text{If } \psi_\varepsilon \text{ is positive and } \dot{\psi} \text{ is zero} && \text{then } \delta = -0.4811\psi_\varepsilon - 1.081\dot{\psi} - 0.06111\\
&\text{If } \psi_\varepsilon \text{ is positive and } \dot{\psi} \text{ is positive} && \text{then } \delta = -0.6596\psi_\varepsilon - 1.311\dot{\psi} + 0.7814
\end{aligned}
$$

Where ψ_ε is the yaw angle error given by yaw angle demand minus actual yaw angle, $\dot{\psi}$ is the yaw rate and δ is the desired canard angle.

The roll channel ANFIS controller,

If ϕ_ε is negative and $\dot{\phi}$ is negative then δ_p = -0.4861ϕ_ε-0.8793$\dot{\phi}$-0.0289

If ϕ_ε is negative and $\dot{\phi}$ is zero then δ_p = -0.4887ϕ_ε-0.9022$\dot{\phi}$+0.0014

If ϕ_ε is negative and $\dot{\phi}$ is positive then δ_p = -0.4864ϕ_ε-0.8960$\dot{\phi}$+0.0030

If ϕ_ε is zero and $\dot{\phi}$ is negative then δ_p = -0.2992ϕ_ε-0.7034$\dot{\phi}$-0.1231

If ϕ_ε is zero and $\dot{\phi}$ is zero then δ_p = -0.4876ϕ_ε-0.8912$\dot{\phi}$+0.0044 (3.19)

If ϕ_ε is zero and $\dot{\phi}$ is positive then δ_p = -0.3045ϕ_ε-0.3063$\dot{\phi}$-0.0370

If ϕ_ε is positive and $\dot{\phi}$ is negative then δ_p = -0.5901ϕ_ε-0.8388$\dot{\phi}$-0.1174

If ϕ_ε is positive and $\dot{\phi}$ is zero then δ_p = -0.4805ϕ_ε-1.0810$\dot{\phi}$-0.0611

If ϕ_ε is positive and $\dot{\phi}$ is positive then δ_p = -0.6590ϕ_ε-1.3110$\dot{\phi}$+0.7808

Where ϕ_ε is the roll angle error given by yaw angle demand minus actual roll angle, $\dot{\phi}$ is the roll rate and δ_p is the desired plane angle.

3.3. IDENTIFICATION OF AUV DYNAMICS

The aim of this thesis is to produce fault tolerant controllers capable of handling sensor and actuator faults in the given AUV. One of the tasks that must be completed before work can begin to design and develop the fault tolerant controller is to produce identification modules. These modules will be used to detect if a fault has occurred in any of the AUV's systems. For this work two channels are being considered, the yaw channel and the roll channel. Two identification modules are required, the first need only detect failures within the yaw channel, and the second only within the roll channel. Firstly to consider a failure in the yaw channel, either a sensor or actuator fault, and the simplest way of identifying a fault is to have a model of the vehicle and compare outputs with the given AUV.

The process used to develop both models was identical, in the interests of brevity the development of only the yaw identification model will be shown. The objective of the

yaw identification module is to take the information from the controller and produce an estimate of the yaw and yaw rate of the AUV.

Three possible approaches were considered for producing a model of the AUV. The first was to use an Elman network, this is a recurrent artificial neural network (ANN) and so can be trained to recognise patterns in time as well as space. The second was the ANFIS method, this was chosen because the controller to be used in the AUV is of an ANFIS design, as previously discussed in section 3.2.3. A linear model was also considered as this is the more traditional approach to modelling a dynamic system. Three approaches were considered because of the non-linear nature of the AUV and it was not clear, in advance, which one would produce the best results.

All of this work has been carried out within the Matlab environment. Early tests of both the Elman network and ANFIS technique showed that they were far better at modelling yaw rate as opposed to yaw and therefore in this thesis the results are only shown for yaw rate as the output. With the yaw angles being obtained from the yaw rates simply by integration.

All of the work described in this thesis was carried out using the same conditions. For all of these simulations the AUV was given an initial speed of 7.5 knots and the thrusters were given a constant input of this velocity throughout. The AUV is controlled by a set of rudders positioned on its hull which are controlled by actuators. The yaw angle is controlled using two of these actuators (the upper and lower canards). These two canards can move to +/- 25.2 degrees at a maximum rate of 9.9 degrees per second. There is also a set of thrusters, which can be used for controlling yaw angle, but they are only used when the AUV is moving at a low velocity and therefore are not considered during this work.

3.3.1. Training and Testing Data

To create and develop all of the required models, training data was essential. This was obtained by using a specific input pattern in the open loop AUV set-up as shown in Figure 3.3. In the diagram input refers to the signal being given to the AUV and output refers to the yaw rate the AUV produces in response to this signal. The input used for training purposes is shown in Figure 3.4. The corresponding yaw rate for this can be seen in Figure 3.5. Also included is the yaw angles the AUV produced during this process seen in Figure 3.6. For the training path, a random input lasting for 1000

seconds was used. The random values did not exceed a modulus value of 25.2 degrees, this is because the canards used to control the yaw angle of the AUV cannot exceed these values. It is worth noting that the signal is permitted to remain at the maximum value long enough for the canards to reach the saturation levels.

Figure 3.3 Open Loop AUV Set-up

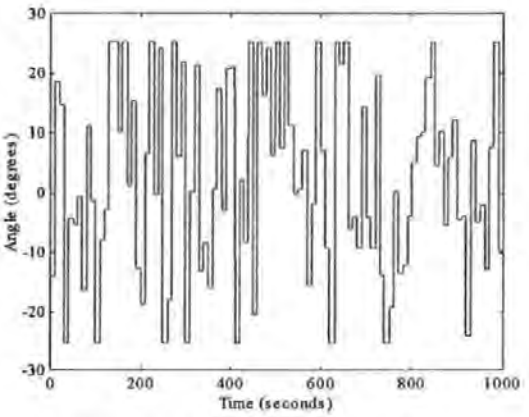


Figure 3.4 Random Input Used for Training

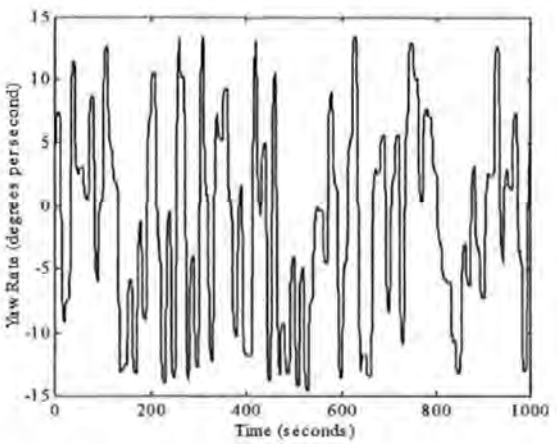


Figure 3.5 AUV Yaw Rate Response to Training Data

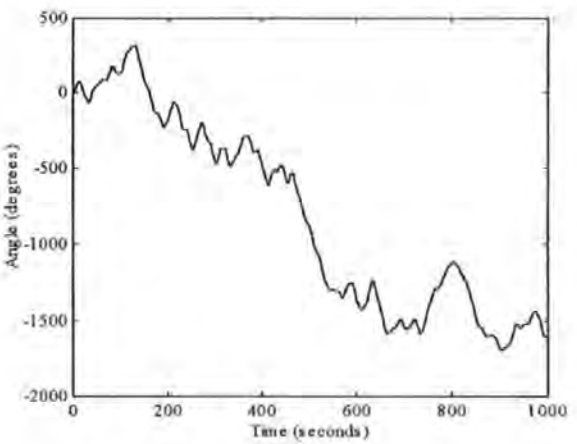


Figure 3.6 AUV Yaw Response to Training Data

To test the models further sets of data were required. A set of step input paths [Derradji and Mort (1996)] was selected to test the models.

Three different levels of step inputs were considered. A small value (5 degrees), a medium value (15 degrees), and a large value (25.2 degrees). These three simulations

were all run for 60 seconds with the step input occurring after 5 seconds. The input signals can be seen in Figure 3.7 with the corresponding yaw rates shown in Figure 3.8.

The large value was selected to be 25.2 degrees as this is the largest angle any of the canards can achieve and hence any larger value would be meaningless. Due to the symmetrical shape of the AUV it is unnecessary to consider negative input values when testing as these will produce the same responses, but in negative values.

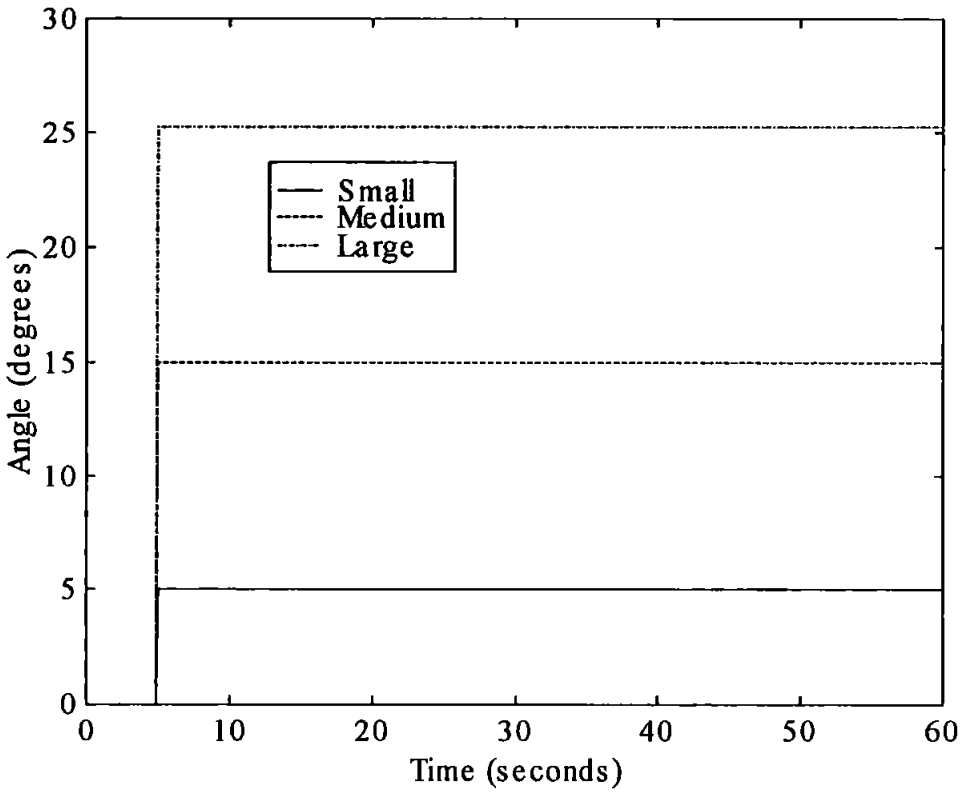


Figure 3.7 Step Input Signals for Testing Models

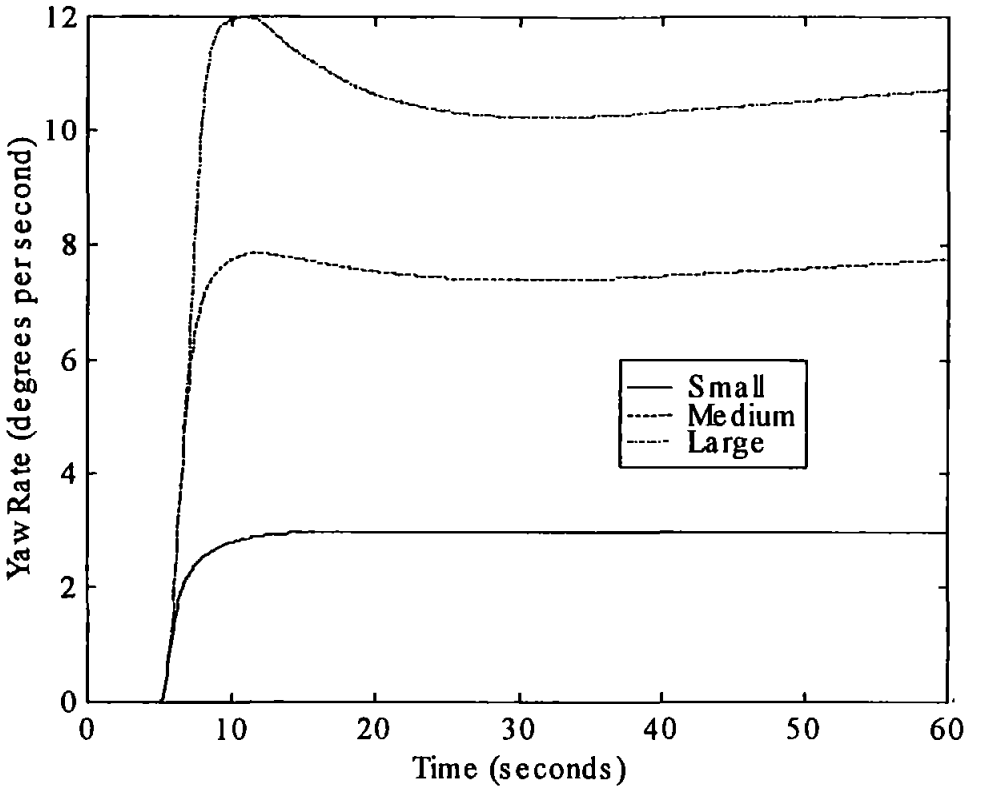


Figure 3.8 Corresponding Yaw Rates used for Testing Models

In the training of these models three possible inputs and one output were considered. All three inputs to be used were from the input data and the output used was the yaw rate of the AUV. This information was obtained by running the AUV along the training path in the Matlab/Simulink environment and recording the four sets of data. These four sets were the input data, the input data delayed by one second, the input data delayed by two seconds and, the yaw rate. All the data was recorded at one tenth of a second intervals to create 10001 pairs of input-output data.

3.3.2. Elman Networks

Elman networks [Warwick et *al* (1992)] are a form of ANN. This type network is made-up of two layers, a recurrent layer and an output layer. The training method used for this ANN is the backpropagation method with a momentum value of 0.95 and an adaptive learning rate with initial value of 0.01.

If the ANN repeatedly improves then the learning rate is increased to 105% of the previous value, however if the ANN is not improving the value is reduced to 70% of the previous value. This helps the ANN to find the best solution to the problem and remain at that setting. The momentum value is used to stop the ANN from getting stuck in any local minima it may encounter.

These two values are used along with the input-output data pairs to train the ANN using the backpropagation method. The backpropagation method will not be explained within this thesis, however many explanations of this procedure have been written, such as Haykin (1994).

As the AUV model is highly non-linear a recursive network is the best choice of ANN as they can handle patterns in time. The results for the model that produced the best results can be found in Appendix B.

3.3.3. Adaptive Neuro-Fuzzy Inference System (ANFIS)

Another approach considered was to use fuzzy logic to create a model of the AUV. It is a simple process to generate a FIS based on the AUV dynamics but these lack accuracy. Tuning can solve this problem, to tune the FIS an ANFIS [Jang (1991)] program was used within the Matlab environment. The ANFIS method tunes the FIS using a backpropagation algorithm [Haykin (1994)], which uses a set of input-output data. This is the same approach as was previously used for designing the control system as explained in section 3.2.3. The set of results for the model that produced the best results of this kind can again be seen in Appendix B.

3.3.4. Linear Models

The third method used was a state space linear model [Fairman (1998)]. The main reason for adopting this approach was that it is the classical approach to the modelling of a dynamic system such as the AUV. A second reason for adopting this approach is that a Kalman filter will be used in further work and a state space linear model is required for this. The mathematical form of the state space model used is shown in Equation (3.20).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{3.20}$$

Where **A** and **C** are the state vector coefficients matrices, **B** and **D** are the control vector coefficients matrices, **u** is the control input vector, **x** is the current state vector, $\dot{\mathbf{x}}$ is the new state vector and **y** is the output vector.

Again, there were already programs within the Matlab environment to generate and train state space models using given input-output data. The results for the model that produced the best results of this type can also be seen in Appendix B.

Several linear models of the AUV were obtained from DERA. These models were for the AUV moving with several different velocities. The control systems developed by Craven (1999), which are being used for this work, were designed for use with the AUV having a velocity of 7.5 knots. It was therefore logical to continue to work with the AUV having this velocity and therefore a 7.5 knots linear model was used as a benchmark for the models which have been developed. The results for this benchmark model using the training and testing data, are shown in Appendix B, and have been included purely for evaluation purposes.

3.3.5. Results and Comparisons

Having defined the types of models to be used and the methods used to implement them, the next stage was to decided on a testing method for the models. As the AUV operated in an open loop set-up when the training data was recorded, the models were all tested in exactly the same way to enable a fair comparison. This set-up is shown in Figure 3.9.

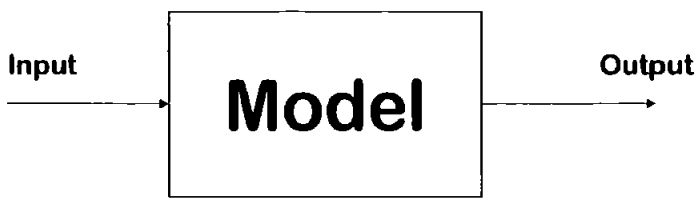


Figure 3.9 Open Loop Set-up for Testing of Models

Figure 3.9 shows the set-up used to obtain training and testing results for the models. An input signal is sent to the model which then produces its estimate for the current yaw rate, which is then integrated to give the yaw angle.

The statistical method used to evaluate the performance of the models was the root mean squared error (RMSE). This will give the average difference between the model and the AUV over the considered path. The equation used to calculate this is shown in Equation (3.21).

$$RMSE = \sqrt{\frac{\sum_{i=0}^{i=nrmse}(AUV_i - MOD_i)^2}{nrmse}} \tag{3.21}$$

Where $RMSE$ is the error value, $nrmse$ is the number of points measured, AUV_i is the information received from the vehicle at point i, and MOD_i is the information received from the model also at point i.

The first set of training was performed with the Elman network. The procedure for this was first to initialise a random Elman network and then to train it for 500 epochs. The training was then repeated until 2500 epochs was reached, with the sum-squared error (SSE) being recorded for later reference. The SSE was used as the Matlab program used this to evaluate the network. The units for SSE in this work are degrees per second squared. This process was repeated three times for each number of neurons in the recursive layer and input sets. The training was repeated with different starting values in order eliminate the possibility of a false result. The best model of each group of twelve was then tested over all of the training and testing paths in the open loop set-up as shown in Figure 3.9. The RMSE for both the yaw and yaw rate were recorded for each path and used to determine which of the models most closely approximated the AUV.

Examination of these results showed that the best Elman model was the one which had 32 neurons within its hidden layer, used all three input signals and had been trained for 1000 epochs. The model had the lowest errors when all four parameters were taken into account. The outputs of the Elman model, when attempting the medium step input (15 degrees), are shown in Figures 3.10 and 3.11.

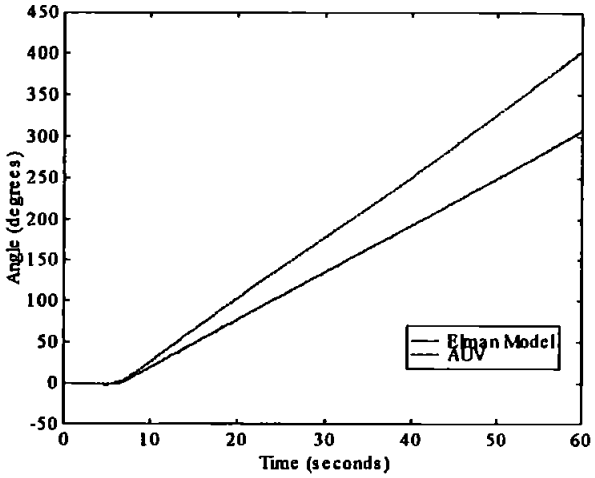


Figure 3.10 Elman Model Yaw Response to Testing Path

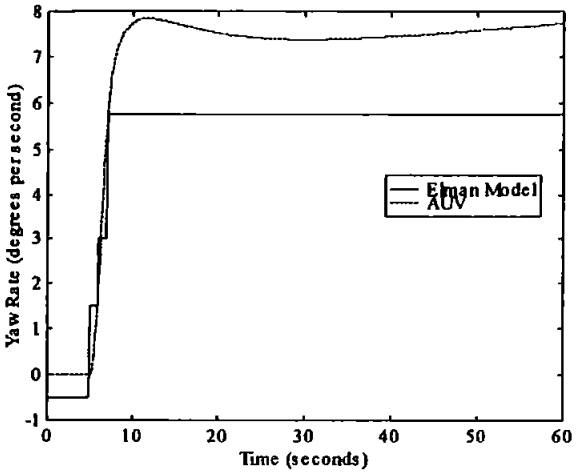


Figure 3.11 Elman Model Yaw Rate Response to Testing Path

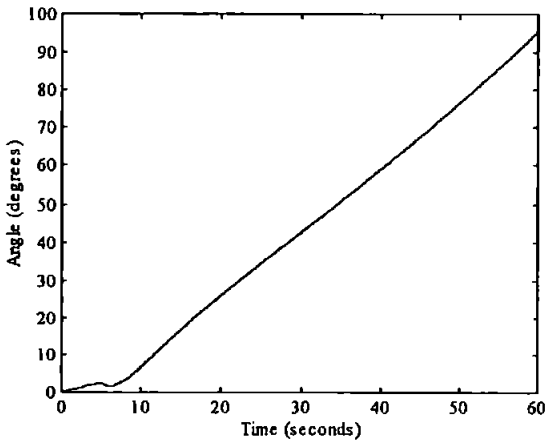


Figure 3.12 Yaw Error for Elman Model on Testing Path

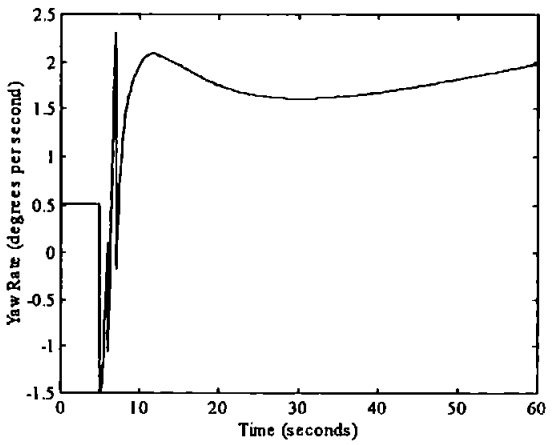


Figure 3.13 Yaw Rate Error for Elman Model on Testing Path

The error in the outputs were then calculated by taking the model outputs from the AUV outputs. These are shown in Figures 3.12 and 3.13. The second set of training was performed using the ANFIS approach. For this the procedure was to first initialise a FIS model using the 10001 pairs of training data and then train the model using the Matlab commands. After each training of 100 epochs the models were tested and the RMSE recorded. Due to the initial FIS being based on the training data, and not just a random starting point as used by the Elman network. It is acceptable to train each size

of model only once. The best model for this approach was fuzzy model which had a 27 fuzzy rule base using all three inputs and had been trained for 100 epochs, the outputs for the 15 degree input test can be seen in Figures 3.14 and 3.15.

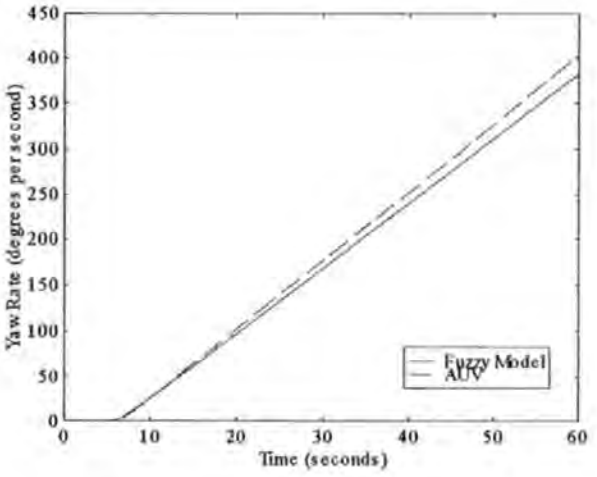


Figure 3.14 Fuzzy Model Yaw Response to Testing Path

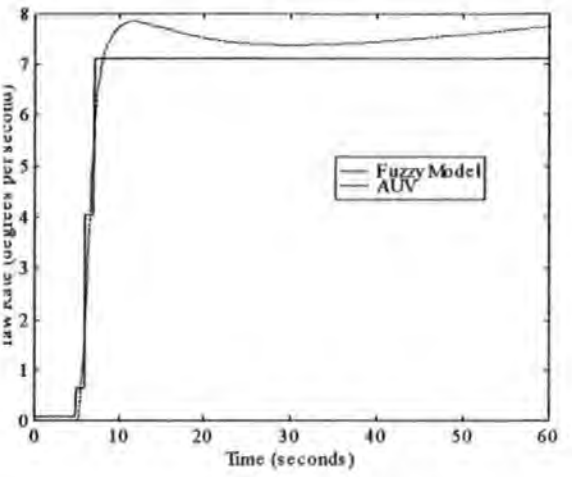


Figure 3.15 Fuzzy Model Yaw Rate Response to Testing Path

The error in the outputs were then calculated by taking the model outputs from the AUV outputs. These are shown in Figures 3.15 and 3.16.

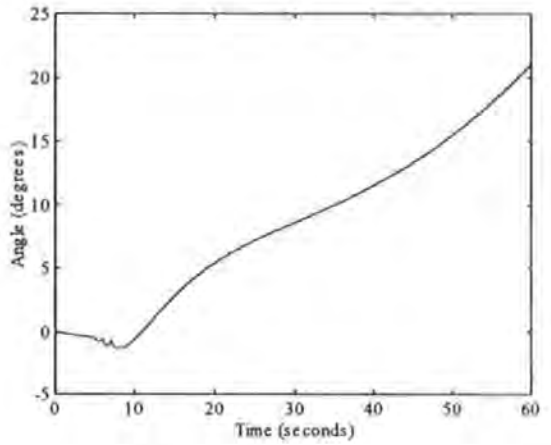


Figure 3.16 Yaw Error for Fuzzy Model on Testing Path

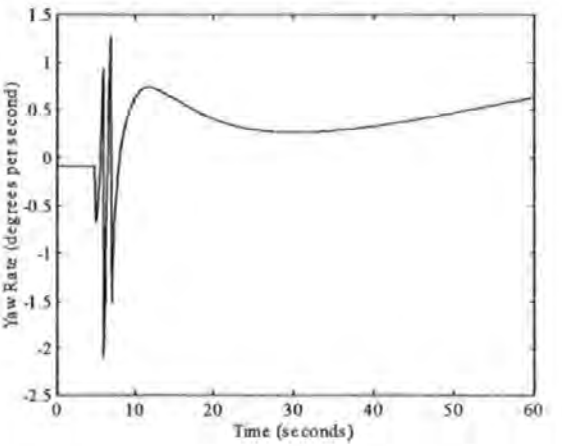


Figure 3.17 Yaw Rate Error for Fuzzy Model on Testing Path

For the state space linear model a different approach was used. The only input to the model used was the controller output. This time however, both outputs from the model were considered. Three types of model were trained one to produce a yaw output, another to produce a yaw rate output and a third which models both the yaw and yaw rate. The results from the most accurate state space linear model can be found in Appendix B. For the models which produced only one output (yaw or yaw rate) the other output was obtained by integration or differentiation. From the results the best model was one trained for 300 epochs, which produced both yaw and yaw rate as its outputs. The outputs of the linear state space model when presented with the medium step input test (15 degrees) can be seen in Figures 3.18 and 3.19.

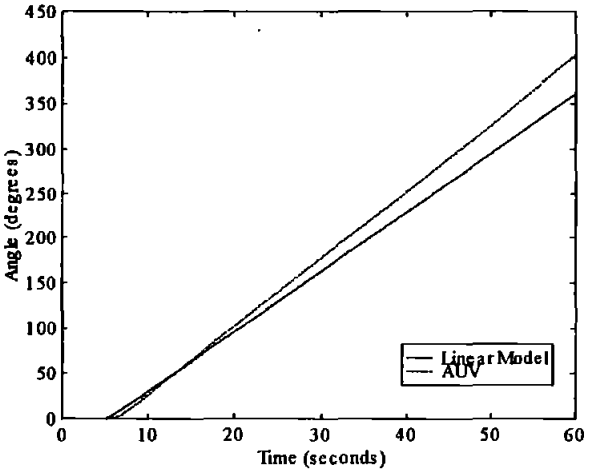


Figure 3.18 Linear Model Yaw Response to Testing Path

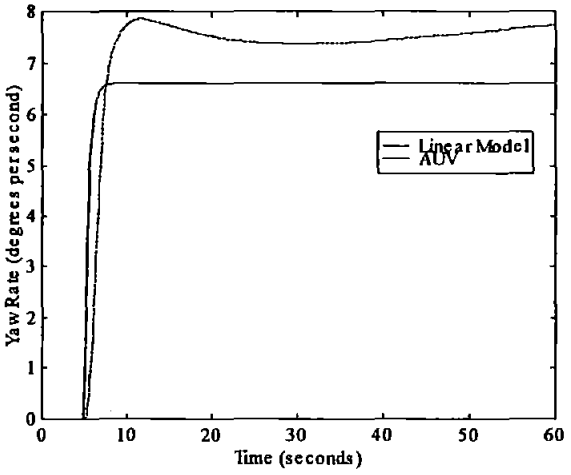


Figure 3.19 Linear Model Yaw Rate Response to Testing Path

The error in the outputs were then calculated by taking the model outputs from the AUV outputs. These are shown in Figures 3.20 and 3.21.

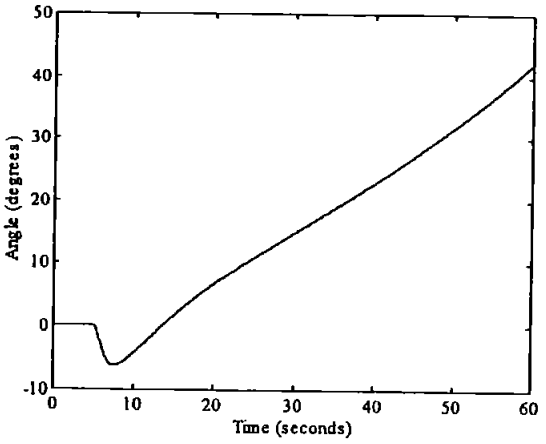


Figure 3.20 Yaw Error for Linear Model on Testing Path

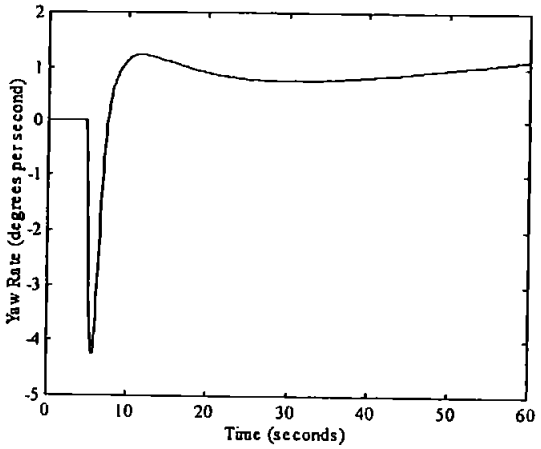


Figure 3.21 Yaw Rate Error for Linear Model on Testing Path

The DERA model was also used in the set-up shown, the results are shown along side the other model results in Appendix B. The DERA model was run over all of the training and testing paths and then the RMSE were calculated for both the yaw and yaw rates. These four values now provide a minimum standard that all the other models must improve upon in order to be considered for any subsequent work. The outputs of the closed loop DERA model being subjected to the medium step input testing path can be seen in Figures 3.22 and 3.23.

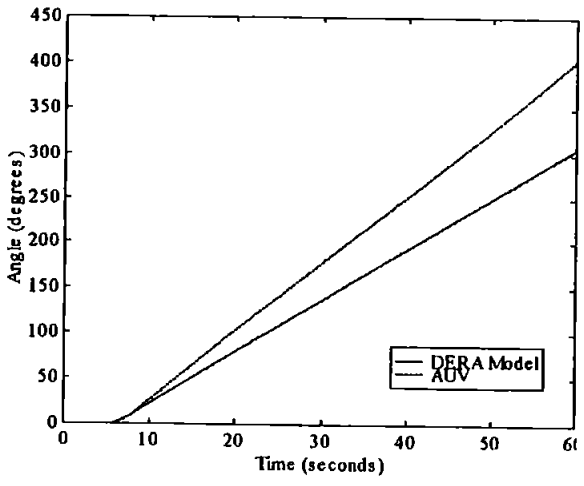


Figure 3.22 DERA Linear Model Yaw Response to Testing Path

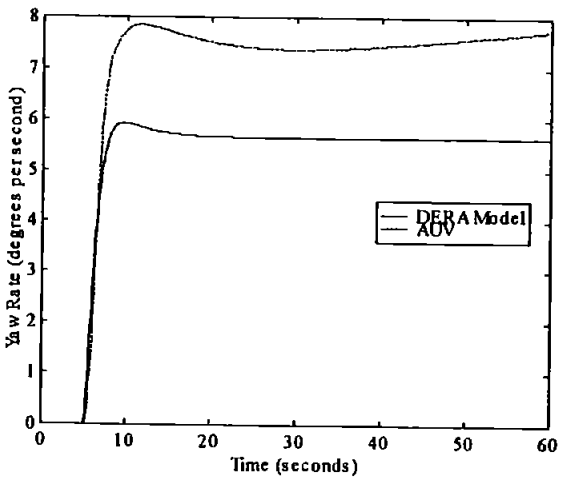


Figure 3.23 DERA Linear Model Yaw Rate Response to Testing Path

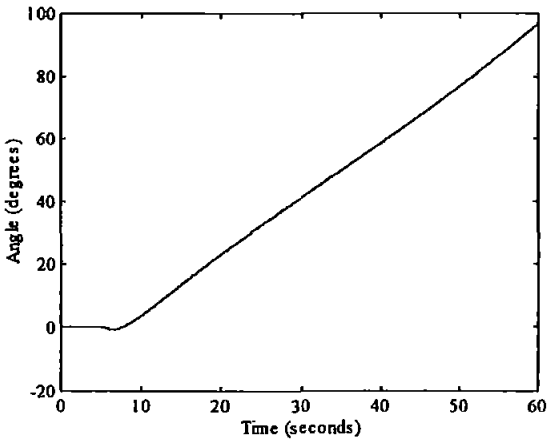


Figure 3.24 Yaw Error for DERA Linear Model on Testing Path

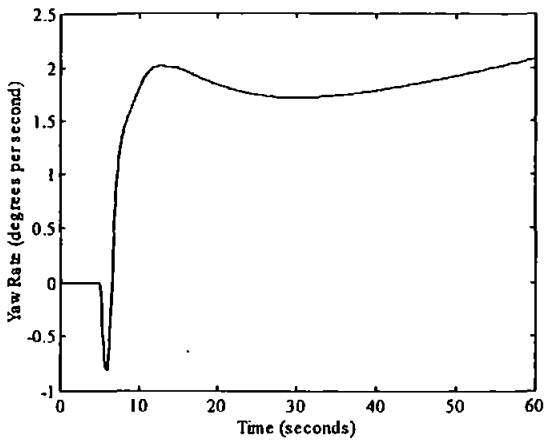


Figure 3.25 Yaw Rate Error for DERA Linear Model on Testing Path

The error in the outputs for these results were then calculated by taking the model outputs from the AUV outputs, which are shown in Figures 3.24 and 3.25.
All of the tests carried out within this work have the same error measurement (RMSE) in order to make comparison easier.

3.3.6. Summary of Models

Having tested the identification models, the best of each type will now be displayed. Due to the size of the Elman network (the matrix for the weights to the recursive layer is 32x35) weight matrices and bias vectors they are not displayed within this Chapter, but are shown in Appendix C.

The information for the ANFIS trained fuzzy logic model is given in Equation (3.22):

If α_1 is negative and α_2 is negative and α_3 is negative then

$$\zeta_f = 1.645\ \alpha_1 + 14.83\ \alpha_2 + 0.3387\ \alpha_3 - 1071$$

If α_1 is negative and α_2 is negative and α_3 is zero then

$$\zeta_f = 65.16\ \alpha_1 - 72.70\ \alpha_2 - 0.1930\ \alpha_3 - 345.9$$

If α_1 is negative and α_2 is negative and α_3 is positive then

$$\zeta_f = -278.4\ \alpha_1 + 277.4\ \alpha_2 + 0.3777\ \alpha_3 - 80.20$$

If α_1 is negative and α_2 is zero and α_3 is negative then

$$\zeta_f = -52.44\ \alpha_1+680.0\ \alpha_2+47.79\ \alpha_3+5944$$

If α_1 is negative and α_2 is zero and α_3 is zero then

$$\zeta_f = -6.693\alpha_1+31.74\ \alpha_2-5.210\ \alpha_3-724.3$$

If α_1 is negative and α_2 is zero and α_3 is positive then

$$\zeta_f = -42.68\ \alpha_1+82.69\ \alpha_2-3.733\ \alpha_3+335.8$$

If α_1 is negative and α_2 is positive and α_3 is negative then

$$\zeta_f = -128.5\ \alpha_1-6171\ \alpha_2+137.3\ \alpha_3-600.3$$

If α_1 is negative and α_2 is positive and α_3 is zero then

$$\zeta_f = 66.36\ \alpha_1-315.7\ \alpha_2-50.46\ \alpha_3+3918$$

If α_1 is negative and α_2 is positive and α_3 is positive then

$$\zeta_f = 3.297\ \alpha_1+83.73\ \alpha_2-114.4\ \alpha_3+1412$$

If α_1 is zero and α_2 is negative and α_3 is negative then

$$\zeta_f = 0.9743\ \alpha_1+42.29\ \alpha_2-50.81\ \alpha_3-379.1$$

If α_1 is zero and α_2 is negative and α_3 is zero then

$$\zeta_f = 30.50\ \alpha_1-96.33\ \alpha_2+9.961\ \alpha_3-39.14$$

If α_1 is zero and α_2 is negative and α_3 is positive then

$$\zeta_f = -97.43\ \alpha_1+57.57\ \alpha_2+12.16\ \alpha_3-481.1$$

If α_1 is zero and α_2 is zero and α_3 is negative then

$$\zeta_f = -22.26\ \alpha_1+39.42\ \alpha_2+8.172\ \alpha_3-628.9$$

If α_1 is zero and α_2 is zero and α_3 is zero then (3.22)

$$\zeta_f = -3.354\ \alpha_1+2.579\ \alpha_2-1.362\ \alpha_3-71.23$$

If α_1 is zero and α_2 is zero and α_3 is positive then

$$\zeta_f = -18.74\ \alpha_1+29.94\ \alpha_2-1.250\ \alpha_3+110.0$$

If α_1 is zero and α_2 is positive and α_3 is negative then

$$\zeta_f = -86.09\ \alpha_1-229.2\ \alpha_2-75.17\ \alpha_3+3235$$

If α_1 is zero and α_2 is positive and α_3 is zero then

$$\zeta_f = 27.92\ \alpha_1 - 117.0\ \alpha_2 + 9.974\ \alpha_3 + 852.0$$

If α_1 is zero and α_2 is positive and α_3 is positive then

$$\zeta_f = 1.191\ \alpha_1 - 5.780\ \alpha_2 + 0.3952\ \alpha_3 + 184.4$$

If α_1 is positive and α_2 is negative and α_3 is negative then

$$\zeta_f = 3.174\ \alpha_1 + 644.4\ \alpha_2 - 638.2\ \alpha_3 + 109.0$$

If α_1 is positive and α_2 is negative and α_3 is zero then

$$\zeta_f = -77.02\ \alpha_1 + 99.43\ \alpha_2 - 130.4\ \alpha_3 - 605.3$$

If α_1 is positive and α_2 is negative and α_3 is positive then

$$\zeta_f = -71.16\ \alpha_1 - 1183\ \alpha_2 - 99.40\ \alpha_3 - 456.5$$

If α_1 is positive and α_2 is zero and α_3 is negative then

$$\zeta_f = 80.79\ \alpha_1 - 42.12\ \alpha_2 + 108.3\ \alpha_3 + 752.7$$

If α_1 is positive and α_2 is zero and α_3 is zero then

$$\zeta_f = 7.474\ \alpha_1 + 7.351\ \alpha_2 - 3.011\ \alpha_3 + 101.9$$

If α_1 is positive and α_2 is zero and α_3 is positive then

$$\zeta_f = 50.61\ \alpha_1 + 142.5\ \alpha_2 + 5.202\ \alpha_3 - 1870$$

If α_1 is positive and α_2 is positive and α_3 is negative then

$$\zeta_f = -96.67\ \alpha_1 + 79.50\ \alpha_2 + 9.073\ \alpha_3 + 798.6$$

If α_1 is positive and α_2 is positive and α_3 is zero then

$$\zeta_f = -48.07\ \alpha_1 + 42.92\ \alpha_2 + 1.317\ \alpha_3 + 187.8$$

If α_1 is positive and α_2 is positive and α_3 is positive then

$$\zeta_f = 2.290\ \alpha_1 + 3.913\ \alpha_2 + 1.784\ \alpha_3 + 220.1$$

Where α_1 is the output from the ANFIS controller, α_2 is the output from the ANFIS controller delayed by one second, α_3 is the output from the ANFIS controller delayed by two seconds, and ζ_f is the fuzzy logic models estimate for the AUV's yaw rate.

The linear model used is of the from in Equation (3.20) with values:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -2.1326 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ -0.9398 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

With a state vector $\mathbf{x} = [\psi \quad \dot{\psi}]^T$, a control vector $\mathbf{u} = \delta_{br}$ the actuator input controlling the yaw motion and an output vector $\mathbf{y} = [\psi \quad \dot{\psi}]^T$.

For completeness the DERA linear model is also included and just as for the linear model is of the form shown in Equation (3.20):

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1.178 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.178 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.178 & 0 & 0 & 0 \\ -2.12 & 0 & 0 & -0.135 & -0.125 & 1.99 & -0.745 & 1.50 & -0.434 \\ -0.000286 & 0 & 0 & -0.700 & 0.685 & 0.000268 & -0.00320 & -1.48 & -0.184 \\ -0.0298 & 0 & 0 & -0.623 & -0.580 & 0.0216 & -0.0293 & -0.805 & -0.413 \end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1.1779 & 0 \\ 1.1779 & 0 & 0 \\ 0 & 0 & 1.1779 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C}=\begin{bmatrix} 0 & 0 & 57.296 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 57.296 & 0 \end{bmatrix}$$

$$\mathbf{D}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

With a state vector $\mathbf{x}=[\phi \quad Y \quad \psi \quad \delta_{sr} \quad \delta_{cr} \quad \delta_{br} \quad p \quad r \quad v]^T$ where δ_{cr} is the actuator input controlling the sway motion, and δ_{sr} is the actuator input controlling the roll motion, a control vector $\mathbf{u}=[\delta_{sr} \quad \delta_{cr} \quad \delta_{br}]^T$ and an output vector $\mathbf{y}=[\psi \quad \dot{\psi}]^T$.

3.3.7. Model Selected

On the whole all of the models presented thus far have outperformed the DERA model in the open-loop simulations detailed in section 3.3.5 and therefore using any of these for further work would be acceptable.

For the test results displayed in section 3.3.5 the DERA model had a final error value of approximately 100 degrees for the yaw angle and a yaw rate error of just above 2 degrees per second, after only 60 seconds. The Elman ANN model used failed to show much improvement over this step size. It did however manage to improve on the DERA model when the small step size was being used, but performed very poorly on the largest step size. The fuzzy logic model used to produce the results within this report performed closest to the actual AUV model and only once produced a result inferior to the DERA model. The linear model used herein also produced an impressive set of results again only once failing to improve on the DERA model being used as a benchmark.

From examining the results presented within Appendix B it is clear that the model which should be used for further work is the fuzzy logic model. The only concern is that this model produces a small yaw rate (approximately 0.1 degrees per second) when a zero signal is applied. This can clearly be seen by examining the first five seconds of Figure 3.15. This will produce a considerable error whenever the model is used for an extended period of time.

The only model where this did not occur was the linear model. This combined with the fact that the results for the linear model were of a similar standard to the fuzzy logic model, have led to the conclusion that the best model for the further work being considered for this thesis, will be the linear model.

3.3.8. Closed Loop Responses

Having tested all of the identification models as open loop systems and deduced which of them responded closest to the AUV, the models were placed in a closed loop system with the ANFIS controller to test their response. The responses of the best of each type of model are shown in Figure 3.26 along with the DERA model and the response of the AUV.

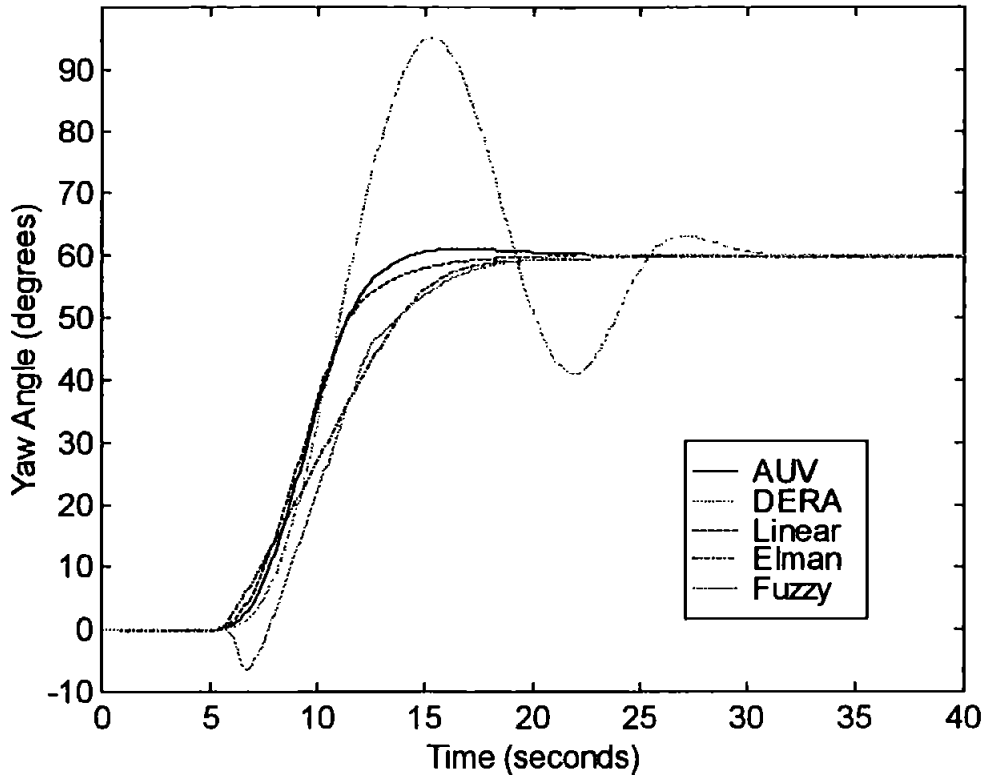


Figure 3.26 Yaw Responses for a Closed Loop 60 Degrees Step Input

From this graph it is clear to see that the linear model performed better than either the fuzzy logic model or the Elman ANN. The poor performance of the DERA model is highlighted in this figure, with its overshoot of 35 degrees and its highly oscillatory response.

3.3.9. Roll Model

The process used to develop the yaw models was then repeated to produce models for the roll channel. Again three types were considered, the ANFIS fuzzy logic model, the Elman ANN model and the state space linear model. After testing in both open and closed loop cases the best linear model again proved to be the closest approximation to the given AUV model. Once again the open and closed loop test also showed that the best of each model outperformed the given DERA model. The closed loop test results for an initial roll angle of 5 degrees are show in Figure 3.27. Therefore the linear model will be used for further work. This model is of the form in Equation (3.20) with values:

$$\mathbf{A} = \begin{bmatrix} 0.2210 & 1.1462 \\ -2.0047 & -0.9984 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -0.2252 \\ 1.7977 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

With a state vector $\mathbf{x} = \begin{bmatrix} \phi & \dot{\phi} \end{bmatrix}^T$, a control vector $\mathbf{u} = \delta_{sr}$ and an output vector $\mathbf{y} = \begin{bmatrix} \phi & \dot{\phi} \end{bmatrix}^T$.

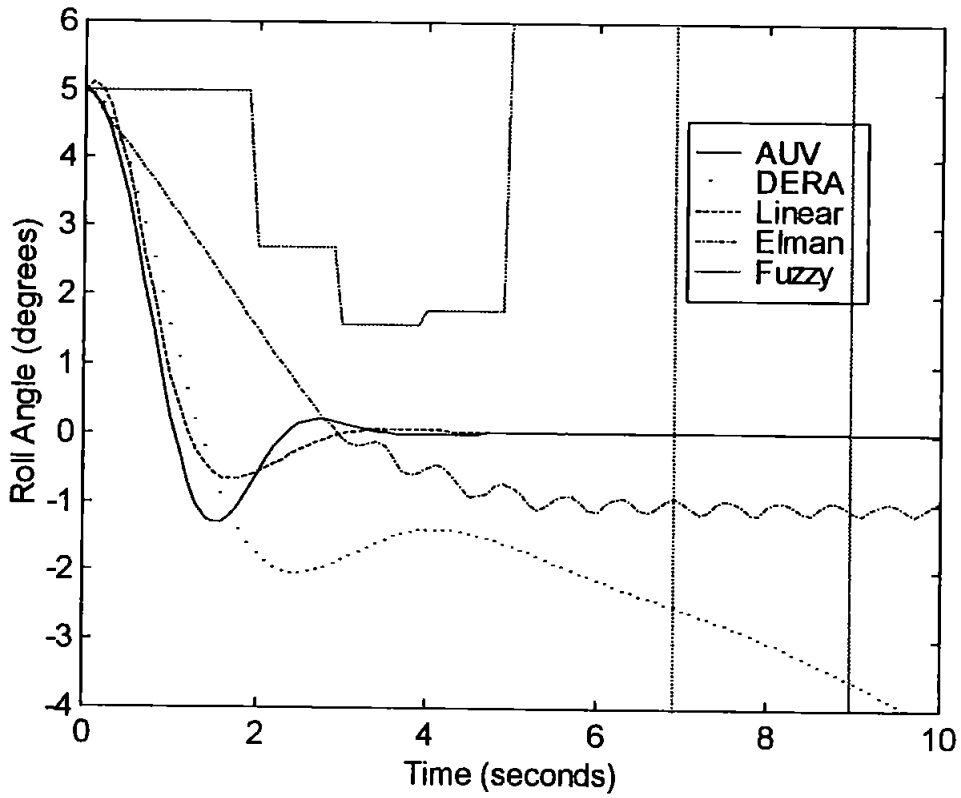


Figure 3.27 Roll Responses for a Closed Loop Initial Roll Angle of 5 Degrees

Figure 3.27 demonstrates the typical responses of the models when the ANFIS roll controller is placed within the control loop. The linear model is able to both improve

on the DERA linear model and achieve the required final angle, where as the ANFIS model becomes unstable and Elman ANN model produces a steady state oscillating error.

3.4. FAULT SET-UP

Having designed several identification models, the next step was to decide on the types of fault that will occur and how they will be implemented. Two types of faults are being considered in this thesis, the first fault occurring in the sensor feedback and the second occurring in the actuators which position the canards.

3.4.1. Sensor Faults

From a survey of work on the subject of sensor failure, the following types of faults were selected:

(1) Percentage signal failure [Napolitano et *al* (1998) and Yang et *al* (1999)]

(2) Intermittent signal failure [Mclean et *al* (1997)]

(3) Noise on the sensor [Fossen and Fjellstad (1995)].

For the yaw channel all three types of sensor failure were considered for each sensor (yaw and yaw rate) individually over three step input demands (10, 50, and 90 degrees). While for the roll channel all three types of sensor failure were considered for each sensor (roll and roll rate) individually, but instead of a demanded step input, three initial roll angles were used (5,15, and 25 degrees) with the demanded roll angle being zero degrees..

The first type of failure involved placing a gain on the sensor feedback to permit only a given percentage of the signal to return to the controller. The percentages considered were 100, 75, 50, 25 and 0. Where zero is total failure and 100% is no fault. This was achieved within the Matlab environment by setting up the system shown in Figure 3.28 and replacing the 1 value with the relevant value in the gain block. For the roll channel work the system is identical but roll replaces yaw.

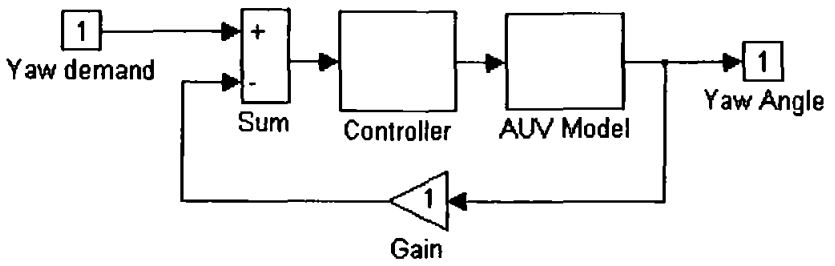


Figure 3.28 Gain on Sensor Feedback for Yaw Channel

The second type of failure involved using a random signal to intermittently create a total signal failure on the given sensor. This sensor failure was implemented within the Matlab model as shown in Figure 3.29. For the roll channel work the system is identical but roll replaces yaw.

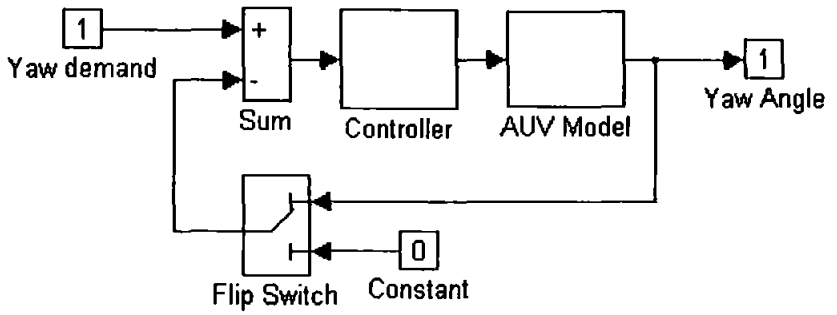


Figure 3.29 Flip Switch Between Total and Zero Failure on Yaw Sensor

The third type of sensor failure to be considered is simply adding noise to the signal. Adding band-limited white noise to the sensor output before it has reached the controller accomplished this. Figure 3.30 shows how this sensor fault was simulated. This third type of fault is not a true fault as all sensors suffer from some level of noise and this has been included as all types of controller should be able to cope with noise. The noise shown in Figure 3.30 and used within this study has a pseudo-random normally distributed (Gaussian) value, with a mean of zero and a variance of one determined by the seed value of 23341 (default value within Matlab Simulink block). A sample time of two seconds and a noise level which is a percentage of maximum sensor signal, know as the signal to noise ratio (SNR) [Anderson and Edmonson (1997)]. For the roll channel work the system is identical but roll replaces yaw.

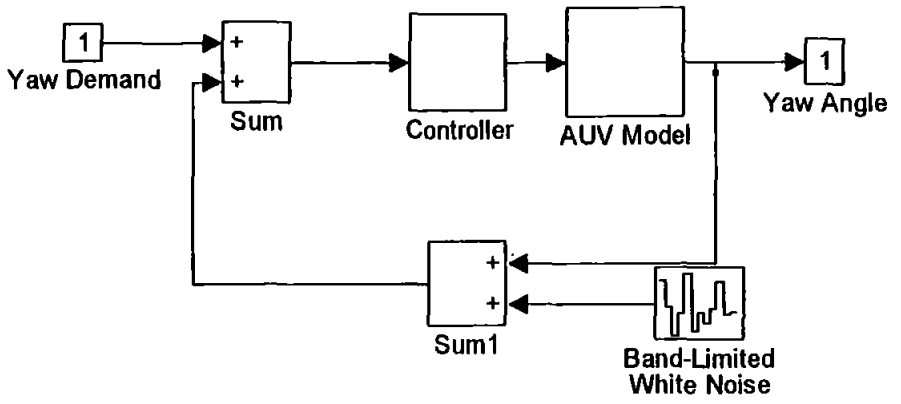


Figure 3.30 Noise Added to Yaw Sensor Feedback

3.4.2. Actuator Faults

Also to be considered are actuator failures. For this thesis only actuator failures in the yaw channel are considered. From a literature survey of the subject, the following types of faults were used:

(1) Total actuator failure at a zero angle [Looze et *al* (1985)]

(2) Actuator loss of effectiveness (LOE) [Derradji and Mort (1996)]

Both of these failures were considered for one of the control surfaces used by the control system, when being subjected to three step inputs (10, 20, and 30 degrees). Within the AUV model the actuators are formed using three blocks, a rate limiter, a saturation block and a transfer function, as shown in Figure 3.31. This is where all of the actuator failures will be implemented.

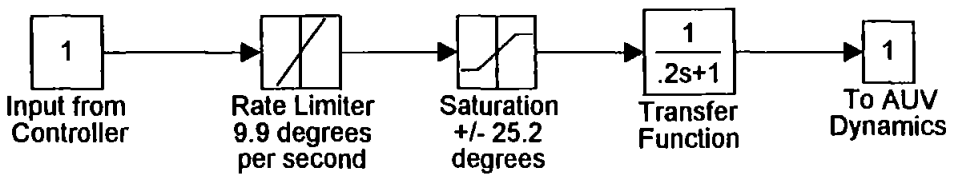


Figure 3.31 Actuator Simulator Within Matlab AUV Model

The first type of failure is total failure of one control surface at zero degrees. This failure has been chosen to simulate the rudder becoming locked and thus being unable

to move. This failure was accomplished by setting the input from the controller (shown in Figure 3.31) to zero.

The second type of failure used is one whereby one of the two control surfaces suffers a LOE [Derradji and Mort (1996)]. The upper canard was the actuator chosen for the faults to be simulated within. The percentages considered were 75, 50, and 25 LOE. Where 100 would be total LOE and zero would be no fault. There are three types of failure which can be implemented using this approach for actuator failures. These failures can be achieved by altering the values within the saturation and the rate limiter blocks shown in Figure 3.31. For this work the actuator failures were simulated firstly by altering only the value in the saturation block, secondly by altering only the value in the rate limiter block, and thirdly by altering both values simultaneously to the same level of failure.

3.5. METHODS USED TO TUNE FUZZY INFERENCE SYSTEMS

Throughout this study three methods are used to tune the FISs used in the construction of the fault tolerant control systems. These three tuning methods are ANFIS [Jang (1991)], simulated annealing [Kirkpatrick, et *al* (1983)], and the tabu search [Denna et *al* (1999)]. This section contains a brief explanation of each one of these methods highlighting the advantages of each method and giving reasons why each one has been chosen.

3.5.1. ANFIS

The ANFIS [Jang (1991)] method has already been explained within this Chapter (see section 3.2.3.), and therefore the information will not be reiterated here.

3.5.2. Simulated Annealing

The simulated annealing method for tuning the FIS is a stochastic search method based on an analogy of a certain physical system and was first employed by Kirkpatrick et *al* (1983). Annealing is the process whereby a substance is initially heated to a high temperature and then allowed to cool gradually.

Energies of the algorithm adapted by Kirkpatrick et *al* (1983) described a Boltzman probability distribution as shown in Figure 3.32. Clearly, the probability of any given energy E, is an exponentially decreasing function of E.

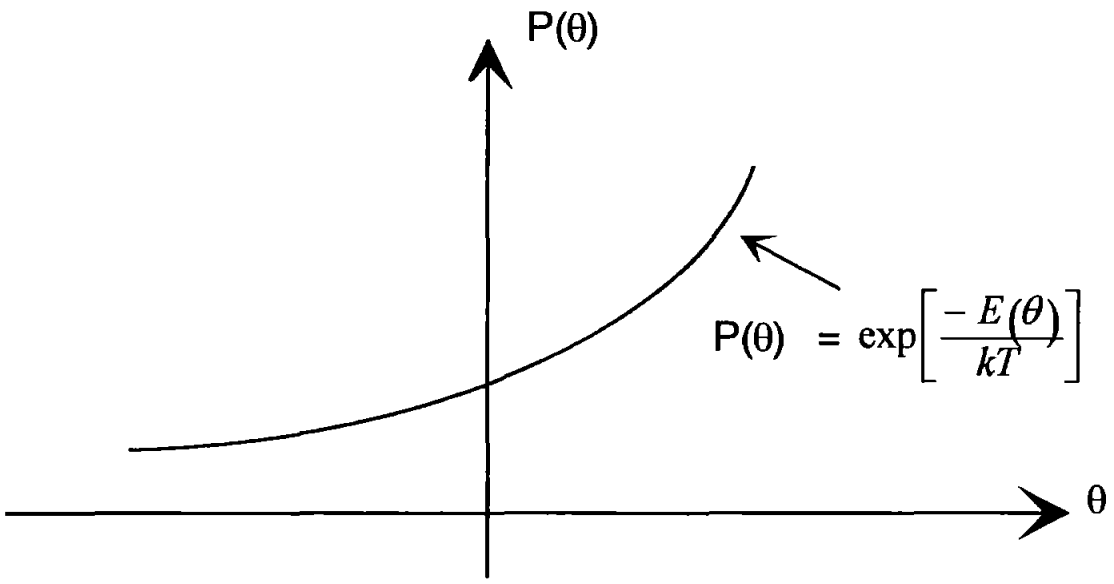


Figure 3.32 The Boltzman Probability Distribution

Thus, if a new matrix of parameters θ, (which have been perturbed by a randomly generated amount), leads to an improved performance of the system then they are accepted. The process is then repeated. However, if this new matrix leads to a worsened performance of the system the new parameters may occasionally be accepted with probability P(θ) such that:

$$P(\theta)=\exp\left[\frac{-E(\theta)}{kT}\right] \tag{3.23}$$

Where $E(\theta)$is the energy associated with the state θ, k is a constant and T is a temperature parameter which decays training according to:

$$T=\frac{T_0}{1+an_e}. \tag{3.24}$$

Where T_0 is the initial temperature, a is a constant which governs the rate of decay and n_e is the training epoch. By including this probability function (Equation (3.24)) the system is allowed to escape the local minima of the error hyper-surface.

The simulated annealing method has three clear steps. Firstly, a random change is induced in the system parameters. The new parameters are then tested. This will lead to the new parameters being accepted if they are an improvement or an improvement within the Boltzman probability distribution. Finally, this process will be stopped if there is sufficient evidence that the global minimum has been reached within the limits of a specified accuracy or when some pre-specified iteration number is reached. This method will, given enough time, converge to the best solution to a given problem due to the random search of the error surface. The simulated annealing algorithm is summarised in Table 3.1.

Table 3.1 The Simulated Annealing Algorithm.

1. Simulate the dynamic system using initial parameter set.
2. Perform random changes to the parameter set and then re-simulate the dynamic system.
3. If the performance of the system has improved then retain parameter set changes and re-apply.
4. If the performance of the system has degraded, compute the probability of accepting the poorer parameter set.
5. Generate random number between 0 and 1 and compare to probability from 4. If random number is less than probability, then accept poorer parameters; otherwise reject.
6. Re-simulate and return to 3 until convergence

The advantage of the simulated annealing method is that it does not rely on the backpropagation training algorithm, which can become trapped in local minimum of the error surface. The acceptance of slightly poorer results also gives the method more robustness when searching the error surface for the global minimum.

3.5.3. Tabu Search

A method previously used by Denna et *al* (1999) to tune FISs is the tabu search algorithm. This algorithm will be used in a later Chapter to tune fault tolerant FISs for use in the AUV, therefore there now follows an explanation of this method.

The tabu search method being that is being considered for tuning the fault tolerant FISs is an iterative process. It also takes advantage of the stochastic properties of the simulated annealing method. The algorithm can splint into three individual steps. The first step is called the move and is where the FIS is altered. The second step is the tabu list, this is where the altered FIS is compared to the list of previous FISs. The third step is the application and is where the altered FIS is placed in the control loop and tested.

For this work there are two possible moves which can take place. The first to be considered is that of a random move, where the current FIS is randomly altered to create the trial FIS. Mathematically this is show in Equation (3.25).

$$\mathbf{X}_{Trial} = \mathbf{X}_{Current} + \boldsymbol{\Omega} \tag{3.25}$$

Where $\mathbf{X}_{Trial}$ is the vector of parameters defining the altered FIS, $\mathbf{X}_{Current}$ is the vector of parameters defining the current FIS and $\boldsymbol{\Omega}$ is the vector of random changes made to the parameters. It should be noted that all three vectors must be of the same length.

The second to be considered is that of a calculated move, where the current FIS is proportionately altered to create the trial FIS. Mathematically this is shown in Equation (3.26).

$$\mathbf{X}_{Trial} = \mathbf{X}_{Current} + \Delta\boldsymbol{\Xi} \tag{3.26}$$

Where $\mathbf{X}_{Trial}$ is the vector of parameters defining the altered FIS, $\mathbf{X}_{Current}$ is the vector of parameters defining the current FIS, Δ is a value to be determined and $\boldsymbol{\Xi}$ is the vector of previous changes made to the parameters in the last iteration. It should be noted that all three vectors must be of the same length.

The random move is used for the first iteration and any iteration when the trial FIS has already been tested and is hence on the tabu list. The proportionate move is used for all other iterations.

The tabu list is a key part of the algorithm and is where the name tabu search is derived from. The tabu list is a list of previous FIS that have already been tested for fault tolerance. In this work the list is updated at the end of every iteration with the newest FIS to be tested and an old FIS may be removed if it has been in the tabu list for a given number of iterations.

After a trial FIS has been created the next step is the application section. In this section the FIS is placed into the control loop and then the simulation is run. The results of this run are then compared to the best of the previous FIS results.

The complete tabu search algorithm for this work is defined as follows, it starts with a random change being made to the parameter set which is compared against the tabu list. This parameter set is then tested and if it leads to an improvement in performance it will be accepted. The next step is for the algorithm to calculate the best changes to make to parameter set, which are compared to the tabu list to ensure it has not been previously simulated. This parameter set is then tested and if it leads to an improvement in performance it will be accepted and the next changes will be calculated, if it does not lead to an improvement then a random change will be made. The tabu search algorithm is summarised in Table 3.2.

Table 3.2 The Tabu Search Algorithm.

1. Simulate the dynamic system using initial parameter set.
2. Perform random changes to the parameter set and then re-simulate the dynamic system.
3. If the performance of the system has improved then retain parameter set changes and calculate next changes.
4. If the performance of the system has degraded perform random changes to the parameter set.
5. Re-simulate and return to 3 until convergence

The random search part of the algorithm means that the best possible solution will be located, given enough time. The length of time required for this process is reduce by two processes the first being the proportional move and the second being the checking of previous tests. Also the lack of a probability function means that only an improvement in performance will be accepted. The advantages of the tabu search method are that it does not rely on the backpropagation training algorithm and it checks to make sure it is not repeating a parameter set which has been previously tested. Also the calculation by the algorithm to identify the best changes, means it will explore local minimum to the full before randomly moving position on the error hyper-surface.

3.6. CONCLUSIONS

This Chapter began by presenting background information on the given UUV model and its associated dynamic equations. The standard PD controller and ANFIS controllers developed by Craven (1999) were also discussed.

Then the three methods used for developing the identification models were explained and several of the best results were presented. Having clarified the methods behind the models and compared the open-loop and closed-loop responses the best model for further work within the yaw channel case was chosen. This model was the linear model. The methods were then repeated to create models for the roll channel case. The best model for the roll channel was also a linear state space model.

There was then an explanation of the types of sensor and actuator faults to be used in later Chapters. It was also shown how each of these faults are implemented within the Matlab Simulink environment.

Finally, the three methods to be used within this thesis to tune the FISs, for the fault tolerant control systems, were then presented.

The next Chapter deals with the Kalman filter and its approach to fault tolerant control. This provides benchmark control systems for both types of faults, which will be used later in this study.

CHAPTER 4

THE KALMAN FILTER APPROACH TO FAULT TOLERANT CONTROL

4.1. INTRODUCTION

The aim of this Chapter is to explain the Kalman filter and show how it has been applied to the fault tolerant control problem being considered in this AUV study. Results from simulations using both the sensor and actuator failures, discussed in the previous Chapter, are presented to show how the Kalman filter approach handles them. The sensor failures are considered for both cases, whereas the actuator failures are only considered in the yaw channel for reasons discussed in the previous Chapter. The identification models developed in Chapter 3 are used to form the basis of the Kalman filters to be used here.

4.2. KALMAN FILTER THEORY

In this section there will be a brief description of the basic concept of the Kalman filter. For the interested reader there are, however, books that look at Kalman filtering in far greater depth [Grewal and Andrews (1993) and Bozic (1979)].

A Kalman filter is an estimator, which uses a statistical method for removing noise from a linear dynamic model. In order for the Kalman filter to remove the noise from a system it requires several pieces of information. It requires the dynamics of the system, the type and level of noise, the initial Kalman gain matrix and the initial error for the sensors.

There are three major steps to the Kalman filter process. The first is to update the error coefficient matrix, by using the Riccati equation. Next the Kalman gain matrix is calculated. Finally the Kalman gain matrix is used with both the sensor signal and linear model outputs to calculate the best estimate of the sensor signal. Then time

moves on one step to the next position and the whole process is repeated to find the new best estimate of the sensor signals in a recursive manner.

As the Kalman filter updates the error coefficient matrix for every run, this will increase the accuracy of the matrix and hence produce a better Kalman gain matrix as it is used. This means that the longer the Kalman filter is used the better the estimate will become.

The three main equations of the Kalman filter process are shown in Equations (4.1), (4.2) and (4.3).

$$\mathbf{P}(t) = \mathbf{\Phi}(t-1)\mathbf{P}(t-1)\mathbf{\Phi}(t-1)^{\mathrm{T}} \tag{4.1}$$

$$\mathbf{KG} = \mathbf{P}(t)\mathbf{C}'(\mathbf{C}\mathbf{P}(t)\mathbf{C}'+\mathbf{R})^{-1} \tag{4.2}$$

$$\mathbf{X} = \mathbf{S} + \mathbf{KG}(\mathbf{L}-[\mathbf{S}]) \tag{4.3}$$

Where **P** is the error covariance matrix, Φ is the transition matrix, t is time, **KG** is the Kalman gain matrix, **C** is the output matrix from the linear model (and for this work is the identity matrix), **C'** is the inverse of **C** (and for this work also the identity matrix), **R** is the initial error covariance matrix, **S** is the vector from the sensors, **L** is the vector from the linear model, and **X** is the best estimate output.

From the three Equations (4.1), (4.2) and (4.3) it is clear how important the error covariance matrix is to the Kalman filter. Therefore to help understand how the Kalman filter operates, the method used to calculate the error covariance matrix will be shown.

First recall that the linear model being used is of form shown in Equation (3.20) from Chapter 3. It follows that the state and the expected value can be described as in Equations (4.4) and (4.5) respectively.

$$\mathbf{X}_t = \mathbf{\Phi}_{t-1}\mathbf{X}_{t-1} + \boldsymbol{\omega}_{t-1} \tag{4.4}$$

$$\begin{aligned} \mathrm{E}\langle \mathbf{X}(t) \rangle &= \mathbf{\Phi}(t-1)\mathrm{E}\langle \mathbf{X}(t-1) \rangle + \mathrm{E}\langle \boldsymbol{\omega}(t-1) \rangle \\ \mathrm{E}\langle \mathbf{X}(t) \rangle &= \mathbf{\Phi}(t-1)\mathrm{E}\langle \mathbf{X}(t-1) \rangle \quad (as \quad \mathrm{E}\langle \boldsymbol{\omega}(t-1) \rangle = 0) \end{aligned} \tag{4.5}$$

Where ω is noise.

Therefore the equations for the covariance matrix can be derived as shown in Equation (4.6).

$$
\begin{aligned}
\mathbf{P}(t) &= \mathrm{E}\left\langle \left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t)\rangle\right]\left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t)\rangle\right]^{\mathrm{T}} \right\rangle \\
&= \mathrm{E}\left\langle \left[\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] + \boldsymbol{\omega}(t-1)\right]\left[\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] + \boldsymbol{\omega}(t-1)\right]^{\mathrm{T}} \right\rangle \\
&= \mathrm{E}\left\langle \begin{array}{l}
\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \\
\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\boldsymbol{\omega}^{\mathrm{T}}(t-1) + \\
\boldsymbol{\omega}(t)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}}(t-1) \\
+ \boldsymbol{\omega}(t-1)\boldsymbol{\omega}^{\mathrm{T}}(t-1)
\end{array} \right\rangle \\
&= \boldsymbol{\Phi}(t-1)\mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle \boldsymbol{\Phi}^{\mathrm{T}}(t-1) \\
&\quad + \mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\boldsymbol{\omega}^{\mathrm{T}}(t-1) \right\rangle \\
&\quad + \mathrm{E}\left\langle \boldsymbol{\omega}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle \boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \mathrm{E}\left\langle \boldsymbol{\omega}(t-1)\boldsymbol{\omega}^{\mathrm{T}}(t-1) \right\rangle \\
&\textit{as} \quad \mathbf{P}(t-1) = \mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle \\
&\textit{and} \quad \textit{letting} \quad \mathbf{Q}(t-1) = \mathrm{E}\left\langle \boldsymbol{\omega}(t-1)\boldsymbol{\omega}^{\mathrm{T}}(t-1) \right\rangle \\
&\mathbf{P}(t) = \boldsymbol{\Phi}(t-1)\mathbf{P}(t-1)\,\boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \mathbf{Q}(t-1)
\end{aligned}
\tag{4.6}
$$

As $\mathbf{Q}(t\text{-}1)=0$, this will then give the equation used within the Kalman filter program and shown in Equation (4.1). This has shown how the covariance matrix was derived from the second moment of the state.

For this project the Kalman filter will be used in a slightly different way. Instead of being used to filter noise from sensor feedback, it will attempt to compensate for failures within the AUV system. It will be using the linear model from the previous Chapter. The Kalman filter will, when there is no fault present, produce a best estimate between the AUV sensor information and the linear model output.

For the remaining sections of this Chapter the Kalman filter will be used as now described.

The Kalman filter used for work in the yaw channel will have two inputs (yaw and yaw rate) from the sensors and two inputs from the linear model (linear yaw and linear yaw rate). The Kalman filter then compares these two sets of information by

using the Riccati Equation (4.1) to update the previous error covariance matrix. The covariance matrix is then used in Equation (4.2) to update the Kalman gain matrix. This is then used by Equation (4.3) to produce a best estimate for the yaw and yaw rate of the AUV.

The information is then passed on to the ANFIS controller and in turn this sends a control signal to both the AUV and linear model, which respond accordingly. Which will change the yaw and yaw rate readings from both the sensors and linear model. This information is then sent on to the Kalman filter where the whole process is repeated.

The process can be seen in a flow chart form in Figure 4.1.

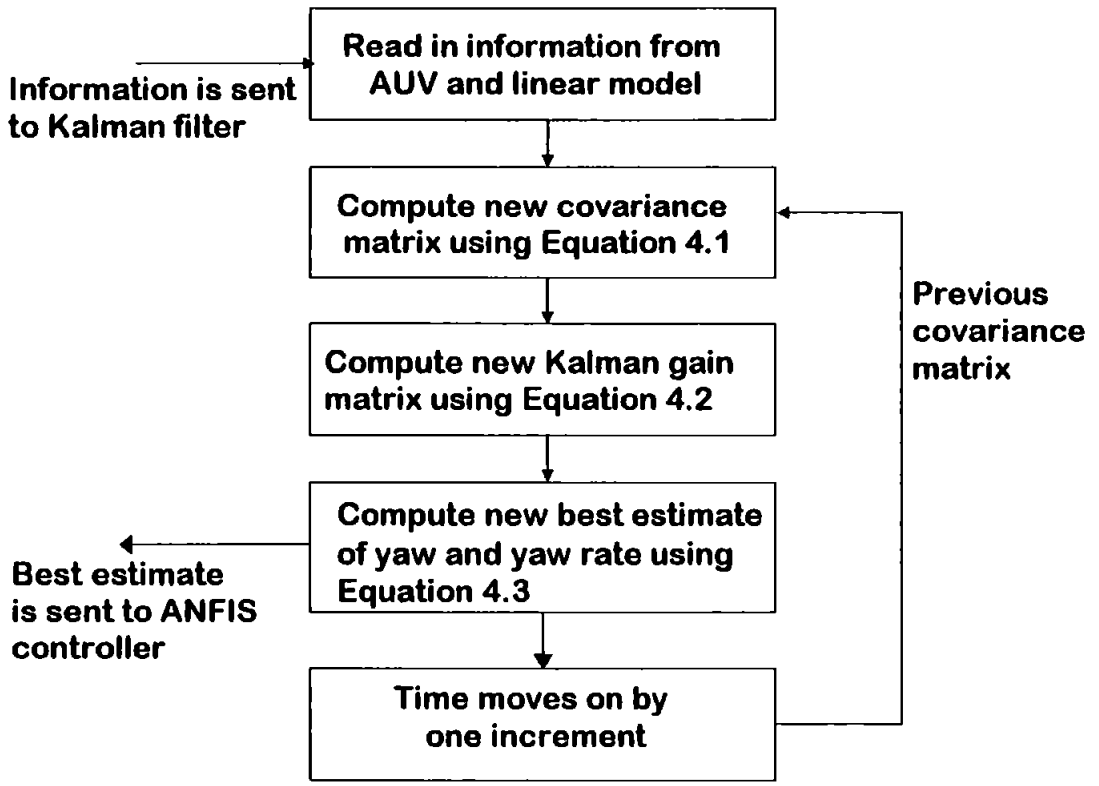


Figure 4.1 Kalman Filter Flow Chart

This concept was programmed within the Matlab environment using the information presented to produce a Kalman filter. The Kalman filter will be able to cope with certain types of failures within the AUV, as the remainder of this Chapter will show.

4.3. KALMAN RESPONSES TO SENSOR FAULTS

Having decided on the best models from the previous Chapter and written a Kalman filter program using the theories described in section 4.2, the next step was to simulate failures within the system. The Kalman filter program will be placed in the closed loop system as shown in Figure 4.2.

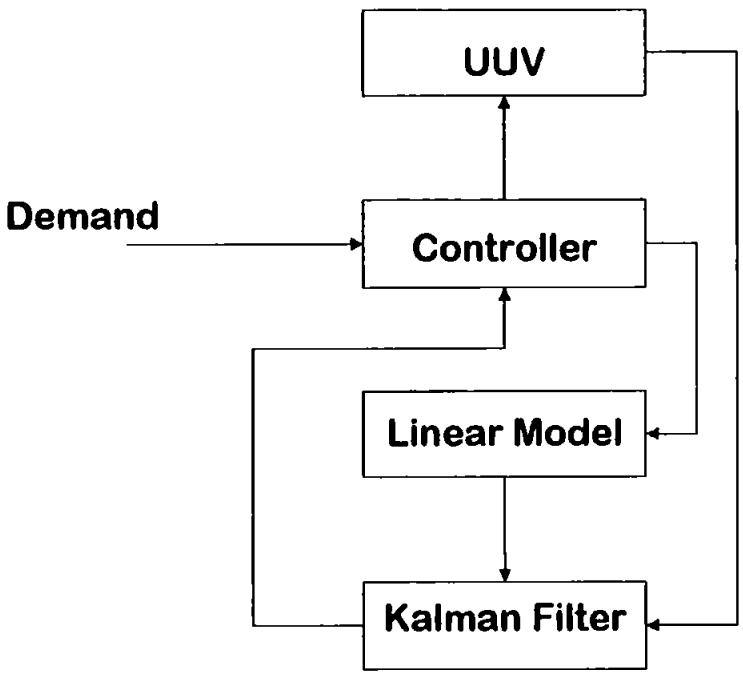


Figure 4.2 The Closed Loop Kalman Filter Control System

For the system shown in Figure 4.2 the Kalman filter is unaffected by the choice of controller used. The reason for this is that the linear model was trained using open loop input-output data and the information used for the Kalman filter was also open loop. This leaves a free choice as to what type of controller can be placed within the system. The ANFIS controllers developed by Craven (1999) have been used for this work.

Before the results could be obtained it was necessary to decide on initial values for both the Kalman gain matrix and the error coefficient matrix. The initial Kalman gain matrix was chosen to be zero, as this implies the Kalman filter will believe totally the output from the sensors. This is acceptable because it is reasonable to assume that at

the beginning of the simulation there are no faults with the information from the sensors. The Kalman gain matrix will also begin to be updated as soon as the simulation begins. The initial error coefficient matrix was determined using the errors obtained by running the given AUV and open loop linear model over the random input path for 100 seconds. The difference between results for the yaw and yaw rate were then used to find the coefficient matrix shown in Equation (4.7).

$$\text{Coefficient matrix} = \begin{bmatrix} 350.07 & -0.027 \\ -0.027 & 13.029 \end{bmatrix} \tag{4.7}$$

The faults to be considered in this section are those stated previously in section 3.4. The first simulation was performed with no fault occurring in the system. This was to show how the AUV reacted to having the Kalman filter in the system. The results show that the final yaw angle was slightly larger than the standard control system, but that the yaw rate was near to the standard response (the results for a demand of 50 degrees can be seen in Figures 4.3 and 4.4). The results for all three sizes of step inputs (10, 50 and 90 degrees as defined in Chapter 3) were similar, but only the 50 degrees input has been shown here.

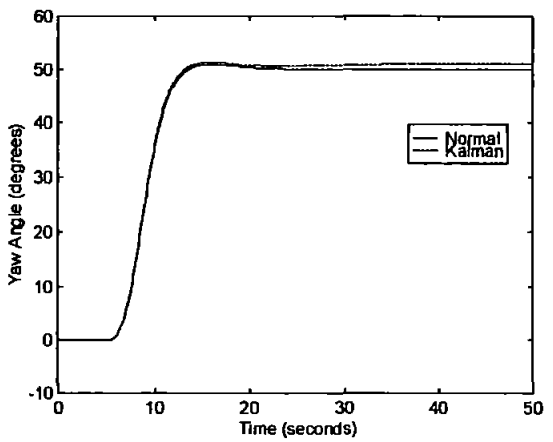


Figure 4.3 Yaw Responses of Normal and Kalman Filter Controllers

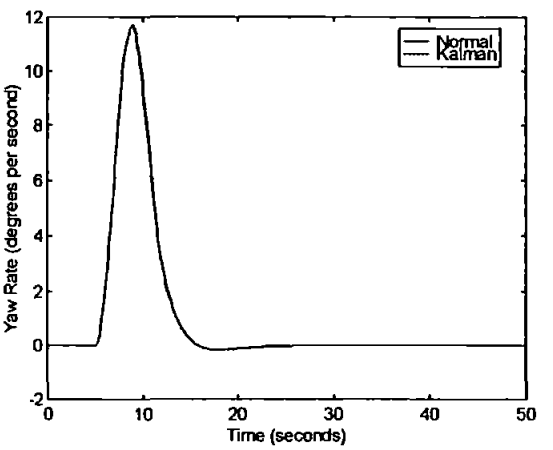


Figure 4.4 Yaw Rate Responses of Normal and Kalman Filters Controllers

For all of the sizes of step inputs considered the demand was implemented after 5 seconds during a simulation of length 50 seconds. The sensor failures in the yaw sensor were implemented after 15 seconds. This was chosen so that the AUV would

have almost completed the manoeuvre before the failure occurred. The sensor failures in the yaw rate sensor were implemented after 8 seconds. This time was chosen because the fault must occur after the demand has been made (5 seconds) and before the AUV has reached the demanded yaw (about 20 seconds). For faults on both the yaw and yaw rate sensors the intermittent and noise failures were both implemented at the beginning of the simulation.

In an attempt to keep this thesis as concise as possible all of the graphs for yaw channel sensor failures are presented within Appendix D. The major points from the graphs are discussed within the next few sections, however only some of the relevant graphs will be shown at this point.

4.3.1. Yaw Sensor Failures

The first sensor failures were carried-out on the yaw sensor. All three types of sensor failure discussed in Chapter 3 were implemented over all three sizes of step inputs. The complete set of results for this work can be found in Appendix D, therefore only certain graphs are shown within this section.

(a) Percentage Signal Failure Tests

The first type of failure to be implemented was the percentage signal failure. Thirty simulations were run with this fault occurring with the correct percentage at the given time. From this set of results it can be seen that as the percentage of information which is being sent back from the AUV decreases the standard ANFIS controller becomes less able to find the demanded yaw angle. This was at its worst when a total failure occurred, at which the ANFIS controller sent the AUV into a circular path.

The Kalman filter was far better at handling this fault. As can be seen from the results in Appendix D, the control system using the Kalman Filter to attempt to correct for yaw sensor failures, was able to minimise the error occurring. For small errors the Kalman filter had only a small effect if any. However as the level of failure increased, so did the effect of the Kalman filter. This can be seen in Figure 4.5.

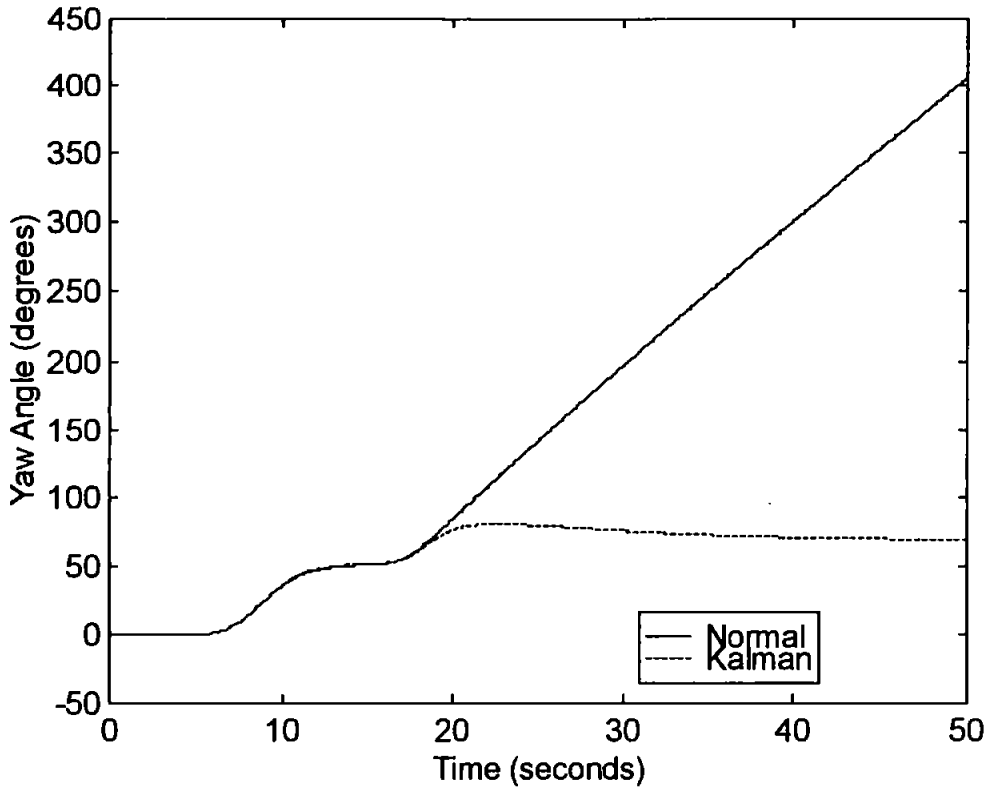


Figure 4.5 Yaw Angle Responses to Total Yaw Sensor Failure Over a Step Demand of 50 Degrees

Figure 4.5 shows how for this level of failure the ANFIS controller forced the AUV to a yaw angle of just over 400 degrees by the end of the 50 second simulation as opposed to the 60 degree yaw angle of the Kalman filter enhanced ANFIS controller. This was typical of the kind of results being produced by both the Kalman filter enhanced control system and ANFIS control system.

To show how the Kalman filter was adapting during this test the Kalman gains were recorded and can be seen in Figure 4.6.

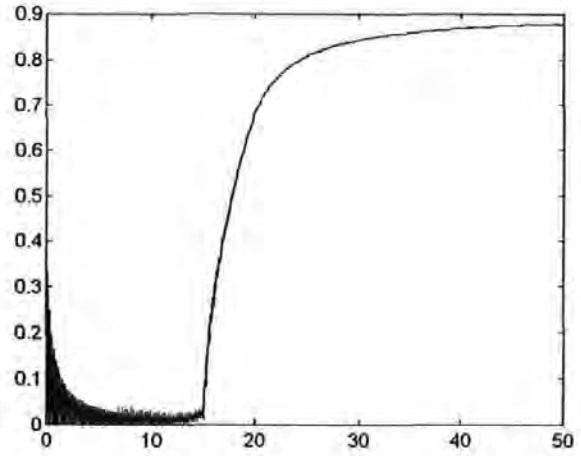


Figure 4.6(a) Kalman Filter Gain Value for Matrix Position (1,1)

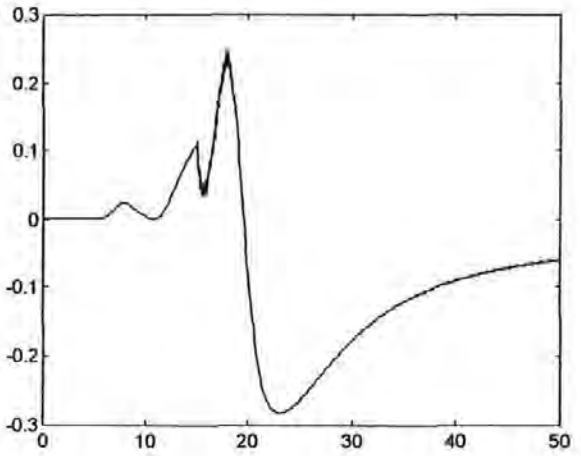


Figure 4.6(b) Kalman Filter Gain Value for Matrix Position (1,2)

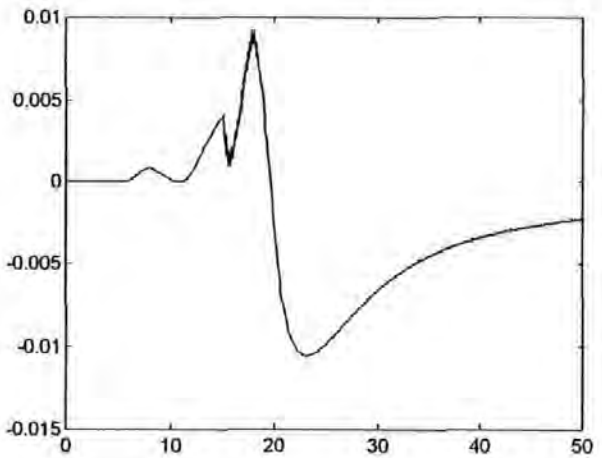


Figure 4.6(c) Kalman Filter Gain Value for Matrix Position (2,1)

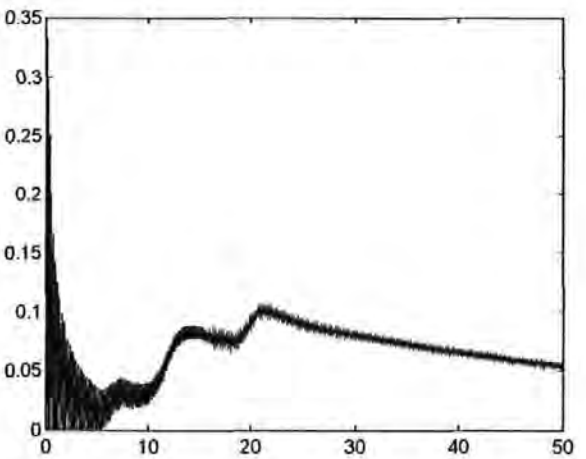


Figure 4.6(d) Kalman Filter Gain Value for Matrix Position (2,2)

At this point it is important to note that a 2x2 Kalman gain matrix for the yaw and yaw rate is being used for this work since the system model is second order. The Figure 4.6(a) relates to the value being used purely by the yaw channel as can be expected this starts to settle to just above zero until the failure occurs then it rapidly approaches a value of one. This represents a shift in the Kalman filter from almost totally believing the sensor (if the value is zero) to almost totally believing the linear model (if the value is one).

Figures 4.6(b) and 4.6(c) relate to both the yaw and yaw rates and as you should expect both change significantly when the failure occurs. Figure 4.6(d) is related to only the yaw rate and is hardly affected by the failure occurring and remains close to zero throughout the test. This is because no failure has occurred in the yaw rate sensor and hence a value of, or about, zero would be expected. Other Kalman filter gain values have been recorded from other test and are in Appendix D.

(b) Intermittent Signal Failure Tests

For the intermittent signal failure test all of the results were of a poor standard, however of the two types of control system used the one using the Kalman filter was slightly better. This can be seen in Figure 4.7.

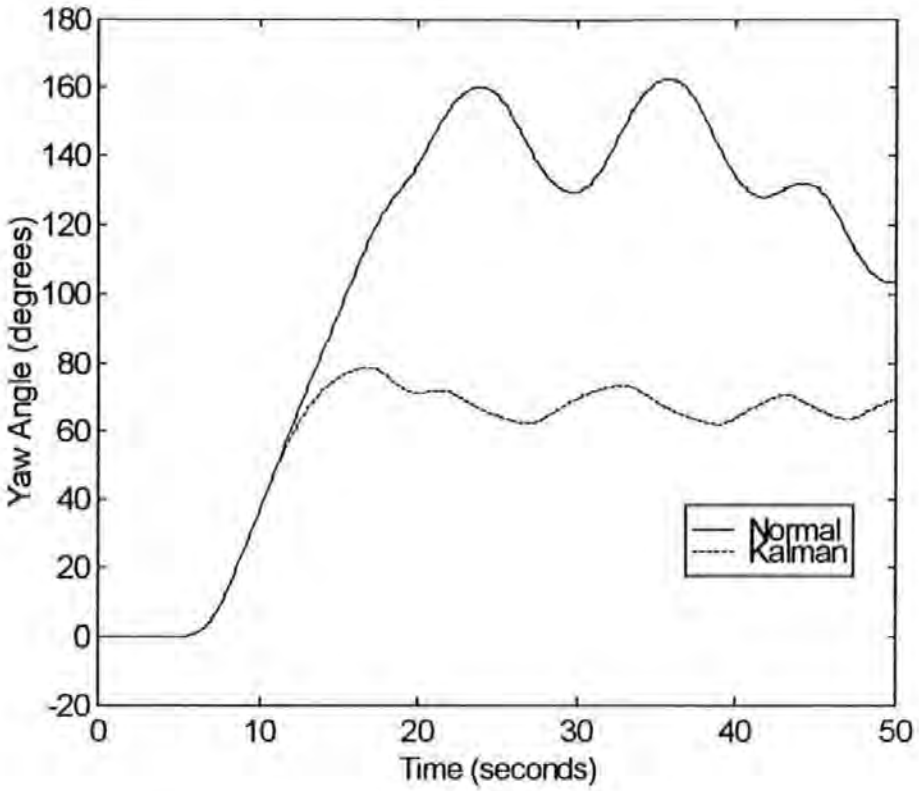


Figure 4.7 Yaw Angle Responses to Intermittent Yaw Signal Failure Over a Step Demand of 50 Degrees

This graph shows the how controller handled the failure over a step input of 50 degrees. The Kalman filter enhanced ANFIS control system had the AUV maintaining a yaw angle around 60 degrees, which is a lot closer to the desired angle then the normal ANFIS controller which had the AUV maintaining a yaw angle around 130 degrees. Similar results were achieved for the other two step sizes with the Kalman filter enhanced ANFIS controller always maintaining a yaw angle closer to the desired one.

(c) Signal to Noise Ratio Tests

Finally the control systems were both submitted to three levels of noise (1%, 5% and 10% SNR) as determined previously in Chapter 3. Both controllers produced almost identical responses to this type of failure and it was only for the 50 degrees step input that any difference was noticeable. However both controllers did manage to keep the

AUV within 10% of the demanded yaw angle. The 50 degrees step input responses for both controllers, for all three levels of SNR, can be seen in Figure 4.8 and Figure 4.9.

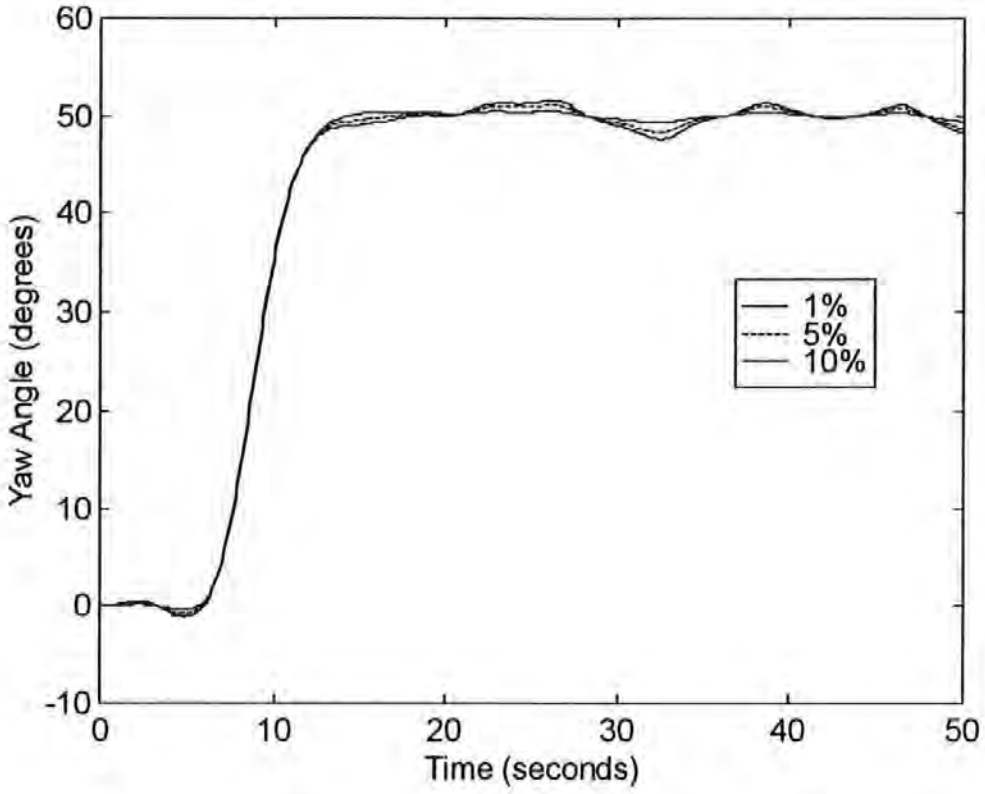


Figure 4.8 Yaw Angle Responses to SNR Faults
Over a Step Demand of 50 Degrees for The ANFIS Controller

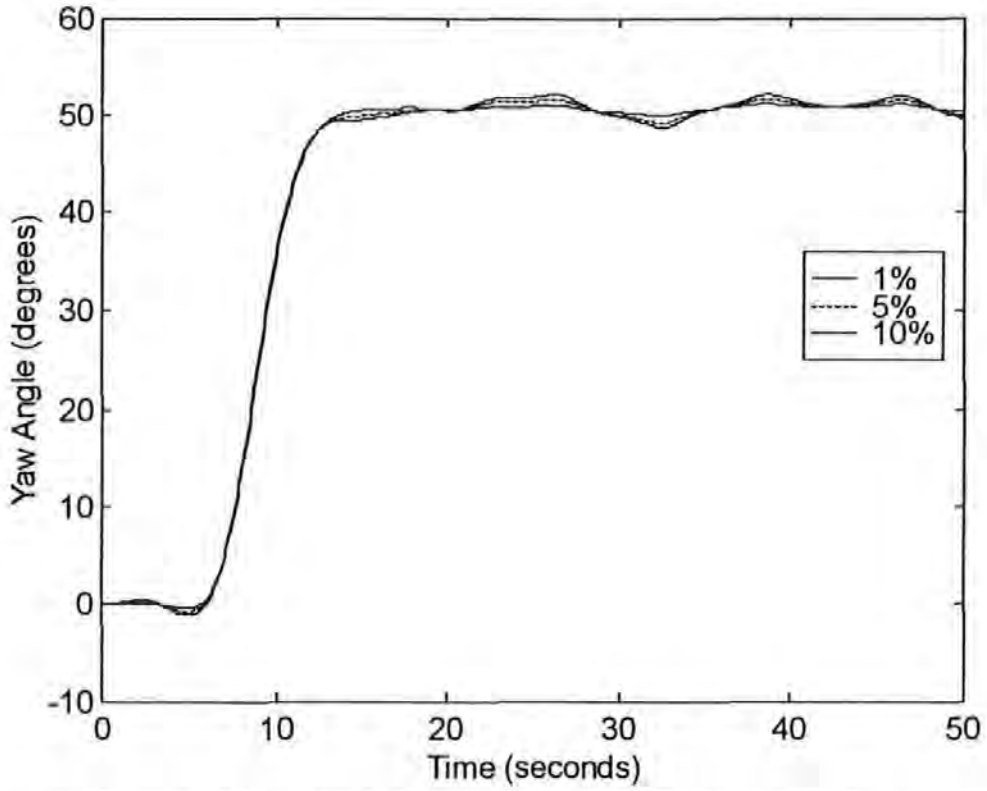


Figure 4.9 Yaw Angle Responses to SNR Faults Over a Step Demand of 50 Degrees for The Kalman Filter Enhanced Controller

This completes a review of all the failures being considered for the yaw sensor.

4.3.2. Yaw Rate Sensor Failures

The second sensor failures were carried-out on the yaw rate sensor. All three types of sensor failure discussed previously were implemented over all three sizes of step. The complete set of results for this work can be found in Appendix D, therefore, again, only certain graphs will be shown within this section.

(a) Percentage Signal Failure Tests

The first type of failure to be implemented was the percentage signal failure. Thirty simulations were run with this fault occurring with the correct percentage at the given time. From this set of results it can be seen that as the percentage of information which is being sent back from the AUV decreases the ANFIS controller struggled to

cope, with an increase in the overshoot being the most noticeable effect of the failures. It did however always manage to reach the desired yaw angle, this was due to the ANFIS controller design which relied mainly on the yaw sensor the yaw rate sensor being used to provide a damping effect. The controller using the Kalman filter did manage to overcome this problem, but only for large failures on the two largest step inputs. The best example of this is for total yaw rate sensor failure over the 90 degrees step input and is shown in Figure 4.10.

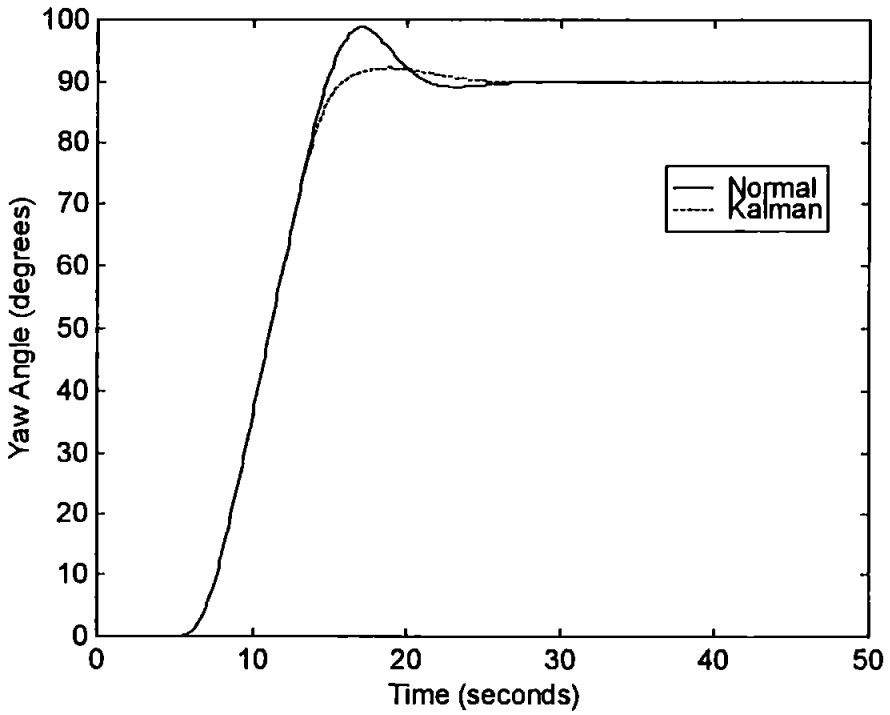


Figure 4.10 Yaw Angle Responses to Total Yaw Rate Signal Failure Over a Step Demand of 90 Degrees

Where an overshoot of 9 degrees has been recorded for the total signal loss test for the normal ANFIS controller and an overshoot of only 2 degrees when the Kalman filter enhanced system was used. The other tests showed similar results with the Kalman filter enhanced ANFIS controller always have the smallest overshoot.

(b) Intermittent Signal Failure Tests

For the intermittent signal failure on the yaw rate sensor the results were very similar to those for the total sensor failure. This is due to the fact that the signal is switching between total failure and prefect feedback signal. This may be occurring too fast for the controller to respond, again this is due to the ANFIS controller being concerned far more with the yaw sensor.

The similarity of the these results can clearly be seen by comparing Figure 4.12 to those results shown in Figure 4.11.

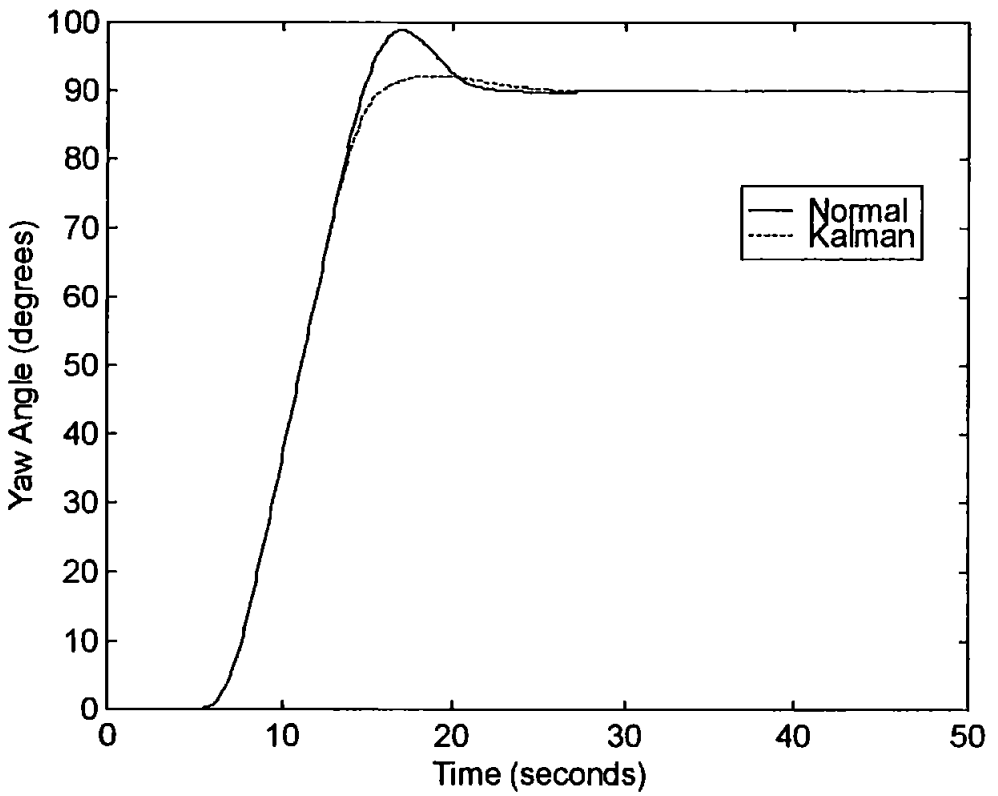


Figure 4.12 Yaw Angle Responses to Intermittent Yaw Rate Signal Failure Over a Step Demand of 90 Degrees

(c) Signal to Noise Ratio Tests

Finally the control systems were both submitted to noise as determined previously. Both controllers produced almost identical responses to this type of failure and the

difference was unnoticeable for all levels of failure and for all sizes of step input. However both controllers did manage to keep the AUV within 10 percent of the demanded angle for tests. The noise itself presented less of a problem as the step size was increased. The 90 degrees step input responses for all levels of SNR for both control systems are shown in Figure 4.13 and Figure 4.14.

These two graphs show how little the noise is affecting the control systems, with neither of them ever going over 93 degrees or below 87 degrees after reaching the desired yaw angle.

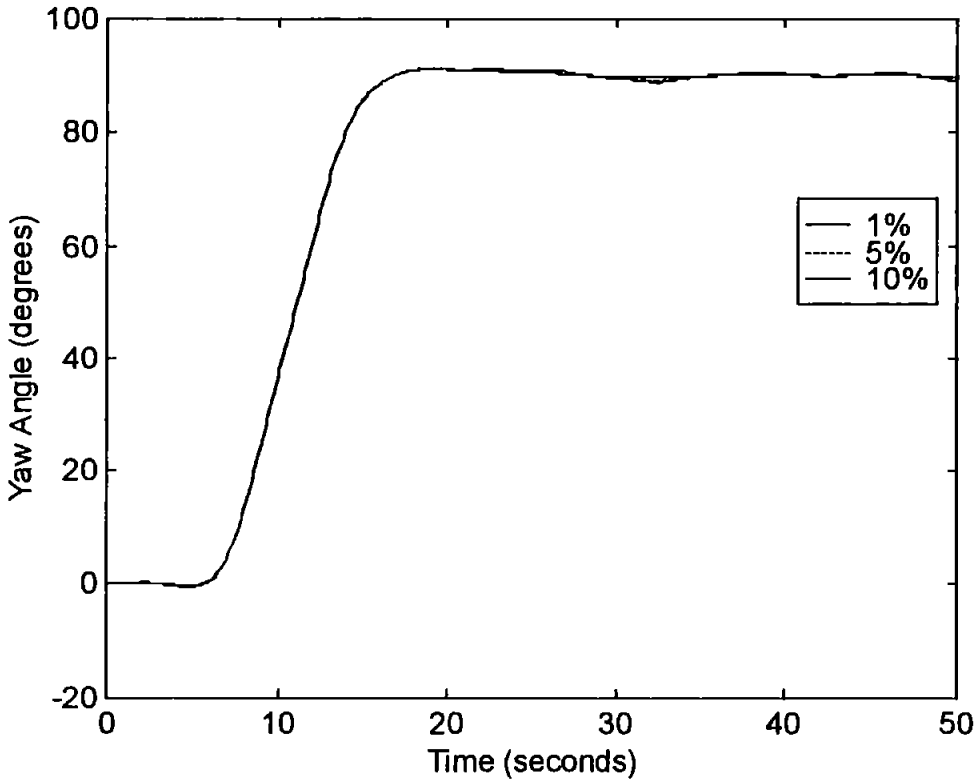


Figure 4.13 Yaw Angle Responses to SNR Faults Over a Step Demand of 90 Degrees For The ANFIS Controller

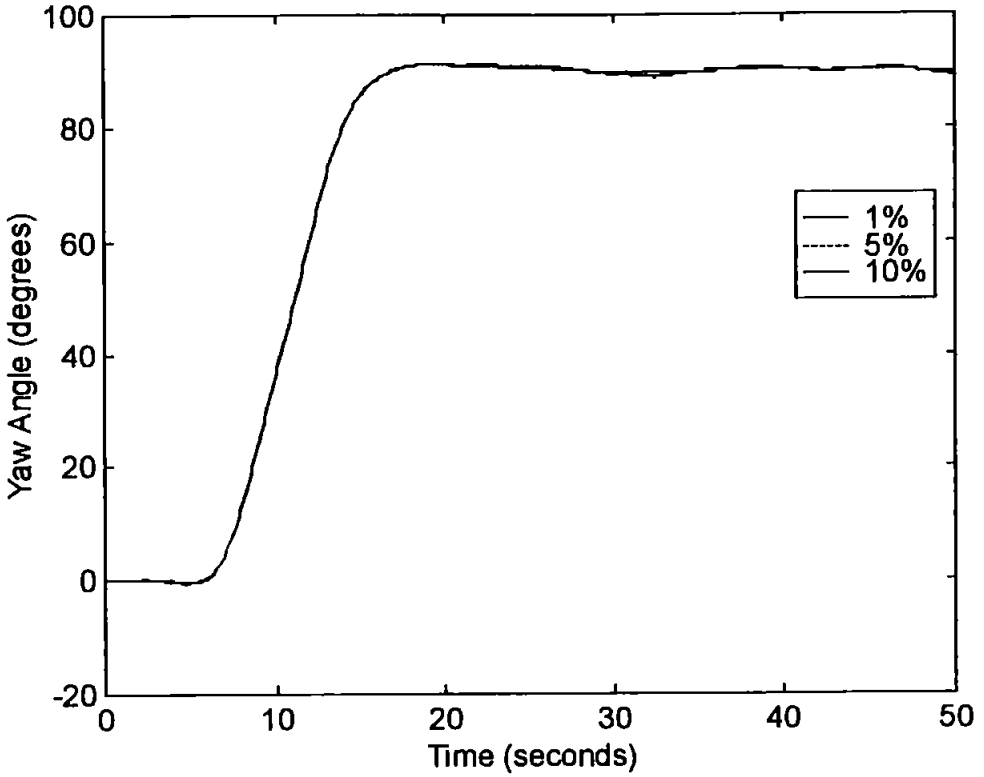


Figure 4.14 Yaw Angle Responses to SNR Faults Over a Step Demand of 90 Degrees For The Kalman Filter Enhanced Controller

All of the yaw channel sensor faults have now been covered, further examples of all types of failures discussed can be found in the Appendices D.

4.3.3. Roll Sensor Failures

The next sensor failures were carried-out on the roll sensor. A Kalman filter using the linear roll model from the previous Chapter was executed in the same way that the yaw channel system had been. The initial error coefficient matrix was determined using the errors obtained by running the given AUV and open loop linear model over the random input path for 100 seconds. The difference between results for the roll and roll rate were then used to find the coefficient matrix shown in Equation (4.8).

$$\text{Coefficient matrix} = \begin{bmatrix} 0.0070 & -0.0107 \\ -0.0107 & 0.0938 \end{bmatrix} \tag{4.8}$$

All three types of sensor failure discussed previously were implemented over all three sizes of initial roll angle. The complete set of results for this work can be found in Appendix E, therefore only certain graphs will be shown within this section.

(a) Percentage Signal Failure Tests

The first type of failure to be implemented was the percentage signal failure. Thirty simulations were run with this fault occurring with the appropriate percentage at the given time. The first size of initial roll angle considered was that of 5 degrees. The standard ANFIS controller shows a noticeable degradation in performance as the level of failure increases. This is shown for this set of results by the AUV having a decrease in overshoot until for total failure the vehicle just returns to its natural roll angle of zero degrees. However when the Kalman filter is placed within the system it handles all levels of fault with the same response. Hence the Kalman filter is unaffected by the level of fault.

Similarly for an initial input of 15 degrees the ANFIS controller shows less of an overshoot as the level of failure increases. For all levels of failure the AUV returns to the demanded angle of zero degrees. Once again for the Kalman filter the results show how it was unaffected by each level of failure.

For the 25 degrees initial roll angle the ANFIS controller displayed similar results to the 15 degrees initial roll angle. As can be seen in Figure 4.14 the AUV has less overshoot as the failure increases.

The Kalman filter enhanced control system produced its best results for this initial angle. It estimates the roll angle very closely achieve the desired overshoot of 10 degrees. The key feature, which can be seen in Figure 4.15, is that of the slight oscillatory motion of the vehicle, which continues after it should have achieved the demanded angle.

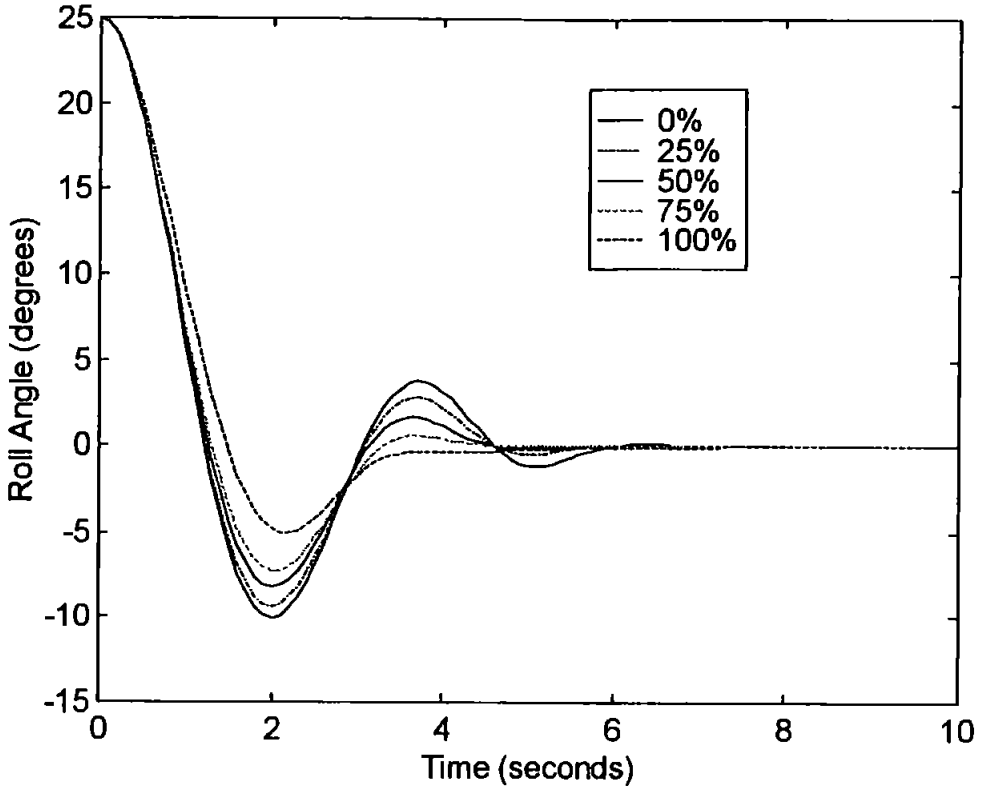


Figure 4.14 Roll Angle Responses to Percentage Signal Loss Tests Roll Signal Failure Over an Initial Angle of 25 Degrees For The ANFIS Controller

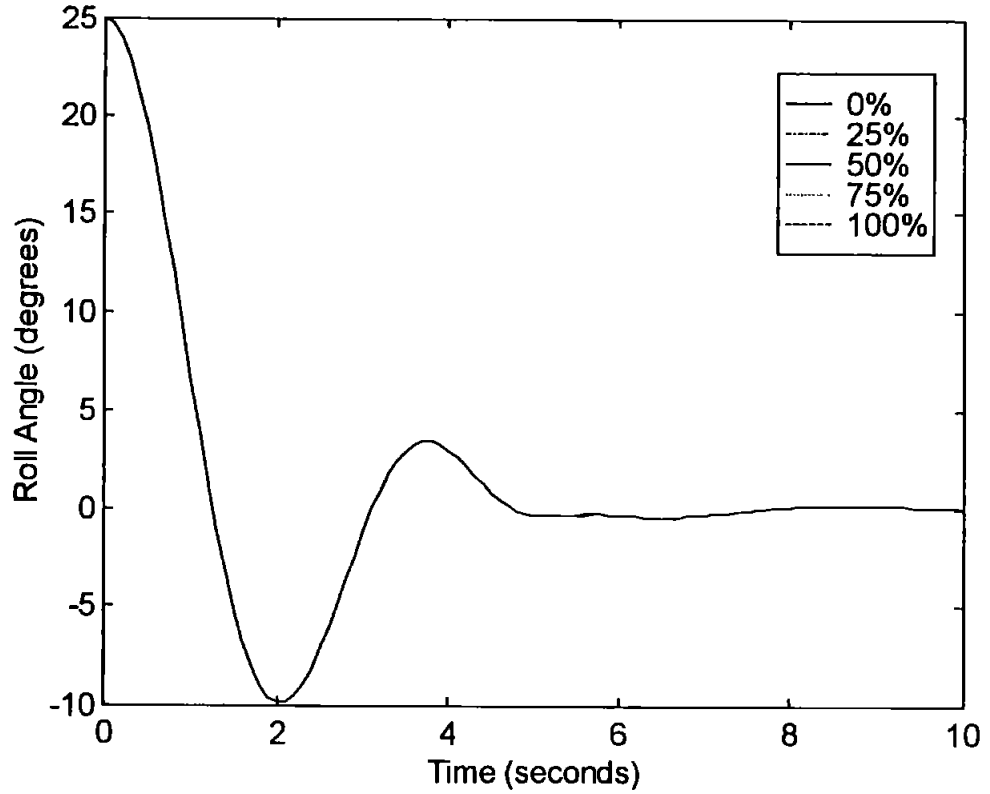


Figure 4.15 Roll Angle Responses to Percentage Signal Loss Tests Roll Signal Failure Over an Initial Angle of 25 Degrees For The Kalman Filter Controller

(b) Intermittent Signal Failure Tests

For the intermittent signal failure tests the Kalman filter enhanced controller was able to approximate the unaffected system more closely. The most noticeable effect were the reductions of overshoots. This can be most clearly seen on the 25 degrees initial roll angle. The AUV should respond with a first overshoot of approximately 10 degrees but only manages 5 degrees, as can be seen in Figure 4.16. This is also seen in the second overshoot where the ANFIS controller again can only force a decrease size of overshoot. The Kalman filter enhanced control system performed well for all initial roll angles, as can be seen in Figure 4.16. It was able to produce an accurate approximation when the initial angle was 25 degrees with a RMSE of 0.407 degrees.

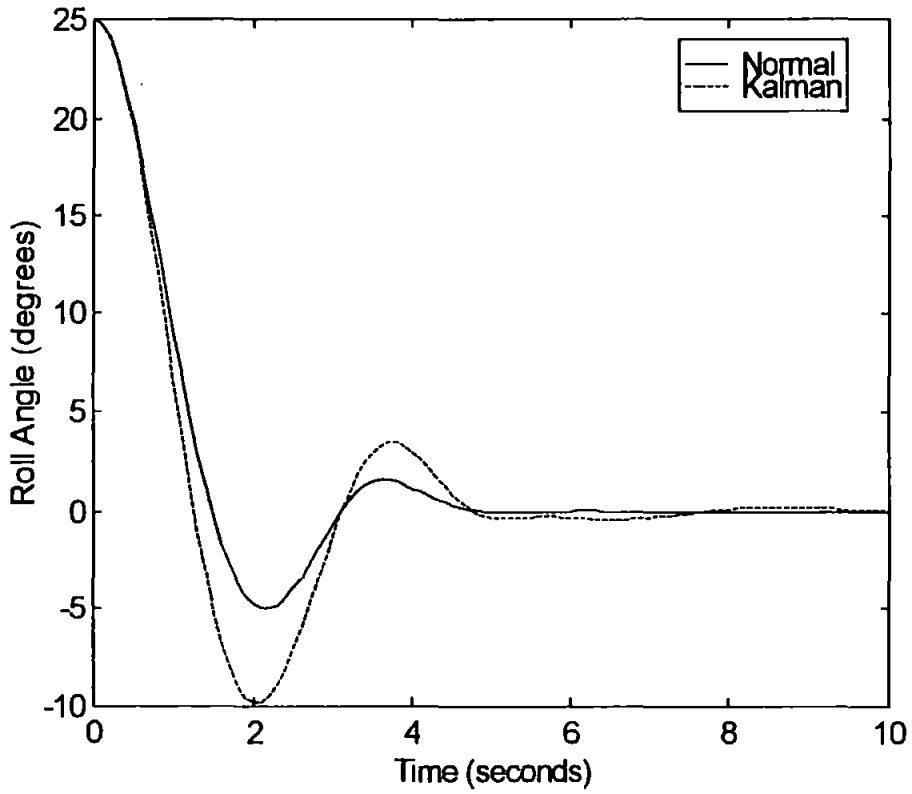


Figure 4.16 Roll Angle Responses to Intermittent Signal Failure For an Initial Angle of 25 Degrees

(c) Signal to Noise Ratio Tests

Finally both the control systems were both submitted to three levels of noise as previously determined. For the initial angle of 5 degrees the ANFIS controller was unable to adapt and produced a relatively slow response. Also the controller failed to force the overshoot shown on the fault free system. For each level of noise the ANFIS controller produced a similar result. For the initial angle of 15 degrees the ANFIS controller did force an overshoot of just under 4 degrees, it should have been 6 degrees. Once again the results for all three levels of noise for this size of initial angle were very alike. For the initial angle of 25 degrees the ANFIS controller did achieve the desired overshoot, but also had trouble maintaining the final roll angle of zero degrees as can be seen in Figure 4.17.

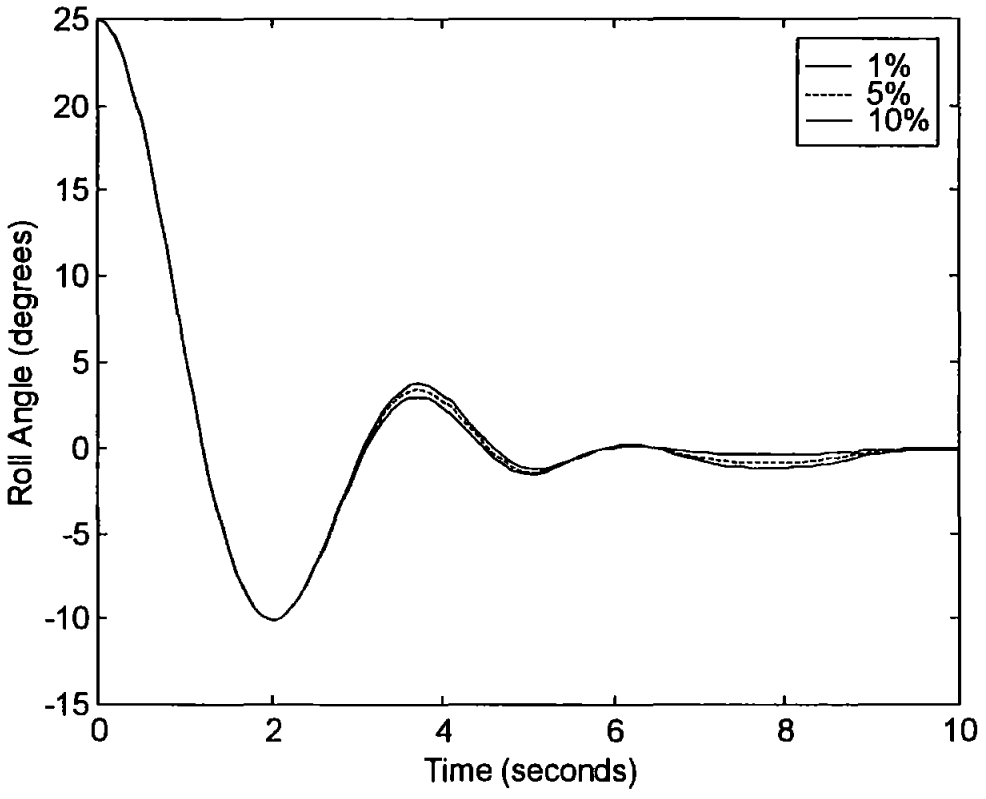


Figure 4.17 Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees For The ANFIS Controller

Over all three initial sizes of roll angles the Kalman filter enhanced control system displayed no signs of being troubled by the level of noise and produced three identical results as is shown in Figure 4.18. The previously noted oscillatory motions associated with the Kalman filter have again been present on all three sizes of initial angles. Overshoots were also forced in all cases, although only the 25 degrees initial angle managed to achieve the required magnitude as shown in Figure 4.18.

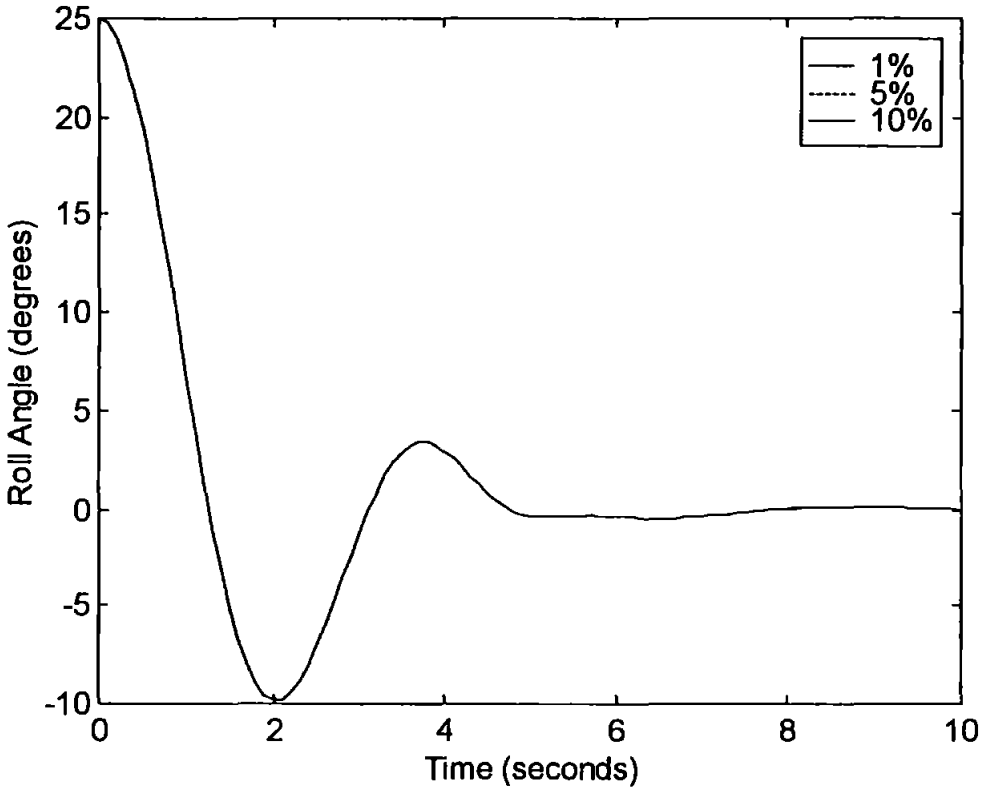


Figure 4.18 Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees For The Kalman Filter Controller

The graphs shown are representative of the results achieved and once again it is worth pointing out for interested reader that the full set of results can be seen in Appendix E.

4.3.4. Roll Rate Sensor Failures

The next sensor failures were carried-out on the roll rate sensor. All three types of sensor failure discussed previously were implemented over all three sizes of initial angles. The same Kalman filter, with the same initial error coefficient matrix, as used

for the roll sensor failures have been used for the roll rate sensor failures. The complete set of results for this work can be found in Appendix E, therefore only certain graphs will be shown within this section.

(a) Percentage Signal Loss Tests

The first type of failure to be implemented was the percentage signal failure. Thirty simulations were run with this fault occurring with the appropriate percentage at the given time. The standard ANFIS controller shows a noticeable degrade in performance as the level of failure increases. As with the yaw rate sensor, the roll rate sensor produces a damping effect. This is shown for this set of results by the AUV having an increase in overshoot until for total failure the vehicle has not achieved the required angle of zero degrees. The results were similar for all initial angle sizes. Shown in Figure 4.19 are the responses of the 25 degrees initial roll angle.

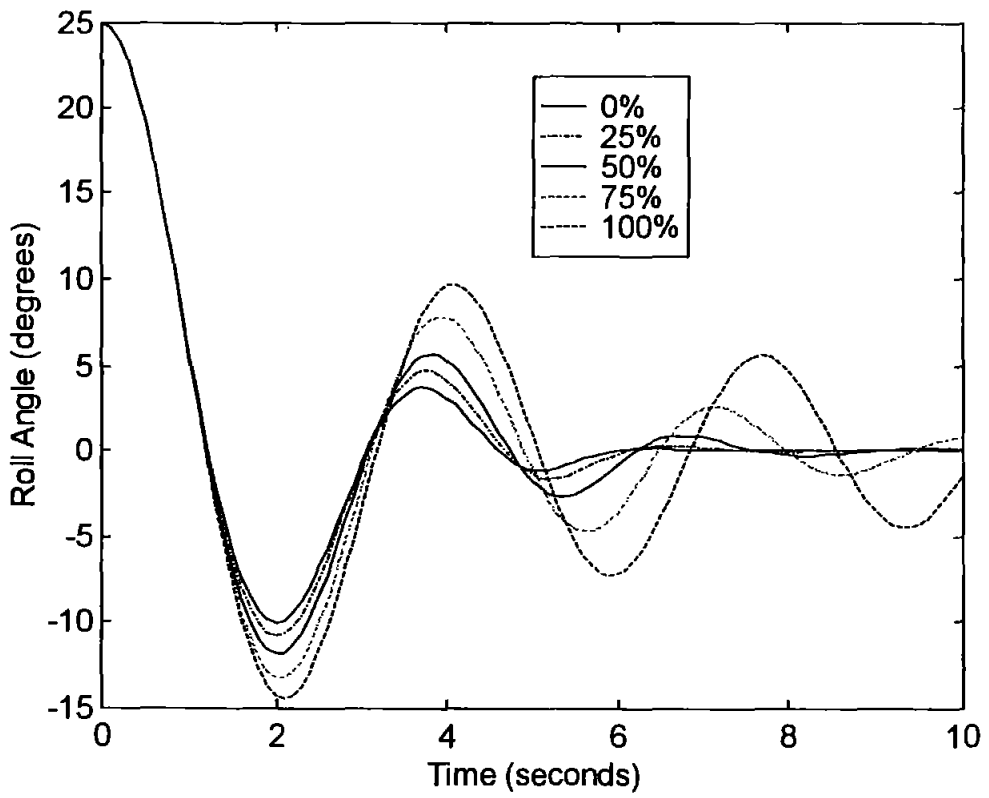


Figure 4.19 Roll Angle Responses to Percentage Signal Loss Tests Roll Rate Signal Failure Over an Initial Angle of 25 Degrees For The ANFIS Controller

However when the Kalman filter is placed within the system it handles all levels of fault with almost the same the same response. This shows that the Kalman filter is slightly affected by the level of fault. Also the oscillatory motion is present again for all sizes of initial angle considered. The Kalman filter enhanced system does for these failures produce overshoots of almost the desired magnitudes. All of these features can be seen in Figure 4.20.

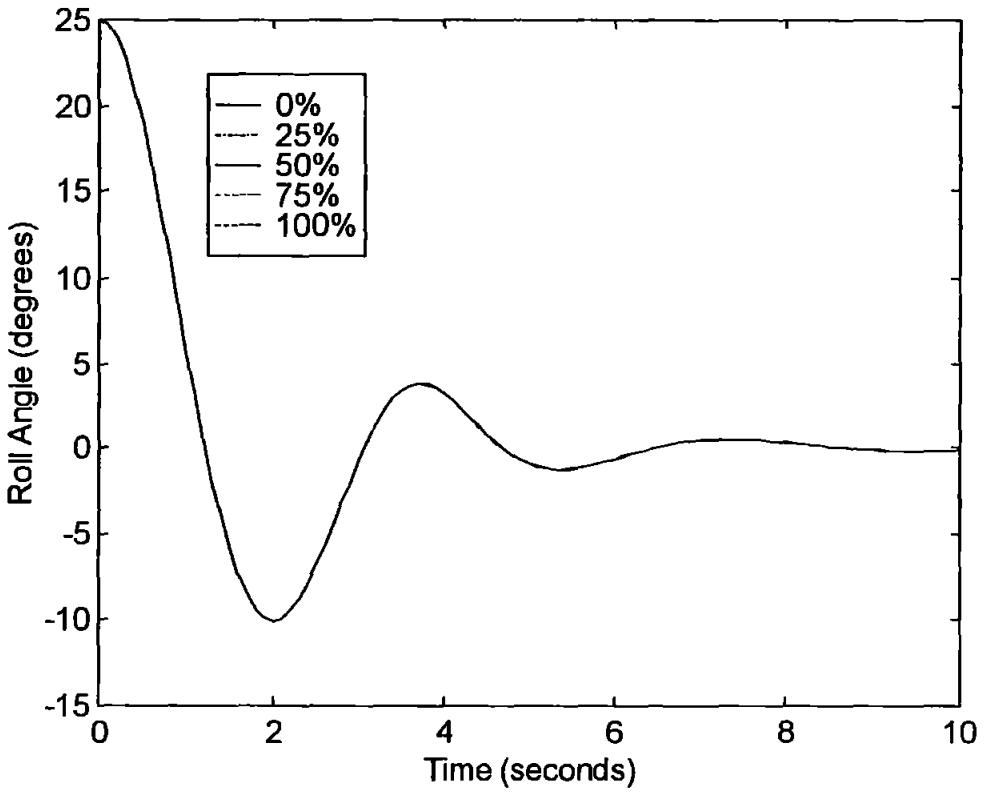


Figure 4.20 Roll Angle Responses to Percentage Signal Loss Tests
Roll Rate Signal Failure Over an Initial Angle of 25 Degrees
For The Kalman Filter Controller

(b) Intermittent Signal Failure Tests

Next the intermittent signal failure test was considered. The normal controller produced an increase in the size of overshoot angles for each size of initial roll angle. Both controllers displayed an oscillatory motion during each simulation, however each graph shows the Kalman filter enhanced control system's oscillations to be of

smaller magnitude. The best example of this is for the initial roll angle of 25 degrees and is shown in Figure 4.21.

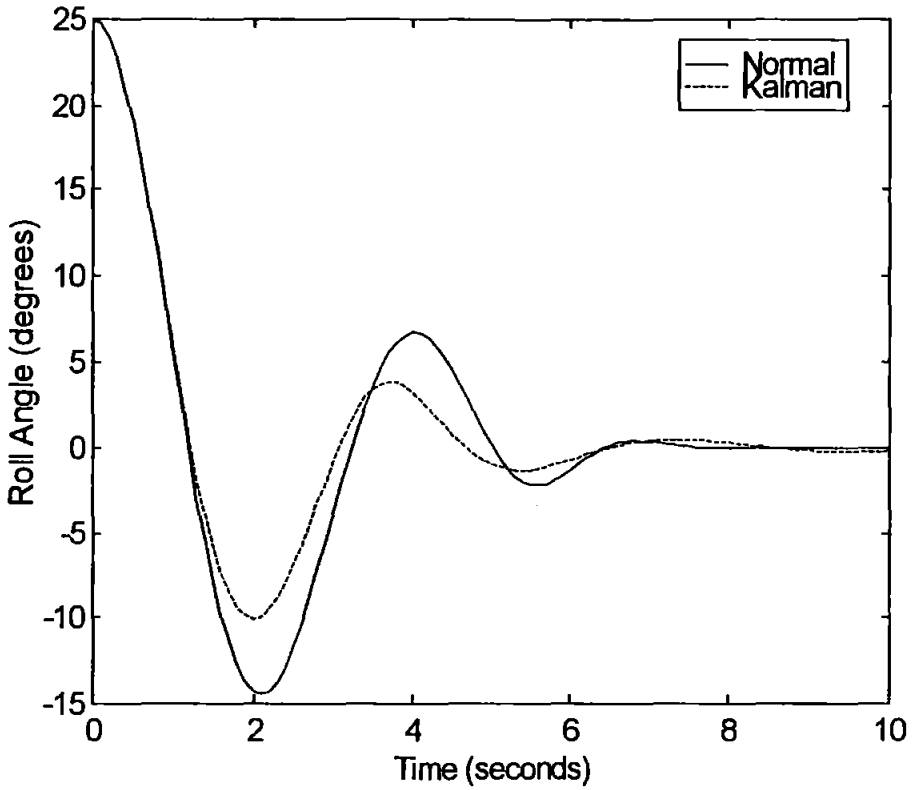


Figure 4.21 Roll Angle Responses to Intermittent Roll Rate Signal Failure Over an Initial Angle of 25 Degrees

(c) Signal to Noise Ratio Tests

Finally both the control systems were submitted to three levels of noise as previously determined. For the initial angle of 5 degrees the ANFIS controller was unable to adapt and produced an oscillatory motion. Also the controller forced the AUV to overshoot by a larger that shown on the fault free system. For each level of noise the ANFIS controller produced a similar result. For the initial angle of 15 degrees the ANFIS controller did force an overshoot of 9 degrees, it should have been only 6 degrees. Once again the results for all three levels of noise for this size of initial angle were very alike. For the initial angle of 25 degrees the ANFIS controller did achieve the desired overshoot and displayed only a small level of oscillatory motion as can be seen in Figure 4.22.

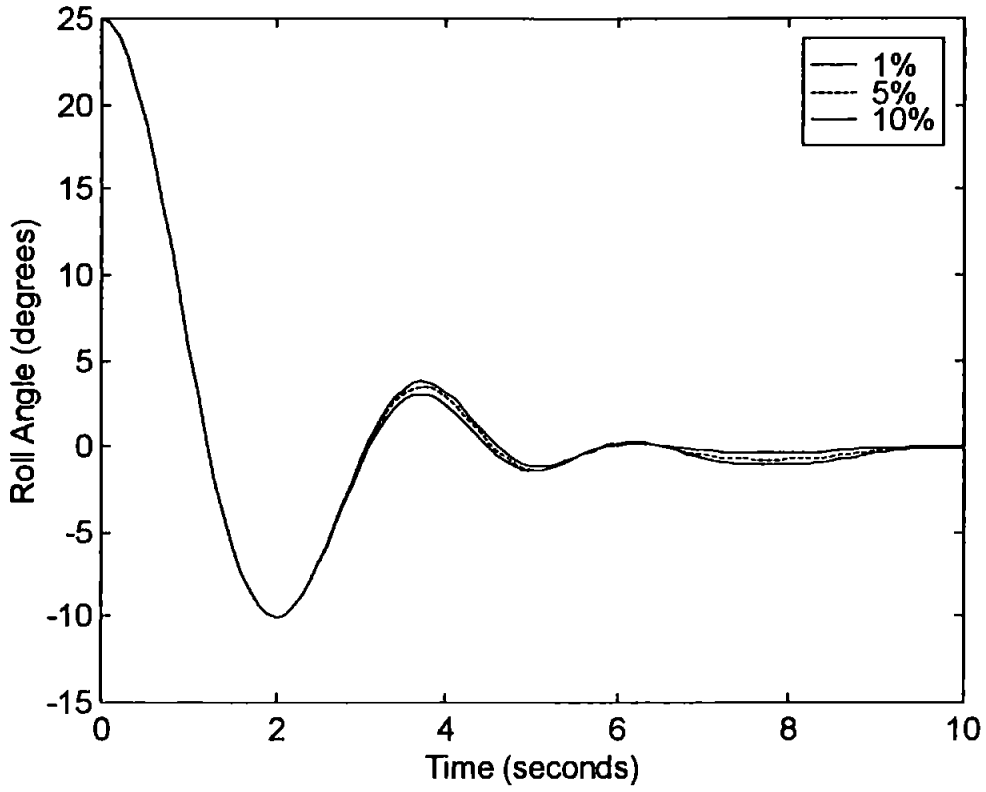


Figure 4.22 Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees For The ANFIS Controller

Over all three initial sizes of roll angles the Kalman filter enhanced control system displayed no signs of being troubled by the level of noise and produced three identical results as is shown in Figure 4.23. The previously noted oscillatory motions associated with the Kalman filter are again present on all three sizes of initial angles. The desired initial overshoots were achieved in all cases. Shown in Figure 4.23 are the roll responses to all three considered levels of noise for the initial roll angle of 25 degrees.

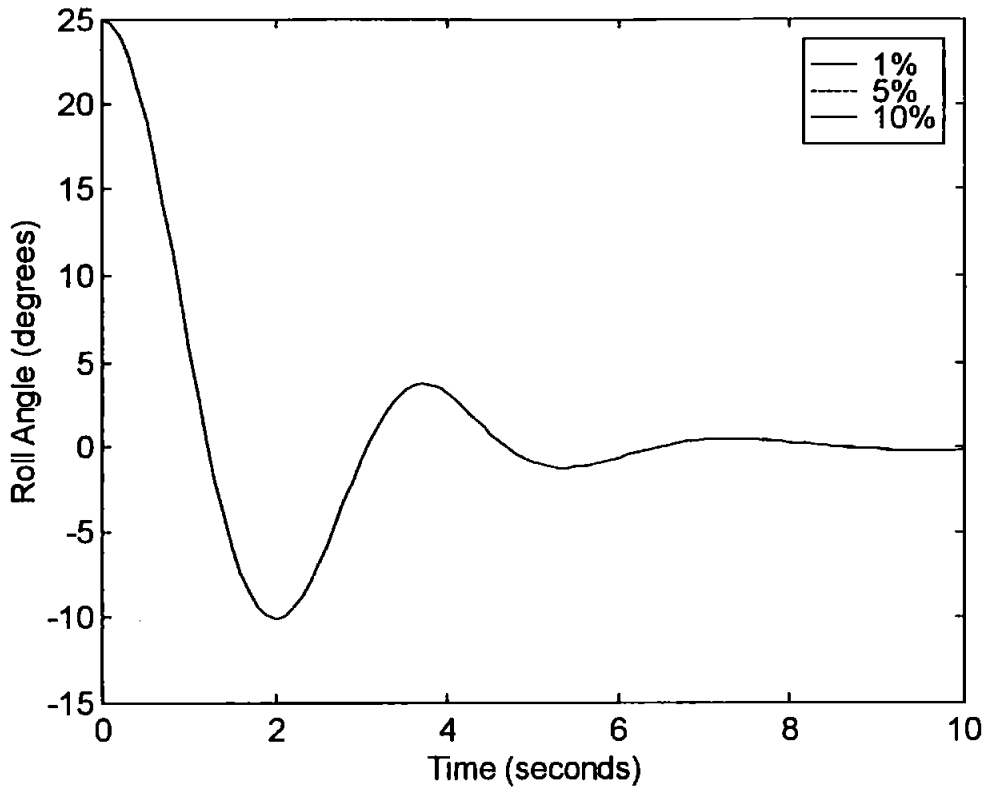


Figure 4.23 Roll Angle Responses of All Levels of Noise Over an Initial Angle of 25 Degrees For The Kalman Filter Controller

The graphs shown are representative of the results achieved and once again it is worth pointing out for interested reader that the full set results of can be viewed at the back of this thesis in Appendix E.

4.4. KALMAN RESPONSES TO ACTUATOR FAULTS

Now that the sensor faults have been dealt with in the previous sections, attention now switches to faults occurring in the control actuators. The faults considered for this work are those based on work by Derradji and Mort (1996), where a loss of effectiveness (LOE) occurring within an actuator was used within the arena of manned submersibles. The LOE and its application to AUVs has previous been explained within this work in Chapter 3. All of the actuator failures considered have been described in section 3.4. For this work, failures were implement at the beginning of the simulations. Due to all actuators being of an identical nature for this vehicle and

hence any actuator recovery system found to be effective for the yaw channel must be effective for all other actuators as presented earlier (Chapter 3) only faults occurring within the yaw channel are considered herein. The AUV uses two actuators to control the yaw angle. These are the two actuators which move the upper and lower bow canards as shown in Figure 3.1. The upper canard actuator is the one in which the faults occur. Only the results for the yaw angle have been displayed here as very little information can be gained from showing the yaw rates, as unlike previously the yaw rate will give no further information when evaluating the systems performance. For the entire set of simulations step input demands on the yaw angle after 3 seconds were used. The first fault considered is that of a LOE occurring to the saturation level of the actuator, this reduces the maximum angle the canard can accomplish. The second fault is that of a LOE occurring to the rate limiter of the actuator, this reduces the speed the actuator can reach its desired position. The third fault considered is that of a LOE occurring to both the saturation and rate limiter of the actuator, this reduces both the maximum angle the canard can accomplish and the speed at which it can achieve that angle.

This section will show how three different control systems handled the three faults over three different step input demands (10, 20, and 30 degrees). The first system is the PD controller developed by Craven (1999). The second controller is the ANFIS controller also developed by Craven (1999). The final system is that described earlier within this Chapter of the Kalman filter enhanced ANFIS controller. All of the results for actuator failures using the standard PD controller, the ANFIS controller and the Kalman filter with the ANFIS controller can be seen in Appendix F.

4.4.1. Yaw Step Inputs of 10 Degrees

(a) The PD Controller

For this size of step input demand the PD controller under a fault free set up displayed a small overshoot of under 0.1 degrees and a rise time of 3.9 seconds. For the saturation fault this systems is totally unaffected for both 25% and 50% LOE. However 75% LOE produces a small effect and for 100% LOE the AUVs rise time has increased to 7.6 seconds. For the rate limiter fault the PD controller shows more signs of a fault occurring with small faults being noted for each level of LOE being

considered. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance. Again for 100% LOE the result is identical to previous 100% LOE's. Figure 4.24 shows the response of the AUV when it is subjected to both faults occurring when the PD controller is used. These results are displayed in Appendix F.

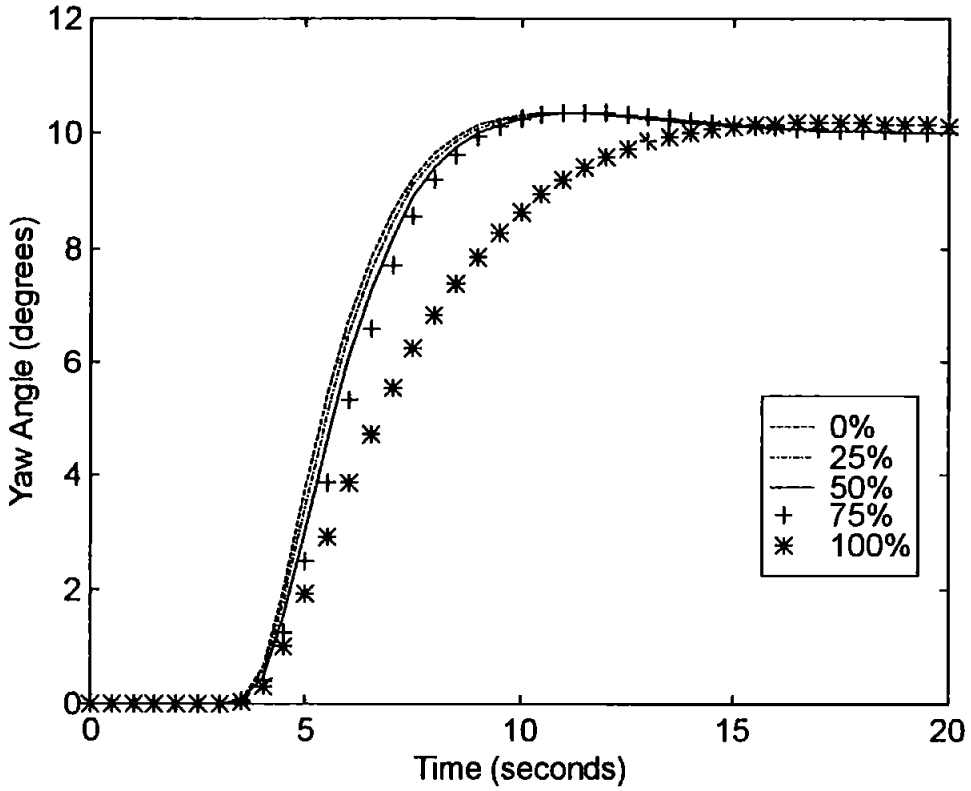


Figure 4.24 PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(b) The ANFIS Controller

The ANFIS controller for this size of step input demand under a fault free set up displayed almost no overshoot and a rise time of 4.5 seconds. For the saturation fault this systems is totally unaffected for both 25% and 50% LOE. The 75% LOE produces a small effect, but is a slight improvement on the PD controller. The 100% LOE produces the biggest effect on the AUV, which has its rise time reduced to 7.2 seconds. For the rate limiter fault the ANFIS controller shows more signs of a fault

occurring with small faults being noted for each level of LOE being considered. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance. It should also be noted at this point that the results for a LOE in both sections are identical to the LOE in the rate limiter for this step size. Again for 100% LOE the result is identical to previous 100% LOE's. These results are displayed in Appendix F. Figure 4.25 shows the response of the AUV when it is subjected to both faults occurring when the ANFIS controller is used.

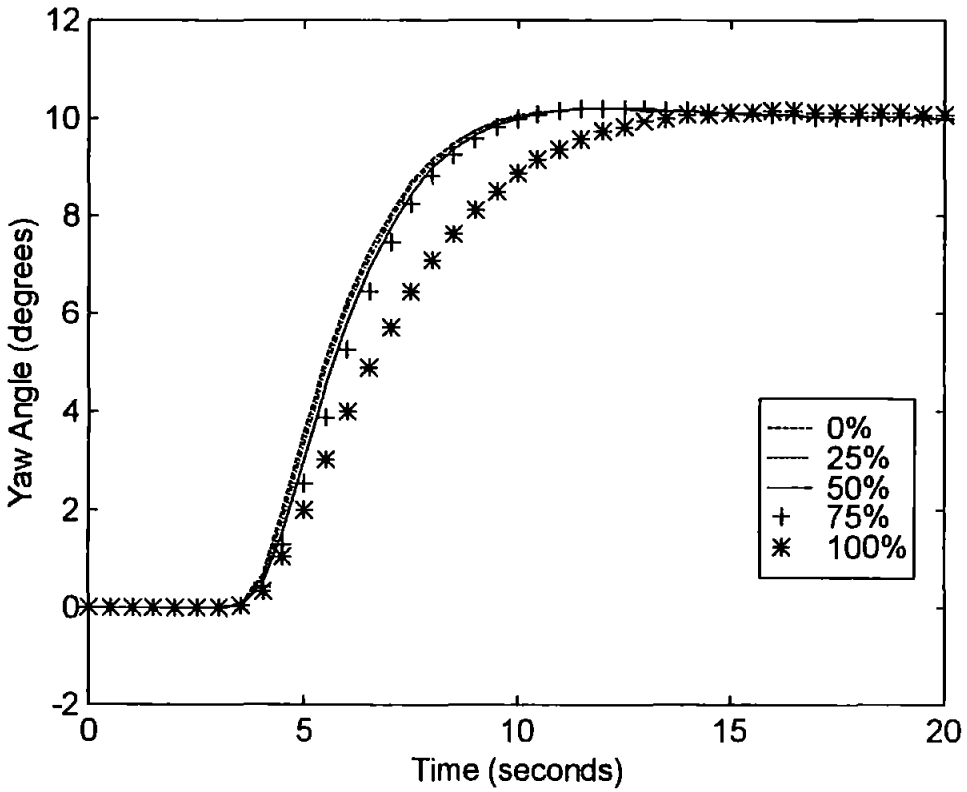


Figure 4.25 ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(c) The Kalman Filter Enhanced ANFIS Controller

When the Kalman filter is placed within the control system along with the ANFIS controller for this size of step input demand under a fault free set up displayed almost no overshoot and a rise time of 4.4 seconds. For the saturation fault this system is totally unaffected for both 25% and 50% LOE. The 75% LOE produces a small effect.

The 100% LOE produces the biggest effect on the AUV, which has its rise time reduced to 7.9 seconds and fails to reach the desired yaw angle. For the rate limiter fault the Kalman filter enhanced controller shows more signs of a fault occurring with small faults being noted for each level of LOE being considered. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance. Once again for 100% LOE the result is identical to previous 100% LOE's, this means that for the Kalman filter enhanced controller fails to achieve the desired angle for all types of fault when a 100% LOE occurs. These results are shown in Appendix F. Figure 4.26 shows the response of the AUV when it is subjected to both faults occurring when the Kalman filter enhanced ANFIS controller is used.

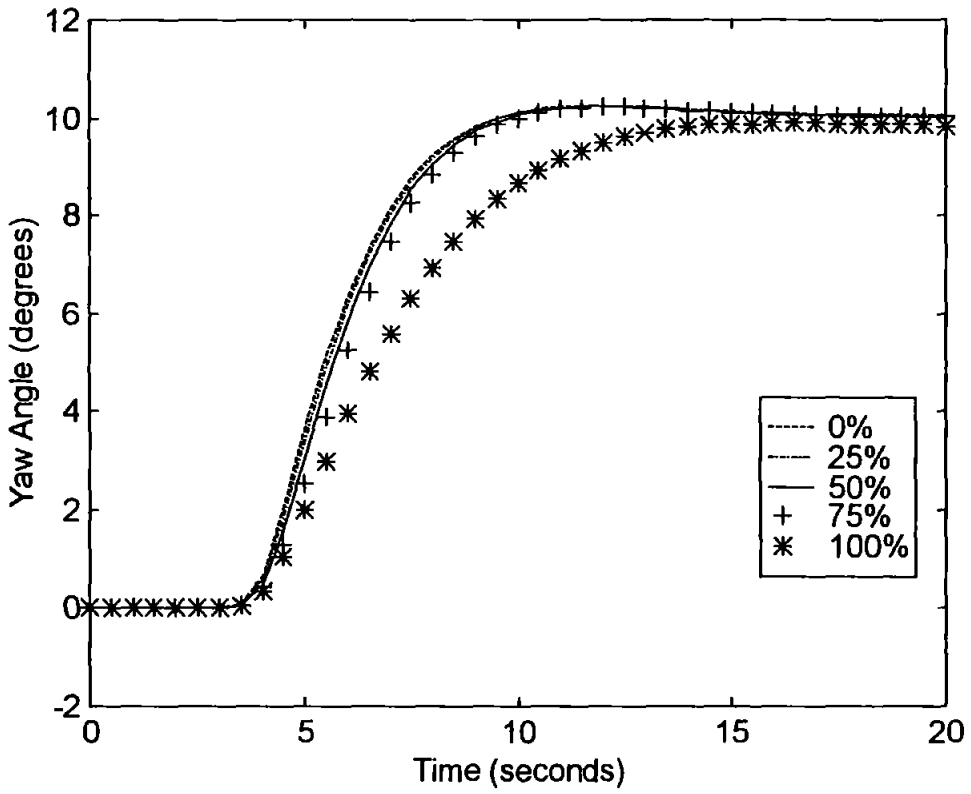


Figure 4.26 Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

Below is shown all three control systems attempting to handle a 100% LOE occurring in both sections of the actuator (Figure 4.27).

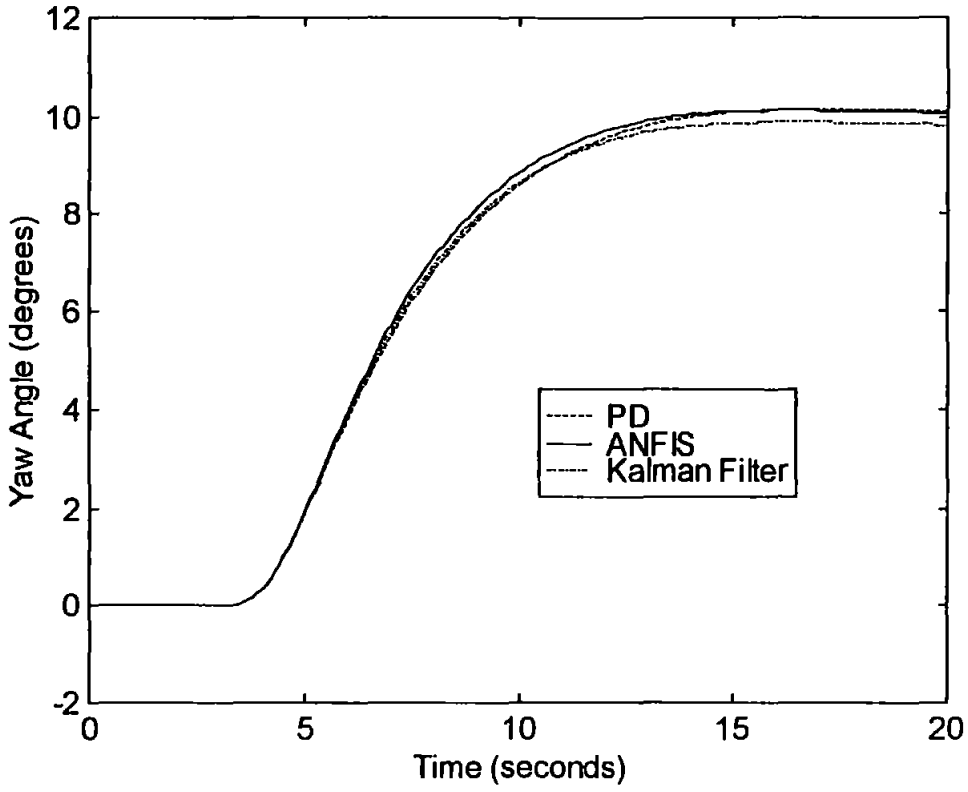


Figure 4.27 Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 10° Step Input Demand

Having seen how the three control systems handle the faults it is interesting to see the rise times of AUV for each test (Table 4.1). If it was on rise times alone it is clear to see the Kalman filter enhanced ANFIS control system performed best of the three systems considered on everything but the 100% LOEs. This means that for the 10 degrees step input demand the Kalman filter enhanced ANFIS control system is the fastest at getting to within 5% of the desired angle in most cases of faults, but as has been shown in the graphs it does not always achieve the desired angle. Unfortunately it is the most important thing for the AUV to achieve its desired yaw angle.

Table 4.1 The Rise Times For A Yaw Step Input of 10 Degrees.

		Rise Times (seconds)		
Type	LOE	PD Controller	ANFIS Controller	Kalman Filter
SAT	0%	5.4	4.5	4.4
	25%	5.4	4.5	4.4
	50%	5.4	4.5	4.4
	75%	5.6	4.7	4.6
	100%	8.1	7.2	7.9
RATE	0%	5.4	4.5	4.4
	25%	5.5	4.6	4.5
	50%	5.6	4.7	4.6
	75%	5.8	4.8	4.8
	100%	8.1	7.2	7.9
BOTH	0%	5.4	4.5	4.4
	25%	5.5	4.6	4.5
	50%	5.6	4.7	4.6
	75%	5.8	4.8	4.8
	100%	8.1	7.2	7.9

4.4.2. Yaw Step Inputs of 20 Degrees

(a) The PD Controller

For this size of step input demand the PD controller under a fault free set up displayed a small overshoot and a rise time of 4.4 seconds. For the saturation fault this system is totally unaffected for the 25% LOE. The 50% LOE reduces slightly the performance of the AUV. The 75% LOE produces a noticeable effect and for 100% LOE the AUVs performance decreased and the rise time has increased to 8.2 seconds. For the rate limiter fault the PD controller shows more signs of a fault occurring with small faults being noted for each level of LOE being considered. The 75% LOE also shows an increase in overshoot. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance and the 75% LOE again displays an increase overshoot. Again for 100% LOE the result is identical to previous 100% LOE's. The complete sets of results are displayed in Appendix F.

Figure 4.28 shows the response of the AUV when it is subjected to both faults occurring when the PD controller is used.

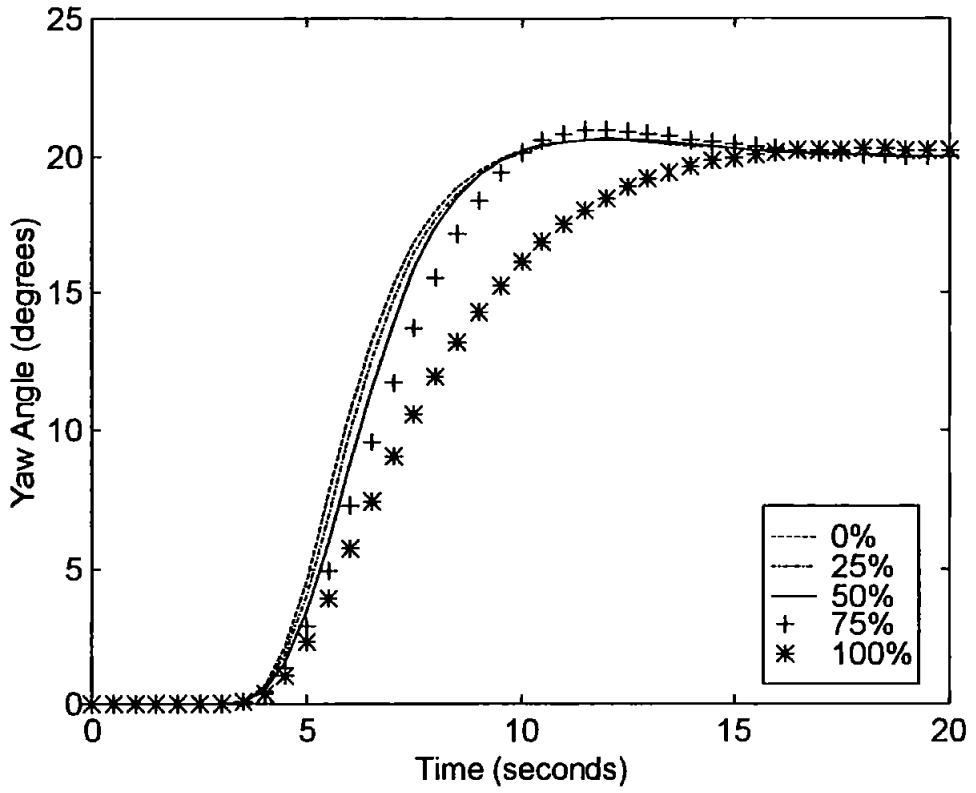


Figure 4.28 PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(b) The ANFIS Controller

The ANFIS controller for this size of step input demand under a fault free set up displayed no overshoot and a rise time of 4.9 seconds. For the saturation fault this system is totally unaffected for the 25% LOE. The 50% LOE produce a slight effect which slowed the response of the AUV. The 75% LOE produces a more noticeable effect. The 100% LOE produces the biggest effect on the response of the AUV, which has its rise time increased to 7.6 seconds. For the rate limiter fault the ANFIS controller shows more signs of a fault occurring with small faults being noted for each level of LOE being considered. The 75% LOE as well the decrease in performance also displays a slight overshoot. The 100% LOE is identical to that produced when the

saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance for all levels of LOE. Again for the 100% LOE the result is identical to previous 100% LOE's. The complete set of results are displayed in Appendix F. Figure 4.29 shows the response of the AUV when it is subjected to both faults occurring when the ANFIS controller is used.

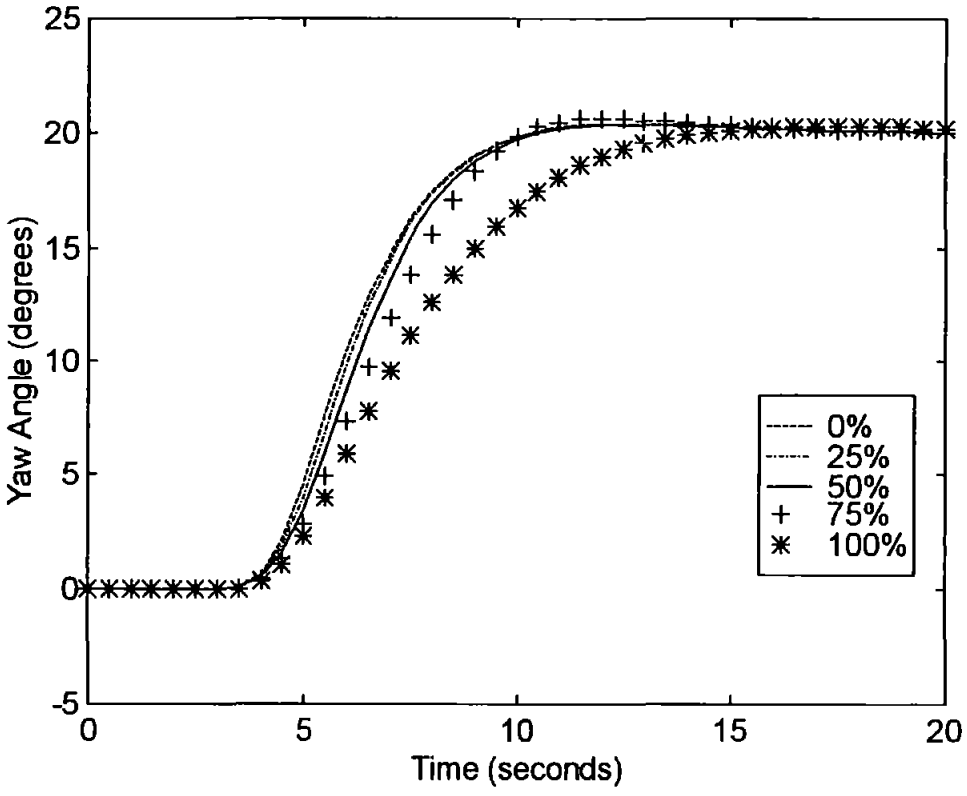


Figure 4.29 ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(c) The Kalman Filter Enhanced ANFIS Controller

When the Kalman filter is placed within the control system along with the ANFIS controller for this size of step input demand under a fault free set up displayed a slight overshoot and a rise time of 4.7 seconds. For the saturation fault this system is totally unaffected for the 25% LOE. The 50% LOE produces a small effect. The 75% LOE produced more of an effect with the final yaw angle being fractionally the desired 20

degrees demand. The 100% LOE produces the biggest effect on the AUV, which has its rise time reduced to 15.6 seconds and fails to reach the desired yaw angle. The final angle achieved by this system for this fault is 18.6 degrees. For the rate limiter fault the Kalman filter enhanced controller shows more signs of a fault occurring with small faults being noted for each level of LOE being considered. The 75% LOE once again produced the most significant overshoot. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show significant degradation in performance. Once again for 100% LOE the result is identical to previous 100% LOE's, this means that the Kalman filter enhanced controller fails to achieve the desired angle for all types of fault when a 100% LOE occurs. The results are for all of these simulations are displayed in Appendix F. Figure 4.30 shows the response of the AUV when it is subjected to both faults occurring when the Kalman filter enhanced ANFIS controller is used.

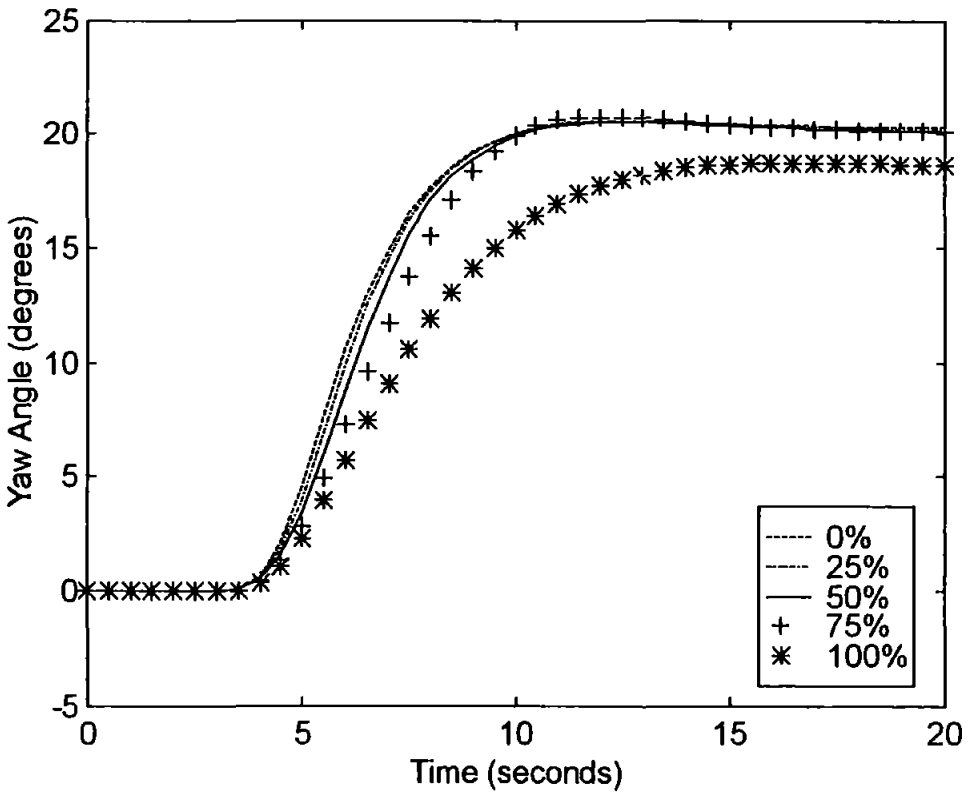


Figure 4.30 Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

Shown in Figure 4.31 is all three control systems attempting to handle a 100% LOE occurring in both sections of the actuator. Which shows particularly how the Kalman filter enhanced control system fails to reach the desired yaw angle.

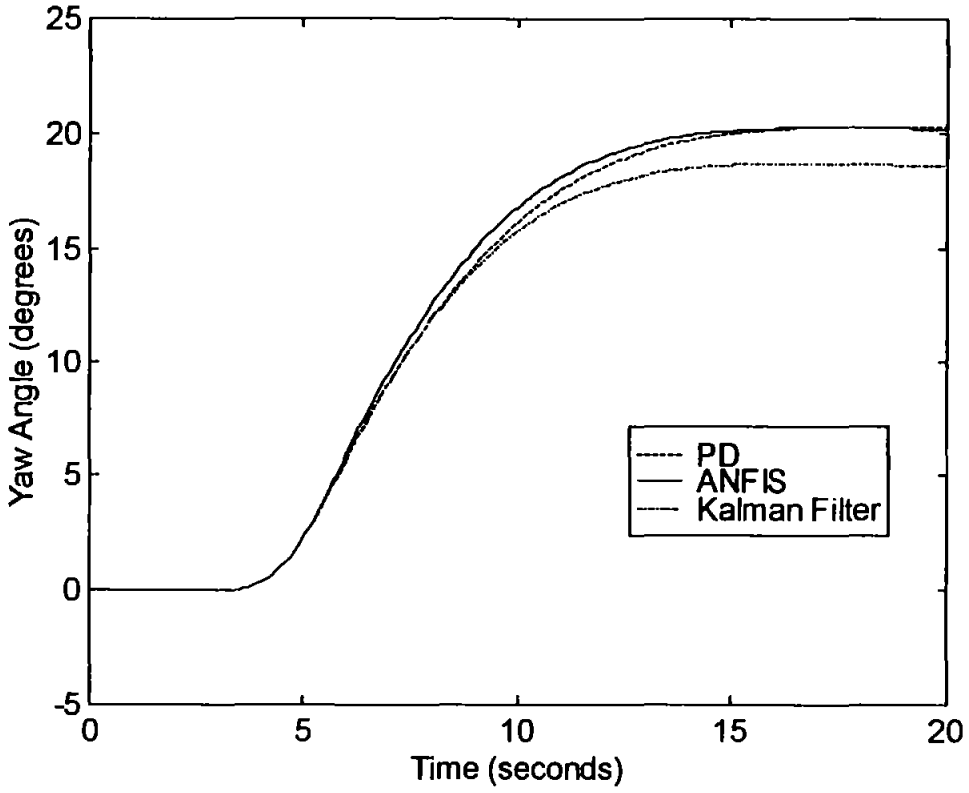


Figure 4.31 Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 20° Step Input Demand

Having seen how the three control systems handle the faults it is interesting to see the rise times of AUV for each test. These are shown in Table 4.2. If it was on rise times alone it is clear to see the Kalman filter enhanced ANFIS control system performed best of the three systems considered on everything but the 100% LOEs. This means again that for the 20 degrees step input demand the Kalman filter enhanced ANFIS control system is the fastest at getting to within 5% of the desired angle in most cases of faults, but as has been shown in the graphs it does not always achieve the desired angle. Unfortunately it is the most important thing for the AUV to achieve its desired yaw angle.

Table 4.2 The Rise Times For A Yaw Step Input of 20 Degrees.

		Rise Times (seconds)		
Type	LOE	PD Controller	ANFIS Controller	Kalman Filter
SAT	0%	6	6	4.7
	25%	6	6	4.7
	50%	6	6.1	4.8
	75%	6.5	6.6	5.4
	100%	8.7	9	15.6
RATE	0%	6	6	4.7
	25%	5.9	6	4.7
	50%	5.9	6.2	4.8
	75%	6.2	6.2	4.9
	100%	8.7	9	15.6
BOTH	0%	6	6	4.7
	25%	5.9	6	4.7
	50%	5.9	6.2	4.8
	75%	6.3	6.3	5
	100%	8.7	9	15.6

4.4.3. Yaw Step Inputs of 30 Degrees

(a) The PD Controller

For this size of step input demand the PD controller under a fault free set up displayed a small overshoot of approximately one degree and a rise time of 4.8 seconds. For the saturation fault this system is affected for all levels of LOE. As would be expected from the previous results as the LOE increases the performance of the AUV decreases. With the 100% LOE giving the worst result and the AUVs rise time has been increased to 8.7 seconds. For the rate limiter fault the PD controller shows more signs of a fault occurring with faults being noted for each level of LOE being considered. For this size of step input there are now more overshoots being detected, with them being clearly viable for all but the 100% LOE. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance and overshoots increasing for each level of LOE apart from the 100%

which again is identical to previous 100% LOE's. The results for all of these are displayed in Appendix F. Figure 4.32 shows the response of the AUV when it is subjected to both faults occurring when the PD controller is used.

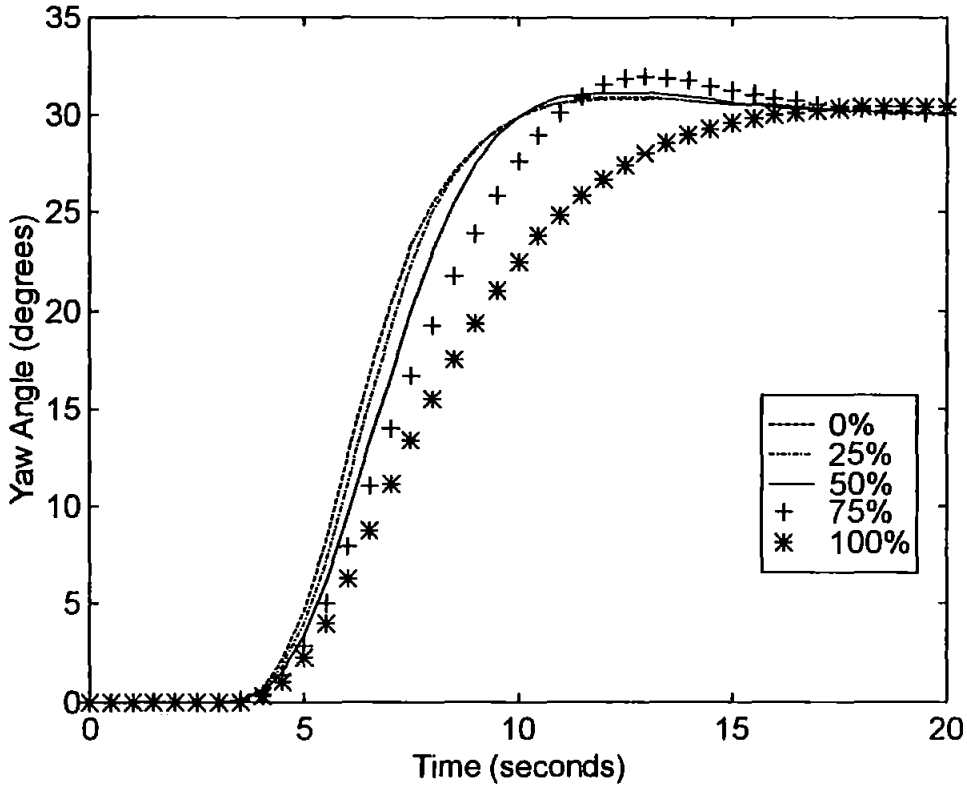


Figure 4.32 PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(b) The ANFIS Controller

The ANFIS controller for this size of step input demand under a fault free set up displayed no overshoot and a rise time of 4.9 seconds. For the saturation fault this system shows decreases in performance for all levels of LOE. As would be expected from the previous results as the LOE increases the performance of the AUV decreases. The 100% LOE produces the biggest effect on the AUV, which has its rise time reduced to 7.6 seconds. Unlike for the PD controller, the AUV does not overshoot the demand yaw angle for any level of LOE for this fault. For the rate limiter fault the ANFIS controller shows more signs of a fault occurring with degrading of performance being noted for each level of LOE being considered. Also

overshoots were detected for all levels of LOE apart from the 100% LOE. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show degradation in performance and overshoots. Again for 100% LOE the result is identical to previous 100% LOE's. These results are all displayed in Appendix F. Figure 4.33 shows the response of the AUV when it is subjected to both faults occurring when the ANFIS controller is used.

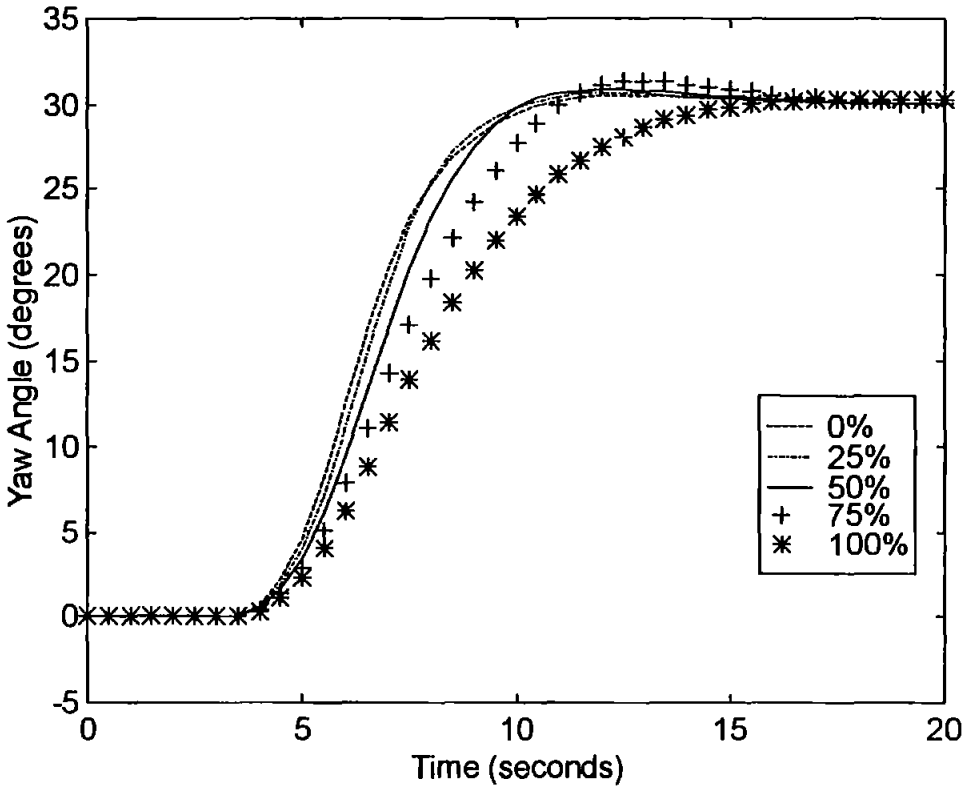


Figure 4.33 ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

(c) The Kalman Filter Enhanced ANFIS Controller

Finally for this section, when the Kalman filter is placed within the control system along with the ANFIS controller for this size of step input demand under a fault free set-up displayed almost no overshoot and a rise time of 4.7 seconds. For the saturation fault this system is slightly effected by the 25% LOE. The 50% LOE produces a more noticeable effect. The 75% LOE produced more of an effect with the final yaw angle

being fractionally the desired 30 degrees demand. The 100% LOE produces the biggest effect on the AUV, which has its rise time reduced to 15.3 seconds and fails to reach the desired yaw angle. The final yaw angle achieved by this system for this fault is 25.8 degrees. For the rate limiter fault the Kalman filter enhanced controller shows more signs of a fault occurring with small faults being noted for each level of LOE being considered. Overshoots were recorded for all but the 100% LOE, with the 75% LOE once again producing the most significant overshoot. The 100% LOE is identical to that produced when the saturation fault of 100% LOE is implemented. When a LOE occurs in both sections of the actuator the AUV responses show significant degradation in performance but with less overshoots than for a LOE occurring in just the saturation section. Once again for 100% LOE the result is identical to previous 100% LOE's, this means that the Kalman filter enhanced controller fails to achieve the desired angle for all types of fault when a 100% LOE occurs. The results for all of these simulations are displayed in Appendix F. Figure 4.34 shows the response of the AUV when it is subjected to both faults occurring when the ANFIS controller is used.

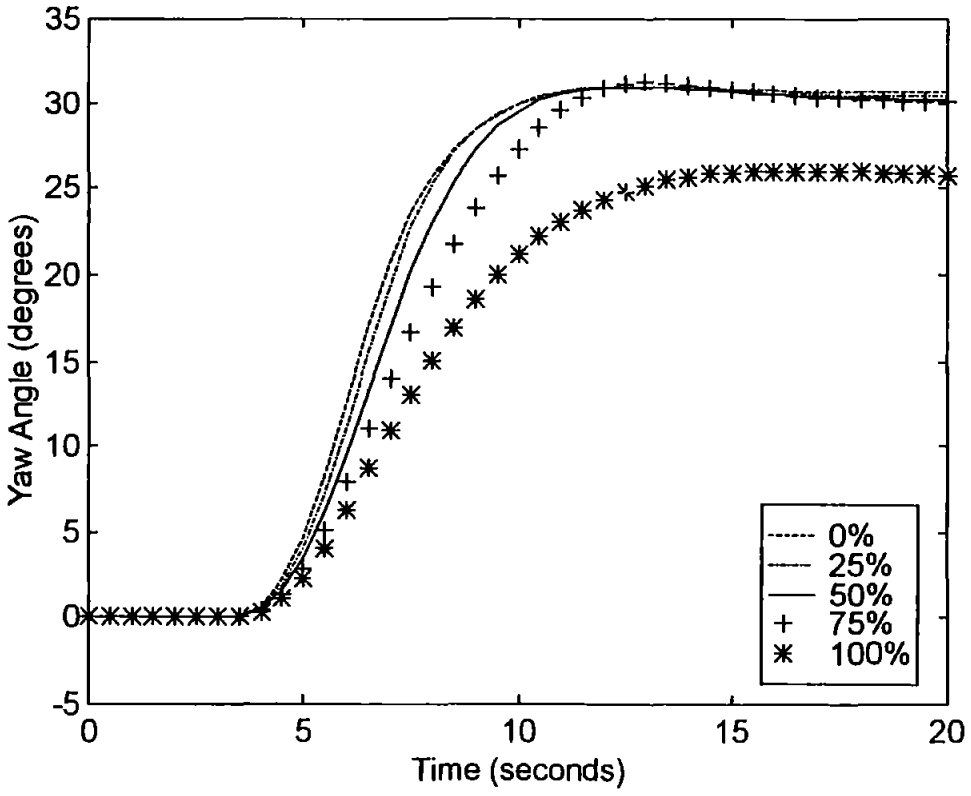


Figure 4.34 Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

Figure 4.35 shows all three control systems attempting to handle a 100% LOE occurring in both sections of the actuator.

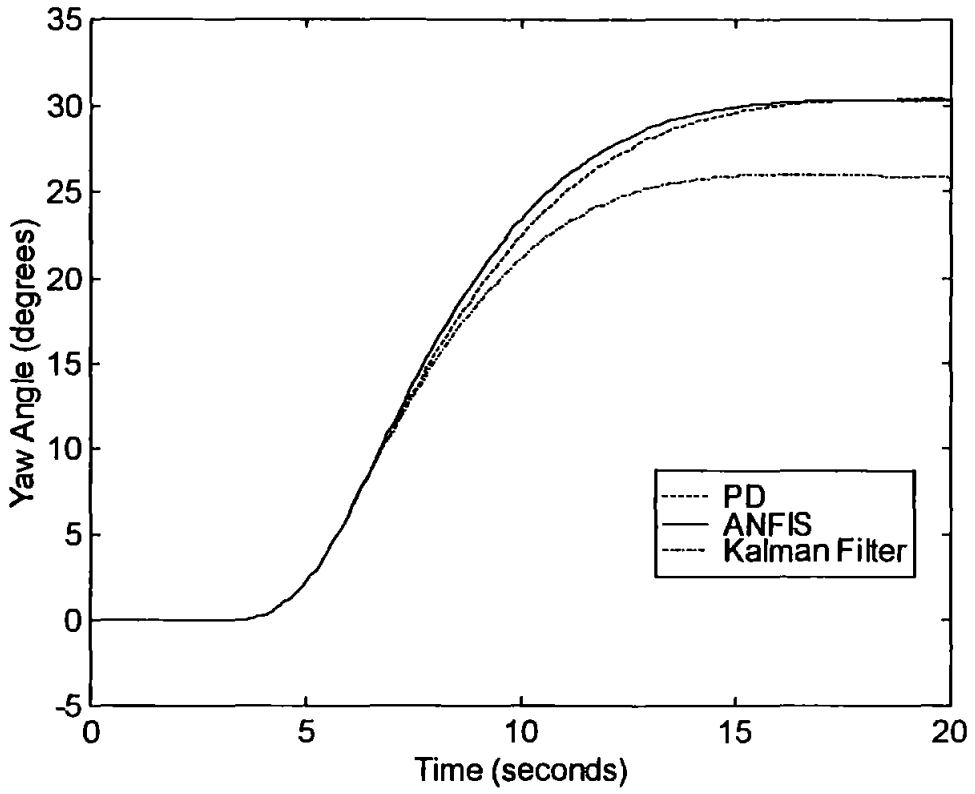


Figure 4.35 Yaw Angle Responses of All Three Control Systems to a 100% LOE Both Fault for a 30° Step Input Demand

Having seen how the three control systems handle the faults it is interesting to note the rise times of AUV for each test. These are shown in Table 4.3. As was the case for the other step inputs, if it was on rise times alone it is clear to see the Kalman filter enhanced ANFIS control system performed best of the three systems considered on everything but the 100% LOEs. This means again that for the 30 degrees step input demand the Kalman filter enhanced ANFIS control system is the quickest at getting to within 5% of the desired angle in most cases of faults. As has been shown in the graphs the ANFIS control system does not always achieve the desired angle. Unfortunately it is the most important thing for the AUV to achieve its desired yaw angle.

Table 4.3 The Rise Times For A Yaw Step Input of 30 Degrees.

		Rise Times (seconds)		
Type	LOE	PD Controller	ANFIS Controller	Kalman Filter
SAT	0%	6.2	6.2	4.7
	25%	6.2	6.3	4.8
	50%	6.5	6.7	5.3
	75%	7.4	7.5	6.2
	100%	9.5	9.8	15.3
RATE	0%	6.2	6.2	4.7
	25%	5.6	5.9	4.7
	50%	5.8	6	4.9
	75%	6.7	6.9	5.4
	100%	9.5	9.8	15.3
BOTH	0%	6.2	6.2	4.7
	25%	5.6	6	4.7
	50%	6.1	6.3	5
	75%	7.2	7.3	5.9
	100%	9.5	9.8	15.3

Having considered all of the failures defined during Chapter 3, this concludes the section dealing with actuator failures.

The graphs shown are representative of the results achieved and it is worth pointing out for interested reader that the full set of results can be seen in Appendix F.

4.5. CONCLUSIONS

This Chapter has shown how the ANFIS controller and the Kalman filter enhanced ANFIS controller coped with sensor failures in both the yaw and roll channels and how these controllers and the standard PD controller coped with failures in the actuators controlling the yaw motions of the AUV.

When considering work in the yaw channel for the fault free system the Kalman filter enhanced ANFIS controller made the AUV attain and then maintain a yaw angle close to the correct yaw angle for most of the tests. When a small error was detected this was due to the linear model being an imperfect representation of the AUV.

When the yaw sensor failures were considered the Kalman filter enhanced control system was able to recover partially for most levels of the failures. The ANFIS controller could not handle this failure and produced some poor results as shown in section 4.3.1.

When yaw rate sensor failures were considered the affect on the AUV was far less noticeable. The main reason for this is that the ANFIS controller is based predominantly on the yaw sensor feedback and the yaw rate sensor is used as a dampener. As the level of fault was increased there was a decrease in performance. The Kalman filter controller produced some very good results for the larger step input demands (50 and 90 degrees).

The work in the roll channel for the fault free system the Kalman filter enhanced controller showed clear signs of problems in all tests. An oscillatory motion was detected in even the fault free tests and was a continued feature of the responses given by this system.

When roll sensor failures were considered the Kalman filter enhanced control system was able recover to a certain degree for most levels of the failures. The results also showed that this system was unaffected by the size of type of failure occurring. The ANFIS controller could not handle this failure and produced a very noticeable degrading of performance.

When roll rate sensor failures were considered the affect on the AUV was far more noticeable. The main reason for this was that the ANFIS controller is based predominantly on the roll rate sensor feedback, rather than the roll sensor. As the level of fault was increased there was a decrease in performance with the ANFIS controller forcing the AUV to oscillate to angles which are a high percentage of the initial angles. The Kalman filter controller produced results similar to those for the roll sensor failures, again being almost unaffected by the type or size of fault occurring.

For the set of actuator failures in the upper canard, which were discussed in section 4.4, the outcome is quite different. The PD controller was able to recover for all types and sizes of faults for all three angles of yaw demands and succeeded in having the AUV at the required yaw angle by the end of every test. The rise times did increase for every type of failure at every level. The ANFIS control system also managed to recover for all types and sizes of failures, but once again the rise times of the vehicles were effected. The Kalman filter enhanced ANFIS controller was also able to produce some level of recovery for all types and sizes of faults. The key feature of the Kalman

filter results was the inability of the system to achieve the desired yaw angle when a large LOE was occurring. This was due to the Kalman filters dependence on the linear model. When an actuator fault occurs within the AUV, it changes the dynamics of the AUV and hence a new linear model of the system is required by the Kalman filter in order for it to continue operating effectively.

In summary, the Kalman filter control system was far more fault tolerant than the standard ANFIS control system for the sensor faults. For the actuator faults the Kalman filter enhanced ANFIS controller was less fault tolerant than the standard ANFIS controller and the PD controller. For further work it will be best, therefore, to use the Kalman filter enhanced ANFIS control system as a benchmark when considering sensor failures. As the ANFIS controller is to be considered as part of the fault tolerant system for the actuator faults, and the Kalman filter enhanced ANFIS control system was out performed by the PD controller on the actuator faults, then for the actuator faults it will be best to use the PD controller as the benchmark controller.

CHAPTER 5

SENSOR RECOVERY SYSTEM

5.1. INTRODUCTION

Chapter 4 established benchmark results for all types and levels of faults being considered within this work. The sensor faults considered therein are now revisited with the aim of producing a fault tolerant control system capable of improving on the benchmark results.

This will be accomplished by placing a FIS within the control loop in a style similar to that used with the Kalman filter within the control loop in Chapter 4. This Chapter will see the creation of eight fault tolerant FISs, two for each of the four sensors being considered. The four sensors are the yaw sensor, the yaw rate sensor, the roll sensor and the roll rate sensor. An ANFIS [Jang (1991)] tuned FIS will be created for each sensor, as will a simulated annealing [Kirkpatrick et *al* (1983)] tuned FIS. Therefore there will be a total of eight FISs.

These eight FIS will be presented with the sensor faults as stated in Chapter 3. They are a percentage loss of signal, an intermittent total signal failure and the addition of random white noise. Such faults are considered to be representative of faults that would occur within a typical sensor used in an AUV. The results for each sensor will be compared for both tuned FISs and the benchmark results.

5.2. FAULT TOLERANT SYSTEM

The nature of the faults being considered in this Chapter are to corrupt the information from the AUV's sensors which the ANFIS controller receives. Therefore it is logical to place the sensor fault recovery FIS after the vehicle dynamics but before the ANFIS controller. The FIS will replace the corrupted information with an estimate of the sensor signal allowing the AUV to continue in its mission. The ANFIS controller used is again that developed by Craven (1999). The placing of this sensor recovery

FIS in the control loop will give the overall control system, and hence the AUV, a level of fault tolerance to the failures which are being considered in this thesis. Figure 5.1 shows how the FIS is placed in the control loop.

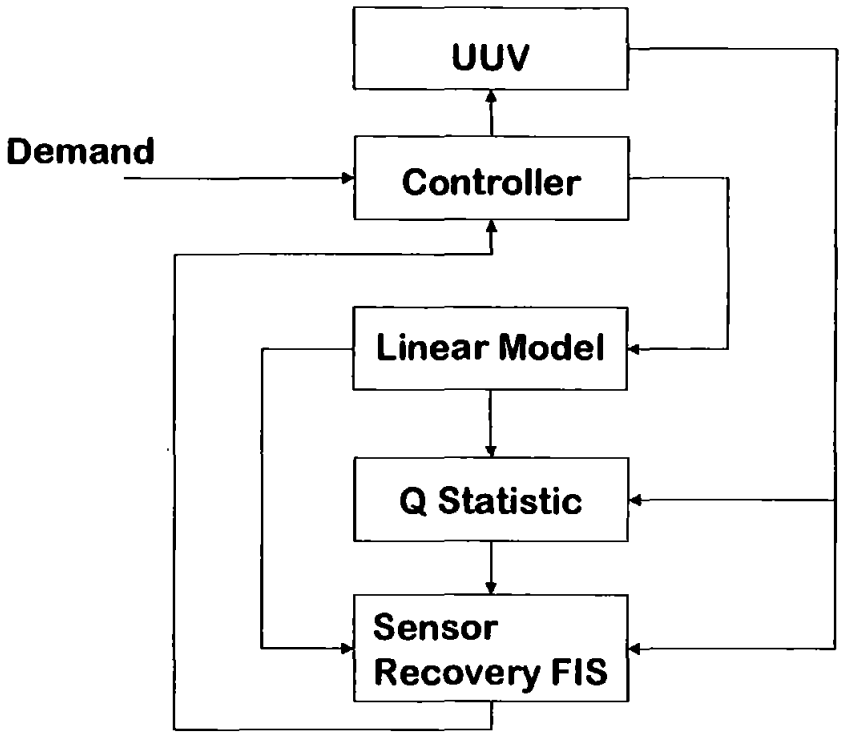


Figure 5.1 The Sensor Fault Tolerant Control System

Figure 5.1 shows how the sensor information is replaced by an estimate from the FIS. It also shows that the FIS requires three inputs, one from the faulty sensor, a second from the linear model and the third from a block labelled 'Q Statistic'. The first input is there so that the FIS knows the signal being produced by the actual sensor. The second input is from the linear model developed in Chapter 3 and is there to provide the FIS with an estimate of the sensor signal based on the output of the controller and will be discussed in section 5.3. The third input is the Q statistic used by the Kalman filter, and is used here to provide a statistical assessment of the level of fault in the actual sensor. Figure 5.2 shows the input output structure for the FIS being considered, with the output being the FISs estimate of the sensor signal.

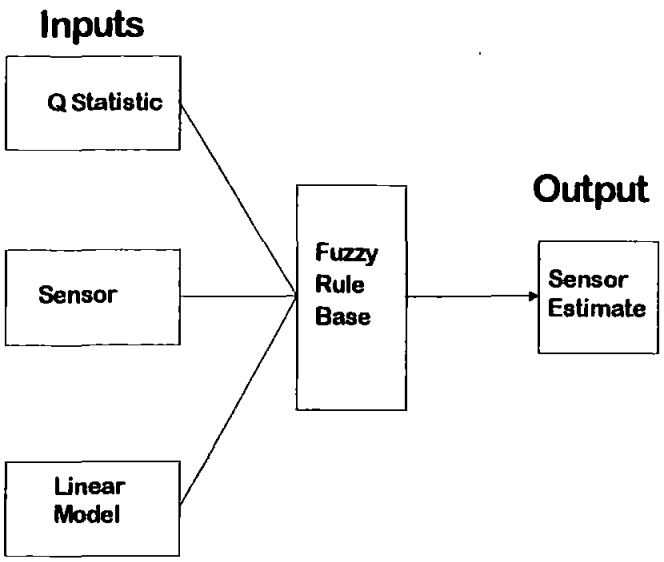


Figure 5.2 The Input Output Structure of The Sensor Recovery FIS

The faults being considered are percentage sensor failures of 100%, 75%, 50% and 25%, an intermittent total signal failure and adding noise to the sensor signal. The full description of these faults and how they are implemented within the AUV is given in Chapter 3. For the yaw channel all three types of sensor failure were considered for each sensor (yaw and yaw rate) individually over three sizes of step input demands (10, 50, and 90 degrees). While for the roll channel all three types of sensor failure were considered for each sensor (roll and roll rate) individually, but instead of a demanded step input, three initial roll angles were used (5,15, and 25 degrees) with the demanded roll angle being zero degrees.

The tuned FISs are to be compared using both a statistical method and by considering rise times. The statistical method used to evaluate the performance of the FISs was the root mean squared error (RMSE). This will give the average difference between the fault free ideal system and the sensor recovery FISs performance over the considered path. The equation used to calculate the RMSE is shown in Equation (5.1).

$$E = \sqrt{\frac{\sum_{i=0}^{i=n} (AUV_i - IDEAL_i)^2}{n}} \tag{5.1}$$

Where E is the error value, n is the number of points measured, AUV_i is the information received from the vehicle at point i, and $IDEAL_i$ is the information recorded when the vehicle followed the same path in a fault free situation also at point i.

5.3. Q STATISTIC

The Q statistic used herein is that of the error covariance matrix used by the Kalman filter.

By taking into account the state and the expected value of the system it is possible to derive the error covariance matrix from Equation (5.2).

$$
\begin{aligned}
\mathbf{P}(t) &= \mathrm{E}\left\langle \left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t)\rangle\right]\left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t)\rangle\right]^{\mathrm{T}} \right\rangle \\
&= \mathrm{E}\left\langle \left[\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] + \omega(t-1)\right]\left[\boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] + \omega(t-1)\right]^{\mathrm{T}} \right\rangle \\
&= \mathrm{E}\left\langle \begin{array}{l} \boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \\ \boldsymbol{\Phi}(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] \omega^{\mathrm{T}}(t-1) + \\ \omega(t)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}}(t-1) \\ + \omega(t-1)\omega^{\mathrm{T}}(t-1) \end{array} \right\rangle \\
&= \boldsymbol{\Phi}(t-1)\mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle \boldsymbol{\Phi}^{\mathrm{T}}(t-1) \\
&\quad + \mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right] \omega^{\mathrm{T}}(t-1) \right\rangle \\
&\quad + \mathrm{E}\left\langle \omega(t-1)\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle \boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \mathrm{E}\left\langle \omega(t-1)\omega^{\mathrm{T}}(t-1) \right\rangle
\end{aligned}
$$

as $\mathbf{P}(t-1) = \mathrm{E}\left\langle \left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]\left[\mathbf{X}(t-1) - \mathrm{E}\langle \mathbf{X}(t-1)\rangle\right]^{\mathrm{T}} \right\rangle$

and letting $\mathbf{Q}(t-1) = \mathrm{E}\left\langle \omega(t-1)\omega^{\mathrm{T}}(t-1) \right\rangle$

$$\mathbf{P}(t) = \boldsymbol{\Phi}(t-1)\mathbf{P}(t-1)\,\boldsymbol{\Phi}^{\mathrm{T}}(t-1) + \mathbf{Q}(t-1) \tag{5.2}$$

Where **P** is the Q statistic (error covariance matrix), Φ is the transition matrix and t is time. As Q(t-1)=0, this will then give the equation used within the Kalman filter program and is shown in Equation (5.3). This has shown how the Q statistic is derived from the second moment of the state.

$$\mathbf{P}(t) = \mathbf{\Phi}(t-1)\mathbf{P}(t-1)\mathbf{\Phi}(t-1)^{\mathrm{T}} \tag{5.3}$$

The Q statistic is used as an error measurement of the sensor in the work below.

5.4. FUZZY TUNING

The FISs used as sensor recovery systems in this Chapter have been tuned using two methods. The first method used is the ANFIS [Jang (1991)] method and the second is the simulated annealing [Kirkpatrick et *al* (1983)] method.

The ANFIS method includes a subroutine to create an initial FIS from the training data and hence there is no need for an initial heuristic FIS to be developed for this approach. However the simulated annealing tuning method does not include such a subroutine and therefore an initial heuristic FIS is required.

A heuristic FIS has been design using a simple yet logical approach to the problem. The Q statistic provides a value between zero and one, which is a statistical measure of the fault within the sensor signal. This value is one when the signal is fault free and zero when the sensor signal is totally unusable. First by letting the fuzzy output be the sensor input when the Q value is small, then by letting the fuzzy output be half of the sensor input and half of the linear model input when the Q value is medium and by letting the fuzzy output be the linear model input when the Q value is large the three rule fuzzy rule base show in Equation (5.4) was produced.

If Q is S and SEN is N and LS is N then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is N and LS is Z then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is N and LS is P then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is Z and LS is N then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is Z and LS is Z then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is Z and LS is P then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is P and LS is N then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is P and LS is Z then SE = 0 Q + 1 SEN + 0 LS +0
If Q is S and SEN is P and LS is P then SE = 0 Q + 1 SEN + 0 LS +0
If Q is M and SEN is N and LS is N then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is N and LS is Z then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is N and LS is P then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is Z and LS is N then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is Z and LS is Z then SE = 0 Q + 0.5 SEN + 0.5 LS +0 (5.4)
If Q is M and SEN is Z and LS is P then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is P and LS is N then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is P and LS is Z then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is M and SEN is P and LS is P then SE = 0 Q + 0.5 SEN + 0.5 LS +0
If Q is L and SEN is N and LS is N then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is N and LS is Z then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is N and LS is P then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is Z and LS is N then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is Z and LS is Z then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is Z and LS is P then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is P and LS is N then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is P and LS is Z then SE = 0 Q + 0 SEN + 1 LS +0
If Q is L and SEN is P and LS is P then SE = 0 Q + 0 SEN + 1 LS +0

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The Q input is defined as having a maximum value of 1 and a minimum value of 0. There are three membership functions for this input which are all generalised bell

curve membership functions and are shown in Figure 5.3. These are the heuristic membership functions for this input, but the tuned membership functions are derived from these and have similar features.

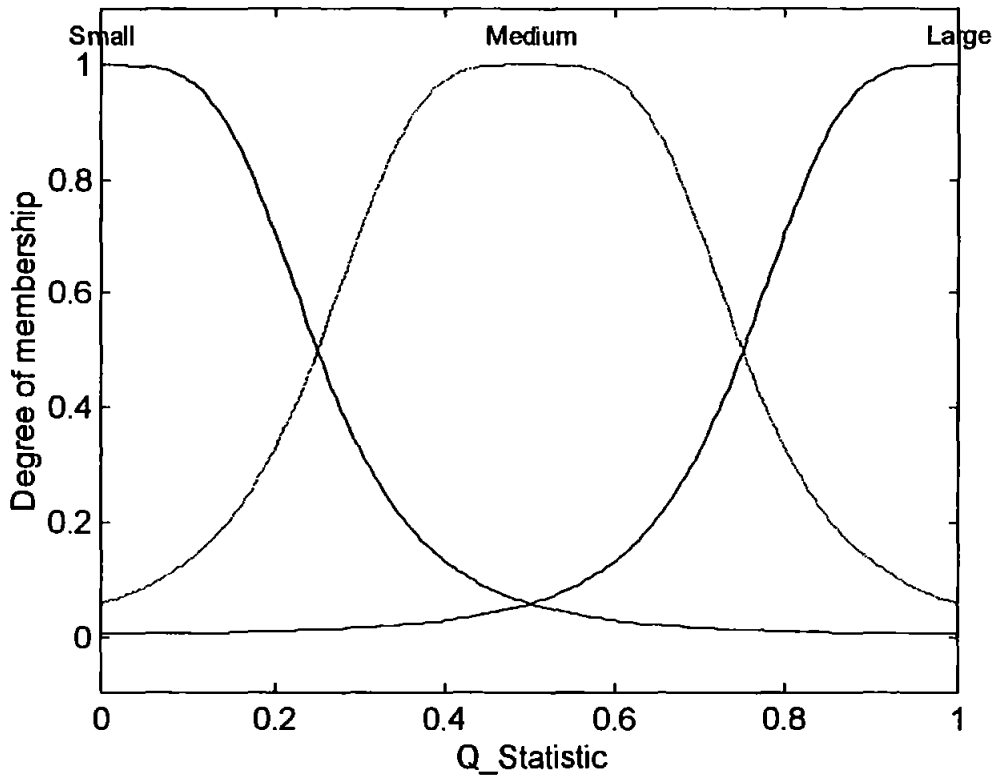


Figure 5.3 Heuristic Membership Functions for Q Input

The sensor input is defined as having a maximum value of 250 degrees and a minimum value of –250 degrees for the yaw sensor. The sensor input is defined as having a maximum value of 15 degrees per second and a minimum value of –15 degrees per second for the yaw rate sensor. The sensor input is defined as having a maximum value of 35 degrees and a minimum value of –35 degrees for the roll sensor. The sensor input is defined as having a maximum value of 35 degrees per second and a minimum value of –35 degrees per second for the roll rate sensor. There are three membership functions for this input which are all generalised bell curve membership functions and the ones for the yaw sensor are shown in Figure 5.4. These are the heuristic membership functions for this input, but the tuned membership functions are derived from these and have similar features.

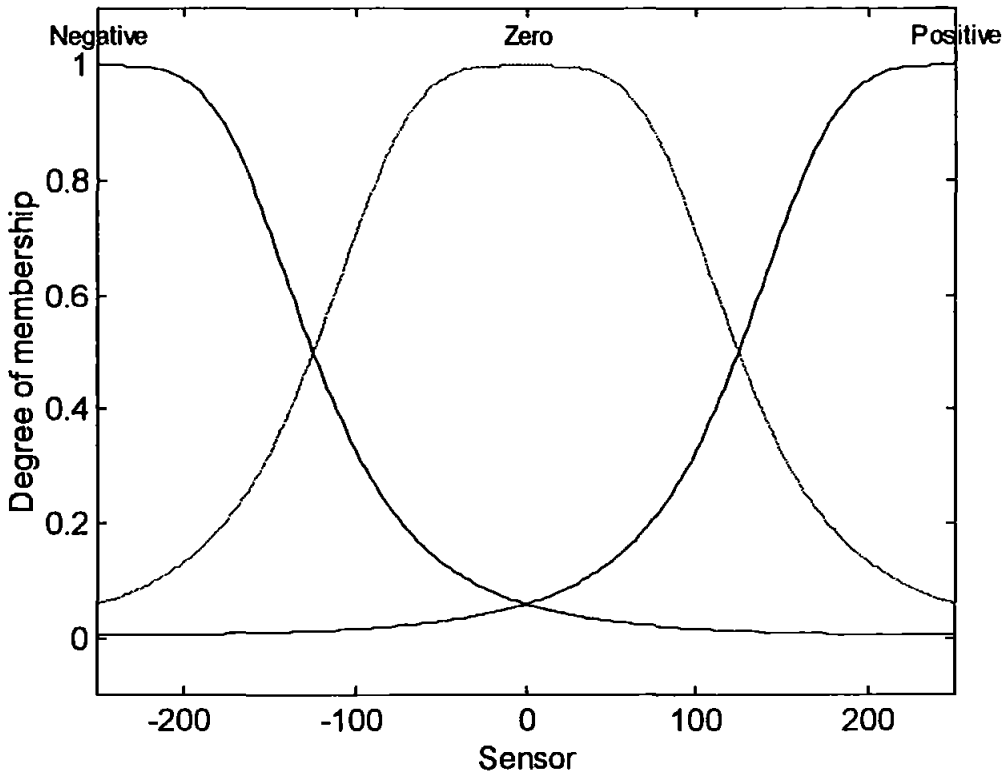


Figure 5.4 Heuristic Membership Functions for Sensor Input

The linear model sensor input is defined as having a maximum value of 250 degrees and a minimum value of –250 degrees for the yaw sensor. The linear model sensor input is defined as having a maximum value of 15 degrees per second and having a minimum value of –15 degrees per second for the yaw rate sensor. The linear model sensor input is defined as having a maximum value of 35 degrees and a minimum value of –35 degrees for the roll sensor. The linear model sensor input is defined as having a maximum value of 35 degrees per second and a minimum value of -35 degrees per second for the roll rate sensor. There are three membership functions for this input which are all generalised bell curve membership functions and the ones for the yaw sensor are shown in Figure 5.5. These are the heuristic membership functions for this input, but the tuned membership functions are derived from these and have similar features.

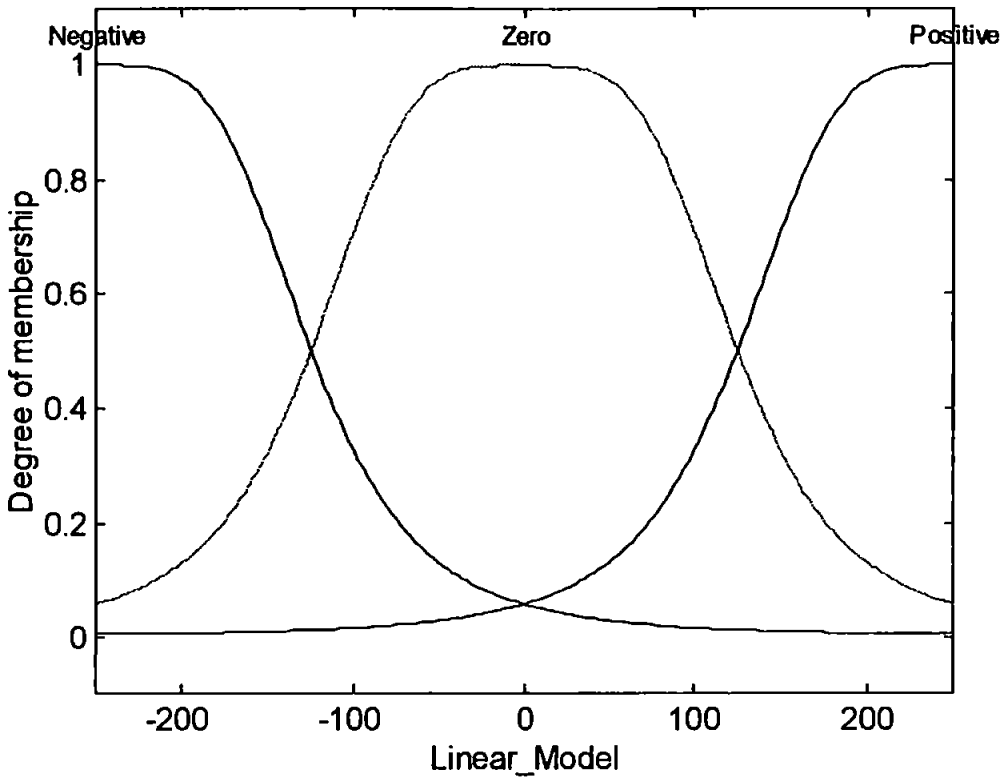


Figure 5.5 Heuristic Membership Functions for Linear Sensor Input

The heuristic FIS was then tuned by the simulated annealing method using training data. The training data was obtained by running the AUV over the path in Figure 5.6 for the yaw channel sensor recovery FISs.

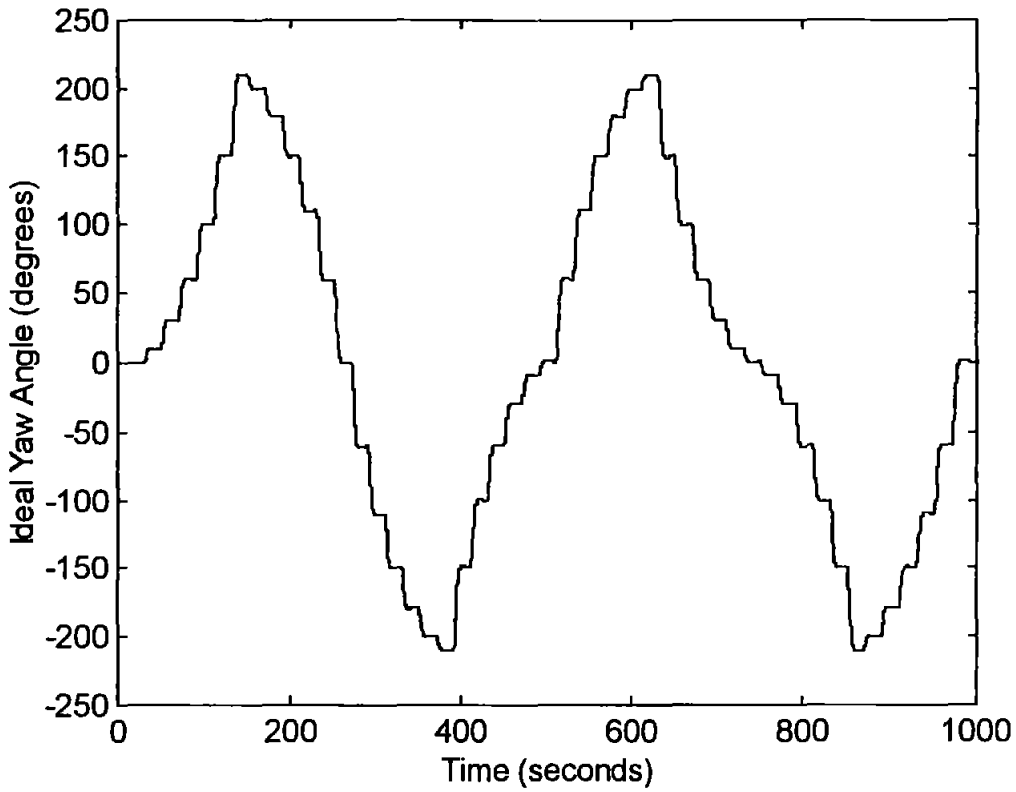


Figure 5.6 AUV Yaw Response to Training Path

The AUV was presented with forty-eight step input demands which lead to the response shown in Figure 5.6 when there was no fault occurring. This set of data represents the ideal output of the FIS when the AUV is given the same set of step input demands. For training purposes the failure percentages considered were 0%, 33.3%, 66.7% and 100%. This produced a training path of 4000 seconds with data being recorded every 0.1 seconds. The linear model was presented with the output from the ANFIS controller and the linear model's output was recorded. The Q statistic was calculated from the sensor and linear model output. This gives the three sets of input data and the output data required to tune the FISs. With the output data being the fault free response of the AUV. These three input data sets and the output data set can be seen in Figures 5.7, 5.8, 5.9 and 5.10 for the yaw sensor.

Figure 5.7 shows for each level of fault the Q statistic initially has a value of zero and the value increases as the sensor signal and linear model signal drift apart. For the 0% failure the Q value settles at around 0.7, for the 33.3% failure the Q value is around 0.55. For the 66.7% failure the Q value settles at around 0.95 and for the 100% failure the value is almost 1, this means that the Q statistic considers the sensor information to be almost unusable.

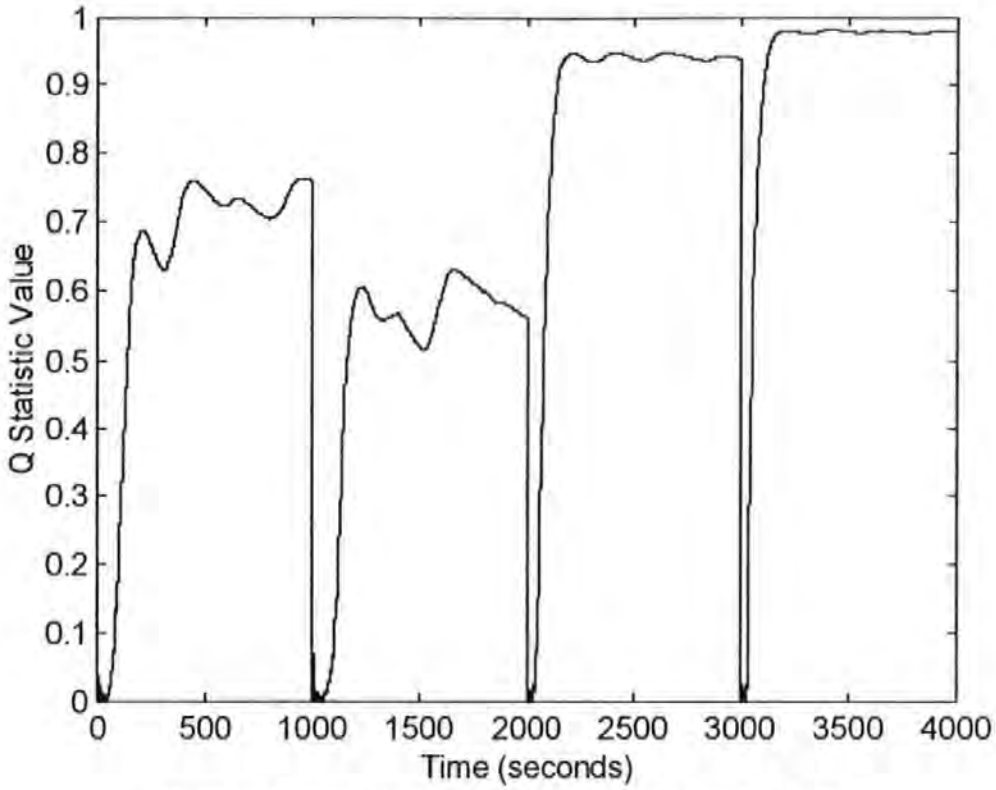


Figure 5.7 Complete Yaw Q Statistic Input Data Set

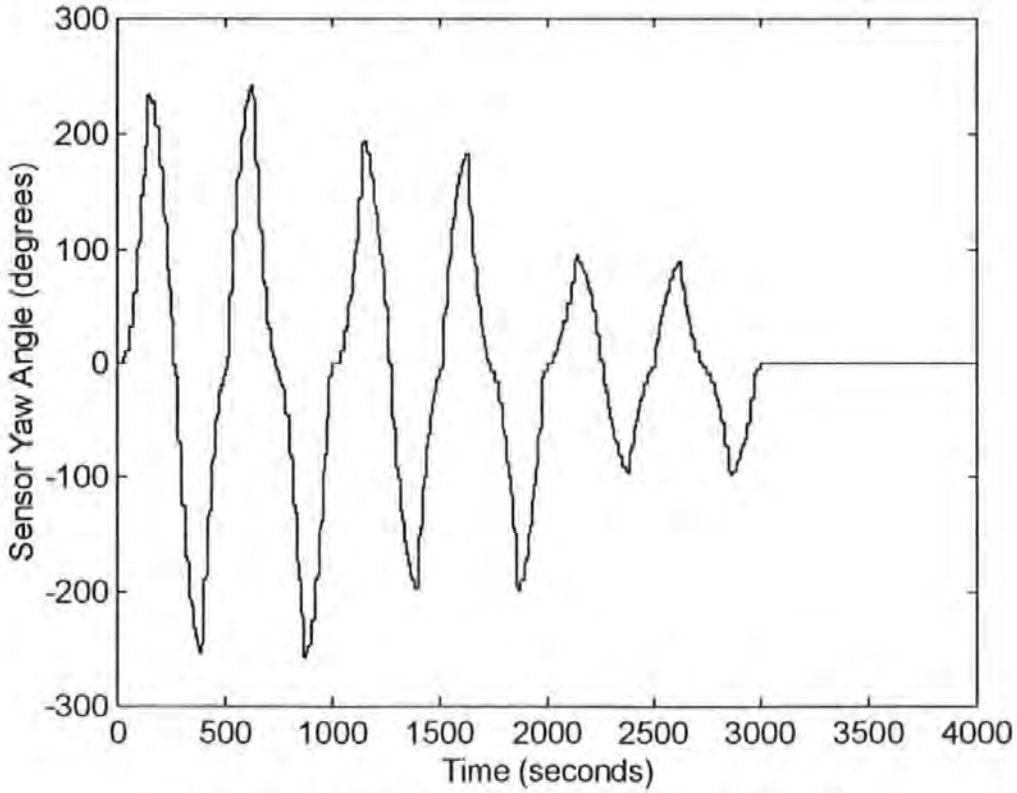


Figure 5.8 Complete Yaw Sensor Input Data Set

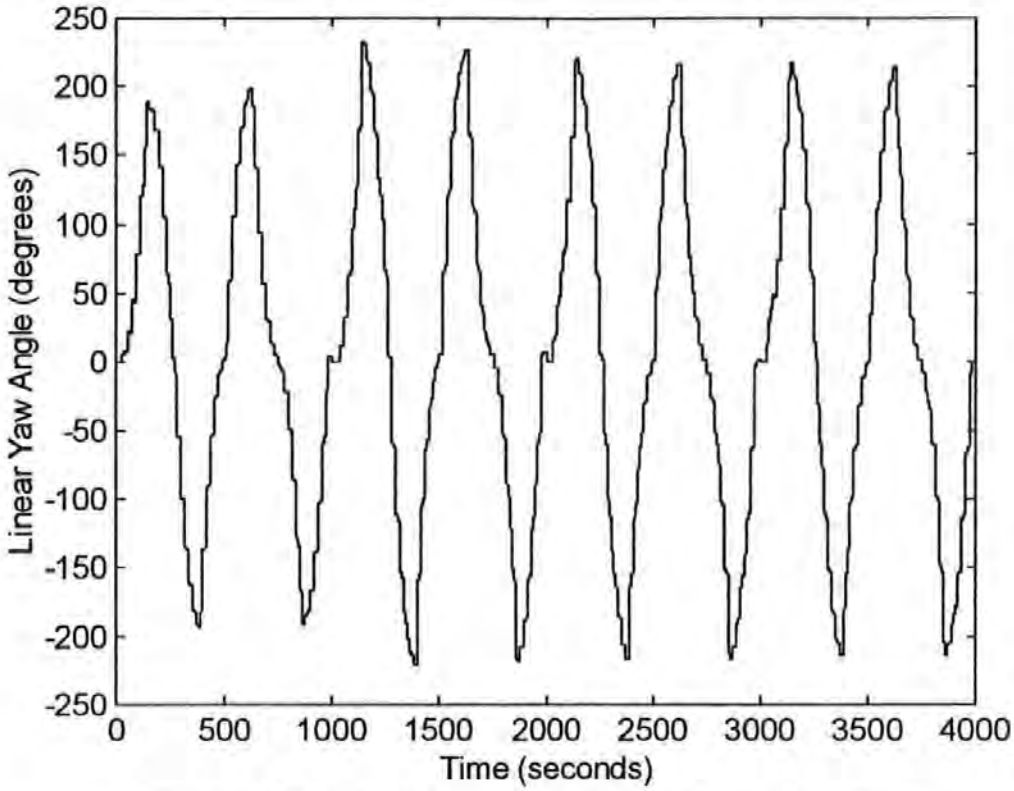


Figure 5.9 Complete Linear Model Yaw Input Data Set

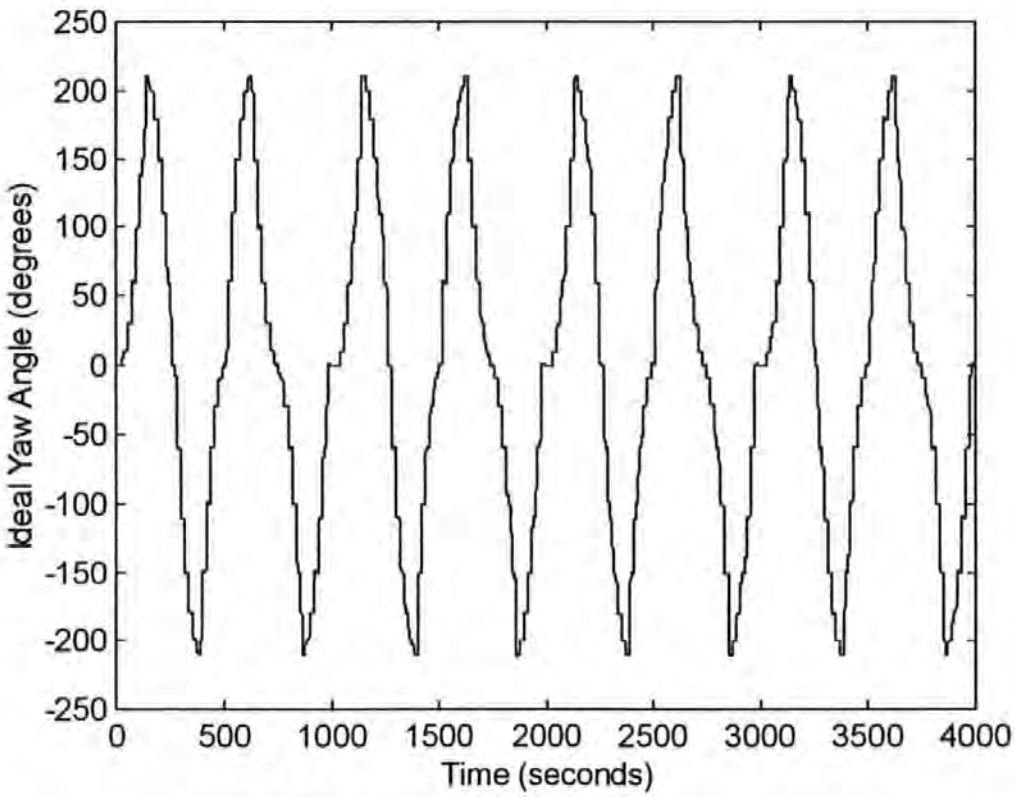


Figure 5.10 Complete Yaw Sensor Output Data Set

For the yaw rate sensor the same step input demands shown in Figure 5.6 were used, but the yaw rate information was recorded and used for tuning. The tuning data can be seen in Figures 5.11, 5.12, 5.13 and 5.14.

Figure 5.11 shows that for each level of fault the Q statistic initially has a value of zero and as each simulation progress the value increases as the sensor signal and linear model signal drift apart. For the 0% failure the Q value settles at around 0.7, for the 33.3% failure the Q value is around 0.55. For the 66.7% failure the Q value settles at around 0.95 and for the 100% failure the value is almost 1, this means that the Q statistic considers the sensor information to be almost unusable. These results are very similar to those obtained when considering faults in the yaw sensor.

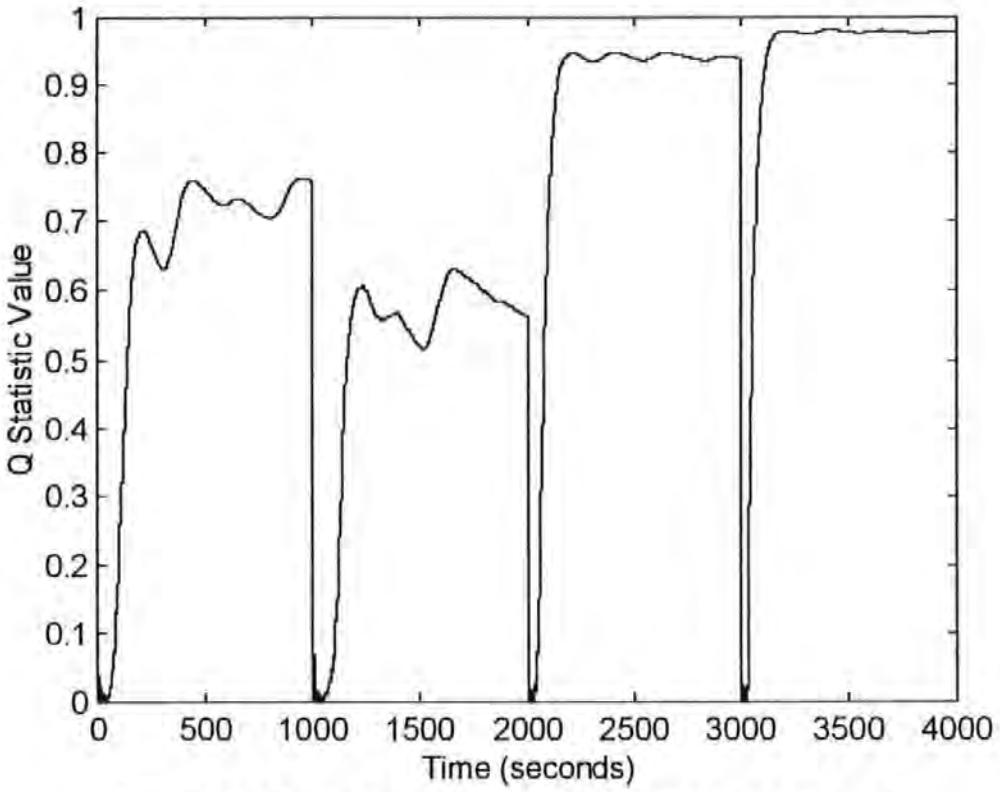


Figure 5.11 Complete Yaw Rate Q Statistic Input Data Set

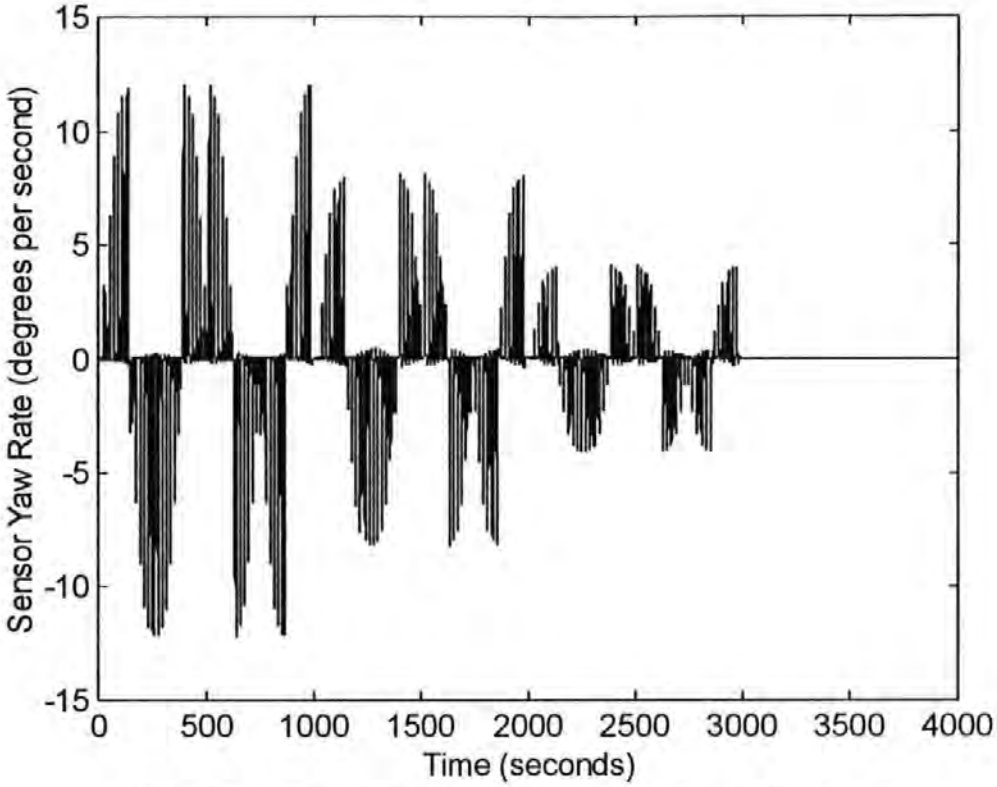


Figure 5.12 Complete Yaw Rate Sensor Input Data Set

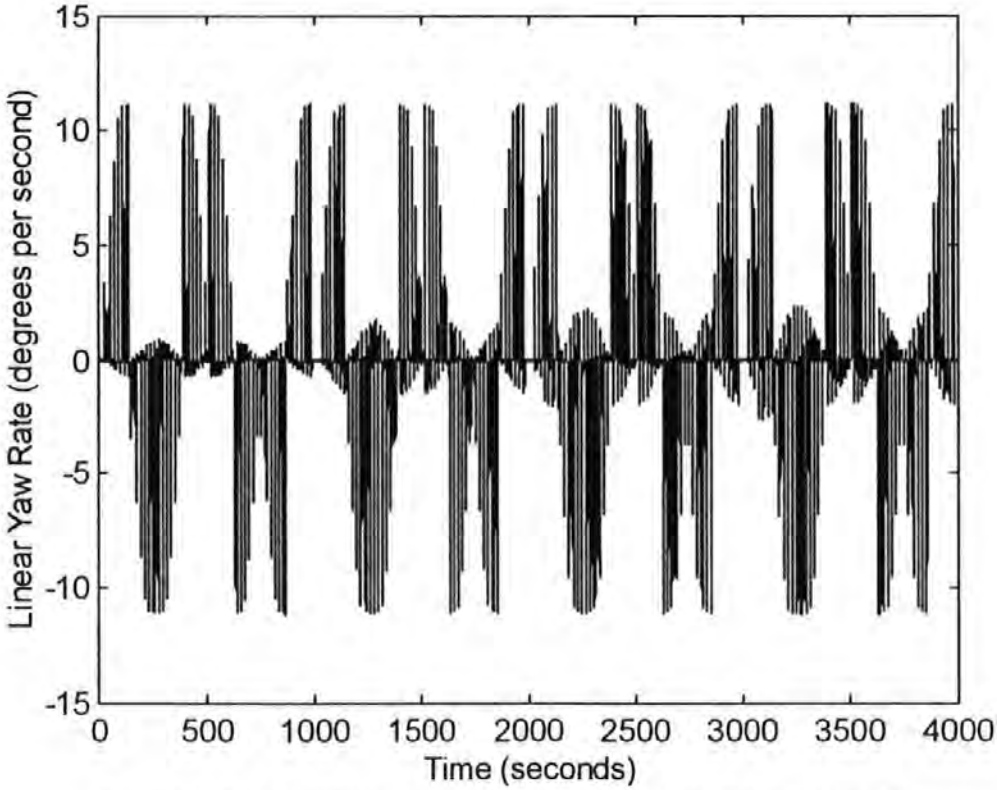


Figure 5.13 Complete Linear Model Yaw Rate Input Data Set

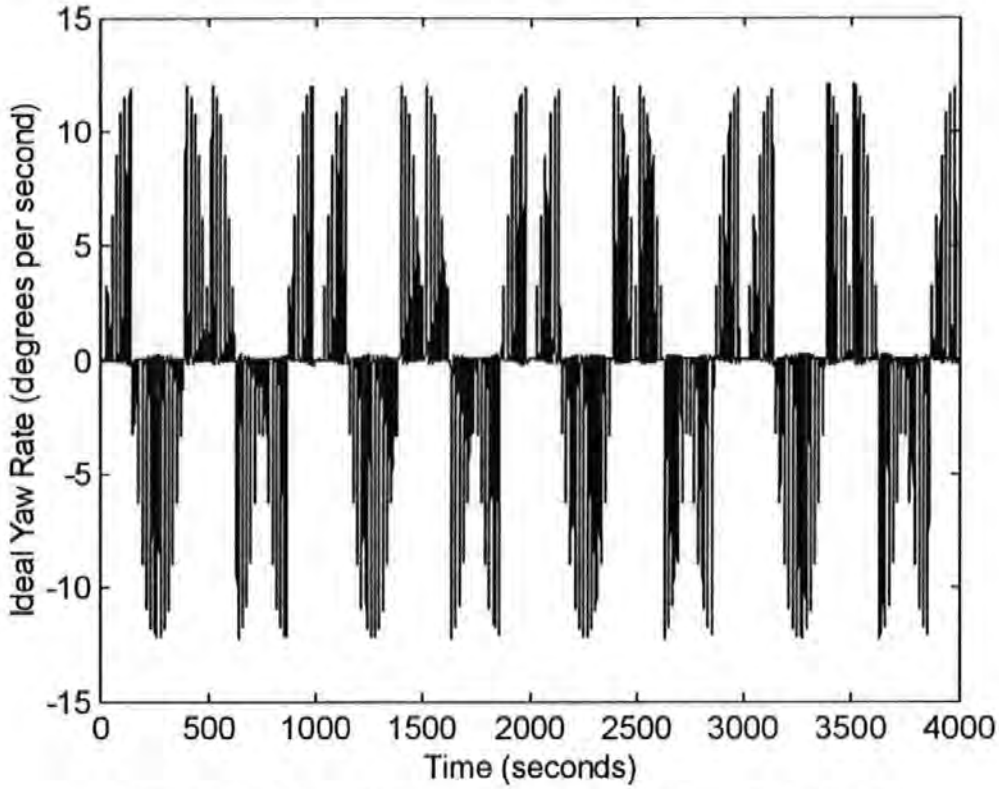


Figure 5.14 Complete Yaw Rate Sensor Output Data Set

For the roll channel FISs a different approach was needed, a series of initial input roll angles was required. This involved running six shorter simulations of twenty seconds each and then placing the results one after another to form a data vector of the required length. The roll response of the AUV to these initial roll angles can be seen in Figure 5.15.

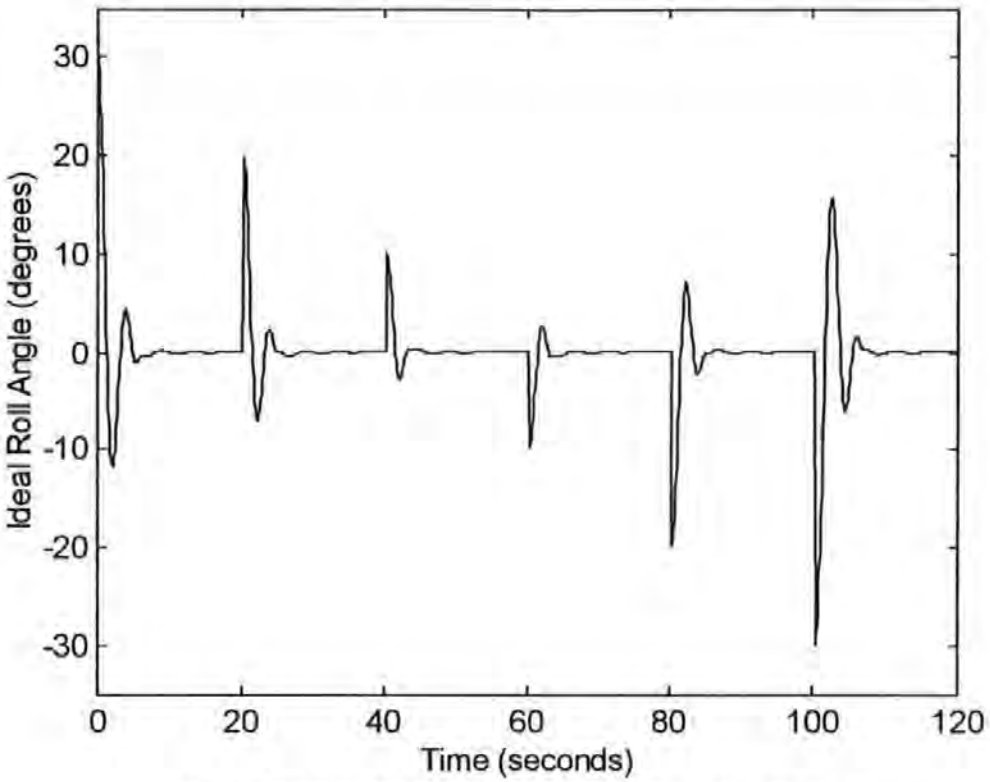


Figure 5.15 AUV Roll Response to Training Path

The input data sets and the output data set were recorded in the same way as for the yaw channel work which lead to the creation of the data sets shown in Figures 5.16, 5.17, 5.18 and 5.19. Once again the roll rate input and output data sets were created using the same process, and hence the data set can be seen in Figures 5.20, 5.21, 5.22 and 5.23.

Figures 5.16 and 5.20 again show the Q statistic initially has a value of zero and the value increases as the sensor signal and linear model signal drift apart. For every simulation run the Q statistic very quickly calculates that the sensor signal and linear model signal are different and considers the linear model signal the most reliable. For this reason both Figures 5.16 and 5.20 show the Q value to 1 for most of the simulation time.

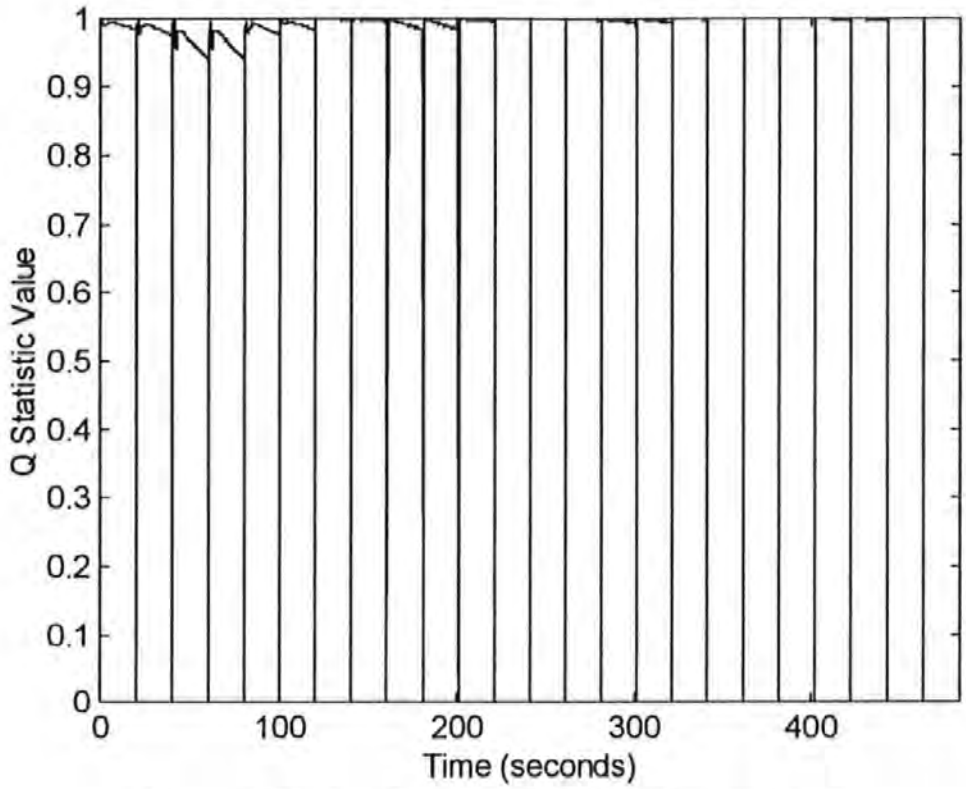


Figure 5.16 Complete Roll Q Statistic Input Data Set

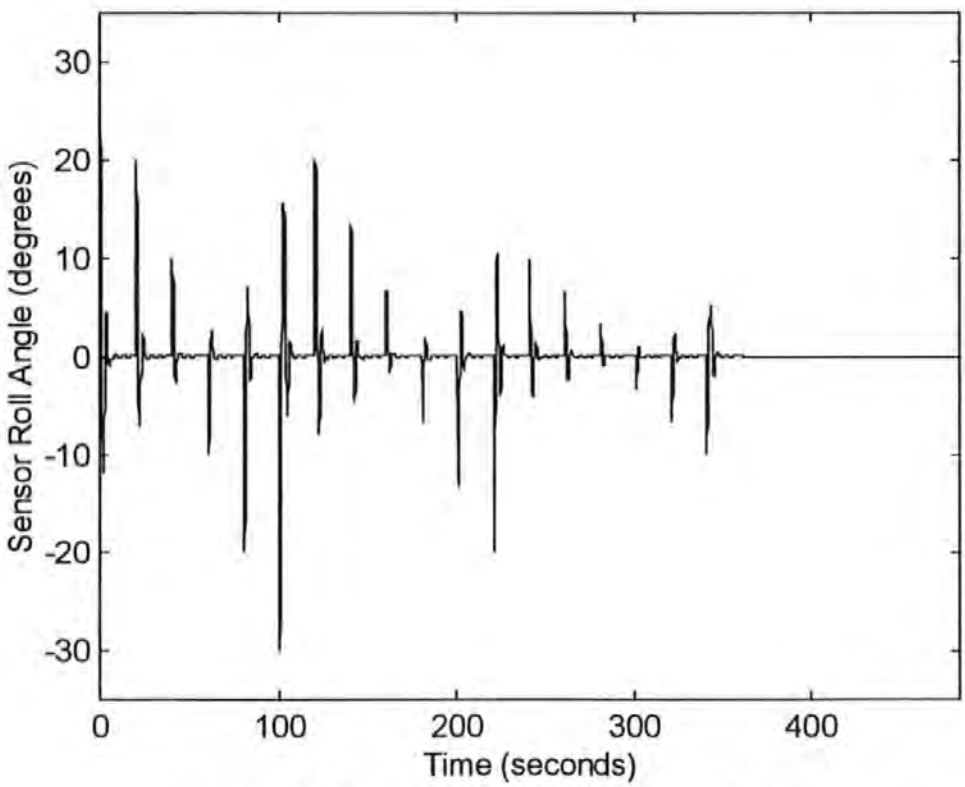


Figure 5.17 Complete Roll Sensor Input Data Set

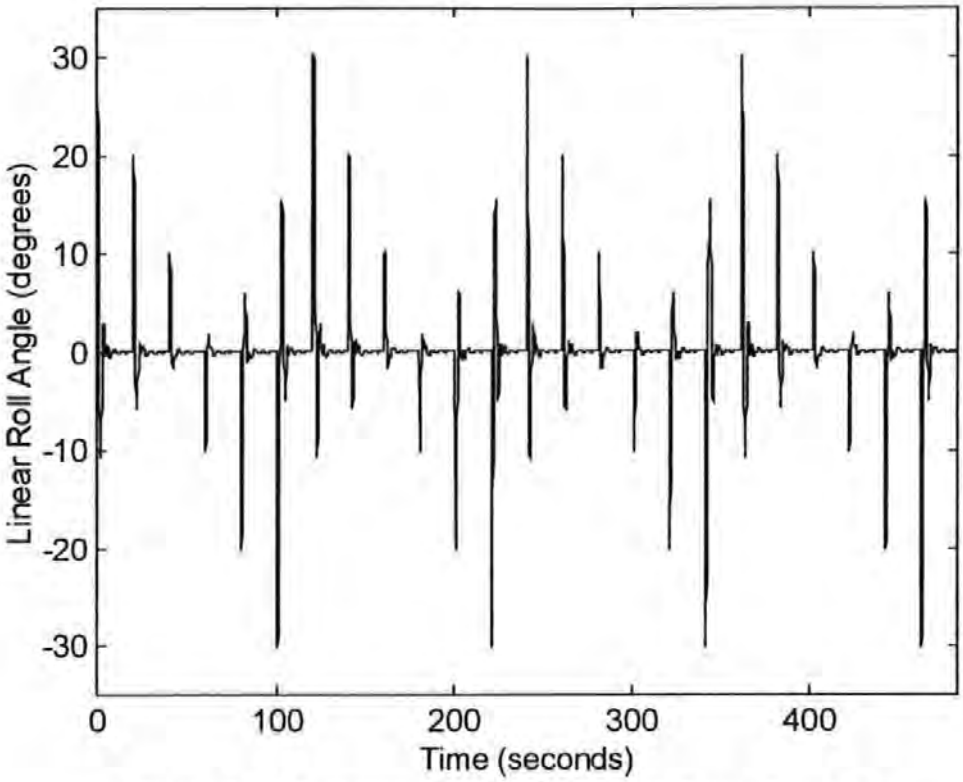


Figure 5.18 Complete Linear Model Roll Input Data Set

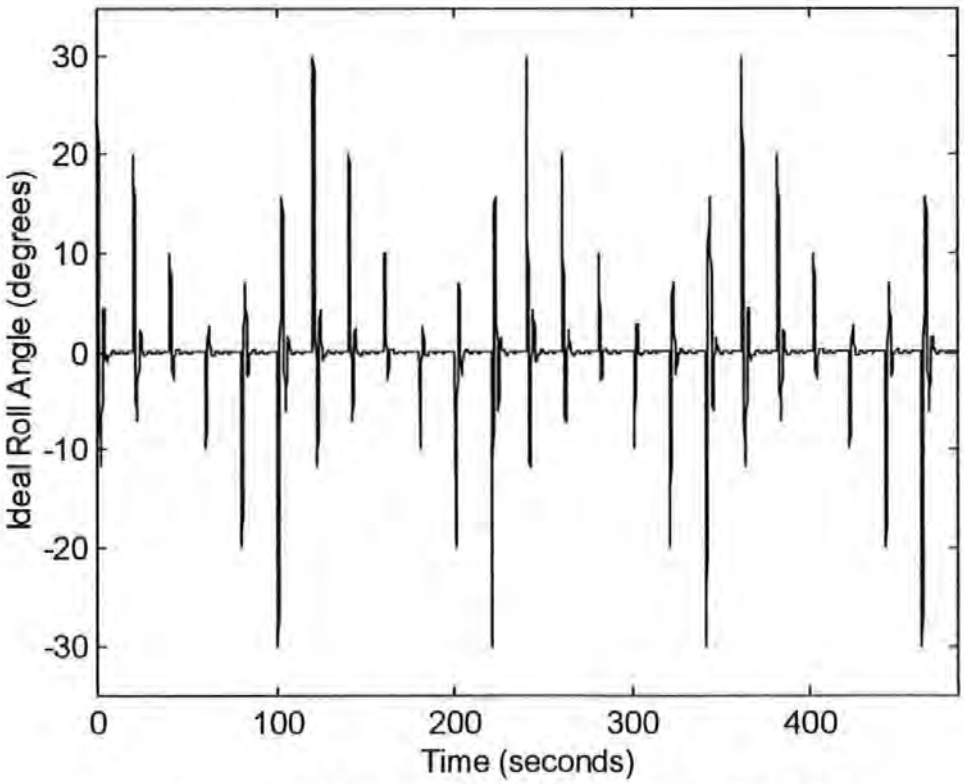


Figure 5.19 Complete Roll Sensor Output Data Set

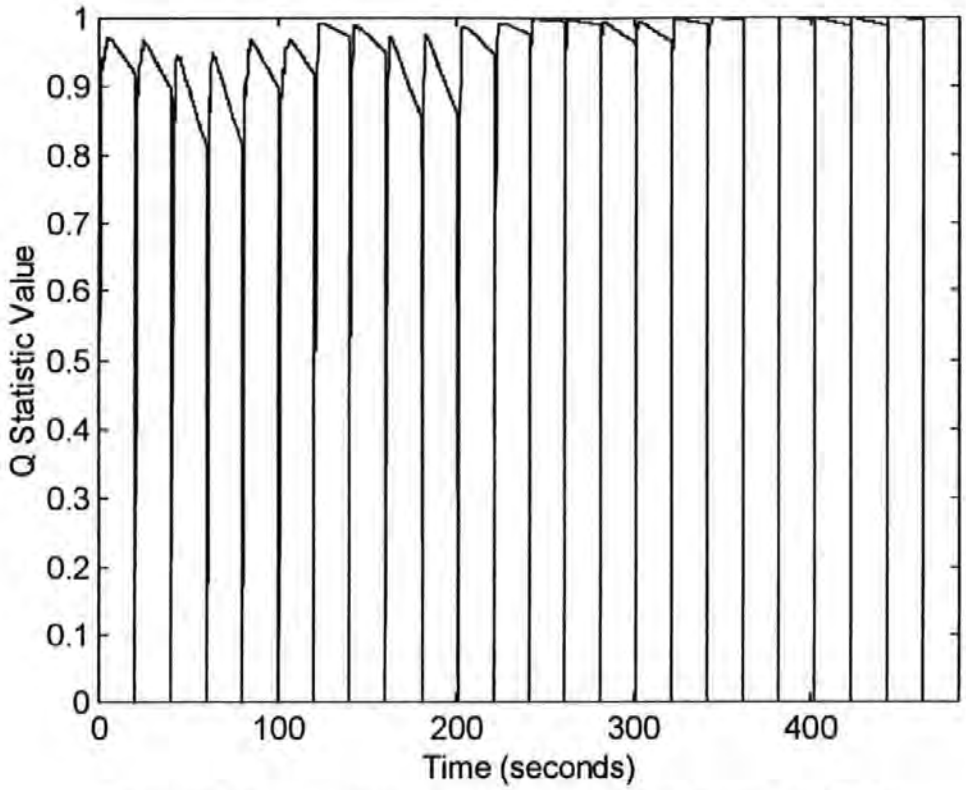


Figure 5.20 Complete Roll Rate Q Statistic Input Data Set

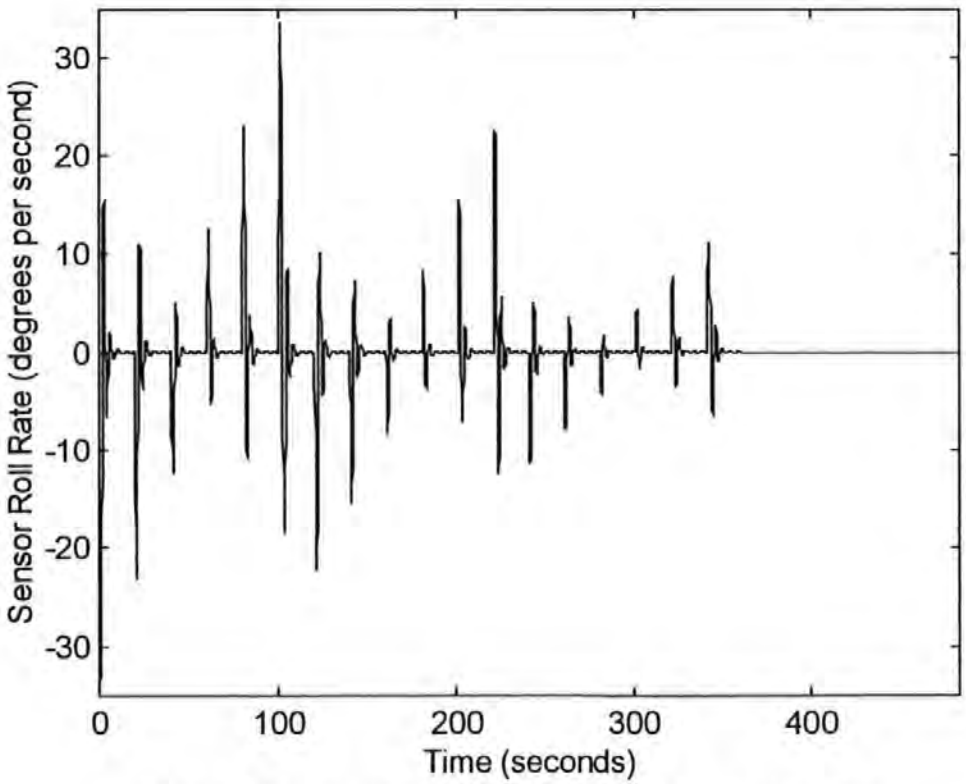


Figure 5.21 Complete Roll Rate Sensor Input Data Set

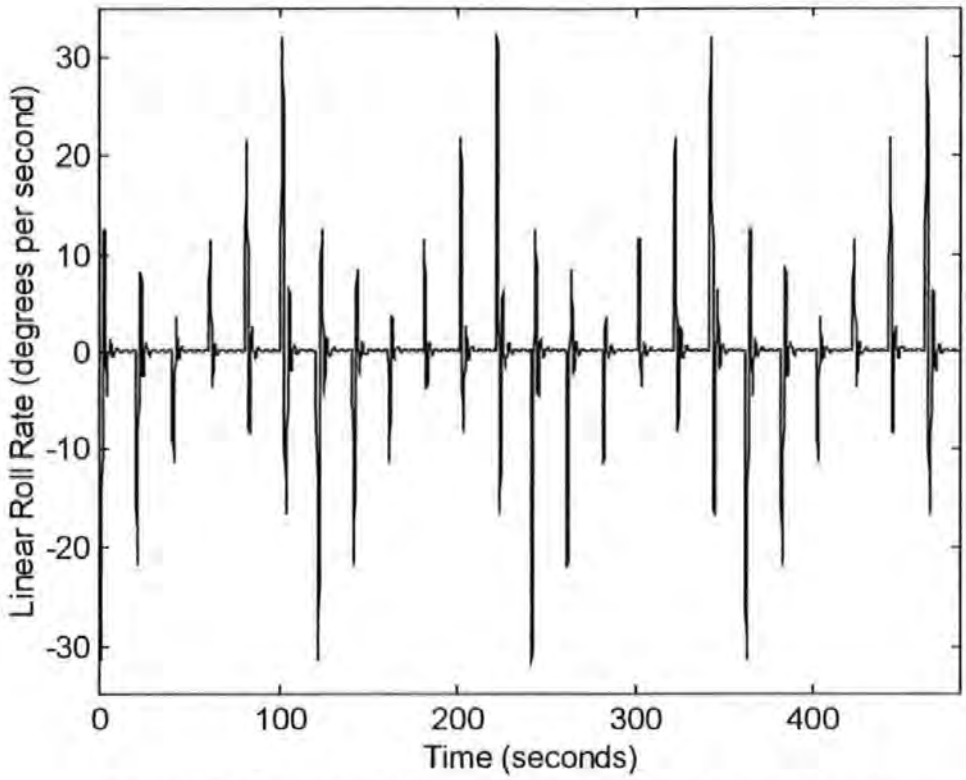


Figure 5.22 Complete Linear Model Roll Rate Input Data Set

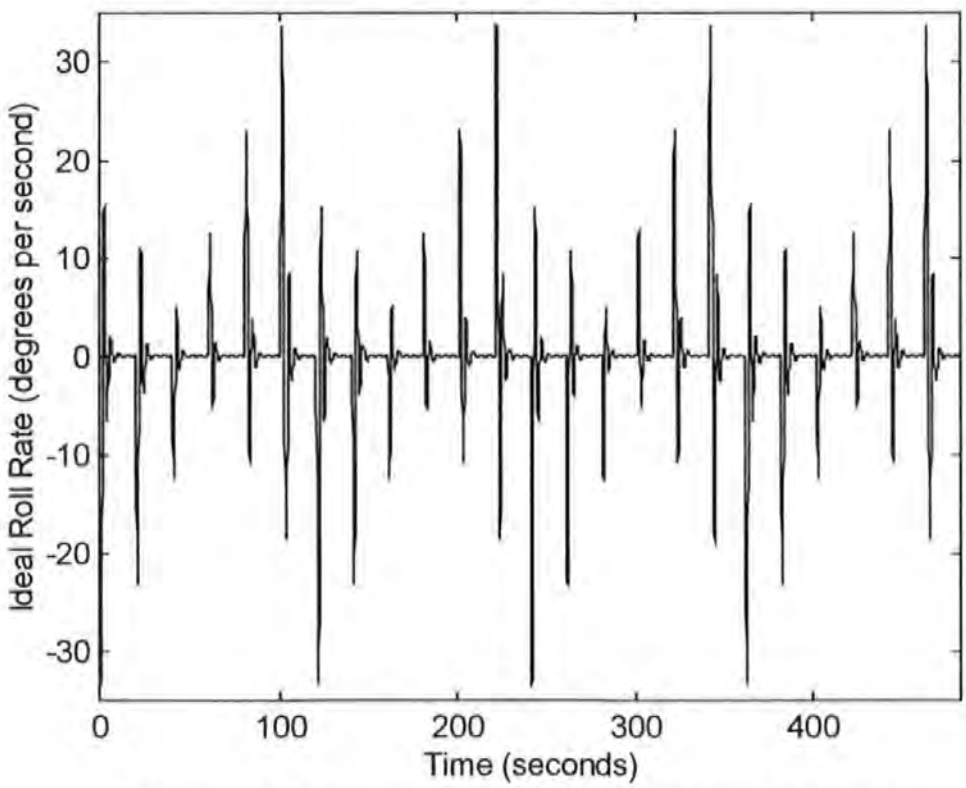


Figure 5.23 Complete Roll Rate Sensor Output Data Set

After each tuning for 100 epochs the FISs were tested on the step input demands and types of faults introduced in Chapter 3. These are the same tests the Kalman filter enhanced ANFIS control systems were subjected too. This will make comparison of the three sets of results both easy and fair. The FISs were all tuned for a total of 500 epochs by the simulated annealing method. This was repeated three times due to the statistical nature of the method to remove the chance of an anomalous result. This produced fifteen FISs for each case being considered.

The ANFIS tuning method did not require the heuristic FIS as a starting point, but did use the same data sets to tune the FISs. Again the FISs were tuned for a total of 500 epochs with the FISs being recorded after each 100 epochs. The was no need to repeat the process three times as the ANFIS method does not depend on the same statistical approach as the simulated annealing method.

5.5. RESULTS

The results of the best tuned FISs for each tuning method (ANFIS and simulated annealing) for each of the four considered sensors are now presented. All of the levels for all the faults for all levels of demanded inputs being considered are shown.

5.5.1. Yaw Sensor FISs

First let the yaw sensor case be considered. A total of twenty FISs were created for this sensor fault, five using the ANFIS method and fifteen using the simulated annealing method. Taking the number of FISs and the number of tests each FIS was subjected to into account leads to 480 tests, which is clearly too many results to include in this thesis, therefore only the best FIS tuned by each of the two considered methods will be presented.

Before the results are presented in section 5.5.5 both FISs will be displayed. The best ANFIS tuned FIS was the one tuned for 200 epochs. The three input membership functions for the best ANFIS tuned yaw sensor recovery FIS are shown in Figures 5.24, 5.25 and 5.26.

The skews shown in Figure 5.24 are due to the input membership functions being based on the data show in Figure 5.7 where most of the data was of high value. The

skews shown in Figures 5.25 and 5.26 are not as prominent as the data used for these inputs was evenly distributed.

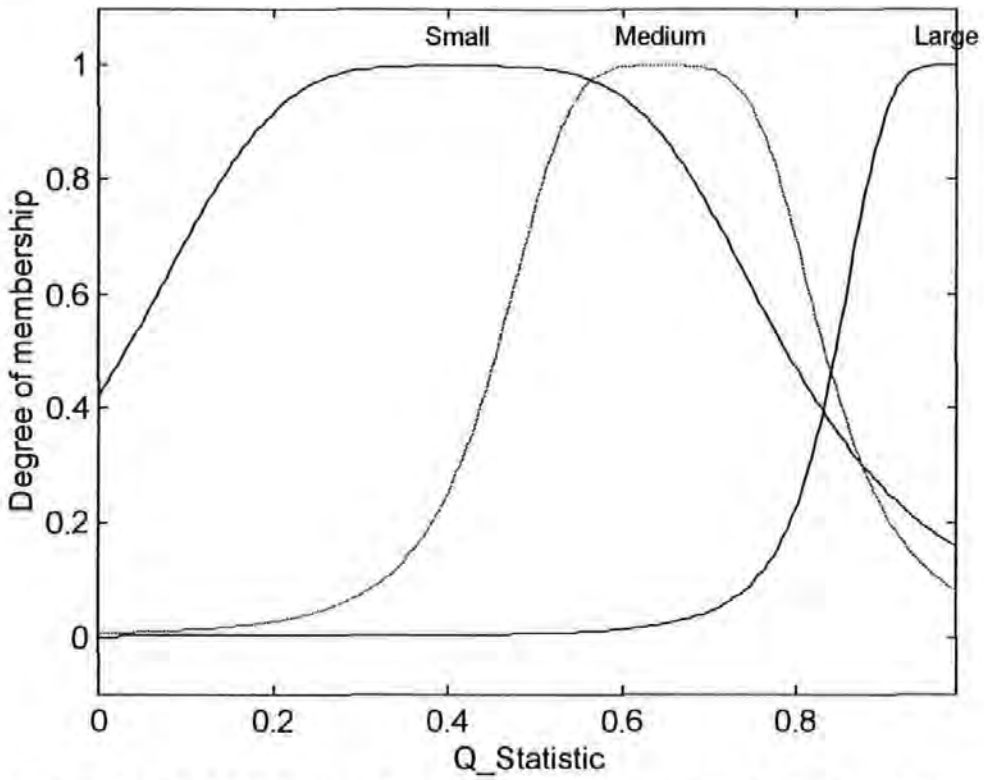


Figure 5.24 ANFIS Tuned Membership Functions for Q Statistic Input

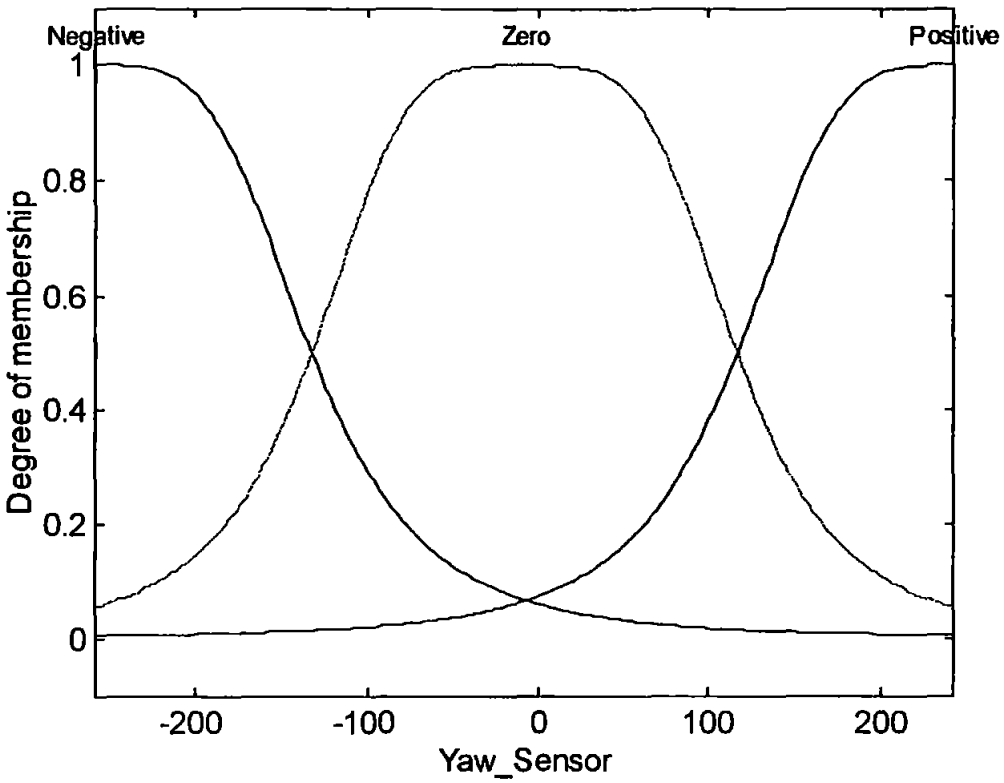


Figure 5.25 ANFIS Tuned Membership Functions for Yaw Sensor Input

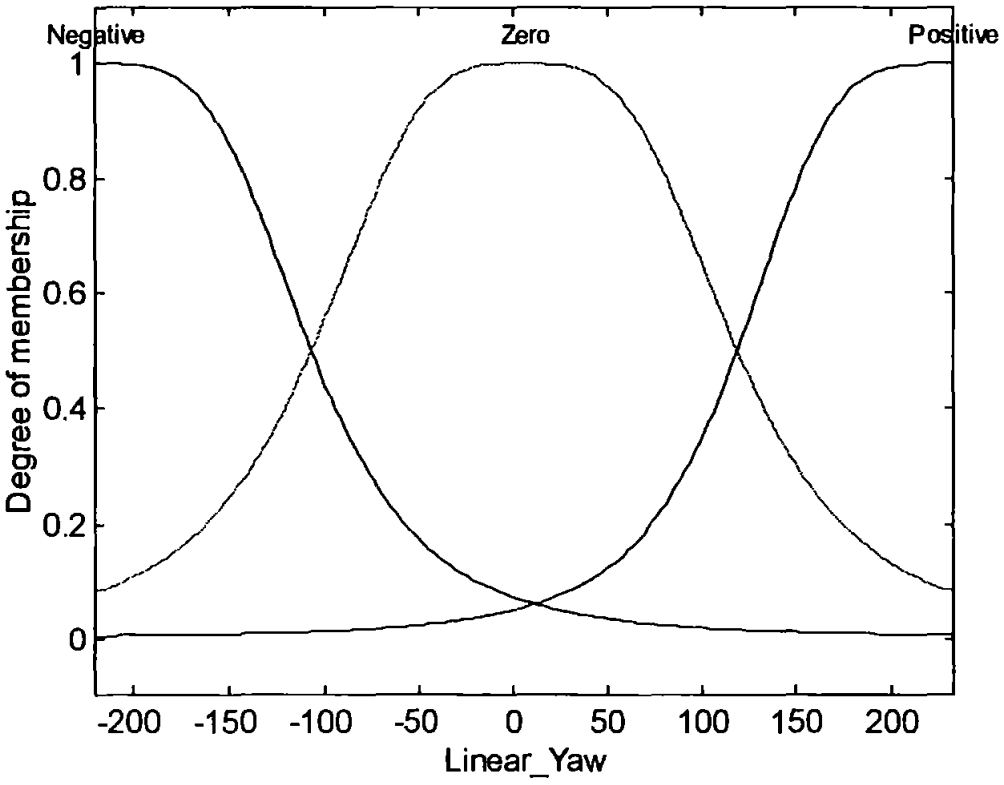


Figure 5.26 ANFIS Tuned Membership Functions for Linear Yaw Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.5). Due to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

SE = 3650 Q –5.267 SEN +4.654 LS –4512

If Q is S and SEN is N and LS is Z then

SE = -3051 Q –16.19 SEN +25.52 LS +1551

f Q is S and SEN is N and LS is P then

SE = 2140 Q +298.8 SEN –407.1 LS +2901

If Q is S and SEN is Z and LS is N then

SE = 839.3 Q –5.733 SEN +1.203 LS –731.5

If Q is S and SEN is Z and LS is Z then

SE = 62.71 Q +1.573 SEN –0.827 LS –29.17

If Q is S and SEN is Z and LS is P then

SE = 83.42 Q –18.57 SEN +21.46 LS –563.7

If Q is S and SEN is P and LS is N then

SE = 4394 Q +589.2 SEN –564.8 LS -3150

If Q is S and SEN is P and LS is Z then

SE = -296.6 Q –21.66 SEN +21.64 LS +645.1

If Q is S and SEN is P and LS is P then

SE = -170.2 Q –1.242 SEN +2.671 LS +70.10

If Q is M and SEN is N and LS is N then

SE = -7470 Q +5.058 SEN –2.078 LS +6839

If Q is M and SEN is N and LS is Z then

SE = 1069 Q +19.55 SEN –21.76 LS +195.4

If Q is M and SEN is N and LS is P then

SE = -6723 Q –302.5 SEN +346.6 LS -1319

If Q is M and SEN is Z and LS is N then

SE = -944.6 Q +7.015 SEN –1.264 LS +863.0

If Q is M and SEN is Z and LS is Z then

SE = 44.16 Q –0.427 SEN +1.956 LS –47.82 (5.5)

If Q is M and SEN is Z and LS is P then

SE = 358.1 Q +15.16 SEN –13.79 LS +25.63

If Q is M and SEN is P and LS is N then

SE = 3498 Q –552.6 SEN +601.7 LS –3108

If Q is M and SEN is P and LS is Z then

SE = -491.6 Q +20.67 SEN –18.07 LS –234.4

If Q is M and SEN is P and LS is P then

SE = 180.8 Q +2.203 SEN –1.209 LS –181.9

If Q is L and SEN is N and LS is N then

SE = -5156 Q –8.535 SEN +3.543 LS +4154

If Q is L and SEN is N and LS is Z then

SE = -944.4 Q –8.383 SEN +6.132 LS +720.4

If Q is L and SEN is N and LS is P then

SE = -9850 Q +17.36 SEN –2.832 LS +1864

If Q is L and SEN is Z and LS is N then

SE = 237.2 Q –0.943 SEN +0.4138 LS –844.8

If Q is L and SEN is Z and LS is Z then

SE = 351.7 Q –0.962 SEN +0.501 LS –531.7

If Q is L and SEN is Z and LS is P then

SE = 963.1 Q –8.587 SEN +1.632 LS –602.0

If Q is L and SEN is P and LS is N then

SE = 12349 Q –217.0 SEN +49.06 LS –3012

If Q is L and SEN is P and LS is Z then

SE = -5137 Q –26.57 SEN +2.862 LS +7829

If Q is L and SEN is P and LS is P then

SE = 213.3 Q –3.299 SEN –3.154 LS +1910

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The best simulated annealing tuned FIS was the one tuned for 400 epochs. The three input membership functions for the best simulated annealing tuned yaw sensor recovery FIS are shown in Figures 5.27, 5.28 and 5.29.

The skews shown in Figures 5.27, 5.28 and 5.29 are due to tuning performed by the simulated annealing method.

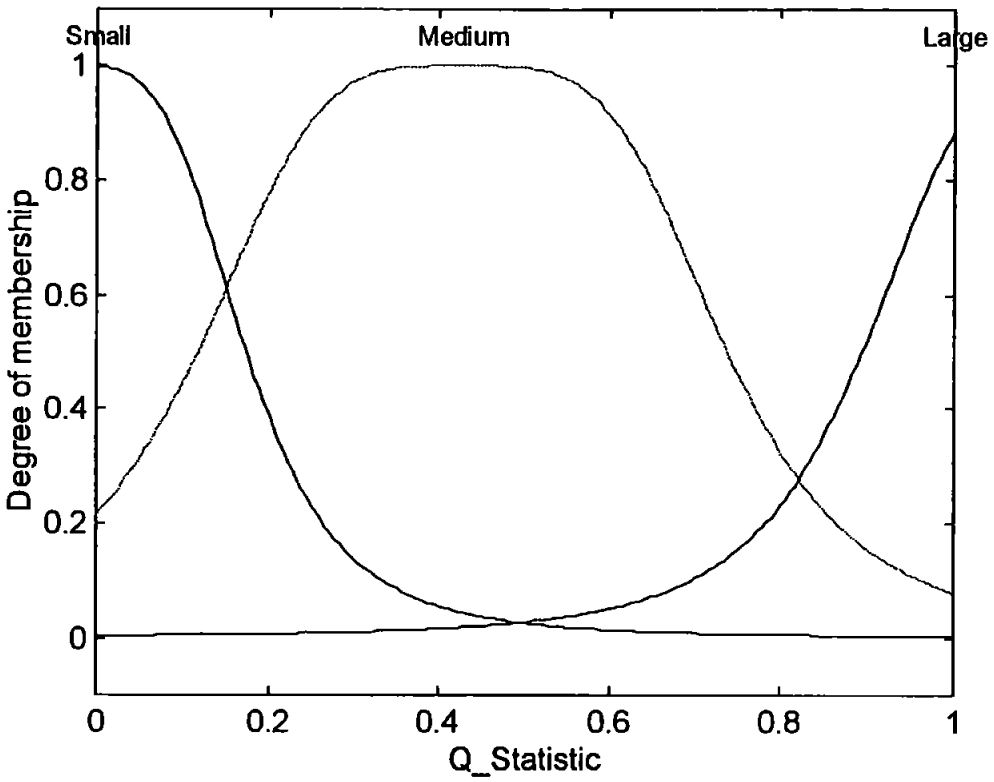


Figure 5.27 Simulated Annealing Tuned Membership Functions for Q Statistic Input

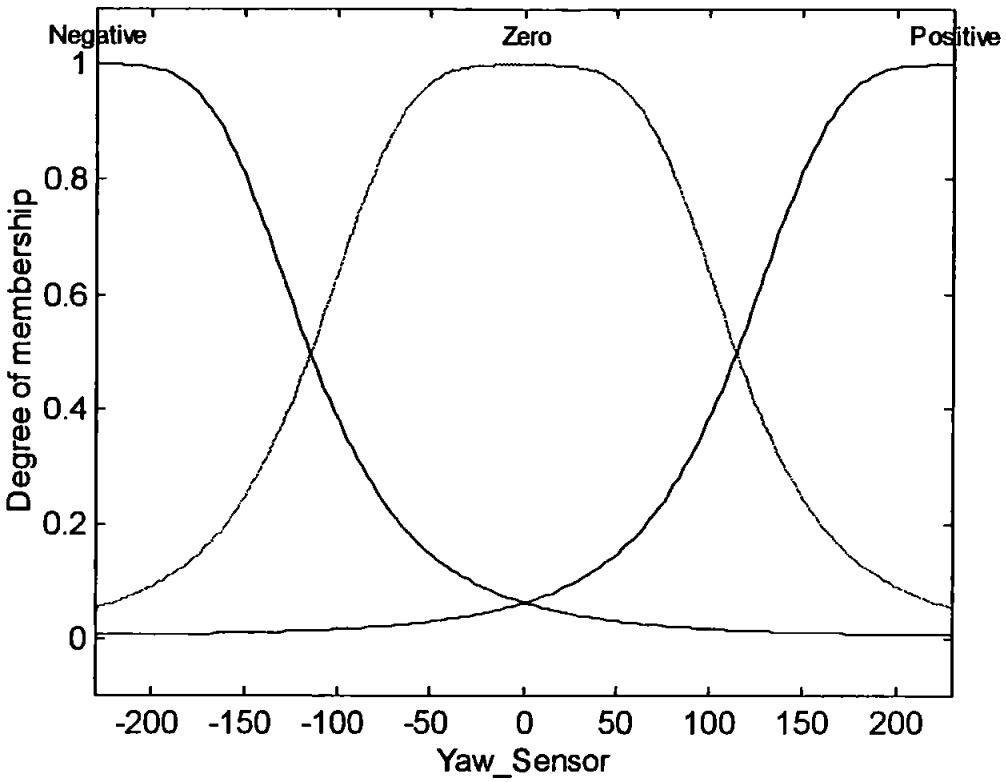


Figure 5.28 Simulated Annealing Tuned Membership Functions for Yaw Sensor Input

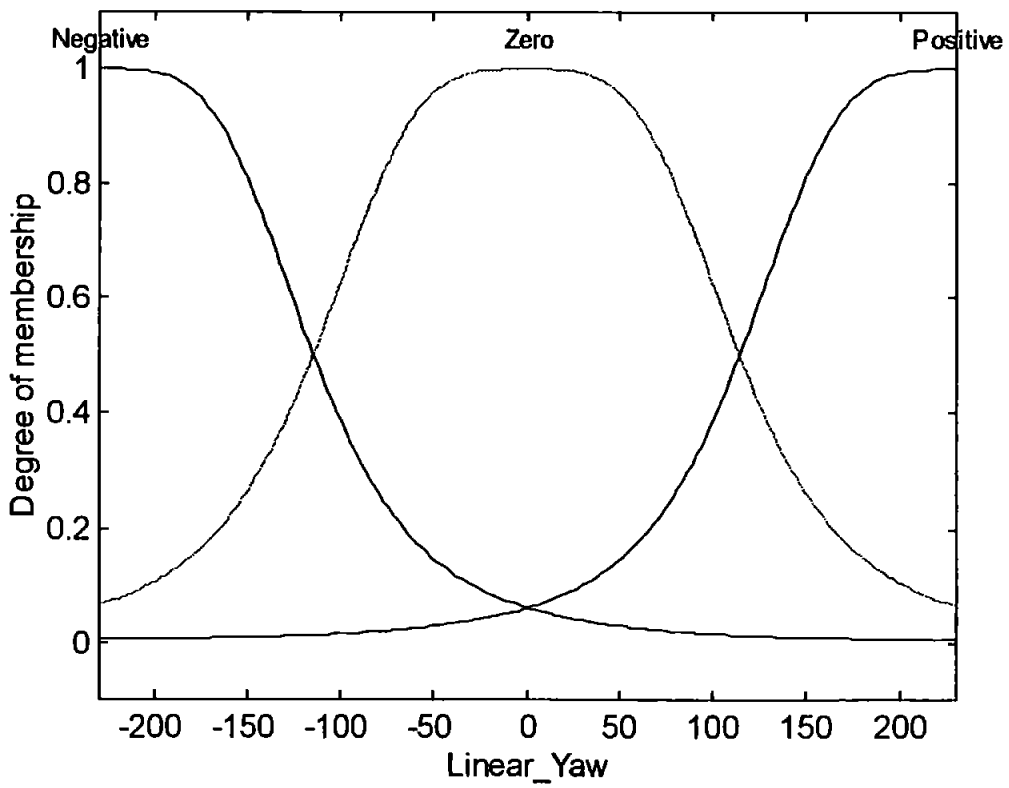


Figure 5.29 Simulated Annealing Tuned Membership Functions for Linear Yaw Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.6). Owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

SE = -0.063 Q +0.932 SEN +0.060 LS –0.058

If Q is S and SEN is N and LS is Z then

SE = 0.060 Q +0.626 SEN +0.680 LS –0.032

If Q is S and SEN is N and LS is P then

SE = -0.003 Q +0.034 SEN +0.976 LS +0.084

If Q is S and SEN is Z and LS is N then

SE = -0.131 Q +0.980 SEN +0.027 LS +-0.047

If Q is S and SEN is Z and LS is Z then

SE = 0.048 Q +0.728 SEN +0.704 LS +0.025

If Q is S and SEN is Z and LS is P then

SE = -0.005 Q –0.045 SEN +1.035 LS +0.023

If Q is S and SEN is P and LS is N then

SE = -0.039 Q +0.967 SEN +0.025 LS +0.005

If Q is S and SEN is P and LS is Z then

SE = 0.025 Q +0.590 SEN +0.640 LS –0.058

If Q is S and SEN is P and LS is P then

SE = -0.098 Q –0.043 SEN +1.020 LS –0.058

If Q is M and SEN is N and LS is N then

SE = -0.005 Q +1.032 SEN +0.003 LS +0.035

If Q is M and SEN is N and LS is Z then

SE = 0.009 Q +0.716 SEN +0.758 LS –0.019

If Q is M and SEN is N and LS is P then

SE = -0.032 Q +0.019 SEN +1.082 LS +0.098

If Q is M and SEN is Z and LS is N then

SE = -0.033 Q +0.935 SEN –0.014 LS +0.029

If Q is M and SEN is Z and LS is Z then

SE = 0.078 Q +0.665 SEN +0.694 LS –0.0132 (5.6)

If Q is M and SEN is Z and LS is P then

SE = 0.034 Q +0.000 SEN +0.982 LS +0.011

If Q is M and SEN is P and LS is N then

SE = 0.014 Q +0.897 SEN –0.008 LS +0.049

If Q is M and SEN is P and LS is Z then

SE = 0.066 Q +0.659 SEN +0.574 LS –0.030

If Q is M and SEN is P and LS is P then

SE = 0.037 Q +0.023 SEN +0.972 LS +0.007

If Q is L and SEN is N and LS is N then

SE = -0.076 Q +1.070 SEN +0.041 LS –0.070

If Q is L and SEN is N and LS is Z then

SE = -0.088 Q +0.656 SEN +0.619 LS +0.004

If Q is L and SEN is N and LS is P then

SE = -0.043 Q –0.024 SEN +0.933 LS +0.074

If Q is L and SEN is Z and LS is N then

SE = -0.010 Q +0.914 SEN +0.026 LS –0.055

If Q is L and SEN is Z and LS is Z then

SE = -0.042 Q +0.696 SEN +0.658 LS –0.031

If Q is L and SEN is Z and LS is P then

SE = -0.085 Q +0.010 SEN +1.059 LS –0.143

If Q is L and SEN is P and LS is N then

SE = -0.035 Q +0.915 SEN +0.036 LS –0.087

If Q is L and SEN is P and LS is Z then

SE = -0.095 Q +0.742 SEN +0.631 LS +0.043

If Q is L and SEN is P and LS is P then

SE = 0.041 Q +0.020 SEN +1.049 LS +0.042

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

5.5.2. Yaw Rate Sensor FISs

As there are the same number of yaw rate sensor recovery FISs which are tested on the same number of faults as the yaw sensor, then once again there are too many results too display fully. Therefore the results for the two best tuned FISs for the yaw rate sensor will be displayed. The two FISs will now be displayed.

The best ANFIS tuned FIS was the one tuned for 100 epochs. The three input membership functions for the best ANFIS tuned yaw rate sensor recovery FIS are shown in Figures 5.30, 5.31 and 5.32.

The skews shown in Figure 5.30 are due to the input membership functions being based on the data show in Figure 5.11 where most of the data was of high value. The skews shown in Figures 5.31 and 5.32 are not as prominent as the data used for these inputs was evenly distributed.

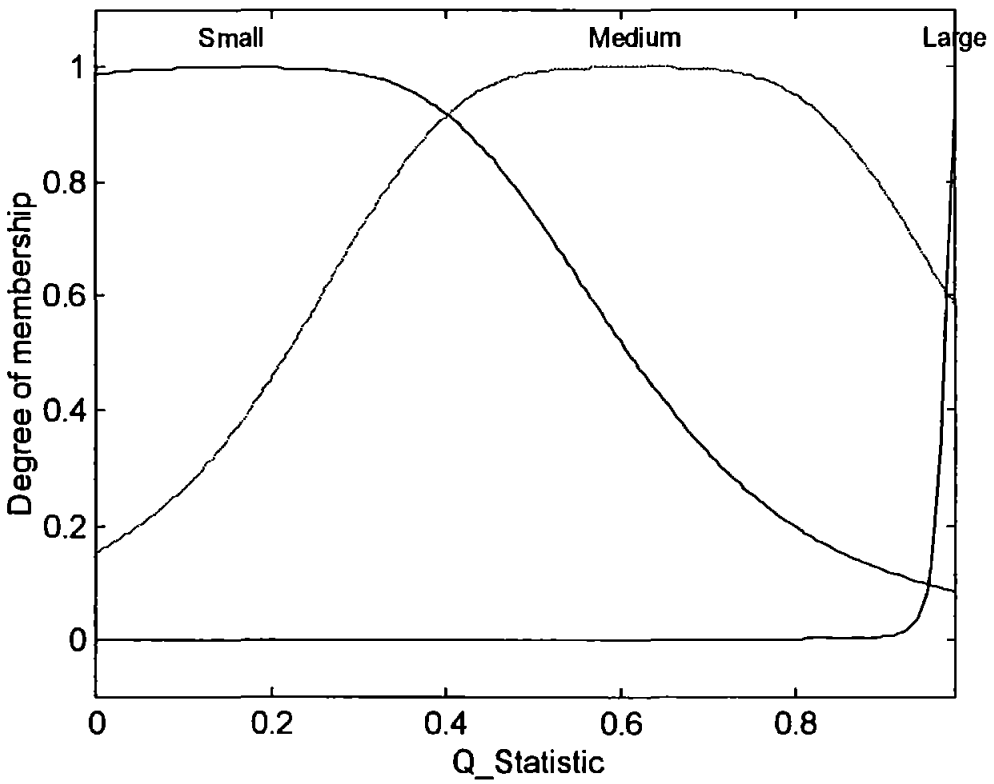


Figure 5.30 ANFIS Tuned Membership Functions for Q Statistic Input

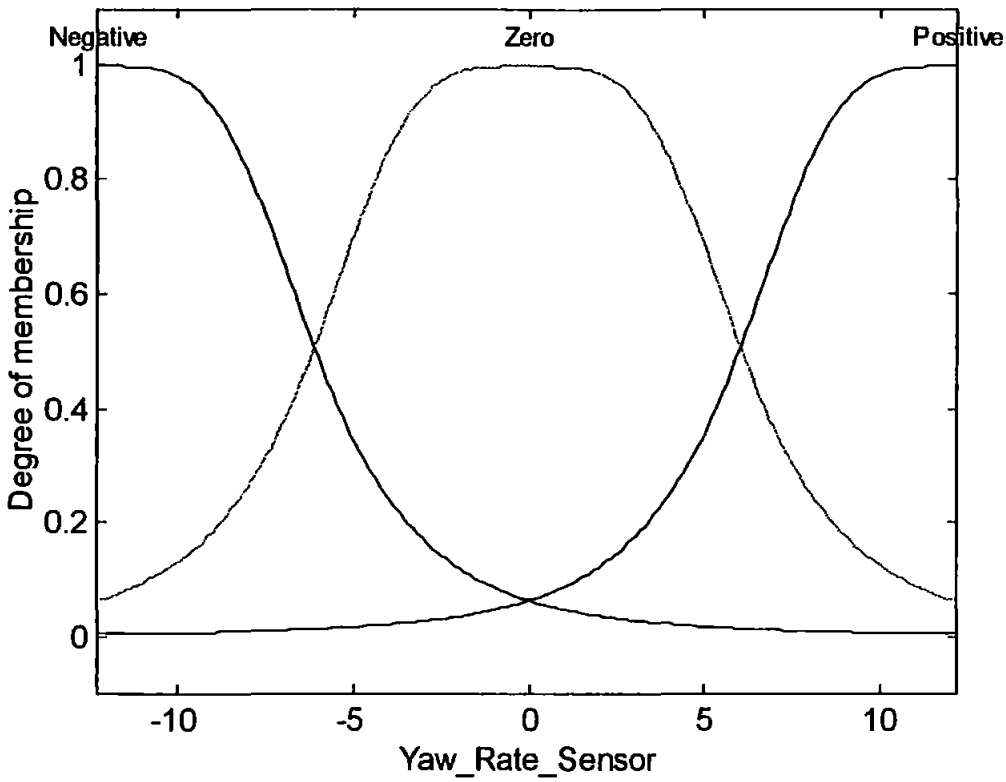


Figure 5.31 ANFIS Tuned Membership Functions for Yaw Rate Sensor Input

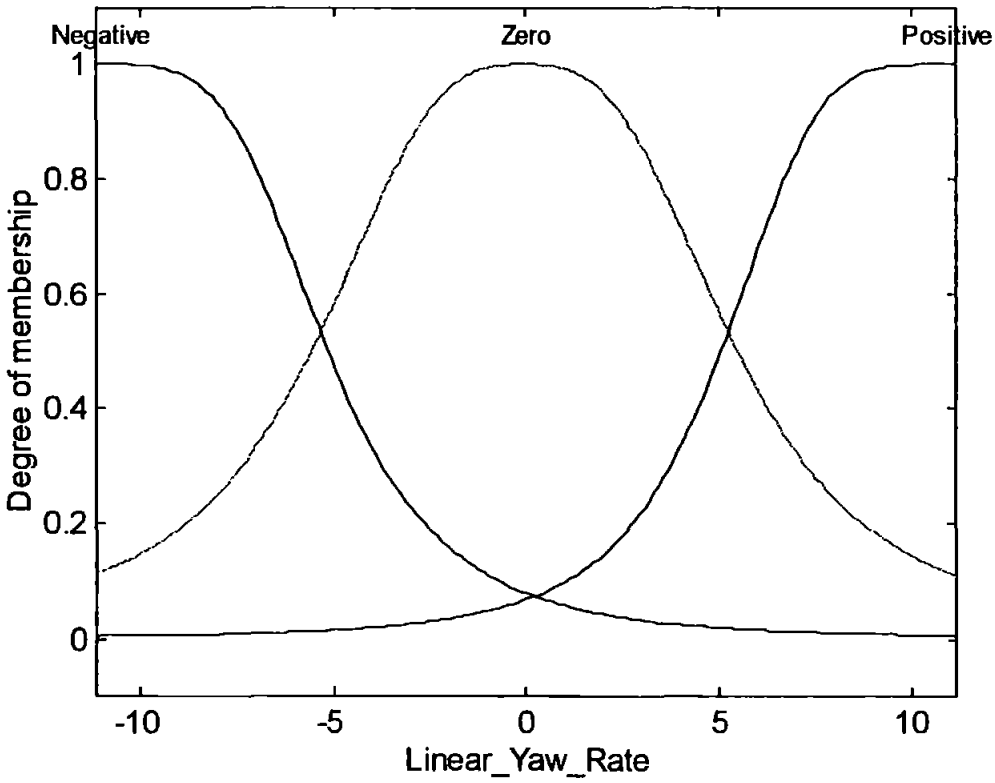


Figure 5.32 ANFIS Tuned Membership Functions for Linear Yaw Rate Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.7). Owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then
SE = -441.7 Q –12.23 SEN +11.29 LS +42.83
If Q is S and SEN is N and LS is Z then
SE = -1414 Q –15.21 SEN +27.07 LS –65.84
If Q is S and SEN is N and LS is P then
SE = -131.6 Q +246.5 SEN –241.8 LS +1890
If Q is S and SEN is Z and LS is N then
SE = 167.6 Q –5.249 SEN +8.490 LS –86.60
If Q is S and SEN is Z and LS is Z then
SE = 63.05 Q +8.262 SEN –5.257 LS +9.533
If Q is S and SEN is Z and LS is P then
SE = -6.151 Q –3.801 SEN +5.906 LS –44.85
If Q is S and SEN is P and LS is N then
SE = 1625 Q –133.5 SEN +-683.9 LS –698.5
If Q is S and SEN is P and LS is Z then
SE = 143.3 Q +11.14 SEN +19.01 LS –0.847
If Q is S and SEN is P and LS is P then
SE = 10.46 Q –1.011 SEN +3.287 LS +1.334
If Q is M and SEN is N and LS is N then
SE = -184.2 Q +4.322 SEN –2.045 LS +230.4
If Q is M and SEN is N and LS is Z then
SE = -1233 Q +8.852 SEN –7.118 LS +1213
If Q is M and SEN is N and LS is P then
SE = -261.0 Q-33.99 SEN –3.139 LS +366.5
If Q is M and SEN is Z and LS is N then
SE = 73.61 Q –3.931 SEN +2.859 LS –38.05
If Q is M and SEN is Z and LS is Z then
SE = 61.05 Q +1.358 SEN –0.707 LS –57.58 (5.7)

If Q is M and SEN is Z and LS is P then

SE = -4.194 Q –0.176 SEN +1.921 LS –1.002

If Q is M and SEN is P and LS is N then

SE = 1646 Q +332.5 SEN +29.93 LS -2407

If Q is M and SEN is P and LS is Z then

SE = 105.4 Q –6.980 SEN +7.555 LS –127.9

If Q is M and SEN is P and LS is P then

SE = 4.825 Q –0.386 SEN +1.951 LS –25.98

If Q is L and SEN is N and LS is N then

SE = -66.20 Q +220.3 SEN –97.83 LS –373.8

If Q is L and SEN is N and LS is Z then

SE = 372.2 Q -1187 SEN +705.3 LS +95.00

If Q is L and SEN is N and LS is P then

SE = 53.77 Q +705.8 SEN –24.95 LS –32.70

If Q is L and SEN is Z and LS is N then

SE = 493.8 Q –69.66 SEN +17.50 LS –449.0

If Q is L and SEN is Z and LS is Z then

SE = 90.43 Q +124.0 SEN –72.19 LS –92.65

If Q is L and SEN is Z and LS is P then

SE = -71.02 Q –49.91 SEN +7.216 LS +76.72

If Q is L and SEN is P and LS is N then

SE = -108.3 Q +487.1 SEN –151.6 LS –115.6

If Q is L and SEN is P and LS is Z then

SE = -379.0 Q –581.6 SEN +469.5 LS –15.77

If Q is L and SEN is P and LS is P then

SE = -294.1 Q +242.9 SEN –67.58 LS +112.2

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The best simulated annealing tuned FIS was the one tuned for 500 epochs. The three input membership functions for the best simulated annealing tuned yaw rate sensor recovery FIS are shown in Figures 5.33, 5.34 and 5.35.

The skews shown in Figures 5.33, 5.34 and 5.35 are due to tuning performed by the simulated annealing method.

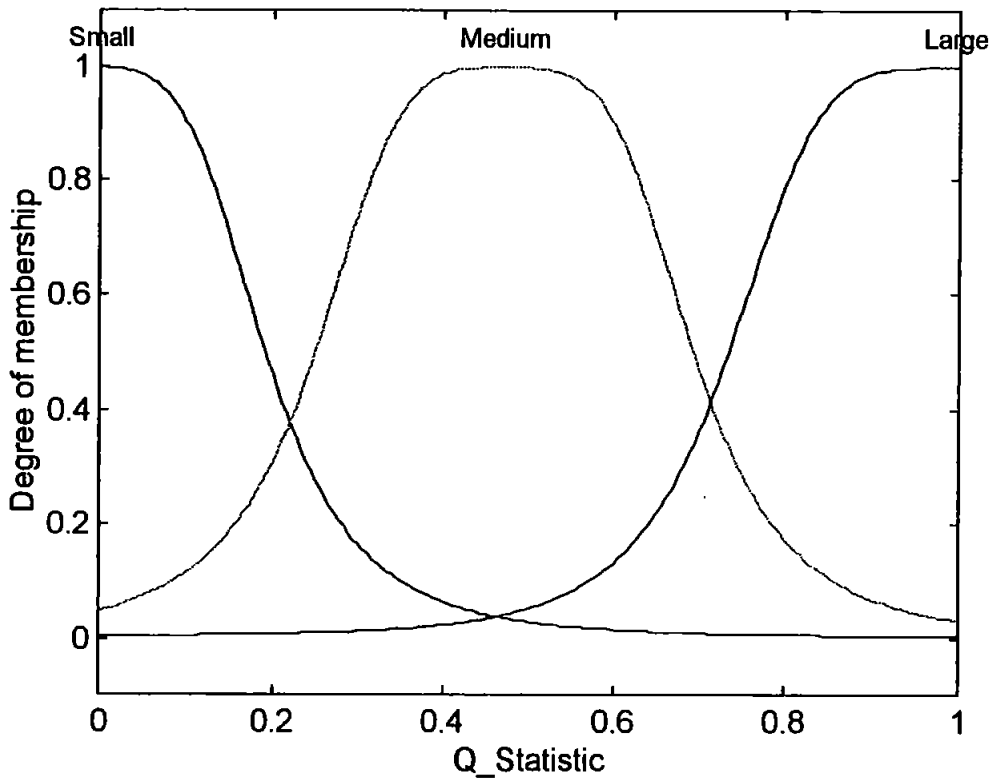


Figure 5.33 Simulated Annealing Tuned Membership Functions for Q Statistic Input

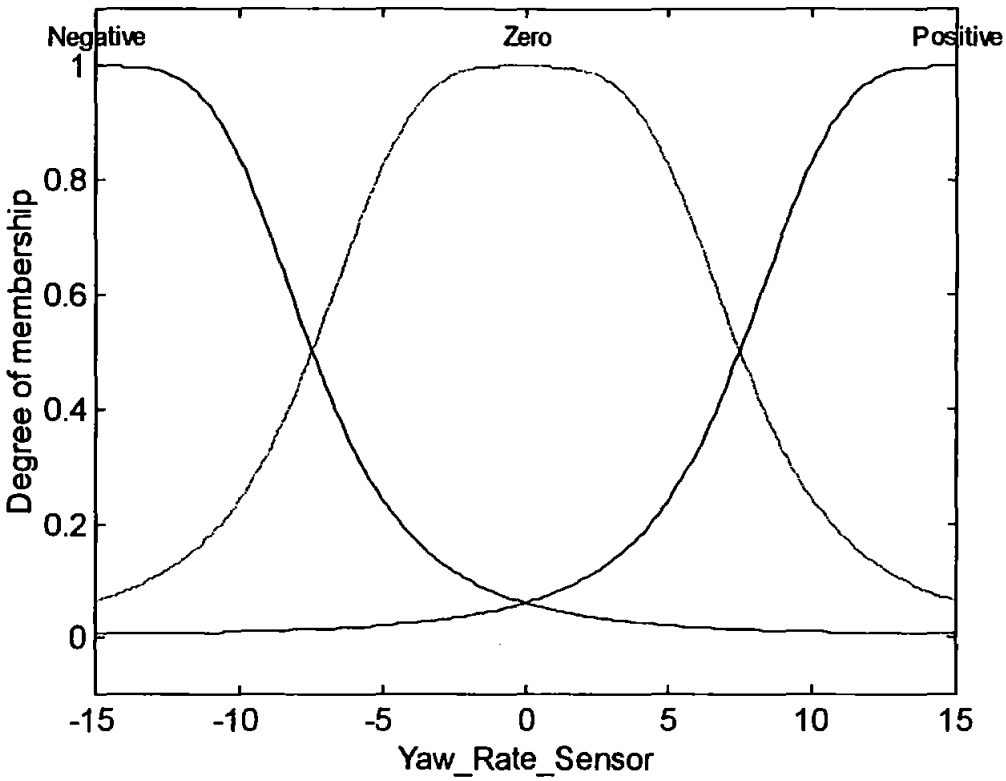


Figure 5.34 Simulated Annealing Tuned Membership Functions for Yaw Rate Sensor Input

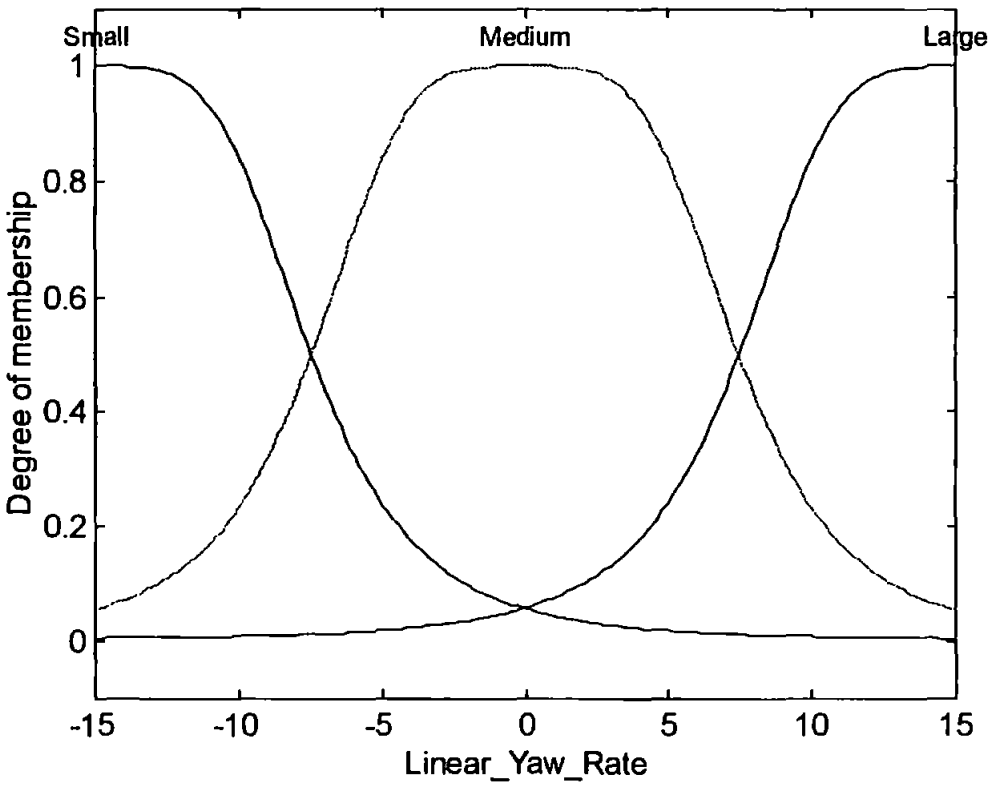


Figure 5.35 Simulated Annealing Tuned Membership Functions for Linear Yaw Rate Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.8). Owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

SE = 0.011 Q +0.487 SEN +0.492 LS +0.107

If Q is S and SEN is N and LS is Z then

SE = -0.022 Q +0.496 SEN +0.532 LS +0.105

If Q is S and SEN is N and LS is P then

SE = 0.003 Q +0.499 SEN +0.527 LS +0.105

If Q is S and SEN is Z and LS is N then

SE = -0.028 Q +0.496 SEN +0.521 LS +0.122

If Q is S and SEN is Z and LS is Z then

SE = -0.013 Q +0.530 SEN +0.537 LS +0.106

If Q is S and SEN is Z and LS is P then

SE = -0.017 Q +0.501 SEN +0.489 LS +0.113

If Q is S and SEN is P and LS is N then

SE = -0.044 Q +0.497 SEN +0.543 LS +0.142

If Q is S and SEN is P and LS is Z then

SE = 0.020 Q +0.519 SEN +0.447 LS +0.116

If Q is S and SEN is P and LS is P then

SE = 0.007 Q +0.502 SEN +0.496 LS +0.126

If Q is M and SEN is N and LS is N then

SE = -0.001 Q +0.530 SEN +0.526 LS +0.125

If Q is M and SEN is N and LS is Z then

SE = 0.007 Q +0.506 SEN +0.486 LS +0.123

If Q is M and SEN is N and LS is P then

SE = 0.019 Q +0.509 SEN +0.521 LS +0.132

If Q is M and SEN is Z and LS is N then

SE = -0.014 Q +0.470 SEN +0.487 LS +0.093

If Q is M and SEN is Z and LS is Z then

SE = 0.026 Q +0.508 SEN +0.523 LS +0.131 (5.8)

If Q is M and SEN is Z and LS is P then

SE = 0.001 Q +0.486 SEN +0.522 LS +0.142

If Q is M and SEN is P and LS is N then

SE = 0.007 Q +0.504 SEN +0.471 LS +0.108

If Q is M and SEN is P and LS is Z then

SE = -0.004 Q +0.489 SEN +0.461 LS +0.146

If Q is M and SEN is P and LS is P then

SE = 0.041 Q +0.452 SEN +0.475 LS +0.105

If Q is L and SEN is N and LS is N then

SE = -0.012 Q +0.526 SEN +0.445 LS +0.137

If Q is L and SEN is N and LS is Z then

SE = -0.0001 Q +0.498 SEN +0.475 LS +0.147

If Q is L and SEN is N and LS is P then

SE = 0.027 Q +0.501 SEN +0.557 LS +0.123

If Q is L and SEN is Z and LS is N then

SE = -0.039 Q +0.476 SEN +0.492 LS +0.119

If Q is L and SEN is Z and LS is Z then

SE = -0.003 Q +0.533 SEN +0.548 LS +0.108

If Q is L and SEN is Z and LS is P then

SE = -0.025 Q +0.532 SEN +0.518 LS +0.125

If Q is L and SEN is P and LS is N then

SE = -0.003 Q +0.477 SEN +0.464 LS +0.109

If Q is L and SEN is P and LS is Z then

SE = -0.006 Q +0.477 SEN +0.497 LS +0.093

If Q is L and SEN is P and LS is P then

SE = -0.014 Q +0.468 SEN +0.498 LS +0.123

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

5.5.3. Roll Sensor FISs

As there are the same number of roll sensor recovery FISs which are tested on the same number of faults as the yaw sensor, then once again there are too many results too display fully. Therefore the results for the two best tuned FISs for the roll sensor will be displayed. The two FISs will now be displayed.

The best ANFIS tuned FIS was the one tuned for 200 epochs. The three input membership functions for the best ANFIS tuned roll sensor recovery FIS are shown in Figures 5.36, 5.37 and 5.38.

The skews shown in Figure 5.36 are due to the input membership functions being based on the data show in Figure 5.16 where most of the data was of high value. The skews shown in Figures 5.37 and 5.38 are not as prominent as the data used for these inputs was evenly distributed.

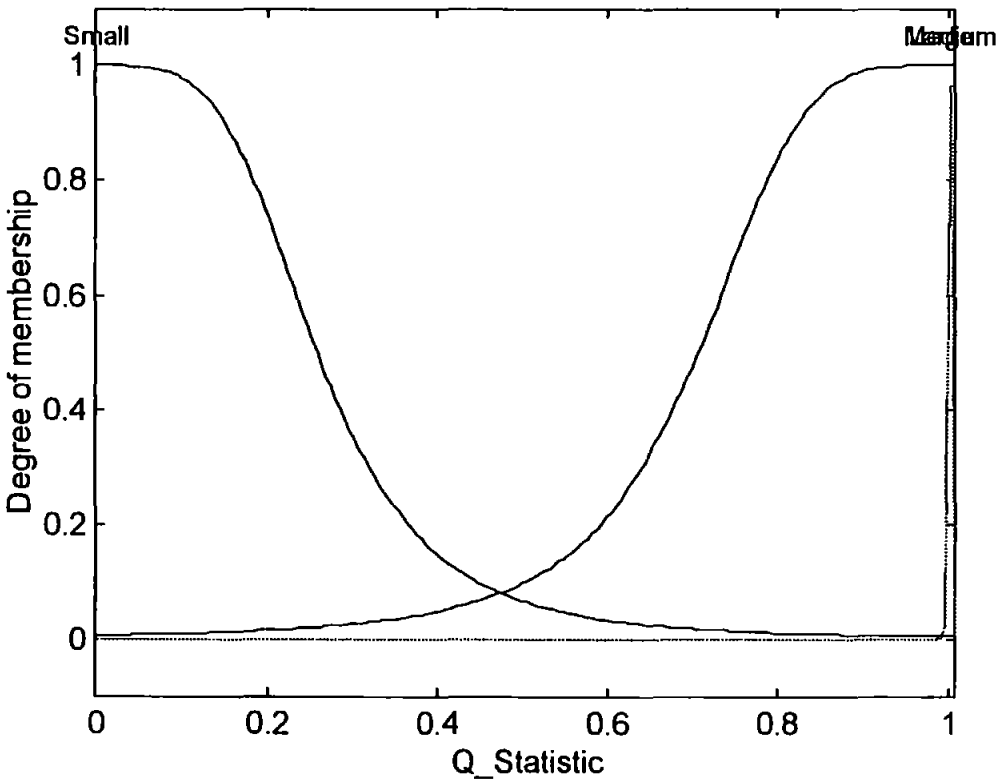


Figure 5.36 ANFIS Tuned Membership Functions for Q Statistic Input

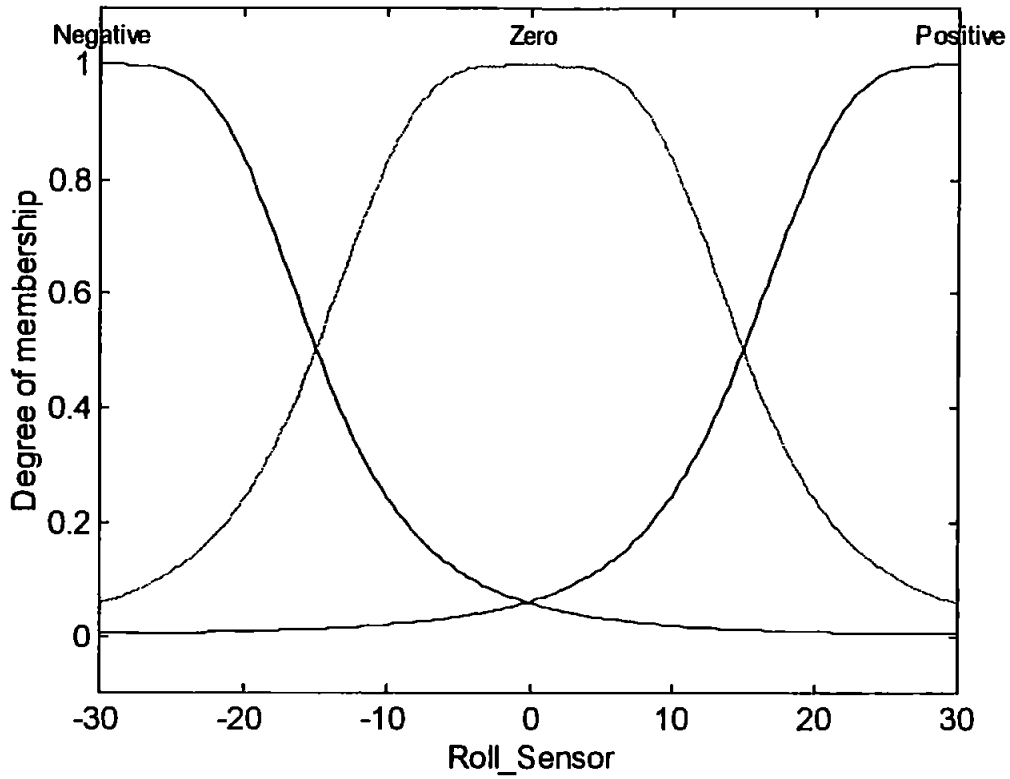


Figure 5.37 ANFIS Tuned Membership Functions for Roll Sensor Input

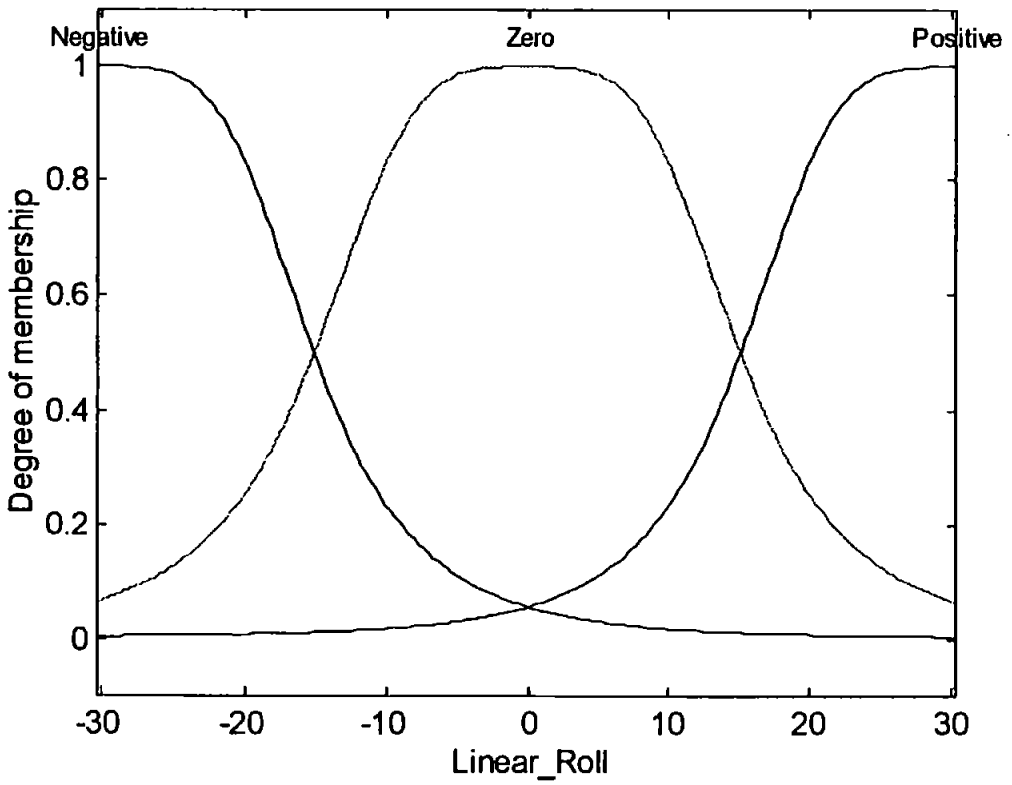


Figure 5.38 ANFIS Tuned Membership Functions for Linear Roll Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.9). Owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

SE = -0.908 Q +0.009 SEN +1.054 LS +2.813

If Q is S and SEN is N and LS is Z then

SE = -67.78 Q –0.662 SEN +1.845 LS –2.077

If Q is S and SEN is N and LS is P then

SE = 0.088 Q +1.897 SEN –0.130 LS +1.930

If Q is S and SEN is Z and LS is N then

SE = -81.73 Q –0.123 SEN +1.049 LS +0.937

If Q is S and SEN is Z and LS is Z then

SE = -18.73 Q +0.031 SEN +0.908 LS –0.132

If Q is S and SEN is Z and LS is P then

SE = 74.21 Q –0.153 SEN +1.056 LS +0.081

If Q is S and SEN is P and LS is N then

SE = -0.687 Q +1.499 SEN +0.106 LS –2.755

If Q is S and SEN is P and LS is Z then

SE = 69.84 Q –0.774 SEN +1.804 LS +3.169

If Q is S and SEN is P and LS is P then

SE = 9.920 Q +0.020 SEN +1.030 LS –1.883

If Q is M and SEN is N and LS is N then

SE = -31.98 Q +1.278 SEN +1.934 LS –107.6

If Q is M and SEN is N and LS is Z then

SE = -431.6 Q –10.47 SEN –65.24 LS –816.1

If Q is M and SEN is N and LS is P then

SE = 72.94 Q +3.916 SEN +78.62 LS +96.27

If Q is M and SEN is Z and LS is N then

SE = 639.1 Q +5.384 SEN +0.162 LS –670.3

If Q is M and SEN is Z and LS is Z then

SE = -9.890 Q –15.39 SEN +11.39 LS +82.45 (5.9)

If Q is M and SEN is Z and LS is P then

SE = -386.3 Q +12.51 SEN –5.735 LS +405.8

If Q is M and SEN is P and LS is N then

SE = 487.8 Q –258.2 SEN +11.80 LS +447.8

If Q is M and SEN is P and LS is Z then

SE = -223.1 Q +102.1 SEN –78.06 LS +213.3

If Q is M and SEN is P and LS is P then

SE = -0.052 Q –2.630 SEN +8.210 LS –1.79.0

If Q is L and SEN is N and LS is N then

SE = 31.77 Q –0.935 SEN +2.585 LS +11.83

If Q is L and SEN is N and LS is Z then

SE = -155.8 Q +15.09 SEN –6.085 LS +317.7

If Q is L and SEN is N and LS is P then

SE = -266.2 Q –17.02 SEN +37.32 LS –205.8

If Q is L and SEN is Z and LS is N then

SE = 143.8 Q +0.148 SEN –0.227 LS –150.5

If Q is L and SEN is Z and LS is Z then

SE = 1.247 Q +1.840 SEN +0.358 LS +2.571

If Q is L and SEN is Z and LS is P then

SE = 9.146 Q +1.350 SEN +0.390 LS –3.227

If Q is L and SEN is P and LS is N then

SE = 47.92 Q +81.26 SEN +31.28 LS +14.76

If Q is L and SEN is P and LS is Z then

SE = 0.031 Q +12.15 SEN –1.390 LS –191.9

If Q is L and SEN is P and LS is P then

SE = -24.95 Q –0.940 SEN +2.803 LS –23.61

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The best simulated annealing tuned FIS was the one tuned for 500 epochs. The three input membership functions for the best simulated annealing tuned roll sensor recovery FIS are shown in Figures 5.39, 5.40 and 5.41.

The skews shown in Figures 5.39, 5.40 and 5.41 are due to tuning performed by the simulated annealing method.

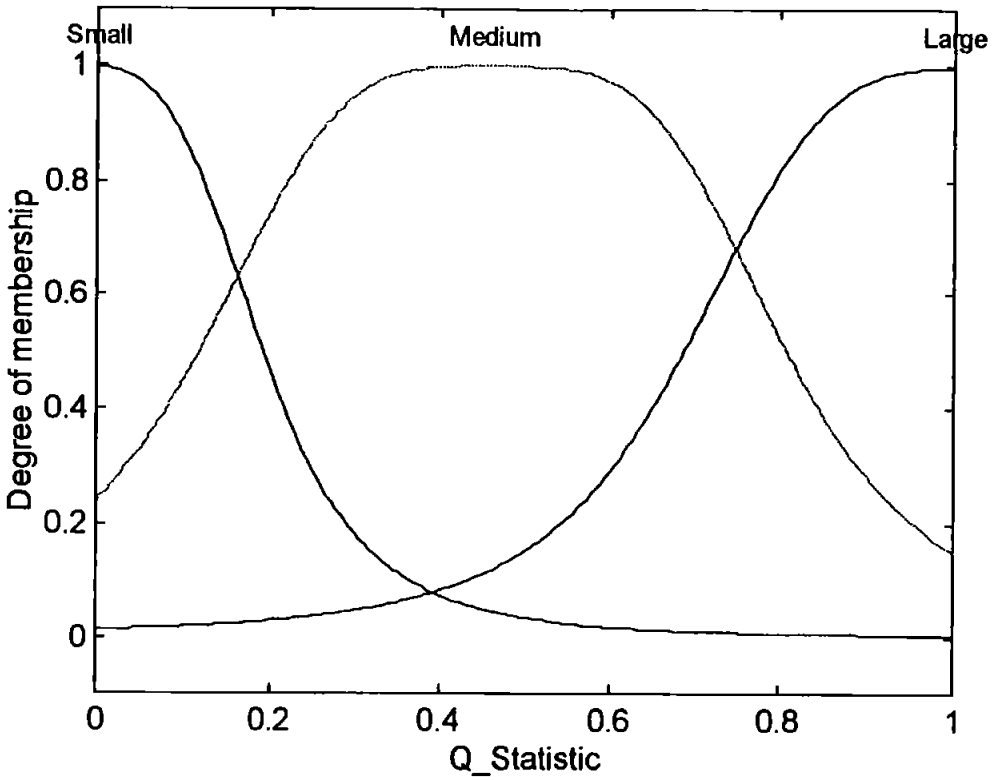


Figure 5.39 Simulated Annealing Tuned Membership Functions for Q Statistic Input

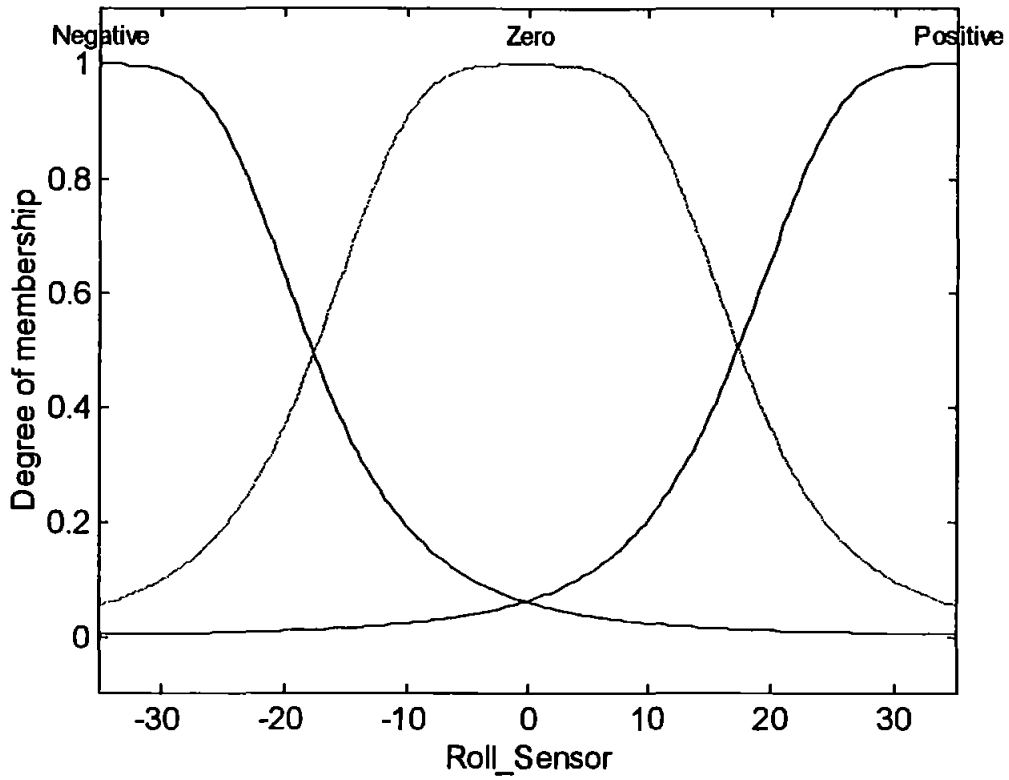


Figure 5.40 Simulated Annealing Tuned Membership Functions for Roll Sensor Input

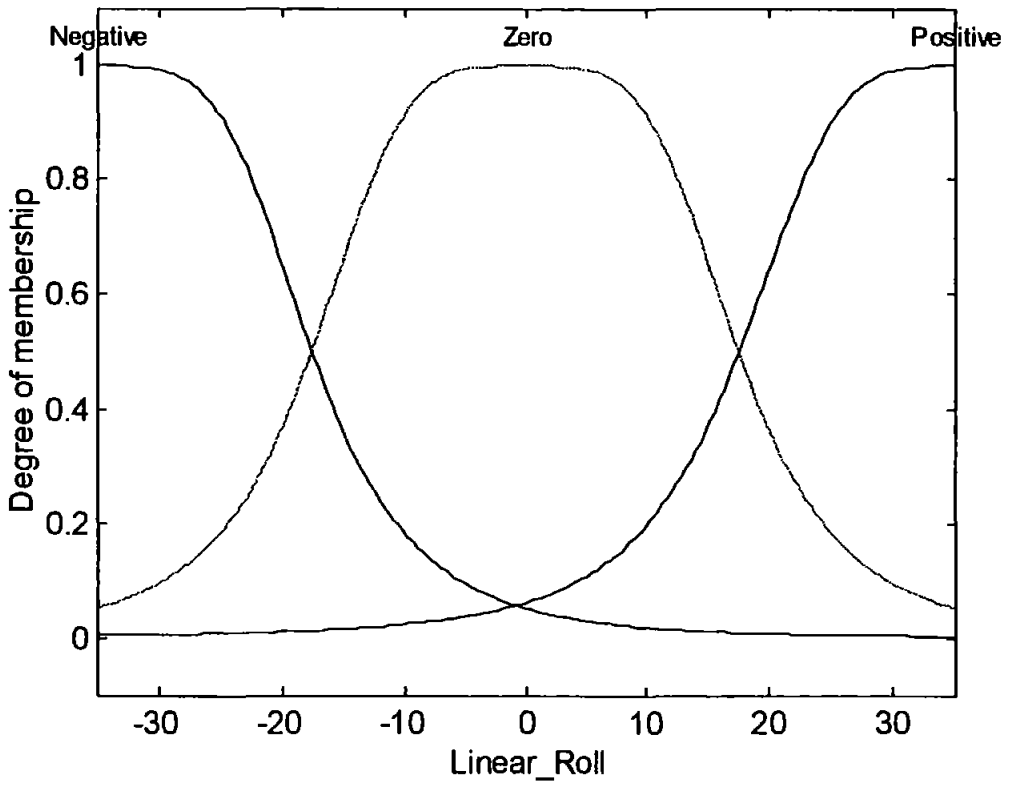


Figure 5.41 Simulated Annealing Tuned Membership Functions for Linear Roll Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.10). Again owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

$$SE = -0.031\,Q + 1.073\,SEN - 0.022\,LS - 0.056$$

If Q is S and SEN is N and LS is Z then

$$SE = 0.076\,Q + 0.431\,SEN + 0.470\,LS - 0.047$$

If Q is S and SEN is N and LS is P then

$$SE = 0.040\,Q + 0.007\,SEN + 0.956\,LS + 0.181$$

If Q is S and SEN is Z and LS is N then

$$SE = -0.026\,Q + 1.001\,SEN + 0.052\,LS - 0.017$$

If Q is S and SEN is Z and LS is Z then

$$SE = 0.011\,Q + 0.424\,SEN + 0.581\,LS + 0.012$$

If Q is S and SEN is Z and LS is P then

$$SE = 0.111\,Q - 0.117\,SEN + 1.108\,LS + 0.228$$

If Q is S and SEN is P and LS is N then

$$SE = 0.031\,Q + 1.109\,SEN - 0.056\,LS + 0.214$$

If Q is S and SEN is P and LS is Z then

$$SE = -0.078\,Q + 0.357\,SEN + 0.531\,LS - 0.024$$

If Q is S and SEN is P and LS is P then

$$SE = 0.076\,Q - 0.039\,SEN + 1.016\,LS - 0.019$$

If Q is M and SEN is N and LS is N then

$$SE = 0.032\,Q + 0.904\,SEN + 0.036\,LS + 0.001$$

If Q is M and SEN is N and LS is Z then

$$SE = -0.110\,Q + 0.541\,SEN + 0.499\,LS + 0.165$$

If Q is M and SEN is N and LS is P then

$$SE = 0.136\,Q - 0.061\,SEN + 0.873\,LS + 0.078$$

If Q is M and SEN is Z and LS is N then

$$SE = -0.005\,Q + 1.033\,SEN + 0.065\,LS - 0.227$$

If Q is M and SEN is Z and LS is Z then

$$SE = 0.036\,Q + 0.499\,SEN + 0.455\,LS + 0.038 \qquad (5.10)$$

If Q is M and SEN is Z and LS is P then

SE = -0.093 Q +0.043 SEN +0.969 LS +0.009

If Q is M and SEN is P and LS is N then

SE = -0.129 Q +0.930 SEN +0.177 LS –0.063

If Q is M and SEN is P and LS is Z then

SE = -0.077 Q +0.333 SEN +0.546 LS –0.150

If Q is M and SEN is P and LS is P then

SE = 0.101 Q +0.040 SEN +1.034 LS –0.164

If Q is L and SEN is N and LS is N then

SE = -0.163 Q +1.036 SEN +0.005 LS –0.129

If Q is L and SEN is N and LS is Z then

SE = -0.011 Q +0.569 SEN +0.589 LS –0.041

If Q is L and SEN is N and LS is P then

SE = -0.028 Q –0.144 SEN +1.085 LS +0.018

If Q is L and SEN is Z and LS is N then

SE = -0.057 Q +0.996 SEN –0.033 LS +0.065

If Q is L and SEN is Z and LS is Z then

SE = 0.021 Q +0.532 SEN +0.371 LS +0

If Q is L and SEN is Z and LS is P then

SE = 0.088 Q +0.020 SEN +1.190 LS –0.043

If Q is L and SEN is P and LS is N then

SE = 0.047 Q +0.860 SEN +0.079 LS –0.021

If Q is L and SEN is P and LS is Z then

SE = 0.101 Q +0.479 SEN +0.385 LS –0.041

If Q is L and SEN is P and LS is P then

SE = -0.102 Q +0.007 SEN +0.803 LS –0.041

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

5.5.4. Roll Rate Sensor FISs

As there are the same number of roll rate sensor recovery FISs which are tested on the same number of faults as the roll sensor, then once again there are too many results too display fully. Therefore the results for the two best tuned FISs for the roll sensor will be displayed. The two FISs will now be displayed.

The best ANFIS tuned FIS was the one tuned for 100 epochs. The three input membership functions for the best ANFIS tuned roll rate sensor recovery FIS are shown in Figures 5.42, 5.43 and 5.44.

The skews shown in Figure 5.42 are due to the input membership functions being based on the data show in Figure 5.20 where most of the data was of high value. The skews shown in Figures 5.43 and 5.44 are not as prominent as the data used for these inputs was evenly distributed.

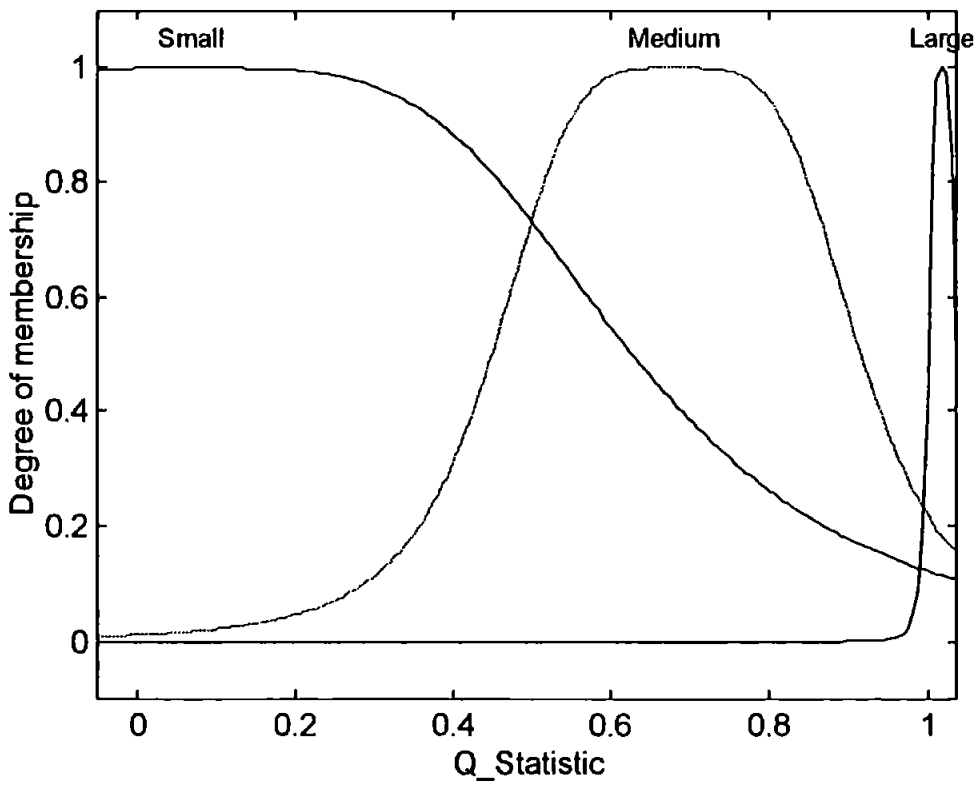


Figure 5.42 ANFIS Tuned Membership Functions for Q Statistic Input

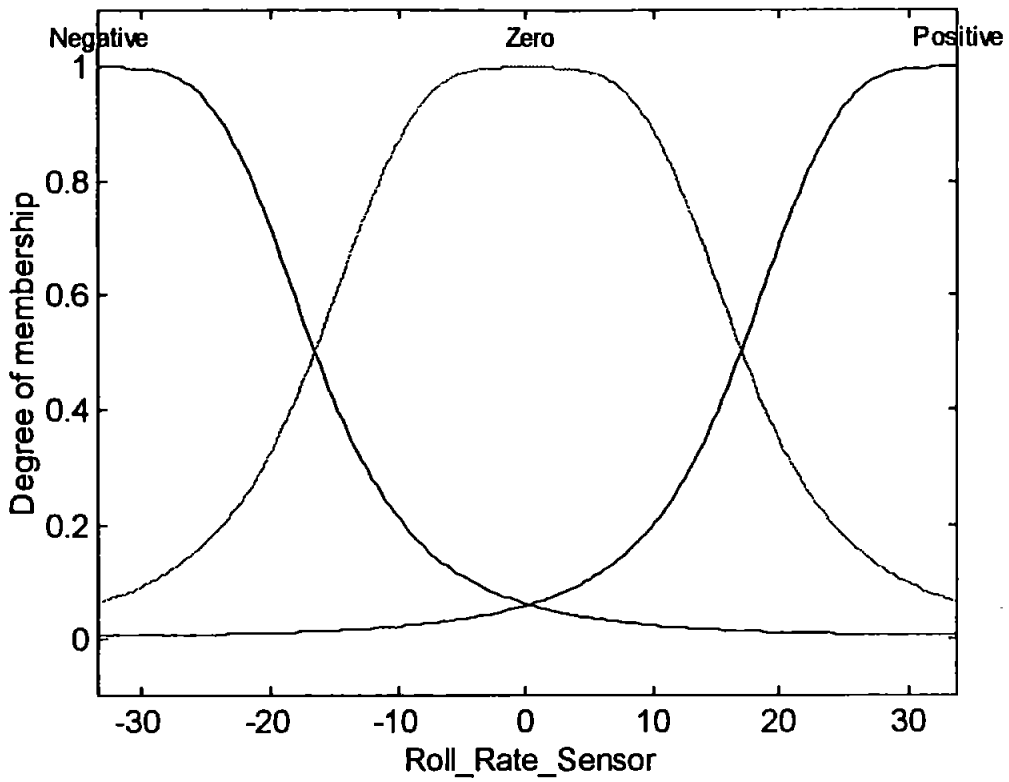


Figure 5.43 ANFIS Tuned Membership Functions for Roll Rate Sensor Input

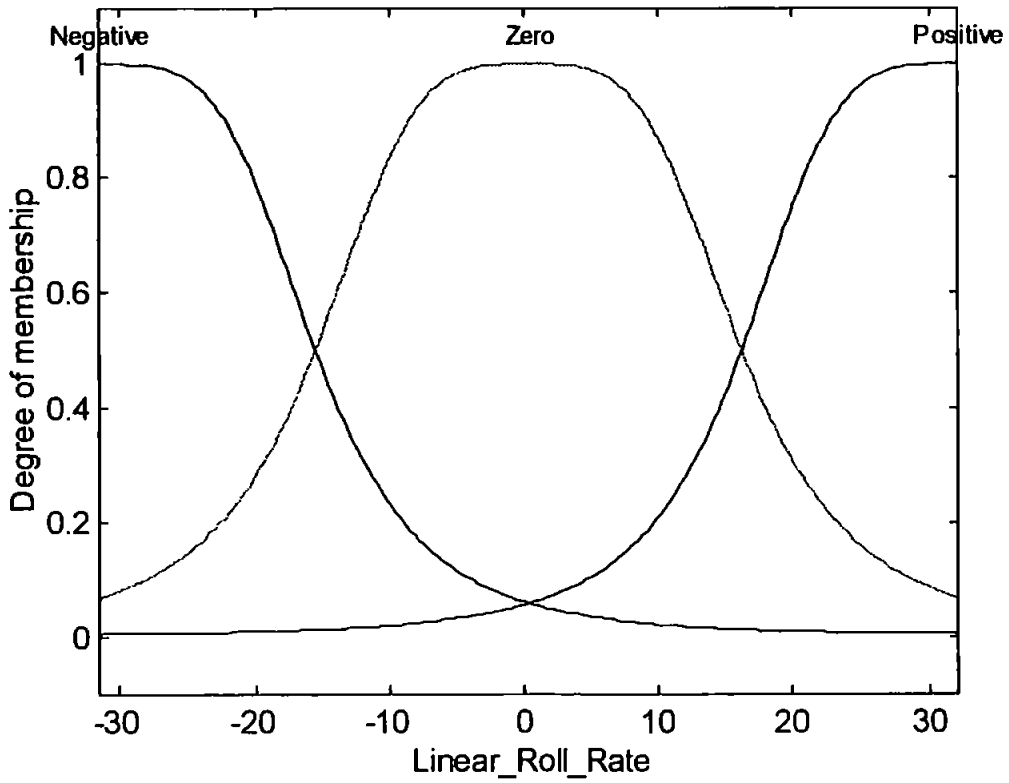


Figure 5.44 ANFIS Tuned Membership Functions for Linear Roll Rate Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.11). Owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results.

If Q is S and SEN is N and LS is N then

$$SE = -15.16\,Q + 7.795\,SEN - 6.745\,LS + 21.37$$

If Q is S and SEN is N and LS is Z then

$$SE = -152.2\,Q - 24.61\,SEN + 37.49\,LS + 193.1$$

If Q is S and SEN is N and LS is P then

$$SE = 281.6\,Q - 1005\,SEN + 1220\,LS + 149.8$$

If Q is S and SEN is Z and LS is N then

$$SE = 105.6\,Q + 14.64\,SEN - 9.439\,LS + 78.53$$

If Q is S and SEN is Z and LS is Z then

$$SE = 0.634\,Q + 10.99\,SEN - 8.191\,LS - 1.003$$

If Q is S and SEN is Z and LS is P then

$$SE = -105.6\,Q + 19.72\,SEN - 21.52\,LS - 23.00$$

If Q is S and SEN is P and LS is N then

$$SE = -529.3\,Q - 1193\,SEN + 929.5\,LS - 514.4$$

If Q is S and SEN is P and LS is Z then

$$SE = 151.9\,Q + 6.437\,SEN + 11.26\,LS - 227.0$$

If Q is S and SEN is P and LS is P then

$$SE = 11.24\,Q + 8.340\,SEN - 7.534\,LS - 14.42$$

If Q is M and SEN is N and LS is N then

$$SE = -6.534\,Q - 5.038\,SEN + 5.932\,LS + 27.82$$

If Q is M and SEN is N and LS is Z then

$$SE = 28.54\,Q + 0.873\,SEN - 10.67\,LS - 118.0$$

If Q is M and SEN is N and LS is P then

$$SE = 772.7\,Q + 769.8\,SEN - 661.1\,LS + 530.7$$

If Q is M and SEN is Z and LS is N then

$$SE = -28.45\,Q - 15.87\,SEN + 10.30\,LS - 32.82$$

If Q is M and SEN is Z and LS is Z then

$$SE = -1.244\,Q - 3.271\,SEN + 3.114\,LS + 5.087 \qquad (5.11)$$

If Q is M and SEN is Z and LS is P then

SE = 23.46 Q –14.95 SEN +13.33 LS +3.871

If Q is M and SEN is P and LS is N then

SE = -868.5 Q +963.1 SEN –654.9 LS –700.3

If Q is M and SEN is P and LS is Z then

SE = 15.24 Q –14.81 SEN +5.848 LS +71.82

If Q is M and SEN is P and LS is P then

SE = -2.209 Q –4.960 SEN +6.265 LS –36.93

If Q is L and SEN is N and LS is N then

SE = 743.9 Q +45.67 SEN +12.25 LS +545.5

If Q is L and SEN is N and LS is Z then

SE = 415.0 Q +126.0 SEN –105.4 LS +441.6

If Q is L and SEN is N and LS is P then

SE = -117.4 Q +556.4 SEN +1121 LS –9.898

If Q is L and SEN is Z and LS is N then

SE = 799.6 Q +32.53 SEN –10.70 LS –896.0

If Q is L and SEN is Z and LS is Z then

SE = -77.96 Q –15.89 SEN +17.74 LS –37.25

If Q is L and SEN is Z and LS is P then

SE = -745.9 Q +78.27 SEN –74.13 LS +769.0

If Q is L and SEN is P and LS is N then

SE = 244.5 Q –433.2 SEN +201.9 LS +124.2

If Q is L and SEN is P and LS is Z then

SE = 545.5 Q +46.04 SEN –163.0 LS +545.3

If Q is L and SEN is P and LS is P then

SE = -182.7 Q –76.93 SEN +105.9 LS –98.03

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The best simulated annealing tuned FIS was the one tuned for 200 epochs. The three input membership functions for the best simulated annealing tuned roll rate sensor recovery FIS are shown in Figures 5.45, 5.46 and 5.47.

The skews shown in Figures 5.45, 5.46 and 5.47 are due to tuning performed by the simulated annealing method.

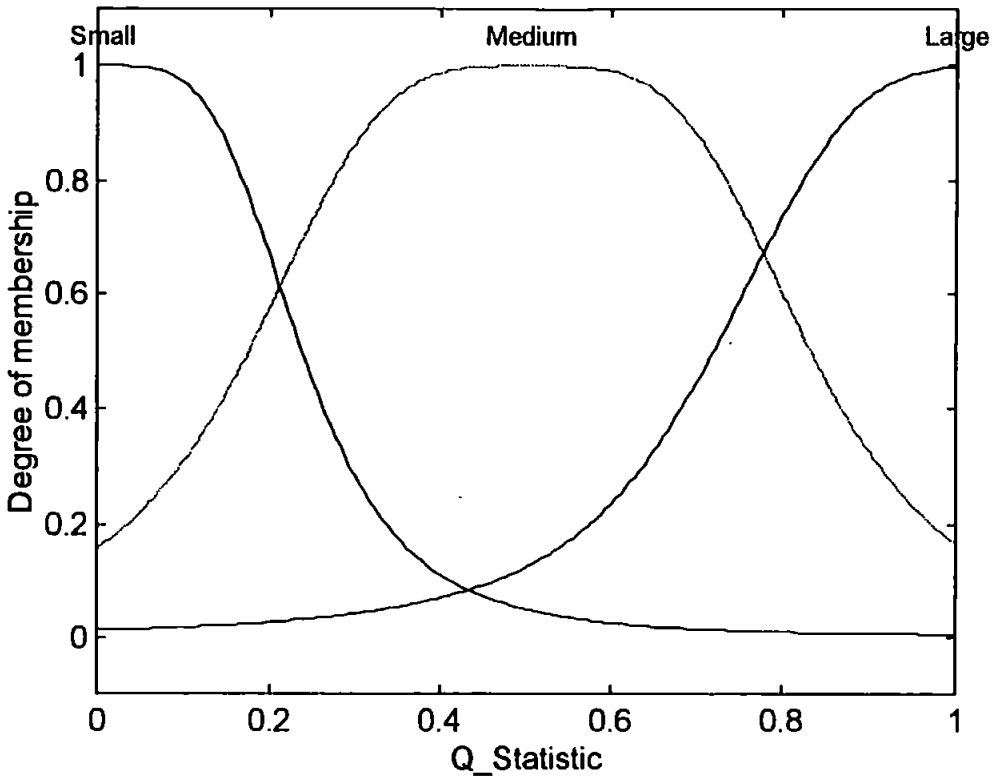


Figure 5.45 Simulated Annealing Tuned Membership Functions for Q Statistic Input

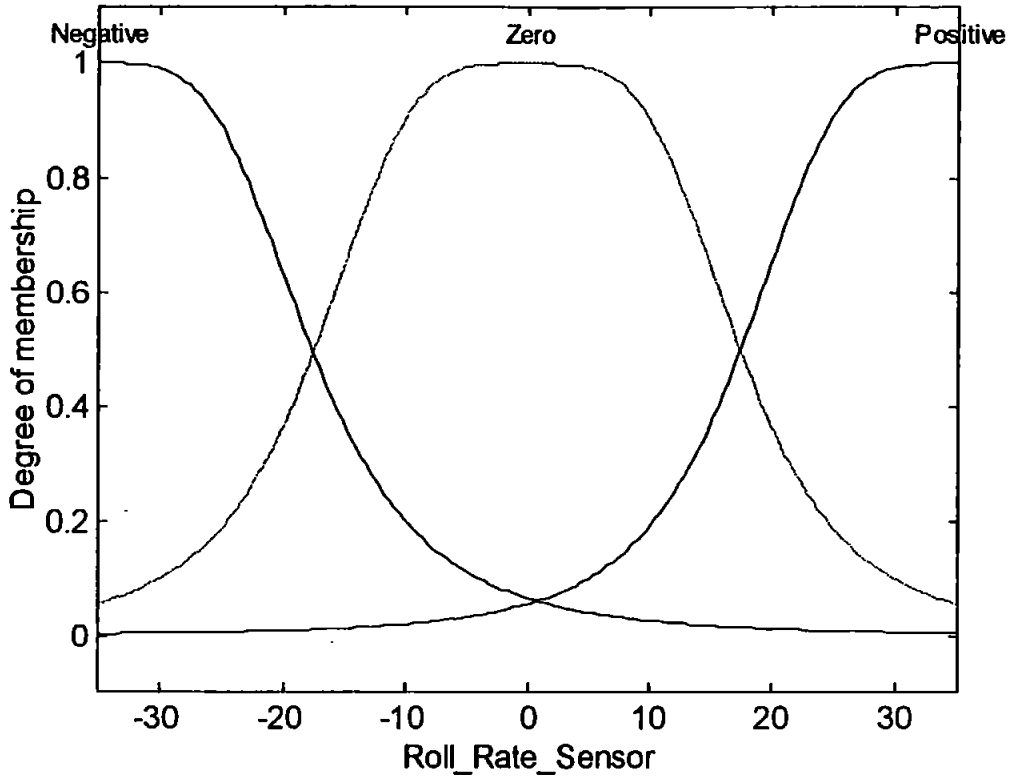


Figure 5.46 Simulated Annealing Tuned Membership Functions for Roll Rate Sensor Input

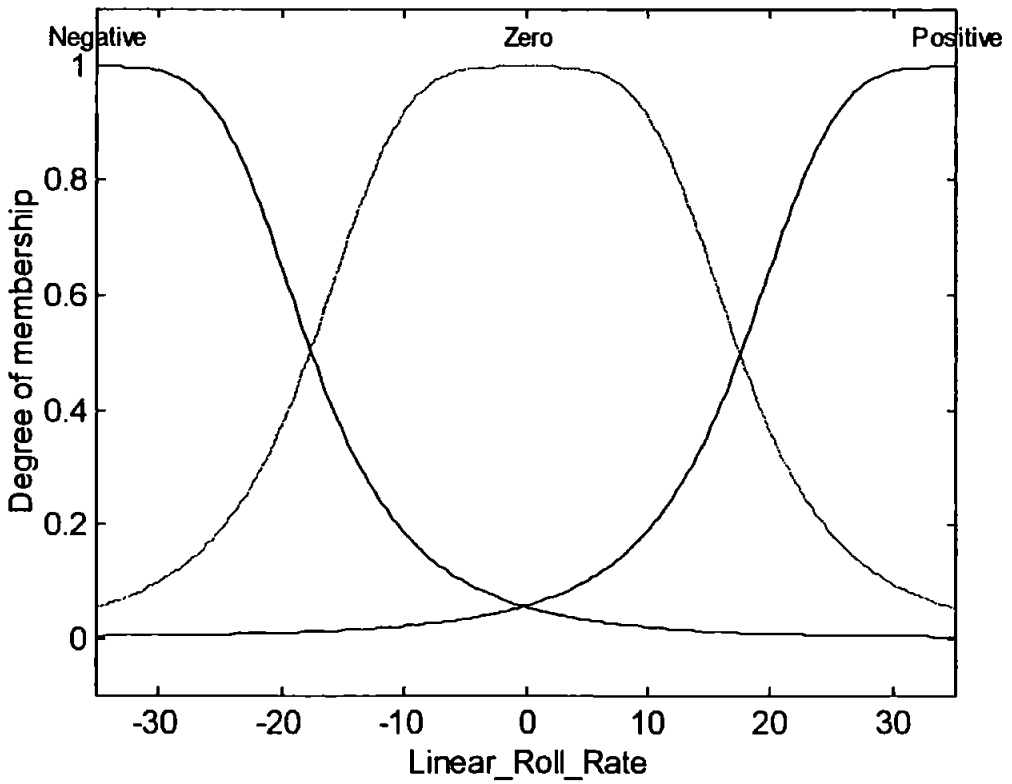


Figure 5.47 Simulated Annealing Tuned Membership Functions for Linear Roll Rate Input

These three sets of input membership functions when combined with the output function lead to the fuzzy rule base shown in Equation (5.12). Again owing to the FIS having three inputs it is impossible to show the output function as it requires a four dimensional surface to display the results

If Q is S and SEN is N and LS is N then

SE = -0.092 Q +1.047 SEN +0.063 LS –0.042

If Q is S and SEN is N and LS is Z then

SE = 0.123 Q +0.394 SEN +0.454 LS –0.035

If Q is S and SEN is N and LS is P then

SE = -0.010 Q –0.005 SEN +0.937 LS +0.166

If Q is S and SEN is Z and LS is N then

SE = -0.018 Q +0.927 SEN +0.065 LS +0.029

If Q is S and SEN is Z and LS is Z then

SE = -0.073 Q +0.448 SEN +0.570 LS +0.012

If Q is S and SEN is Z and LS is P then

SE = 0.123 Q –0.105 SEN +1.110 LS +0.255

If Q is S and SEN is P and LS is N then

SE = -0.040 Q +1.172 SEN –0.080 LS +0.203

If Q is S and SEN is P and LS is Z then

SE = -0.064 Q +0.361 SEN +0.591 LS –0.072

If Q is S and SEN is P and LS is P then

SE = -0.006 Q –0.031 SEN +1.053 LS –0.038

If Q is M and SEN is N and LS is N then

SE = 0.0002 Q +0.927 SEN +0.034 LS –0.015

If Q is M and SEN is N and LS is Z then

SE = -0.124 Q +0.544 SEN +0.487 LS +0.189

If Q is M and SEN is N and LS is P then

SE = 0.101 Q –0.051 SEN +0.891 LS +0.072

If Q is M and SEN is Z and LS is N then

SE = -0.038 Q +0.951 SEN +0.093 LS –0.196

If Q is M and SEN is Z and LS is Z then

SE = 0.056 Q +0.453 SEN +0.490 LS +0.008 (5.12)

If Q is M and SEN is Z and LS is P then

SE = -0.034 Q +0.061 SEN +0.973 LS +0.059

If Q is M and SEN is P and LS is N then

SE = -0.147 Q +0.882 SEN +0.131 LS +0.004

If Q is M and SEN is P and LS is Z then

SE = -0.052 Q +0.309 SEN +0.606 LS –0.104

If Q is M and SEN is P and LS is P then

SE = 0.061 Q +0.051 SEN +0.977 LS –0.126

If Q is L and SEN is N and LS is N then

SE = -0.124 Q +1.055 SEN +0.002 LS –0.085

If Q is L and SEN is N and LS is Z then

SE = -0.081 Q +0.587 SEN +0.577 LS –0.022

If Q is L and SEN is N and LS is P then

SE = -0.091 Q –0.109 SEN +1.087 LS –0.028

If Q is L and SEN is Z and LS is N then

SE = -0.111 Q +1.001 SEN –0.004 LS +0.030

If Q is L and SEN is Z and LS is Z then

SE = -0.043 Q +0.503 SEN +0.428 LS +0.002

If Q is L and SEN is Z and LS is P then

SE = 0.081 Q –0.010 SEN +1.131 LS –0.030

If Q is L and SEN is P and LS is N then

SE = 0.025 Q +0.904 SEN +0.092 LS –0.081

If Q is L and SEN is P and LS is Z then

SE = 0.066 Q +0.472 SEN +0.435 LS +0.022

If Q is L and SEN is P and LS is P then

SE = -0.078 Q +0.031 SEN +0.734 LS +0.006

Where Q is the value from the Q statistic, SEN is the sensor signal, LS is the linear model's signal, S is small, M is medium, L is Large, N is negative, Z is zero, P is positive and SE is the signal estimate produced by the FIS.

The results of placing these eight FISs in the control loop will now be displayed in the next four sections.

5.5.5. Yaw Sensor Failures

This first section will consider faults occurring in the yaw sensor. The section is divided into three subsections, each presenting results concerned with one of the three step input sizes. Each subsection will present results from both considered FISs and the Kalman filter enhance control system developed in Chapter 4. The complete set of results are presented in Appendix G.

(a) Yaw Step Inputs of 10 Degrees

This first subsection will look at faults occurring in the yaw sensor when a 10 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.1.

Table 5.1 The 10 Degrees Step Input RMSEs for Yaw Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.047	2.633	7.398	14.293	17.624	9.510	0.144	0.323	0.458
ANFIS	0.139	2.298	5.940	13.907	18.008	11.408	0.204	0.342	0.459
Simulated Annealing	0.200	2.599	6.512	10.078	11.310	6.091	0.228	0.339	0.443

Table 5.1 shows how both the ANFIS and simulated annealing tuned sensor recovery FISs performed for the considered sensor faults when attempting a yaw step input of 10 degrees. When there was no fault on the system (0% percentage signal loss) both FISs produced RMSEs larger than the Kalman filter enhanced control system. For the other considered levels of percentage signal loss tests considered the ANFIS tuned FIS was able to produce lower RMSE values for all but the 100% fault and the simulated annealing tuned FIS was able to produce lower RMSE values for all levels considered. When the intermittent total sensor failure is simulated within the system, the simulated annealing FIS produced a RMSE less than the Kalman filter enhanced

control system, but the ANFIS tuned FIS could not. When the signal to noise ratio (SNR) faults were implemented in the AUV both FISs failed to improve on the Kalman filter enhanced system for all but the highest level of noise considered. For the highest level of noise there was a small improvement in performance.

To show the level of improvements produced by using the FISs in place of the Kalman filter Table 5.2 has been calculated showing the percentage improvements for all tests.

Table 5.2 The 10 Degrees Step Input Percentage Increases for Yaw Sensor Faults.

Control System	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-194%	13%	20%	3%	-2%	-20%	-41%	-6%	0%
Simulated Annealing	-322%	1%	12%	29%	36%	36%	-58%	-5%	3%

Table 5.2 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.48 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

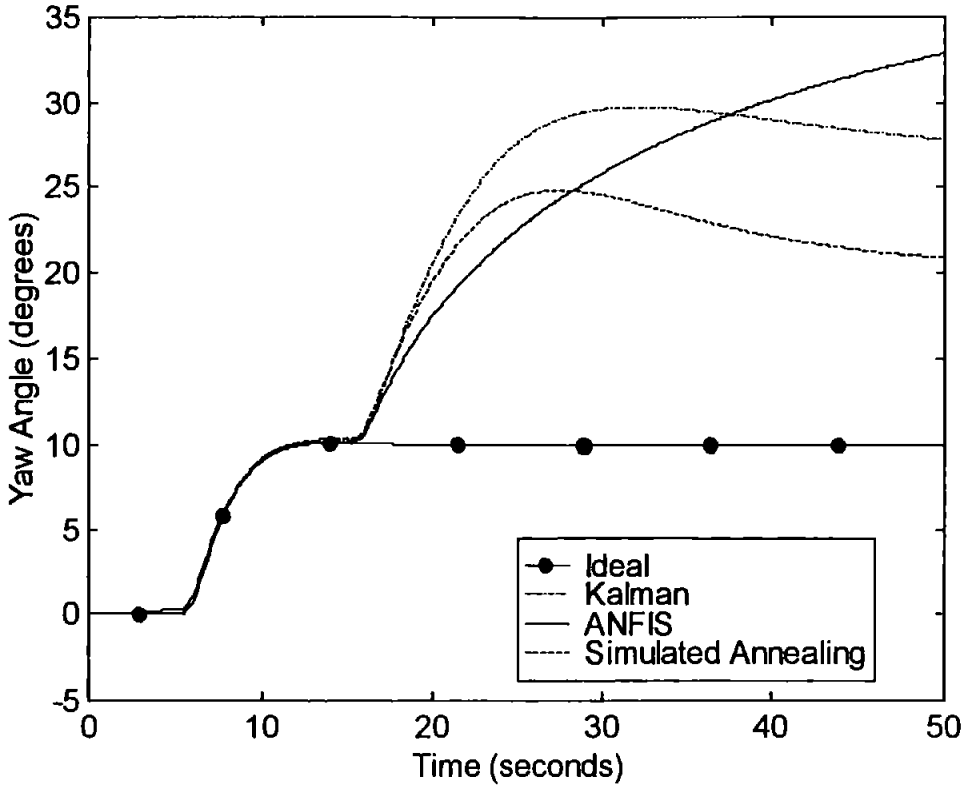


Figure 5.48 The Yaw Responses to a 75% Yaw Signal Loss for a 10 Degrees Step Yaw Demand

(b) Yaw Step Inputs of 50 Degrees

This subsection will look at faults occurring in the yaw sensor when a 50 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.3.

Table 5.3 The 50 Degrees Step Input RMSEs for Yaw Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.697	12.707	20.751	20.210	18.820	10.598	0.580	0.841	1.102
ANFIS	2.051	12.912	20.421	21.680	20.011	11.974	2.083	2.139	2.238
Simulated Annealing	1.138	10.520	6.922	14.150	17.575	7.666	1.352	1.469	1.639

As was the case for the 10 degrees step input demand, neither of the FISs are capable of handling the fault free situation. Again both FISs produced an increase in RMSE values for these tests. The simulated annealing tuned FIS was capable of improving on the Kalman filter enhanced control system for the other four levels of signal loss and the intermittent total signal loss. For the same set of tests the ANFIS tuned FIS was only once able to produce a lower RMSE value, for the 50% signal loss. When considering the SNR faults both FISs produced high RMSE values for all levels of noise considered.

To show the level of improvements produced by using the FISs in place of the Kalman filter Table 5.4 has been calculated showing the percentage improvements for all tests.

Table 5.4 The 50 Degrees Step Input Percentage Increases for Yaw Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-194%	-2%	2%	-7%	-6%	-13%	-259%	-154%	-103%
Simulated Annealing	-63%	17%	67%	30%	7%	28%	-133%	-75%	-49%

Table 5.4 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.49 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

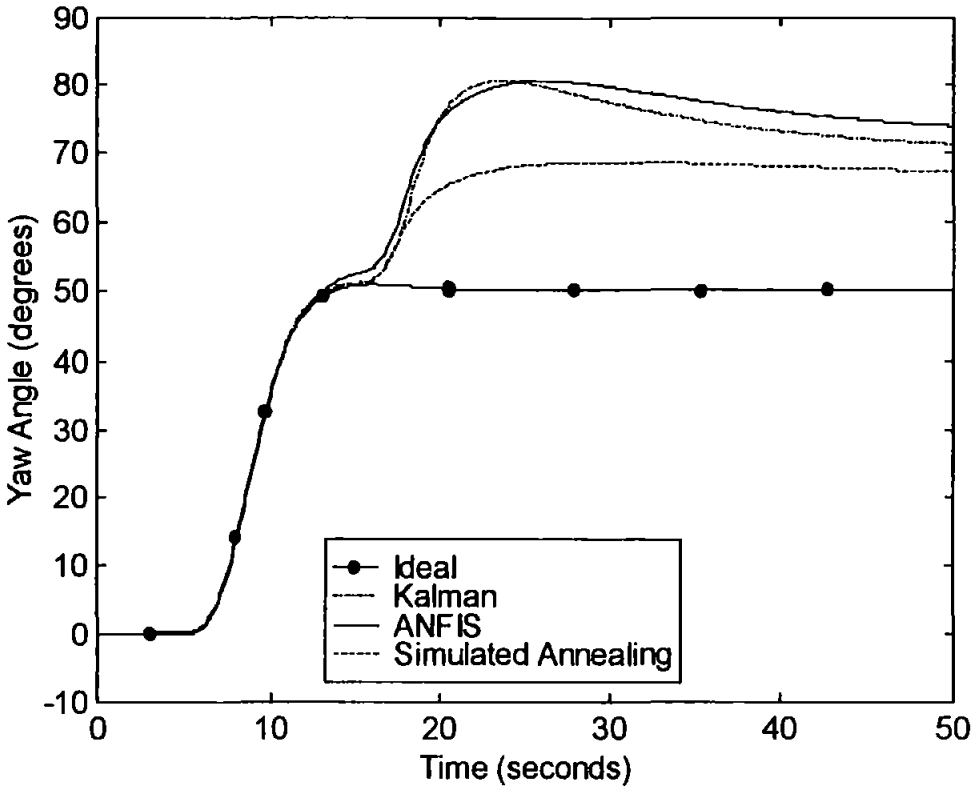


Figure 5.49 The Yaw Responses to a 75% Yaw Signal Loss for a 50 Degrees Step Yaw Demand

(c) Yaw Step Inputs of 90 Degrees

This subsection will look at faults occurring in the yaw sensor when a 90 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.5.

Table 5.5 The 90 Degrees Step Input RMSEs for Yaw Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.079	10.865	7.475	5.463	4.695	12.483	0.384	0.849	1.199
ANFIS	0.505	12.127	12.559	4.524	3.609	7.373	0.560	1.054	1.353
Simulated Annealing	0.451	9.455	0.908	7.229	5.709	6.817	0.439	0.913	1.336

Table 5.5 once again shows how both the FISs were unable to improve on the Kalman filter enhanced control system for the fault free situation and for all levels of noise in the SNR tests. For this, the large size of yaw step input demand considered the ANFIS tuned FIS did produce lower RMSE values for the 75% and 100% signal loss's and the intermittent total failure. The simulated annealing tuned FIS produced lower RMSE values for the 25% and 50% signal loss's and the intermittent total failure.

Again to show the level of improvements produced by using the FISs in place of the Kalman filter Table 5.6 has been calculated showing the percentage improvements for all tests.

Table 5.6 The 90 Degrees Step Input Percentage Increases for Yaw Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-543%	-12%	-68%	17%	23%	41%	-46%	-24%	-13%
Simulated Annealing	-475%	13%	88%	-32%	-22%	45%	-14%	-8%	-11%

Table 5.6 shows how much the performance has been changed due to the use of the sensor estimation FISs.

Figure 5.50 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss. The Kalman filter enhanced control system produced an estimate that enabled the AUV to achieve its desired yaw angle for the case shown. This is evidence of how well the Kalman filter can perform and highlights how effective the FISs need to be to be considered an improvement on the Kalman filter.

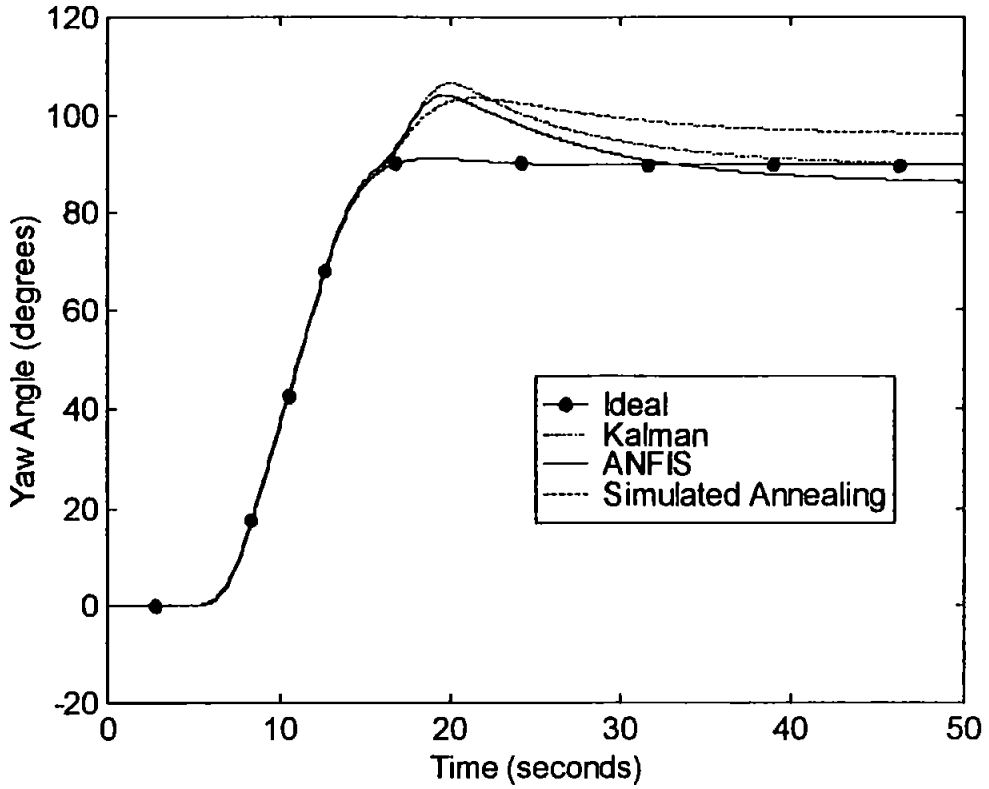


Figure 5.50 The Yaw Responses to a 75% Yaw Signal Loss for a 90 Degrees Step Yaw Demand

5.5.6. Yaw Rate Sensor Failures

The second section will look at faults occurring in the yaw rate sensor. The section is again divided into three subsections, each presenting results concerned with one of the three step input sizes. Each subsection will present results from both considered FISs and the Kalman filter enhance control system developed in Chapter 4. The complete set of results are presented in Appendix G.

(a) Yaw Step Inputs of 10 Degrees

The first subsection in this section will look at faults occurring in the yaw rate sensor when a 10 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.7. It should be noted about

these values is that they are in degrees per second as the yaw rate is now being considered.

Table 5.7 The 10 Degrees Step Input RMSEs for Yaw Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.003	0.025	0.054	0.089	0.130	0.218	0.045	0.100	0.141
ANFIS	0.044	0.050	0.074	0.105	0.139	0.186	0.054	0.094	0.131
Simulated Annealing	0.033	0.045	0.059	0.075	0.093	0.135	0.041	0.063	0.082

As was the case for the yaw sensor tests, the yaw rate sensor recovery FIS for both methods of tuning have failed to improve on the Kalman filter enhanced system when the system is fault free. The ANFIS tuned FIS also produced larger RMSE values for all percentage signal loss tests. The simulated annealing tuned FIS produced lower RMSE values for both the 75% and 100% signal loss tests. Both FISs were capable of producing RMSE values lower than that of the Kalman filter enhanced control system for the intermittent total signal failure. When considering the SNR tests the FISs designed for the yaw rate sensor produced a better set of results at this step size than the yaw sensor FISs. With only one case of a decrease in performance being noted, that of the 1% SNR when the ANFIS tuned FIS was tested.

Again as was the case for the yaw sensor, to show the level of improvements produced by using the FISs in place of the Kalman filter Table 5.8 has been calculated showing the percentage improvements for all tests.

Table 5.8 The 10 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-1226%	-105%	-36%	-18%	-7%	14%	-20%	6%	7%
Simulated Annealing	-893%	-83%	-9%	16%	28%	38%	8%	37%	42%

Table 5.8 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.51 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

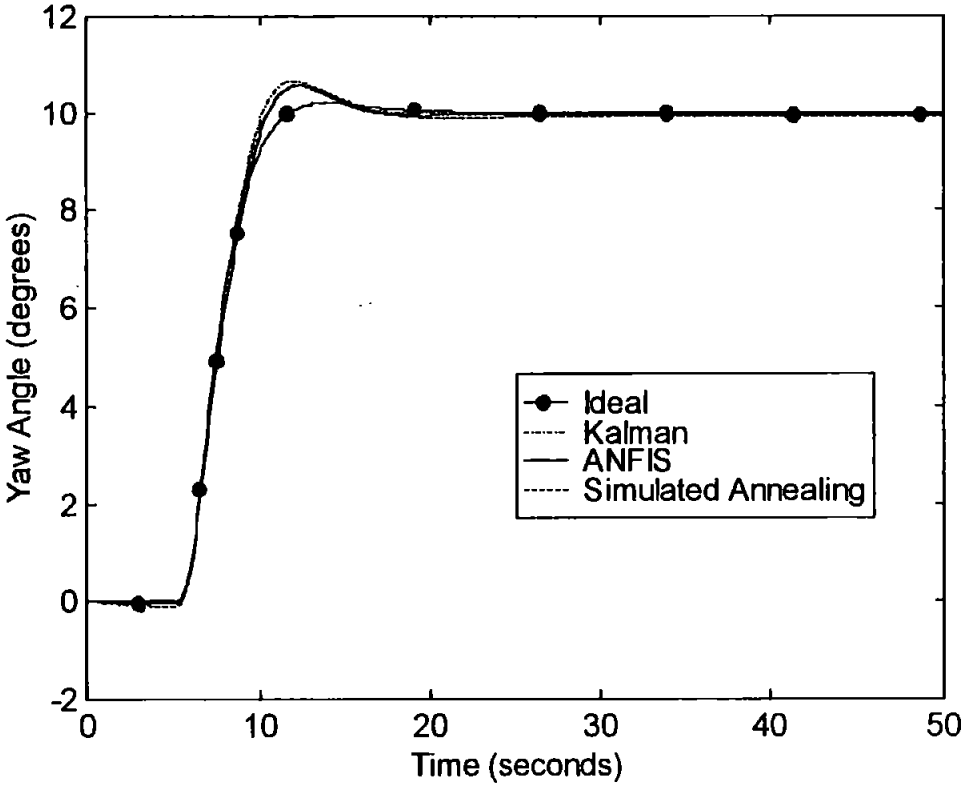


Figure 5.51 The Yaw Responses to a 75% Yaw Rate Signal Loss for a 10 Degrees Step Yaw Demand

(b) Yaw Step Inputs of 50 Degrees

This subsection will look at faults occurring in the yaw rate sensor when a 50 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.9.

Table 5.9 The 50 Degrees Step Input RMSEs for Yaw Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.007	0.206	0.341	0.435	0.432	0.249	0.084	0.182	0.251
ANFIS	0.122	0.046	0.151	0.212	0.798	0.430	0.113	0.190	0.272
Simulated Annealing	0.068	0.156	0.246	0.341	0.441	0.436	0.065	0.095	0.127

Once again the first column of results show how both FISs produced high RMSE values for the fault free system. For the tests where there was a 25%, 50% and 75% signal loss both FISs were able to generate RMSE values lower than those recorded for the Kalman filter enhanced control system. For both the total and intermittent total signal failures both FIS performed worse than the benchmark system. The SNR results show a difference in performance between the two considered FISs, with the ANFIS tuned FIS producing high RMSE values and the simulated annealing tuned FIS producing low RMSE values, relative to the benchmark results, for all levels of noise.

Again to show the level of improvements produced by using the FISs in place of the Kalman filter Table 5.10 has been calculated showing the percentage improvements for all tests.

Table 5.10 The 50 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-1564%	78%	56%	51%	-85%	-73%	-36%	-4%	-8%
Simulated Annealing	-824%	24%	28%	22%	-2%	-75%	22%	48%	49%

Table 5.10 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.52 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

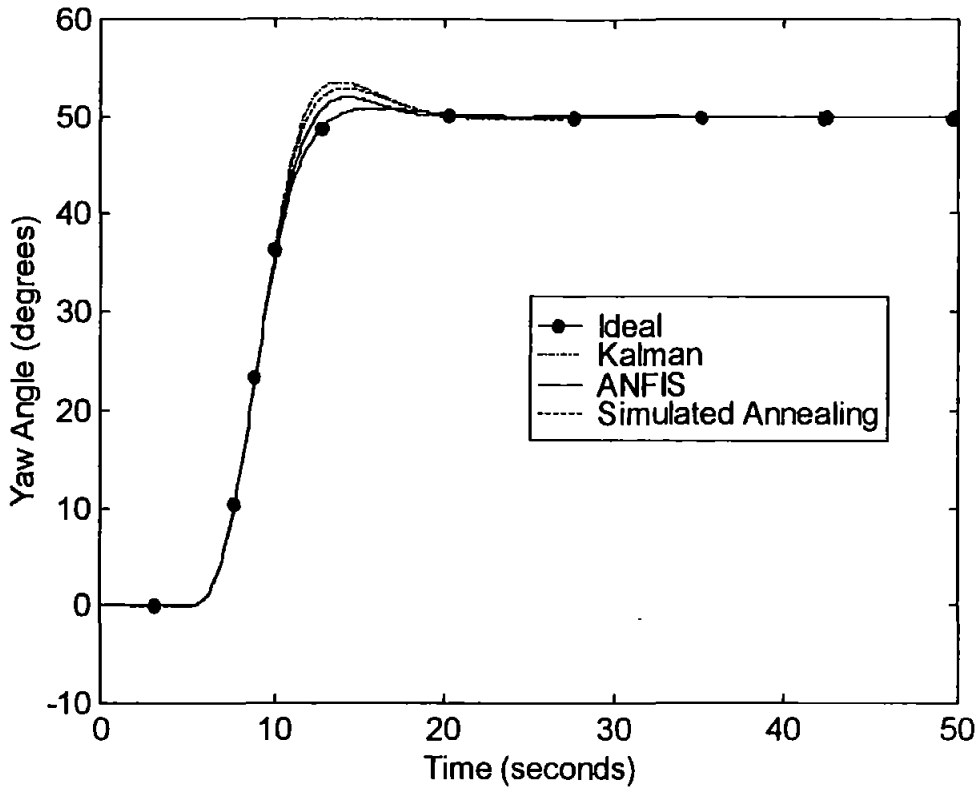


Figure 5.52 The Yaw Responses to a 75% Yaw Rate Signal Loss for a 50 Degrees Step Yaw Demand

(c) Yaw Step Inputs of 90 Degrees

The final subsection is concerned with faults in the yaw channel will look at faults occurring in the yaw rate sensor when a 90 degrees yaw step input demand has been placed on the AUV. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.11.

Table 5.11 The 90 Degrees Step Input RMSEs for Yaw Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.008	0.180	0.253	0.248	0.212	0.275	0.078	0.181	0.244
ANFIS	0.096	0.030	0.065	0.120	0.203	0.125	0.115	0.197	0.272
Simulated Annealing	0.057	0.107	0.208	0.304	0.425	0.423	0.068	0.109	0.148

For this size of step input demand, as with the two other sizes and all tests in the yaw sensor, both FISs again produced RMSE values higher than the Kalman filter enhanced control system. The ANFIS tuned FIS generated RMSE values less than the benchmark system for all levels of percentage signal loss tests, however the simulated annealing tuned FIS could only accomplish this for the 25% and 50% tests. For the intermittent total signal failure tests the ANFIS tuned FIS produced a low RMSE value, while the simulated annealing tuned FIS produced a high RMSE value, relative to the benchmark value. This leaves the SNR tests in the yaw rate sensor as the last to be considered for the yaw channel. The results show that the simulated annealing tuned FIS produced RMSE values lower than the benchmark values for all levels of noise and the ANFIS one produced higher RMSE values for all levels of noise considered.

As has been the case for all work in the yaw channel thus far the level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests and are shown in Table 5.12.

Table 5.12 The 90 Degrees Step Input Percentage Increases for Yaw Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	-1100%	83%	74%	52%	4%	54%	-47%	-9%	-12%
Simulated Annealing	-604%	41%	18%	-23%	-100%	-54%	13%	40%	39%

Table 5.12 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.53 shows the yaw responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

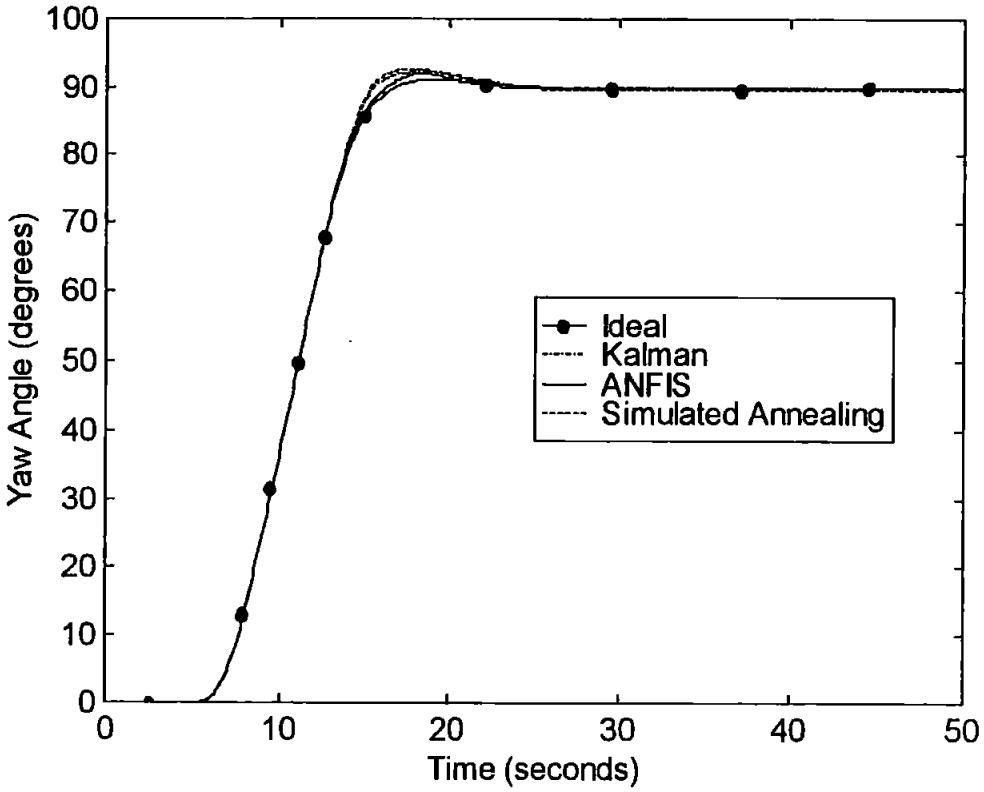


Figure 5.53 The Yaw Responses to a 75% Yaw Rate Signal Loss for a 90 Degrees Step Yaw Demand

This brings to a close the sections concerned with faults occurring within the sensors related to the yaw channel of the AUV.

5.5.7. Roll Sensor Failures

The third section will look at faults occurring in the roll sensor. The section is again divided into three subsections. Each presenting results concerned with one of the three initial roll angle sizes. Each subsection will present results from both considered FISs and the Kalman filter enhance control system developed in Chapter 4. The complete set of results are presented in Appendix H.

(a) Initial Roll Angles of 5 Degrees

The first subsection concerned with faults in the roll channel will look at faults occurring in the roll sensor when an initial roll angle of 5 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.13.

Table 5.13 The Initial Angle of 5 Degrees RMSEs for Roll Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.285	0.283	0.283	0.283	0.283	0.283	0.283	0.283	0.283
ANFIS	0.081	0.188	0.172	0.196	0.203	0.193	0.199	0.210	0.221
Simulated Annealing	0.106	0.111	0.118	0.182	0.305	0.304	0.104	0.102	0.101

The first point of interest in Table 5.13 is that both FISs were able to improve on the benchmark results for the fault free system. This was not achieved by any of the yaw channel systems. The ANFIS tuned FIS produced lower RMSE values for all tests at this size of initial roll angle. The simulate annealing tuned FIS did not perform as well with two tests (100% signal loss and intermittent total failure) producing RMSE values larger than the benchmark results.

As has been the case for all work in the yaw channel, the level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing

the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.14.

Table 5.14 The Initial Angle of 5 Degrees Percentage Increases for Roll Sensor Faults.

Control System	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	51%	48%	40%	27%	26%	28%	28%	28%	28%
Simulated Annealing	62%	60%	58%	37%	-8%	-7%	37%	37%	37%

Table 5.14 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.54 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

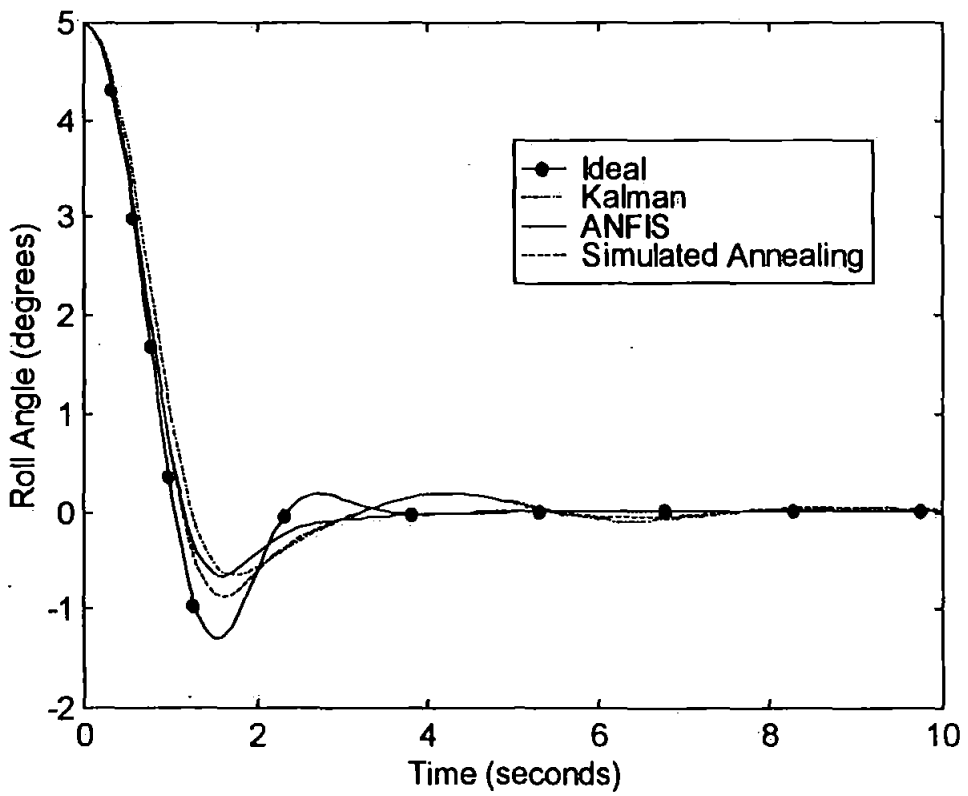


Figure 5.54 The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 5 Degrees

(b) Initial Roll Angles of 15 Degrees

The next subsection will look at faults occurring in the roll sensor when an initial roll angle of 15 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.15.

Table 5.15 The Initial Angle of 15 Degrees RMSEs for Roll Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.475	0.475	0.475	0.475	0.475	0.475	0.475	0.475	0.475
ANFIS	0.087	0.178	0.325	0.363	0.419	0.332	0.369	0.377	0.385
Simulated Annealing	0.234	0.223	0.227	0.232	0.248	0.246	0.233	0.230	0.229

Once again, as was the case for the 5 degrees initial roll angle, the ANFIS tuned FIS was able to generate smaller RMSE values than the Kalman filter enhanced system for all nine tests. For this size of initial roll angle the simulated annealing tuned FIS also produced smaller RMSE values than the Kalman filter enhanced system for all nine tests.

Again the level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.16.

Table 5.16 The Initial Angle of 15 Degrees Percentage Increases for Roll Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	82%	62%	32%	24%	12%	30%	22%	21%	19%
Simulated Annealing	51%	53%	52%	51%	48%	48%	51%	52%	52%

Table 5.16 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.55 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

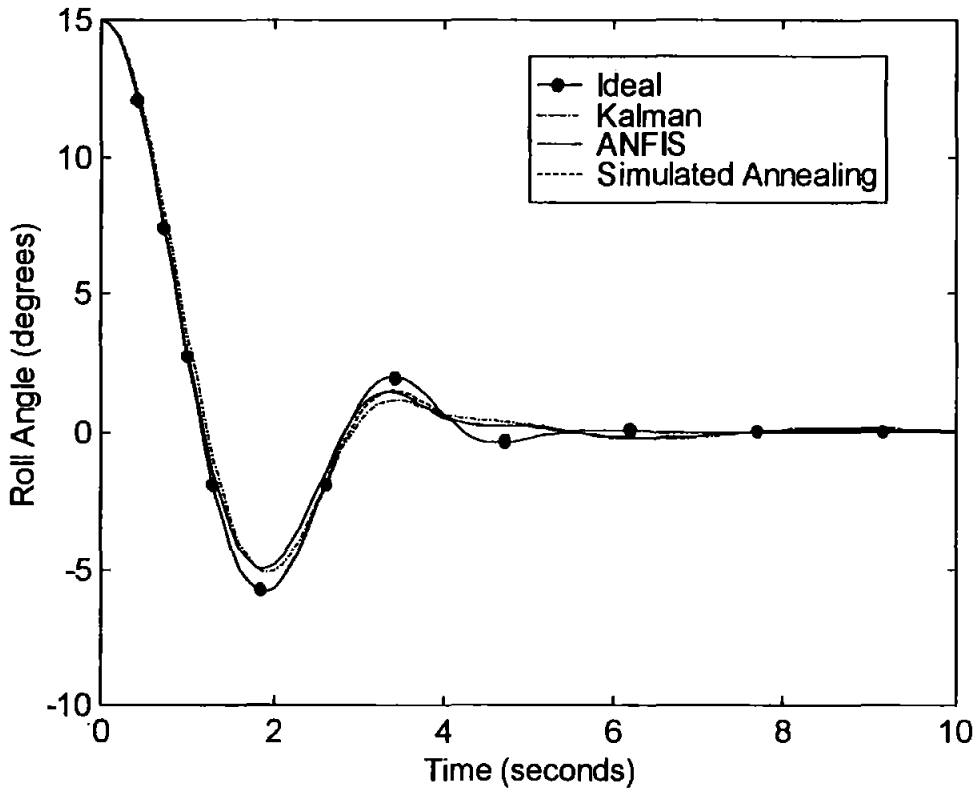


Figure 5.55 The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 15 Degrees

(c) Initial Roll Angles of 25 Degrees

This subsection will look at faults occurring in the roll sensor when an initial roll angle of 25 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.17.

Table 5.17 The Initial Angle of 25 Degrees RMSEs for Roll Sensor Faults.

	RMSEs (degrees)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.406	0.407	0.407	0.407	0.407	0.407	0.406	0.406	0.406
ANFIS	0.098	0.048	0.154	0.225	0.274	0.674	0.191	0.443	0.572
Simulated Annealing	0.226	0.228	0.255	0.323	0.383	0.229	0.225	0.226	0.228

The size of initial roll angle considered here produces the first negative results for the ANFIS tuned FIS. The FIS failed to improve on the benchmark results for the intermittent total failure and the 5% and 10% SNR tests. The simulated annealing tuned FIS was able to improve on every one of the benchmark results.

Again the level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.18.

Table 5.18 The Initial Angle of 25 Degrees Percentage Increases for Roll Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	76%	88%	62%	45%	33%	-66%	53%	-9%	-41%
Simulated Annealing	44%	44%	37%	21%	6%	44%	45%	44%	44%

Table 5.18 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.56 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

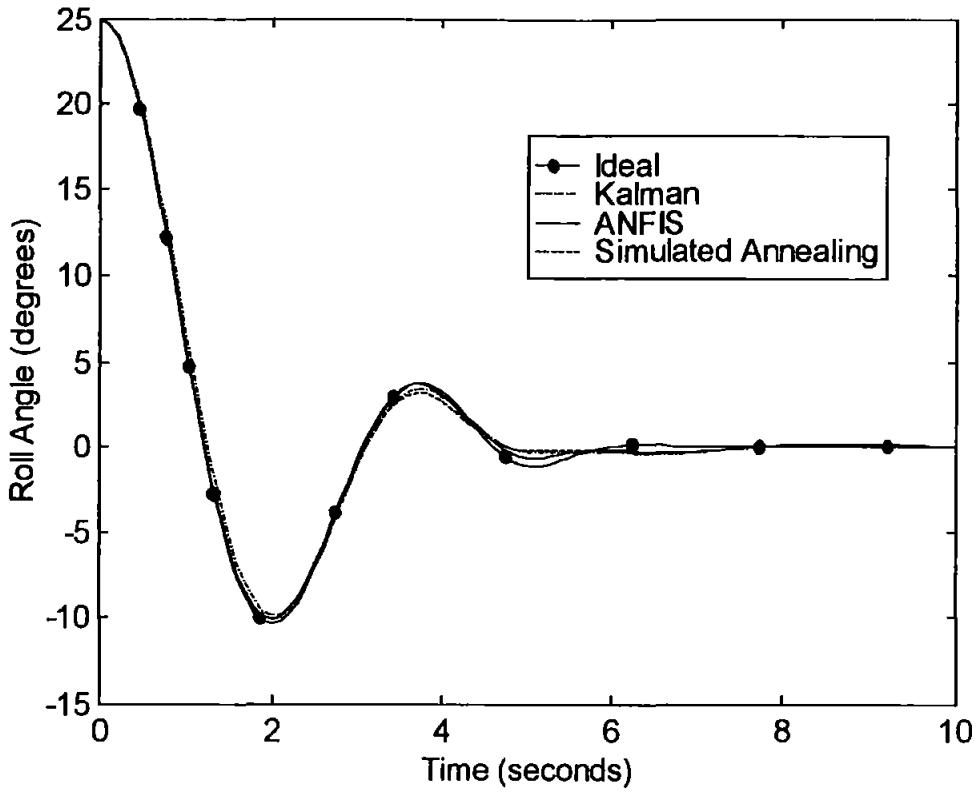


Figure 5.56 The Roll Responses to a 75% Roll Signal Loss for an Initial Roll Angle of 25 Degrees

5.5.8. Roll Rate Sensor Failures

The final section presenting results will look at faults occurring in the roll rate sensor. The section is again divided into three subsections. Each presenting results concerned with one of the three initial roll angle sizes. Each subsection will present results from both considered FISs and the Kalman filter enhance control system developed in Chapter 4. The complete set of results are presented in Appendix H.

(a) Initial Roll Angles of 5 Degrees

The first subsection of this section will look at faults occurring in the roll rate sensor when an initial roll angle of 5 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.19.

Table 5.19 The Initial Angle of 5 Degrees RMSEs for Roll Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.302	0.306	0.325	0.332	0.335	0.333	0.332	0.332	0.331
ANFIS	0.170	0.220	0.253	0.293	0.379	0.183	0.286	0.279	0.276
Simulated Annealing	0.317	0.339	0.362	0.377	0.391	0.362	0.312	0.306	0.301

As was the case for the roll sensor recovery FIS, the roll rate recovery ANFIS tuned FIS was able to improve on the benchmark results for a fault free system. As can been from Table 5.19 only one test lead to a lower RMSE value being recorded, that of the 100% signal loss test. The simulated annealing tuned FIS produced RMSE values similar to those of the Kalman filter enhanced control system. It can be seen from Table 5.20 that every RMSE was within 20% of the benchmark values.

As was the case for the roll sensor tests the level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.20.

Table 5.20 The Initial Angle of 5 Degrees Percentage Increases for Roll Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	44%	28%	22%	12%	-13%	45%	14%	16%	17%
Simulated Annealing	-5%	-11%	-11%	-13%	-17%	-9%	6%	8%	9%

Table 5.20 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.57 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

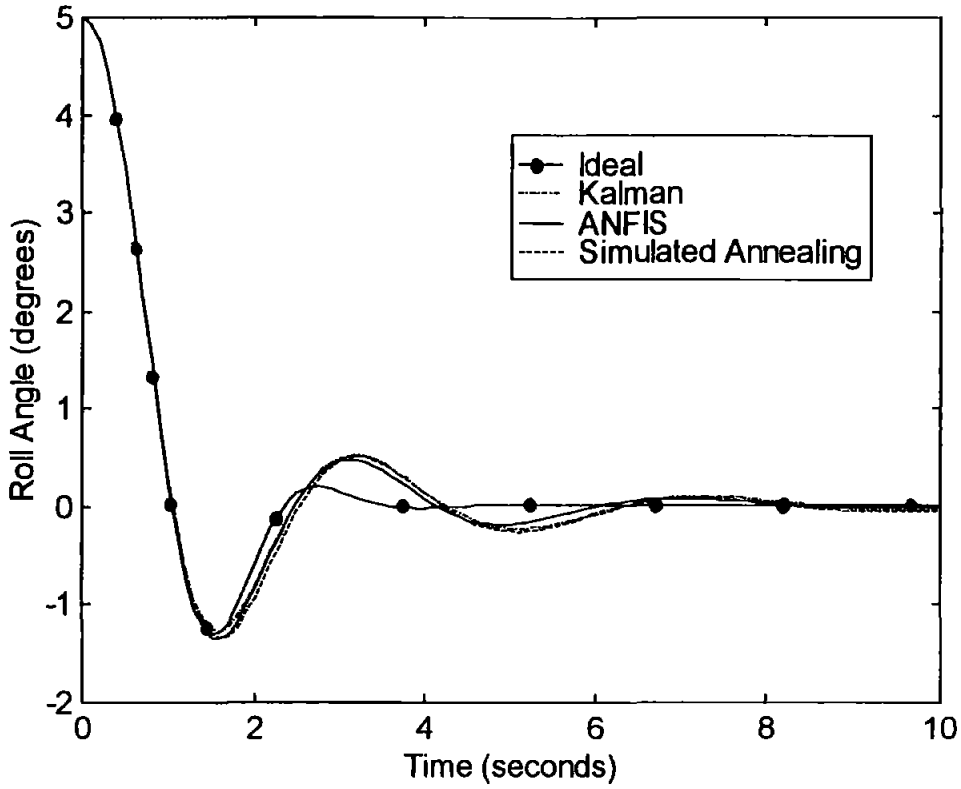


Figure 5.57 The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 5 Degrees

(b) Initial Roll Angles of 15 Degrees

This subsection will look at faults occurring in the roll rate sensor when an initial roll angle of 15 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.21.

Table 5.21 The Initial Angle of 15 Degrees RMSEs for Roll Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.516	0.527	0.538	0.539	0.540	0.539	0.539	0.539	0.539
ANFIS	0.365	0.384	0.433	0.500	0.684	0.578	0.478	0.456	0.440
Simulated Annealing	0.522	0.536	0.549	0.561	0.576	0.546	0.520	0.515	0.508

The ANFIS tuned FIS again produced lower RMSE for most of the tests including all three SNR tests, despite the FIS having never been tuned using SNR data. The two tests where the FIS failed to improve on the benchmark results were the 100% signal loss and the intermittent total failure tests. The simulated annealing FIS again produced results similar to those of the benchmark ones. For this initial roll angle all the results were within 7% of the benchmark values.

Again level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.22.

Table 5.22 The Initial Angle of 15 Degrees Percentage Increases for Roll Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	29%	27%	19%	7%	-27%	-7%	11%	15%	18%
Simulated Annealing	-1%	-2%	-2%	-4%	-7%	-1%	4%	4%	6%

Table 5.22 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.58 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

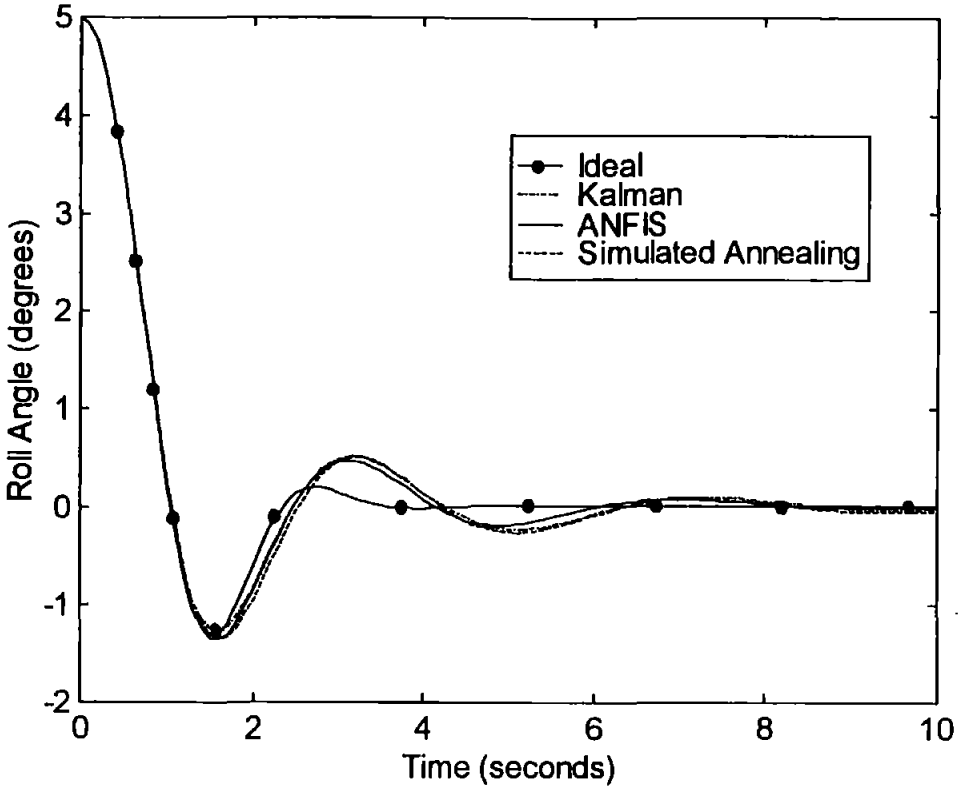


Figure 5.58 The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 15 Degrees

(c) Initial Roll Angles of 25 Degrees

The final subsection of this section will look at faults occurring in the roll rate sensor when an initial roll angle of 25 degrees is considered. The three types of faults defined in Chapter 3 will all be considered. The two FISs have been compared to the Kalman filter enhanced control system of Chapter 4 using RMSE values. The complete set of RMSEs are shown in Table 5.23.

Table 5.23 The Initial Angle of 25 Degrees RMSEs for Roll Rate Sensor Faults.

	RMSEs (degrees per second)								
Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
Kalman Filter	0.566	0.574	0.582	0.584	0.584	0.584	0.567	0.568	0.570
ANFIS	0.412	0.486	0.729	0.655	0.620	0.690	0.241	0.290	0.342
Simulated Annealing	0.563	0.615	0.631	0.647	0.663	0.606	0.557	0.549	0.543

The ANFIS tuned FIS was again able to produce a RMSE value lower than the benchmark one for the fault free test, and has now managed that at all three initial roll angles for the roll rate sensor tests. It could only manage to produce a lower RMSE value for the 25% signal loss of the percentage signal loss tests and also failed to produce a lower result for the intermittent total failure. The FIS produced lower RMSE values for all three SNR tests for this initial roll angle, as has been the case for the previous two initial roll angles considered.

The simulated annealing tuned FIS again produced RMSE values similar to the Kalman filter enhanced system. For this size of initial roll angle the RMSE values were all within 11% of the benchmark ones.

Again level of improvements produced by using the FISs in place of the Kalman filter has been calculated showing the percentage improvements for all tests for this size of initial roll angle and are shown in Table 5.24.

Table 5.24 The Initial Angle of 25 Degrees Percentage Increases for Roll Rate Sensor Faults.

Control	Percentage Signal Loss					Intermittent	Signal to Noise Ratio		
System	0%	25%	50%	75%	100%	Total Failure	1%	5%	10%
ANFIS	27%	15%	-25%	-12%	-6%	-18%	58%	49%	40%
Simulated Annealing	0%	-7%	-8%	-11%	-14%	-4%	2%	3%	5%

Table 5.24 shows how much the performance has been changed due to the use of the sensor estimation FISs. Figure 5.59 shows the roll responses of both considered FISs along with the Kalman filter enhanced system and the ideal system for a 75% signal loss.

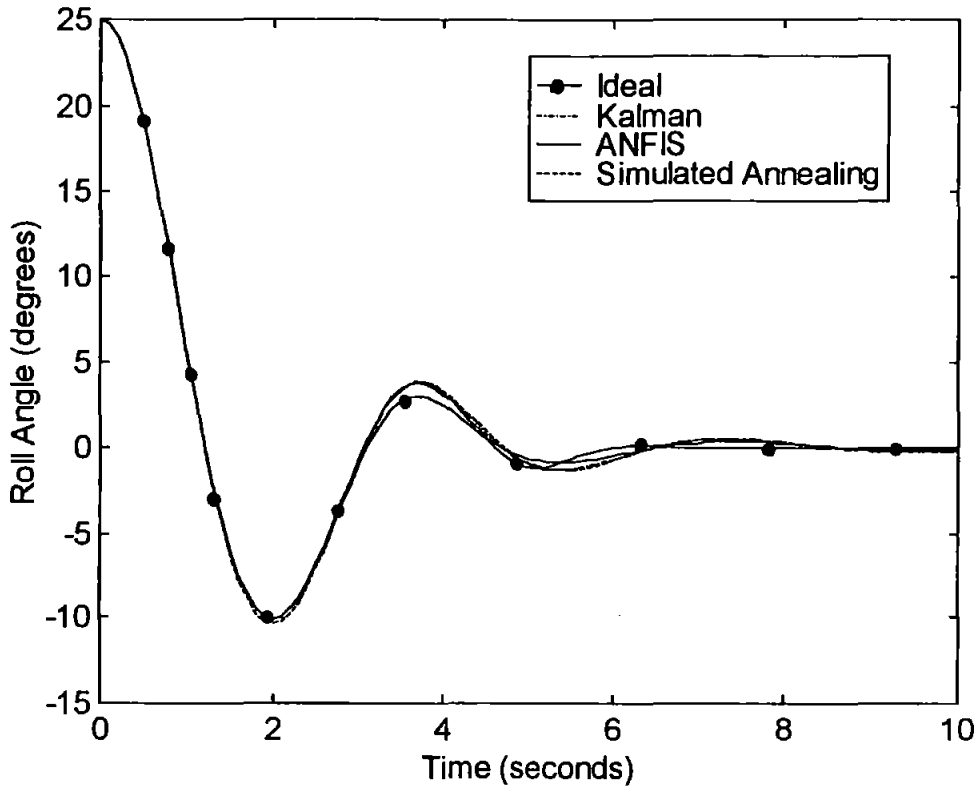


Figure 5.59 The Roll Responses to a 75% Roll Rate Signal Loss for an Initial Roll Angle of 25 Degrees

5.6. DISCUSSION AND CONCLUDING REMARKS

The aim of this Chapter was to replace the Kalman filter with a sensor recovery FIS. Two FISs, tuned using different methods ANFIS [Jang (1991)] and simulated annealing [Kirkpatrick et *al* (1983)]), were developed for each of the four considered sensors and then compared to the results of the Kalman filter enhanced control system developed in Chapter 4.

5.6.1. Yaw Sensor

The yaw channel results are discussed first. The first point to be noted is the inability of any of the FISs, for either the yaw or yaw rate sensors to achieve desirable RMSE values for the fault free system. This problem is probably due to the inaccuracy of the linear model used. For a fault free system the linear model would act identically to the actual system. The fault free tests can be considered a special case as for all other tests

there is some form of fault occurring. It would be possible to add a fuzzy rule to system for this special case, thereby allowing the sensor signal to be transmitted unaltered. This is a conceptually simple idea, but would require the FIS to be almost totally restructured to allow for the added rule. It should also be noted that the Kalman filter did produce some small RMSE values for the fault free system.

When considering tests where faults have occurred in the system there is an increase in performance of all of the tuned FISs. The first sensor recovery FISs tested in this Chapter are those used in the yaw sensor of the AUV. The ANFIS tuned FIS for this sensor was able to improve on the benchmark results for eight of the twenty-four tests where a fault has occurred. The best performance was for the 100% signal loss (90 degrees step input) test where an increase of 23% was calculated, and the worst performance was for the 1% SNR (50 degrees step input) test where a decrease of 259% was calculated. The simulated annealing tuned FIS for this sensor was able to improve on the benchmark results for fourteen of the twenty-four tests where a fault has occurred. The best performance was for the 50% signal loss (90 degrees step input) test where an increase of 88% was calculated, and the worst performance was for the 1% SNR (50 degrees step input) test where a decrease of 133% was calculated. It should be noted here that neither of the FISs were trained using SNR data and so it not surprising that both FIS performed poorly on those tests. It was important to test them on those faults, as an increase in performance would have been very impressive. Of the two tuned FISs presented here the one that made the overall system most fault tolerant to the considered faults is the one tuned by the simulated annealing method. It is possible that if both FISs had been tuned on SNR data, as well as the data used, they would have produced better results.

5.6.2. Yaw Rate Sensor

The yaw rate sensor recovery FISs were tested next. The ANFIS tuned FIS for this sensor was able to improve on the benchmark results for eleven of the twenty-four tests where a fault has occurred. The best performance was for the 25% signal loss (90 degrees step input) test where an increase of 83% was calculated, and the worst performance was for the 25% signal loss (10 degrees step input) test where a decrease of 105% was calculated. The simulated annealing tuned FIS for this sensor was able to improve on the benchmark results for nineteen of the twenty-four tests where a

fault has occurred. The best performance was for the 10% SNR (50 degrees step input) test where an increase of 49% was calculated, and the worst performance was for the 100% signal loss (90 degrees step input) test where a decrease of 100% was calculated. Again of the two tuned FISs presented here the one that made the overall system most fault tolerant to the considered faults is the one tuned by the simulated annealing method. This FIS despite also not having been tuned using SNR data, was able to improve on the benchmark systems test results for all SNR tests at every level of noise considered. This shows that it is possible for the FIS to provide sensor recovery information on a fault previously unobserved by the FIS.

5.6.3. Roll Sensor

After the yaw channel sensor had been considered sections 5.5.4 and 5.5.5 considered faults occurring within the roll channel sensors.

The roll sensor recovery FISs were the first of the roll channel FISs to be tested. The ANFIS tuned FIS for this sensor was able to improve on the benchmark results for twenty-one of the twenty-four tests where a fault has occurred and all three of the fault free tests. The best performance was for the 25% signal loss (initial angle of 25 degrees) test where an increase of 88% was calculated, and the worst performance was for the intermittent total failure (initial angle of 25 degrees) test where a decrease of 66% was calculated. The results presented showed that the FIS was capable of improving the fault tolerance of the system, when compared to the Kalman filter enhanced control system. Even when considering faults that the FIS had not been tuned to handle (the SNR faults) it was still capable of generating RMSE values less than the benchmark results.

The simulated annealing tuned FIS for this sensor was able to improve on the benchmark results for twenty-two of the twenty-four tests where a fault has occurred and all three of the fault free tests. The best performance was for the 25% signal loss (initial angle of 5 degrees) test where an increase of 60% was calculated, and the worst performance was for the 100% signal loss (initial angle of 5 degrees) test where a decrease of 8% was calculated. Again the results presented showed that the FIS was capable of improving the fault tolerance of the system, when compared to the Kalman filter enhanced control system. The FIS also proved it was capable of handling the SNR faults for which it had not been tuned.

5.6.4. Roll Rate Sensor

The roll rate sensor recovery FISs were the final two FISs to be tested in this Chapter. The ANFIS tuned FIS for this sensor was able to improve on the benchmark results for seventeen of the twenty-four tests where a fault has occurred and all three of the fault free tests. The best performance was for the 1% SNR (initial angle of 25 degrees) test where an increase of 58% was calculated, and the worst performance was for the 100% signal loss (initial angle of 15 degrees) test where a decrease of 27% was calculated. The FIS produced RMSE values which showed that it was capable of improving the fault tolerance of the AUV.

The simulated annealing tuned FIS for this sensor was able to improve on the benchmark results for only nine of the twenty-four tests where a fault has occurred and one of the three fault free tests. The best performance was for the 10% SNR (initial angle of 5 degrees) test where an increase of 9% was calculated, and the worst performance was for the 100% signal loss (initial angle of 5 degrees) test where a decrease of 17% was calculated. The FISs results were similar to the Kalman filter enhanced control systems and so there would be no advantage to using this FIS instead of the Kalman filter in the AUV.

CHAPTER 6

ACTUATOR RECOVERY SYSTEM WITH ERROR SENSOR

6.1. INTRODUCTION

A fault tolerant fuzzy sensor recovery system was developed in Chapter 5 where it was clearly shown how this was an improvement on both the standard AUV ANFIS control system and the Kalman filter enhanced ANFIS control system. Attention now switches from the sensor faults to the task of dealing with actuator faults. The aim of this Chapter is to consider the actuator faults occurring within the canards controlling the yaw channel motion of the AUV as described in Chapter 3.

Within this Chapter fuzzy inference systems (FISs) will be developed which will allow the AUV to more effectively accommodate actuator faults. The set-up of the basic FISs and tuning of these FISs is explained. The FISs are tuned by the various methods described in Chapter 3, some of which have been used in Chapter 5 for tuning the sensor recovery FISs. The faults considered are those introduced and explained in Chapter 3. They are a loss of effectiveness (LOE) in the saturation and/or rate limiter blocks of the actuator that controls the upper canard. These three types of faults are considered to be representative of faults that could occur within an actuator of the AUV.

6.2. FAULT RECOVERY SYSTEM

The sensor recovery system developed was placed before the ANFIS controller within the control loop. This was due to the nature of the fault being to disrupt the information going to the ANFIS controller. The faults now being considered do not affect the information the ANFIS controller receives. As the ANFIS controller has been proved to be both an effective and robust controller for the considered AUV [Craven (1999)], there is no need to place any form of actuator fault recovery system before the controller. Instead for the faults currently being considered it is proposed to

place a component that will alter the strength of the signal being produced by the ANFIS controller.

To attempt to cope with the failures it is proposed to situate a FIS between the ANFIS controller and the AUV dynamics. This will multiply the control signal to the actuators, as shown in Figure 6.1. Multiplication was chosen above any other method of altering the control signal due to its identity function affect on a zero control signal. The placing of this multiplication FIS in the control loop will give the overall control system, and hence the AUV, a level of fault tolerance to the failures that are being considered in this thesis.

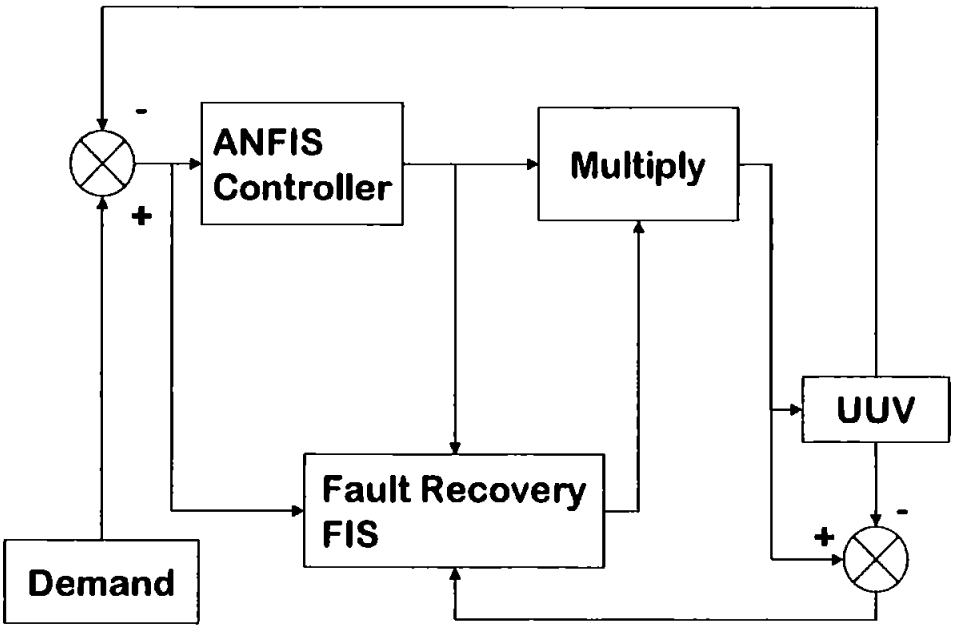


Figure 6.1 The Actuator Fault Tolerant System

As can clearly be seen from Figure 6.1 the FIS being developed here has three inputs and only one output. The three inputs being used are the demand being placed on the AUV, the control signal the ANFIS controller sends to achieve this demand and the error between the demanded and actual position of the damaged actuator. The output of the FIS will be a fuzzy simpleton derived from the inputs using a fuzzy rule base. Figure 6.2 illustrates the input output structure of the FISs being used.

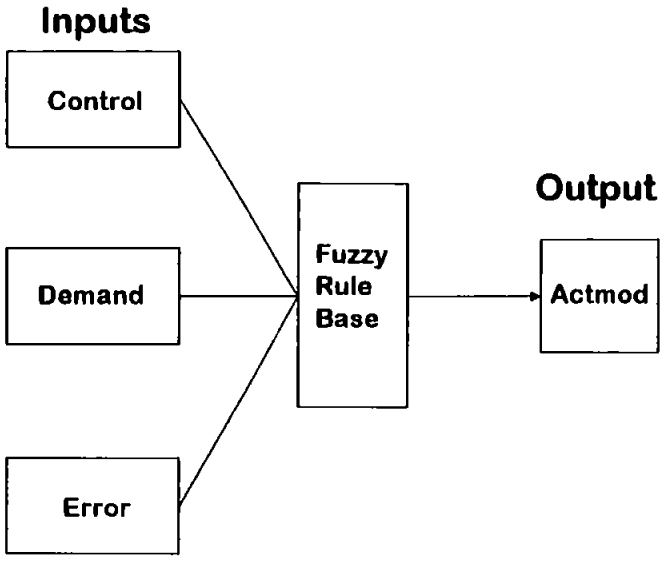


Figure 6.2 Basic Input Output Structure

The faults being considered are saturation and/or rate limiter LOE of 25%, 50%, 75%, and 100% of the actuator controlling the upper canard of the AUV. The full description of these faults and how they are implemented within the AUV is given in Chapter 3 of this thesis.

The tuned FISs are to be compared using a statistical method and the rise times of the AUV.

The statistical method used to evaluate the performance of the models was the root mean squared error (RMSE). This will give the average difference between the fault free ideal control system and the actuator recovery FISs performance over the considered path. The equation used to calculate this is shown in Equation (6.1).

$$E = \sqrt{\frac{\sum_{i=0}^{i=n}(AUV_i - IDEAL_i)^2}{n}} \tag{6.1}$$

Where E is the error value, n is the number of points measured, AUV_i is the information received from the vehicle at point i, and $IDEAL_i$ is the information recorded when the vehicle followed the same path in a fault free situation also at point i.

The rise time is defined as the time taken for the AUV to move from 5% to 95% of the demanded final yaw angle.

6.3. FUZZY TUNING

There are two basic starting actuator recovery FISs which have been considered for this study, the first of which is the identity FIS, and the second is a heuristic FIS. The identity FIS was chosen as before tuning it will not affect the system and hence any tuning which leads to an improvement will lead to a more fault tolerant system.

6.3.1. Identity FIS

The identity FIS was designed using the input-output structure shown in Figure 6.2. The first input shown there is that of the control signal from the ANFIS controller. It is defined as having a maximum value of 25.2 degrees and minimum value of –25.2 degrees. There are three membership functions for this input which are all generalised bell curve membership functions and are shown in Figure 6.3.

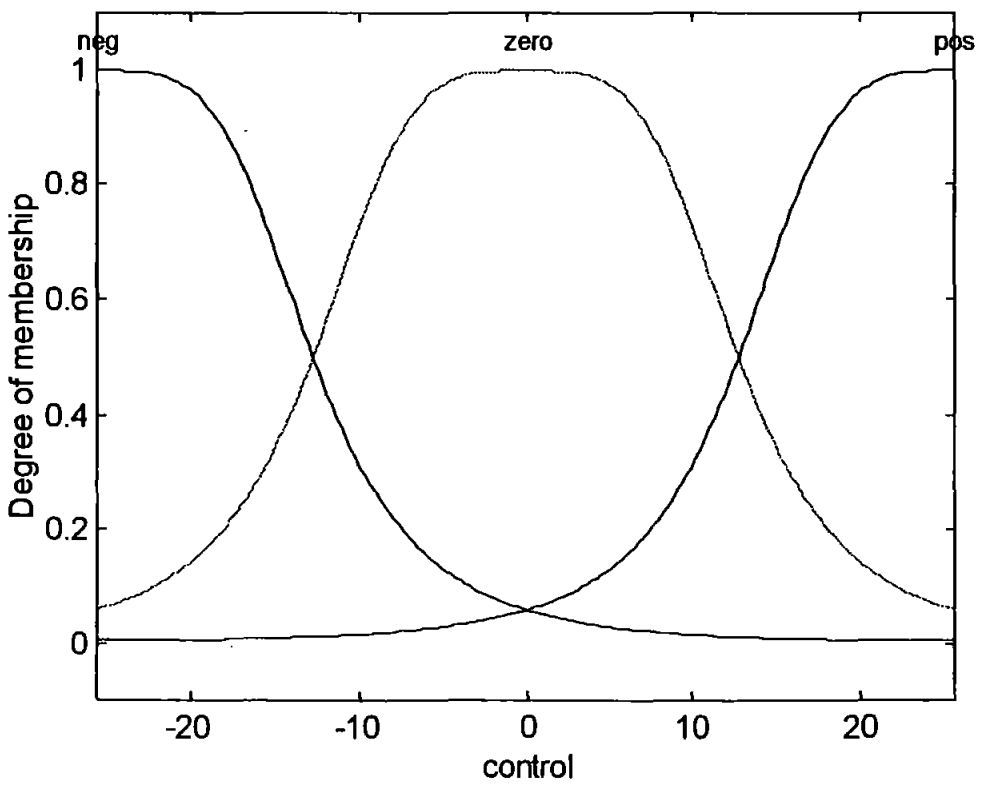


Figure 6.3 Membership Functions for Control Input

The second input shown there is that of the demand signal which is the signal being sent to the ANFIS controller. It is defined as having a maximum value of 30 degrees and minimum value of –30 degrees, although it is possible to demand a bigger angle than this it was not necessary to consider such angles here. These larger angles are not being considered as the ANFIS controller limits all of its input signals to this level. There are three membership functions for this input which are all generalised bell curve membership functions and are shown in Figure 6.4.

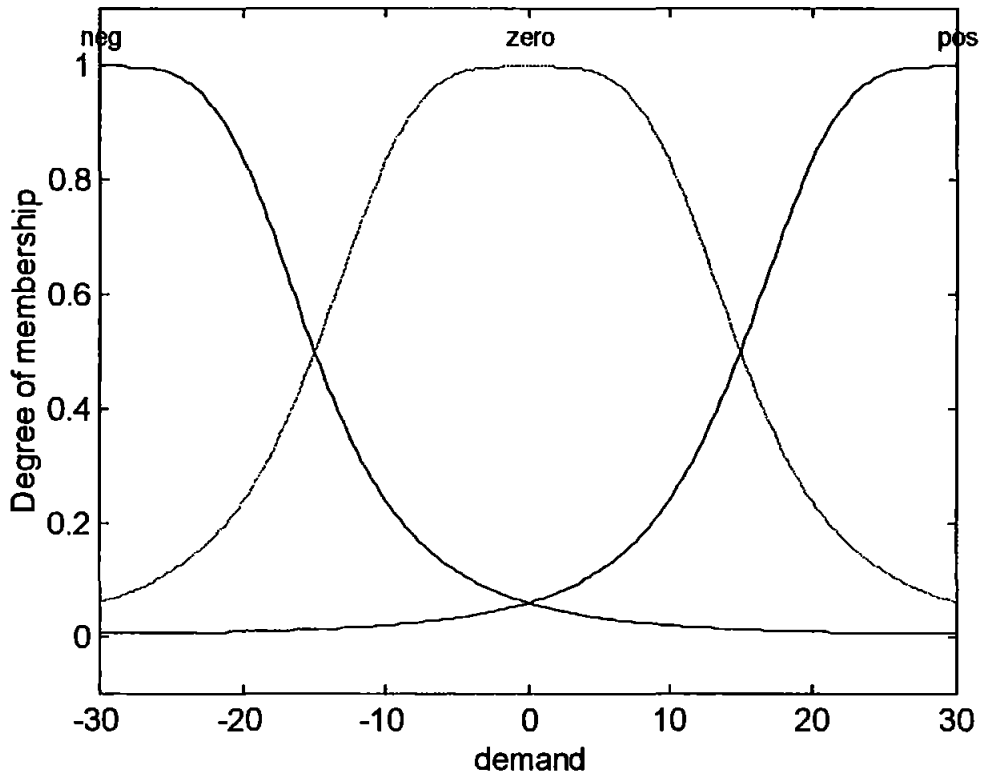


Figure 6.4 Membership Functions for Demand Input

The third input shown there is that of the error signal which is defined as the modular difference between the position the actuator should be in and the actual position of the actuator. It is defined as having a maximum value of 25.2 degrees and minimum value of 0 degrees. There are three membership functions for this input which are all generalised bell curve membership functions and are shown in Figure 6.5.

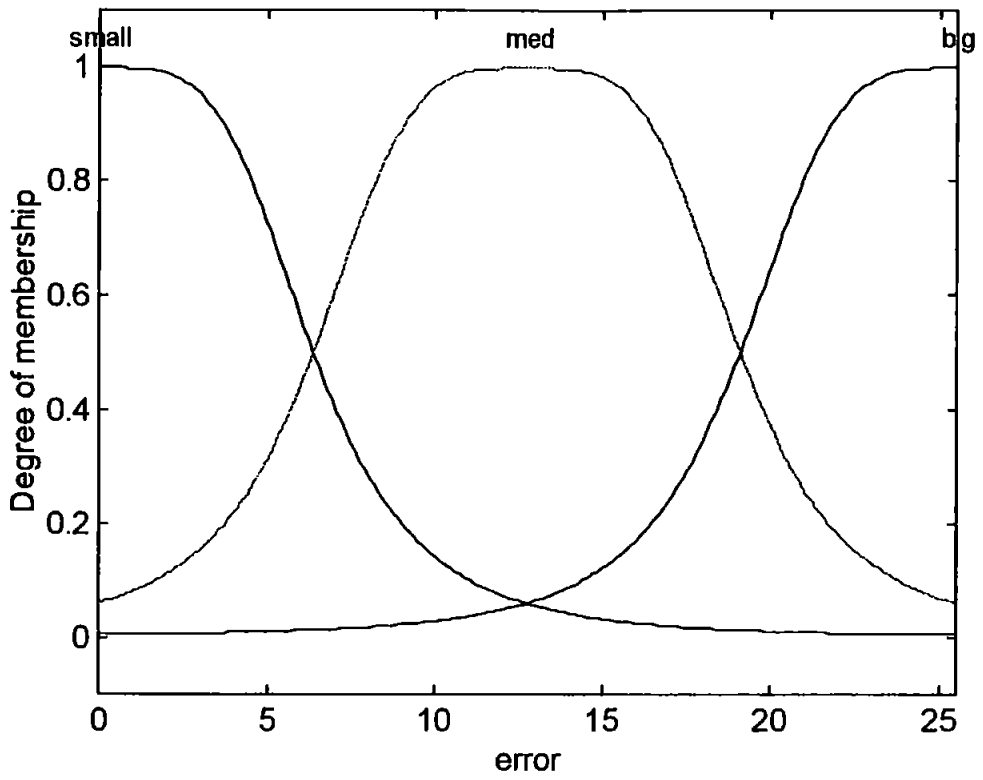


Figure 6.5 Membership Functions for Error Input

The output for the identity FIS is that of a fuzzy singleton with a value of one. This combined with the input membership functions described leads to the fuzzy rule base shown in Equation (6.2).

If δ is Neg and ψ_ε is Neg and ηe is Small then $\beta = 1$

If δ is Neg and ψ_ε is Neg and ηe is Medium then $\beta = 1$

If δ is Neg and ψ_ε is Neg and ηe is Big then $\beta = 1$

If δ is Neg and ψ_ε is Zero and ηe is Small then $\beta = 1$

If δ is Neg and ψ_ε is Zero and ηe is Medium then $\beta = 1$

If δ is Neg and ψ_ε is Zero and ηe is Big then $\beta = 1$

If δ is Neg and ψ_ε is Pos and ηe is Small then $\beta = 1$

If δ is Neg and ψ_ε is Pos and ηe is Medium then $\beta = 1$

If δ is Neg and ψ_ε is Pos and ηe is Big then $\beta = 1$

If δ is Zero and ψ_ε is Neg and ηe is Small then $\beta = 1$

If δ is Zero and ψ_ε is Neg and ηe is Medium then $\beta = 1$

If δ is Zero and ψ_ε is Neg and ηe is Big then $\beta = 1$
If δ is Zero and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Zero and ηe is Medium then $\beta = 1$ (6.2)
If δ is Zero and ψ_ε is Zero and ηe is Big then $\beta = 1$
If δ is Zero and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Pos and ηe is Medium then $\beta = 1$
If δ is Zero and ψ_ε is Pos and ηe is Big then $\beta = 1$
If δ is Pos and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Neg and ηe is Medium then $\beta = 1$
If δ is Pos and ψ_ε is Neg and ηe is Big then $\beta = 1$
If δ is Pos and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Zero and ηe is Medium then $\beta = 1$.
If δ is Pos and ψ_ε is Zero and ηe is Big then $\beta = 1$
If δ is Pos and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Pos and ηe is Medium then $\beta = 1$
If δ is Pos and ψ_ε is Pos and ηe is Big then $\beta = 1$

Where δ is the control signal from the ANFIS controller, ψ_ε is the demanded yaw angle ηe is the error in the actuator position, β is the value the control signal is to be multiplied by, Neg is negative and Pos is positive.

By considering Equation (6.2) it is clear that for whatever values any of the three input functions take the output of the FIS will be simply a value of one. The output value of the FIS will multiply the control signal of the ANFIS controller and hence for this FIS the ANFIS controller's signal is unaffected. This is why it is the identity FIS for this work.

6.3.2. Heuristic FIS

The second was a heuristic FIS and was designed by taking a simple approach to the problem, the bigger the error, the larger the value the control signal must be multiplied by in order to compensate for the fault.

For the heuristic FIS the same input-output structure shown in Figure 6.2 was used. Also identical input membership functions to those of the identity FIS were used. The main difference between the heuristic FIS and the identity FIS is in the output function. Fuzzy singletons are again used as the output of the FIS. A value of 1 is used when the error is considered to be small as for a small error the signal may not need to be altered by a huge amount. A value of 1.05 is used when the error value is considered to be medium as this is in the middle of the small and large values. A value of 1.1 is used for the large error values as this will increase the signal by 10%. This combined with the input membership functions describe leads to the fuzzy rule base shown in Equation (6.3).

If δ is Neg and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Neg and ηe is Medium then $\beta = 1.05$
If δ is Neg and ψ_ε is Neg and ηe is Big then $\beta = 1.1$
If δ is Neg and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Zero and ηe is Medium then $\beta = 1.05$
If δ is Neg and ψ_ε is Zero and ηe is Big then $\beta = 1.1$
If δ is Neg and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Pos and ηe is Medium then $\beta = 1.05$
If δ is Neg and ψ_ε is Pos and ηe is Big then $\beta = 1.1$
If δ is Zero and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Neg and ηe is Medium then $\beta = 1.05$
If δ is Zero and ψ_ε is Neg and ηe is Big then $\beta = 1.1$
If δ is Zero and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Zero and ηe is Medium then $\beta = 1.05$ (6.3)
If δ is Zero and ψ_ε is Zero and ηe is Big then $\beta = 1.1$
If δ is Zero and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Pos and ηe is Medium then $\beta = 1.05$
If δ is Zero and ψ_ε is Pos and ηe is Big then $\beta = 1.1$
If δ is Pos and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Neg and ηe is Medium then $\beta = 1.05$
If δ is Pos and ψ_ε is Neg and ηe is Big then $\beta = 1.1$

If δ is Pos and ψ_ε is Zero and ηe is Small then $\beta = 1$

If δ is Pos and ψ_ε is Zero and ηe is Medium then $\beta = 1.05$.

If δ is Pos and ψ_ε is Zero and ηe is Big then $\beta = 1.1$

If δ is Pos and ψ_ε is Pos and ηe is Small then $\beta = 1$

If δ is Pos and ψ_ε is Pos and ηe is Medium then $\beta = 1.05$

If δ is Pos and ψ_ε is Pos and ηe is Big then $\beta = 1.1$

Where δ is the control signal from the ANFIS controller, ψ_ε is the demanded yaw angle ηe is the error in the actuator position, β is the value the control signal is to be multiplied by, Neg is negative and Pos is positive.

It is now clear to see from Equation (6.3) that as the size of the error increase so does the output of the FIS. The result of an increase in the value of the output of the FIS is to multiply the ANFIS controller's output signal. This will lead to an increase in the demand being placed on the actuators and will make the AUV recover partially from the fault being considered.

To tune these FISs a suitable tuning path for the AUV was required. The key features of the tuning path are that it must excite as many levels as possible of each input function being considered and that it is a realistic path for the AUV to follow.
The chosen path was a series of step input demands, ranging form 5 degrees to 35 degrees. The AUV was given adequate time between each demand to achieve and maintain the next yaw angle. The total simulation lasted for a period of 400 seconds and can be seen in Figure 6.6. During the tuning this path was presented to this system four times. The first time there was no fault within the system, this was to ensure the fault recovery FIS would not inhibit the operation of the AUV during a fault free scenario. The second time a LOE of 33.3% was implemented in both the rate limiter and saturation blocks. The third time a LOE of 66.6% was implemented in both the rate limiter and saturation blocks. For the forth and final time the path of input demands were presented to the system when a LOE of 100% was implemented in both the rate limiter and saturation blocks. This produced a total tuning simulation time of 1600 seconds for every epoch. The tuning path also considered four levels of LOE for seven different magnitudes of step input yaw demands.

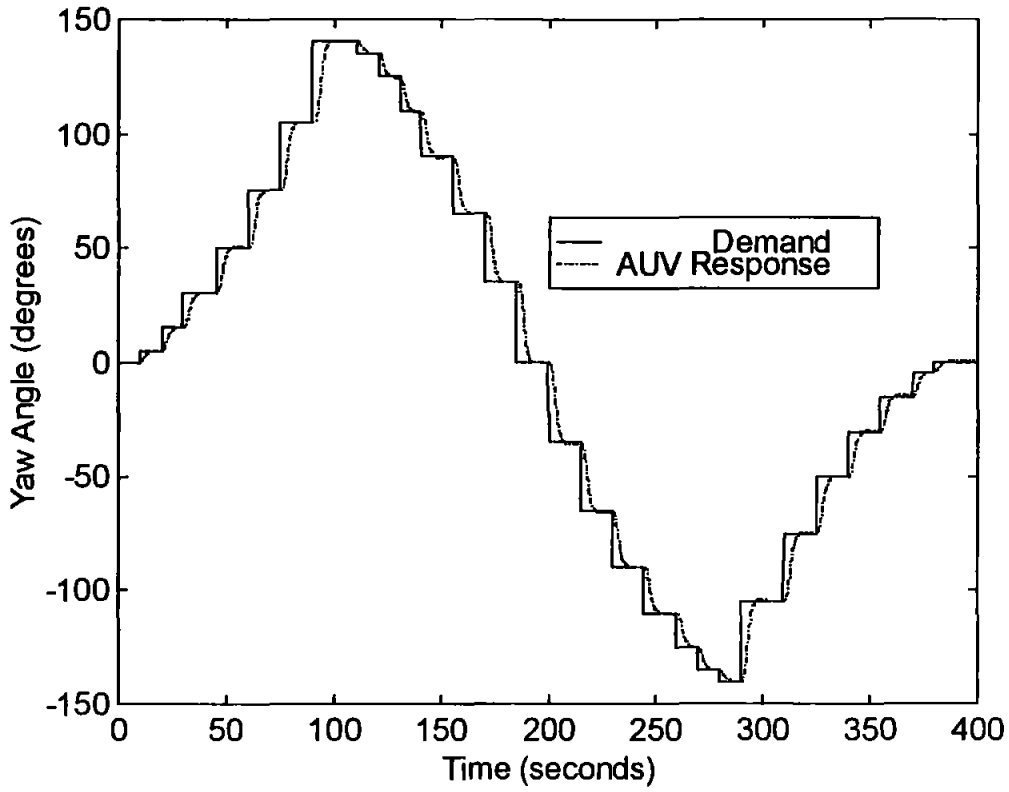


Figure 6.6 AUV Tuning Path for Actuator Faults

Both FISs to be tuned within this Chapter have now been defined, as has the tuning path to be used to tune them.

6.3.3. ANFIS

Unfortunately it was not possible to use the ANFIS [Jang (1991)] tuning method for this work. This is due to its need of both input and output training data. Unlike the sensor faults, where the sensor information required to be produced by the sensor recovery FIS was already known, for this section of the work the desired control signal for the AUV is an unknown quantity. Each type and level of fault alters the dynamics of the control surfaces being used for this work. Therefore each fault would require a new optimal control FIS to be designed to maintain performance. This requires a huge amount of work to calculate and would require many different control systems to be carried in the AUV at all times.

6.3.4. Simulated Annealing

As the ANFIS approach cannot be used, for reasons discussed in section 6.3.3, the first approach used to tune the basic FISs is simulated annealing [Kirkpatrick, et *al* (1983)]. This is the same method used in Chapter 5 to tune the sensor recovery FISs, and has been explained in detail within Chapter 3.

Both initial FISs were tuned using this method. Two approaches to the tuning have been considered, the first being where the simulated annealing program can alter all of the values and the second where the output for small error values are fixed at a value of 1. This created four separate cases of FIS tuning, two different starting positions and two different tuning parameters. For each case the FIS was tuned for five runs of 100 epochs. This was repeated three times to ensure no anomalous results were obtained due to the probabilistic nature of the simulated annealing program. A mean of these three FISs was then calculated to produce an average FIS, it was hoped this would create a more fault tolerant FIS, unfortunately this was not the case.

After training the FISs were tested over all three step sizes (10, 20, and 30 degrees), for all three types of faults (saturation, rate limiter and both), for all four levels of LOE (25%, 50%, 75%, and 100%). The FIS which was the most fault tolerant on all of these tests will be presented in the results section.

6.3.3. Tabu Search

The second and alternative approach considered for tuning the two basic FISs is that of the tabu search [Denna et *al* (1999)] method. This method has been explained in detail within Chapter 3.

Both initial FISs were tuned using this method. Two approaches to the tuning have again been considered the first being where the tabu search can alter all of the values and the second where the output for small error values are fixed at a value of 1. This again created four separate cases of FIS tuning, two different starting positions and two different tuning parameters. For each case the FIS was tuned for five runs of 100 epochs. As this method does not use the statistical approach employed by simulated annealing there was no reason to repeat the process.

After training the FISs were tested over all three step sizes (10, 20, and 30 degrees), for all three types of faults (saturation, rate limiter and both), for all four levels of

LOE (25%, 50%, 75%, and 100%). The FIS which performed most fault tolerant on all of these tests will be presented in the results section.

6.4. RESULTS

The results of the most fault tolerant tuned FIS for both the simulated annealing and tabu search algorithms are now presented. The results (RMSEs and rise times) for all the levels of all the faults for all levels of demanded yaw angle inputs being considered are shown. Figures are presented showing the AUVs response to each type of fault. Also shown are the canard responses to selected types of fault, the complete set of canard responses can be found in Appendix I.

6.4.1. Simulated Annealing Tuned FIS

First let the simulated annealing case be considered, the method described leads to the creation of sixty-four FISs. With each FIS being tested on three step input sizes (10, 20 and 30 degrees) for three kinds of faults (saturation and/or rate limiter blocks) for four levels of LOE (25%, 50%, 75% and 100%), this would lead to a total of thirty-six tests for each FIS and hence a total of two thousand three hundred and four individual tests. Clearly this would be too many to tests to be included in this thesis, hence only the test results for the most fault tolerant FIS tuned using simulated annealing will be displayed herein.

After all of the tests had been performed and the results had been compared and correlated the most fault tolerant FIS tuned using the simulated annealing method was the FIS tuned for 500 epochs from the heuristic FIS when the output for the occurrence of the small error input was fixed at a value of one. SA FIS as it shall be referred to from this point.

The input membership functions for the SA FIS were similar to those of the heuristic FIS, despite the algorithm having the ability to change them. The minimum and maximum values of each function were not permitted to be tuned. The lack of change is due to the structure of the tuning path used. It considered many different step sizes and LOEs in an unbiased manner, which covered the complete input space of the control, demand and error inputs. The tuning path was chosen so that the tuned FIS

would be able to handle all levels of LOE for all step inputs. The three input membership functions for the SA FIS are shown in Figures 6.7, 6.8 and 6.9.

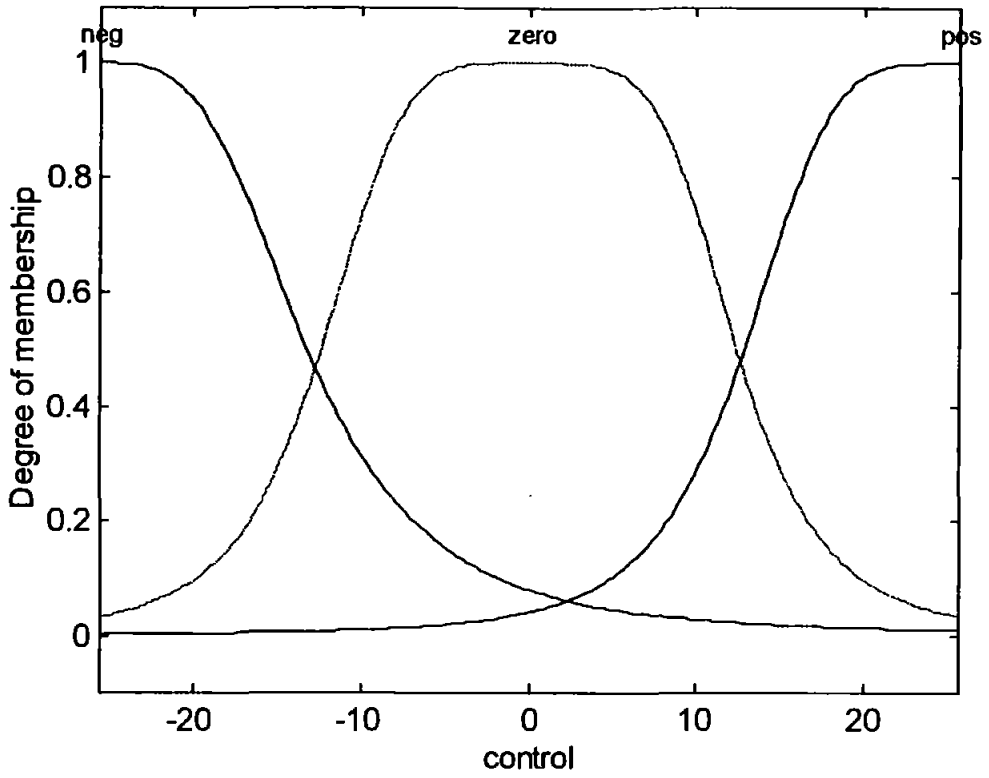


Figure 6.7 Membership Functions for Control Input of SA FIS

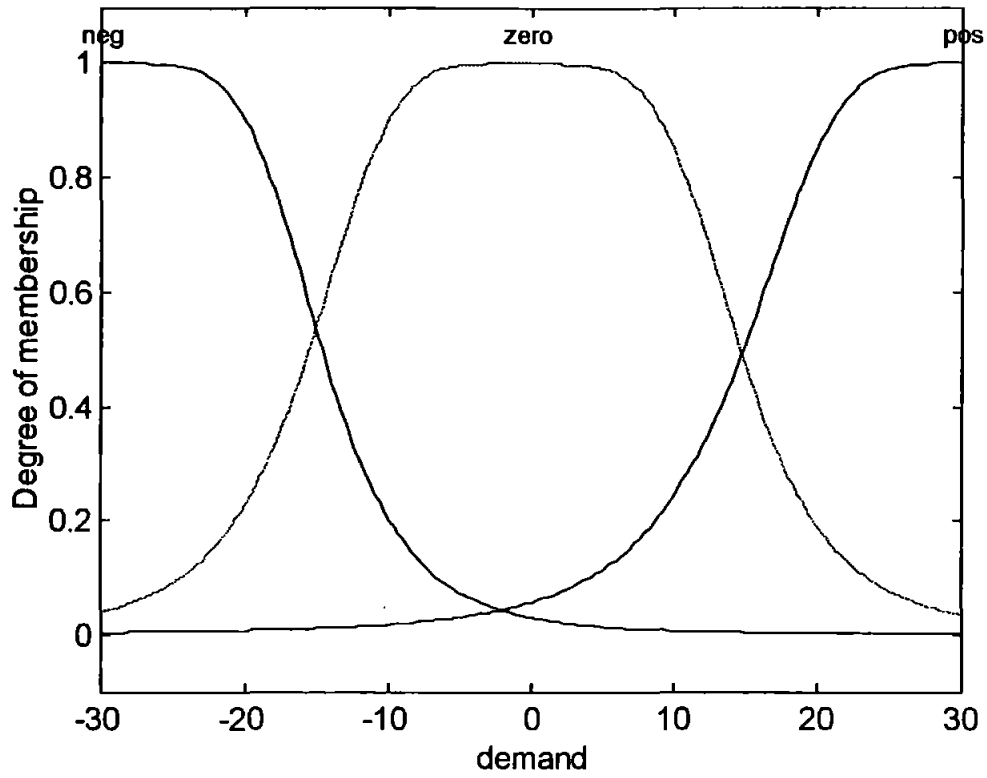


Figure 6.8 Membership Functions for Demand Input of SA FIS

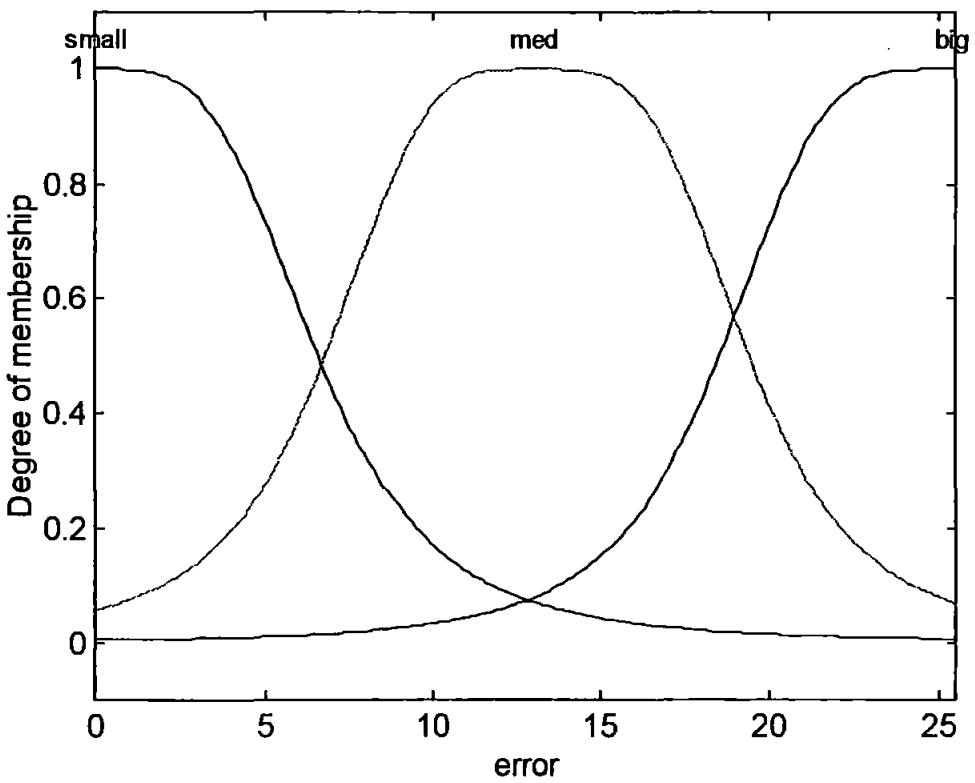


Figure 6.9 Membership Functions for Error Input of SA FIS

It is in the output fuzzy singletons that the simulated annealing tuning had its greatest effect. For the SA FIS eighteen of the twenty-seven singletons could be tuned. The fuzzy singletons, when combined with the input membership functions, lead to the tuned fuzzy rule base shown in Equation (6.4).

If δ is Neg and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Neg and ηe is Medium then $\beta = 0.87$
If δ is Neg and ψ_ε is Neg and ηe is Big then $\beta = 2.19$
If δ is Neg and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Zero and ηe is Medium then $\beta = 1.15$
If δ is Neg and ψ_ε is Zero and ηe is Big then $\beta = 1.86$
If δ is Neg and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Neg and ψ_ε is Pos and ηe is Medium then $\beta = 0.83$
If δ is Neg and ψ_ε is Pos and ηe is Big then $\beta = 1.08$
If δ is Zero and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Neg and ηe is Medium then $\beta = 1.29$
If δ is Zero and ψ_ε is Neg and ηe is Big then $\beta = 1.85$
If δ is Zero and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Zero and ηe is Medium then $\beta = 1.67$ (6.4)
If δ is Zero and ψ_ε is Zero and ηe is Big then $\beta = 1.57$
If δ is Zero and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Zero and ψ_ε is Pos and ηe is Medium then $\beta = 0.98$
If δ is Zero and ψ_ε is Pos and ηe is Big then $\beta = 1.01$
If δ is Pos and ψ_ε is Neg and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Neg and ηe is Medium then $\beta = 0.97$
If δ is Pos and ψ_ε is Neg and ηe is Big then $\beta = 0.88$
If δ is Pos and ψ_ε is Zero and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Zero and ηe is Medium then $\beta = 0.77$
If δ is Pos and ψ_ε is Zero and ηe is Big then $\beta = 1.59$
If δ is Pos and ψ_ε is Pos and ηe is Small then $\beta = 1$
If δ is Pos and ψ_ε is Pos and ηe is Medium then $\beta = 0.99$
If δ is Pos and ψ_ε is Pos and ηe is Big then $\beta = 1.96$

Where δ is the control signal from the ANFIS controller, ψ_s is the demanded yaw angle ηe is the error in the actuator position and β is the value the control signal is to be multiplied by.

The β values in Equation (6.4) have been tuned by the simulated annealing algorithm from those in Equation (6.3). The values before and after tuning can be seen in Figure 6.10.

It can be seen from Figure 6.10 that no tuning took place to the nine fuzzy rules where the error is considered to be small. The other eighteen values have all been changed by the tuning algorithm. The tuning process did increase eleven of these values, which means that the FIS will increase the ANFIS controller's signal for these cases. The tuning process reduced the values for the other seven rules which means that the FIS will decrease the ANFIS controller's signal for these cases

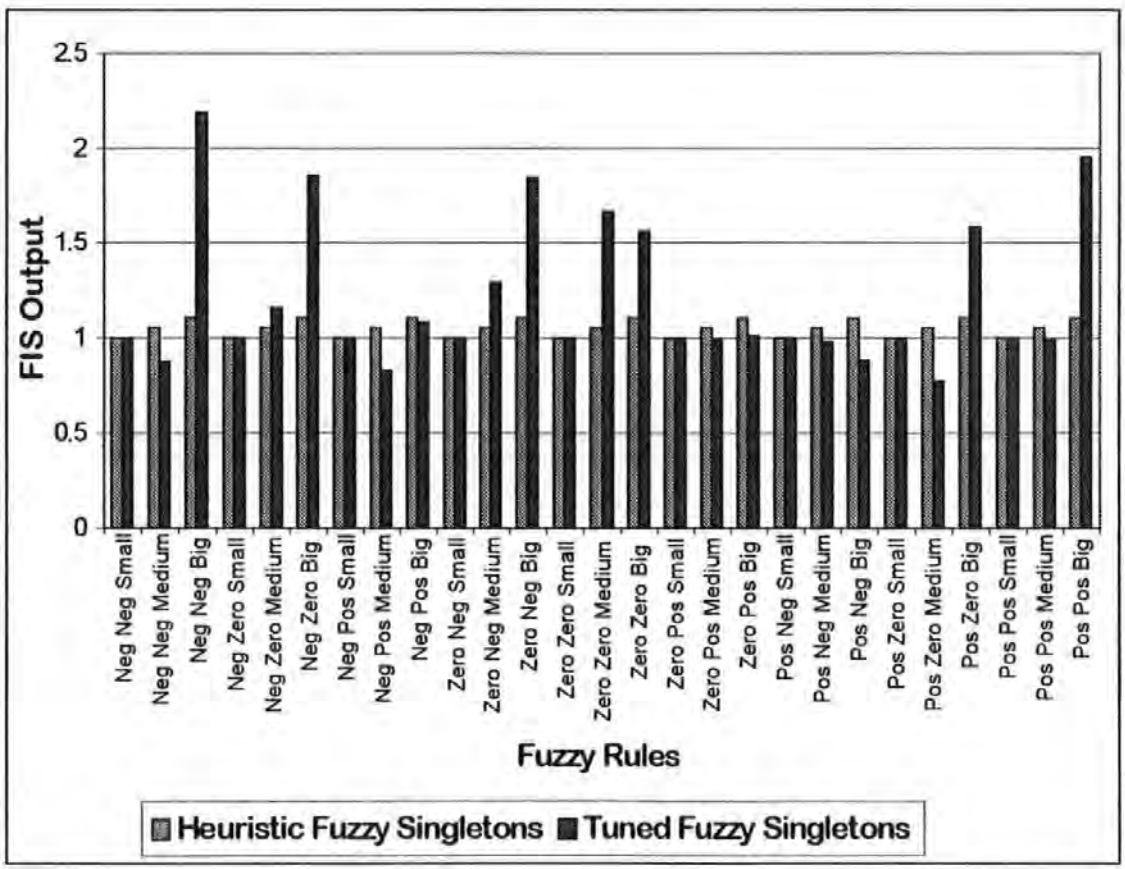


Figure 6.10 Values for The Fuzzy Singletons Before and After Tuning by The Simulated Annealing Algorithm

6.4.2. Tabu Search Tuned FIS

Now let the tabu search case be considered, the method described leads to the creation of fifty-six FISs. With again each FIS being tested on three step input sizes for three kinds of faults for four levels of LOE, this leads to a total of thirty-six tests for each FIS and hence a total of two thousand and sixteen individual tests. As was the case before this would be too many to tests to be included in this thesis, hence only the test results for the most fault tolerant FIS tuned using tabu search will be displayed herein. After all of the tests had been performed and the results had been compared and correlated the most fault tolerant FIS tuned using the tabu search method was the FIS tuned for 500 epochs from the heuristic FIS. TABU FIS as it shall be referred to from this point.

The input membership functions for the TABU FIS were similar to those of the heuristic FIS, despite the algorithm having the ability to change them. The reasons for this are the same as those stated when using the simulated annealing approach. The three input membership functions for the TABU FIS are shown in Figures 6.11, 6.12 and 6.13.

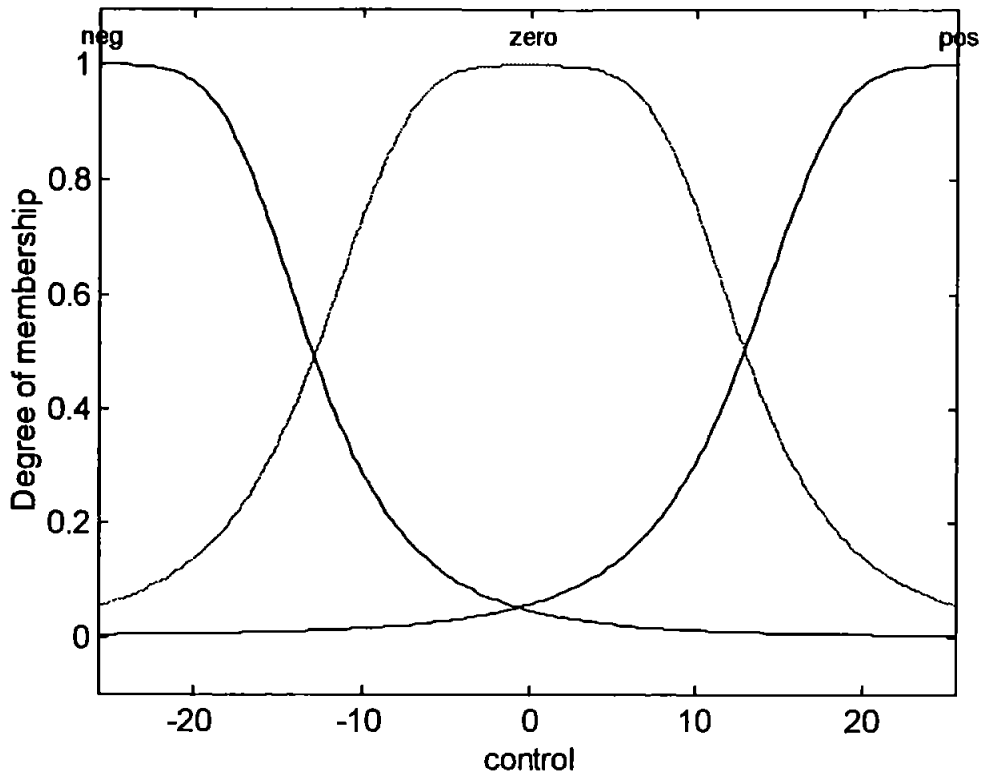


Figure 6.11 Membership Functions for Control Input of TABU FIS

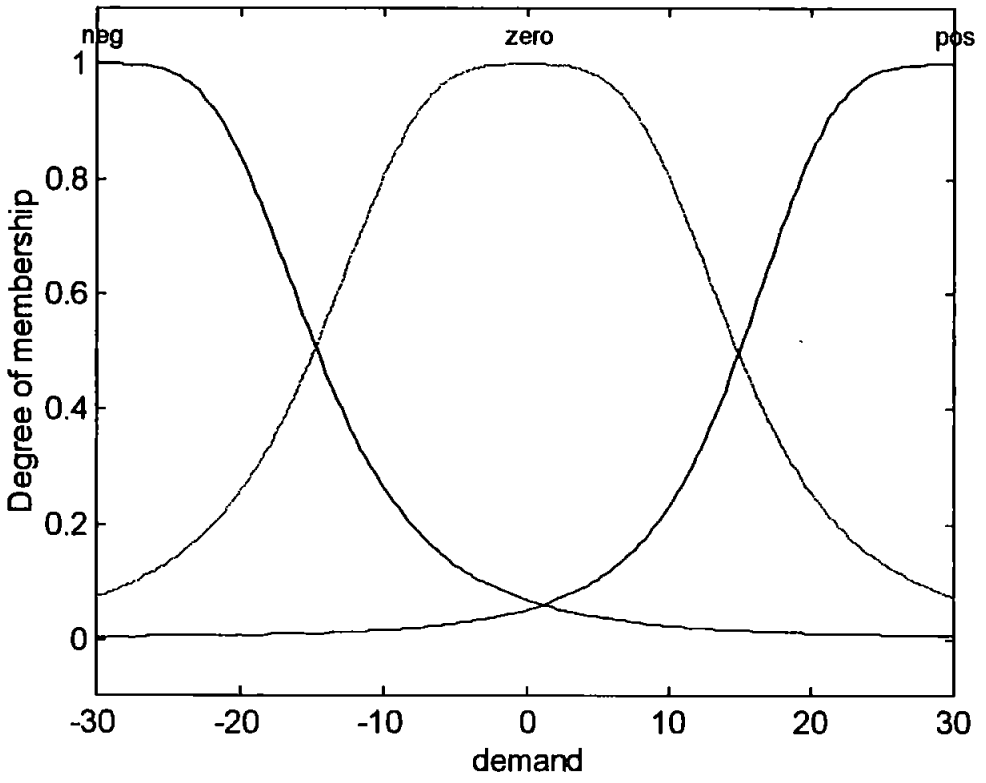


Figure 6.12 Membership Functions for Demand Input of TABU FIS

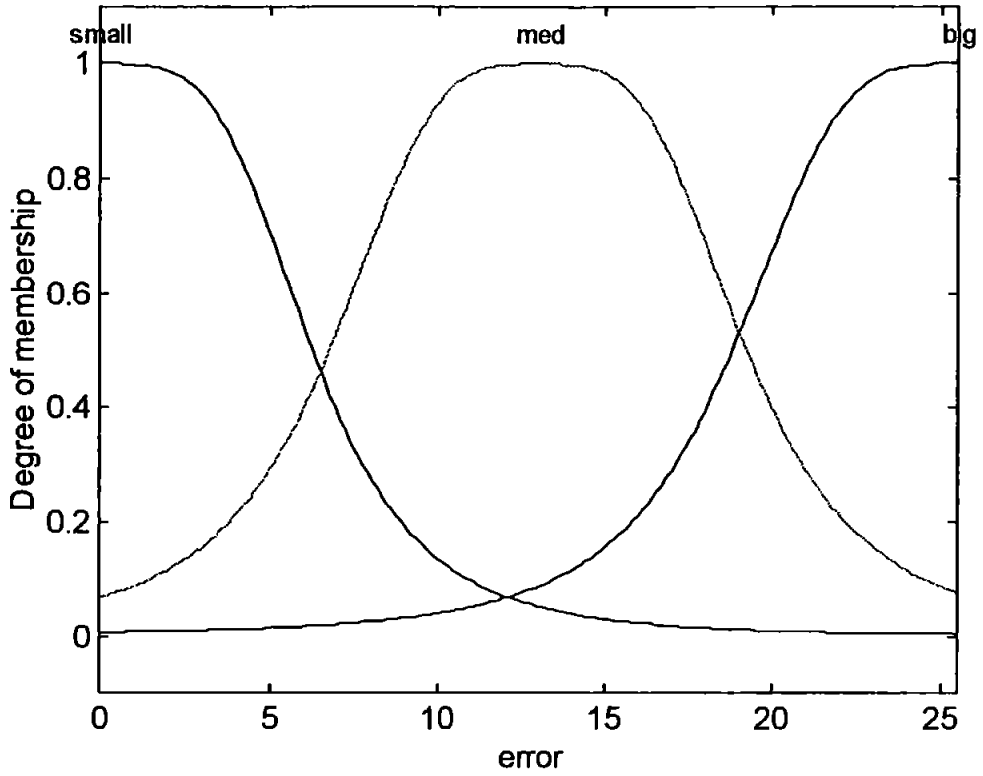


Figure 6.13 Membership Functions for Error Input of TABU FIS

It is in the output fuzzy singletons that the tabu search tuning had its greatest effect. For the TABU FIS all of the twenty-seven singletons could be tuned. The fuzzy singletons, when combined with the input membership functions, lead to the tuned fuzzy rule base shown in Equation (6.5).

If δ is Neg and ψ_ε is Neg and ηe is Small then $\beta = 1.00$

If δ is Neg and ψ_ε is Neg and ηe is Medium then $\beta = 1.14$

If δ is Neg and ψ_ε is Neg and ηe is Big then $\beta = 1.11$

If δ is Neg and ψ_ε is Zero and ηe is Small then $\beta = 1.05$

If δ is Neg and ψ_ε is Zero and ηe is Medium then $\beta = 1.02$

If δ is Neg and ψ_ε is Zero and ηe is Big then $\beta = 1.24$

If δ is Neg and ψ_ε is Pos and ηe is Small then $\beta = 1.07$

If δ is Neg and ψ_ε is Pos and ηe is Medium then $\beta = 0.96$

If δ is Neg and ψ_ε is Pos and ηe is Big then $\beta = 1.30$

If δ is Zero and ψ_ε is Neg and ηe is Small then $\beta = 0.68$

If δ is Zero and ψ_ε is Neg and ηe is Medium then $\beta = 0.78$

If δ is Zero and ψ_ε is Neg and ηe is Big then $\beta = 1.19$

If δ is Zero and ψ_ε is Zero and ηe is Small then $\beta = 1.10$

If δ is Zero and ψ_ε is Zero and ηe is Medium then $\beta = 1.43$ (6.5)

If δ is Zero and ψ_ε is Zero and ηe is Big then $\beta = 1.27$

If δ is Zero and ψ_ε is Pos and ηe is Small then $\beta = 0.98$

If δ is Zero and ψ_ε is Pos and ηe is Medium then $\beta = 1.1$

If δ is Zero and ψ_ε is Pos and ηe is Big then $\beta = 1.17$

If δ is Pos and ψ_ε is Neg and ηe is Small then $\beta = 2.35$

If δ is Pos and ψ_ε is Neg and ηe is Medium then $\beta = 2.61$

If δ is Pos and ψ_ε is Neg and ηe is Big then $\beta = 2.49$

If δ is Pos and ψ_ε is Zero and ηe is Small then $\beta = 1.03$

If δ is Pos and ψ_ε is Zero and ηe is Medium then $\beta = 1.11$

If δ is Pos and ψ_ε is Zero and ηe is Big then $\beta = 1.07$

If δ is Pos and ψ_ε is Pos and ηe is Small then $\beta = 0.98$

If δ is Pos and ψ_ε is Pos and ηe is Medium then $\beta = 1.02$

If δ is Pos and ψ_ε is Pos and ηe is Big then $\beta = 1.28$

Where δ is the control signal from the ANFIS controller, ψ_ε is the demanded yaw angle ηe is the error in the actuator position and β is the value the control signal is to be multiplied by.

The β values in Equation (6.5) have been tuned by the tabu search algorithm from those in Equation (6.3). The values before and after tuning can be seen in Figure 6.14. It can be seen from Figure 6.14 that all values have been tuned by the algorithm. The tuning process did increase seventeen of the values, which means that the FIS will increase the ANFIS controller's signal for these cases. The tuning process reduced the values for eight rules which means that the FIS will decrease the ANFIS controller's signal for these cases. The tuning process also left two of the values unchanged, these values were changed during the tuning.

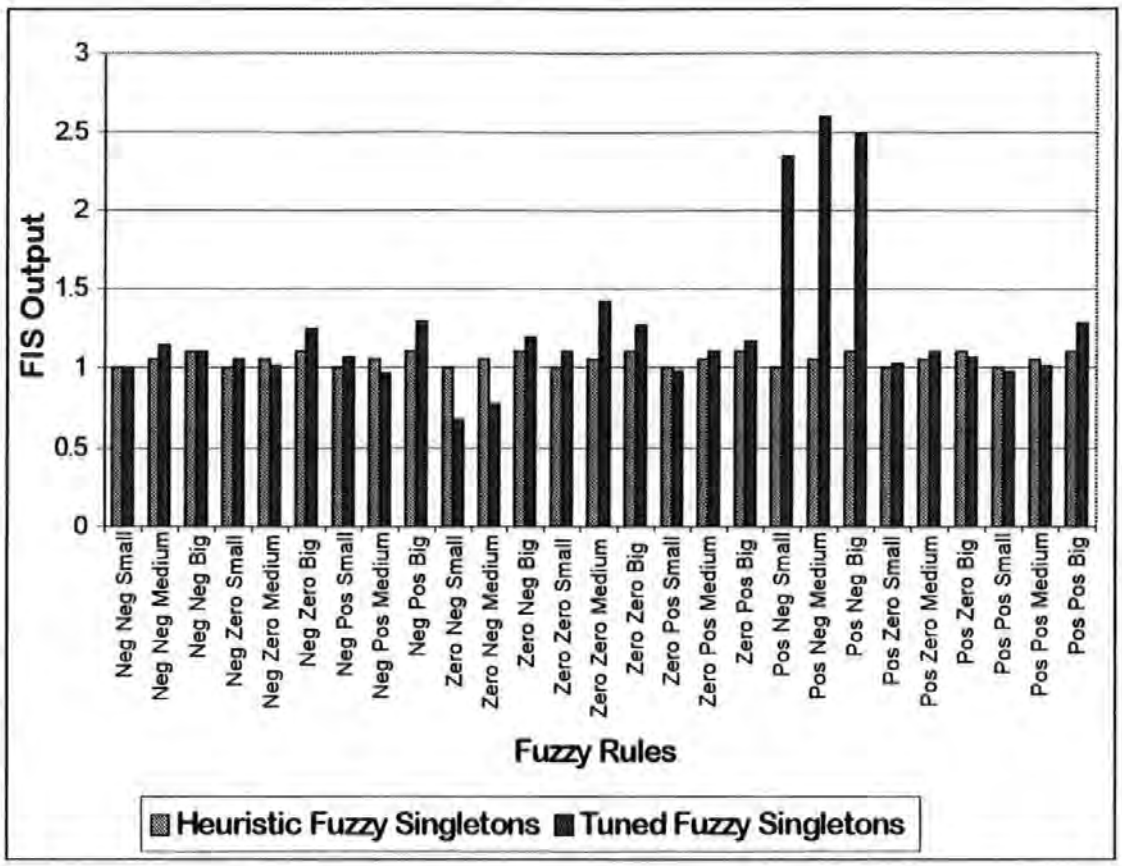


Figure 6.14 Values for The Fuzzy Singletons Before and After Tuning by The Tabu Search Algorithm

The results of placing both of these FISs in the control loop will now be displayed in three sections.

6.4.3. Yaw Step Inputs of 10 Degrees

The first section will look at LOEs for the 10 degrees yaw step input demands set of tests. The three types of faults defined in Chapter 3 will all be considered. The FISs have been compared using the RMSEs and rise times as defined previously.

The complete set of results (RMSEs and rise times) for both of the FISs described, along with the benchmark PD controller, are shown in Tables 6.1 and 6.2. At this point it is important to note that the unaffected ANFIS controlled system has a rise time of 4.5 seconds and a RMSE of zero degrees (as to be expected).

Table 6.1 The 10 Degrees Step Input RMSEs.

	RMSEs (degrees)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	0.227	0.227	0.131	1.183	0.172	0.171	0.358	1.183	0.172	0.171	0.358	1.183
Simulated Annealing	0.028	0.028	0.072	0.782	0.053	0.142	0.294	0.782	0.053	0.142	0.294	0.782
Tabu Search	0.072	0.072	0.053	0.820	0.049	0.129	0.290	0.820	0.049	0.129	0.290	0.820

Table 6.2 The 10 Degrees Step Input Rise Times.

	Rise Times (seconds)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	5.4	5.4	5.6	8.1	5.5	5.6	5.8	8.1	5.5	5.6	5.8	8.1
Simulated Annealing	4.5	4.5	4.6	6.4	4.5	4.6	4.6	6.4	4.5	4.6	4.6	6.4
Tabu Search	4.3	4.3	4.5	6.4	4.4	4.5	4.5	6.4	4.4	4.5	4.5	6.4

It is clear from these results that both fault tolerant FISs are an improvement on the PD controller. With respect to both the RMSE and rise times methods of evaluation, the FISs have smaller RMSE values and rise times for every test performed.

(a) Saturation Block Faults

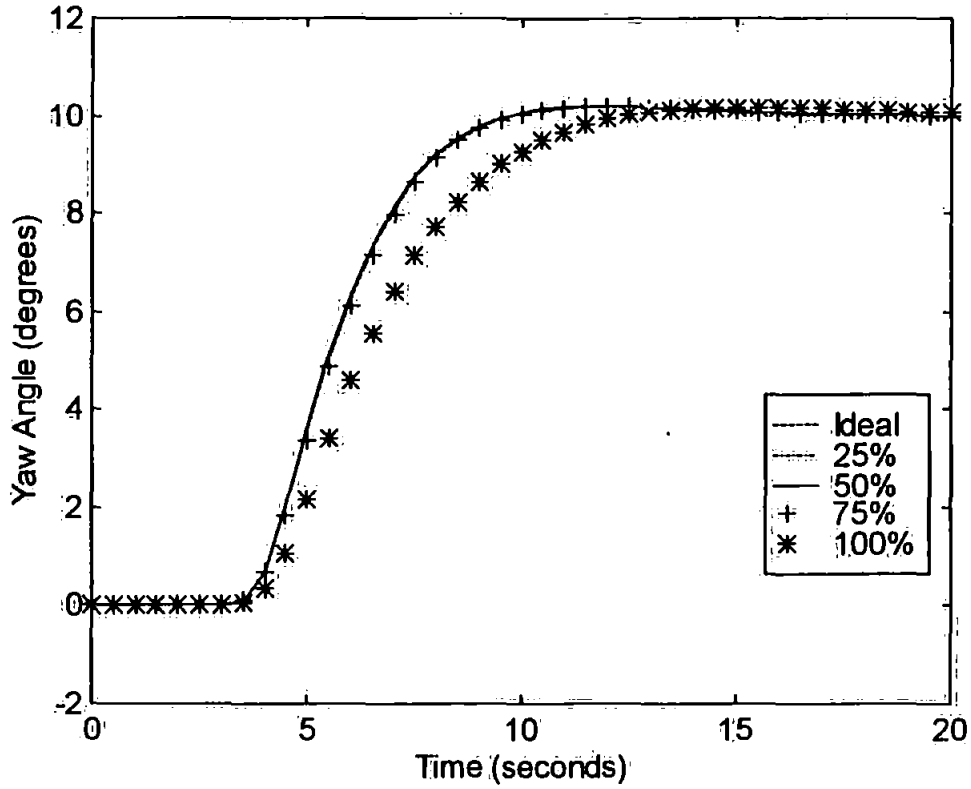


Figure 6.15 Simulated Annealing Results for The Saturation Block LOEs for a 10 Degrees Demanded Yaw Angle

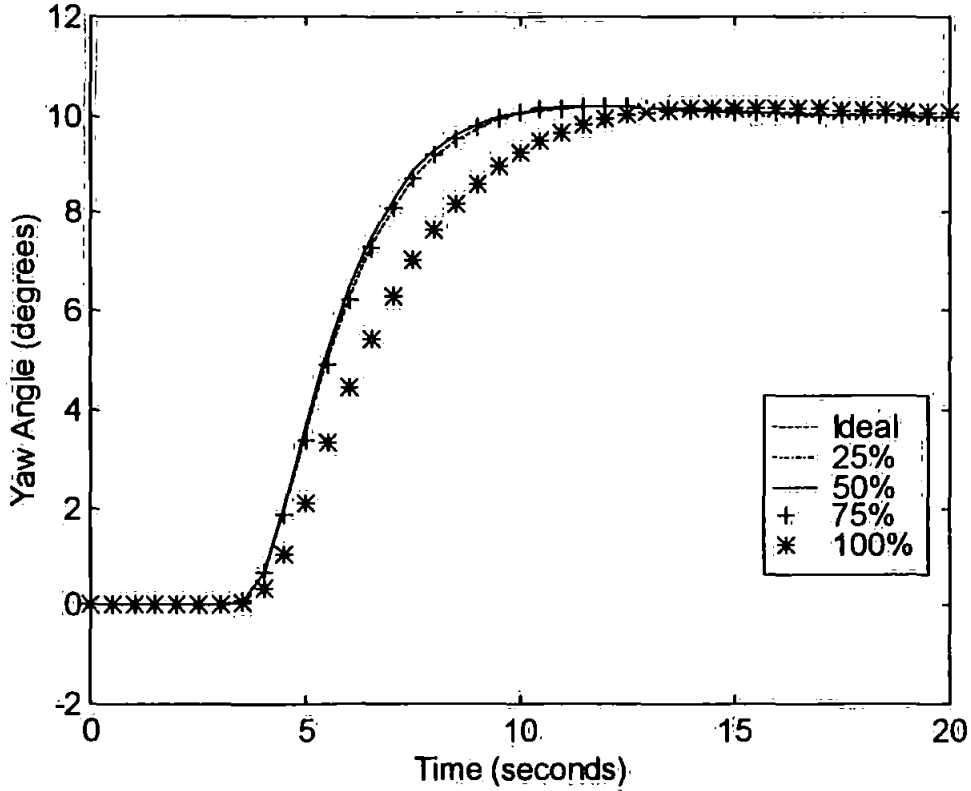


Figure 6.16 Tabu Search Results for The Saturation Block LOEs for a 10 Degrees Demanded Yaw Angle

On the previous page are shown the results for both the FISs for the saturation block LOEs in Figures 6.15 and 6.16.

The SA FIS for the 25% and 50% LOEs produced identical responses and were able to achieve the rise times of the ideal system and only produced a small RMSE (0.028 degrees). From Figure 6.15 it is possible to see the change in performance, with the AUV altering course at a slightly faster rate. For the 75% LOE the rise time was increased by only 0.1 seconds and the RMSE of 0.072 degrees is seen in Figure 6.15 as the AUVs response being slowed down. For the 100% LOE both the rise time and RMSE have increased considerably to 6.4 seconds and 0.782 degrees. This is illustrated in Figure 6.15, where the AUVs yaw angle can be seen to be below that of the ideal situation.

The TABU FIS for the 25% and 50% LOEs produced identical responses and were able to achieve a rise time 0.2 seconds shorter that the ideal system and only produced a small RMSE of 0.072 degrees. From Figure 6.16 it is clear to see that the TABU FIS has forced the AUV to alter yaw angle at a faster rate, which explains the decrease in rise times. For the 75% LOE the rise time is that of the ideal system and the RMSE of 0.053 degrees, which is closer to the ideal situation that either the 25% or 50% LOEs, is seen in Figure 6.16 as the AUVs response being slowed down. For the 100% LOE both the rise time and RMSE have increased considerably to 6.4 seconds and 0.820 degrees. This is illustrated in Figure 6.16, where the AUVs yaw angle can be seen to be below that of the ideal situation.

(b) Rate Limiter Block Faults

The next type of fault to be considered in this section is that of the rate limiter block LOE. The results for both of the fault tolerant FISs can be seen in Figures 6.17 and 6.18.

For this fault the SA FIS showed far more signs of a fault occurring. For the 25% LOE produced a response which showed a rise time identical to that of the ideal system and only produced a small RMSE (0.053 degrees). From Figure 6.17 it is possible to see the change in performance, with the AUV altering course at a slightly slower rate. The 50% LOE produced a further drop in performance with the rise time increasing to 4.6 seconds and a RMSE of 0.142 degrees. For the 75% LOE the rise time was again 4.6 seconds, the same as for the 50%, but an increase in the RMSE to

0.294 degrees and can be seen in Figure 6.17 as the AUVs response being slowed down. For the 100% LOE the results are identical to those of saturation block 100% LOE, this is due to 100% LOE in any of the considered faults being the same as the canard becoming locked in the zero position.

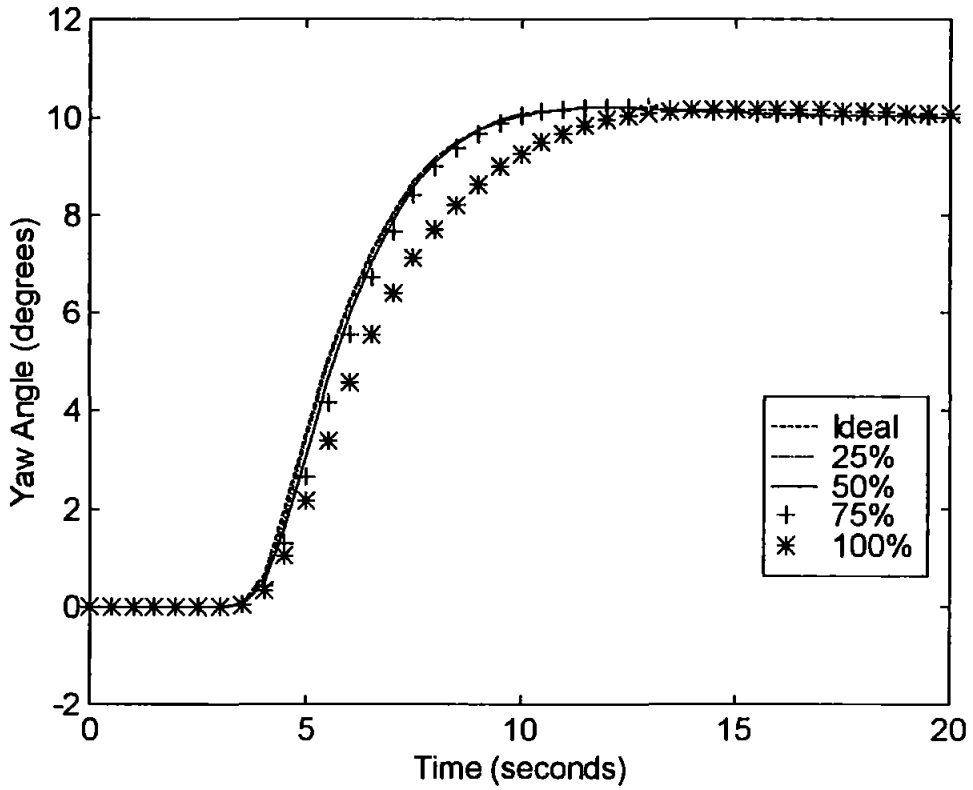


Figure 6.17 Simulated Annealing Results for The Rate Limiter Block LOEs for a 10 Degrees Demanded Yaw Angle

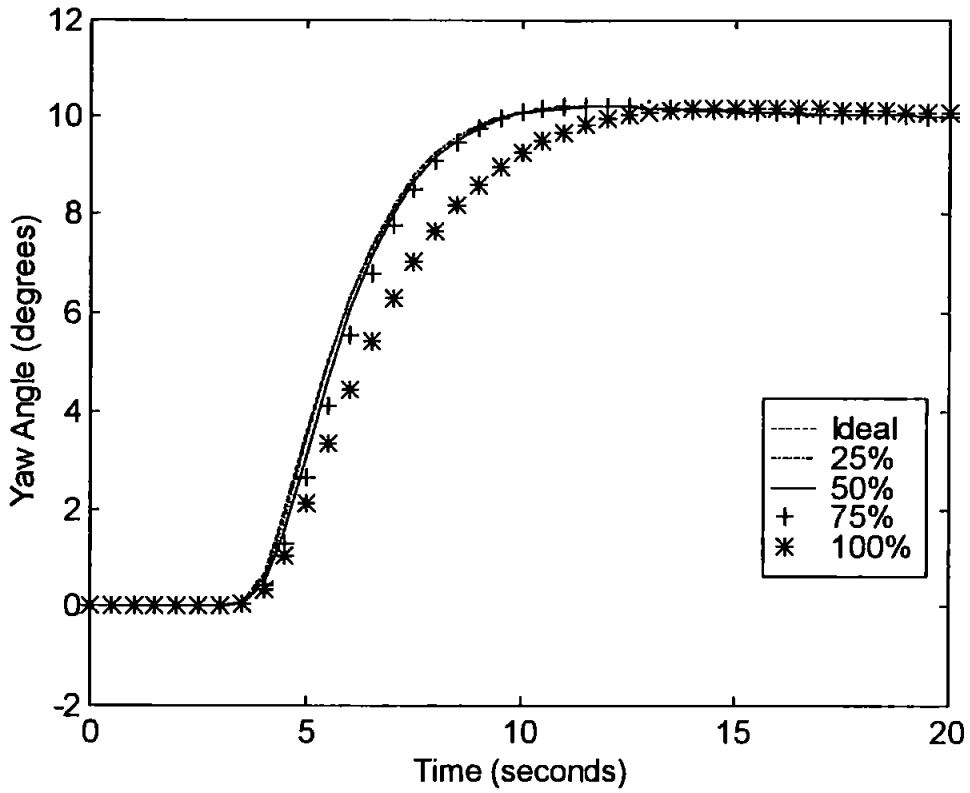


Figure 6.18 Tabu Search Results for The Rate Limiter Block LOEs for a 10 Degrees Demanded Yaw Angle

For this type of fault the TABU FIS also showed far more signs of a fault occurring. The 25% LOE produced a response which showed a rise time of 4.4 seconds, 0.1 seconds less than that of the ideal system and only produced a small RMSE (0.049 degrees). From Figure 6.18 it is possible to see the change in performance, with the AUV altering course at a slightly slower rate. The 50% LOE produced a further drop in performance with the rise time increasing to 4.5 seconds (that of the ideal system) and a RMSE of 0.129 degrees. For the 75% LOE the rise time was again 4.5 seconds, the same as for the 50%, but has an increase in the RMSE to 0.290 degrees and can be seen in Figure 6.18 as the AUVs response being slowed down. For the 100% LOE the results are identical to those of saturation block 100% LOE, for the same reasons as given in section 6.4.3 (a).

(c) Saturation and Rate Limiter Block Faults

Finally for this size of step input the fault where both the saturation and rate limiter blocks suffer a simultaneous LOE is considered. The results for both of the fault tolerant FISs can be seen in Figures 6.19 and 6.20.

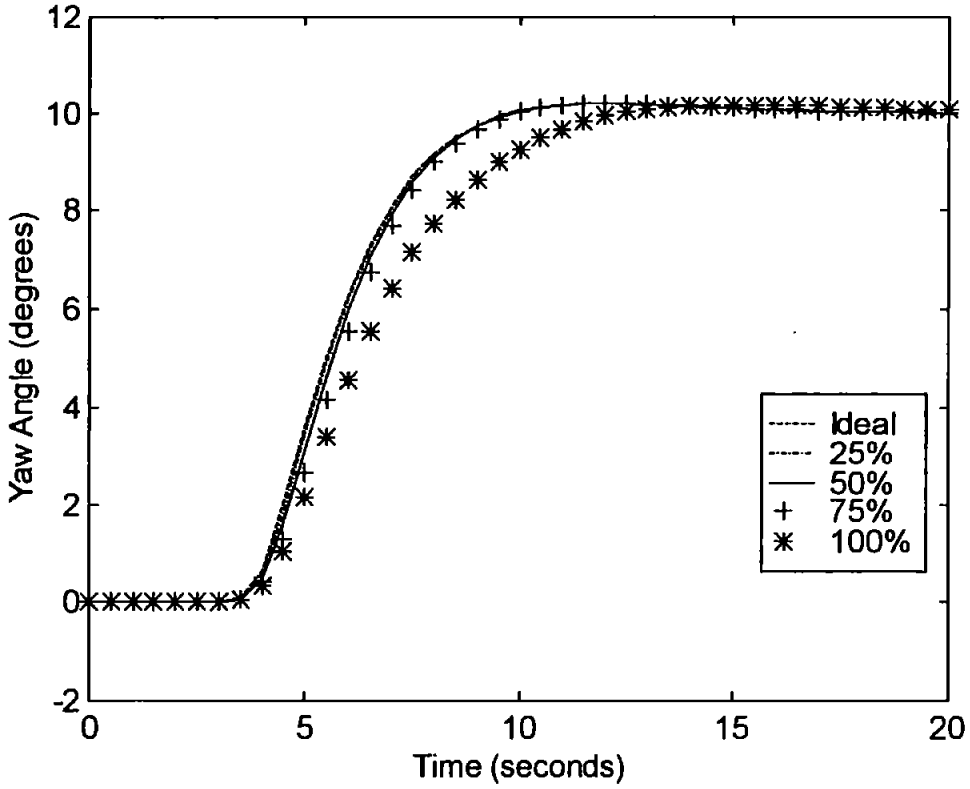


Figure 6.19 Simulated Annealing Results for Both Block LOEs for a 10 Degrees Demanded Yaw Angle

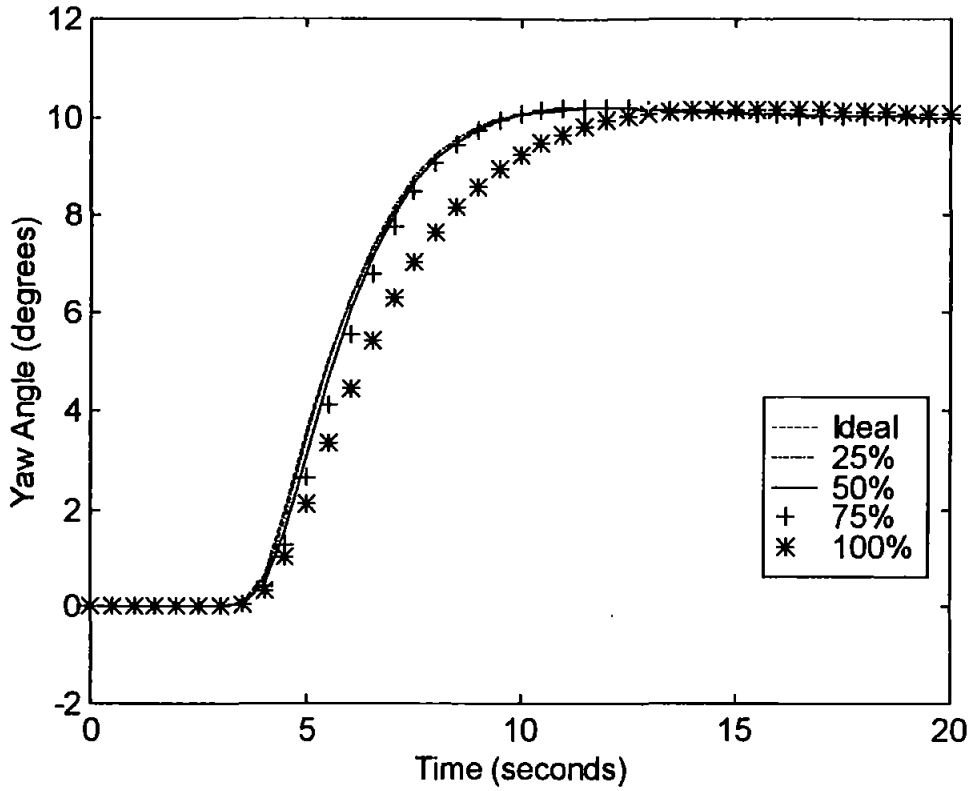


Figure 6.20 Tabu Search Results for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle

When both faults occurred simultaneously the results shown in Tables 6.1 and 6.2 and in Figures 6.19 and 6.20 indicate that the results are identical to those of the rate limiter LOEs. The explanation for this is that when both faults occur simultaneously, for this small size of step input, one of the effects of the rate limiter fault is to prevent the actuator obtaining an angle for which the saturation fault will affect the results. Hence for this size of step input when both faults occur it is effectively just a rate limiter block fault occurring and both FISs treated it as such.

The canard responses for the SA FIS are shown in Figures 6.21 and 6.22.

Figure 6.21 shows how the upper canard is affected by the fault with the canard's response being slowed as the level of LOE increases up to the maximum 100% LOE. The canard does not function for this level of fault as is shown by the canard remaining at angle of zero degrees.

Figure 6.22 shows how the lower canard compensates for the faults. This is shown as the canard achieving a larger angle for a larger LOE. From the figures it is also clear that neither actuator reaches saturation apart from the upper canard when considering the 100% LOE, where this is inevitable.

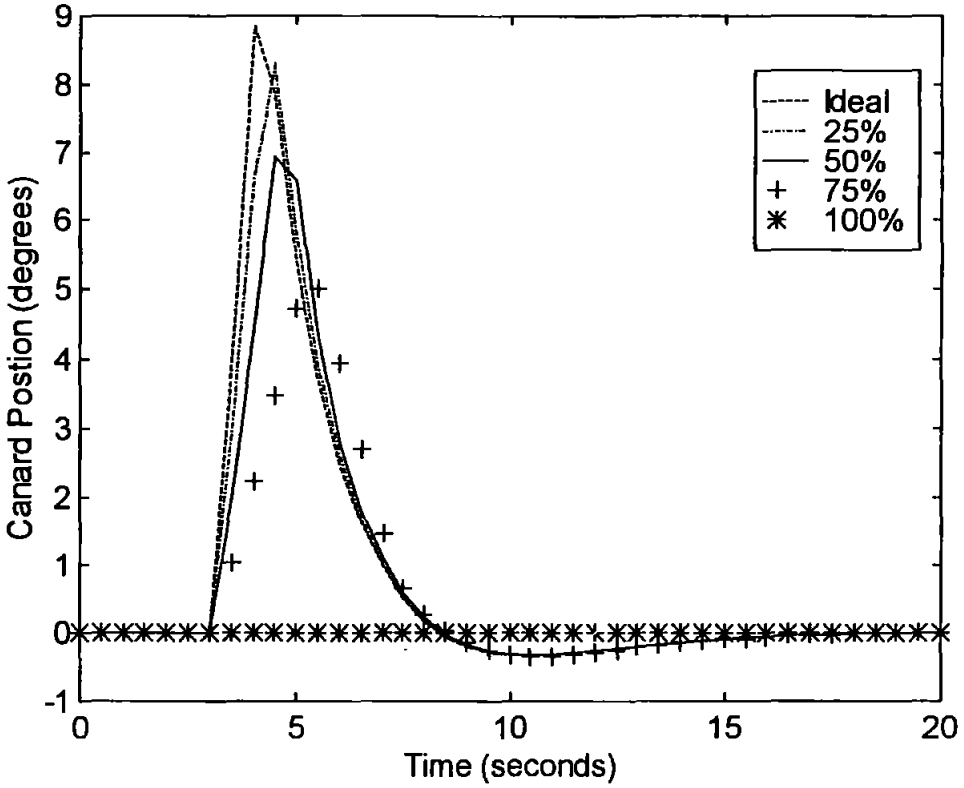


Figure 6.21 Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle

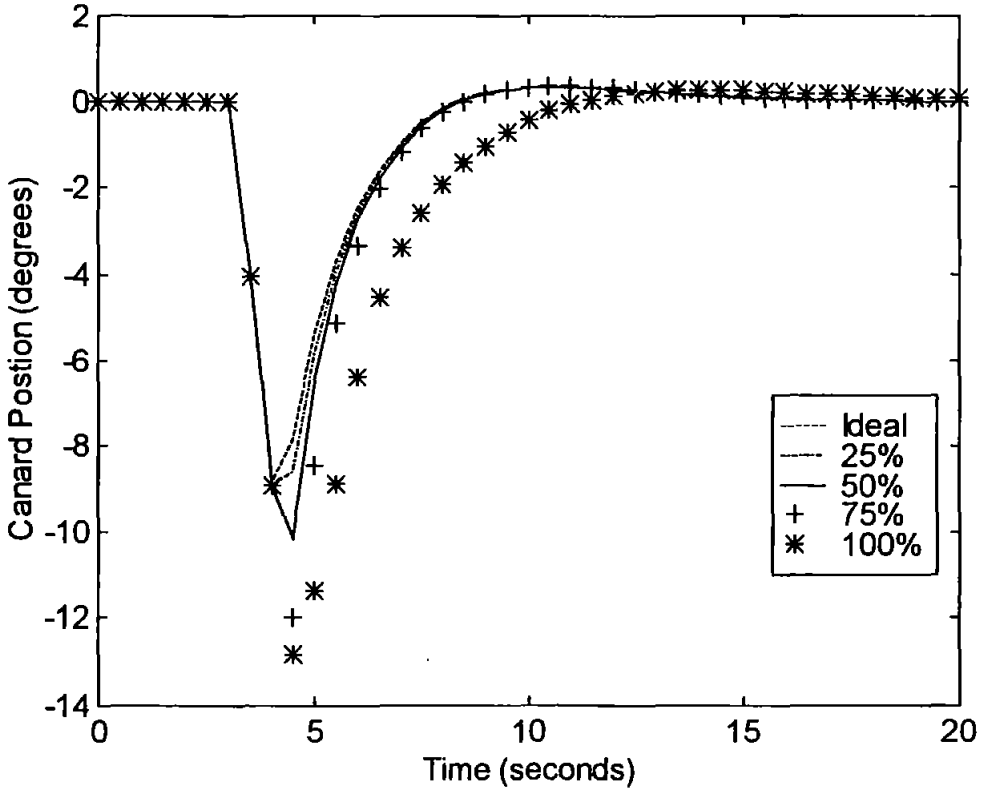


Figure 6.22 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 10 Degrees Demanded Yaw Angle

Having considered all types of faults for this size of yaw angle demand a further calculation was performed to show the percentage improvement of the FISs compared to the PD controlled AUV. These can be seen in Table 6.3.

Table 6.3 The 10 Degrees Step Input Percentage Improvements.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
Simulated Annealing	88%	88%	45%	34%	69%	17%	18%	34%	69%	17%	18%	34%
Tabu Search	68%	68%	59%	31%	72%	25%	19%	31%	72%	25%	19%	31%

6.4.4. Yaw Step Inputs of 20 Degrees

Having presented the results produced by both fault tolerant FISs in the case of a 10 degrees step input yaw demand, the next step is to increase that yaw angle to 20 degrees and observe how both FISs respond.

The complete set of results (RMSEs and rise times) for both of the FISs described, along with the benchmark PD controller, are shown in Tables 6.4 and 6.5. At this point it is important to note that the unaffected ANFIS controlled AUV has a rise time of 4.9 seconds and a RMSE of zero degrees.

Table 6.4 The 20 Degrees Step Input RMSEs.

	RMSEs (degrees)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	0.261	0.193	0.732	2.646	0.251	0.537	1.151	2.646	0.251	0.537	1.194	2.646
Simulated Annealing	0.023	0.111	0.694	1.678	0.203	0.531	1.029	1.678	0.203	0.531	1.072	1.678
Tabu Search	0.095	0.071	0.621	2.024	0.184	0.466	0.985	2.024	0.184	0.467	1.026	2.024

Table 6.5 The 20 Degrees Step Input Rise Times.

	Rise Times (seconds)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	6	6	6.5	8.7	5.9	5.9	6.2	8.7	5.9	5.9	6.3	8.7
Simulated Annealing	4.8	4.9	5.4	6.3	4.8	4.9	4.8	6.3	4.8	4.9	4.9	6.3
Tabu Search	4.7	4.8	5.3	6.9	4.6	4.5	4.7	6.9	4.6	4.5	4.8	6.9

It is again obvious from these results that both fault tolerant FISs are an improvement on the PD controller. With respect to both the RMSE and rise times methods of evaluation, the FISs have smaller RMSE values and rise times for every test performed.

(a) Saturation Block Faults

In Figures 6.23 and 6.24 the results are shown for both the FISs for the saturation block LOEs in.

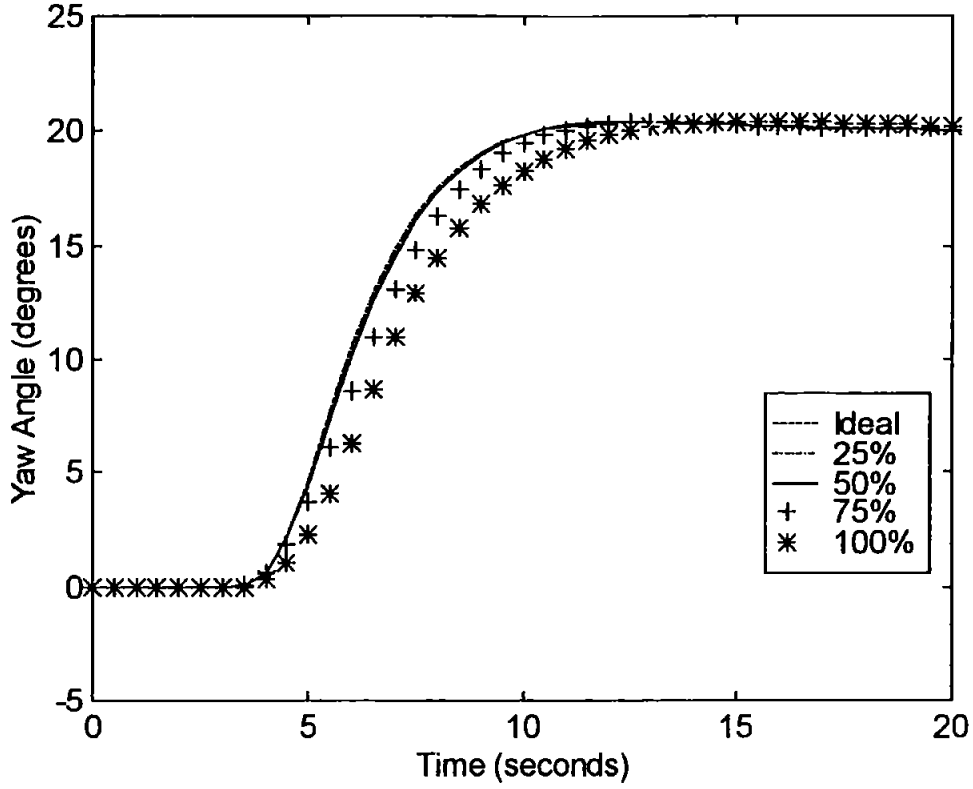


Figure 6.23 Simulated Annealing Results for The Saturation Block LOEs for a 20 Degrees Demanded Yaw Angle

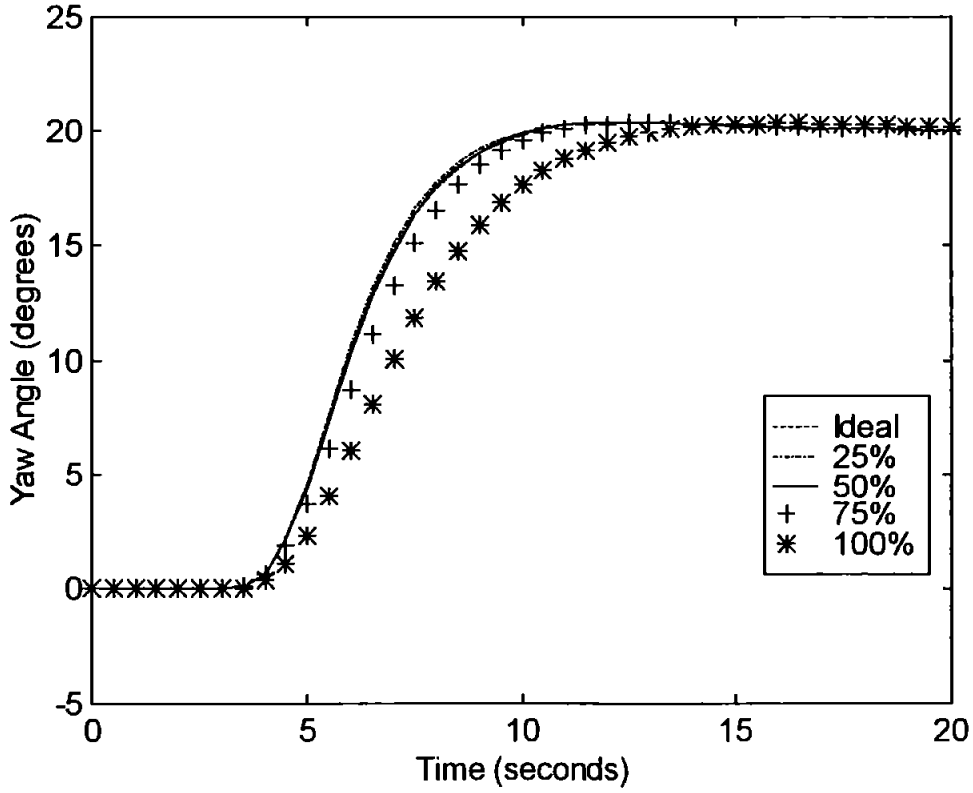


Figure 6.24 Tabu Search Results for The Saturation Block LOEs for a 20 Degrees Demanded Yaw Angle

The SA FIS for the 25% LOE produced a very good response with a RMSE of only 0.023 degrees and a rise time 4.8 seconds, 0.1 seconds less than that of the ideal system. From Figure 6.23 it is possible to see the change in performance. For the 50% LOE the rise time is identical to that of the idea system and a RMSE of 0.111 degrees indicates the drop in the level of performance. For the 75% LOE the rise time has increased considerably to 5.4 seconds and the RMSE to 0.694 degrees. As shown in Figure 6.23, this indicates that for the increased demand yaw angle the actuator is more susceptible to a fault in the saturation block. For the 100% LOE both the rise time and RMSE have increased considerably to 6.3 seconds and 1.678 degrees. This is illustrated in Figure 6.23, where the AUVs yaw angle can be seen to be below that of the ideal situation.

The TABU FIS for the 25% LOE was able to achieve a rise time 0.2 seconds shorter that the ideal system and only produced a small RMSE of 0.095 degrees. From Figure 6.24 it is clear to see that the TABU FIS has forced the AUV to alter yaw angle at a faster rate, which explains the decrease in rise times. This has also been the case for the 50% LOE where a rise time of 4.8 seconds has been observed and a RMSE of 0.071 degrees is less than that of the 25% LOE. For the 75% LOE the rise time has increased considerably to 5.3 seconds and a RMSE of 0.621 degrees. For the 100% LOE both the rise time and RMSE have increased considerably to 6.9 seconds and 2.024 degrees. This is illustrated in Figure 6.24, where the AUVs yaw angle can be seen to be further below that of the ideal situation as the level of LOE increases and hence has more of an effect on the AUV.

(b) Rate Limiter Block Fault

The next type of fault to be considered in this section is that of the rate limiter block LOE. The results for both of the fault tolerant FISs can be seen in Figures 6.25 and 6.26.

Firstly for this type of fault the SA FIS shall be examined. For the 25% LOE a response which showed a rise time of 4.8 seconds and a RMSE of 0.203 degrees was recorded. From Figure 6.25 it is possible to see the change in performance, with the AUV altering course at a slightly slower rate. The 50% LOE produced a further drop in performance with the rise time increasing to 4.9 seconds and a RMSE of 0.531 degrees. For the 75% LOE the rise time decreased to 4.8 seconds, but an increase in

the RMSE to 1.029 degrees was recorded and can be seen in Figure 6.25 where an overshoot is now being induced by the FIS to compensate for the failure. For the 100% LOE the results are identical to those of saturation block 100% LOE for reasons previously given.

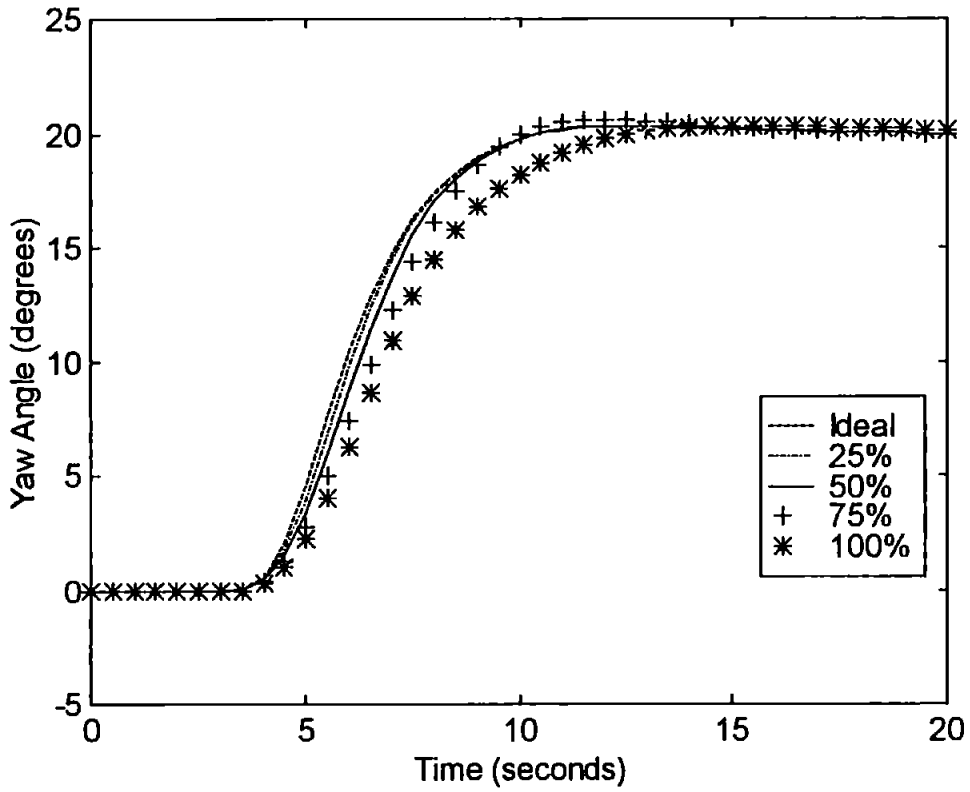


Figure 6.25 Simulated Annealing Results for The Rate Limiter Block LOEs for a 20 Degrees Demanded Yaw Angle

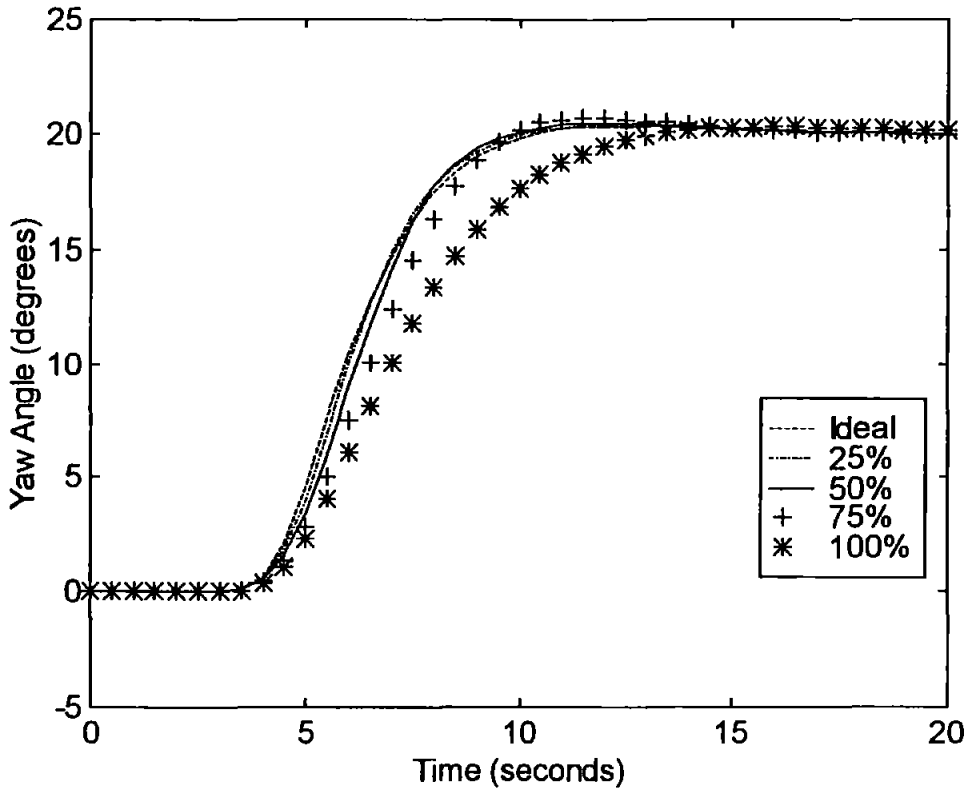


Figure 6.26 Tabu Search Results for The Rate Limiter Block LOEs for a 20 Degrees Demanded Yaw Angle

For this fault the TABU FIS also showed more signs of a fault occurring. For the 25% LOE that produced a response which showed a rise time of 4.6 seconds, 0.2 seconds less than that of the ideal system and only produced a RMSE of 0.184 degrees. From Figure 6.26 it is possible to see the change in performance, with the AUV altering course at a slightly slower rate. The 50% LOE produced a further drop rise time to 4.5 seconds and a RMSE of 0.466 degrees. For the 75% LOE the rise time was 4.7 seconds, again less than the ideal response, and an increase in the RMSE to 0.985 degrees was detected. Figure 6.26 shows how for all three LOEs the AUVs response has been slowed down, but an induced overshoot can be seen in. For the 100% LOE the results are identical to those of saturation block 100% LOE, for the same reasons as given in section 6.4.4 (a).

(c) Saturation and Rate Limiter Block Faults

Finally for this size of step input the fault where both the saturation and rate limiter blocks endure a simultaneous LOE is considered. The results for both of the fault tolerant FISs can be seen in Figures 6.27 and 6.28.

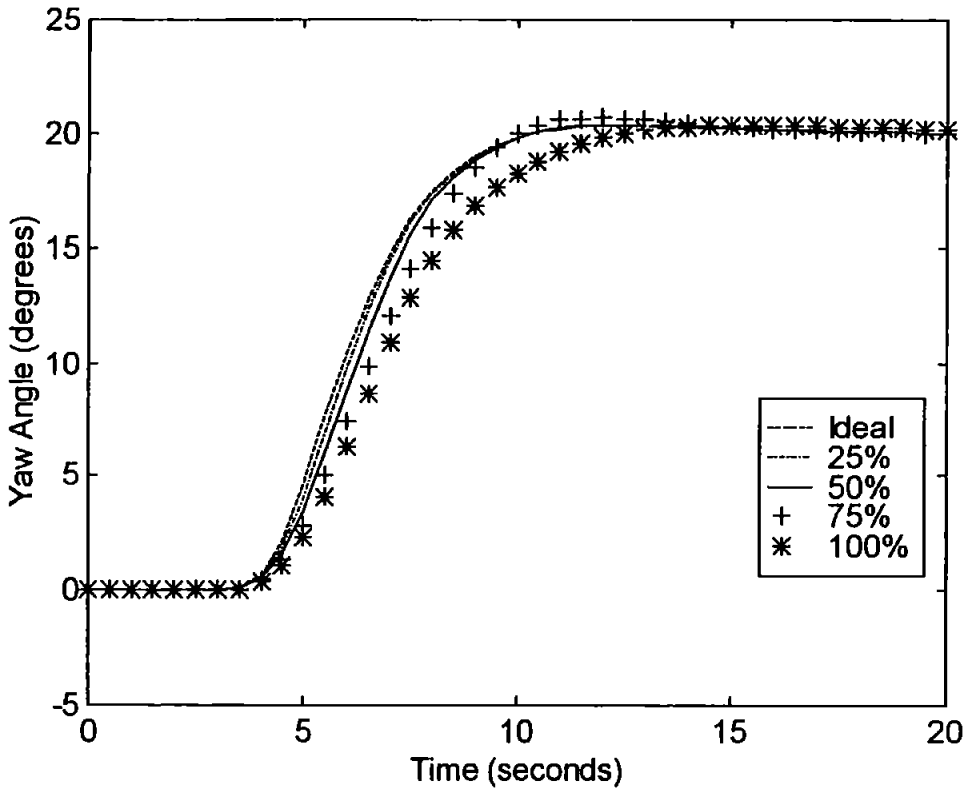


Figure 6.27 Simulated Annealing Results for Both Block LOEs for a 20 Degrees Demanded Yaw Angle

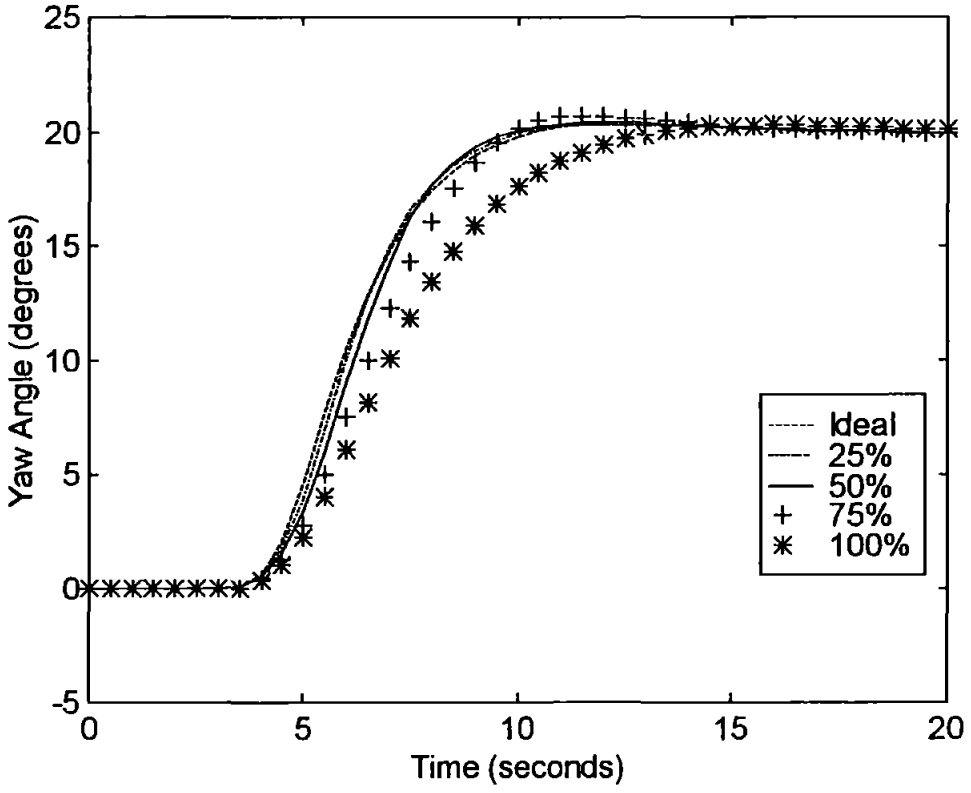


Figure 6.28 Tabu Search Results for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle

As was the case for the 10 degrees yaw step input, when the SA FIS is presented with simultaneous LOEs it produces identical rise times and RMSEs to those of the rate limiter block LOEs for the 25% and 50% LOE cases. The reason given for this occurrence is equally valid for these cases. The 75% LOE is the first test where it is apparent that both faults are effecting the performance of the AUV. The rise time of 4.9 seconds is greater than that of the rate limiter block 75% LOE and the RMSE is 1.072 degrees which is also an increase. The only possible explanation for this is that the saturation block LOE is having an effect and the SA FIS is attempting to alter the response of the AUV accordingly. Once again the 100% LOE is identical to previous results, for the reasons given in section 6.4.4 (a).

As for the SA FIS, when the TABU FIS is presented with simultaneous LOEs it produces identical rise times and RMSEs to those of the rate limiter block LOEs, but only for the 25% LOE case. The reason given for this occurrence is equally valid for these cases. The 50% LOE also has a rise time identical to that for the rate limiter

block LOE, of 4.5 seconds, but has an increase of 0.001 degrees in its RMSE. The 75% LOE is the first test where it is apparent that both faults are effecting the performance of the AUV. The rise time of 4.8 seconds is greater than that of the rate limiter block 75% LOE and the RMSE is 1.026 degrees which is also an increase. The only possible explanation for this is that the saturation block LOE is having an effect and the TABU FIS is attempting to alter the response of the AUV accordingly. Once again the 100% LOE is identical to previous results, for the reasons given in section 6.4.4 (a).

The canard responses for the SA FIS are shown in Figures 6.29 and 6.30.

Figure 6.29 shows how the upper canard is affected by the fault with the canard's response being slowed as the level of LOE increases up to the maximum 100% LOE. The canard does not function for this level of fault as is shown by the canard remaining at angle of zero degrees. For the first time it is also clear the effect the saturation fault has on the system. This is shown by the 75% LOE where the canard becomes saturated at 6.3 degrees.

Figure 6.30 shows how the lower canard compensates for the faults. This is shown as the canard achieving a larger angle for a larger LOE. The lower canard achieves a maximum angle of just over 24 degrees, which is just below the saturation level for this canard. This shows that the lower canard never reaches saturation for any level of fault, when a 20 degrees yaw angle is being demanded.

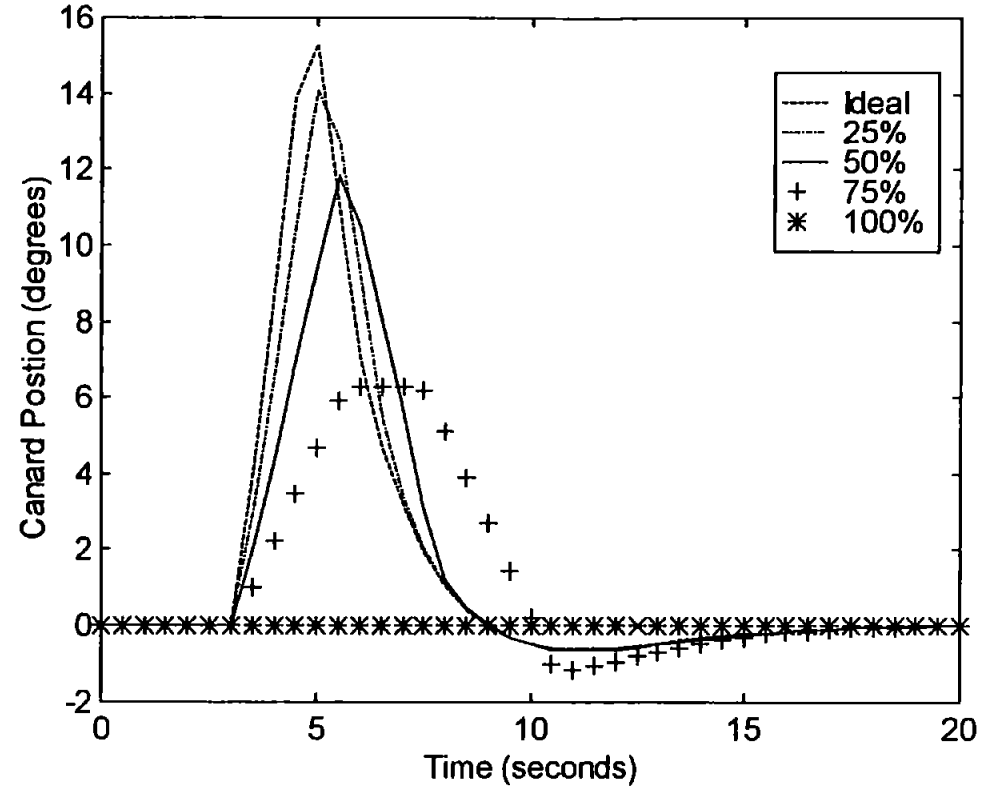


Figure 6.29 Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle

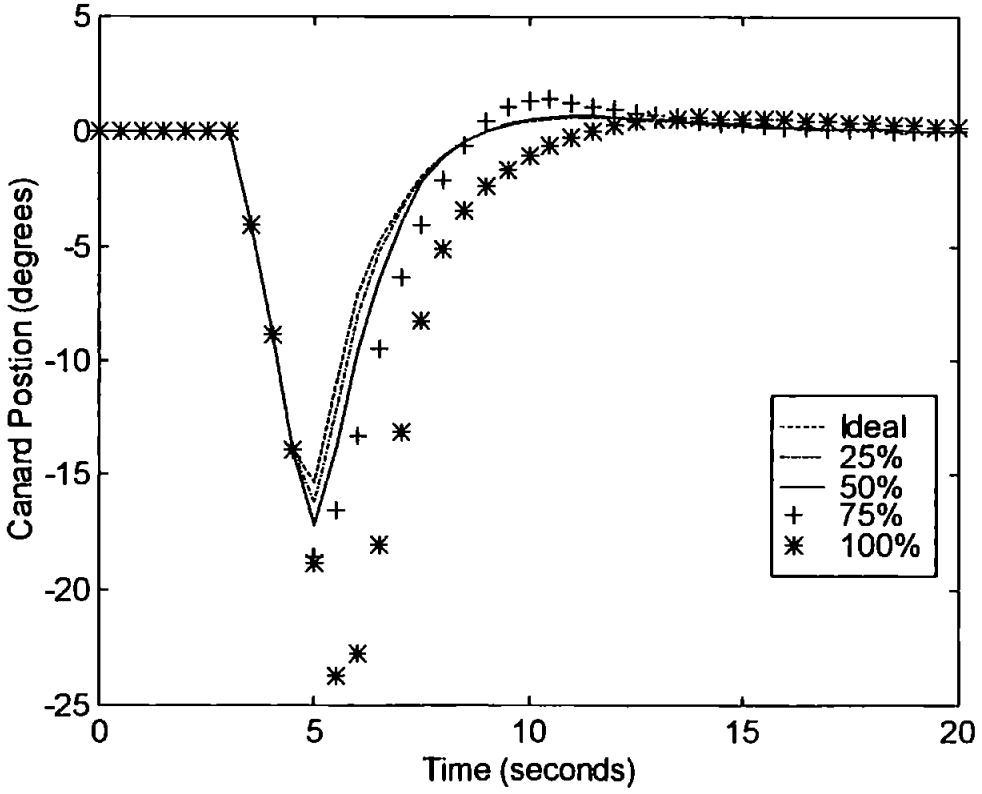


Figure 6.30 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 20 Degrees Demanded Yaw Angle

Having considered all types of faults for this size of yaw angle demand a further calculation was performed to show the percentage improvement of the FISs compared to the PD controlled AUV. These can be seen in Table 6.6.

Table 6.6 The 20 Degrees Step Input Percentage Improvements.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
Simulated Annealing	91%	42%	5%	37%	19%	1%	11%	37%	19%	1%	10%	37%
Tabu Search	64%	63%	15%	24%	27%	13%	14%	24%	27%	13%	14%	24%

6.4.5. Yaw Step Inputs of 30 Degrees

The final size of step input yaw angle being considered for actuator faults, within this thesis, is that of 30 degrees. This is the largest step size considered for reasons given in Chapter 3. Once again all types and sizes of LOEs previously discussed have been implemented for this step size.

The complete set of results (RMSEs and rise times) for both the SA FIS and TABU FIS, along with the benchmark PD controller, are shown in Tables 6.7 and 6.8. At this point it is important to note that the unaffected ANFIS controlled system has a rise time of 4.9 seconds and a RMSE of zero degrees (as to be expected). The ideal system having a rise time of 4.9 seconds, the same as for the 20 degrees step input yaw angle demand, is explained by the very good performance of the original ANFIS controller as discussed by Craven (1999).

It is again obvious from these results that both fault tolerant FISs are an improvement on the benchmark PD controller. With respect to both the RMSE and rise times methods of evaluation, the FISs have smaller RMSE values and shorter rise times for every test performed.

Table 6.7 The 30 Degrees Step Input RMSEs.

	RMSEs (degrees)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	0.175	0.686	1.994	4.494	0.489	1.176	2.340	4.494	0.494	1.256	2.538	4.494
Simulated Annealing	0.082	0.673	1.514	3.535	0.435	1.094	2.055	3.535	0.437	1.140	2.129	3.535
Tabu Search	0.059	0.562	1.720	3.817	0.563	1.131	2.061	3.817	0.542	1.138	2.237	3.817

Table 6.8 The 30 Degrees Step Input Rise Times.

	Rise Times (seconds)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
PD	6.2	6.5	7.4	9.5	5.6	5.8	6.7	9.5	5.6	6.1	7.2	9.5
Simulated Annealing	5	5.4	5.6	6.8	4.6	4.5	4.8	6.8	4.6	4.7	5.2	6.8
Tabu Search	4.9	5.2	5.9	7.4	4.3	4.4	5.1	7.4	4.3	4.7	5.6	7.4

(a) Saturation Block Faults

Shown in Figures 6.31 and 6.32 are the results for both the FISs for the saturation block LOEs.

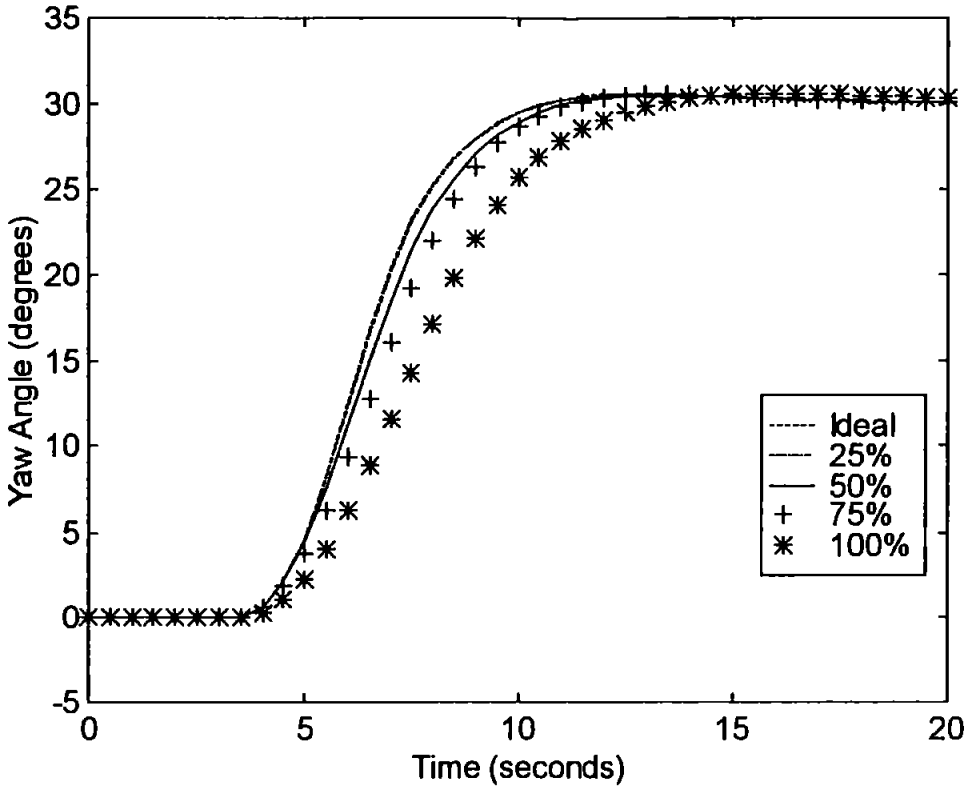


Figure 6.31 Simulated Annealing Results for The Saturation Block LOEs for a 30 Degrees Demanded Yaw Angle

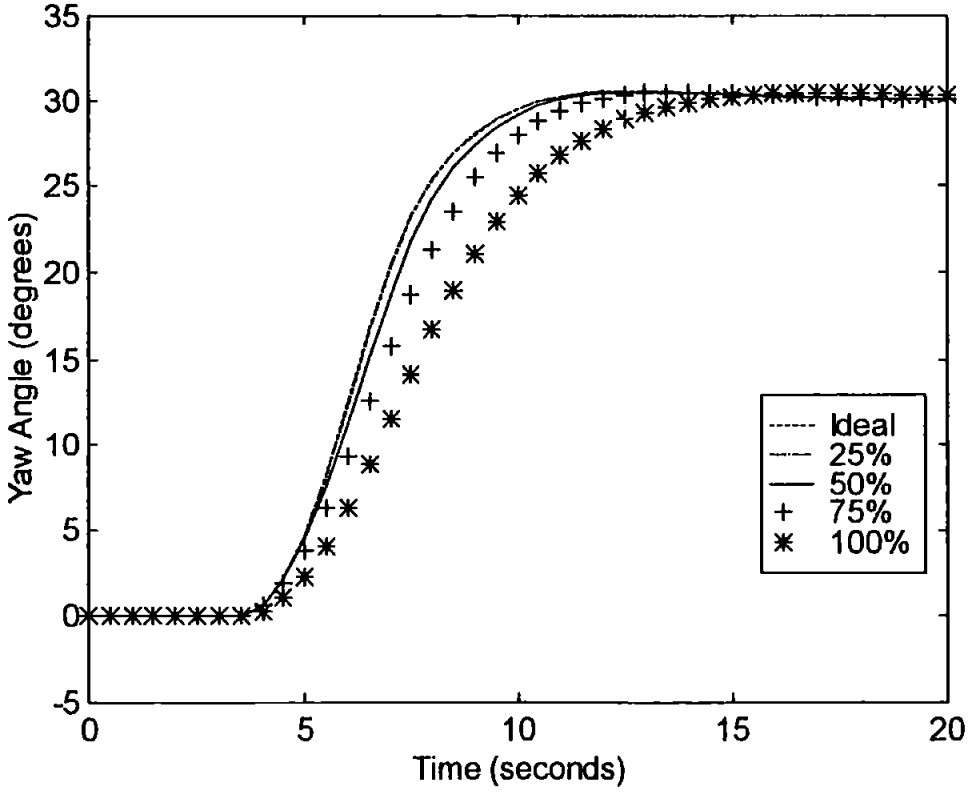


Figure 6.32 Tabu Search Results for The Saturation Block LOEs for a 30 Degrees Demanded Yaw Angle

The results for the SA FIS when considering the saturation block LOEs are as to be expected. The rise times and RMSEs increase as the percentage LOE increases. There are no overshoots detectable for any level of LOE. The SA FIS has some effect for all levels of saturation block LOEs, but is limited at this angle as the canard very quickly achieves the maximum obtainable angle and once saturation has been reach, there is no effect the control system can have. This is a problem when dealing with such a large yaw angle.

The results for the TABU FIS on this type of fault are of a similar nature, due once again too the limiting situation created by the large yaw angle demand.

(b) Rate Limiter Faults

The next type of fault to be considered in this section is that of the rate limiter block LOE. The results for both of the fault tolerant FISs can be seen in Figures 6.33 and 6.34.

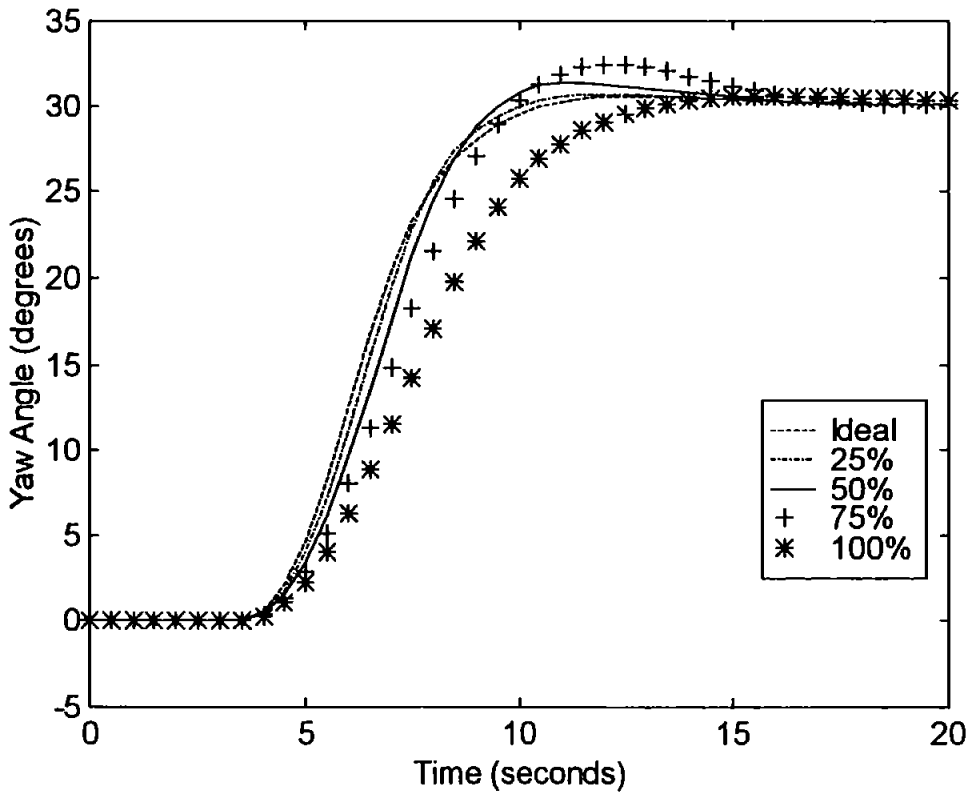


Figure 6.33 Simulated Annealing Results for The Rate Limiter Block LOEs for a 30 Degrees Demanded Yaw Angle

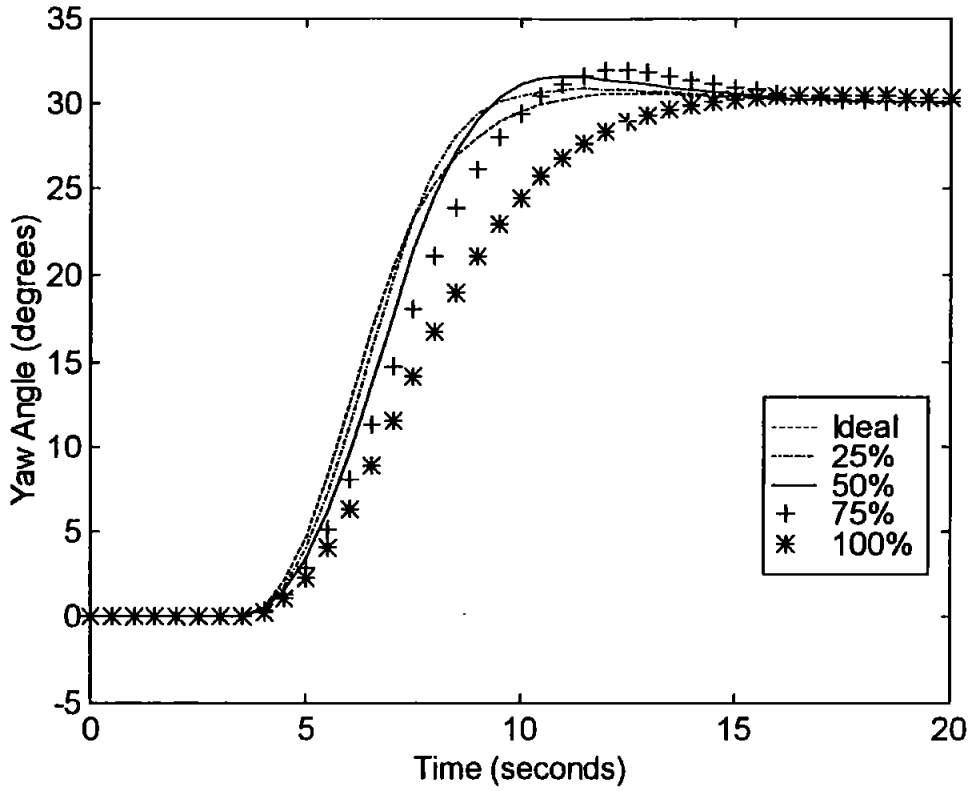


Figure 6.34 Tabu Search Results for The Rate Limiter Block LOEs for a 30 Degrees Demanded Yaw Angle

For this type of fault over this step input size the SA FIS has its most noticeable effect yet. For the 25% LOE the rise time has been reduced by 0.6 seconds and despite a RMSE of 0.435 degrees, a look at Figure 6.33 shows how the FIS forces a slight overshoot and yet obtains the desired yaw angle before the ideal system. If the criteria of the tests were not to reproduce the unaffected system this result would be an improvement on that ideal situation. For both the 50% and 75% LOEs both rise times were still below that of the ideal system (4.5 and 4.8 seconds respectively), which were once again due to the FIS inducing an overshoot. Unfortunately unlike for the 25% LOE case, for these cases the overshoots were of such magnitude that the system required a longer time to return to the desired yaw angle. This is shown as RMSEs of 1.094 degrees for the 50% LOE and 2.055 degrees for the 75% LOE, and can be seen in Figure 6.33. The 100% LOE results where again identical to those of the saturation block 100% LOE, for reasons given prior to this point.

The TABU FIS also showed similar results to those of SA FIS. For the 25% LOE the TABU FIS forced an overshoot in the AUV which, unlike the SA FIS on this test, the

control system could not recover from quickly enough from to achieve the desired yaw angle in a short time than the ideal system. It had a rise time of 4.3 seconds, 0.6 seconds less than the ideal system and a RMSE of 0.563 degrees which is bigger than that of the benchmark PD controller. This is the first test where either fault tolerant FIS has failed to improve on the benchmark results. The 50% LOE test does not cause the same problem for this FIS and a rise time of 4.4 seconds and a RMSE of 1.131 degrees are produced. The Figure 6.34 shows how the TABU FIS once again induces an overshoot in an attempt to compensate for the fault. The 75% LOE shows an increase in rise time to 5.1 seconds and an increase in RMSE to 2.061 degrees, and again an overshoot is seen in Figure 6.34. The 100% LOE results where again identical to those of the saturation block 100% LOE, for reasons given prior to this point.

(c) Saturation and Rate Limiter Faults

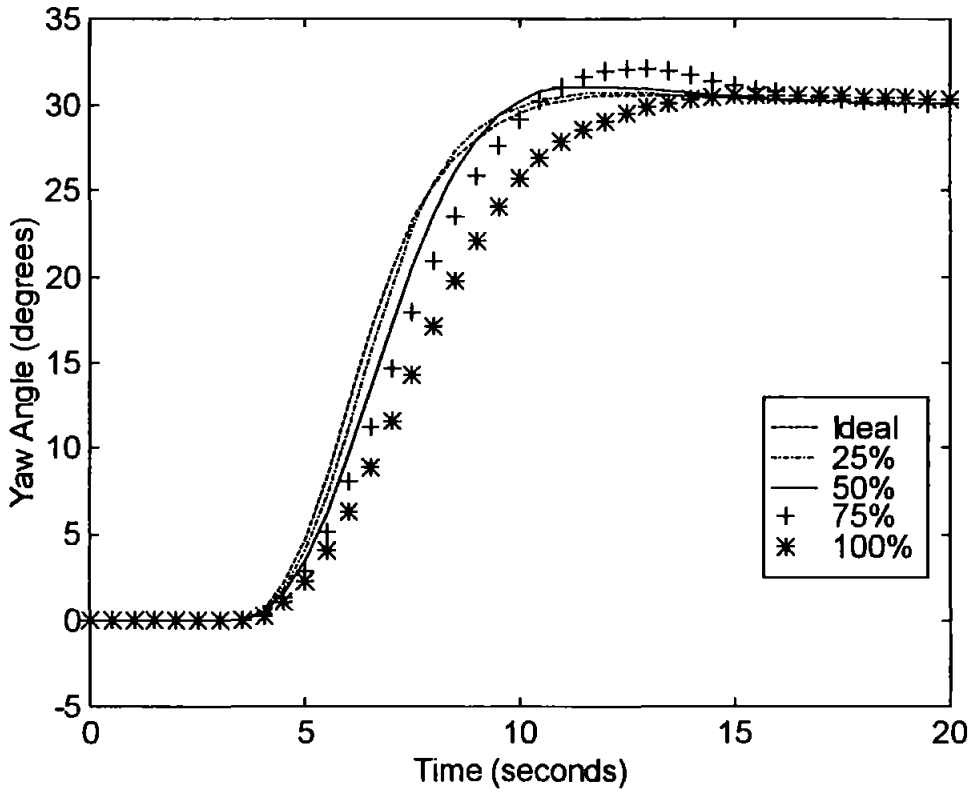


Figure 6.35 Simulated Annealing Results for Both Block LOEs for a 30 Degrees Demanded Yaw Angle

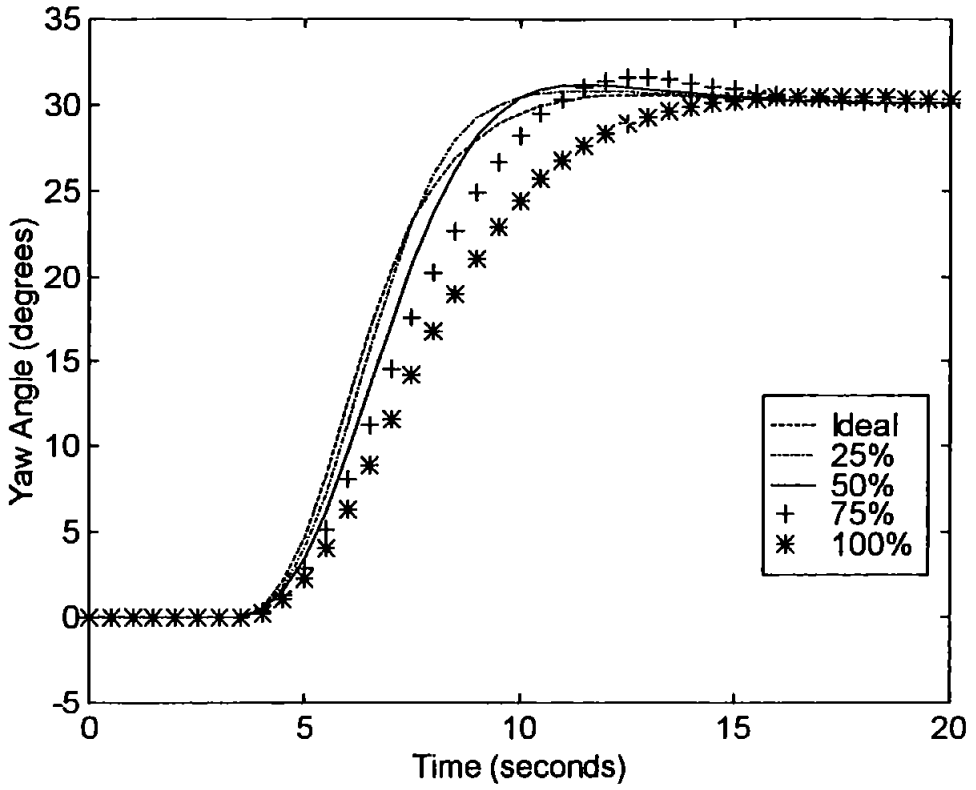


Figure 6.36 Tabu Search Results for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle

Unlike for the previous step sizes, for a 30 degrees step input demand, when the SA FIS is presented with simultaneous LOEs it does not produce any identical rise times and RMSEs to those of the rate limiter block LOEs. As the step is now of such a large magnitude that both faults have an effect on the actuator for all considered levels of LOE. The 25% LOE shows little evidence of this with the same rise time of 4.6 seconds and an increase of only 0.002 degrees in its RMSE value. The 50% LOEs rise time has decreased by 0.2 seconds compared to the rate limiter block 50% LOE, and its RMSE has increased to 1.140 degrees. This can be seen in Figure 6.35 as a decrease in the size of the overshoot. The 75% LOE has a rise time of 4.9 seconds, which is that of the ideal system for this step size and a RMSE of 2.129 degrees. Once again the 100% LOE is identical to previous results as a 100% LOE is the same as locking the actuator at zero degrees.

The TABU FIS when presented with simultaneous LOEs does not produce any identical rise times or RMSEs to those of the rate limiter block LOEs. The 25% LOE once again shows a RMSE larger than that for the benchmark controller, but does

have the decreased rise time of 4.3 seconds. This can be seen in Figure 6.36 as an overshoot. The 50% LOE also has a rise time of 4.7 seconds, which is an increase to that for the rate limiter block 50% LOE, of 4.5 seconds and has an increase of 0.007 degrees in its RMSE. The 75% LOE show significant signs that both faults are effecting the performance of the AUV. The rise time of 5.6 seconds is half a second greater than that of the rate limiter block 75% LOE and the RMSE is 2.237 degrees which is also an increase. Once again the 100% LOE is identical to previous results.

The canard responses for the SA FIS are shown in Figures 6.37 and 6.38.

Figure 6.37 shows how the upper canard is affected by the fault with the canard's response being slowed as the level of LOE increases up to the maximum 100% LOE. The canard does not function for this level of fault as is shown by the canard remaining at angle of zero degrees. Again it is clear the effect the saturation fault has on the system with the canard now becoming saturated for both the 50% and 75% LOE tests.

Figure 6.38 shows how the lower canard compensates for the faults. This is shown as the canard achieving a larger angle for a larger LOE. The lower canard does for this level of step input achieve the saturation level for this canard, but only for the 100% LOE test. The complete set of upper and lower canard responses are presented in Appendix I.

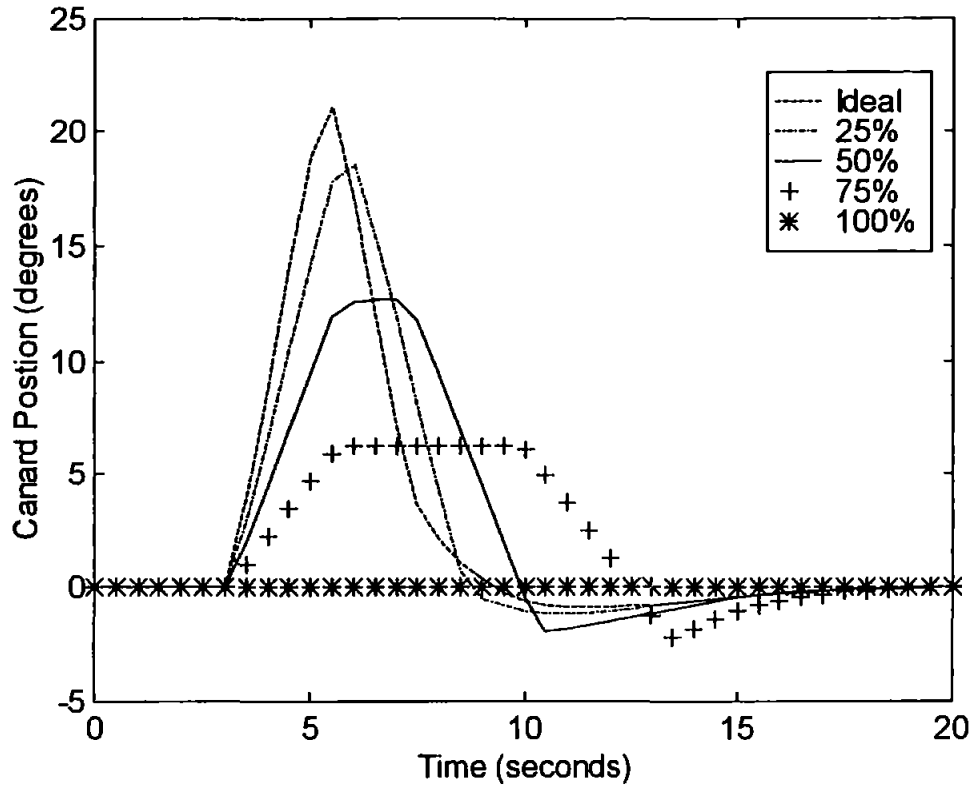


Figure 6.37 Upper Canard Responses When Using SA FIS for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle

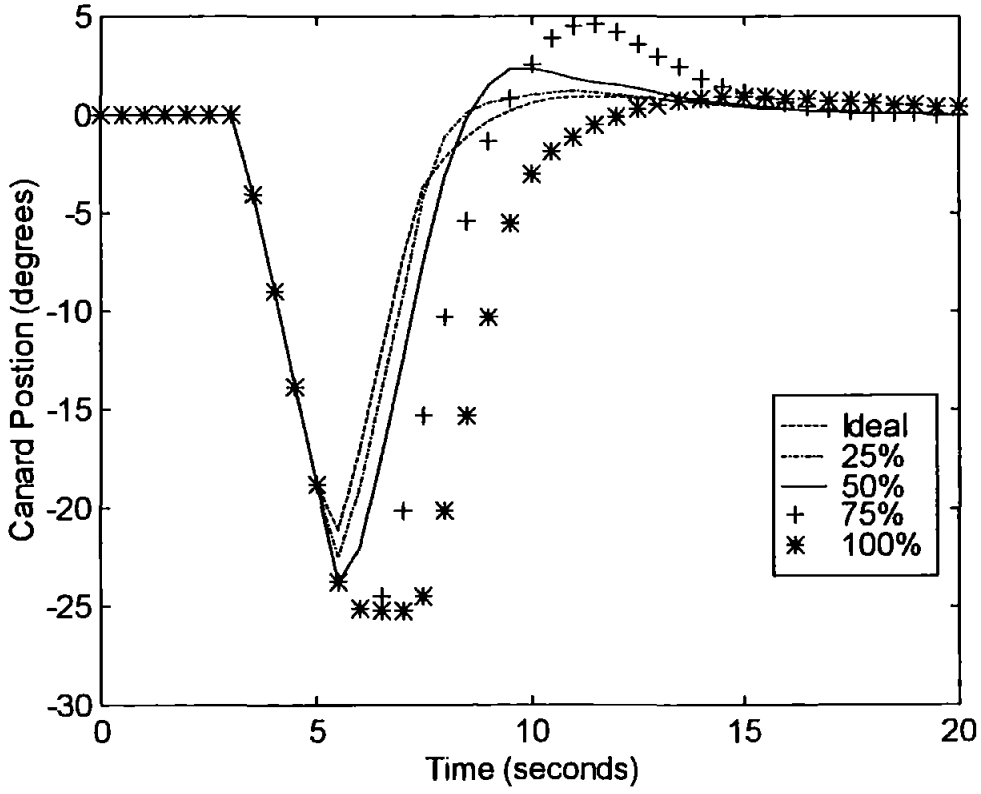


Figure 6.38 Lower Canard Responses When Using SA FIS for Both Blocks LOEs for a 30 Degrees Demanded Yaw Angle

Having considered all types of faults for this size of yaw angle demand a further calculation was performed to show the percentage improvement of the FISs compared to the PD controlled AUV. These can be seen in Table 6.9.

Table 6.9 The 30 Degrees Step Input Percentage Improvements.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
Simulated Annealing	53%	2%	24%	21%	11%	7%	12%	21%	12%	9%	16%	21%
Tabu Search	66%	18%	14%	15%	-15%	4%	12%	15%	-10%	9%	12%	15%

This brings to a conclusion all of the sizes of step input yaw demand angles being considered in this study. All types and levels of LOE previously stated in Chapter 3 have been tested and the results presented in the last three sections for the most fault tolerant simulated annealing and tabu search tuned FISs.

6.5. CONCLUSIONS

The aim of this Chapter was to produce a FIS capable of handling the actuator faults discussed in Chapter 3. This was accomplished by starting with a basic FIS and tuning it, using two methods, to create a FIS capable of improving the fault tolerance of the control AUV. The two tuning methods were simulated annealing [Kirkpatrick, et *al* (1983)] and tabu search [Denna et *al* (1999)], both described in Chapter 3. The results (rise times and RMSEs) for both of these fault tolerant FISs have been presented. For comparison the results for the benchmark PD controller are also given.

The SA FIS was able to improve on all of the RMSEs of the PD controller, with percentage increases in performance ranging from 1% to 88%. The rise times of all the SA FIS's test results were below those of the benchmark controller and for eight of tests were below the ideal system rise times. This was achieved by the SA FIS forcing the AUV to turn steeply for a slightly longer period, of time, inducing a slight overshoot in some cases.

The TABU FIS also had shorter rise times for all tests compared to the PD benchmark controller, and again was able to produce a rise time less than the ideal system for sixteen of the thirty six tests. This was achieved by the TABU FIS, using the same

method employed by the SA FIS. By forcing the AUV to turn steeply for a slightly longer period of time, and so induce a slight overshoot in some cases, a shorter rise time was recorded, as seen in the results. For all but two of the RMSEs recorded the results were similar to those of the SA FIS. The two tests where a difference is clearly apparent are the rate limiter block 25% LOE and the simultaneous blocks 25% LOE, where a decrease in performance compared to the PD benchmark controller was observed. These results were not uncommon for the tabu search tuned FISs, when presented with these types and levels of LOE.

Both the actuator fault tolerant FISs presented in this Chapter are a significant improvement on the benchmark PD controllers responses. Either FIS could be placed within the control loop of the AUV and greatly improve its performance when considering actuator faults, as shown in the results, sections 6.4.3 to 6.4.5.

In order to choose which of the FISs produced a better performance the two FISs have to be directly compared. First compare the rise times of the AUV for both FISs. The SA FIS had a shorter rise time for nine of the thirty-six tests, the TABU FIS had a shorter rise time for twenty-three of the thirty-six tests, with four of the thirty-six test rise times being equal. Therefore for rise time performance the TABU FIS is clearly a better choice. Secondly compare the RMSE performance. The SA FIS had a smaller RMSE for eighteen of the thirty-six tests, the TABU FIS had a smaller RMSE eighteen of the thirty-six tests, with none of the thirty-six test RMSEs being equal. This shows that overall there is not a great difference in the FISs when considering RMSEs.

Despite the TABU FIS clearly having a large number of shorter rise times and an equal number of smaller RMSEs, the conclusion of this Chapter must be that the SA FIS would be the better of the two to use for further work. The main reasons are that the TABU FIS produced a higher RMSE, relative to the benchmark controller, for two tests and that the SA FIS produced better results for the most severe faults being considered, i.e. the 100% LOE for all types of faults at all step sizes.

A potential drawback of the fault tolerant system developed here is the need of the FIS to have the level of failure given to it via a sensor on the actuator. Chapter 7 considers a method to remove this need and still maintain an acceptable level of fault tolerance.

CHAPTER 7

ACTUATOR RECOVERY SYSTEM WITHOUT ERROR SENSOR

7.1. INTRODUCTION

Chapter 6 dealt with the design and development of a fault tolerant FIS capable of handling specified actuator faults. The FIS developed uses three inputs, namely: the control signal from the ANFIS controller, the demand being placed on the AUV, and the error between the actual and desired position of the damaged actuator. The use of the error sensor in the fault tolerant FIS could conceivably lead to a sensor fault of the types defined in Chapter 3 and considered in Chapter 5 of this thesis. A fault occurring within this sensor would lead to a loss of performance by the actuator fault tolerant FIS of Chapter 6.

To eliminate this potential problem this Chapter considers a way of removing the need for the error sensor. The error sensor used in the fault tolerant FIS is replaced by an error estimation FIS. When placed within the control loop, in addition to the fault tolerant FIS, the error estimation FIS removes the need for a sensor to be placed on the actuator. This further increases the fault tolerant performance of the overall control system. In this case the overall control system is defined as the ANFIS controller [Craven (1999)] working in conjunction with both the actuator fault tolerant FIS, developed in Chapter 6, and the novel error estimation FIS developed herein.

7.2. REPLACING THE ERROR SENSOR

The effects of sensor faults have been explored within this thesis (Chapter 4 and Chapter 5). Removing the need for a sensor to give the error in the position of the actuator will make the overall control system more fault tolerant. To this end a novel estimation device will is designed to replace the actuator error sensor. The estimation FIS will be placed within the overall control structure as shown in Figure. 7.1.

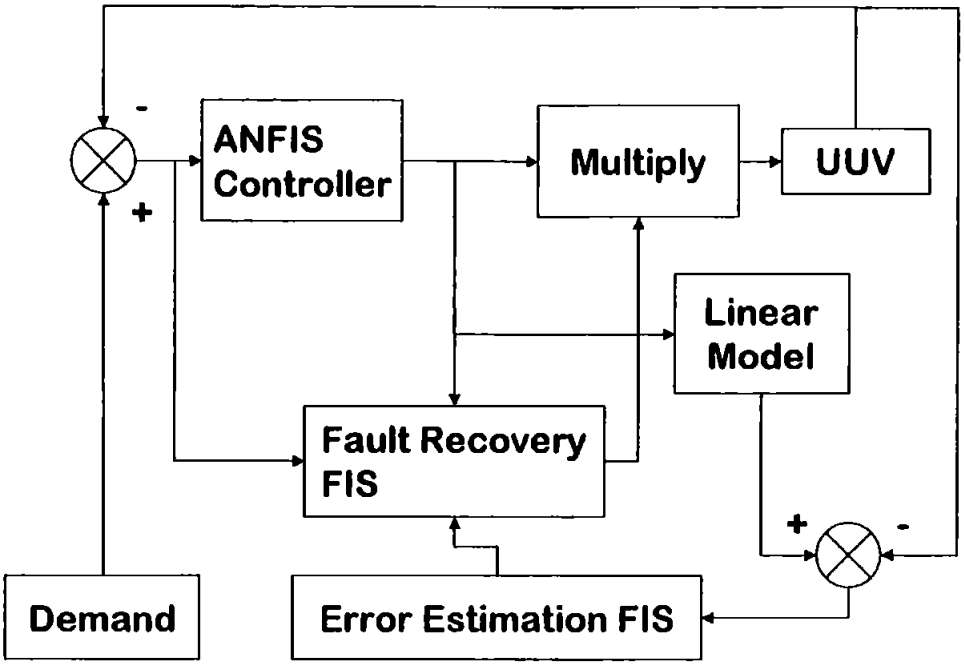


Figure 7.1 The Actuator Fault Tolerant System

The fault tolerant control system shown in Figure 7.1 is a natural progression of the control structure shown in Figure 6.1. For this Chapter the fault tolerant FIS developed in Chapter 6 will be used in the block labelled 'Fault Recovery FIS'. The conclusion of Chapter 6 is that the most effective fault tolerant FIS was shown to be the simulated annealing tuned FIS which was tuned for 500 epochs and hence will be used in this Chapter.

As can be seen in Figure 7.1 the error estimation FIS has only one input and one output. As only one input is required for the FIS, then it is possible to increase the number of fuzzy rules used. An increase in the number of fuzzy rules should lead to an increase in performance of the estimator FIS. The one input to be used by the estimation FIS is that of the modular difference between the linear model's yaw rate and the actual yaw rate of the AUV. The yaw rate was selected because there can only be an error in the actuator position when the AUV is altering yaw angle and hence when a yaw rate is evident. The minimum value used is zero degrees per second due to the fault recovery FIS needing only a modular input. The maximum value of the input is 8.72 degrees per second. This value was the maximum difference detected

from the open loop training data. The input membership functions for the heuristic case where only three fuzzy rules were used can be seen in Figure 7.2.

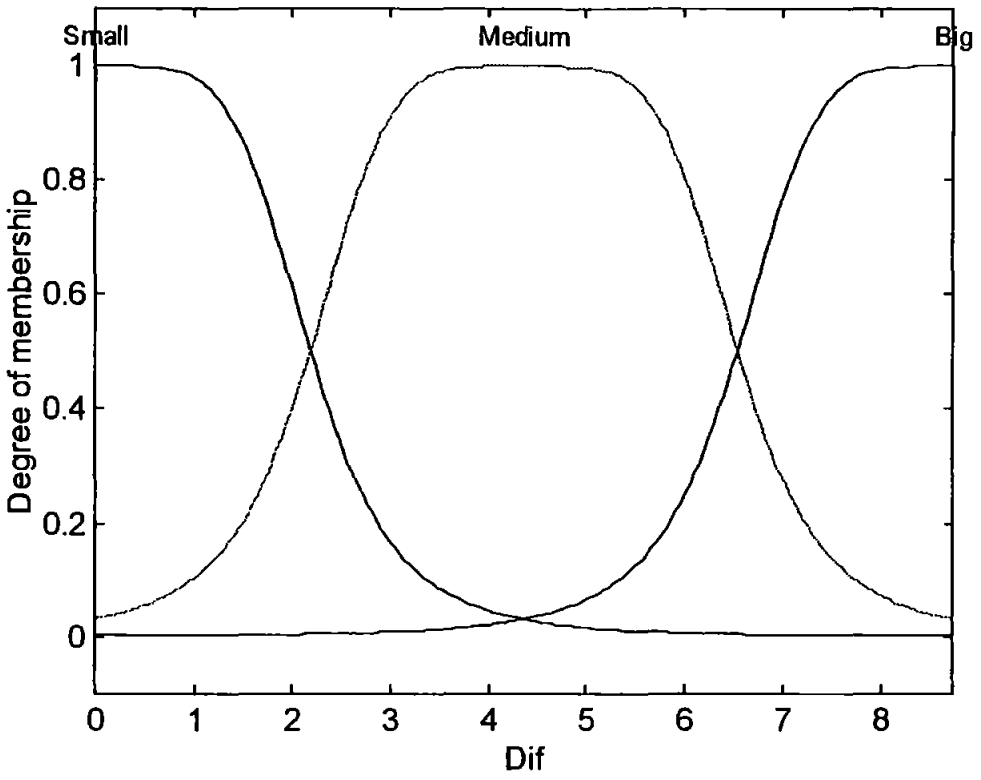


Figure 7.2 Heuristic Error Estimation FIS Input Membership Functions

The output functions associated with the same system is that shown in Figure 7.3. It has a minimum value of zero degrees and a maximum value of 30.99 degrees. As only a modular value is required by the fault recovery FIS a minimum value of zero degrees is the optimal value. However the fault recovery FIS requires a maximum value of only 25.2 degrees for reasons given in Chapter 6, this is achieved by placing a saturation block in-between the estimation FIS and the fault recovery FIS. The high value of 30.99 degrees is used by the estimation FIS because of a high error detected in the open loop training data when a LOE of 60% were considered.

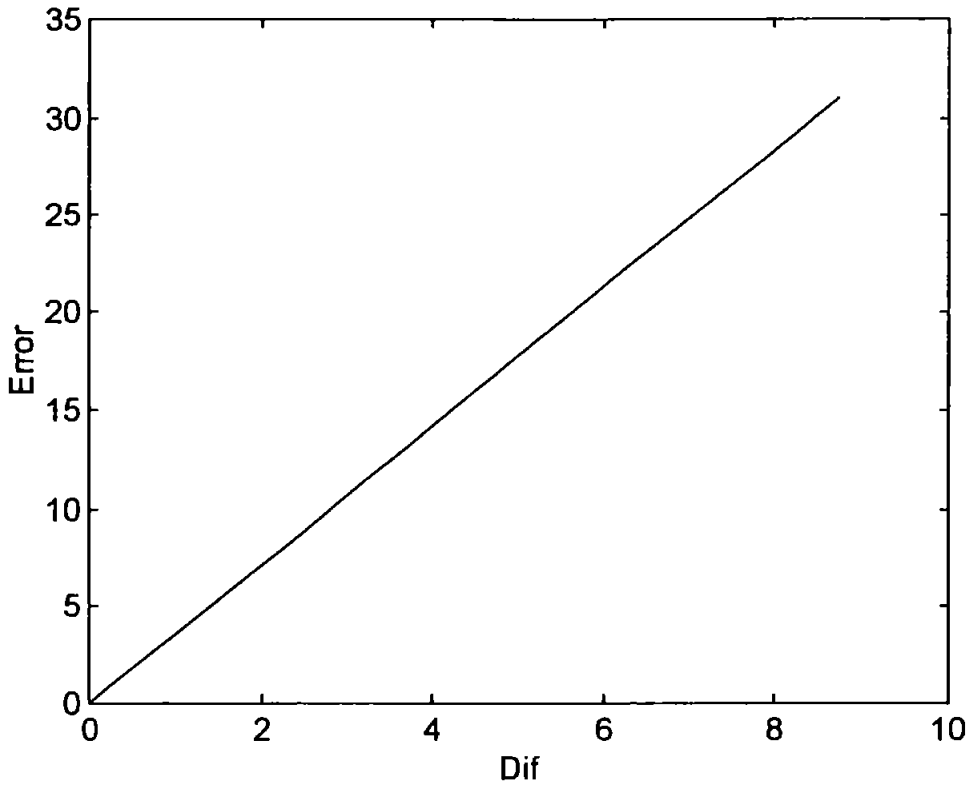


Figure 7.3 Heuristic Error Estimation FIS Output Line

The input and output membership functions given lead to the simple heuristic fuzzy rule base shown in Equation 7.1.

$$\begin{aligned} &\text{If Dif is Small} && \text{then Error} = 3.5539\,\text{Dif} + 0 \\ &\text{If Dif is Medium} && \text{then Error} = 3.5539\,\text{Dif} + 0 \\ &\text{If Dif is Big} && \text{then Error} = 3.5539\,\text{Dif} + 0 \end{aligned} \tag{7.1}$$

Where Dif is the modular difference between the linear model yaw rate and the AUV yaw rate, and Error is the modular difference in the desired and actual position of the damaged actuator controlling the upper canard.

The three rules fuzzy system described has the general principals of all of the FISs being used in this Chapter. The maximum and minimum values do not alter for any of the FISs being considered. The only differences being in the number of rules used and the number of epochs a FIS has been trained.

7.3. FUZZY TUNING

The FIS presented is a general representation of the FISs being used to estimate the error sensor information. The FISs used were derived and tuned using input-output data from the given AUV model. As the estimating of the information is akin to modelling the dynamics of the actuator then the information used must be obtained in the open loop system set-up. To obtain the data an identical random demand was placed simultaneously on both the AUV and the linear yaw model. The linear yaw model used herein is that derived in Chapter 3. The random demand data was created using a random number generator in the Matlab/Simulink environment. Preliminary tests showed that when the upper canard was damaged, in the ways being considered, the AUV became unstable after a short period of time in an open loop system. The problem was removed by running twenty short programs of 50 seconds instead of one long program with a different random number seed value used in every 50 seconds run. The random input data used was created using the white noise generator in MATLAB/Simulink, a noise power of 1.7 and a sample time of twenty seconds, along with a different seed value for each of the twenty short programs. The values of the seed used were that of the numbers 1 to 20. This created a random input data path lasting 1000 seconds. This input data path can be seen in Figure 7.4 and the open loop yaw rate response of the fault free AUV can be seen in Figure 7.5.

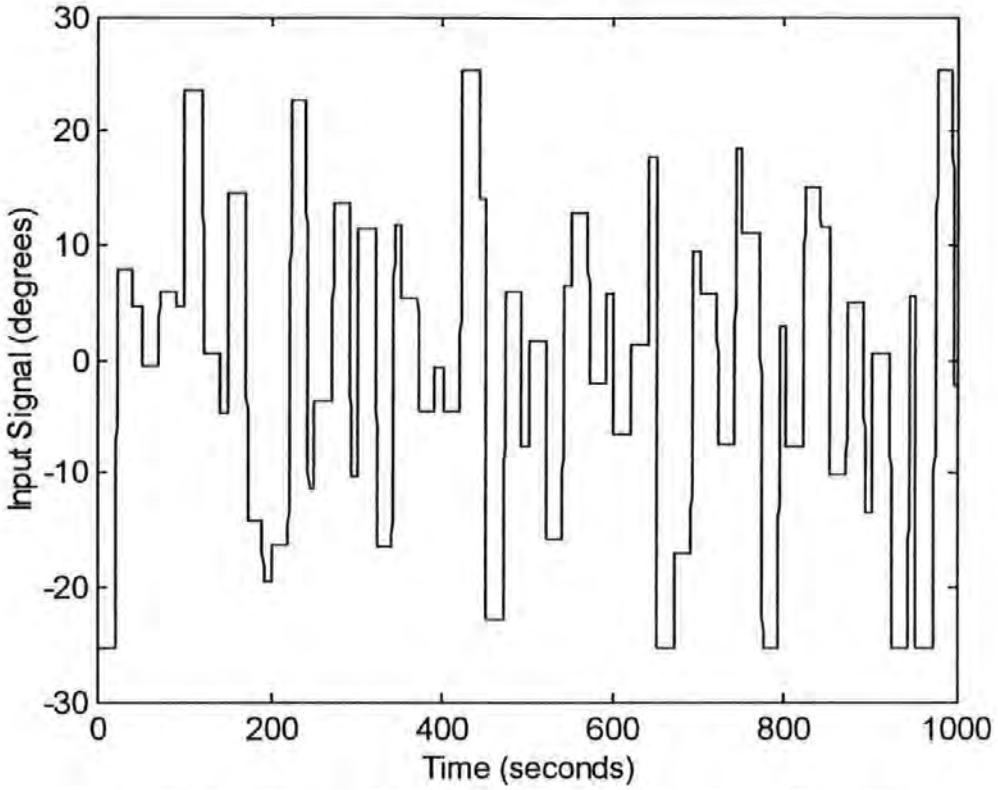


Figure 7.4 Error Estimation FIS Random Input Signal

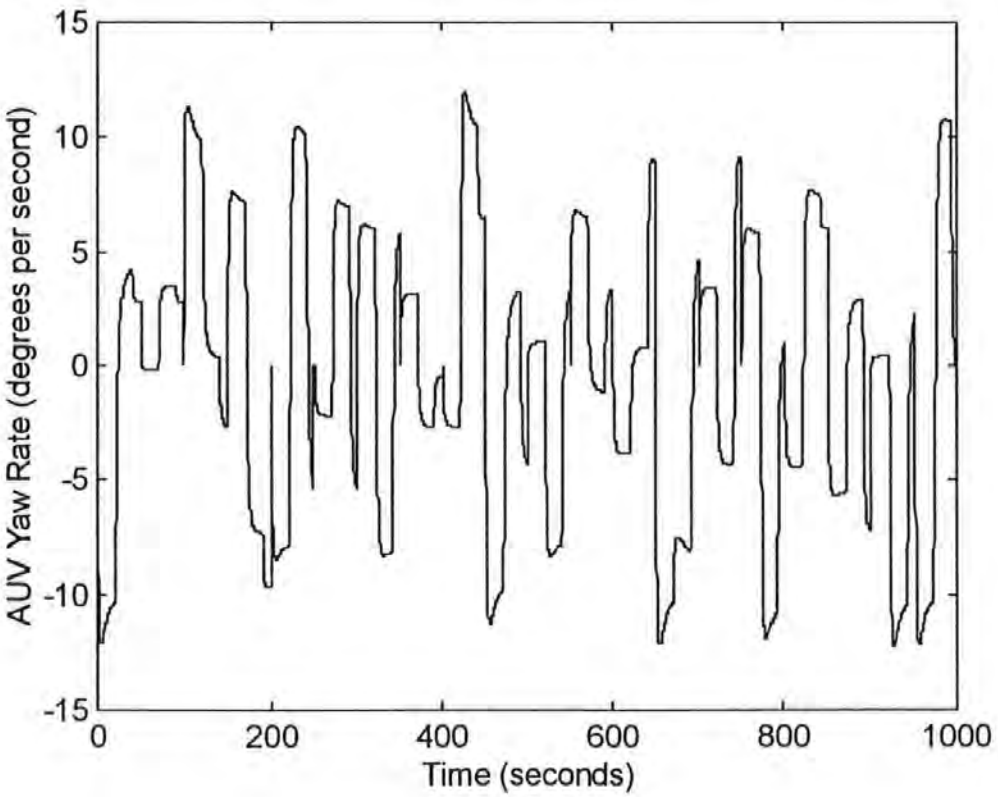


Figure 7.5 AUV Yaw Rate Response

Presenting this data to the AUV and linear models in an open loop system stimulated the complete range of responses from the systems. As the estimator FISs would need to handle all levels of faults six different levels of LOE (0%, 20%, 40%, 60%, 80% and 100%) in the rate limiter and saturation blocks were induced and the system was given the same random demand data. The information was recorded every one tenth of a second to create a vector of length 60120. The input data was that of the difference between the linear models yaw rate and the actual AUV yaw rate and can be seen in Figure 7.6. The output data was the recorded error in the actuator controlling the upper canard and can be seen in Figure 7.7.

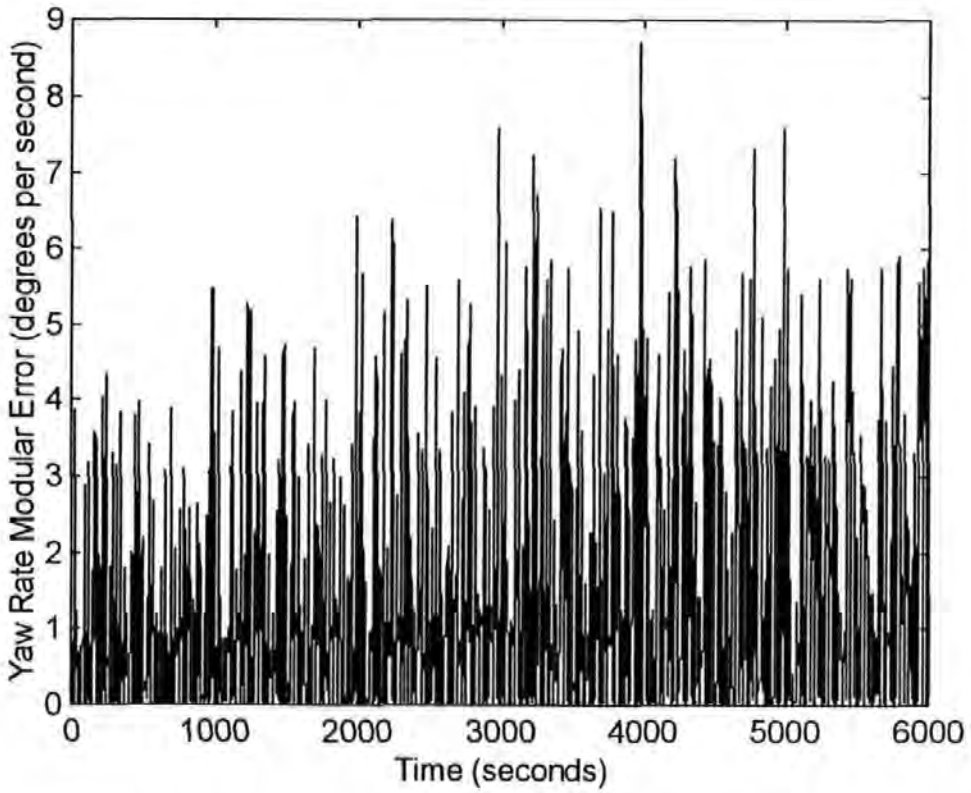


Figure 7.6 The Modular Difference Between The Linear Model and AUV Yaw Rates

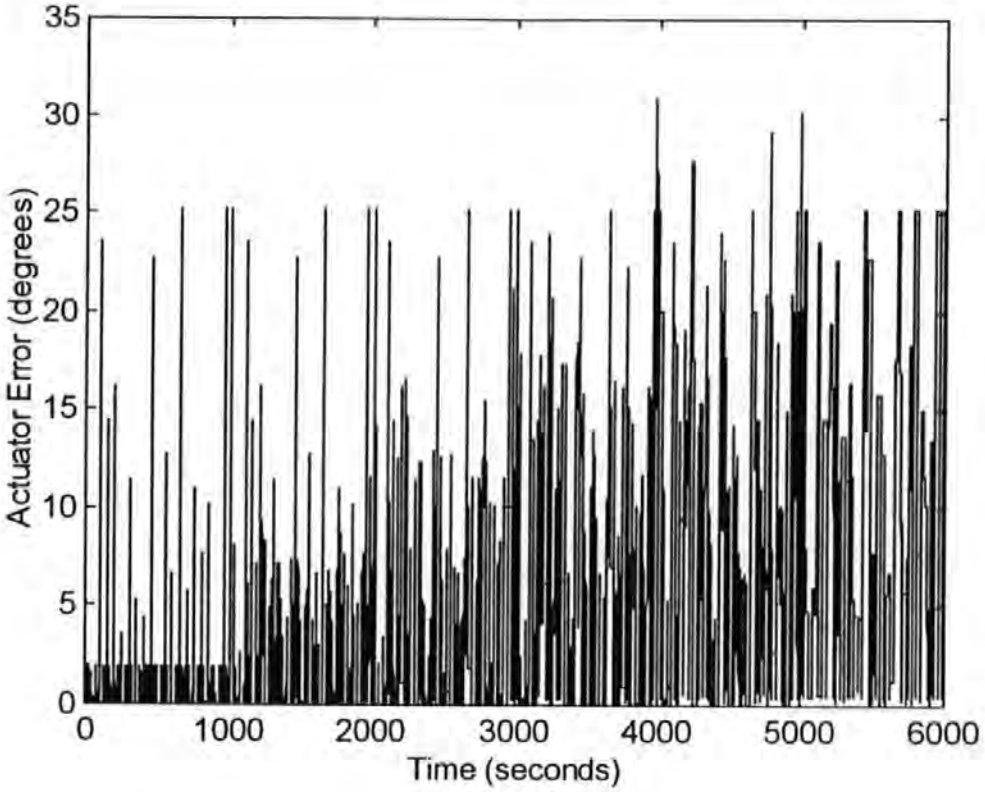


Figure 7.7 The Modular Error in The Actuator Position

The input-output data shown was then used to create and tune the error estimator FISs.

7.3.1. ANFIS

As the required input-output data pairs are available for this tuning process the ANFIS [Jang (1991)] tuning method will be used. As the FIS has only one input, which greatly reduces the computational time necessary for each tuning epoch, it is possible to consider larger numbers of fuzzy rules. Therefore in this Chapter FISs with rule bases using 3, 5, 7, and 9 rules have been tuned for a maximum of 800 epochs, with the FIS being recorded after each 100 epochs. The input membership functions, output membership functions and the fuzzy rule bases associated with the most effective of each size of FIS will now be presented.

The results for these four FISs will now be presented and then compared in the next section.

7.4. RESULTS

The results of the tuned FISs for each size of rule base are now presented. The results for all the levels of all the faults for all levels of demanded yaw inputs being considered are shown. All faults are those given in Section 3.4.2 of this thesis.

7.4.1. The Three Rules FIS

First let the three rules case be considered. After all of the tests had been performed and the results had been compared and correlated the most accurate three rules FIS tuned is the FIS tuned for 800 epochs.

The input membership functions for this FIS is shown in Figure 7.8. The functions have been tuned by the ANFIS method using the data in Figure7.6. Most of the data of low value hence when there are only three membership function, it is necessary to move them to the left of the graph as shown.

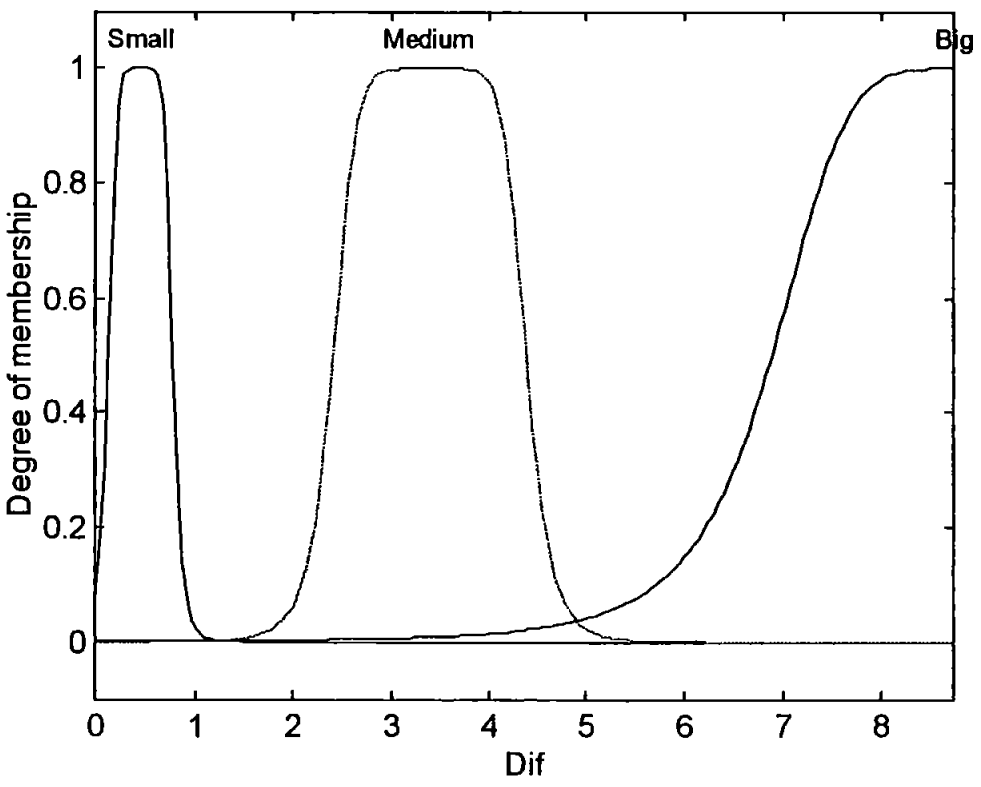


Figure 7.8 The Three Rules FIS Input Membership Functions

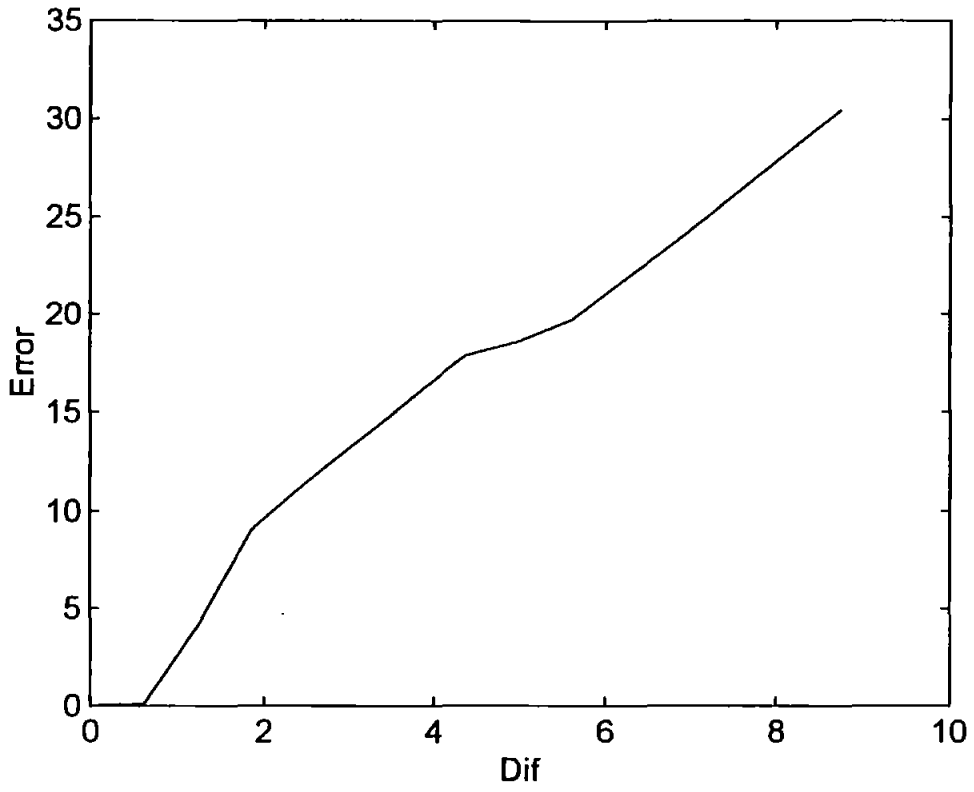


Figure 7.9 The Three Rules FIS Output Line

The output line for the tuned FIS is shown in Figure 7.9 and along with the input membership functions lead to the fuzzy rule base shown in Equation 7.2.

$$\begin{aligned} &\text{If Dif is Small then} && \text{Error} = 0.213\,\text{Dif} + 0 \\ &\text{If Dif is Medium then} && \text{Error} = 3.492\,\text{Dif} + 2.696 \\ &\text{If Dif is Big then} && \text{Error} = 3.425\,\text{Dif} + 0.483 \end{aligned} \tag{7.2}$$

Where Dif is the modular difference between the linear model yaw rate and the AUV yaw rate, and Error is the modular difference in the desired and actual position of the faulty actuator.

7.4.2. The Five Rules FIS

Next let the five rules case be considered. After all of the tests had been performed and the results had been compared and correlated the most accurate five rules FIS tuned was the FIS tuned for 800 epochs.

The input membership functions for this FIS is shown in Figure 7.10. The functions have been tuned by the ANFIS method using the data in Figure7.6. As was the case for the three rules FIS, it is again necessary to move the membership functions to the left hand side of the graph, so as to cover the data in a more effective way.

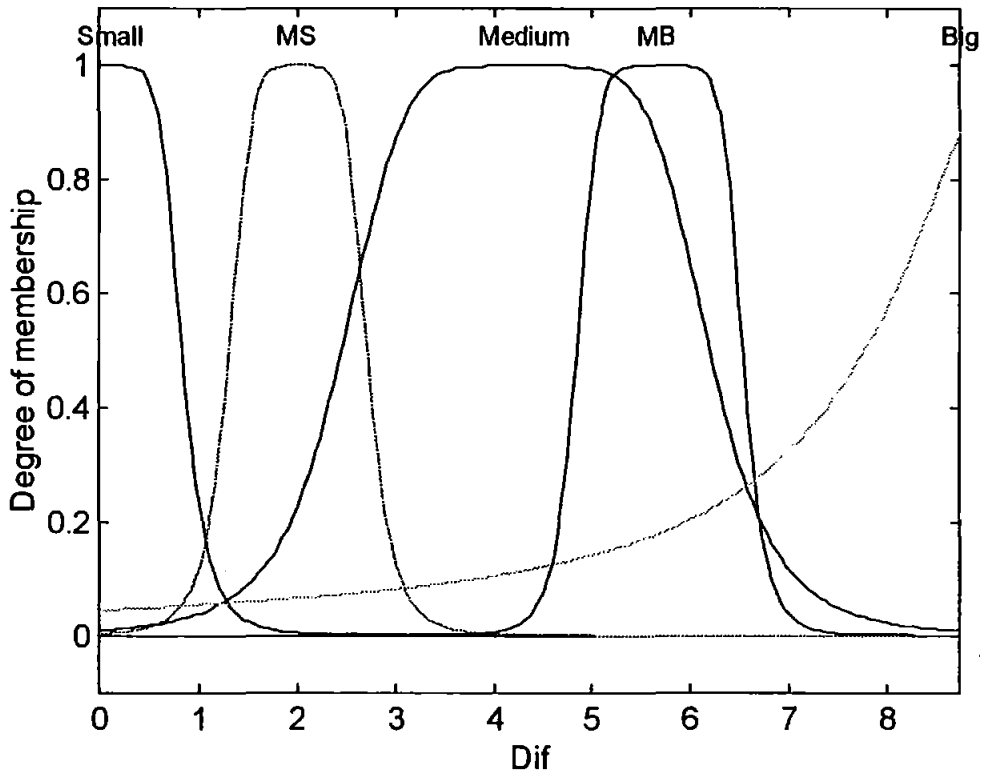


Figure 7.10 The Five Rules FIS Input Membership Functions

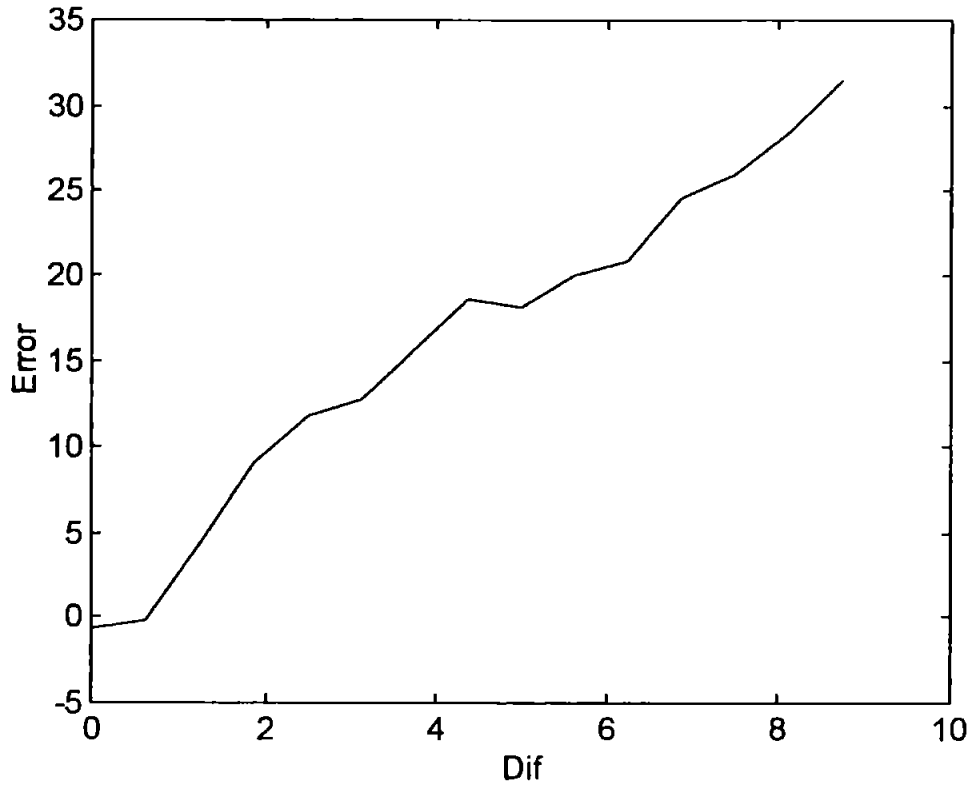


Figure 7.11 The Five Rules FIS Output Line

The output line for the tuned FIS is shown in Figure 7.11 along with the input membership functions lead to the fuzzy rule base shown in Equation 7.3.

$$\begin{aligned}
&\text{If Dif is Small then} && \text{Error} = 0.466\,\text{Dif} + 0 \\
&\text{If Dif is MS then} && \text{Error} = 5.513\,\text{Dif} + 0.190 \\
&\text{If Dif is Medium then} && \text{Error} = 5.367\,\text{Dif} - 3.388 \\
&\text{If Dif is MB then} && \text{Error} = 3.454\,\text{Dif} - 4.597 \\
&\text{If Dif is Big then} && \text{Error} = 5.273\,\text{Dif} - 14.637
\end{aligned} \tag{7.3}$$

Where Dif is the modular difference between the linear model yaw rate and the AUV yaw rate, MS is medium small, MB is medium big and Error is the modular difference in the desired and actual position of the faulty actuator.

7.4.3. The Seven Rules FIS

Next let the seven rules case be considered. After all of the tests had been performed and the results had been compared and correlated the most accurate seven rules FIS tuned was the FIS tuned for 800 epochs.

The input membership functions for this FIS is shown in Figure 7.12. The functions have been tuned by the ANFIS method using the data in Figure7.6. Now that the number of membership functions has increased to seven each one is not required to cover the input function space as much, which leads to Figure 7.12.

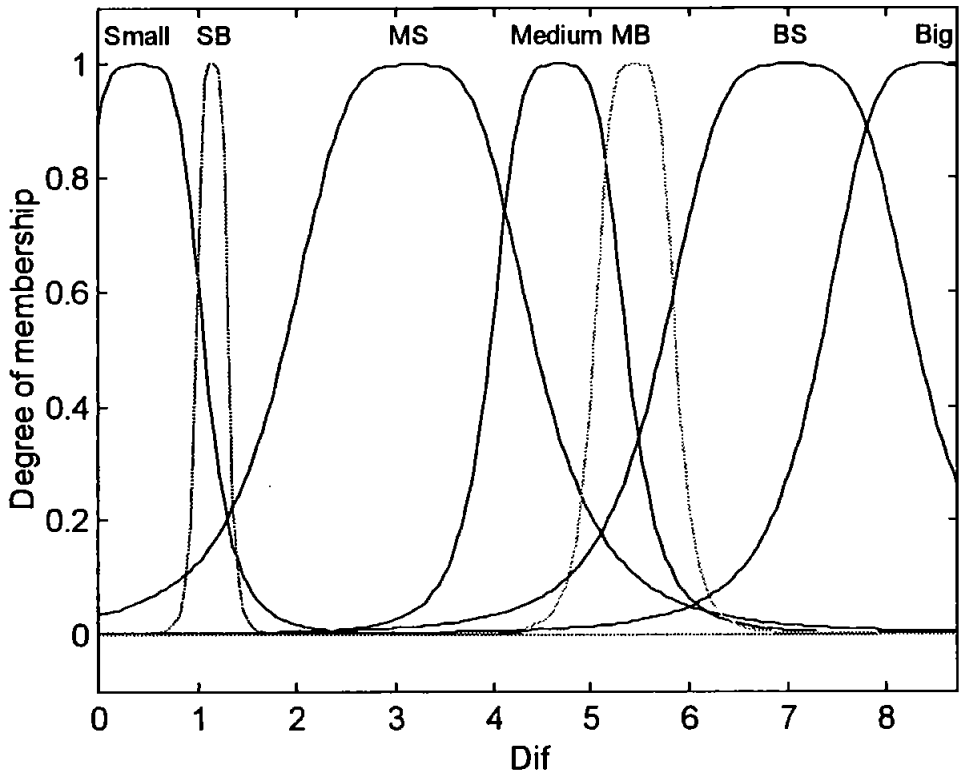


Figure 7.12 The Seven Rules FIS Input Membership Functions

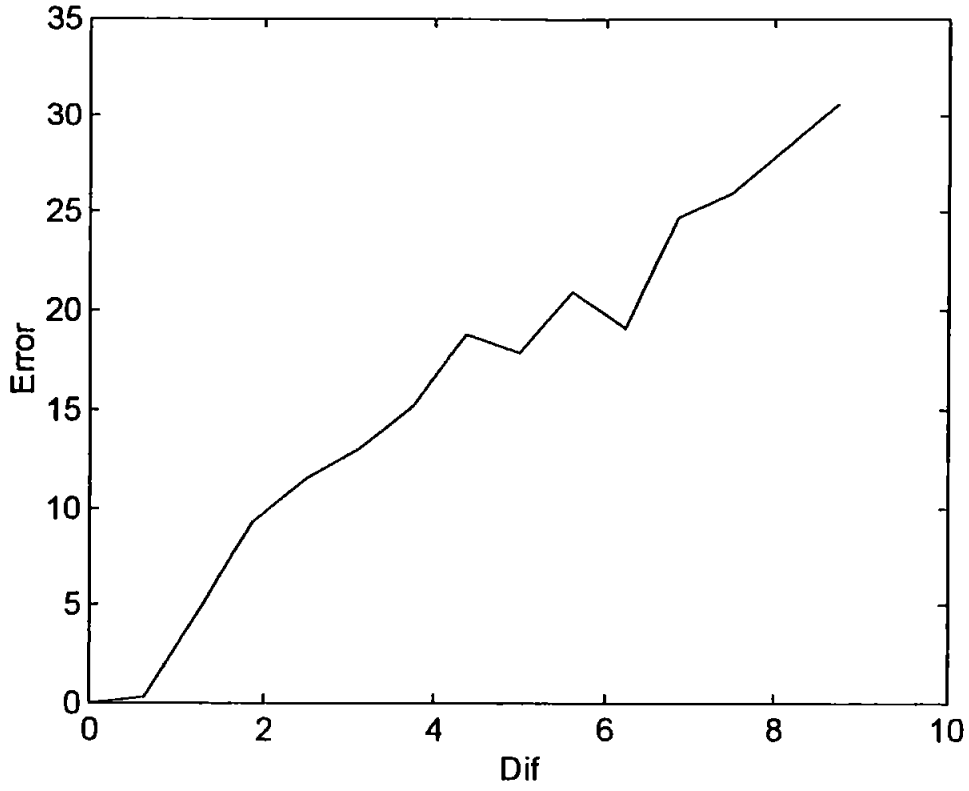


Figure 7.13 The Seven Rules FIS Output Line

The output line for the tuned FIS is shown in Figure 7.13 along with the input membership functions lead to the fuzzy rule base shown in Equation 7.4.

$$
\begin{aligned}
&\text{If Dif is Small then} && \text{Error} = 0.025\,\text{Dif} + 0 \\
&\text{If Dif is SB then} && \text{Error} = 16.464\,\text{Dif} - 15.228 \\
&\text{If Dif is MS then} && \text{Error} = 2.613\,\text{Dif} + 5.656 \\
&\text{If Dif is Medium then} && \text{Error} = 2.723\,\text{Dif} + 10.714 \\
&\text{If Dif is MB then} && \text{Error} = 23.892\,\text{Dif} - 107.185 \\
&\text{If Dif is BS then} && \text{Error} = 19.655\,\text{Dif} - 101.314 \\
&\text{If Dif is Big then} && \text{Error} = 20.282\,\text{Dif} - 156.605
\end{aligned}
\tag{7.4}
$$

Where Dif is the modular difference between the linear model yaw rate and the AUV yaw rate, MS is medium small, MB is medium big, SB is small big, BS is big small and Error is the modular difference in the desired and actual position of the faulty actuator.

7.4.4. The Nine Rules FIS

Finally let the nine rules case be considered. After all of the tests had been performed and the results had been compared and correlated the most accurate nine rules FIS tuned was the FIS tuned for 700 epochs.

The input membership functions for this FIS is shown in Figure 7.14. The functions have been tuned by the ANFIS method using the data in Figure7.6. Due to the number of membership functions now being used by the FIS there is little need for the ANFIS method to alter the functions greatly. The input space is covered effectively for this FIS as shown in Figure 7.14.

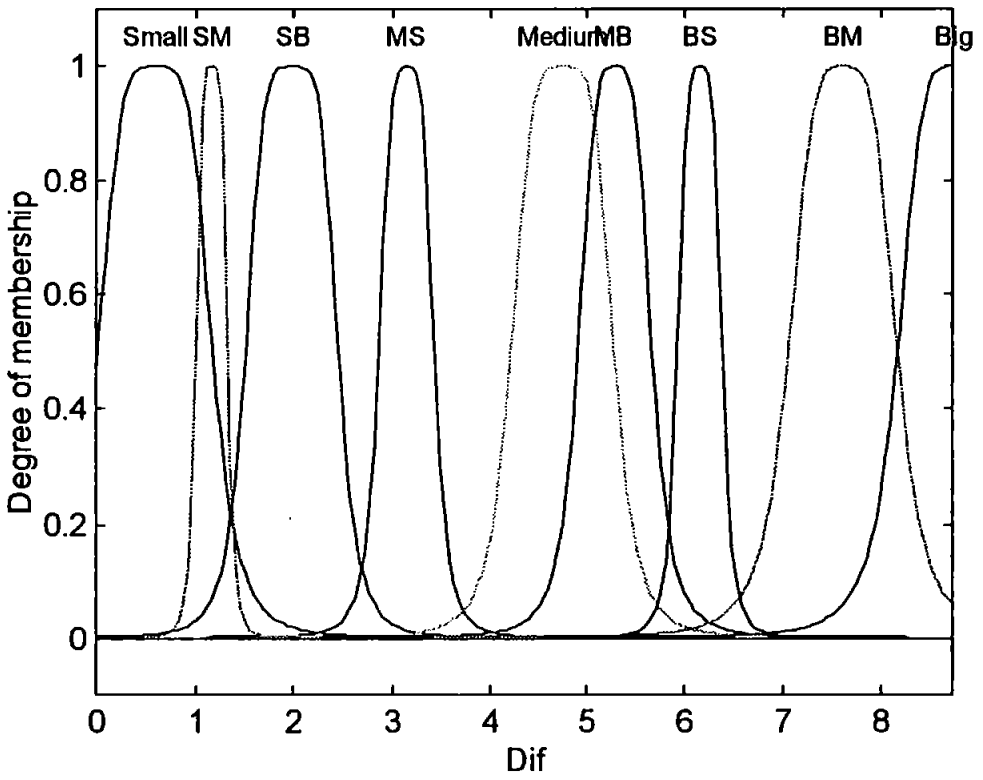


Figure 7.14 The Nine Rules FIS Input Membership Functions

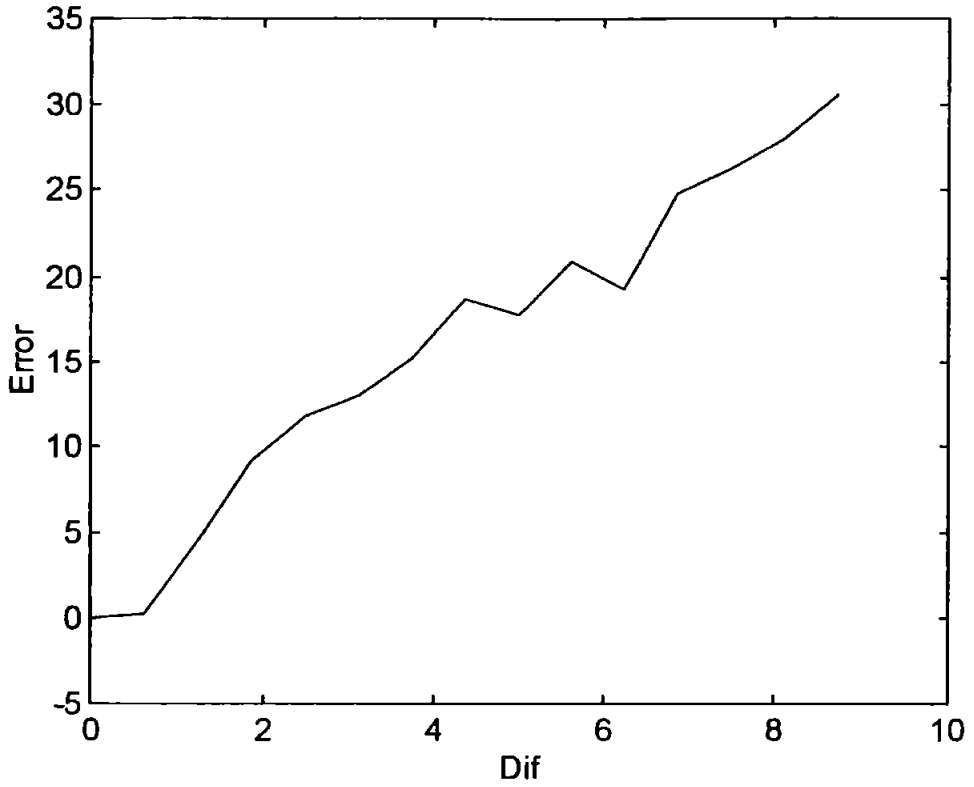


Figure 7.15 The Nine Rules FIS Output Line

The output line for the tuned FIS is shown in Figure 7.15 along with the input membership functions lead to the fuzzy rule base shown in Equation 7.5.

$$
\begin{aligned}
&\text{If Dif is Small then} && \text{Error} = 0.392\,\text{Dif} + 0 \\
&\text{If Dif is SM then} && \text{Error} = 17.255\,\text{Dif} - 15.679 \\
&\text{If Dif is SB then} && \text{Error} = 4.157\,\text{Dif} + 1.630 \\
&\text{If Dif is MS then} && \text{Error} = 3.979\,\text{Dif} + 0.572 \\
&\text{If Dif is Medium then} && \text{Error} = 4.236\,\text{Dif} + 0.852 \\
&\text{If Dif is MB then} && \text{Error} = 14.058\,\text{Dif} - 58.295 \\
&\text{If Dif is BS then} && \text{Error} = 3.404\,\text{Dif} - 2.359 \\
&\text{If Dif is BM then} && \text{Error} = 2.473\,\text{Dif} + 7.726 \\
&\text{If Dif is Big then} && \text{Error} = 3.826\,\text{Dif} - 2.733
\end{aligned}
\tag{7.5}
$$

Where Dif is the modular difference between the linear model yaw rate and the AUV yaw rate, MS is medium small, MB is medium big, SM is small medium, SB is small big, BS is big small, BM is big medium and Error is the modular difference in the desired and actual position of the faulty actuator.

The four FISs given are now tested on the same testing paths as used to determine the most fault tolerant actuator recovery FIS in Chapter 6. Again both the rise times of the AUV and the RMSEs will be used to compare the performance of the FIS. As the actuator fault recovery SA FIS is being used as a basis for this work, its results will be the benchmark for the testing. Therefore if the error estimator FIS perfectly recreates the error in the actuator the overall system will give identical results to those obtained for the simulated annealing tuned actuator recovery FIS in Chapter 6. To measure this the percentage decrease in performance has also been calculated for each test. As the testing of the four FISs over 36 different simulations gave 144 results, not all will be shown within this work, however for the interested reader the complete results are presented in Appendix J. Also both the upper and lower canard responses have been recorded and are presented in Appendix K.

7.4.5. Yaw Step Inputs of 10 Degrees

This first section will look at LOEs for the 10 degrees yaw step input demands sets of tests. The three types of faults defined in Chapter 3 will all be considered. The FISs have been compared using RMSEs and rise times as defined previously in Chapter 6. The complete set of results (RMSEs and rise times) for all four estimator FISs described, along with the results for the system when the error sensor is used, are shown in Tables 7.1 and 7.2.

When examining the results for the RMSEs of the four error estimator FISs it is clear to see how similar the results are, with only a 0.008 degrees change being the largest difference recorded. For the saturation fault identical results are recorded for both the 25% and 50% LOE levels, with the three and five rules FISs having the lowest values. The lowest values for the 75% LOE level was 0.079 degrees and was produced by the seven rules FIS. While the three rules FIS produced the lowest recorded RMSE for the 100% LOE test. When the fault was implemented in the rate limiter block the seven rules FIS produced the lowest RMSEs for both the 25% and 75% LOE tests. The three rules FIS produced the lowest RMSEs for the 50% and 100% LOE tests. These results were repeated when the faults occurred in both blocks, with the three rules FIS producing the lowest RMSEs for the 50% and 100% LOE tests and the seven rules FIS producing the lowest RMSEs for both the 25% and 75% LOE tests.

Table 7.1 The 10 Degrees Step Input RMSEs.

Controller	RMSEs (degrees)											
	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	0.028	0.028	0.072	0.782	0.053	0.142	0.294	0.782	0.053	0.142	0.294	0.782
3 Rules FIS	0.023	0.023	0.081	0.901	0.056	0.143	0.312	0.901	0.056	0.143	0.312	0.901
5 Rules FIS	0.023	0.023	0.080	0.902	0.055	0.144	0.311	0.902	0.055	0.144	0.311	0.902
7 Rules FIS	0.024	0.024	0.079	0.902	0.054	0.150	0.310	0.902	0.054	0.150	0.310	0.902
9 Rules FIS	0.024	0.024	0.080	0.902	0.055	0.151	0.311	0.902	0.055	0.151	0.311	0.902

When considering the rise times recorded from these tests it is clear, from Table 7.2, of the performance of each of the FISs. All four of the FISs were able to estimate the actuator error to a level where the overall system was capable of reproducing the rise times identically for all faults apart from the 100% LOE tests. When the error estimation FISs were used an increase in the rise times for the 100% LOE tests of 0.4 seconds to 6.8 seconds were recorded for every FIS and every type of fault. When considering the rise times for this size of step input there is no difference recorded between any of the four FISs.

Table 7.2 The 10 Degrees Step Input Rise Times.

Controller	Rise Times (seconds)											
	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	4.5	4.5	4.6	6.4	4.5	4.6	4.6	6.4	4.5	4.6	4.6	6.4
3 Rules FIS	4.5	4.5	4.6	6.8	4.5	4.6	4.6	6.8	4.5	4.6	4.6	6.8
5 Rules FIS	4.5	4.5	4.6	6.8	4.5	4.6	4.6	6.8	4.5	4.6	4.6	6.8
7 Rules FIS	4.5	4.5	4.6	6.8	4.5	4.6	4.6	6.8	4.5	4.6	4.6	6.8
9 Rules FIS	4.5	4.5	4.6	6.8	4.5	4.6	4.6	6.8	4.5	4.6	4.6	6.8

Tables 7.1 and 7.2 have shown how similar the performances of the four FISs are when presented with the faults being considered. This is further shown in Figure 7.16 where the responses of the AUV can be seen for all four error estimation FISs and for

the control system when the error sensor is used for the 50% LOE level of fault in both blocks.

It is clear from Figure 7.16 how the overall system has been affected by replacing the error sensor with the estimation FIS.

The results for all four FISs in Figure 7.16 are very similar, therefore the error traces between each estimation FISs response and the response of the control system with the error sensor are displayed in Figure 7.17.

Figure 7.17 shows that all four error estimation FISs effects on the system are near identical. These results are typical, for this size of step input showing very little change of performance between each of the four error estimation FISs, of those shown in Appendix J.

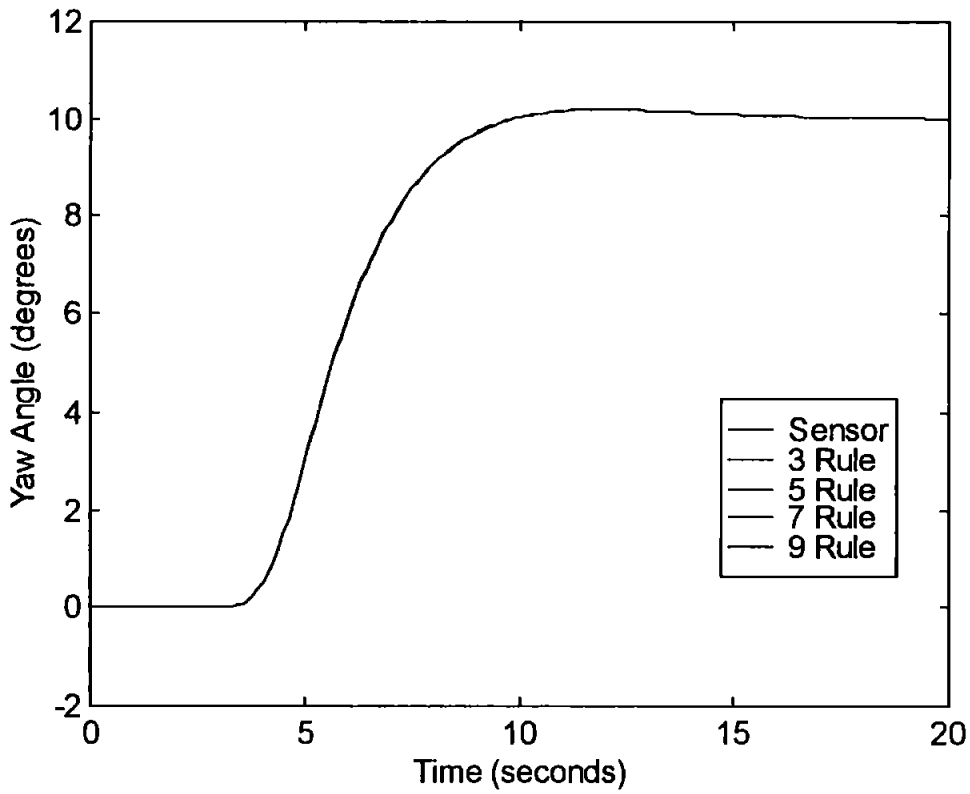


Figure 7.16 The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 10 degrees

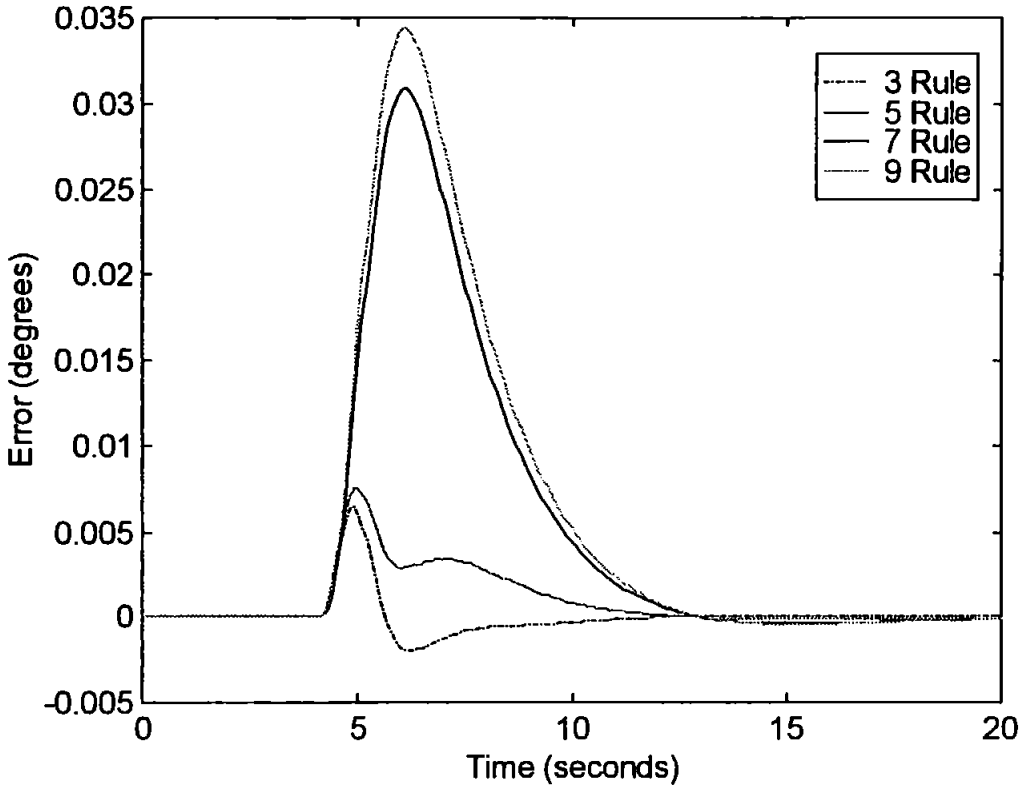


Figure 7.17 The Error Traces for The 50% LOE in Both Blocks for a Yaw Step Input of 10 degrees

Figures 7.18 and 7.19 show the upper and lower canard responses for the three rules FIS when a fault is occurring simultaneously in both the saturation and rate limiter blocks.

Figure 7.18 shows the upper canard responses to the control system for increasing levels of LOE. It is clear the effect the saturation fault has on the canard, with the decrease in gradient as the fault increases. There is no indication of the effect the rate limiter fault has at this size of step input.

Figure 7.19 shows the lower canard responses to the same faults. This canard has not been damaged and the system attempts to compensate for the faulty canard by increasing the angle it achieves.

The results when the other three FISs are used in the control system are presented in Appendix K

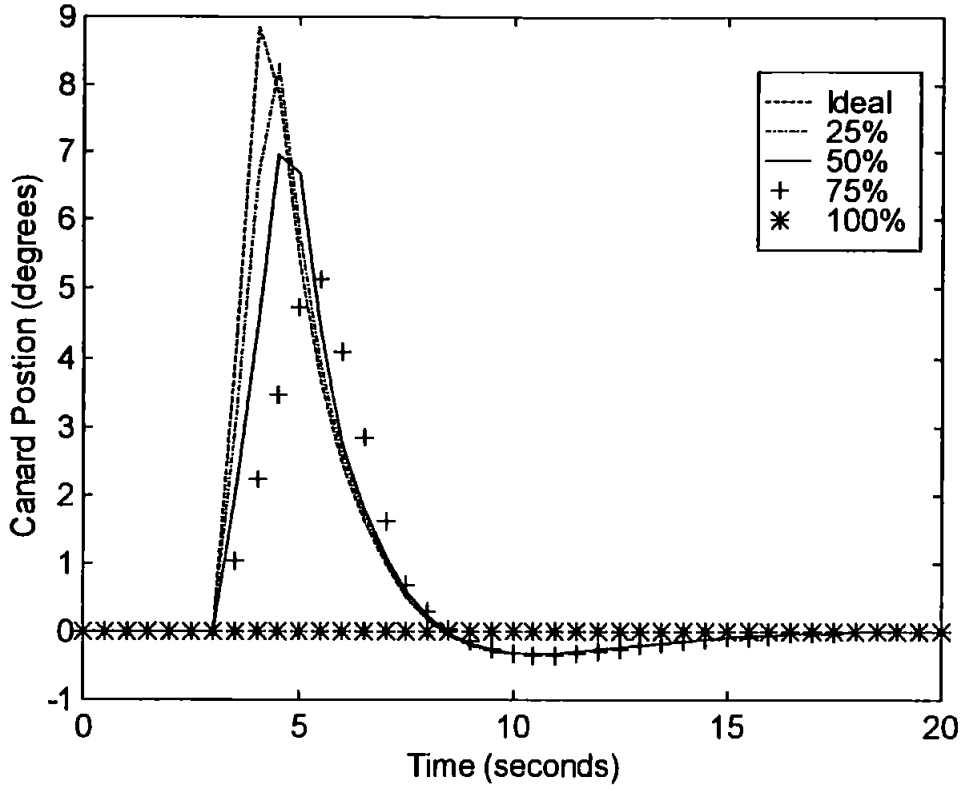


Figure 7.18 Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 10 degrees

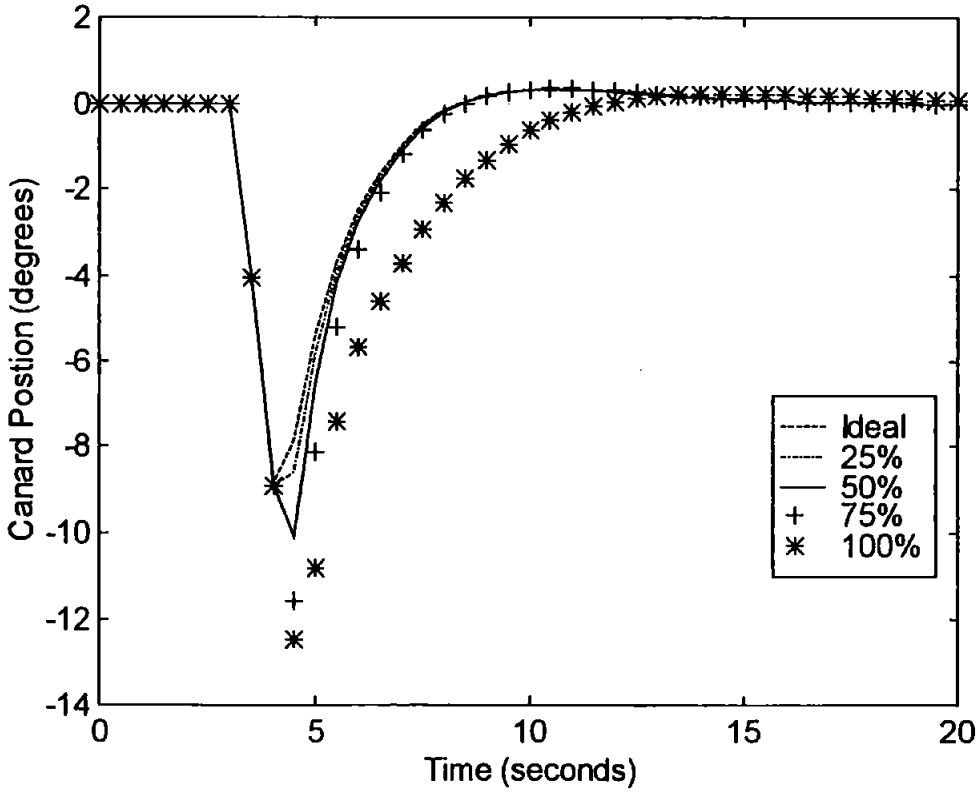


Figure 7.19 Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 10 degrees

To show the change in performance of the AUV for each of the error estimation FISs the percentage decrease in performance for each test, relative to the control system using the error sensor, has been calculated in Table 7.3.

The replacement of the error sensor with an error estimation FIS was expected to lead to a decrease in performance of the AUV. This is the case for ten of the twelve tests at this step size. However for the 25% and 50% LOE levels in the saturation block fault the performance has been increased by the use of an error estimation FIS.

Table 7.3 The 10 Degrees Step Input Percentage Decrease.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
3 Rules FIS	-17.1%	-17.1%	12.4%	15.3%	5.1%	0.4%	6.0%	15.3%	5.1%	0.4%	6.0%	15.3%
5 Rules FIS	-19.3%	-19.3%	11.7%	15.4%	4.5%	1.1%	5.8%	15.4%	4.5%	1.1%	5.8%	15.4%
7 Rules FIS	-13.6%	-13.6%	10.1%	15.3%	1.5%	5.3%	5.4%	15.3%	1.5%	5.3%	5.4%	15.3%
9 Rules FIS	-15.0%	-15.0%	10.7%	15.3%	2.8%	6.2%	5.6%	15.3%	2.8%	6.2%	5.6%	15.3%

7.4.6. Yaw Step Inputs of 20 Degrees

Next increase the yaw input demand to 20 degrees and retest all four FISs.

The complete set of results (RMSEs and rise times) for all four estimator FISs described, along with the results for the control system when the error sensor is used, are shown in Tables 7.4 and 7.5.

When considering the RMSE results for this level of yaw input demand, it is again clear from the Table 7.4 that the four estimator FISs performed to a similar standard. There is a larger variance in results for this set of tests, with the biggest difference recorded in any one type of test being 0.029 degrees.

For the saturation block faults the three rules FIS produced the lowest RMSE value for both the 25% and 50% LOE levels. The five and seven rules FIS both produced the lowest RMSE value for the 75% LOE level and the seven rules FIS also had the lowest RMSE value for the 100% LOE level test. For the rate limiter block faults all four FIS produced an identical result for the 25% LOE level test. When examining the 50% LOE level results it is seen that three of the FISs (five, seven and nine rules)

have identically the lowest RMSEs. The results for the 75% and 100% LOE level tests shown a single FIS produce the lowest RMSE for each, being the three rules FIS and seven rules FIS respectively. When the faults were implemented in both blocks identical RMSEs were recorded for all four FISs for the 25%, 50% and 100% LOE levels tests. For the 75% LOE level tests the three rules FIS produce a RMSE 0.001 degrees less than the other three FISs.

Table 7.4 The 20 Degrees Step Input RMSEs.

	RMSEs (degrees)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	0.023	0.111	0.694	1.678	0.203	0.531	1.029	1.678	0.203	0.531	1.072	1.678
3 Rules FIS	0.125	0.116	0.697	2.153	0.213	0.506	1.037	2.153	0.213	0.506	1.086	2.153
5 Rules FIS	0.127	0.117	0.695	2.133	0.213	0.504	1.039	2.133	0.213	0.504	1.087	2.133
7 Rules FIS	0.130	0.117	0.695	2.124	0.213	0.504	1.038	2.124	0.213	0.504	1.087	2.124
9 Rules FIS	0.128	0.117	0.696	2.129	0.213	0.504	1.038	2.129	0.213	0.504	1.087	2.129

When considering the rise times recorded from these tests, it is clear from Table 7.5 of the performance of each of the FISs. All four of the FISs were able to estimate the actuator error to a level where the overall system was capable of reproducing the rise times identically for the 75% LOE tests for both the rate limiter and both faults. When the error estimation FISs were used an increase in the rise times for the 100% LOE was recorded. An increase of 1 second was recorded for the three and five rules FISs and an increase of 0.9 seconds for the seven and nine rules FISs for every type of fault at the 100% level LOE. For the remaining twenty-eight tests decreases in rise times were recorded. For the 25% LOE tests all FISs performed identically producing rise times of 4.6 seconds, 0.2 seconds less than when the error sensor was used. For the saturation 50% LOE fault all four FISs produced rise times 0.2 seconds less than the when the error sensor was used and for the 75% LOE all four FISs produced rise times 0.1 seconds less than the standard result. For the rate limiter 50% LOE fault the three FIS produced a rise time of 4.7 seconds, 0.2 seconds less than expected. The other three FIS improved on this result by a further decrease of 0.1 seconds. These results were then reproduced for when 50% LOE occurred in both blocks. When

considering the rise times for this size of step input there is never more than a difference of 0.1 seconds recorded between any of the four considered FISs.

The seven and nine rules FISs produced identical rise times, for tests considered, which were the lowest of the rise times recorded by the error estimation FISs.

Table 7.5 The 20 Degrees Step Input Rise Times.

	Rise Times (seconds)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	4.8	4.9	5.4	6.3	4.8	4.9	4.8	6.3	4.8	4.9	4.9	6.3
3 Rules FIS	4.6	4.7	5.3	7.3	4.6	4.7	4.8	7.3	4.6	4.7	4.9	7.3
5 Rules FIS	4.6	4.7	5.3	7.3	4.6	4.6	4.8	7.3	4.6	4.6	4.9	7.3
7 Rules FIS	4.6	4.7	5.3	7.2	4.6	4.6	4.8	7.2	4.6	4.6	4.9	7.2
9 Rules FIS	4.6	4.7	5.3	7.2	4.6	4.6	4.8	7.2	4.6	4.6	4.9	7.2

Tables 7.4 and 7.5 have shown how close the performances of the four FISs are when presented with the faults being considered at this increased yaw input step demand size. This is further shown in Figure 7.20 where the responses of the AUV can be seen for all four error estimation FISs and for the control system when the error sensor is used for the 50% LOE level of fault in both blocks.

It is clear from the figure how the overall system has been affected by replacing the error sensor with the estimation FIS.

The results for all four FISs in Figure 7.20 are very similar, therefore the error traces between each estimation FISs response and the response of the control system with the error sensor are displayed in Figure 7.21.

Figure 7.21 shows that all four error estimation FISs effects on the system are near identical. These results are typical, for this size of step input showing very little change of performance between each of the four error estimation FISs, of those shown in Appendix J.

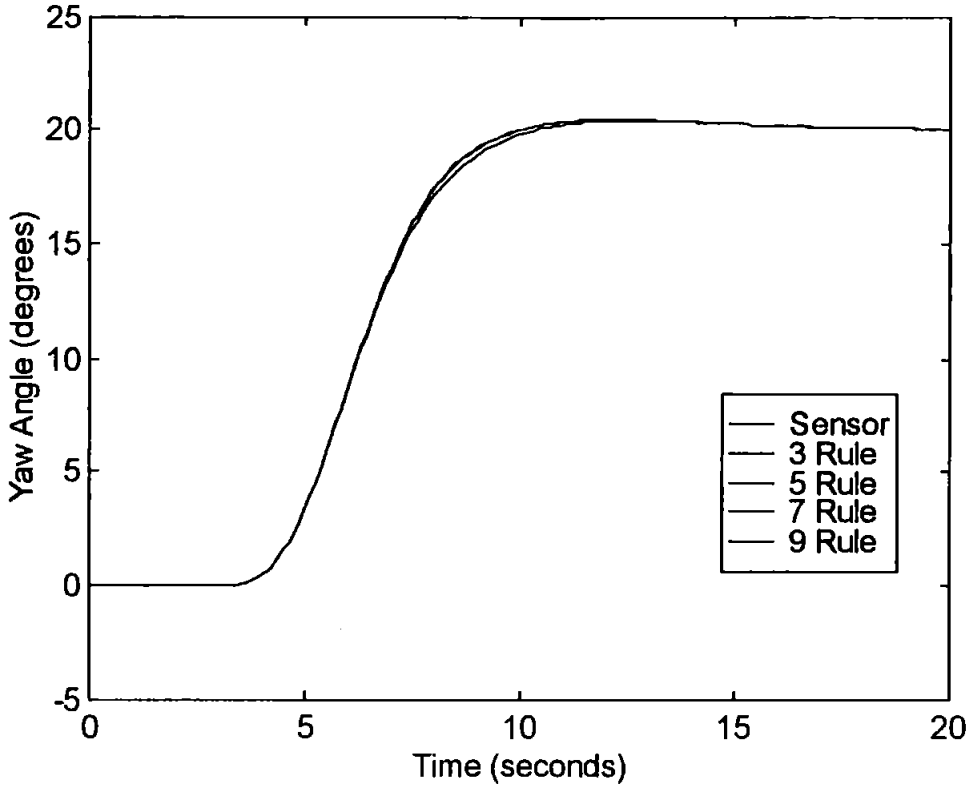


Figure 7.20 The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 20 degrees

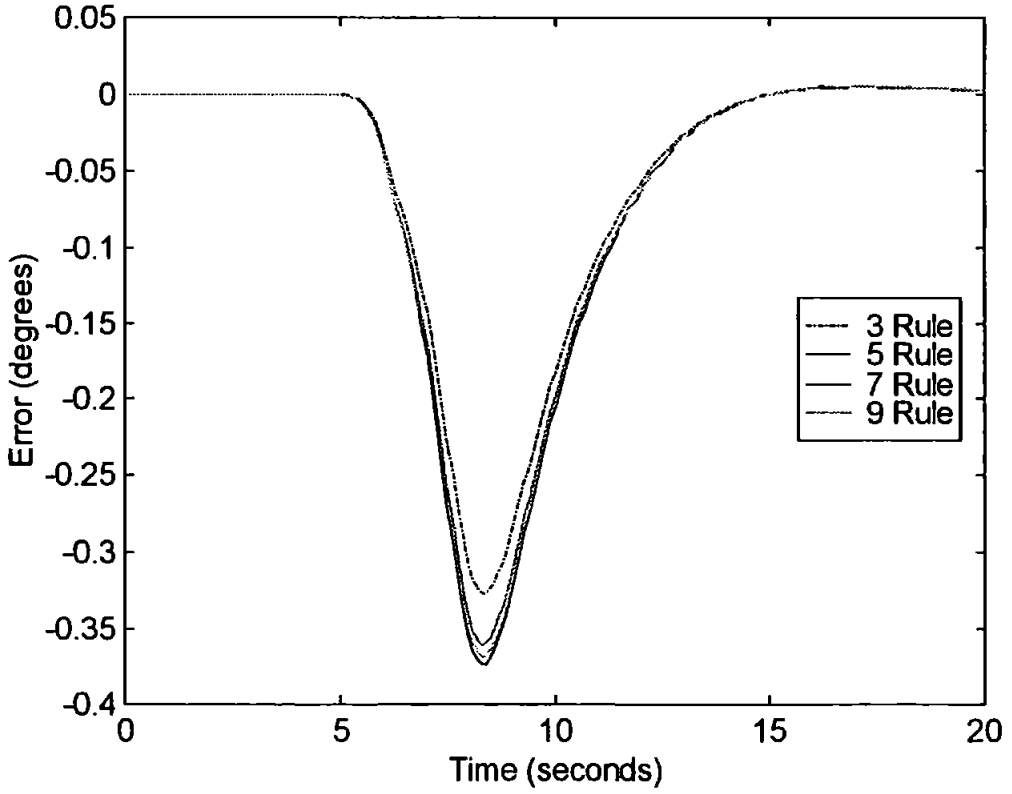


Figure 7.21 The Error Traces for The 50% LOE in Both Blocks for a Yaw Step Input of 20 degrees

Figures 7.22 and 7.23 show the upper and lower canard responses for the three rules FIS when a fault is occurring simultaneously in both the saturation and rate limiter blocks.

Figure 7.22 shows the faulty actuators canard response to an increasing level of LOE when the three rules FIS is used. For this level of LOE it is again clear that the saturation block fault is having an effect on the larger LOE tests.

Figure 7.23 shoes how the undamaged actuator is used to increase the work load of its canard to compensate to the fault.

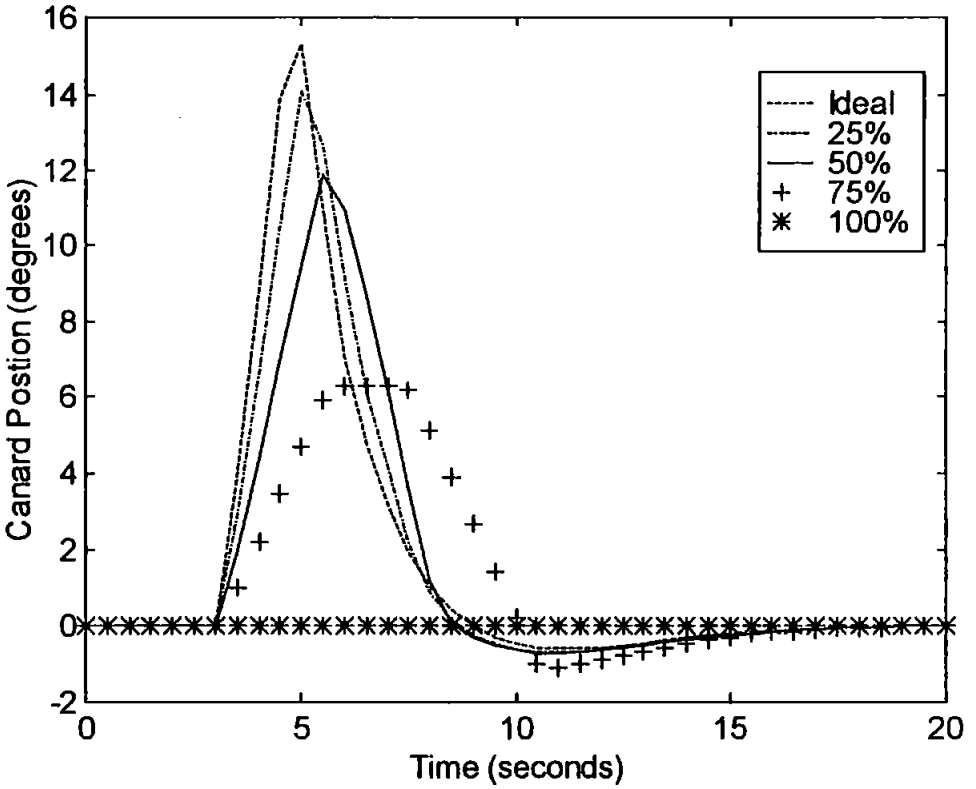


Figure 7.22 Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 20 degrees

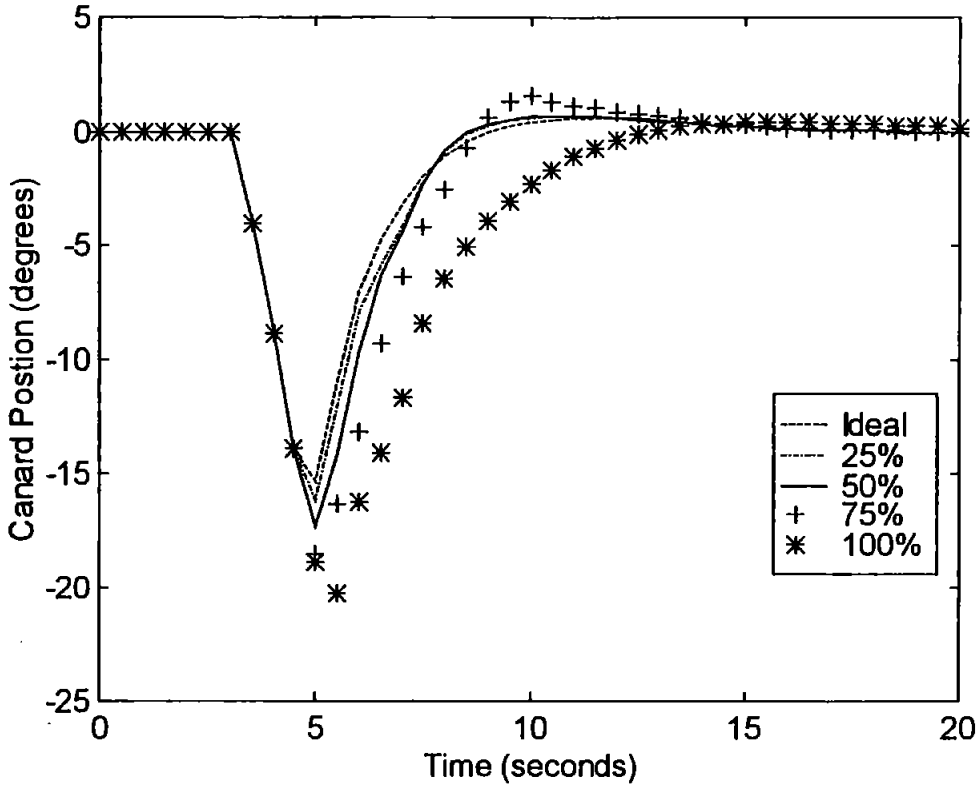


Figure 7.23 Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 20 degrees

To show the change in performance of the AUV for each of the error estimation FISs the percentage decrease in performance for each test, relative to the control system using the error sensor, has been calculated in Table 7.6.

Table 7.6 shows how all of the FISs could not handle the 25% LOE level in the saturation block, this was the same for all estimation FIS produced. Again it was expected that the use of an error estimation FIS would lead to a decrease in performance. This is the case for ten of the twelve tests at this step size. However for the 50% LOE levels in both the rate limiter block and both block faults the performance has been increased by the use of an error estimation FIS.

Table 7.6 The 20 Degrees Step Input Percentage Decrease.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
3 Rules FIS	442%	4.9%	0.4%	28.3%	4.7%	-4.7%	0.8%	28.3%	4.7%	-4.6%	1.3%	28.3%
5 Rules FIS	451%	5.1%	0.2%	27.1%	4.8%	-5.1%	0.9%	27.1%	4.8%	-5.1%	1.4%	27.1%
7 Rules FIS	467%	5.5%	0.2%	26.6%	5.0%	-5.1%	0.9%	26.6%	5.0%	-5.1%	1.4%	26.6%
9 Rules FIS	456%	5.2%	0.3%	26.9%	5.0%	-5.1%	0.9%	26.9%	5.0%	-5.1%	1.4%	26.9%

7.4.7. Yaw Step Inputs of 30 Degrees

Finally the yaw input demand is increased to 30 degrees. All of the FISs are tested on this, the largest step size considered for actuator faults in this thesis.

The complete set of results (RMSEs and rise times) for all four estimator FISs described, along with the results for the system when the error sensor is used, are presented in Tables 7.7 and 7.8.

When considering the RMSE results for this level of yaw input demand, it is again clear from Table 7.7 that the four estimator FISs performed to a similar standard. There is a variance in results of similar size to that of the 20 degrees yaw input demand for this set of tests, with the biggest difference recorded in any one type of test being 0.027 degrees.

For the saturation block fault of 25% LOE all the estimation FISs produced identical results of a RMSE of 0.093 degrees. The seven and nine rules FISs both produced the lowest RMSE values for the 50% and 100% LOE level tests. While the three rules FIS individually produced the lowest RMSE value for the 75% LOE test. When the fault was implemented in the rate limiter block the 25% LOE was handled most effectively by the three, five and nine rules FISs which all produced RMSE values 0.001 degrees less than that of the seven rules FIS. For the 50% LOE level it was the five and nine rules FISs producing the lowest RMSE values, while the five, seven and nine rules FIS produced the lowest RMSE values for the 75% LOE level. As to be expected, due to the fault being identical to that of the 100% LOE in the saturation block as explained in Chapter 6 the lowest RMSE values were again produced by the seven and nine rules FISs. Finally let the results from the set of tests where the fault was

implemented in both the saturation and rate limiter blocks simultaneously be considered. When a fault of only 25% LOE was simulated all four FISs produced identical RMSE values. For the 50% LOE level tests the three rules FIS produced clearly the lowest RMSE value and the seven rules FIS produced the lowest value for the 75% LOE level tests. The seven and nine rules FISs once again produced the lowest RMSE values for the 100% LOE level tests.

Table 7.7 The 30 Degrees Step Input RMSEs.

	RMSEs (degrees)											
Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	0.082	0.673	1.514	3.535	0.435	1.094	2.055	3.535	0.437	1.140	2.129	3.535
3 Rules FIS	0.093	0.641	1.766	3.859	0.431	1.074	2.043	3.859	0.435	1.130	2.241	3.859
5 Rules FIS	0.093	0.641	1.774	3.843	0.431	1.073	2.041	3.843	0.435	1.138	2.232	3.843
7 Rules FIS	0.093	0.638	1.793	3.842	0.432	1.074	2.041	3.842	0.435	1.131	2.226	3.842
9 Rules FIS	0.093	0.638	1.792	3.842	0.431	1.073	2.041	3.842	0.435	1.135	2.233	3.842

When considering the rise times recorded from these tests, it is clear from Table 7.8 of the performance of each of the FISs. All four of the FISs were able to estimate the actuator error to a level where the overall system was capable of reproducing the rise times identically for the rate limiter and both faults for both the 25% and 50% LOE levels. For the saturation fault at these LOE levels a decrease in rise times of 0.2 seconds were recorded for all eight tests. When the LOE level was 75% the results show an increase in rise times of 0.3 seconds for every fault and every FIS for all but one, the three rules FIS required an extra 0.1 seconds. When the error estimation FISs were used an increase in the rise times for the 100% LOE tests of 0.9 seconds to 7.7 seconds were recorded for every FIS and every type of fault.

In the 48 tests show in Table 7.8 for the four error estimation FISs there is only one results where any change between the FISs can be detected. The difference is in the 75% LOE fault where both the saturation and rate limiter blocks are affected. The three rules FIS produces a rise time 0.1 seconds slower than the other FISs. When considering rise times for the 30 degrees step input demand, this is the only difference between the contemplated FISs.

Table 7.8 The 30 Degrees Step Input Rise Times.

Controller	Rise Times (seconds)											
	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
With Error Sensor	5	5.4	5.6	6.8	4.6	4.5	4.8	6.8	4.6	4.7	5.2	6.8
3 Rules FIS	4.8	5.2	5.9	7.7	4.6	4.5	5.1	7.7	4.6	4.7	5.6	7.7
5 Rules FIS	4.8	5.2	5.9	7.7	4.6	4.5	5.1	7.7	4.6	4.7	5.5	7.7
7 Rules FIS	4.8	5.2	5.9	7.7	4.6	4.5	5.1	7.7	4.6	4.7	5.5	7.7
9 Rules FIS	4.8	5.2	5.9	7.7	4.6	4.5	5.1	7.7	4.6	4.7	5.5	7.7

Tables 7.7 and 7.8 have shown how close the performances of the four FISs are when presented with the faults being considered at this increased yaw input step demand size. This is further shown in Figure 7.24 where the responses of the AUV can be seen for all four error estimation FISs and for the control system when the error sensor is used for the 50% LOE level of fault in both blocks.

It is clear from the figure how the overall system has been affected by replacing the error sensor with the estimation FIS.

The results for all four FISs in Figure 7.24 are again very similar, therefore the error traces have again been calculated and are displayed in Figure 7.25.

Figure 7.25 shows the effect of all four error estimation FISs on the. These results are typical, for this size of step input showing very little change of performance between each of the four error estimation FISs, of those presented in Appendix J.

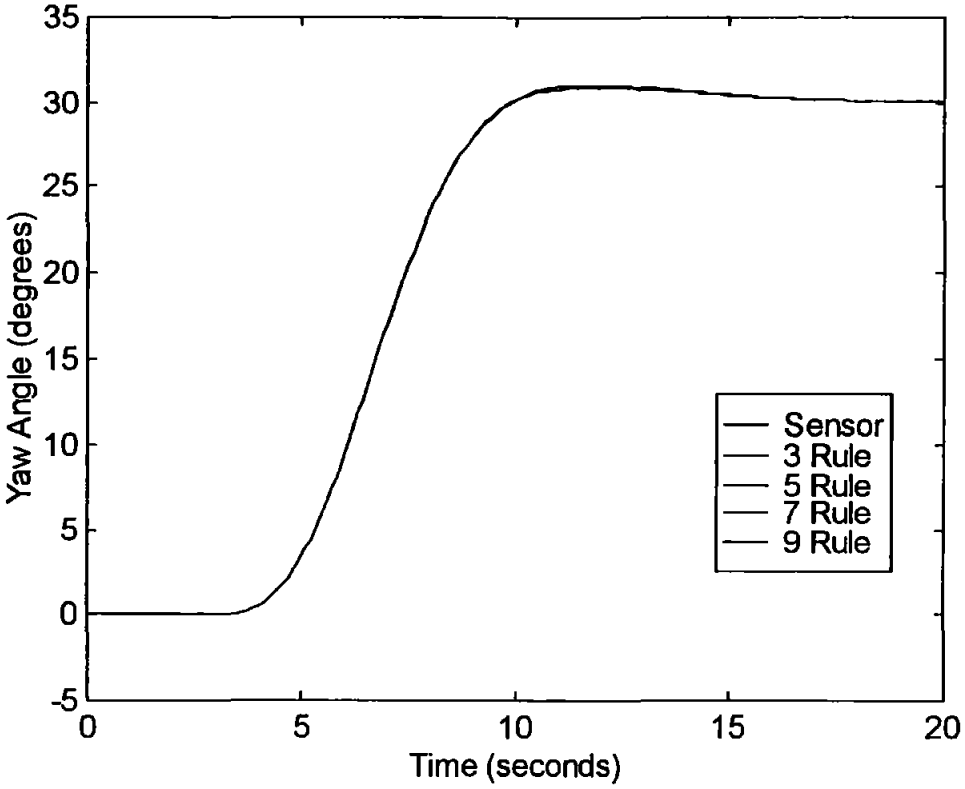


Figure 7.24 The AUV Responses to 50% LOE in Both Blocks for a Yaw Step Input of 30 degrees

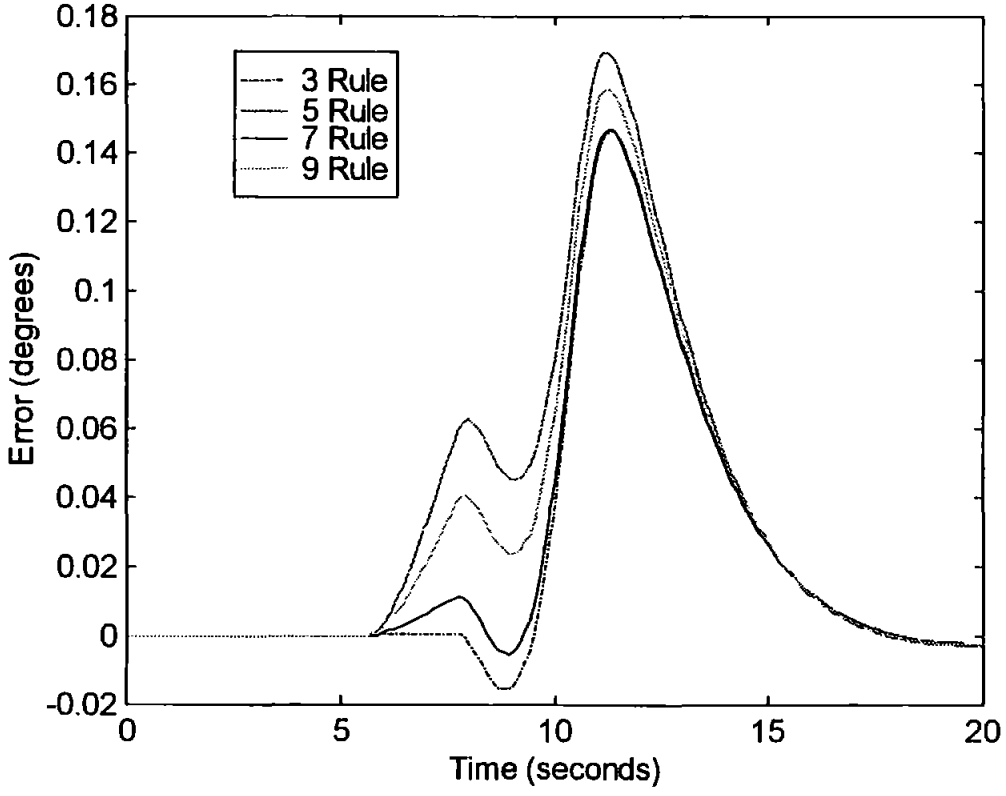


Figure 7.25 The Error Traces for The 50% LOE in Both Blocks for a Yaw Step Input of 30 degrees

Figures 7.26 and 7.27 show the upper and lower canard responses for the three rules FIS when a fault is occurring simultaneously in both the saturation and rate limiter blocks.

In Figure 7.26 the canard response for the damaged actuator is again show for an increasing level of LOE for this increased size of yaw angle. Due to the increased demanded yaw the effects of the faults are more clearly shown.

Figure 7.27 shows how the lower canard is used to compensate for the fault occurring within the upper canard. The complete set of results for all types and sizes of LOE for all four considered error estimator FISs are presented in Appendix K.

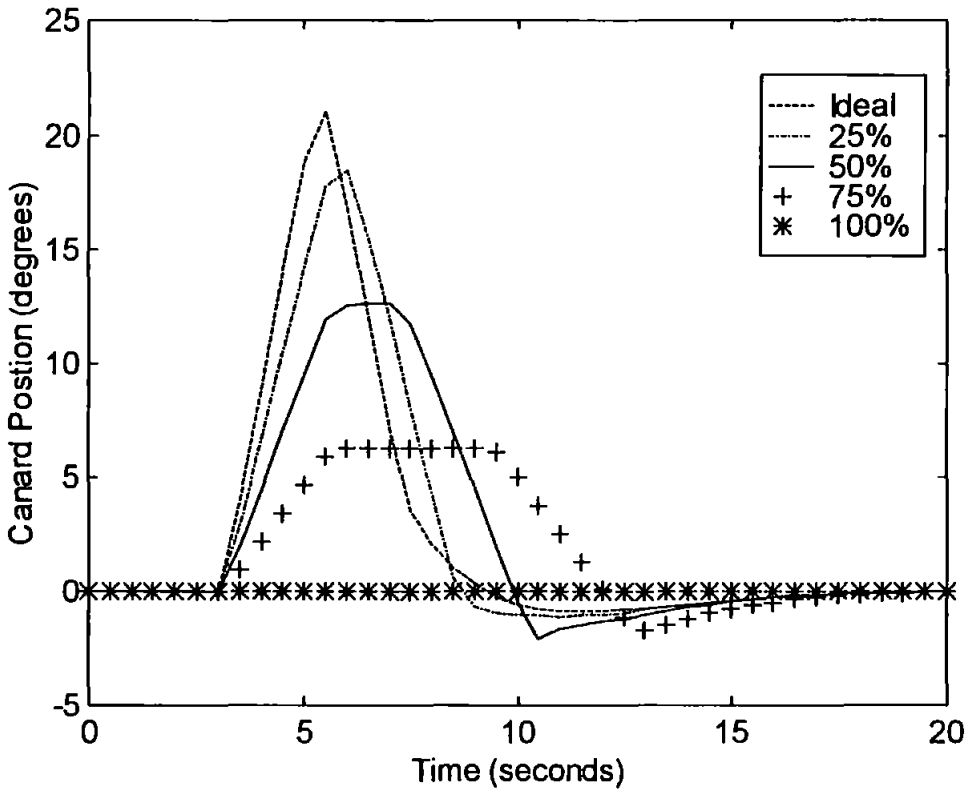


Figure 7.26 Upper Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 30 degrees

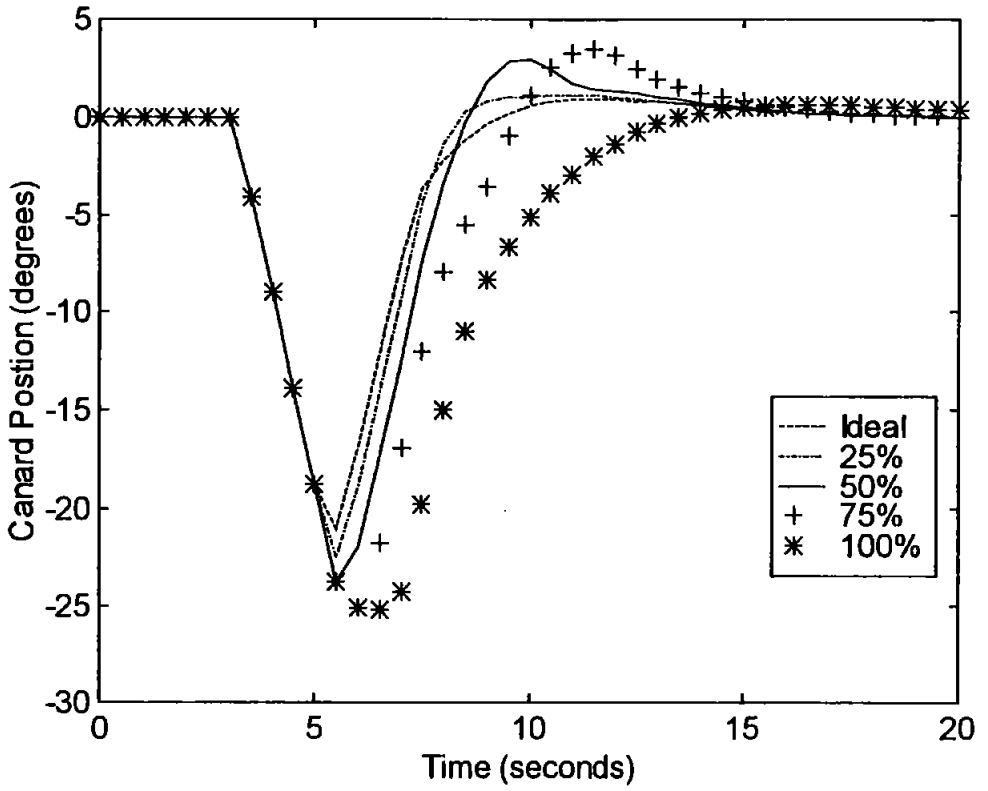


Figure 7.27 Lower Canard Responses When Using Three Rules FIS for Both Blocks LOEs for a Yaw Step Input of 30 degrees

To show the change in performance of the AUV for each of the error estimation FISs below the percentage decrease in performance for each test, relative to the control system using the error sensor, has been calculated in Table 7.9.

Again it was expected that the use of an error estimation FIS would lead to a decrease in performance. This is the case for six of the twelve tests at this step size. However for the half of the considered faults, for all the sizes of FISs, the inclusion of an error estimator improved the performance of the overall control system.

Table 7.9 The 30 Degrees Step Input Percentage Decrease.

Controller	Saturation				Rate Limiter				Both			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
3 Rules FIS	13.4%	-4.8%	16.7%	9.2%	-0.9%	-1.8%	-0.6%	9.2%	-0.6%	-0.9%	5.3%	9.2%
5 Rules FIS	13.5%	-4.8%	17.2%	8.7%	-0.9%	-1.9%	-0.7%	8.7%	-0.6%	-0.2%	4.9%	8.7%
7 Rules FIS	13.9%	-5.3%	18.4%	8.7%	-0.8%	-1.8%	-0.7%	8.7%	-0.5%	-0.8%	4.5%	8.7%
9 Rules FIS	13.9%	-5.2%	18.4%	8.7%	-0.8%	-1.9%	-0.7%	8.7%	-0.5%	-0.5%	4.9%	8.7%

7.5. CONCLUSIONS

The aim of this Chapter was to replace the error sensor used by the actuator fault tolerant FIS with an error estimation FIS. Four ANFIS [Jang (1991)] tuned FISs of differing numbers of rule bases were developed and compared. It was discovered that all four FISs were capable of estimating the error in the actuator such that the system could continue to operate with minimal performance degradation.

After close examination of the results shown in section 7.4, in the form of both tables and graphs, along with the graphs presented in Appendix J, it is difficult to draw a conclusion as to which of the four estimation FISs performed most accurately.

When considering the RMSE tables and the percentage decrease tables one result is highly noticeable, that of the 25% LOE for the saturation block fault over a 20 degrees yaw step input demand. A large increase of between 442% and 467% was recorded. This result was repeated throughout all the error estimation FISs created. This may have occurred due to the large number of and types of faults being considered over several sizes of step inputs. Of the other tests various FISs performed well on different types of fault, levels of LOE and sizes of step inputs. The lack of a clearly most accurate error estimation FIS makes it hard to conclude which should be used in the AUV control system.

When considering the rise times the error estimation FISs all performed to a very similar standard, with only a small difference apparent (0.2 seconds or less) for 96 of the 144 tests. The largest increases were always in the 100% LOE tests and for several types of faults and levels the control system was able to decrease the rise times of the AUV when using the error sensor. This was an unexpected result, as an identical

estimation of the error in the actuator would lead to an identical result. It may be that the FISs inability to model perfectly the error has lead to an overall increase in performance for some cases. Again it is vary hard to conclude, from the rise time results, which error estimation FIS should be used in the AUV control system.

From the remarks in this section it is not possible to recommend which of the four FISs would be most capable of estimating the actuator error. It is clear that any of the four FIS could be used in the overall control system to remove the actuator fault recovery FISs need for a sensor on the actuator. The implication of this is an increase in fault tolerance due to the removal of sensors within the vehicle.

CHAPTER 8

CONCLUDING REMARKS

8.1. CONCLUSIONS

The aim of this work was to produce novel FISs capable of handling AUV sensor faults and canard controlling actuator faults. This aim has been achieved by the successful design of fault tolerant FISs as show within this thesis. The work presented in Chapters 5, 6 and 7 clearly demonstrates the superiority of the these FIS enhanced controllers over the benchmark methods using Kalman filters and standard control systems presented in Chapter 4.

Chapter 5 produced novel FISs capable of handling sensor faults within the yaw and roll channels of the AUV. These sensor recovery FISs, tuned using the both the ANFIS [Jang (1991)] and the simulated annealing [Kirkpatrick et *al* (1983)] method, show a notable increase in fault tolerance when compared to the benchmark Kalman filter enhanced control system of Chapter 4. The best results were obtained by the simulated annealing tuned FIS designed for the roll sensor. The results presented show that the FIS was able to produce RMSE values which were less than the benchmark Kalman filter enhanced control system for twenty-five of the twenty-seven considered tests by between 6% and 62%.

The results of Chapter 5 show that it is possible to develop fault tolerant FISs that are more effective at handling both types of sensor faults the FISs were tuned on i.e. the percentage signal loss and the intermittent signal loss. The FISs were also capable of handling some sensor faults they had not been tuned on i.e. signal to noise ratio (SNR). This demonstrates the flexibility of the FISs to handle unexpected types of sensor faults.

The results presented clearly show that it is possible to create FISs which improve on the fault tolerance performance of the AUV. However there was no clear indication as to which tuning method provides the most fault tolerant FIS. The ANFIS tuned FIS

achieving smaller RMSE values for some tests and the simulated annealing tuned FIS achieving smaller RMSE values for other tests.

The main conclusion of Chapter 5 was that the FISs were capable of improving on the Kalman filter enhanced ANFIS control system.

The actuator recovery FISs developed in Chapter 6 and enhanced in Chapter 7 significantly increased the AUVs fault tolerance with regards to the canards becoming damaged. In Chapter 6, FISs were designed which considered the control signal being sent to the canard, the demand being placed on the ANFIS controller, and the error within the damaged canard actuator, in order to produce a control signal modifying fuzzy singleton. The results presented showed how both the simulated annealing and the tabu search method tuned FISs were capable of handling the faults far better than the benchmark control system. The conclusion of the discussion that took place at the end of Chapter 6 was that the simulated annealing tuned FIS made the system more fault tolerant than the tabu search method tuned FIS. It was also shown that either FIS could be used to enhance significantly the fault tolerant performance of the AUV. The most fault tolerant FIS as identified from results of Chapter 6, was then used for the work of Chapter 7.

Further enhancement of the fault tolerance of the system was achieved in Chapter 7. The sensor used by the FISs in Chapter 6 to measure the error within the damaged canard actuator was replaced by a FIS. The error estimation FIS, developed in this Chapter, removed the actuator recovery FIS's dependence on the error sensor making it more fault tolerant. Results clearly show that replacing the sensor with the fuzzy estimator actually improved the overall performance of the AUV for some of the considered faults.

Chapter 7 also considered FIS with different sizes of rule bases (3, 5, 7 and 9 rules). The results presented showed that there was very little change in performance as the number of rules was increased. Leading to the conclusion that all four FISs were capable of estimating the error with a similar success and therefore to use the three rule FIS to replace the error sensor in the overall control system.

A further contribution of this thesis is to advance knowledge and understanding of methods to tune FISs by considering the tuning of FISs by various methods. Specifically the tabu search [Denna et *al* (1996)] method of tuning has been developed and used to tune the actuator fault recovery FISs of Chapter 6.

The novel FISs presented in Chapters 5, 6 and 7 of this thesis clearly demonstrated increased levels of fault tolerance of the AUV to both sensor and actuator failure occurring during normal operation of the vehicle.

8.2. RESEARCH OBJECTIVES

The objectives required to achieve the aim of this programme of research were outlined in Chapter 1 and are reproduced here for ease of reference:

(a) Critically review the current fault tolerant control literature.

This objective has been achieved by the work presented in Chapter 2.

(b) Investigate and then develop identification models of the AUV dynamics using linear modeling, neuro-fuzzy and ANN approaches.

Several types of model of the AUV were investigated, developed and compared in Chapter 3 accomplishing the requirements of this objective.

(c) Devise sensor/actuator failure scenarios within the AUV model to assess the model responses.

This objective has been successfully achieved in Chapter 3 where all sensor/actuator faults have been defined.

(d) Investigate and then develop Kalman filter based fault tolerant control systems to be used as benchmarks. Test the robustness of the systems to various levels of faults.

The successful completion of this objective was achieved in Chapter 4 where benchmark systems were investigated before two were defined for later use.

(e) Develop intelligent fault control systems (IFCSs) for the yaw channel sensors based on neuro-fuzzy approaches such as the adaptive neuro-fuzzy inference system (ANFIS).

The FISs developed in Chapter 5 successfully accomplished this objective.

(f) Develop IFCSs for the roll channel sensors based on neuro-fuzzy approaches similar to those for yaw channel.

The roll channel work presented in Chapter 5 successfully accomplished this objective.

(g) Develop IFCSs for the yaw channel actuator faults using suitable fuzzy approaches.

The FISs designed in Chapter 6 effectively satisfy this objective.

(h) Remove and replace error sensor used in IFCS for actuator faults.

The work successfully satisfying this objective is documented in Chapter 7.

The work presented in these six chapters covers all eight objectives of the original proposal. All are successfully achieved by the work presented in Chapters 2, 3, 4, 5, 6 and 7 of this thesis.

8.3. RECOMMENDATIONS FOR FUTURE RESEARCH

Several different directions for future research have been highlighted through the completion of this work. The following points provide a summary of these areas:

- The combing of the sensor recovery FISs presented in Chapter 5 with the actuator fault tolerant FIS of Chapter 6 and the error estimation FIS of Chapter 7 to create a multi-fault tolerant control system.

- The implementation of this work within an AUV and evaluation of the fault tolerant control systems when acting in the real world.

- Investigation into the effects of varying the speed of the vehicle during sensor/actuator failures. With the possibility of developing IFCSs to work at any given speed including the possible use of the AUVs thrusters at low speeds.

- The Q statistic used by the sensor recovery FISs of Chapter 5 depend on both sensor information and a linear model of the vehicle. A different error measurement could be developed which does not depend on the linear model. This would further enhance the fault tolerance of the AUV.

- The FISs produced in this thesis contained some fuzzy rules that may be unnecessary during the operations of the AUV. The removal of these rules via a pruning algorithm would lead to a reduction in the dimension of the parameter space and hence a reduction in computation effort.

REFERENCES

Alessandri. A., Caccia. M. and Veruggio. G. (1999). Fault detection of actuator faults in unmanned underwater vehicles. Control Engineering Practice, 7: 357-368.

Anderson. J. M. M. and Edmonson. W. (1997). System identification with noisy input-output data using a cumulant-based Steiglitz-McBride algorithm. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 44, No 5: 407-409.

Antsaklis. P. J., Passino. K. M. and Wang. S. J. (1991). An introduction to autonomous control systems. IEEE Control System, June: 5-13.

Basseville. M. (1988). Detecting changes in signals and systems- a survey. Automatica, Vol. 24, No 3: 309-326.

Bodson. M. and Groszkiewicz. J. E. (1997). Multivariable adaptive algorithms for reconfigurable flight control. IEEE Transactions on Control Systems Technology, Vol. 5, No 2: 217-229.

Bozic. S. M. (1979). Digital and Kalman filtering. Edward Arnold.

Caccia. M. and Veruggio. G. (2000). Guidance and control of a reconfigurable unmanned underwater vehicle. Control Engineering Practice. Vol. 8, No 1: 21-37.

Craven. P. J. (1999). Intelligent control strategies for an autonomous underwater vehicle. PhD thesis, University of Plymouth, U.K.

Craven. P. J., Sutton. R. and Burns. R. S. (1998). Control strategies for unmanned underwater vehicles. Journal of Navigation, Vol. 51, No 1: 79-105.

Denna. M., Mauri. G. and Zanaboni. A. (1999). Learning fuzzy rules with Tabu search- an application to control. IEEE Transactions on Fuzzy Systems, Vol. 7, No 2: 295-318.

Derradji. D. A. and Mort. N. (1996). Control reconfiguration in submersible vehicles using artificial neural networks. 13th World Congress of IFAC, San Francisco 96: 333-338.

Diao. Y. and Passino K. M. (2001). Stable fault-tolerant adaptive fuzzy/neural control for a turbine engine. IEEE Transactions on Control Systems Technology, Vol. 9, No 3: 494-509.

Dorato. P., Abdallah. C. and Cerone. V. (1995). Linear-quadratic control- an introduction. Prentice Hall, Englewood Cliffs. NJ. USA.

Eddy. P., Potter. E. and Page. P. (1976). Destination disaster. Hart-Davis Mac Gibbon Book Co. London. U.K.

Eryurek. E. and Upadhyaya. B. R. (1995). Fault-tolerant control and diagnostics for large-scale systems. IEEE Control Systems, October: 34-42.

Fairman. F. W. (1998). Linear Control Theory-the state space approach. John Wiley and Sons, Chichester. U.K.

Fossen. T. I. and Fjellstad. O. E. (1995). Robust control of underwater vehicles: a comparative study. Proceedings of 3rd IFAC Workshop on CAMS, Trondheim: 43-48.

Gao. Z. and Antsaklis. P. J. (1992). Reconfigurable control system design via perfect model following. International Journal of Control, Vol. 56, No 4: 783-798.

Grewal. M. S. and Andrews. A. P. (1993). Kalman filtering: theory and practice. Prentice-hall, Upper Saddle Rivere. NJ. USA.

Hassoun. M. H. (1995). Fundamentals of artificial neural networks. A Bradford Book, The MIT Press, London. U.K.

Hatton. K. and Fennell. H. (1999). The Hall effect. http://math.sunyit.edu/projects/vector/hall.htm.

Haykin. S. (1994). Neural networks- a comprehensive foundation. Prentice Hall, Upper Saddle Rivere. NJ. USA.

Healey. A. J. and Lienard. D. (1993). Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE Journal of Oceanic Engineering, Vol. 18, No 3: 327-339.

Ishii. K., Fujii. T. and Ura. T. (1995). An on-line adaptation method in a neural network based control system for AUVs. IEEE Journal of Oceanic Engineering, Vol. 20, No 3: 221-228.

Jang. J. S. R. (1991). ANFIS: Adaptive network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23: 665-685.

Katebi. M. R. and Grimble. M. J. (1999). Integrated control guidance and diagnosis for reconfigurable autonomous underwater vehicles control. Special Issue of the International Journal of Systems Science on UUV Control, Vol. 30, No 9: 1021-1032.

Kim. R. R., Ware. J. R. and Ammeen. E. S. (1997). Applications of neural networks and fuzzy logic in failure detection and fault tolerant control system design. Eleventh Ship Control Systems Symposium, Vol. 2: 189-208.

Kirkpatrick. S., Gelatt. C. and Vecchi. M. (1983). Optimisation by simulated annealing. Science, Vol. 220: 671-680.

Kosko. B. (1994). Fuzzy thinking. Flamingo, London. U.K.

Kuo. B. C. (1982). Automatic control systems forth edition. Prentice Hall, Upper Saddle Rivere. NJ. USA.

Leonard. N. E. (1994). Averaging and motion control of systems on lie groups. Doctorate Thesis. University of Maryland.

Leonard. N. E. (1995). Control synthesis and adaptation for an underactuated autonomous underwater vehicle. IEEE Journal of Oceanic Engineering, Vol. 20, No 3: 211-220.

Looze. D. P., Weiss. J. L., Eterno. J. S. and Barrett. N. M. (1985). An automatic redesign approach for restructurable control systems. IEEE Control Systems Magazine, May: 16-22.

Lopez-Toribio. C. J., Patton. R. J. and Daley. S. (2000). Takagi-Sugeno fuzzy fault-tolerant control of an induction motor. Neural Computing and Applications, Vol. 9: 19-28.

McLean. D., Saade. C. and Zinsch. F. (1997). A sensor fault accommodation system using a neural network. Transactions of the Institute of Measurement and Control, Vol. 19, No 3: 166-168.

Naidu. S. R., Zafiriou. E. and McAvoy. T. J. (1990). Use of neural networks for sensor failure detection in a control system. IEEE Control Systems Magazine, April: 49-55.

Napolitano. M. R., Windon II. D. A., Casanova. J. L., Innocenti. M. and Silvestri. G. (1998). Kalman filters and neural-network schemes for sensor validation in flight control systems. IEEE Transactions on Control Systems Technology, Vol. 6, No 5: 596-611.

Ochi. Y. and Kanai. K. (1991). Design of restructurable flight control systems using feedback linearization. Journal of Guidance, Vol. 14, No 5: 903-911.

Patton. R. J., Chen. J. and Nielsen. S. B. (1995). Model-based methods for fault diagnosis: some guide-lines. Transactions of the Institute of Measurement and Control, Vol. 17, No 2: 73-83.

Perrault. D. and Nahon. M. (1999). Fault-tolerant control of an autonomous underwater vehicle. Oceanic Engineering International, Vol. 3, No 2: 85-94.

Podder. T. K. and Sarker. N. (2001). Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy. Robotics and Autonomous Systems, Vol. 34: 39-56.

Rauch. H. E. (1994). Intelligent fault diagnosis and control reconfiguration. IEEE Control Systems, Vol. 14, No 3: 6-12.

Rauch. H. E. (1995). Autonomous control reconfiguration. IEEE Control Systems, December: 37-47.

Stengel. R. F. (1991). Intelligent failure-tolerant control. IEEE Control Systems, June: 14-23.

Tacconi. G. and Tiano. A. (1989). Reconfigurable control of an autonomous underwater vehicle. 6th Int. Symposium of Unmanned Untethered Submersible Technology Proceedings, Uni. of New Hampshire, USA: 486-493.

Thompson. S. (1993). Robust control-an introduction. Measurement and Control, Vol. 26: 235-241.

Valavanis. K. P., Gracanin. D., Matijasevic. M., Kolluru. R. and Demetriou. A. (1997). Control architectures for autonomous underwater vehicles. IEEE Control Systems, Vol. 17, No 6: 48-64.

Waldock. M. I. (1996). Artificial neural networks for control and modelling of an unmanned underwater vehicle. MPhil University of Plymouth, U.K.

Wang. H. (1995). Actuator fault diagnosis for non-linear dynamic systems. Transactions of the Institute of Measurement and Control, Vol. 17, No 2: 63-71.

Warwick. K., Irwin. G. W. and Hunt. K. J. (1992). Neural Networks for control and systems. Peter Peregrinus Ltd. U.K.

Yang. K. C. H., Yuh. J. and Choi. S. K. (1999). Fault-tolerant system design of an autonomous underwater vehicle- ODIN: an experimental study. International Journal of Systems Science, Vol. 30, No 9: 1011-1019.

Appendix A Publications

Papers from this work are:

Pearson. A. R., Sutton. R., Burns. R. S., Robinson. P. and Tiano. A. (2000). Fault tolerant control strategies for uninhabited underwater vehicles. Underwater Technology, Vol. 24, No 2: 61-72.

Pearson. A. R., Sutton. R., Burns. R. S. and Robinson. P. (2000). A Kalman filter approach to fault tolerant control in autonomous underwater vehicles. Fourteenth International Conference on Systems Engineering, Coventry, U.K. 12-14 September 458-463.

Pearson. A. R., Sutton. R., Burns. R. S. and Robinson. P. (2001). A fuzzy fault tolerant control scheme for an autonomous underwater vehicle. IFAC CAMS2001, Glasgow, U.K. 17-20 July

Sutton. R.. Pearson. A. R. and Tiano. A. (2001). A fuzzy fault tolerant control scheme for an autonomous underwater vehicle. IEEE Proceedings of MMAR 2001 Conference, Międzyzdroje, Poland, August 28-31. Vol. 2: 595-600.

ISSN 0141 0814

Underwater Technology

Journal of the Society for Underwater Technology

Volume 24 Number 2 Winter 1999/2000

CONTENTS

Page

A Personal View:
...Got the tee shirt 39
B. Loth

Notes for Contributors. 40

Buoyancy Mechanisms of Marine Organisms: Lessons from Nature 41
P. J. Molloy and M. J. Cowling

Electric Field Signatures of Ships in Southampton Water 51
M. Varney, S. Bailey and M. Siddall

Fault Tolerant Control Strategies for Uninhabited Underwater Vehicles 61
A. R. Pearson, R. Sutton, R. S. Burns, P. Robinson and A. Tiano

Deepwater Development Opportunities in Angola. 73
—Report on Meeting
M. Hibbert

6th Atlantic Frontier Workshop: Waves, Steep Waves and Statistics—Operating in Harsh Environment. . . . 81
—Report on Meeting
S. Archer

$10 Oil: Is Underwater Robotics an Answer?. . . . 85
—Report on Meeting
I. N. L. Gallett

Book Reviews. 89

ISSN 0141 0814. Journal of the Society for Underwater Technology Vol 24 No 2 pp. 61-71, 1999/2000

Fault Tolerant Control Strategies for Uninhabited Underwater Vehicles

A. R. PEARSON[1], R. SUTTON[1], R. S. BURNS[2], P. ROBINSON[3] and A. TIANO[4]

[1] *Department of Mechanical and Marine Engineering,* [2] *Department for Business Development,* [3] *School of Electronic, Communication and Electrical Engineering, University of Plymouth, Plymouth, UK.* [4] *Department of Information and Systems, University of Pavia, Pavia, Italy*

Abstract

Commercial, naval and scientific operational specifications for uninhabited underwater vehicles (UUVs) continue to become more challenging in line with the advances being made in control engineering. In order to survive actuator and/or sensor failure during a mission, such vehicles need to possess a reconfigurable or fault tolerant control system. This paper explains the basic principles of fault tolerant control systems. It then reviews their application in the design of UUVs and other systems where it is considered a technology transfer is possible.

1. Introduction

By steering an aircraft via differential engine thrust, the captian of a crippled American Airlines DC10 landed safely at Windsor, Ontario, under circumstances similar to those which claimed the lives of 346 passengers and crew of a Turkish Airlines aeroplane in the forest of Ermenouville, France on 3 March 1974 [1]. The successful survival of the DC10 at Windsor can be attributed to the fast adaptation ability of the pilot to control what had become, in effect, a different vehicle.

It is interesting to note that the DC10 pilot is claimed to have the maxim 'he who hesitates will probably survive' insofar as hasty action may make a situation irrecoverable, but calm experimentation in a high stress environment may well lead to success. The event itself was 'recoverable' because it was not totally unexpected, a similar problem having luckily been identified and solved on a training simulator some months earlier by the particular pilot concerned. Once he recognised the symptoms of the flight failure, the pilot was relatively well equipped to land the aircraft.

As will be seen from material which is to be presented here, the incident recounted above provides an excellent example of a human reconfigurable control scheme. Given the ongoing advances being made in control engineering and artificial intelligence techniques, serious consideration is now being given to the development of automated reconfigurable control systems (RCSs) that will operate autonomously whether or not there is a human in the loop.

In recent years, considerable interest has been shown in the commercial, scientific and naval use of uninhabited underwater vehicles (UUVs). UUV being a generic expression to describe both an autonomous underwater vehicle (AUV) and a remotely operated vehicle (ROV). An AUV is a marine craft that fulfils a mission or task without being constantly monitored and supervised by a human operator, whilst an ROV is a marine vessel that requires instruction from an operator via a tethered cable. Demands are growing for the requirement for AUVs to be able to operate at extreme depths and/or in confined areas such as under packed ice. Unfortunately, owing to the nature of these vehicles, data transmissions to and from the craft to the mother station through the sea water medium are poor. Thus, an AUV has to be totally self-sufficient during the duration of a mission. Hence, in order for an AUV to survive sensor and/or actuator failure in this environment, it is paramount to have on board an RCS. Such systems could also be beneficially installed in ROVs. Their employment within ROVs would lighten the work load of the human operators, whilst at the same time allowing them to maintain overall supervisory control.

Thus, the following two questions may be posed:

(i) What are the essential elements in a non-human RCS? and
(ii) How does such a system function?

This paper attempts to answer these questions and also to review applications of this technology in the underwater vehicle field and other areas. For the interested reader some other excellent reviews already written on this subject area can be found in [2], [3], [4] and [5].

Throughout this text it is assumed the reader has a basic understanding of control engineering principles; if this is not the case, reference should be made to [6] for an introduction to the subject. Control strategies which have been applied to UUVs are reviewed by Craven *et al.* [7]. Some

of the control schemes described were developed using artificial intelligence techniques. Background reading on artificial intelligence approaches can be found in [8] and [9].

2. Fault Tolerant Control

If a vehicle is damaged or has a sensor or actuator failure during operation, it may not be possible to repair the damage or fault immediately. There may, however, be a requirement for it to continue operating until it is possible to carry out the necessary repairs. Under such circumstances, the overall system is called an impaired system. When such a scenario as this occurs it is very unlikely that a standard control scheme will be able to cope and therefore a different kind of control system, which can handle the anomalies, would be more appropriate. A controller with such capabilities is considered to have a fault tolerant, a reconfigurable or a restructurable structure. Collectively, such controllers are generically called fault tolerant systems; however each approach operates in a different manner. As well as being the generic term for this type of system it is also possible to have a fault tolerant controller that is neither reconfigurable nor restructurable.

All controllers have authority over active parts, such as surfaces or motors of some kind, which they use in a certain configuration to perform given tasks. These controllers can be simple or very complex, depending on the performance requirements of the system, and will be optimally designed accordingly.

Having dealt with the basic ideas concerned with a fault tolerant controller, it would now be helpful to describe the objectives of this approach. The first priority of such a controller is to stabilise the system. Having stabilised the system, the next step is to try to return the system as close as is possible to its normal operating conditions and this will necessitate the use of different control laws.

2.1 Fault tolerance

A fault tolerant controller is capable of maintaining a system at a given level after it has been damaged. The controller does not necessarily return the system to a perfect condition after the damage but, obviously, this would be an advantage. Such a device has three subsystems or components that constitute the control system.

The first subsystem deals with detecting if a problem has arisen in the system. The second subsystem copes with identifying the location of the fault and its seriousness. Finally, the third is the actual reconfiguration subsystem itself. A simple block diagram of these subsystems and their integration into a system is shown in Figure 1.

The fault tolerant controller is only activated when there is a fault in the system. Hence, the first part of the controller, which deals with detecting if a fault has occurred, is continually monitoring the plant. This subsystem is not concerned with what the problem is or how to fix it but only whether the system is performing as required. This can be achieved by having an analytical method for checking out the system; this method compares what should be happening to what is actually happening in the system. When a fault occurs this subsystem informs the next subsystem that there is a problem and continues to do so until the fault has been corrected.

The function of the second subsystem is to identify what fault has occurred in the system and the seriousness of the fault. There are many different kinds of analytical methods that have been developed for this purpose, such as the multiple model method or the generalised likelihood ratio method; these will be discussed in more detail later in the paper. It is important to know the seriousness of the problem, as this will help the controller decide how to deal with the problem. Basseville [10] and Wang [11] have investigated both of these types of subsystem in some detail.

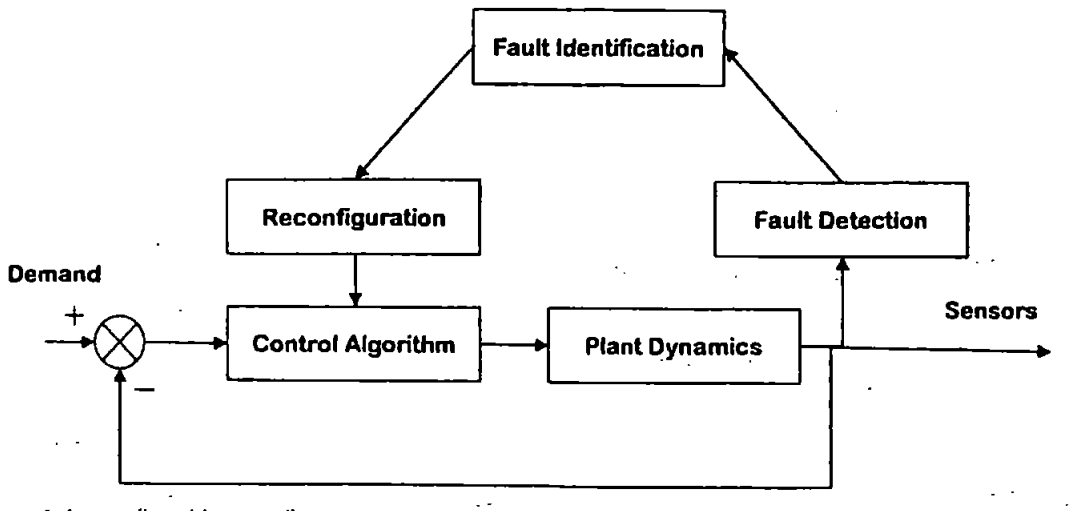


Figure 1 A reconfigurable controller.

A considerable amount of research has been undertaken in the area of using model-based methods to detect and identify faults in systems. Patton *et al.* [12] describe several methods based on more traditional approaches and give some general guidelines for the implementation of these types of subsystems. In addition, artificial neural networks (ANNs) have also been used by Naidu *et al.* [13] for sensor failure detection.

Alessandri *et al.* [14] considered actuator failures in an ROV, using approximate models of its dynamics. The output of the model is compared to the actual output to detect and diagnose the fault. This is performed by a bank of estimators, which are extended Kalman filters. Extended Kalman filters are used because of the non-linearities of the model. Results are reported to show the effectiveness of this approach compared to the unfiltered model output.

It is the final component in which the type of controller is defined. A fault tolerant controller may use many different methods to handle faults in the system. One way is for the fault tolerant controller to make use of a second backup component. This allows the system to continue operating until the first component can be repaired or replaced.

2.2 Reconfigurable control

A reconfigurable controller deals with a fault by reconfiguring the control laws of the system. The reconfigurable controller has a basic set-up very similar to the one described above and again consists of three subsystems.

The fault detection and identification subsystems are as explained above in Section 2.1. Once it has been established that there is a problem and identification of it has taken place, the only remaining operation is the reconfiguration of the controller. In this subsystem one of a set of predefined reconfigurations, with different control laws, is used to accommodate the fault. This is achieved by using components of the system for purposes other than their designed task. The predefined reconfigurations being those predicted by the designer. Thus the system will only be able to handle as many problems as envisaged by the designer.

2.3 Restructurable control

The *modus operandi* of a restructurable controller is similar to that described in Section 2.1, and again the first two subsystems are used for detection and identification. Once again the difference lies in the third subsystem. This type of controller tries to restructure the control laws to accommodate the fault. This is achieved by using every available component of the vehicle for purposes other than their designed task in the same fashion as the reconfigurable controller. However, this method does not need any predefined sets of control laws, unlike the reconfigurable controller, and therefore may be very flexible at handling unanticipated faults. Such an approach lends itself to solution by artificial intelligence techniques.

3. Fault Tolerant Control Systems for Non-UUVs

The vast majority of research in this area has been concerned with plants other than UUVs and therefore this section has been included. It contains explanations of fault tolerant, reconfigurable and restructurable controllers used in other related systems to handle problems. The actual ideas discussed below, however, can be modified and applied to underwater vehicles.

3.1 Fault tolerant control

The fault tolerant controller, used in the regulation of the feedwater system in a four-loop pressurised water reactor power plant, presented by Eryurek and Upadhyaya [15], is capable of handling both sensor faults and controller failure. However, this controller cannot handle equipment malfunctions (actuator faults) or multiple simultaneous faults. The controller is made up of five major components:

1. Parallel control module
2. Signal validation module
3. Command validation module
4. Decision making module
5. System executive module

These five modules and the method by which they are connected can be seen in Figure 2. The control system has three different controllers working in parallel with each other. The system uses a method called horizontal redundancy to decide which of the controllers to use at any given time. Horizontal redundancy feeds different subsets of measurements to each of the controllers and then compares the outputs. When all the outputs are the same the horizontal redundancy procedure has no effect. When a fault has occurred one of the outputs changes and the horizontal redundancy procedure overrules the controller with a different output. This allows any fault to be overruled by the other controllers that have fault free information. The three controllers used in the example are a reconstructive inverse dynamics controller, a fuzzy logic controller and a conventional proportional integral derivative controller. These three different controllers provide different methods to reach the same result when controlling the plant. As they use different inputs a single fault will only affect one controller; this will then be out-voted by the other two unaffected controllers. The same result

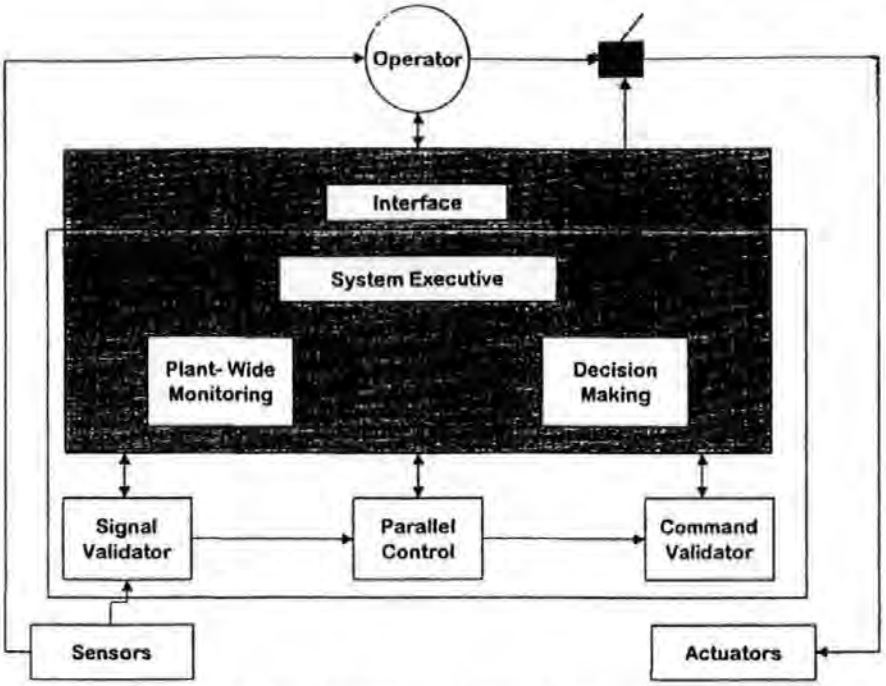


Figure 2 Integrated-system components for advanced plant control (after Eryurek and Upadhyaya [15]).

would be achieved if one of the controllers were to develop a failure during normal operations.

This is the main fault tolerant feature of the system, but one of the others is the signal validation module which uses two different routines, process empirical modelling and an ANN technique, to check against the actual output of the plant before going to the controllers. In a similar method the command validation module checks the outputs of each controller before the decision making module chooses the best response.

Tests were performed where faults were introduced to one of the controllers. The results show that the decision module was capable of handling these faults without the system becoming unstable. It was also shown that the decision module changed from one controller to another without a fault occurring; this took place because the controller's approach to the situation was an improvement on the original.

Despite this fault tolerant controller being designed for use in large scale systems, the idea of a parallel controller as presented here could easily be applied in the area of marine underwater technology. In addition, the ability to change from one controller to another just to improve the performance could provide an UUV with a much more flexible control structure.

Other papers in this area are [16] which gives a general review of fault tolerant controllers some of which use ANNs, and [17] which deals with sensor faults in a helicopter using ANNs.

3.2 Reconfigurable control

Rauch [18] considers autonomous control reconfiguration in relation to fault accommodation and learning systems. One of the approaches considered is multiple models, another is a single model with adaptive techniques for updating system characteristics. This may be considered a restructural technique and is not discussed here.

When considering the multiple model approach, the general form for a non-linear system (eqns (1) and (2)) is presented and the appropriate assumptions are stated.

$$x_{k+1} = f[x_k, u_k, q] + \text{plant noise} \quad (1)$$

$$z_k = h[x_k, u_k, q] + \text{measurement noise} \quad (2)$$

where q is the parameter values vector, x is the state vector, z is the measurements vector, u is the control vector, $u_k = g[x_{k^*}, q^*]$, which is a function of x^* (estimated state) and q^* (estimated vector parameters), and f and h are the state transition and measurement functions. The subscript k is the value at the kth time.

An example is then given for terminal guidance of an interceptor missile. The target that the missile is attempting to hit can make unknown manoeuves. The multiple models represent the sets of possible manoeuvres the missile can perform.

The multiple models are run off-line and a single extended Kalman filter is used on-line to measure the target state and compare with stored estimates from the off-line multiple models. The multiple models are generated using a general regression neural network.

At first glance this may not appear to be an RCS, but it does fit the definition given above. This example, however, is not concerned with handling a fault within the system, but is concerned with a changing variable outside of its control. It does have many models each with its

own control configuration and the system does change between them, depending on the situation.

A simulation was performed using an unmodified Kalman filter, the multiple model approach discussed above and a perfect guidance approach where the target trajectory is known exactly. The results showed that the multiple model had a hit probability of 50% which, as one would expect, is better than the original unmodified Kalman filter (15%), but worse than the perfect guidance (84%). The results presented show that an increase in performance is possible with minimal increase in on-line computation. It is suggested in the paper that fuzzy logic and artifical neural network techniques could be used in this approach to improve the given method and produce even better results.

Gao and Antsaklis [19] develop a different approach to RCS design, called perfect model following. This is a development of standard linear model following methods that are designed to make the output of the plant match the output of a model system with the desired behaviour. There is an explanation of standard linear model following methods, which states that they need both a feedforward and feedback controller in order to fulfil the task. The difference between a standard linear model following method and the perfect model following method is that for the latter the state variables of the model must match the state variables of the plant.

For the perfect model following to be achieved then Erzberger's conditions must be satisfied. These conditions are formed using equations given in [19], which describe the reference model and the plant coeffcients.

In order to satisfy Erzberger's conditions a system must have the same number of inputs as states. This is very rare and so it is very difficult to find an appropriate reference model that represents the desired dynamics and satisfies the conditions. However, even if the conditions are not satisfied then it is still possible to find a close solution by reducing the error to a minimum.

There are a couple of drawbacks with this method: if the conditions are not met then the system can become unstable and there is no control over the location of the poles of the system. The poles are the key to determine if a controller is stable; they are the roots of the characteristic equation associated with the system transfer function. If any of these poles are located on the right-hand side of the s-plane the system can then become unstable [6]. One advantage is that this type of control system is not very complex and the method does not use the output of the plant.

Kim *et al.* [20] use fuzzy logic and an ANN for the detection and isolation section of the reconfigurable control system, before the remaining section reconfigures the control laws of the system to handle failures. The block diagram of this system can be seen in Figure 3, with the neural network using the control signals and the measurable system outputs as its inputs. The ANN is originally trained off-line to detect faults but is then further trained on-line to update the network.

These ideas were then used in conjunction with a small waterplane area twin hull (SWATH) vessel. A back propagation ANN is used which has 16 inputs, two hidden layers with 16 processing elements in each layer, and one output. A fuzzy logic block takes the output for the ANN and decides if a failure has occurred.

ANNs are made up of a collection of neurons that are arranged in layers. All ANNs have an input and output layer; however, they also have

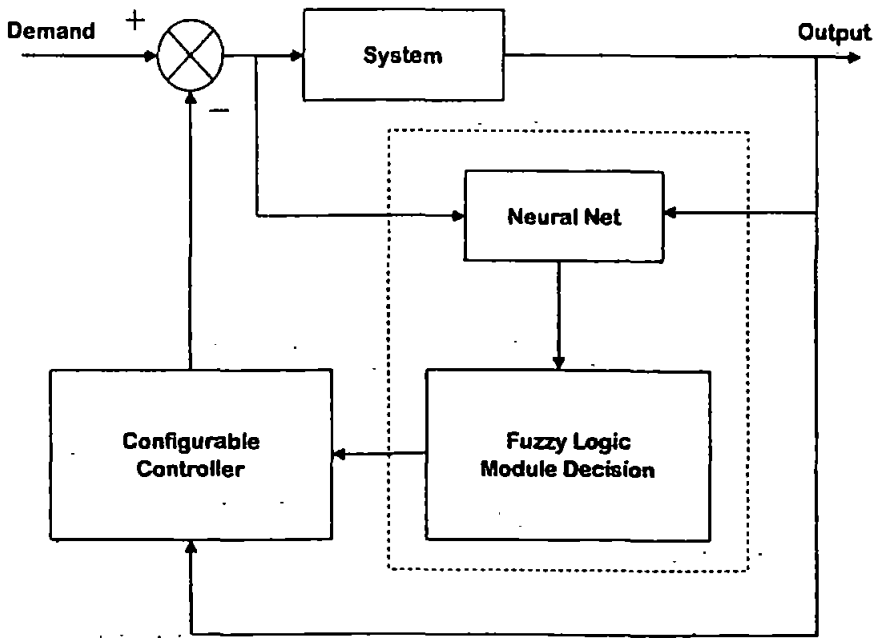


Figure 3 Reconfigurable control system set-up (after Kim *et al.* [20]).

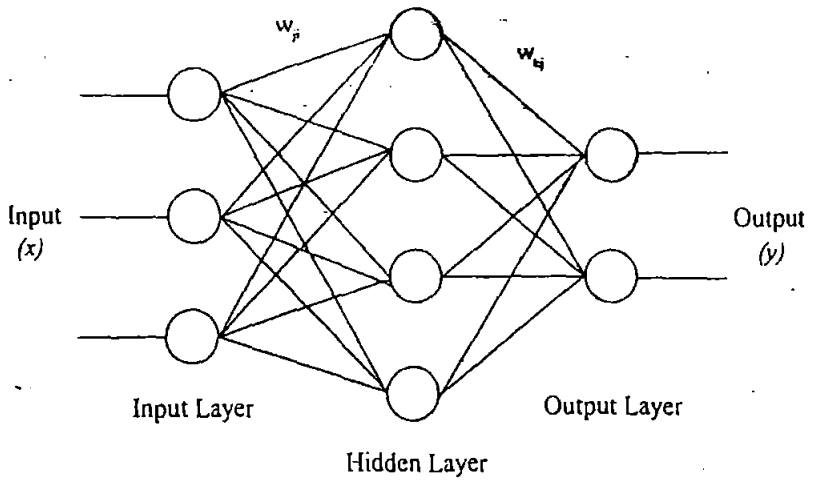


Figure 4 The feedforward multilayer perceptron.

a varying number of hidden layers depending on their function. For fault detection and isolation a single hidden layer is sufficient. An example of one type of network is the feedforward multilayer perceptron, which can be seen in Figure 4. Back propagation uses information that is fedback from the output layer to the input layer as a further input.

Two examples of failure detection using a SWATH vessel are then performed and explained. The first example shows how the failure has been detected but no reconfiguration occurs. For this example, the ANN detection system was trained on four sets of data with one control surface failing. A further 10 tests with varying levels of failure were then performed. It took the system between 8 and 44 seconds to detect the failure. No false alarms were recorded during any of these tests.

In the second example, reconfiguration does take place after the failure has been detected but there was no explanation of how the system is reconfigured to handle the failure. The same ANN is used for this example and one of the previous tests is repeated. The ANN and fuzzy logic subsystems take 19 seconds to detect the failure. After this, the reconfiguration subsystem makes the necessary modifications and the system returns to a stable state.

This system could be improved by training the ANN with further examples. This should speed up the process of identifying failures, but may increase the risk of a false alarm if the ANN was overtrained. Overtraining an ANN is where the network has been trained on the training data to such a degree that it will match it very well, but will lose its generalisation and thus, when checking other data, it would incur large errors. The fuzzy logic decision module could also be optimised after further testing. The paper does not explain the actual reconfiguration but does provide a good example of the other two subsystems used, i.e. failure detection and identification.

Other work in this area presents several different failure scenarios with respect to aircraft control reconfiguration during flight [4]. The aim was to control an aircraft after it had suffered surface damage and/or actuator failures by using some form of adaptive controller. Three algorithms are presented and compared. The result being that the direct input error algorithm is deemed the most applicable to the problem.

3.3 Restructurable control

Rauch [18] also considers a method that fits the definition of restructurable control. This is illustrated for a non-linear system using an adaptive controller in a SWATH ship. This uses three proportional integral-derivative controllers to control heading, pitch, and roll. It then has an adaptive controller consisting of two blocks, computation and decision, which choose the appropriate control laws for the conditions and adapt the control parameters continuously as the mission progresses. This can be seen in block diagram form in Figure 5.

The algorithm to generate the non-linear function used in the adaptive non-linear model is given but there are no examples of it in use. The model could be of any system that requires a restructurable controller. The basic idea of this algorithm is to take a set of training data and find a function that fits the data. This forms the basic starting model and then new information is input, as it becomes available. The model is then updated, with the new information being the most important, but the old information is also taken into account. This could be used in some kind of fault tolerant system, as the new information would be from the damaged system. It is suggested that fuzzy logic and artificial neural network techniques could be used to improve this approach.

Other papers in this field of research are by Looze *et al.* [21], who used linear quadratic design techniques to produce a control system

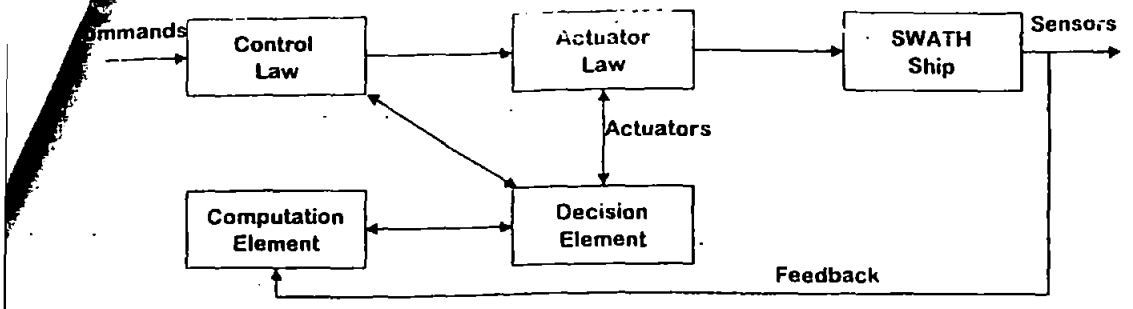


Figure 5 Block diagram of adaptive controller (after Rauch [18]).

for an aircraft suffering one or more control element failures, and by Ochi and Kanai [22] who present a restructurable flight control system based on a linearisation method.

4. Uninhabited Underwater Vehicles

This section considers the ideas from the previous sections and new ideas with respect to UUVs. Many of the concepts use a basic controller and then append a fault handling portion or redesign a standard controller. Eleven control architectures for underwater vehicles are given in [23], whilst [24] shows that the notion of an UUV being able to operate after a fault has occurred is a reasonable one. This is achieved by considering the six degrees of freedom that most UUVs have as a mathematical set of dynamic equations and showing that losing motion in one direction does not effect the domain of the set and hence is a recoverable fault. This is then demonstrated using a model that performs a yaw movement by using only roll and pitch movements. It is shown that by a positive roll and then pitch motion followed by a negative roll and then pitch motion, the nett motion of the underwater vehicle will be in the yaw direction.

4.1 Fault tolerant controllers

In the area of underwater vehicles, fault tolerant controllers can be considered the most general type of controller and use some of the simplest methods. Yang *et al.* [25] describe two fault tolerant systems for the ODIN (Omni-Directional Intelligent Navigator) vehicle. ODIN is an AUV with six degrees of freedom. These two fault tolerant systems focus on thruster and sensor failures in the vehicle. Results are presented to show the effectiveness of the systems.

The first fault tolerant system described is for ODIN's thrusters. This system uses the thruster control matrix (TCM) which represents the thruster output force to input force relationship. This matrix is used in the calculations to find the required thruster force for each thruster as shown in eqn (3):

$$[\text{Thruster force}] = [\text{TCM}]^{-1}[\text{Force input}] \quad (3)$$

This can then be used to find the correct input voltage to produce the required movement by ODIN.

There were two constraints placed on the fault detection and isolation subsystems. The first limits the number of thrusters that could fail during a mission, one vertical and one horizontal thruster. The second constraint is that once a thruster fault is detected then it is out of operation throughout the mission.

The fault detection and isolation processes are implemented as one process due to each thruster being fitted with its own Hall effect sensor [26]. The desired voltage is then compared with the voltage measured by the sensor, using the conditional algorithm shown below:

```
IF (Input Signal-Output Signal)/
  Input Signal > TOLERANCE
THEN count # of times TOLERANCE is
  continuously exceeded
IF # of times > TOLERANCE TIME
THEN send signal that Thruster is
  Faulty
ELSE reset counter and repeat
  monitoring routine
```

The thruster fault accommodation subsystem, having determined where the fault is located, then eliminates the corresponding column in the TCM and recalculates the required input voltage for the remaining thrusters. This effectively reconfigures the TCM to permit ODIN to continue with a mission.

Two tests where two of the thrusters failed during a simple mission, were performed and showed that the system could handle this fault by doubling the voltage to the remaining thrusters. This allowed the vessel to finish a mission and remain at the desired depth. These two tests do show that the system can handle some simple faults, but it is not shown if it could handle a fault where it is not possible to simply double the voltage to the remaining thrusters. This would be the case if the thrusters were already operating at their maximum when the fault occurred.

The second fault tolerant system presented is concerned with sensor faults. The sensor fault considered is in the heave direction, for which ODIN has two different sensors and one virtual

sensor obtained by an analytical model of itself. For this system three assumption are made about the fault that may occur. The fault is permanent, only one fault may occur and if a sensor is faulty, it is completely inactive and outputs zero. These assumptions limit the range of this system. It is not too hard to envisage a situation where the faulty sensor gives a reading of a fixed dc level or the fault is only temporary or intermittent.

Once again the fault detection and isolation processes are implemented as one process. This process is a series of IF–THEN logic rules that compare the outputs of the two sensors and one virutal sensor to determine which one has the fault.

The fault accommodation subsystem for this system could not be simpler, as the algorithm simply switches over to the good sensor and ignores the faulty sensor output. This shows why only one fault is allowed to occur, as a second fault would leave no good sensors left for the controller to use.

This subsystem was then tested for a fault occurring in each sensor. Both tests showed that the subsystem worked well with the only noticeable effect being the change in oscillation of ODIN. This was caused by the different characteristics of the sensors and not the fault tolerant system. The oscillation was greater when the sonar sensor was used. This was due to its relatively low resolution when compared to the pressure sensor.

The fault tolerant systems presented were reasonably effective at detecting, isolating and accommodating faults for the ODIN vehicle. These systems are unique to the ODIN vehicle, but the approach and concept can be extended for use in other underwater vehicles in order to deal with similar types of faults.

4.2 Reconfigurable controllers

This section is concerned with methods used in reconfigurable controllers and shows how they can be used with respect to UUVs in order to handle faults and in some cases improve general performance.

In the paper by Katebi and Grimble [27] a whole control scheme is proposed for an AUV, which is composed of three fully integrated layers. These can be seen in Figure 6. The top layer is the navigation layer and this is where all of the reconfiguration will take place. The middle layer is the guidance subsystem and the bottom layer is the AUV autopilot.

The AUV model used for this work is described by a set of non-linear differential equations. A linear state-space model of the system is then presented for use in the local controller and diagnostic subsystems.

The local controller is based on H-infinity theory [28] and is a trade-off between the plant controller and the diagnostic controller. This controller is designed particularly for the model used, but the same H-infinity approach could be used to design controllers for other UUV models.

The guidance system of this AUV is a predictive controller (PC) which does not suffer from the problems of a line of sight [29] algorithm. This is because the PC predicts where the AUV

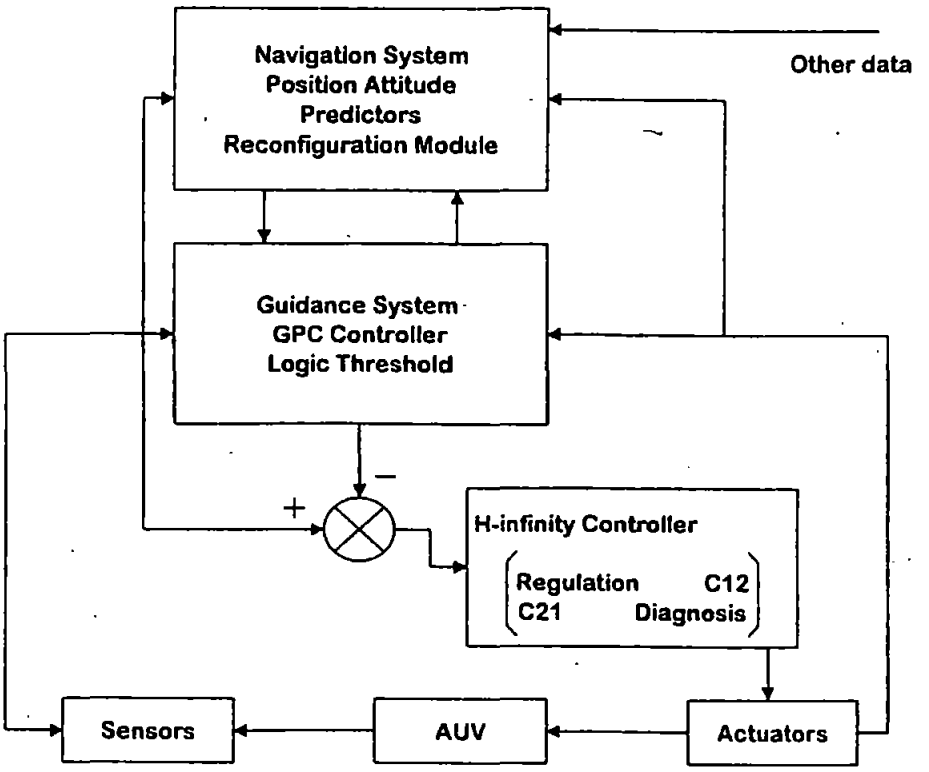


Figure 6 AUV control scheme (after Katebi and Grimble [27]).

will need to be and so has less overshoot when way points (target points for the AUV) are close together. The robustness of the H-infinity and PC are discussed with the outcome being that optimisation can be obtained by appropriate choice of weighting functions for the H-infinity controller and then suitable tuning of the PC.

The main area of interest, reconfigurable control, is then considered. Two approaches to reconfigurable control are examined: multiple models, and single models with adaptive techniques. For the multiple model approach a number of system models, each with their own corresponding control law, are first obtained and then a decision element chooses which is the most appropriate model and associated control law. For fault diagnosis a model must be included for each particular fault condition and a normal no-fault condition model.

The single non-linear model, which is continuously adapting, is then presented as the second approach to reconfigurable control. In general, the initial model is based on prior information and then continuously adjusted as new information is received. An example is given which uses a gain scheduling scheme.

Simulations were carried out to demonstrate the improved performance of the new controller compared with a simple H_2 controller [28]. The H_2 controller works in a similar manner to a H-infinity controller but is less complicated in its structure. A further simulation was performed in which the steering system developed a fault and the gyro failed simultaneously (sensor and actuator faults). The reconfiguration controller was activated when a set level had been passed in the heading. The new control system stabilised the plant after 30 seconds of activity.

The proposed scheme has some advantages over standard control strategies and is worthy of further research.

Tacconi and Tiano [30] explore reconfigurable control techniques applied to an AUV where they consider both sensor and actuator faults.

A mathematical model of an AUV using a mobile reference system is used in the study. The model of the vehicle uses six non-linear coupled equations and by considering a linearisation with respect to a given equilibrium condition, a state-space model of the type shown in eqn (4) is obtained.

$$\dot{x} = Ax + Bu + G\omega \tag{4}$$

where $x = [uvwpqr\phi\theta\varphi X_0 Y_0 Z_0]^T$ is an augmented state vector, which contains u, v, w the velocity components along the three axes; p, q, r the three components of pitch, roll and yaw angular velocities, θ, ϕ, φ the corresponding angles and X_0, Y_0, Z_0 the fixed reference system of the vehicles position; $u = [n\delta_1\delta_2\delta_3\delta_4]^T$ is the control vector, which contains n the propulsion rpm, δ_1 and δ_2 the upper and lower rudder angles, δ_3 and δ_4 the port and starboard elevators; ω is a vector which takes external disturbances into account. The three matrices, A, B and G are determined by both the hydrodynamic derivatives and the vehicle's speed.

A linear quadratic Gaussian (LQG) method, which is robust and can handle small faults, is discussed. This removes the need to consider such faults later and allows the focus to be on larger faults caused by system failures.

An LQG control system design requires the control vector be chosen during each transition from one way point of the AUV mission task to the subsequent one, in such a way that the expected value of a quadratic cost function J of state and control vector $u(t)$ is given by:

$$J = \sum \int_{t_0}^{t_f} (x^T(t)Qx(t) + u^T(t)Ru(t))\,dt \tag{5}$$

Different types of missions can be easily managed by a proper choice of the weighting matrices Q and R.

If it is assumed that $t_f \gg t_0$, then a computationally simpler problem can be solved, which supplies a linear feedback of the type shown in eqn (6).

$$u(t) = -K\hat{x}(t) \tag{6}$$

where the matrix K is obtained by solving a time-invariant Riccati equation, while $\hat{x}(t)$ is an optimal estimate of the state vector $x(t)$ supplied by a Kalman filter:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + H(y - C\hat{x}(t)) \tag{7}$$

The linear quadratic method for controlling plants is a robust form of controller and therefore very good at compensating for noise in the system. For further details of LQG control theory, see Dorato *et al.* [31].

Brief consideration is then given to integrated navigation systems, on-line monitoring and fault detection. The model uses Kalman filters in the navigation system and a statistical decision test for the fault detection and identification module.

When faults occur in the AUV, structural changes may occur in the mathematical model, which will cause the vehicle's performance to decrease to an irreparable level. The robust LQG design can be used to provide a reconfigurable controller. This is achieved by using a previously computed mathematical model to handle the new system. It computes the new model as required; however, it would be computationally advantageous to have the solutions stored in a look-up table.

These methods were originally proposed for use in the aerospace field and have easily been adapted for implementation in an AUV control system. Unfortunately, despite discussing the method in detail, no results are presented.

Derradji and Mort [32] describe and test two methods of reconfigurable control for a sub-

mersible vehicle, using an ANN approach. The traditional algorithm for an ANN controller used in such vehicles has problems if any form of control failure occurs. In order to deal with this, these two new methods are presented, both of which are capable of handling faults.

The first of these is the linear model-following approach, which uses a state-space model of the normal plant as the ideal model and the plant as a state-space model of the impaired model. Whilst in operation this approach modifies the signal from the neural network controller to make the impaired system act the same as the ideal system.

The second method is the error vector suppression (EVS) method. This approach simply disregards the error vector element in the adjustment algorithm and thus forces the impaired closed loop system output to be the same as the ideal systems output.

These two methods were tested using a linear multivariable state-space model of a large submarine vehicle.

For the tests, a three layer neural network with an input layer containing eight linear neurons, an output layer containing four linear neurons and 30 non-linear neurons in the hidden layer, were used as the controller. The tests involved simulating several levels of control surface failure, first in the rudder and then the starboard stern plane. Both methods provided satisfactory performance over a variety of conditions. The EVS and linear model following both managed to reconfigure the remaining control surfaces to accommodate the failures. There was very little to choose between the methods in performance, but the EVS used less computer memory and so makes it a much better option for real-time applications.

Ishii *et al.* [33] control an AUV using an adaptive ANN, which is continuously updated as the AUV operates. For the method described, an ANN is first trained off-line to be the controller for the given AUV and then a second ANN is used to model the output of the system, known as the identification network. These two ANNs act together almost as a single ANN to control the AUV. In the improved method, however, the identification network is regularly updated as the mission develops. This simply improves the performance of the controller, but could, with a little work, be adapted to become some form of reconfigurable controller or even a restructurable controller, where the ANN adapts to the new input of the now impaired system to restructure the controller and regain total control of the AUV.

Caccia and Veruggio [34] use a proportional plus integral-type guidance algorithm to control a prototype ROV's depth and motion. This controller has a three level hierarchical architecture. The motion, during operation, is estimated by a set of sensors, all with different capabilities. This information is compared with the mission tasks in order to keep a theoretical track of what the ROV has accomplished and is still required to undertake. Once the controller is informed of the intentions of the ROV, the information is used to compute the force and torque that must be applied to complete the given tasks. The next step performed by the controller is to take the required force and torque information and translate it into actuator outputs. This is where reconfiguration takes place. In theory, when an actuator fails, the translation will simply ignore that actuator as a possible output. As the same force and torque are still required, the workload will need to be redistributed between the actuators that remain functioning, hence reconfiguring the controller to cope with the loss of an actuator. Their reconfigurable controller is capable of handling total failure in one or more thrusters, as was shown in the presented test results.

5. Concluding Remarks

In this paper several different approaches to the problem of handling a fault in a given system are presented. The type of approach depends heavily on the system and its requirements. It can be seen that as the complexity and non-linearities in a system increase, the more elaborate both the control and fault tolerant subsystems have to become in order to cope with such phenomena. As UUVs tend to be highly non-linear, there is a need for most of their control strategies to be complex. Thus, many of the systems presented here have used concepts such as fuzzy logic and/or artificial neural networks in order to solve respective problems. In addition, several other different approaches to handling a failure within an UUV have been described. It is clear that there is considerable scope for research work to be undertaken in this area.

The simplest method of fault tolerance is to have more than one actuator for each process and to switch to the backup one after a failure occurs. This method is highly effective but is unacceptable in UUVs, as such an approach increases the payload of the vehicle and is expensive.

The most appropriate type of fault tolerant system for an UUV is a reconfigurable controller. This type of controller does not require the extra space required for redundant parts. It has been shown here that within certain limits most methods can recover the original performance after failure. The biggest drawback for this type of control system is its lack of ability to handle unpredicted failures. However, they are capable of handling a large range of fault conditions.

The restructurable controller should, in an ideal world, be a standard subsystem for all UUVs. This is not possible due to the large computer memory required to operate such subsystems. As computer memory is limited onboard

AUVs, a reconfigurable controller which has a comprehensive portfolio of stored predefined failure solutions, is an acceptable solution to the problem.

References

1. Eddy, P., Potter, E., and Page, P., 1976, *Destination Disaster*. Hart-Davis MacGibbon, London.
2. Antsaklis, P.J., Passino, K.M., and Wang, S.J., 1991, An introduction to autonomous control systems. *IEEE Control Systems*, June, pp. 5–13.
3. Rauch, H.E., 1994, Intelligent fault diagnosis and control reconfiguration. *IEEE Control Systems*, 14(3), pp. 6–12.
4. Bodson, M., and Groszkiewicz, J.E., 1997, Multivariable adaptive algorithms for reconfigurable flight control. *IEEE Trans. Control Systems Technology*, 5(2), pp. 217–229.
5. Fossen, T.I., and Fjellstad, O.E., 1996, Robust adaptive control of underwater vehicles: a comparative study. *Modeling Identification and Control*, 17(1), pp. 47–61.
6. Golten, J., and Verwer, A., 1991, *Control System Design and Simulation*. McGraw-Hill, Maidenhead.
7. Craven, P.J., Sutton, R., and Burns, R.S., 1998, Control strategies for unmanned underwater vehicles. *Journal of Navigation*, 51(1), pp. 79–105.
8. Kosko, B., 1994, *Fuzzy Thinking*. Flamingo, London.
9. Hassoun, M.H., 1995, *Fundamentals of Artificial Neural Networks*. MIT Press, London.
10. Basseville, M., 1988, Detecting changes in signals and systems—a survey. *Automatica*, 24(3), pp. 309–326.
11. Wang, H., 1995, Actuator fault diagnosis for nonlinear dynamic systems. *Trans. Inst. MC*, 17(2), pp. 63–71.
12. Patton, R.J., Chen, J., and Nielsen, S.B., 1995, Model-based methods for fault diagnosis: some guide-lines. *Trans. Inst. MC*, 17(2), pp. 73–83.
13. Naidu, S.R., Zafiriou, E., and McAvoy, T.J., 1990, Use of neural networks for sensor failure detection in a control system. *IEEE Control Systems*, April, pp. 49–55.
14. Alessandri, A., Caccia, M., and Veruggio, G., 1999, Fault detection of actuator faults in unmanned underwater vehicles. *Control Engineering Practice*, 7, pp. 357–368.
15. Eryurek, E., and Upadhyaya, B.R., 1995, Fault-tolerant control and diagnostics for large-scale systems. *IEEE Control Systems*, October, pp. 34–42.
16. Stengel, R.F., 1991, Intelligent failure-tolerant contol. *IEEE Control Systems*, June, pp. 14–23.
17. McLean, D., Saade, C., and Zinsch, F., 1997, A sensor fault accommodation system using a neural network. *Trans. Inst. MC*, 19(3), pp. 166–168.
18. Rauch, H.E., 1995. Autonomous control reconfiguration. *IEEE Control Systems*, December, pp. 37–47.
19. Gao, Z., and Antsaklis, P.J., 1992, Reconfigurable control system design via perfect model following. *Int.J. Control*, 56(4), pp. 783–798.
20. Kim, R.R., Ware, J.R., and Ammeen, E.S., 1997, Applications of neural networks and fuzzy logic in failure detection and fault tolerant control system design. *Eleventh Ship Control Systems Symposium*, Vol. 2, pp. 189–208.
21. Looze, D.P., Weiss, J.L., Eterno, J.S., and Barrett, N.M., 1985, An automatic redesign approach for restructurable control systems. *IEEE Control Systems*, May, pp. 16–22.
22. Ochi, Y., and Kanai, K., 1991, Design of restructurable flight control systems using feedback linearization. *Journal of Guidance*, 14(5), pp. 903–911.
23. Valavanis, K.P., Gracanin, D., Matijasevic, M., Kolluru, R., and Demetriou, A., 1997, Control architectures for autonomous underwater vehicles. *IEEE Control Systems*, 17(6), pp. 48–64.
24. Leonard, N.E., 1995, Control synthesis and adaptation for an underactuated autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 20(3), pp. 211–220.
25. Yang, K.C.H., Yuh, J., and Choi, S.K., 1999, Fault-tolerant system design of an autonomous underwater vehicle-ODIN: an experimental study. *Int.J. of Systems Science on UUV Control*, Special issue.
26. Hatton, K., and Fennell, H., 1999, 'The Hall effect.' http://math.sunyit.edu/projects/vector/hall.htm.
27. Katebi, M.R., and Grimble, M.J., 1999, Integrated control guidance and diagnosis for reconfigurable autonomous underwater vehicles control. *Int.J. of Systems Science on UUV Control*, Special issue.
28. Thompson, S., 1993, Robust control—an introduction. *Measurement and Control*, 26, pp. 235–241.
29. Healey, A.J., and Lienard, D., 1993, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3), pp. 327–339.
30. Tacconi, G., and Tiano, A., 1989, Reconfigurable control of an autonomous underwater vehicle. *Proceedings of 6th Int. Symposium of Unmanned Untethered Submersible Technology*, University of New Hampshire, pp. 486–493.
31. Dorato, P., Abdallah, C., and Cerone, V., 1995, *Linear-quadratic Control—An Introduction*. Prentice Hall, Englewood Cliffs, NJ.
32. Derradji, D.A., and Mort, N., 1996, Control reconfiguration in submersible vehicles using artificial neural networks. *13th World Congress of IFAC*, San Francisco, pp. 333–338.
33. Ishii, K., Fujii, T., and Ura, T., 1995, An on-line adaptation method in a neural network based control system for AUVs. *IEEE Journal of Oceanic Engineering*, 20(3), pp. 221–228.
34. Caccia, M., and Veruggio, G., 1999, Guidance and control of a reconfigurable unmanned underwater vehicle. *Control Engineering Practice* (to appear).

Proc. 14th Int. Conf. on Systems Engineering, ICSE'2000
Coventry University, UK, 12-14 September 2000

A KALMAN FILTER APPROACH TO FAULT TOLERANT CONTROL IN AUTONOMOUS UNDERWATER VEHICLES

A. R. Pearson[1], R. Sutton[1], R. S. Burns[1] and P. Robinson[2].

[1] Department of Mechanical and Marine Engineering
[2] Department of Communication and Electronic Engineering
University of Plymouth,
Drakes Circus, Plymouth,
Devon, PL1 8AA.
Tel: (01752) 232679 Fax: (01752) 232583 E-mail: a.pearson@plymouth.ac.uk

Keywords: Kalman filter, fault tolerance, AUV, sensor faults, actuator faults.

Abstract

This paper is concerned with using Kalman filter theory in a fault tolerant control scheme for a given autonomous underwater vehicle (AUV). Both sensor and actuator failures are simulated within the yaw and roll channels to test the ability of both a normal adaptive neuro-fuzzy inference system (AANFIS controller and the Kalman filter enhanced ANFIS controller. Results are presented to show how both the control systems cope with faults occurring within the sensor feedback system and the control actuators.

Introduction

In recent years, considerable interest has been shown into the commercial, scientific and naval use of AUVs. An AUV being a marine craft which fulfils a mission or task without being constantly monitored and supervised by a human operator.

AUVs have many uses in areas such as the fields of scientific research, military and commercial activity. They can operate in places that would otherwise be inaccessible, such as at extreme depths and/or in confined areas. Unfortunately, owing to the nature of these vehicles, data transmissions to and from the craft to the mother station through the sea water medium are poor. However, if these AUVs become damaged during operation in some way, they will be lost unless they have some form of fault tolerant control system on-board. It is therefore vital to the future work of AUVs that fault tolerant control systems, such as the Kalman filter approach discussed in this paper, are developed.

Within this paper, two of the most common types of failure will be considered. The first addressed are sensor failures in the yaw and roll channels. The following two types of faults considered are, percentage and intermittent signal failure. Both of these faults were considered over three yaw step inputs (10, 50, and 90 degrees) for the work with the yaw channel. The faults were considered over three initial roll angles (5, 15, and 30 degrees) for the roll channel. The second are actuator failures, again within both the yaw and roll channels. The faults considered are total actuator failure at a given angle and percentage actuator failure. These faults were also considered while the AUV was manoeuvring through the three step inputs for the yaw study and with three initial roll angles for the roll study.

The controllers used throughout this work are those developed by Craven [1] specifically for the given AUV and are based upon an ANFIS approach. The control systems were not designed to cope with the faults considered within this paper, however it was vital to test them to create a benchmark set of results for comparison.

The complete control authority of the AUV model is shown in Figure 1. It should be noted at this time that the yaw ANFIS controller only has command of the upper and lower canards, and that the roll ANFIS controller only has command of the port and starboard planes.

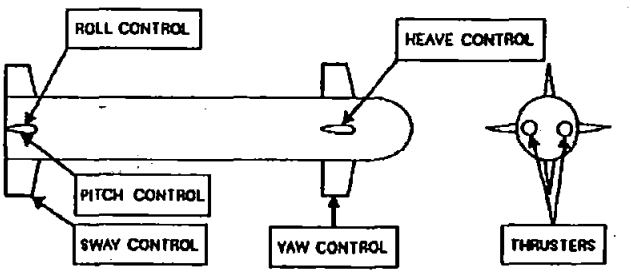


Figure 1. Control authority of the AUV

2 Fault tolerant controller

To attempt to cope with the failures it is proposed to place both a linear model of the AUV and a Kalman filter in-between the sensor feedback and the ANFIS controller. This improvement should give the control system a level of fault tolerance to the failures that are being considered. Before the Kalman filter could be developed several identification models around the given AUV were required. The optimum model developed of the yaw dynamics was the continuous linear state space model shown in equation (1). The optimum model developed of the roll dynamics was the continuous linear state space model shown in equation (2).

$$\dot{\psi} = \begin{bmatrix} 0 & 1 \\ 0 & -2.1326 \end{bmatrix} \psi + \begin{bmatrix} 0 \\ -0.9398 \end{bmatrix} \alpha \quad (1)$$

$$\dot{\phi} = \begin{bmatrix} 0.2210 & 1.1462 \\ -2.0047 & -0.9984 \end{bmatrix} \phi + \begin{bmatrix} -0.2252 \\ 1.7977 \end{bmatrix} \beta \quad (2)$$

Where $\dot{\psi}$ is the new vector for yaw and yaw rates, ψ is ne old vector for yaw and yaw rate, α is the control signal rom the ANFIS yaw controller, $\dot{\phi}$ is the new vector for roll nd roll rate, ψ is the old vector for roll and roll rate, and β is ne control signal form the ANFIS roll controller.

These two models were then used, in conjunction with ensor information from the AUV, to produce two Kalman ilters. One for the yaw channel and the other for the roll hannel. The standard use for a Kalman filter is to remove oise from sensor feedback within a given plant. However, or this paper the Kalman filter will be used in a slightly ifferent way. Instead of being used to filter noise from ensor feedback, it will attempt to compensate for failures ithin the AUV system. It will be using the linear models iven above (1) and (2). The Kalman filter will, when there is o fault present, produce a best estimate between the AUV nsor information and the linear model output and when a ult does occur it will compensate for it and allow the AUV continue operating.

In the interests of brevity the standard Kalman filter uations have been omitted however for the interested reader ese maybe found in [2].

Faults

o test the Kalman filters ability to handle faults it was cessary to create several failure scenarios. Sensor and tuator faults were considered for this paper. These two pes of faults will present the Kalman filter with two very fferent problems. The sensor faults will test the Kalman ters ability to recover lost information. The actuator faults ll change the way the AUV responds to control signals and ange its dynamic behaviour.

1 Sensor faults

ter researching many papers on the subject of sensor lure, the following types of faults were selected:

Percentage signal failure. [3] and [4]

Intermittent signal failure. [5]

For the yaw channel both of the types of sensor failure re considered for the yaw sensor over three step inputs (10, and 90 degrees). The failures were implemented in the w sensor, as opposed to the yaw rate sensor, due to the trollers bias towards the yaw information. Similarly for roll channel both types of faults were considered for roll sensor over three initial roll values (5, 15, and 30 rees). The failures were implemented in the roll rate sor, as opposed to the roll sensor, due to the controllers towards the roll rate information.

The first type of failure involved placing a gain on the sor feedback to permit only a given percentage of the al to return to the controller. The percentages considered e 100, 75, 50, 25 and 0. Where zero is total failure and 100% is no fault. This was achieved within a Matlab environment by setting up the system shown below in Figure 2 and putting the relevant value in the gain block. The roll rate sensor was set-up in an identical fashion for the relevant tests.

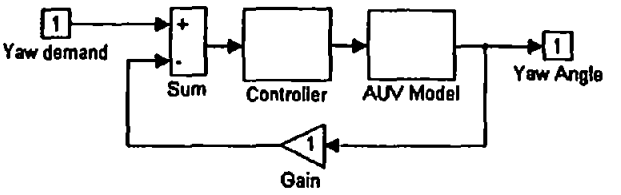


Figure 2. Gain on yaw sensor feedback for yaw channel

The second type of failure involved using a random signal to intermittently create a total signal failure on the given sensor. This sensor failure was implemented within the Matlab model as shown below in Figure 3. The roll rate sensor was set-up in an identical fashion for the relevant tests.

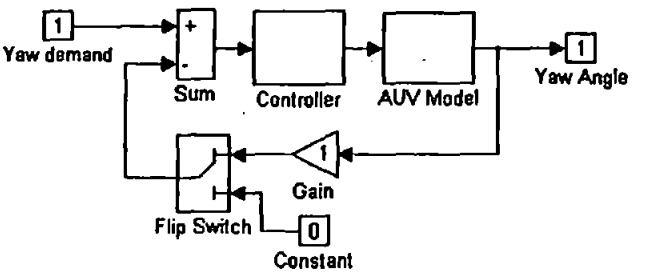


Figure 3. Flip switch between total and partial failure on yaw sensor

3.2 Actuator faults

Also to be considered are actuator failures, from a literature survey of the subject, the following types of faults were used:

(1) Total actuator failure at a zero angle. [6]

(2) Total actuator failure at a given angle. [7]

(3) Percentage actuator failure. [8]

For the yaw channel all of these failures were considered for the upper stern rudder which was used by the control system to control yaw, when being subjected to three step inputs (10, 50, and 90 degrees). For the roll channel the failures were implement in the same manor but in the starboard canard. Within the AUV model the actuators are formed using three blocks, a rate limiter, a saturation block and a transfer function, as shown in Figure 4. This is where all of the actuator failures will be implemented.

Figure 4. Actuator simulator within Matlab AUV model

The first type of failure is total failure of one control surface at zero degrees. This failure has been chosen to simulate the control surface becoming locked and thus being unable to move. This failure was accomplished by setting the input from the controller (see Figure 4) to zero.

The second failure is of a similar nature but involves the rudder being frozen at a given angle. This simulates the

ontrol surface becoming inoperable when the AUV is erforming a manoeuvre. This failure was achieved by setting he output to the AUV dynamics (see Figure 4) to the required ngle before the simulation began.

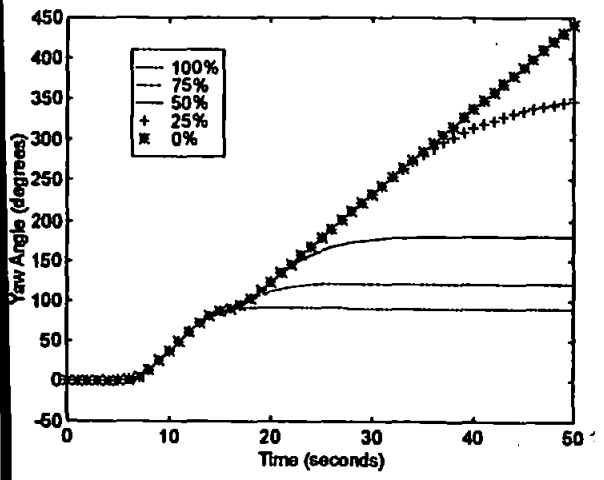
The third type of failure is to reduce the effectiveness of he controller by limiting the maximum angle it turn as a ercentage of the normal maximum angle (25.2 degrees for he yaw channel and 5 degrees for the roll channel). The ercentages considered were 75, 50, and 25. Where zero vould be total failure and 100 would be no fault. This failure vas achieved simply by altering the values within the aturation block shown in Figure 4.

4 Results

n the previous sections the Kalman filter, linear models and he faults to be used were presented. The faults were then imulated in the AUV and the two types of control systems ttempted to handle them.

4.1 Yaw sensor faults

he first type of sensor failure to be implemented was the ercentage signal failure. These failures were implemented 5 seconds into the simulation. The results for the standard NFIS controller attempting to achieve a yaw angle of 90 egrees can be seen in Figure 5. From the results it can be en that as the percentage of information which is being sent ack form the AUV decreases the standard ANFIS controller ecomes more unable to find the demanded yaw angle. This as at its worst when a total failure occurred, at which the NFIS controller sent the AUV into a circular path.



igure 5. The yaw responses of the AUV when the normal ANFIS control system is used

These results can be explained by a little bit of simple hematics. For all of the results shown above the control em believed it was reaching the demanded angle. This can shown by taking the final achieve yaw angle and tiplying it by the percentage signal failure. For all but the % gain failure this will give a final angle of 90 degrees. The Kalman filter was far better at handling this fault. The trol system using the Kalman Filter to attempt to correct for yaw sensor failures, was able to minimise the error occurring. For small errors the Kalman filter had only a small effect if any. However as the level of failure increased, so did the effect of the Kalman filter. This can be seen below in Figure 6.

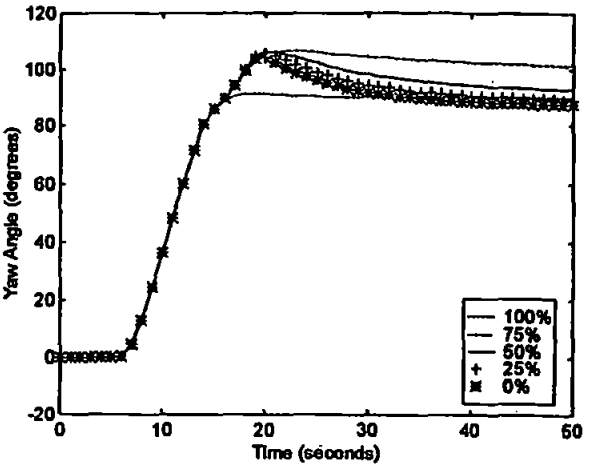


Figure 6. The yaw responses of the AUV when the Kalman filter is used in the control loop

At the point of failure the Kalman filter takes approximately 5 seconds to start compensating. The positive aspect of these results is that the control system does handle the failure, but the AUV does not achieve the demanded yaw angle of 90 degrees.

For the intermittent signal failure test the failure was implemented at the beginning of the simulation. The failure will not affect the AUV until the yaw angle moves away from zero. The results for both control systems attempting to handle this failure during manoeuvre consisting of a final demanded yaw angle of 90 degrees are shown below in Figure 7.

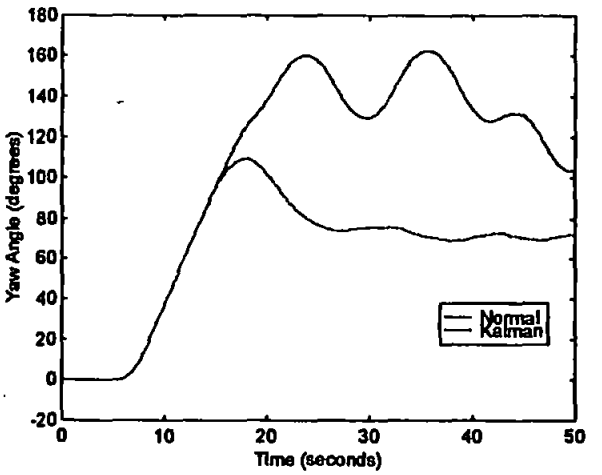


Figure 7. The normal and Kalman filter AUV yaw responses to the intermittent failure scenario

This shows how the Kalman filter enhanced control system was able to achieve and maintain an almost constant yaw angle with more success than the normal ANFIS control system. Despite not achieving the desired yaw angle the Kalman filter was did manage to stay closer than the standard ANFIS controller.

4.2 Roll rate sensor faults

For the roll rate sensor the first type of failure to be considered was the percentage signal failure. The failures, outlined above, were implemented after one second. This was because of the speed at which the ANFIS controller was able achieve the desired roll angle of zero degrees. The results for the standard ANFIS controller attempting to achieve the desired roll angle of zero degrees from the largest initial roll angle (30 degrees) can be seen in Figure 8.

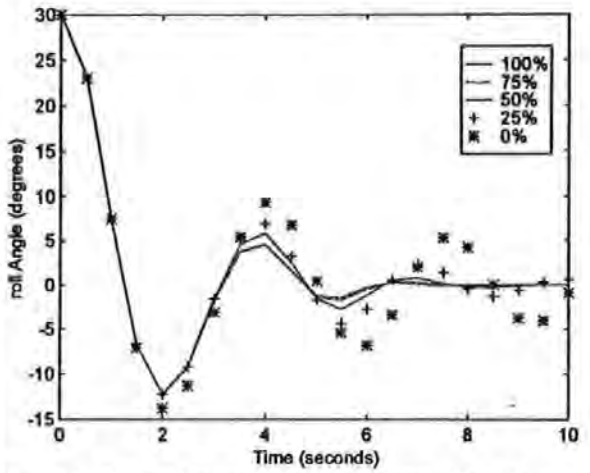


Figure 8. The roll responses of the AUV when the normal ANFIS control system is used

These results show how the controller became more unstable as the percentage of information the controller receives drops. Despite taking an increasingly longer time to settle down the AUV did always reach the required roll angle.

The same set of failures were then presented to the system when the Kalman filter was used. The results to this set of test can be seen below in Figure 9.

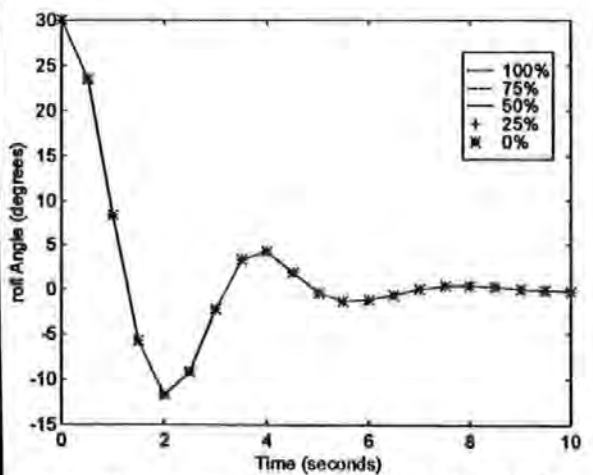


Figure 9. The roll responses of the AUV when the Kalman filter is used in the control loop

These results show how the Kalman filter was able to handle the loss of roll rate information was incredible ease. The fact that the AUV responded almost exactly the same for all levels of failure proves that the Kalman filter managed to apply the ANFIS controller the correct information regardless of what sensor information it received.

4.3 Yaw actuator faults

Next the actuator failures within the yaw channel are considered. All of the results presented in this section are for the AUV attempting to achieve a yaw angle of 90 degrees. Firstly the locked actuator set of failures as described above, are presented. When the actuator controlling the upper canard becomes locked at a given angle the standard ANFIS controller is able to achieve a yaw angle close to the demand for all but one case. This is when the upper canard is locked at –25.2 degrees. As one canard is locked at the maximum angle against the desired outcome, it is impossible for the lower canard to both compensate for this level of failure and then steer the AUV to the desired yaw angle using the given controller. For the other sizes of failure the AUV achieved a yaw angle around the desired mark. As can be seen in Figure 10.

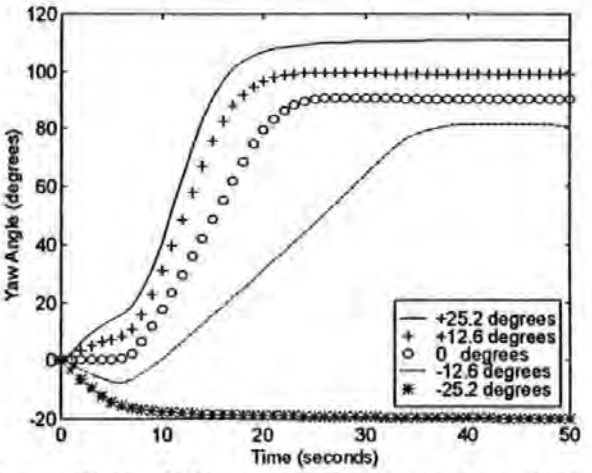


Figure 10. The AUV responses to the locked actuator set of failures using the standard ANFIS controller

The most important feature to notice from these results is the time taken to reach the final yaw angle. The worst case of this is when the rudder is fixed at –12.6 degrees, it takes the AUV almost 30 seconds to reach the yaw angle of just over 80 degrees.

For the same set of failures the Kalman filter controller performed poorly. It did manage to attain a constant yaw angle with the rudder locked at zero degrees, however it did not get as close to the demanded angle as the standard ANFIS controller. For all the other failures of this type the Kalman filter controller sent the AUV into a circular path in an attempt to achieve the demanded 90 degree yaw angle. As can be seen in Figure 11.

For the reduced maximum angle set of failure scenarios the standard ANFIS controller performed quite well. It reached the 90 degree yaw angle for every level of failure, but as the size of the failure increased the AUV was taking increasingly longer to reach that angle. This can be seen in Figure 12.

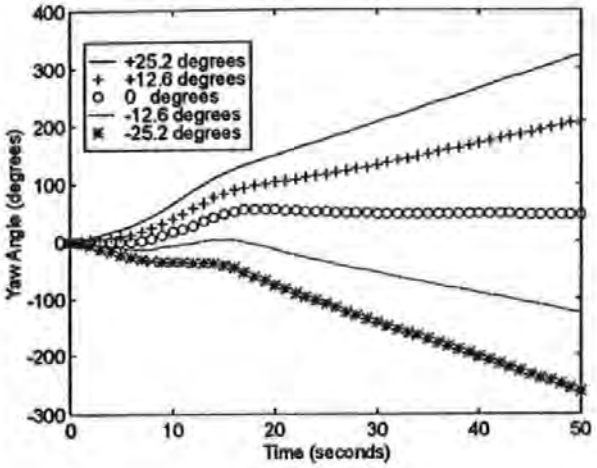


Figure 11. The AUV responses to the locked actuator set of failures using the Kalman filter controller

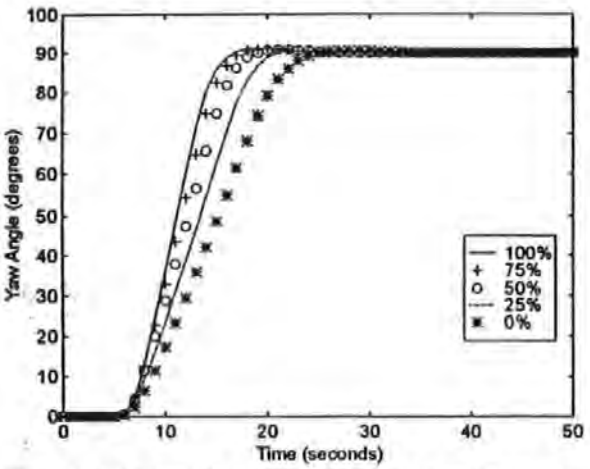


Figure 12. The AUV responses to the reduced maximum angle set of failures using the normal controller

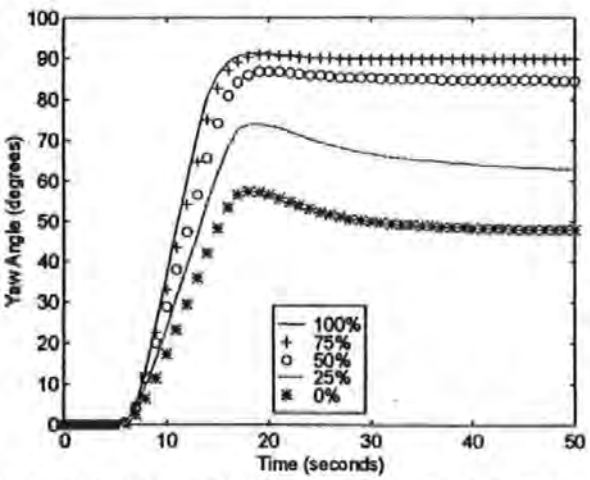


Figure 13. The AUV responses to the reduced maximum angle set of failures using the Kalman filter controller

As can be seen above in Figure 13 the Kalman filter control system did not have as much success with this set of failures. For the 100% and 75% failures the AUV did manage to achieve the demanded 90 degrees. However when the fault is increased the AUV's final yaw angle became less than the required 90 degrees. Just like for the standard controller the AUVs response becomes slower as the level of failure increases.

4.4 Roll actuator faults

The final collection of failures to be considered in this paper are the actuator failures in the roll channel. The actuator affected in these tests is the starboard stern plane. All of the results presented in this section are for the AUV attempting to achieve a roll angle of zero degrees from an initial value of 30 degrees. Firstly the locked actuator set of failures as described above, are presented. When the actuator controlling the starboard plane becomes locked at a given angle the standard ANFIS controller is able to achieve its aim only for the zero degrees failure. For all other failures AUV eventually maintains a roll angle of approximately one third the value of the angle of the locked plane. As can be seen in Figure 14.

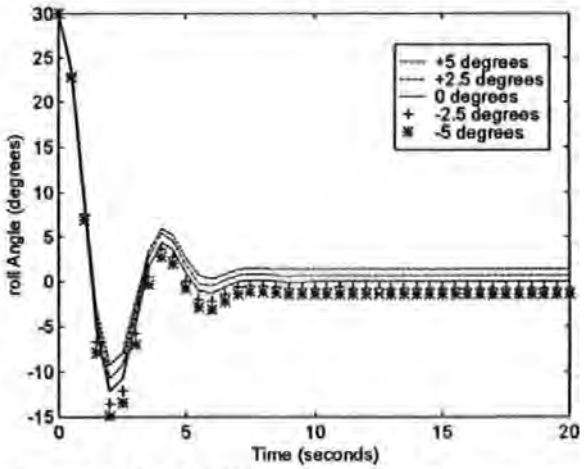


Figure 14. The ANFIS controller locked actuator results

The same set of failures were then simulated with the Kalman filter enhanced ANFIS control system used. These results were very similar to the previous set of results, the final roll angles were approximately half the value of the angle of the locked plane and it took almost five seconds longer for the roll angle to settle at its final value. This results can be seen in Figure 15.

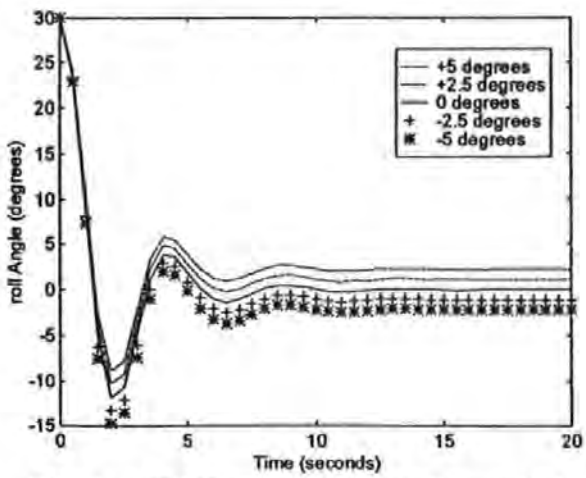


Figure 15. The Kalman filter locked actuator results

The starboard stern plane was then subjected to the reduced maximum angle set of failure scenarios. The standard ANFIS controller performed quite well. It reached the desired zero degree roll angle for every level of failure, but as the size of the failure increased the AUV was taking increasingly longer to reach that angle. Also it should be noted that as the level of failure increased then the overshoot also increased. These results can be seen in Figure 16.

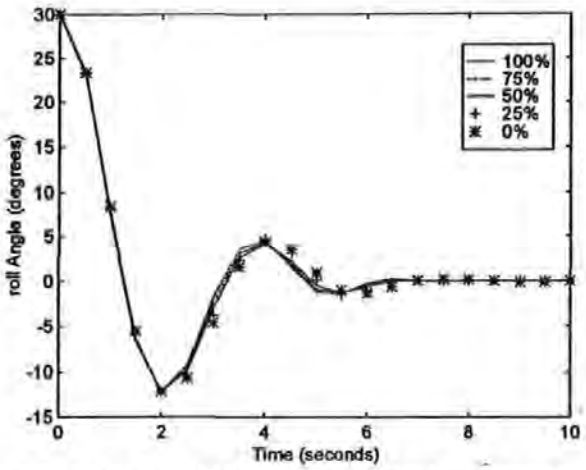


Figure 16. The ANFIS controller reduced maximum angle results

The same set of failures were then simulated using the Kalman filter enhanced ANFIS control system. These results were very similar to the previous set of results. The major difference when the Kalman filter was used, was that it took over two seconds longer for the roll angle to settle at its final value. This results can be seen in Figure 17.

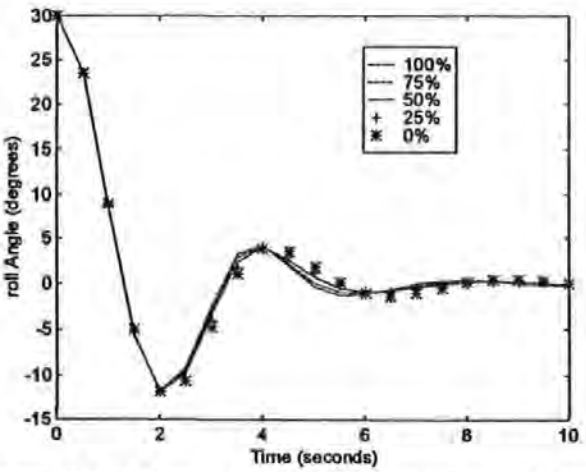


Figure 17. The Kalman filter reduced maximum angle results

5 Conclusions

From the results given in this paper there are several conclusions which can be drawn. Firstly the Kalman filter showed great promise for recovering sensor information. This was highlighted by the results from the roll rate failures where the Kalman filter was almost unaffected by the failures.

The Kalman filter did manage to handle the roll actuator failures, but not as well as the standard ANFIS controller. The Kalman filters worst test results were for the yaw actuator failures, were it was unable to control the AUV with any level of success.

The control system could be more fault tolerant if it was able to control some of the other actuators shown in Figure 1. The control system may also be improved by using a fuzzy logic system to attempt both sensor and actuator failures.

Acknowledgements

The authors wish to thank the Sea Systems Sector, DERA, Winfrith, for supplying the AUV model.

References

[1] Craven, P., Intelligent control strategies for an autonomous underwater vehicle, *Ph.D. thesis, University of Plymouth, U.K,* (1999).

[2] Dean, G., An introduction to Kalman filters, *Measurement and Control,* Vol. 19, 69-73. (1986).

[3] Napolitano, M., Windon II, D., Casanova, J., Innocenti, M, and Silvestri, G., Kalman filters and neural-network schemes for sensor validation in flight control systems, *IEEE Transactions on Control Systems Technology,* Vol. 6, No 5, 596-611. (1998).

[4] Yang, K., Yuh, J. and Choi, S., Fault-tolerant system design of an autonomous underwater vehicle- ODIN: an experimental study, *Int. J of Systems Science,* Vol. 30, No 9, 1011-1019. (1999).

[5] McLean, D., Saade, C. and Zinsch, F., A sensor fault accommodation system using a neural network, *Trans. Inst. MC,* Vol. 19, No 3, 166-168. (1997).

[6] Looze, D., Weiss, J., Eterno, J. and Barrett, N., An automatic redesign approach for restructurable control systems, *IEEE Control Systems Magazine,* May, 16-22. (1985).

[7] Ochi, Y. and Kanai, K., Design of restructurable flight control systems using feedback linearization, *Journal of Guidance,* Vol. 14, No 5, 903-911. (1991).

[8] Derradji, D. and Mort, N., Control reconfiguration in submersible vehicles using artificial neural networks. *13th World Congress of IFAC,* 333-338. San Francisco 96, (1996).

A FUZZY FAULT TOLERANT CONTROL SCHEME FOR AN AUTONOMOUS UNDERWATER VEHICLE

A.R.Pearson[1], R.Sutton[1], R.S.Burns[1] and P.Robinson[2].

[1] *Department of Mechanical and Marine Engineering*
[2] *Department of Communication and Electronic Engineering*
University of Plymouth,
Drakes Circus, Plymouth,
Devon, PL1 8AA.
Tel: (01752) 232679 Fax: (01752) 232583
E-mail: a.pearson@plymouth.ac.uk

Abstract: This paper is concerned with using fuzzy logic in a fault tolerant control scheme for a given autonomous underwater vehicle (AUV). Actuator failures are simulated within the yaw channel to test the ability of both a normal adaptive neuro-fuzzy inference system (ANFIS) controller and a fault tolerant fuzzy control system tuned using a simulated annealing algorithm. Results are presented to show how each of the control systems coped with faults occurring within one of the two control actuators of the AUV.

Keywords: autonomous underwater vehicle, fault tolerant, fuzzy logic, simulated annealing.

1. INTRODUCTION

Commercial, naval and scientific operational specifications for uninhabited underwater vehicles (UUVs) continue to become more challenging in line with the advances being made in the field of control engineering. In order to survive actuator failure during a mission, such vehicles need to possess either a reconfigurable or a fault tolerant control system.

In recent years, considerable interest has been shown into the commercial, scientific and naval use of UUVs. UUV being a generic expression to describe both an AUV and a remotely operated vehicle (ROV). An AUV is a marine craft which fulfils a mission or task without being constantly monitored and supervised by a human operator, whilst a ROV is a marine vessel that requires instruction from an operator via a tethered cable.

Demands are growing for the requirement of AUVs to be able to operate effectively at extreme depths and/or in confined areas such as under packed ice. Unfortunately communication to and from the craft to the mother station through the sea water medium are poor. Thus, an AUV has to be self-sufficient for the duration of a mission. Fault tolerant control systems could also be beneficially installed within ROVs. Their employment within ROVs would significantly lighten the workload of the human operators allowing them to concentrate on the mission, whilst permitting them to maintain overall supervisory control.

This paper starts with the dynamics of the vehicle and the set-up of both the control system and the faults to be considered. Then the approach used to make the AUV fault tolerant is described. Finally the results are presented to show how the controllers handle the faults.

2. AUV DYNAMICS

The standard controller referred to throughout this paper is that developed by Craven (1999) specifically for the given AUV model used within this study. It is an intelligent fuzzy logic controller developed using the ANFIS approach (Jang 1991). The control system was not designed to cope with the types of fault

which are considered within this paper, however it was vital to test this controller to create a benchmark set of results. These results will be used for comparisons with the new control system design.

2.1 The Control Authority

The complete control authority of the AUV model is shown in Fig 1. It should be noted at this time that both the standard yaw ANFIS controller and the new fuzzy fault tolerant controllers only have command of the upper and lower canards (forward control surfaces) during this work.

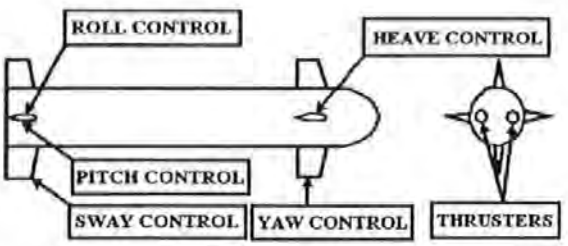


Fig. 1. Control Authority of the AUV

2.2 The Yaw Controller

The fuzzy logic rules for the standard ANFIS yaw controller used in this paper are given below in equation 1.

$$
\begin{aligned}
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.4863\,\psi_\varepsilon - 0.8791\,\dot{\psi} - 0.02926\\
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4890\,\psi_\varepsilon - 0.9021\,\dot{\psi} + 0.001381\\
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.4858\,\psi_\varepsilon - 0.8962\,\dot{\psi} + 0.003143\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.2994\,\psi_\varepsilon - 0.7034\,\dot{\psi} - 0.1227\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4879\,\psi_\varepsilon - 0.8910\,\dot{\psi} + 0.003723\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.3053\,\psi_\varepsilon - 0.3055\,\dot{\psi} - 0.03744\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.5902\,\psi_\varepsilon - 0.8387\,\dot{\psi} - 0.1172\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4811\,\psi_\varepsilon - 1.081\,\dot{\psi} - 0.06111\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.6596\,\psi_\varepsilon - 1.311\,\dot{\psi} + 0.7814
\end{aligned}
\qquad (1)
$$

Where ψ_ε is the yaw angle error given by yaw angle demand minus actual yaw angle, $\dot{\psi}$ is the yaw rate, δ is the desired rudder angle, Neg stands for negative and Pos stands for positive.

3. FAULT TOLERANT CONTROLLER

To attempt to cope with the failures it is proposed to situate a fuzzy logic system between the ANFIS controller and the AUV dynamics. This will multiply the control signal to the actuators, as shown in Fig 2. This improvement should give the control system a level of fault tolerance to the failures that are being considered in this paper.

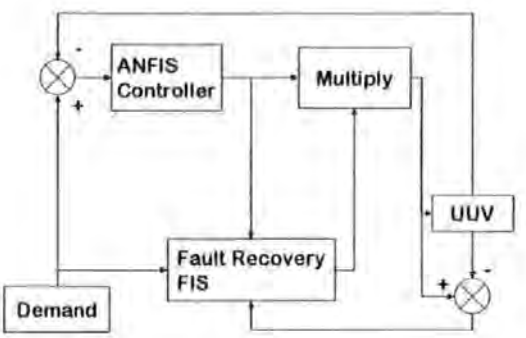


Fig. 2. Fault Tolerant Control System Set-up

The fuzzy logic system can be seen to take three batches of information for use in determining the degree of failure which has occurred and then makes the necessary correction for it. The three pieces of information the system will have as inputs are, (1) the demand being placed on the standard ANFIS controller, (2) the demand being placed on the actuators by the standard ANFIS controller, (3) the error in the damaged actuator. The fuzzy logic system compensates for this and returns the AUV to a fully functional or near fully functional situation. When there is no fault present, the fault tolerant system will have a minimal affect on the ANFIS control signal. However when a fault does occur it will compensate for it and allow the AUV to continue operating and complete its mission.

For this paper two fuzzy fault tolerant systems were tested. The first was a heuristic fuzzy inference system (FIS). This was designed by taking a simple approach to the problem, the bigger the error, the larger the value the control signal must be multiplied by in order to compensate. This was then tuned using the simulated annealing method (Kirkpatrick, *et al* 1983) to create the second FIS.

4. ACTUATOR FAULTS

To test the fault recovery ability of the fuzzy logic controllers to handle faults it was necessary to create several failure scenarios. Faults occurring within only one of the two control actuators are considered in this paper. The actuator faults will change the way the AUV responds to control signals and will also affect the dynamic behaviour of the AUV. The fault used here is one whereby one of the two control surfaces suffers a loss of efficiency (LOE) (Mort, and Derradji, 1999). The upper canard was the actuator chosen for the faults to be simulated within.

When the AUV is undamaged the LOE for each of the actuators would be 0%. As an actuator becomes more damaged this value will rise up to a maximum of 100%, which would be total failure where the actuator would not be able to move.

For this study, all of the work has been executed within the yaw channel. The same principles and techniques can be used in any of the other channels to produce similar results. The failures have been simulated in the upper canard which is used to control yaw. The system was tested by being subjected to three step inputs (10, 20, and 30 degrees). Within the AUV model the actuators are formed using three blocks, a rate limiter, a saturation block and a transfer function, as shown in Fig 3.

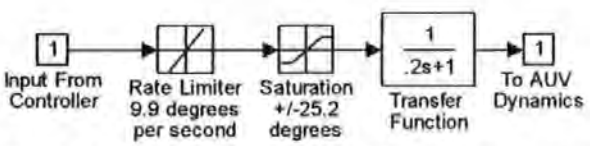


Fig. 3. Actuator simulator within Matlab AUV model

The first type of failure is total LOE, which can also be thought of total failure of one control surface at zero degrees. This failure has been chosen to simulate the control surface becoming locked. This failure was accomplished by setting the input from the controller (see Fig 3) to zero.

The second type of failure is to reduce the effectiveness of the controller. The percentages of LOE considered for this work were 75%, 50%, and 25% (Mort, and Derradji, 1999). There are three types of failure which can be implemented using this approach for actuator failures. These failures can be achieved by altering the values within the saturation and the rate limiter blocks shown in Fig 3. For this paper the actuator failures were simulated firstly by altering only the value in the saturation block (shown in the tables as Saturation), secondly by altering only the value in the rate limiter block (shown in tables as Rate limiter), and thirdly by altering both values simultaneously to the same level of failure (shown in tables as Both).

5. RESULTS

The standard non-fault tolerant ANFIS controller and the fault tolerant controllers were all subjected to the complete range of faults described it the previous section using the Matlab SIMULINK computer package.

5.1 The Standard ANFIS Controller

The first set of tests were performed on the standard ANFIS control system. Below the performance of this controller can be seen as it attempts to achieve the 30 degree demanded yaw angle while enduring an increase in the level of failure (using the failure defined as Both) for successive simulations (Fig 4).

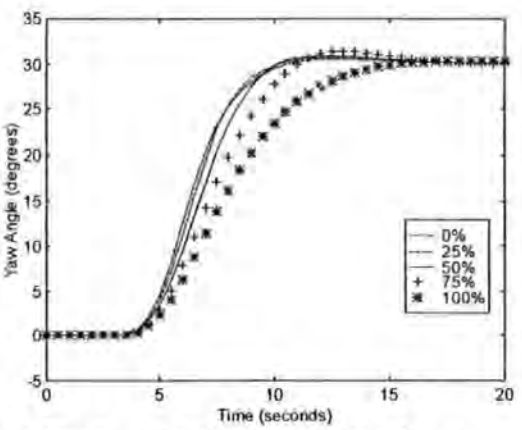


Fig. 4. The Standard ANFIS Controller System Attempting to Handle Various Levels of the Both Failure Scenario.

In order to make comparison of the controllers easier two forms of information have been recorded. Firstly the root mean squared error (RMSE) value, between the desired and the actual results was calculated. The RMSEs for the standard control system, for all of the given faults for the given demanded yaw angles, have been compiled in Table 1 shown below.

Table 1 The RMSEs (degrees) for the Complete Set of Results for the Standard ANFIS Controller

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0	0	0.099	1.065
	Rate limiter	0.075	0.188	0.380	1.065
	Both	0.075	0.188	0.380	1.065
20°	Saturation	0	0.127	0.741	2.360
	Rate limiter	0.205	0.546	1.080	2.360
	Both	0.205	0.546	1.132	2.360
30°	Saturation	0.094	0.691	1.904	4.125
	Rate limiter	0.434	1.088	2.172	4.125
	Both	0.436	1.172	2.390	4.125

The second type of information used to evaluate the systems was the rise times. Where the rise time is defined as the time taken for the AUV to rise from five to ninety five percent of the demanded yaw angle. The rise times for the standard control system, when attempting to handling the all of the given faults for the given demanded yaw angles, have been compiled in Table 2 shown below.

Table 2 The Complete Set of Rise Times (seconds) of The Chosen Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Sat	4.5	4.5	4.7	7.1
	Rate limiter	4.6	4.7	4.8	7.1
	Both	4.6	4.7	4.8	7.1
20°	Sat	4.8	4.9	5.4	7.5
	Rate limiter	4.7	4.7	4.9	7.5
	Both	4.7	4.7	5	7.5
30°	Sat	4.9	5.3	6.1	7.9
	Rate limiter	4.3	4.5	5.3	7.9
	Both	4.4	4.7	5.7	7.9

5.2 The Heuristic Controllers

Several heuristic controllers were created as starting points for the tuning programs. The first of them was to have a FIS whose starting point did not affect the output of the standard controller. As expected this gave results identical to the standard ANFIS controller shown above. The others all worked on the principle that as the size of fault increased then the signal must also be increased. Below is shown the results for one such FIS attempting to compensate for the set of failures defined above as Both over the 30 degree step input (Fig 5).

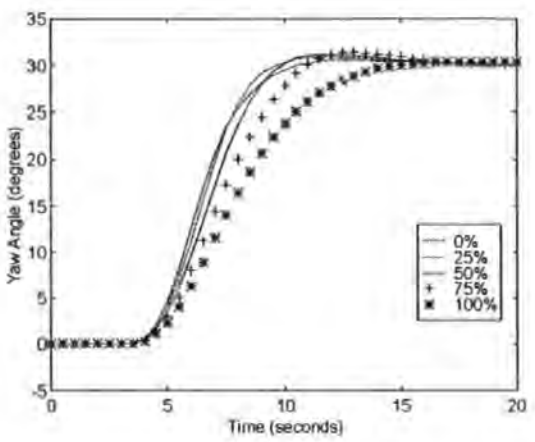


Fig. 5. A Heuristic Control System Attempting to Handle Various Levels of the Both Failure Scenario.

The complete set of RMSEs for the same heuristic controller when subjected to all the given failure scenarios have been compiled below in Table 3.

Table 3 The Complete Set of RMSEs (degrees) of The Chosen Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0.004	0.004	0.095	1.024
	Rate limiter	0.072	0.183	0.367	1.024
	Both	0.072	0.183	0.367	1.024
20°	Saturation	0.112	0.113	0.711	2.255
	Rate limiter	0.181	0.494	1.051	2.255
	Both	0.181	0.494	1.098	2.255
30°	Saturation	0.060	0.603	1.829	4.001
	Rate limiter	0.548	1.120	2.114	4.001
	Both	0.529	1.145	2.320	4.001

5.3 The Tuned Controllers

The heuristic FIS mentioned above was then tuned for 500 epochs using a set of training data, which comprised of various sizes of steps with different levels of failures occurring. The FIS was tuned using the simulated annealing method (Kirkpatrick 1983). The FISs which performed best on the training data were then re-tested on the failures used above. After the training and testing of the FISs had taken place it was possible to determine which of the FISs handled the faults optimally. This was done by taking into account the performance of each FIS with respect to all of the faults. The fuzzy logic rule base for this system is shown below in equation 2.

If δ is Neg and ψ_ε is Neg and η is Small then β =0.9784

If δ is Neg and ψ_ε is Neg and η is Medium then β =1.0048

If δ is Neg and ψ_ε is Neg and η is Big then β = 1.6817

If δ is Neg and ψ_ε is Zero and η is Small then β = 1.0810

If δ is Neg and ψ_ε is Zero and η is Medium then β = 0.9574

If δ is Neg and ψ_ε is Zero and η is Big then β = 1.3934

If δ is Neg and ψ_ε is Pos and η is Small then β = 1.3150

If δ is Neg and ψ_ε is Pos and η is Medium then β = 1.0566

If δ is Neg and ψ_ε is Pos and η is Big then β = 0.9854

If δ is Zero and ψ_ε is Neg and η is Small then β = 0.9792

If δ is Zero and ψ_ε is Neg and η is Medium

then $\beta = 1.1607$
If δ is Zero and ψ_ε is Neg and η is Big
then $\beta = 1.3252$
If δ is Zero and ψ_ε is Zero and η is Small
then $\beta = 0.8898$
If δ is Zero and ψ_ε is Zero and η is Medium
then $\beta = 1.5735$
If δ is Zero and ψ_ε is Zero and η is Big
then $\beta = 1.3664$ (2)
If δ is Zero and ψ_ε is Pos and η is Small
then $\beta = 1.0124$
If δ is Zero and ψ_ε is Pos and η is Medium
then $\beta = 1.0105$
If δ is Zero and ψ_ε is Pos and η is Big
then $\beta = 1.2517$
If δ is Pos and ψ_ε is Neg and η is Small
then $\beta = 0.9901$
If δ is Pos and ψ_ε is Neg and η is Medium
then $\beta = 1.0883$
If δ is Pos and ψ_ε is Neg and η is Big
then $\beta = 1.1980$
If δ is Pos and ψ_ε is Zero and η is Small
then $\beta = 1.0779$
If δ is Pos and ψ_ε is Zero and η is Medium
then $\beta = 1.0762$
If δ is Pos and ψ_ε is Zero and η is Big
then $\beta = 1.3498$
If δ is Pos and ψ_ε is Pos and η is Small
then $\beta = 0.9751$
If δ is Pos and ψ_ε is Pos and η is Medium
then $\beta = 1.0641$
If δ is Pos and ψ_ε is Pos and η is Big then $\beta = 1.3499$

Where Pos, Neg, δ and ψ_ε are as defined above and η is the error in actuator and β is the control signal multiplier.

The results, expressed as RMSEs, for the chosen FIS are shown below in Table 4.

Table 4 The Complete Set of RMSEs (degrees) for The Best Tuned Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0.016	0.016	0.101	0.820
	Rate limiter	0.079	0.158	0.317	0.820
	Both	0.079	0.158	0.317	0.820
20°	Saturation	0.024	0.101	0.657	1.949
	Rate limiter	0.172	0.469	0.995	1.949
	Both	0.172	0.470	1.036	1.949
30°	Saturation	0.070	0.557	1.695	3.749
	Rate limiter	0.579	1.174	2.062	3.749
	Both	0.557	1.152	2.210	3.749

Below is shown the results for this best tuned FIS attempting to compensate for the set of failures defined above as Both over the 30 degree step input (Fig 6).

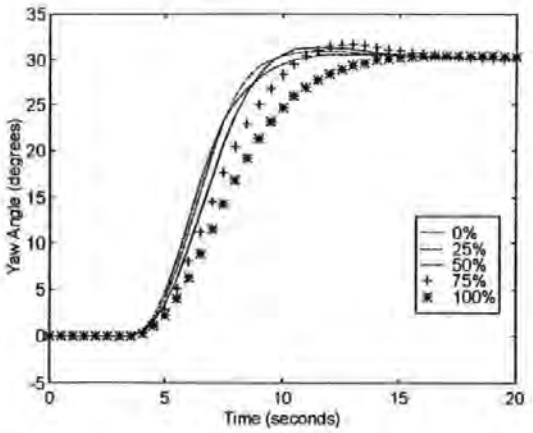


Fig. 6. The Best of the Tuned Control Systems Attempting to Handle Various Levels of the Both Failure Scenario.

The rise times for the chosen FIS, when attempting to handling the all of the given faults for the given demanded yaw angles, have been compiled in Table 5 shown below.

Table 5 The Complete Set of Rise Times (seconds) of The Best Tuned Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Sat	4.6	4.6	4.7	6.6
	Rate limiter	4.7	4.7	4.8	6.6
	Both	4.7	4.7	4.8	6.6
20°	Sat	4.9	4.9	5.4	6.9
	Rate limiter	4.7	4.6	4.8	6.9
	Both	4.7	4.6	5	6.9
30°	Sat	5	5.2	6	7.4
	Rate limiter	4.3	4.3	5	7.4
	Both	4.3	4.6	5.5	7.4

The RMSEs displayed within Table 1 and Table 4 were then used to calculate the percentage increase in performances achieved by the tuned FIS for the faults which have been considered in this paper. These percentages are shown below in Table 6.

Table 6. The Percentage Increases For the Best Tuned Fuzzy Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation			-2.3%	23.0%
	Rate limiter	-0.5%	16.4%	16.6%	23.0%
	Both	-0.5%	16.4%	16.6%	23.0%
20°	Saturation		20.5%	11.3%	17.4%
	Rate limiter	16.0%	14.1%	7.9%	17.4%
	Both	16.0%	14.0%	8.5%	17.4%
30°	Saturation	26.4%	19.4%	11.1%	9.1%
	Rate limiter	-33.6%	-7.9%	5.1%	9.1%
	Both	-27.8%	1.7%	7.5%	9.1%

6. ANALYSIS

Fig 4 and Tables 1 and 2 show how the standard ANFIS controller struggled to cope with the faults presented to it. It was also clear that as the level of faults increased then the performance of the controller degraded and that the rise times increased. This was certainly to be expected due to the design used for the standard ANFIS controller used for the AUV.

It has then been shown via Fig 5 and Table 3 that even a basic heuristic FIS can enhance the fault tolerant performance of the AUV.

Finally the performance of the best-tuned FIS was displayed. The results of these tests showed how the simulated annealing program had adapted the FIS in order improve the performance of the AUV. The most significant improvement was 26.4%, which was in the saturation fault of 25% over a 30 degree step input. With the worst result being a drop in performance of 33.6% in the rate limiter fault of 25% over a 30 degree step input. This may be due to the starting FIS used in the tuning. From Table 3 it is clear that a large RMSE is produced by this fault. The training does improve the results, but not to a sufficient level to produce an improvement in all of the faults considered.

The rise times shown in Tables 2 and 5 present a far better view of the tests, with two thirds of the rise times being reduced, and only four cases show an increase in time taken. This shows that the FIS is forcing the AUV to turn slightly faster, to try and compensate for the faults.

7. CONCLUSIONS

The analysed results show how the performance of the AUV has been enhanced by the fault tolerant FIS. The results for the large errors are good showing improvements of between 9.1% and 23%. Overall twenty seven of the tests produced an increase in performance and only five showed a decrease. It is unmistakably clear that for a small error over the large step the fault tolerant controller actually inhibits the performance of the AUV. This problem was due to training method used, whereby a general improvement is accepted even if for some of the results the performance has decreased. This may well be correctable if the FIS could be trained for a great number of epochs.

The rise time results show that for a large LOE all of the rise times have decreased. For a small LOE the rise times have remained the same or as shown in six of the tests, actually increased. Nineteen of the thirty-six results displayed the rise time decreasing. With further training it should be possible to have all thirty-six rise times reduced.

One solution to this would be to limit the maximum angle the AUV turns in one motion, or only use the FTCS when there is a large fault occurring within the actuator. This would perhaps slow the process of any mission down, but not to the extent that continuing on the mission with the non-fault tolerant controller in place would.

REFERENCES

Craven, P. J. (1999). *Intelligent control strategies for an autonomous underwater vehicle.* PhD thesis, University of Plymouth, U.K.

Jang, J. S. R. (1991). ANFIS: Adaptive network-based fuzzy inference system. *IEEE transactions on systems, man and cybernetics,* **Vol. 23**, 665-685.

Kirkpatrick, S., C. Gelatt and M. Vecchi (1983). Optimization by simulated annealing. *Science*, **220**, 671-680.

Mort, N. and D. A. Derradji (1999). Control reconfiguration in submersible vehicles using artificial neural networks. *International Journal of Systems Science*, **Vol. 30, No. 9**, 989-1010.

MMAR 2001 28-31 August 2001, Międzyzdroje, Poland
7th IEEE International Conference on Methods and Models in Automation and Robotics

A FUZZY FAULT TOLERANT CONTROL SCHEME FOR AN AUTONOMOUS UNDERWATER VEHICLE

ROBERT SUTTON†, ANDREW PEARSON†, ANTONIO TIANO‡

† Department of Mechanical and Marine Engineering, University of Plymouth, Drake Circus, Plymouth, Devon, PL4 8AA., *r.sutton@plymouth.ac.uk*
‡ Department of Information and Systems, University of Pavia , Pavia, Italy

Abstract: This paper is concerned with using fuzzy logic in a fault tolerant control scheme for a given autonomous underwater vehicle (AUV). Actuator failures are simulated within the yaw channel to test the ability of both a normal adaptive neuro-fuzzy inference system (ANFIS) controller and a fault tolerant fuzzy control system tuned using a tabu algorithm. Results are presented to show how each of the control systems coped with faults occurring within one of the two control actuators of the AUV.

Keywords: autonomous underwater vehicle, fault tolerant, fuzzy logic, tabu search.

1. INTRODUCTION

Commercial, naval and scientific operational specifications for uninhabited underwater vehicles (UUVs) continue to become more challenging in line with the advances being made in the field of control engineering. In order to survive actuator failure during a mission, such vehicles need to possess either a reconfigurable or a fault tolerant control system.

In recent years, considerable interest has been shown into the commercial, scientific and naval use of UUVs. UUV being a generic expression to describe both an AUV and a remotely operated vehicle (ROV). An AUV is a marine craft which fulfils a mission or task without being constantly monitored and supervised by a human operator, whilst a ROV is a marine vessel that requires instruction from an operator via a tethered cable.

Demands are growing for the requirement of AUVs to be able to operate effectively at extreme depths and/or in confined areas such as under packed ice. Unfortunately communication to and from the craft to the mother station through the sea water medium are poor. Thus, an AUV has to be self-sufficient for the duration of a mission. Fault tolerant control systems could also be beneficially installed within ROVs. Their employment within ROVs would significantly lighten the workload of the human operators allowing them to concentrate on the mission, whilst permitting them to maintain overall supervisory control.

This paper begins with the dynamics of the vehicle and the set-up of both the control system and the faults to be considered. Then the approach used to make the AUV fault tolerant is described. The results are then presented to show how the controllers handle the faults before, finally, the concluding remarks are given.

2. AUV DYNAMICS

The standard controller referred to throughout this paper is that developed by Craven [1] specifically for the given AUV model used within this study. It is an intelligent fuzzy logic controller developed using the ANFIS approach [3]. The control system was not designed to cope with the types of fault which are considered within this paper, it is only designed to control the yaw angle for the given AUV when it is fully operational. It was vital to test this controller to create a benchmark set of results. These results will be used for comparisons with the new control system design.

2.1. The Control Authority

The complete control authority of the AUV model is shown in Fig 1. It should be noted at this time that both the standard yaw ANFIS controller and the new fuzzy fault tolerant controllers only have command of the upper and lower canards (forward control surfaces) during this work.

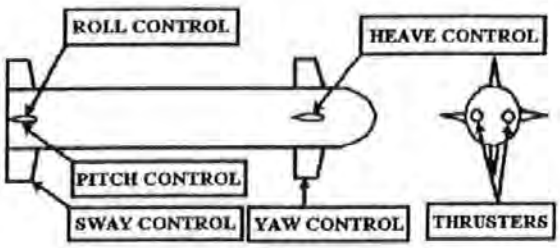


Fig. 1. Control Authority of the AUV

2.2. The Yaw Controller

The fuzzy logic rules for the standard ANFIS yaw controller used in this paper are given below in equation 1.

$$
\begin{aligned}
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.4863\,\psi_\varepsilon - 0.8791\,\dot{\psi} - 0.02926\\
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4890\,\psi_\varepsilon - 0.9021\,\dot{\psi} + 0.001381\\
&\text{If } \psi_\varepsilon \text{ is Neg and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.4858\,\psi_\varepsilon - 0.8962\,\dot{\psi} + 0.003143\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.2994\,\psi_\varepsilon - 0.7034\,\dot{\psi} - 0.1227\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4879\,\psi_\varepsilon - 0.8910\,\dot{\psi} + 0.003723 \qquad (1)\\
&\text{If } \psi_\varepsilon \text{ is Zero and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.3053\,\psi_\varepsilon - 0.3055\,\dot{\psi} - 0.03744\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Neg then}\\
&\delta = -0.5902\,\psi_\varepsilon - 0.8387\,\dot{\psi} - 0.1172\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Zero then}\\
&\delta = -0.4811\,\psi_\varepsilon - 1.081\,\dot{\psi} - 0.06111\\
&\text{If } \psi_\varepsilon \text{ is Pos and } \dot{\psi} \text{ is Pos then}\\
&\delta = -0.6596\,\psi_\varepsilon - 1.311\,\dot{\psi} + 0.7814
\end{aligned}
$$

Where ψ_ε is the yaw angle error and is the difference between the yaw angle demand and the actual yaw angle, $\dot{\psi}$ is the yaw rate, δ is the desired rudder angle, Neg and Pos, and Zero are fuzzy sets.

3. FAULT TOLERANT CONTROLLER

To attempt to cope with the failures it is proposed to situate a fuzzy logic system between the ANFIS controller and the AUV dynamics. This will multiply the control signal used by the actuators, as shown in Fig 2. This improvement should give the control system a level of fault tolerance to the failures that are being considered in this paper.

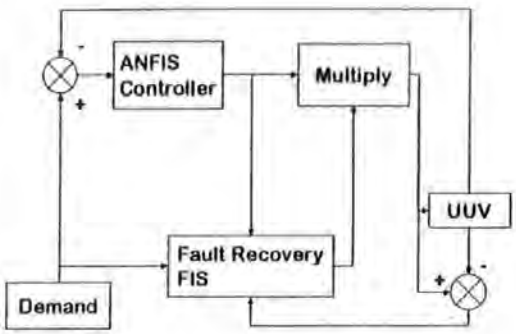


Fig. 2. Fault Tolerant Control System Set-up

The fuzzy logic system can be seen to take three batches of information for use in determining the degree of failure which has occurred and then makes the necessary correction for it. The three pieces of information the system will have as inputs are, the demand being placed on the standard ANFIS controller, the demand being placed on the actuators by the standard ANFIS controller and, the error in the damaged actuator. The fuzzy logic system compensates for the fault and returns the AUV to a fully functional or near fully functional situation. When there is no fault present, the fault tolerant system will have a minimal affect on the ANFIS control signal. However when a fault does occur it will compensate for it and allow the AUV to continue operating and complete its mission.

For this paper two fuzzy fault tolerant systems were tested. The first was a heuristic fuzzy inference system (FIS). This was designed by taking a simple approach to the problem, the bigger the error, the larger the value the control signal must be multiplied by in order to compensate. This was then tuned using the tabu method [2] to create the second FIS.

4. ACTUATOR FAULTS

To test the fault recovery ability of the fuzzy logic controllers it was necessary to create several failure scenarios. Faults occurring within only one of the two control actuators are considered in this paper. The actuator faults will change the way the AUV responds to control signals and will also affect the dynamic behaviour of the AUV. The fault used here is one whereby one of the two control surfaces

suffers a loss of efficiency (LOE) [4]. The upper canard was the actuator chosen for the faults to be simulated within.

When the AUV is undamaged the LOE for each of the actuators would be 0%. As an actuator becomes more damaged this value will rise up to a maximum LOE of 100%, which would be total failure where the actuator would not be able to move.

For this study, all of the work has been executed within the yaw channel. The same principles and techniques can be used in any of the other channels to produce similar results. The failures have been simulated in the upper canard which is used to control yaw. The system was tested by being subjected to three step inputs (10, 20, and 30 degrees). Within the AUV model the actuators are formed using three blocks, a rate limiter, a saturation block and a transfer function, as shown in Fig 3.

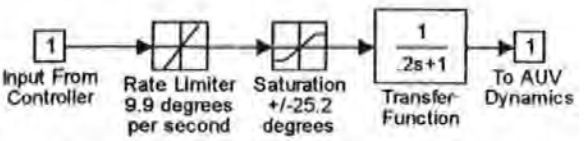


Fig. 3. Actuator Simulator Within Matlab AUV Model

The first type of failure is total LOE, which can also be thought of total failure of one control surface at zero degrees. This failure has been chosen to simulate the control surface becoming locked. This failure was accomplished by setting the input from the controller (see Fig 3) to zero.

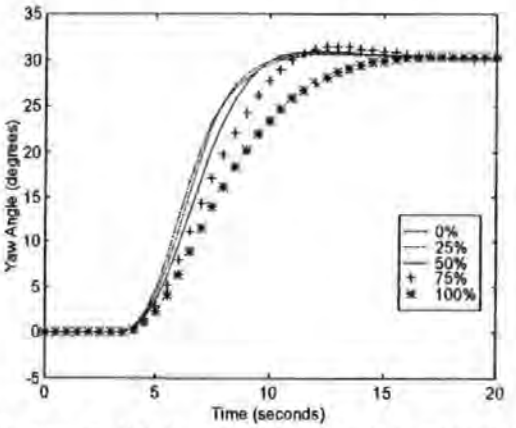
The second type of failure is to reduce the effectiveness of the controller. The percentages of LOE considered for this work were 75%, 50%, and 25% [4]. There are three types of failure which can be implemented using this approach for actuator failures. These failures can be achieved by altering the values within the saturation and the rate limiter blocks shown in Fig 3. For this paper the actuator failures were simulated firstly by altering only the value in the saturation block (shown in the tables as 'Saturation'), secondly by altering only the value in the rate limiter block (shown in tables as 'Rate limiter'), and thirdly by altering both values simultaneously to the same level of failure (shown in tables as 'Both').

5. RESULTS

The standard non-fault tolerant ANFIS controller and the fault tolerant controllers were all subjected to the complete range of faults described in the previous section using the Matlab SIMULINK computer package.

5.1. The Standard ANFIS Controller

The first set of tests were performed on the standard ANFIS control system. Below the performance of this controller can be seen as it attempts to achieve the 30 degree demanded yaw angle while enduring an increase in the level of failure (using the failure



defined as 'Both') for successive simulations (Fig 4).

Fig. 4. The Standard ANFIS Controller System Attempting to Handle Various Levels of the 'Both' Failure Scenario.

In order to make comparison of the controllers easier two forms of information have been recorded. Firstly the root mean squared error (RMSE) value, between the desired and the actual results was calculated. Where the desired results are defined as the recorded yaw angle of the AUV when LOE was 0%. The RMSEs for the standard control system, for all of the given faults for the given demanded yaw angles, have been compiled in Table 1 shown below.

Table 1. The RMSEs (degrees) for the Complete Set of Results for the Standard ANFIS Controller.

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0	0	0.099	1.065
	Rate limiter	0.075	0.188	0.380	1.065
	Both	0.075	0.188	0.380	1.065
20°	Saturation	0	0.127	0.741	2.360
	Rate limiter	0.205	0.546	1.080	2.360
	Both	0.205	0.546	1.132	2.360
30°	Saturation	0.094	0.691	1.904	4.125
	Rate limiter	0.434	1.088	2.172	4.125
	Both	0.436	1.172	2.390	4.125

The second type of information used to evaluate the systems was the rise times. Where the rise time is defined as the time taken for the AUV to rise from five to ninety five per cent of the demanded yaw angle. The rise times for the standard control system, when attempting to handling the all of the given faults for the given demanded yaw angles, have been compiled in Table 2 shown below.

Table 2. The Complete Set of Rise Times (seconds) of The Chosen Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Sat	4.5	4.5	4.7	7.1
	Rate limiter	4.6	4.7	4.8	7.1
	Both	4.6	4.7	4.8	7.1
20°	Sat	4.8	4.9	5.4	7.5
	Rate limiter	4.7	4.7	4.9	7.5
	Both	4.7	4.7	5	7.5
30°	Sat	4.9	5.3	6.1	7.9
	Rate limiter	4.3	4.5	5.3	7.9
	Both	4.4	4.7	5.7	7.9

5.2. The Heuristic Controllers

Several heuristic controllers were created as starting points for the tuning process. The first of them was to have a FIS whose starting point did not affect the output of the standard controller. As expected this gave results identical to those of the standard ANFIS controller shown above. The others all worked on the principle that as the size of fault increased then the signal must also be increased. Below is shown the results for one such FIS attempting to compensate for the set of failures defined above as 'Both' over the 30 degree step input demand (Fig 5).

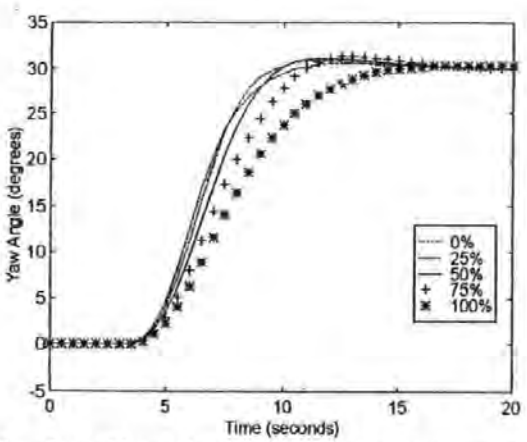


Fig. 5. A Heuristic Control System Attempting to Handle Various Levels of the 'Both' Failure Scenario.

The complete set of RMSEs for the same heuristic controller when subjected to all the given failure scenarios have been compiled in Table 3.

Table 3. The Complete Set of RMSEs (degrees) of The Chosen Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0.004	0.004	0.095	1.024
	Rate limiter	0.072	0.183	0.367	1.024
	Both	0.072	0.183	0.367	1.024
20°	Saturation	0.112	0.113	0.711	2.255
	Rate limiter	0.181	0.494	1.051	2.255
	Both	0.181	0.494	1.098	2.255
30°	Saturation	0.060	0.603	1.829	4.001
	Rate limiter	0.548	1.120	2.114	4.001
	Both	0.529	1.145	2.320	4.001

5.3. The Tuned Controllers

The heuristic FIS mentioned above was then tuned for 500 epochs using a set of training data, which comprised of various sizes of steps with different levels of failures occurring. The FIS was tuned using the tabu method [2]. The FISs which performed best on the training data were then re-tested on the failures used above. After the training and testing of the FISs had taken place it was possible to determine which of the FISs handled the faults optimally. This was done by taking into account the performance of each FIS with respect to all of the faults. The fuzzy logic rule base for the best system is shown below in equation 2.

If δ is Neg and ψ_ε is Neg and η is Small then β =0.8802

If δ is Neg and ψ_ε is Neg and η is Medium then β =1.3311

If δ is Neg and ψ_ε is Neg and η is Big then β = 1.1293

If δ is Neg and ψ_ε is Zero and η is Small then β = 1.1034

If δ is Neg and ψ_ε is Zero and η is Medium then β = 1.2944

If δ is Neg and ψ_ε is Zero and η is Big then β = 1.1031

If δ is Neg and ψ_ε is Pos and η is Small then β = 0.9496

If δ is Neg and ψ_ε is Pos and η is Medium then β = 0.9229

If δ is Neg and ψ_ε is Pos and η is Big then β = 1.3010

If δ is Zero and ψ_ε is Neg and η is Small then β = 0.9940

If δ is Zero and ψ_ε is Neg and η is Medium then β = 0.7219

If δ is Zero and ψ_ε is Neg and η is Big then β = 1.2037

If δ is Zero and ψ_ε is Zero and η is Small then β = 1.0284

If δ is Zero and ψ_ε is Zero and η is Medium then β = 1.0490

If δ is Zero and ψ_ε is Zero and η is Big then $\beta = 1.2851$ (2)

If δ is Zero and ψ_ε is Pos and η is Small then $\beta = 1.0503$

If δ is Zero and ψ_ε is Pos and η is Medium then $\beta = 1.1073$

If δ is Zero and ψ_ε is Pos and η is Big then $\beta = 1.0012$

If δ is Pos and ψ_ε is Neg and η is Small then $\beta = 6.7378$

If δ is Pos and ψ_ε is Neg and η is Medium then $\beta = 7.2062$

If δ is Pos and ψ_ε is Neg and η is Big then $\beta = 7.2363$

If δ is Pos and ψ_ε is Zero and η is Small then $\beta = 1.2028$

If δ is Pos and ψ_ε is Zero and η is Medium then $\beta = 0.9834$

If δ is Pos and ψ_ε is Zero and η is Big then $\beta = 1.0905$

If δ is Pos and ψ_ε is Pos and η is Small then $\beta = 0.8235$

If δ is Pos and ψ_ε is Pos and η is Medium then $\beta = 1.0272$

If δ is Pos and ψ_ε is Pos and η is Big then $\beta = 0.8852$

Where Pos, Neg, Zero, δ and ψ_ε are as defined above and η is the error in actuator and β is the control signal multiplier.

The results, expressed as RMSEs, for the chosen FIS are shown below in Table 4.

Table 4. The Complete Set of RMSEs (degrees) for The Best Tuned Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation	0.064	0.064	0.058	0.951
	Rate limiter	0.044	0.139	0.325	0.951
	Both	0.044	0.139	0.325	0.951
20°	Saturation	0.209	0.086	0.618	2.127
	Rate limiter	0.285	0.509	0.975	2.127
	Both	0.284	0.507	1.017	2.127
30°	Saturation	0.081	0.489	1.763	3.938
	Rate limiter	0.698	1.281	2.093	3.938
	Both	0.660	1.180	2.278	3.938

Below is shown the results for this best tuned FIS attempting to compensate for the set of failures defined above as 'Both' over the 30 degree step input demand (Fig 6).

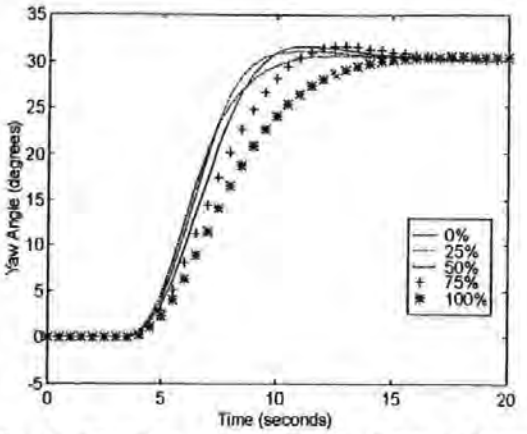


Fig. 6. The Best of the Tuned Control Systems Attempting to Handle Various Levels of the 'Both' Failure Scenario.

The rise times for the chosen FIS, when attempting to handle all of the given faults for the given demanded yaw angles, have been compiled in Table 5 shown below.

Table 5. The Complete Set of Rise Times (seconds) of The Best Tuned Heuristic Control System.

step size	type of failure	25%	50%	75%	100%
10°	Sat	4.4	4.4	4.5	6.8
	Rate limiter	4.5	4.6	4.6	6.8
	Both	4.5	4.6	4.6	6.8
20°	Sat	4.6	4.7	5.3	7.2
	Rate limiter	4.3	4.2	4.7	7.2
	Both	4.3	4.2	4.8	7.2
30°	Sat	4.9	5.1	6	7.7
	Rate limiter	4.1	4.3	5.2	7.7
	Both	4.2	4.5	5.6	7.7

The RMSEs displayed within Table 1 and Table 4 were then used to calculate the percentage increase in performances achieved by the tuned FIS for the faults which have been considered in this paper. These percentages are shown below in Table 6.

Table 6. The Percentage Increases For the Best Tuned Fuzzy Control System.

step size	type of failure	25%	50%	75%	100%
10°	Saturation			41.7%	10.7%
	Rate limiter	41.8%	26.2%	14.3%	10.7%
	Both	41.8%	26.2%	14.3%	10.7%
20°	Saturation		32.2%	16.6%	9.9%
	Rate limiter	-39%	6.7%	9.8%	9.9%
	Both	-39%	7.2%	10.1%	9.9%
30°	Saturation	14.6%	29.2%	7.4%	4.5%
	Rate limiter	-61%	-18%	3.7%	4.5%
	Both	-51.3%	-0.7%	4.7%	4.5%

6. ANALYSIS

Fig 4 and Tables 1 and 2 show how the standard ANFIS controller struggled to cope with the faults presented to it. It was also clear that as the level of faults increased then the performance of the controller degraded and that the rise times increased. This was certainly to be expected due to the design used for the standard ANFIS controller used for the AUV and the faults used.

It has then been shown via Fig 5 and Table 3 that even a basic heuristic FIS can enhance the fault tolerant performance of the given AUV.

Finally the performance of the best tabu search algorithm tuned FIS was displayed. The results of these tests showed how the tabu program had adapted the FIS in order improve the performance of the AUV. The highest percentage increase was 41.8%, which was in two tests, the 'Saturation' fault and the 'Both' fault of 25% over a 10 degree step input. With the worst result being a drop in performance of 61% in the 'Rate limiter' fault of 25% over a 30 degree step input. This may be due to the starting FIS used in the tuning. From Table 3 it is clear that a large RMSE is produced by this fault. The training does not improve the results, which may be due to the tuning process accepting a decrease in perform on this fault if overall it judged the controller to be an improvement.

The rise times shown in Tables 2 and 5 present a far better view of the tests, with all but one of the rise times being reduced, and the remaining one result having the same rise time. This shows that the FIS is forcing the AUV to turn slightly faster, to try and compensate for the faults.

7. CONCLUDING REMARKS

The analysed results show how the performance of the AUV has been enhanced by the fault tolerant FIS. The results for the large errors are good showing improvements of between 4.5% and 10.7%. Overall twenty seven of the tests produced an increase in performance and only six showed a decrease. It is unmistakably clear that for a small error over the medium and large steps that the fault tolerant controller actually inhibits the performance of the AUV. This problem was due to the training method used, whereby a general improvement is accepted even if for some of the results the performance has decreased. This may well be correctable if the FIS was to be trained for a great number of epochs.

The rise time results show that for all levels of LOE, all but one of the rise times have decreased. With further training it should be possible to have all thirty-six rise times reduced. This shows that the training is effective at producing a FIS which will speed up the process of reaching the desired yaw angle and hence reduce the rise time back to that of the LOE 0% results.

One solution to the poor results for small LOE faults would be to limit the maximum angle the AUV turns in one motion. This would perhaps slow the process of any mission down, but not to the same extent as with the non-fault tolerant controller. A second possible solution would be to only use the fault tolerant control system when there is a large fault occurring within the actuator.

8. REFERENCES

1. Craven P. J.: Intelligent control strategies for an autonomous underwater vehicle. PhD thesis, University of Plymouth, U.K. 1999
2. Denna M., Mauri G., Zanaboni A.: Learning fuzzy rules with Tabu search- an application to control. IEEE Transactions on Fuzzy Systems, vol. 7, No. 2, 1999, pp. 295-318
3. Jang J. S. R.: ANFIS: Adaptive network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, 1991, pp. 665-685
4. Mort N., Derradji D. A.: Control reconfiguration in submersible vehicles using artificial neural networks. International Journal of Systems Science, Vol. 30, No. 9, 1999, pp. 989-1010

Appendix B Open loop Results

Type of Model	Input Path Type	Outputs		Input Path Type	Outputs	
	Random (training data)	yaw rate RMSEs (degrees per second)	yaw RMSEs (degrees)	Small Step (5 degrees)	yaw rate RMSEs (degrees per second)	yaw RMSEs (degrees)
DERA		3.7968	94.6567		0.9733	29.8461
Linear		4.1625	280.7301		0.6888	19.5272
fuzzy		2.4877	40.4470		0.5128	14.7005
Elman		4.1328	683.8166		0.5373	12.9213
	Medium Step (15 degrees)			Large Step (25.2 degrees)		
DERA		1.7359	51.7064		1.1376	28.5099
Linear		1.0489	20.8646		1.4189	24.1161
fuzzy		0.4660	10.6065		1.1361	31.8414
Elman		1.6798	51.8552		3.9686	122.7443

Appendix C Elman ANN Yaw Channel Model

The weight matrices and bias vector for the Elman ANN used in Chapter 3 are shown below.

The weight matrix for recurrent layer is:

Columns 1 through 7

-0.0249	-0.0275	-0.0890	0.2150	0.2031	-0.0047	0.0662
-0.0211	0.0022	-0.0729	-0.2253	-0.1736	-0.0878	0.1446
0.0521	0.0575	0.0079	-0.1764	0.0089	-0.1753	0.2336
0.0471	0.0589	0.0746	-0.1292	0.1220	0.1937	0.2128
0.1284	0.1566	0.0831	0.2019	-0.1120	-0.1683	0.0108
0.0072	0.0328	0.1127	-0.1930	0.0215	0.0395	0.0600
0.0163	-0.0147	0.0514	0.1886	0.2272	-0.0833	0.0320
-0.0370	-0.0016	-0.1389	0.0622	-0.1682	0.1925	-0.0490
0.0081	-0.0046	-0.0163	0.1377	0.1875	-0.1468	0.0757
0.0056	0.0158	0.0042	0.0781	-0.1826	0.0980	-0.2076
0.0168	-0.2415	0.1830	-0.1852	-0.1557	0.2196	0.0598
0.0052	0.0164	0.0211	0.2683	0.0327	-0.1071	0.0596
0.0040	0.0200	-0.0055	0.0074	0.0538	0.0313	-0.1425
0.2208	0.1860	0.1226	-0.0730	-0.2015	0.1379	0.0841
0.0069	-0.0042	-0.0177	-0.2174	0.0414	-0.1172	0.0841
-0.0078	0.0117	-0.0408	0.1495	-0.0312	-0.1480	0.1419
0.0034	-0.0050	0.0427	-0.1832	-0.0308	0.1336	-0.0799
0.1695	0.1507	0.0798	-0.1690	-0.1433	0.1856	0.0964
0.3320	0.3424	0.2625	0.1907	-0.2525	-0.2189	-0.1671
0.0030	0.0005	0.0019	-0.0099	0.0716	0.1836	-0.1044
0.0389	-0.0379	-0.0013	-0.0574	0.0979	0.1913	0.0651
-0.1163	-0.1123	-0.0376	0.1948	0.0151	-0.1462	-0.0618
-0.0044	-0.0043	-0.0112	-0.2107	-0.0769	-0.0175	0.0690
0.0332	0.0157	0.1384	0.0078	0.1217	0.0224	-0.1956
-0.0047	0.0135	-0.0381	-0.0817	-0.0511	-0.1356	-0.1996
0.1333	0.1055	0.3932	0.0996	0.0419	0.0169	0.0495
0.0103	0.0013	0.0568	0.1197	-0.2355	0.1910	0.0902
0.0016	-0.0214	0.0294	-0.0345	0.0863	0.1153	-0.0481
0.2202	0.2420	0.2012	-0.1329	-0.0314	-0.0101	-0.1398
-0.1272	-0.1187	-0.0432	0.1874	0.1824	-0.1028	-0.0506
-0.0033	0.0106	-0.0055	0.0452	0.1564	-0.0967	-0.1288
0.0188	0.0304	0.1319	-0.2055	0.2132	0.0217	0.0829

Columns 8 through 14

0.1152	0.1486	-0.1121	0.0064	0.1775	-0.1705	0.1632
0.1742	-0.1973	0.0914	-0.1693	0.1654	0.0195	-0.1313
0.0755	0.0413	-0.1290	-0.1067	0.1820	-0.0008	0.1788
-0.0253	-0.1914	0.0361	0.2029	0.0891	0.1839	0.1752
0.1185	0.0053	-0.0643	-0.2001	0.0594	-0.1756	-0.1143
-0.1091	0.1711	-0.2231	-0.0204	0.0467	0.0453	-0.1515
-0.0155	0.1106	0.0829	0.1704	-0.0865	0.1561	-0.1491

0.0704	0.0992	0.1007	-0.1251	-0.2326	-0.2294	-0.1248
0.0220	-0.0212	-0.0002	0.0353	0.0849	-0.0809	-0.0539
-0.2013	-0.2313	0.1753	-0.0430	-0.0067	0.2274	0.1427
0.1973	0.1235	0.0988	0.1523	-0.0448	-0.0557	-0.2190
0.0047	0.0664	0.1648	0.1626	0.2659	0.1930	0.0834
-0.0882	-0.0710	0.0755	-0.0364	0.2039	-0.0009	0.0022
0.1649	0.1982	0.0103	0.0892	-0.0996	-0.0289	0.1673
0.0357	-0.0206	-0.2184	0.1262	0.1549	0.3150	0.2245
-0.0642	-0.0538	0.1156	0.0217	-0.0765	-0.0495	0.0816
-0.1202	-0.1646	-0.2264	0.0176	0.1144	-0.0245	-0.0560
-0.0185	-0.0207	-0.1911	-0.0619	0.1996	-0.1148	0.0358
0.2853	0.2377	-0.0999	-0.1276	0.0917	-0.1010	-0.1878
0.0469	-0.0584	-0.1812	0.2279	0.0960	0.2143	0.1987
0.0943	0.1492	-0.0177	0.0031	0.3174	-0.0514	0.1128
0.1430	-0.1982	-0.1230	-0.0960	0.1255	-0.1260	-0.0878
-0.1942	-0.0796	-0.0796	-0.0058	-0.1898	0.0805	0.1265
-0.2670	-0.0974	-0.2553	0.1042	0.0877	0.1422	-0.1626
0.0941	-0.2116	0.2070	0.1336	0.1845	0.0468	-0.0651
-0.1735	-0.1895	-0.2506	-0.1412	0.1618	0.2430	0.0748
-0.0465	0.0094	-0.0027	-0.1295	-0.0135	0.1960	-0.0690
0.1342	0.1469	-0.1383	0.0347	-0.1255	0.0434	-0.0752
0.1722	-0.1108	0.0357	-0.1690	0.0779	-0.1202	-0.2425
-0.1658	0.1135	-0.0809	0.0060	0.2178	0.2405	-0.1507
-0.0682	0.1506	0.0351	0.1683	0.0147	-0.1830	-0.1984
0.0473	0.0228	-0.1013	-0.1715	-0.1441	-0.1125	-0.0049

Columns 15 through 21

-0.0416	0.0241	-0.0743	-0.2009	0.0498	-0.2000	0.0578
-0.1647	-0.0558	0.1618	0.0418	0.0471	0.2247	0.0029
-0.1870	-0.0711	-0.1401	0.1802	-0.0172	0.1489	-0.0646
-0.0295	-0.1346	0.1054	-0.1635	-0.0971	0.0278	0.1307
0.1453	0.2092	0.1488	0.0826	0.1872	-0.1380	0.0921
-0.0834	-0.0958	-0.1925	0.1172	-0.0530	0.1176	-0.1472
0.0807	-0.0432	0.0995	-0.1571	0.0171	-0.1360	0.1230
-0.2036	-0.1245	-0.2500	-0.0753	-0.0809	-0.0320	-0.1989
0.2396	0.1435	0.2076	0.1751	-0.1710	-0.1035	0.1765
0.0730	0.0505	0.1677	-0.0508	0.0309	-0.1747	-0.1163
-0.0661	-0.1884	0.0661	0.1293	0.1438	0.1256	0.2151
-0.1060	-0.1490	0.0394	0.1650	0.1592	0.0878	-0.1663
-0.0698	0.1011	-0.0552	-0.1327	0.1827	-0.1093	-0.2842
-0.1432	-0.0013	-0.1979	-0.1123	0.1191	-0.0557	0.1719
-0.1009	0.0042	0.1605	0.2669	0.1105	-0.2170	0.0585
-0.2142	0.1995	-0.0261	0.2397	0.0717	-0.1588	0.1955
-0.0234	-0.1084	-0.0956	-0.0538	0.0930	0.2437	-0.1562
0.0752	0.1536	0.0055	0.1753	0.0133	0.1773	0.0861
-0.2358	-0.0804	0.0537	-0.1336	-0.1623	0.0826	0.1260
-0.0808	0.1782	0.1827	0.1134	0.0062	0.1580	-0.1823
0.2085	-0.0967	0.0492	0.2100	-0.0837	0.0261	-0.1489
0.1224	-0.0739	0.1202	0.1460	0.2022	-0.0069	0.0979

-0.1073	0.1118	-0.0026	-0.0614	-0.0025	0.0471	-0.1683
0.1521	0.1993	-0.2424	0.1096	0.0168	0.0294	-0.0830
-0.1203	-0.1124	-0.0762	0.1238	-0.0773	0.0010	-0.2328
-0.0342	0.0462	-0.0547	-0.0720	0.1330	0.0955	-0.1435
0.0724	-0.0545	-0.0211	0.0883	0.1530	-0.2142	-0.0068
-0.0849	-0.0813	0.1839	-0.2598	-0.1807	-0.1074	0.2000
-0.2678	-0.2314	0.1844	-0.1946	-0.0786	-0.1966	0.0298
-0.1621	-0.1060	0.0746	0.0006	-0.0853	0.1705	0.1407
-0.1214	-0.0887	-0.1894	-0.0849	-0.2077	0.1761	-0.1651
-0.1692	0.1125	-0.0329	0.0260	-0.0164	0.2554	0.2887

Columns 22 through 28

0.1928	-0.0797	0.1425	-0.1594	0.1499	0.1336	-0.1335
-0.0577	-0.0399	0.0074	-0.0148	-0.0748	0.0679	0.0581
-0.0802	0.0989	-0.1589	-0.1803	0.1257	-0.1071	0.0490
0.1858	-0.0131	-0.0693	0.1151	-0.1703	-0.0640	-0.0282
0.1325	0.1182	0.0568	-0.1366	-0.0874	0.0026	0.1618
0.2293	0.0482	-0.1068	-0.2121	-0.1898	-0.0595	-0.1285
-0.2097	-0.1486	-0.0843	0.0376	-0.0905	-0.1641	-0.0968
-0.0142	-0.1931	-0.2017	-0.0657	0.1527	-0.0914	0.1799
0.2005	0.1906	-0.1463	-0.0046	0.1445	-0.1113	-0.1881
-0.1344	0.2065	0.1755	0.1335	-0.1357	0.1073	-0.0678
-0.1072	0.1059	0.0859	-0.1713	-0.2340	-0.2170	0.0729
-0.0419	0.2271	0.0857	0.1157	-0.2921	-0.1920	0.1582
-0.1320	-0.1642	0.0691	0.0866	0.0290	-0.1667	0.2737
0.1939	-0.0648	0.0849	0.0778	0.1936	0.1144	0.1385
-0.0018	-0.0747	0.2154	0.0159	-0.1926	-0.0557	-0.0319
-0.2113	-0.0611	-0.1479	0.0142	-0.1866	-0.0104	0.1035
0.0681	-0.1902	0.0990	-0.1370	-0.0558	-0.1938	0.1564
-0.1115	-0.1632	-0.1946	-0.0006	-0.0756	0.1651	-0.1673
-0.1137	-0.2284	-0.2094	-0.0025	0.0498	0.0393	0.2686
0.1620	0.2608	-0.0648	-0.0245	-0.2464	-0.2114	0.2310
-0.1264	0.3102	0.1415	0.0098	-0.3261	-0.0412	0.0127
0.1270	0.0626	0.0047	-0.0732	0.1944	0.1685	0.0606
-0.1248	-0.1244	-0.2122	-0.1946	0.2862	0.1419	-0.0982
-0.2198	0.1815	0.2078	-0.1196	0.0362	-0.0746	-0.1948
-0.0507	0.1764	-0.2193	-0.0500	0.2020	-0.1534	0.0149
0.0391	-0.1843	0.0067	0.0288	0.0607	-0.0149	0.1660
-0.1913	0.0142	0.1558	-0.1914	0.1343	-0.1205	-0.0417
0.2017	0.0860	0.1790	0.1260	-0.0896	0.0261	-0.1927
-0.1398	-0.0291	-0.0452	-0.2049	0.0491	0.1010	0.0953
-0.0455	0.0721	0.1962	-0.1227	-0.0478	0.0007	-0.0378
-0.0345	0.0366	-0.0969	-0.0843	0.1673	0.1869	-0.2168
-0.0200	-0.1628	0.2756	0.0676	0.0113	0.0759	-0.0757

Columns 29 through 35

0.0605	-0.1687	0.1190	0.0685	-0.1627	0.0122	-0.0677
-0.0678	-0.1135	-0.0146	0.2352	0.0273	-0.2360	0.2507
-0.0798	-0.0785	0.2473	0.0541	-0.1110	0.1765	0.0738
-0.0551	-0.0921	-0.1253	0.1234	-0.1806	-0.2161	-0.1959
0.0404	-0.0260	0.0766	-0.1514	0.1059	-0.1369	0.1745
0.1529	0.1583	0.1668	0.1409	0.1472	0.0221	0.0683
0.0239	0.0545	0.0412	-0.1570	-0.1972	0.0341	-0.2704
0.0542	-0.2645	-0.2159	-0.2541	-0.0487	0.1389	0.0544
0.0733	0.0598	-0.0657	-0.0734	-0.0151	-0.2757	-0.1033
-0.0695	-0.2203	0.0916	0.0577	-0.1334	-0.2488	-0.1531
-0.0352	0.0675	-0.1052	0.0917	-0.2338	-0.1523	-0.1493
-0.0051	-0.1551	0.0445	-0.1275	0.0663	-0.1208	0.1667
0.1896	-0.0975	-0.1910	-0.1911	-0.2349	-0.1571	-0.0086
-0.1654	0.1953	0.1675	-0.0279	-0.0795	-0.0382	0.1597
0.2200	-0.0280	-0.3103	0.1987	-0.1148	-0.1205	-0.0733
-0.0510	-0.1850	-0.2177	0.1019	0.1577	0.1993	-0.0006
0.1635	0.1962	-0.0784	0.0507	-0.1488	0.1286	0.1751
0.1683	0.0401	0.1184	-0.1186	0.1154	-0.1921	0.1711
0.0009	-0.0363	0.1400	-0.0155	0.0650	0.0213	-0.1033
-0.0839	0.1019	0.0787	0.0840	-0.0599	0.0672	-0.0417
0.1271	-0.2176	-0.2148	-0.2016	-0.1119	0.0874	0.1567
-0.1932	0.1516	0.0106	-0.0646	0.0608	-0.2126	-0.1962
0.1519	-0.0604	0.1341	0.0608	-0.0781	0.2906	0.1776
-0.3036	0.0092	-0.1975	-0.2564	-0.1131	-0.1607	-0.0952
0.1673	-0.1669	-0.0791	-0.2412	-0.0057	-0.0864	0.1043
-0.0535	0.1070	0.0573	0.1811	-0.0435	-0.2051	0.1332
-0.1453	0.2017	0.2578	0.1903	-0.0170	-0.0342	0.1453
0.1337	0.2111	-0.0643	0.1255	0.1399	0.1187	-0.1007
0.1994	-0.2316	0.1366	0.0263	0.1478	-0.0739	0.0694
-0.0328	0.1472	0.0915	-0.1344	0.2043	0.0840	-0.1585
-0.1472	0.1631	0.1236	0.0663	0.2060	0.1494	0.1701
-0.0430	-0.1768	-0.1726	0.0422	-0.0421	-0.1127	0.2350

The bias (column) vector for recurrent layer is:

-0.6493
-0.4405
0.0936
-0.2617
-0.2907
0.0386
0.2674
0.0212
-0.2241
0.4002
-0.4078
-0.5000
-0.4250
0.1793
0.6843
0.5057
-0.5170
0.5878
0.6792
0.3297
-0.4496
0.3577
0.1658
-0.0017
-0.6466
0.7286
0.3128
-0.6597
-0.3003
-0.0021
0.3660
0.6124

The weight matrix for output layer is:

Columns 1 through 7

0.1516 0.4586 -0.2028 -0.3301 -0.7525 -0.2961 -0.9439

Columns 8 through 14

0.9270 0.4822 0.7508 0.7957 0.8822 0.8143 -0.0569

Columns 15 through 21

1.0863 0.4672 -0.2682 -0.1295 0.4110 0.5524 1.0385

Columns 22 through 28

0.0912 -0.7070 -1.0058 0.4212 -0.6026 -0.1907 -0.7691

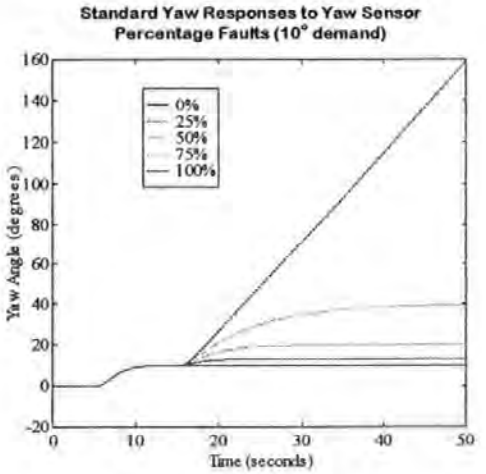
Columns 29 through 32

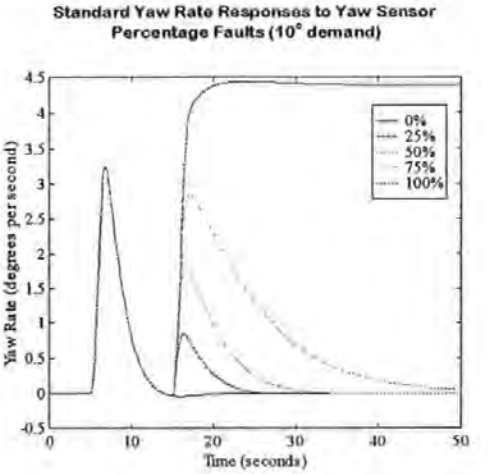
0.2242 0.0997 -0.5035 -0.4377

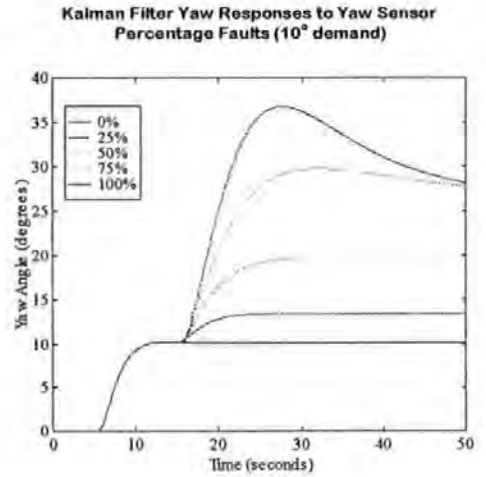
The bias (column) vector for output layer is:

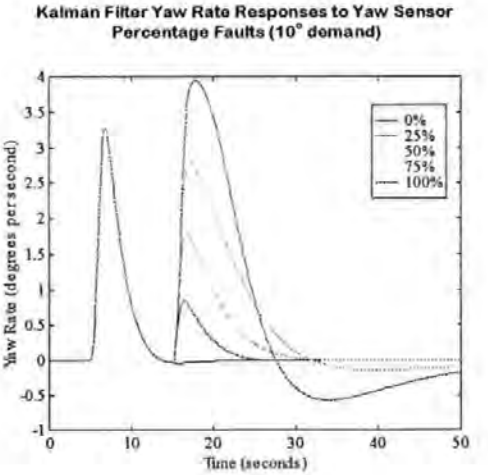
0.4930

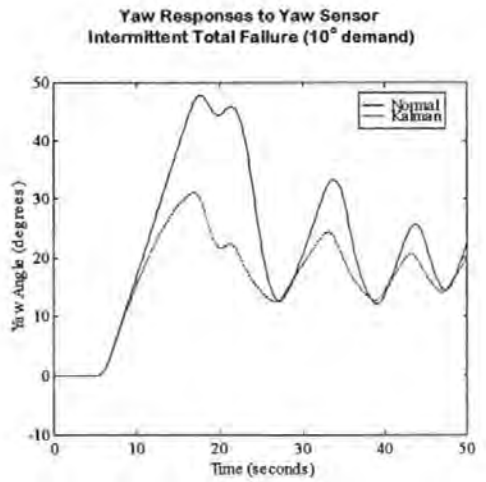
Appendix D Yaw Channel Sensor Failure Results

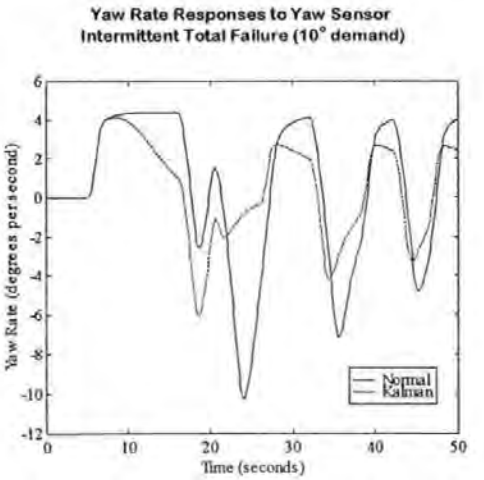




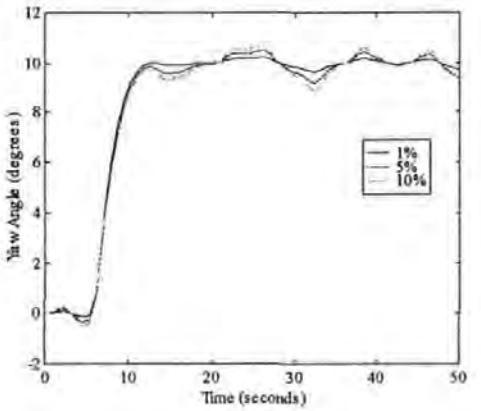




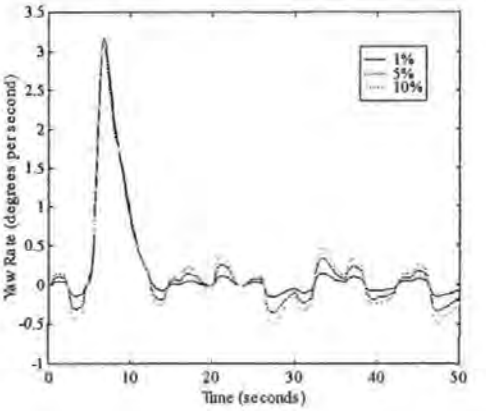




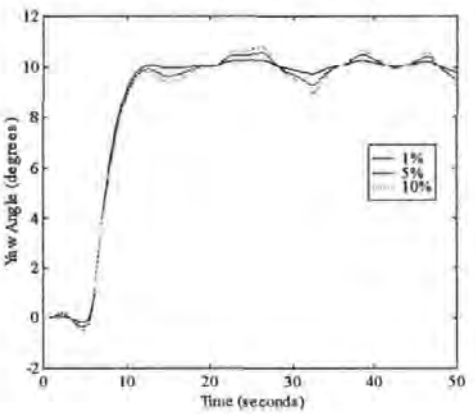
Standard Yaw Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



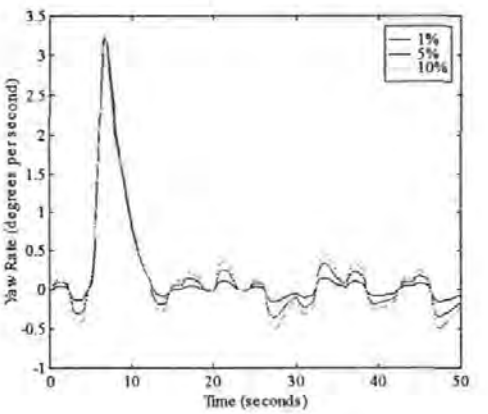
Standard Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



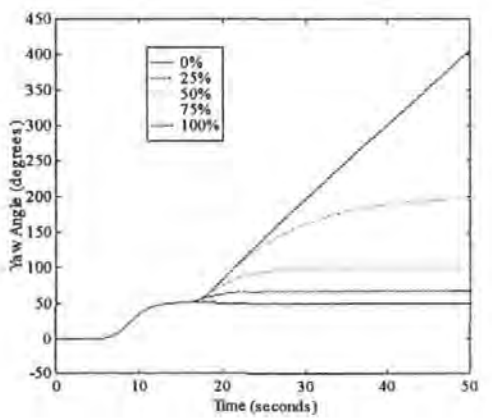
Kalman Filter Yaw Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



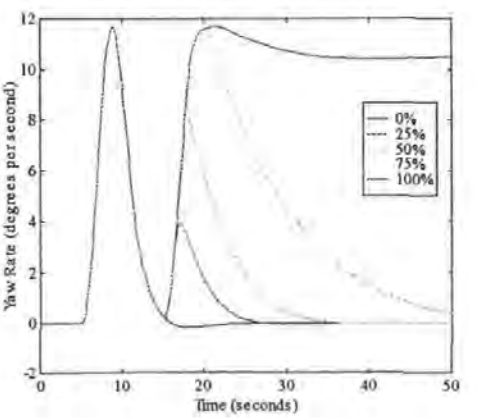
Kalman Filter Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



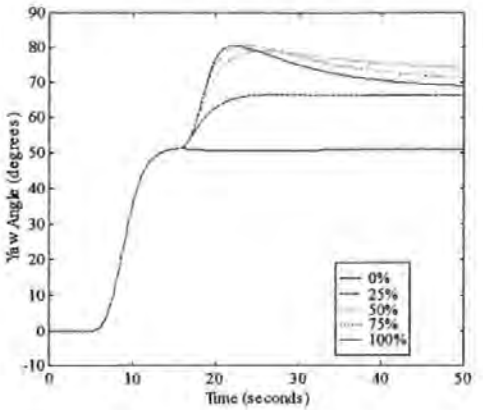
Standard Yaw Responses to Yaw Sensor Percentage Faults (50° demand)



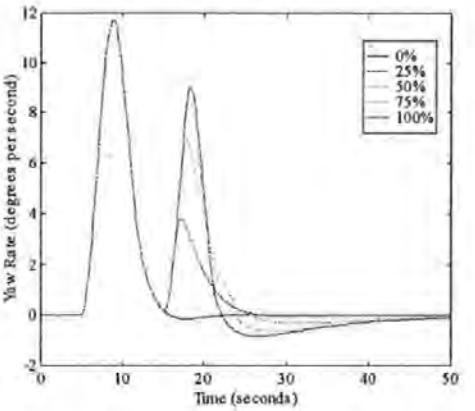
Standard Yaw Rate Responses to Yaw Sensor Percentage Faults (50° demand)



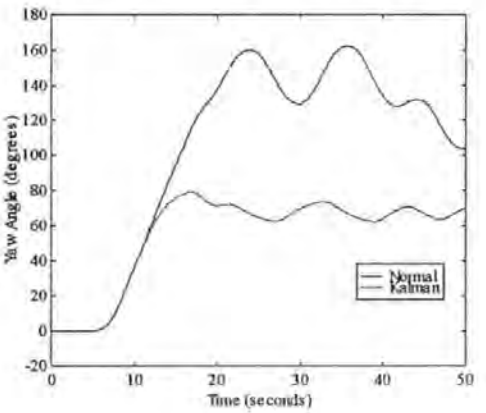
Kalman Filter Yaw Responses to Yaw Sensor Percentage Faults (50° demand)



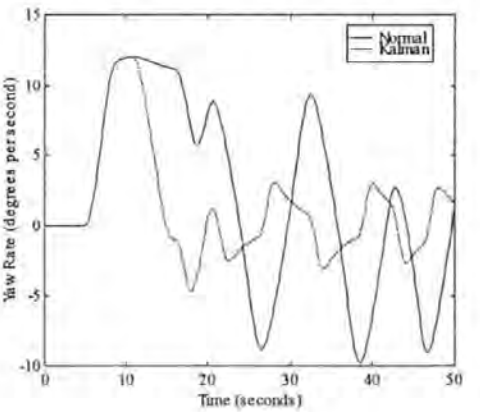
Kalman Filter Yaw Rate Responses to Yaw Sensor Percentage Faults (50° demand)



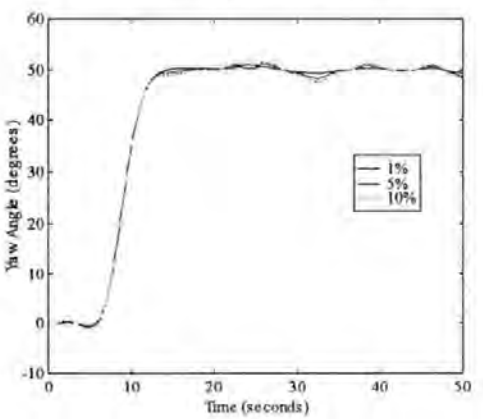
Yaw Responses to Yaw Sensor Intermittent Total Failure (50° demand)



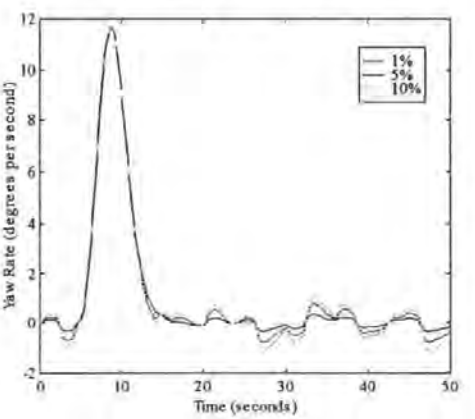
Yaw Rate Responses to Yaw Sensor Intermittent Total Failure (50° demand)



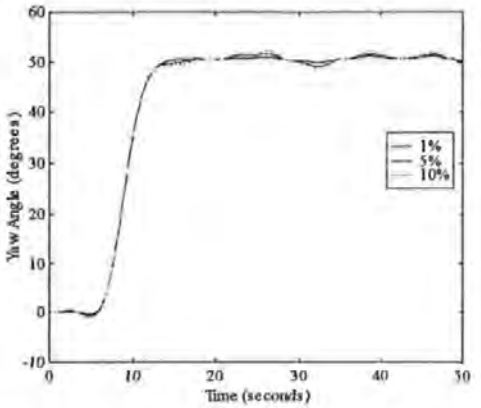
Standard Yaw Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



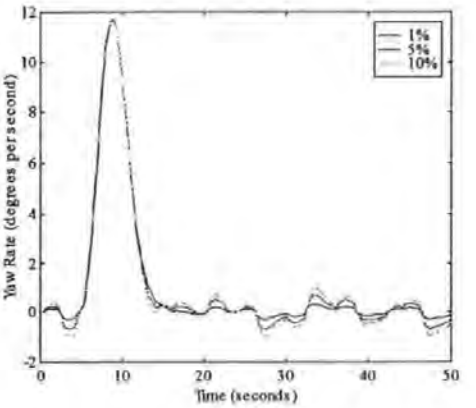
Standard Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



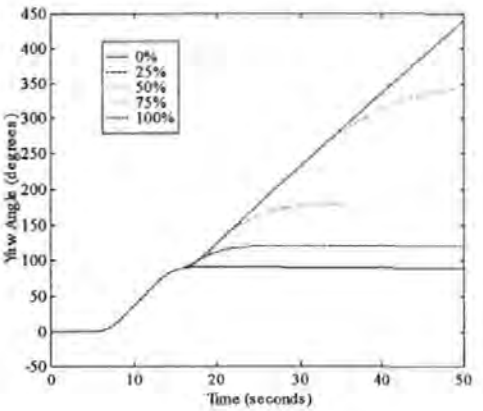
Kalman Filter Yaw Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



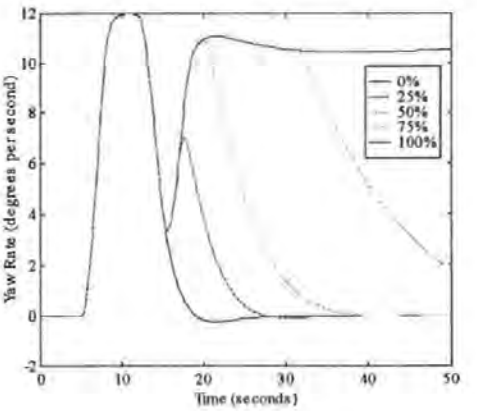
Kalman Filter Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



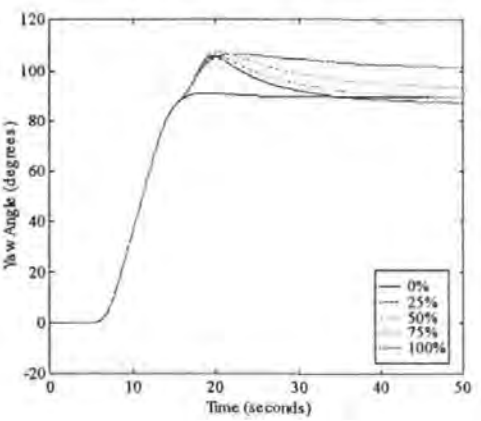
Standard Yaw Responses to Yaw Sensor Percentage Faults (90° demand)



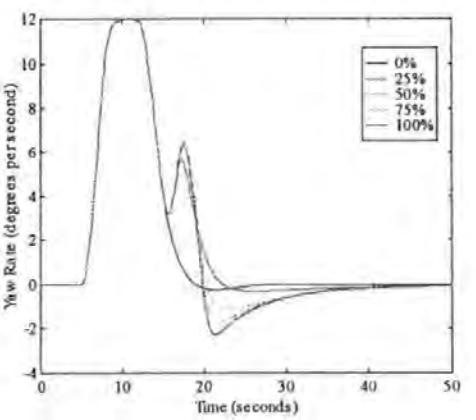
Standard Yaw Rate Responses to Yaw Sensor Percentage Faults (90° demand)



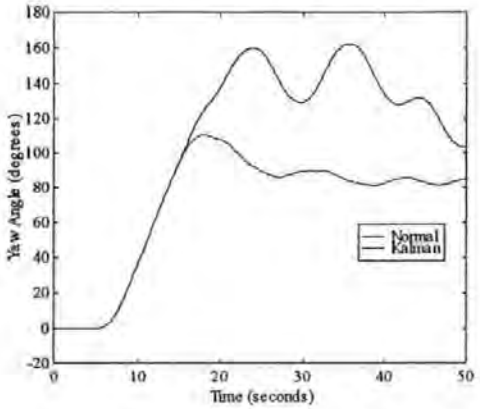
Kalman Filter Yaw Responses to Yaw Sensor Percentage Faults (90° demand)



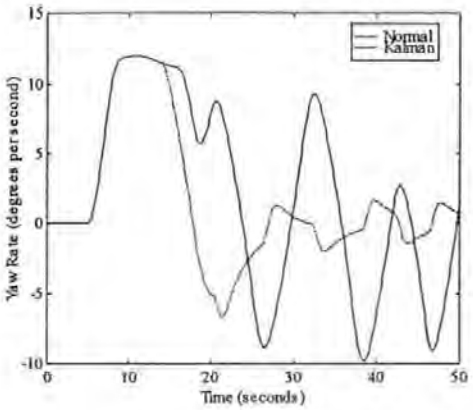
Kalman Filter Yaw Rate Responses to Yaw Sensor Percentage Faults (90° demand)



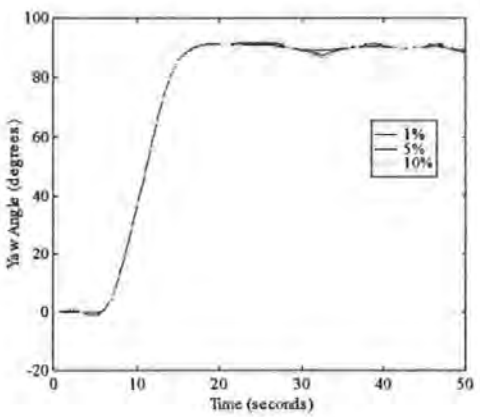
Yaw Responses to Yaw Sensor Intermittent Total Failure (90° demand)



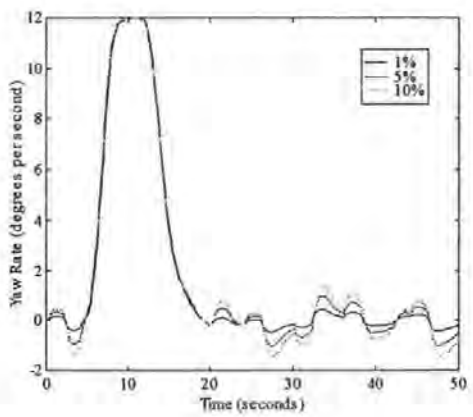
Yaw Rate Responses to Yaw Sensor Intermittent Total Failure (90° demand)



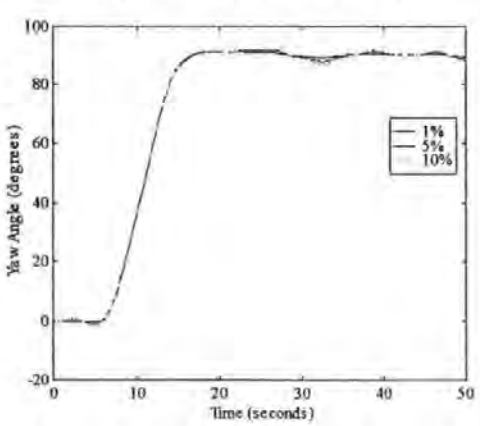
Standard Yaw Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



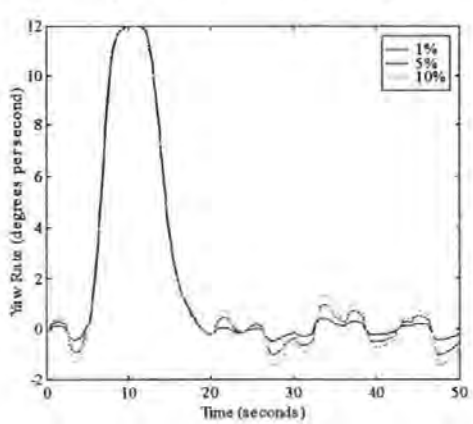
Standard Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



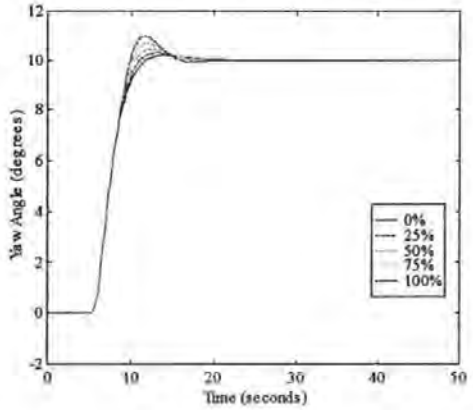
Kalman Filter Yaw Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



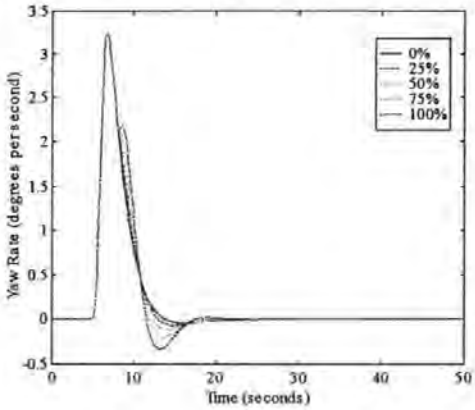
Kalman Filter Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



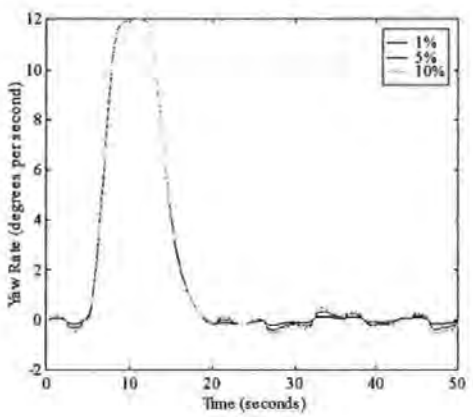
Standard Yaw Responses to Yaw Rate Sensor Percentage Faults (10° demand)



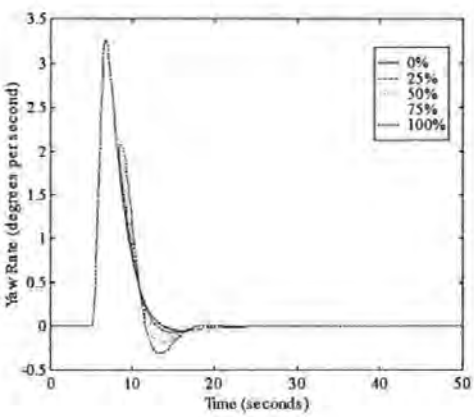
Standard Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (10° demand)



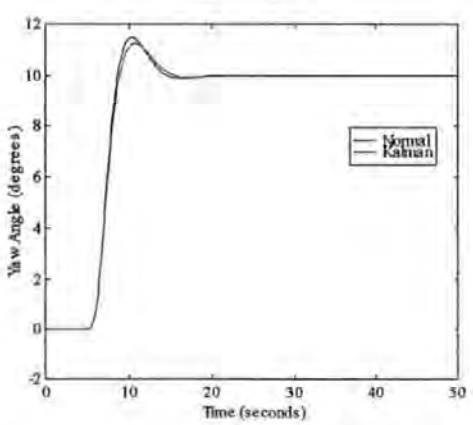
Kalman Filter Yaw Responses to Yaw Rate Sensor Percentage Faults (10° demand)



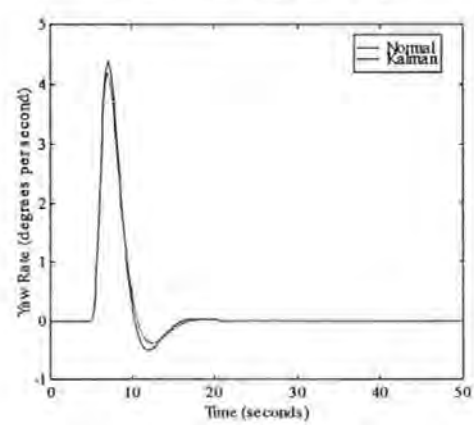
Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (10° demand)



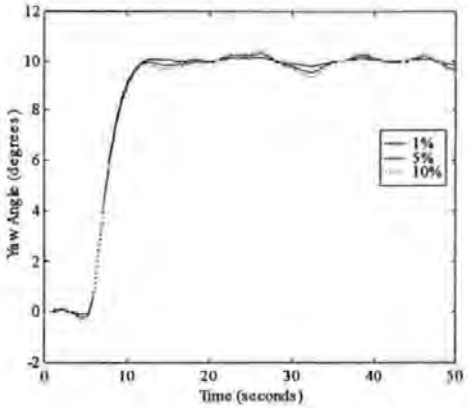
Yaw Responses to Yaw Rate Sensor Intermittent Failure (10° demand)



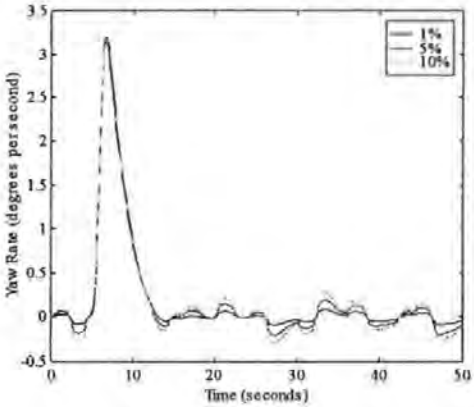
Yaw Rate Responses to Yaw Rate Sensor Intermittent Failure (10° demand)



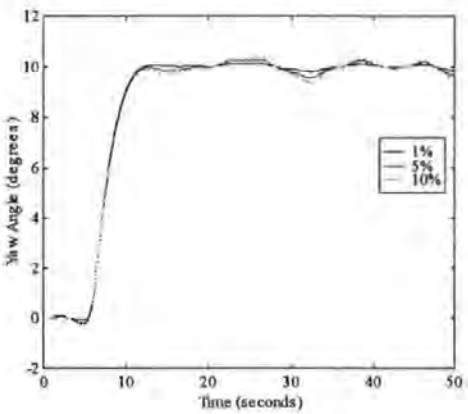
Standard Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



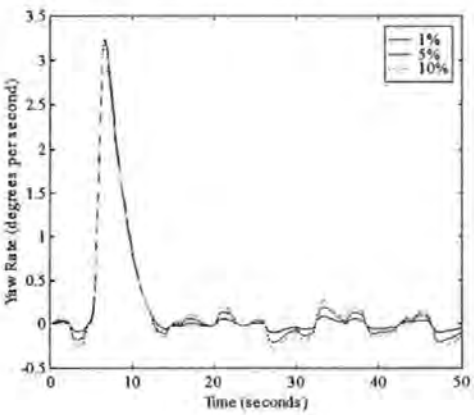
Standard Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



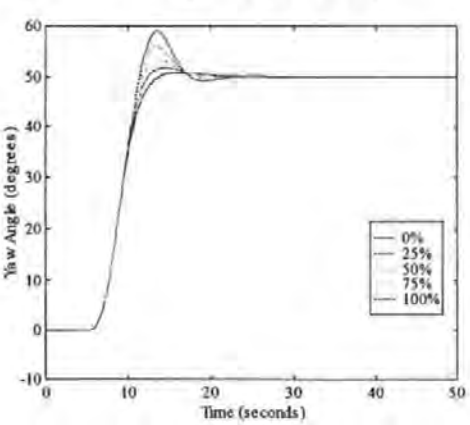
Kalman Filter Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



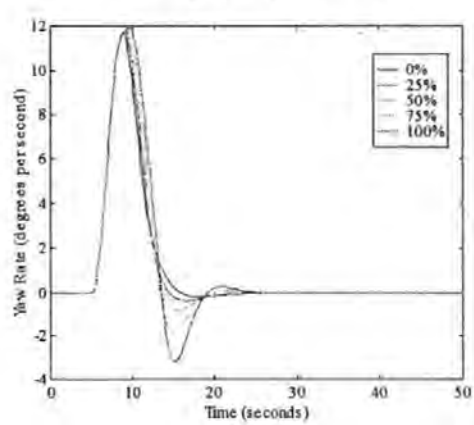
Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



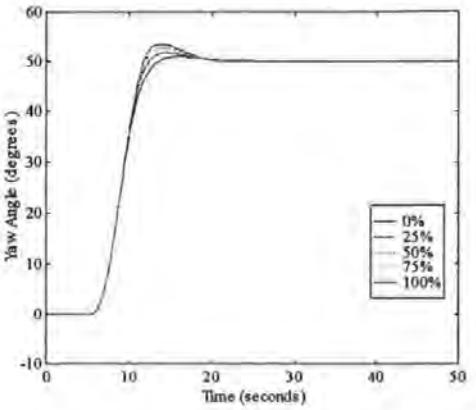
Standard Yaw Responses to Yaw Rate Sensor Percentage Faults (50° demand)



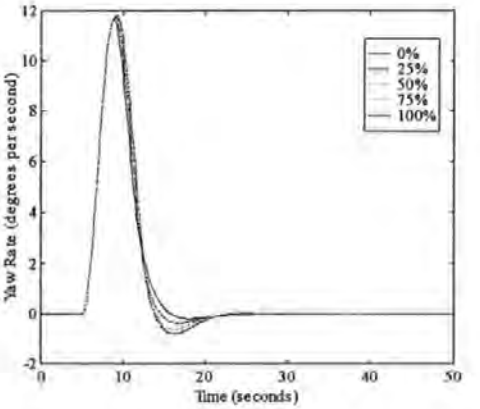
Standard Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (50° demand)



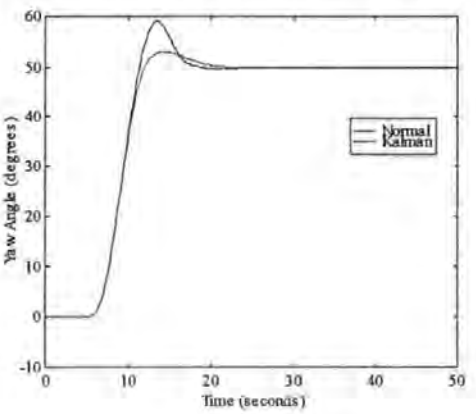
Kalman Filter Yaw Responses to Yaw Rate Sensor Percentage Faults (50° demand)



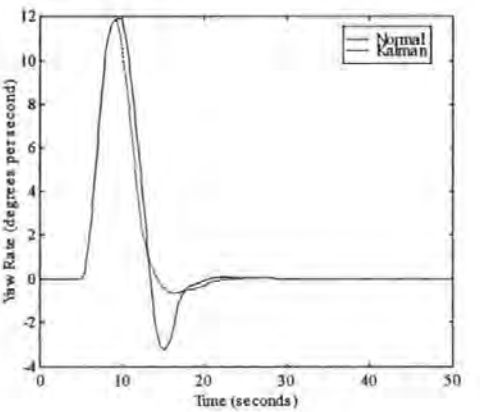
Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (50° demand)



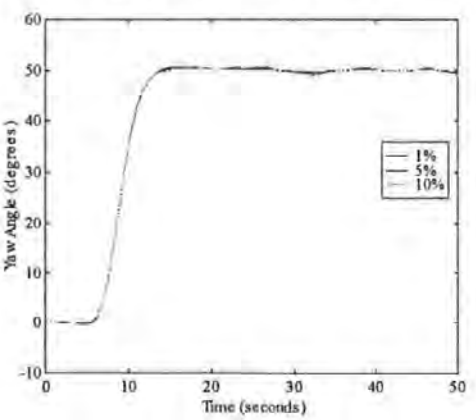
Yaw Responses to Yaw Rate Sensor Intermittent Total Failure (50° demand)



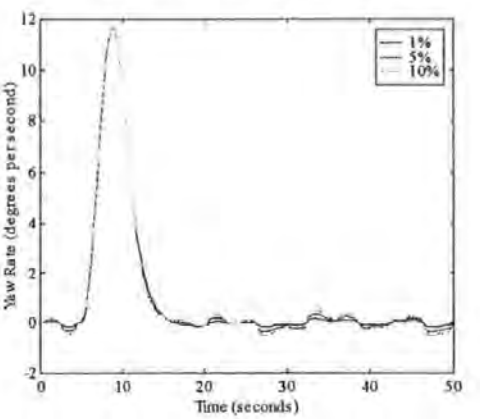
Yaw Rate Responses to Yaw Rate Sensor Intermittent Total Failure (50° demand)



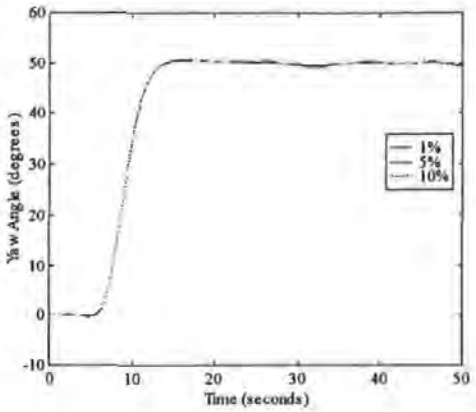
Standard Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



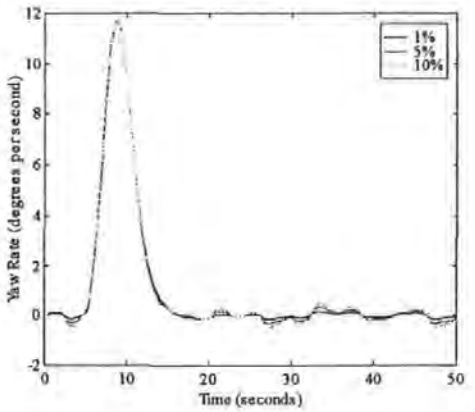
Standard Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



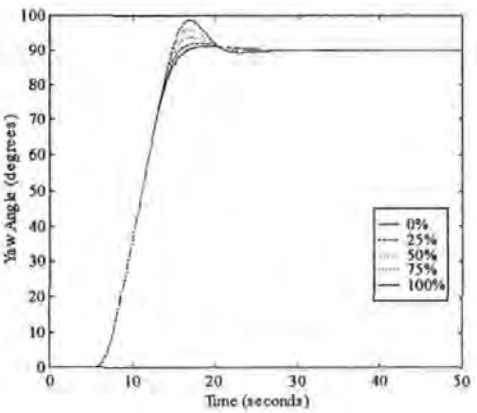
Kalman Filter Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



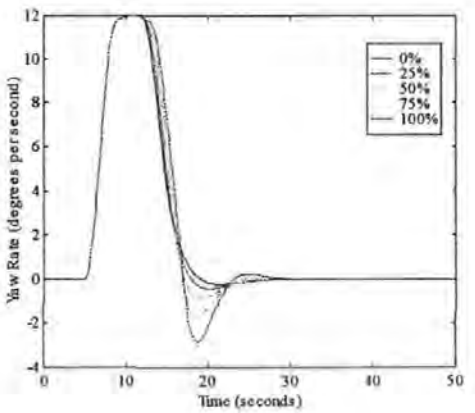
Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



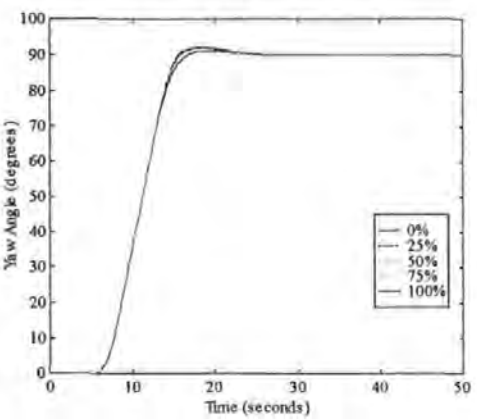
Standard Yaw Responses to Yaw Rate Sensor Percentage Faults (90° demand)



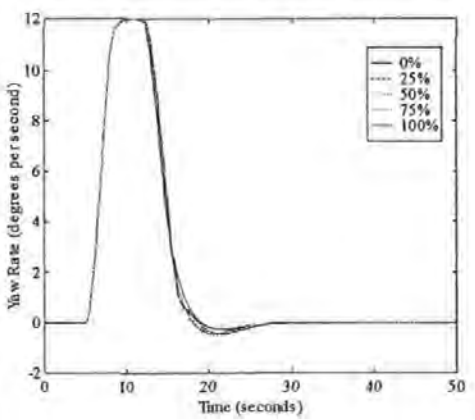
***Standard Yaw Rate Responses* to Yaw Rate Sensor Percentage Faults (90° demand)**



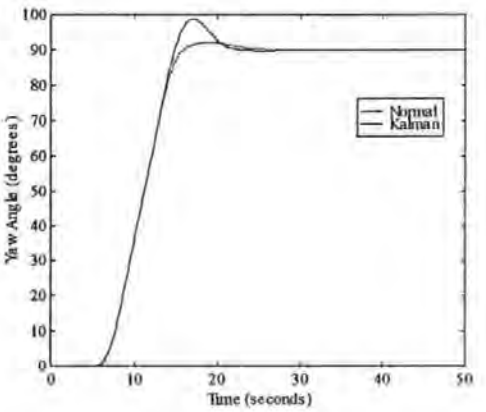
Kalman Filter Yaw Responses to Yaw Rate Sensor Percentage Faults (90° demand)



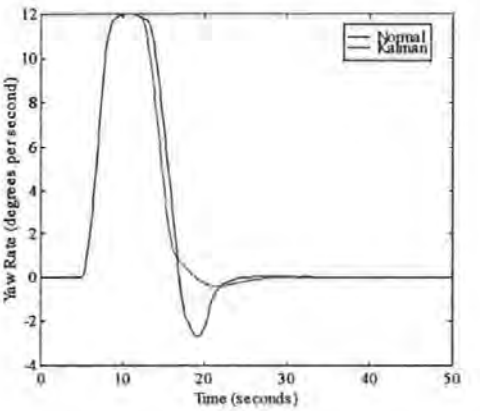
Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (90° demand)



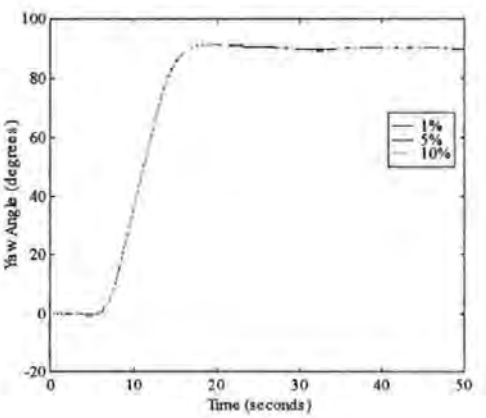
Yaw Responses to Yaw Rate Sensor Intermittent Total Failure (90° demand)



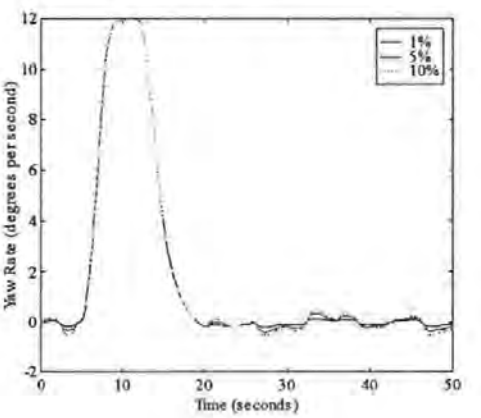
Yaw Rate Responses to Yaw Rate Sensor Intermittent Total Failure (90° demand)



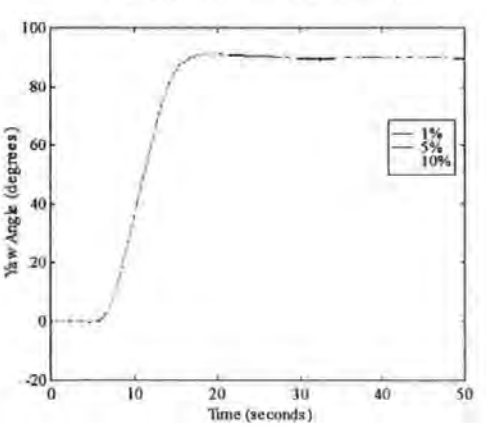
Standard Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)



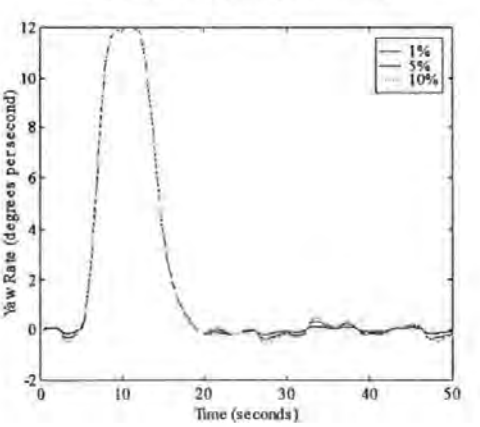
Standard Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)



Kalman Filter Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)

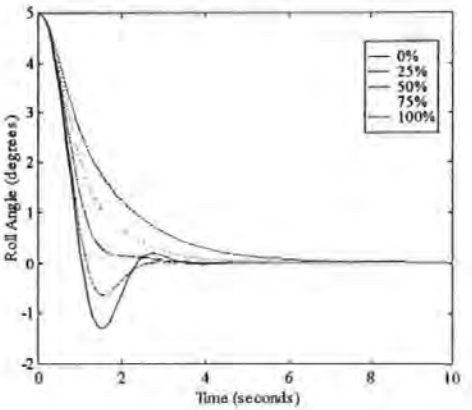


Kalman Filter Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)

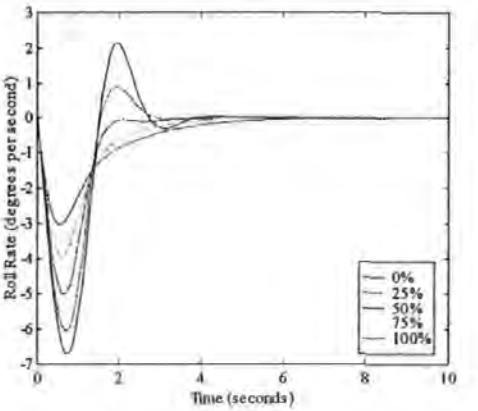


Appendix E Roll Channel Sensor Failure Results

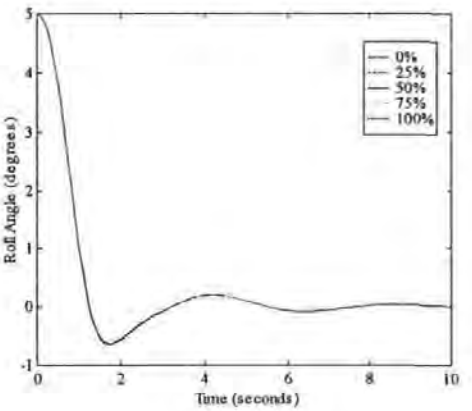
***Standard* Roll Responses to Roll Sensor Percentage Faults (5° initial position)**



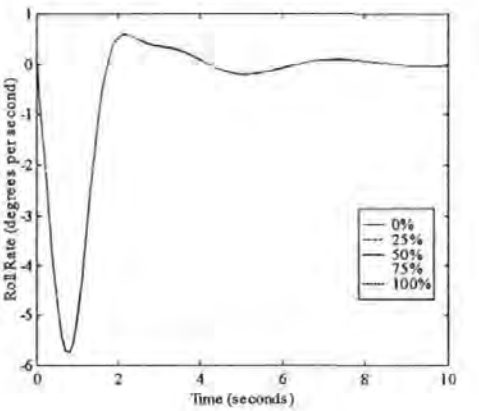
Standard Roll Rate Responses to Roll Sensor Percentage Faults (5° initial position)



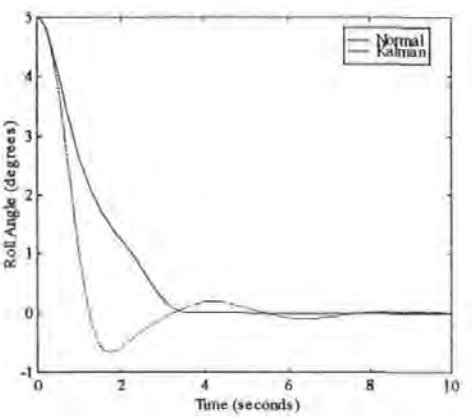
Kalman Filter Roll Responses to Roll Sensor Percentage Faults (5° initial position)



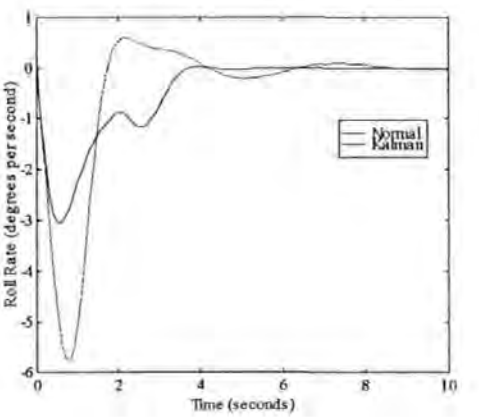
Kalman Filter *Roll Rate Responses* to Roll Sensor Percentage Faults (5° initial position)



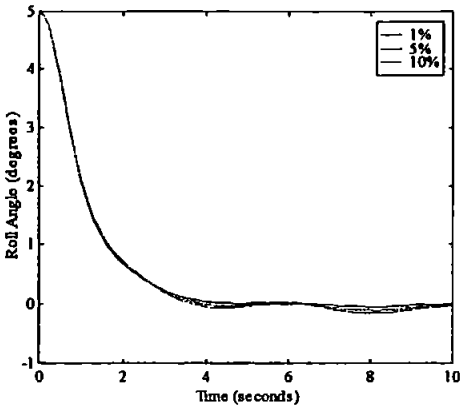
Roll Responses to Roll Sensor Intermittent Total Failure (5° initial position)



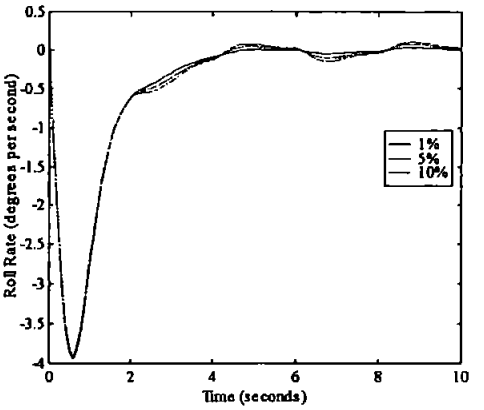
Roll Rate Responses to Roll Sensor Intermittent Total Failure (5° initial position)



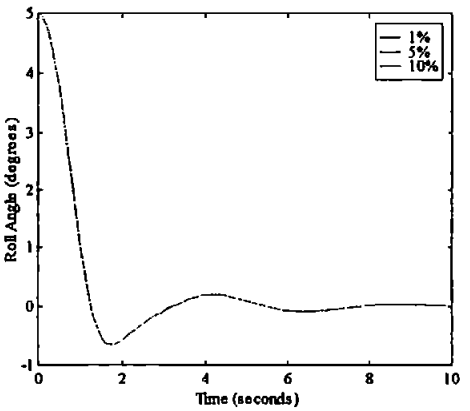
Standard Roll Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



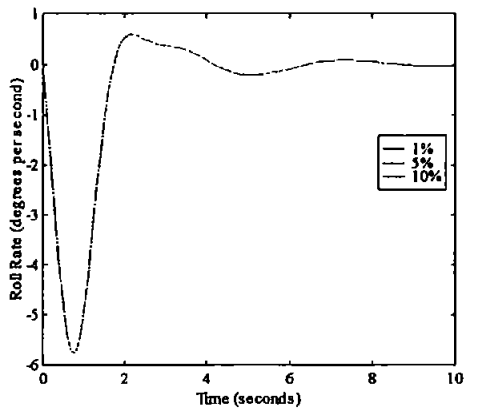
Standard Roll Rate Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



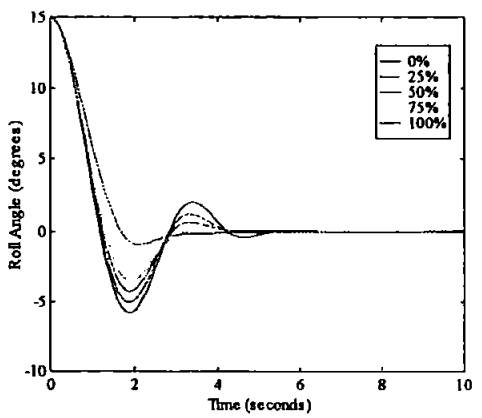
Kalman Filter Roll Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



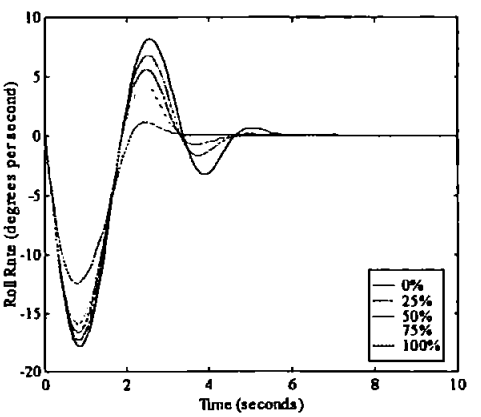
Kalman Filter Roll Rate Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



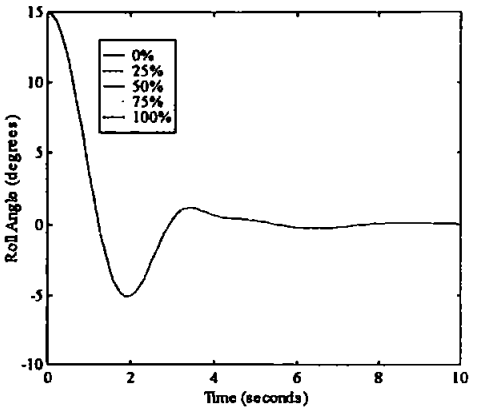
Standard Roll Responses to Roll Sensor Percentage Faults (15° initial position)



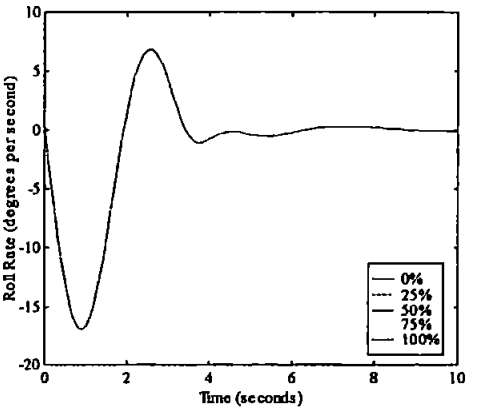
Standard Roll Rate Responses to Roll Sensor Percentage Faults (15° initial position)



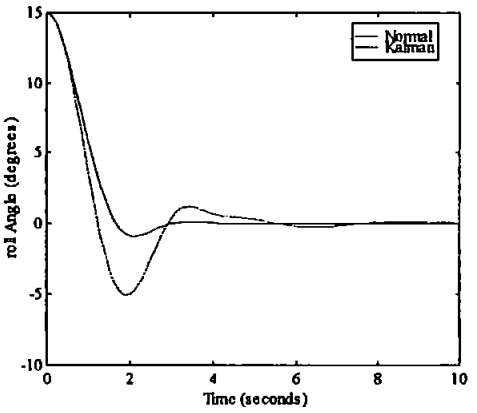
Kalman Filter Roll Responses to Roll Sensor Percentage Faults (15° initial position)



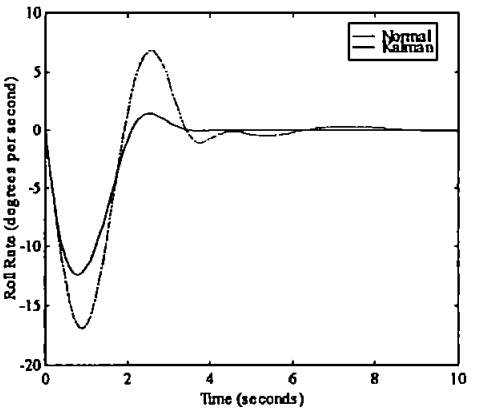
Kalman Filter Roll Rate Responses to Roll Sensor Percentage Faults (15° initial position)



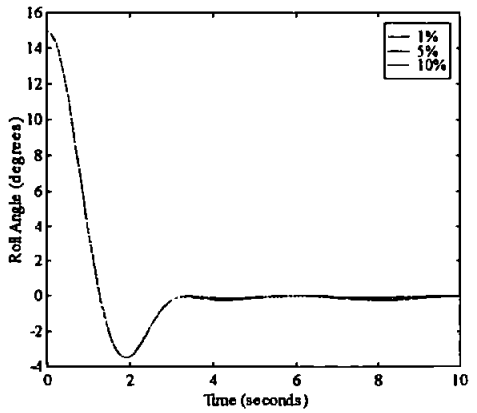
Roll Responses to Roll Sensor Intermittent Total Failure (15° initial position)



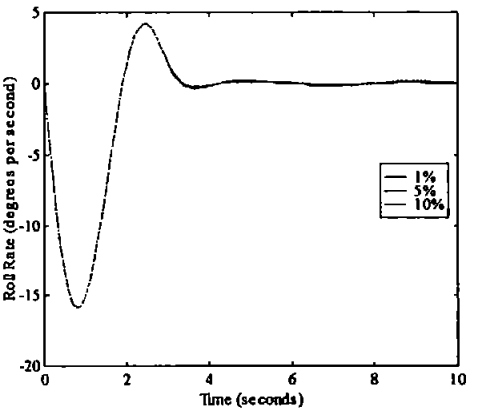
Roll Rate Responses to Roll Sensor Intermittent Total Failure (15° initial position)



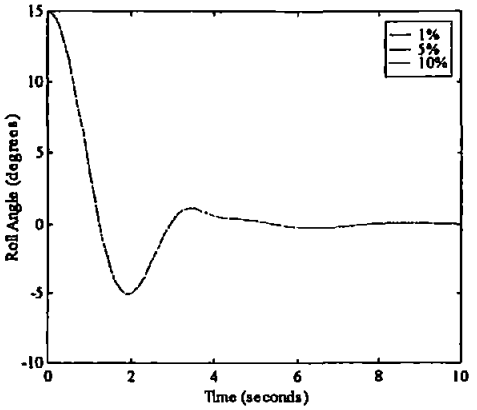
Standard Roll Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



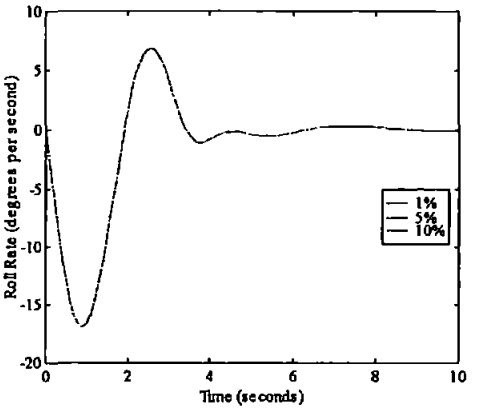
Standard Roll Rate Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



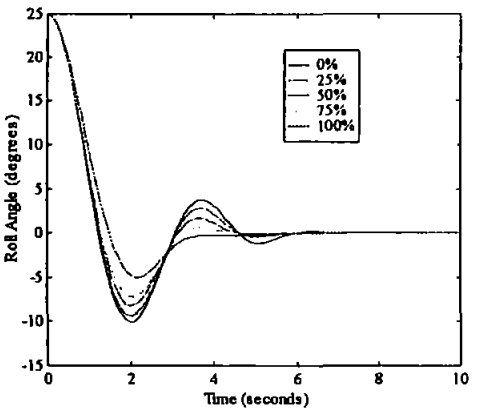
Kalman Filter Roll Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



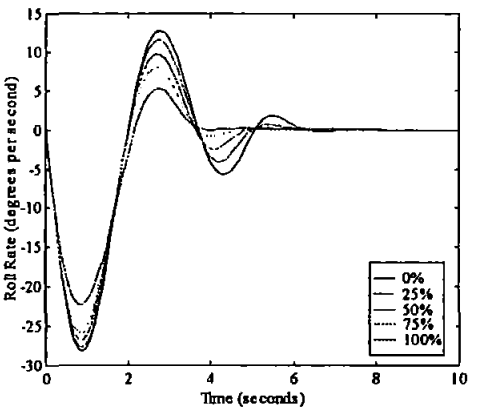
Kalman Filter Roll Rate Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



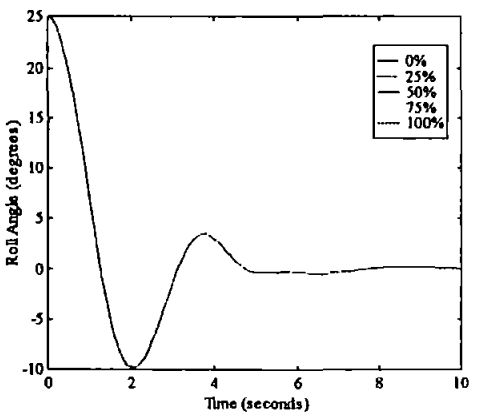
Standard Roll Responses to Roll Sensor Percentage Faults (25° initial position)



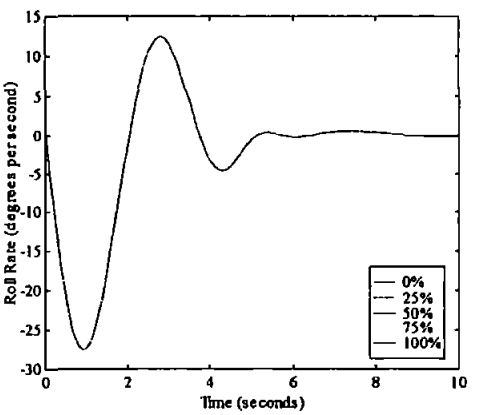
Standard Roll Rate Responses to Roll Sensor Percentage Faults (25° initial position)



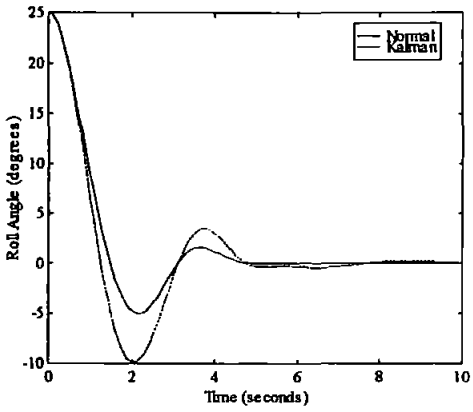
Kalman Filter Roll Responses to Roll Sensor Percentage Faults (25° initial position)



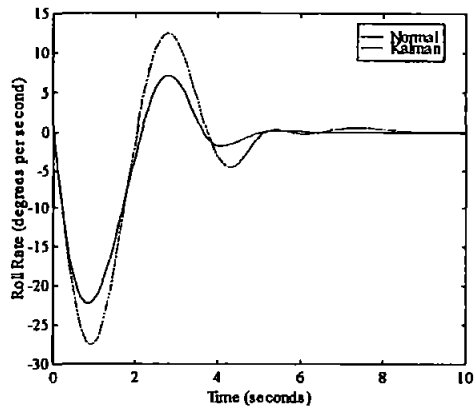
Kalman Filter Roll Rate Responses to Roll Sensor Percentage Faults (25° initial position)



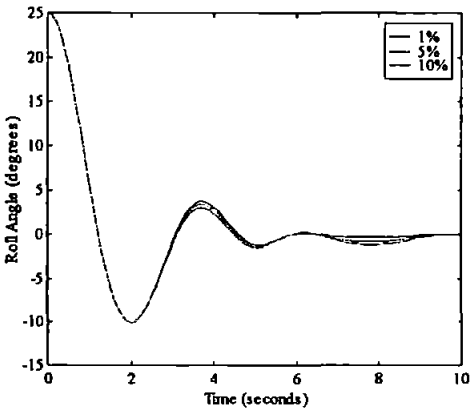
Roll Responses to Roll Sensor Intermittent Total Failure (25° initial position)



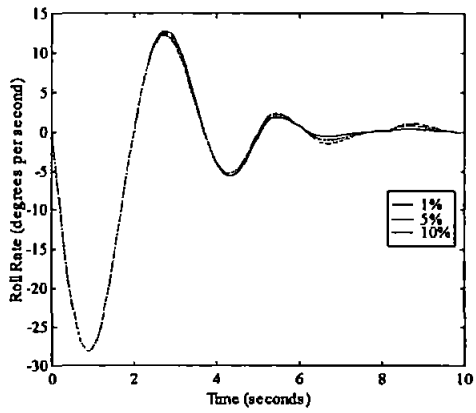
Roll Rate Responses to Roll Sensor Intermittent Total Failure (25° initial position)



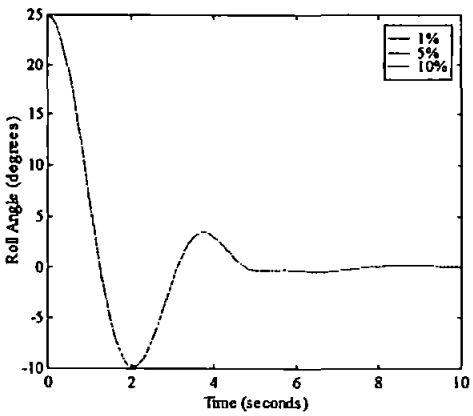
Standard Roll Responses to Roll Sensor Signal to Noise Ratios (25° initial position)



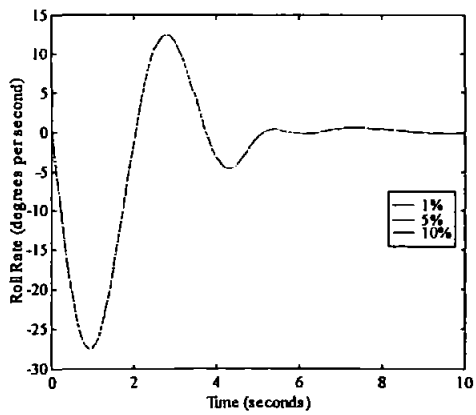
Standard Roll Rate Responses to Roll Sensor Signal to Noise Ratios (25° initial position)



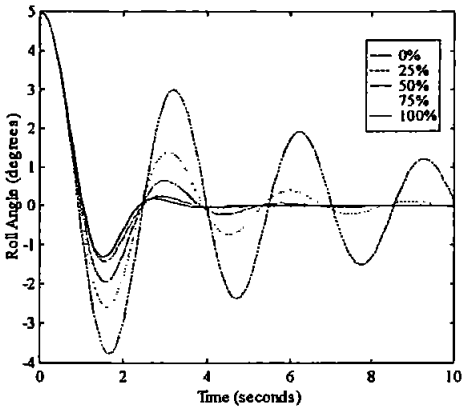
Kalman Filter Roll Responses to Roll Sensor Signal to Noise Ratios (25° initial position)



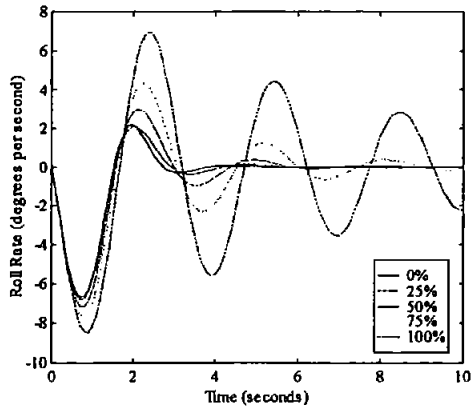
Kalman Filter Roll Rate Responses to Roll Sensor Signal to Noise Ratios (25° initial position)



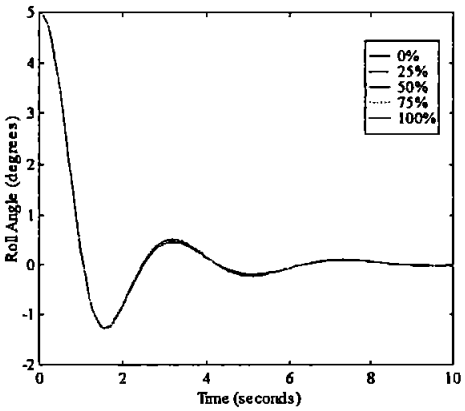
Standard Roll Responses to Roll Rate Sensor Percentage Faults (5° initial position)



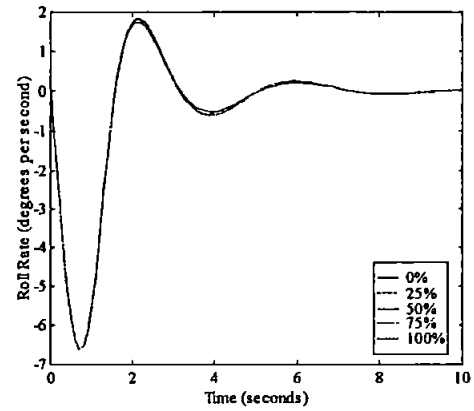
Standard Roll Rate Responses to Roll Rate Sensor Percentage Faults (5° initial position)



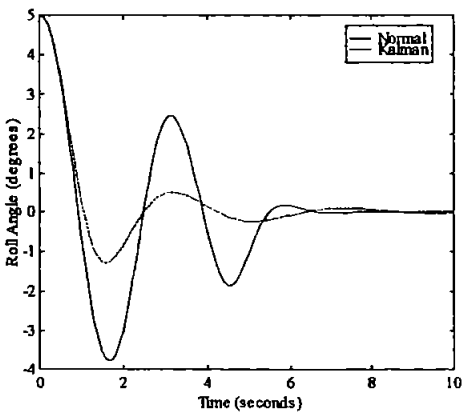
Kalman Filter Roll Responses to Roll Rate Sensor Percentage Faults (5° initial position)



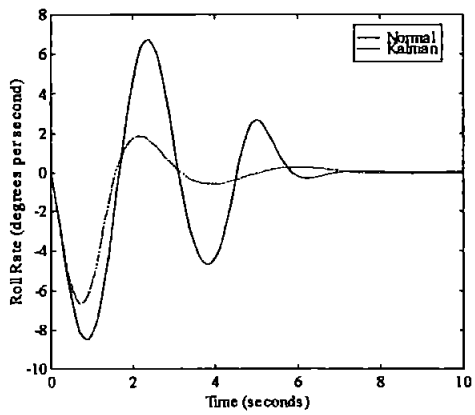
Kalman Filter Roll Rate Responses to Roll Rate Sensor Percentage Faults (5° initial position)



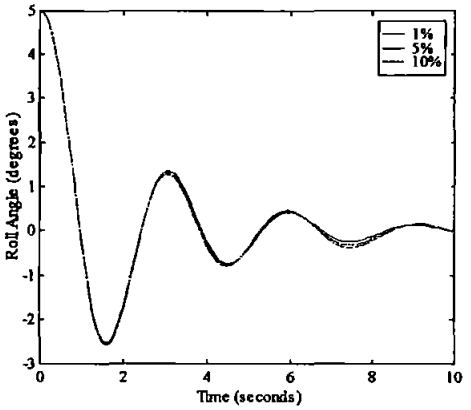
Roll Responses to Roll Rate Sensor Intermittent Failure (5° initial position)



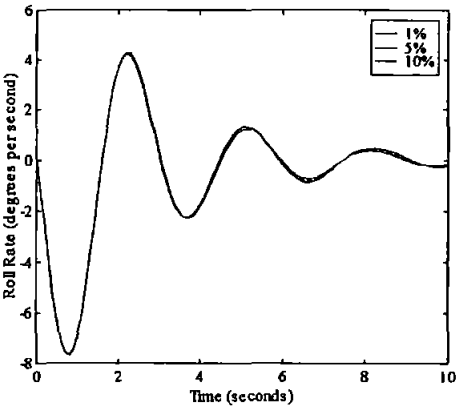
Roll Rate Responses to Roll Rate Sensor Intermittent Failure (5° initial position)



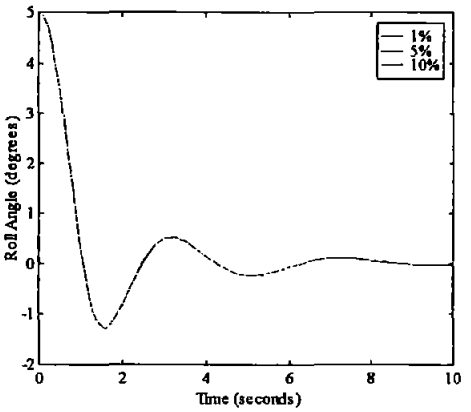
Standard Roll Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



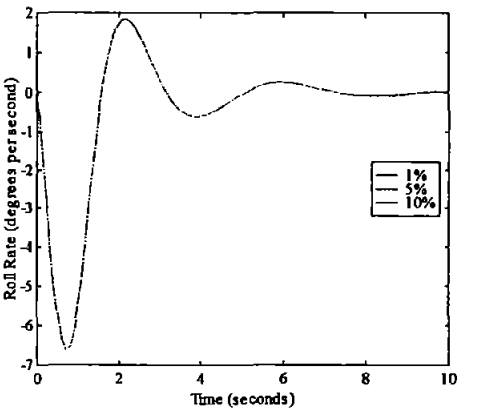
Standard Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



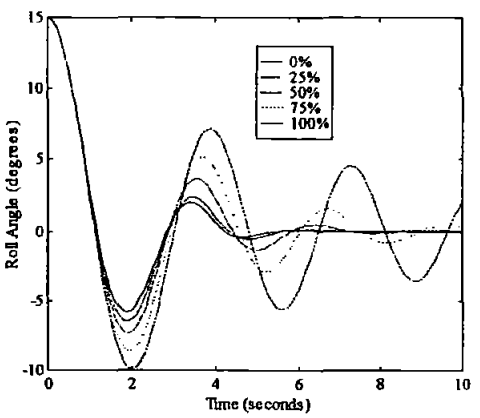
Kalman Filter Roll Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



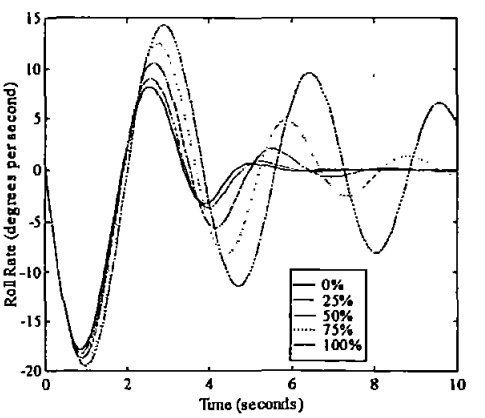
Kalman Filter Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



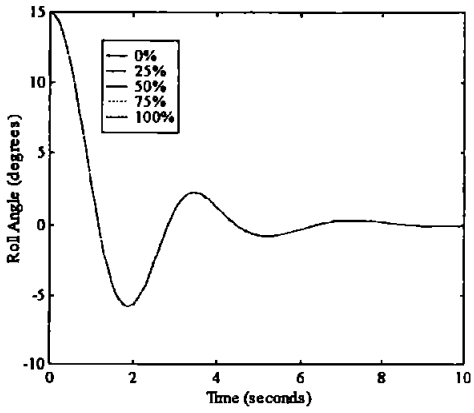
Standard Roll Responses to Roll Rate Sensor Percentage Faults (15° initial position)



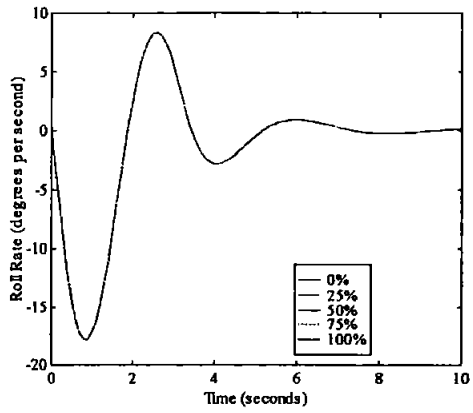
Standard Roll Rate Responses to Roll Rate Sensor Percentage Faults (15° initial position)



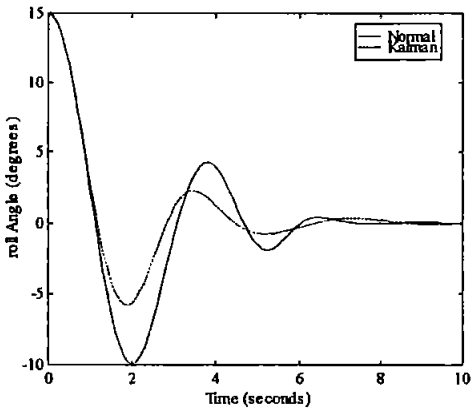
Kalman Filter Roll Responses to Roll Rate Sensor Percentage Faults (15° initial position)



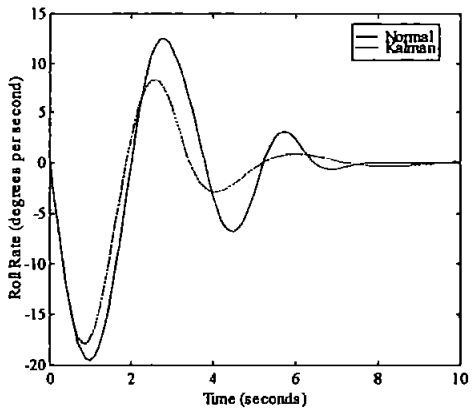
Kalman Filter Roll Rate Responses to Roll Rate Sensor Percentage Faults (15° initial position)



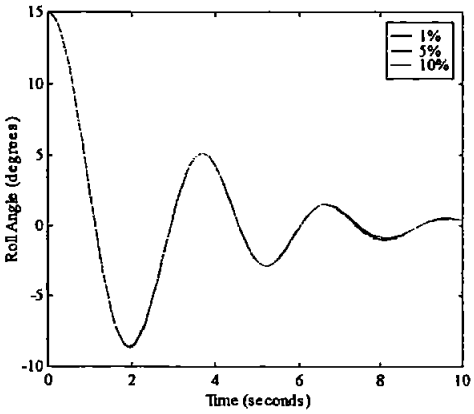
Roll Responses to Roll Rate Sensor Intermittent Total Failure (15° initial position)



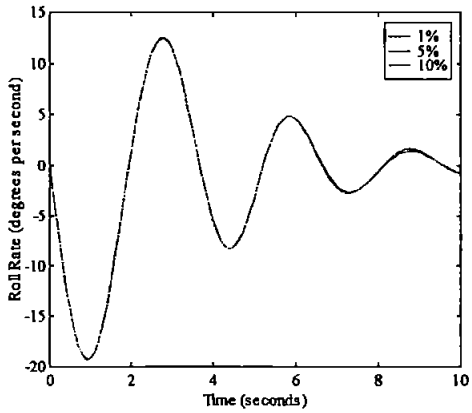
Roll Rate Responses to Roll Rate Sensor Intermittent Total Failure (15° initial position)



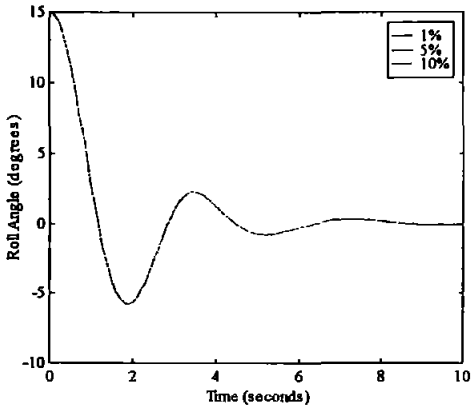
Standard Roll Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



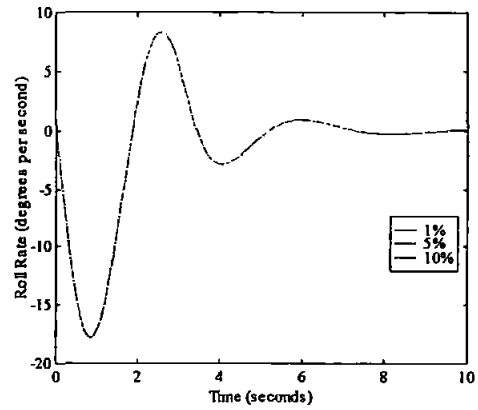
Standard Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



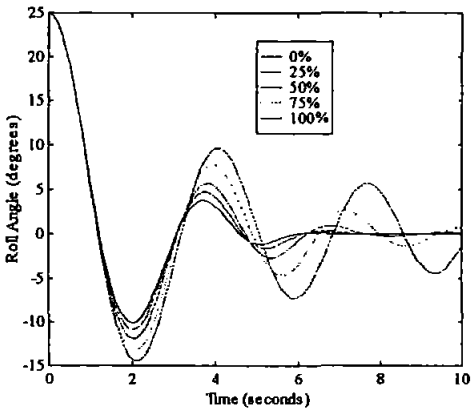
Kalman Filter Roll Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



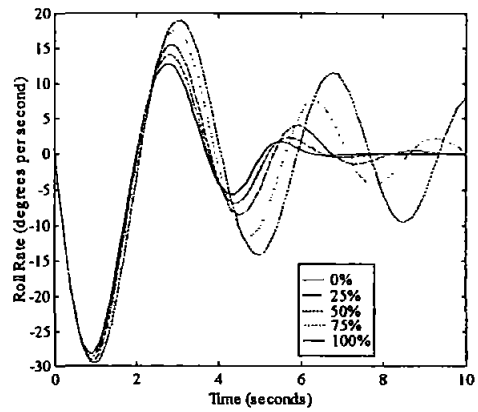
Kalman Filter Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



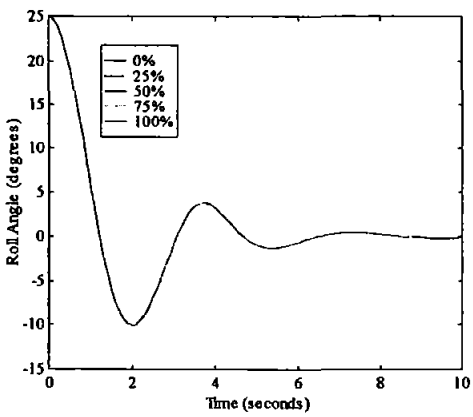
Standard Roll Responses to Roll Rate Sensor Percentage Faults (25° initial position)



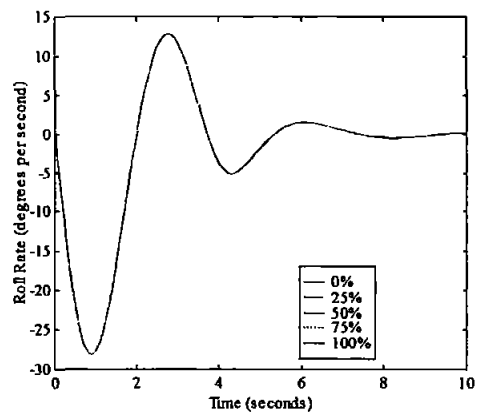
Standard Roll Rate Responses to Roll Rate Sensor Percentage Faults (25° initial position)



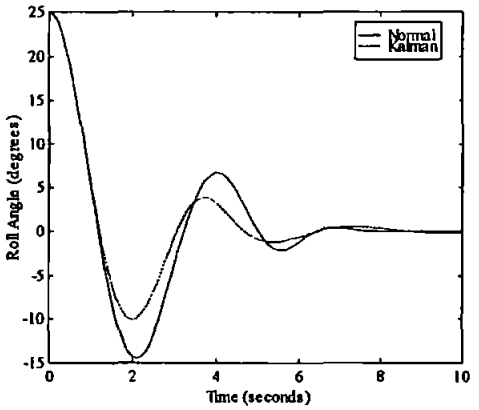
Kalman Filter Roll Responses to Roll Rate Sensor Percentage Faults (25° initial position)



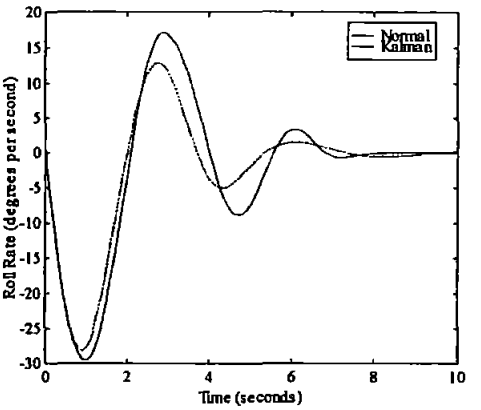
Kalman Filter Roll Rate Responses to Roll Rate Sensor Percentage Faults (25° initial position)



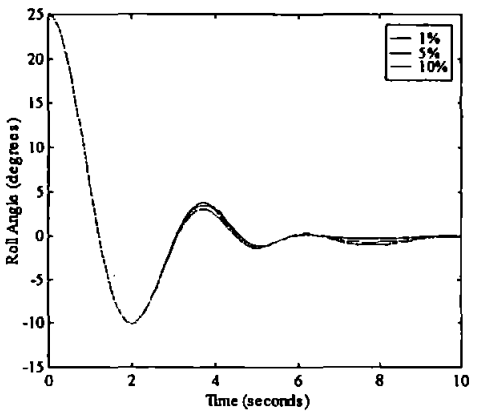
Roll Responses to Roll Rate Sensor
Intermittent Total Failure (25° initial position)



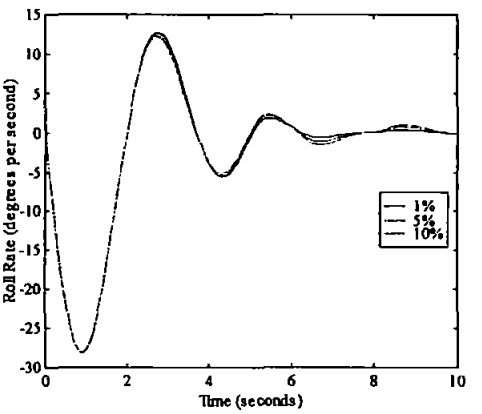
Roll Rate Responses to Roll Rate Sensor
Intermittent Total Failure (25° initial position)



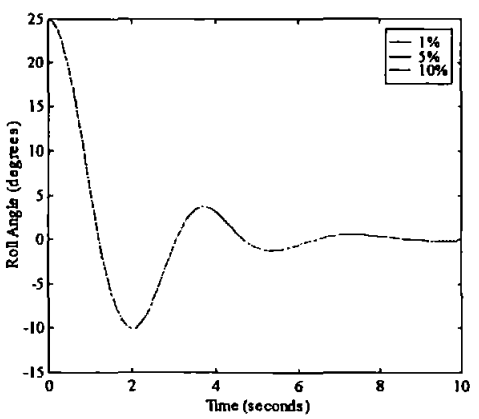
Standard Roll Responses to Roll Rate Sensor
Signal to Noise Ratios (25° initial position)



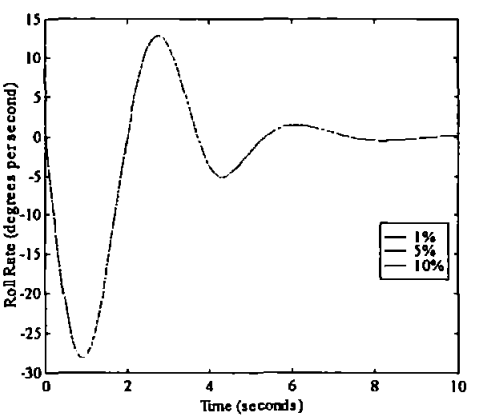
Standard Roll Rate Responses to Roll Rate Sensor
Signal to Noise Ratios (25° initial position)



Kalman Filter Roll Responses to Roll Rate Sensor
Signal to Noise Ratios (25° initial position)

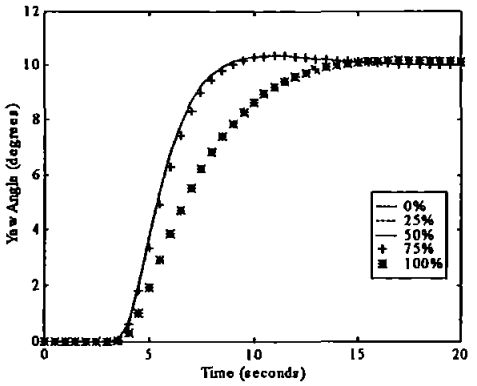


Kalman Filter Roll Rate Responses to Roll Rate Sensor
Signal to Noise Ratios (25° initial position)

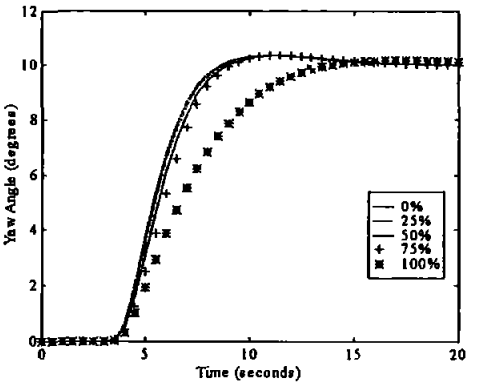


Appendix F Yaw Actuator Failure Results

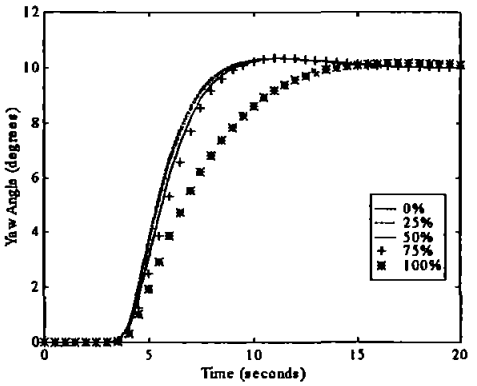
PD Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



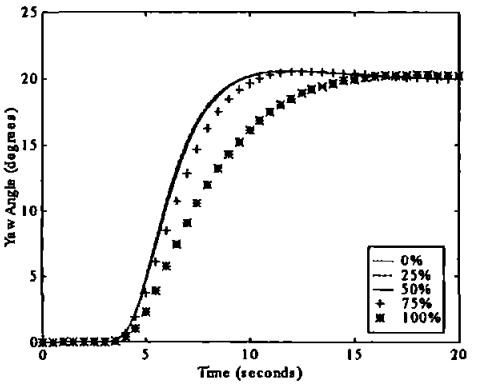
PD Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



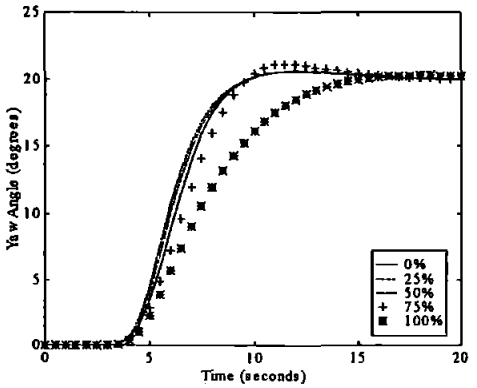
PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



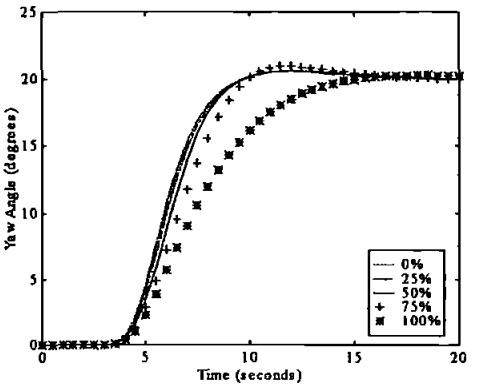
PD Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



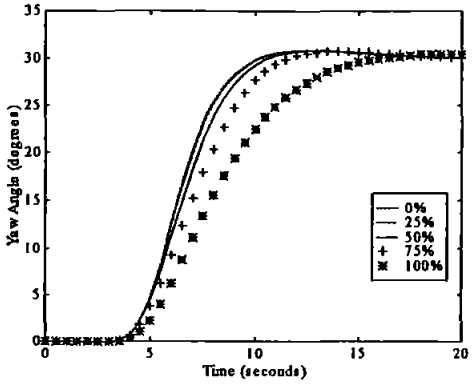
PD Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



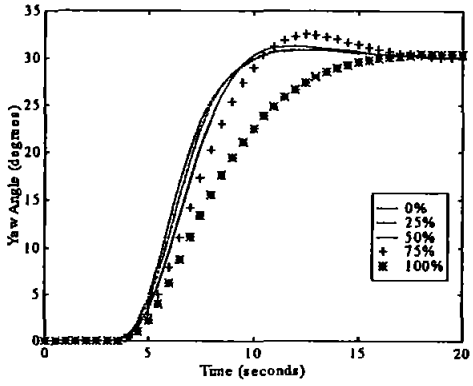
PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



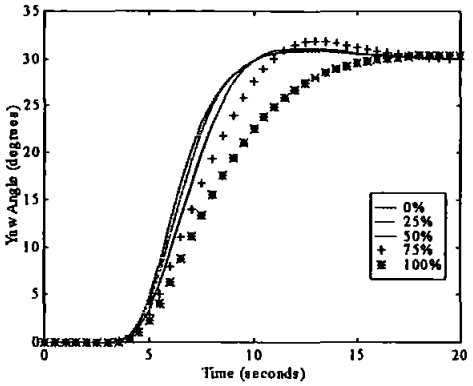
PD Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



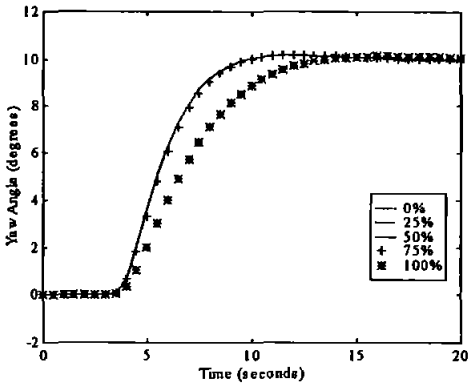
PD Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



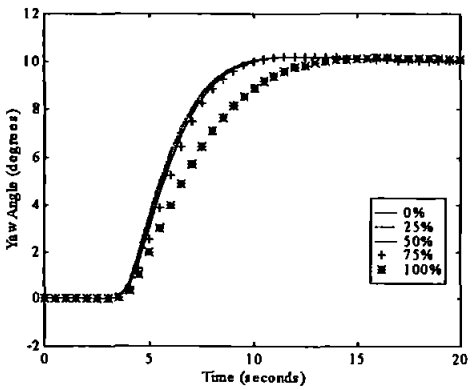
PD Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



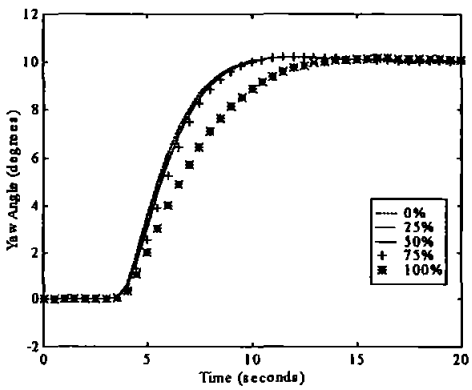
ANFIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



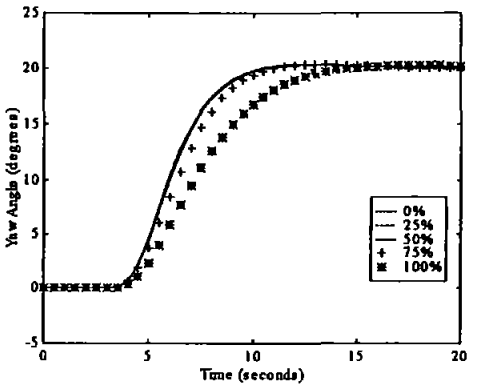
ANFIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



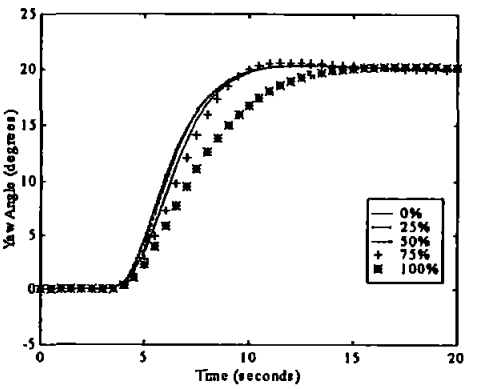
ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



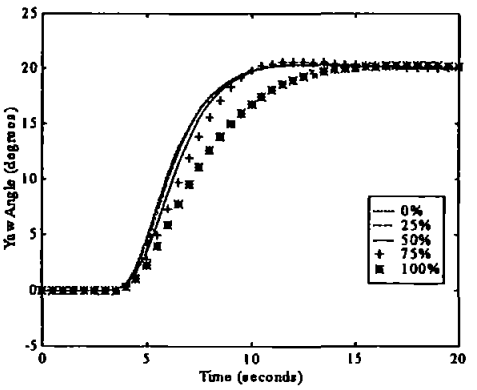
ANFIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



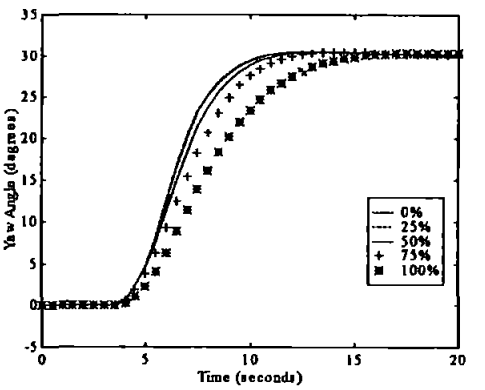
ANFIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



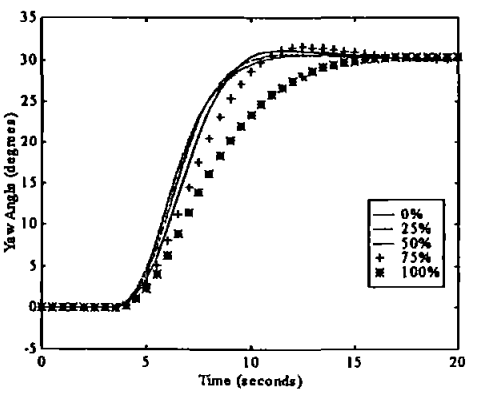
ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



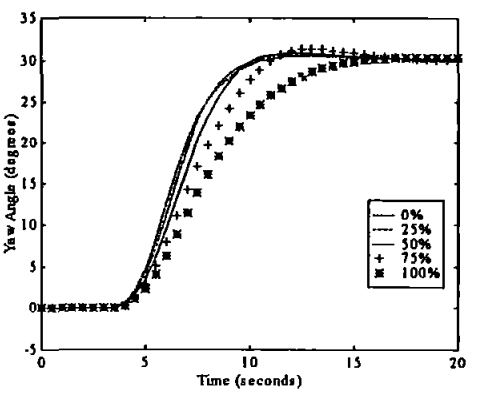
ANFIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



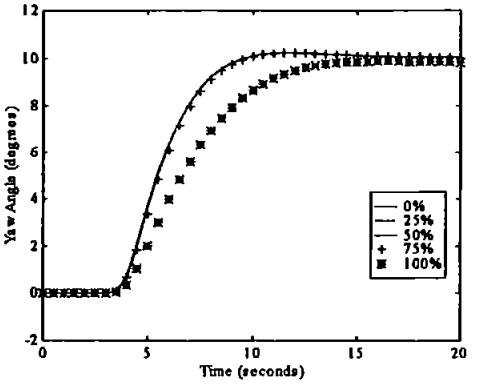
ANFIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



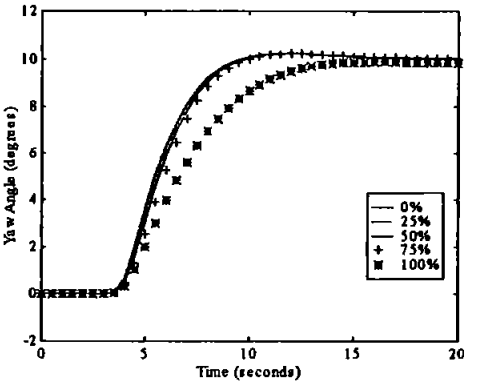
ANFIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



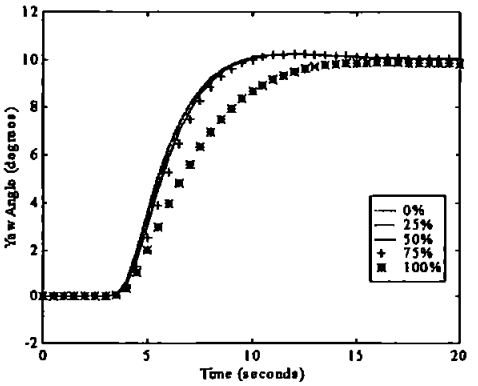
Kalman Filter Enhanced Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



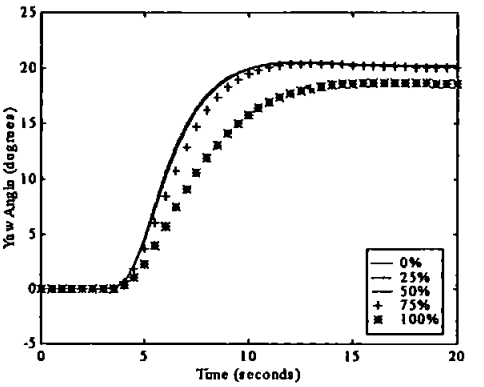
Kalman Filter Enhanced Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



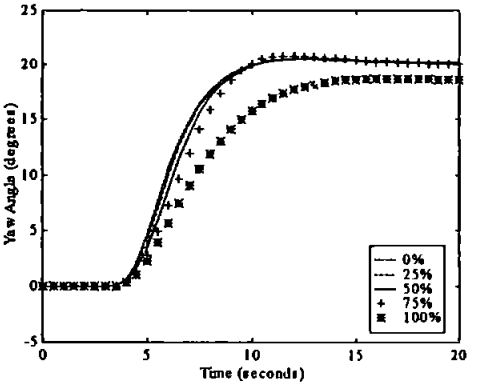
Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



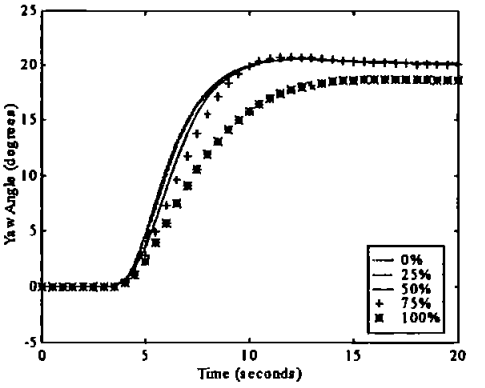
Kalman Filter Enhanced Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



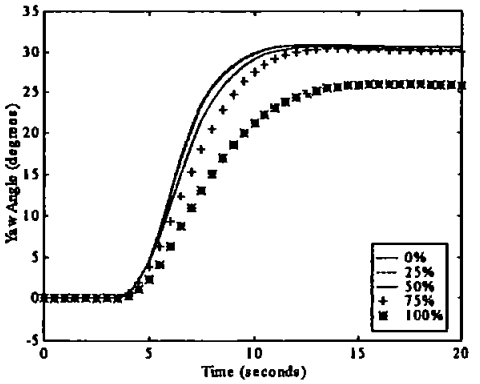
Kalman Filter Enhanced Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



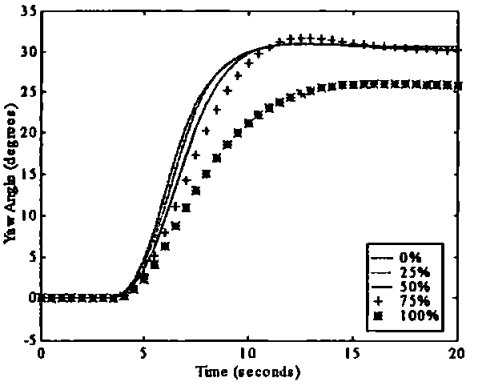
Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



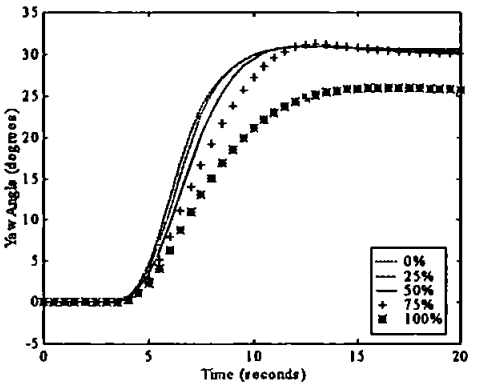
Kalman Filter Enhanced Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



Kalman Filter Enhanced Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults

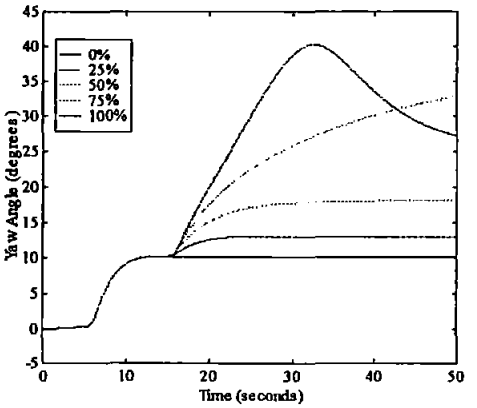


Kalman Filter Enhanced Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

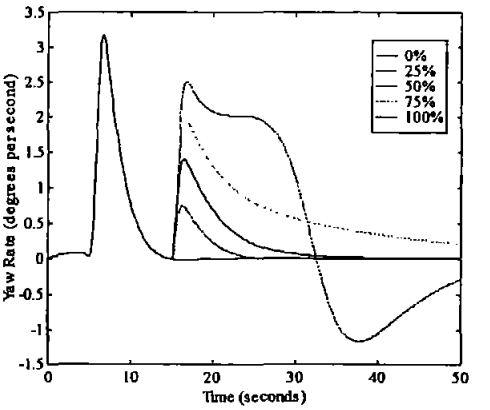


Appendix G Yaw Channel Sensor Recovery FISs Results

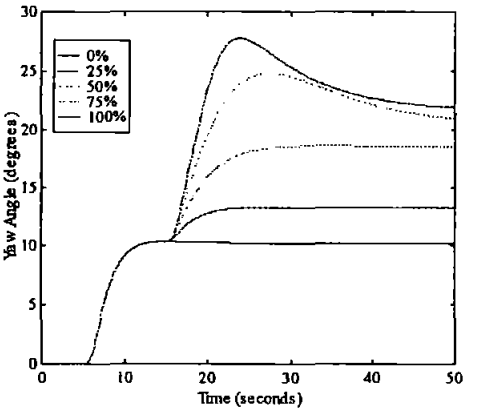
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (10° demand)



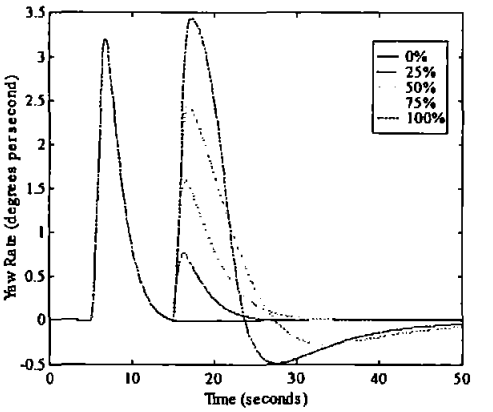
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (10° demand)



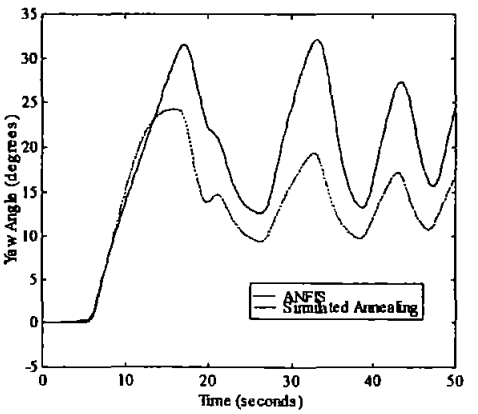
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (10° demand)



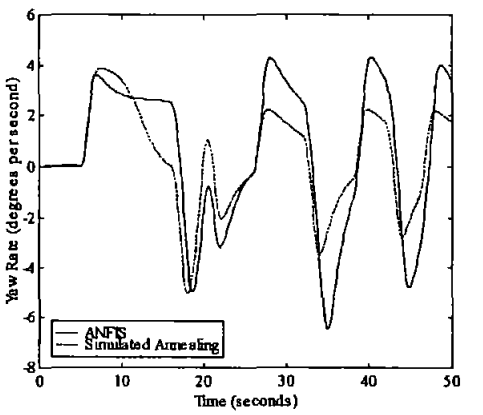
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (10° demand)



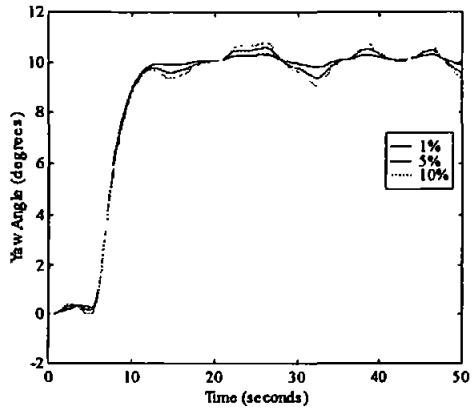
Yaw Responses to Yaw Sensor Intermittent Total Failure (10° demand)



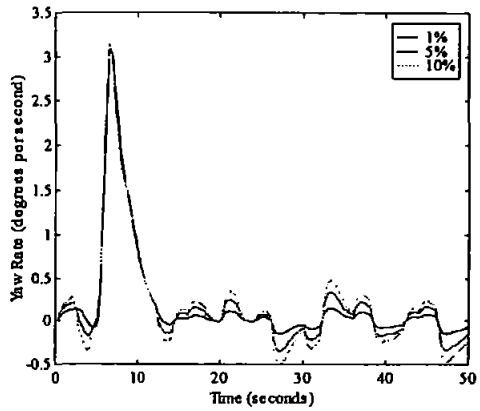
Yaw Rate Responses to Yaw Sensor Intermittent Total Failure (10° demand)



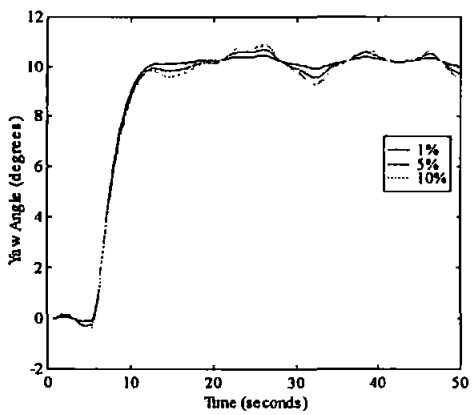
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



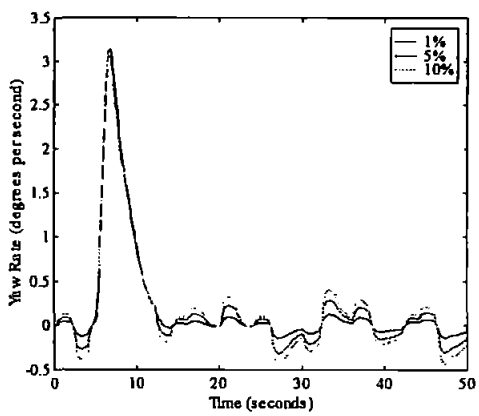
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



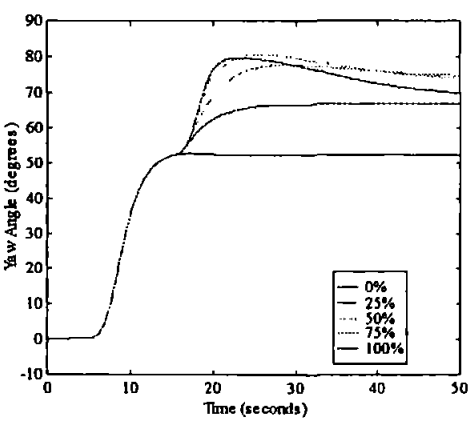
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



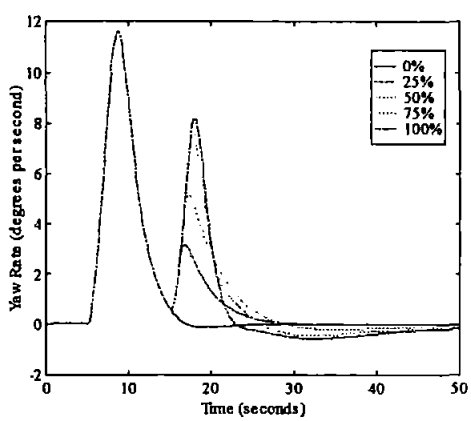
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (10° demand)



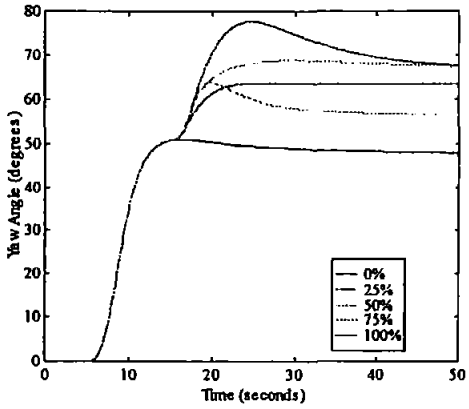
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (50° demand)



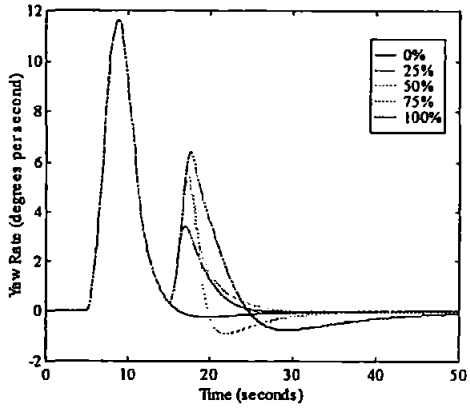
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (50° demand)



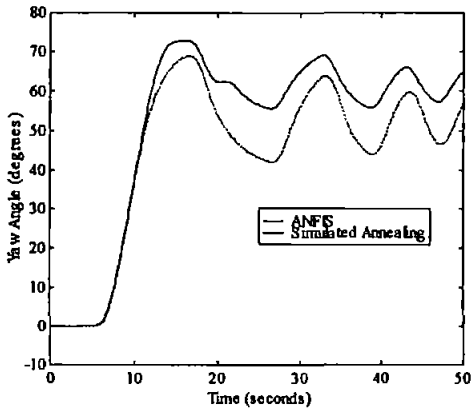
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (50° demand)



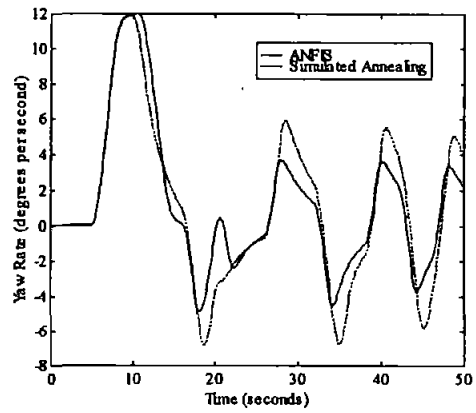
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (50° demand)



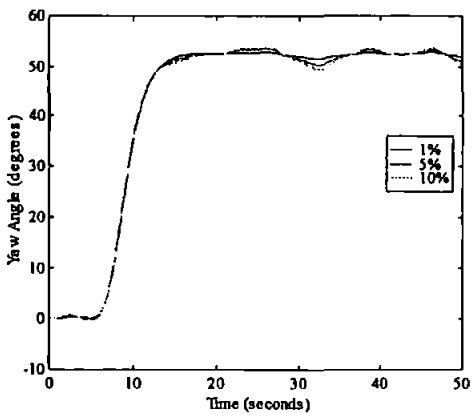
Yaw Responses to Yaw Sensor Intermittent Total Failure (50° demand)



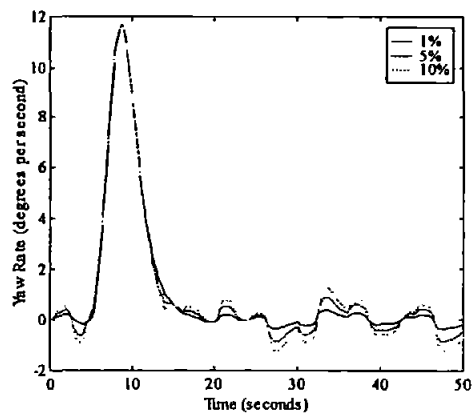
Yaw Rate Responses to Yaw Sensor Intermittent Total Failure (50° demand)



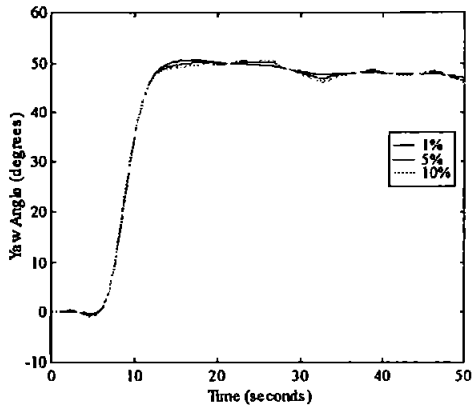
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



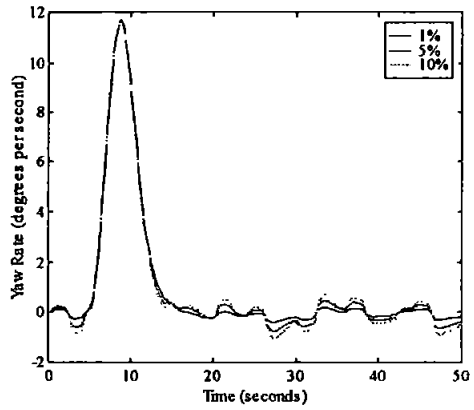
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



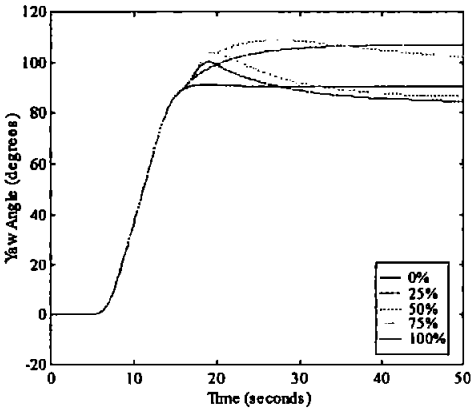
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



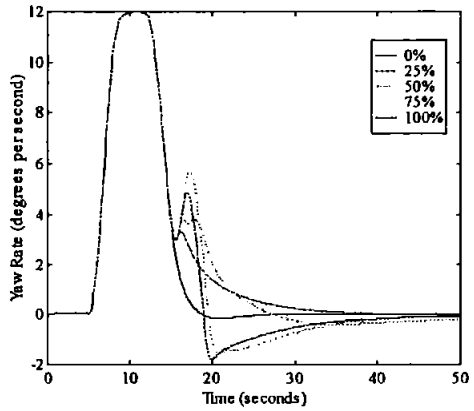
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (50° demand)



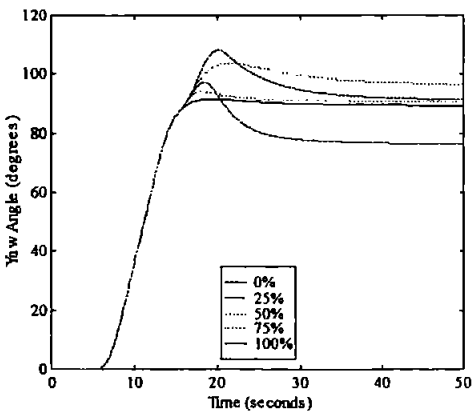
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (90° demand)



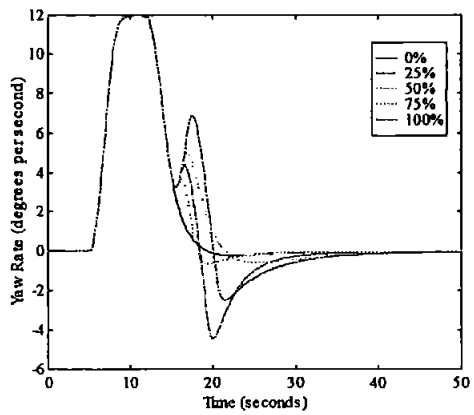
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (90° demand)



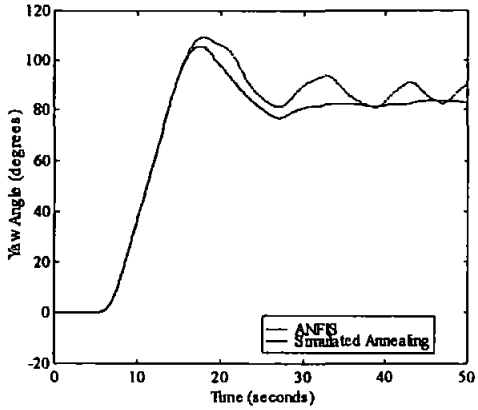
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Percentage Faults (90° demand)



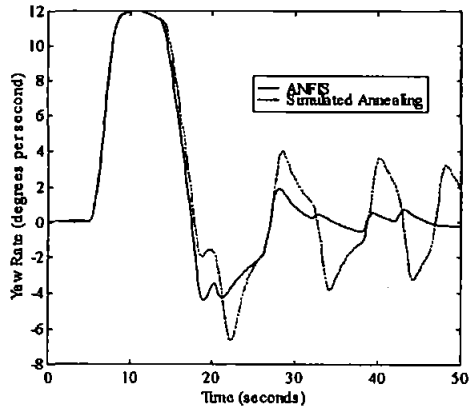
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Percentage Faults (90° demand)



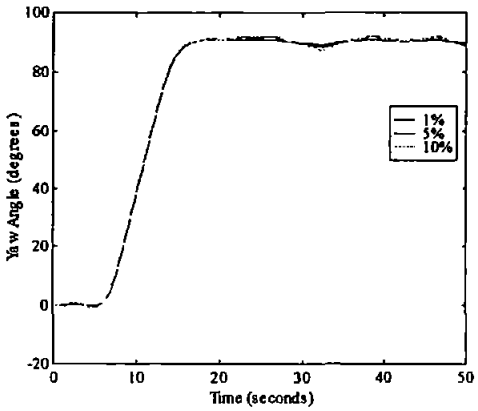
Yaw Responses to Yaw Sensor Intermittent Total Failure (90° demand)



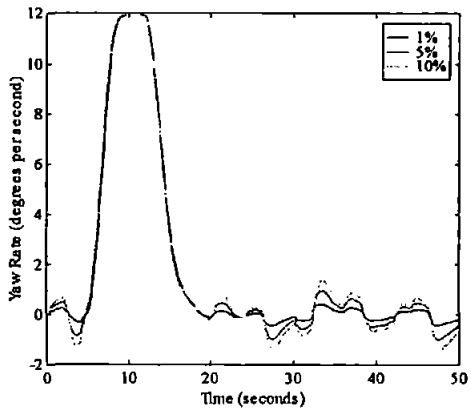
Yaw Rate Responses to Yaw Sensor Intermittent Total Failure (90° demand)



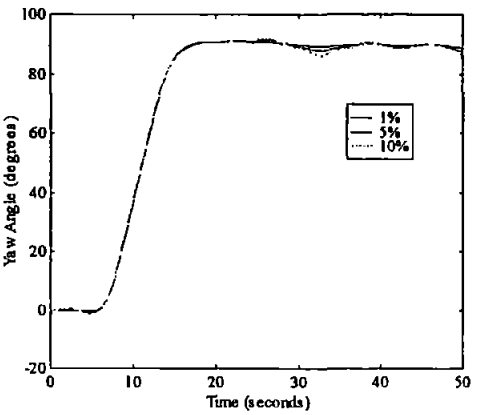
ANFIS Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



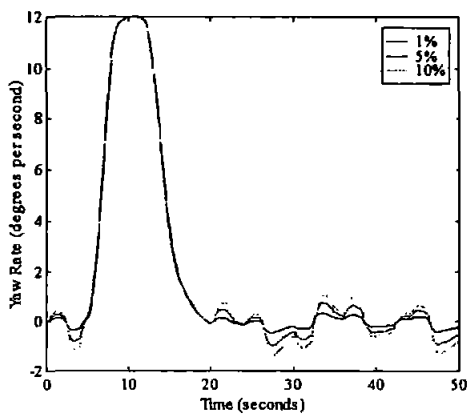
ANFIS Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



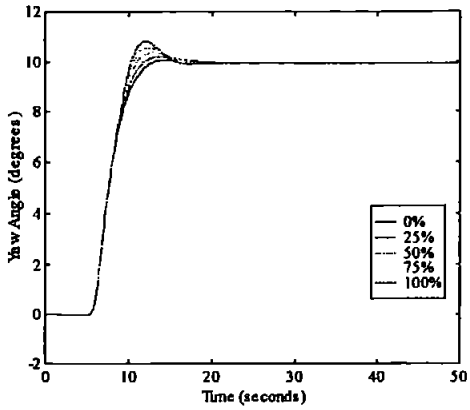
Simulated Annealing Tuned FIS Yaw Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



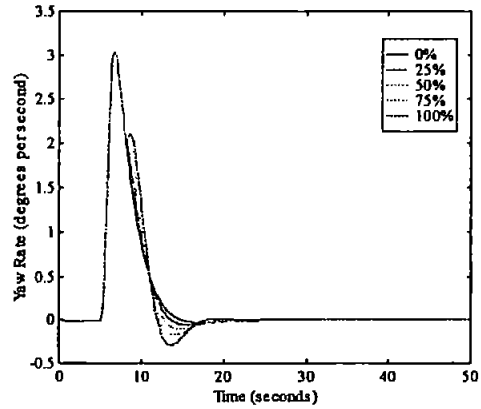
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Sensor Signal to Noise Ratios (90° demand)



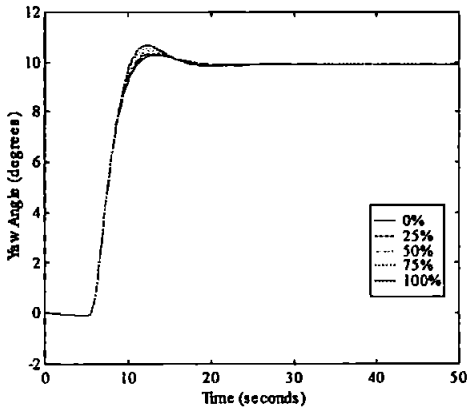
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (10° demand)



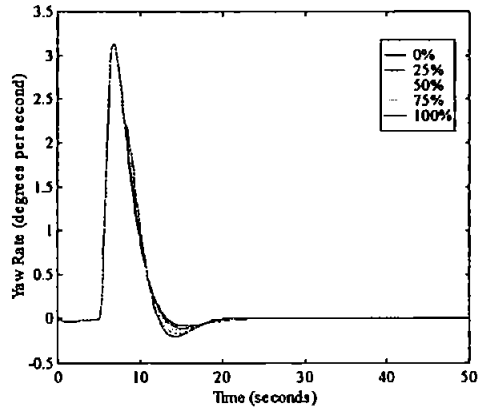
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (10° demand)



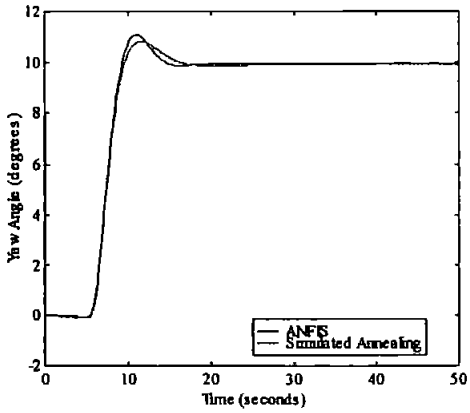
Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (10° demand)



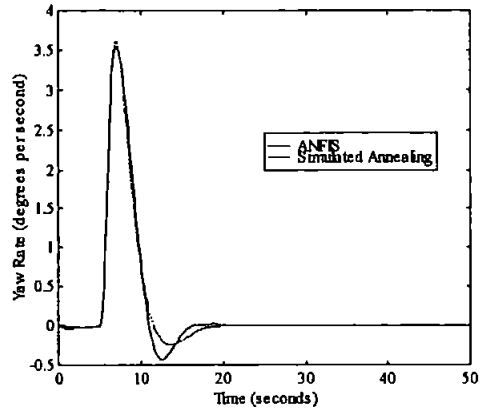
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (10° demand)



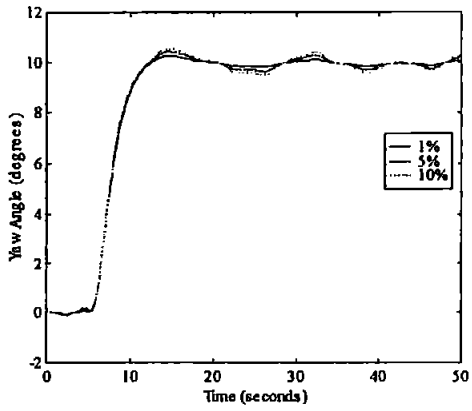
Yaw Responses to Yaw Rate Sensor Intermittent Failure (10° demand)



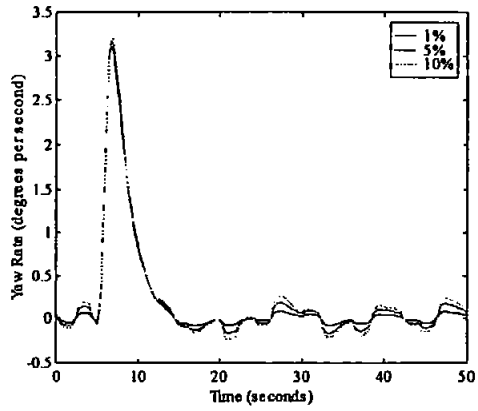
Yaw Rate Responses to Yaw Rate Sensor Intermittent Failure (10° demand)



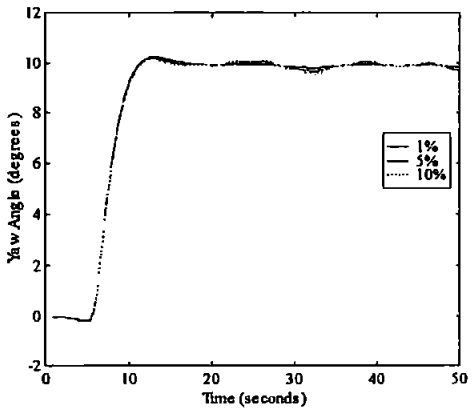
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



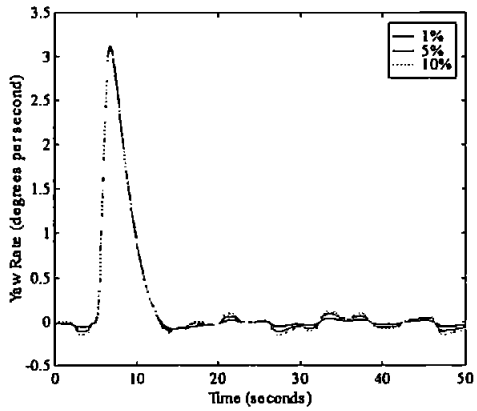
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



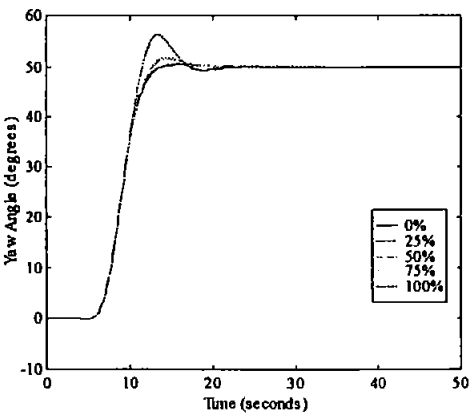
Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



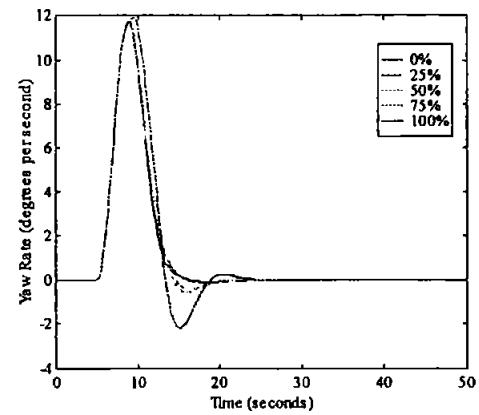
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (10° demand)



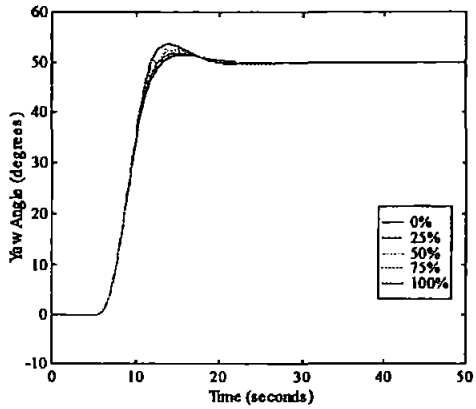
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (50° demand)



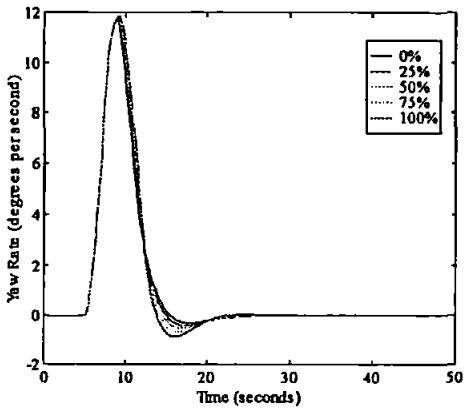
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (50° demand)



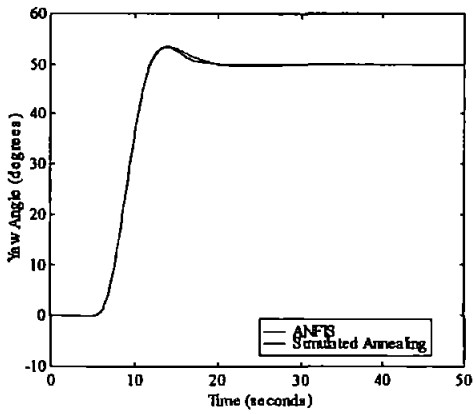
Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (50° demand)



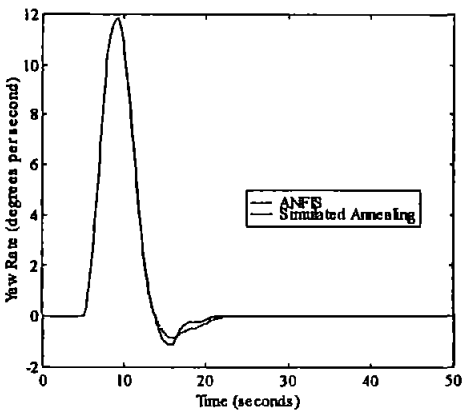
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (50° demand)



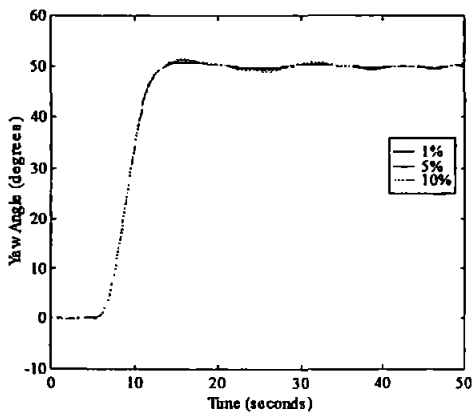
Yaw Responses to Yaw Rate Sensor Intermittent Total Failure (50° demand)



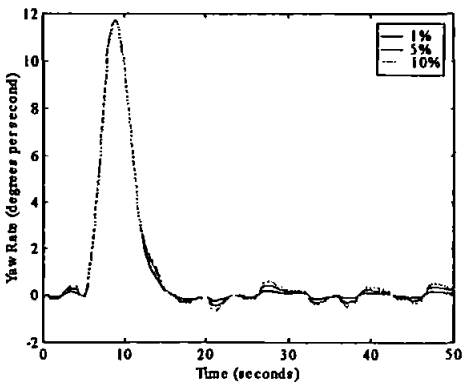
Yaw Rate Responses to Yaw Rate Sensor Intermittent Total Failure (50° demand)



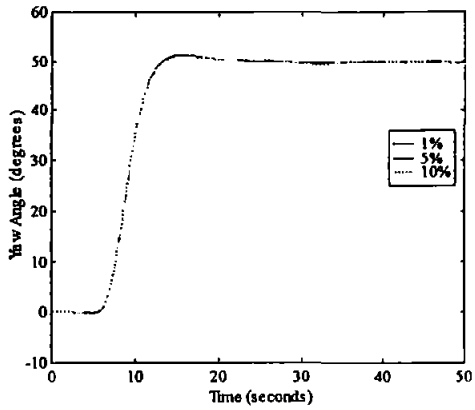
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



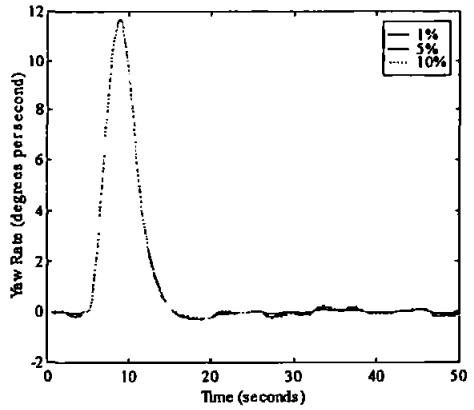
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



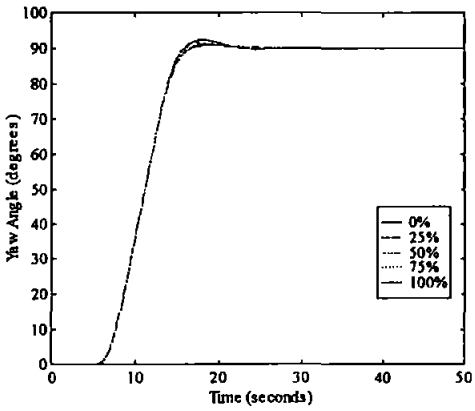
Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



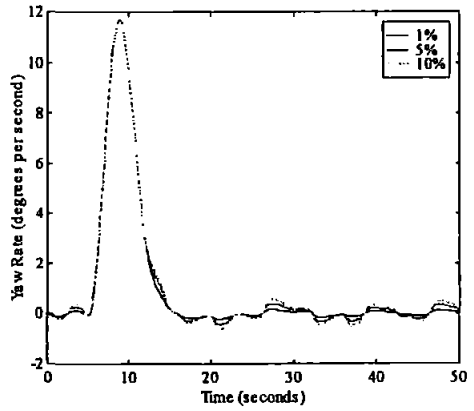
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (50° demand)



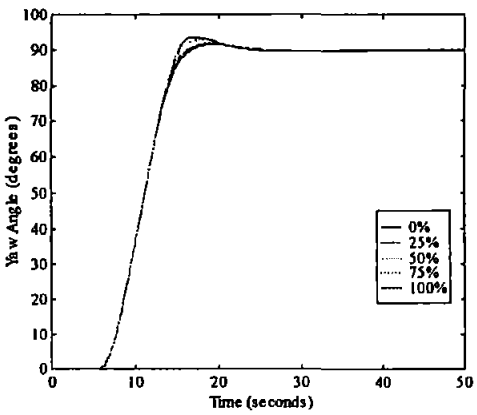
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (90° demand)



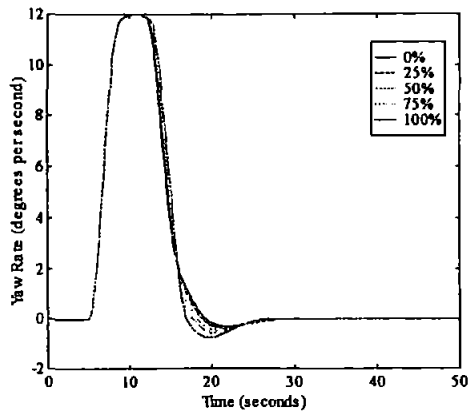
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (90° demand)



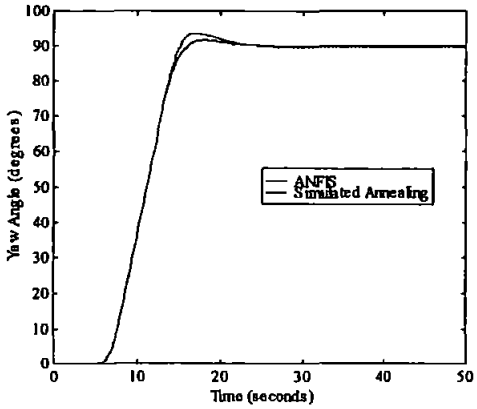
Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Percentage Faults (90° demand)



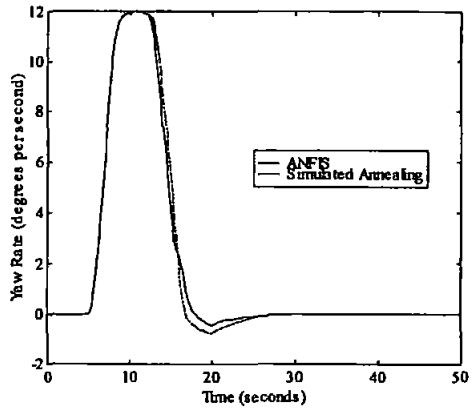
Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Percentage Faults (90° demand)



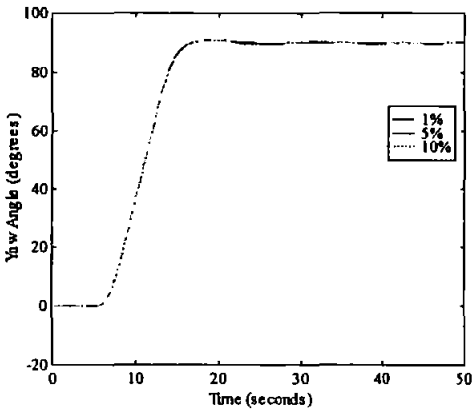
Yaw Responses to Yaw Rate Sensor Intermittent Total Failure (90° demand)



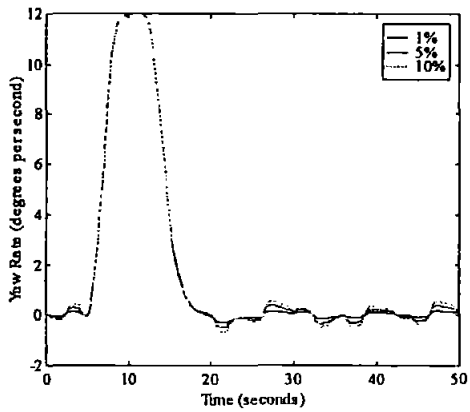
Yaw Rate Responses to Yaw Rate Sensor Intermittent Total Failure (90° demand)



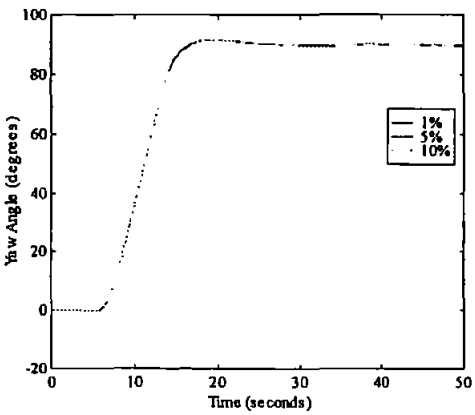
ANFIS Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)



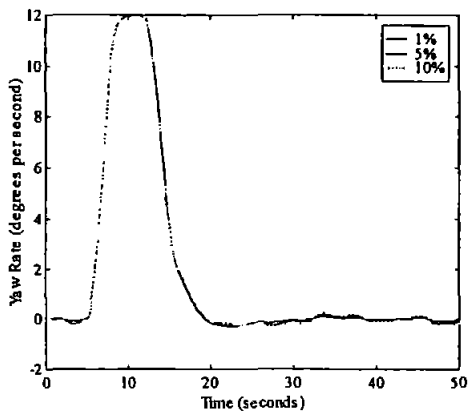
ANFIS Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)



Simulated Annealing Tuned FIS Yaw Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)

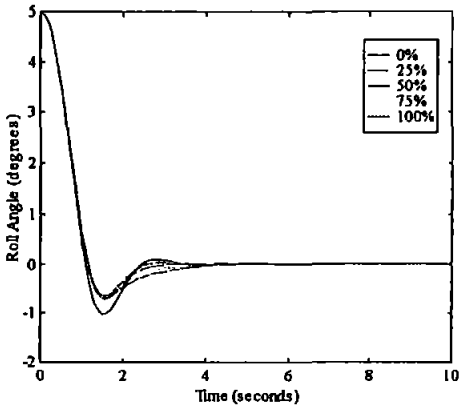


Simulated Annealing Tuned FIS Yaw Rate Responses to Yaw Rate Sensor Signal to Noise Ratios (90° demand)

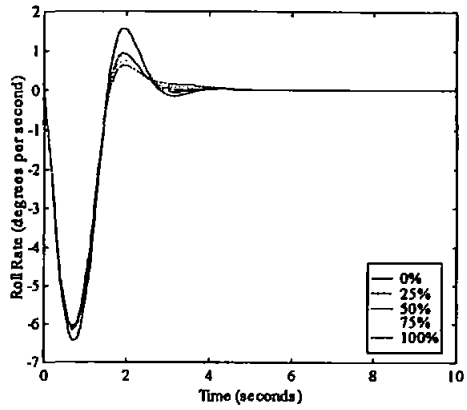


Appendix H Roll Channel Sensor Recovery FISs Results

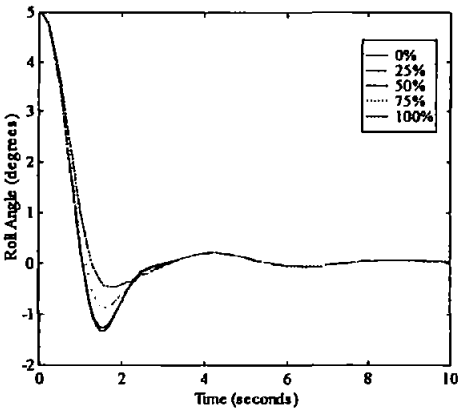
ANFIS Tuned FIS Roll Responses to Roll Sensor Percentage Faults (5° initial position)



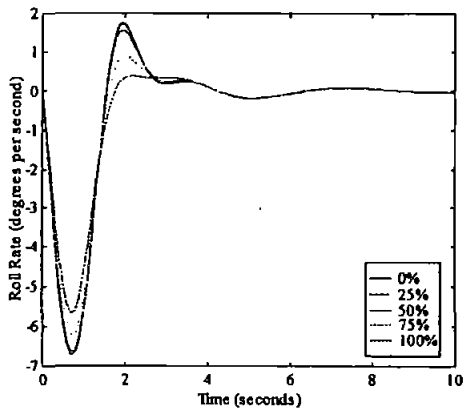
ANFIS Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (5° initial position)



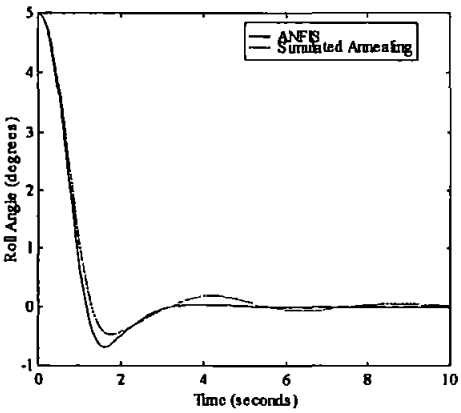
Simulated Annealing Tuned FIS Roll Responses to Roll Sensor Percentage Faults (5° initial position)



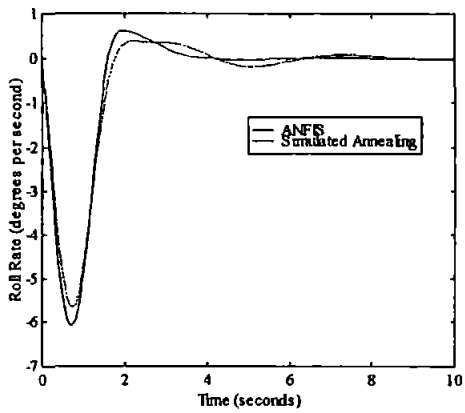
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (5° initial position)



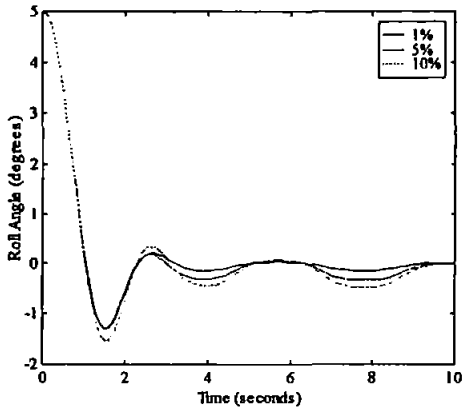
Roll Responses to Roll Sensor Intermittent Total Failure (5° initial position)



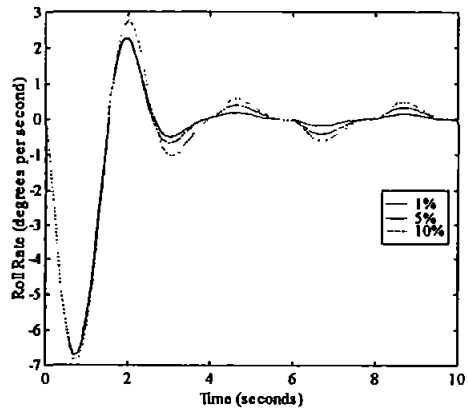
Roll Rate Responses to Roll Sensor Intermittent Total Failure (5° initial position)



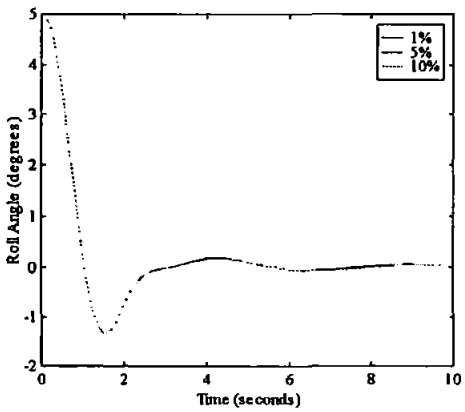
ANFIS Tuned FIS Roll Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



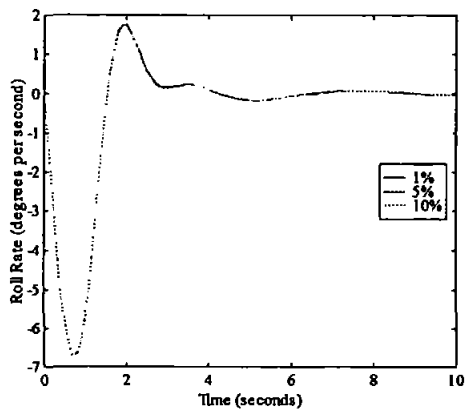
ANFIS Tuned FIS Roll Rate Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



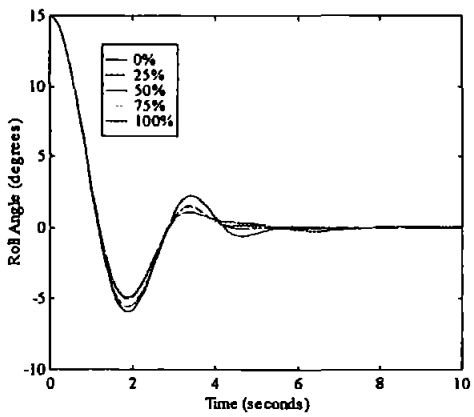
Simulated Annealing Tuned FIS Roll Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



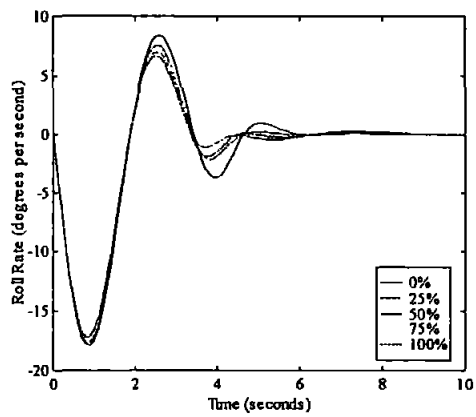
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Sensor Signal to Noise Ratios (5° initial position)



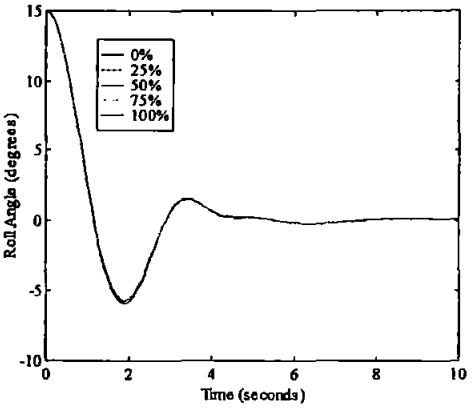
ANFIS Tuned FIS Roll Responses to Roll Sensor Percentage Faults (15° initial position)



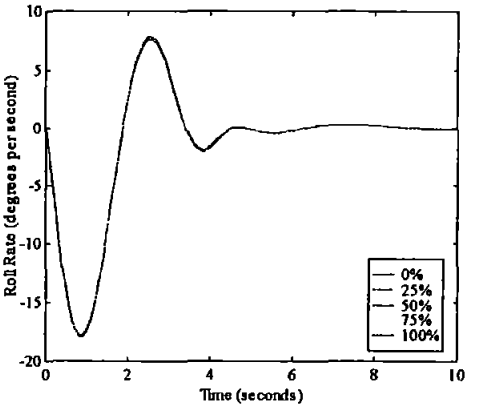
ANFIS Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (15° initial position)



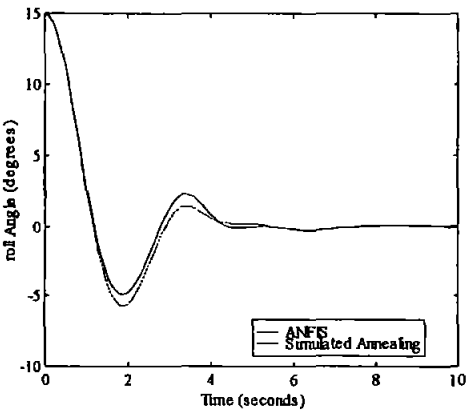
Simulated Annealing Tuned FIS Roll Responses to Roll Sensor Percentage Faults (15° initial position)



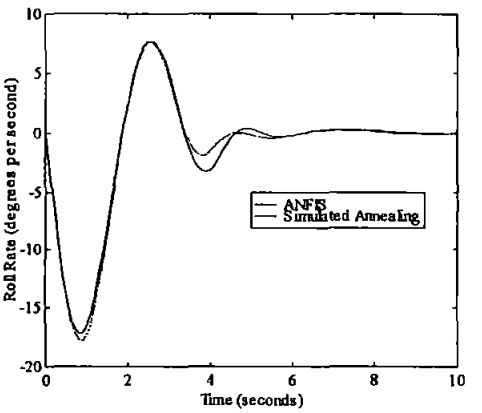
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (15° initial position)



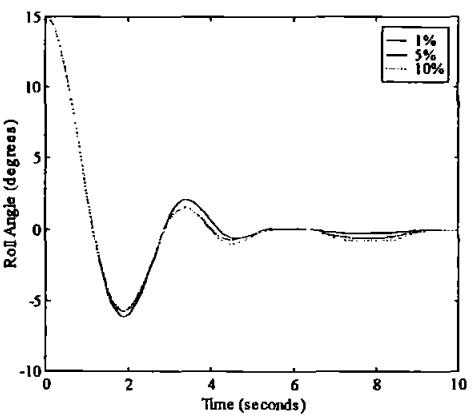
Roll Responses to Roll Sensor Intermittent Total Failure (15° initial position)



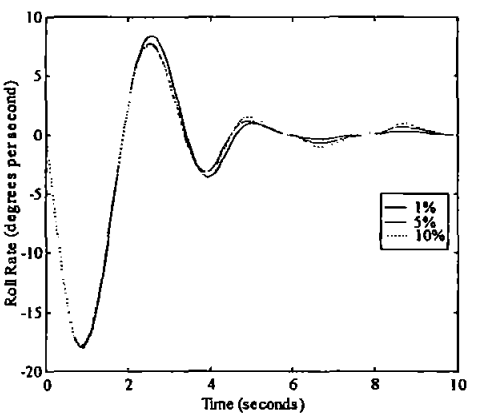
Roll Rate Responses to Roll Sensor Intermittent Total Failure (15° initial position)



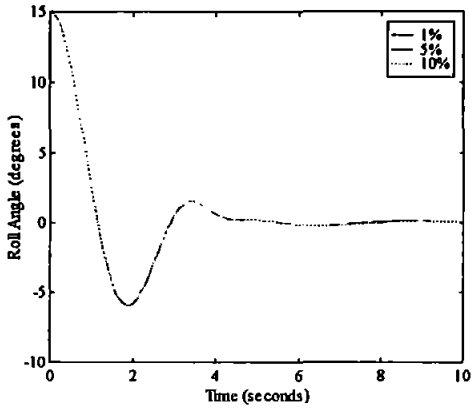
ANFIS Tuned FIS Roll Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



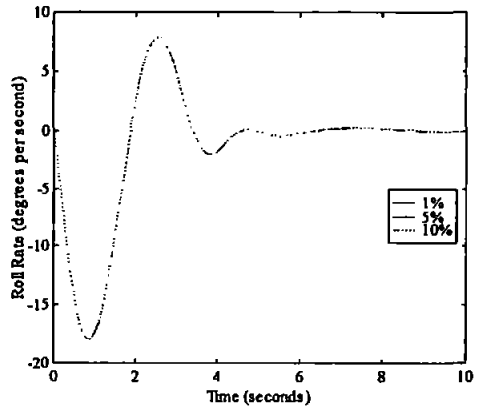
ANFIS Tuned FIS Roll Rate Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



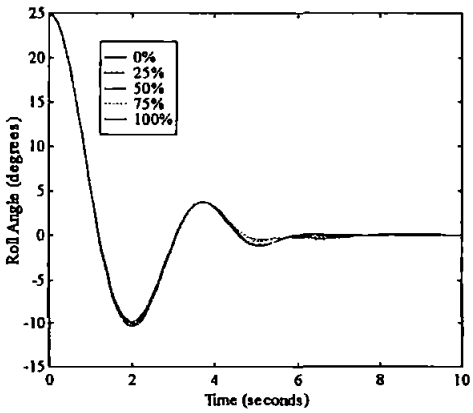
Simulated Annealing Tuned FIS Roll Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



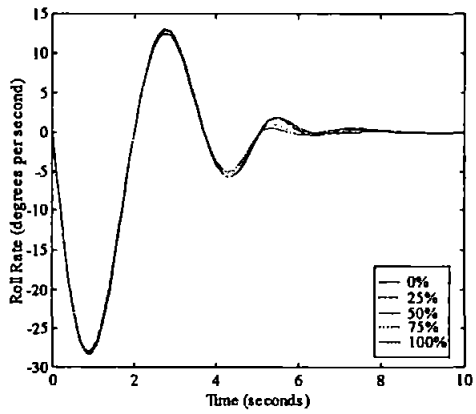
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Sensor Signal to Noise Ratios (15° initial position)



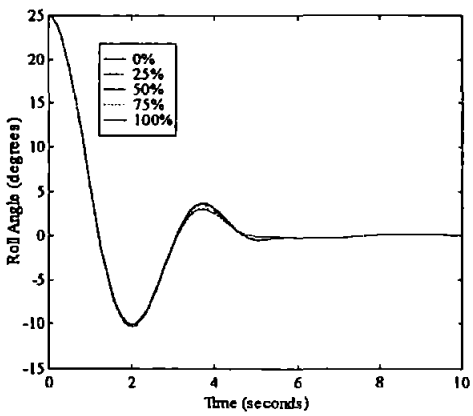
ANFIS Tuned FIS Roll Responses to Roll Sensor Percentage Faults (25° initial position)



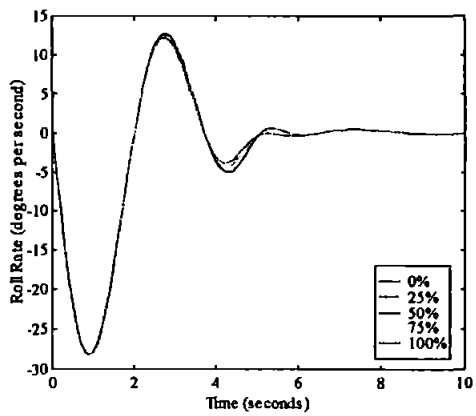
ANFIS Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (25° initial position)



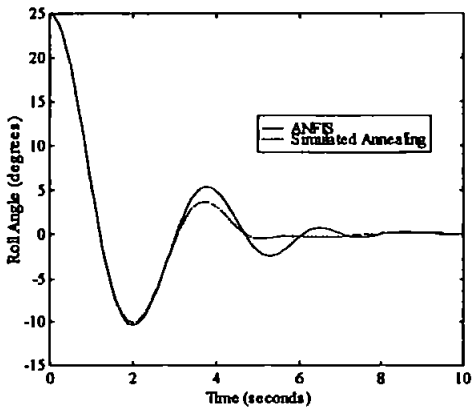
Simulated Annealing Tuned FIS Roll Responses to Roll Sensor Percentage Faults (25° initial position)



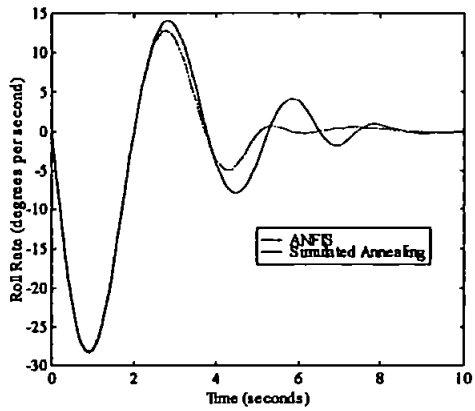
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Sensor Percentage Faults (25° initial position)



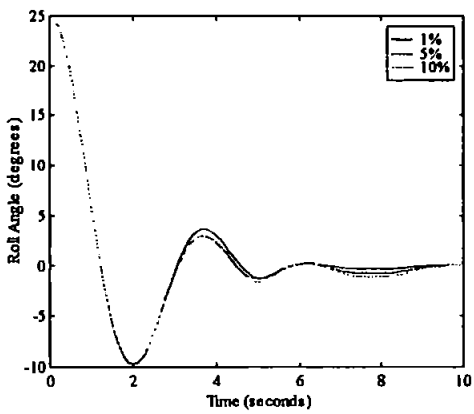
Roll Responses to Roll Sensor
Intermittent Total Failure (25° initial position)



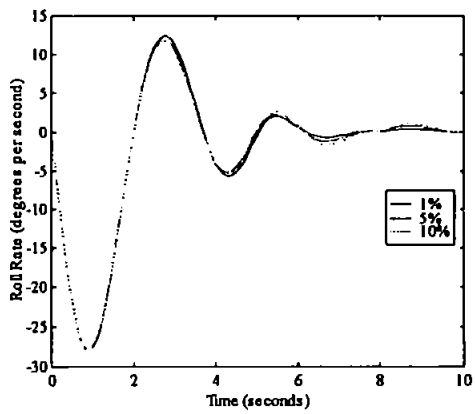
Roll Rate Responses to Roll Sensor
Intermittent Total Failure (25° initial position)



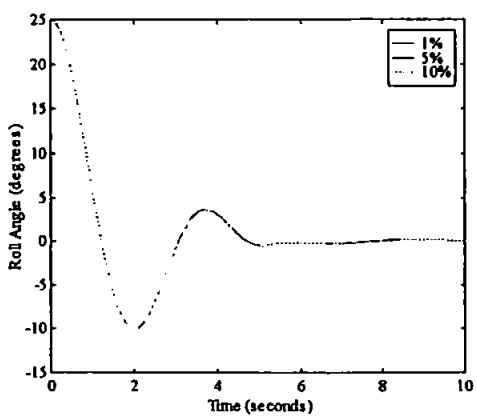
ANFIS Tuned FIS Roll Responses to Roll Sensor
Signal to Noise Ratios (25° initial position)



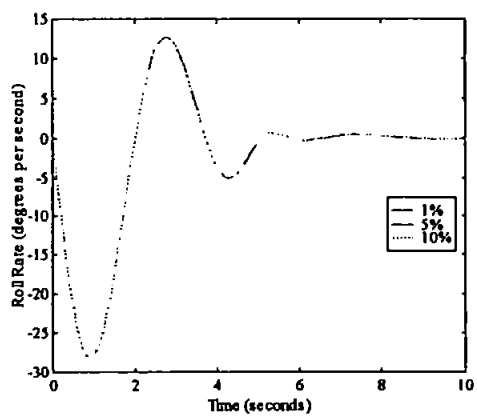
ANFIS Tuned FIS Roll Rate Responses to Roll
Sensor Signal to Noise Ratios (25° initial position)



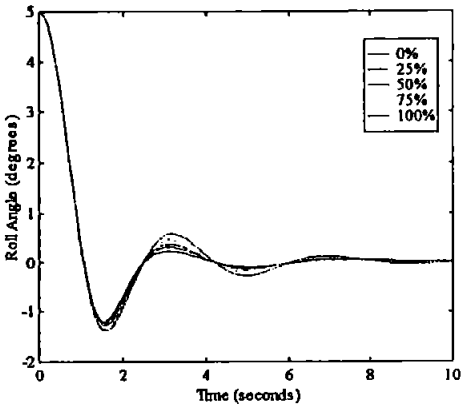
Simulated Annealing Tuned FIS Roll Responses to
Roll Sensor Signal to Noise Ratios (25° initial position)



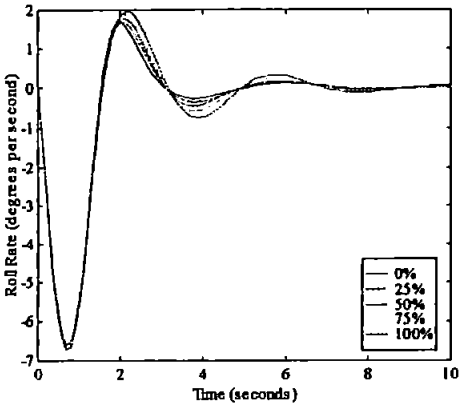
Simulated Annealing Tuned FIS Roll Rate Responses to
Roll Sensor Signal to Noise Ratios (25° initial position)



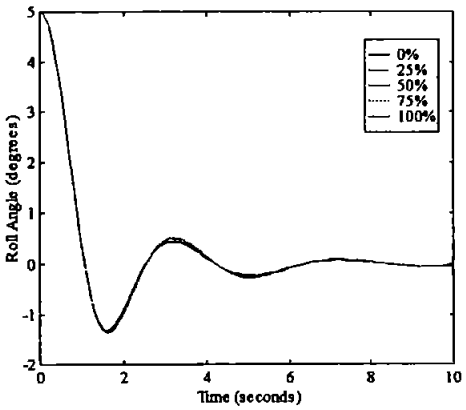
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (5° initial position)



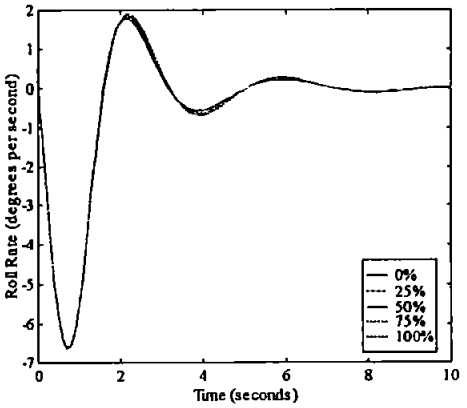
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (5° initial position)



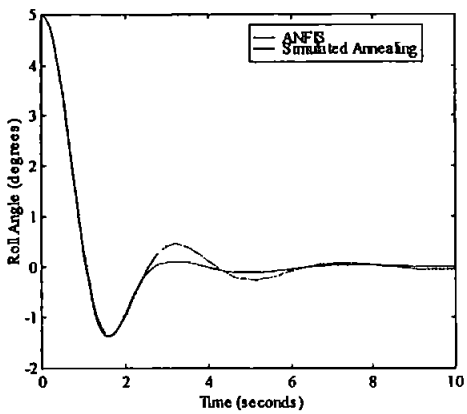
Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (5° initial position)



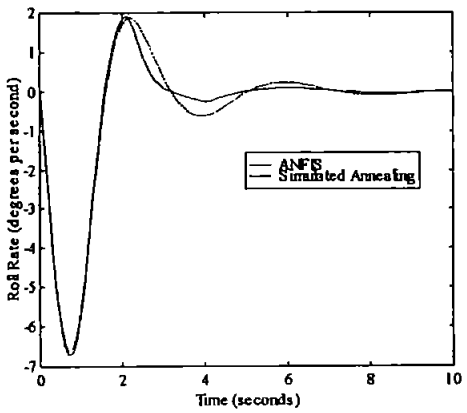
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (5° initial position)



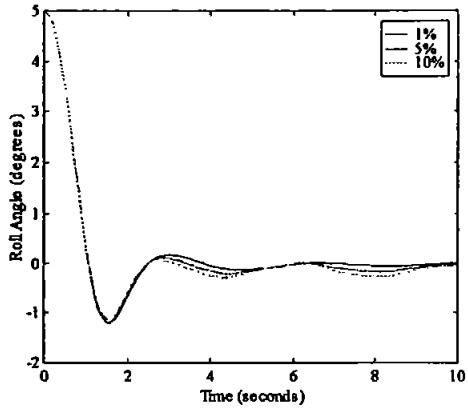
Roll Responses to Roll Rate Sensor Intermittent Failure (5° initial position)



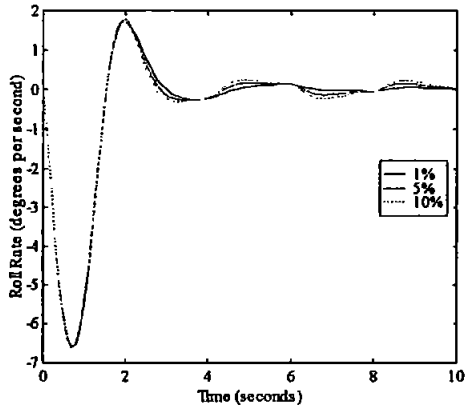
Roll Rate Responses to Roll Rate Sensor Intermittent Failure (5° initial position)



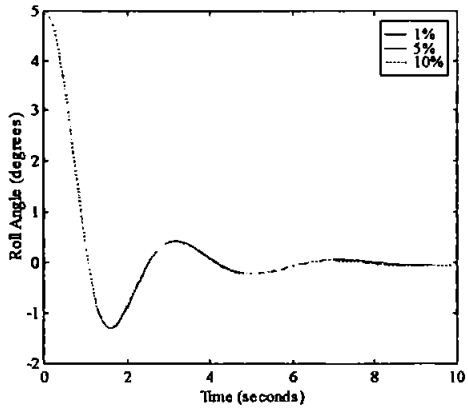
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



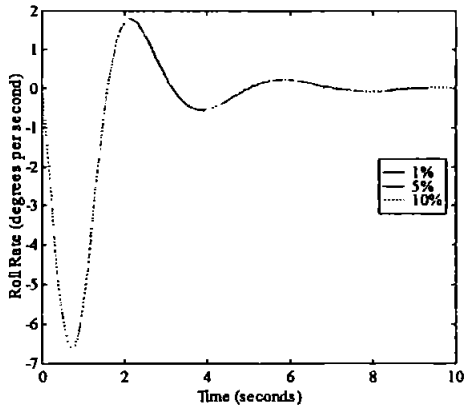
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



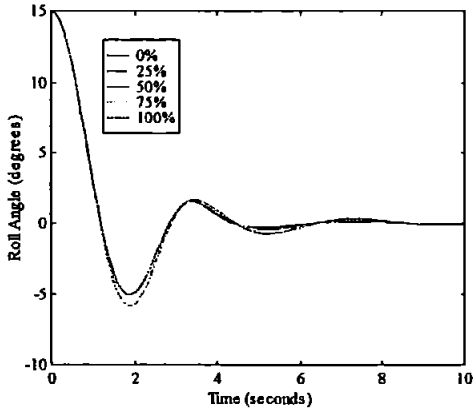
Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



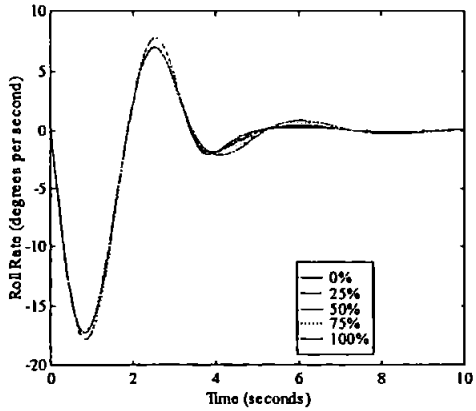
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (5° initial position)



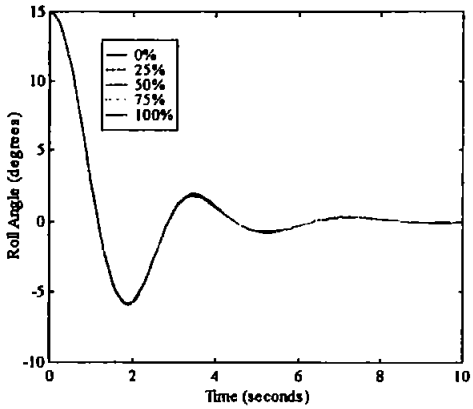
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (15° initial position)



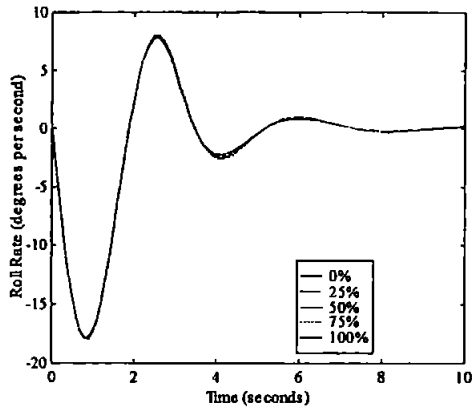
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (15° initial position)



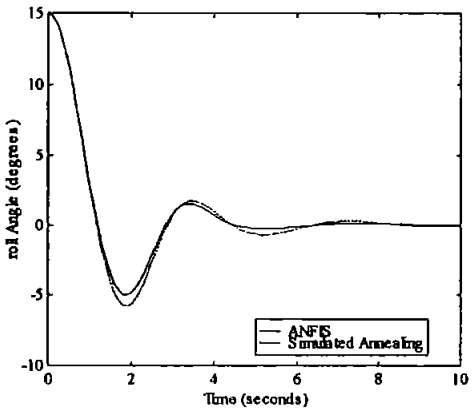
Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (15° initial position)



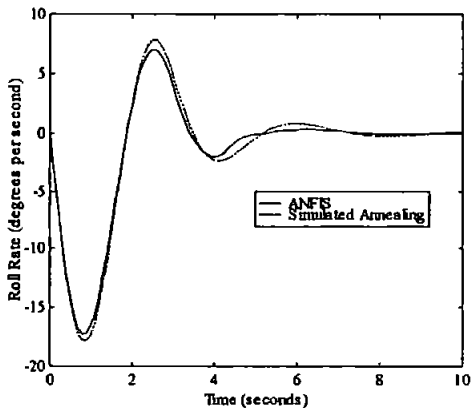
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (15° initial position)



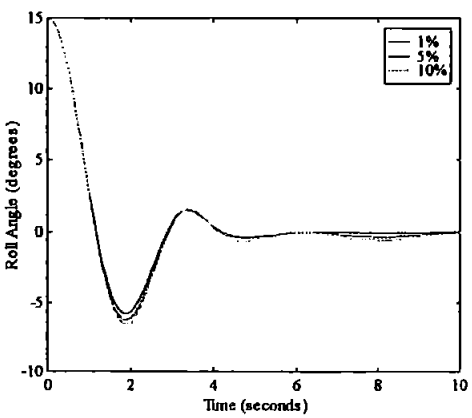
Roll Responses to Roll Rate Sensor Intermittent Total Failure (15° initial position)



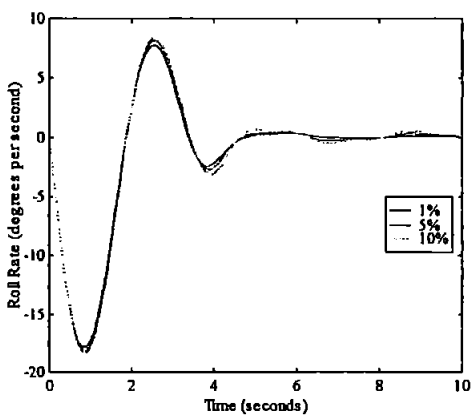
Roll Rate Responses to Roll Rate Sensor Intermittent Total Failure (15° initial position)



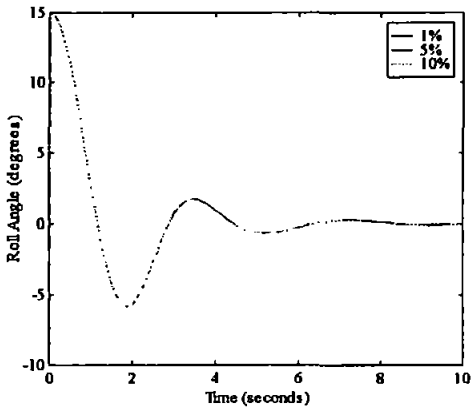
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



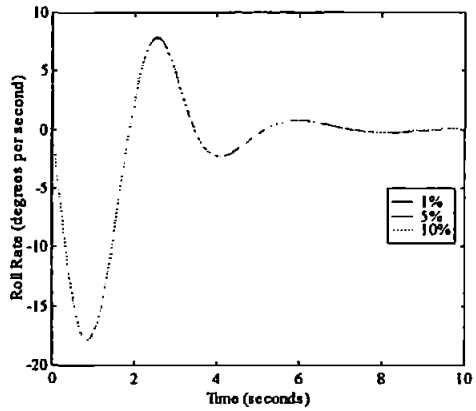
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



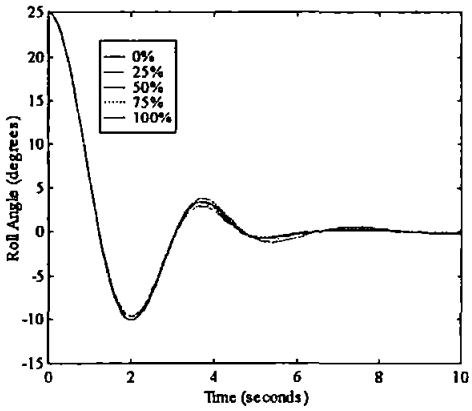
Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



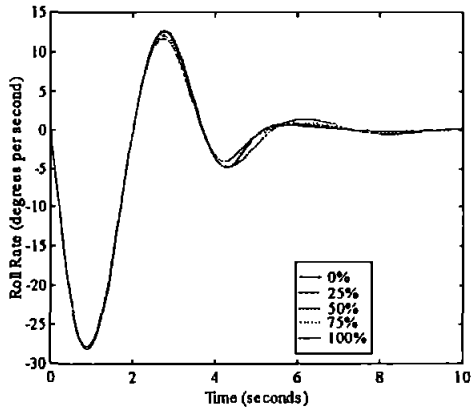
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (15° initial position)



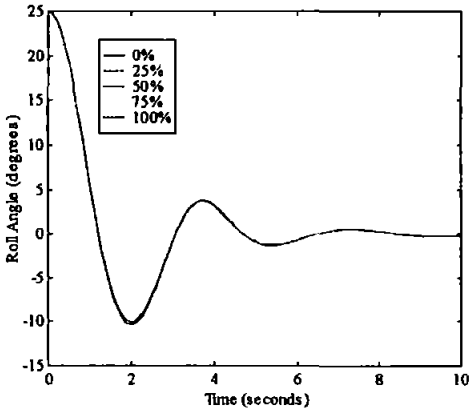
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (25° initial position)



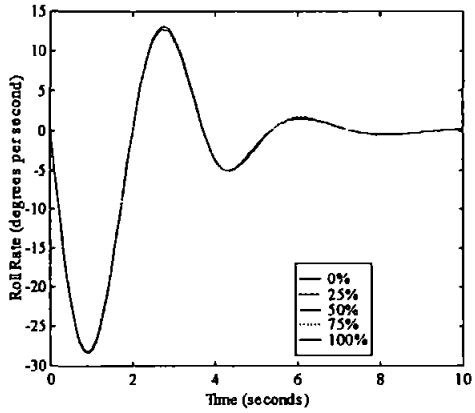
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (25° initial position)



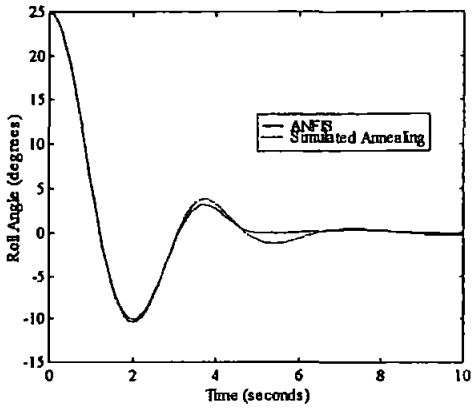
Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Percentage Faults (25° initial position)



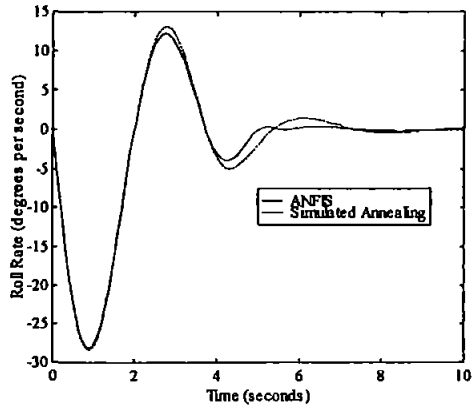
Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Percentage Faults (25° initial position)



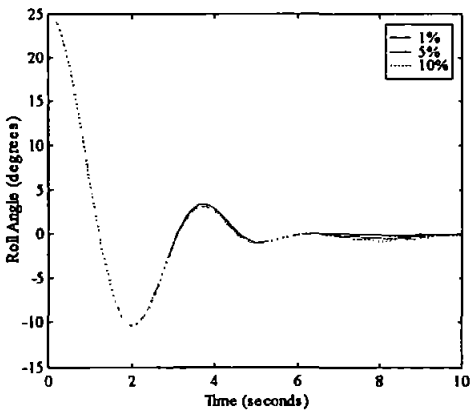
Roll Responses to Roll Rate Sensor Intermittent Total Failure (25° initial position)



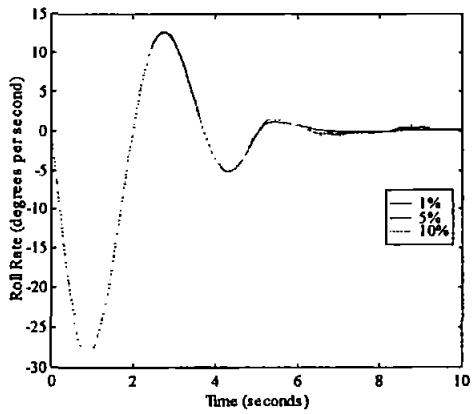
Roll Rate Responses to Roll Rate Sensor Intermittent Total Failure (25° initial position)



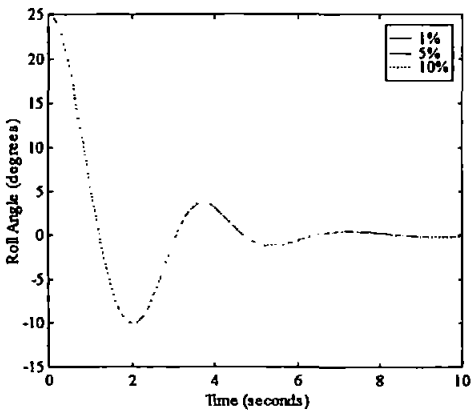
ANFIS Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (25° initial position)



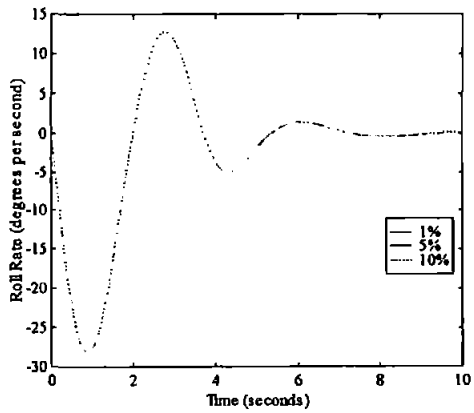
ANFIS Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (25° initial position)



Simulated Annealing Tuned FIS Roll Responses to Roll Rate Sensor Signal to Noise Ratios (25° initial position)

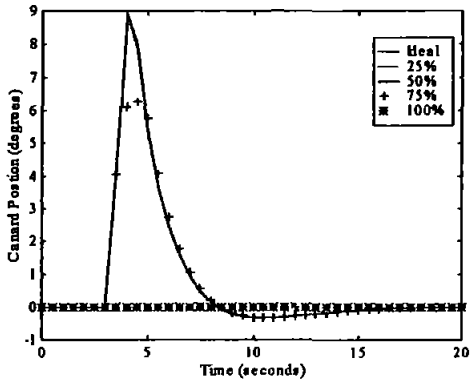


Simulated Annealing Tuned FIS Roll Rate Responses to Roll Rate Sensor Signal to Noise Ratios (25° initial position)

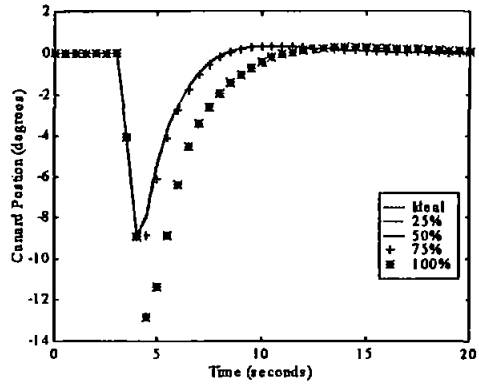


Appendix I Actuator Recovery FISs Canard Responses

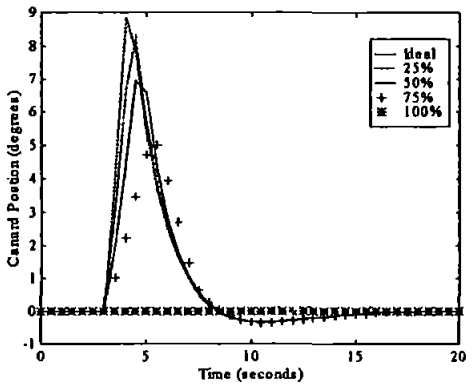
Upper Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (10° demand)



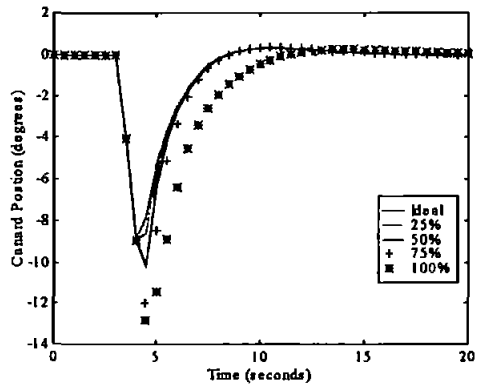
Lower Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (10° demand)



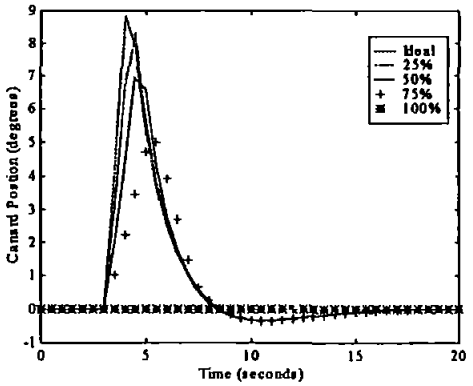
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (10° demand)



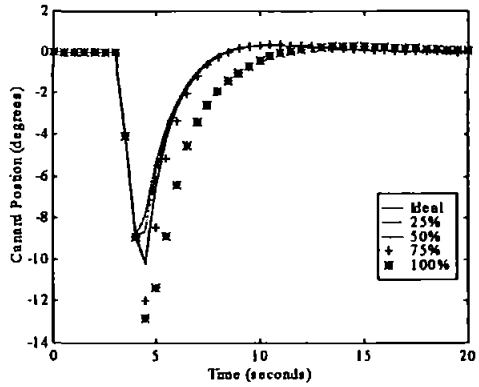
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (10° demand)



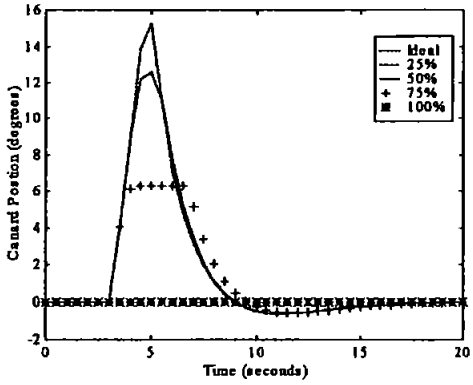
Upper Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (10° demand)



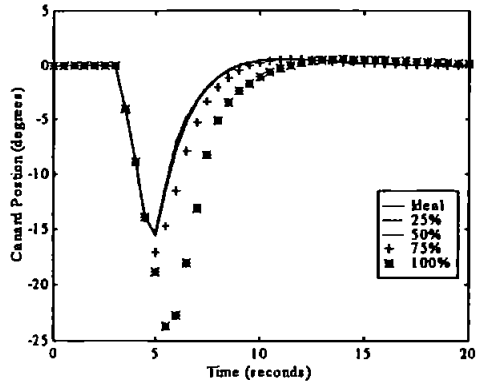
Lower Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (10° demand)



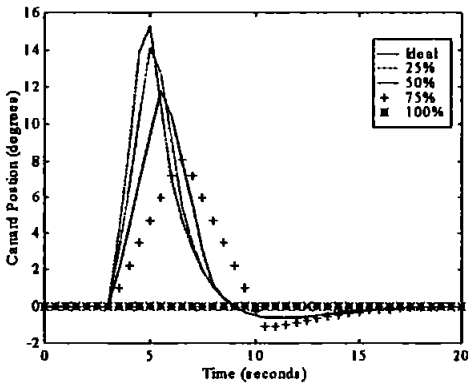
Upper Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (20° demand)



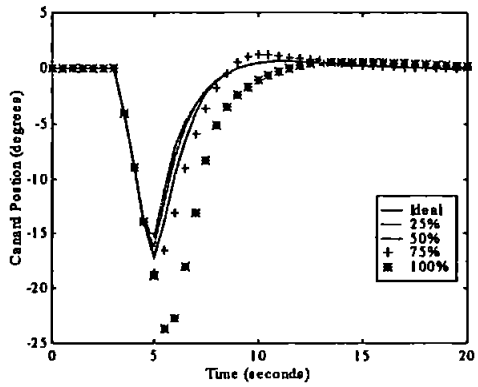
Lower Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (20° demand)



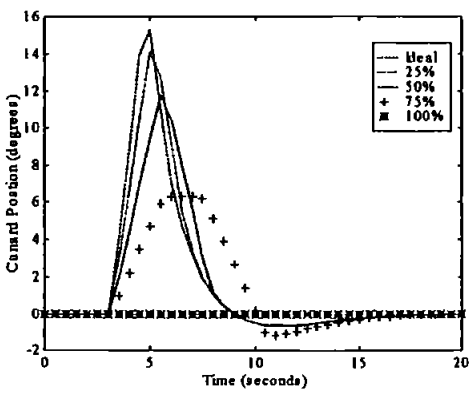
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (20° demand)



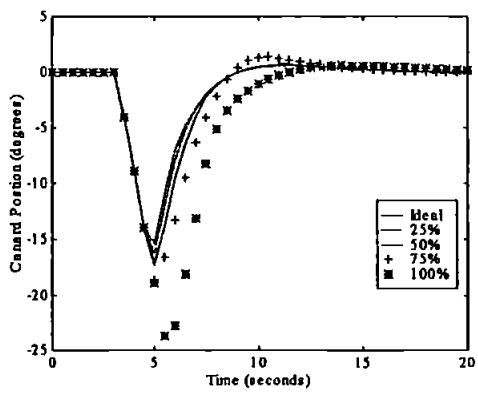
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (20° demand)



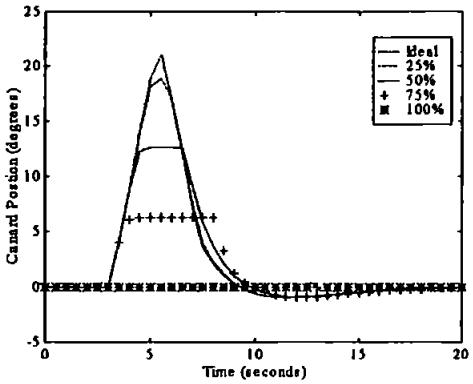
Upper Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (20° demand)



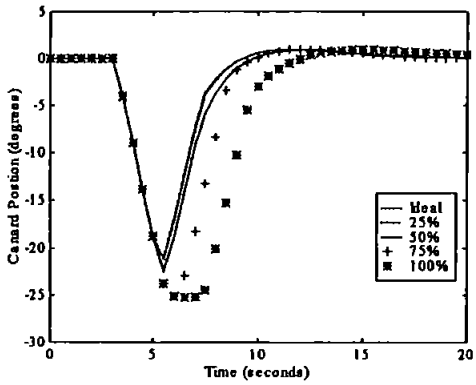
Lower Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (20° demand)



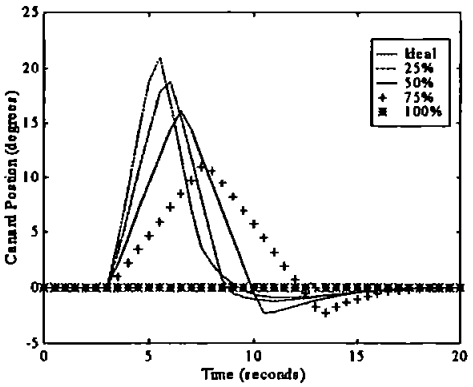
Upper Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (30° demand)



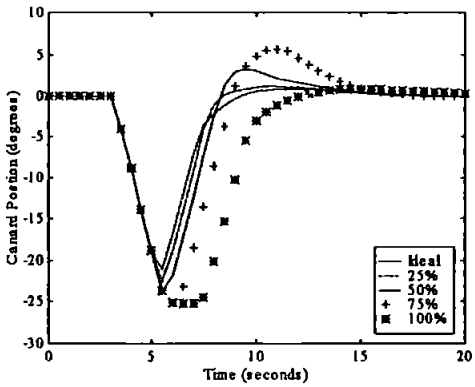
Lower Canard Responses to Actuator LOE in Saturation Block Using Simulated Annealing Tuned FIS (30° demand)



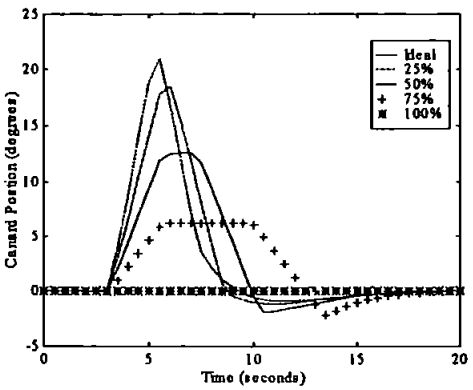
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (30° demand)



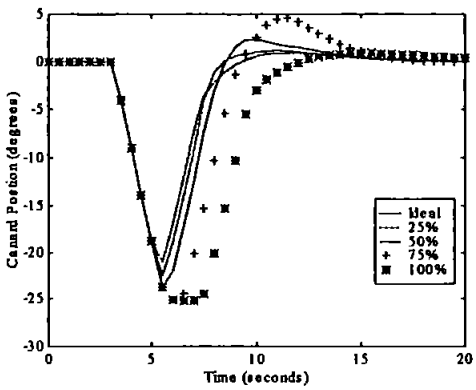
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Simulated Annealing Tuned FIS (30° demand)



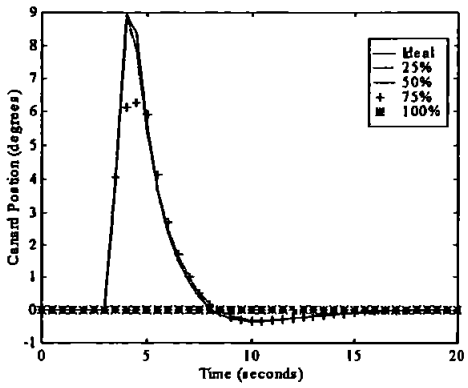
Upper Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (30° demand)



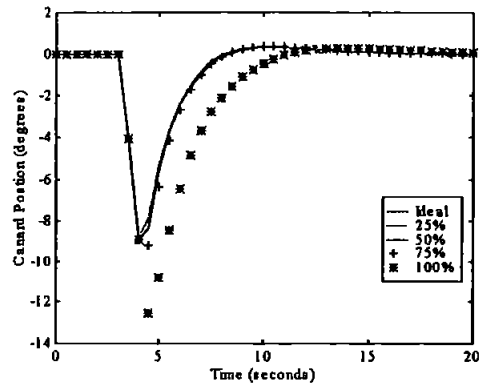
Lower Canard Responses to Actuator LOE in Both Blocks Using Simulated Annealing Tuned FIS (30° demand)



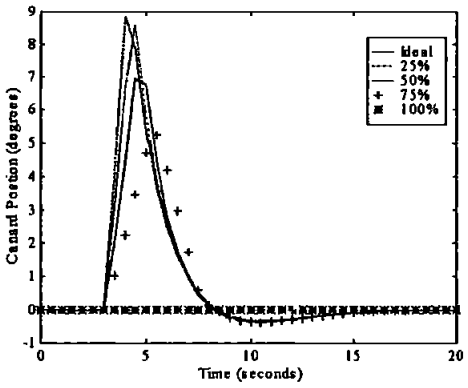
Upper Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (10° demand)



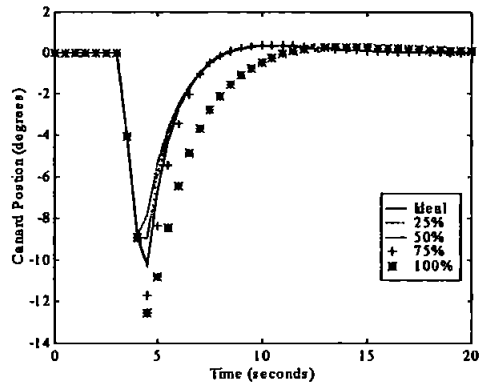
Lower Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (10° demand)



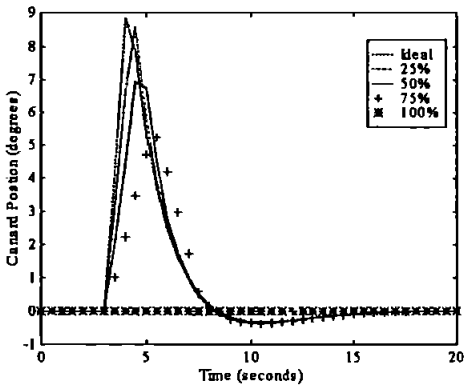
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (10° demand)



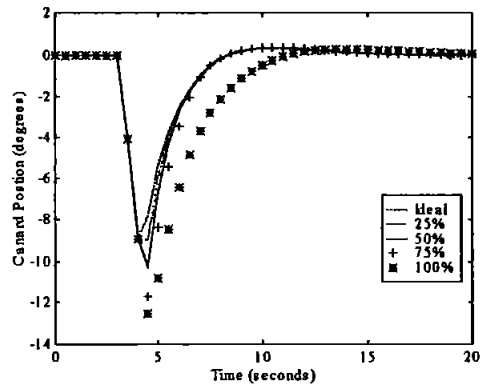
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (10° demand)



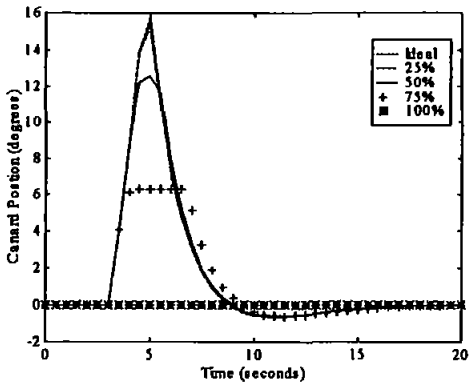
Upper Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (10° demand)



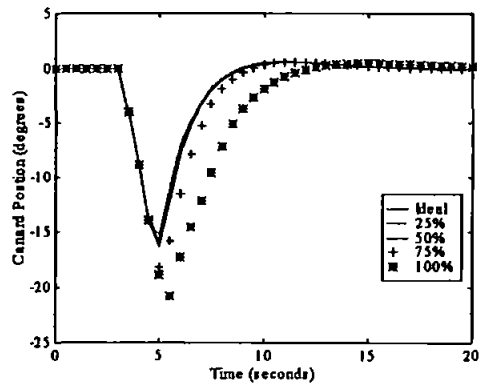
Lower Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (10° demand)



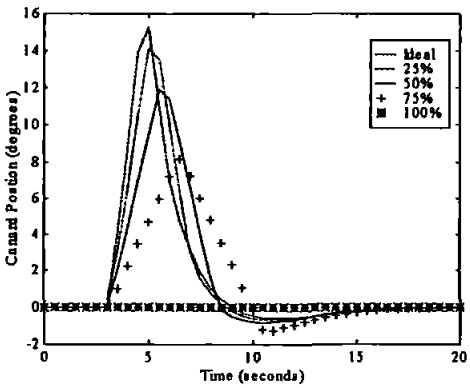
Upper Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (20° demand)



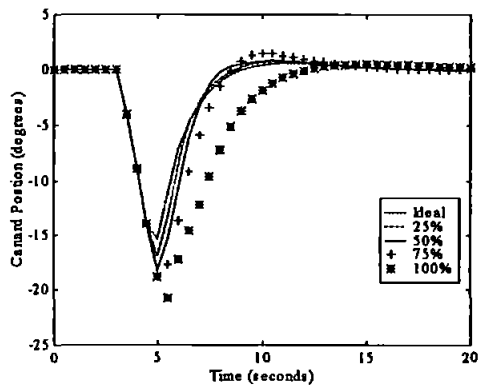
Lower Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (20° demand)



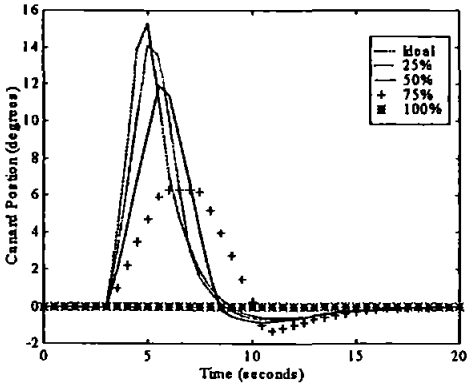
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (20° demand)



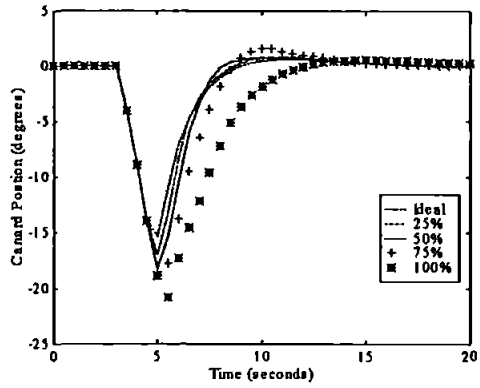
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (20° demand)



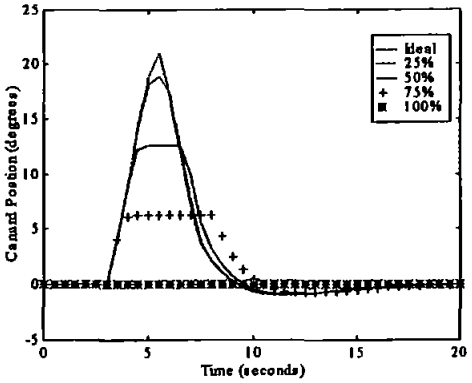
Upper Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (20° demand)



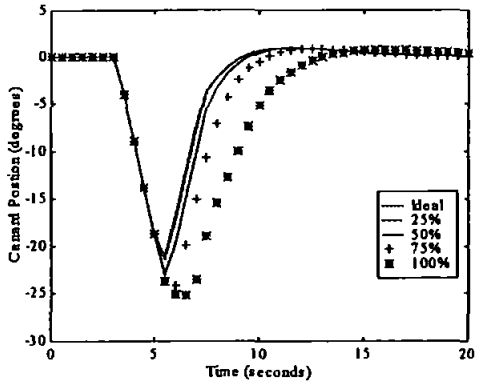
Lower Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (20° demand)



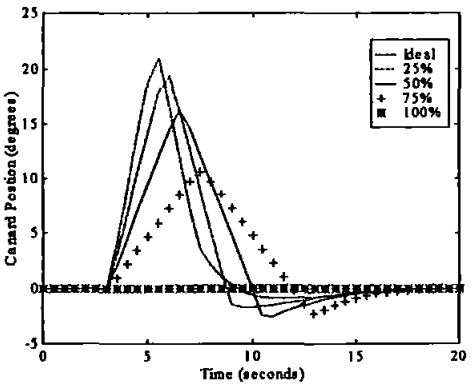
Upper Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (30° demand)



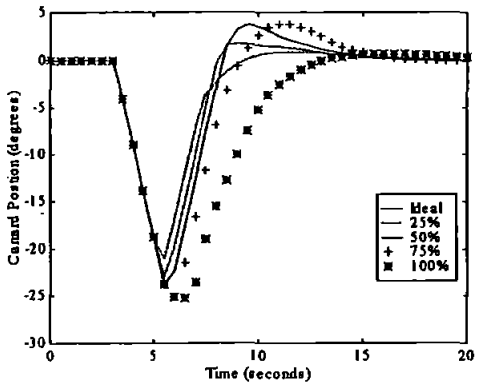
Lower Canard Responses to Actuator LOE in Saturation Block Using Tabu Tuned FIS (30° demand)



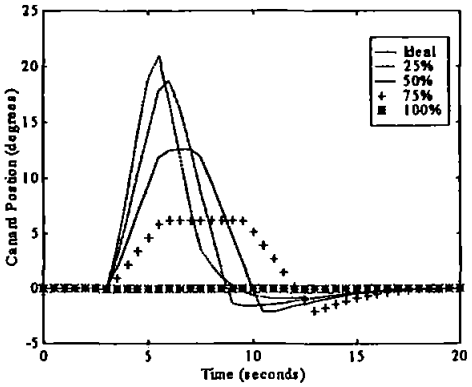
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (30° demand)



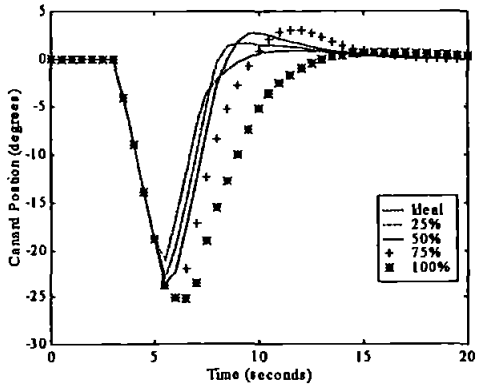
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using Tabu Tuned FIS (30° demand)



Upper Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (30° demand)

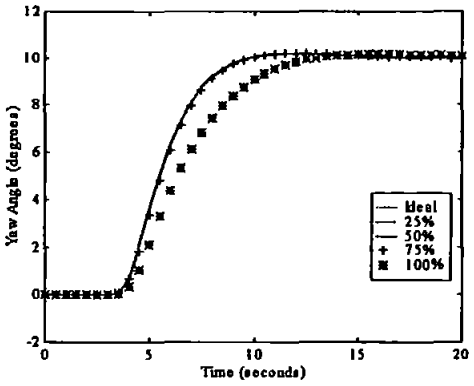


Lower Canard Responses to Actuator LOE in Both Blocks Using Tabu Tuned FIS (30° demand)

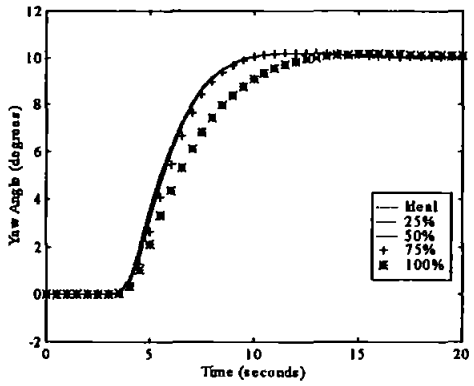


Appendix J Actuator Recovery Without Error Sensor Results

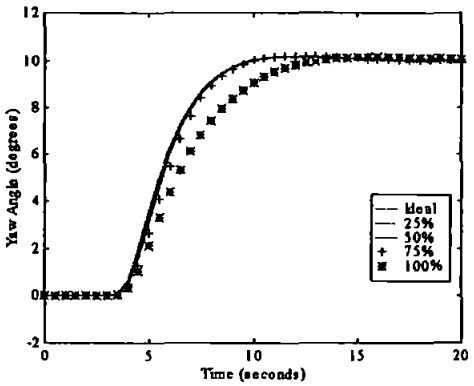
3 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



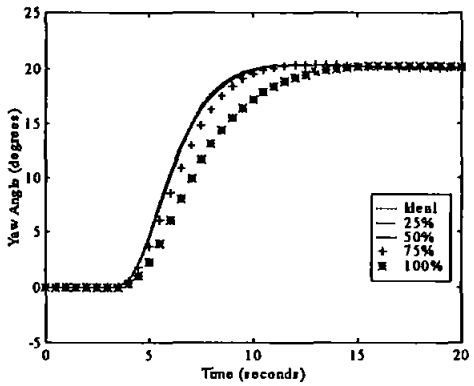
3 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



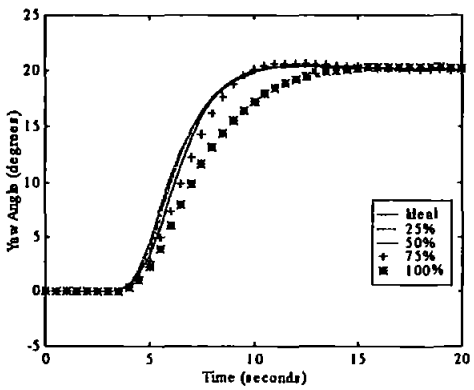
3 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



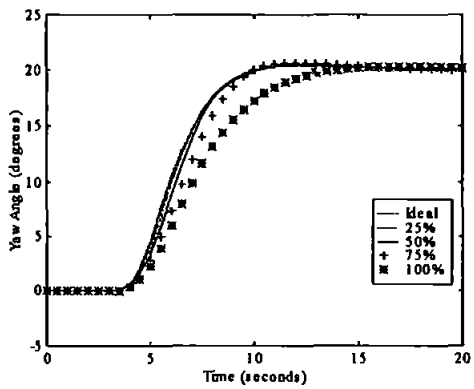
3 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



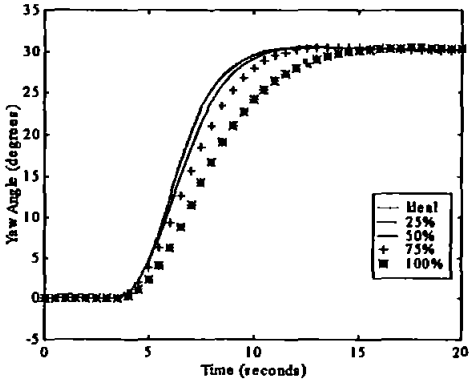
3 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



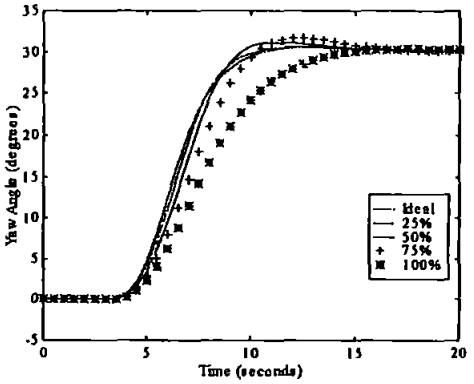
3 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



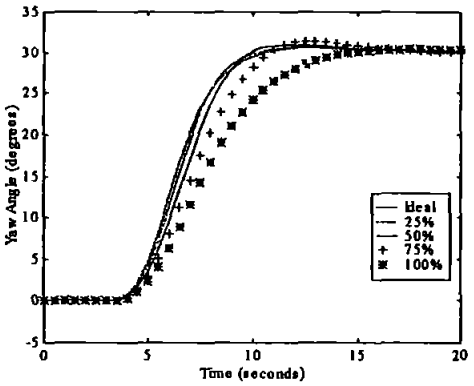
3 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



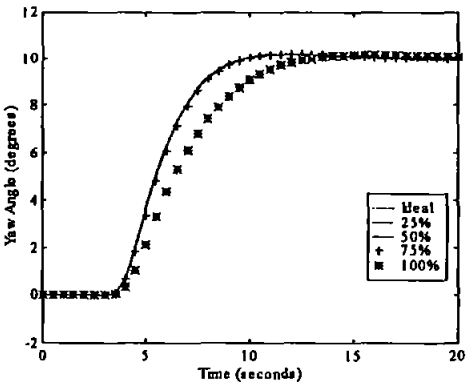
3 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



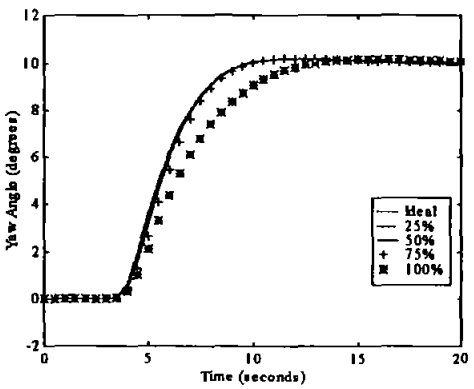
3 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



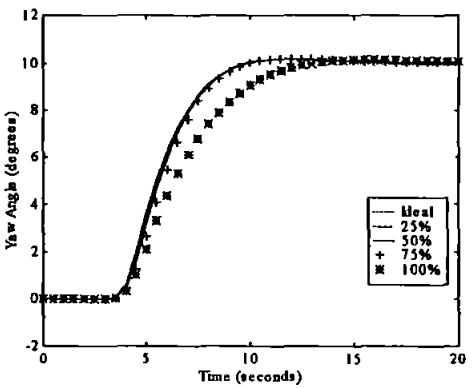
5 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



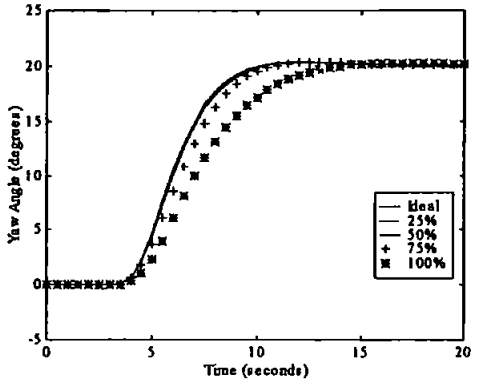
5 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



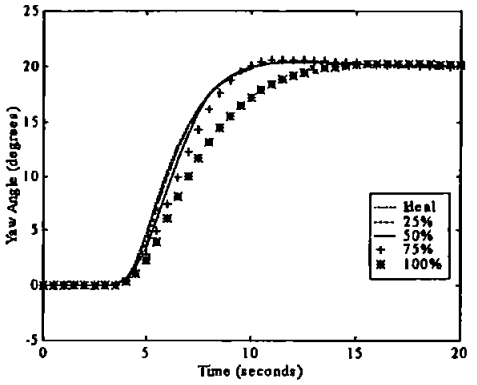
5 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



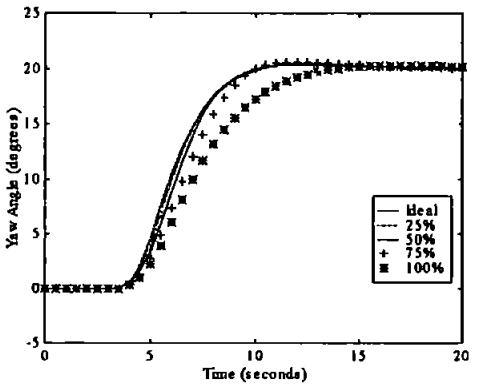
5 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



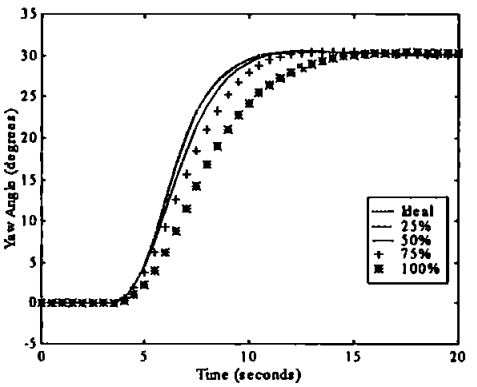
5 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



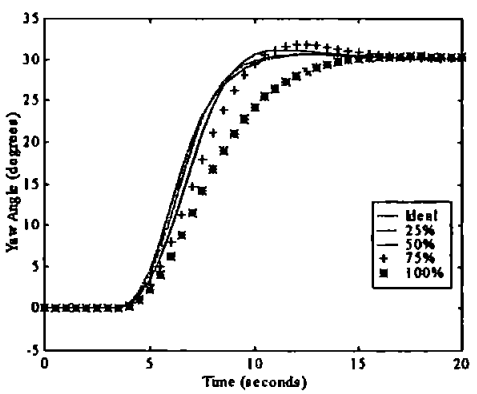
5 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



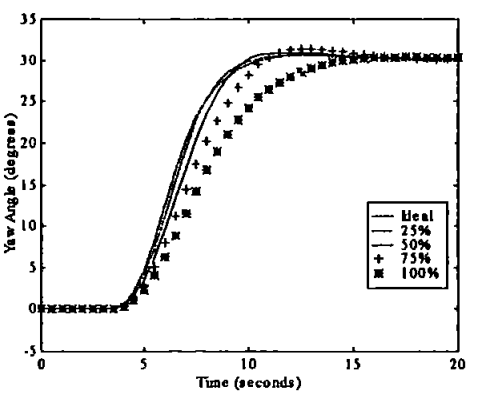
5 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



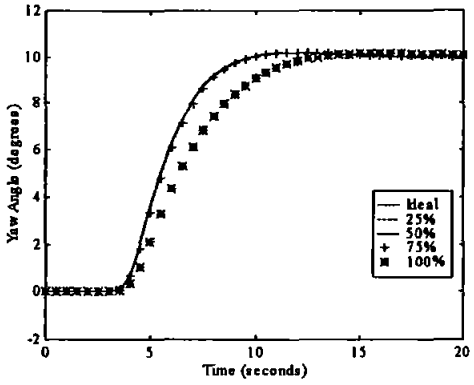
5 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



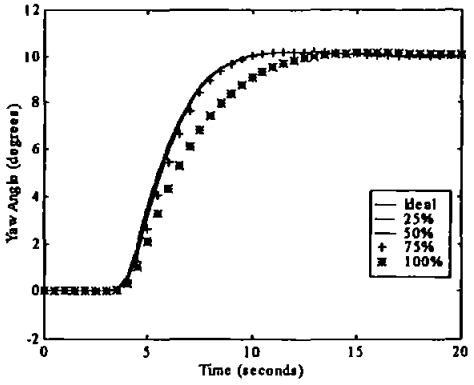
5 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



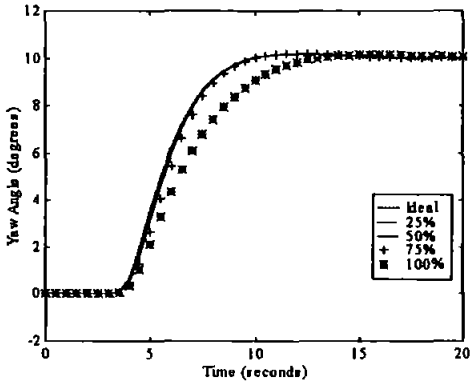
7 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



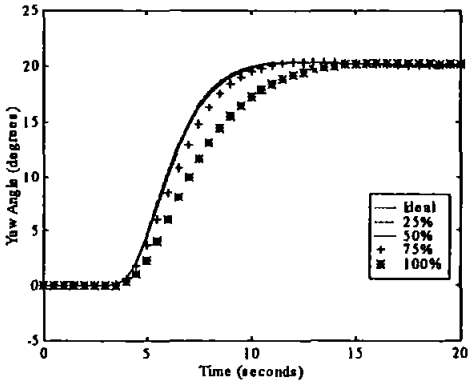
7 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



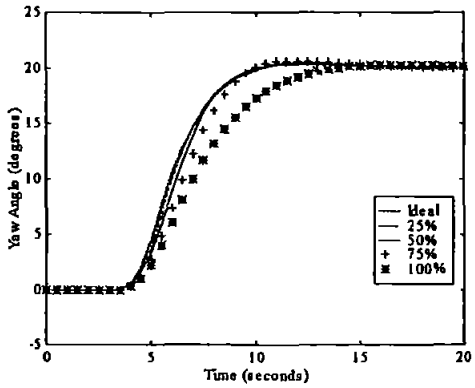
7 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



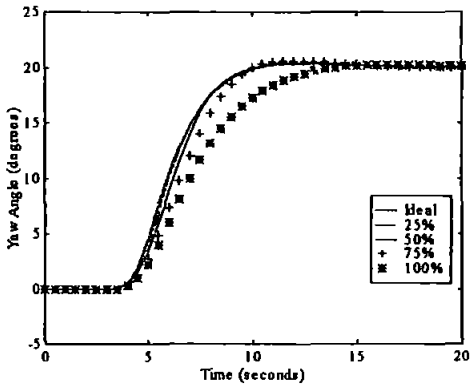
7 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



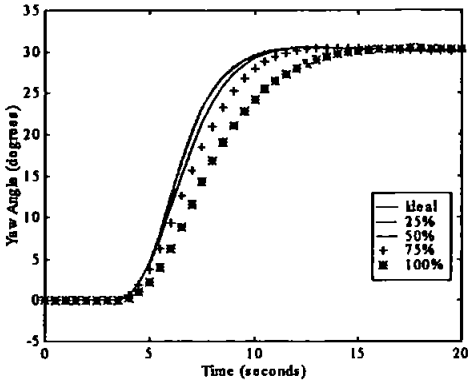
7 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



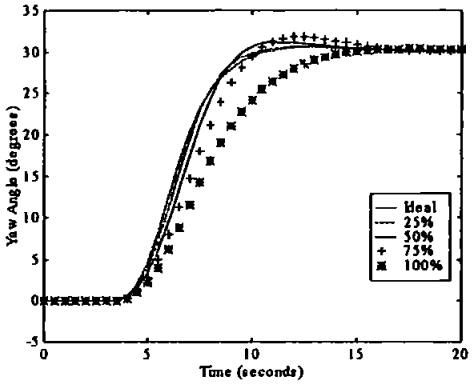
7 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



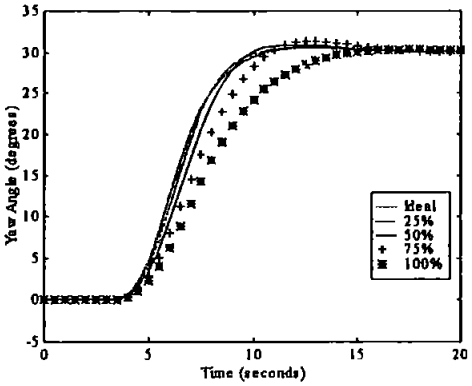
7 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



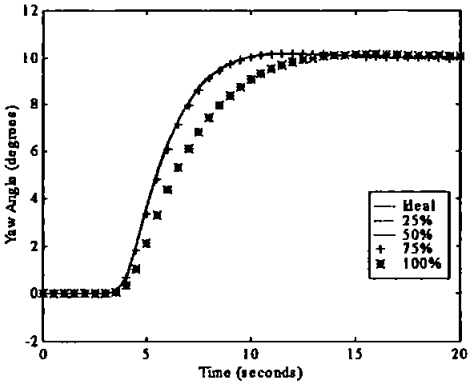
7 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



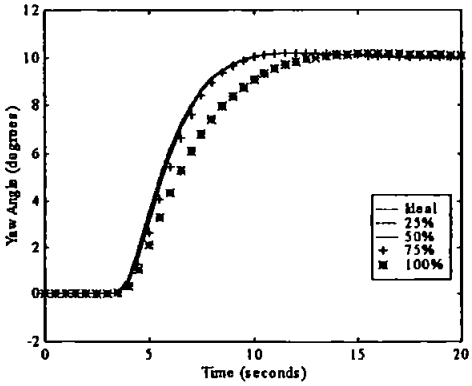
7 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



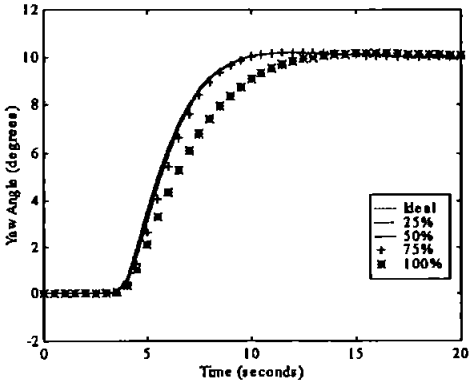
9 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



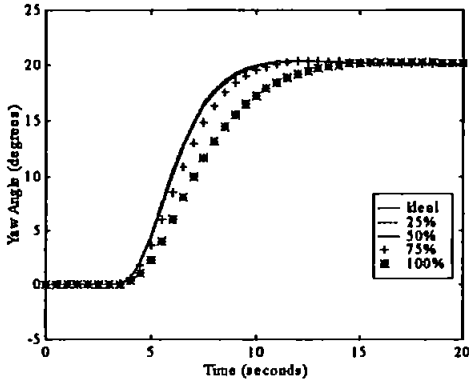
9 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



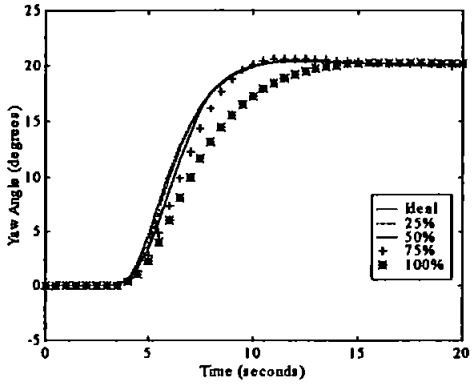
9 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



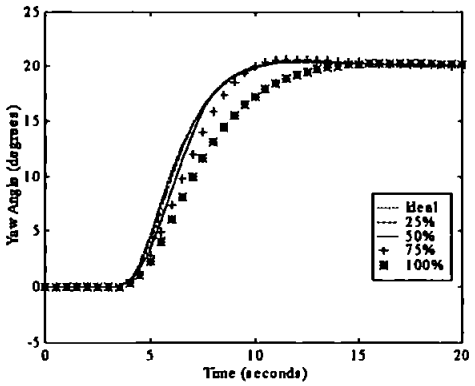
9 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



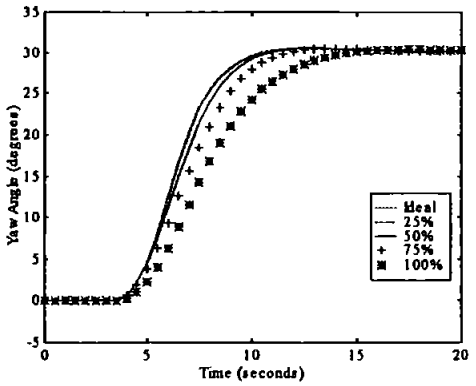
9 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults



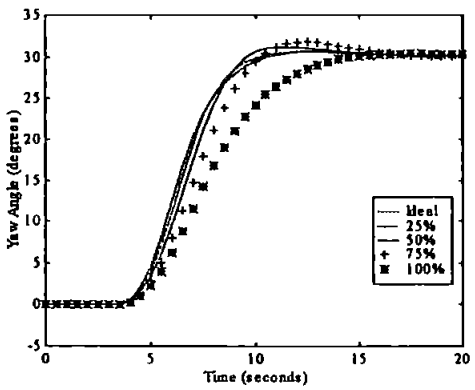
9 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults



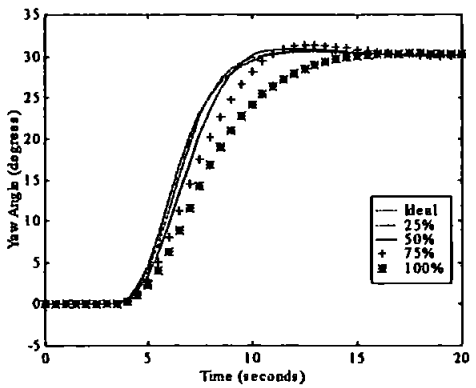
9 Rule FIS Controller Yaw Angle Responses for Saturation Block Actuator Percentage LOE Faults



9 Rule FIS Controller Yaw Angle Responses for Rate Limiter Block Actuator Percentage LOE Faults

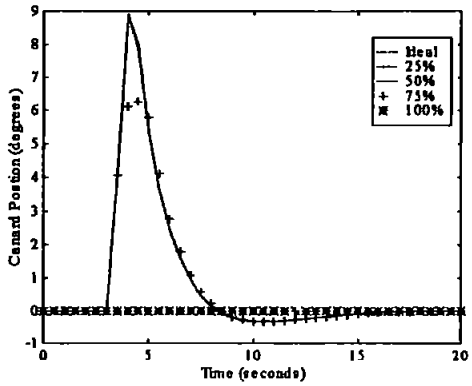


9 Rule FIS Controller Yaw Angle Responses for Both Block Actuator Percentage LOE Faults

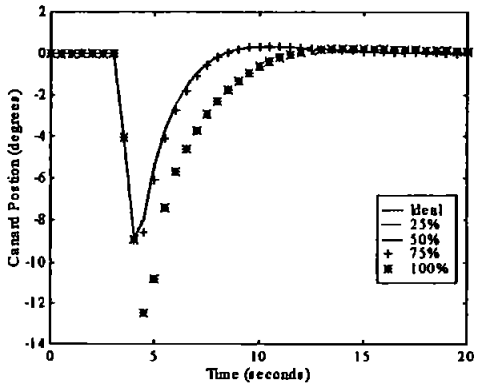


Appendix K Sensorless Actuator Recovery FISs Canard Responses

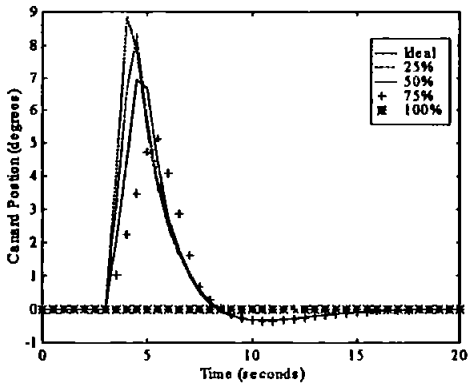
Upper Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (10° demand)



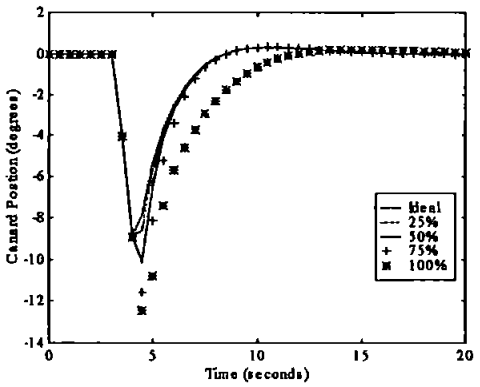
Lower Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (10° demand)



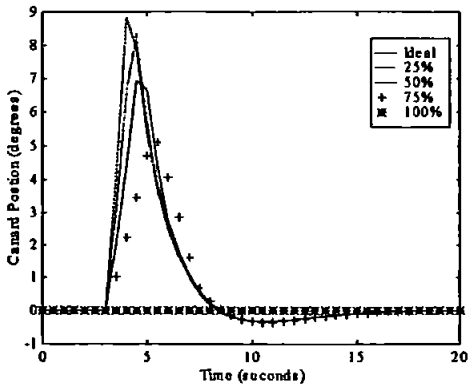
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (10° demand)



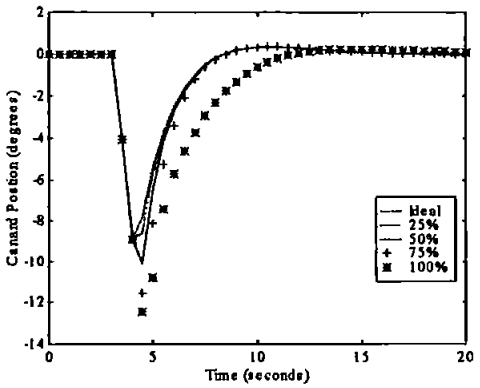
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (10° demand)



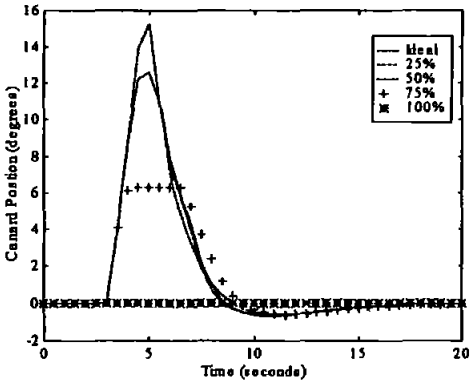
Upper Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (10° demand)



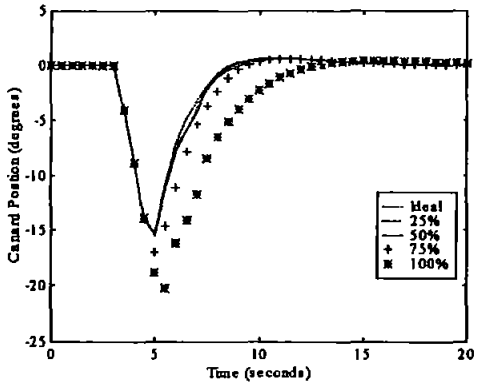
Lower Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (10° demand)



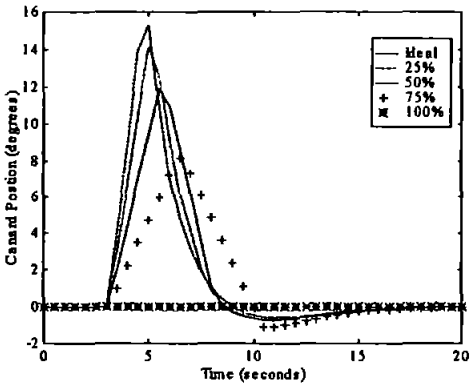
Upper Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (20° demand)



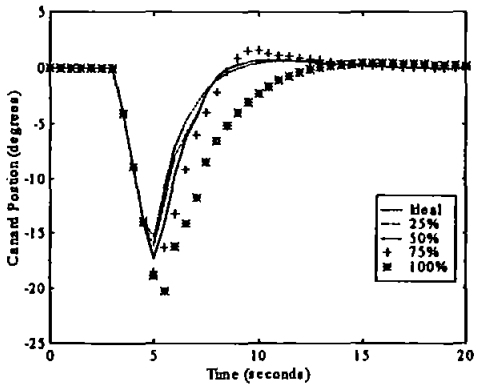
Lower Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (20° demand)



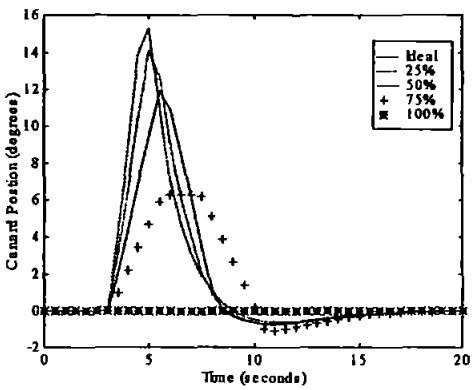
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (20° demand)



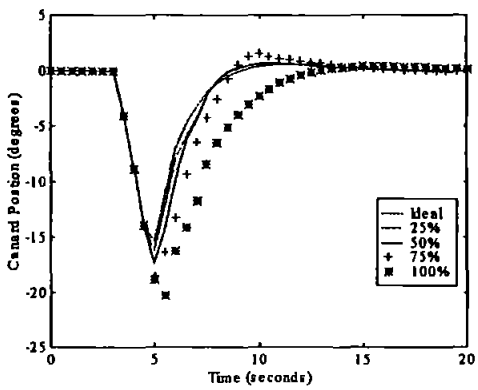
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (20° demand)



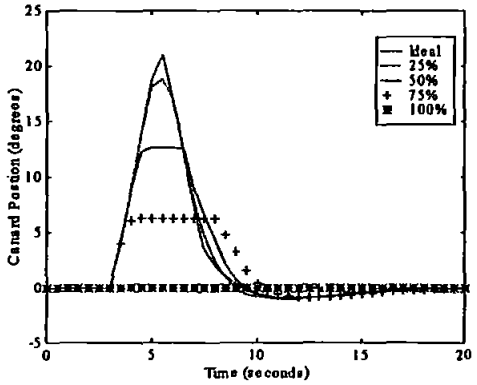
Upper Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (20° demand)



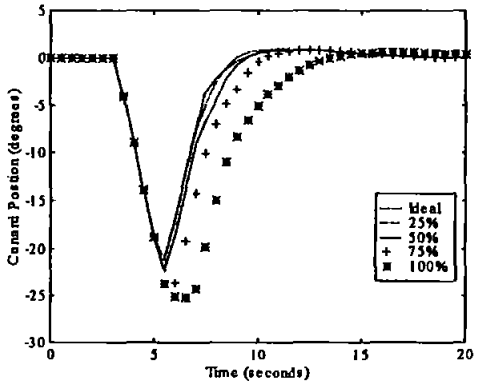
Lower Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (20° demand)



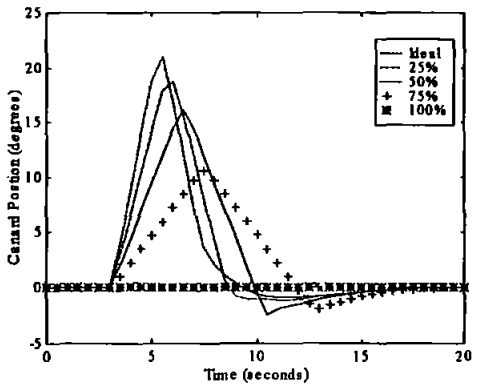
Upper Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (30° demand)



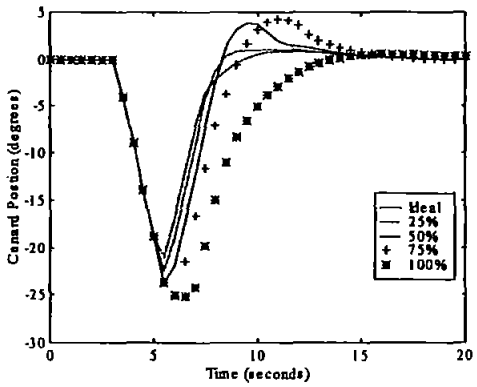
Lower Canard Responses to Actuator LOE in Saturation Block Using 3 Rule Estimator FIS (30° demand)



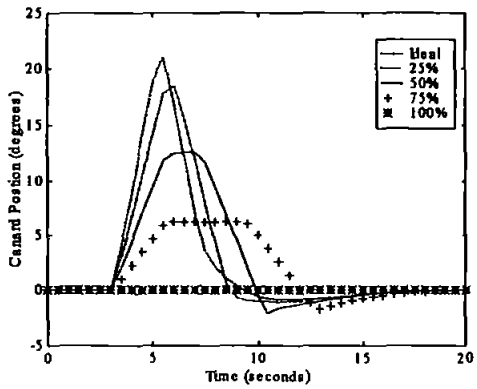
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (30° demand)



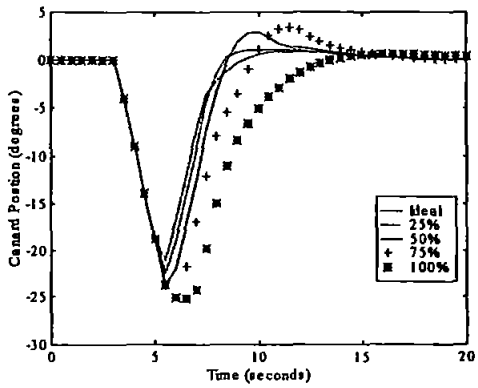
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 3 Rule Estimator FIS (30° demand)



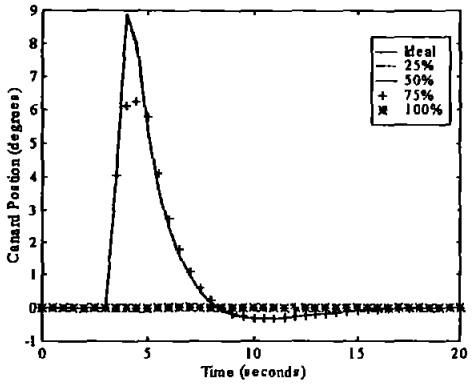
Upper Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (30° demand)



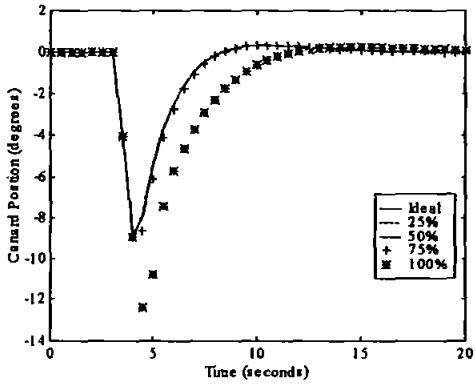
Lower Canard Responses to Actuator LOE in Both Blocks Using 3 Rule Estimator FIS (30° demand)



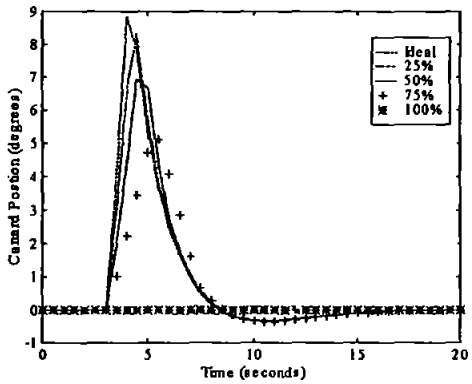
Upper Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (10° demand)



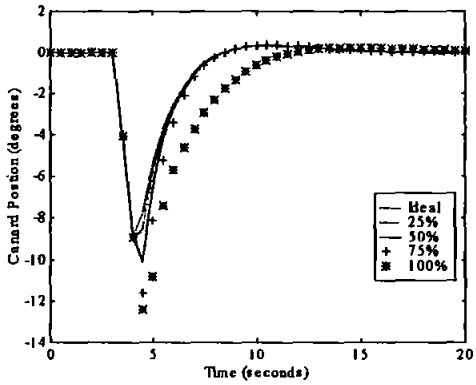
Lower Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (10° demand)



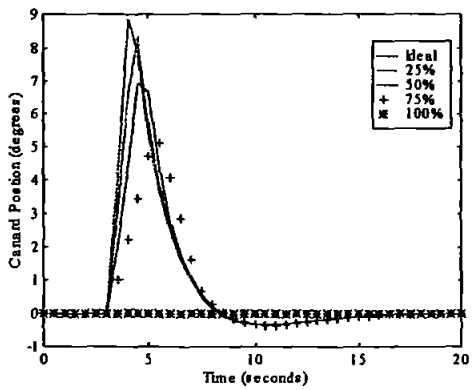
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (10° demand)



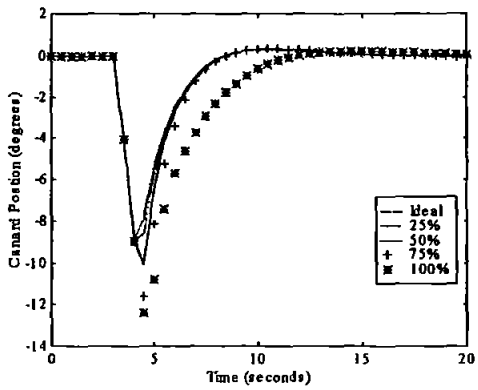
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (10° demand)



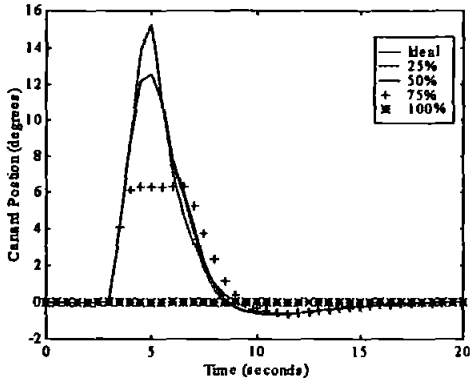
Upper Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (10° demand)



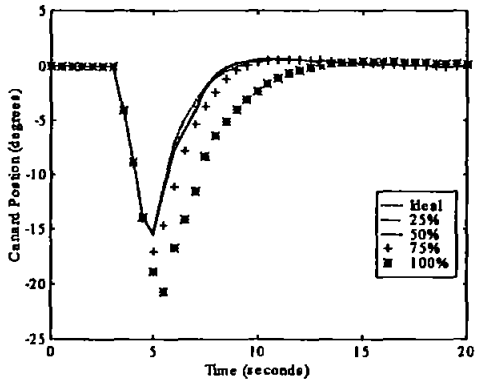
Lower Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (10° demand)



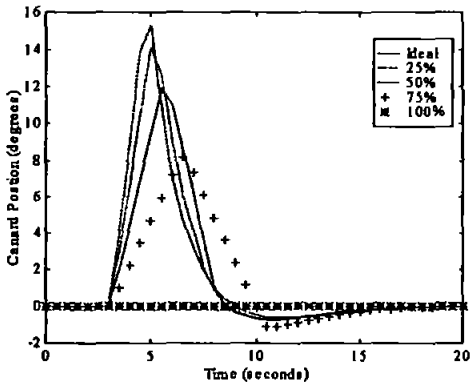
Upper Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (20° demand)



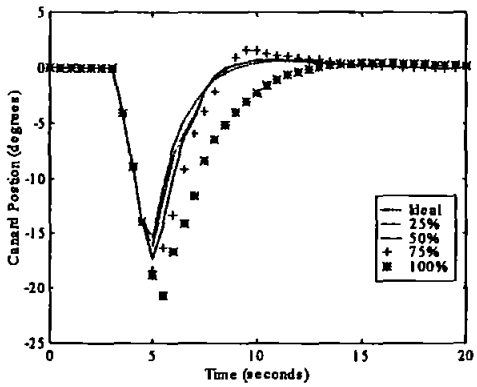
Lower Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (20° demand)



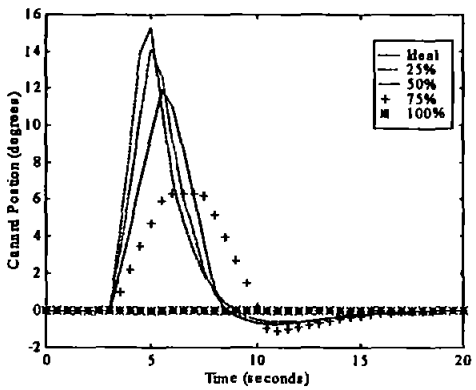
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (20° demand)



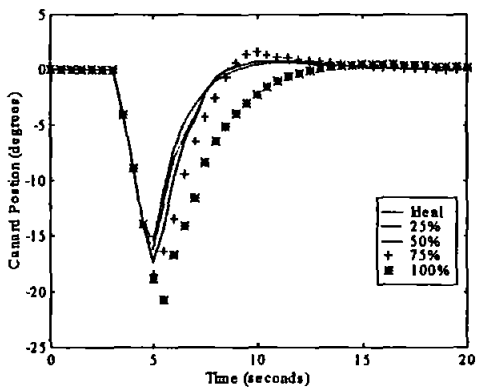
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (20° demand)



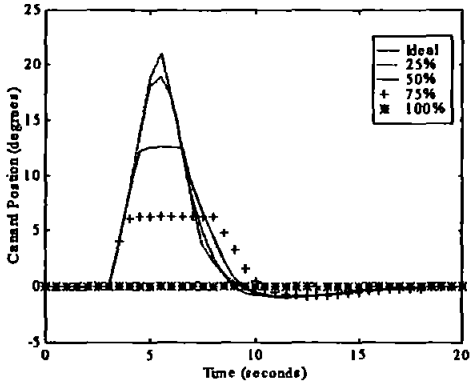
Upper Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (20° demand)



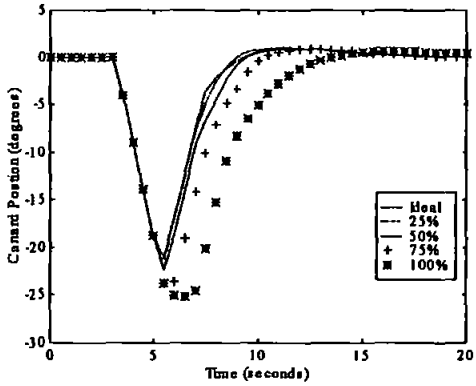
Lower Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (20° demand)



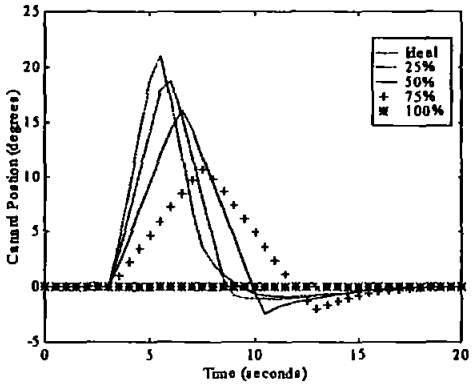
Upper Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (30° demand)



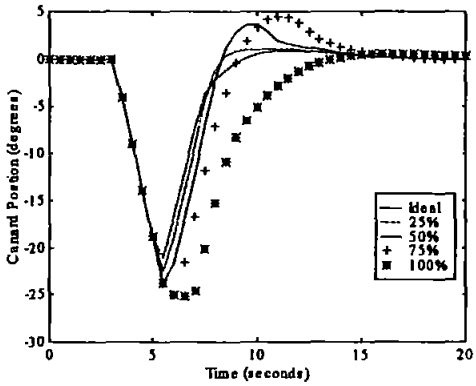
Lower Canard Responses to Actuator LOE in Saturation Block Using 5 Rule Estimator FIS (30° demand)



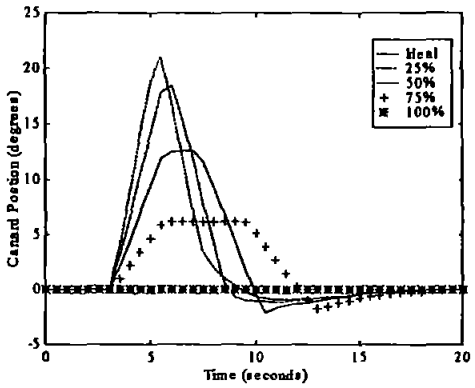
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (30° demand)



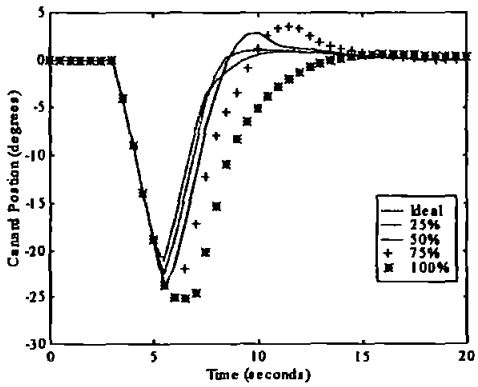
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 5 Rule Estimator FIS (30° demand)



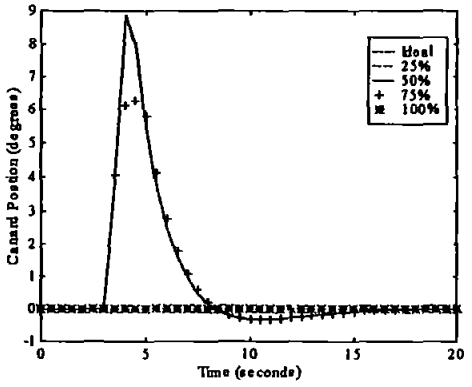
Upper Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (30° demand)



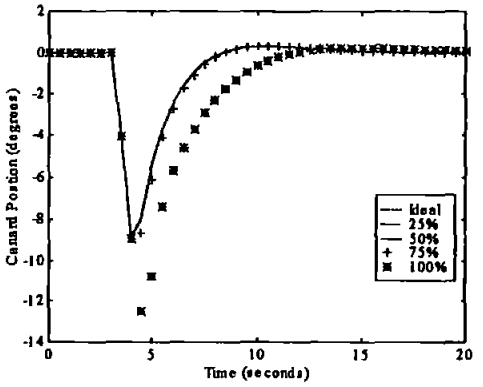
Lower Canard Responses to Actuator LOE in Both Blocks Using 5 Rule Estimator FIS (30° demand)



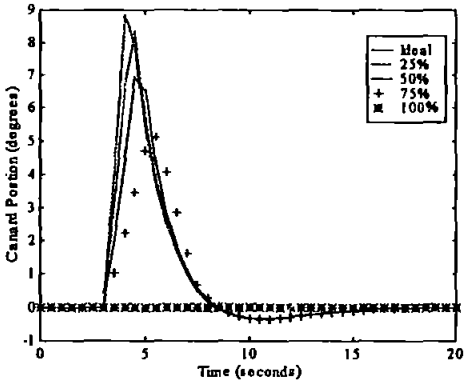
Upper Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (10° demand)



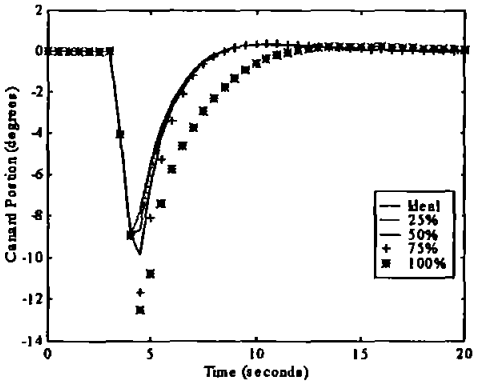
Lower Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (10° demand)



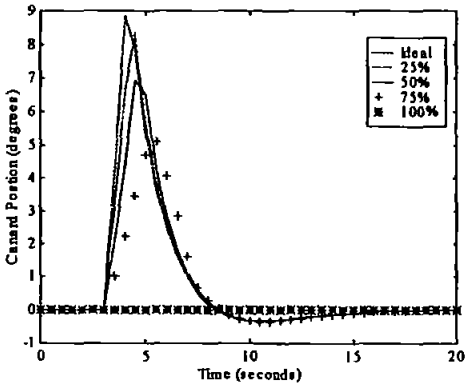
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (10° demand)



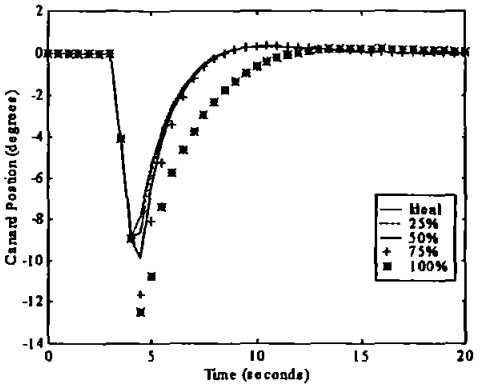
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (10° demand)



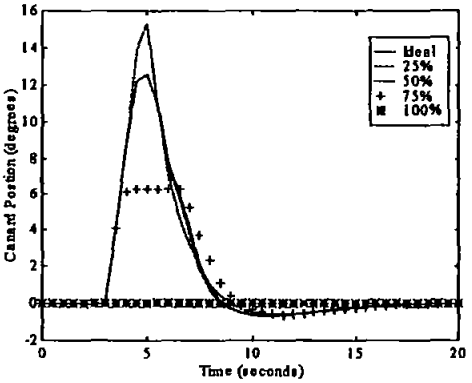
Upper Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (10° demand)



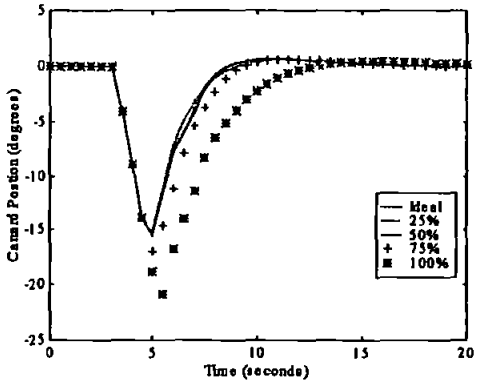
Lower Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (10° demand)



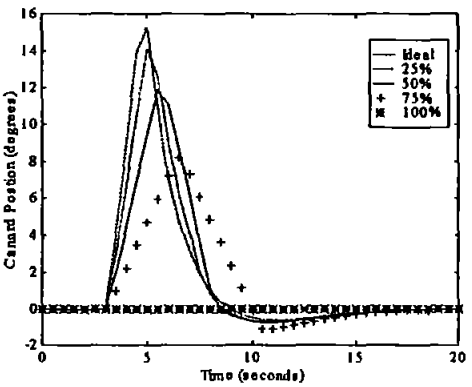
Upper Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (20° demand)



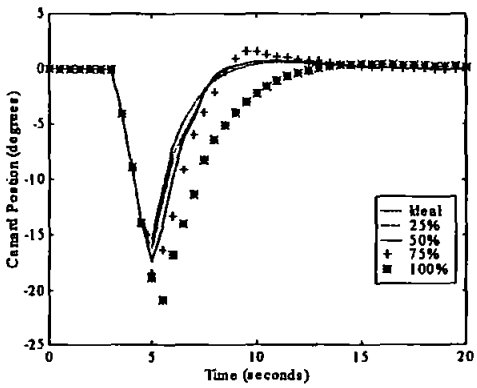
Lower Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (20° demand)



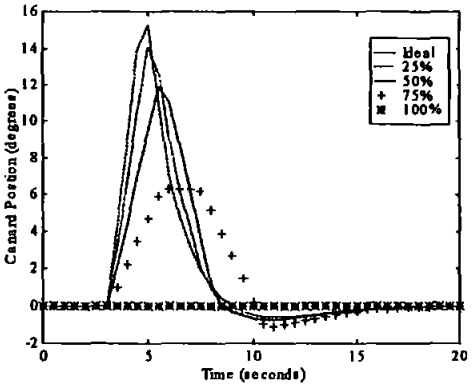
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (20° demand)



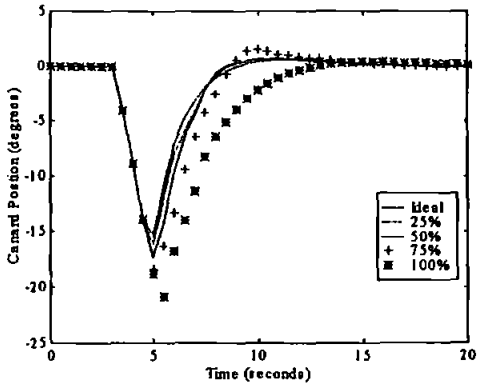
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (20° demand)



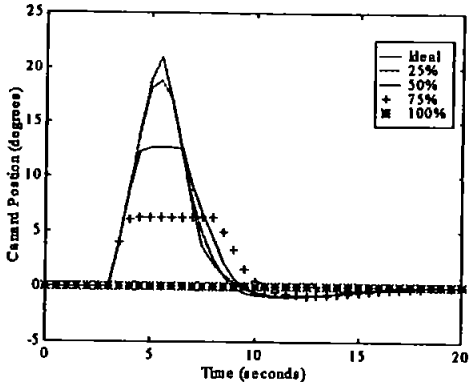
Upper Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (20° demand)



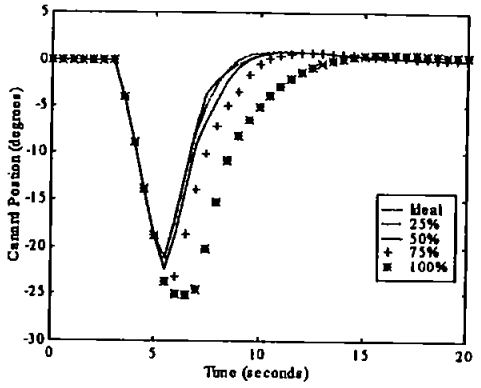
Lower Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (20° demand)



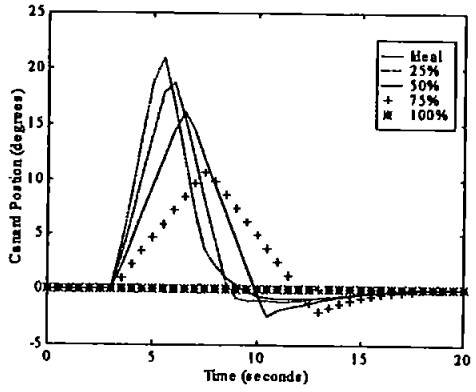
Upper Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (30° demand)



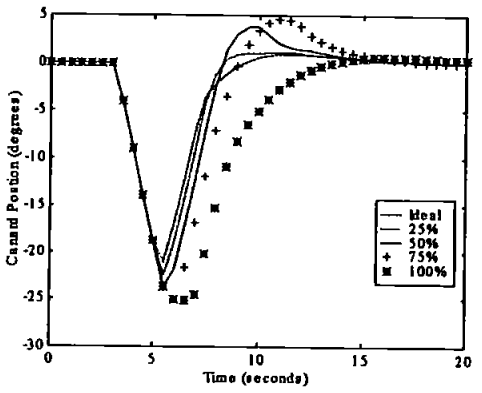
Lower Canard Responses to Actuator LOE in Saturation Block Using 7 Rule Estimator FIS (30° demand)



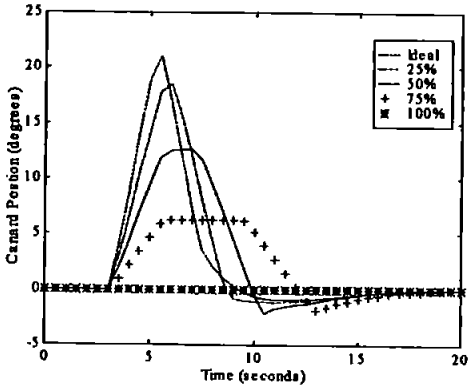
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (30° demand)



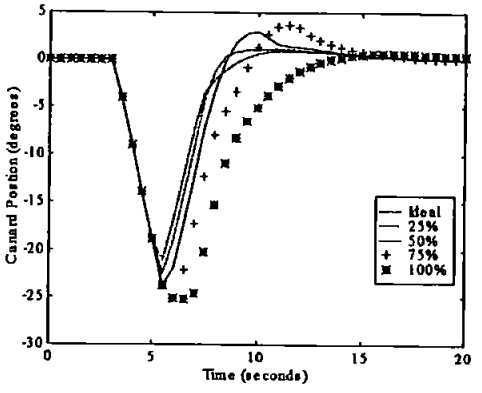
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 7 Rule Estimator FIS (30° demand)



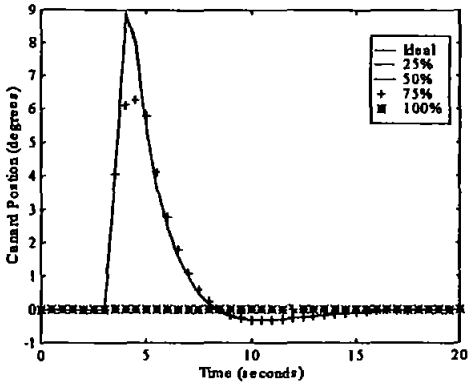
Upper Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (30° demand)



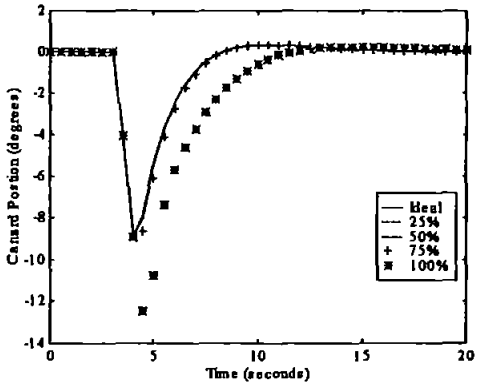
Lower Canard Responses to Actuator LOE in Both Blocks Using 7 Rule Estimator FIS (30° demand)



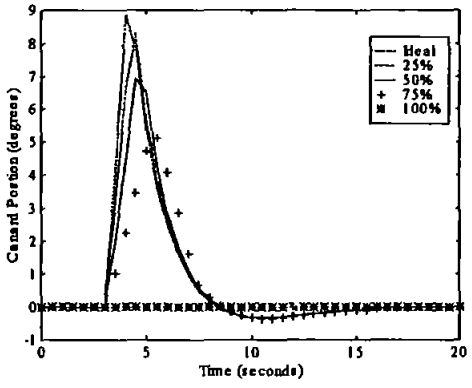
Upper Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (10° demand)



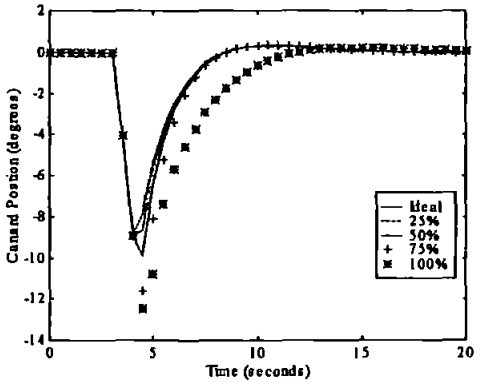
Lower Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (10° demand)



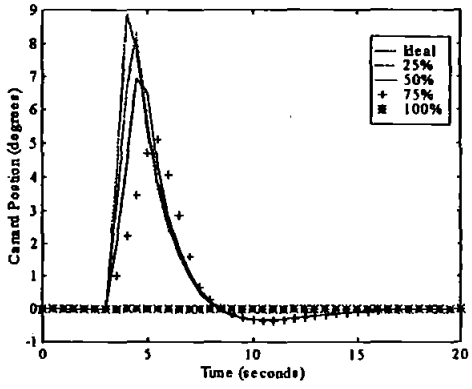
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (10° demand)



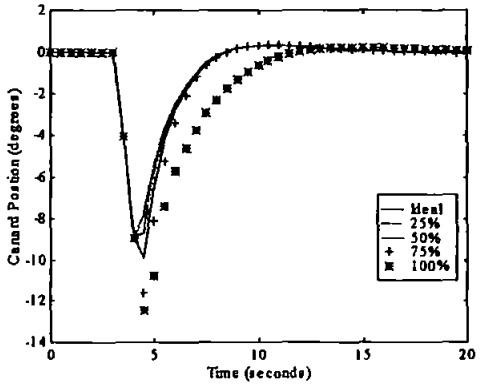
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (10° demand)



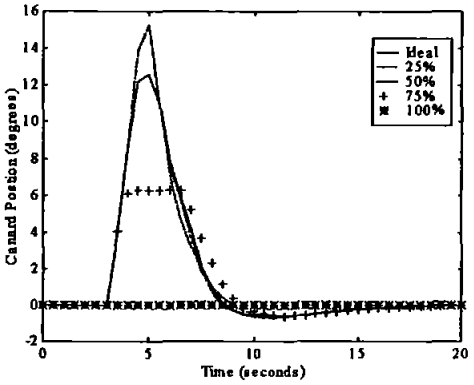
Upper Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (10° demand)



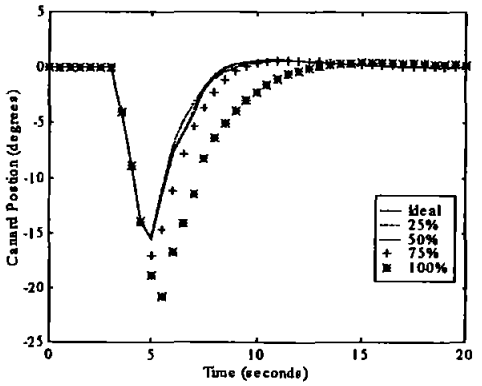
Lower Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (10° demand)



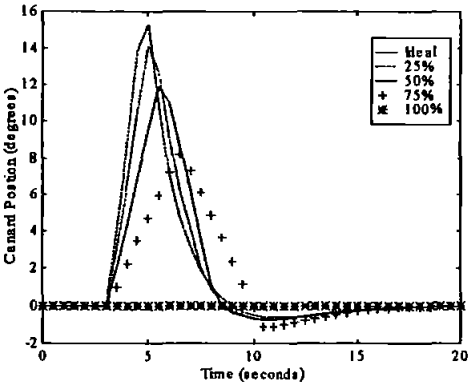
Upper Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (20° demand)



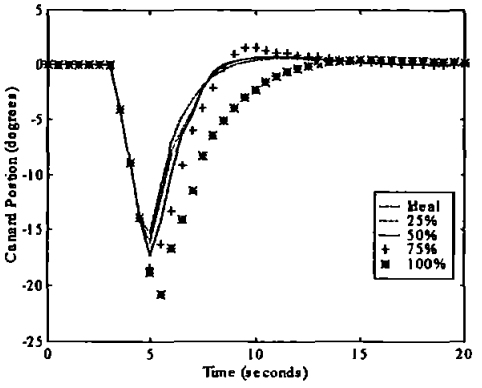
Lower Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (20° demand)



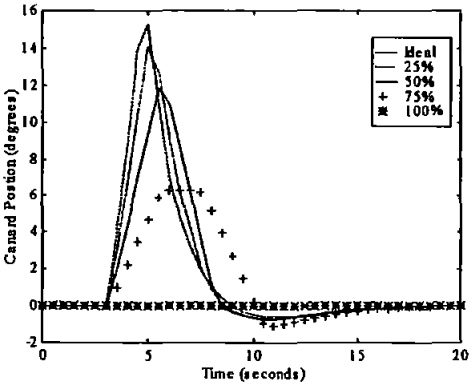
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (20° demand)



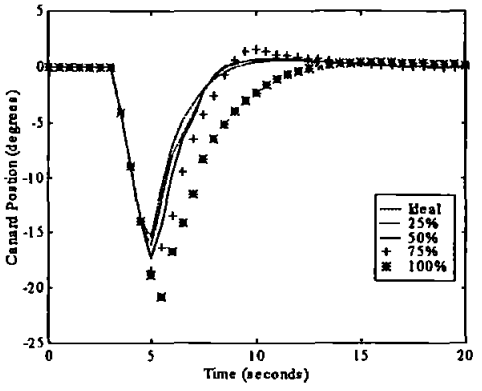
Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (20° demand)



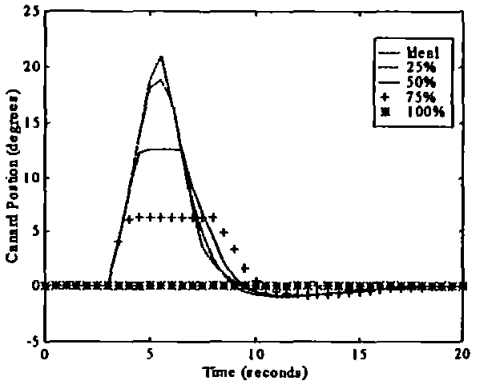
Upper Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (20° demand)



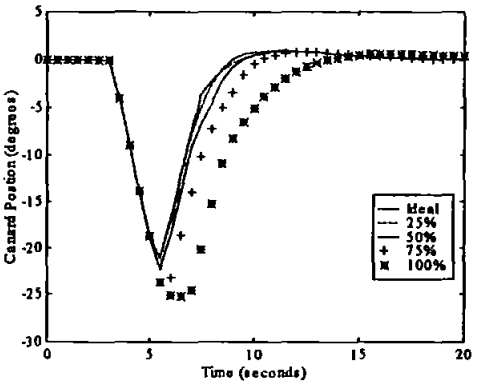
Lower Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (20° demand)



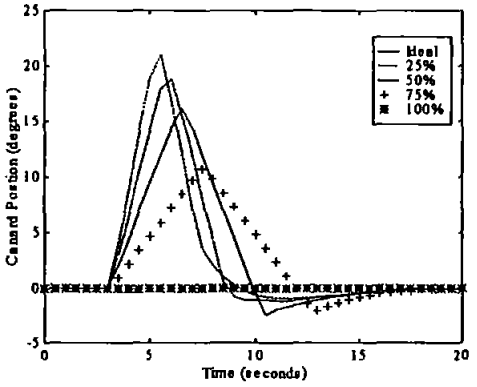
Upper Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (30° demand)



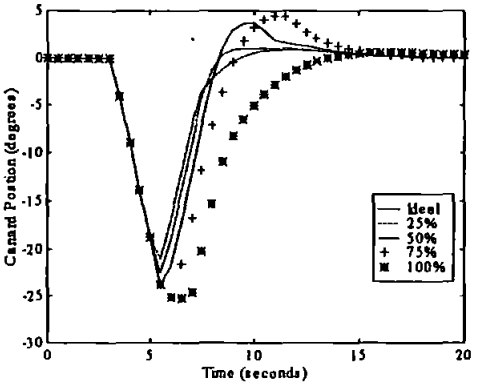
Lower Canard Responses to Actuator LOE in Saturation Block Using 9 Rule Estimator FIS (30° demand)



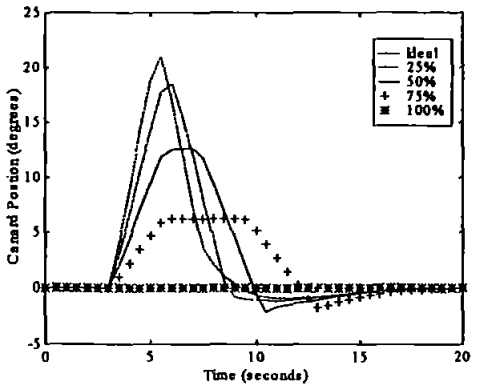
Upper Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (30° demand)



Lower Canard Responses to Actuator LOE in Rate Limiter Block Using 9 Rule Estimator FIS (30° demand)



Upper Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (30° demand)



Lower Canard Responses to Actuator LOE in Both Blocks Using 9 Rule Estimator FIS (30° demand)