Investigation into Fault Detection and Diagnosis Techniques

Prepared for: **Prof. M Braae**
Department of Electrical Engineering
University of Cape Town

Prepared by: **B Patel**
Department of Electrical Engineering
University of Cape Town

April 2000

This dissertation is prepared in fulfilment of the requirements for the Degree of Master of Science in Electrical Engineering at the University of Cape Town.

University of Cape Town

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

ACKNOWLEDGEMENTS

As with any complex project, it would be very insincere and vain for the author not to thank the many people without whom, this contribution would not be made possible. So, I would like to take this opportunity to thank Prof. Braae, for the invaluable help on the academic side of my thesis, and for his persistent motivation. Many go to all of my colleagues: Angus, Austin, Bernard, Craig, Garron, Riaan and Rory, for their support and help. Further thanks are given to my family and friends for their motivation and support, without which I would definitely have not succeeded so far. Thanks especially to my friend Nilan, for going the extra mile and being there for me at the end. On this note, special thanks are given to Indira for her inspiration, uncompromising support, and understanding. Finally, I would like to give thanks to DEBID for their financial assistance.

SYNOPSIS

The early detection and diagnosis of faults in systems is one of the most important tasks assigned to the computers supervising these systems. The early indication of abrupt or incipient failures can assist in avoiding major system breakdowns and catastrophes, which could otherwise result in substantial material damage and even human fatalities. Furthermore, these systems should be designed with the properties that any good fault detection and diagnosis scheme should incorporate. These properties include isolability, sensitivity, robustness, reliability, maintainability, survivability and the ability to be reconfigurable. From the beginning of process control, most practical systems have contained some form of fault detection and diagnosis. In the majority of these systems, the detection and diagnostic function is rather simple, being based on straight limit checking. The development of computational equipment has set the scene for the general application of more sophisticated and powerful methods.

Ideally, one would like to detect both classes of failure: additive and multiplicative. Within the classes fall types of failures represented as: vibration, actuator, sensor and process. However, no such system is available to perform total fault detection, yet alone diagnosis. Most fault detection schemes are designed to perform one or two types of fault detection.

A good fault detection scheme would use analytical and heuristic knowledge of the process. Furthermore, the scheme should be sensitive enough to detect small faults but not detect spurious faults (which could occur because of noise), and be robust to modelling errors. Another desirable property is the detection of multiple simultaneous failures.

There are two categories of fault detection and diagnostic schemes – those that do not use the plant model, and those that do. The former category includes methods such as using special or multiple or intelligent sensors (for sensor fault detection only), expert systems (for fault diagnosis only), artificial neural networks (for fault detection only) and neuro-fuzzy networks. Of the

methods listed above, the neuro-fuzzy networks, neural networks and expert system have the most desirable properties.

Model-based schemes of fault detection include three types: process model-based, signal model-based and change detection. The adaptive parity space method has the most desirable properties of the process model-based methods of fault detection, while maximum entropy estimation has the best properties of the signal model-based methods, and finally, both cumulative sum and bayes algorithms exhibit the best properties in methods of change detection.

Generally, most model-based methods of fault detection require a fault diagnostic system built on it, unlike most non-model-based methods. Neural networks have a slight edge in having the most desirable properties of a fault diagnostic scheme, over the other methods.

The three methods chosen for simulation and online were observer, parity equation, and adaptive parity space. The process used for the simulation purposes was a two-tank system. The system has one actuator (one input) and two level sensors (two outputs). Three additive faults were simulated as the two sensor bias faults and actuator bias fault. Three multiplicative faults were simulated as the two sensor scale faults and actuator scale fault. The faults were simulated as the response of the system was in the transient states. The system was required to track a setpoint, which was represented by a step waveform with an amplitude of one volt. All faults were simulated at different times, with no fault occurring at the same time as another fault. Simultaneous faults were not simulated since there was no fault diagnostic scheme implemented.

Simulations were also run with varied fault parameters, fault width, duration, amplitude, type and noise level. The adaptive parity space method was found to be the best fault detection method, as it showed the best isolability results in each simulation case. The method was however found to have a large computational burden, and was difficult to implement. The parity equation and observer methods of fault detection did not show good isolability results as the adaptive parity space method, but were easier to implement and had a lower computational burden. All methods were found to be reliable, have a quick fault detection response time, detect actuator, sensor, abrupt and incipient faults. The parity space method was found to be the most sensitive to noise,

but was found to detect faults with small amplitude and duration, better than the other methods of fault detection investigated. The observer and parity space methods, showed similar results and properties. As a result, the choice of the better method between the two could not be determined. Rather a preferred method can be chosen for any specific application.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS I

SYNOPSIS II

TABLE OF CONTENTS V

LIST OF FIGURES XII

LIST OF TABLES XIX

LIST OF ABBREVIATIONS XX

CHAPTER 1 **INTRODUCTION** 1

1.1 Background on the Dissertation 1

1.2 Problems with the Dissertation 1

1.3 Objectives of the dissertation 1

1.4 Methods of Information Capture 2

1.5 Plan of Development 2

CHAPTER 2 **LITERATURE REVIEW OF FAULT DETECTION AND DIAGNOSIS** 4

2.1 Introduction To Fault Detection And Diagnosis 4

2.2 Developments in Fault Detection and Diagnosis 5

2.3 Description of Fault Detection and Diagnosis 6

2.4 Desirable Properties of Fault Detection and Diagnostic Schemes 7

2.5 Classes Of Failures 8

2.6 Methodology For Model-Based Fault Detection And Diagnosis 9
2.6.1 Analytical Symptoms 10
2.6.2 Heuristic Symptoms 11
2.6.3 Fault Detection 12
2.6.4 Fault Diagnosis 12

2.7 Process Model-Based Fault Detection Methods 12
2.7.1 State observers 13
2.7.2 Parameter Estimation 15
2.7.3 Parity space 15
2.7.4 Adaptive Observer 16
2.7.5 Adaptive Parity Space 17

2.8 Signal Model-Based Fault Detection Methods 18
2.8.1 Spectral Analysis 18
2.8.2 Bandpass Filters 18
2.8.3 Maximum-entropy estimation 19

2.9 Methods Of Change detection 19
2.9.1 Direct Parallel Testing 20
2.9.2 Likelihood Ratio Tests 20
2.9.3 Bayes Decision 21
2.9.4 Multivariate Testing 22
2.9.5 CUSUM algorithm 23
2.9.6 Compound Scalar Testing 23
2.9.7 Other Methods 24

2.10 Fault Diagnosis Methods 25
2.10.1 Geometrical Distance and Probabilistic Methods 25
2.10.2 Neural Networks 26
2.10.3 Fuzzy Classification 27

2.11 Non Model-Based Methods of Fault Detection and Diagnosis 29
2.11.1 Neuro–Fuzzy Networks 29
2.11.2 Neural networks (two-stage) 34
2.11.3 Intelligent sensors 36

2.11.4 Expert Systems 38

2.12 Applications Of Fault Detection and Diagnosis Methods 40

2.12.1 Observer 40

2.12.2 Parameter Estimation 41

2.12.3 Parameter estimation with hypothesis testing 41

2.12.4 Parity space 42

2.12.5 Passive FDI system with neural networks 42

2.12.6 Adaptive Parity Space 43

2.12.7 Spectral analysis & neural networks 43

2.12.8 Multivariate testing with geometric distance and probabilistic methods for diagnosis 44

2.12.9 CUSUM algorithm with fuzzy classification 44

2.12.10 Expert system with neuro-fuzzy networks 45

2.12.11 Overview of applications 46

2.13 Trends in fault detection and diagnosis 47

2.14 Summary 47

2.15 Conclusions 50

2.16 Recommendations 56

CHAPTER 3 **PROCESS DESCRIPTION 58**

3.1 Mathematical Model Of The Process 58

3.2 Determination of the Characteristics of the Process 59

3.2.1 Pump characteristic 60

3.2.2 Depth sensor calibrations 61

3.3 Process Transfer Function 64

3.3.1 Determination of linear operating region 64

3.3.2 Methodology to determine the process model 65

3.3.3 Process model 66

3.4 **Closed Loop Response** 80

3.5 **Nyquist Stability** 82

3.6 **Concluding Remarks** 83

CHAPTER 4 **THEORETICAL DESCRIPTION OF PARITY SPACE, ADAPTIVE PARITY SPACE AND OBSERVER METHODS OF FAULT DETECTION** 84

4.1 **Description of Process and Failure Models** 84

4.2 **Description Of Methods** 87

4.2.1 Parity space 88

4.2.2 Adaptive parity space 95

4.2.3 Observer 102

4.3 **Concluding Remarks** 104

CHAPTER 5 **FAULT DETECTION SIMULATION STUDIES** 106

5.1 **Overview of Simulation Studies** 106

5.2 **Parity Space Results** 109

5.2.1 Performance Criterion for Fault Detection 114

5.2.2 Use of Individual thresholds on each residual 116

5.2.3 Kalman filter implementation 120

5.2.4 Summary for the Parity Space method 123

5.3 **Adaptive Parity Space Results** 124

5.3.1 Least squares 124

5.3.2 Recursive Least Squares 129

5.3.3 Recursive Least Squares with Forgetting Factor 134

5.3.4 Recursive Least Squares with forgetting and stabilising factors 139

5.3.5 Summary for the Adaptive Parity Space method 145

5.4 **Observer Results** 145

5.4.1 Use of individual thresholds on each residual 148

5.4.2 Summary for the Observer method 150

5.5 Performance of Fault Detection Methods with Varying Fault Parameters..... 150

5.5.1 Parity Space Performance..... 152

5.5.2 Adaptive Parity Space Performance 156

5.5.3 Observer Performance 162

5.5.4 Comparison of the fault detection methods..... 166

5.6 Performance of Fault Detection Methods with Varying Noise Levels 167

5.6.1 Parity Space method 167

5.6.2 Adaptive Parity Space method 169

5.6.3 Observer method..... 170

5.6.4 Comparison of the methods..... 170

5.7 Discussion on Reliability, Ease of Implementation and Computational Burden 171

5.8 Concluding Remarks..... 172

CHAPTER 6 **CONCLUSIONS 176**

CHAPTER 7 **RECOMMENDATIONS..... 180**

BIBLIOGRAPHY 181

GLOSSARY 185

APPENDIX A: SUMMARY OF PUBLISHED FAULT DETECTION & DIAGNOSIS PUBLICATION A-1

APPENDIX B: DETAILED PROCESS DESCRIPTION B-1

B.1 System specifications and operation..... B-1

B.2 Mathematical model of the system B-3

APPENDIX C: PROCESS CHARACTERISTICS C-1

C.1 Pump Characteristic C-1

C.2 Depth Sensor Characteristics C-5

APPENDIX D: LS & RLS THEORY D-1

D.1 Least Squares D-1

D.1.1 Practical Considerations D-2

D.2 Recursive Least Squares D-2

D.2.1 Initial Conditions D-3

D.2.2 Practical Considerations D-3

D.2.3 Time Varying Systems D-4

APPENDIX E: KALMAN FILTER E-1

E.1 Calculation of R matrix E-1

E.2 Calculation of Q matrix E-2

E.3 Applied results using the Kalman Filter E-3

APPENDIX F: EFFECT & DETERMINATION OF STABILISING & FORGETTING FACTORS ON THE APS FAULT DETECTION F-1

F.1 Forgetting Factor and Stabilising Factor values F-1

F.2 Determination of Forgetting Factor F-2

F.3 Determination of Stabilising Factor F-7

F.4 Effect of Forgetting and Stabilising Factors on APS Performance F-13

APPENDIX G: CORRELATION RESULTS FOR VARYING FAULT PARAMETERS FOR THE FAULT DETECTION METHODS G-1

APPENDIX H: RESIDUAL AND FAULT INDICATION GRAPHS FOR VARIED FAULT PARAMETERS H-1

H.1 Parity Space Method H-2

H.2 Adaptive Parity Space Method H-10

H.3 Observer Method H-21

APPENDIX I: REVISED CORRELATION GRAPHS ANDDATA FOR THE APS FAULT DETECTION METHOD I-1

I.1 Correlation Graphs I-1

I.2 Correlation Results I-4

APPENDIX J: CORRELATION RESULTS FOR VARIED NOISE LEVELS J-1

APPENDIX K: APS WITH RLS RESIDUAL GRAPHS K-1

LIST OF FIGURES

Figure 2-1 Scheme of model-based fault detection and diagnosis (Fig. 1. Isermann and Ballé, 1996) 10
Figure 2-2 Scheme for model-based fault detection (Fig. 2. Isermann and Ballé, 1996) 11
Figure 2-3 Structure of a neuro-fuzzy network (adapted Fig 8. Halfmann 1996) 30
Figure 2-4 Schematic diagram of fault detection and diagnosis system (Fig. 3 Maki and Loparo, 1997) 35
Figure 2-5 The self-validating sensor (Fig. 2 Henry and Clarke, (1991)) 37
Figure 2-6 Expert system architecture (Fig. 15-1 Patton et al., 1989)) 39
Figure 3-1 The Two-Tank Process 58
Figure 3-2 Pump Characteristic 60
Figure 3-3 LS Fit to Sensor 1 Characteristic 62
Figure 3-4 LS Fit to Sensor 2 Characteristic 62
Figure 3-5 Typical Step Test 65
Figure 3-6 Typical LS fit to Step Data 67
Figure 3-7 Simulated Step Responses for Tank 2 72
Figure 3-8 Model Deviations for Tank 2 73
Figure 3-9 Typical Step Test for Tank 1 with an Offset of 2V 75
Figure 3-10 Model Deviations for Tank 1 76
Figure 3-11 Average Step Responses for Tank 1 78
Figure 3-12 Approximated G1(s) Models with Average Step Response for Tank 1 79
Figure 3-13 Closed Loop Step Responses: Simulated and Real time of the System 81
Figure 3-14 Nyquist Plot of the System 82
Figure 4-1 Fault Detection System 87
Figure 4-2 Fault Detection system with Kalman Filter 93
Figure 4-3 Standard observer configuration (Fig 10-1 Chipperfield and Fleming, 1993) 94
Figure 4-4 Adaptive Parity Space Fault Detection Scheme 102
Figure 4-5 Observer Fault Detection Scheme 104
Figure 5-1 A Typical Simulation of the Process to a Step with Faults included 108
Figure 5-2 Parity Space Residuals 1 & 2 in a fault-free environment 110
Figure 5-3 Parity Space Residuals 3 & 4 in a fault-free environment 110
Figure 5-4 Parity Space Residuals 1 & 2 subject to faults 111

Figure 5-5 Parity Space Residuals 3 & 4 subject to faults 112
Figure 5-6 Parity Space Fault Indication 113
Figure 5-7 A Typical Residual Response to a Fault 115
Figure 5-8 Parity Space Fault Indication with Four Thresholds 117
Figure 5-9 Parity Space Residuals 1 & 2 with actuator faults of 5V amplitude 118
Figure 5-10 Parity Space Residuals 3 & 4 with actuator faults of 5V amplitude 118
Figure 5-11 Parity Space Fault Indication with actuator faults of 5V amplitude 119
Figure 5-12 Parity Equation Residuals 1 & 2 with Kalman filter 120
Figure 5-13 Parity Equation Residuals 3 & 4 with Kalman filter 121
Figure 5-14 Parity Space Fault Indication with Kalman filter 122
Figure 5-15 Adaptive Parity Space Residuals M1 & V1 using LS estimation in a fault-free environment 125
Figure 5-16 Adaptive Parity Space Residuals M2 & V2 using LS estimation in a fault-free environment 125
Figure 5-17 Adaptive Parity Space Residuals S & D using LS estimation in a fault-free environment 126
Figure 5-18 Adaptive Parity Space Residuals M1 & V1 using LS estimation subject to faults 127
Figure 5-19 Adaptive Parity Space Residuals M2 & V2 using LS estimation subject to faults 127
Figure 5-20 Adaptive Space Residuals S & D using LS estimation subject to faults 128
Figure 5-21 Adaptive Parity Space Fault Indication using LS estimation 128
Figure 5-22 APS Residuals M1 & V1 using RLS in a fault-free environment 129
Figure 5-23 Adaptive Parity Space Residuals M2 & V2 using RLS in a fault-free environment 130
Figure 5-24 Adaptive Parity Space Residuals S & D using RLS in a fault-free environment 130
Figure 5-25 Adaptive Parity Space Residuals M1 & V1 using RLS with faults 131
Figure 5-26 Adaptive Parity Space Residuals M2 & V2 using RLS with faults 132
Figure 5-27 Adaptive Parity Space Residuals S & D using RLS with faults 132
Figure 5-28 Adaptive Parity Space Fault Indication using RLS 133
Figure 5-29 Adaptive Parity Space Residuals M1 & V1 using LS estimation subject to faults 134

Figure 5-30 Adaptive Parity Space Residuals M2 & V2 using LS estimation in a fault-free environment 135
Figure 5-31 Adaptive Space Residuals S & D using LS estimation subject to faults 135
Figure 5-32 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with λ, subject to faults 136
Figure 5-33 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with λ, subject to faults 137
Figure 5-34 Adaptive Space Residuals S & D using RLS estimation with λ, subject to faults 137
Figure 5-35 Adaptive Parity Space Fault Indication using RLS estimation with λ 138
Figure 5-36 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with α and λ, subject to no faults 139
Figure 5-37 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with α and λ, subject to no faults 140
Figure 5-38 Adaptive Space Residuals S & D using RLS estimation with α and λ, subject to no faults 140
Figure 5-39 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with α and λ, subject to faults 141
Figure 5-40 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with α and λ, subject to faults 142
Figure 5-41 Adaptive Space Residuals S & D using RLS estimation with α and λ, subject to faults 142
Figure 5-42 Adaptive Parity Space Fault Indication using RLS estimation with α and λ, subject to faults 143
Figure 5-43 New Adaptive Parity Space Fault Indication using RLS estimation with α and λ, subject to faults 144
Figure 5-44 Observer residuals in a fault-free environment 146
Figure 5-45 Observer residuals subjected to faults 147
Figure 5-46 Observer Fault Indication 147
Figure 5-47 Observer Fault Indication with two thresholds 149
Figure 5-48 Parity Space Correlation Graph for Ramp Waveform 152
Figure 5-49 Parity Space Correlation Graph for Saw Waveform 153
Figure 5-50 Parity Space Correlation Graph for Sine Waveform 153
Figure 5-51 Parity Space Correlation Graph for Square Waveform 154

Figure 5-52 Parity Space Correlation Graph for Step Waveform 154
Figure 5-53 Adaptive Parity Space Correlation Graph for Ramp Waveform 157
Figure 5-54 Adaptive Parity Space Correlation Graph for Saw Waveform 157
Figure 5-55 Adaptive Parity Space Correlation Graph for Sine Waveform 158
Figure 5-56 Adaptive Parity Space Correlation Graph for Square Waveform 158
Figure 5-57 Adaptive Parity Space Correlation Graph for Step Waveform 159
Figure 5-58 New Adaptive Parity Space Correlation Graph for Saw Waveform 161
Figure 5-59 Observer Correlation Graph for Ramp Waveform 162
Figure 5-60 Observer Correlation Graph for Saw Waveform 163
Figure 5-61 Observer Correlation Graph for Sine Waveform 163
Figure 5-62 Observer Correlation Graph for Square Waveform 164
Figure 5-63 Observer Correlation Graph for Step Waveform 164
Figure 5-64 Parity Space Percentage of Correlation against Noise 168
Figure 5-65 Adaptive Parity Space Percentage of Correlation against Noise 169
Figure 5-66 Observer Percentage of Correlation against Noise 170
Figure B-1 Schematic View of the Pump System B-2
Figure C-1 Pump Characteristic C-3
Figure C-2 LS Error on Pump Characteristic C-4
Figure C-3 Sensor Characteristics C-6
Figure C-4 Sensor 1 Characteristic C-8
Figure C-5 LS Error on Sensor 1 Characteristic C-8
Figure C-6 Sensor 2 Characteristic C-9
Figure C-7 LS Error on Sensor 2 Characteristic C-9
Figure C-8 LS Fit to Sensor 1 Characteristic C-12
Figure C-9 Revised LS Error on Sensor 1 Characteristic C-12
Figure C-10 LS Fit to Sensor 2 Characteristic C-13
Figure C-11 LS Error on Sensor 2 Characteristic C-13
Figure D-1 Process with measurement noise D-1
Figure E-1 Parity Space Residuals 1 & 2 with Kalman filter, using Q1 ..... E-3
Figure E-2 Parity Space Residuals 3 & 4 with Kalman filter, using Q1 ..... E-4
Figure E-3 Parity Space Residuals 1 & 2 with Kalman filter, using Q2 ..... E-4
Figure E-4 Parity Space Residuals 3 & 4 with Kalman filter, using Q2 ..... E-5
Figure E-5 Parity Space Residuals 1 & 2 with Kalman filter, using Q3 ..... E-5
Figure E-6 Parity Space Residuals 3 & 4 with Kalman filter, using Q3 ..... E-6

Figure E-7 Parity Space Residuals 1 & 2 with Kalman filter, using Q4 ..... E-6
Figure E-8 Parity Space Residuals 3 & 4 with Kalman filter, using Q4 ..... E-7
Figure E-9 Parity Space Residuals 1 & 2 with Kalman filter, using Q5 ..... E-7
Figure E-10 Parity Space Residuals 3 & 4 with Kalman filter, using Q5 ..... E-8
Figure E-11 Parity Space Residuals 1 & 2 with Kalman filter, using Q6 ..... E-8
Figure E-12 Parity Space Residuals 3 & 4 with Kalman filter, using Q6 ..... E-9
Figure E-13 Parity Space Residuals 1 & 2 with Kalman filter, using Q7 ..... E-9
Figure E-14 Parity Space Residuals 3 & 4 with Kalman filter, using Q7 ..... E-10
Figure E-15 Parity Space Residuals 1 & 2 with Kalman filter, using Q7 ..... E-10
Figure E-16 Parity Space Residuals 3 & 4 with Kalman filter, using Q8 ..... E-11
Figure F-1 APS Residuals M1 & V1 for Simulation1 ..... F-2
Figure F-2 APS Residuals M2 & V2 for Simulation 1 ..... F-3
Figure F-3 APS Residuals S & D for Simulation 1 ..... F-3
Figure F-4 APS Residuals M1 & V1 for Simulation 2 ..... F-4
Figure F-5 APS Residuals M2 & V2 for Simulation 2 ..... F-4
Figure F-6 APS Residuals S & D for Simulation 2 ..... F-5
Figure F-7 APS Residuals M1 &V1 for Simulation 3 ..... F-5
Figure F-8 APS Residuals M2 & V2 for Simulation 3 ..... F-6
Figure F-9 APS Residuals S & D for Simulation 3 ..... F-6
Figure F-10 APS Residuals M1 & V1 for Simulation 4 ..... F-8
Figure F-11 APS Residuals M2 & V2 for Simulation 4 ..... F-8
Figure F-12 APS Residuals D & S for Simulation 4 ..... F-9
Figure F-13 APS Residuals M1 & V1 for Simulation 5 ..... F-9
Figure F-14 APS Residuals M2 & V2 for Simulation 5 ..... F-10
Figure F-15 APS Residuals D & S for Simulation 5 ..... F-10
Figure F-16 APS Residuals M1 & V1 for Simulation 6 ..... F-11
Figure F-17 APS Residuals M2 & V2 for Simulation 6 ..... F-11
Figure F-18 APS Residuals D & S for Simulation 6 ..... F-12
Figure F-19 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation F-13
Figure F-20 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (zoomed) F-14
Figure F-21 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (contour map) F-15

Figure F-22 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (zoomed with contours) F-15
Figure H-1 Parity Space Residuals 1 & 2 for Simulation 1 ..... H-2
Figure H-2 Parity Space Residuals 3 & 4 for Simulation 1 ..... H-3
Figure H-3 Parity Space Fault Indication for Simulation 1 ..... H-3
Figure H-4 Parity Space Residuals 1 & 2 for Simulation 2 ..... H-4
Figure H-5 Parity Space Residuals 3 & 4 for Simulation 2 ..... H-4
Figure H-6 Parity Space Fault Indication for Simulation 2 ..... H-5
Figure H-7 Parity Space Residuals 1 & 2 for Simulation 3 ..... H-5
Figure H-8 Parity Space Residuals 3 & 4 for Simulation 3 ..... H-6
Figure H-9 Parity Space Fault Detection for Simulation 3 ..... H-6
Figure H-10 Parity Space Residuals 1 & 2 for Simulation 4 ..... H-7
Figure H-11 Parity Space Residuals 3 & 4 for Simulation 4 ..... H-7
Figure H-12 Parity Space Fault Detection for Simulation 4 ..... H-8
Figure H-13 Parity Space Residuals 1 & 2 for Simulation 5 ..... H-8
Figure H-14 Parity Space Residuals 3 & 4 for Simulation 5 ..... H-9
Figure H-15 Parity Space Fault Indication for Simulation 5 ..... H-9
Figure H-16 Adaptive Parity Space Residuals M1 & V1 for Simulation 1 ..... H-10
Figure H-17 Adaptive Parity Space Residuals M2 & V2 for Simulation 1 ..... H-11
Figure H-18 Adaptive Parity Space Residuals S & D for Simulation 1 ..... H-11
Figure H-19 Adaptive Parity Space Fault Indication for Simulation 1 ..... H-12
Figure H-20 Adaptive Parity Space Residuals M1 & V1 for Simulation 2 ..... H-12
Figure H-21 Adaptive Parity Space Residuals M2 & V2 for Simulation 2 ..... H-13
Figure H-22 Adaptive Parity Space Residuals S & D for Simulation 2 ..... H-13
Figure H-23 Adaptive Parity Space Fault Indication for Simulation 2 ..... H-14
Figure H-24 Adaptive Parity Space Residuals M1 & V1 for Simulation 3 ..... H-14
Figure H-25 Adaptive Parity Space Residuals M2 & V2 for Simulation 3 ..... H-15
Figure H-26 Adaptive Parity Space Residuals S & D for Simulation 3 ..... H-15
Figure H-27 Adaptive Parity Space Fault Indication for Simulation 3 ..... H-16
Figure H-28 Adaptive Parity Space Residuals M1 & V1 for Simulation 4 ..... H-16
Figure H-29 Adaptive Parity Space Residuals M2 & V2 for Simulation 4 ..... H-17
Figure H-30 Adaptive Parity Space Residuals S & D for Simulation 4 ..... H-17
Figure H-31 Adaptive Parity Space Fault Indication for Simulation 4 ..... H-18
Figure H-32 Adaptive Parity Space Residuals M1 & V1 for Simulation 5 ..... H-18

Figure H-33 Adaptive Parity Space Residuals M2 & V2 for Simulation 5 .....H-19
Figure H-34 Adaptive Parity Space Residuals S & D for Simulation 5.....H-19
Figure H-35 Adaptive Parity Space Fault Indication for Simulation 5 .....H-20
Figure H-36 Adaptive Parity Space new Fault Indication for Simulation 5 .....H-21
Figure H-37 Observer Residuals 1 & 2 for Simulation 1 .....H-22
Figure H-38 Observer Fault Detection for Simulation 1 .....H-22
Figure H-39 Observer Residuals 1 & 2 for Simulation 2 .....H-23
Figure H-40 Observer Fault Detection for Simulation 2 .....H-23
Figure H-41 Observer Residuals 1 & 2 for Simulation 3 .....H-24
Figure H-42 Observer Fault Detection for Simulation 3 .....H-24
Figure H-43 Observer Residuals 1 & 2 for Simulation 4 .....H-25
Figure H-44 Observer Fault Detection for Simulation.....H-25
Figure H-45 Observer Residuals 1 & 2 for Simulation 5 .....H-26
Figure H-46 Observer Fault Detection for Simulation 5 .....H-26
Figure I-1 New APS Correlation Graph for Ramp waveform..... I-1
Figure I-2 New APS Correlation Graph for Saw waveform I-2
Figure I-3 New APS Correlation Graph for Sine waveform I-2
Figure I-4 New APS Correlation Graph for Square waveform I-3
Figure I-5 New APS Correlation Graph for Step waveform I-3
Figure K-1 APS Residuals M1 and V1 using RLS K-1
Figure K-2 APS Residuals M2 and V2 using RLS K-2
Figure K-3 APS Residuals S and D using RLS.....K-2
Figure K-4 APS Residuals S and D using RLS with λ K-3

LIST OF TABLES

Table 2-1 Table of Properties of Fault Detection Methods 48
Table 2-2 Table of Properties of Fault Diagnosis Methods 49
Table 3-1 Sensor Level Heights for Different Pump Drive Input Voltages 64
Table 3-2 Step Test Results for Tank 2 68
Table 3-3 Step Test Results for Tank 1 74
Table 5-1 Summary of the Percentages of Correlation obtained for the investigated Fault Detection Methods 172
Table 5-2 Comparison of Simulated and Predicted Properties of the Investigated Fault Detection Methods 174
Table C-2 Sensor Calibration Data C-5
Table C-1 Pump Characteristic with LS Data and Error C-2
Table C-3 Sensor Characteristics with LS Data and Errors C-7
Table C-4 Sensor Characteristics with LS Data and Errors C-11
Table E-I Covariance Matrices for different Simulations E-1
Table E-II Diagonal elements of the Q matrices with different "rootsize", for three simulations E-2
Table F-1 Fault Parameters used for the Simulations F-2
Table F-2 Fault Parameters used for the Simulations F-7
Table H-1 Fault Parameters used for the Simulations H-2

LIST OF ABBREVIATIONS

ACC:	Adaptive Cruise Control
APS:	Adaptive Parity Space
ARMA:	Autoregressive Moving Average
CSTR:	Continuously Well - Stirred Tank Reactor
CUSUM:	Cumulative Sum
DSM:	Diagnostic State Machine
ES:	Expert System
FDF:	Fault Detection Filter
FDI:	Fault Detection and Isolation
FFT:	Fast Fourier Transform
GA:	Genetic Algorithm
GLR:	General Likelihood-Ratio
HTM:	Hypothesis Testing Modules
IFAC:	International Federation of Automatic Control
IFD:	Instrument Fault Detection
IV:	Instrumental Variable
LS:	Least Squares
MIMO:	Multiple Input Multiple Output
MV:	Measurement Value

NLU:	Next Level Up
NN:	Neural Network
NOR:	Normal Operating Region
PCA:	Principal Component Analysis
PLS:	Partial Least Squares
RBF:	Radial Basis Function
RLS:	Recursive Least Squares
RMS:	Root-Mean-Square
RMV:	Raw Measurement Value
RU:	Raw Uncertainty
SEE:	Standard Error of Estimate
SPE:	Squared Prediction Error
SVF:	State Variable Filter
VI:	Validity Index
VMV:	Validated Measurement Value
VU:	Validated Uncertainty

Chapter 1 INTRODUCTION

This chapter begins by introducing the background to this dissertation. A few of the problems encountered by the author are then stated. This is followed by the objectives that this dissertation intends to fulfill. The methods of information capture follow, and finally, the plan of development that this dissertation follows, is given.

1.1 Background on the Dissertation

Fault detection and diagnosis has always been adopted into many process systems, from the beginning. In most cases, the methods used were primitive, with the emphasis being on productivity and profitability. Such systems were found to be very unsafe to the human operators and to the processes themselves. Much gratitude is owed to the pioneers of fault detection and diagnosis for providing the tools so that current researchers may be innovative in their thought and be able to create efficient and safe processes.

1.2 Problems with the Dissertation

The major problems experienced were a lack of resources, which included both proceedings to conferences and software.

1.3 Objectives of the dissertation

The objectives of this dissertation are:

1. To summarise the methods of fault detection and diagnosis.
2. To draw conclusions and make recommendations on the fault detection and diagnosis literature obtained.
3. To describe the parity space, adaptive parity space and observer methods of fault detection.

4. To analyse the results of simulating the above mentioned methods using a model for a simple laboratory process.
5. To draw comparisons between theory and the simulated results.
6. To draw comparisons between the three fault detection methods.
7. To draw conclusions of these methods and make recommendations.

1.4 Methods of Information Capture

Information was captured via the internet, but mostly discarded due to the lack of detail in the articles that were found. Relevant literature was obtained from various journals and books found in the University of Cape Town, University of Pretoria and University of the Witwatersrand libraries. The proceedings of various conferences were also a source of information. Since the literature review given in Chapter Two was compiled in 1998, only developments up to this time have been analysed.

1.5 Plan of Development

The dissertation begins by describing fault detection and diagnosis in Chapter Two. A description on the classes of failure, symptoms and desirable properties of fault detection and diagnosis schemes follows. The various methods of fault detection and diagnosis are discussed and properties given. Conclusions are drawn and recommendations are made at the end of the literature review.

Chapter Three describes the mathematical model of a two-tank system used for the fault detection simulation. The calibration of the system is then described. The method and tests performed in order to obtain the process transfer function follows. The results of the step tests (used to obtain the system transfer function) are then analysed. Finally, the closed loop response of the process is discussed.

In Chapter Four, the theory for the parity space method of fault detection is given. This is followed by detailed descriptions of the adaptive parity space and observer methods of fault detection.

The beginning of Chapter Five describes the simulation parameters used for the investigation of the three chosen fault detection methods. The results of simulating these methods are given and an analysis of the results is then described. Further simulations are then described, with a discussion on the results of varied fault and simulation parameters. Finally, the results obtained for methods of fault detection are compared and summarised.

Chapters Six describes the conclusions drawn on the project. Recommendations made for future research are stated in Chapter Seven.

Note:

Because a colour bubble jet printer was utilised to print some diagrams, the font on those pages is more dense than the rest of the pages.

Chapter 2 LITERATURE REVIEW OF FAULT DETECTION AND DIAGNOSIS

An introduction into fault detection and diagnosis starts off the chapter. Developments and an overview of fault detection and diagnosis follow. Desirable properties are then outlined. This is followed by a description of the classes of failure. The author then describes the fault detection and diagnosis methods, for which literature has been obtained, together with the properties of these methods. Applications of these methods are briefly given. Trends and a summary of fault detection and diagnosis follow. Finally, conclusions are drawn and recommendations made.

2.1 Introduction To Fault Detection And Diagnosis

The early detection and diagnosis of faults in processes is one of the most important tasks assigned to the computers supervising these processes. The early indication of abrupt or incipient failures can assist in avoiding major plant breakdowns and catastrophes, which could otherwise result in substantial material damage and even human fatalities. Furthermore, preventative maintenance on the plant may be performed. From the beginning of process control, most practical systems have contained some form of fault detection and diagnosis.

A good overview on the fault detection and diagnosis procedure is given by Patton et al., 1989. Briefly, original methods of fault detection and diagnosis included straight limit checking of the measured signals of the system. A fault was indicated when the signal exceeded a predetermined threshold, and diagnosis deduced by the system operators. The assumption with this method was that the hardware elements (actuators, process components, sensors, etc.) were all fault-free. As a result, methods of fault detection began to use hardware redundancy, by distributing many hardware elements around the system. These schemes checked the generated outputs for consistency. Usually, there were three or four repeated elements measuring the same variable, whereas one would be sufficient if it were fault-free. A logic circuit was then used to declare a fault if a measurement from a hardware element deviated out of a predetermined bound set around the average of the other elements. Problems with using hardware redundancy configurations included the extra software, cost of

the equipment used and the space that was unnecessarily used, or unavailable for the equipment. Furthermore, the life expectancies of equipment are generally the same, so if one becomes faulty, the others would soon follow.

The development of computational equipment has set the scene for the general application of more sophisticated and powerful methods, which eliminate some or all of the hardware redundancy. Such methods, which use the dynamic state of the system to aid fault detection, are said to be analytically or artificially redundant.

2.2 Developments in Fault Detection and Diagnosis

The development on the field of fault detection and diagnosis began in the early 1970's. Beard (1971) and Jones (1973) reported an observer based fault detection in linear systems. Wilsky (1976) summarised this development. Then Rault, et al. (1971) considered the application of identification methods to the fault detection of jet engines. Himmelblau (1978) wrote a book on model – based methods of fault detection and diagnosis of chemical processes. Clark (1978) showed instrument failure detection based on the inherent analytical redundancy of multiple observers. Baskiotis et al. (1976), Geiger (1982), Filbert and Metzger (1982) demonstrated the use of parameter estimation methods for fault detection of technical systems. The development of process fault detection methods based on modelling parameter and state estimation was then summarised by Isermann (1984). Parity space based methods shown by Chow and Wilsky (1984), were further developed by Patton and Chen (1991), Gertler (1991), Höffling and Pfeufer (1994).

The development of fault detection methods up to the respective time is summarised in the books Pau (1981), Patton et al. (1989), Basseville and Nikiforov (1993) and in survey papers by Gertler (1988), Frank (1990) and Isermann (1994). Further developments can be found in the proceedings of the International Federation of Automatic Control (IFAC) Symposium SAFEPROCESS that have been held in 1991, 1994 and 1997.

2.3 Description of Fault Detection and Diagnosis

Patton et al. in 1989,and Isermann and Ballé in 1996 have provided a thorough description of fault detection and diagnosis. Briefly, all automatically controlled systems are required to indicate undesired or unpermitted process states and then take the appropriate actions to maintain control of the systems, thereby avoiding damages and accidents. The functions that are required to maintain the above functionality are process monitoring, automatic protection and supervision with fault diagnosis. Furthermore, the following criteria must be met in any fault detection and diagnostic scheme:

- Early detection of faults with abrupt or incipient time behaviour.
- Diagnosis of actuator, process and sensor faults.
- Detection of faults in closed loop.
- Supervision of the process in transient state.
- Low false alarm rate.
- Detection of as many faults as possible (good reliability).

The function of any fault detection scheme is to register an alarm when a failure has occurred, and to identify it. Abrupt faults (such as step changes) have to be detected so as to reconfigure the system, thus avoiding catastrophic consequences. Furthermore, these faults must especially be detected promptly. Incipient faults (such as bias or drift) are much more difficult to detect since they are small faults. This can be overcome by using a large time for detection, at the expense of detecting faults early. Furthermore, incipient faults are usually associated with maintenance problems of degraded equipment. Having a low false alarm rate is generally unacceptable in fault detection schemes, as this leads to a lack of confidence in the scheme, and psychologically affects the operators of the system since they might be mislead when an actual fault does occur. Good reliability is important, but is usually obtained at the expense of abrupt detection and/or increase in false alarm rates.

The compromises in prompt detection, false alarm rate and sensitivity are dependent on the requirements of the application that the fault detection scheme will be implemented on. Generally, the more properties required by the fault detection scheme, the more complex the

scheme becomes, with the possibility of using both analytical and hardware redundancy. The faults can be detected by analytical and/or heuristic symptoms.

Fault diagnosis determines the type, size, location and time of the fault(s), based on the observed analytic and heuristic symptoms. There are two types of fault detection and diagnosis methods, those that use the model of the system and those which do not. These will be discussed after the desirable properties of fault detection and diagnosis schemes are given, and the classes of faults are described.

2.4 Desirable Properties of Fault Detection and Diagnostic Schemes

The following describes the desirable properties for any fault detection and diagnostic scheme (as described by Gertler, 1988; Stengel, 1997):

1. **Isolability** – the need of the scheme to isolate non-spurious, specific faults. The isolability is dependent on the statistical methods used for the fault detection and on the structure of the plant model. It is difficult to determine the statistical threshold used to detect faults. Implementing a low threshold would result in intermediate and large faults being detected, but at the expense of detecting spurious faults. Implementing a high threshold would result in large faults being detected, intermediate faults possibly being detected and small faults left undetected.
2. **Sensitivity** – a qualitative measure that characterises the size of faults – that can be isolated under specific conditions. Sensitivity is dependent on the size of the elements in the system matrices, and on the noise within the system.
3. **Robustness** – the ability of the scheme to detect faults in the presence of modelling errors and component failures. Modelling errors are inevitable and can interfere with the fault detection. The time history of a multiplicative fault is the only way in which modelling errors can be differentiated from multiplicative faults.
4. **Reconfigurable** – the structure or parameters of the scheme must adjust to faults. Such a scheme is said to be both adaptive and redundant. The scheme is adaptive since it can adjust to off-nominal behaviour such as degradation of sensors, actuators and power

supplies. The scheme is redundant in its ability to overcome lost capabilities with its remaining resources.

5. **Reliability** – the ability of the scheme to complete a task satisfactorily over the period that it is required to do so. Good reliability allows the normal completion of tasks after failures. The overall system reliability depends on the reliability of the physical components of the system and the computer. The computer is however assumed to be fault-free, especially with the advances in technology, which promise ever-increasing reliability for computer systems. The computer faults are therefore not addressed for any fault detection and diagnosis scheme.
6. **Maintainability** – the scheme needs to allow for easy repair to the system, without compromising performance. A good scheme would allow for increased time between maintenance actions and use simpler repair procedures.
7. **Survivability** – measures the ability of the scheme to operate safely (without danger to human operators or the system), irrespective of whether the tasks are completed. A permissible scheme allows for degradation in performance after a fault, as long as the system is brought to an acceptable state of operation.

The above properties are influenced by the selection/transformation of the plant model. Designing for a more robust fault detection scheme would result in a decrease in sensitivity, and vice versa. It therefore becomes necessary to choose a model that meets as much of the above criteria with an acceptable compromise.

2.5 Classes Of Failures

There are three classes of failures, as outlined by Gertler in 1988:

1. Additive measurement faults are the discrepancies between the measured and true values of plant output or input variables. They can also be used to describe actuator malfunctions; now the discrepancy is between the intended (computed) control and its realisation by the actuator.

2. Additive process faults are the disturbances acting on the plant, which are normally zero and cause a shift in the plant outputs independent of the measured inputs. Such faults best describe plant leaks, loads, etc.
3. Multiplicative (process) faults are the abrupt or gradual changes of the plant parameters. Such faults best describe the deterioration of plant equipment, such as partial or total loss of power, surface contamination, etc.

Given that any system is split up into three subsystems (Patton et al., 1989): actuators, actual process and sensors (or instrumentation), if a sensor fault occurs, the fault can be corrected by electronic switching techniques without any reconfiguration of the mechanical parts. The compensation for actuator faults is generally more difficult than a redirection of electric signals. Compensation for the process faults is the most difficult.

2.6 Methodology For Model-Based Fault Detection And Diagnosis

Model-based methods of fault detection and diagnosis are the most researched and widely used. The available knowledge for fault detection comprises both quantifiable, analytical information and qualitative, heuristic information on the process. This section describes the two symptoms that are generated for fault detection and the task of fault diagnosis, as outlined by Isermann and Ballé in 1996. The general scheme for model (or knowledge)-based fault detection and diagnosis is shown in Figure 2-1.

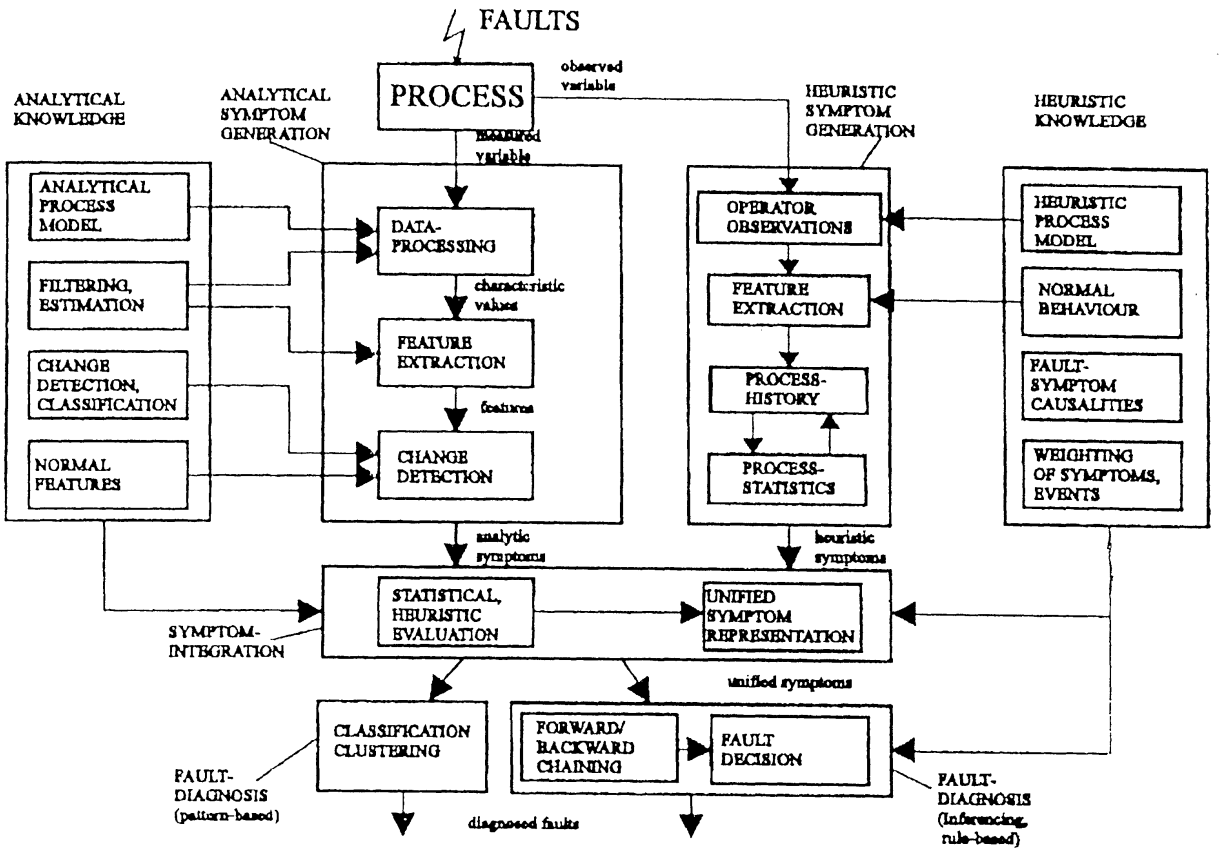


Figure 2-1 Scheme of model-based fault detection and diagnosis (Fig. 1. Isermann and Ballé, 1996)

2.6.1 Analytical Symptoms

The analytical symptoms (obtained from the analytical information) combined with processed data of the measured process variables, are used for fault detection. The processed data is actually the first characteristic values generated from either of the following:

- Limit value checking of measurable signals. The characteristic values are exceeded signal tolerances.
- Signal analysis of measurable signals using signal models such as correlation functions and frequency spectra. The characteristic values are then variances and frequencies, for example.
- Process analysis using mathematical models together with observer, parameter estimation and parity space methods. Characteristic values are state variables, parameters or residuals.

It is possible sometimes to extract special features like physically defined process coefficients, from the characteristic values. These special features when compared with features of the non-faulty process yield discrepancies, which are also analytical symptoms. The comparison is performed by methods of change detection (statistical tests such as mean and variance estimation). There are two types of model-based fault detection. Process model-based methods make use of the input, **U(t)** and output, **Y(t)** variables, while Signal model-based methods use only the output variables, **Y(t)**. The following figure shows the scheme used for model-based fault detection.

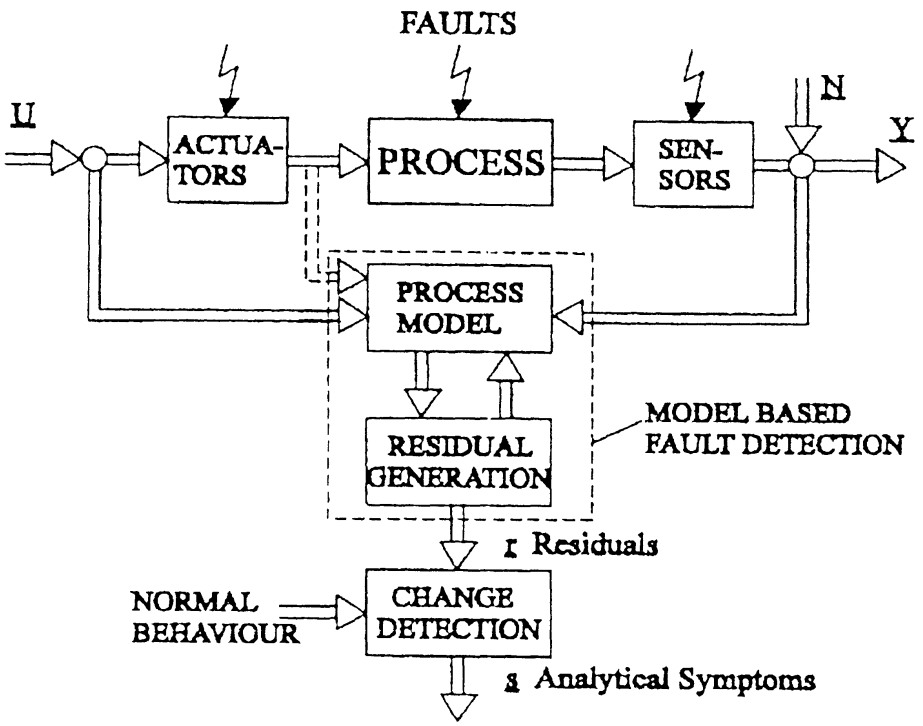


Figure 2-2 Scheme for model-based fault detection (Fig. 2. Isermann and Ballé, 1996)

2.6.2 Heuristic Symptoms

Heuristic symptoms are generated from the qualitative information supplied by human operators of the process and the process history. The heuristic symptoms are represented as linguistic variables or vague numbers. These symptoms together with analytic symptoms provide a more accurate method of fault detection as compared to using analytic symptoms only.

2.6.3 Fault Detection

The following have researched process model-based fault detection methods extensively: Isermann and Pfeufer, 1996; Gertler, 1988 and Frank, 1990. Briefly, the analytic and heuristic symptoms together with the mathematical model of the plant system are used to generate residuals (functions of the fault vector). Under analytical redundancy, the residual generator performs a validation of the nominal relationships of the measured inputs and outputs of the plant system. A fault then occurs when one of the residual vectors is not approximately equal to zero. There are three models that are required: nominal, actual (observed) and faulty. For a good fault detection scheme with a low false alarm rate, the nominal model should be tracked and updated by the actual model.

2.6.4 Fault Diagnosis

Fault diagnosis is concerned with the task of determining the type, size and location of the fault, and the time of detection – based upon the observed analytical and heuristic symptoms. Classification methods use multidimensional symptom vectors to obtain the distance and direction of these vectors compared to a reference vector of the normal steady state. The distances and directions can be found, for example, using artificial neural networks, probabilistic or geometrical methods. Clustering methods group the symptom vectors according to their similarity as compared to the reference vector. Another method is the use of heuristic knowledge in the form of heuristic process (qualitative) models, fault-symptom causalities and weighting of effects. Different diagnostic reasoning strategies such as: forward and backward chaining via fault-symptom trees or approximate reasoning via IF – THEN rules and fuzzy logic can be applied. Finally, a fault decision of the type, size and location of the most possible fault with the time of detection will be indicated.

2.7 Process Model-Based Fault Detection Methods

The general properties of these methods include:

- Able to detect actuator, process and sensor faults.
- Able to detect all classes of faults which can be incipient or abrupt.
- Unable to detect vibration faults.

- Are reliable.
- Do not require *a priori* information.

There are five methods that are described in this section: State Observers (or Estimators), Parameter Estimation, Parity Space, Adaptive Observer and Adaptive Parity Space.

2.7.1 State observers

This method can simply be called Observer. The methodology is as follows: the outputs of the system are reconstructed from the measurements, or subsets of the measurements, with the aid of observers or estimators. These outputs are reconstructed using the estimation error as a residual for the detection and isolation of the faults. Full or reduced-order state observers can be used for state estimation in the deterministic case, while Kalman filters can be used in the stochastic case, when noise and input disturbances have to be considered. A fault is reported when the value of the residual is not equal to zero, by using threshold logic. To avoid false alarms, the threshold must be chosen larger than zero, though this reduces the sensitivity to faults (Frank, 1990).

In the dedicated observer scheme for Instrument Fault Detection (IFD) by Clark (1978), each estimator is driven by a different sensor output and the complete or part (as is possible) of the output vector is estimated. With this scheme multiple simultaneous faults can be principally detected and isolated by checking properly structured sets of estimation errors, for example, with the aid of threshold logic.

The generalised observer scheme (Frank 1987), uses an estimator dedicated to a certain sensor, to drive all outputs except that of the respective sensor. However, this IFD scheme is only able to detect and isolate a single fault in any of the sensors.

A duplex sensor system uses two identical sets of instruments; each supervised by an IFD scheme (Deckert et al., 1977; Onken and Stuckenberg, 1979). When a fault occurs, it is detected by using observer schemes, and the system switches to the healthy sensor. Therefore, hardware redundancy is used to detect the fault, while analytical redundancy is used for the fault isolation.

When a system is non-linear, the model in the observer is non-linear so as to avoid modelling errors arising from linearisation. Non-linear estimators can then be used for FDI. Another approach is to use non-linear decoupling (Wünnenberg and Frank, 1990). This leads to a state estimation error equation that is linear in the estimation error space, which is easy to evaluate.

The Fault Detection Filter (FDF) is a full-order state estimator with the feedback gain matrix being chosen such that the residual vector has certain directional properties when a fault occurs (Wilbers and Speyer, 1989). A fault is detected when one or more of the residual projections along the known fault direction or in the known fault plane are sufficiently large. The appealing feature of the FDF is that the size, time or history of the fault does not affect the residual direction for the fault. However, this approach does not account for disturbances, parameter variations or measurement noises and therefore requires precise modelling. As a result, this method has not been widely used for fault detection.

Note:

The following conclusions and comments have been compiled and deduced by the author, from the various journals and literature that have been obtained. These remarks cannot be strictly applied to all adaptations of the original method, but are general. Furthermore, these remarks have originated from the results of applications which use the method, rather than what the method has been said to be or prove, in theory. It must be added, that the methods performance and properties vary according to the application and the design of the method. This format will continue for all the other methods of fault detection and diagnosis that are described.

Conclusions/Comments:

- Robust to parameter variations and noise (accurately modelled as Gaussian processes).
- Cannot detect sensor failures reliably.
- Implementation becomes complex when detecting multiple faults.
- Not very sensitive to fault effects and modelling uncertainties.
- Low computational burden and can be easily implemented.
- Unable to detect multiple simultaneous faults.
- Prompt detection with bad to average isolability.

2.7.2 Parameter Estimation

This method detects faults by estimating the parameters of the mathematical model by the following procedure (Isermann, 1984; Frank, 1990). The parametric model of the system, which is usually a linear model with lumped parameters and input/output differential equations, is chosen. The relationships between the model and physical parameters are then determined. The model parameter vector using the input and output variables of the actual system is formed. The physical parameter vector is then determined. Next, the vector of deviations from the nominal value taken from the nominal model, is calculated. A fault is recognised by exploiting the relationships between faults and changes in the physical parameters.

Conclusions/Comments:

- Sensitivity to parametric faults is dependent on the current process state.
- Process excitation is required.
- Assumes sensors are failure free.
- Does not always detect faults correctly when the process parameters do not change.
- Detects faults correctly in processes that have changes in their parameters.
- Detects faults promptly.
- Unable to detect multiple simultaneous faults.
- Requires accurate mathematical models.
- Insensitive towards disturbed measurement data.
- Has a low computational burden and is easily implemented.

2.7.3 Parity space

This method checks the parity of the mathematical equations of the system (analytical redundancy relations), by using the actual measurements. The two forms of analytical redundancy are:

- Direct redundancy – which are the relationships between instantaneous redundant sensor outputs (algebraic relations).
- Temporal redundancy – which are the dynamic relationships between sensor outputs and actuator inputs (differential or difference equations)

A fault occurs once the pre-assigned error bounds are surpassed by the actual process variables by the use of parity checks. In the direct redundancy relation, the parity equations indicate the errors due to the faults, independent of the directly measured values. To detect a single fault among p components, at least $(p-1)$ parity relations are required (Chow and Wilsky, 1984; Lou et al., 1986). Using the temporal redundancy relations, the relation is a simple input – output model for a part of the dynamics of the system. This implies that the most reliable relations can be selected, since the consistency of the parts of the model can be checked, instead of the model as a whole (Frank, 1990).

Conclusions/Comments:

- Flexible in design of its structure, which accounts for its ability to detect faults accurately, but detects spurious faults.
- Has high sensitivity towards noise.
- Unable to detect multiple simultaneous faults.
- Implementation is complex when detecting multiple sensor and actuator faults.
- Has a low computational burden.
- Is robust and has an average isolability.

2.7.4 Adaptive Observer

Since only one paper has been found that describes this hybrid method, together with an application, the method is described with reference to the application in the following text.

The combination was developed so that all possible faults could be detected on a two-pump system. Furthermore, the model of the two-pump system needed to be robust with respect to model uncertainties and unknown disturbances. The observer part of the scheme was used to calculate the disturbances on the system and hence find the approximated, associated error. The observer was then designed to replicate the plant. This resulted in the final scheme becoming more sensitive to faults, since the thresholds set on the residuals became narrower, hence resulting in smaller sized faults being allowed to exceed the threshold. The observers were used to validate the results of the parameter estimation. Furthermore, model parameters did not adversely affect the residuals. Parameter estimation was performed on steady state data only, because the sampling period of the available plant data could not permit dynamic analysis. Only process and actuator faults were considered on the system. Results showed

correct detection of the tested faults. Small faults were also correctly detected, with the system being intolerant to changes in the state of the plant (Dalton et al., 1996).

Conclusions/Comments:

- Unable to detect multiple simultaneous faults.
- Able to detect faults promptly.
- Has good isolability and average sensitivity towards noise.
- Can be easily implemented.

2.7.5 Adaptive Parity Space

This method uses the parity space approach as a temporal analytical redundancy (i.e. the static and dynamic relationships between the actuator inputs and the sensor outputs). The formulation and calculation of the parity spaces are done in either the continuous or discrete time domains. It is assumed that the parameters do not change when a fault is not present. The structure is designed so that at least one measurement will not affect a specific residual. A faulty measurement will result in the decoupled residual remaining small, but the other residuals will increase in value. The residuals are then recursively updated by a parameter estimation scheme. As a result, a residual is formed for each possible fault, thus allowing simultaneous fault detection.

Conclusions/Comments:

- Detects faults slowly.
- Able to detect multiple simultaneous faults.
- Insensitive to noise and has good isolability.
- Implementation is complex.

2.8 Signal Model-Based Fault Detection Methods

These schemes are able to detect faults left undetected in process-model based method. Such faults include vibrations from rotating machinery and electrical circuits, and are termed vibration faults. These methods can also detect incipient and process type faults, but cannot detect all classes of faults. However, they are unable to detect actuator or sensor faults and abrupt faults. These methods are also robust, reliable and do not need *a priori* information. The three methods described are Spectral analysis, Bandpass filters and Maximum entropy estimation.

2.8.1 Spectral Analysis

The first step of this method requires the output signals of the system to be sampled. To reduce the influence of possible offsets, the difference between the actual and last sampled data, is calculated. The Fast Fourier Transform (FFT) is then taken of a certain number of samples. In the fault-free case, all the orders of the Fourier transform are zero. So, a fault occurs when the orders of the transform are nonzero.

Conclusions/Comments:

- Quick computation time with easy implementation.
- Is very sensitive to noise but has bad isolability.

2.8.2 Bandpass Filters

With this method, the output signals of the model are sampled and then mapped onto the frequency domain. These discrete mappings are then passed into one or more bandpass filters. The resulting signals are indicative of the faults that have occurred. The frequency range of the bandpass filter must be carefully chosen to allow for sensitivity to faults and for false alarms.

Conclusions/Comments:

- Is very sensitive to noise and has bad isolability.
- Has a low computational burden and can be easily implemented.

2.8.3 Maximum-entropy estimation

Since only one paper by Neuman in 1991 was found on this method, the description of the method together with the application is given. An Auto Regressive Moving Average (ARMA)-type signal model is used to model the sampled signals. The parameters are estimated by a two-step identification technique called maximum entropy (Papoulis, 1984). This technique estimates the amplitudes of the signals and the process parameters. The signal amplitudes are based on the solution of the undamped oscillation's homogenous differential equation. The autocorrelation function is used to eliminate the noise, instead of the sampled signal itself. The estimated parameters are used to describe the significant eigenfrequencies of the measured signal – which are represented by frequency loci and concatenated amplitudes. These parameters are compared to reference sets, in order to determine if faults have occurred. The deviations that occur from the comparison (due to faults) are fed into a diagnostic system. The diagnostic system determines the reason and location of the developing fault automatically. The efficiency of the method was tested on a metal hacksaw machine. All faults were correctly detected and diagnosed. The errors in frequency and amplitude were less than 1%.

Conclusions/Comments:

- Is insensitive to noise and has good isolability.
- Statistical properties have to be strictly adhered to for good fault detection.
- Has a heavy computational burden.

2.9 Methods Of Change detection

If the characteristic quantities or features (residuals) show stochastic behaviour with mean values and variances, then the deviations from the normal behaviour must be detected by change detection methods. All these methods require *a priori* information and their algorithms are in general, complex to implement. Furthermore, they all have very similar results (which show reliability and fair sensitivity to noise) in their fault detection since they are all statistical tests that differ narrowly. They can all be used for both on-line and off-line fault detection, and can easily be adapted to suit specific processes. However, they are unable

to detect vibration faults, but can detect actuator, process and sensor faults which are either incipient or abrupt. All classes of faults can be detected, but simultaneous multiple faults cannot be detected. Lastly, the change time of the symptoms and the sample path of the past observations determine the speed of detection (Basseville and Nikiforov, 1993). Such methods include Direct parallel testing, Likelihood ratio tests, Bayes decision, Multivariate test, CUSUM algorithm and Compound scalar test.

2.9.1 Direct Parallel Testing

With this method, a separate test is applied to each element of the residual vector. A signature is then generated – this is a logical pattern showing which residuals can be considered normal and those that can indicate faults. This test is easy to apply to the residuals in an input/output model because the residuals in a fault-free system are the outputs of a discrete moving average (non-recursive) system driven by the noise. This implies that the fault-free distribution of the residuals can be derived from the noise distribution. A covariance matrix consisting of the fault-free residual variances on the diagonal is generated from the noise covariances. The thresholds of each residual can be calculated using normal distribution of the variances and a selected level of confidence (usually 99% to prevent false detection). If the noise covariances are unavailable, then the residual variances may be estimated from observation of fault-free periods. In this case, t-distribution is used (Vaclavek, 1974).

A disadvantage of this method, is that the additional information in the off-diagonal elements in the covariance matrix, is not utilised (Gertler, 1988).

Conclusions/Comments:

- Statistical properties have to be strictly adhered to.
- Is not robust and can misdetect faults, but is reliable.
- Is complex to implement.

2.9.2 Likelihood Ratio Tests

The likelihood ratio is a ratio formed between the probabilities of the hypotheses of nonzero residual mean and zero mean vectors. The method then uses the natural logarithm of this ratio, so that a decision can be made as to whether a fault has occurred or not. If the residuals

are independent and normally distributed, the logarithm likelihood ratio is the difference of two sums (Gertler, 1988).

The sequential likelihood ratio test is when the likelihood ratio is compared to two limits (normal/faulty), with a gray range between them. When the ratio is in this gray range, there is no decision, and further innovation values are taken. The numerical value for this innovation can be found by maximum likelihood estimation from the innovations (Wilsky and Jones, 1976) or is a pre-set nominal value (Chien and Adams, 1976). A difficulty in this method is the accumulation of normal state information in the likelihood ratio before the situation changes from normal to possibly faulty. Chien and Adams proposed that the accumulations be automatically reset to zero. Uosaki (1985) proposed a backward evaluation of the ratio starting with the last residual vector, until a decision is reached. Wilsky and Jones included an explicit, but unknown variable for the time of change, thus making the algorithm more complex, but problem formulation more precise.

The test limits are obtained either from specified probabilities for false/missed alarms (Uosaki), or from the expected time between two false alarms and the expected time of detecting a fault. Running a set of parallel tests, with different alternative hypotheses based on the same innovations sequence enhances failure isolation (Gertler and Yin, 1996).

Conclusions/Comments:

- Are not robust and at times misdetect faults and detect spurious faults, but are reliable.
- Are simple algorithms to implement.
- Lack of *a priori* information can cause misdetection of faults (however, does not need as much as other methods).

2.9.3 Bayes Decision

This method utilises an *a priori* probability distribution for the detection of faults. An *a priori* distribution is obtained from the heuristic knowledge of the plant or the distribution may be assumed as design parameters. The non-linear filtering scheme (Loparo et al., 1986; Eckert et al., 1986) models faults as sudden shifts in parameters ruled by a random jump process with an *a priori* known distribution. System models are determined for each of the different failure modes. The non-linear filter calculates the conditional probability of the

system operating in each of the failure modes. Use of Kalman filters and the solution to a stochastic differential equation obtain reasonable approximations to the problem, since no finite solution is possible. The algorithm is complex, but shows good robustness properties (Gertler, 1988).

Conclusions/Comments:

- Is robust, reliable and has good isolability.
- Is complex to implement and has heavy computational burden.

2.9.4 Multivariate Testing

The elements in a fault-free residual vector are not independent of each other, even if the noise is white and uncorrelated. Newman and Perkins (1979), used this additional information to describe any constant probability density of residual vectors in space, as a closed hypersurface (constant 'isotherm'). Each hypersurface is a selection of a level of confidence. A fault occurs when any residual vector is outside the limit of the surface. This method becomes difficult when dealing with high-dimensional residual vectors. The main disadvantage of the method, is that it provides a faulty/not-faulty decision, which does not facilitate the isolation of the fault (Gertler 1988).

Principal Component Analysis (PCA) is used to model data clusters formulated by process operation caused by various disturbances. Data is assumed to be ergodic and stationary. This method is not sensitive to low levels of autocorrelation. The method uses linear combinations of original correlated measurements to make new uncorrelated variables (Jackson, 1991; Kresta et al., 1991). PCA is performed by eigenvalue decomposition of the covariance matrix of the original measurements. Effectively, PCA reduces the dimensionality of the process data. PCA is used to draw conclusions about the probability of different events. The boundary between likely and unlikely events is viewed as a statistical confidence threshold that can be used to identify faults (Raich and Çinar, 1996).

Non-linear PCA (NLPCA) is used for highly non-linear processes. The main difference between NLPCA and PCA is the introduction of a non-linear mapping between the original and reduced dimensional space. Linear PCA minimises the sum of the orthogonal deviations between a straight line and the data, while NLPCA summarises the data by a smooth curve.

Conclusions/Comments:

- Does not facilitate the isolation of the fault and is unreliable.
- Able to detect actuator, process and sensor faults fairly quickly.
- Can misdetect faults at times.
- Is complex to implement.

2.9.5 CUSUM algorithm

This algorithm is lesser known by its expanded name, Cumulative Sum (Basseville and Nikiforov, 1993). It is a hypothesis test of the generated symptoms of the system. The test indicates a fault when there is a change in the symptoms as defined by the hypothesis. CUSUM is well known for detecting abrupt and incipient changes in the stationary Gaussian case, increasing or decreasing in mean or variance of the Gaussian distributions (Boudaoud et al., 1996).

Conclusions/Comments:

- Is reliable but not robust.
- Lack of *a priori* information can cause misdetection of faults.
- Is simple to implement.

2.9.6 Compound Scalar Testing

A single scalar statistic is formed by the multiplication of the residual vector with its transpose and covariance matrix (Romagnoli and Stephanopoulos, 1981). If the noises are normally distributed and there are no faults, then the scalar statistic follows the chi-square distribution with μ degrees of freedom (μ is the number of residuals). The test provides a single binary inference for the method. A sequential procedure was proposed, where equations are systematically eliminated until the test indicates a fault-free system. An important component of the algorithm was the recursive generation of the statistic, following each elimination step, without complete computation.

Conclusions/Comments:

- Is not reliable or robust.
- Can misdetect faults and detect spurious faults.
- Is simple to implement.

2.9.7 Other Methods

The following methods have been stated for completeness. Fault detection application literature on them has not been found, although some theory has been found on Elementary algorithms, but not on the Run-sum test or Two probe t-test. These methods are probably not being used extensively.

1. Elementary Algorithms

The following is a brief outline from Basseville and Nikiforov. There are four methods, which are considered to be part of this section:

- Shewhart Control Charts
- Geometric Moving Average Control Charts
- Finite Moving Average
- Filtered Derivative Algorithm

These methods of change detection differ narrowly. Most of the algorithms use samples of data with fixed size. The general methodology for the methods is given.

At the end of each sample, a decision is made as to whether a fault has occurred or not. For a fault to occur, the hypothesis test on a parameter must be greater than the residual value. The parameter is the sum of the log-likelihood ratios of the observed signal.

2. Run-sum test

3. Two-probe t-test

2.10 Fault Diagnosis Methods

Fault diagnosis is performed by manipulation of the residuals. All the methods are reliable. The following three methods are described: Geometric distance and probabilistic methods, Neural networks and Fuzzy classification.

2.10.1 Geometrical Distance and Probabilistic Methods

These methods are commonly used with statistical multivariate methods of fault detection (e.g. PCA and partial least squares (PLS)). The concept of quadratic-distance based discriminants is derived from the multivariate Normal probability distribution. To choose the most likely source population for an observation, the probability of it being in one of the known classes can be calculated and compared. An alternative assumes that the residuals contain the discrimination information, instead of the scores (observed samples of the transformation matrix). An example is when known groups contain variation in disjoint sets of variables, the most likely class for an observation would be the class yielding residuals farthest inside the threshold. Classification is improved by combining score and residual discriminants, since both have restrictive assumptions (Naes and Isaksson, 1991). The combined discriminant uses distance information from both scores and residuals, as well as the relative sizes of the different disturbance regions. If a group of observations fail to fit within any of the known groups, it becomes a new group and is added to the discrimination scheme. Correct diagnosis is made when the discriminant for the true group has the minimum value (Raich and Çinar, 1996).

The angular discriminant is the minimum angle between the test point and the mean of the group, with the vertex positioned at the mean of the normal operation region (NOR). Decision boundaries for angular discriminants describe open-ended conical regions in space, while distance-based discriminants enclose elliptical regions. The difference between these types of boundaries may be especially important to diagnose points extending beyond the ellipse and those just leaving normal operation. Diagnosis with each new sample is done by selecting the group with the minimum angle to the sample (Raich and Çinar, 1996).

Conclusions/Comments:

- Needs strict statistical properties in the symptoms, otherwise diagnosis becomes poor.
- A large number of faults can be diagnosed.
- Isolability ranges from bad to good, dependent on the types of faults that need to be diagnosed.
- Unable to diagnose multiple simultaneous faults.
- No training is required.
- Computational burden is heavy.

2.10.2 Neural Networks

The following is a brief outline from Maki and Loparo, 1997; Kavuri and Venkatasubramanian, 1993. The neuron is the fundamental element in this method. The neuron has multiple inputs and a single output. Each input is multiplied by a weight, then all inputs are summed and operated by the transfer function of the neuron to generate the output (activity level). Each application has a different number of neurons for the input and output layers of the neural network. The number of neurons in the hidden layer must be adjusted during the learning phase, so that the network can be trained properly. The neural network receives information on its inputs to produce information on the faults. Neural Networks (NNs) are characterised by their choice of activation function, squashing function and error criteria. The standard feedforward NN uses a linear activation function (or fan-in function), which is a linear, weighted summation of the values coming in from the input connections. It serves to map the inputs into a scalar, which is then delimited between the bounds through the use of a saturating non-linear function – called the squashing function. The network must be trained with target patterns of the faults or combinations of the faults in order to diagnose all the different types of faults correctly. Although the NN is able to learn samples accurately, its reasoning conclusions are ambiguous.

Conclusions/Comments:

- Require optimal training of the network for good diagnosis.
- An increase in the number of training patterns that have to be learnt decreases the accuracy.
- Is able to diagnose many faults simultaneously.
- Potentially very fast (in parallel hardware implementation).
- Can learn from experience so can better diagnose faults with time.
- Has complex implementation and heavy computational burden.

2.10.3 Fuzzy Classification

Faults often show fuzziness when it is very difficult to describe the relationship between the phenomena and the reasons for faults. Zadeh (1965) suggested that fuzzy set theory be utilised to deal with uncertainties. In fuzzy set theory, the membership grade function is proposed to translate uncertain or qualitative information into quantitative data (Zadeh, 1983; Negoita, 1985). In this section the fuzzy classification method of fault diagnosis is described, this is followed by a description of a fuzzy model-based method of fault detection and diagnosis (Maruyama et al., 1996).

2.10.3.1 Fuzzy Classification Fault Diagnosis

This diagnosis scheme identifies faults by comparing a model describing the current behaviour of the system with a set of reference models. For each reference model, a degree of similarity is calculated using fuzzy rules. The reference models can either be computer simulated (which may lead to ambiguities in fault diagnosis to real applications), or a learning algorithm can be implemented in normal plant operation to obtain fault-free models. Fault diagnosis is then achieved by statistical or probabilistic analysis and testing of the degree of similarity (Maruyama et al., 1996).

Conclusions/Comments:

- Test data needs to be close to steady state.
- Diagnosis is correctly performed when the generated symptoms are significantly different.
- Gives correct results when there are few measurements and accurate system models.
- Able to diagnose multiple simultaneous faults.
- Unable to diagnose many faults.
- Training is required and implementation is simple.

2.10.3.2 Fuzzy Model-Based Fault Detection And Diagnosis

The system data is first pre-processed to remove noise and disturbances. A rectangular window was used by a transient detector to calculate the average activity of each of the measured variables. This is done to locate the sections of the data close enough to the static mode's steady state. A recursive least squares algorithm was used to estimate the positions of output member functions for the training data. Fault detection is based on two models of correct operation, one specific, while the other generic. A fault is detected when the residuals (generated by the model of the correct operation) exceed a certain threshold, which varies with the operating point to take modelling errors into account. The faults were predicted using singleton fuzzification, product inference and centroid defuzzification. The faults were diagnosed by comparing a partial model, describing the current state of the system, with a set of references. Fault detection and diagnosis was performed on the cooling coil sub-system of a full-scale air conditioning test facility. The results showed that fault detection was correctly performed, although the diagnosis was not correct, unless the symptoms generated were significantly different. This is due to the fact that the test data needed to be very close to steady state. This method could be advantageous in applications where few measurements are available and accurate models of the system are unavailable (Maruyama et al., 1996).

Conclusions/Comments for fuzzy model-based fault detection include:

- Able to detect all types and classes of faults, except vibration faults.
- Unable to detect multiple simultaneous faults.
- Is insensitive to noise, robust, reliable and has good isolability.
- Does not require *a priori* information.

2.11 Non Model-Based Methods of Fault Detection and Diagnosis

Non model-based methods of fault detection are lesser used than model-based methods of fault detection and diagnosis. Their obvious advantage over model-based methods is that they do not need a model of the system. Non model-based methods are more likely to be used in complex processes where there are many faults and combinations of faults that need to be detected and diagnosed. The traditional methods of non-model based methods of fault detection, (which have since become redundant) included:

1. **Limit Checking** – a fault is indicated when the plant measurements exceed a given threshold. Many systems contain two thresholds of which the first serves as a warning and the second triggers emergency action.
2. **Special Sensors** – which perform limit checking in hardware (such as limiting temperature), or measure special variables (e.g. vibration).
3. **Multiple Sensors** – which act as a form of physical redundancy. The multiple sensor measurements of the same variable are compared. If a discrepancy arises in the measurements, at least one sensor failure can be detected and isolated.

The rest of this section will describe current methods used. It must be noted that these methods are reliable, able to detect and diagnose multiple simultaneous faults, *a priori* information is necessary in these methods. Neuro-fuzzy networks, Neural networks (two-stage), Intelligent sensors and Expert systems will be described.

2.11.1 Neuro–Fuzzy Networks

Neuro–fuzzy networks can be used in a variety of ways for fault detection, fault diagnosis or for fault detection and diagnosis. The structure of a neuro-fuzzy network is shown below. The description of neuro-fuzzy networks in each of its three uses, with applications follow.

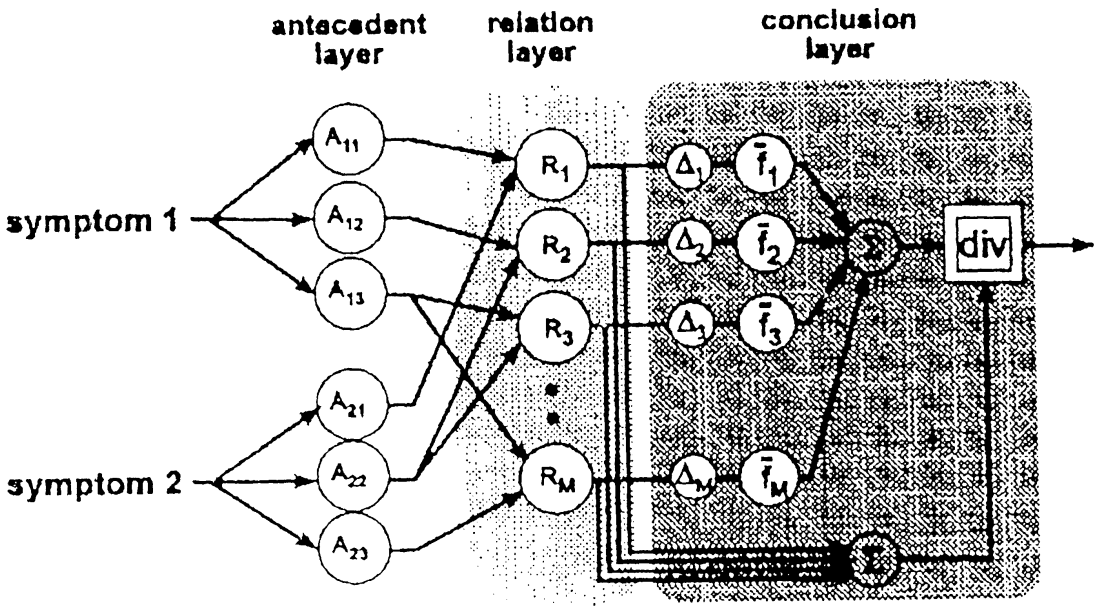


Figure 2-3 Structure of a neuro-fuzzy network (adapted Fig 8. Halfmann 1996)

2.11.1.1 Neuro-Fuzzy Fault Detection

A neuro-fuzzy fault detector is used to detect bearing wear and winding insulation failure in a split-phase squirrel cage inductor motor, as described by Goode and Chow, 1995a and 1995b. The method of failure detection consists of two modules – **a.** and **b.**, with results described in sub-sections **c.** and **d.**.

a. The Fuzzy Membership Function Module

This module is used to obtain fuzzy membership functions of the motor current and the rotor speed. These membership functions provide heuristic information in the form of grades (Zadeh 1965). These grades are then able to assign low, medium and high values of current and speed to conditions: bad, fair and good. The fuzzy membership function module is composed of 2 independent sub-networks. One network uses normalised motor current as input, while the other has normalised rotor speed as input. The measurements are obtained using a current sensor and a tachometer. The sub-networks partition the normalised values into fuzzy membership function space, which then become the outputs of the modules. The information for the fuzzy membership functions are contained in the weights of the sub-networks. The weights also determine the shape of the membership functions. Sub-networks

allow for complex membership function representation (Lin and Lee, 1990), which result in a flexible adaptation for decision classification. The neuro-fuzzy system adaptively determines the membership functions, but a good starting point will help the training of the system. Different membership functions were evaluated to find a good initialisation (by Goode and Chow). The final form was a triangular or trapezoidal shaped membership function with between 20% and 45% overlap. The outputs of the sub-networks are then normalised values of the motor current and rotor speed in the range [0, 1], in terms of the low, medium and high fuzzy membership functions.

b. The Fuzzy Rule Module

The fuzzy rule module provides the rule-based statements that govern the linguistic operating ranges of the inputs (antecedents) to the output consequences that would detect faults (as being in conditions: bad, fair and good). The structure of the module is a two-layer feedforward NN. The input layer nodes are fuzzy membership functions, which are also the conditional part of the fuzzy rules. The output layer nodes indicate the consequences. A 'best guess' was used to start the fuzzy rules training. The final rules were determined through training of the neuro-fuzzy system. The fuzzy rules information is contained in the weights connecting the antecedent nodes to the consequence nodes. If a positive weight connects the antecedent node to the consequence node, then that is the consequence for that particular antecedent. A negative weight connecting nodes indicates that the consequence does not belong to that antecedent. The generalised delta rule is used as the learning algorithm to backpropagate the error through the network. Therefore, the weights connecting the correct consequence for an antecedent will continue to increase positively in value, whereas the weights connecting the incorrect consequence for an antecedent will continue to decrease negatively in value. So, the consequence node (of which there should only be one) to which the positive weight connects, is the correct consequence for that antecedent.

The modules are then trained with initial values of the weights. If the training results in incorrect fuzzy rules and false information, the weights are reinitialised with different values. Once the training has been completed successfully, the unnecessary antecedent rules in module two are removed. These unnecessary rules are those contained in the weights, that have changed sign after the training. The network is then retrained until all of the weight values increase in absolute value. The results for the scheme were:

c. Motor Bearing Wear

The neuro-fuzzy bearing fault detector was trained and classified 100% of the training data, but only 93 % of the testing data. This was because the 'medium' membership function for the rotor speed violated the heuristic reasoning behind the fuzzy logic membership function. The network was then retrained allowing the rules of the rule module to change. However, some consequences that were not rules tried to become rules, while some rules tried to change from one consequence to another, resulting in incorrect classification. The rules were then changed and the network retrained, which then resulted in 98% classification of the testing data. After the unnecessary rules had been removed, there was a 100% classification of the testing data. The Root-Mean-Square (RMS) error converged to 0.0064 after 5000 iterations.

d. Motor Insulation Wear

The fault classifications were based on the life expectancy of the insulation as the stator winding temperature increased. The neuro-fuzzy network was trained, from which results showed that insulation wear was caused by the motor current only. There was a 100% classification of training and test data. The RMS error was approximately 0.089 after 2000 iterations.

Conclusions/Comments:

- Able to detect all types of fault including those that are abrupt and incipient.
- Able to detect multiple simultaneous faults.
- Is reliable, robust and insensitive to noise.
- Has good isolability.
- Requires *a priori* information.
- Had a heavy computational burden and is complex to implement.

2.11.1.2 Neuro Fuzzy Fault Diagnosis

The fault detection network was developed through principal component decomposition, which results in a model for the process under nominal operating conditions. A fault is detected once the fault signal exceeds a certain threshold. A neuro-fuzzy network is a conventional feedforward network with a fuzzification layer. This network is developed from normal and faulty process operating data and is able to locate the faults. The training algorithm was based on the non-linear iterative PLS algorithm (Wold, 1966). The

fuzzification layer converted increments in the online measurements into three fuzzy sets: increase, steady and decrease, by association with three types of neurons in the fuzzification layer. The activity functions for the neurons were: sigmoidal, Gaussian and complement sigmoidal for increase, steady and decrease, respectively. As a result, the training data was minimised and robustness increased. This method was applied to a simulation of a continuously well-stirred tank reactor (CSTR). Eleven possible faults were considered for the experiments. The same training sets and methods were applied to a conventional feedforward network, using the back propagation training algorithm. Faults were tested on both systems – all of which were correctly detected. However the conventional network was not able to correctly diagnose all the faults as the neuro-fuzzy network had done (Zhang et al., 1996a).

Conclusions/Comments:

- Determination of the activity function can lead to a robust scheme.
- A relatively small amount of training data is required.
- Correct diagnosis of multiple simultaneous faults is possible.
- Able to diagnose a large amount of faults quickly.
- Computational load is large and implementation is complex.
- Isolability is good.

2.11.1.3 Neuro Fuzzy Fault detection and diagnosis

The following has been described by Chow et al., 1993. Briefly, the Duke Power Company has used neuro-fuzzy networks to diagnose faults in their power distribution, in particular those faults caused by animals. There were 22 items of information collected for each fault that had occurred, including the expert knowledge of the repair crew. Of the 22 items of information, five essential items concerning animal caused faults were selected. These five items were pre-processed, to obtain fuzzy conditional probabilities (using Bayes' Rule) for input to the NN. This meant that the conditional probabilities were dependent on the history of the faults in the distribution system. The five information items were expanded to seven as input to the NN. The NN then mapped the non-linear probabilistic relationships of the input variables to the output variable. The NN was trained with 734 training samples, using the backpropagation algorithm. The NN had fifteen hidden layers. Results showed that the NN learned the training data with 99% accuracy after 200 iterations, after which the mean squared training error had become negligible. When trained with 1337 training sets, the accuracy was

98%. This meant that the errors were due to other faults aside from animals, or faults that had not been detected. It was suggested that this method could easily be generalised for other fault detection and diagnosis.

The conclusions/comments section for neuro-fuzzy fault detection and diagnosis is a combination of those described under the fault detection and fault diagnosis sections.

2.11.2 Neural networks (two-stage)

Even though neural networks are classified as a model based method of fault diagnosis, there exists a two-stage neural network method used for fault detection and diagnosis (Maki and Loparo, 1997), which does not make use of the system model. The following describes the neural network method of fault detection and diagnosis with reference to the one application it has been used in.

A simulation of a CSTR is studied. The primary network is used to detect the transients of each measurement, dynamically, using data pre-processed by a moving time window. The secondary network detects and diagnoses pre-known faults. The back propagation algorithm (considered better than Hopfield, for training feedforward networks), together with momentum and the adaptive learning rate method, was used to train the networks with patterns for normal operating conditions and faulty conditions. The secondary network would then diagnose faults by comparing the current state with patterns it has learnt and then decide which pattern was best matched. The schematic diagram is shown on the following page.

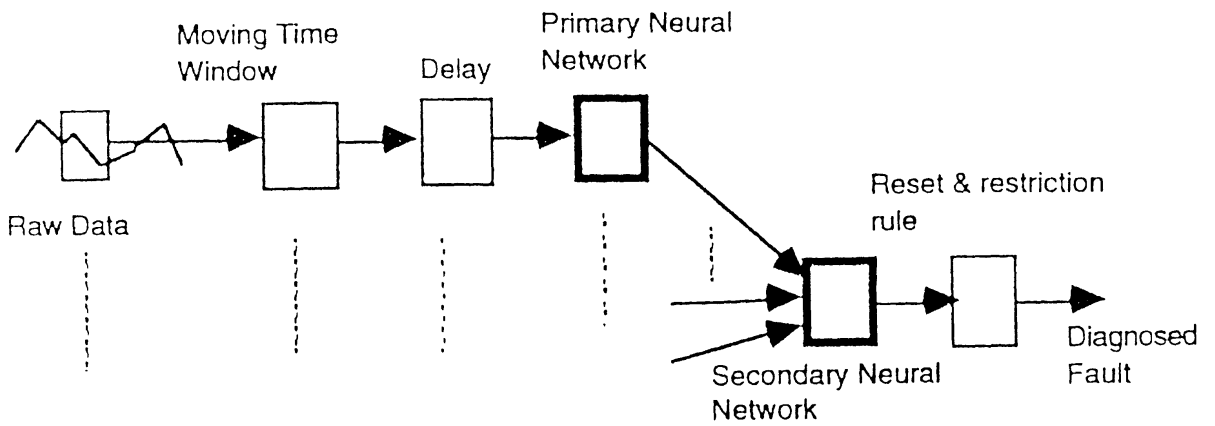


Figure 2-4 Schematic diagram of fault detection and diagnosis system (Fig. 3 Maki and Loparo, 1997)

The failures were diagnosed quickly and correctly, but as the number of trained fault patterns that had to be learnt, became greater, the correct diagnosis of the failures became lower. Furthermore, when unknown faults were initiated, the diagnosis was a normal operating condition. The secondary network can generalise on noise disturbances (up to a certain bound) and the severity of faults.

Conclusions/Comments of the fault detection part of this method include (the fault diagnosis properties for neural networks are the same as those described earlier):

- Can detect all types of faults quickly, which are either incipient or abrupt, except for vibration faults.
- Requires little or no *a priori* information.
- The system requires resetting after faults have occurred.
- Able to detect multiple simultaneous faults.
- Is robust.
- Implementation is complex and the computational burden is heavy.

2.11.3 Intelligent sensors

Although there are similar techniques such as that described by Duyar et al., 1996, the technique described in this section is that proposed by Henry and Clarke, 1991. The method is described with reference to an application.

A sensor provides information to the 'next level up' (NLU), which may be a control, alarm or monitoring system, or an operator. This method suggests that an intelligent sensor should interact with the NLU so as to provide enhanced information – not just the best estimate of the measurand, but an associated validity. This validity is approximated to have two principal components: 'uncertainty' (a real value) and 'status' (a discrete value of a specified set). These two validity components encompass fault detection and other factors to which the data will be processed. All types of sensor fault: bias, drifting, precision degradation and complete failure, should be detected and diagnosed.

The sensor performs conventional data processing to generate a Raw (i.e. invalidated) Measurement Value (RMV). The RMV is then validated to generate a Validated Measurement Value (VMV) and corresponding Validity Index (VI). If a sensor fault were to occur, then an abnormal RMV sequence would be generated. If the fault were to be correctly detected and diagnosed, then the best estimate of the true process value, based on recent history, would be used as the VMV. This value of the VMV and the VI (which would indicate that the value of the VMV was only an estimate) would be sent to the NLU for the duration of the fault. The self-validating sensor is shown in Figure 2-5.

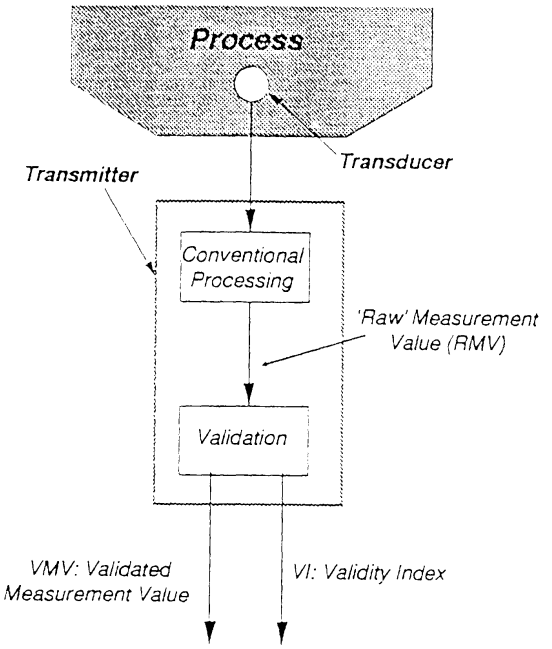


Figure 2-5 The self-validating sensor (Fig. 2 Henry and Clarke, (1991))

The VI properties require it to be: applicable to all measurements, have unambiguous meaning, be consistent in all situations, manipulated for all measurements, accurate and relate the sensor status at the time of Measurement Value (MV) generation. The VI should include the uncertainty associated with VMV, making sure that the uncertainty is increased when a fault occurs. Also included in the VI should be the confidence in the RMV according to the status of the MV, and according to the duration of the fault.

The information sources available to the diagnostic logic for the sensor are:

1.) **Raw Data:** the first signal available in the sensor, generally an electric image of a transducer output.

2.) **RMV Signal:** a signal that relates directly to process parameters.

3.) **Auxiliary Data:** other signals available within the device.

4.) **NLU Inputs:** generally discrete values and thresholds relating to expected process behaviour.

5.) **Hardware Errors:** discrete flags summarising raw data.

6.) **Product Database:** relations between signals and device characteristics over time.

The Diagnostic State Machine (DSM) processes all of the available information to diagnose the faults, using device-specific rules. The diagnosis derives all the parameters transmitted to the NLU (which could also be used for maintenance). The diagnosis is used to select the appropriate device and MV status values via a lookup table, thereby affecting the calculation of the VMV and Validated Uncertainty (VU).

The intelligent sensor scheme was implemented on a commercial mass flowmeter in both faulty and non-faulty conditions. The results proved to be correctly detected and diagnosed when a temperature sensor failed (Henry and Clarke, 1991).

Conclusions/Comments:

- Only one fault of a particular sensor can be detected at any one time.
- Computational burden is heavy.
- Validity of the information to be processed is given.
- Able to detect and diagnose multiple simultaneous faults.
- Isolability is good and is robust.
- No training or *a priori* information is required.

2.11.4 Expert Systems

Expert Systems (ES) are only used for fault diagnosis. The diagnostic system consists of 5 parts:

1. **Data Base** – the function of the database is to store and manage all of the test data.
2. **Knowledge Base** – the ES needs an effective knowledge base to store the expert knowledge. The knowledge base contains quantitative and qualitative knowledge. The quantitative knowledge is obtained from the analytic process models (e.g. residuals) and the qualitative knowledge from rules and soft computing.
3. **Reasoning Engine** – draws conclusions by quantitative reasoning (e.g. Boolean methods) or by qualitative reasoning (e.g. possibilistic methods).
4. **User Interface** – is used to exchange information between the operator and the machine.
5. **Work Space** – is an area of memory used to store a description of the problem constructed from the facts supplied by the user or inferred from the knowledge base.

Fault diagnosis is then performed by the reasoning engine, which uses information from the data base, knowledge base and user to make the decisions. The limitations of the ES development are:

- The ES knowledge base is usually imperfect because of the insufficiency of expert knowledge and the difficulty of rule translation. The imperfection shows that fault diagnosis is incorrect when none of the rules correspond to the fault phenomena.
- Since the fault diagnosis procedure is complex, ensuring the correctness of inference results is a difficult problem.
- The ES knowledge is not discreetly distributed over the network.

(Patton et al., 1989;Xu et al., 1997; Isermann, 1996).

The architecture of the expert system is shown in Figure 2-6.

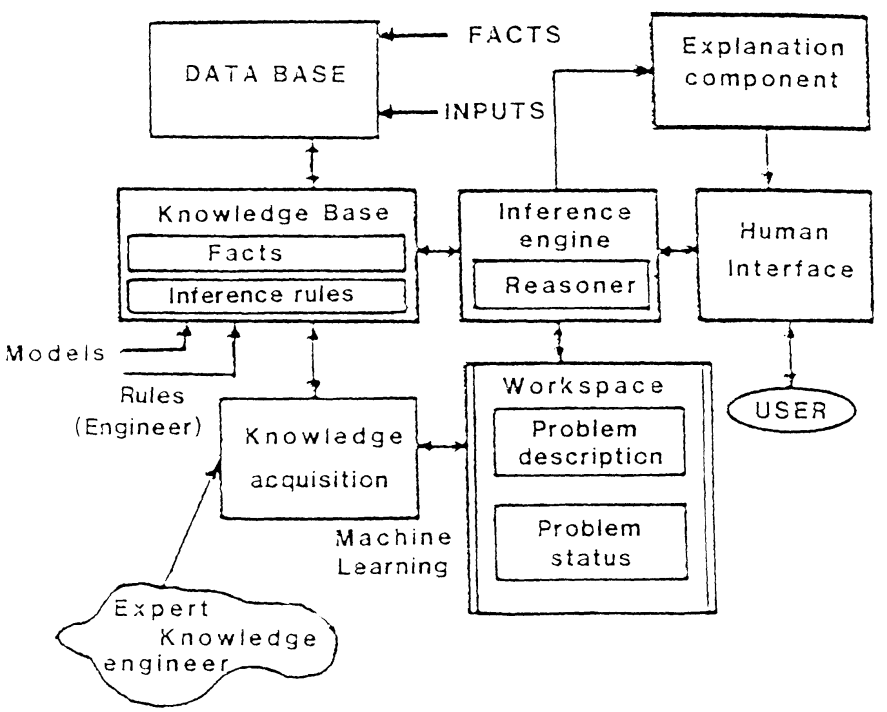


Figure 2-6 Expert system architecture (Fig. 15-1 Patton et al., 1989))

Conclusions/Comments:

- Highly skilled operators are needed.
- Development of the system is difficult and time consuming.
- Computational burden is heavy and implementation is complex.
- Reconfiguration is problematic when it is necessary.
- Able to diagnose a large number of faults quickly.
- Able to diagnose multiple simultaneous faults.
- Training is required.

2.12 Applications Of Fault Detection and Diagnosis Methods

In this section some of the various methods of fault detection and diagnosis are described with reference to applications that they have been used in.

2.12.1 Observer

The Sliding Mode Observer by Hermans and Zarrop, 1996 is used to generate robust residuals for fault detection. Extra conditions are specified on the observer to generate unbiased estimates of the system, despite modelling errors. These extra conditions involve Sliding Surfaces – which are situated where the output errors are zero. When the deviations of the model and the real plant are within a certain range, the residuals are virtually zero – indicating that no fault has occurred. In the sliding mode context, the observer is said to be sliding i.e. it is situated on its surface. When a fault occurs, the observer is disturbed and the sliding stops. The system is designed to detect actuator and sensor faults. However, when there are more actuators than outputs, the fault detection can be incorrect. The fault detection system was tested on a non-linear thermodynamic system, which is under feedback control. Mismatches between the plant and the model resulted from the use of a linear model. However, the sensor faults were correctly detected, both inside and outside of the robustness interval. Small drifts and offsets produced a few incorrect results when detected. Furthermore, a fault in one sensor caused spurious measurements out of other sensors.

2.12.2 Parameter Estimation

The following is an outline from Reiß, 1991. Additional sensors, such as impact-sound sensors and dynamometers are expensive and supervision with them is often unreliable. Furthermore installation of such sensors on machine tools is nearly impossible. Current and speed signals and the drive positions were then used for the supervision of the process. In the drilling process, the operational monitoring system had no information about the cutting conditions, the estimated parameter set of a new tool was used as a reference set. The heuristic knowledge base included the reference sets for all the different tools, including those used for various cuts. The values for tool wear were obtained by summing the multiplications of the deviations of the estimated parameters (relative to the reference sets), with weighting coefficients. Thus, various wear curves were obtained and added to the knowledge base. The process parameters were identified via RLS estimation. As a result, the tool wear state of a newly used tool could be determined, and the operator could determine the number of workpieces that could still be produced with that tool. Experimental results showed accurate results for the tool wear, as compared to the simulations that were carried out.

2.12.3 Parameter estimation with hypothesis testing

The fault diagnostic system was formed for a space shuttle main engine by Duyar and Eldem, 1991. The fault parameters were estimated by using a real time multivariable parameter estimation algorithm (Duyar et al., 1990). It was assumed that neither of the actuator or sensor or process faults would occur simultaneously. Process or measurement noise, modelling errors and parameter uncertainties were not taken into account in determining the mathematical model of the engine. Three different models of the faulty processes were developed using the parameter values of faulty sensors, actuators and processes. The diagnostic system had two steps. The first step was composed of three Hypothesis Testing Modules (HTM) - which were parallel processed to test each type of suspected fault. These modules compared the estimated fault parameters with the actual measured output, to find which faults have occurred, thereby generating the Standard Error of Estimate (SEE). The second step is the fault diagnosis module - which is used to check the information from the HTM, isolate the fault and determine its magnitude. This is done by:

1.) comparing the fault parameters against the predetermined signatures

2.) comparing the SEE's against the pre-selected thresholds

3.) comparing the relative magnitude of the SEE's among all the HTM's.

The diagnostic system was simulated for actuator and sensor faults. The diagnosis produced correct results. The advantage of the system is that *a priori* knowledge about the fault signatures was not required.

2.12.4 Parity space

This application introduces an Adaptive Cruise Control (ACC) feature for motor vehicles by Holzmann et al., 1996. The difference of the ACC from conventional cruise control systems, is that apart from keeping a fixed speed, the ACC provides distance control if a slower vehicle is detected ahead. The second part of the ACC control system is to supervise the lateral dynamics of the vehicle by using the parity space approach (which is what will be described in the rest of this section). The supervision is performed by cyclically checking the important sensor signals, and then warning the driver if critical limits are approached. The state-space model of the system was first transformed into discrete time, and the parity equations formed in accordance with a certain design procedure for each operating point. A low-pass Butterworth filter was used to reduce high frequency disturbances in the residuals. Two residuals were designed so as to obtain the maximum decoupling of the sensor and parameter faults (Gertler and Singer, 1990). When the system was tested, a critical situation was detected by both residuals when the vehicle reached a wet surface from a dry surface. A sensor fault was also correctly detected. However, the faults took longer to be detected due to the phase lag of the Butterworth filter. The method was robust and reliable, with good results.

2.12.5 Passive FDI system with neural networks

This method is based on the parity-space approach (Gertler, 1991) and adapted by Danai and Jammu, 1996. The primary residuals are classified into a set of transformed residuals via a Transformation Network that represents the dynamics of the system. The transformed residuals are then mapped by a Diagnostic Network into decision regions for diagnosis. The Transformation Network uses a batch of fault measurement based on the diagnostic results from the Diagnostic Network (through feedback), for iterative training. The data used for training is obtained from a perturbed model of the process that reflects the levels of uncertainty in the nominal model, disturbances are also added. A time-delay net is used to transform the residuals, and a multi – valued influence matrix (MVIM) (Danai and Chin, 1991), is used to estimate the fault signatures and to perform diagnosis. The time-delay net is tuned to improve uniqueness and repeatability of the fault signatures. After each iteration, the

detection capability of the transformed residuals is based upon the number of false alarms and undetected faults within the training set. The parameters of the time-delay net are trained to have better detection and diagnostic performance. The bootstrap adaptation (Widrow et al. ,1973) is used to train the net. This method was applied to a model simulation of a 1.6l engine using four-cylinder multipoint injection (Kamei et al., 1978). The diagnosability of the faults under the presence of bias, noise and modelling uncertainty was good, but could have been better if the training of the network was done until it reached the optimum.

2.12.6 Adaptive Parity Space

The parity space with parameter estimation scheme was tested on a throttle valve actuator driven by a DC motor by Pfeufer and Isermann, 1996. The mechanism is able to detect all of process, sensor and actuator faults. All the parameters were estimated in the fault free case by RLS estimation, in the continuous time domain. The identified process model was then the basis for the controller design and the fault detection scheme. The design of the digital controller was based on the parameterised mathematical model and from the information given by the deviations in the parameters. When faults occurred, the residual, deviated parameter and process signal parameters were passed to a diagnostic rule base, which detected and diagnosed the faults. Furthermore, when a sensor fault occurred, the controller was switched to a redundant sensor. All faults were correctly detected and diagnosed quickly, resulting in small controller error.

2.12.7 Spectral analysis & neural networks

Isermann et al., 1996 used the FFT to detect engine misfire. A four-cylinder diesel engine was used for this purpose, with the output speed signal being used to apply the FFT. Misfires were detected. Neural networks were used for fault detection of a turbocharger. A RBF network was trained (Leonhardt, et al., 1995) with the experimental test data. Dynamic Multilayer Perceptrons were used as process models to simulate the normally operating system, and some faulty operation modes. The fault detection of the 'fault' model was better than the 'nominal' model (only one fault was considered). For the detection of multiple faults, a static multilayer perceptron (MLP) structure was trained using a combined second order gradient method called the Levenberg – Marquardt algorithm. This algorithm includes the advantages of a gradient method and a Gauss – Newton method. All faults and

combinations of faults were diagnosed correctly, provided that the inputs provided sufficient information to the neural network.

2.12.8 Multivariate testing with geometric distance and probabilistic methods for diagnosis

Raich and Çinar, 1996 performed this application. The PCA model was tuned to discard the small variations of the residuals. Thresholds were calculated using Normal distribution and corrections were made for sampling effects to approximate fit to the normal distribution. Mahalanobis or statistical distance (Johnson and Wichern, 1992), and Euclidian and Mahalanobis angles were used to check whether the samples were within the score bounds. Score, residual, angular and a combination of score and residual distance ratios were used to find the discriminants for fault diagnosis. The method was applied to a simulation of a detailed chemical plant, including several measurements and lengthy data records. Four PCA models were created, but insufficient sample size caused the models to be statistically unreliable. Half of the faults were correctly diagnosed with step or ramp disturbances. For random disturbances, the angular information was in some cases, more reliable than distance measures in diagnosis. Residual, score and the combination discriminants were unsuccessful in fault diagnosis. Generally, the angular discriminants diagnosed fault more correctly than any other discriminant.

2.12.9 CUSUM algorithm with fuzzy classification

The fault detection system by Boudaoud et al., 1996 was based on both a learning and a supervisor module. The learning module acts as an adaptive fuzzy classifier (i.e. its parameters are updated online). The fuzzy-minmax algorithm, FMMC (Simpson, 1993), was chosen for the pattern learning. The supervisor module controls the learning periods of the classifier by making the adaptation of the classifier parameters on steady states. These parameters are saved without changes as evolutions occur (e.g. slowly or abruptly). The fuzzy Mahalanobis distance, FMD (Bezdek, 1981), was used to characterise the evolutions between different categories. A CUSUM algorithm (Basseville and Nikiforov, 1993), incorporated with the FMD, was the basis for the supervisor module. The diagnosis system is then able to detect new operation modes and slow or abrupt varying changes on the plant operation modes. This method was tested on a DC motor. Results showed that single faults were detected and diagnosed correctly, at any specific time.

2.12.10 Expert system with neuro-fuzzy networks

The following is a brief outline from Xu et al., 1997. Unlike the ES, the NN knowledge is discreetly distributed over the network and is based on sample learning. Although the NN is able to learn samples accurately, its reasoning conclusions are ambiguous. Combining the ES and NN eliminates the shortcomings of both diagnosis methods, thereby increasing the perfection of the knowledge base and the reliability of fault diagnosis. This combination is known as the consultative mechanism. NN is the main means of fault diagnosis, while ES is used to obtain new knowledge and make up for the deficiency of NN.

If ES is the main means for fault diagnosis, the knowledge base will be enlarged continuously. NN is then used for knowledge acquisition, but would need a greater number of training samples - which reflect many kinds of faults. The ES contains rules from the fuzzy rules, which are added to those rules learned by NN. However, this mechanism would have problems of contradictory inference, resulting in the incorrect diagnosis of faults.

The uncertainty characteristics of the diagnostic system are:

- **Numerical Uncertainty Translation** – is the process where test data, some expert experience and some criteria can be directly numerated into fuzzy numbers.
- **Linguistic Uncertainty Translation** – is the process by which some expert experience and some criteria, which are expressed in linguistic language (which cannot be numerated directly), are changed to fuzzy numbers.

The method of Numerical Uncertainty Translation is to establish suitable membership grade functions. If the function is not given by the ES, it can be based on fuzzy set theory. To obtain membership grade functions for Linguistic Uncertainty Translation, the relationship between 'true' and 'false' and its degree to which the fuzziness is reflected, needs to be expressed clearly. Some typical fuzzy languages are: completely false, very false, false, rather false, unknown, rather true, true, very true, completely true (Xu et al., 1997).
The diagnostic system consists of four parts:

1.) **Database** – the function of the database is to store and manage all of the test data.
2.) **Knowledge base** – the ES needs an effective knowledge base to store the expert knowledge. The knowledge base is obtained from the rules given by the Numerical Uncertainty Translation and the Linguistic Uncertainty Translation.

3.) **Reasoning Engine** – forward reasoning is adopted to control the inference process. This is done by, not considering the result corresponding to the negative premise, and by not inquiring about the identical premise more than once. The inference form is uncertain reasoning.

4.) **User Interface** – is used to exchange information between the operator and the machine.

The scheme was applied to a power transformer, using Dissolved-gas analysis (DGA), as the method used for the fault diagnosis. All faults were correctly and quickly detected. The problem that arose was that of contradictory diagnosis between the NN and the ES. Furthermore, a lot of training was required and few faults were tested, which resulted in some misdiagnosis.

2.12.11 Overview of applications

The results obtained for fault detection and diagnosis vary widely according to the methods used for the applications, and are by no means specific. Many adaptations of the methods described thus far have been described, with results and properties that differ from those stated in the descriptions of the methods.

The main contributor to bad results in model-based methods of fault detection, is the inaccurate modelling of the parameters. Another contribution is that as the number of faults to be diagnosed are increased, the lower is the rate of correct diagnosis. Many of the methods are only able to diagnose a few specific faults and not others. Most of the applications discussed have been performed on model simulations, which have taken into account the effects of noise and various uncertainties, which can be closely approximated to real processes. On the other hand, some applications have not taken the various disturbances into account, and have obtained good results, but in practice, these results should not be as good. In some applications, the results that have been obtained do not correspond to that predicted by theory. Furthermore, the objectives of some of the methods were not achieved by the results obtained. One cannot analyse which method for fault detection or fault diagnosis, is necessarily the best method, but can see which combination of fault detection and diagnosis is most likely to suit a specific process.

2.13 Trends in fault detection and diagnosis

Observers and parameter estimation are the two most commonly used methods of fault detection. Observers are mostly used to detect actuator faults, followed by the parameter estimation and neural networks methods of fault detection. Parameter estimation is mostly used for the detection of process faults. Nevertheless, the importance of neural network based and combined methods for fault detection is steadily growing. For fault diagnosis, neural networks and fuzzy rule-based logic are mostly used. Analytical redundancy based methods have their best application areas in mechanical systems, where the process models are relatively precise (Isermann and Ballé, 1996). A summary of published applications on fault detection and diagnosis is given in Appendix A.

2.14 Summary

From the literature review presented, the author has been able to deduce and summarise the properties of the fault detection and diagnosis methods in the following two tables. However, it must be noted that these properties were deduced from the literature and resources that were available, and that the properties are general, since adaptations of the methods do exist and exhibit different properties. For example, an adapted version of the observer method, which is designed to be robust to modelling uncertainties, could detect faults slower than the original method.

Table 2-1 Table of Properties of Fault Detection M

METHOD	Detection of faults related to Actuators	Detection of faults related to Processes	Detection of faults related to Sensors	Detection of all classes of failures	Detection of Multiple Simultaneous faults	Detection of Abrupt faults	Detection of Incipient faults	Detection of Vibration faults	Promptness of detection	Noise Sensitivity	Robust
Observer	✓	✓	✓	✓	✗	✓	✓	✗	Fast	Low	✗
Parameter Estimation	✓	✓	✓	✓	✗	✓	✓	✗	Fast	Average	✗
Parity Space	✓	✓	✓	✓	✗	✓	✓	✗	Average	High	✓
Adaptive Observer	✓	✓	✓	✓	✗	✓	✓	✗	Fast	Average	✓
Adaptive Parity Space	✓	✓	✓	✓	✓	✓	✓	✗	Slow	Low	✓
Spectral Analysis	✗	✓	✗	✗	✗	✗	✓	✓	Fast	High	✓
Bandpass Filters	✗	✓	✗	✗	✗	✗	✓	✓	Fast	High	✓
Maximum Entropy Estimation	✗	✓	✗	✗	✗	✗	✓	✓	Fast	Low	✓
Direct Parallel Testing	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✗
Likelihood Ratio Tests	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✗
Bayes Decision	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✓
Multivariate Test	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✓
CUSUM Algorithm	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✗
Compound Scalar Test	✓	✓	✓	✓	✗	✓	✓	✗	*	Average	✗
Intelligent Sensors	✗	✗	✓	✓	✓	✓	✓	✗	Average	Average	✓
Neural Networks	✓	✓	✓	✓	✓	✓	✓	✗	Fast	Average	✓
Fuzzy Classification	✓	✓	✓	✓	✗	✓	✓	✗	Average	Low	✓
Neuro-Fuzzy Networks	✓	✓	✓	✓	✓	✓	✓	✓	Average	Low	✓

* Determined by the change time of the symptoms and the sample path of the past observations of the system

ties of Fault Detection Methods

s n	Noise Sensitivity	Robust	Reliability	Isolability	Requirement of *a priori* Information	Ease of Implementation	Computational Burden	Assumptions & Comments
	Low	✗	✓	Bad – Average	✗	Simple	Low	• Can be insensitive to modelling uncertainty. • Implementation is complex for detecting multiple faults. • Sensors are failure-free. • Requires process excitation.
	Average	✗	✓	Average	✗	Average	Low	
	High	✓	✓	Average	✗	Average	Low	Implementation is complex for detecting multiple faults.
	Average	✓	✓	Good	✗	Simple	Average	
	Low	✓	✓	Good	✗	Complex	Average	
	High	✓	✓	Bad	✗	Simple	Low	
	High	✓	✓	Bad	✗	Simple	Low	
	Low	✓	✓	Good	✗	Average	Heavy	
	Average	✗	✓	Average	✓	Complex	Average	Statistical properties have to be strictly adhered to for good fault detection.
	Average	✗	✓	Bad – Average	✓	Simple	Average	
	Average	✓	✓	Good	✓	Complex	Heavy	
	Average	✓	✗	Average	✓	Complex	Average	Does not facilitate fault isolation.
	Average	✗	✓	Bad – Average	✓	Simple	Average	
	Average	✗	✗	Average	✓	Simple	Average	
	Average	✓	✓	Good	✓	Average	Heavy	Detects one fault per sensor at any one time.
	Average	✓	✓	Average – Good	Little or none	Complex	Heavy	• Requires resetting after fault detection. • Isolability dependent on number of faults required for detection.
	Low	✓	✓	Good	✗	Average	Average	
	Low	✓	✓	Good	✓	Complex	Heavy	

Table 2-2 of Properties of Fault Diagnosis M

METHOD	Amount of faults that can be diagnosed	Ability to diagnose Multiple Simultaneous faults	Promptness of diagnosability	Reliability	Isolability	Training Required
Geometric Distance & Probablistic Methods	Large	✗	Average	✓	Bad – Good	✗
Neural Networks	Large	✓	Average	✓	Average – Good	✓
Fuzzy Classification	Small – Average	✓	Average	✓	Average	✓
Neuro-Fuzzy Networks	Large	✓	Fast	✓	Good	✓
Intelligent Sensors	As many as required	✓	Average	✓	Good	✗
Expert Systems	Large	✓	Fast	✓	Average	✓

ble 2-2 of Properties of Fault Diagnosis Methods

liability	Isolability	Training Required	Ease of Implementation	Computational Burden	Assumptions & Comments
✓	Bad – Good	✗	Average	Heavy	Strict statistical properties need to be adhered to.
✓	Average – Good	✓	Complex	Heavy	• Needs to be trained optimally for good diagnosis with as much information as possible. • An increase in the amount of faults to be diagnosed, decreases the methods accuracy.
✓	Average	✓	Simple	Average	• Test data must be close to steady state. • System models must be accurate. • The lower the amount of faults to be diagnosed, the better the diagnosis.
✓	Good	✓	Complex	Heavy	Fuzzification rules need to be tried and tested.
✓	Good	✗	Average	Heavy	Is expensive, but could be cheaper than using hardware redundancy.
✓	Average	✓	Complex	Heavy	Large amounts of human expertise and system information are required.

2.15 Conclusions

The following conclusions may be drawn on the methods of fault detection and diagnosis. These conclusions are largely based on the general properties of the fault detection and diagnosis methods. Firstly, conclusions based on process model-based methods of fault detection will be stated, followed by signal model-based methods, the fuzzy model-based method, methods of change detection, model-based fault diagnosis methods, non model-based methods of fault detection and diagnosis and finally general conclusions.

Process Model-Based Methods of Fault Detection

1. Only the adaptive parity space method of fault detection is able to detect multiple simultaneous faults, of the process model-based methods of fault detection. This can be a problem where there are many inputs/outputs in a plant, where it is likely that more than one fault can occur at the same time.
2. Of the process model-based methods of fault detection, the parity space method is most sensitive to noise which results in the method giving good results only in the presence of a small amount of noise. The adaptive parity space method and some observer methods have low sensitivity to noise, giving these methods an advantage over the others, by the fact that noisy signals should not affect their performance.
3. Observer methods cannot reliably detect sensor faults unlike other process model-based methods. Furthermore, the method has other disadvantages of not being robust and having bad isolability at times. It is however, the simplest process model-based fault detection method.
4. Parameter estimation methods do not provide a robust method of fault detection. Furthermore, it has more requirements of the plant model than the other methods, and isolability and noise sensitivity are only average, resulting in it being the least preferred of the process model-based methods of fault detection.
5. The adaptive observer method has many advantages and promises to be a good method of fault detection. However, it is fairly sensitive to noise and is not able to detect multiple simultaneous faults.
6. The adaptive parity space method has the most advantages as compared to all process model-based methods (possibly excluding the adaptive observer method) in terms of

detecting multiple simultaneous faults, low sensitivity to noise, robustness, and detection of all classes of fault. The method is structured in a manner so as to ease the diagnosability of the faults, which is not the case in some of the other methods.

7. The adaptive parity space method is complex to implement with an average computational burden. This is unlike other process model-based methods, which are easier to implement and have lower computational burdens (except for adaptive observer). However the advantages of the adaptive parity space method far outweigh the advantages of the other methods.

8. Process model-based methods of fault detection are able to detect all types of faults (except vibration faults), all classes of failures, incipient and abrupt faults and are reliable. It is because of the above properties, together with the fact that they are generally easy to implement with low computational burden, that makes these methods the most widely used methods of fault detection.

Signal Model-Based Methods of Fault Detection

9. Of the signal model methods, maximum entropy estimation is the best method of fault detection. It has low sensitivity to noise and good isolability, unlike the spectral analysis and bandpass filters methods. It is however the slowest of the methods because it has the highest computational burden. Maximum entropy estimation is also not as easy to implement as the other signal model-based methods. Its advantages outweigh the other methods advantages.

10. Although signal model-based methods are reliable and robust and have the ability to detect incipient vibration and process faults quickly, these methods are very application specific. This is due to the fact that the methods are unable to detect actuator and sensor faults, or all the classes of failures, or abrupt faults or multiple simultaneous faults. These methods are therefore not generally used as methods of fault detection.

Fuzzy Model-Based Fault Detection

11. This method is able to detect all types and classes of faults (except vibration faults). Fuzzy model-based fault detection is unable to detect multiple simultaneous faults, but has low sensitivity to noise, is robust and reliable, and has good isolability. This

method promises to be a good fault detection method, but due to limited resources, this method cannot be justified as either a good or bad method.

Methods of Change Detection

12. The Bayes decision method is the most reliable of the change detection methods because it is robust and reliable with the best isolability of the change detection methods. The only other robust method is the multivariate test. Other reliable methods include direct parallel testing, likelihood ratio tests and CUSUM algorithm. Although the Bayes decision is more difficult to implement (aside from the multivariate test) and has a heavier computational burden, it has the most advantages from all change detection methods.

13. However, the Likelihood Ratio tests are simple and needs less *a priori* information, which could be more useful to certain processes, than having a complex though more reliable and robust method of fault detection (like Bayes). The multivariate test is not useful because it does not facilitate fault isolation. Direct parallel testing is not used in practice because the statistical properties are very stringent and the loss of information in the process results in poor fault detection. The CUSUM algorithm is simple to implement and is reliable, however it needs a lot of *a priori* information for average isolability, indicating that it is an average method for fault detection. The compound scalar test though simple to implement, is unreliable, not robust and does not have good isolability, resulting in it being the worst of the change detection methods.

14. Change detection methods are able to detect all types and classes of failures except for vibration faults and multiple simultaneous faults. These methods are useful for detecting faults, when residuals show stochastic behaviour with mean values and variances, since other fault detection methods are incapable of detecting these faults. Unfortunately though, the amount of *a priori* information required is generally large.

Model-Based Methods of Fault Diagnosis

15. Neural networks are good fault diagnosis schemes because of their ability to diagnose a large number of multiple simultaneous faults. They do however require training and are complex to implement with a heavy computational burden. Furthermore, an increase in the number of faults to be diagnosed decreases the methods ability to

diagnose faults correctly. An advantage of neural networks over other model-based methods of fault diagnosis is their ability to learn, so with time their ability increases to diagnose faults correctly.

16. Fuzzy classification methods are able to diagnose multiple simultaneous faults, but cannot diagnose as many faults as neural network and geometric distance and probabilistic methods. Furthermore, fuzzy classification methods require training, but are easier to implement with less computational burden than other model-based methods of fault diagnosis. However, the isolability of the method is average. Furthermore, these methods are good to use for diagnosis because of their ease of implementation, average computational burden and the good isolability achieved when few faults need to be diagnosed.

17. Geometric distance and probabilistic methods are able to diagnose a large amount of faults, though not multiple simultaneous faults. Their isolability varies from bad to good, depending on the types of faults that need to be diagnosed and on how strictly the statistical properties are adhered to. Although these methods do not require training, their computational burden is heavy. These methods do not match the good properties of neural networks and fuzzy classification methods. Furthermore, the ability of these methods to diagnose some types of fault correctly and others incorrectly, make these methods inferior to other model-based methods of fault diagnosis. As a result, these methods are not as widely used as the other methods.

18. All model-based fault diagnosis methods have similar requirements to ensure their reliability. The use of an optimally trained neural network has an edge over the other methods, because of its capability to learn and adapt to changes within the process, thus giving it good isolability properties.

Non Model-Based Methods of Fault Detection and Diagnosis

19. Limit checking is the simplest method of fault detection, which detects spurious faults and at times does not detect faults at all. This is caused by the fact that the thresholds used to detect the faults are set in such a manner to have a bias on either detecting spurious faults, or not detecting small faults. An optimum threshold band is difficult to obtain. This method is the least effective means of fault detection.

20. Special sensors use limit checking, and therefore have the same advantages and disadvantages. A further advantage is that they are able to indicate faulty equipment, but at an extra expense.

21. The use of multiple sensors is costly since at least three sensors have to be used to measure any one variable. The use of more sensors implies a further cost on the software required to operate the fault detection system. Another disadvantage is that space has to be found to accommodate the sensors.

22. Intelligent sensors provide a potentially good method for sensor fault diagnosis. The sensors however have drawbacks in that they are costly and that a fault in only one sensor can be detected at any one time. Furthermore, implementation of these sensors could require a reconfiguration of the plant in order to accommodate them. The advantages of this method such as good isolability in detection and diagnosis, robustness, reliability and detection of all classes of failures, outweigh its disadvantages of being complex to implement and heavy computational burden.

23. Of the methods for detecting sensor faults, the intelligent sensor provides the best method of fault detection and diagnosis. The fact that the validity of the data is being sent with the data to the diagnostic system, provides an abundance of information that neither multiple nor special sensors can provide. This provides a reliable method of fault diagnosis, which was found to be lacking in the previous methods discussed. The disadvantages (cost in particular) associated with special sensors are negligible as compared to those of the previously discussed methods.

24. The expert system is heavily dependent on correct *a priori* information, and to a limited extent on the knowledge of the plant operator. If these requirements are met, then the expert system will correctly diagnose faults when on-line. The computational burden increases if many rules have to be applied for the fault diagnosis. Such a situation may arise when many faults have to be diagnosed. Although expert systems used to diagnose faults are able to diagnose multiple simultaneous faults quickly, they are complex to implement with heavy computational burden and require training. All the information and knowledge that the expert system requires, make it an undesirable scheme for fault diagnosis. As a result, this scheme is not widely used.

25. Neural networks provide a robust and reliable method of fault detection, if they can be optimally trained to learn the required *a priori* information. They are able to quickly detect all types and classes of failures (excluding vibration faults). Problems which have been noted are the decrease in isolability: a) when there is an increase in the

information that needs to be learnt and b) if the data is noisy. Furthermore, this method is complex to implement with heavy computational burden and requires resetting after fault detection (this can disturb the process). As a result, this method is not as good as other methods of fault detection, because of its requirements and properties as described above.

26. Neuro-fuzzy networks are able to detect and diagnose all types and classes of failures, multiple simultaneous faults, have low sensitivity to noise, good isolability and are robust and reliable. However, they require *a priori* information and training, and are complex to implement with heavy computational burden. However, these latter properties are negligible in comparison to all the good properties that these methods possess. These methods are generally used in complex processes, which require accurate fault detection and diagnosis because of their complexity and heavy computational burden.

General

27. In general, process model-based methods of fault detection are better than signal model-based methods. The reasons being that the former is able to: detect multiple faults, detect faults correctly, detect all types of fault, avoid detecting spurious faults and robust, all of which is not generally true for signal model-based methods. However, process model-based methods are not able to detect vibration faults as signal model-based methods do.

28. Neuro-fuzzy networks provide the most competent method of fault detection and diagnosis of all of the non-model based methods of fault diagnosis. Neuro-fuzzy networks overcome all of the shortfalls of the other methods discussed so far. Since neuro-fuzzy networks are basically a combination of the fuzzy classification and neural network approaches, the shortfalls of each approach are overcome by the forte of the other.

29. Generally, the methods of fault detection and diagnosis, which exhibit the best results are not easy to implement or do not have a low computational burden. This is to be expected, since the more complex the method, the more thorough it is in computing or diagnosing the faults. Furthermore, complex methods generally have a heavy computational burden, since the two properties are almost always associated with each other.

30. Each of the fault detection methods is suited to specific processes – a method, which works on one system, may be totally inappropriate for use on another. For example, a fault detection method that functions correctly when applied to a mechanical plant, may function disastrously on a chemical process.

31. It is not physically meaningful to identify all failures because the system model (in model-based fault detection) is used to identify the failure. Some information should be assumed to be reliable (e.g. some sensors are reliable, some types of failures can be validated by alternate procedures and hence can be ruled out of the present model, etc.) so that the other information can be better manipulated.

32. There are many adaptations of the methods that have been described in this literature review. As a result, the properties that have been stated, and the conclusions drawn thus far, may differ according to the adaptation.

2.16 Recommendations

1. For sensor fault detection only, intelligent sensors should be used.
2. The best non model-based method of fault detection and diagnosis is neuro-fuzzy networks, which should be implemented in complex processes.
3. Of the model-based methods of fault detection, the adaptive parity space method is the best.
4. The Bayes decision method should be used as a method of change detection.
5. Of the fault diagnosis methods, the neural network method should be used for complex processes.
6. Fuzzy classification should be used for fault diagnosis in simple processes.
7. For the purposes of this project, the detection and diagnosis of many faults will not be needed, as the process used for the investigation is simple. Therefore, neuro-fuzzy networks would be too complicated to implement and would have an unnecessary high computational burden.
8. The use of change detection methods is also unnecessary for use on a system, which does not have *a priori* information.
9. The adaptive parity space method of fault detection should be further investigated and compared with the two methods of fault detection that show the next best properties: Parity Space and Observer. This is because the adaptive parity space method has the

best properties for simple process fault detection. The adaptive observer method which although it shows good properties, should be further investigated to confirm these properties. The author did not investigate this method, since the method became a research option, as the author had already progressed with this project beyond the point where the considerable time which would have been necessary to do so was not possible.

10. Since parameter estimation is used mostly for process fault detection, it should not be further investigated, as the faults to be tested are actuator and sensor faults only.

11. Since process model-based methods of fault detection are the only fault detection methods that can be used for the purposes of this project, the adaptive parity space method of fault detection exhibits the best properties of the process model-based methods of fault detection. There are five process model-based methods of fault detection. The reason given for not investigating the adaptive observer method is described in Point 9. Parameter estimation, parity space and observer methods exhibit similar properties, however, due to the reason given in Point 10, it was decided to investigate the parity space and observer methods.

Chapter 3 PROCESS DESCRIPTION

This chapter starts with a description of the mathematical model of the process. This is followed by a description of the characteristics and transfer function of the process. The performance of the system in closed loop is also analysed.

3.1 Mathematical Model Of The Process

The experimental system used in this study is a two-tank system, which can be seen below. The specifications and detailed operation of the process can be found in Appendix B.

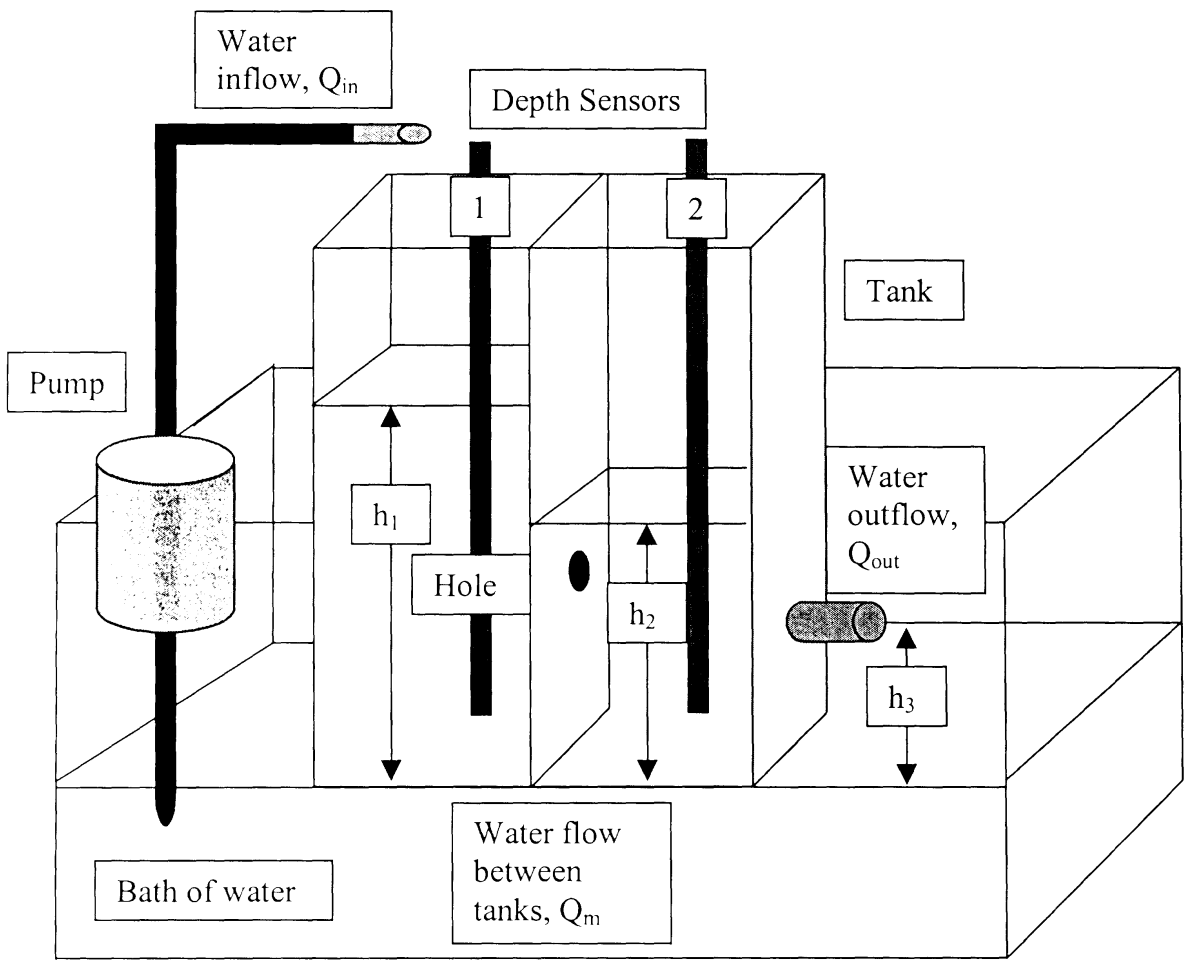


Figure 3-1 The Two-Tank Process

The mathematical model of the process is derived from a volume balance analysis as described in Appendix B, and can be expressed as the following state space model:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} \frac{-k_1}{A} & \frac{k_1}{A} \\ \frac{k_1}{A} & -\frac{k_1+k_2}{A} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A} \\ 0 \end{bmatrix} \cdot Q_{in}$$

Where

k_1, k_2 are the gain constants,

A is the cross-sectional area of each tank

Taking the Laplace transform gives:

$$\frac{h_2(s)}{Q_{in}(s)} = \frac{\frac{1}{k_2}}{(1+sT_1)\cdot(1+sT_2)} \quad \frac{[V]}{[cc]} \tag{1a}$$

With

$$\frac{h_1(s)}{Q_{in}(s)} = \frac{\frac{k_1+k_2}{k_1k_2} + \frac{A}{k_1k_2}s}{(1+sT_1)\cdot(1+sT_2)} \quad \frac{[V]}{[cc]} \tag{1b}$$

Where

$T_1T_2 = \frac{A^2}{k_1k_2}$ and $T_1 + T_2 = \frac{A(2k_1+k_2)}{k_1k_2}$ define the process time constants implicitly.

3.2 Determination of the Characteristics of the Process

In order to determine the system's parameters, the pump and the depth sensors need to be considered. This is because these devices are used to determine the process model, as described by the above equations. The determination of the pump characteristic is described followed by the determination of the sensor characteristics.

3.2.1 Pump characteristic

The pump needs to be calibrated in order to determine its characteristic. The calibration details and data used to obtain the characteristic are described in Appendix C. The relationship between the controlling signal and the pump flow is shown in Figure 3-2 on which a Least Squares (LS) fit is superimposed.

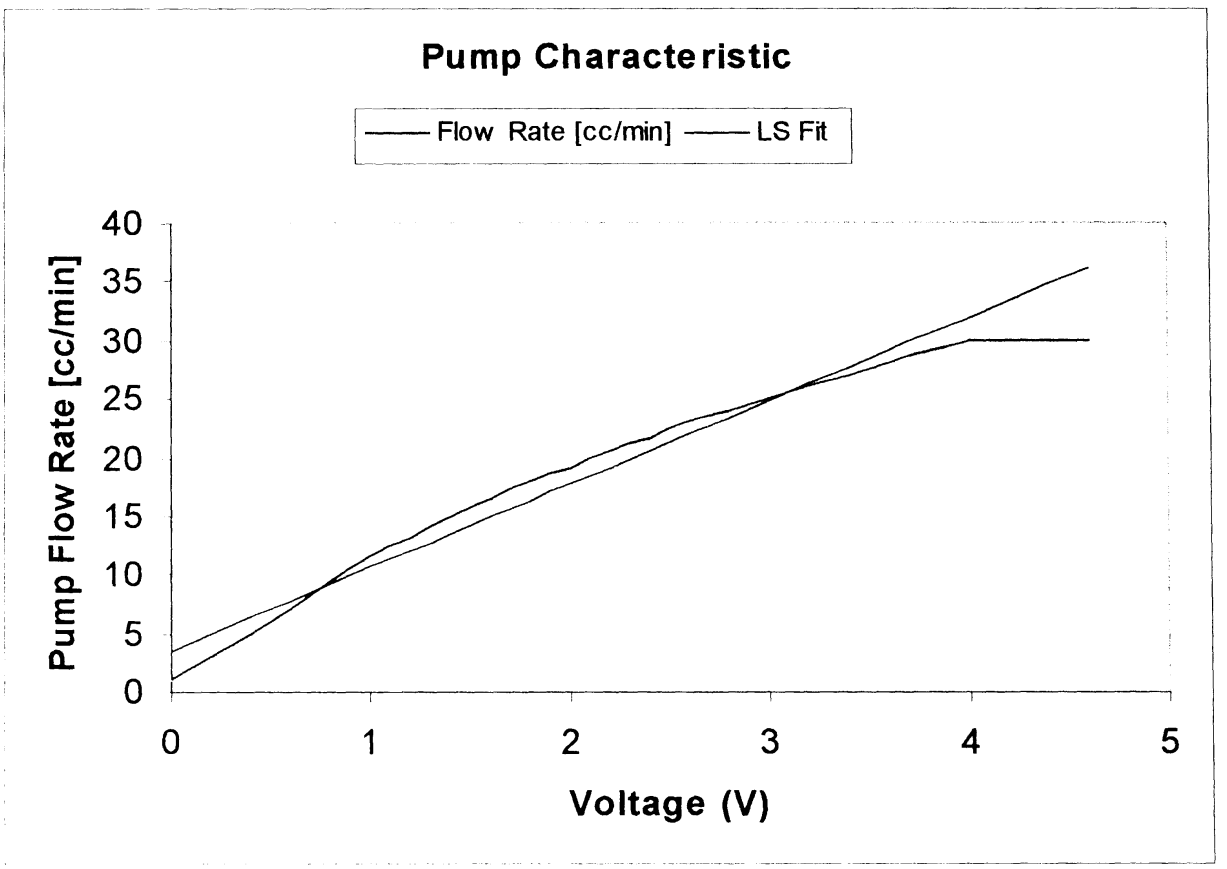


Figure 3-2 Pump Characteristic

From the graph, it can be seen that the LS approximation is a fairly accurate representation of the actual pump characteristic. Furthermore, the pump flow can be seen to reach its maximum flow rate when driven by an input voltage of 4V.

As a result, the following linear equation may be used to describe the pump characteristic (over the input voltage range of 0 – 4V).

$$y(cc\ \ min) = mx(V) + b \quad (\pm error)$$

The "LINEST" function in Microsoft EXCEL was used to obtain the gradient, m and the point of intersection, b of the LS linear equation.

$\therefore y = 7.11x + 3.50 \quad (\pm 2.50)$

Discussion:

The resolution of the flow meter is 0.5 [cc/min]. As a result, the observed flow rate values are not as accurate as they could possibly be. The flow rates were recorded when steady state had been reached. The pump is nonlinear, as can be seen from its characteristic.

The pump should not be operated without water as this would cause unnecessary wear and tear on the motor and its components. Therefore, there should always be enough water in the bath (at least 2/3 full) before operation, and the level switch should at all times be free to move vertically.

3.2.2 Depth sensor calibrations

The sensors were also calibrated (refer to Appendix C for details) in order to determine the sensor characteristics. The relationships between the water level heights and the sensor voltages are shown in the following two figures on which LS fits have been superimposed.

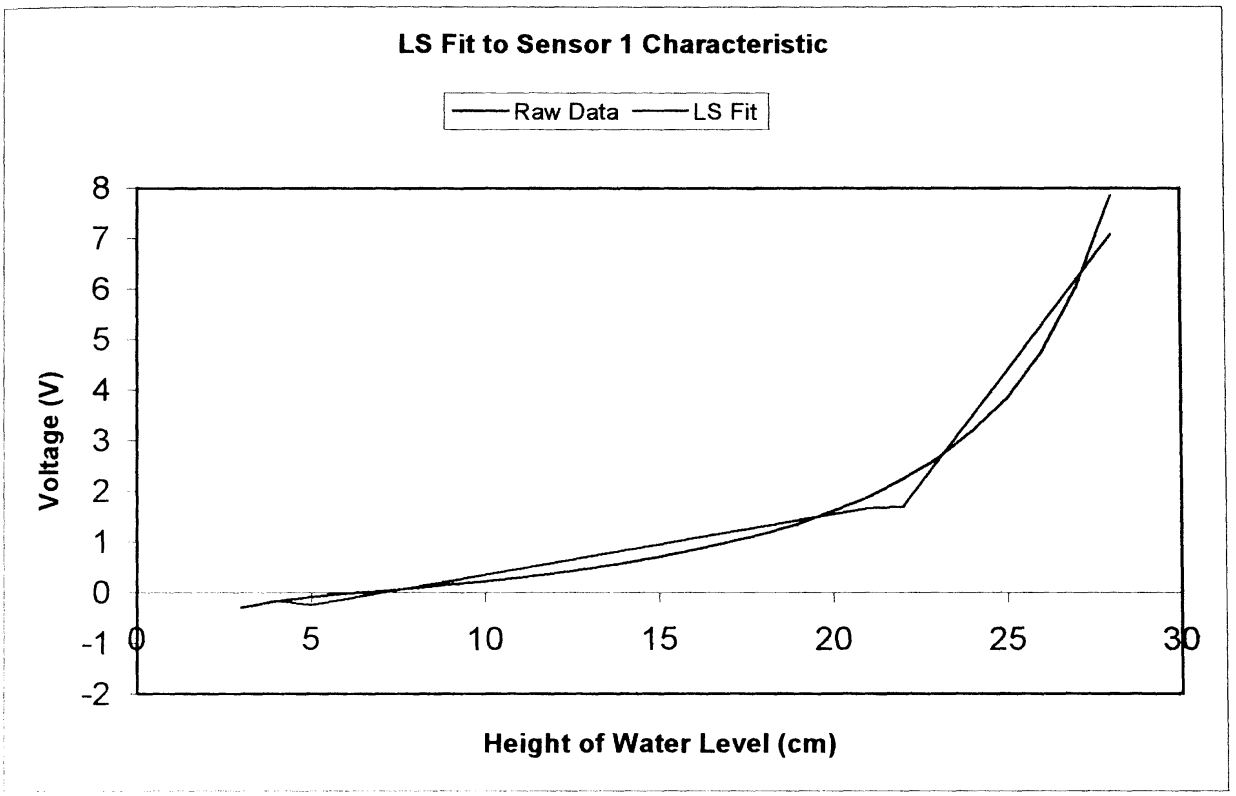


Figure 3-3 LS Fit to Sensor 1 Characteristic

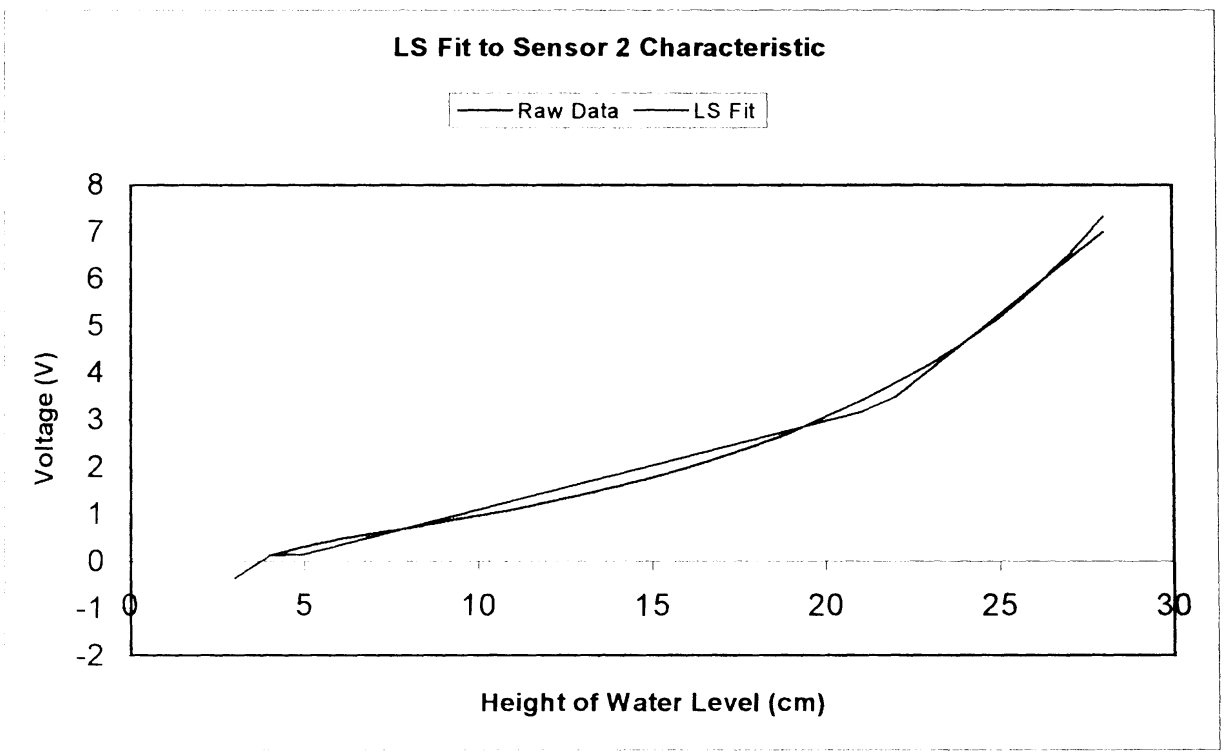


Figure 3-4 LS Fit to Sensor 2 Characteristic

From the previous two graphs, it can be seen that the LS fits are a fairly accurate representation of the sensor characteristics. In order to obtain these accurate fits, three LS equations were used to estimate each characteristic, split into the following regions (water level heights): 3 – 4 cm, 5 – 21 cm and 22 – 28 cm. There are regions between 4 – 5 cm and 21 – 22 cm where the LS fits are nonlinear. These regions represent the transition regions between the three LS fits.

Furthermore, the plots of the characteristics show that the sensors are highly nonlinear. Since the LS fits are fairly accurate, the following linear equations are used to describe the sensor characteristics. Given the general equation for the sensor characteristic:

$$y(V) = mx(cm) + b \quad (\pm error)$$

For Sensor 1:

$$y = \begin{cases} 0.13x - 0.70 \ (\pm 0.00) & for \ 3 \le x \le 4 \\ 0.12x - 0.85 \ (\pm 0.25) & for \ 5 \le x \le 21 \\ 0.90x - 18.10 \ (\pm 0.77) & for \ 22 \le x \le 28 \end{cases}$$

For Sensor 2:

$$y = \begin{cases} 0.98x - 0.00 \ (\pm 0.00) & for \ 3 \le x \le 4 \\ 0.19x - 0.80 \ (\pm 0.26) & for \ 5 \le x \le 21 \\ 0.56x - 9.40 \ (\pm 0.33) & for \ 22 \le x \le 28 \end{cases}$$

Discussion:

When the calibration was performed, there was noticeable drift in the output voltages of the sensors. The sensor output voltages were recorded once steady state had been reached. The accuracy of these values was to 0.005V.

The level sensors behave as an electrical resistance which vary with the water levels of the tanks. Wiping the sensors would cause a significant change in the output voltages, as the relationship between the heights of the water levels and the output voltages, would be affected (as the sensors are exteremely sensitive). Displacement of the sensors would also cause a

disparity in the output voltages. Distilled water should not be used in the system, as it does not contain the impurities that the sensors use to function.

3.3 Process Transfer Function

This section describes the method used to determine the linear operating region of the process. The method used to obtain the process transfer function follows.

3.3.1 Determination of linear operating region

From the previous section, the largest linear range of the sensor characteristics is between water level heights of 5 – 21cm. In order to find the linear operating region of the process, the corresponding input voltage range (fed into the pump) had to be determined. Therefore, voltages were applied to the pump drive input, and the corresponding sensor levels observed once steady state had been reached. The results are displayed in the following table. The linear operating region, which is a pump drive input voltage range of between 1 - 3V, has been shaded.

Table 3-1 Sensor Level Heights for Different Pump Drive Input Voltages

Input Voltage (V)	Sensor 1 Level Height (cm)	Sensor 2 Level Height (cm)
0	2.7	2.7
0.5	3.7	3.4
1.0	6.5	5.0
1.5	10.0	7.8
2.0	13.7	11.1
2.5	16.7	13.3
3.0	19.9	17.1
3.5	23.9	19.7
4.0	28.0	22.5
4.5	28.0	22.5

3.3.2 Methodology to determine the process model

The parameters (k_1,k_2,A) of the process model, defined in Equations 1 & 2, were determined from experimental data. The method is explained as follows.

The process's pump drive input was connected to a DAC and the depth sensors connected to an ADC. Two in-house programs, "Control Example" (which the author had adapted) and "Megalomania", were used to perform step tests on the process. The step test was performed by driving the pump input with an input voltage, (via the DAC), which would in turn determine the flow rate. The depth sensors would then output a voltage, which would be read as the process responses (via the ADC). The step tests were performed within the linear operating range of 1 to 3 Volts. The amplitude of the steps was 1V, while the voltage offsets were either 1V or 2V. Before each step was performed, the system was allowed to reach its steady state conditions. The flow rate was checked with the output voltages from the depth sensors and water level heights, to ensure consistency in the step tests. Furthermore, many step tests were performed to model the highly nonlinear process as accurately as possible. A graph showing a typical step test follows.

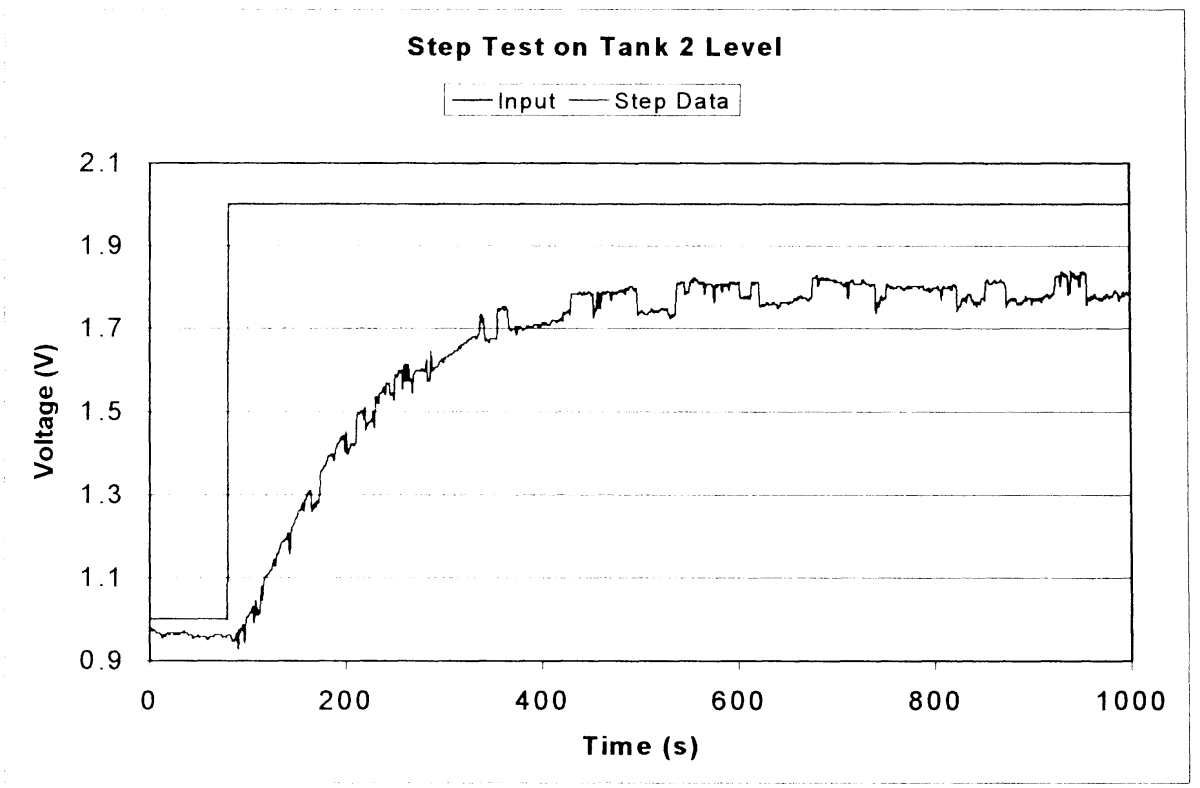


Figure 3-5 Typical Step Test

Discussion:

The accuracy of the program, as shown in the graph is ±0.05V on the y-axis and ±0.5[s] on the x-axis. The water level of the tank was allowed to reach steady state before the step was initiated. The response seems to be that of a first order system with dead time. The response shows that there are events represented by jumps which are approximately ± 0.12V. These events were caused by the internal electronics of the system. After many attempts, the author and the laboratory technician could not eliminate these events. As a result, the accuracy of the step responses was undermined. Since the calibration error of the operating region (±0.26V) was greater than the observed events, the author considered the observed step test results to be acceptable. Most of the noise in the system was filtered by the low-pass filters; with the resultant observed noise being approximately ±0.02V. Since the gain of this step test is 0.85±0.06V, the noise level is acceptable at 2.35±0.33%.

Although measures were taken to ensure that the process remained in the same state for all the step tests, it was unavoidable that discrepancies arose. These discrepancies arose from people interfering with the equipment and thus altering the settings such as the pump drive and depth sensor positions. The settings were put back to their original positions, but it is impossible that they would have been in the exact positions, before the interference occurred.

3.3.3 Process model

Where possible, a LS estimation to the step data was used to determine the process gain, A time constant, T and dead time, τ from the step tests (using an Excel equation solver). The model that was used for the estimations was that of a first order system with dead time:

$$G(s) = \frac{\mathrm{A}}{1 + sT} \cdot e^{-s\tau}$$

The flow rate between the two tanks gives rise to the dead time. Hence the diameter of the orifice between the tanks influences the effect that the dead time has on the system. A typical LS fit to step test data is shown in the graph that follows.

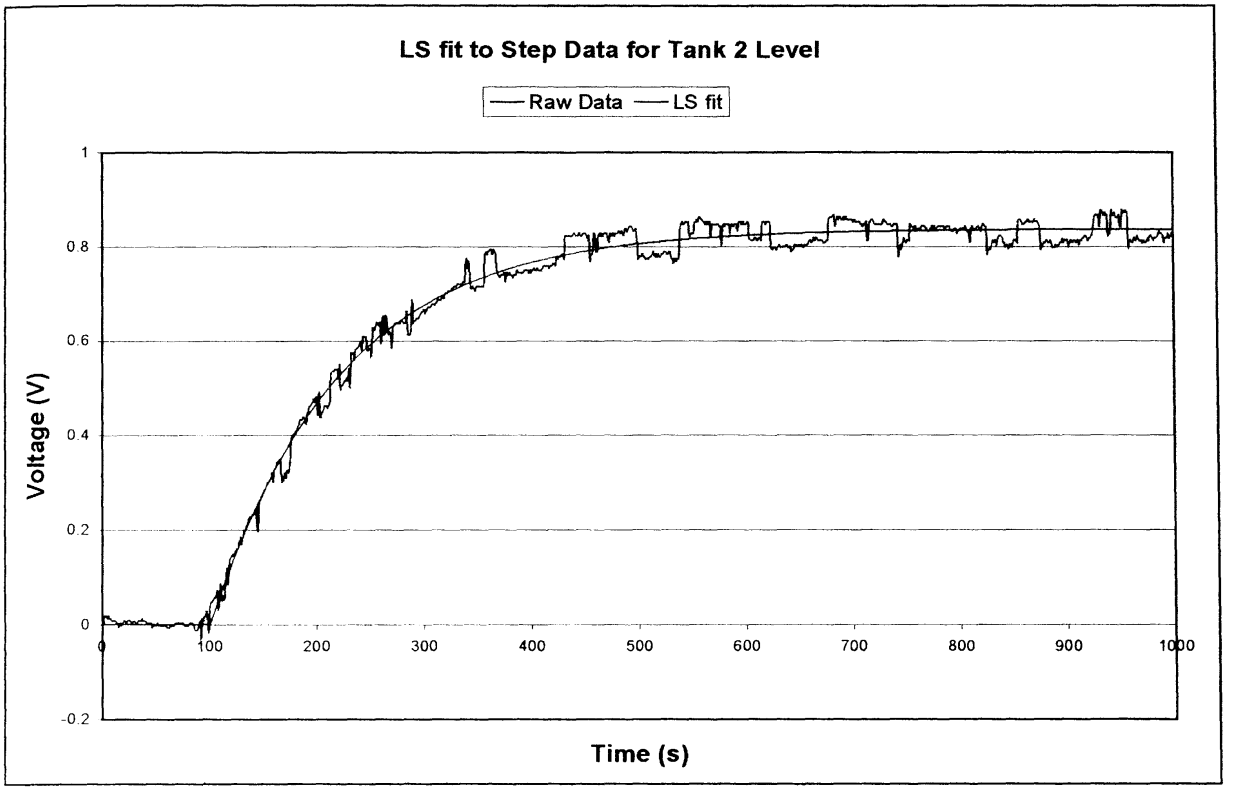


Figure 3-6 Typical LS fit to Step Data

From the above graph, it can be seen that LS provides a good estimation to the response. The LS model does not include a model of the noise or the drift.

This section is divided into the results obtained for Tank 2 and Tank 1. In the sections that follow, the tables display the data together with the Mean Squared Error (MSE) value (an index showing how accurate the LS model fits the raw step data).

3.3.3.1 Model for Tank 2

The following is a table showing the step test results.

Table 3-2 Step Test Results for Tank 2

	Offset Volts (V)	Gain, A Volts (V)	Time Constant, T Time (s)	Dead Time, τ Time (s)	Mean Squared Error Index
	1	0.759	105.0	2.2	0.073
	1	0.704	108.9	1.3	0.063
	1	0.724	107.1	0.8	0.144
	1	0.694	99.9	1.1	0.131
	1	0.745	100.6	0.1	0.315
	1	0.731	94.4	0.1	0.261
	1	0.762	107.7	0.2	0.091
	1	0.793	103.2	0.5	0.257
Average for 1V Offset	1	0.739±0.033	103.4±4.9	0.8±0.7	0.167
	2	0.879	158.4	13.9	0.238
	2	0.822	124.3	18.3	0.237
	2	0.888	146.8	16.6	0.262
	2	0.892	156.9	11.8	0.224
	2	0.839	136.9	18.1	0.207
	2	0.858	148.8	11.6	0.215
	2	0.867	136.4	12.6	0.188
	2	0.867	113.4	14.7	0.989
	2	0.793	118.1	14.6	0.651
	2	0.837	121.7	18.7	0.528
	2	0.893	148.4	14.1	0.371
	2	0.891	125.4	19.3	0.464
	2	0.814	133.7	21.4	0.289
Average for 2V Offset	2	0.857±0.033	136.1±14.9	15.8±3.1	0.374
Total Average	1.5	0.798±0.067	119.7±20.2	8.3±7.9	0.271

Discussion:

As can be seen from the above table, the variations in the data for the step tests for the same input voltage offset are quite large. Furthermore, there is a significant increase in the gain, time constant and dead time of the responses, as the offset is increased from 1V to 2V. Both these observations confirm the nonlinearity of the process. These effects can be explained by the following theory.

Considering Tank 2 only (at steady state), by the conservation of energy:

Loss in Potential Energy = Gain in Kinetic Energy

$$m \cdot g \cdot (h_2 - h_3) = \frac{1}{2} \cdot m \cdot v^2$$

$$\therefore v = \sqrt{2 \cdot g \cdot (h_2 - h_3)}$$

Where

m is the mass of water in the tank

g is the gravitational constant

h_2 is the height of water in the tank

h_3 is the height of the drain tap

v is the velocity of the water

The outflow, Q_{out} of Tank 2 is then given as:

$$Q_{out} = C_{d1} \cdot A_o \cdot v$$

$$\therefore Q_{out} = C_{d1} \cdot A_o \cdot \sqrt{2 \cdot g \cdot (h_2 - h_3)} \qquad (2a)$$

Where

A_o is the cross sectional area of the orifice of the drain tap

C_{d1} is the discharge coefficient of the orifice

Explanation of process parameters for Tank 2:

When the inflow increases (due to a step test with a larger voltage offset), the height of the water level in the tank has to increase, (by Equation 2a) governing the outflow. As a result, the gain of the process should increase with increasing voltage offset. Furthermore, the time constant of the model should also increase with an increased offset voltage, since the process should take longer to reach its increased steady state value.

The increase in the dead time can be explained by considering the pump flow rates (refer to Table C-1 in Appendix C). When a 1V step test with an offset of 1V is performed, the difference between the final and initial flow rates is 7.5[cc/min]. The difference between the final and initial flow rates for a 1V step test with 2V offset is 6[cc/min]. This implies an increase of 65% of the initial flow rate for the 1V offset step test, and an increase of 32% for

the 2V offset step test. As can be seen, the increase of the 1V offset step test is approximately twice that of the 2V offset step test. This implies that the rate of increase in the inflow for the 1V offset is twice that for the 2V offset step tests. If Tank 2 is now considered, the implication that arises from the above statements is that the rate of change of inflow is greater for a 1V offset step test than for a 2V offset step test. As a result, the response and therefore the dead time associated with 1V offset step tests should be quicker than that associated for step tests with an offset of 2V.

Model Determination:

In order to determine the transfer function of the tank, an average of the gain, time constant and dead time of the data calculated from within the linear region of the process was used. The values used were:

Gain, *A2* (V): 0.798 ± 0.067

Time Constant, *T2* (s): 119.7 ± 20.2

Dead Time, τ (s): 8.3 ± 7.9

The transfer function, *G2(s)* for the tank (in Laplacian form) is modelled as a first-order process with a dead time term as:

$$G2(s) = \frac{A2}{1 + sT2} \cdot e^{-s\tau}$$

The dead time term in the transfer function is approximated by the Maclaurin series approximation as:

$$e^{-s\tau} = \frac{1}{1 + \frac{s\tau}{1!} + \frac{(s\tau)^2}{2!} + \cdots}$$

In order to keep the model for Tank 2 simple, only the first two terms in the denominator of the Maclaurin expansion are used:

$$e^{-s\tau} = \frac{1}{1 + s\tau}$$

Furthermore, the resultant transfer function for Tank 2 satisfies the derived mathematical model as outlined in Equation 1a. The transfer function is now in the form:

$$G2(s) = \frac{A2}{(1+sT2)\cdot(1+s\tau)}$$

Substituting the values obtained for the gain, A2, time constant, T2 and dead time, τ gives:

$$G2(s) = \frac{0.798}{(1+s119.7)\cdot(1+s8.3)}$$

$$\therefore G2(s) = \frac{0.00080}{0.00101 + 0.12832\,s + s^2}$$

The transfer function has two poles, a non-dominant pole at s = -0.12048, and a dominant pole at s = -0.00835. The process is stable but slow since it has a dominant time constant of 119.7[s], and takes 478.8[s] to reach within ±2% of its final steady state value.

Figure 3.7 shows the simulated average step responses of Tank 2, with offsets of 1V and 2V and the total average (used to determine the model, *G2(s)*).

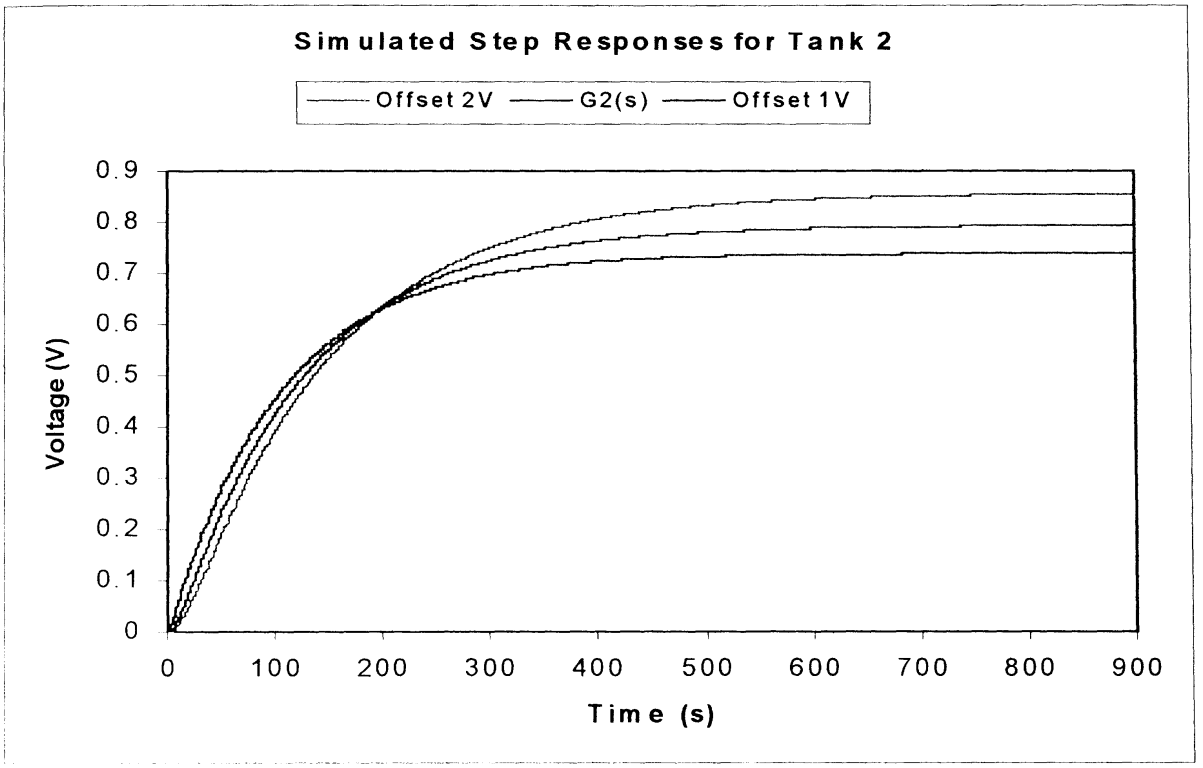
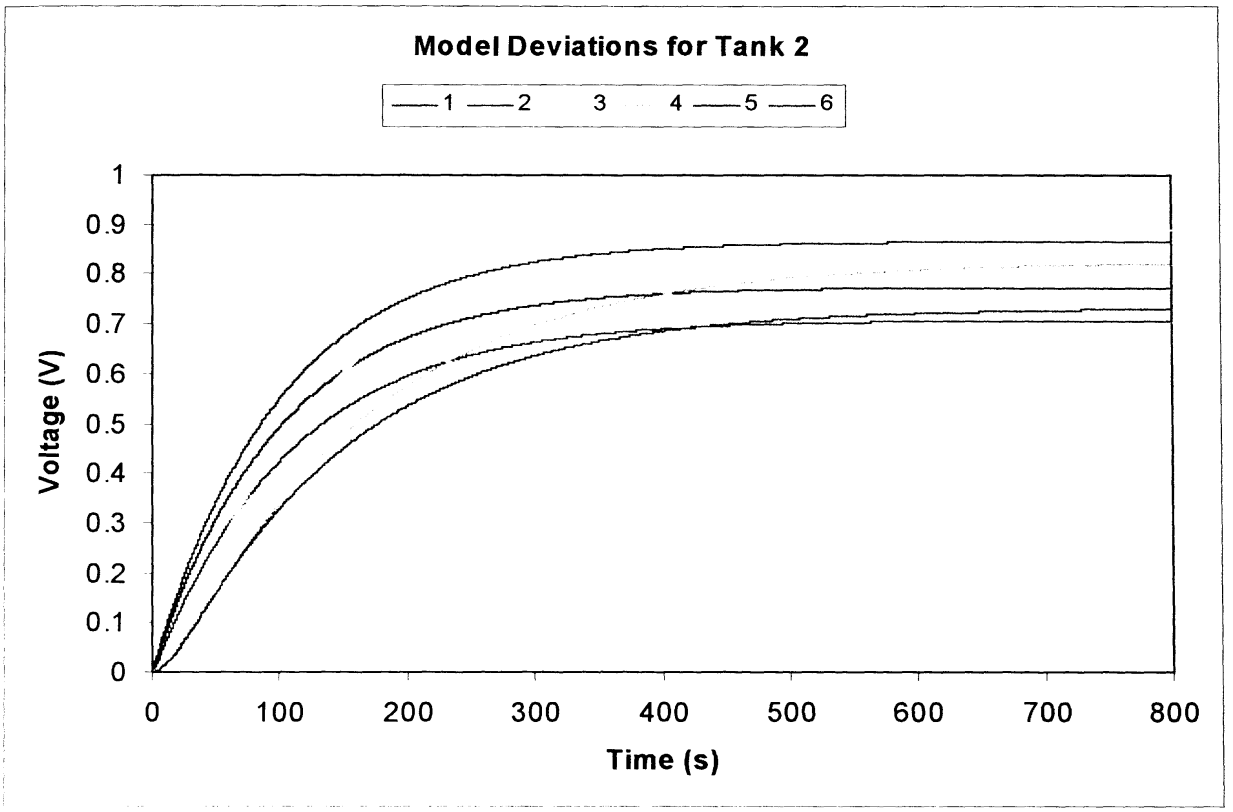


Figure 3-7 Simulated Step Responses for Tank 2

From the plots, it can be seen that the model obtained for *G2(s)* is a good average of the models predicted by the other responses. The fact that there is a difference in the step responses indicates that Tank 2 is nonlinear.

The following graph shows the simulated deviations of the models obtained from the step tests. The generated models were calculated from the standard deviations obtained for the gain, time constant and dead time, then fitted to the process model. The generated models were chosen from the two possible extremes that could be obtained for each voltage offset. The first type of model uses the maximum standard deviation for the gain and the minimum standard deviation for the time constant and dead time. The second type of model uses the maximum deviations for the time constant and dead time with the minimum deviation for the gain. Details can be seen in Figure 3-8.



Curve Number	Offset Volts (V)	Gain Volts (V)	Time Constant Time (s)	Dead Time Time (s)	Model
1	1	0.772	98.5	0.1	$\frac{0.772}{(1+98.5s)\cdot(1+0.1s)}$
2	1	0.706	108.3	1.5	$\frac{0.706}{(1+108.3s)\cdot(1+1.5s)}$
3	2	0.890	121.2	12.7	$\frac{0.890}{(1+121.2s)\cdot(1+12.7s)}$
4	2	0.824	151.0	18.9	$\frac{0.824}{(1+151.0s)\cdot(1+18.9s)}$
5	1.5	0.865	99.5	0.4	$\frac{0.865}{(1+99.5s)\cdot(1+0.4s)}$
6	1.5	0.731	139.9	16.2	$\frac{0.731}{(1+139.9s)\cdot(1+16.2s)}$

Figure 3-8 Model Deviations for Tank 2

The preceding graph shows the modelling uncertainty associated with the process model for Tank 2. The graph also justifies the nonlinearity of Tank 2.

3.3.3.2 Model for Tank 1

It is known that the model for Tank 1 should assume the form represented in Equation 1b (Section 3.1). However, using a LS estimation to the step test data obtained for 1V offsets was the simplest and most accurate method of obtaining a model to the responses. The LS estimation gives the following results as shown in Table 3-3.

Table 3-3 Step Test Results for Tank 1

	Offset Volts (V)	**Gain, A Volts (V)**	**Time Constant, T Time (s)**	**Dead Time, τ Time (s)**	**Mean Squared Error Index**
	1	0.329	88.8	0	0.198
	1	0.395	104.1	0	0.251
	1	0.295	107.7	0	0.261
	1	0.341	95.3	0	0.188
	1	0.327	88.6	0	0.191
	1	0.322	104	0	0.322
Average for 1V Offset	1	0.335±0.033	98.1±8.3	0	0.235

The transfer function for an offset of 1V, $G1(s)_{1VOffset}$ for the tank (in Laplacian form) is modelled as a first-order system:

$$G1(s)_{1VOffset} = \frac{A1}{1+sT1}$$

Where

Gain, $A1$ (V) = 0.335 ± 0.033

Time Constant, $T1$ (s) = 98.1 ± 8.3

From the values obtained in Table 3-3. The resultant model is:

$$G1(s)_{1VOffset} = \frac{0.335}{1+s98.1}$$

A LS estimation for a first or second order system could not be used on the step data obtained for step tests with an offset of 2V. This is because the response has the characteristic as shown in Figure 3-8 and the estimation could not accurately be used to determine the model.

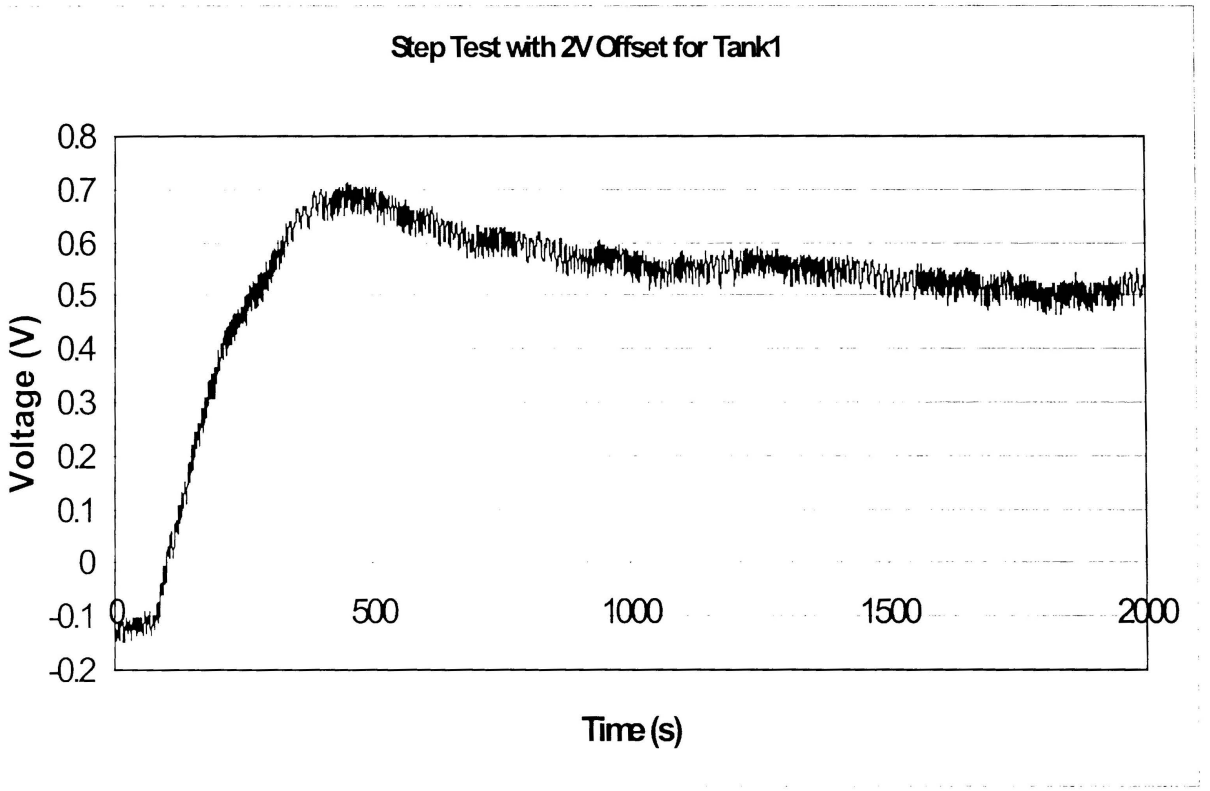


Figure 3-9 Typical Step Test for Tank 1 with an Offset of 2V

Discussion:

The response most closely resembles that of a second-order system with a zero in the left half of the *s*-plane with drift and noise. It can be seen from the above graph that the response is stable but slow, settling to steady state at approximately 1800[s]. There is also overshoot in the response of approximately 25% of the steady state value. The drift is in the form of a random low-frequency (between 0.004and 0.010Hz) sinusoidal wave. Noise in the response can be seen to be ±0.03V, which is 5% of the gain (0.6V).

The following graph shows the deviations in the step test results obtained. Actual step test results are used to show the deviations associated with an offset of 2V. The model deviations for an offset of 1V were calculated from the standard deviations obtained for the gain, time constant and dead time then fitted to the process model (as in the preceding section).

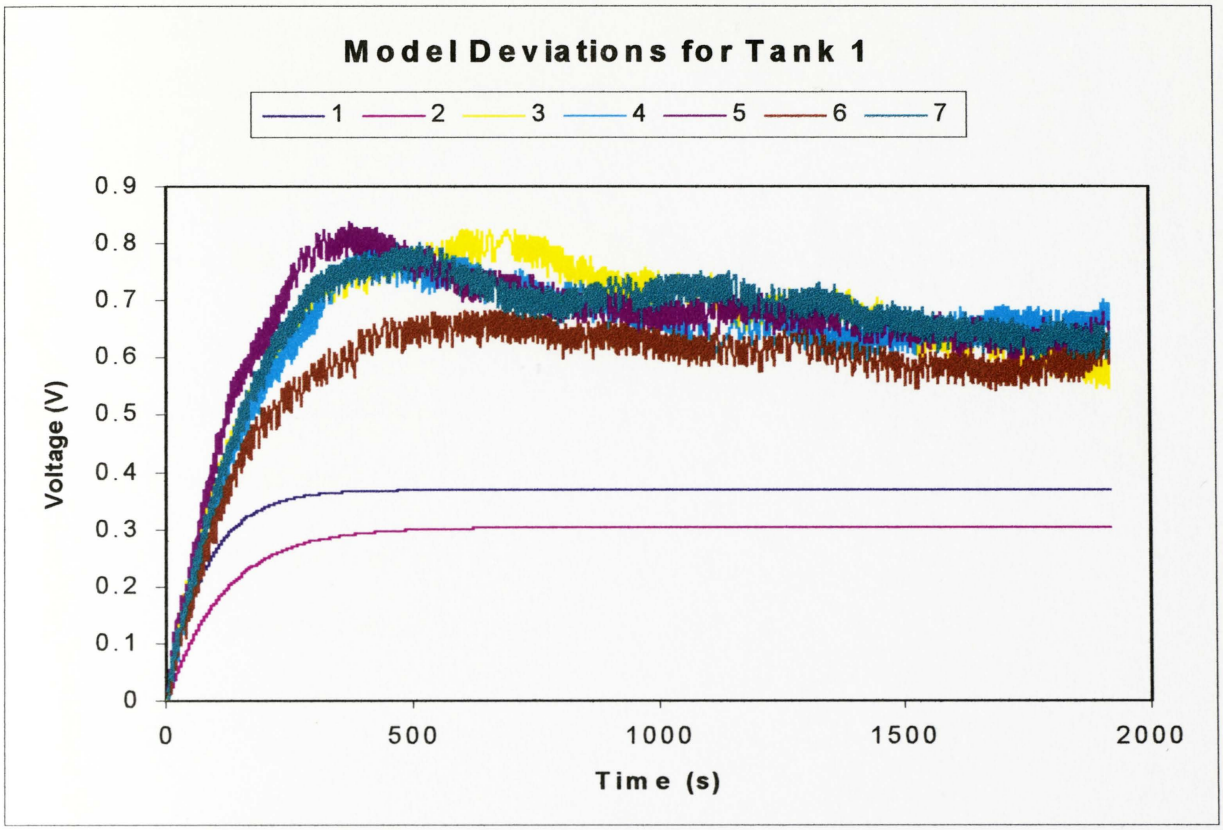


Figure 3-10 Model Deviations for Tank 1

Curves 1 & 2 in the graph represent the deviations in the model associated with an offset of 1V. Curves 3 to 7 show the actual step test results for an offset of 2V. The models that were used to obtain Curves 1 & 2 are:

Curve 1: $\frac{0.368}{1+79.8s}$ and Curve 2: $\frac{0.302}{1+116.4s}$

Discussion:

From the graph, it can be seen that the gains and time constants of the step responses vary significantly, indicating that Tank 1 is highly nonlinear. There is an increase in the gain and time constant of the responses, as the offset is increased from 1V to 2V. These effects can be explained by the following theory.

Considering Tank 1 only (at steady state), by the conservation of energy:

Loss in Potential Energy = Gain in Kinetic Energy

$$m \cdot g \cdot (h_1 - h_2) = \frac{1}{2} \cdot m \cdot v^2$$

$$\therefore v = \sqrt{2 \cdot g \cdot (h_1 - h_2)}$$

Where

m is the mass of water in the tank

g is the gravitational constant

h_1 is the height of water in Tank 1

h_2 is the height of water in Tank 2

v is the velocity of the water

The outflow, Q_m of Tank 1 is then given as:

$$Q_m = C_{d2} \cdot A_m \cdot v$$

$$\therefore Q_m = C_{d2} \cdot A_m \cdot \sqrt{2 \cdot g \cdot (h_1 - h_2)} \qquad (2b)$$

Where

A_m is the cross sectional area of the orifice between the tanks

C_{d2} is the discharge coefficient of the orifice

Explanation for process parameters of Tank 1:

When the inflow increases (due to a step test with a larger voltage offset), the height of the water level in the tank has to increase (by Equation 2b). As a result, the gain of the process should increase with increasing voltage offsets. Furthermore, the time constant of the model should also increase with an increased offset voltage, since the process should take longer to reach its increased steady state value.

If the step responses of Tank 2 (Figure 3-7) are compared to those shown in Figure 3-10 for Tank 1, it can be seen that step responses for Tank 1 are more varied than those obtained for Tank 2. As a result, Tank 1 can be said to be more nonlinear than Tank 2.

Model Determination:

An average of the step responses with offsets of 2V was then obtained. The model for Tank 1 was determined by averaging the responses obtained from both offsets. The responses can be seen in the following graph.

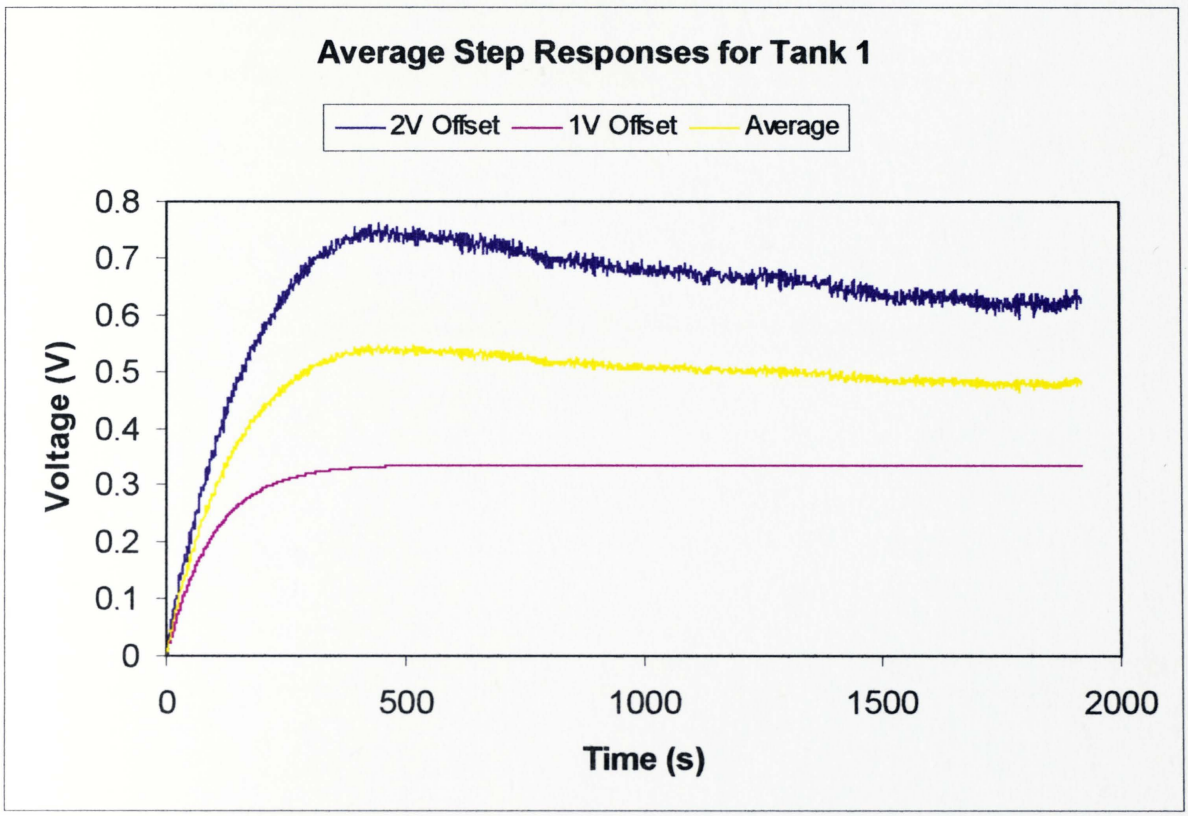


Figure 3-11 Average Step Responses for Tank 1

From Equations 1a & 1b in section 3.1, it is known that models for both Tank 1 and Tank 2 have the same characteristic function, ϑ:

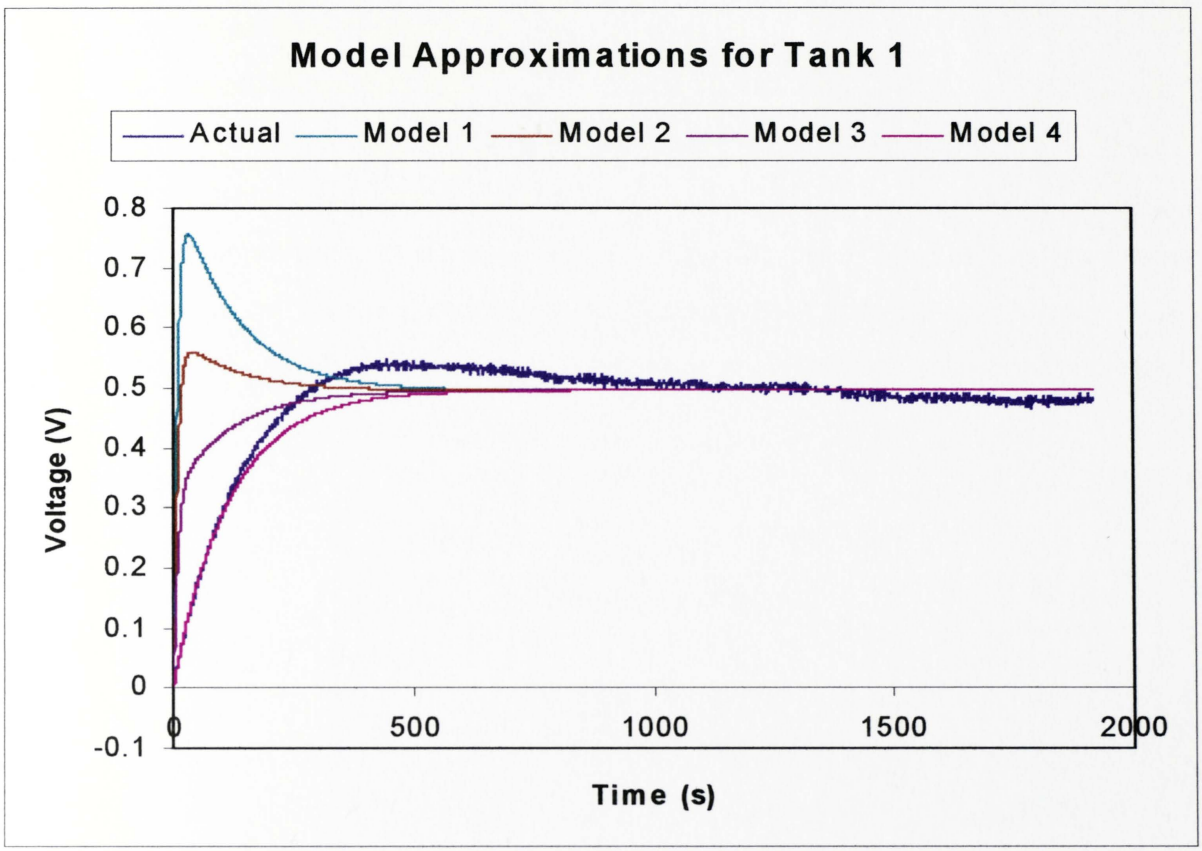
$$\vartheta = (1+sT_1)\cdot(1+sT_2)$$

From the model obtained for Tank 2, it is known that the characteristic function is:

$$\vartheta = (1+s119.7)\cdot(1+s8.3)$$

$$\therefore \vartheta = s^2 + 0.12832s + 0.00101$$

Since the characteristic function for the model of Tank 2 was now known, only the left half plane zero in the *s*-plane needed to be determined. The model for Tank 1 was then obtained by determining the zero through trial and error (using "Megalomania"). The following graph shows the approximated models for Tank 1 and the actual model that has been evaluated.



Model	Zero
1	0.100s+0.0005
2	0.070s+0.0005
3	0.010s+0.0005
4	0.005s+0.0005

Figure 3-12 Approximated G1(s) Models with Average Step Response for Tank 1

As can be seen the approximation closest to the actual step response is Model 4. Although this approximation is not as accurate as it could be, it is the best available option, given the constraint on the denominator of the transfer function for Tank 1. The model for *G1(s)* is then: $G1(s) = \dfrac{0.005s + 0.0005}{s^2 + 0.12832s + 0.00101}$

3.4 Closed Loop Response

The structure of the process model, *G(s)* is given in MIMO notation, because although only Tank 2 is controlled under negative feedback, the model for Tank 1 is necessary for fault detection implementation (so as to be able to use model-based fault detection methods to detect faults that could occur on Sensor 1). A controller is used to improve the dynamics of the process, by making it faster or ensuring that it correctly tracks its setpoint. Since it is unnecessary to improve the dynamics of the process for fault detection, a Type 1, Type 2 or non-unity Type 0 controller, which would traditionally have been designed for such a process, is not required. A unity Type 0 controller has therefore been implemented because it is the simplest controller available. Given that the system is in closed loop with negative feedback, the models are:

Controller, $K(s) = 1$

Process, $G(s) = \begin{bmatrix} G2(s) \\ G1(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.00080}{s^2 + 0.12832s + 0.00101} \\ \dfrac{0.00500s + 0.00050}{s^2 + 0.12832s + 0.00101} \end{bmatrix}$

Post-Compensator, $L(s) = [1\ 0]$

Feedback, $F(s) = 1$

The Open Loop Response, $Q(s)$ is given by:

$$Q(s) = L(S) \cdot G(s) \cdot K(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{0.00080}{s^2 + 0.12832s + 0.00101} \\ \dfrac{0.00500s + 0.00050}{s^2 + 0.12832s + 0.00101} \end{bmatrix} \cdot 1 = \begin{bmatrix} \dfrac{0.00080}{s^2 + 0.12832s + 0.00101} \end{bmatrix}$$

The Closed Loop Response, *H(s)* is given by:

$$H(s) = \frac{Q(s)}{1 + F(s) \cdot Q(s)} = \frac{\left[\frac{0.00080}{s^2 + 0.12832s + 0.00101}\right]}{1 + 1 \cdot \left[\frac{0.00080}{s^2 + 0.12832s + 0.00101}\right]} = \frac{\left[\frac{0.00080}{s^2 + 0.12832s + 0.00101}\right]}{\left[\frac{s^2 + 0.12832s + 0.00181}{s^2 + 0.12832s + 0.00101}\right]}$$

$$= \left[\frac{0.00080}{s^2 + 0.12832s + 0.00181}\right]$$

The following graph shows the closed loop step responses (simulated and real time) of the system. The size of the step was 1V, and the voltage offset was 1.5V.

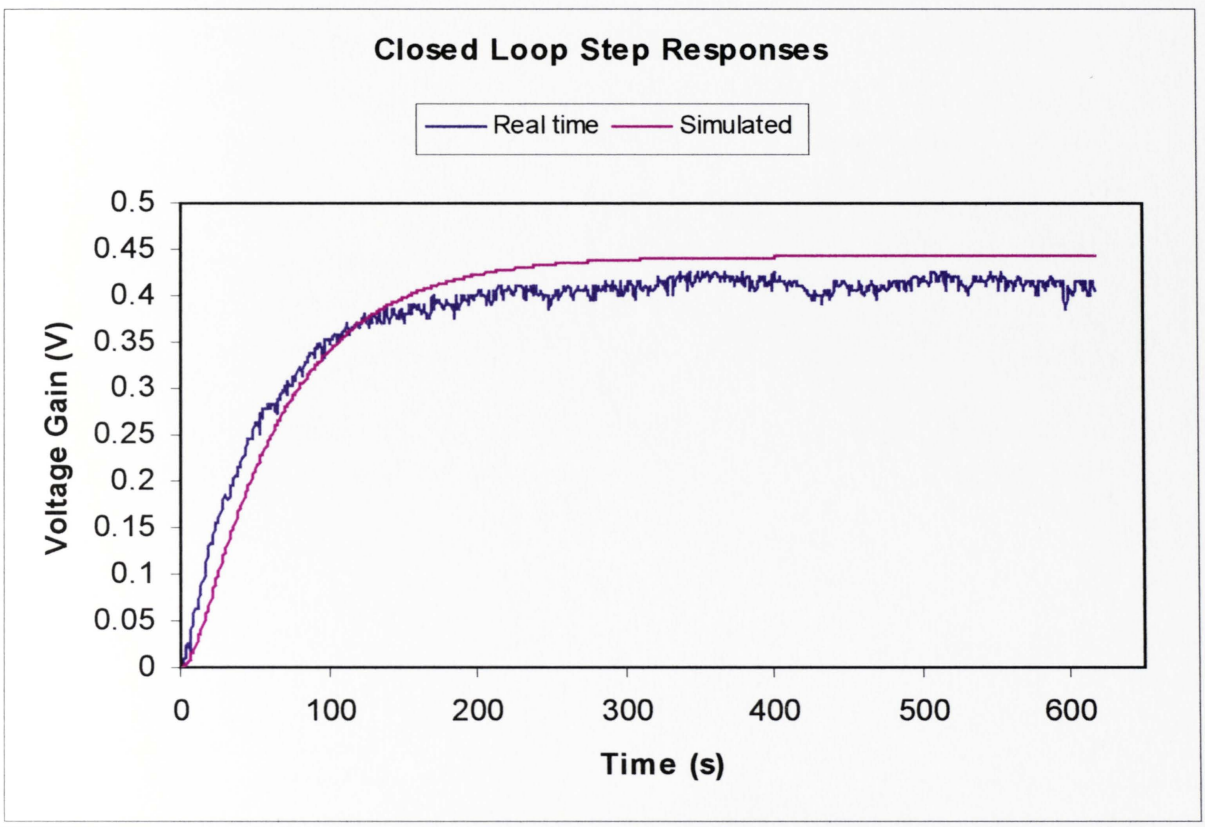


Figure 3-13 Closed Loop Step Responses: Simulated and Real time of the System

As can be seen from the two plots, the real time response is faster than that predicted by the simulation. The real time response has a time constant of 50[s] and takes 200[s] to reach within ±2% of its final steady state value, whereas the simulated response has a time constant of 76[s] and takes 304[s] to reach within ±2% of its final steady state value. The real time

response does however have a smaller gain of 0.405, while the simulated response has a larger gain of 0.442. These two phenomena can be explained by the fact that there was modelling uncertainty involved when deriving the model, and that the process is highly nonlinear. The responses are stable and there is no oscillation or overshoot.

The closed loop response is much faster than the open loop, which has a dominant time constant of 119.7[s], and takes 478.8[s] to reach within ±2% of its final steady state value. The gain of the closed loop response is however smaller than 0.798, which is the gain of the open loop response. Both these observations are in accordance with theory.

3.5 Nyquist Stability

The following is a Nyquist plot of the system for *G2(s)*.

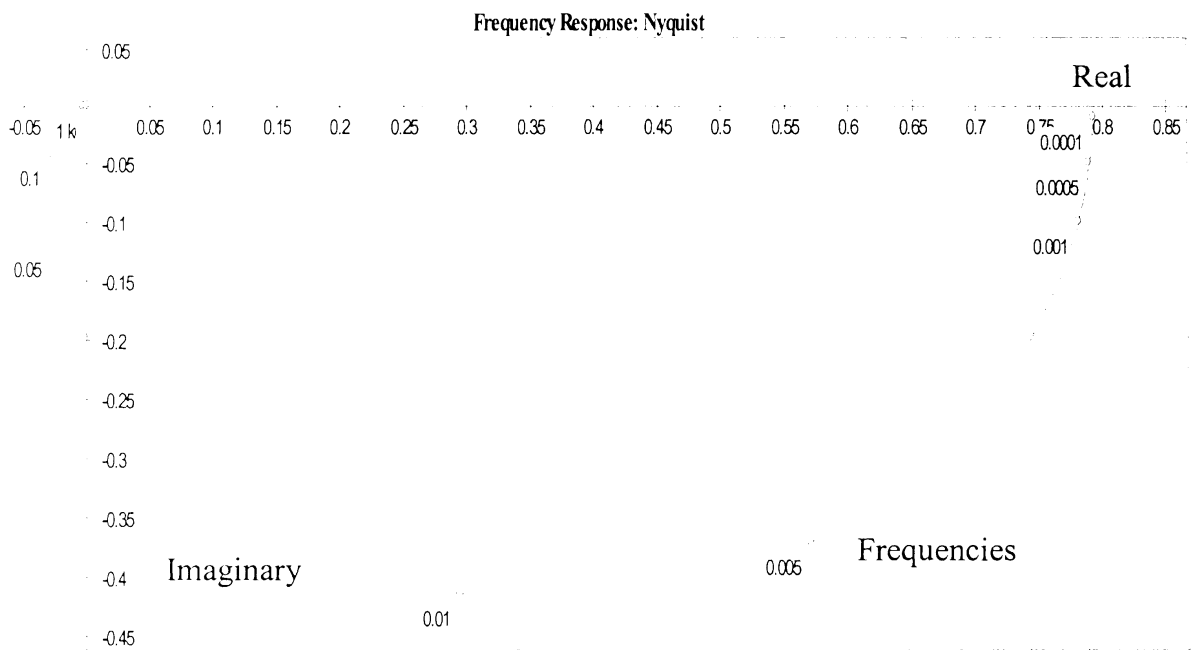


Figure 3-14 Nyquist Plot of the System

From *Q(s)* it is clear that there are no unstable poles in the open loop. Therefore, for closed loop stability, $<N>$ must be zero. By inspection of the Nyquist plot, this is true for $F(s) > 0$. This means that the closed loop system is stable for all gains in negative feedback configuration. Since $F(s) = 1$, the phase margin is infinity, which means that an infinite

amount of phase lead may be added to the system without making it unstable, and that the closed loop poles of the system are not damped at all.

The Gain Margin, *GM* of the Nyquist plot is defined by:

$$GM = -\frac{1}{aF(s)} = 20Log_{10}\left[-\frac{1}{aF(s)}\right] \qquad in \ \ [dB]$$

Where, a is the point at which the Nyquist plot intersects the negative real axis. In the case of the process:

$$GM = 20Log_{10}\left[-\frac{1}{-0.00002 \cdot 1}\right] = 94dB$$

This means that if the gain margin were to be multiplied by the open loop transfer function, the Nyquist plot would pass through the critical point and result in a closed loop system that would oscillate forever. Increasing the gain further would make the closed loop system unstable. Since the gain margin is large for the system, it can be said to be a very stable system.

3.6 Concluding Remarks

This chapter has presented a detailed description of the process on which the chosen methods of fault detection and diagnosis are to be applied. The process has been seen to be highly nonlinear, but the model obtained is the best possible under the constraints imposed by the mathematical formulations. It was found that there was modelling uncertainty, the degree of the effect it will have on the fault detection system, can only be determined by online implementation of the chosen fault detection methods. Through analysis of the two-pump system, it would seem that normal operation is almost impossible since the nonlinearity of the process is so great (especially noticed in Tank 1). So, practically the detection of small (up to 0.25V for the Sensor 1 faults, up to 0.26V for Sensor 2 faults, and up to 2.5 [cc/min] for the actuator faults) process and additive faults (Scale and Bias faults, respectively) would be very difficult, if not impossible. The reasons for these selections is because the associated error of the linearised Least Squares fit to the pump characteristic (Section 3.2.1) and the sensor characteristics (Section 3.2.2). The following chapter presents detailed descriptions of the methods that are to be used for fault detection.

Chapter 4 THEORETICAL DESCRIPTION OF PARITY SPACE, ADAPTIVE PARITY SPACE AND OBSERVER METHODS OF FAULT DETECTION

This chapter describes the methods of parity space, adaptive parity space and observer methods of fault detection that were simulated. A theoretical description of each method with application data is included.

4.1 Description of Process and Failure Models

The system that was used for the fault detection implementation was the two-tank system, which has one actuator and two level sensors. The types of faults that can be tested on the system are:

- Sensor 1 & 2 bias and scale (or change in scale) faults
- actuator bias and scale faults
- parametric (change in model) fault

A bias fault is an additive fault, whereas a scale fault is multiplicative. For the purposes of this project, only the sensor and actuator faults will be considered.

The transfer function of the system is:

$$G(s) = \begin{bmatrix} \dfrac{0.00080}{s^2 + 0.12832s + 0.00101} \\ \\ \dfrac{0.00500s + 0.00050}{s^2 + 0.12832s + 0.00101} \end{bmatrix} \quad (3)$$

The state space model of a linearised dynamic system can be represented as:

$$\dot{x}(t) = \Lambda x(t) + \Pi u(t) \quad (4)$$

$$y(t) = H_x x(t) + H_u u(t) \quad (5)$$

where $x \in R^i$, $u \in R^m$ *and* $y \in R^q$. For the process, $i = 2$, $m = 1$ and $q = 2$. The vector $u(t)$ represents the control input to the process. The disturbance inputs were not modeled for the process. The following are the actual matrices used for the system, in its canonical form:

$$\Lambda = \begin{bmatrix} 0 & 1.00000 \\ -0.00101 & -0.12832 \end{bmatrix} \tag{6}$$

$$\Pi = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{7}$$

$$H_x = \begin{bmatrix} 0.00050 & 0.00500 \\ 0.00080 & 0 \end{bmatrix} \tag{8}$$

$$H_u = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9}$$

A linear-time-invariant, discrete time representation of the system is:

$$\Delta x(k+1) = \Phi \Delta x(k) + \Gamma \Delta u(k) \tag{10}$$

$$\Delta y(k) = H_x \Delta x(k) + H_u \Delta u(k) \tag{11}$$

with

$$\Delta x(k) = x(k) - x_0 \tag{12}$$

$$\Delta u(k) = u(k) - u_0 \tag{13}$$

$$\Delta y(k) = y(k) - y_0 \tag{14}$$

and

$$\Phi = \begin{bmatrix} 0.98724 & -0.00010 \\ -0.09936 & 0.99999 \end{bmatrix} \tag{15}$$

$$\Gamma = \begin{bmatrix} 0.09936 \\ 0.00497 \end{bmatrix} \tag{16}$$

$$H_x = \begin{bmatrix} 0.00050 & 0.00500 \\ 0.00080 & 0 \end{bmatrix} \tag{17}$$

$$H_u = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{18}$$

where, k is the sampling instant, and the sampling interval is set at $\Delta t = 0.1$[s] for the system. Furthermore, x_0, u_0, and y_0 are the operating values. Control is assumed to be constant

between any two samples. The outputs of the system are obtained through the two imperfect sensors. A vector of measurements is defined as:

$$\Delta z(k) = M(k)\Delta y(k) + \Delta v(k) + \Delta n(k) \tag{19}$$

where, Δz, Δv, $\Delta n \in R^q$, and $M \in R^{q\times q}$ (given $q = 2$ for the process). The measurement-error noise sequence, $\{\Delta n(k)\}$, has zero mean and covariance, R The vector $\Delta v(k)$ shows the bias of the sensors, and

$$M(k) = diag[m_1(k), m_2(k), \cdots, m_q(k)] = diag[m(k)] \tag{20}$$

gives the scaling of the sensors with respect to the output. For example, if Sensor 1 is stuck m_1 would equal zero, and Δv_1 would be the value at which it is stuck. Also, if the instrument cable is poorly installed and subjected to cross coupling from its surrounding cables, this would result in Δn being non-zero. The effect of waves in the tanks is attributed as noise in the process. The displacement of Sensor 2, would cause the component m_2 to be nonzero and equal to the corresponding sensor voltage displacement.

For actuator failures, the actual input $\Delta u(k)$, can differ from the commanded input $\Delta u^c(k)$. The difference can be described by:

$$\Delta u(k) = S(k)\Delta u^c(k) + \Delta d(k) \tag{21}$$

where, Δu^c, $\Delta d \in R^m$, $S(k) \in R^{m\times m}$ ($m = 1$ for the process), and

$$S(k) = diag[s_1(k), s_2(k), \cdots, s_m(k)] = diag[s(k)] \tag{22}$$

For example, if the actuator blows up, s is zero, and Δd takes its extreme value. The component s would become non-unity if the actuator were to have an object partially blocking its outflow orifice. For the process,

$$M(k) = \begin{bmatrix} m_1(k) & 0 \\ 0 & m_2(k) \end{bmatrix} \quad \text{and} \quad S(k) = [s(k)]$$

(23), (24)

The complete fault detection system is shown Figure 4-1. The block F is the feedback function (which is unity), and the fault detection scheme is seen to have u^c and z as inputs. Note R is the setpoint of the system.

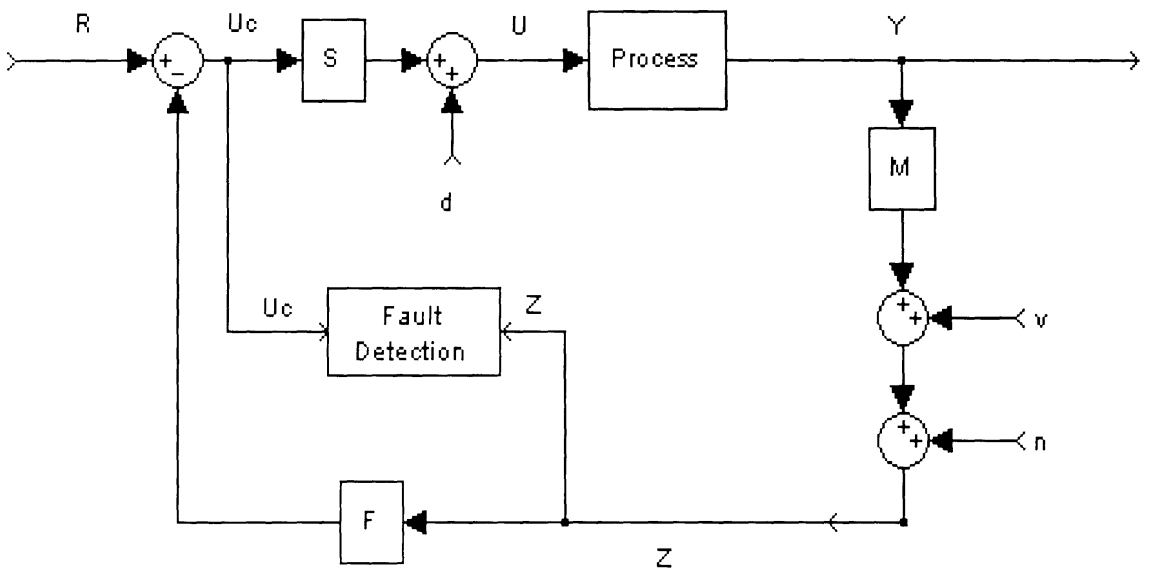


Figure 4-1 Fault Detection System

4.2 Description Of Methods

This section describes both parity space and adaptive parity space methods of fault detection. The theory of the aforementioned methods was given by Gopisetty and Stengel, 1998. Also included is the observer method of fault detection. All matrices and equations that are displayed have been calculated using Matlab.

The residuals are formed from a time-window of information consisting of measured outputs and commanded inputs to the process. The residual is not structured in terms of inputs and outputs, so as to allow maximum design freedom to reject disturbances and parametric uncertainty. The sensor and actuator faults are parameterised by biases and changes in scale, so as to obtain an explicit dependence of these parameters on the residuals. A fault is then

indicated when a residual exceeds a certain threshold. The fault-free case is described by the residual approximately equaling zero.

4.2.1 Parity space

The observed outputs, Δz, and the commanded inputs, Δu^c constitute the available information for a fault detection scheme. Together these inputs and outputs can be analytically related to each other in their response time-histories as input-output relations. Consider, for example, the $(i + 1)$ most recent outputs, $\Delta y(k - i)$ to $\Delta y(k)$. In the absence of disturbance inputs, they can be expressed as functions of the state and control:

$$\Delta y(k-i) = H_x \Delta x(k-i) + H_u \Delta u(k-i) \tag{25}$$

$$\Delta y(k-i+1) = H_x \Phi \Delta x(k-i) + H_x \Gamma \Delta u(k-i) + H_u \Delta u(k-i+1) \tag{26}$$

$$\vdots$$

$$\Delta y(k) = H_x \Phi^i \Delta x(k-i) + H_x \Phi^{i-1} \Gamma \Delta u(k-i) + H_x \Phi^{i-2} \Gamma \Delta u(k-i+1) + \cdots + H_x \Gamma \Delta u(k-1) + H_u \Delta u(k)$$

Or

$$y(k) = T \Delta x(k) + B u(k) \tag{28}$$

where,

$$y(k) = [\Delta y(k-i)^T \quad \Delta y(k-i+1)^T \quad \cdots \quad \Delta y(k)^T]^T \tag{29}$$

$$u(k) = [\Delta u(k-i)^T \quad \Delta u(k-i+1)^T \quad \cdots \quad \Delta u(k)^T]^T \tag{30}$$

$$T = \begin{bmatrix} H_x \\ H_x \Phi \\ \vdots \\ H_x \Phi^i \end{bmatrix} \tag{31}$$

$$B = \begin{bmatrix} H_u & 0 & \cdots & 0 \\ H_x \Gamma & H_u & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ H_x \Phi^{i-1} \Gamma & H_x \Phi^{i-2} \Gamma & \cdots & H_u \end{bmatrix} \tag{32}$$

with $y(k) \in R^{(i+1)q}$, $u(k) \in R^{(i+1)m}$, $T \in R^{(i+1)q \times i}$, and $B \in R^{(i+1)q \times (i+1)m}$.

Since $i = 2$ for the process, the last 3 samples are considered:

$$\Delta y(k-2) = H_x \Delta x(k-2) + H_u \Delta u(k-2) \tag{33}$$

$$\Delta y(k-1) = H_x \Phi \Delta x(k-2) + H_x \Gamma \Delta u(k-2) + H_u \Delta u(k-1) \tag{34}$$

$$\Delta y(k) = H_x \Phi^2 \Delta x(k-2) + H_x \Phi \Gamma \Delta u(k-2) + H_x \Gamma \Delta u(k-1) + H_u \Delta u(k) \tag{35}$$

and

$$y(k) = \left[\Delta y(k-2)^T \quad \Delta y(k-1)^T \quad \Delta y(k)^T\right]^T \tag{36}$$

$$u(k) = \left[\Delta u(k-2)^T \quad \Delta u(k-1)^T \quad \Delta u(k)^T\right]^T \tag{37}$$

$$T = \begin{bmatrix} H_x \\ H_x \Phi \\ H_x \Phi^2 \end{bmatrix} \tag{38}$$

$$B = \begin{bmatrix} H_u & 0 & 0 \\ H_x \Gamma & H_u & 0 \\ H_x \Phi \Gamma & H_x \Gamma & H_u \end{bmatrix} \tag{39}$$

Substituting the values obtained for Φ, Γ, H_x and H_u from Equations 15 - 18 respectively, into Equations 38 and 39 gives:

$$T = \begin{bmatrix} 0.00080 & 0 \\ 0.00050 & 0.00500 \\ 0.00080 & 0.00008 \\ 0.00050 & 0.00499 \\ 0.00080 & 0.00016 \\ 0.00050 & 0.00497 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.000004 & 0 & 0 \\ 0.000499 & 0 & 0 \\ 0.000012 & 0.000004 & 0 \\ 0.000498 & 0.000499 & 0 \end{bmatrix}$$

with $y(k) \in R^6$, $u(k) \in R^3$, $T \in R^{6\times 2}$, and $B \in R^{6\times 3}$, since $i = q = 2$ and $m = 1$. Making $T\Delta x(k)$ the subject of Equation 28:

$$T\Delta x(k) = y(k) - Bu(k) \tag{40}$$

Pre-multiplying Equation 40 by $\Omega \in R^{N\times(i+1)q}$, and imposing the condition,

$$\Omega T = \mathbf{0} \tag{41}$$

gives

$$\Omega T \Delta x(k) = \Omega[y(k) - Bu(k)] = \mathbf{0} \tag{42}$$

with

$$\Omega = \begin{bmatrix} -0.81234 & -0.00702 & 0.41440 & -0.00653 & 0.41018 & -0.00644 \\ 0 & -0.56376 & 0.10470 & 0.78298 & -0.10616 & -0.21653 \\ 0 & -0.17521 & -0.68796 & 0.08274 & 0.69457 & 0.08216 \\ 0 & -0.56237 & 0.10362 & -0.21631 & -0.10621 & 0.78418 \end{bmatrix}$$

Where, Ω is called the *parity matrix* and each row of Ω is called a *parity vector*. For the process, $\Omega \in R^{4\times 6}$. The *parity matrix* is computed by using the "null" function in Matlab, so:

$\Omega = \text{null}(T^T)$

The reason why T is transposed is because Matlab gives the right null space, whereas the left null space is required. From the Matlab definition, Ω is defined as an orthonormal basis for the null space of T^T obtained from the singular value decomposition. That is, $\Omega * T^T$ has negligible elements, and $\Omega * \Omega^T = I$, where I is the identity matrix.

Each row of Equation 42 defines a unique analytical relation between the inputs and the outputs, which is independent of the state and is called a *parity equation*. From its definition, Ω is a null space, and its dimension, N can at most equal $[(i + 1)q - i]$. Physically, N represents the number of independent analytical relations that can be derived from the dynamics of the system, and is equal to four for the process. The *parity matrix* for the process is defined as:

$$\Omega = \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{pmatrix} \tag{43}$$

where Ω_1, Ω_2, Ω_3 and $\Omega_4 \in R^{1\times 6}$ are the *parity vectors*. Then the *parity equations* are:

$$\begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{pmatrix} \cdot \left\{ \begin{bmatrix} \Delta y(1)^T \\ \Delta y(2)^T \\ \Delta y(3)^T \end{bmatrix} - \begin{bmatrix} H_u & 0 & 0 \\ H_x\Gamma & H_u & 0 \\ H_x\Phi\Gamma & H_x\Gamma & H_u \end{bmatrix} \cdot \begin{bmatrix} \Delta u(1)^T \\ \Delta u(2)^T \\ \Delta u(3)^T \end{bmatrix} \right\} = \mathbf{0} \tag{44}$$

which show the four analytical relationships obtained from substitution of Equations 36, 37 and 43 into Equation 42. The length of the observation window in the original equation, $(i + 1)$ is generally $(s + 1)$, where s is defined as the *order* of the *parity space* Ω, which has been selected as two (since there are two states in the dynamic model of the tank system).

The residual vector is defined as:

$$r(k) = \Omega[y(k) - Bu(k)] \tag{45}$$

The residual vector is equal to zero if there is no model-mismatch (fault-free and noise free situation, i.e. $\Delta v=\Delta d=\Delta n=S=M=0$). The model used for the process does not include disturbance inputs in its model. The resulting parity equations represent hyper-planes in the input-output space on which the true inputs and outputs of the system propagate regardless of the state and disturbance inputs; each parity equation describes a part of the dynamics of the system. By replacing the actual outputs and inputs by the measured outputs and commanded inputs, a new window of available observations z and commanded inputs u^c can be defined:

$$z(k) = [\Delta z(k-n)^T \quad \Delta z(k-n+1)^T \quad \cdots \quad \Delta z(k)^T]^T \tag{46}$$

$$u^c(k) = [\Delta u^c(k-n)^T \quad \Delta u^c(k-n+1)^T \quad \cdots \quad \Delta u^c(k)^T]^T \tag{47}$$

Using the measured outputs and commanded inputs in the parity equations, a residual estimate that can serve as a fault detection criterion is formed:

$$\hat{r}(k) = \Omega[z(k) - Bu^c(k)] \tag{48}$$

then substituting Equations 39, 46 and 47 into Equation 47, for $k = 3$ gives:

$$\hat{r}(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \\ r_3(k) \\ r_4(k) \end{bmatrix} = \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{pmatrix} \cdot \left\{ \begin{bmatrix} \Delta z(k-2)^T \\ \Delta z(k-1)^T \\ \Delta z(k)^T \end{bmatrix} - \begin{bmatrix} H_u & 0 & 0 \\ H_x\Gamma & H_u & 0 \\ H_x\Phi\Gamma & H_x\Gamma & H_u \end{bmatrix} \cdot \begin{bmatrix} \Delta u^c(k-2)^T \\ \Delta u^c(k-1)^T \\ \Delta u^c(k)^T \end{bmatrix} \right\} \tag{49}$$

where

r_1 is Residual 1

r_2 is Residual 2

r_3 is Residual 3

r_4 is Residual 4

Residual 1 for example, has the following analytical relationship, as a result of substituting Equations 19 and 21 into the Residual 1 parity equation formed from Equation 49:

$$r_1(k) = \Omega_1 \left\{ \begin{pmatrix} [M(k-2)\cdot\Delta y(k-2) + \Delta v(k-2) + \Delta n(k-2)]^T \\ [M(k-1)\cdot\Delta y(k-1) + \Delta v(k-1) + \Delta n(k-1)]^T \\ [M(k)\cdot\Delta y(k) + \Delta v(k) + \Delta n(k)]^T \end{pmatrix} - \begin{bmatrix} H_u & 0 & 0 \\ H_x\Gamma & H_u & 0 \\ H_x\Phi\Gamma & H_x\Gamma & H_u \end{bmatrix} \cdot \begin{pmatrix} \left[\frac{\Delta u(k-2) - \Delta d(k-2)}{S(k-2)}\right]^T \\ \left[\frac{\Delta u(k-1) - \Delta d(k-1)}{S(k-1)}\right]^T \\ \left[\frac{\Delta u(k) - \Delta d(k)}{S(k)}\right]^T \end{pmatrix} \right\} \quad (50)$$

If all the residuals are considered, then the following variables are known (provided $k \geq 3$):

Ω_1, Ω_2, Ω_3, Ω_4, Δy(k-2), $\Delta y(k-1)$, $\Delta y(k)$, $\Delta u(k-2)$, $\Delta u(k-1)$, $\Delta u(k)$, H_x, H_u, Φ and Γ

while the following are unknown:

$M(k-2)$, $M(k-1)$, $M(k)$, $S(k-2)$, $S(k-1)$, $S(k)$, $\Delta d(k-2)$, $\Delta d(k-1)$, $\Delta d(k)$, $\Delta v(k-2)$, $\Delta v(k-1)$, $\Delta v(k)$, $\Delta n(k-2)$, $\Delta n(k-1)$ and $\Delta n(k)$.

Since there are more unknown variables than are known, it is not possible to uniquely identify each of the unknown variables.

4.2.1.1 Fault Detection

A mismatch in either the observation or commanded inputs from the real outputs and inputs results in a non-zero estimated residual vector that serves as an indication of a failure. The error signature left on the residual vector by any failure is unique, and this can be used to detect the failure. This error signature should be a deviation in the residual corresponding to the shape and the duration of the fault. The signature should also have deviations with amplitudes that correspond to the severity of the faults, i.e. small faults should produce little deviations on the residual, while large faults should cause significant deviations. Furthermore, a single fault should only effect one residual, hence facilitate fault isolation. But, in practice this would not be feasible, since the process variables are dependent on each other. For example, if the actuator blows up, theory states that component s would be zero, and component Δd would take its extreme value (Section 4.1). However, the sensors would

both be affected since there would either be a large flow rate or no flow rate into Tank 1, which would cause components in Δv to be non-zero.

A fault is indicated if the modulus of the residual exceeds a certain (positive) threshold. This can be stated as:

$$\begin{aligned} |\hat{r}(k)| &\geq \Psi \qquad fault \quad has \quad occurred \\ |\hat{r}(k)| &< \Psi \qquad fault-free \quad case \end{aligned} \tag{51}$$

where Ψ is the residual threshold setting. The threshold must be carefully chosen in order to be sensitive to relatively small faults and large enough not to detect process noise as spurious faults.

Since this method does not perform well in the presence of noise (Section 2.2.3), a Kalman filter was used to filter out the noise of the output measurement signals. The revised fault detection scheme is shown in Figure 4-2. The output of the Kalman filter is seen too be z_{kf}, which is the new input to the fault detection scheme.

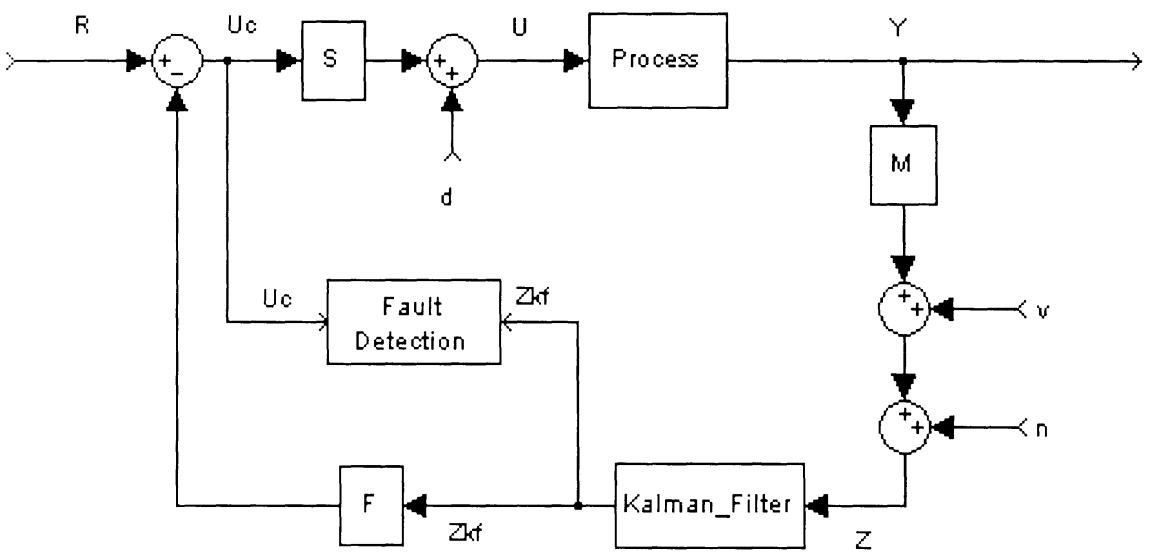


Figure 4-2 Fault Detection system with Kalman Filter

4.2.1.2 Theory on Kalman Filter

The following theory has been extracted from Chipperfield and Fleming, 1993. The Kalman filter is essentially an observer, as shown in Figure 4-3.

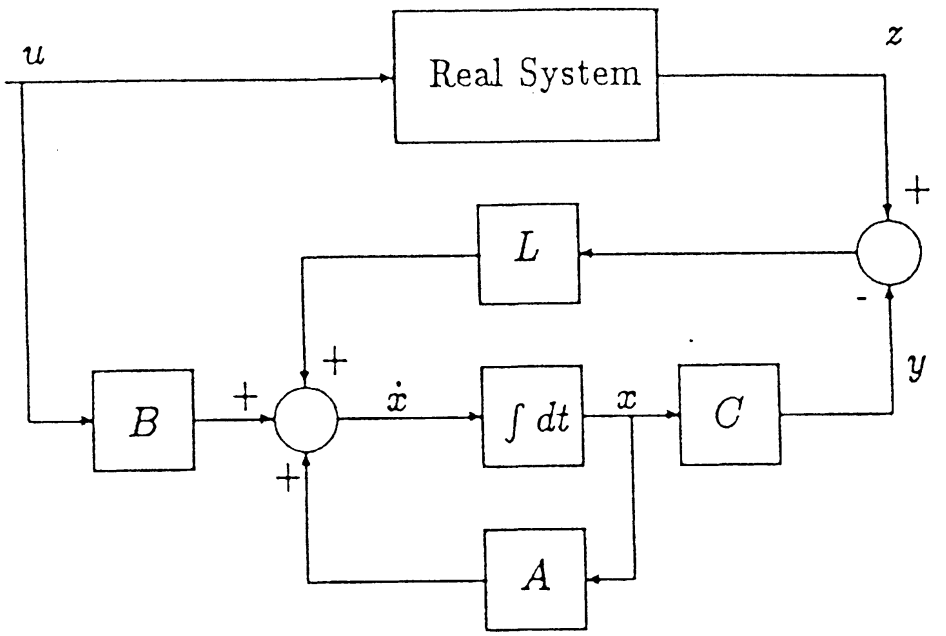


Figure 4-3 Standard observer configuration (Fig 10-1 Chipperfield and Fleming, 1993)

The observer dynamics can be written as:

$$\Delta\hat{x}(k+1) = \Phi\Delta\hat{x}(k) + \Gamma\Delta u(k) + L\cdot\left[\Delta z(k) - \Delta y(k)\right] \quad (52a)$$

$$\Delta\hat{y}(k) = H_x\Delta\hat{x}(k) + H_u\Delta u(k) \quad (52b)$$

Substitution for Equation 52b into Equation 52a gives:

$$\Delta\hat{x}(k+1) = \left[\Phi - LH_x\right]\hat{x}(k) + \left[\Gamma - LH_u \mid L\right]\cdot\begin{bmatrix}\Delta u(k)\\ \Delta z(k)\end{bmatrix} \quad (53)$$

The discrete Kalman filter differs from the observer structure in that the gain matrix is variable and is calculated recursively in three steps:

1. Prediction

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k)$$
$$P(k+1) = \Phi P(k)\Phi^T + Q$$

State error covariance propagation

2. Gain Update

$$L = P(k)\cdot H_x^T \cdot \left[H_x \cdot \Delta P(k)\cdot H_x^T + R\right]^{-1}$$

3. Measurement

$$\hat{x}(k+1) = \hat{x}(k) + L\cdot\left[z(k) - H_x\hat{x}(k) - H_u u(k)\right]$$

State update

$$P(k) = \left[I - LH_x\right]\cdot P(k)$$

State error covariance update

The system model is used to forecast the state in response to a deterministic input, in the first step. The gain matrix, L is updated in the second step. In the final step, a state estimate is formed from the forecast and measurement, which is controlled by the Kalman gains. Both R and Q are weighting matrices. The covariance of the measurement noise, R, is a statistical property determined from the steady state output. The matrix, Q, is said to be a level of confidence in the linear model. In real terms, Q arises from the state error covariance between the system and the model. Large values of Q generate large Kalman gains, which put a low emphasis on the model output and greater emphasis on the incoming measurements. However, low values of Q put more emphasis on the model response than the system inputs. The values of R and Q for the program were found using an optimisation procedure (refer to Appendix E for results).

The R and Q matrices are:

$$R = \begin{bmatrix} 0.02585 & 0.00834 \\ 0.00834 & 0.01518 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The fault detection results obtained for the parity space methods with and without the Kalman filter are given in the next chapter.

4.2.2 Adaptive parity space

This method is an extension of the above method. If there are *ds* disturbances in the system, one can at most detect *(q - ds)* failures simultaneously. This implies that no simultaneous faults can be identified in the process, since the process has two sensors, and at most, six different faults can be injected into the process. Furthermore, it is not possible to simultaneously identify all sensor and actuator failures. To detect a plant (process) failure, the specific failure has to be modelled in the residual, and then only can the parameters be estimated. The theory for the method follows.

The residual error is defined as the difference between the residual (Equation 45) and its estimate (Equation 48):

$$\Delta r(k) = \hat{r}(k) - r(k) = \Delta r(k) = \Omega \cdot \left[(z(k) - Bu^c(k)) - (y(k) - Bu(k)) \right] \tag{54}$$

Substituting Equations 19 and 21 into Equation 54 gives:

$$\begin{aligned} \Delta r(k) &= \Omega\{(M(k)y(k) + v(k) + n(k) - Bu^c(k)) - [y(k) - B(S(k)u^c(k) + d(k))]\} \\ &= \Omega\{(M(k) - I)y(k) + v(k) + B[(S(k) - I)u^c(k) + d(k)]\} + n'(k) \end{aligned} \tag{55}$$

Where the scale factors, biases and measurement errors are contained in:

$$M(k) = block\ diagonal(m(k-i), m(k-i+1), \ldots, m(k)) \tag{56}$$

$$S(k) = block\ diagonal(s(k-i), s(k-i+1), \ldots, s(k)) \tag{57}$$

$$v(k) = [\Delta v(k-i)^T, \Delta v(k-i+1)^T, \ldots, \Delta v(k)^T]^T \tag{58}$$

$$d(k) = [\Delta d(k-i)^T, \Delta d(k-i+1)^T, \ldots, \Delta d(k)^T]^T \tag{59}$$

$$n'(k) = \Omega[\Delta n(k-i)^T, \Delta n(k-i+1)^T, \ldots, \Delta n(k)^T]^T \tag{60}$$

As a result,

$$M(k) = \begin{bmatrix} M(k-2) & 0 & 0 \\ 0 & M(k-1) & 0 \\ 0 & 0 & M(k) \end{bmatrix} \tag{61}$$

$$S(k) = \begin{bmatrix} S(k-2) & 0 & 0 \\ 0 & S(k-1) & 0 \\ 0 & 0 & S(k) \end{bmatrix} \tag{62}$$

$$v(k) = [\Delta v(k-2)^T, \Delta v(k-1)^T, \Delta v(k)^T]^T \tag{63}$$

$$d(k) = [\Delta d(k-2)^T, \Delta d(k-1)^T, \Delta d(k)^T]^T \tag{64}$$

$$n'(k) = \Omega[\Delta n(k-2)^T, \Delta n(k-1)^T, \Delta n(k)^T]^T \tag{65}$$

since i = 2 for the process. Also $y(k)$ is not available for the estimation scheme, so the residual error equation is approximated by using an estimate of the outputs, $\hat{y}(k)$:

$$\Delta r(k) = \Omega\{(M(k) - I)\hat{y}(k) + v(k) + B[(S(k) - I)u^c(k) + d(k)]\} + \Delta n''(k) \tag{66}$$

where the estimation error is defined by $n''(k)$, and

$$\hat{y}(k) = [\Delta\hat{y}(k-i)^T \; \Delta\hat{y}(k-i+1)^T \quad \cdots \quad \Delta\hat{y}(k)^T]^T \tag{67}$$

Substitution for i=2 yields,

$$\hat{y}(k) = \left[\Delta\hat{y}(k-2)^T \quad \Delta\hat{y}(k-1)^T \quad \Delta\hat{y}(k)^T\right]^T \tag{68}$$

The estimated output, $\Delta\hat{y}(k)$, is derived from a Kalman filter with gain K_k:

$$\Delta\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma\Delta u(k) + K_k[z(k) - \Phi\hat{x}(k) - \Gamma\Delta u(k)] \tag{69}$$

$$\Delta\hat{y}(k) = H_x\Delta\hat{x}(k) + H_u\Delta u(k) \tag{70}$$

With the assumption that the fault parameters are constant over the (observation) window length, then the vector of parameters to be identified is:

$$p(k) = [(1 - M(k))^T, v^T(k), (1 - S(k))^T, d^T(k)]^T \tag{71}$$

Then

$$\Delta r(k) = H_p(k)p(k) + n''(k) \tag{72}$$

where

$$H_p = \left[-\Omega\hat{y}(k), \quad \Omega, \quad -\Omega Bu^c(k), \quad \Omega B\right] \tag{73}$$

and is defined as the *fault-signature matrix*. Therefore the Residual 1 error is found through the use of Equations 37, 39, 43, 48, 61 - 66 and 68:

$$\Delta r_1(k) = \Omega_1 \left\{ \left(\left(\begin{bmatrix} M(k-2) & 0 & 0 \\ 0 & M(k-1) & 0 \\ 0 & 0 & M(k) \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} \Delta\hat{y}(k-2)^T \\ \Delta\hat{y}(k-1)^T \\ \Delta\hat{y}(k)^T \end{bmatrix} + \begin{bmatrix} \Delta v(k-2)^T \\ \Delta v(k-1)^T \\ \Delta v(k)^T \end{bmatrix} \right) \right.$$
$$+ \begin{bmatrix} H_u & 0 & 0 \\ H_x\Gamma & H_u & 0 \\ H_x\Phi\Gamma & H_x\Gamma & H_u \end{bmatrix} \cdot \left\{ \left(\left(\begin{bmatrix} S(k-2) & 0 & 0 \\ 0 & S(k-1) & 0 \\ 0 & 0 & S(k) \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} \Delta u^c(k-2)^T \\ \Delta u^c(k-1)^T \\ \Delta u^c(k)^T \end{bmatrix} \right) \right.$$
$$\left. \left. + \begin{bmatrix} \Delta d(k-2)^T \\ \Delta d(k-1)^T \\ \Delta d(k)^T \end{bmatrix} \right) \right\} \right\}$$
$$+ n''(k) \tag{74}$$

For all the residuals, then the following variables are known:

Ω_1, Ω_2, Ω_3, Ω_4, $\Delta\hat{y}(k-2)$, $\Delta\hat{y}(k-1)$, $\Delta\hat{y}(k)$, $\Delta u(k-2)$, $\Delta u(k-1)$, $\Delta u(k)$, $\Delta r_1(k)$, $\Delta r_2(k)$, $\Delta r_3(k)$, $\Delta r_4(k)$, H_x, H_u, Φ and Γ

while the following are unknown:

$M(k-2)$, $M(k-1)$, $M(k)$, $S(k-2)$, $S(k-1)$, $S(k)$, $\Delta d(k-2)$, $\Delta d(k-1)$, $\Delta d(k)$, $\Delta v(k-2)$, $\Delta v(k-1)$, $\Delta v(k)$.

Although, there are more known variables than are unknown, it is not possible to uniquely identify each of the unknown variables. However, a Least Squares (LS) estimate for $p(k)$ was used to yield the bias and scale factors. It was then decided to investigate the use of a Recursive Least Squares (RLS) estimate for $p(k)$ to yield the bias and scale factors. Since the parameters could change abruptly, the estimation scheme was then designed to weight the most recent measurements more, than the earlier measurements. A *forgetting factor* was then implemented in the RLS to account for this (details for the estimation methods can be found in Appendix D). Finally, a *stabilising factor* was also added to the RLS to overcome the possibility of limited excitation at the inputs (the limited excitation could cause the estimation scheme to be unstable). The theory of the RLS with stabilising and forgetting factors is described next.

4.2.2.1 Stabilised RLS with forgetting factor

The stabilised RLS algorithm with forgetting factor, λ (Bodson and Groszkiewicz, 1997), is obtained by including an additional term in the error function used by LS, the stabilising factor is denoted by α. Using the direct input error algorithm for LS estimation, the parameter matrix is defined,

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{pmatrix} \tag{75}$$

where θ_1 is Parameter , θ_2 is Parameter 2 and so forth until θ_N, which is the last parameter. The input error, e can be expressed as a linear equation in the form:

$$e = \theta^T \cdot z - u \tag{76}$$

Where z is the regressor vector, and u the system input. Then the error function is:

$$J(\theta(N)) = \sum_{k=1}^{N} [\theta^T(N) z(k) - u(k)]^2 \lambda^{N-k} + \alpha[\theta(N) - \theta(N-1)]^2 \tag{77}$$

The regressor vector, $z(k)$ should be permanently excited to ensure convergence of the parameters to the nominal values. The additional term changes the parameter matrix, θ. Setting $\partial J / \partial \theta(N) = 0$ gives:

$$\theta(N) = \left(\sum_{k=1}^{N} z(k) z^T(k) \lambda^{N-k} + \alpha I \right)^{-1} \bullet \left(\sum_{k=1}^{N} z(k) u^T(k) \lambda^{N-k} + \alpha \theta(N-1) \right) \tag{78}$$

This is not truly the equivalent of the batch solution of LS, because of the dependence on $\theta(N-1)$. The covariance matrix of the algorithm is:

$$P(N) = \left(\sum_{k=1}^{N} z(k) z^T(k) \lambda^{N-k} + \alpha I \right)^{-1} \tag{79}$$

The recursive form of the inverse of the covariance matrix is:

$$P^{-1}(N) = \lambda P^{-1}(N-1) + z(N) z^T(N) + \alpha(1-\lambda) I \tag{80}$$

Giving the initial condition $P^{-1}(0) = \alpha I$. The recursive formula for θ becomes:

$$\theta(N) = \theta(N-1) + P(N) z(N) [u^T(N) - z^T(N) \theta(N-1)] + \alpha\lambda P(N) [\theta(N-1) - \theta(N-2)] \tag{81}$$

The residual error (Equation 72) can be rewritten as:

$$\begin{aligned}\Delta r(k) &= H_p(k)\cdot p(k) + n''(k)\\ \therefore -n''(k) &= H_p(k)\cdot p(k) - \Delta r(k)\end{aligned} \tag{82}$$

The residual error equation is now in the form of input error equation (Equation 76). Comparing the two equations yields:

$$u = \Delta r(k) \tag{83}$$

$$\theta^T = p(k) \tag{84}$$

$$z = H_p(k) \tag{85}$$

$$e = -n''(k) \tag{86}$$

Since the estimation error, $n''(k)$ can be either positive or negative, then it is assumed that

$$e = n''(k) \tag{87}$$

Substituting Equations 83, 84, 85 and 87 into Equation 81, yields the following recursive formula:

$$p(N) = p(N-1) + P(N)H_p(N)[\Delta r^T(N) - H_p{}^T(N)p(N-1)] + \alpha\lambda P(N)[p(N-1) - p(N-2)] \tag{88}$$

Forgetting Factor and Stabilising Factor values

The optimal values for the forgetting factor, λ and stabilising factor, α were found through trial and error (Detailed results in Appendix F). Given that old data is weighted by a factor of less than 0.05 if it is more than $-3\Delta t/\ln[\lambda]$ time units old, the forgetting factor must be dependent on the sampling time, Δt (refer to Appendix D for more details). The process uses a sampling time of 0.1[s], and in practice, $0.9<\lambda<1.0$ with a sampling rate of 1[s]. Therefore, to get the appropriate forgetting factor for the process, the following equation must be solved:

$$\begin{aligned}\frac{-3\Delta t_{1s}}{\ln\lambda_{1s}} &= \frac{-3\Delta t_{0.1s}}{\ln\lambda_{0.1s}}\\ \therefore \frac{1}{\ln\lambda_{1s}} &= \frac{0.1}{\ln\lambda_{0.1s}}\\ \therefore \ln\lambda_{0.1s} &= 0.1\cdot\ln\lambda_{1s}\\ \therefore \lambda_{0.1s} &= \lambda_{1s}{}^{0.1}\end{aligned} \tag{89}$$

Where λ_{1s} and $\lambda_{0.1s}$ are the forgetting factors for sampling times of 1[s] and 0.1[s] respectively, and Δt_{1s} is a 1[s] sampling time and $\Delta t_{0.1s}$ is a 0.1[s] sampling time. Then, $\lambda_{0.1s}$ is the corrected forgetting factor that should be used for a 0.1[s] sampling time, on the RLS estimation. For consistency, the stabilising factor was also scaled by the same value as that used for the forgetting factor. The optimum values for λ and α are:

$$\lambda = 6.64 \times 10^{-1}$$
$$\alpha = 5.75 \times 10^{-9}$$

4.2.2.2 Fault detection

Faults are detected by estimating the parameters of the matrix *p(k)*. There are six parameters in *p(k)*, which can be said to be the fault indicators or residuals. These are:

m_1 – Sensor 1 scale residual

v_1 – Sensor 1 bias residual

m_2 – Sensor 2 scale residual

v_2 – Sensor 2 bias residual

s – Actuator scale residual

d – Actuator bias residual

Faults are then detected when either of the residuals exceeds a certain positive or negative threshold. This can be stated as:

$$\begin{aligned} |p_j(k)| \geq \Psi \quad & \textit{fault has occurred} \\ |p_j(k)| < \Psi \quad & \textit{fault-free case} \end{aligned} \tag{90}$$

where Ψ is the residual threshold setting and j is any parameter. The complete fault detection scheme for the adaptive parity space method of fault detection is shown in the following diagram. The inputs into the fault detection scheme are seen to be u^c, z and y_{est} ($\hat{y}$).

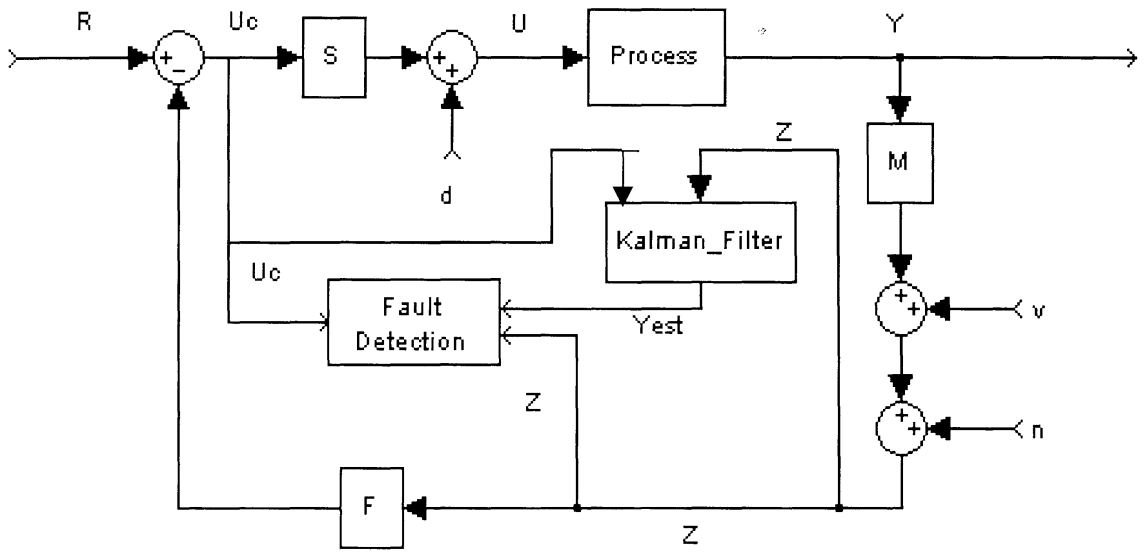


Figure 4-4 Adaptive Parity Space Fault Detection Scheme

4.2.3 Observer

The following theory is extracted from Isermann, 1984 and Frank, 1990. This is one of the simplest methods of fault detection. The method essentially reconstructs the outputs of the system from the measurements, with the use of a Kalman filter. The resultant estimation error is then used as the fault indicator or residual. A Kalman filter in the form of Equations 69 and 70, is used to generate an estimate of the outputs. Residuals are then generated as the error between the actual outputs of the system, $\Delta z(k)$, and the estimated outputs, $\Delta\hat{y}(k)$:

$$\hat{r}(k) = \Delta z(k) - \Delta\hat{y}(k) \tag{91}$$

Substituting Equation 19 into Equation 91 gives:

$$\hat{r}(k) = \begin{pmatrix} r_1(k) \\ r_2(k) \end{pmatrix} = M(k)\Delta y(k) + \Delta v(k) + \Delta n(k) - \Delta\hat{y}(k) \tag{92}$$

where r_1 and r_2 are the two residuals for the method (since there are two outputs), and k is the most recent sample. Then substituting Equations 11 and 70 into Equation 92 gives:

$$\begin{pmatrix} r_1(k) \\ r_2(k) \end{pmatrix} = M(k)\cdot\left[H_x\Delta x(k) + H_u\Delta u(k)\right] + \Delta v(k) + \Delta n(k) - H_x\Delta\hat{x}(k) - H_u\Delta u(k) \quad (93)$$
$$= \left[M(k) - I\right]\cdot H_u\Delta u(k) + M(k)H_x\Delta x(k) - H_x\Delta\hat{x}(k) + \Delta v(k) + \Delta n(k)$$

Then substituting Equation 21 into Equation 93 gives:

$$\begin{pmatrix} r_1(k) \\ r_2(k) \end{pmatrix} = \left[M(k) - I\right]\cdot H_u\cdot\left[S(k)\Delta u^c(k) + \Delta d(k)\right]\cdot + M(k)H_x\Delta x(k) - H_x\Delta\hat{x}(k) + \Delta v(k) + \Delta n(k) \quad (94)$$

Finally, substitution of Equation 9 into Equation 94 yields:

$$\begin{pmatrix} r_1(k) \\ r_2(k) \end{pmatrix} = M(k)H_x\Delta x(k) - H_x\Delta\hat{x}(k) + \Delta v(k) + \Delta n(k) \quad (95)$$

which shows that the residual equations are concerned with only the states and the state estimations of the process, since $H_u = 0$, for the process. Furthermore, Equation 95 shows that only the sensor faults ($M(k)$ and $\Delta v(k)$) and the measurement noise are directly incorporated into the residual equations. Although the actuator faults ($S(k)$ *and* $\Delta d(k)$) are not shown in the residual equations for the process, the system states and their estimations will be effected by the actuator faults. However, the fact that the actuator faults are not incorporated into the residual equations for $H_u = 0$, could be a possible shortfall in the fault detection results of the observer method. Further analysis of the residual equations shows that there are seven unknown variables in:

$M(k)$, $S(k)$, $\Delta d(k)$, $\Delta v(k)$ and $\Delta n(k)$

while the following are known:

$\Delta\hat{y}(k)$, $\Delta x(k)$, $\Delta\hat{x}(k)$, H_x, Φ and Γ.

Since there are more unknown variables than are known, it is not possible to uniquely identify each of the unknown variables. The two residuals are then used to make a decision on whether a fault has occurred if either exceeds the absolute value of a certain threshold, ψ (as seen in Equation 51).

This observer method of fault detection is shown in the following diagram, where z and y_{est} ($\hat{y}$) are seen to be the inputs to the fault detection scheme.

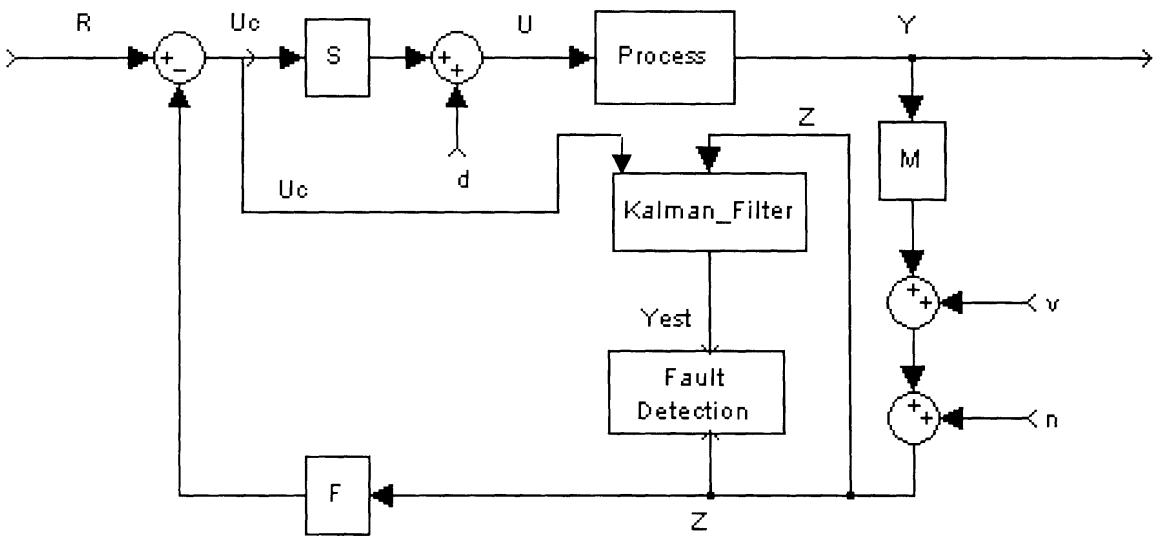


Figure 4-5 Observer Fault Detection Scheme

4.3 Concluding Remarks

It is expected that the adaptive parity space method of fault detection should give the best fault detection results, followed by the parity space and the observer methods. The previous statement is made after considering the number of residuals that are generated for each method. The adaptive parity space method uses six residuals, while the parity space and observer methods use four and two residuals, respectively.

Since the adaptive parity space method uses three "modules":

1. Parity space
2. Kalman filter
3. Parameter estimation

in its implementation, it is the most difficult method to implement of the three fault detection methods. The observer method uses two "modules":

1. Observer
2. Kalman filter

The parity space has a single "module", which is fairly complicated. As a result, the ease of implementation for both the observer and parity space method is approximately equal. The computational burden is expected to be the greatest for the adaptive parity space method, compared to that of the parity space and observer methods.

The following chapter, which shows the simulation studies of the methods, should test these expectations.

Chapter 5 FAULT DETECTION SIMULATION STUDIES

This chapter describes the simulation studies of the three chosen methods of fault detection, which are parity space, adaptive parity space and observer. Analyses of the results obtained are also described.

5.1 Overview of Simulation Studies

As a general rule, the number of faults that can be detected is equal to the number of linearly independent sensors, q in a MIMO case. Additionally, if there are disturbances, ds in the system, (q - ds) is/are the most fault(s) that can be detected. Furthermore, it is not physically meaningful to identify all failures because the system model, which is "known" – is being used to identify all the "unknown" failures. As a result, some information should be assumed to be reliable (e.g. from Sensor 2 in the process), while some types of failures should be validated by alternate procedures so as to remove them from the present model and incorporate the rest of the failures.

All simulations were performed in the Matlab programming language. A Lotus 1-2-3 spreadsheet was used to input parametric data required by the simulation program. The spreadsheet, "Info_ListP&P.wk1", contained the data used to generate the models and signals as well was values for parameters. The simulation program, "P_P.m" and data file can be found on the enclosed disc.

In order to test the three fault detection systems, the following initial parameters were set for the simulations:

- The initial operating conditions, x_0, u_0 and y_0 were set to zero.
- The process was required to track a setpoint, which was a step waveform of 1V amplitude from a steady state value of 0V.

- The duration of the simulation was 250[s] which shows the transient response of the system simulated, steady state settling time = 304[s] of the closed loop response from Section 3.4). The reason that the simulation investigations were performed in the transient state was due to requirements set by the minerals extraction industry. The sampling time was set at 0.1[s], and the initial delay of the setpoint was 1[s].
- Noise injected into the system was 5% of the setpoint amplitude and uniformly distributed for the duration of the simulation. Thus the amplitude of the noise at each sample was chosen randomly from a uniform distribution between 0V and 0.05V.
- Six faults were simulated as square waveforms with amplitudes of 0.5V and each with a duration of 10[s]. The six faults simulated were:

    1. Sensor 1 Scale fault
    2. Sensor 2 Scale fault
    3. Sensor 1 Bias fault
    4. Sensor 2 Bias fault
    5. Actuator Scale fault
    6. Actuator Bias fault

  The scale faults were simulated with an offset of 1V, which is the final value of the step setpoint. This is necessary since the scale faults are multiplicative faults and therefore have to be initiated at a voltage offset equal to the amplitude of the setpoint in order for the fault to be correctly implemented. All the faults were initiated at different times, with no faults occurring simultaneously. The reason for this is that none of the methods investigated, are able to detect multiple simultaneous faults on the process under investigation, for reasons given in Sections 2.7.1, 2.7.3 and 4.2.2. Furthermore, it would be difficult to differentiate between the simultaneous faults without the implementation of a suitable fault diagnosis architecture. The investigation under consideration deals only with fault detection.

The six faults were chosen for simulation for investigative purposes, in order to establish the theory stated in the opening paragraph of this section.

The results of the fault detection schemes with varying parameters, such as fault duration, fault amplitude and noise level, will be discussed later in the chapter.

The following diagram shows a typical simulation of the process under the faulty conditions stated and the faults numbered as previously stated, at their activated postions.

Graph	Colour	Description
First	Red	Setpoint, R
First	Green	Output Measurements, z
First	Magenta	Actuator Scale Fault, S
First	Cyan	Actuator Bias Fault, d
First & Second	Blue	Commanded Input, U^c
Second	Red	Noise, n
Second	Green	Kalman estimate, $\hat{y}$
Second	Magenta	Sensor Scale Faults, M
Second	Cyan	Sensor Bias Faults, v

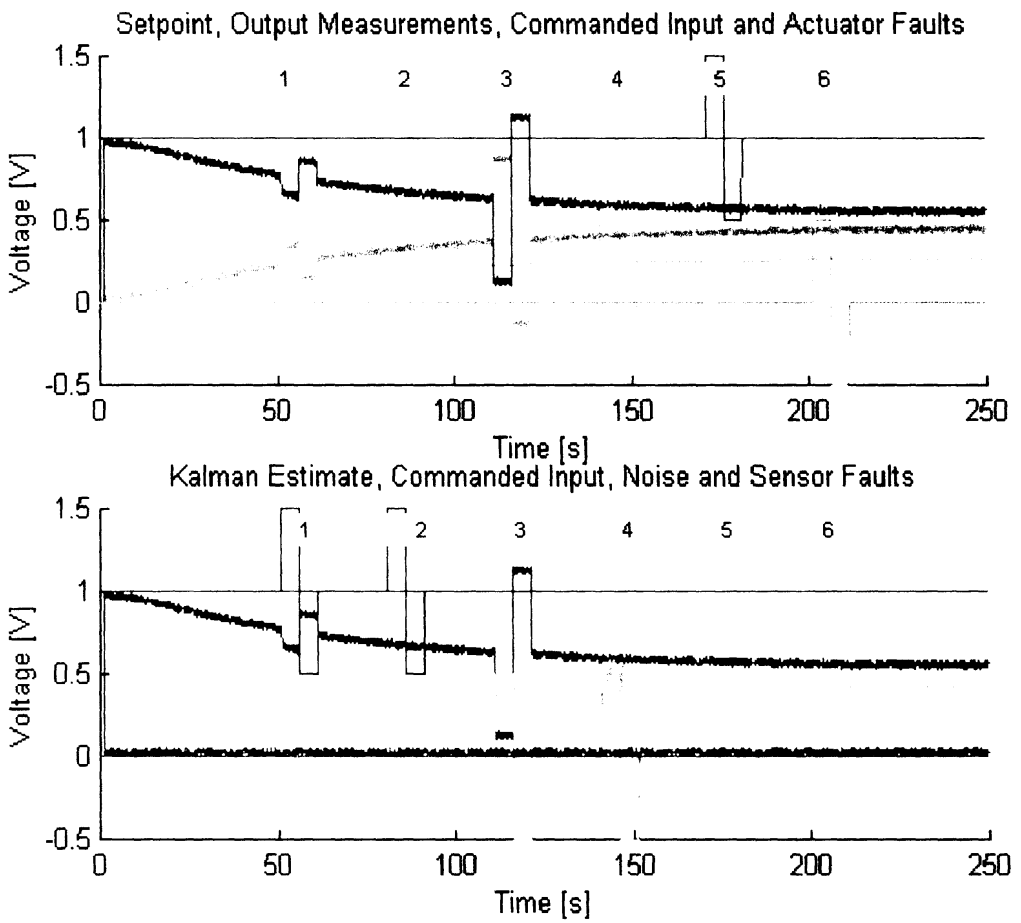


Figure 5-1 A Typical Simulation of the Process to a Step with Faults included

The results of the fault detection schemes with varying parameters, such as fault duration, fault amplitude and noise level, will be discussed later in the chapter.

The following diagram shows a typical simulation of the process under the faulty conditions stated and the faults numbered as previously stated, at their activated postions.

Graph	Colour	Description
First	Red	Setpoint, R
First	Green	Output Measurements, z
First	Magenta	Actuator Scale Fault, S
First	Cyan	Actuator Bias Fault, d
First & Second	Blue	Commanded Input, U^c
Second	Red	Noise, n
Second	Green	Kalman estimate, $\hat{y}$
Second	Magenta	Sensor Scale Faults, M
Second	Cyan	Sensor Bias Faults, v

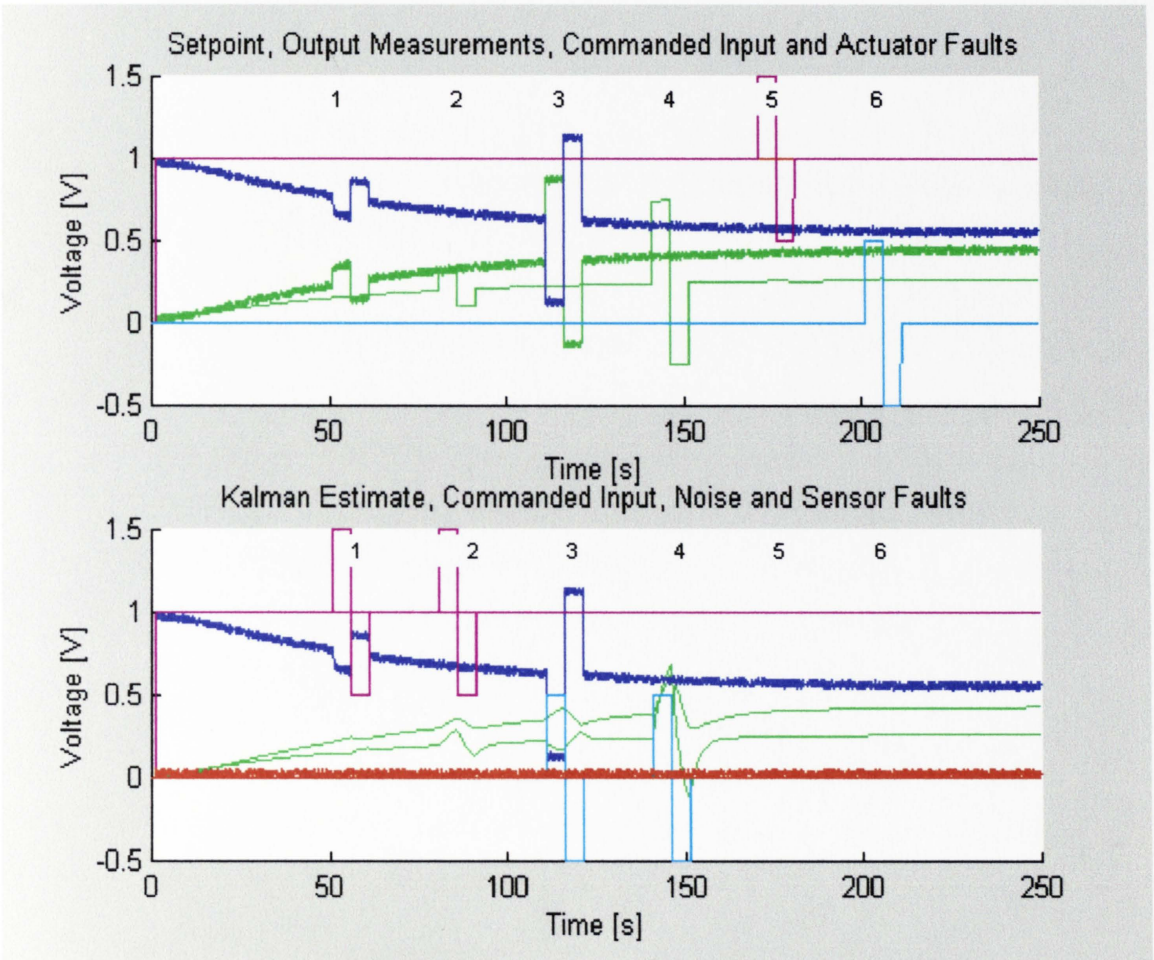


Figure 5-1 A Typical Simulation of the Process to a Step with Faults included

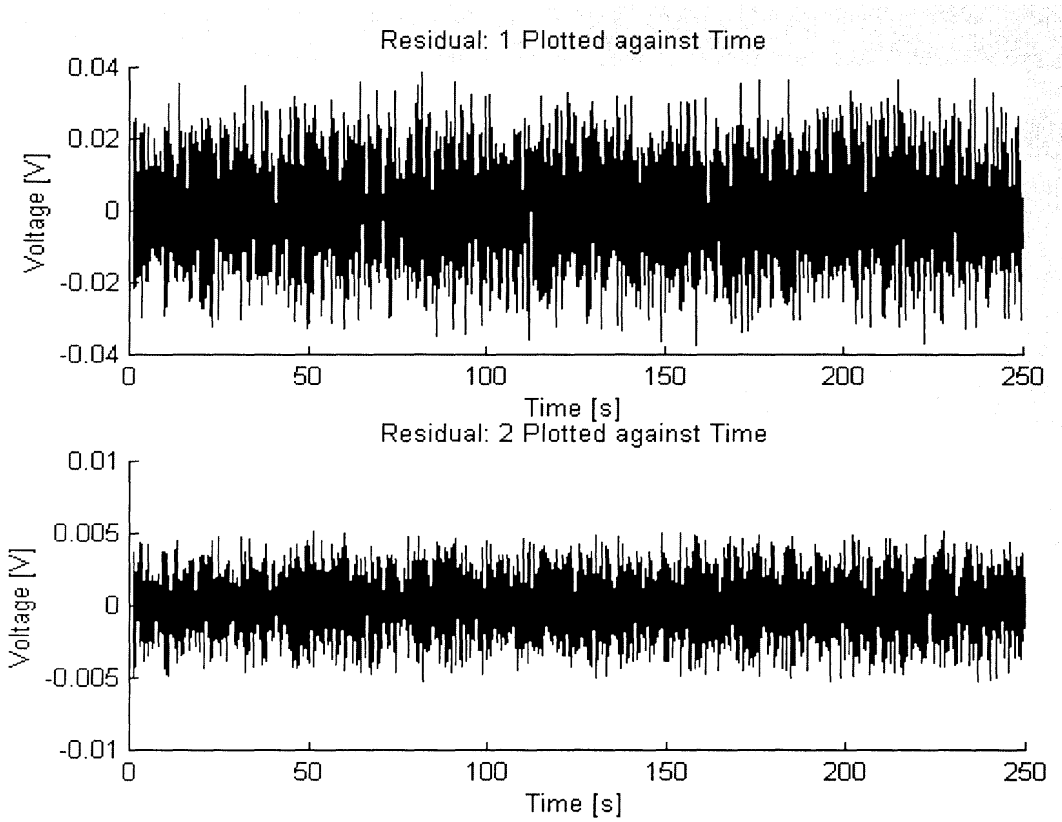


Figure 5-2 Parity Space Residuals 1 & 2 in a fault-free environment

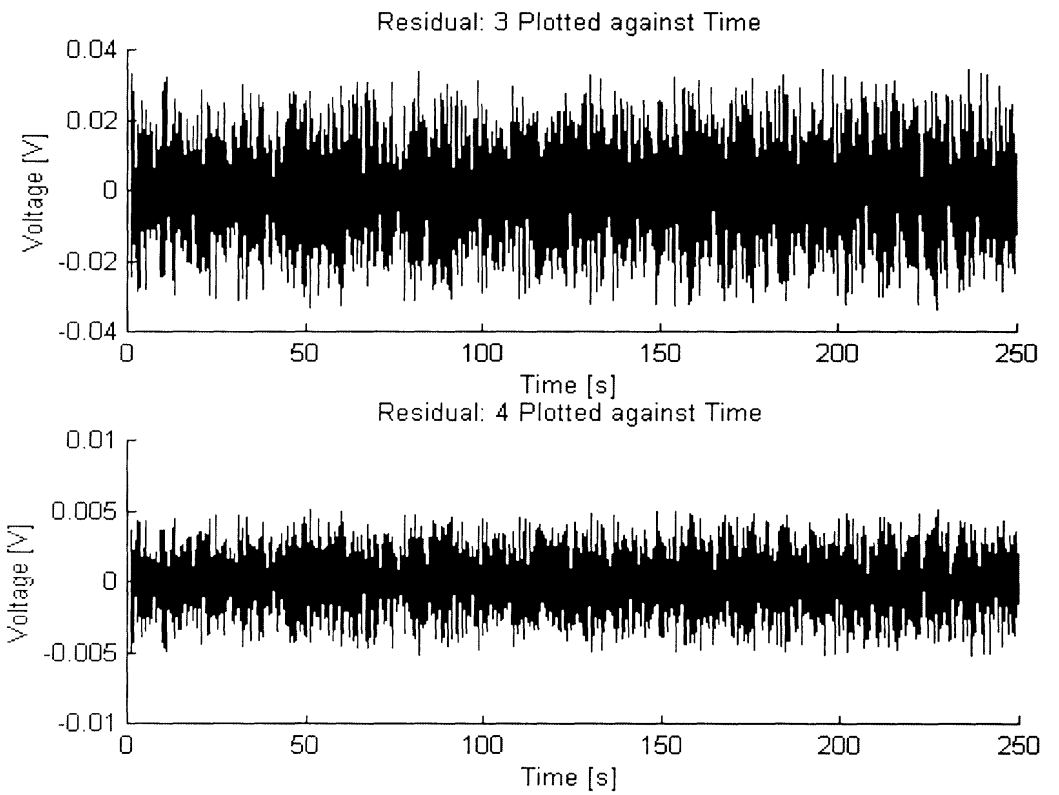


Figure 5-3 Parity Space Residuals 3 & 4 in a fault-free environment

From the above graphs, it can be seen that Residuals 1 & 3 are noisier than Residuals 2 & 4. This is because the noise signal is only being subject to the output of Tank 2 in the process (*G1(s)*) and the output of Tank 2 is incorporated into Residuals 1 & 3, while the output of Tank 1 (*G2(s)*) is incorporated into Residuals 2 & 4. If there were no noise injected into the system, the residuals would be closer to zero. Also the noise seen on the residuals has zero mean.

The largest value out of the fault-free residuals is less than 0.04, which is obtained from Residual 1. Therefore, the chosen threshold for the method was 0.04. A fault is then indicated if the modulus of any residual exceeds the threshold (Equation 51), in other words:

$|\hat{r}| >= 0.04$; fault has occurred

$|\hat{r}| < 0.04$; fault-free case

where, $\hat{r}$ = [Residual 1 Residual 2 Residual 3 Residual 4]^T

for the parity space method of fault detection.

The following graphs show the residuals subject to the six faults described previously.

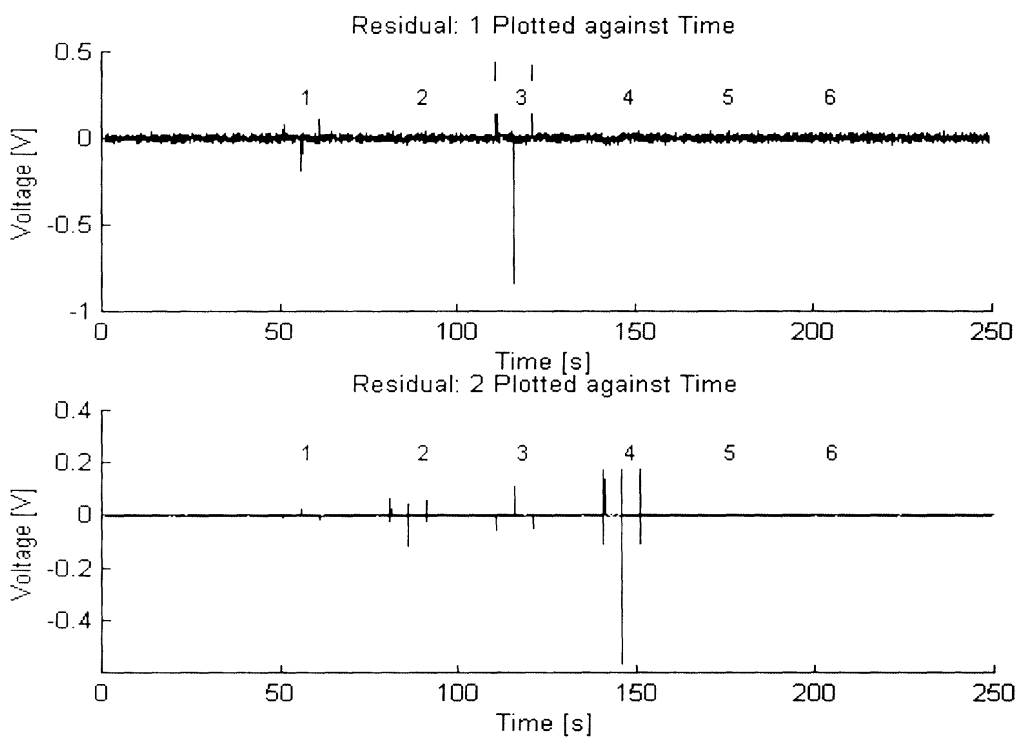


Figure 5-4 Parity Space Residuals 1 & 2 subject to faults

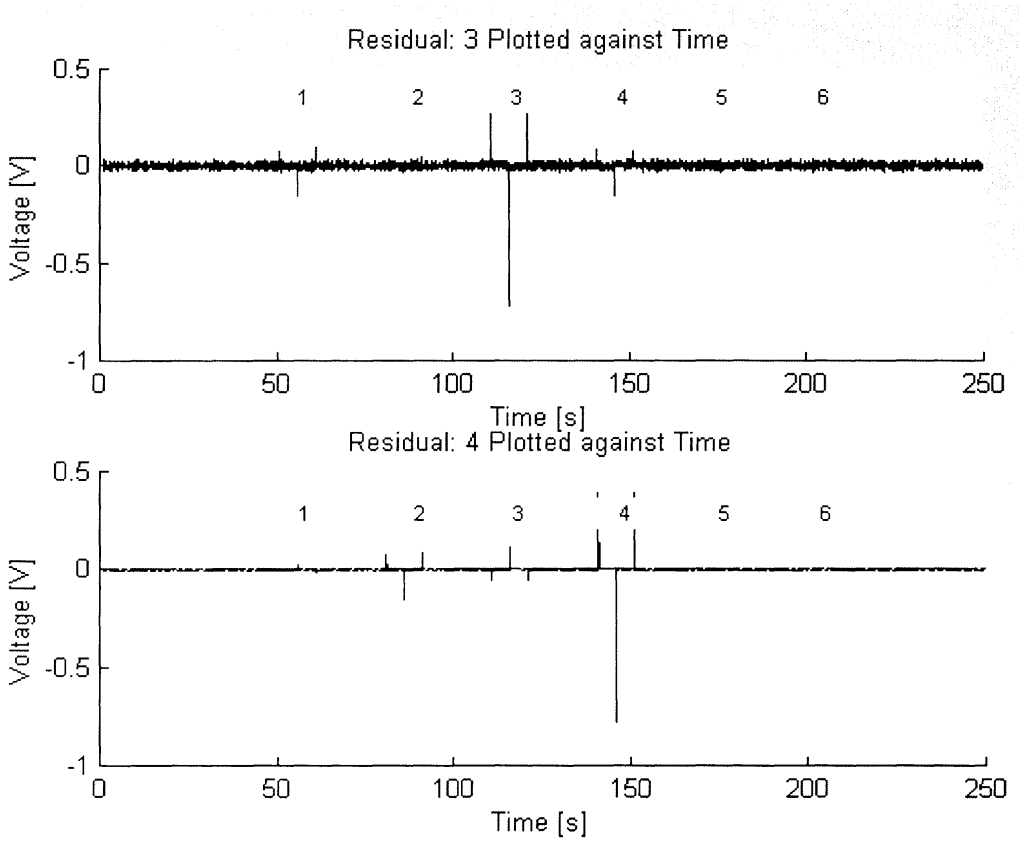


Figure 5-5 Parity Space Residuals 3 & 4 subject to faults

From the graphs, it can be seen that Residuals 1 and 3 show larger deviations for the Sensor 1 faults (positions 1 and 3), while little or no deviation for the Sensor 2 faults (positions 2 and 4). Residuals 2 and 4 show a large deviation for the Sensor 2 bias fault (position 4) and smaller deviations for Sensor 2 scale fault (position 2) and Sensor 1 bias fault (position 3), and a slight deviation for the Sensor 1 scale fault (position 1). However, none of the residuals show visible deviations at the positions (5 and 6) of the actuator faults. Also, the shape of the deviations resemble "spikes", which show that the parity space method shows significant residual deviations only in the transient states of the fault.

In order to see the fault detection ability of the parity space method, the positions where the residuals have exceeded the threshold have to be obtained. The following diagram shows the fault indication graph for the parity space method of fault detection.

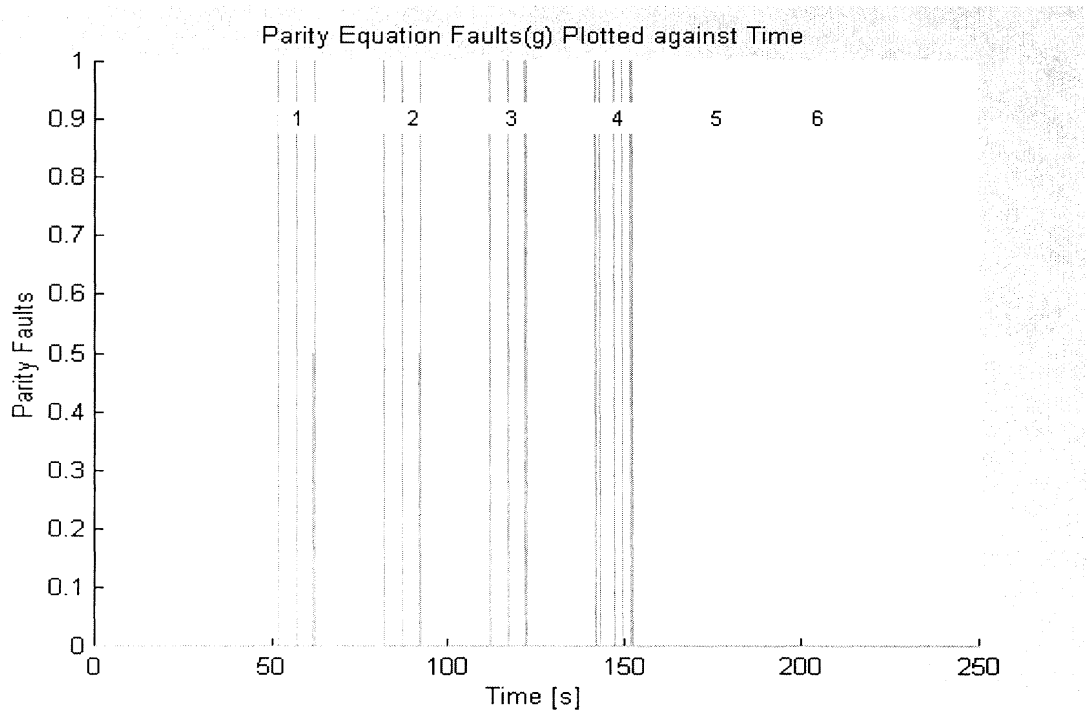


Figure 5-6 Parity Space Fault Indication

Explanation of fault indication graph:

Any non-zero value in a fault indication graph shows a position in time when a fault has been detected. Ideally, the faults should be detected over the duration of the fault in positions 1 to 6, and without any delay in the fault detection. The faults detected outside the fault region are spurious.

Comment on fault indication graph:

The fault indication graph (Figure 5.6) shows spikes at the positions where the faults have been detected. It can be seen that the actuator faults are still left undetected (positions 5 & 6) and that the sensor faults are detected (positions 1,2,3 and 4), but not fully detected over the duration of the fault. However, it is still possible that the residuals could be detecting the actuator faults, since only one threshold is being used for all the residuals. If four different thresholds are used as fault indicators for the four residuals, there remains the possibility of detecting the actuator faults. It is also possible that the fault detection isolability might improve. The delay associated with the residual deviations from the fault injections is found

to be 1[s], for all the deviations. The following is a criterion designed by the author as an indication of fault detection isolability.

5.2.1 Performance Criterion for Fault Detection

Firstly, the following must be defined:

1. Percentage of Correct Faults Detected, $P_{Correct}$: is the percentage obtained when faults are correctly detected, $N_{Correct}$ (for any particular method) in the presence of faults, divided by the time period in which there are faults, N_{Fault} (both values as numbers of samples).

$$P_{Correct}[\%] = \frac{N_{Correct}}{N_{Fault}} \cdot 100 \tag{96}$$

2. Percentage of Spurious Faults Detected, $P_{Spurious}$: is the percentage obtained when faults are incorrectly detected, $N_{Spurious}$ (for any particular method) when there are no faults, divided by the time period in which there are no faults, $N_{Fault\text{-}free}$ (both values as numbers of samples).

$$P_{Spurious}[\%] = \frac{N_{Spurious}}{N_{Fault-free}} \cdot 100 \tag{97}$$

3. Percentage of Correlation, $P_{Correlation}$: is the percentage obtained from subtracting the value of spurious faults, $N_{Spurious}$ detected from the value of correct faults detected, $N_{Correct}$, divided by the total number of fault samples, N_{Fault} of the simulation run.

$$P_{Correlation}[\%] = \frac{N_{Correct} - N_{Spurious}}{N_{Fault}} \cdot 100 \tag{98}$$

where $-\infty \leq P_{Correlation} \leq 100$

The percentage of correlation can be then be said to be a measure of isolability in the fault detection scheme. A negative correlation implies that more spurious faults are being detected than correct faults. A high value in the percentage of correlation indicates a good fault detection scheme. Ideally, a good fault detection scheme would have $P_{Correct}$ = 100%, $P_{Spurious}$ = 0% and $P_{Correlation}$ = 100%. However, this would never be the case even for the very best

fault detection schemes. This is because there would always be a delay in the fault detection, which would result in a lower value for the percentage of correct faults detected.

Furthermore, if the fault detection scheme is able to detect the whole duration of the fault, there would be fault indication when there is no fault present, because of the delay in the fault detection, thus resulting in detection of spurious faults. It is unfortunate that the delay of the fault cannot be easily incorporated into the above equations, but the criteria stated above are nevertheless important as a benchmark test for the performance of the fault detection methods. The following diagram shows a typical residual response to a fault. The delay, area of correct fault detection and area of spurious fault detection are highlighted.

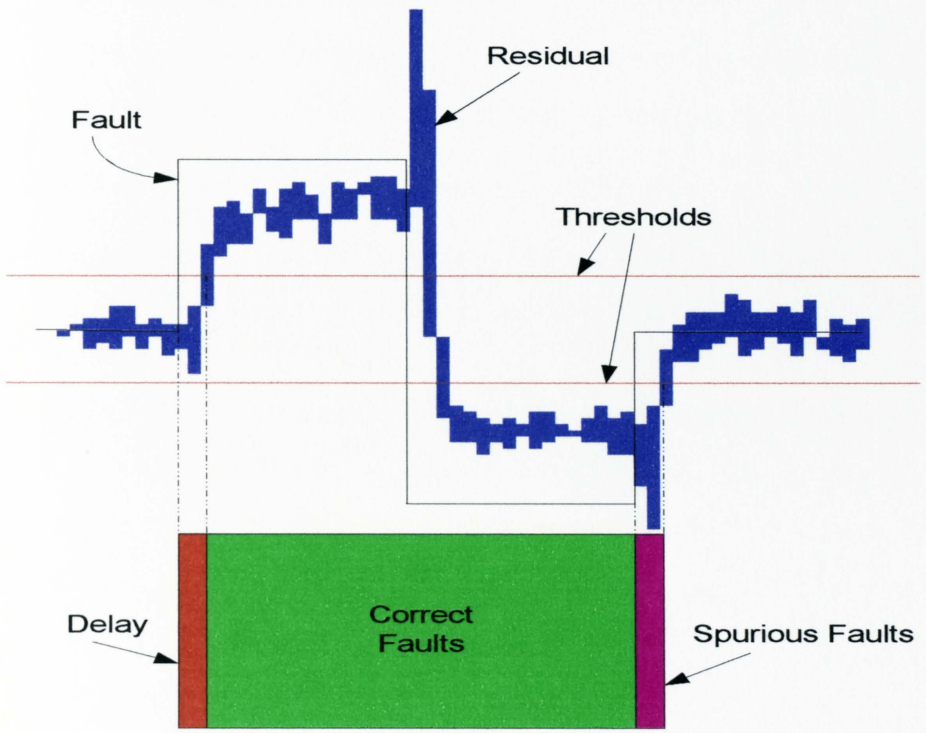


Figure 5-7 A Typical Residual Response to a Fault

From the above diagram, it can be seen that there will always be a further delay in the actual detection of the fault, which is caused by the threshold setting. If a threshold is set very high, the percentage of correct faults detected will be lower than a threshold set just greater than the

fault-free value of the residual. Therefore, the threshold setting strongly determines the performance of any fault detection scheme. The delay in the start of the residual deviation associated with a fault is dependent on the computational burden of the method. Generally, the more complicated the fault detection method (the greater the computational burden), the more accurate it is, and the slower the residuals deviate in the presence of faults. Hence, the fault detection is more delayed than for a simple method. Methods such as limit-checking detect faults quickly, but are inaccurate and not robust, whereas methods such as neuro-fuzzy networks detect faults slowly, but are very accurate and robust. Delays can also be associated with the sample time of the process. If a process has a quick sample time, the fault should be detected more quickly than if the process has a slower sampling time. Whether it is practical to have a quick sampling time is dependent on the process concerned.

It must also be noted that the percentages that are calculated (using Matlab) are largely subject to the noise injected into the process - which is random and is thus different for each simulation run. As a result, the percentages of error associated for each simulation run are:

Error on Percentage of Correct Fault Detection: ±2.00%

Error on Percentage of Spurious Fault Detection: ±2.00%

Error on Percentage of Correlation: ±2.00%

It must be stressed that these errors are applied strictly to the simulation parameters for the specific fault duration and length of simulation run, only. This is because the calculated percentages are dependent on the lengths. So, if the lengths of the faults or the total duration of the simulation were increased, then the associated errors for the percentages would need to be recalculated.

5.2.2 Use of Individual thresholds on each residual

With the use of a single threshold, the following results were obtained:

Percentage of Correct Fault Detection: 4.12%

Percentage of Spurious Fault Detection: 0.89%

Percentage of Correlation: 1.32%

These results show poor performance of the parity space method with the use of one threshold. As previously stated, the use of four thresholds might increase the fault detection performance. The following are the thresholds used for all four residuals:

Residual 1: 0.0400

Residual 2: 0.0055

Residual 3: 0.0350

Residual 4: 0.0055

The new fault indication graph follows.

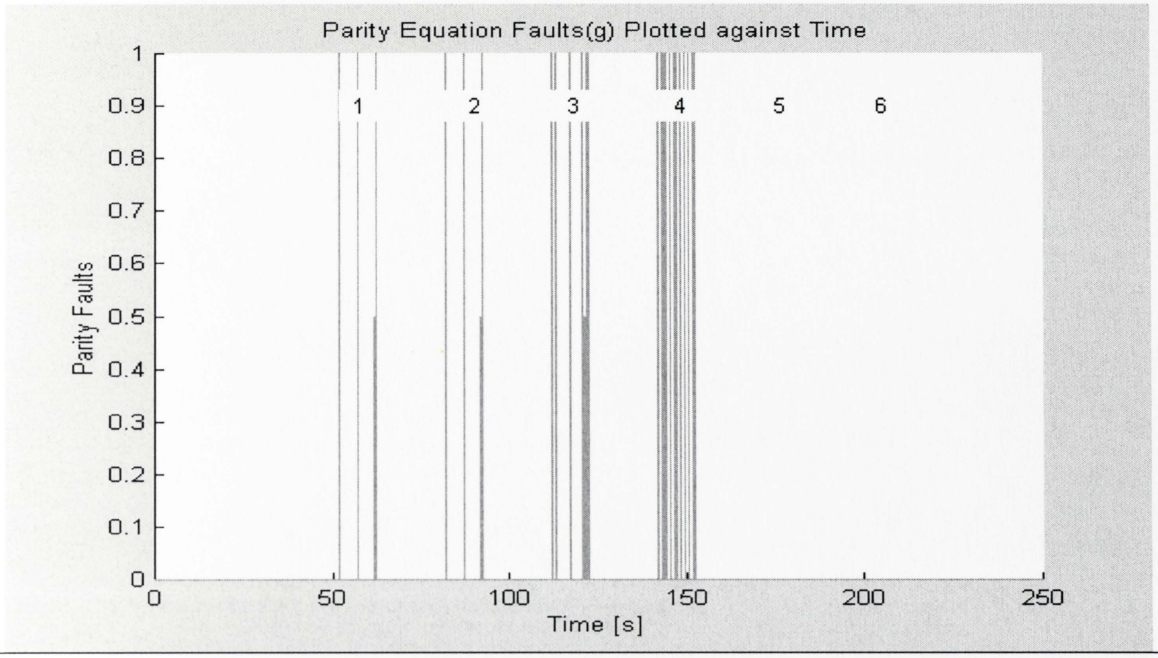


Figure 5-8 Parity Space Fault Indication with Four Thresholds

The above graph shows that there is an increase in the amount of faults detected, when compared to the fault indication graph in Figure 5-6. However, this could imply an increase in either the percentage of correct faults or spurious faults detected. Analysis of the fault detection performance shows an improvement from that described in Section 5.2.1, as there is an increase in the correct fault detection and reduction in the spurious fault detection, leading to an increase in percentage correlation, as given by:

Percentage of Correct Fault Detection: 4.62%

Percentage of Spurious Fault Detection: 0.48%

Percentage of Correlation: 3.13%

Once again, the actuator faults have not been detected, this would suggest an increase in the fault amplitude to see whether or not the method is able to detect the actuator faults. The following graphs show the residuals obtained by using actuator faults with a 5V amplitude.

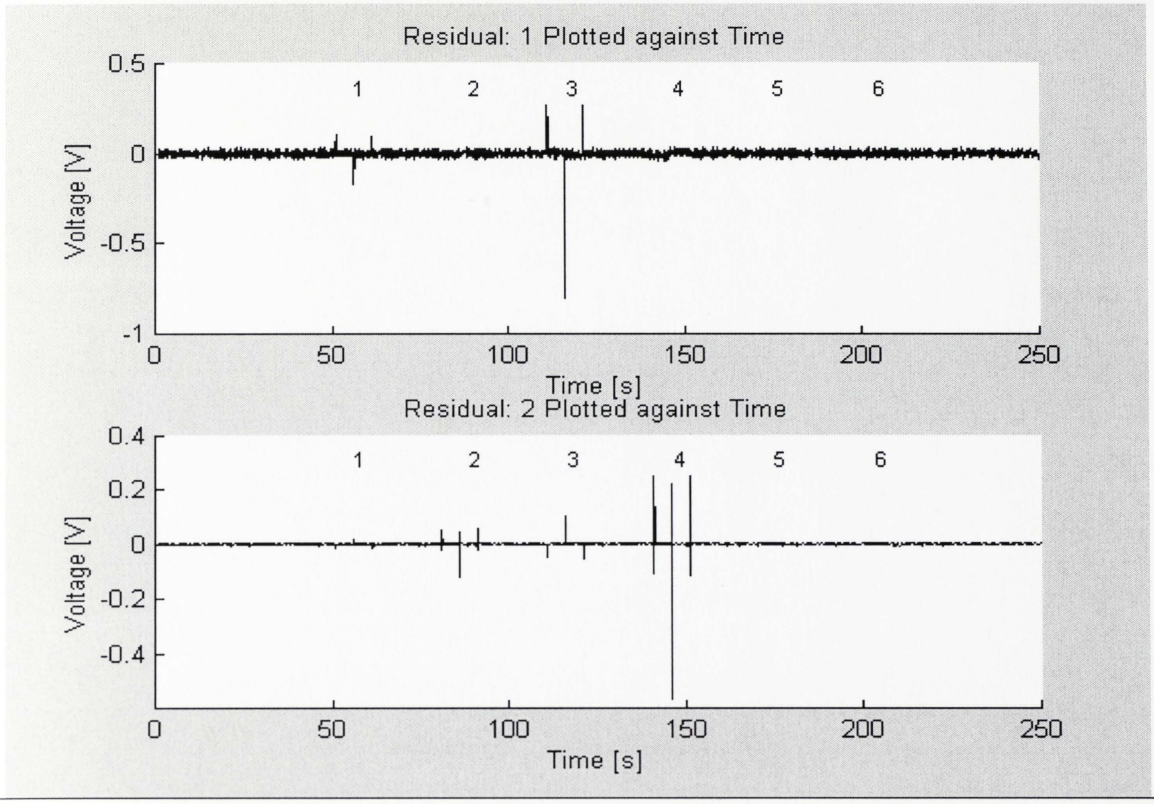


Figure 5-9 Parity Space Residuals 1 & 2 with actuator faults of 5V amplitude

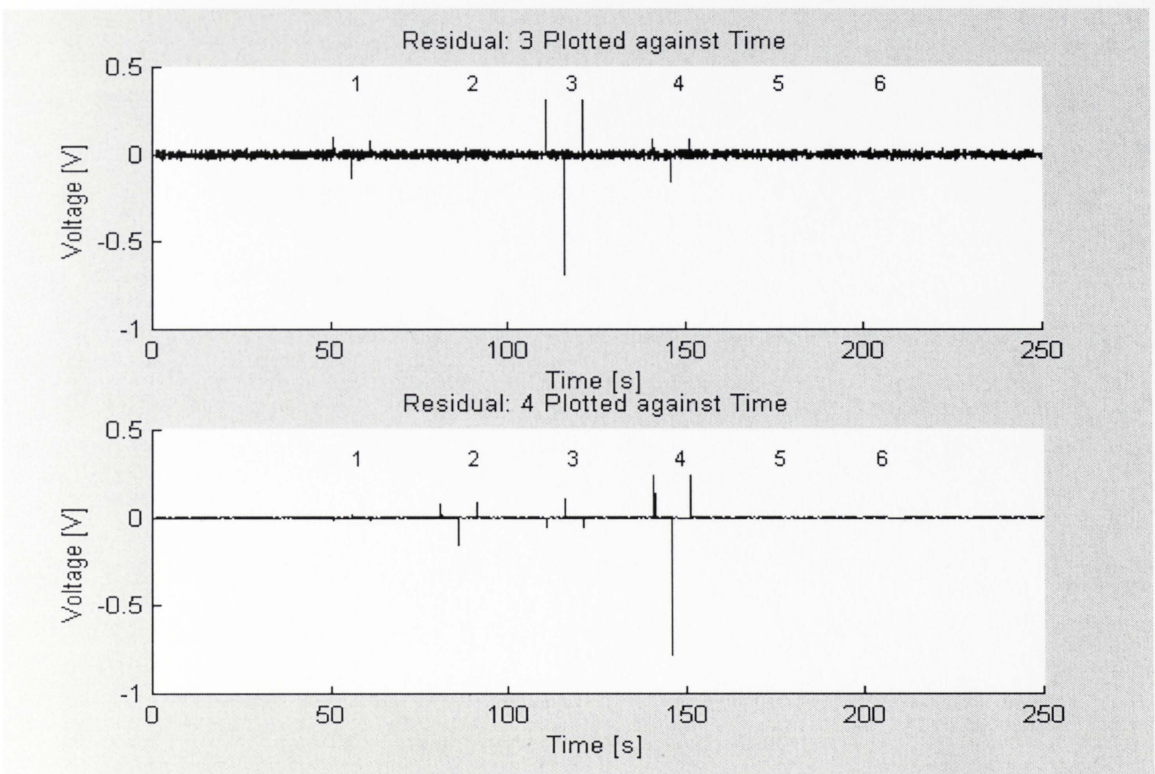


Figure 5-10 Parity Space Residuals 3 & 4 with actuator faults of 5V amplitude

Analysis of the previous graphs shows that Residual 4 recognises the actuator faults, suggesting that they should be seen clearly in the fault indication graph that follows.

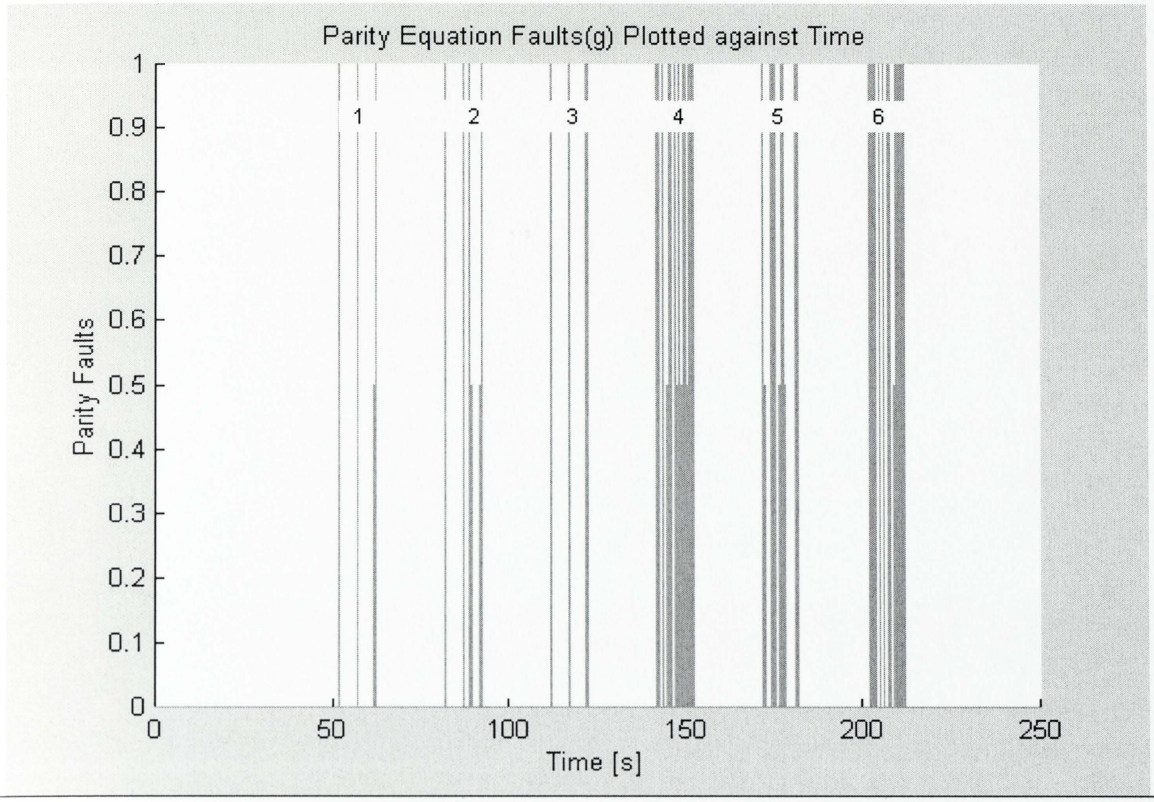


Figure 5-11 Parity Space Fault Indication with actuator faults of 5V amplitude

As expected, the graph shows that the actuator faults have been identified. The fault detection results show:

Percentage of Correct Fault Detection: 8.74%

Percentage of Spurious Fault Detection: 0.74%

Percentage of Correlation: 6.42%

These results show that the detection of the actuator faults increases the performance of the parity space method of fault detection. Further analysis suggests that only large actuator faults are detected by the parity space method. Since the method is able to detect actuator faults, the observations of the simulations may be related to theory in the following manner: From theory, it is known that the parity space method is heavily dependent on the observed input-output relationships of the process (Section 4.2.1). From analyses of the process dynamics (Section 5.1), it is known that they do not change significantly in the presence of actuator faults. So, if the process dynamics do not change significantly in the presence of a

fault, the method would not be able to detect the fault, as the residual would not show a significant enough deviation to exceed the fault threshold.

5.2.3 Kalman filter implementation

From the theoretical explanations given in Sections 2.2.3 & 4.2.1, it is also known that the parity space method is very sensitive to noise. It is possible to filter the process noise by means of a Kalman filter (Section 4.2.1.1). There is also the possibility that the high levels of noise on Residuals 1 & 3 are greater than the deviations that cause the actuator faults. Therefore a reduction of the residual noise could show deviations on the residuals corresponding to actuator faults. Irrespective of this possibility, the performance of the fault detection method should increase. The following are the residuals obtained when using a Kalman filter (refer to Appendix E for further information).

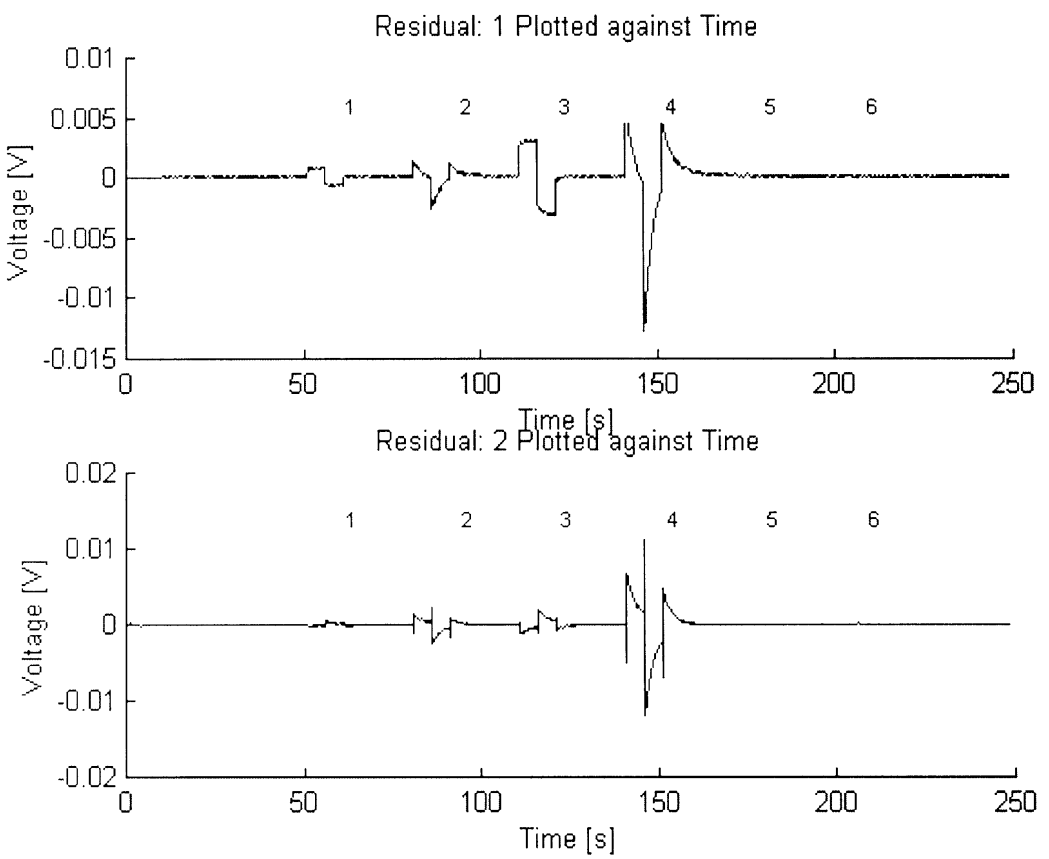


Figure 5-12 Parity Equation Residuals 1 & 2 with Kalman filter

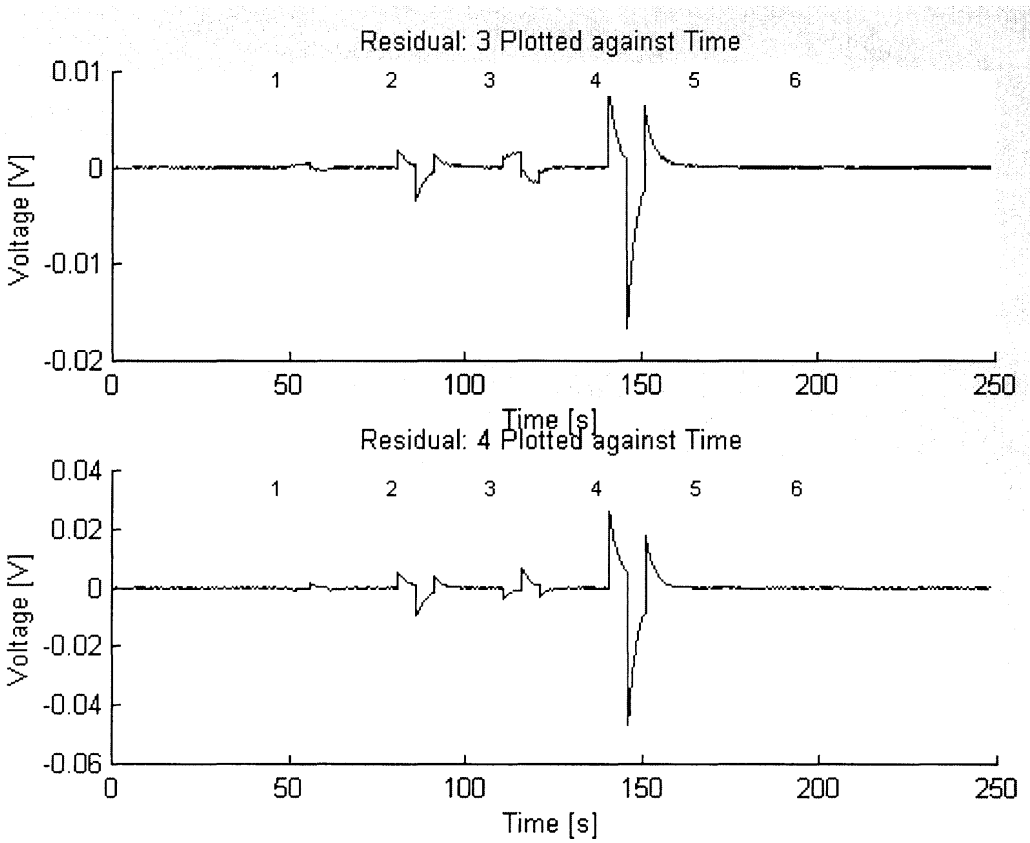


Figure 5-13 Parity Equation Residuals 3 & 4 with Kalman filter

From the new residuals, a significant improvement is seen in the shape of the residuals, as compared to Figures 5-9 and 5-10. The residuals do not show "spikes" when faults are detected, but tend to resemble the fault waveform more closely. This occurs because the Kalman filter introduces a delay in the decay of the residual deviations. The result is better detection of correct faults, as the residual is above the threshold for the duration of the fault. However, the result of increased correct fault detection, is an increase in the detection of spurious faults, which is caused by the delay in the decay of the residual deviations when faults occur. The delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations. There is also a reduction in the noise levels, as expected from theory. The choice of the Q matrix for the Kalman filter is difficult as it influences the shape and noise rejection properties of the residuals. Large values of Q tend to increase the noise levels in the residuals as the sampling rate of the filter is increased, therefore when a fault occurs the deviation in the residual quickly settles down. However, for small values of Q, the sampling time is much slower resulting in good noise reduction, but any deviation in the residual takes long to settle. So essentially the best value of Q is chosen from a trade-off in being able to detect small faults (such as the actuator faults) which could

easily be masked by the residual noise, or residuals with a good shape, thus giving better fault detection performance. For more information refer to Appendix E.

Another observation that can be made is that slight deviations at the positions where the actuator faults have been simulated, are seen in Residuals 3 & 4. These deviations could now be indicated as faults. Firstly though, the new thresholds have to be set by analysing the residuals obtained.

The following are the new threshold values for the residuals:

Residual 1: 0.0004

Residual 2: 0.0004

Residual 3: 0.0002

Residual 4: 0.0005

The following diagram shows the fault indication graph with a Kalman filter.

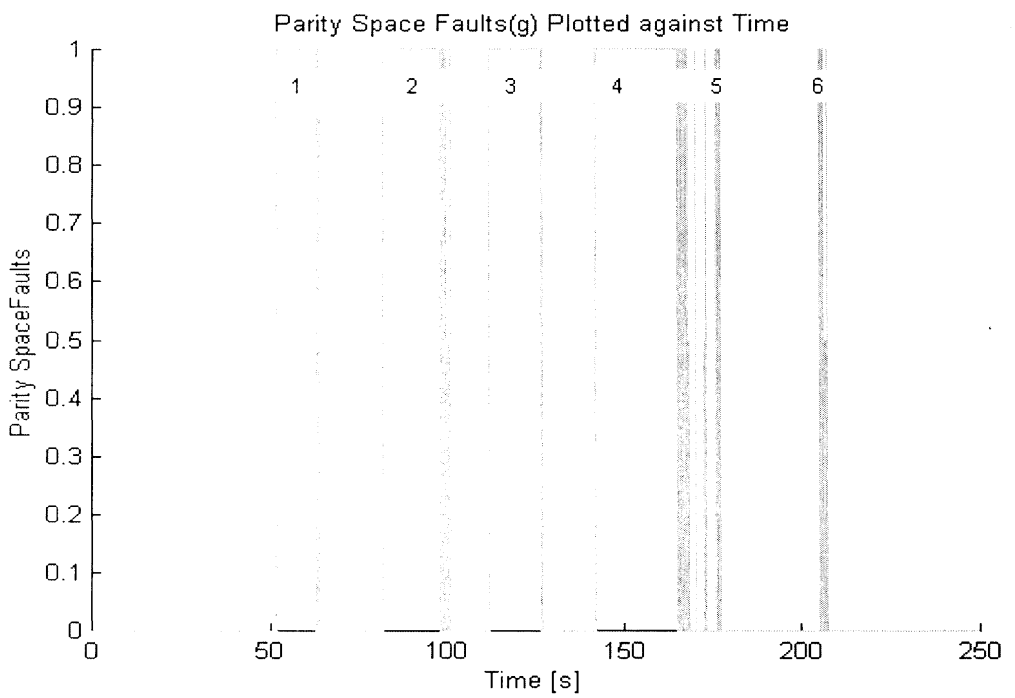


Figure 5-14 Parity Space Fault Indication with Kalman filter

By comparison with Figure 5-6 and 5-8, it is seen that the parity space method with a Kalman filter is able to detect the actuator faults. However, because of the delay of the residual deviations associated with the Kalman filter, the lengths of the sensor faults are more than the

actual fault lengths. This is a result of the low threshold used to detect the actuator faults. So there is a trade off between the detection of the actuator faults and the more accurate detection of the sensor faults. If the actuator faults had a large enough amplitude (e.g. 5V), then a higher threshold could have been implemented, with the possibility of achieving a better fault detection performance. The performance of the current architecture of the parity space method with Kalman filter is:

Percentage of Correct Fault Detection: 69.63%

Percentage of Spurious Fault Detection: 17.52%

Percentage of Correlation: 14.81%

There is a significant increase in the percentage of correct faults detected, but there is also an increase in the percentage of spurious faults. This is expected from the statements above. Since the percentage of spurious faults is quite high, the percentage of correlation is thus low, even though the percentage of correct faults detected is quite high.

5.2.4 Summary for the Parity Space method

As expected from the literature review (Table 2-1, p48), the parity space method is seen to be sensitive to noise. The deviations in the residuals caused by faults, do not last over the whole fault duration, and can be considered as a shortfall in the fault detection method. The introduction of a Kalman filter into the process was able to reduce the noise sensitivity of the fault detection method. The Kalman filter also provided an increased performance of the method and enabled the detection of actuator faults (0.5V amplitudes), which were previously left undetected. The detection of actuator faults is a problem for the method, since they do not affect the inputs or outputs of the system significantly (as the process behaves as a low-pass filter). The use of the Kalman filter in the parity space method of fault detection, is strictly defined as an adaptation of the method. As a result, the analysis of the method's performance in Section 5.5.1, uses the parity space method using only individual thresholds and no Kalman filter.

5.3 Adaptive Parity Space Results

Firstly, the residuals for the Adaptive Parity Space (APS) method are defined as the parameters contained in the matrix, *p(k)*. This method is unlike the previous and following methods of fault detection, which use four and two residuals to detect six faults. Since there are six faults simulated for the process there will therefore be six parameters and residuals. Ideally, each residual should detect the single fault associated with its parameter. So the residual obtained for an actuator scale fault parameter should only detect the actuator scale fault. However, this does not occur because any fault will effect the measured input and output signals, causing deviations in the residuals (as explained and observed with the previous fault detection methods). This section is divided into subsections, which describe the results obtained for the adaptive parity space method of fault detection by using:

- Least Squares (LS)
- Recursive Least Squares (RLS)
- Recursive Least Squares with forgetting factor
- Recursive Least Squares with forgetting factor and stabilising factor

for the estimation of the parameter vector *p(k)* (Chapter 4). Also included are sections that investigate the effect of the forgetting factor and stabilising factor on the adaptive parity space method of fault detection, and a summary section. For the remainder of this dissertation, the Residuals m_1, v_1, m_2, v_2, s and d will be denoted as Residuals M1, V1, M2, V2, S and D, respectively.

5.3.1 Least squares

As with the previous methods, the fault free residuals need to be examined in order to determine the threshold. The following diagrams show the fault-free residuals.

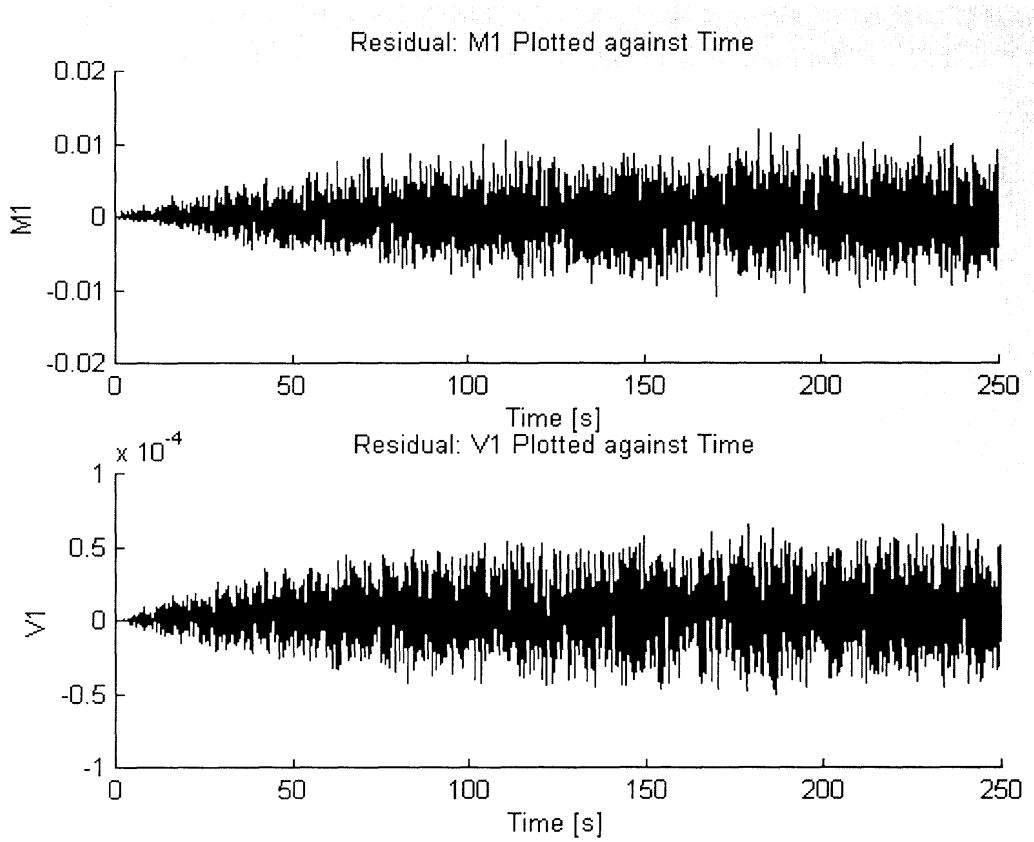


Figure 5-15 Adaptive Parity Space Residuals M1 & V1 using LS estimation in a fault-free environment

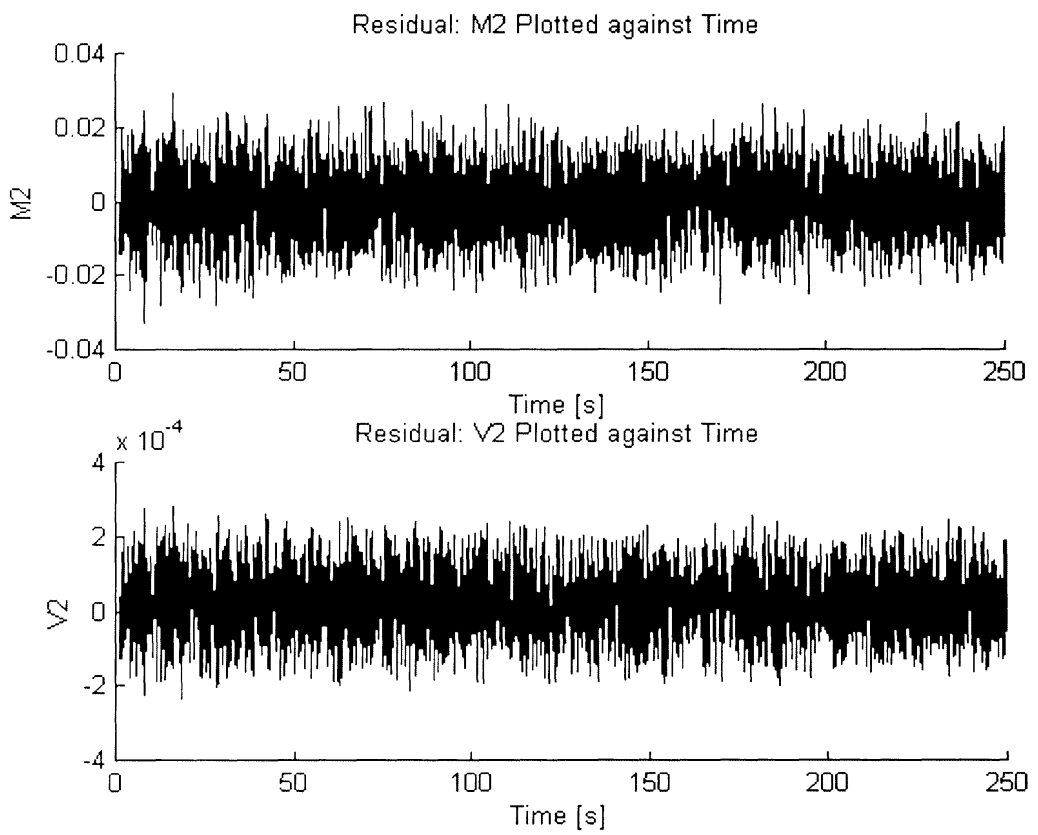


Figure 5-16 Adaptive Parity Space Residuals M2 & V2 using LS estimation in a fault-free environment

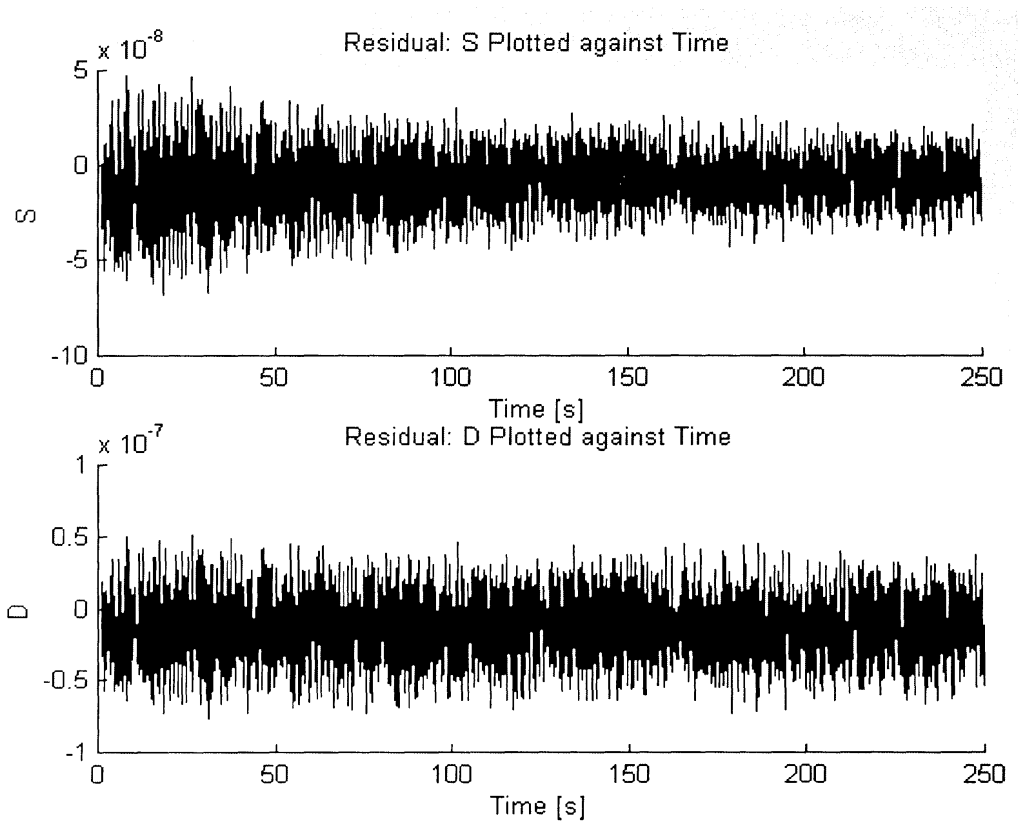


Figure 5-17 Adaptive Parity Space Residuals S & D using LS estimation in a fault-free environment

From the analysis of the residuals, it is seen that there is a wide range in their fault-free values, approximately between 5.0×10^{-8} and 1.1×10^{-2}. There are also transients associated with the residuals, especially noticeable in Residuals M1, V1 and S. These transients are due to the LS estimation error in finding the residuals, and can be said to have a negligible effect on the fault detection scheme (since they do not affect the residuals significantly). It is logical to implement six thresholds, and not investigate the results of the fault detection method with the use of a single threshold. The resultant thresholds obtained for the six residuals are:

Residual M1: 1.2×10^{-2}

Residual V1: 8.0×10^{-5}

Residual M2: 3.5×10^{-2}

Residual V2: 3.0×10^{-3}

Residual S: 1.0×10^{-7}

Residual D: 8.0×10^{-8}

The following graphs show the response of the residuals subject to the six faults, followed by the fault indication graph.

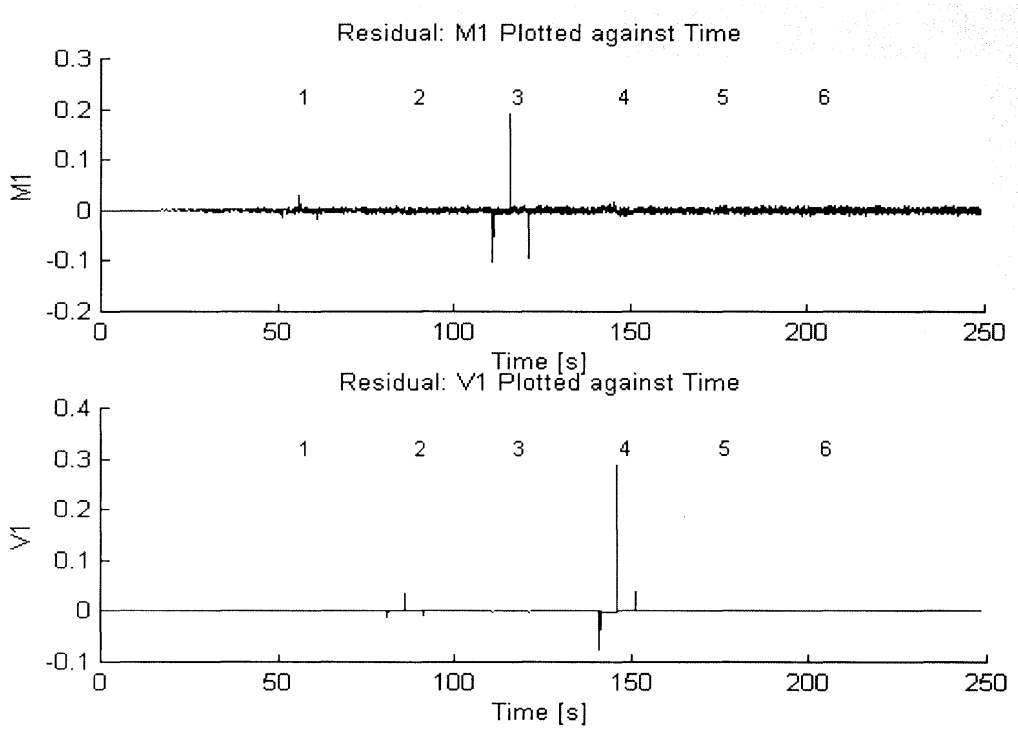


Figure 5-18 Adaptive Parity Space Residuals M1 & V1 using LS estimation subject to faults

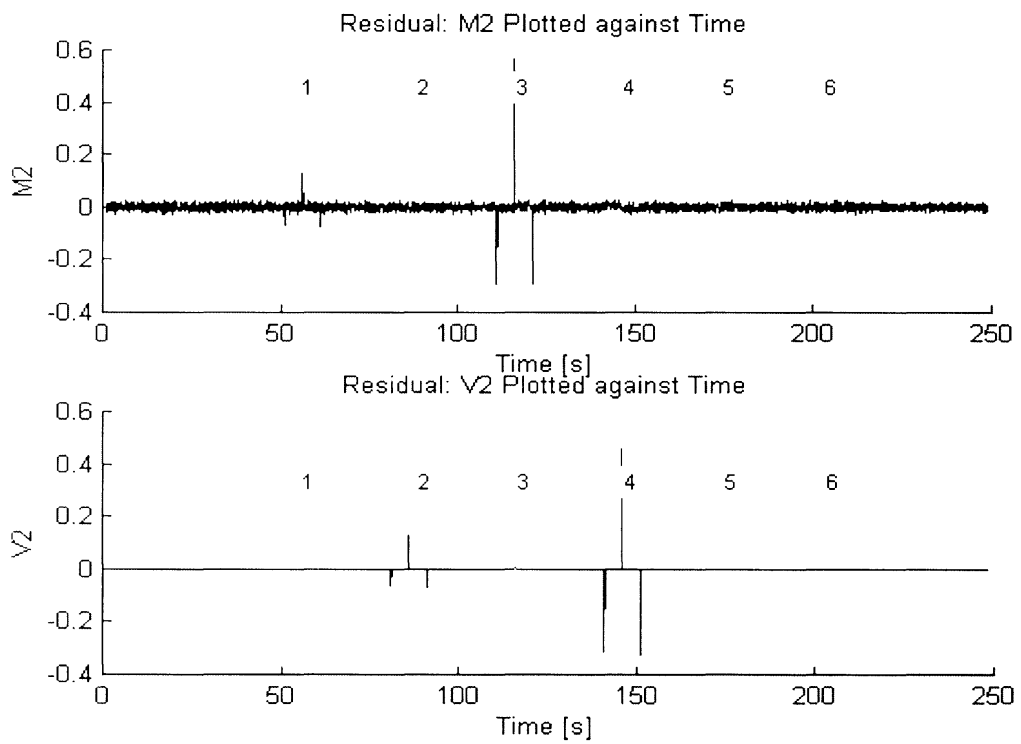


Figure 5-19 Adaptive Parity Space Residuals M2 & V2 using LS estimation subject to faults

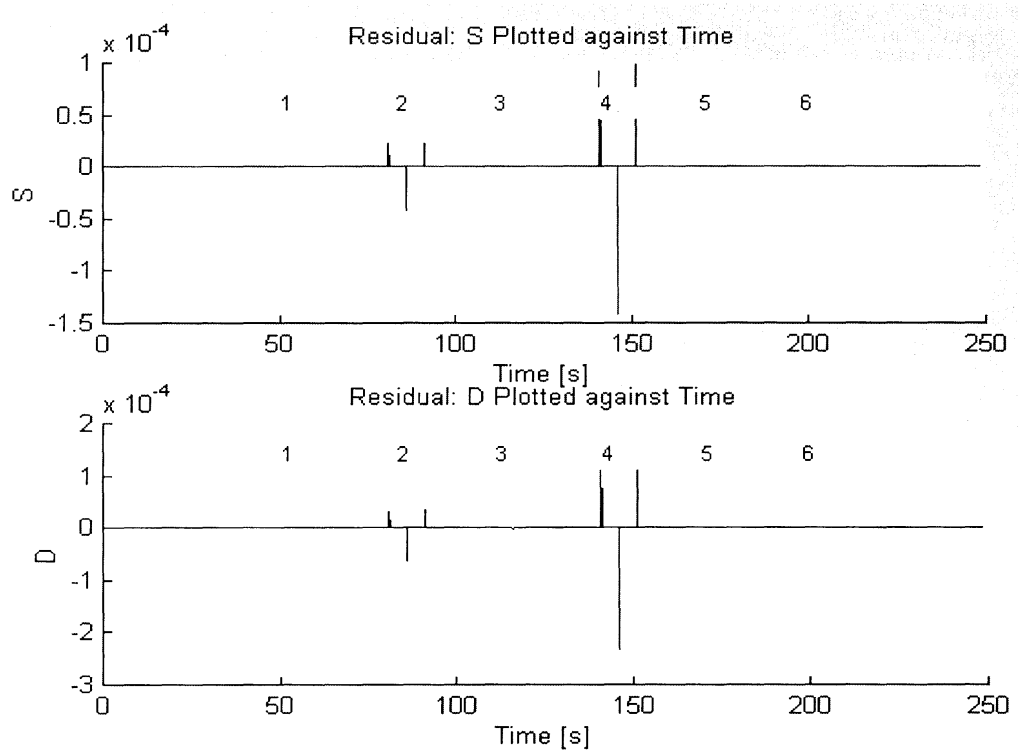


Figure 5-20 Adaptive Space Residuals S & D using LS estimation subject to faults

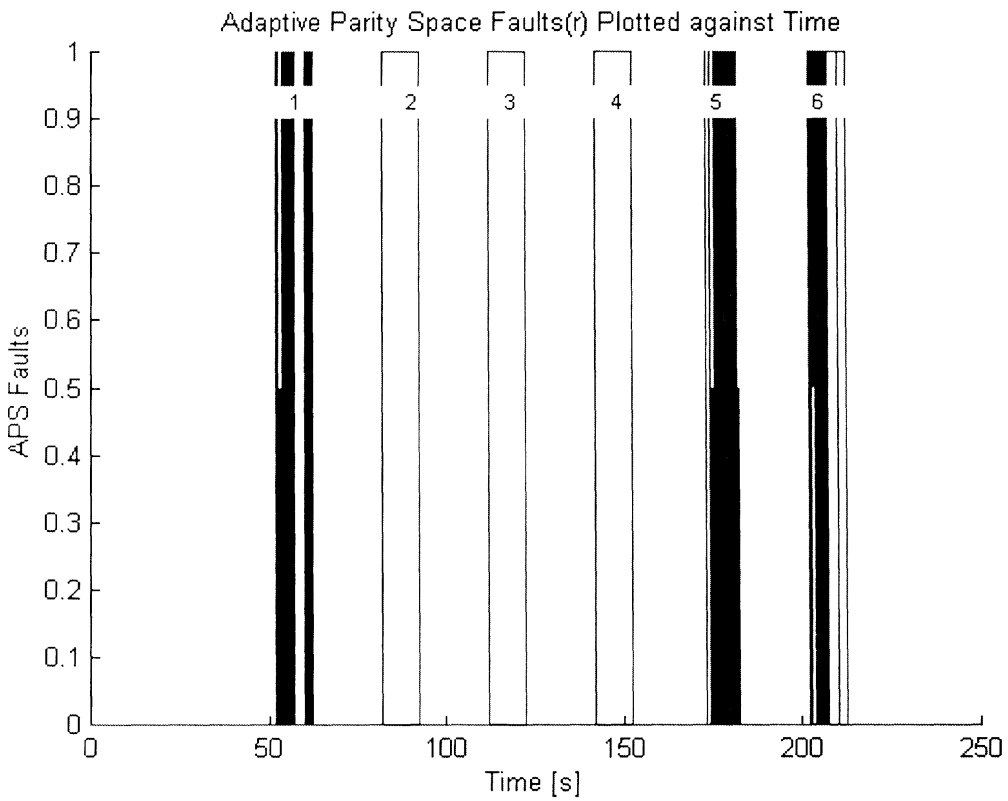


Figure 5-21 Adaptive Parity Space Fault Indication using LS estimation

From the residual graphs, Residuals M1, V1, M2, V2 and D detect Sensor 1 faults fairly accurately, while Residuals M1, V1, V2, S & D detect Sensor 2 faults fairly accurately. The delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations. The actuator faults do not seem to be seen on any of the residuals because they affect the residuals very lightly, as expected from theory (Section 5.1). However, the fault indication graph does show detection of the actuator faults. On closer analysis of the residuals, it was found that Residuals M2, V2 and D show deviations at the positions of the actuator faults. The performance of the method is described by:

Percentage of Correct Fault Detection: 66.00%

Percentage of Spurious Fault Detection: 3.07%

Percentage of Correlation: 56.50%

The method is seen to be a fairly good fault detection method, from the results presented above. But, LS estimations have problems with estimation bias (Appendix D), which could be problematic in practice. The residuals would have to track the faults more accurately to improve on the percentage of faults detected and hence, the percentage of correlation.

5.3.2 Recursive Least Squares

Implementation of a recursive least squares estimation should yield better estimation of the residuals. The following graphs show the fault-free residuals obtained using RLS estimation.

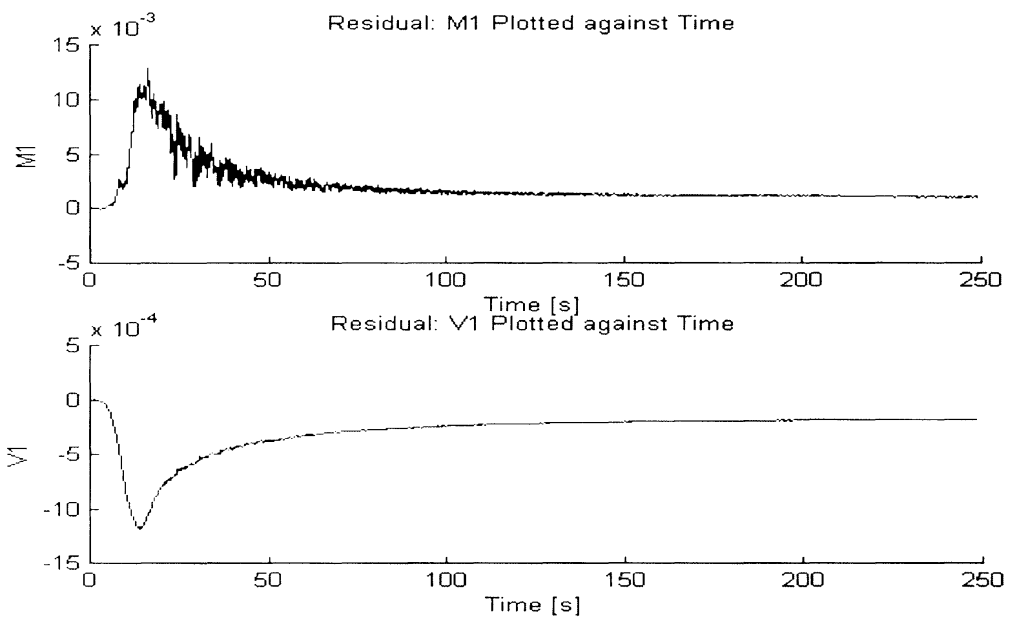


Figure 5-22 APS Residuals M1 & V1 using RLS in a fault-free environment

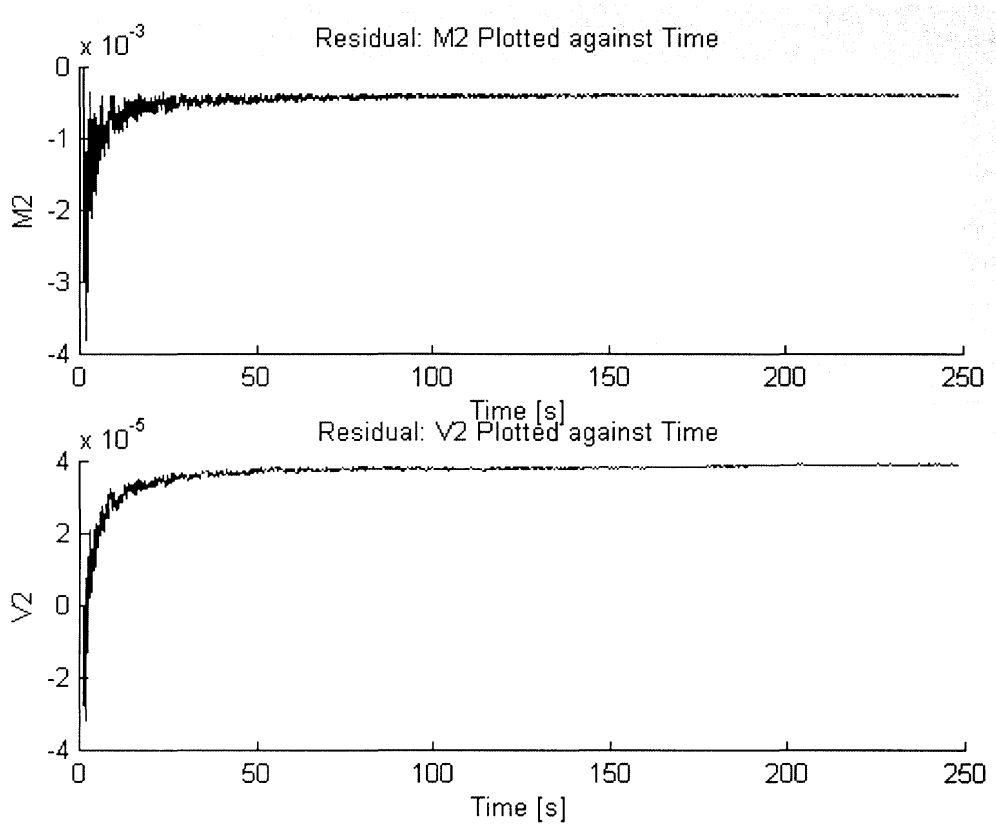


Figure 5-23 Adaptive Parity Space Residuals M2 & V2 using RLS in a fault-free environment

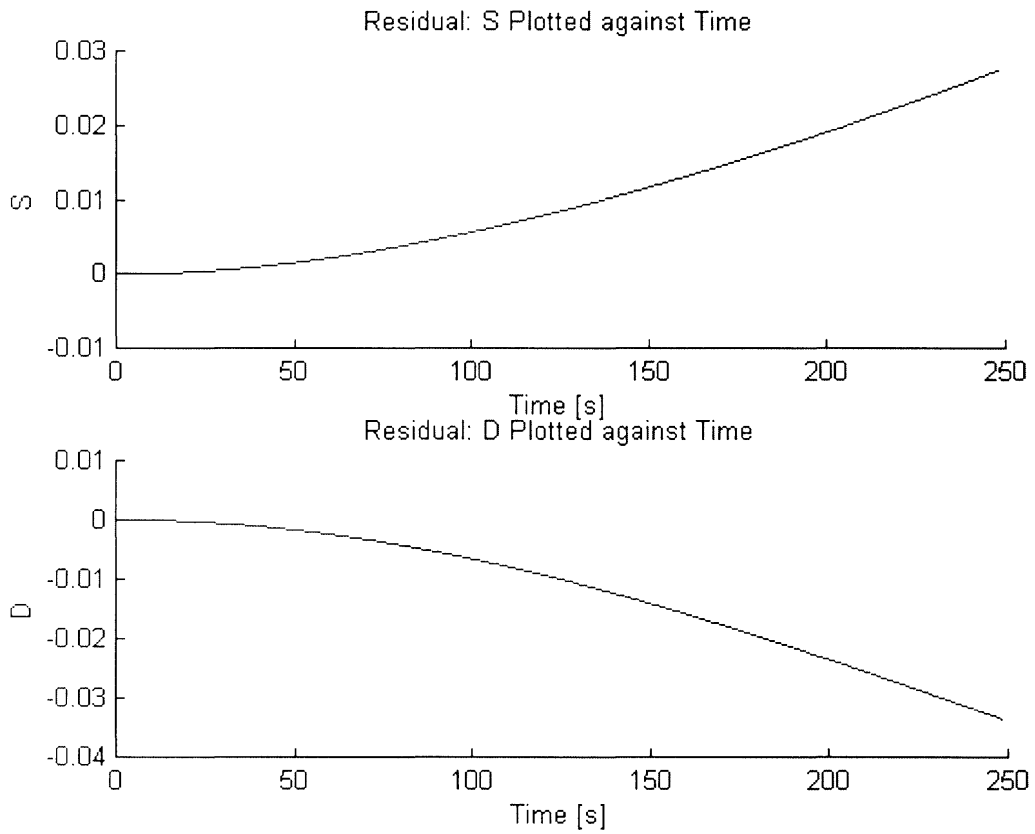


Figure 5-24 Adaptive Parity Space Residuals S & D using RLS in a fault-free environment

From the graphs of Residuals M1, V1, M2 and V2, it is seen that the initial instability in the RLS estimation causes overshoot and high frequency oscillation, when the process is stepped. Thereafter, the residuals settle down to values, which are nonzero as a result of RLS estimation bias. Residuals S & D show overshoot throughout the transient state of the process. The instability of the RLS estimation results from abrupt changes in the parameters/residuals, as the process is stepped. Furthermore, the estimation uses all the measurements, when theory suggests the most recent measurements should be used. However, it is possible that this adaptation could yield good fault detection results. The delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations. Thresholds were not used for Residuals S & D, so the thresholds set for the remaining residuals were:

Residual M1: 1.4×10^{-2}

Residual V1: 1.3×10^{-3}

Residual M2: 4.0×10^{-3}

Residual V2: 4.0×10^{-5}

The following graphs show the residuals obtained when faults were introduced into the system. The resultant fault indication is shown after the residual graphs.

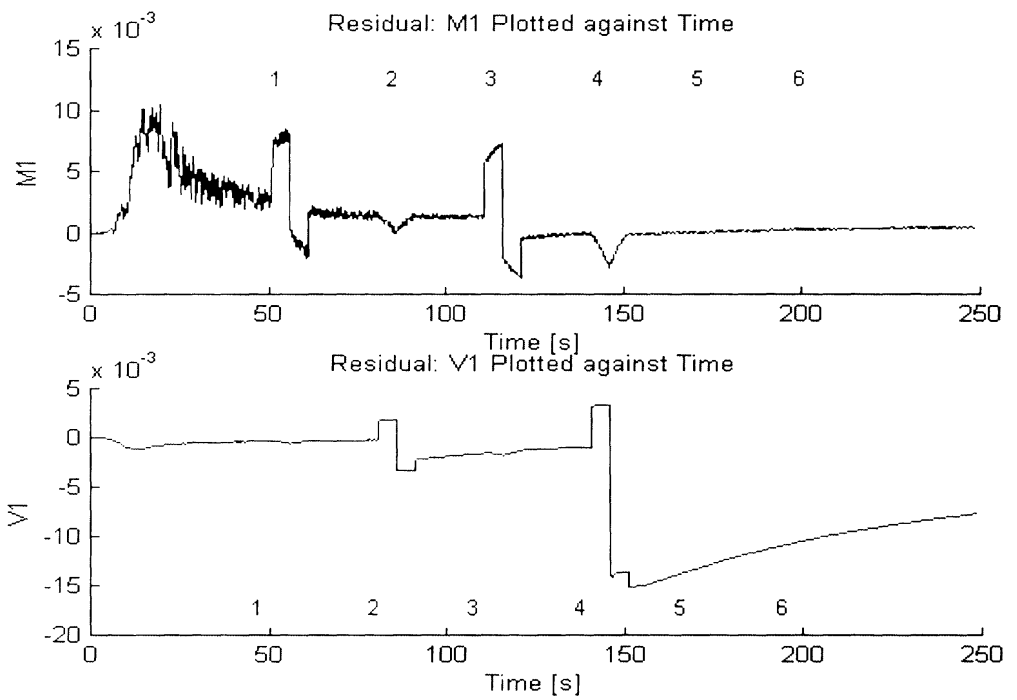


Figure 5-25 Adaptive Parity Space Residuals M1 & V1 using RLS with faults

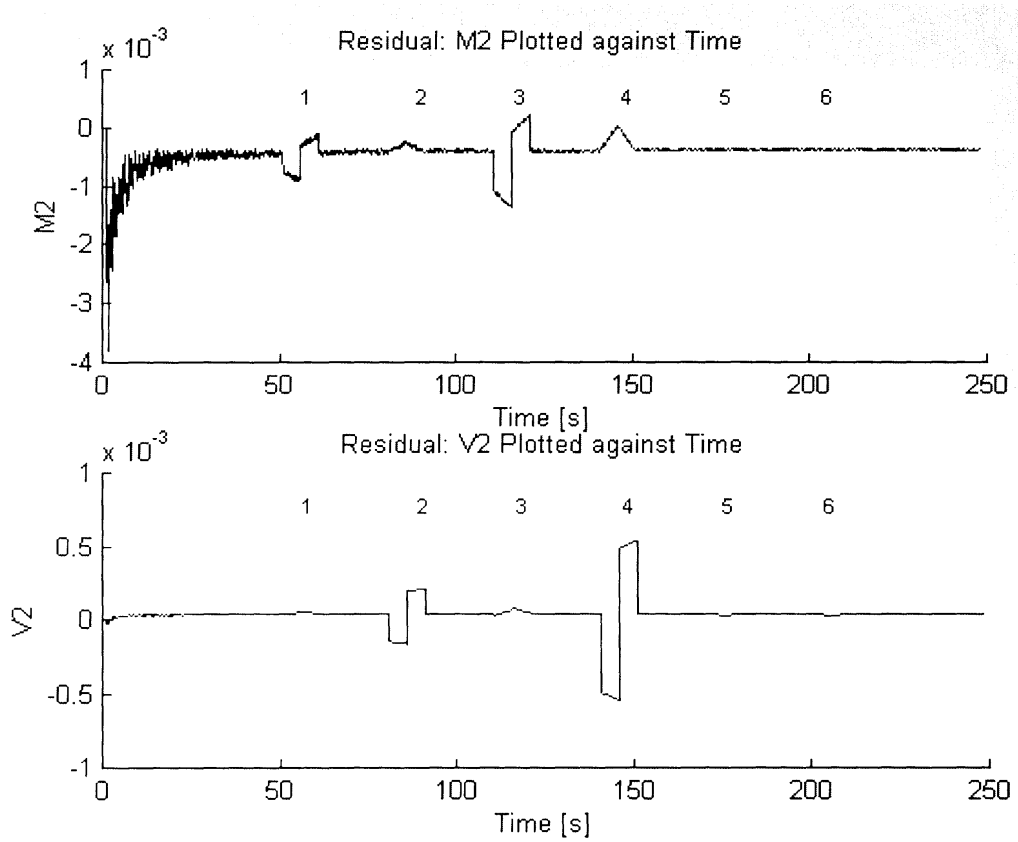


Figure 5-26 Adaptive Parity Space Residuals M2 & V2 using RLS with faults

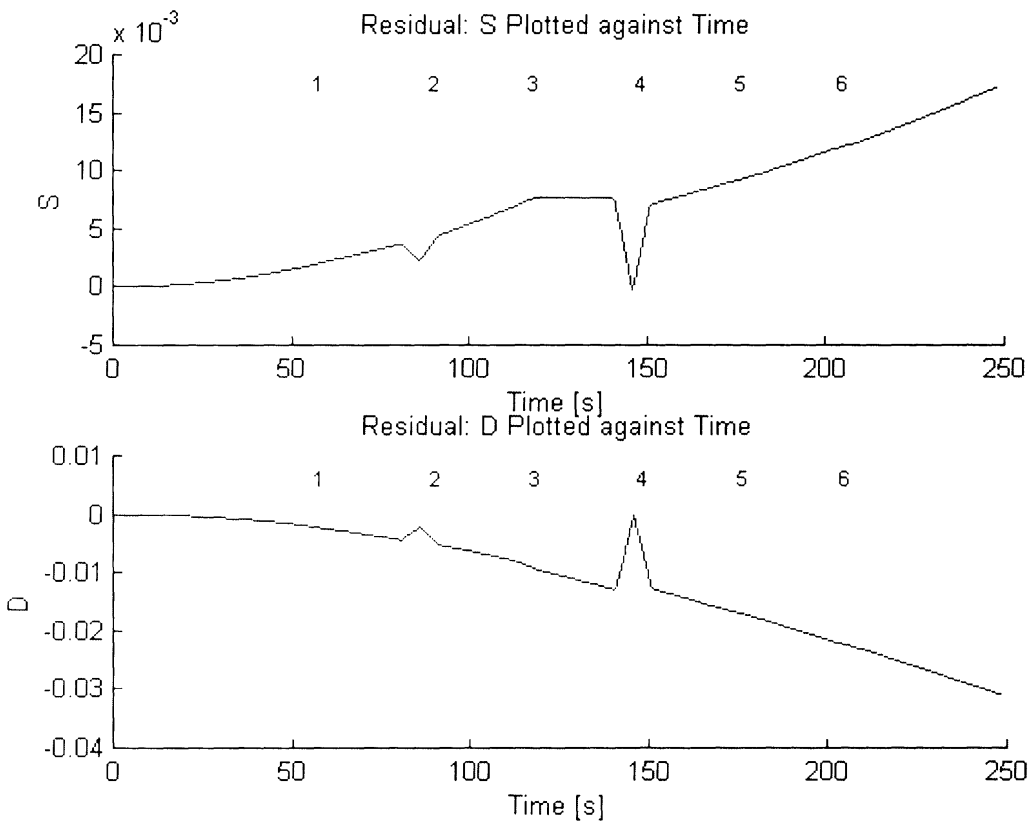


Figure 5-27 Adaptive Parity Space Residuals S & D using RLS with faults

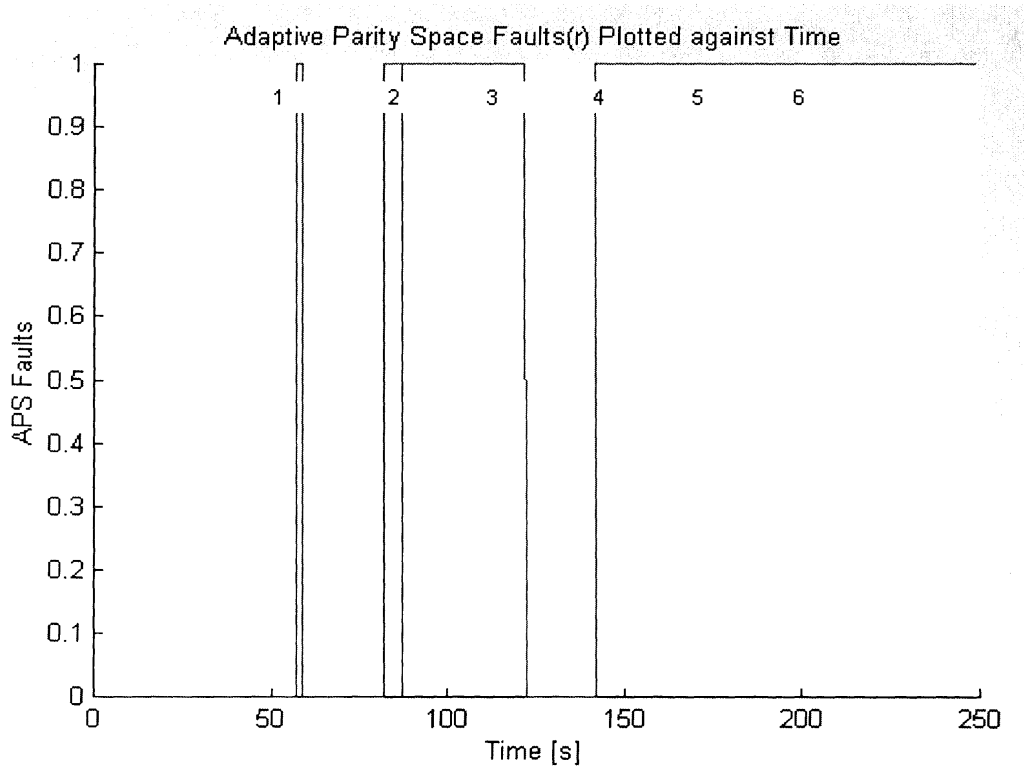


Figure 5-28 Adaptive Parity Space Fault Indication using RLS

From the residual graphs, Residuals M1 & M2 show deviations for both Sensor 1 & 2 faults, but because of the RLS estimation instability (caused by the change of the parameters when the process is stepped), the faults cannot be detected. Residual V1 is able to detect the Sensor 2 faults, but the delay in the decay of the deviations (caused by the RLS estimation), causes spurious faults to be detected, which can be seen in the fault indication graph. Residual V2 shows deviations corresponding to all the faults, but the actuator faults are not detected due to the threshold set for the fault detection. Residuals S & D show deviations corresponding to Sensor 2 faults, Sensor 1 bias fault and the actuator faults. Refer to Appendix K to view the residual variations over a duration of 1000[s]. From the above results, it can be seen that the RLS estimation does not give a good fault detection performance, as expected from the theory given. For completeness, the performance can be described by:

Percentage of Correct Fault Detection: 9.09%

Percentage of Spurious Fault Detection: 53.66%

Percentage of Correlation: -76.01%

The negative value of the percentage of correlation indicates that there were more spurious faults detected than correct faults.

5.3.3 Recursive Least Squares with Forgetting Factor

A RLS estimation with forgetting factor, λ was used to estimate the residuals to observe whether there would be an increase in the fault detection performance with the use of the most recent measurements generated from the process. The chosen value for λ was 0.664 (refer to Appendix F for reasons). The fault-free residuals were analysed in order to determine the thresholds to be used. The following graphs show the fault-free residuals obtained.

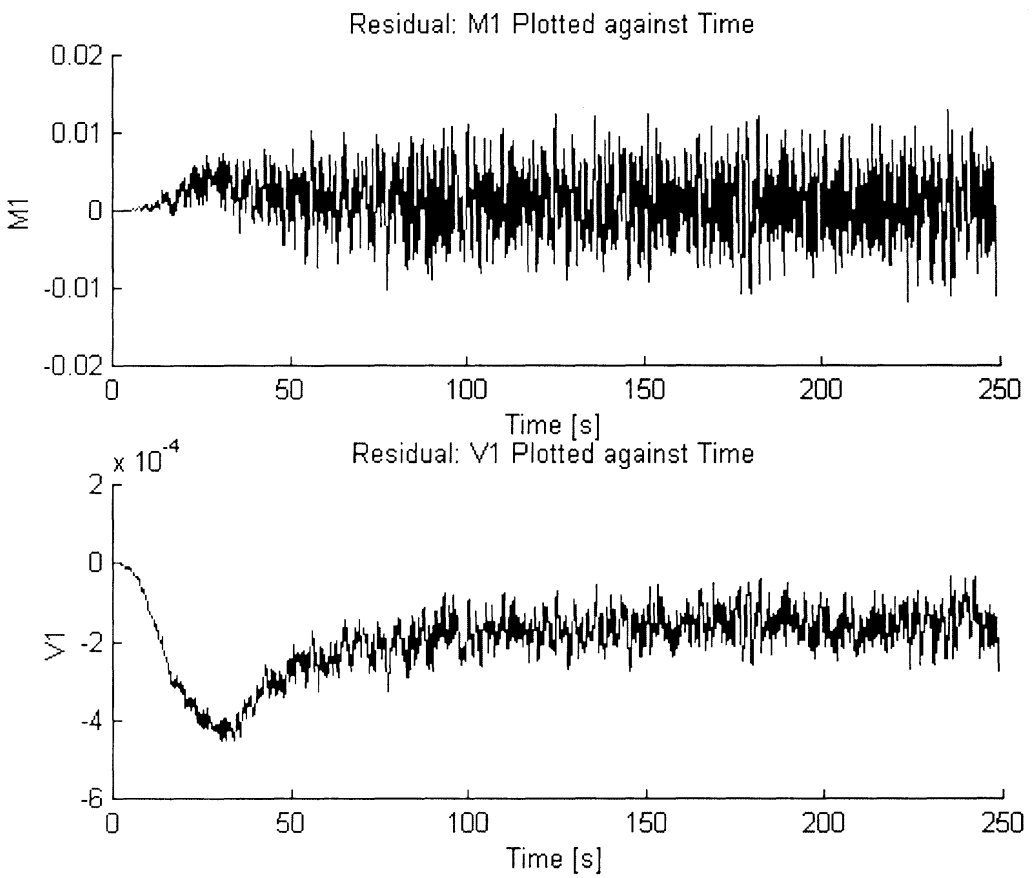


Figure 5-29 Adaptive Parity Space Residuals M1 & V1 using LS estimation subject to faults

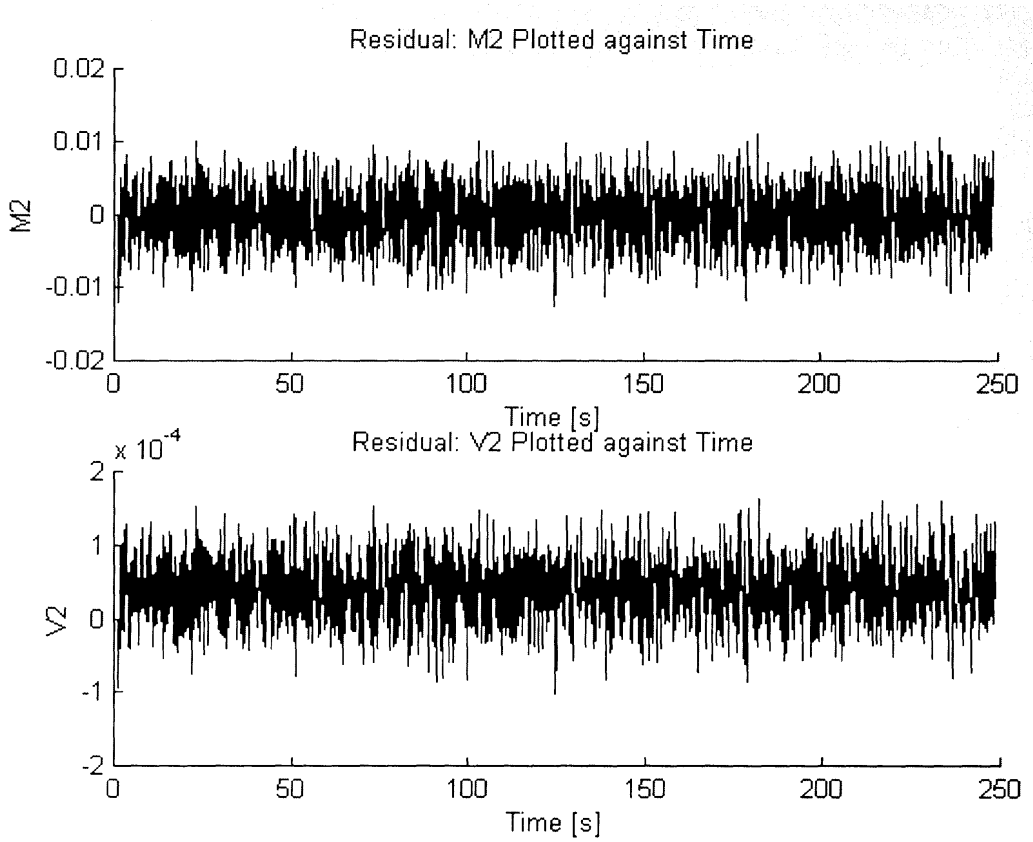


Figure 5-30 Adaptive Parity Space Residuals M2 & V2 using LS estimation in a fault-free environment

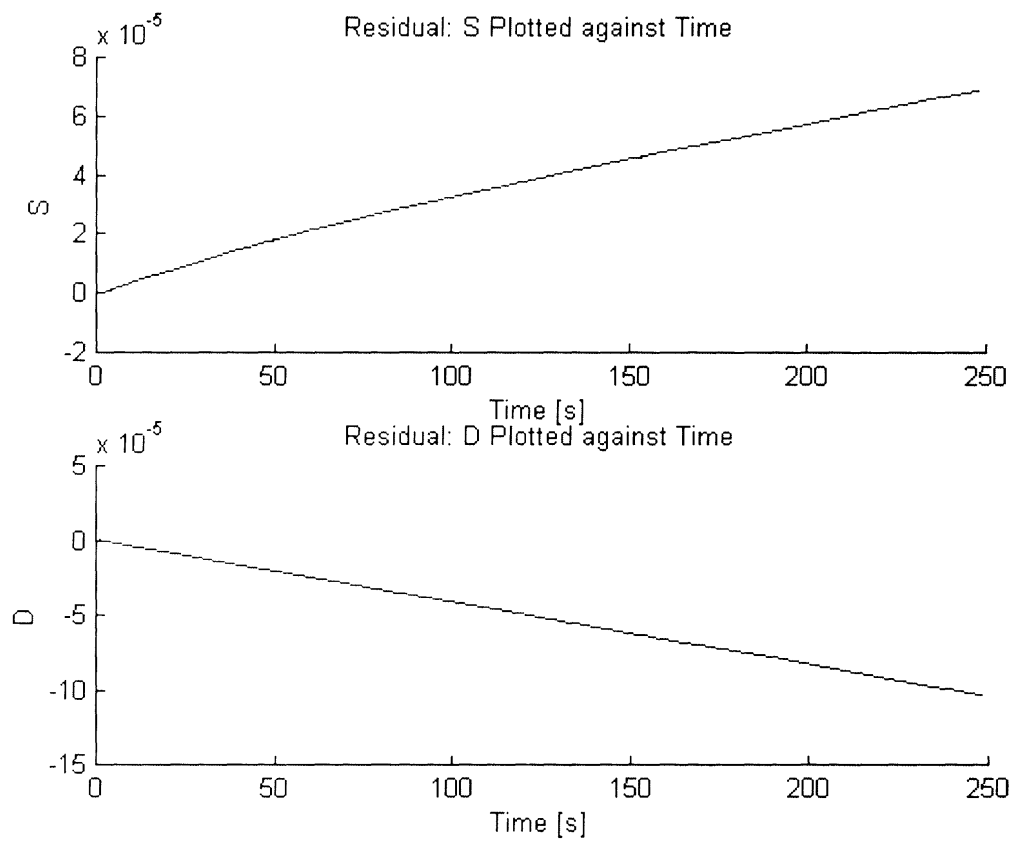


Figure 5-31 Adaptive Space Residuals S & D using LS estimation subject to faults

From the residual graphs, it can be seen that Residuals M1, V1, M2 and V2 have become noisier as compared to the respective residuals of the RLS estimation (Figures 5-22, 5-23 and 5-24). The introduction of λ is the cause of the extra noise, the greater the rate at which old data is forgotten, the noisier the estimation becomes. Furthermore, Residual V1 still shows overshoot when the process is stepped, and Residuals S & D have overshoot throughout the transient state. Also, an estimation bias in Residuals V1, V2, S and D is observed (Appendix K shows the bias in Residuals S and D). However, it can be seen that all the residuals have become more stable with the introduction of λ. But, a stabilising factor in the RLS estimation is seen to be necessary. The following thresholds were then set for the residuals:

Residual M1: 1.5×10^{-2}

Residual V1: 5.0×10^{-4}

Residual M2: 1.5×10^{-2}

Residual V2: 2.0×10^{-4}

Residual S: 1.0×10^{-4}

Residual D: 1.5×10^{-4}

The following graphs show the residuals subject to faults.

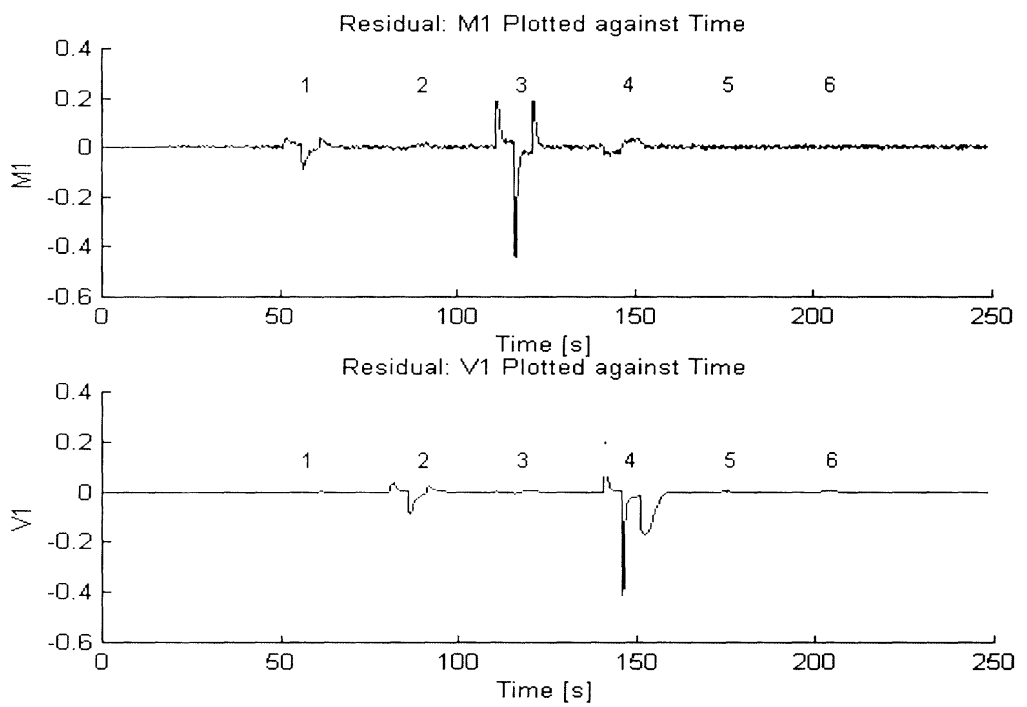


Figure 5-32 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with λ, subject to faults

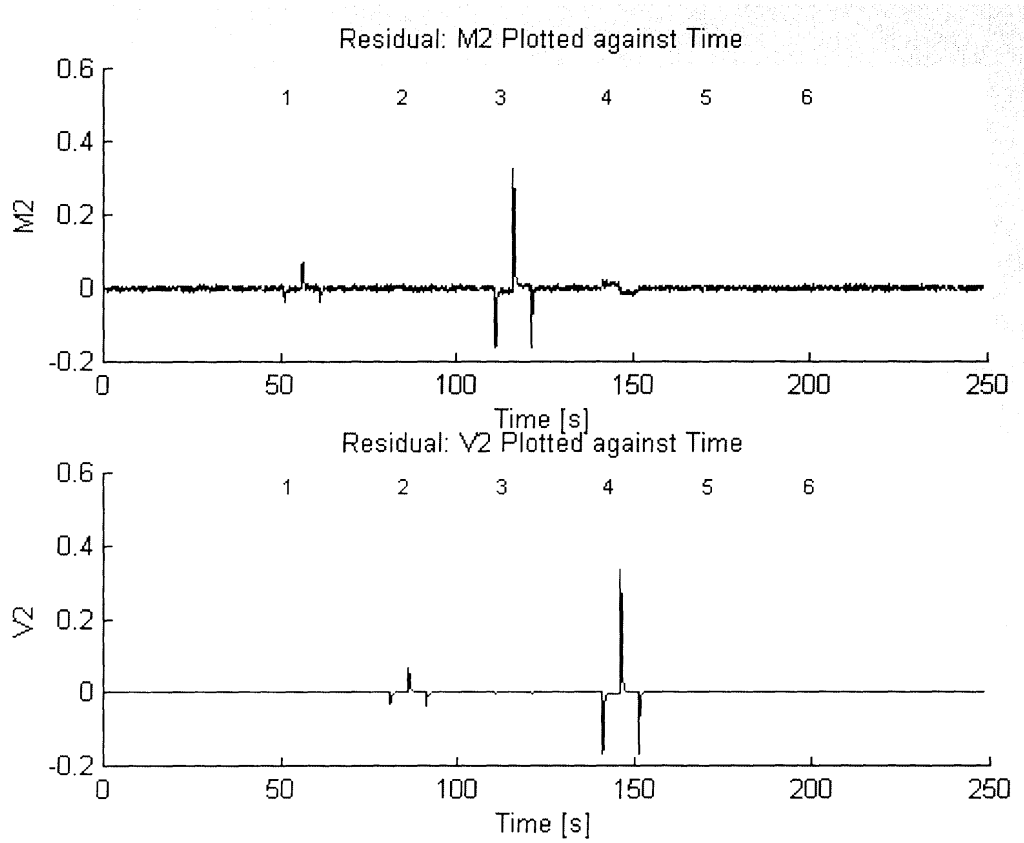


Figure 5-33 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with λ, subject to faults

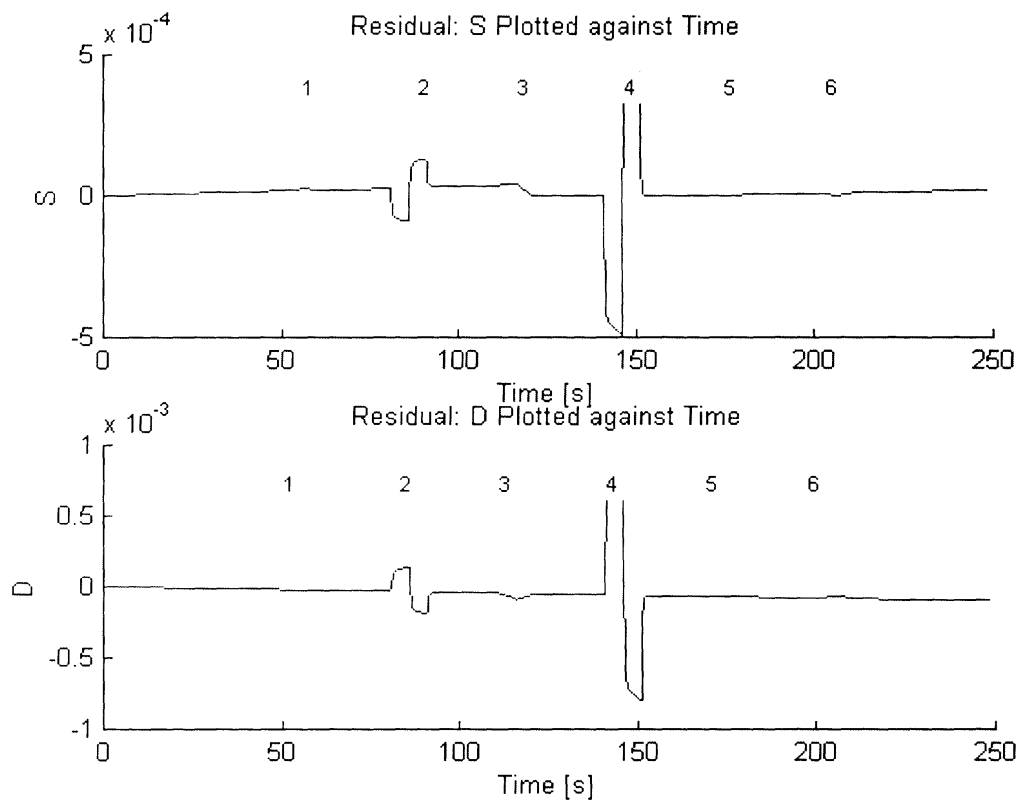


Figure 5-34 Adaptive Space Residuals S & D using RLS estimation with λ, subject to faults

Close analysis of the residuals finds that Residuals M1, V1, M2 and V2 are able to detect Sensor 1 faults, Residuals M1, V1, V2, S and D detect Sensor 2 faults and Residuals V1 and V2 detect the actuator faults. Residual M2 also detects the Sensor 2 bias fault and Residuals S & D show deviations for the Sensor 1 bias fault and the actuator faults, but are not detected because of the threshold setting. The delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations. The fault indication diagram follows.

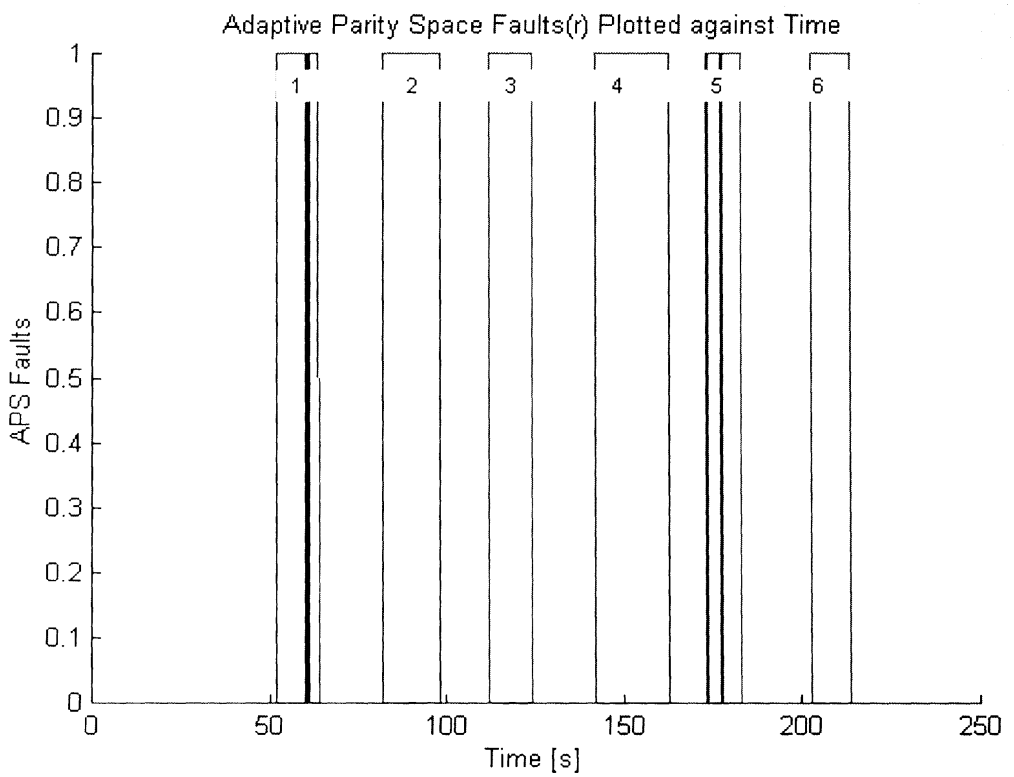


Figure 5-35 Adaptive Parity Space Fault Indication using RLS estimation with λ

The fault indication graph shows that all the faults are detected, although the Sensor 2 bias and scale faults are over detected, causing spurious faults to be detected. This is because of the threshold settings on the residuals, and the delay associated with the deviation on the residuals as the fault clears, corresponding to the Sensor 2 faults. The performance of the fault detection system can be described by:

Percentage of Correct Fault Detection: 84.48%

Percentage of Spurious Fault Detection: 15.60%

Percentage of Correlation: 36.01%

The percentage of spurious faults detected has caused the percentage of correlation to be low, even though the percentage of correct faults detected is high.

5.3.4 Recursive Least Squares with forgetting and stabilising factors

The addition of the stabilising factor into the RLS estimation with λ, should give the best fault detection performance, as this estimation scheme for the residuals should not incorporate the deficiencies outlined in the previous residual estimation schemes (Sections 5.4.1, 5.4.2 and 5.4.3). The choice of the stabilising factor, α was found to be 5.75×10^{-9} (refer to Appendix F for explanation). The fault-free residuals are shown in the following diagrams.

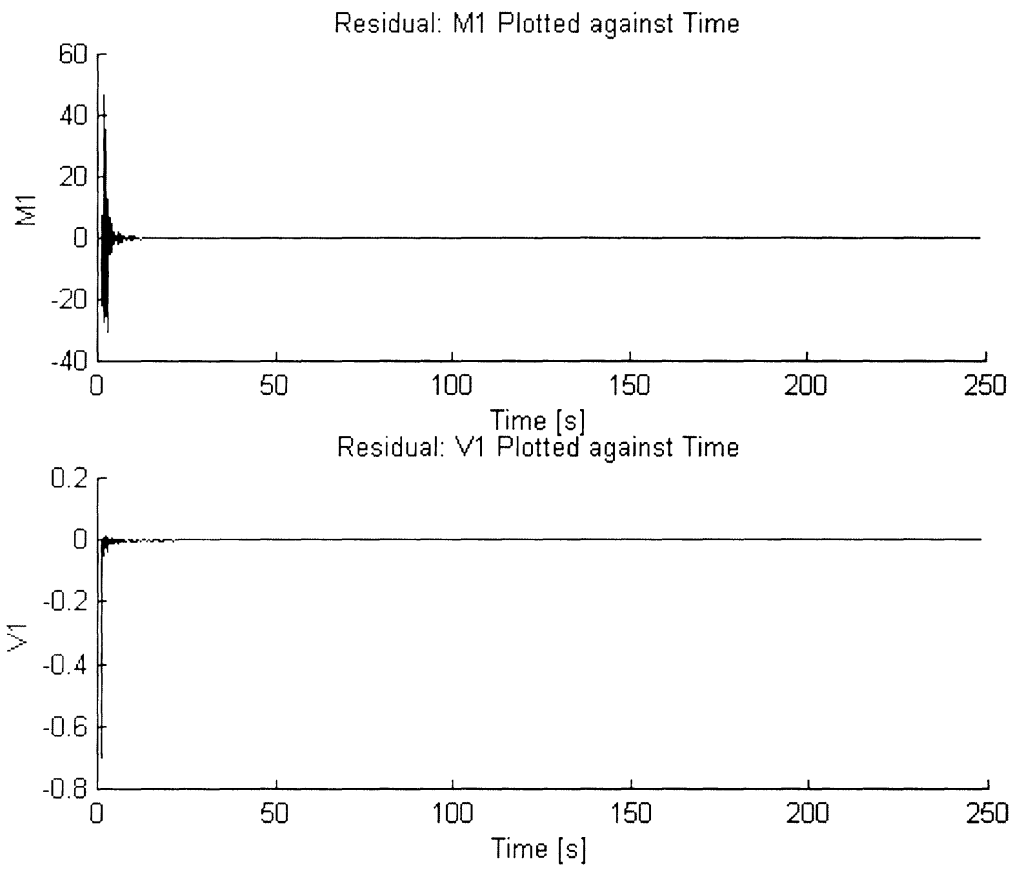


Figure 5-36 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with α and λ, subject to no faults

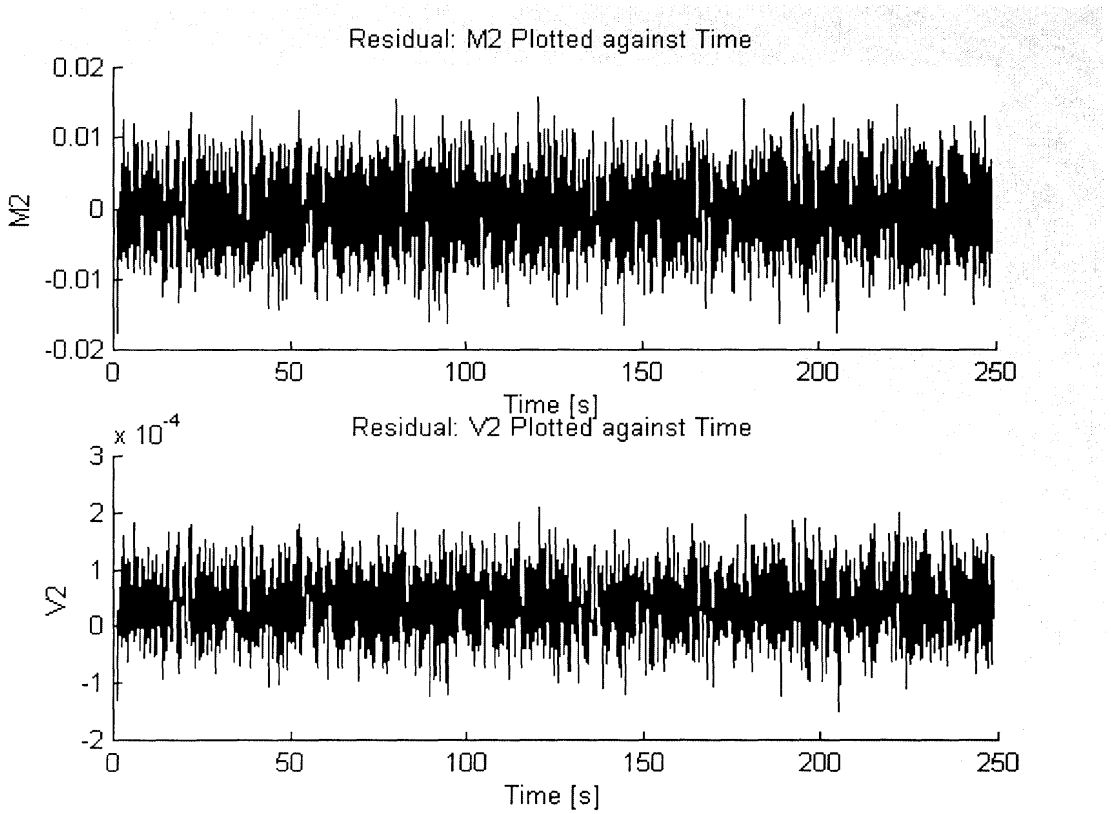


Figure 5-37 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with α and λ, subject to no faults

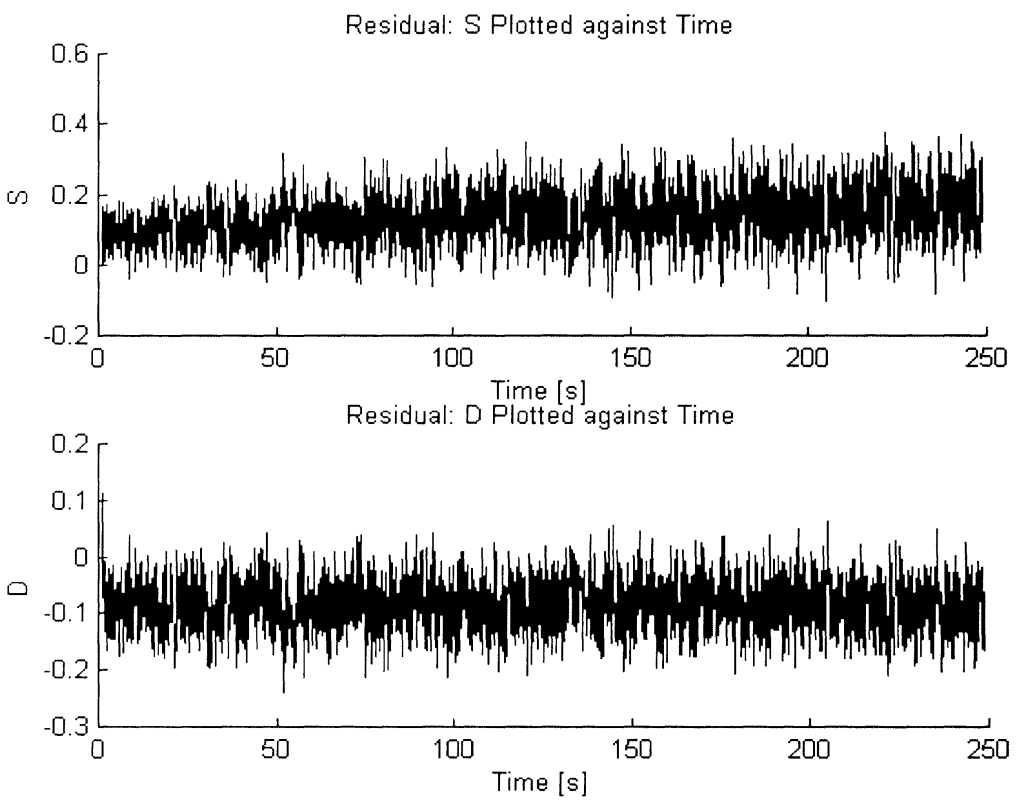


Figure 5-38 Adaptive Space Residuals S & D using RLS estimation with α and λ, subject to no faults

The residuals are seen to be noisy, as expected from the estimation scheme. The significant observations that can be made are that Residuals M1 and V1 have high frequency oscillation as the process is stepped. Residual V1 shows a lower amplitude of the oscillation than Residual M1. However, Residuals S & D are seen to have almost no overshoot in the transient state of the process. These observations can be explained as there is a trade-off in the decrease in stability of Residuals M1 and V1 for increased stability of Residuals S & D (refer to Appendix F for details). It is also noticed that the estimation bias remains in the residuals, but does not affect the fault detection performance of the method. The resultant thresholds that were obtained are:

Residual M1: 6.0×10^{1}

Residual V1: 8.0×10^{-1}

Residual M2: 2.0×10^{-2}

Residual V2: 3.0×10^{-4}

Residual S: 4.0×10^{-1}

Residual D: 2.4×10^{-1}

The following graphs show the residuals subject to faults.

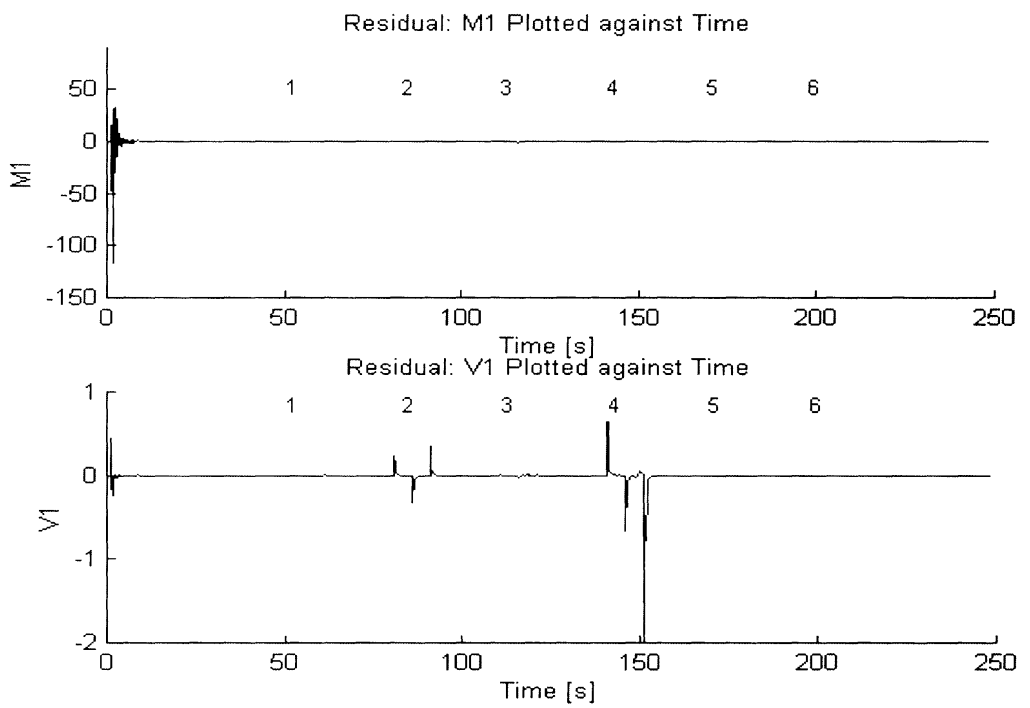


Figure 5-39 Adaptive Parity Space Residuals M1 & V1 using RLS estimation with α and λ, subject to faults

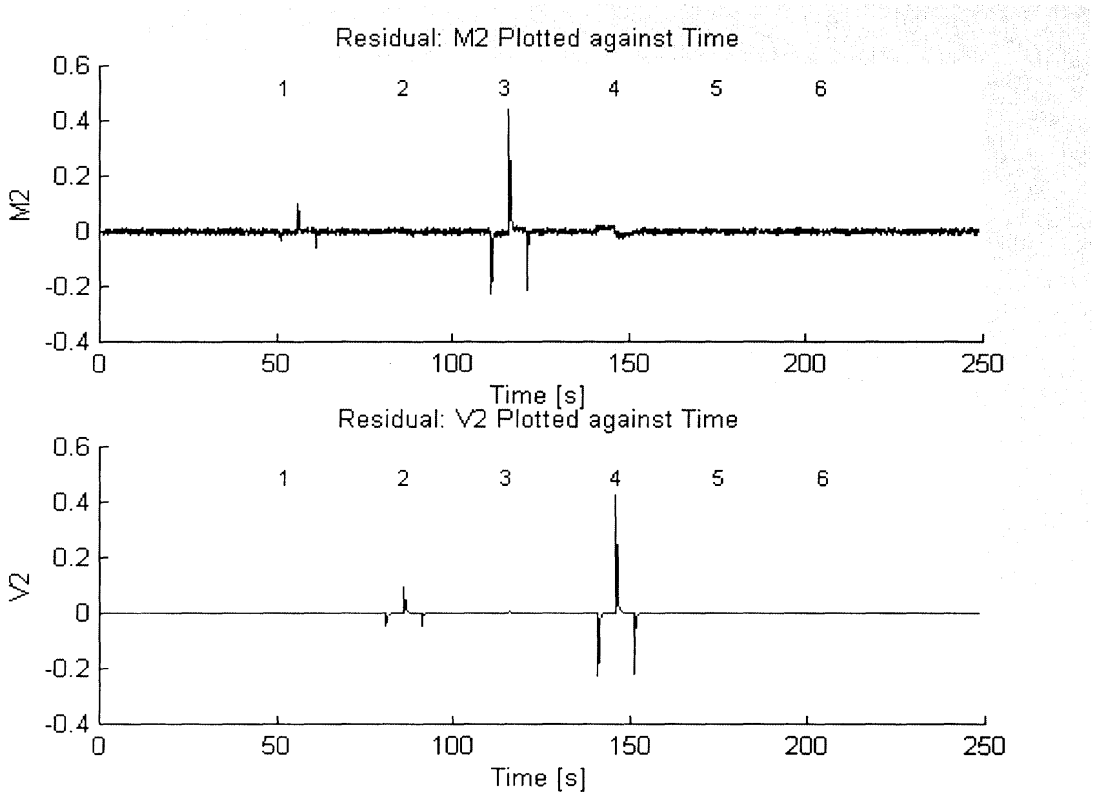


Figure 5-40 Adaptive Parity Space Residuals M2 & V2 using RLS estimation with α and λ, subject to faults

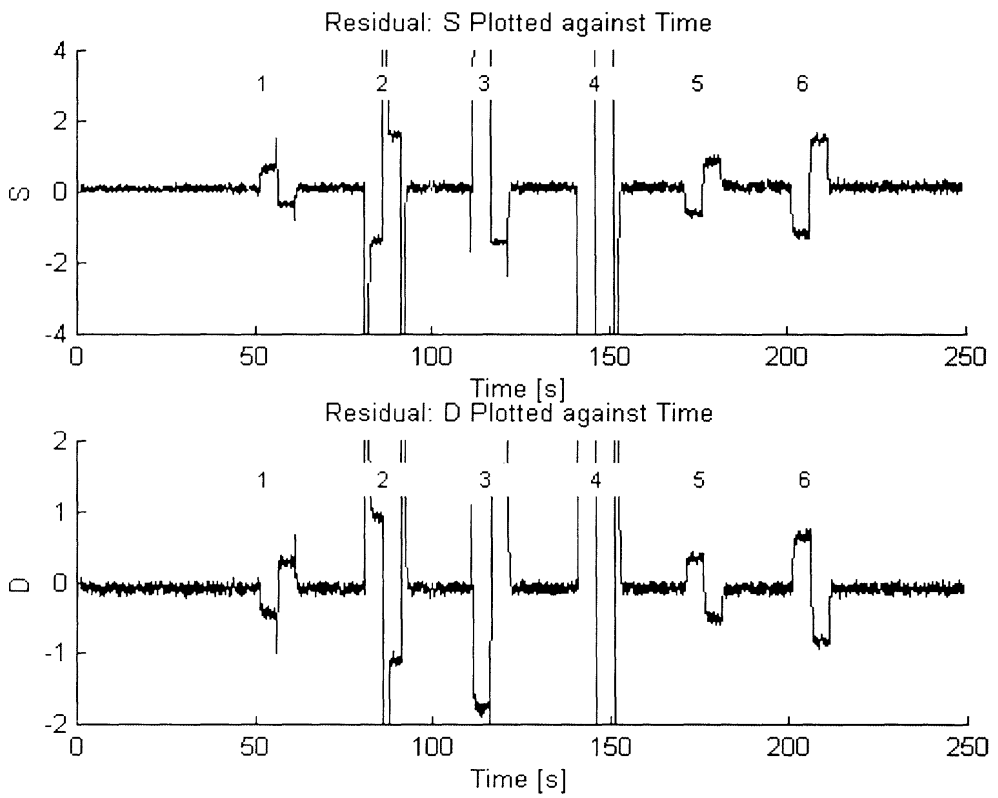


Figure 5-41 Adaptive Space Residuals S & D using RLS estimation with α and λ, subject to faults

Analysis of the residual graphs shows that the initial oscillation on Residual M1 exceeds the current threshold, and causes the indication of spurious faults. This is observed in the fault indication graph, which follows. Furthermore, Residual M1 does show small deviations for Sensor 1 faults and Residual V1 shows deviations for all faults, but none are detected because of the threshold settings. Residuals V2, S and D are able to detect all faults. Also, Residual M2 is able to detect Sensor 1 faults and shows deviations for Sensor 2 faults - which cannot be detected because of the threshold setting. Finally, the delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations.

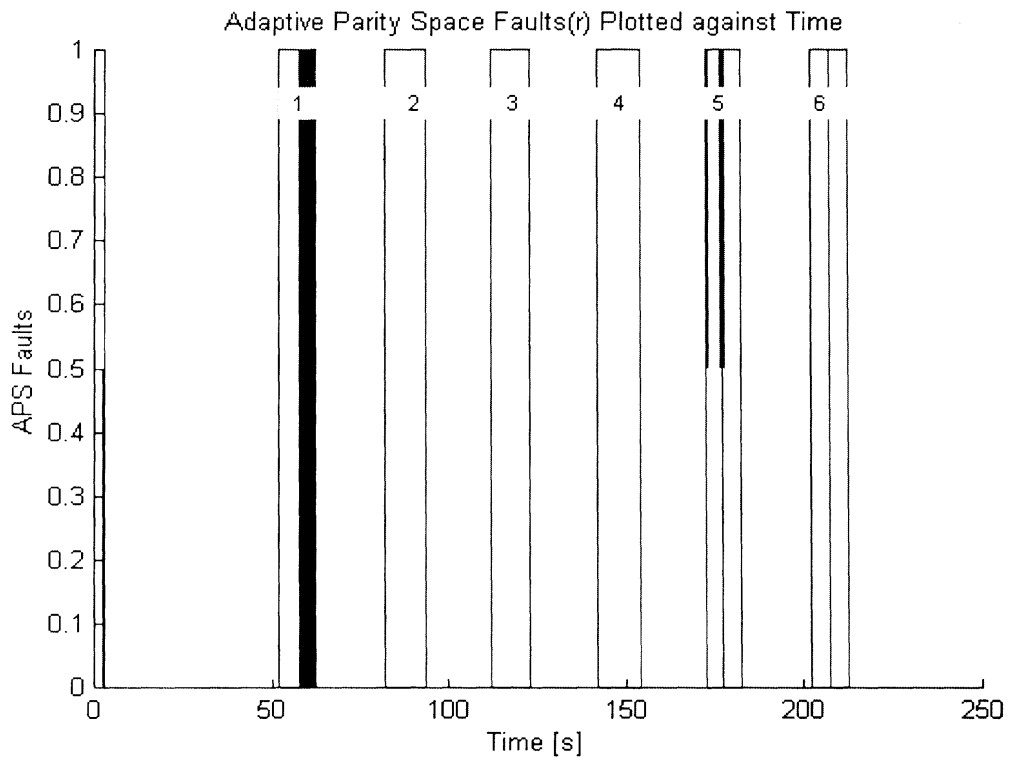


Figure 5-42 Adaptive Parity Space Fault Indication using RLS estimation with α and λ, subject to faults

As can be seen from the fault indication graph, all the faults are fairly accurately detected. However, there are spurious faults detected when the process is stepped, due to Residual M1. In order to prevent the spurious fault detection, no threshold was placed on Residual M1 since it was not able to detect any faults correctly and therefore unable to provide any useful information. The new fault indication graph follows.

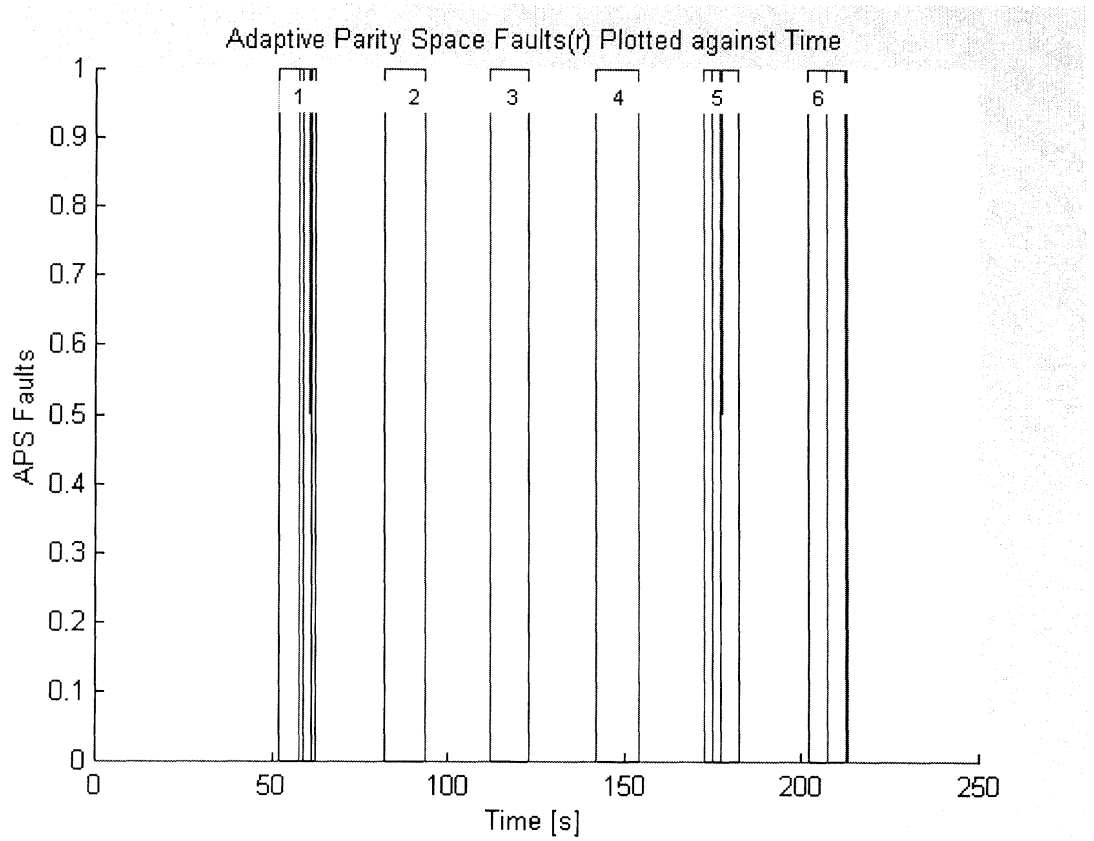


Figure 5-43 New Adaptive Parity Space Fault Indication using RLS estimation with α and λ, subject to faults

From the preceding graph, it can be seen that all faults are correctly and fairly accurately detected. Also, there is not much delay in the fault detection, resulting in a minimal amount of spurious fault detection. The above observations can be justified by the performance:

Percentage of Correct Fault Detection: 86.96%

Percentage of Spurious Fault Detection: 6.21%

Percentage of Correlation: 67.78%

The results show a high percentage of correct fault detection, low percentage of spurious fault detection, and thus a fairly high percentage of correlation. The percentage of correlation would be higher if there was a smaller delay in the fault detection.

5.3.5 Summary for the Adaptive Parity Space method

Using a simple LS estimation for the residuals gives fairly good results. However, the shortfalls of the LS estimate which occur in practice (described in Appendix D) make it an unreliable method for estimation purposes. The reason why LS obtained fairly accurate results is because the estimate uses the complete data range to obtain its parameter estimates. The RLS estimations which used a forgetting factor can be said to be almost like a LS estimate since the forgetting factor is low, which implies that old data is forgotten rapidly (the three most recent measurements are used in the estimations). This also implies that the estimation scheme does not need the most recent measurements for good fault detection performance. In fact, the use of the most recent measurements in the RLS estimation produced the worst fault detection performance of the four estimation methods investigated. The performance of the adaptive parity space method improved with the introduction of firstly, the forgetting factor, and secondly, the stabilising factor on the RLS estimation. The RLS with forgetting factor and stabilising factor showed the best estimation of the residuals, resulting in fairly accurate and correct fault detection.

Refer to Appendix F for details on the effect that the forgetting factor and stabilising factor have on the performance of the adaptive parity space method of fault detection.

5.4 Observer Results

From Section 4.2.3, it is known that the observer method of fault detection uses two threshold to indicate whether a fault has occurred or not. As with the previous sections, the fault-free residuals need to be analysed in order to determine the threshold setting.

The following graph shows the residuals obtained for the observer method, in order to determine the initial threshold for fault indication

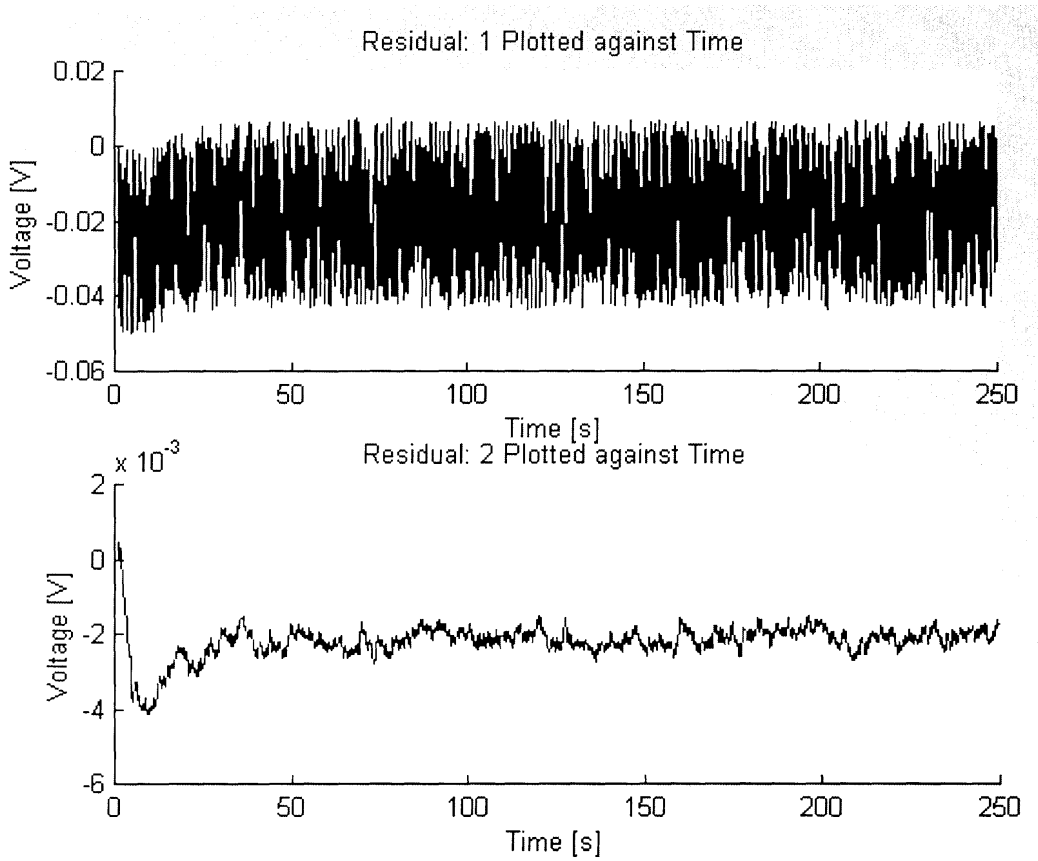


Figure 5-44 Observer residuals in a fault-free environment

Residual 1 corresponds to the parameter change of the level in Tank 2 and thus shows more noise than Residual 2, which corresponds to the parameter change of the Tank 1 level. The initial dip in the residuals is attributed to the change in the parameters when the step is introduced. Thereafter, the residuals settle down to values, which have a negative bias (due to the Kalman filter estimation of the process outputs), and thus have a non-zero mean. This effect could be problematic if the residuals show small deviations in the presence of faults (as expected from the actuator faults), since the threshold will have to be adjusted in order to detect faults outside the maximum value of the bias. The threshold for the method is chosen to be 0.055, as both residuals do not exceed this value in the fault-free case. The following two diagrams show the residuals obtained when subjected to faults and the resultant fault indication graph.

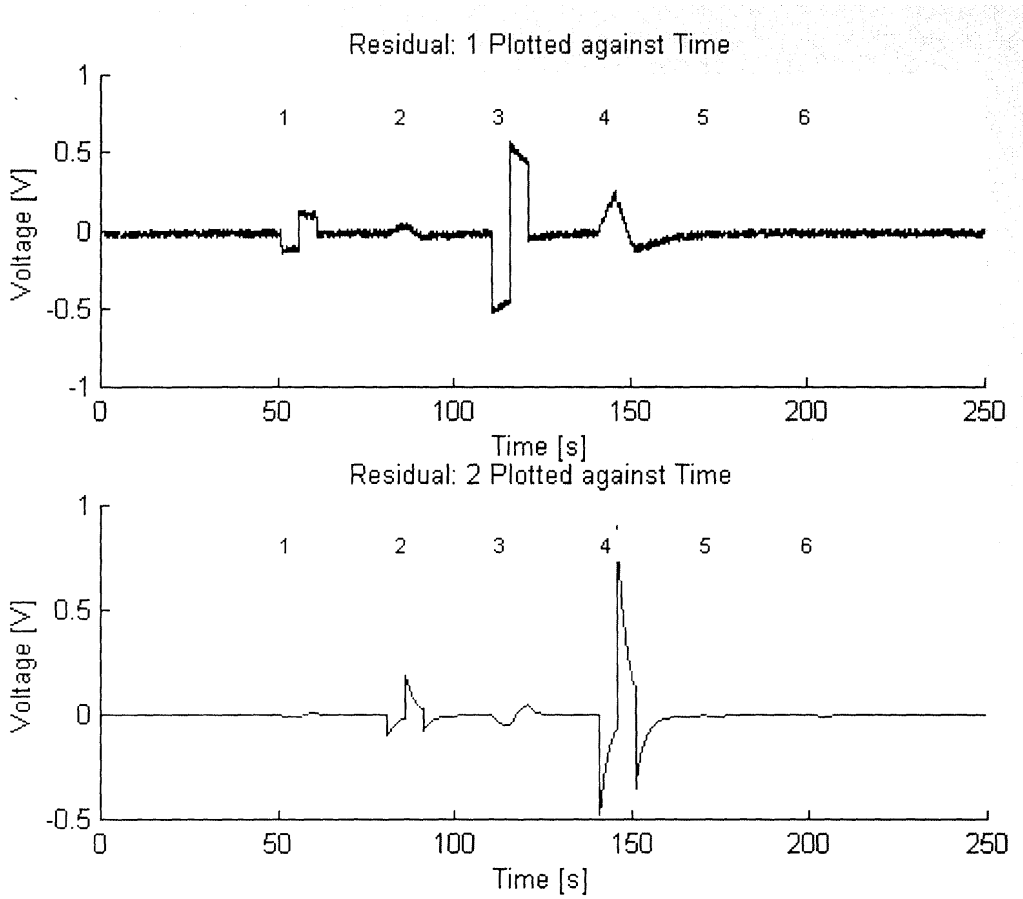


Figure 5-45 Observer residuals subjected to faults

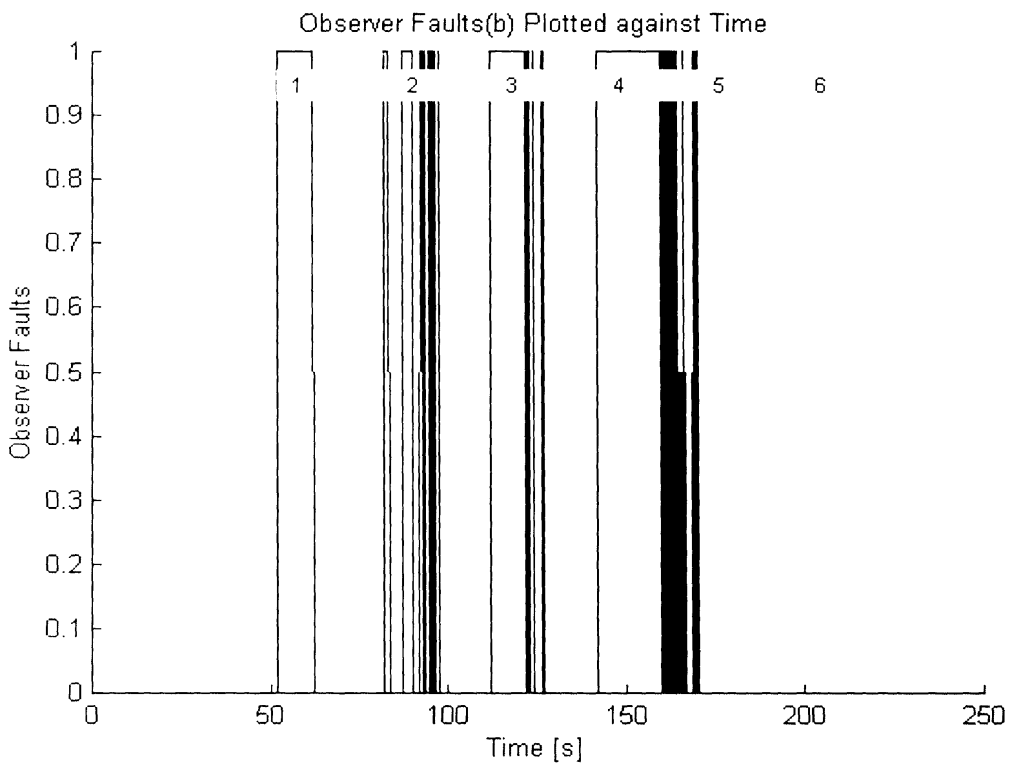


Figure 5-46 Observer Fault Indication

From the residual graphs, it can be seen that Residual 1 does not show visible deviations for the actuator faults, which is confirmed by analysis of the fault indication graph. Residual 1 shows near perfect deviations for Sensor 1 faults and Residual 2 shows good deviations for Sensor 2 faults and slight deviations for the actuator faults. The actuator faults are not detected because of the high threshold used. The use of individual thresholds on the residuals could be used to investigate the actuator fault detection. The decay of the residual deviations is explained in the same manner for the parity equation method (dependent on the Q matrix). The delay associated with the residual deviations from the fault injections is found to be 1[s], for all the deviations. The performance of the method can be described by:

Percentage of Correct Fault Detection: 52.64%

Percentage of Spurious Fault Detection: 8.86%

Percentage of Correlation: 25.10%

The percentage of correct detection could be higher with the detection of the actuator faults. The percentage of correlation is low because of the amount of spurious faults caused by the delay in the residual deviations.

5.4.1 Use of individual thresholds on each residual

From the previous section, it is expected that the observer method of fault detection should detect all faults with the use of individual thresholds. The thresholds used were:

Residual 1: 0.0550

Residual 2: 0.0050

The new fault indication graph is shown in the following diagram.

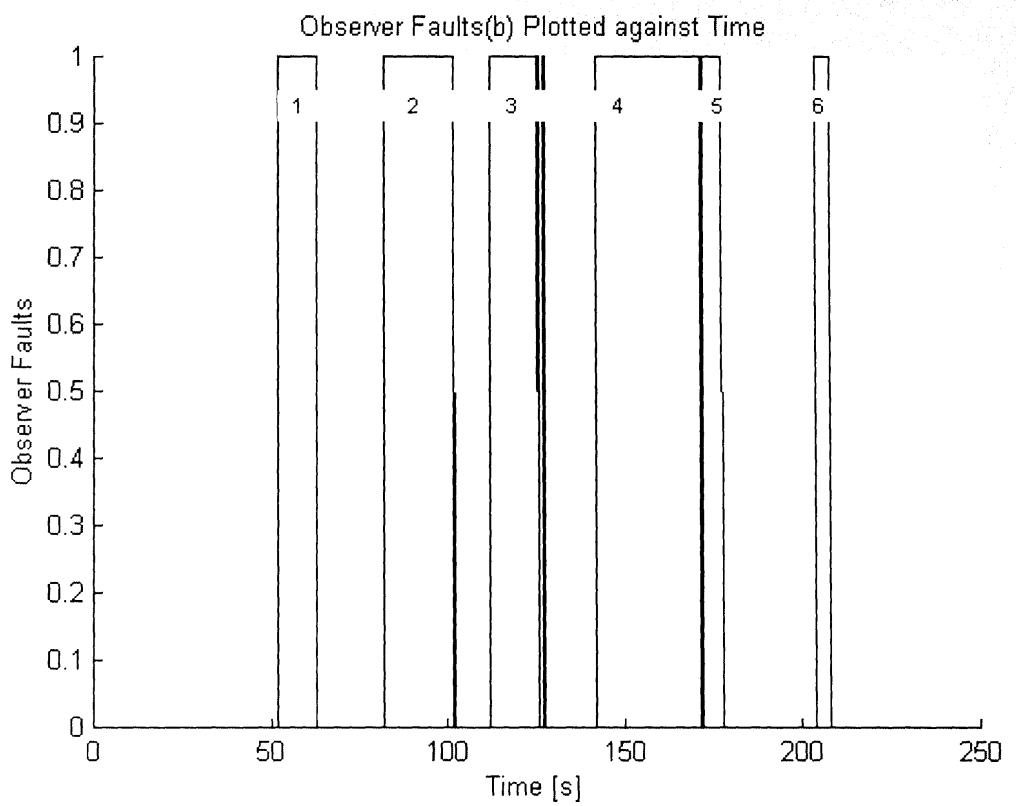


Figure 5-47 Observer Fault Indication with two thresholds

The graph shows that the actuator faults are now detected, at the expense of detecting spurious faults caused by the delay of the residual deviations, as the faults clear. The performance may be described by:

Percentage of Correct Fault Detection: 76.07%

Percentage of Spurious Fault Detection: 19.95%

Percentage of Correlation: 14.03%

The performance results using two thresholds show a significant increase in the performance of the fault detection method. The percentage of correct faults has increased with the percentage of spurious faults detected. This is because of the detection of the actuator faults and the delay of the detection, causing the spurious faults. As a result of the increased percentage of spurious faults, the percentage of correlation has thus decreased.

5.4.2 Summary for the Observer method

The inclusion of two thresholds in the observer method of fault detection has made it possible to detect actuator faults, otherwise left undetected with the use of a single threshold. With the use of two thresholds, came an increase in the percentage of correct faults detected, as well as an increase in the percentage of spurious faults, resulting in a decrease in the correlation percentage obtained from the use of one threshold. The decrease in the correlation percentage would indicate that the use of two thresholds is not better than the use of one threshold. However, it is important to detect faults that exist rather than not to detect them at all, especially in the real world. Bearing this in mind, using two thresholds creates a better observer fault detection architecture, than an architecture that would use a single threshold.

5.5 Performance of Fault Detection Methods with Varying Fault Parameters

This section shows the response of the three fault detection schemes when their fault parameters are changed from the original parameters stated in Section 5.1. The performance of the fault detection methods is analysed through variation of:

- Fault amplitude from 0.25 – 5.00 Volts
- Fault duration from 3 – 30 seconds
- Shape of fault waveform i.e. Ramp, Saw, Sine, Square and Step

The static simulation parameters have been described in Section 5.1. The results and analyses are presented in the following sub-sections. The colours seen on the graphs, which follow, are dependent on the *z-axis* values (percentage correlation). The large correlation values are shown by magenta and red, which lighten through orange to yellow then green and finally indigo, as the values decrease.

Theory predicts that large correlation percentages should be obtained for faults with large amplitudes (approximately 1V and larger) and fault duration (approximately 10[s] and longer), while low correlation percentages should be obtained for faults with low amplitudes (up to approximately 1V) and duration (up to approximately 10[s]). This is explained by the following facts:

- Large fault amplitudes cause large deviations in the residuals. As a result, faults that were previously partially detected (over the fault duration) or not detected at all, can now be detected. The percentage of correct fault detection increases, causing the percentage of correlation to increase, provided that the percentage of spurious faults does not increase. The percentage of spurious faults would only increase if the delay associated with the decay of the residual deviations increases (refer to Section 5.2.1 for further information).
- Low fault amplitudes cause little or no deviation in the residuals. This implies a low percentage of correct fault detection. The percentage of spurious faults detected decreases since there are few residual deviations that decay slowly. As a result, the percentage of correlation is low.
- Faults with a long duration cause an increase in the percentage of correct faults detected as there are more correct faults detected over a long fault duration in comparison to the percentage of corrects fault detected over a shorter duration (from Equation 96). This is explained by the fact that the delay in detection affects the percentage of correct faults detected. Therefore, over a long fault duration the delay is less significant than over a short fault duration. The percentage of spurious faults should also increase, since the fault-free time period becomes larger (Equation 97). The result is that the percentage of correlation becomes larger, provided that the fault detection method is good, else the percentage of spurious faults would increase for a bad fault detection method, causing the percentage of correlation to either decrease or remain unaffected.
- Faults with a short duration should be partially detected, or not detected at all. This is due to the delay in the fault detection, which affects faults with a short duration more than faults with a long duration. The shorter the fault duration, the lower the percentage of correct fault detection and the greater the percentage of spurious fault detection (Equations 96 and 97). The result is a low percentage of correlation.

Theory also predicts that the shape of the fault waveform should affect the performance of the fault detection schemes to a minimal degree. Different waveforms cause different error signatures on the residuals, which result in different fault detection performances of the methods. Different fault types will not affect good fault detection methods. For example, a ramp fault waveform (which exhibits an incipient fault) will leave an error signature on the residual, which should also be incipient. As a result there is a delay in the fault detection as the residual deviation takes a longer time to exceed the threshold. This causes a lower

percentage of correct fault detection, and thus correlation, as compared to the respective percentages obtained for an abrupt fault (such as a fault with a square waveform), which is quickly detected. A good fault detection method would show similar correlation results for both incipient and abrupt faults. Refer to Appendix H for the residuals and fault indication graphs obtained for the different fault waveforms, of the fault detection methods.

From the correlation results obtained in Sections 5.2.2, 5.3.4 and 5.4.1, it is expected that the adaptive parity space method of fault detection should show the best performance, while the parity space method should show the worst performance. The observer method should have an exaggerated performance for reasons given in Section 5.5.3.

5.5.1 Parity Space Performance

The results in this section are based on the parity space fault detection method with the use of individual thresholds, as described in Section 5.2.2 (for reasons given in Section 5.2.4). The following graphs show the percentage correlation results for varying fault amplitude and duration for ramp, saw, sine, square and step fault waveforms respectively.

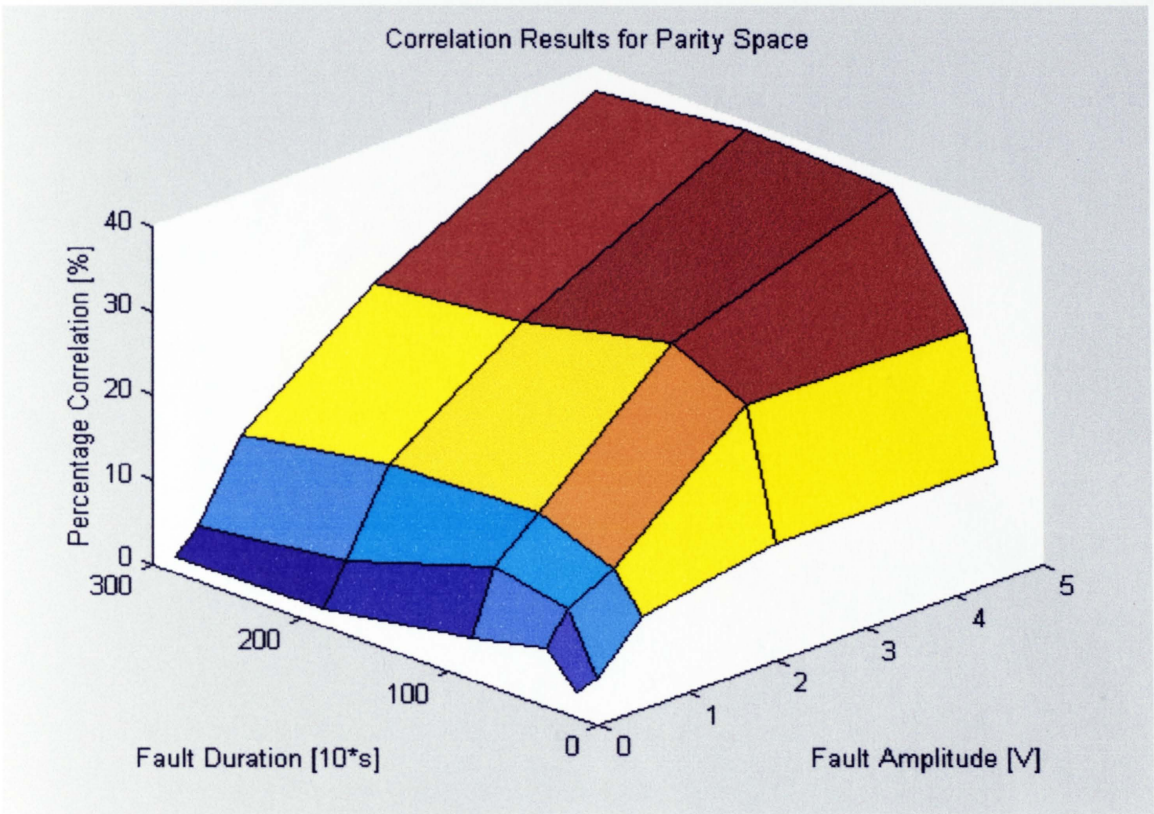


Figure 5-48 Parity Space Correlation Graph for Ramp Waveform

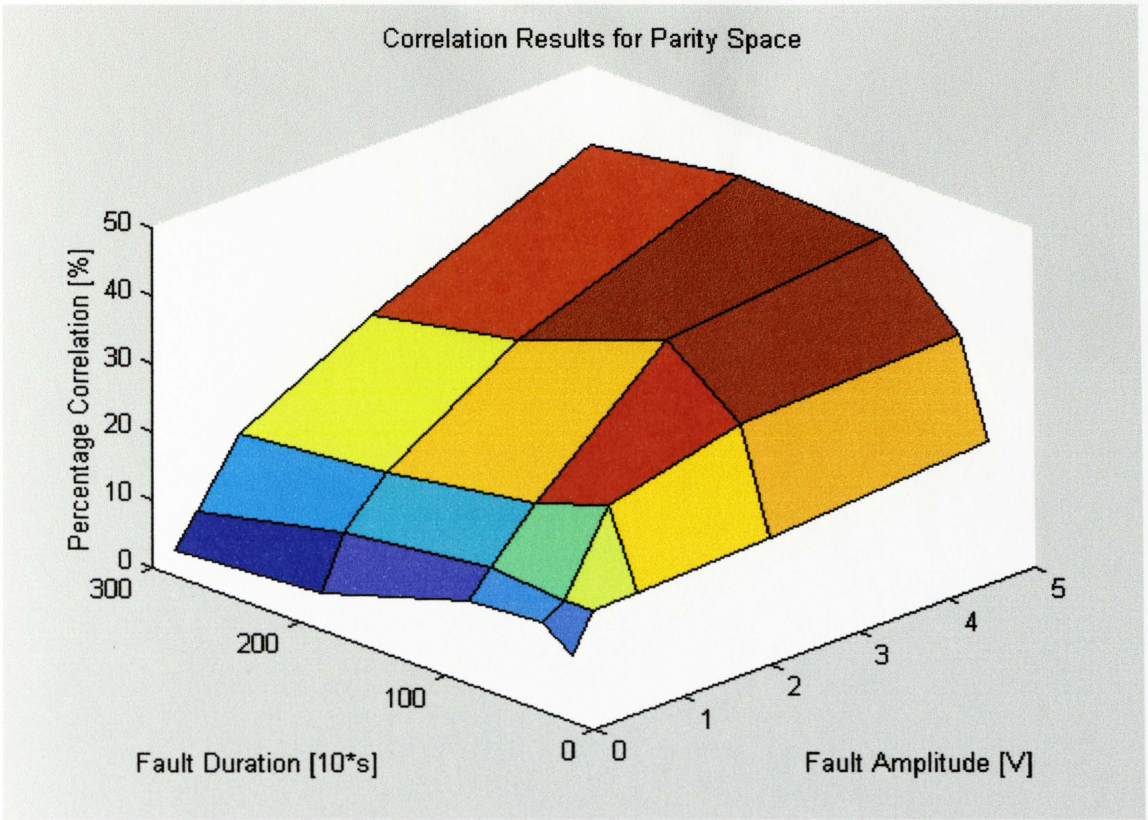


Figure 5-49 Parity Space Correlation Graph for Saw Waveform

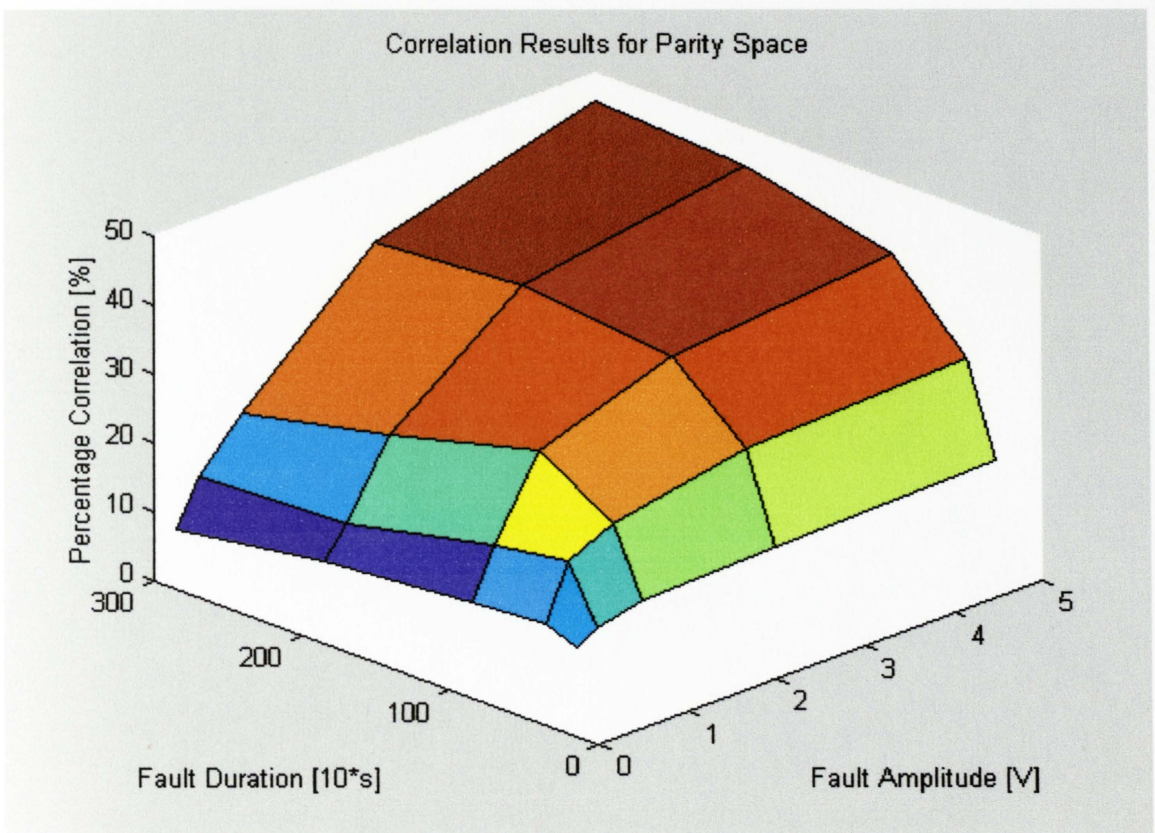


Figure 5-50 Parity Space Correlation Graph for Sine Waveform

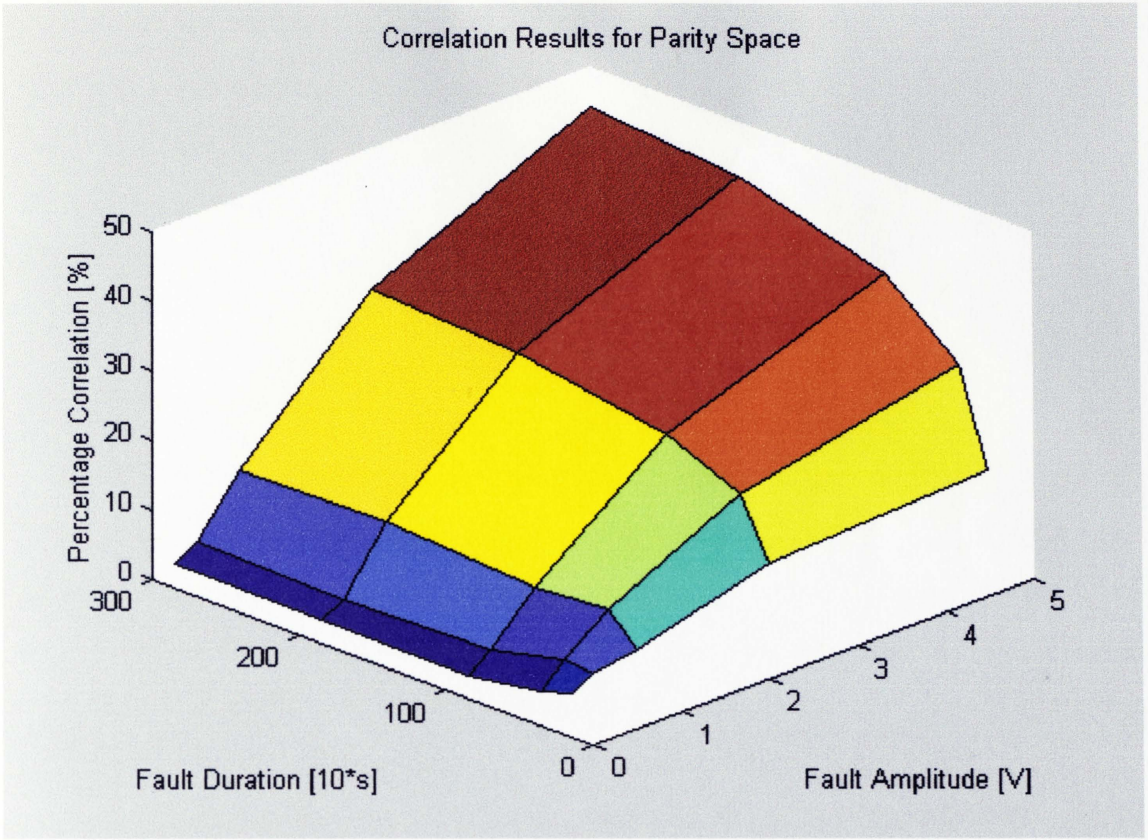


Figure 5-51 Parity Space Correlation Graph for Square Waveform

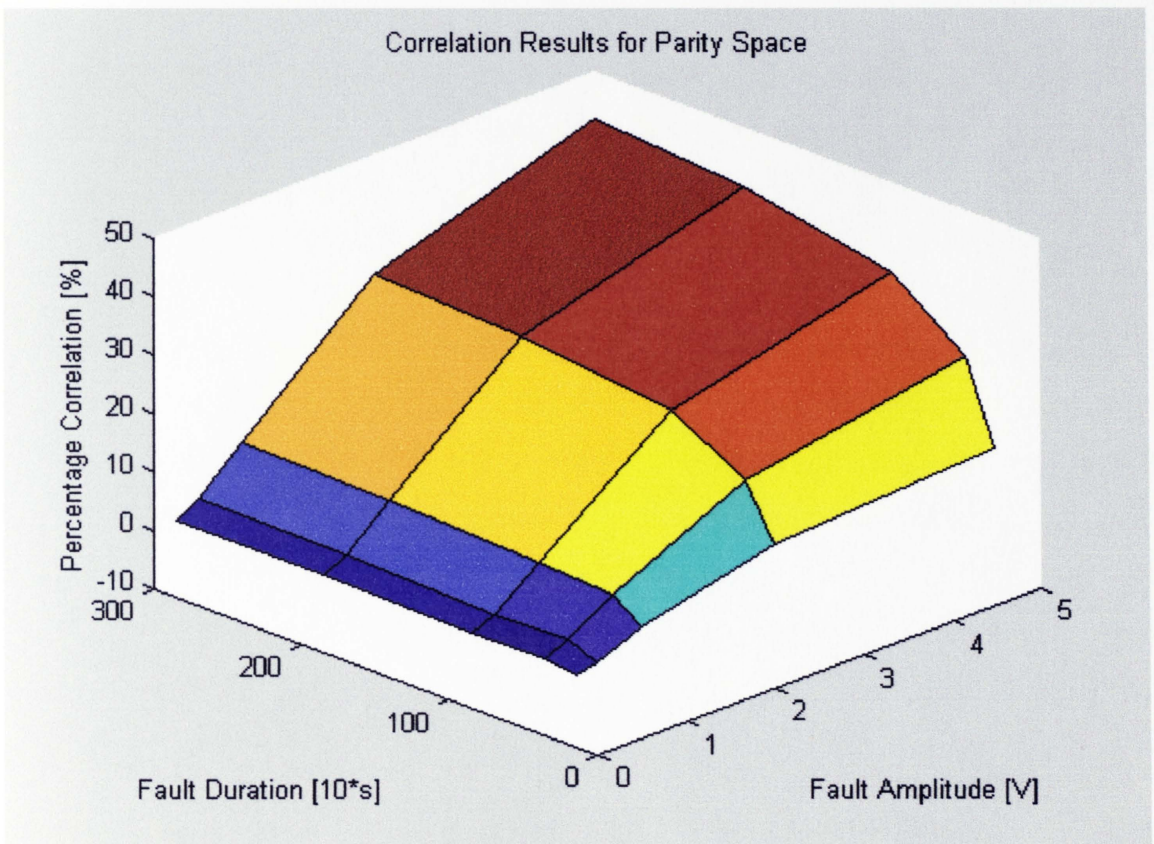


Figure 5-52 Parity Space Correlation Graph for Step Waveform

Effect of Fault Amplitude:

For all fault waveforms, the percentage of correlation increases with the fault amplitude, as predicted by theory. The increase is seen to be fairly linear, although differing for each fault waveform, and in some cases (ramp and saw fault waveforms), the increase is seen to change with fault duration.

Effect of Fault Duration:

The increase in percentage of correlation with fault duration is explained by theory. However, when the percentage of correlation decreases with fault duration, the following explanation is given. The parity space method does not detect actuator faults with low amplitudes. So as the fault duration becomes larger, the percentage of correct faults detected becomes smaller (Equation 96), thus resulting in a lower percentage of correlation than would be obtained for a small fault duration.

Also noticeable is the fact that the incipient faults (saw and ramp waveforms) show a decrease in the percentage of correlation from a fault duration of 30[s] even with a fault amplitude of 5V, whereas the abrupt faults (square and step waveforms) show an increase in the percentage of correlation.

Effect of Fault Waveform:

The incipient faults show similar correlation results with fault duration and amplitude, to each other. The abrupt faults also show similar correlation results with fault duration and amplitude, to each other. The sine fault waveform shows correlation results similar to both abrupt and incipient fault waveforms, since it can either be considered as an incipient or abrupt fault (i.e. a sine fault waveform with small amplitude and long fault duration can be called incipient, while a sine fault waveform with a large amplitude and short fault duration can be called abrupt).

Comment:

The parity space method of fault detection is seen to be heavily dependent on both large fault amplitudes and duration, which yield only average performance (as defined by the percentage of correlation). Generally, the performance of the fault detection method is seen to be more dependent on fault amplitude than fault duration, but the method becomes equally dependent on fault duration as the fault amplitude increases. Furthermore, the fault detection method is capable of detecting faults with a small fault duration with almost equal success as a fault with a large duration, for faults with amplitudes 1V or less.

The parity space method of fault detection does not exhibit good performance, even with large fault parameters, and abrupt faults (since the highest percentage of correlation obtained was 45.89%). Also the abrupt faults showed marginally better correlation results as compared to the incipient faults, as predicted by theory.

5.5.2 Adaptive Parity Space Performance

The results in this section are based on the adaptive parity space method using a recursive least squares estimate with forgetting factor and stabilising factor (Section 5.3.4, for reasons given in Section 5.3.5). The following graphs show the percentage correlation results for varying fault amplitude and duration for ramp, saw, sine, square and step fault waveforms respectively.

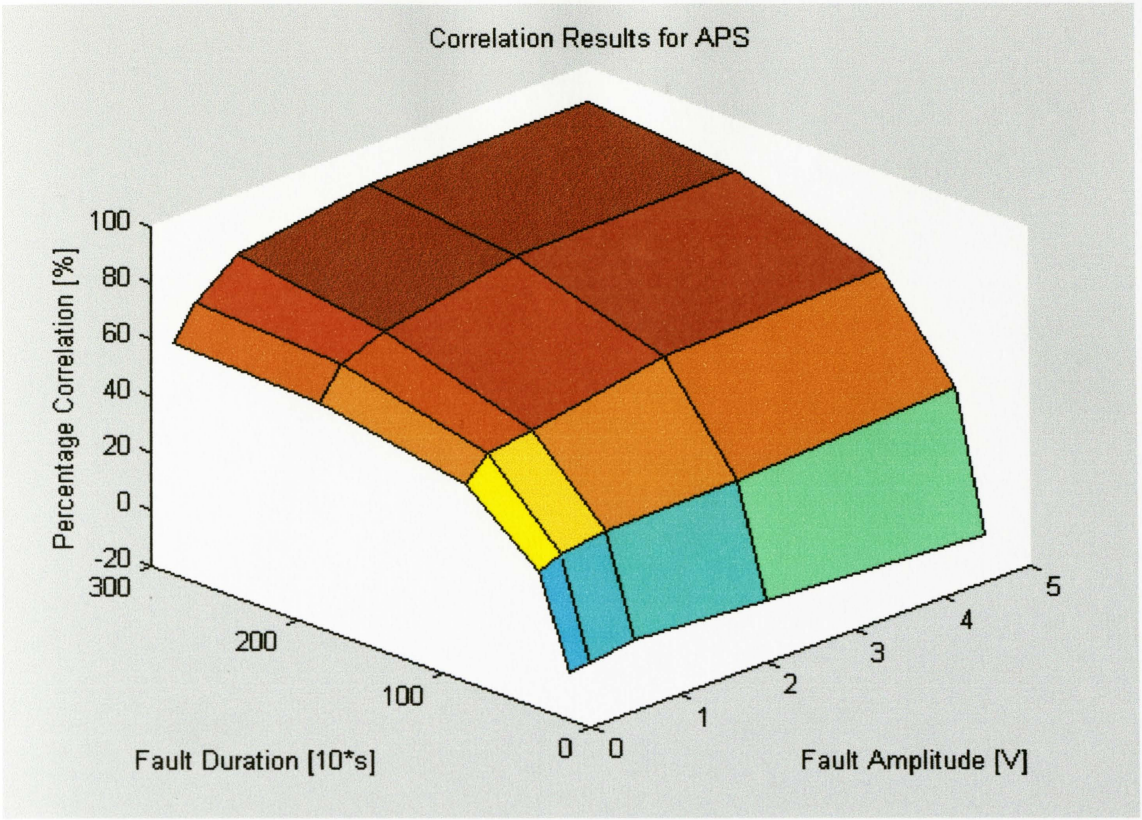


Figure 5-53 Adaptive Parity Space Correlation Graph for Ramp Waveform

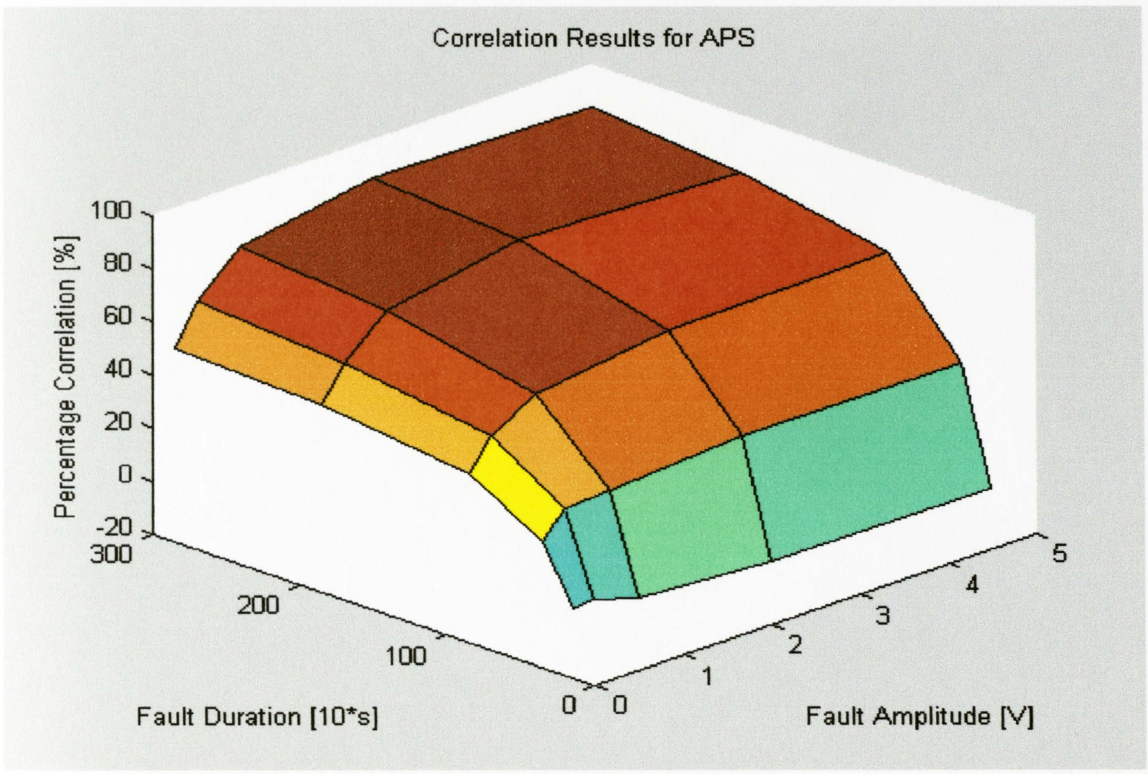


Figure 5-54 Adaptive Parity Space Correlation Graph for Saw Waveform

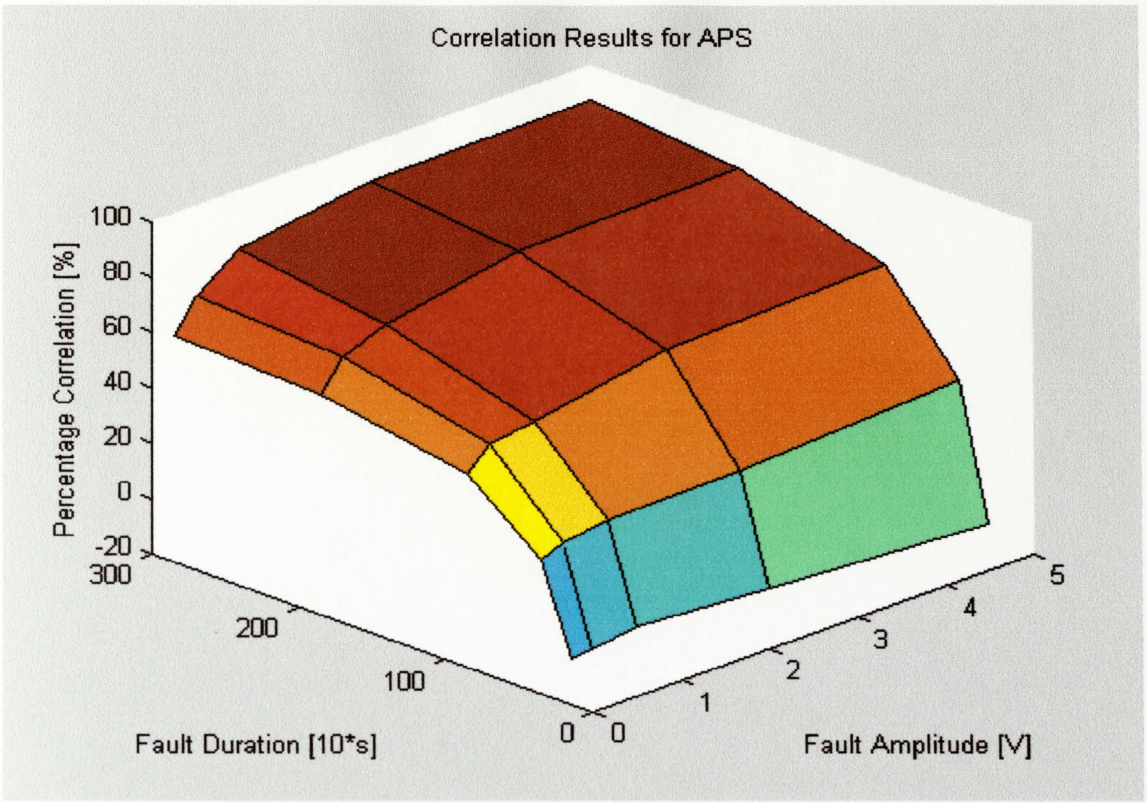


Figure 5-55 Adaptive Parity Space Correlation Graph for Sine Waveform

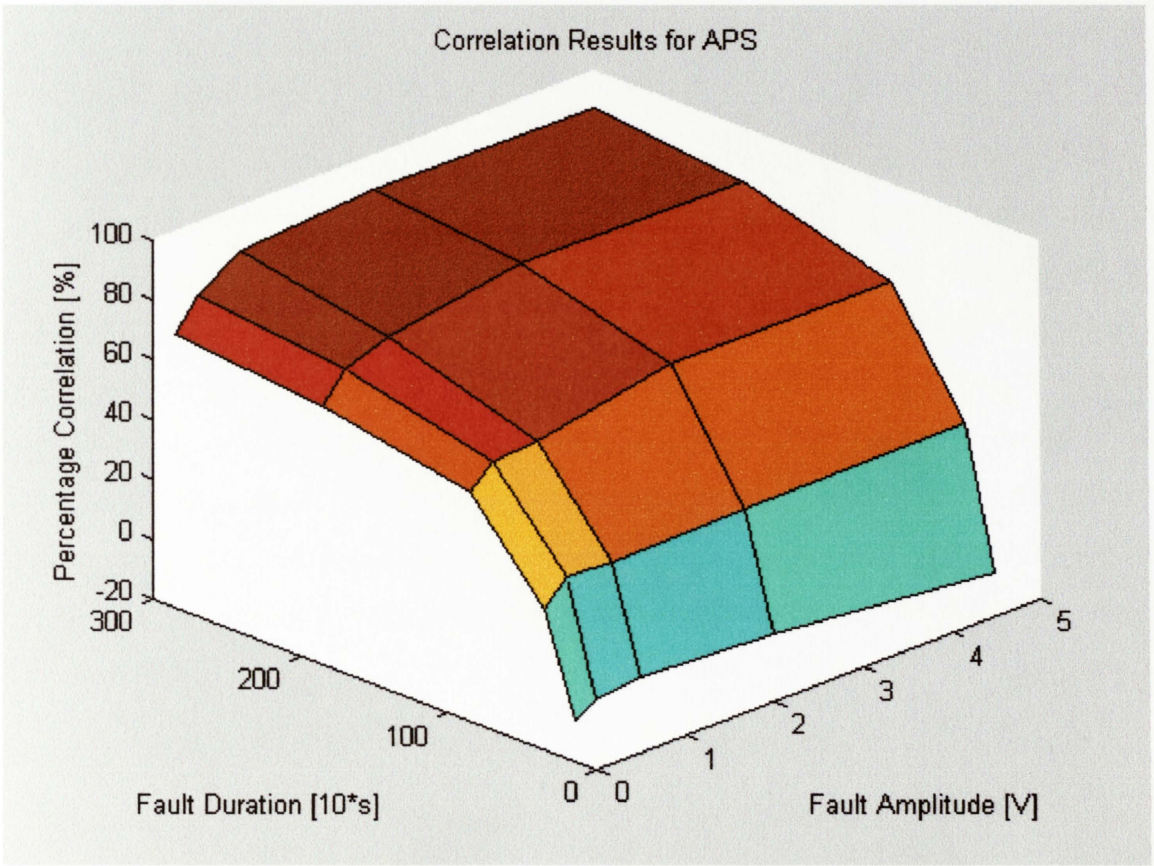


Figure 5-56 Adaptive Parity Space Correlation Graph for Square Waveform

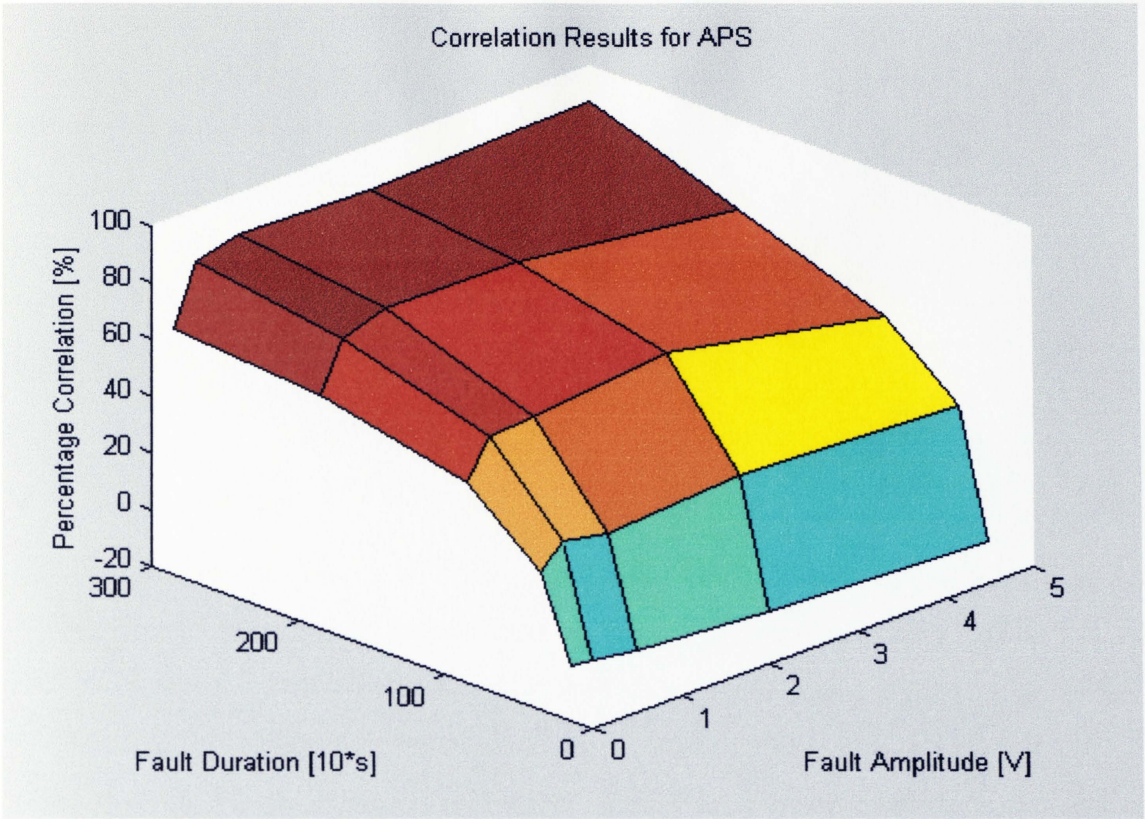


Figure 5-57 Adaptive Parity Space Correlation Graph for Step Waveform

Effect of Fault Amplitude:

The general trend is that the percentage of correlation increases with the fault amplitude for all the fault waveforms. The increase in percentage correlation is larger when the fault amplitudes are small (0.25V to 1.00V). This shows that the percentage of correlation does not significantly increase in the presence of faults with a large amplitude (greater than 1V). The slight decrease in the percentage of correlation, observed when the fault duration is small, is explained by the fact that large fault amplitudes cause larger deviations in the residuals. As a result, the delay in the decay of the residual deviations causes spurious faults to be detected. So the percentage of correlation decreases because there is an increase in the spurious faults detected, while the percentage of correct faults detected remains approximately the same, for all fault amplitudes.

However, it can be seen that the percentage of correlation decreases at amplitude of 2.5V and 5.0V when the fault duration is 30[s] for the saw and square waveforms. Also, the percentage of correlation decreases at an amplitudes of 5.0V when fault duration is 10[s] and 20[s], for

the square and step fault waveforms. This can be explained by the fact that the adaptive parity space method was detecting spurious faults (refer to Appendix H for details). So even if the percentage of correct fault detection remains the same or increases with fault amplitude, the increase in the percentage of spurious faults causes the percentage of correlation to decrease. The increase in the percentage of correlation for the step waveform, when the fault duration is 30[s] is attributed to the fact that there is a decrease in the amount of spurious faults detected, as a result of the increase in fault duration (Equation 98).

Effect of Fault Duration:

For all the fault waveforms, at all fault amplitudes the percentage of correlation is seen to increase with the fault duration, as predicted by theory. The rate of increase in the percentage of correlation is seen to be high when the fault duration is under 10[s], thereafter the rate decreases as the fault duration becomes larger.

Effect of Fault Waveform:

The shapes of the adaptive parity space graphs are seen to be similar. This is confirmed by analysis of the correlation results obtained for the all the fault waveforms. The correlation results are seen to be fairly consistent with each other (within ±10% of each other, for any specific fault amplitude and duration).

Comment:

Before making any substantial comments, the author had decided to investigate the effect on the percentage of correlation with varying fault amplitude and duration, by increasing the threshold of Residual S (which was found to be detecting the spurious faults). The new threshold used was equal to 0.8. The following graph shows the percentage of correlation obtained for the saw fault waveform (since it is known that the shape of the fault waveform does not significantly affect the performance of the adaptive parity space method of fault detection, the graphs and correlation data for the other fault waveforms are not shown this chapter, but can be found in Appendix I).

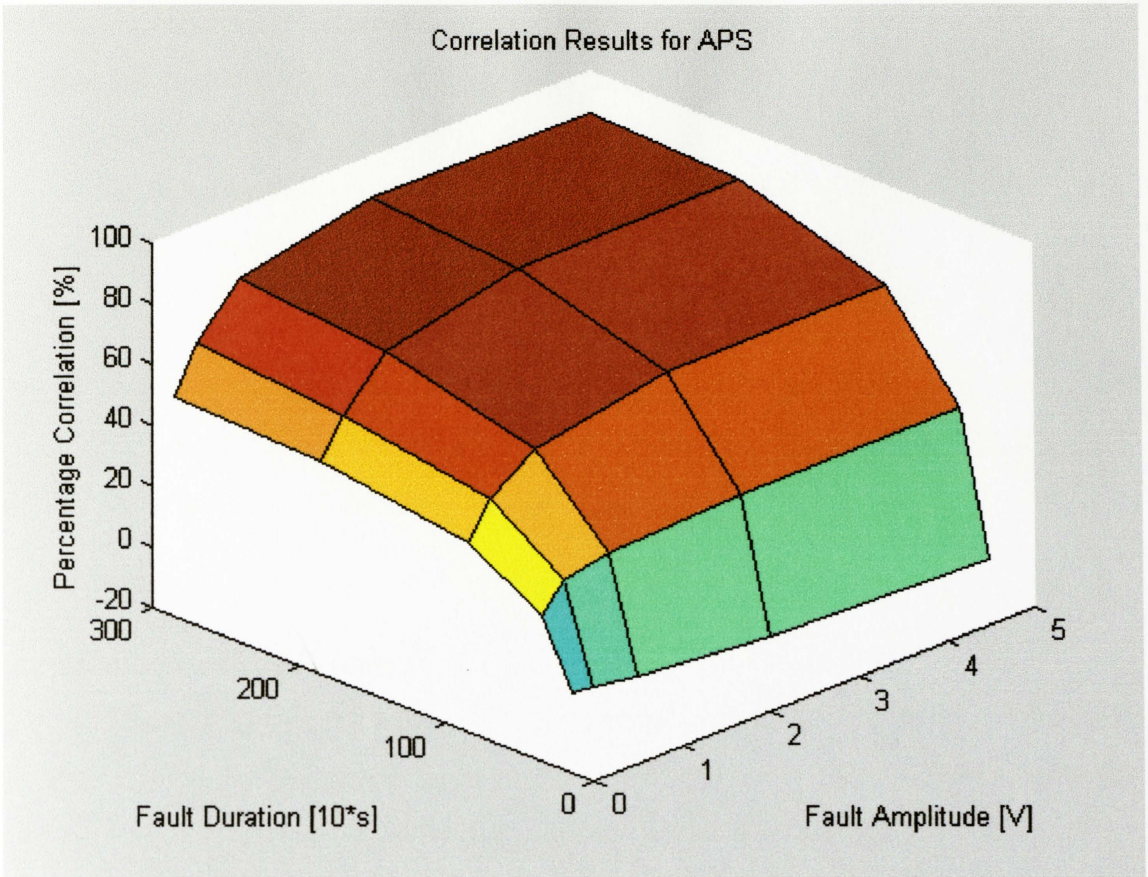


Figure 5-58 New Adaptive Parity Space Correlation Graph for Saw Waveform

Comment:

The percentage of correlation is now seen to increase with fault amplitude, irrespective of the fault duration. The graph also shows that the percentage of correlation remains as high as the percentages obtained before (with the original threshold of 0.4 on Residual S). Analysis of the other fault waveforms show correlation results that are similar to that obtained for the ramp fault waveform. Comparisons of the percentages of correlation obtained for each fault waveform (Appendices G and I) show that the correlation results are similar to those obtained before the threshold increase for Residual S. The above result of the comparison shows that the threshold setting has not actually undermined the performance of the method. In fact, the method performed better than before.

Further analysis of the correlation graphs for the adaptive parity space method show that the percentage of correlation for the adaptive parity space method of fault detection is more dependent on the fault duration than on the fault amplitude. This is noted by the fact that the percentage of correlation increases steeply with a fault duration up to 10[s], whereas the

increase with fault amplitude is seen to settle down (for amplitudes greater than 1V). So, the fault detection method is capable of detecting faults of small amplitude with almost equal success as faults with large amplitudes, irrespective of the fault duration. However, the fault detection method detects faults much more successfully when the fault duration is large as compared to when the fault duration is small.

Finally, it is observed that the fault waveform does not significantly affect the performance of the fault detection method, and thus the fault detection method can be said to detect all the fault waveforms tested with almost equal success.

5.5.3 Observer Performance

The results in this section are based on the observer method using individual thresholds, as described in Section 5.4.1 (for reasons given in Section 5.4.2). The following graphs show the percentage correlation results for varying fault amplitude and width for ramp, saw, sine, square and step fault waveforms respectively.

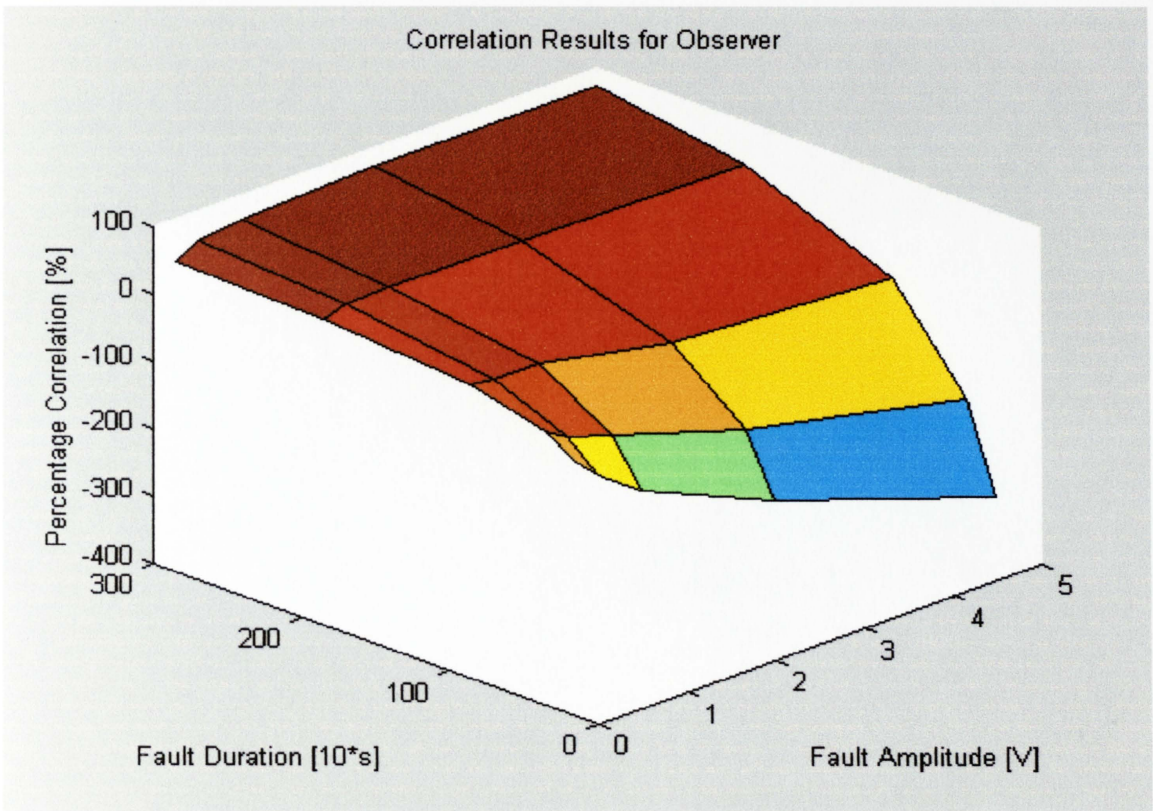


Figure 5-59 Observer Correlation Graph for Ramp Waveform

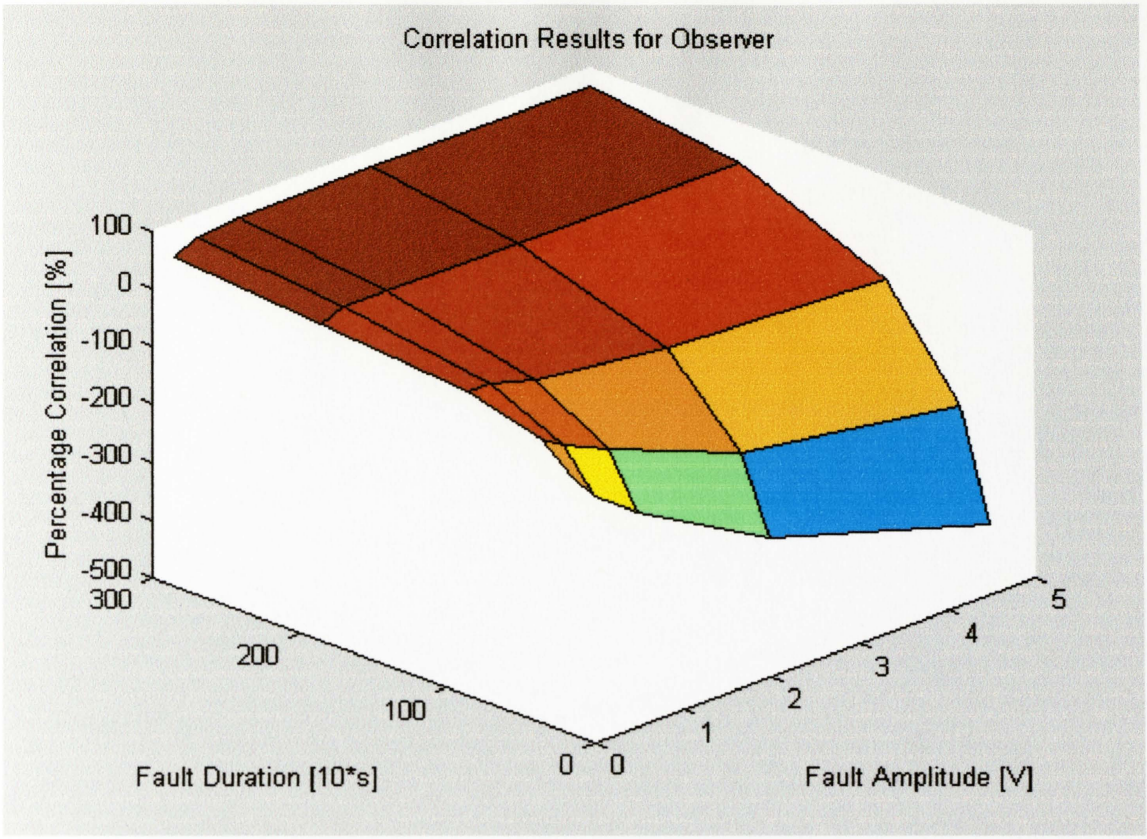


Figure 5-60 Observer Correlation Graph for Saw Waveform

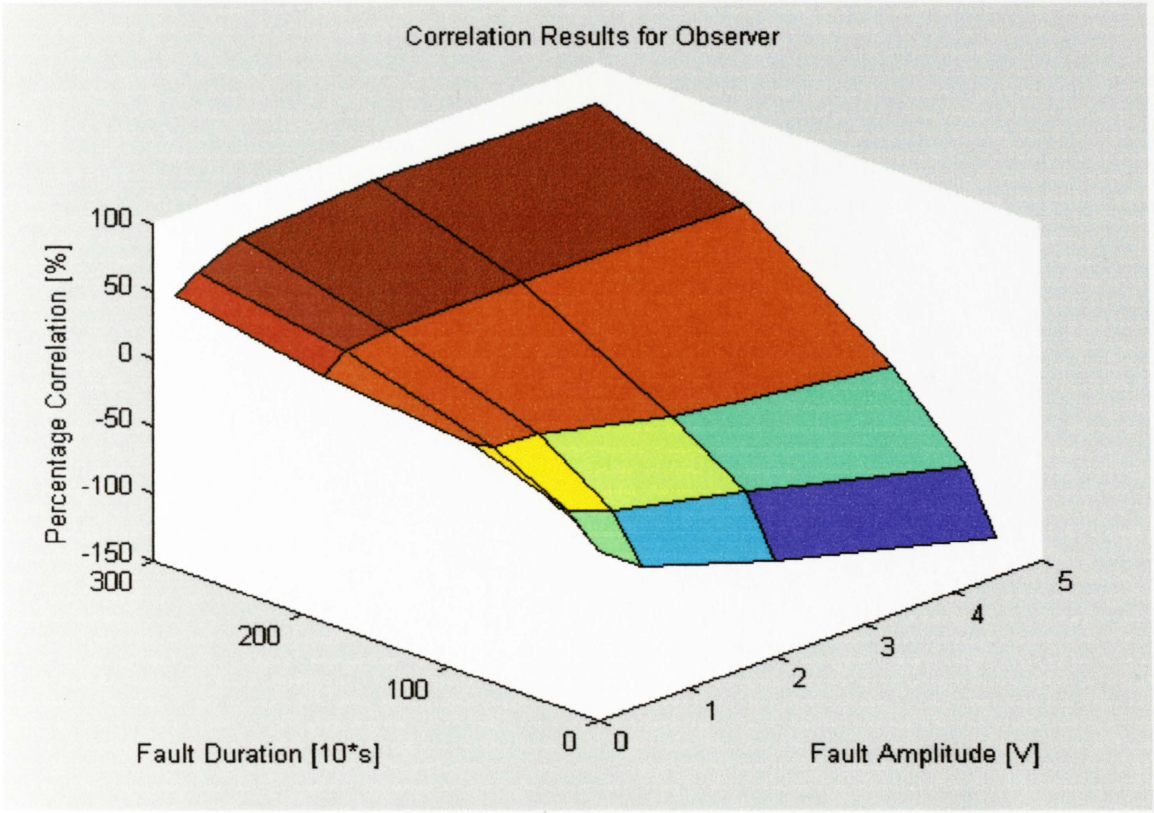


Figure 5-61 Observer Correlation Graph for Sine Waveform

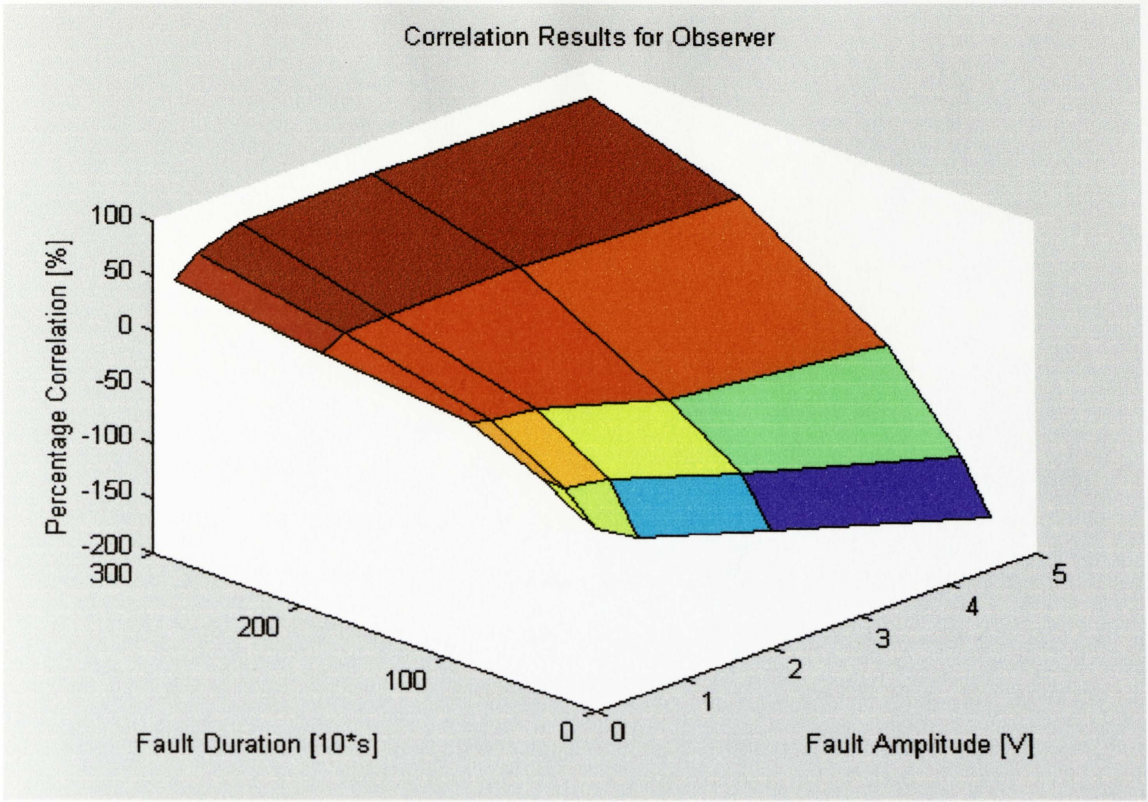


Figure 5-62 Observer Correlation Graph for Square Waveform

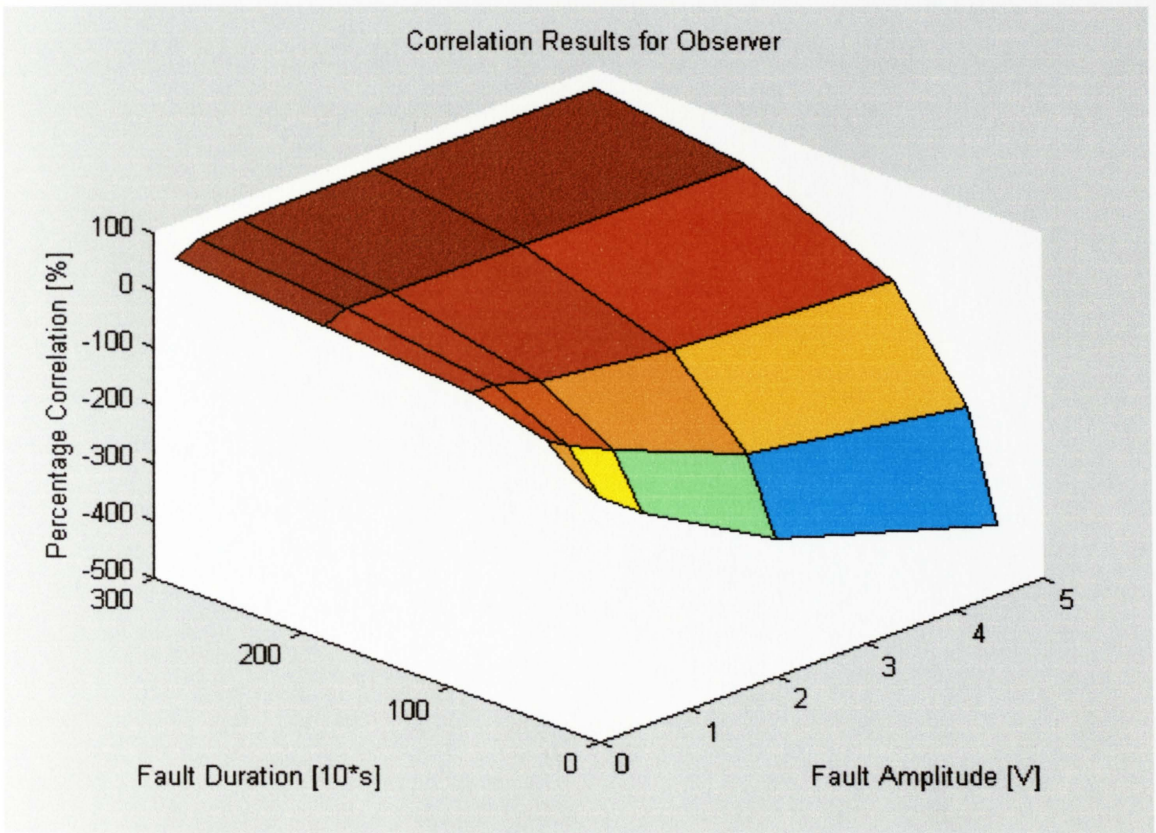


Figure 5-63 Observer Correlation Graph for Step Waveform

Effect of Fault Amplitude:

For all fault waveforms, the percentage of correlation initially increases with the fault amplitude, for a fault duration of 20[s] and 30[s], thereafter the percentage of correlation does not change significantly with an increase in fault amplitude. For a fault duration of 10[s] and smaller, there is a decrease in the percentage of correlation. The rate of decrease becomes higher as the fault duration decreases.

The decrease in the percentage of correlation with fault amplitude is explained by the following reasoning. An increase in fault amplitude causes a large deviation in the residual, which results in a large decay of the deviation. So, as the fault duration becomes smaller, the amount of spurious faults detected, increases. The amount of correct faults detected remains approximately the same for all fault amplitudes, so from Equation 98, the percentage of correlation decreases with fault amplitude. As the fault duration increases, there is a decrease in the amount of spurious faults detected, as the faults are closer together. So, the spurious faults are now detected as correct faults. This results in an increase in the percentage of correlation, as the correct faults detected remains approximately the same with an increase in fault amplitude. So, if the positions of the faults had been further apart, in a simulation with a larger duration (than currently investigated), the percentage of correlation would be seen to decrease with increasing fault amplitude, for reasons previously stated.

Effect of Fault Duration:

For all the fault waveforms, at all fault amplitudes the percentage of correlation is seen to increase steeply when the fault duration is 10[s] and under, and then increase by a lower rate, up to a fault duration of 25[s]. The fact that the percentage of correlation increases with fault duration is predicted by theory.

Effect of Fault Waveform:

As with the parity space method, the incipient faults show similar correlation results with fault duration and amplitude. The abrupt faults also show similar correlation results with fault duration and amplitude. The sine fault waveform shows correlation results similar to both abrupt and incipient fault waveforms. Although the shapes of the correlation graphs are

similar, the correlation results are within ±15% of each other, for any specific fault amplitude and duration.

Comment:

The high percentages of correlation obtained for large fault amplitudes and duration are a false indication of the performance of the observer method. A true indication of the performance would be to accept the correlation results obtained for fault amplitudes of 2.5V and duration of 20[s]. The highest percentages of correlation with the aforementioned fault parameters is now in the region between 21.92% (ramp waveform) and 36.83 (sine waveform). The incipient faults are seen to have lower correlation results than the abrupt faults, as expected. This shows that the method does not perform well, and its performance is dependent on the fault type.

Further analysis of the correlation graphs for the observer method show that the percentage of correlation is dependent to a greater extent on the fault duration than on the fault amplitude. This is noted by the fact that the percentage of correlation increases steeply with a fault duration up to 10[s], whereas there is a general decrease with fault amplitude. The fault detection method detects faults much more successfully when the fault duration is large as compared to when the fault duration is small.

5.5.4 Comparison of the fault detection methods

For a fault duration of 3[s], the parity space method shows the best correlation results, as the fault amplitude increases. Thereafter, the adaptive parity space method shows the best correlation results for the investigated fault amplitudes and for the investigated fault duration of 5[s] to 30[s]. The adaptive parity space method is not significantly affected by the type of fault (incipient or abrupt), whereas the observer and parity space methods show either significant changes in the shapes of the graphs, as a result of inconsistent percentages of correlation (attributed to the different fault waveforms). As a result, the adaptive parity space method is seen to have the best fault detection performance of the three fault detection methods investigated.

Comparison of the parity space and observer methods of fault detection, indicate that the parity space method shows better correlation results for small amplitudes and a small fault duration. The observer method shows marginally better correlation results for a fault amplitude of 2.5V and a fault duration of 20[s]. Both methods show better results for abrupt faults than incipient faults. So for faults with a small fault duration and amplitude, the parity space method is the preferred method.

5.6 Performance of Fault Detection Methods with Varying Noise Levels

The simulation and fault parameters used in this section remain the same to those described in Section 5.1, with the exception of the noise parameter, which is set to be uniformly distributed over the intervals ranging between 0V and 0.01V and 0V to 1.00V.

It is expected that the percentage of correlation should initially remain at approximately the same level, since the thresholds of the fault detection methods have been set above the noise level of 0.05V (maximum amplitude) in the original simulations (described in Sections 5.2.2, 5.3.4 and 5.4.1). For noise levels larger than 0.05V, the percentage of correlation should decrease, since the noise on the residuals should exceed the thresholds and be detected as spurious faults. The decay in the percentage of correlation with noise shows the sensitivity of the fault detection method to noise. The correlation value at each noise level, indicates the sensitivity of the particular method to noise. A large correlation value would mean good noise rejection, while a low correlation value would indicate sensitivity towards noise. The adaptive parity space method should show the least sensitivity to noise, followed by the observer and parity space methods of fault detection. The correlation results for the fault detection methods are described in the following sub-sections (details in Appendix J). This is followed by a comparison between the fault detection methods.

5.6.1 Parity Space method

The following graph shows the percentage of correlation plotted against noise level, for the parity space method of fault detection.

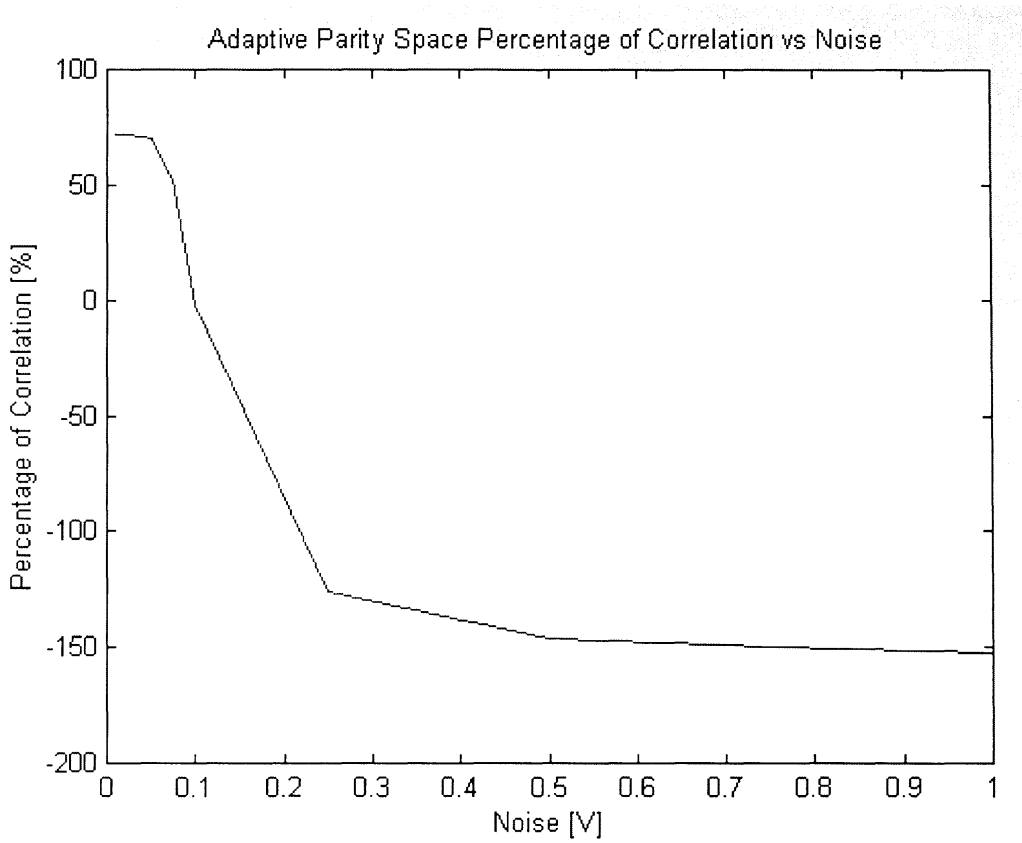


Figure 5-64 Parity Space Percentage of Correlation against Noise

Comment:

Initially, the percentage of correlation is seen to rise before decaying down to a value of approximately –150%. The initial rise in the percentage of correlation can be explained by the following fact. As the noise level increases, the residuals become noisier and thus deviate beyond the thresholds with greater frequency. Since the parity space method of fault detection does not fully detect the faults over the whole fault duration, the noise imposed on the residuals causes more correct faults to be detected, than with a lower noise level. But, there are also more spurious faults detected. The peak in the graph shows the point at which the most correct faults are detected, at a noise level of 0.05V. After this point, the spurious faults detected become increasingly greater than the correct faults detected, causing the percentage of correlation to exponentially decay, as expected.

The method shows low correlation values initially, which decrease further with increased noise levels. Thus, the method is seen to be sensitive to noise, as predicted by theory.

5.6.2 Adaptive Parity Space method

The following graph shows the percentage of correlation plotted against noise level, for the adaptive parity space method of fault detection.

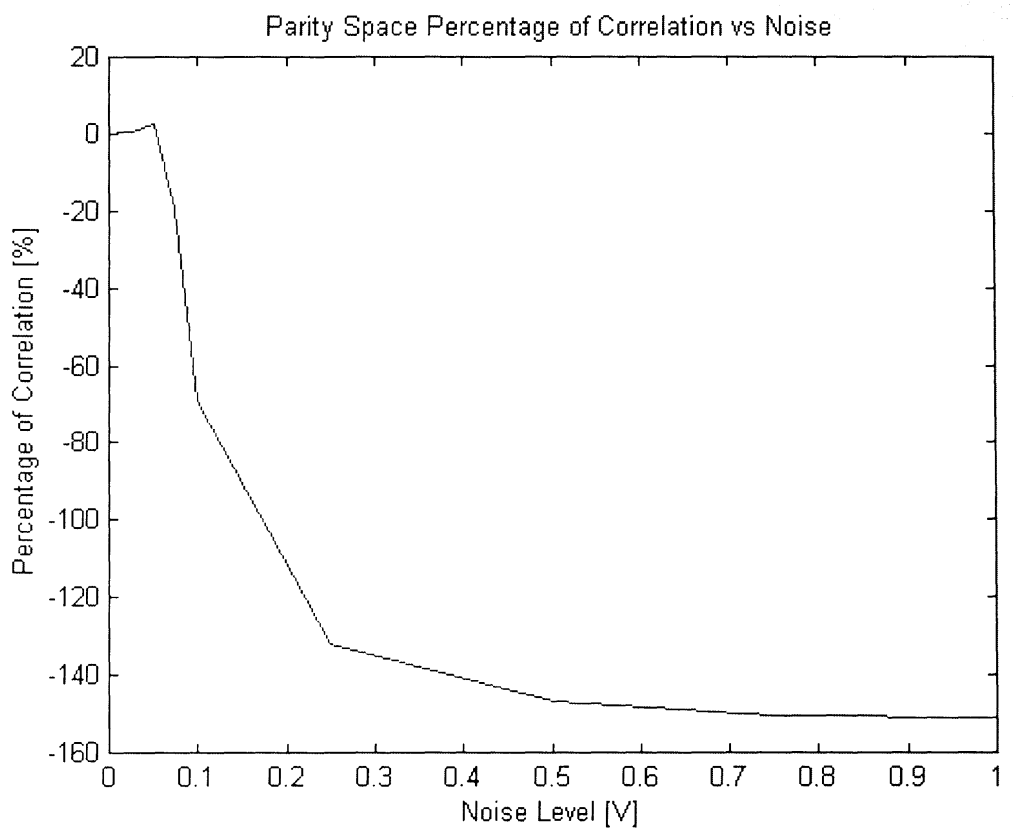


Figure 5-65 Adaptive Parity Space Percentage of Correlation against Noise

Comment:

The adaptive parity space method maintains a fairly high percentage of correlation for noise levels of up to 0.075V, before decaying. This observation is in accordance with the theory predicted. The high values of correlation show that the method is not very sensitive to noise.

5.6.3 Observer method

The following graph shows the percentage of correlation plotted against noise level, for the observer method of fault detection.

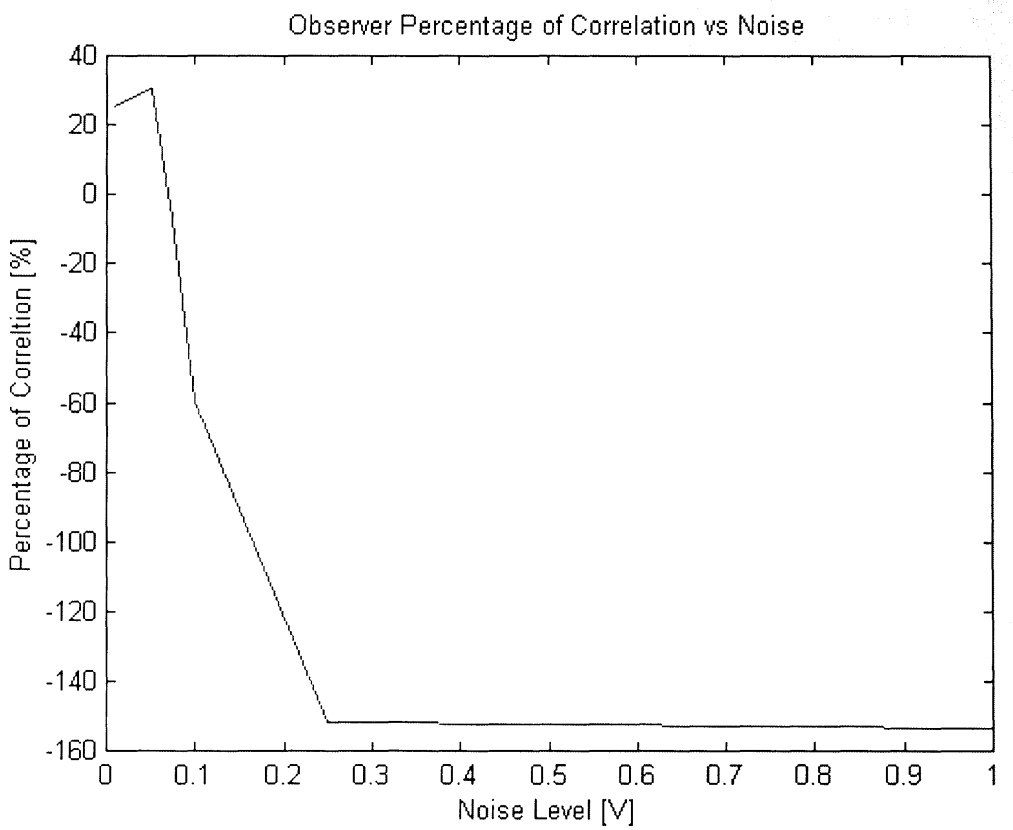


Figure 5-66 Observer Percentage of Correlation against Noise

Comment:

The correlation results seen in the graph, are seen to be similar to that obtained for the parity space method. The explanation for the results is the same as that given for the parity space method.

5.6.4 Comparison of the methods

The adaptive parity space method shows the best correlation results for the noise levels investigated, as expected. The observer method of fault detection is seen to have higher correlation results for all noise levels up to 0.75V, when compared to the parity space method. Since, the percentages of correlation for all the methods are similar at noise levels at

0.5V and greater, the observer method can be said to be marginally less sensitive to noise than the parity space method.

5.7 Discussion on Reliability, Ease of Implementation and Computational Burden

For the fault detection methods investigated in the previous two sections, it was found that the three methods were reliable in the simulations performed. Comparison of the three fault detection methods showed the adaptive parity space method to be the most difficult to implement, as predicted in Section 4.3. The parity space and observer methods were less difficult to implement. It was found that the parity space method was mathematically demanding, while the parameters in the Kalman filter for the observer method were difficult to tune. The adaptive parity space method had the aforementioned difficulties, together with a sensitive parameter estimation scheme.

The computational burden for the methods was determined by approximating the number of *flops* (see Glossary) that each method was required to perform at each sample of the simulation. The number of flops was calculated using Matlab. The greater the number of flops for a fault detection method, the greater the computational burden. The following flops were obtained for each fault detection method:

Parity Space: 2046

Adaptive Parity Space: 2975

Observer: 1098

The results show that the computational burden of the adaptive parity space method is almost twice the burden of both the parity space and observer methods of fault detection.

5.8 Concluding Remarks

The simulation results for the adaptations of the fault detection methods investigated show that the delay in the detection of faults is equal. The following table shows the percentages of correct fault detection, spurious fault detection and correlation obtained for all the adaptations of the fault methods described.

Table 5-1 Summary of the Percentages of Correlation obtained for the investigated Fault Detection Methods

Method	Percentage of Correct Fault Detection [%]	Percentage of Spurious Fault Detection [%]	Percentage of Correlation [%]
Parity Space-single threshold	4.12	0.89	1.32
Parity Space-individual thresholds	4.62	0.48	3.13
Parity Space-individual thresholds (large actuator faults)	8.74	0.74	6.42
Parity Space-Kalman filter	69.63	17.52	14.81
Adaptive Parity Space-LS	66.00	3.07	56.50
Adaptive Parity Space-RLS	9.09	53.66	-76.01
Adaptive Parity Space-RLS with λ	84.48	15.60	36.01
Adaptive Parity Space-RLS with λ and α	86.96	6.21	67.78
Observer-single threshold	52.64	8.86	25.10
Observer-individual thresholds	76.07	19.95	14.03

The table shows that the highest percentage of correlation obtained was for the adaptive parity space method using a RLS estimation with forgetting factor and stabilising factor. The next best results are from the parity space method with Kalman filter, and the observer method using individual thresholds. Comparison of the observer method using individual thresholds with the parity space method using individual thresholds (the method which was further investigated), shows the observer method to have better fault detection performance. The performance of the fault detection methods (as determined by the percentage of correlation) can be considered as a measure of its isolability.

The methods are also seen to detect both multiplicative and additive types of faults. The detection of both actuator and sensor faults is also achieved by the methods, but is dependent on the fault amplitude and duration. The fault detection performance of each method was found to be dependent on fault amplitude, duration, type and noise level. Through the investigation of the varied fault parameters on the fault detection methods, it was found that the parity space method was able to detect faults of small amplitude and duration with more success than the other two fault detection methods. Overall though, the adaptive parity space method showed the best performance with varied fault amplitude and duration. The performance of the adaptive parity space method was not compromised by fault type as the parity space and observer methods (which showed better fault detection performance for abrupt faults than incipient faults). Also, the adaptive parity space method showed the best performance with increased noise levels, followed by the observer and parity space methods.

The adaptive parity space method is the most difficult of the fault detection methods to implement, and has the largest computational burden. The observer and parity space methods show similar ease of implementation and computational burden.

The simulated results of the fault detection methods are now compared to the predictions made by the author in Table 2-1. The following table shows a comparison of the properties predicted and properties obtained from simulation of the fault detection techniques. Process faults, multiple simultaneous faults and vibration faults have not been tested. The robustness of the fault detection methods can only ascertained by online implementation of the methods.

Table 5-2 Comparison of Simulated and Predicted Properties of the investiga

METHOD	Detection of faults related to Actuators	Detection of faults related to Processes	Detection of faults related to Sensors	Detection of all classes of failures	Detection of Multiple Simultaneous faults	Detection of Abrupt faults	Detection of Incipient faults	Detection of Vibration faults	Promptness of detection	Noise Sensitivity	Robust
Predicted Observer	✓	✓	✓	✓	✗	✓	✓	✗	Fast	Low	✗
Simulated Observer	✓	N/A	✓	✓	N/A	✓	✓	N/A	Fast	Average	N/A
Predicted Parity Space	✓	✓	✓	✓	✗	✓	✓	✗	Fast	High	✓
Simulated Parity Space	✓	N/A	✓	✓	N/A	✓	✓	N/A	Fast	High	N/A
Predicted Adaptive Parity Space	✓	✓	✓	✓	✓	✓	✓	✗	Fast	Low	✓
Simulated Adaptive Parity Space	✓	N/A	✓	✓	N/A	✓	✓	N/A	Fast	Low	N/A

N/A – either means does not apply or not tested or cannot be tested.

ulated and Predicted Properties of the investigated Fault Detection Methods

Detection of Vibration faults	Promptness of detection	Noise Sensitivity	Robust	Reliability	Isolability	Requirement of *a priori* Information	Ease of Implementation	Computational Burden	Assumptions & Comments
✗	Fast	Low	✗	✓	Bad – Average	✗	Simple	Low	• Can be insensitive to modelling uncertainty. • Implementation is complex for detecting multiple faults.
N/A	Fast	Average	N/A	✓	Bad – Average	N/A	Average	Average	Kalman filter parameters require optimal tuning
✗	Fast	High	✓	✓	Average	✗	Average	Low	Implementation is complex for detecting multiple faults.
N/A	Fast	High	N/A	✓	Average	N/A	Average	Average	Adaptation with Kalman filter shows promising results
✗	Fast	Low	✓	✓	Good	✗	Complex	Average	
N/A	Fast	Low	N/A	✓	Good	N/A	Complex	Heavy	LS estimation shows better isolability than RLS with and without a forgetting factor

From Table 5-2, it can be seen that most of the properties that the author had compiled for the fault detection methods have been confirmed by the simulation results. The anomalies that arose were fairly insignificant. The anomalies that arose for the parity space method are:

- The promptness of the fault detection was fast and not average as predicted.
- Computational burden was found to be average and not low.

For the adaptive parity space method, the anomalies are:

- The promptness of the fault detection was fast and not slow as predicted.
- Computational burden was heavy and not average.

The anomalies for the observer method of fault detection are:

- Noise sensitivity is average and not low, as predicted.
- Ease of implementation is average and not simple.
- Computational burden is average and not low.

Thus, the fault detection methods are seen to closely match the properties originally predicted.

Chapter 6 CONCLUSIONS

From the findings of this investigation, the following conclusions may be drawn from this project. The author has found it unnecessary to repeat the conclusions based on the literature review. Firstly, conclusions based on the process are discussed. These conclusions are followed by conclusions for the parity space method of fault detection, adaptive parity space method, then the observer method and finally general conclusions on the investigated methods of fault detection.

Process

1. The noise levels measured on the outputs of the process were found to be fairly high, even with low-pass filters on the depth sensor outputs. The effect of the noise and other nonlinearities in the process could not be eliminated. As a result, many step tests were taken to model the process as accurately as possible. However, there remains modelling uncertainty associated with the process model.

2. The process behaves as a low-pass filter that tends to attenuate any disturbances on the input. Hence, the difficult detection of actuator faults by some of the adaptations of the fault detection methods.

Kalman Filter

3. The Kalman filter provided good noise rejection properties on the measured output signals, with the parity space method, as well as providing accurate estimates for the process output. These properties resulted from the derivation of the optimum parameters for the Kalman filter.

Parameter Estimation

4. All adaptations of the least squares estimation schemes produced a bias and initial overshoot on the residuals of the adaptive parity space method. The recursive least

squares estimation showed the worst estimation results, but the introduction of a stabilising and forgetting factor provided significantly better fault detection results. However, the simple least squares estimation provided fairly accurate results. The reason why LS obtained fairly accurate results is because the estimate uses the complete data range to obtain its parameter estimates. The RLS estimations which used a forgetting factor can be said to be almost like a LS estimate since the forgetting factor is low, which implies that old data is forgotten rapidly (the three most recent measurements are used in the estimations). This also implies that the estimation scheme does not need the most recent measurements for good fault detection performance.

Parity Space Fault Detection Method

5. The deviations caused by faults on the parity space residuals do not last through the fault duration. Also, the method is able to detect actuator faults with large fault amplitudes. As a result, the percentage of correct faults detected remains small, which results in poor isolability, even though the amount of spurious faults detected is small.

6. The introduction of a Kalman filter significantly increased the performance of the parity space method of fault detection and provided noise rejection on the process outputs. A contribution to the increased performance was the detection of actuator faults, which were previously left undetected.

7. The parity space method of fault detection with the use of individual thresholds was observed to give only average isolability with large fault duration and amplitude. However, the method gave the best isolability results (compared to the observer and adaptive parity space methods) for faults with low amplitude and duration. The method was also seen to show large sensitivity to noise and detect abrupt faults better than incipient faults. As a result, the method is not a good method of fault detection, even though it has average computational burden and ease of implementation.

Adaptive Parity Space Fault Detection Method

8. The adaptive parity space method of fault detection, with the use of a recursive least squares estimation with forgetting and stabilising factors, showed good isolability,

irrespective of fault amplitude, duration and type. The method was also seen to show average sensitivity to noise. As a result, the method has proved that it is a good fault detection method. However, the method is difficult to implement and has a large computational burden, which could be a deciding factor as to whether to implement this method, in practice.

9. Since the process did not have more sensors, the detection of multiple simultaneous failures was not investigated.

Observer Fault Detection Method

10. The delay in the residual deviations caused the detection of spurious faults in the observer method. As a result, the performance of the method was undermined as it showed good fault detection capability.

11. The observer method of fault detection with the use of individual thresholds was observed to give only bad to average isolability with varied fault duration and amplitude. The method was also seen to show average sensitivity to noise, and detect abrupt faults better than incipient faults. As a result, the method is not a good method of fault detection, even though it has average computational burden and ease of implementation.

Investigated Fault Detection Methods

12. The use of individual thresholds on each of the residuals to indicate faults provides a more accurate method of fault detection, since each residual is unique.

13. The fault detection methods showed reliability and quick detection of faults.

14. The fault detection performance of any method is dependent on the residual threshold setting. The threshold setting has to be sensitive to small faults, yet large enough not to detect noise as spurious faults. So, each threshold setting was carefully chosen above the noise and/or bias of the generated residuals in order to obtain the best possible isolability.

15. The fault detection methods were able to detect all the types of faults that were simulated, such as additive or multiplicative actuator and sensor faults with incipient

or abrupt behaviour. But, the performances of the fault detection methods were observed to be dependent on the fault parameters and the level of noise injected to the process. Thus,

16. While the simulation studies showed that the fault detection methods were able to detect all the simulated faults, the isolation or diagnosis of the faults can be problematic. The residuals showed deviations in the positions of most of the faults, so if all six faults were required to be isolated or diagnosed, the manner in which the fault isolation or diagnosis scheme were to successfully perform, would be impossible. So, for the process investigated, it would be possible to successfully isolate or diagnose one or two faults (dependent on the fault isolation/diagnosis scheme). Also, information from the process should be assumed to be reliable, e.g. from Sensor 2.

17. The best method of fault detection investigated is the adaptive parity space method. It could not be differentiated between the observer or the parity space methods as to which one was the better method. Both have an average ease of implementation and computational burden. This makes them viable methods to use if the process requires these properties, over the better properties that the adaptive parity space method offers. The fact that the observer method has better noise rejection properties might be a deciding factor over the parity space method, which shows better isolabilty to faults with a small duration and amplitude.

Chapter 7 **RECOMMENDATIONS**

From the conclusions, the following recommendations for future research may be made:

1. The fault detection techniques should be implemented online, in order to verify the simulation predictions in practice.
2. A fault diagnosis architecture, such as simple logic rules or fuzzy logic should be implemented, provided that the number of faults to be diagnosed is one or two, as for the process used in this investigation.
3. The parity space method with the use of the Kalman filter should be further investigated, as an alternative adaptive parity space method of fault detection.
4. Research investigating the effects that RLS and its adaptations have on the APS method of fault detection, should be performed.

BIBLIOGRAPHY

Åström, K. J. and K Wittenmark (1989). Adaptive Control. Addison-Wesley Publishing Company. USA.

Basikotis, C., J. Raymond and A. Rault (1979). Parameter identification and discriminant analysis for jet engine mechanical state diagnosis. IEEE Conference on Decision and Control, Fort Lauderdale.

Basseville, M. and I.V. Nikiforov (1993). Detection of abrupt changes: theory and application. Prentice Hall. New Jersey.

Beard, R.V. (1971). Failure accommodation in linear systems through self – reorganisation. Rept. MVT-71-1, Ma Vehicle Lab., Cambridge Mass.

Bodson, M. and J. E. Groszkiewicz (1997). Multivariable and adaptive algorithms for reconfigurable flight control. IEEE Trans. Control Systems Technology, Vol. 5, No. 2, pp 217 – 229.

Bohlin, T. (1994). A case study of grey box identification. Automatica, Vol. 30, pp 307 – 318.

Chipperfield, A. J. and P. J. Fleming (1993). Matlab toolboxes and applications for control. Prentice Hall. New Jersey.

Chow, E.Y. and A.S. Wilsky (1984). Analytical redundancy and the design of robust detection systems. IEEE Trans. Automatic Control, AC-29, pp 603-614.

Chow, M., S.O. Yee and L.S. Taylor (1993). Recognising animal-caused faults in power distribution systems using artificial neural networks. IEEE Trans. On Power Delivery, Vol. 8, No. 3, pp 1268-1274.

Clark, R.N. (1978). Instrument fault detection. IEEE Trans. Aerospace Electron. Syst. Vol. 14, pp 456-465.

Dalton, T., R. Patton and P. Millar (1996). Robust fault diagnosis of a two pump system – a case study. IFAC Symposium SAFEPROCESS '96, paper 7f-031, San Francisco, USA.

Dunia, R., S. Joe Qin, T.F. Edgar and T.J. McAvoy (1991). Sensor fault detection and reconstruction using principal component analysis. IFAC Symposium SAFEPROCESS '96, paper 7f-082, San Francisco, USA.

Duyar, A. and V. Eldem (1991). Fault detection and diagnosis in propulsion systems; a real time identification approach. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 473 - 478.

Duyar, A., V. Eldam, W.C. Merill and T-H Guo (1990). A simplified dynamic model of space shuttle main engine. Submitted to ASME J. of Dynamic Systs. Measurement and Control.

Filbert, D. and K. Metzger (1982). Quality test of systems by parameter estimation. 9th IMEKO-Congress, Berlin..

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy - a survey and some new results. Automatica Volume 26, number 3, pp 459-474.

Geiger, G. (1982). Monitoring of an electrical driven pump using continuous-time parameter estimation methods. 6th IFAC Symposium on Identification and Parameter Estimation, Washington. Pergamon Press, Oxford.

Gertler, J. (1988). Survey of model based failure detection and isolation in complex plants. IEEE control systems magazine.

Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation. IFAC Symposium SAFEPROCESS, Bade-Baden.

Goode, P.V. and M. Chow (1995a). Using a neural/fuzzy system to extract heuristic knowledge of incipient faults in induction motors: part I – methodology. IEEE Trans. On Industrial Electronics, Vol. 42, No. 2, pp 131 – 138.

Goode, P.V. and M. Chow (1995b). Using a neural/fuzzy system to extract heuristic knowledge of incipient faults in induction motors: part II – application. IEEE Trans. On Industrial Electronics, Vol. 42, No. 2, pp 139 – 146.

Gopisetty, S. M. and R. F. Stengel (1998). Detecting and identifying multiple failures in a flight control system. AIAA-98-4488.

Gühmann, C. and D. Filbert (1991). Fault diagnosis of electric low-power motors by analysing the current signal. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 141 - 146.

Haffmann, Ch., H. Holtzmann, M. Ayoubi and R. Isermann (1996). Supervision of vehicles' tyre pressure by measurement of body accelerations. IFAC Symposium SAFEPROCESS '96, paper 8b-013, San Francisco, USA.

Henry, P.M. and D.W. Clarke (1991). A standard interface for self-validating sensors. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 57 – 64.

Hermans, F.J.J. and M.B. Zarrop (1996). Sliding mode observers for robust sensor monitoring. IFAC Symposium SAFEPROCESS '96, paper 7f-066, San Francisco, USA.

Himmelblau, D.M. (1978). Fault detection and diagnosis in chemical and petrochemical processes. Elsevier, Amsterdam.

Höffling, T. and R. Isermann (1996). Adaptive parity equations and advanced parameter estimation for fault detection and diagnosis. IFAC Symposium SAFEPROCESS '96, paper 7f-023, San Francisco, USA.

Höffling, T. and T. Pfeufer (1994). Detection of additive and multiplicative faults – parity space vs. parameter estimation. IFAC Symposium SAFEPROCESS '94, Espoo, Finland

Holzmann, H., Ch. Halfmann, S. Germann, M. Würtenberger and R. Isermann (1996). Longitudinal and lateral control and supervision of autonomous intelligent vehicles. IFAC symposium SAFEPROCESS'96, paper 8b-053, San Francisco.

Isermann, R. and P. Ballé (1996). Trends in application of model based fault detection and diagnosis of technical proceses. IFAC symposium SAFEPROCESS'96, paper 7f-011, San Francisco.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods: a survey. Automatica , Volume 12, pp 387-404.

Isermann, R. (1989). Process fault diagnosis based on process model knowledge. IFAC Symposium on Advanced Information Processing in Automatic Control "AIPAC", Nancy, France.

Isermann, R. (1991). Fault diagnosis of machines via parameter estimation and knowledge processing. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 43 - 55.

Isermann, R., K.-H Lachmann and D. Matko (1992). Adaptive control systems. Prentice Hall (Series in Systems and Control Engineering), London.

Jones, H.L. (1973). Failure detection in linear systems. Ph.D.Thesis Dept. of Aeronautics, MIT, Cambridge, Mass.

Kavuri, S.N. and V. Venkatasubramanian (1993). Using fuzzy clustering with ellipsoidal units in neural networks for robust fault classification. Computers chem. Engng, Vol. 17, No. 8, pp 765 – 784.

Maki, Y. and K. A. Loparo (1997). A neural network approach to fault detection and diagnosis in industrial processes. IEEE Trans. On Control Syst. Tech., Vol. 5, pp 529-541.

Negoita, C.V. (1985). Expert system and fuzzy systems. Cliff Benjamin/ Cummings, Menlo Park.

Neuman, D. (1991). Fault diagnosis of machine-tools by estimation of signal spectra. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 147 - 152.

Papoulis, A. (1984). Probability, random variables and stochastic processes, McGraw-Hill.

Patton, R., P. Frank and R. Clark (1989). Fault diagnosis in dynamic systems. Theory and application. Prentice Hall, Hertfordshire.

Patton, R.J. and M. Hou (1996). A matrix pencil approach to fault detection and isolation observers. IFAC Symposium SAFEPROCESS '96, paper 7f-073, San Francisco, USA.

Patton, R.J. and J. Chen (1991). A review of parity space approaches to fault diagnosis. IFAC Symposium SAFEPROCESS, Baden-Baden.

Pau, L.F. (1981). Failure diagnosis and performance monitoring. Marcel Dekker, New York.

Pfeufer, T. and R. Isermann (1996). Intelligent electromechanical servosystem. IFAC Symposium SAFEPROCESS '96, paper 9d-031, San Francisco, USA.

Rault, A., A. Richalet, A. Barbot and J.P. Sergenton (1971). Identification and modelling of a jet engine. IFAC Symposium on Digital Simulation of Continuous Processes, Gejör.

Reiß, T. (1991). Model based fault diagnosis and supervision of the drilling process. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 211 - 216.

Reiß, T., P. Wanke and R. Isermann (1990). Model based fault diagnosis of a flexible milling centre. 11th IFAC World Congress Vol.13, paper 17, Tallinn, USSR.

Rückwald, R. (1991). A model-based fault detection concept for mechanical systems. IFAC-Symposium SAFEPROCESS, Baden-Baden, Germany, pp 205 - 210.

Van Dooren, P. (1979). The computation of Kronecker's canonical form of a singular pencil. Linear Algebra Appl. Vol.27, pp 103 – 140.

Wang, H. and S. Daley (1996). Fault diagnosis using adaptive observer based techniques. IFAC Symposium SAFEPROCESS '96, paper 7f-022, San Francisco, USA.

Xu, W., D. Wang, Z. Zhou and H. Chen (1997). Fault diagnosis of power transformers: apllication of fuzzy set theory, expert systems and artificial neural networks. IEE Proc.-Sci. Meas. Technol., Vol. 144, No. 1, pp 39 – 44.

Zadeh, L.A. (1965). Fuzzy sets. Inf. Control, Vol. 8, pp 383 – 353.

Zadeh, L.A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets Syst., pp 199 – 227.

Fault isolation: Determination of kind, location and time of detection of a fault. Follows fault detection.

Fault identification: Determination of the size and timevariant behaviour of a fault. Follows fault isolation.

Monitoring: A continuous real time task of determining the conditions of a physical system, by recording information recognising and indicating anomalies of the behaviour.

Protection: Means by which a potentially dangerous behaviour of the system is suppressed if possible, or means by which the consequences of a dangerous behaviour are avoided.

Fault Detection and Isolation systems can be classified as either active or passive.

Active approach: Residuals are generated to have minimum sensitivity to modelling uncertainties and maximum sensitivity to faults.

Passive approach: Concentrates on the stage of residual generation, where the residuals are analysed for diagnosis.

3.) Models

Analytical redundancy:

Use of two or more, but not necessarily identical ways to determine a variable where one way uses a mathematical process model in analytical form.

Diagnostic model: A set of static or dynamic relations which link specific input variables – the symptoms – to specific output models – the faults.

Qualitative model: Use of static and dynamic relations among system variables and parameters in order to describe systems behaviour in qualitative terms such as causality or if-then rules.

Quantitative model: Use of static and dynamic relations among system variables and parameters in order to describe systems behaviour in quantitative mathematical terms.

Flops: Floating point operation count that returns the cumulative number of floating point operations. It is not feasible to count absolutely all floating point operations, but most of the important ones are counted. Additions and subtractions are one flop if real and two if complex. Multiplications and divisions count one flop each if the result is real and six flops if it is not. Elementary functions count one if real and more if complex. (Matlab definition)

APPENDIX A: SUMMARY OF PUBLISHED FAULT DETECTION & DIAGNOSIS PUBLICATIONS

This appendix has been adapted from the paper published by Isermann and Ballé in 1996. A summary of fault detection and diagnosis publications follows.

SUMMARY OF PUBLISHED APPLICATIONS 1991-1995

Because of the many publications and increasing number of applications it should be of interest to summarize some results. Therefore one of the authors, Peter Ballé, has performed a literature study of Conference Preprints. Following Conferences were taken into account: :

(Bad 91) : IFAC Symposium on fault detection, Supervision and safety for technical processes -Safeprocess '91-, Sep 1991, Baden Baden, Germany
(De 92) : IFAC Symposium on artificial intelligence and real time control, June 1992, Delft, Netherlands
(Syd 93) : 12th IFAC World Congress, July 1993, Sydney, Australia
(TD 93) : International Conference on fault diagnosis - Touldiag 93 -, April 1993, Toulouse, France
(Hel 94) : IFAC Symposium on fault detection, supervision and safety for technical processes -Safeprocess '94-, June 1994, Espoo, Finland
(Sysid 94) : 10th IFAC Symposium on system identification, 6.1994, Kopenhagen, Denmark
(NC 95) : On-line fault detection and supervision in the chemical process industries, June 1995, Newcastle, England
(SF 96) : 13th IFAC World Congress, June 1996, San Francisco, USA

Only those contributions were considered which take into account applications with

1.	Simulation of real processes	(55)
2.	Pilot processes (large-scale)	(44)
3.	Laboratory processes (small-scale)	(18)
4.	Industrial processes (full scale)	(48)

The numbers in brackets indicate the number of publications. Altogether 165 publications were taken into account.

The type of considered faults are distinguished by

s:	sensor fault	(69)
a:	actuator fault	(51)
p:	process fault	(83)
c:	control loop or controller fault	(8)

The fault detection methods were classified according to

ob:	observer	(53)
ps:	parity space	(14)
pe:	parameter estimation	(51)
fs:	frequency spectral analysis	(7)
nn:	neural networks	(9)

For change detection and fault classification methods are indicated by

nn:	neural nets	(19)
fl:	fuzzy logic	(5)
bc:	Bayes classification	(4)
ht:	hypothesis testing	(8)

The reasoning strategies for fault diagnosis are

rb:	rule based	(10)
sdg:	sign directed graph	(3)
fst:	fault symptom tree	(2)
fl:	fuzzy logic	(6)

In the appendix the considered references are shown in Table 1 for fault detection and isolation and in Table 2 for fault detection and diagnosis.

Some results of the literature study are presented in the following figures. The evaluation has been limited to laboratory- pilot- and industrial processes (110 publications), i.e. simulations are not taken into account.

Methods: **ob**: observer, **pe**: parameter estimation, **ps**: parity space, **nn**: neural nets, **fs**: frequency spectra analysis, **fl** : fuzzy logic, **ht**: hypothesis testing
Type: **s**: sensorfault, **a**: actuator fault, **p**: process fault, **c**: control loop fault, **ad**: additive fault, **mp**: multiplicative fault
Process: **1**- simulation, **2**- pilot process, **3**-labor process, **4**-industrial process
classification: **nn**: neurol nets, **fl**: fuzzy-logic, **bc**: bayes classifier, **x**: others
Reasoning: **rb**: rule based, **sdg**: signed directed graph, **fst**: fault-symptom- trees, **ht**: hypothesis testing

Table 1 Fault detection and isolation (1991- 1996)

	Process	Num. of faults	Type	Methods	Comments	Author
	drilling machine (2)	1	p	fs, nn	neural network patterns of vibrational signals	Akhmetov (NC 95)
	drilling process(4)	1	p	pe	reduced linear model, use of current and feed rate signals	Reiß (Bad 91)
	grinding process (4)	2	p	pe	combination of process- and signalmodel approach	Fuchs (Bad 91)
	machine tool drive (4)	4	p		analysis of motorcurrent	Martin (Bad 91)

Table 2 Fault detection and diagnosis (1991- 1996)

		Process	Num. of faults	Type	Methods for detection	Methods for diagnosis: classification	Methods for diagnosis: reasoning	Comments	Author
Mechanical engineering	Thermal proc.	power plant feed water process (1)	6	a				diagnostic algorithms based on models describing characteristics of monitored components as function of time and operating points.	Lautala (Syd 93)
		power plant boiler unit (1)	4	a, mp	ob		rb	kalmanfilter, heuristic knowledge base for diagnosis	Korbicz (He 94)
		air to water heat exchanger (2)	> 4	s,a,p	ps		rb	parity space and static characteristics, statistic residual evaluation and logic base reasoning	Millar (Nc 95)
	Fluiddynamic proc.	pipeline system (2)	2	p			sdg	directed graph modeling	Shibata (Bad 91)
		air compressure system (3)	2	a			rb		Juuma (Bad 91)
		3 tank-system (3)	5	s,a			x	predicate formulae description of the process	Lunze (De 92)
		3 tank- system (3)	8	a,p			rb	knowledge based causal structure	Lunze (Syd 93)
		tanksystem (1)	4	p,c			rb	qualitative / heuristic model and causal reasoning	Lunze (TD 93)
	Nuc. proc.	nuc. reactor coolant pump (1)	1				fl	Fuzzy decision making process	
		nuc. reactor coolant pump (1)	30	p,s,a,c			rb	heuristic and causal fault model and reasoning	Porcheron (Td 93)
		nuc. press. water reactor (1)	> 4	s	ob	nn		neural net for threshold gen. and residual classification.	Hwang (Syd 93)
	Combustion engines and turbines	turbofan engine (2)	1	a	pe	nn		quasi linear state space representation	Merril (Bad 91)
		ship diesel engine (2)	24	a		fl		fuzzy logic rule based system	Heude (Td 93)
		jet engine (2)	> 1	s		fl		knowledge based method with fuzzy rules	Lunderstead (Syd 93)
		gas turbine (4)	4				rb	rule based expert system	Milne (Td 93)
		combustion engine (1)	2	c,a		bc		qualitative reasoning with Bayesian net	Kirsch (Td 93)
		turbocharger wastegate (4)	1	a	nn	nn		dynamic neural nets for nominal and fault model	Ayoubi (Hel 94)
		gas turbine (2)	> 1	s		fl		model-based fuzzy classification	Hilleman (Hel 94)
		diesel engine (4)	3	a		nn			Ludwig (Hel 94)
		4 cylinder comb. engine (4)	4	a,s		x			Danai (SF 96)
	Misc.	industrial robot (4)	7	p	pe		fst	using detailed model, statistical evaluation, fault symptom trees, process history	Freyermuth (Bad 91)
		industrial robot (4)	5	s,p			rb	nonlinear model, supervision in time, frequency and parameter domain, knowledge base for diagnosis	Siedler (Hel 94)
		electr. driven throttle valve (4)	12	p,ad, mp	pe	fl-nn		diagnosis of pe symptoms with neuro-fuzzy network	Pfeufer (GI 95)
		electric drive (2)	3	s,c,			sdg	model-based direct graph analysis	Berendsen (Hel 94)
		DC motor (4)	12	p	pe		fl	possibilistic fault model, fuzzy fault trees	Ulieru (95)
		perm. excited DC motor (4)	7	p,s	ps,pe		fl		Höfling (96)
		turbo generator (4)	3	s,p	pe		fl	Hotellings and Scheffees test for fault detection, MISO-ARX modells	Gomez (SF 96)
		power line feeders (2)	1			x		classification of current and voltage patterns	
Chemical engineering		jacketed reactor (1)	2	p	nn	bc		dynamic RBF netwoks as nonlinear predictor bank for faulty/nonfaulty case, Bayes classifier	Sorsa (Td 93)
		bioreactor	>2	s,a			rb	qualitative reasoning, knowledge based diagnosis	Halme (Hel 94)
		stirred tank reactor temperature and level control (1)	5	p		nn	fl, rb	fuzzy-neural network, genetic algorithm, knowledge based and qualitative simulation based approaches	Zhang (Nc 95)
Misc.		aircraft model (1)			nn	bc		bank of feedforeward nn and a multiple model adaptive estimator	Zhang (SF 96)

Comb. Engines	automotive engine exhaust emmission control system (4)	6	s,a	ps	mix of linear dynamic and nonlinear static models	Rizzoni (Td 93)
	variable thrust rocket engine (2)	4	p	nn	Hopfield NN for simulating normal and anomalous conditions	Feng (Td 93)
	gasturbine combustion chamber (1)	1	s	pe	asymptotic local approach for change detection, linearized model	Basseville (TD 93)
	combustion engines (4)	1	a		Levenstein distance and nearest neighbour classification of angular acceleration of crankshaft	Ruckhäberle (Td 93)
	combustion engine (4)	1	p	fl	fuzzy classification of patterns derived by analysis of engine flywheel speed fluctuations	Deng (Hel 94)
	automobile combustion engine (4)	4	a,s	ps	nonlinear parity equation	Rizzoni (Hel 94)
Automotive	automotive power train (2)	3	s	ob	nonlinear state equations and a bank of kalmanfilters	Paolella (Bad 91)
	hydraulic power steering (2)	4	a	pe		Appel (Bad 91)
	vehicle model (1)	2	p	ob, pe, ps	combination of parameter estimation and observer	Rückwald (Bad 91)
	rear axle gearbox (2)	4	p	fs,nn	MLP net, classification of statistical and frequency domain features	Barschdorff (Td 93)
	car break system (1)	1	p	pe	parameter identification and bond-graph modelling	Vergé (Hel 94)
Inverted pendulum	inverted pendulum (3)	3	s, ad, mp	pe		Sproesser (Td 93)
	inverted pendulum (3)	1	s	ob		Hassenforder (Td 93)
	inverted pendulum (3)	3	s, ad	ob, pe		Nowakowski (Td 93)
	inverted pendulum (3)	4	s,a	ob	linear model	Bousghiri (Td 93)
	inverted pendulum (3)	3	s	pe	linear model, evaluation of fault signatures	Sproesser (Td 93)
	inverted pendulum (3)	3	s	ob	bank of observers, linearized model around set point	Köppen (Td 93)
	inverted pendulum (3)	3	s	ps		Frank (Td 93)
	inverted pendulum (3)	4	s,a, ad, mp	ob, pe	bank of unknown input observers	Köppen Seliger (Hel 94)
	inverted pendulum (3)	3	s	ob	adaptive state observer, estimation of friction parameters	Sauter (Hel 94)
	inverted pendulum (3)	3	s	ob	exponential observers	Nowakowski (Hel 94)
	inverted pendulum (3)	3	s,	ob		Patton (Hel 94)
Misc.	coal milling process (4)	2	p,s	pe		Ferretti (Bad 91)
	overhead crane (3)	2	s,p	ob	nonlinear unknown input (loadmass) observers	Frank (Syd 94)
	bearings (2)	1	p	fs, nn	kohonen networks for vibration spectra patterns	Gong (SF 96)
	crane (1)	1	p, mp	pe	missing data RLSD algorithm	Albertos (SF 96)
	industrial robot (2)	2	p	ob	nonlin. treated as known ob inputs, adaptive thresholds	Schneider (Td 93)
	paper machine (2)	1	p	nn	kohonen feature maps	Sorsa (NA 92)
	paper mill (4)	1	a		detec. of oscill. (stick slip motions) in the control loop	Hägglund (Hel 94)
DC Motor	DC motor (4)	3	p	pe	mathematical model and wald sequential test for fault detection	Coirault (Bad 91)
	DC motor (4)	4	p	pe		Filbert (Bad 91)
	DC motors (4)	4	p	pe	current signal analysis (covariance , autocorrelation...), ht for classification	Gühmann (Bad 91)
	DC mot. feed drive of a mach. tool (4)	7	p	pe	statistical decision tests	Isermann (Bad 91)
	DC mot. main dr. of a mill. cent. (4)	3	p	pe	reduced linear model	Wanke (Bad 91)
	DC motor feed drives of a milling center (4)	5	p	pe	reduced linear model	Wanke (Bad 91)
	DC motor (1)	5	p	pe, fl	fuzzy logic classification of estimated parameters	Marcu (Bad 91)
	DC motor (4)	2	p		classification of measurementes for voltage, current, vibration, rotation speed	Barschdorff (Bad 91)
	DC motor (4)	2	p	pe	wald and page hinkley test for isolation	Coirault (Td 93)
	permanently excited DC motor (4)	7	p,s	ps		Höfling (Syd 93)
	PI controlled DC servo motor (1)	6	p,cmp	pe	(15% -25%) size of faults	Fang Jia (Td 93)
	PI controlled DC servo motor (1)	6	p,c,mp	ob	dedicated and unknown input ob (20-30, 20-60%) size	Fang, Jia (Td 93)
	DC motor PI controlled (2)	3	p,c	ob		Berendsen (Td 93)
	DC motor (4)	5	p	ob ,ps, pe (not combined)	Bayes classifier of symptoms	Filbert (Hel 94)

Mechanical engineering	Mill. A. grind. proc.	cutting process (4)	1	p	pe		Schulz (Bad 91)
		turning, grinding processes (4)	1	p	fs, nn	backprop., nearest neighb. nn. signals: force and vibration	Barschdorff (Td 93)
		rotating machines (1)	4	p	fs, nn	classification of frequency spectrum features with MLP	Yan Tinguh (Td 93)
		turning machines (2)	2	p		classification of multi-sensor signals	Ravindra (Td 93)
		cnc machines (4)	10	a,p	pe	off line, models for faulty and faultfree case	Vergé (Td 93)
		milling machine (4)	2	p	fs, fl	vibration signal and fuzzy pattern recognition	Zieba (Hel 94)
		milling machine (4)	3	p	pe,nn	nearest neighbour, threshold and neural nets classification of estimated parameters	Konrad (1995)
		large rotating machines (4)	2	p	fs, nn	probabilistic neural net	Xu (NC 95)
	Power plants and thermal processes	heat exchanger (1)	8	s	nn	MLP with sigmoidal activation function	Himmelblau (Bad 91)
		heat exchanger (4)	> 2	s	ob	measured state var., linearization around nom. op. point	Cecchin (Bad 91)
		steam generator (4)	8	s, mp, ad	ob, pe	kalmanfilter	Wernz (Bad 91)
		thermal plant feed water heating (1)	1	p	pe		Kumamaru (Td 93)
		thermal plant (2)	>2	p	ps	linearized model in a nominal working point	Noura (Td 93)
		liquid -liquid extractor (1)	>7	p,s,a,ad	ob	single and multiple faults	Millar (Td 93)
		pressure vessel of a nuclear reactor (2)	1	p	nn, fl	fuzzy, KNN classification of accoustic sound patterns	Weiss (Hel 94)
		heater of feedwater high pressure line of a power plant (1)	>3	s,p,a,ad	nn	RBF as receptive field classifier and MLP as hyperplane classifier	Guglielmi (Hel 94)
		heat exchanger battery (4)	>4	s	ps	static bilinear model	Cassar (Hel 94)
		heat exchanger (4)	>1	s	ob,ps		Simonart (Hel 94)
		heat exchanger (1)	1	p		long term wear models and generalised fourier pole estimation of a second order transfer function	Neuenschwader (Hel 94)
		steam gen. of a nuclear power plant (4)	>4	s, ad	pe	linearized model,	Dorr (Hel 94)
		thermal plant (4)	8	s,p,a,ad, mp	ps, pe	combined, linearized model around the setpoint	Höfling (Nc 95)
		thermal plant (4)	3	s.a, ad	nn	perceptron with locally distributed dynamics	Ayoubi (NO 95)
		temperature control loop (2)	3	a,s	ob	state space description obtained from bond models	Blanke (NC 95)
		heaters of a feedwater high pressure line (4)	3	a,p, s, ad	nn	multilayer feedforward nn and stochastic approximation, nn and mathematical modelling for distinct subprocesses	Alessandri (SF 96)
		boiler unit (1)	1	p	pe	linearized model	Kwon (SF 96)
		thermodynamic system (2)	>2	s, ad, mp	ob	low dimensional linear model, sliding mode observer	Hermans (SF 96)
	Fluiddynamic processes	tank system with level control (1)	3	s,a,p,mp	ob	Luenberger observer	Deibert (De 92)
		hydraulic pump (1)	1	p		frequency domain , pulsative model	Wang (TD 93)
		3 tank mixing process (3)	>3	s,a,p	pe, ht	linearized model, knowledge based evaluation, hypothesis testing	Jaume (Hel 94)
		pressure system (2)	5	p,a	nn	MLP net	Juuma (Hel 94)
		industrial pump system (1)	6	a,p	ob,pe	observer-, parameter estimation and combination of observer and parameter estimation	Dalton (Nc 95)
		valves in flow control loops (2)	3	a	ob	observer based techniques in addition of signal analysis of control loop signals	Deibert (Hel 94)
		3 tank system (3)	> 3	s,p	ob	linearized model, state and input estimation	Bousghiri (Nc 95)
		3 tank system (1)	2	a,p		steady state model, fault trees	Juricic (SF 96)
		3 tank system (3)	> 3	s, mp, p	pe	slow acting faults, linearization around setpoint, Kofahl test for detection (no isolation possible)	Zolghardi (SF 96)
		jet engine (1)	3	s	ob	unknown input ob in combination with pe	Zhang (Bad 91)
		propulsion system (2)	> 3	s,a, mp,ad	pe		Duyar (Bad 91)
		propulsion system (4)	1	p	ob/pe	nonlin. parameter estimation with extended Kalmanfilter	Bruce (Bad 91)
		gas turbines (4)	>1	s	ob	gas path analysis	Lunderstaedt (Bad 91)
		jet engine (1)		s	ob	unknown input observer	Patton (Bad 91)
		gas turbine engine (2)	1	p	pe	model of transfer function fuel flow to outlet temp.	Villanueva (Bad 91)
		jet engine (1)	> 3	s	ob	output observer, estimation of residual bias due to modeling errors	Patton (De 92)
		rotary turbo machines (1)	4	a		classification of vibration signals	Menard (Td 93)
		automotive engines (1)	5	s,a	ps	linear dynamic and nonlinear static hybrid model	Gertler (Syd 93)

Electrical engineering	DC Motor	DC motor (4)	2	p		classific. of frequency features and statistical moments	Filbert (Hel 94)
		DC motor pump pipe process (1)	5		pe	linear classifier of least squares minimum distance type	Marcu (Hel 94)
		DC servomotor actuator (1)	2	s,a	ob, ht	bank of observers, multi hypothesis testing	Nielsen (Hel 94)
		DC servomotor actuator (1)	2	s,a	ob	eigenstructure assignment	Jorgensen (Hel 94)
		DC servomotor actuator (1)	2	s,a	ob	extended nonlinear kalmanfilter and fixed thresholds	Walker (Hel 94)
		DC servomotor actuator (1)	2	s,a	ob	output observer, fixed thresholds,	Garcia (Hel 94)
		DC servomotor actuator (1)	2	s,a	ob	adaptive threshold	Höfling (Hel 94)
		DC motor (4)	4	s,p,ad mp	ps,pe	adaptive threshold	Höfling (Hel 94)
		DC motor (1)	1	a	ob	nonlinear unknown input observer	Yang (SF 96)
		DC motor (1)	2	p	pe	two sliding time windows with different lengths	Balestrina (SF 96)
	Misc.	4 quadrant chopper (2)	2		ob		Catellani (Bad 91)
		electric power systems (2)	1	a	ob	kalmanfilter and fixed point linearization	Kliokys (Bad 91)
		turbo generator set (4)	?		pe	approximation of parameters of a discrete, linear model	Rachev (Syd 93)
		electrohydraulic drive (4)	?		ob	bilinear modelling for nonlinear observer design	Schwarz (Syd 93)
		electrohydraulic rotary system (1)	2	a mp	nn	MLP for the healthy system modelling, knowledge about effects of known faults is used for detection	Daley (Hel 94)
Chemical engineering	Stirred tank reactor	exothermic reactio in jacketed reactor (1)	6	p	nn	temperature, outlet concentration control loops, feedforeward ANN for steady state and dynamic cases	Venkatasubrama nian (Bad 91)
		stirred tank with heat exchanger (1)	9	a,p	nn	level, flowrate and temperature control loops, MLP, counterpropagation, nearest neighbour classification	Sorsa (Bad 91)
		nitric acid production process (4)	> 2	s	ob	simpified mathematical model of subprocesses	Schlosser (Bad 91)
		stirred tank with heat exchanger (1)	9	a,p	nn	level, flowrate and temperature control loops, direct classification of 14 measurements with MLP, RBF and Kohonen feature maps	Sorsa (93)
		stirred chemical ractor with heat exchanger (1)	3	s, ad	ob	nonliner state observer filter (particle filter)	Viel (Td 93)
		stirred tank reactor with second order exothermic reaction (1)	5	s,p,a, mp	pe	level and temperature control loops general discrete time model	Gomm (Hel 94)
		stirred tank reactor with second order exothermic reaction (1)	4	s,a ad	nn	level and temperature control loops, RBF network with automatically added nodes for new faults	Gomm (NC 95)
		batch polymerisation process (1)	2	p		multivariate statistical analysis, likelihood based confidence region	Martin (NC 95)
		low density polyethylene tubular reactor (2)	2	p		multivariate statistical analysis, likelihood based confidence region	Martin (NC 95)
		polimerization reactor (1)	3	s,a	nn	auto associative neural network	Fujiwara (NC 95)
		batch process 2 buffer and 2 reaction tanks (2)	2	a	ob, nn	nonlinear state observers, cohonen vector quantisation classification	Medl (NC 95)
		high cell-density fed-batch ferment. (2)	2	s,c		dynamic time warping, pattern recognition	Gollmer (NC 95)
		polimerization reactor (2)	2	p	nn	nn as nonlinear model and inverse model of the process	Zhang (SF 96)
		polimerization reactor (2)	3	p	nn	autoassociative neural network;	Zhang (SF 96)
		stirred tank reactor (2)	1	p,mp	nn, ob	nonlinear observers and reinforcement learning for nn	Pany (SF 96)
	Misc.	exothermic batch process (4)	1	p	ob	ext. kalmanfilter, models for faulty and faultfree process	Perne (De 92)
		wastewater plant (1)	2	p	pe	detection of large changes	Fuente (Hel 94)
		catalytic cracking process (4)	7	a,p	fl-nn	fuzzy-neural network for fault models	Yang (NC 95)
		DOx sensor (2)	1			physical model of the diffusion process	Clarke (Nc 95)
Misc.		off shore drilling (4)	1		ob, pe		Bugge (Bad 91)
		airplane (4)	3	s	ob	nonlinear observers	Dunkel (Bad 91)
		navigation system (4)	2		ob, pe	kalmanfilter	Nikiforov (Bad 91)
		track guided bus (2)	2	s	ob	robust state observers	Schrick (Td 93)
		building heating (1)	9	p,s,a,	ob, pe	combined ob and pe	Georgescu (Td 93)
		aircraft (1)	3	s	ob	unknown input observer schemes	Strichander (Syd 93)
		navigation system (4)	2	s,ad		change detection in stochastic processes	Nikiforov (Td 93)
		heating system of large buildings (4)	5		fl	fuzzy model-based method with models for faulty and faultfree process	Benouarets (Nc 95)
		underwater vehicle (1)	>1	p	ps	parity equations for nonlinear dynamic systems	Shumsky (SF 96)

APPENDIX B: DETAILED PROCESS DESCRIPTION

B.1 System specifications and operation

The experimental system used in this study was designed by TecQuipment. The system consists of a glass tank of length 20cm, width 10cm and height 30cm. A centre partition is placed in between the tanks with a hole, with a diameter of 0.635cm, situated near the bottom of the partition (so as to create a water flow between the two tanks). An electric motor is used to drive a variable speed pump, which pumps water from a bath into Tank 1 (as shown in the following diagram). The pump motor can be driven by a voltage of between 0 and 10V applied to the pump drive socket. A flow meter measures the water flow rate into the tank. Depth sensors are placed in each tank to determine the height of the water levels in the tanks. The depth sensors produce an output voltage proportional to the height of water in the tank, in the range of ±10V. An adjustable tap is attached to Tank 2 to allow the water flow out. The signals from the depth sensors are passed through amplifiers, which have adjustable gains. The amplified signals are then passed into low-pass filters, which are used to remove process disturbances and noise. The specifications of the system are:

Voltage: 240V

Current: 1A

Frequency: 50 – 60Hz

Phase: 1

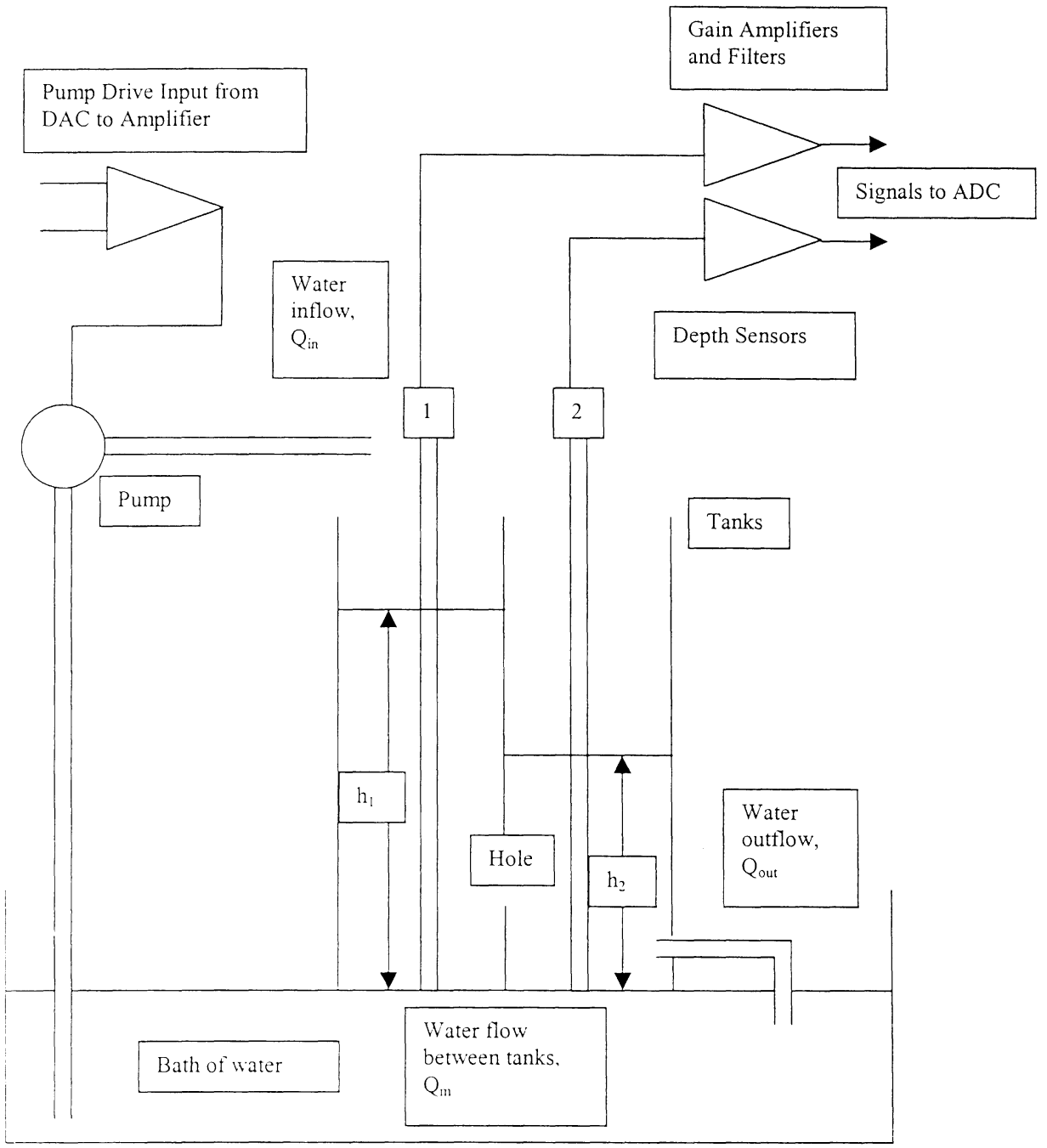


Figure B-1 Schematic View of the Pump System

B.2 Mathematical model of the system

In order to obtain the mathematical model of the system, the input and output of the system need to be determined. The input is the water flow rate into Tank 1, and the output is the height of water in Tank 2 (which the depth sensor converts into a voltage).

The mathematical model of the system is derived from a volume balance analysis of the process:

Water Inflow Rate:	Q_{in}
Water Outflow Rate:	Q_{out}
Water Flow Rate between Tanks:	Q_m
Height of Water Level in Tank 1:	h_1
Height of Water Level in Tank 2:	h_2
Volume of Water in Tank 1:	V_1
Volume of Water in Tank 2:	V_2
Cross-sectional Area of Tank 1 and 2:	A
Gain constants:	k_1, k_2

Assuming steady-state conditions,

For Tank 1:

Flow Rate = Q_{in} - Q_m

$$= \frac{\partial V_1}{\partial t} = A\frac{\partial h_1}{\partial t}$$

For Tank 2:

Flow Rate = Q_m - Q_{out}

$$= \frac{\partial V_2}{\partial t} = A\frac{\partial h_2}{\partial t}$$

And

$Q_{out} = k_2 . h_2$

$Q_m = k_1 (h_1 - h_2)$

Substituting in the flow rate equations for Tanks 1 & 2:

$$Q_{in} - k_1(h_1 - h_2) = A\frac{\partial h_1}{\partial t}$$

$$k_1(h_1 - h_2) - k_2 h_2 = A\frac{\partial h_2}{\partial t}$$

This results in the following state space model:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} \frac{-k_1}{A} & \frac{k_1}{A} \\ \frac{k_1}{A} & -\frac{k_1 + k_2}{A} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A} \\ 0 \end{bmatrix} \cdot Q_{in}$$

Taking the Laplace transform gives:

$$\frac{h_2(s)}{Q_{in}(s)} = \frac{1/k_2}{(1 + sT_1).(1 + sT_2)} \qquad \frac{[V]}{[cc]}$$

and

$$\frac{h_1(s)}{Q_{in}(s)} = \frac{\frac{k_1 + k_2}{k_1 k_2} + \frac{A}{k_1 k_2}s}{(1+sT_1)\cdot(1+sT_2)} \qquad \frac{[V]}{[cc]}$$

where

$T_1 T_2 = \frac{A^2}{k_1 k_2}$ and $T_1 + T_2 = \frac{A(2k_1 + k_2)}{k_1 k_2}$ define the process time constants implicitly.

APPENDIX C: PROCESS CHARACTERISTICS

C.1 Pump Characteristic

In order to calibrate the pump, the zero setting of the pump drive input of the system was set in such a manner that when 0Volt was fed to it, the pump was on the verge of pumping water into Tank 1. The calibration was performed by applying voltages to the pump drive input socket (within the specified voltage range of the pump). The corresponding flow rates from the flow meter were recorded. The following graph shows the pump's characteristic, with a Least Squares (LS) fit to the data (Note: the flow rate values after 4V were not used to determine the LS fit since the maximum flow rate had been reached when the input voltage was 4V).

The following table shows the data obtained for the pump's characteristic. The table also shows the LS values used to fit the data and the squared error values between the characteristic and the LS fit. The "LINEST" function in Microsoft EXCEL was used to obtain the gradient, m and point of intersection, b of the LS linear equation (Note: the flow rate values after 4V were not used to determine the LS fit since the maximum flow rate had been reached when the input voltage was 4V).

Table C-1 Pump Characteristic with LS Data and Error

	Voltage (V)	Flow Rate [cc/min]	LS Data	Squared Error
	0	1.0	3.5	6.25
	0.2	3.0	4.9	3.69
	0.4	5.0	6.3	1.81
	0.6	7.0	7.8	0.59
	0.8	9.5	9.2	0.10
	1.0	11.5	10.6	0.79
	1.2	13.0	12.0	0.94
	1.4	15.0	13.5	2.39
	1.6	16.5	14.9	2.64
	1.8	18.0	16.3	2.89
	2.0	19.0	17.7	1.64
	2.2	20.5	19.1	1.84
	2.4	21.5	20.6	0.87
	2.6	23.0	22.0	1.03
	2.8	24.0	23.4	0.35
	3.0	25.0	24.8	0.03
	3.2	26.0	26.3	0.06
	3.4	27.0	27.7	0.46
	3.6	28.0	29.1	1.20
	3.8	29.0	30.5	2.31
	4.0	30.0	31.9	3.77
	4.2	30.0	33.4	11.31
	4.4	30.0	34.8	22.90
	4.6	30.0	36.2	38.54
Mean Squared Error				1.70

From the table, The initial flow rate for a 1V step test is 11.5 [cc/min], and final flow rate is 19.0 [cc/min]. The difference is 7.5 [cc/min], therefore the increase in initial flow rate is 7.5/11.5*100= 65%. For a 2V step test, the initial flow rate is 19 [cc/min] and final flow rate is 25 [cc/min]. The difference between the final and initial flow rates is 6 [cc/min], therefore the increase in the initial flow rate is 6.0/19.0*100=32%.

The relationship between the controlling signal and the pump flow is shown in Figure C-1 on which a Least Squares (LS) fit is superimposed.

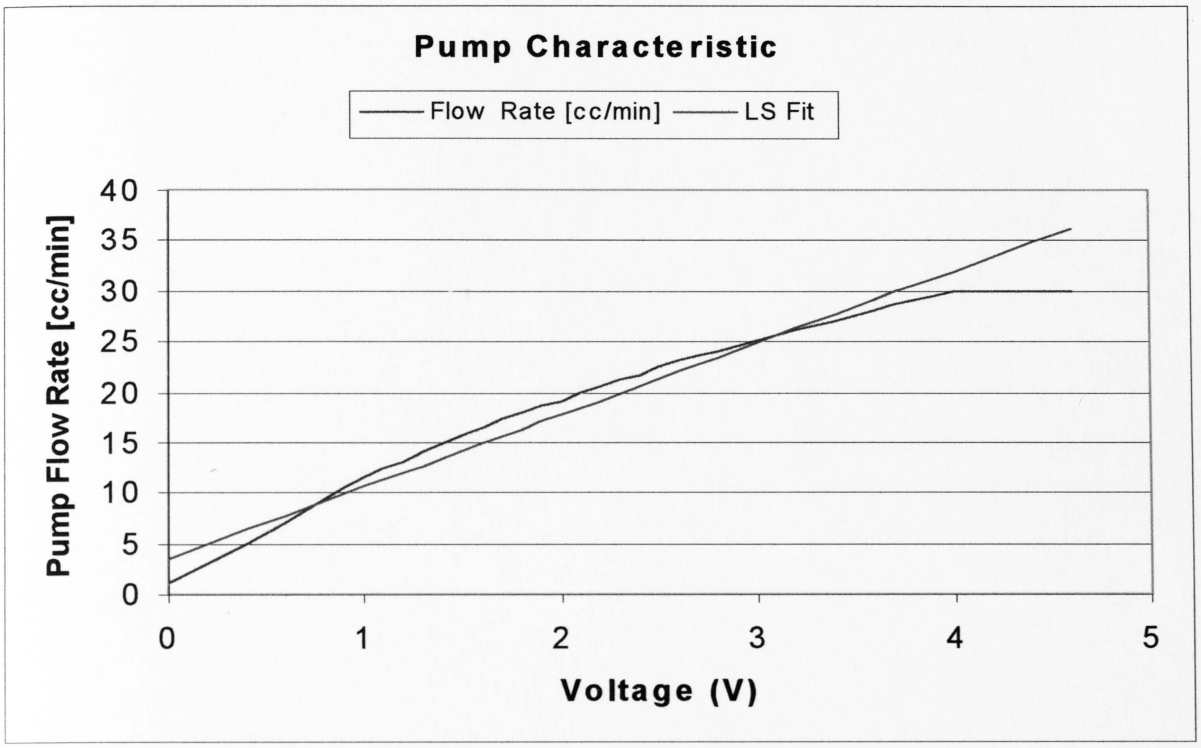


Figure C-1 Pump Characteristic

From the graph, it can be seen that the LS approximation is a fairly accurate representation of the actual pump characteristic. Furthermore, the pump flow can be seen to reach its maximum flow rate when driven by an input voltage of 4V. The following graph shows the squared error values resulting from the differences between the raw data from the pump's characteristic and the LS estimations (up to 4V).

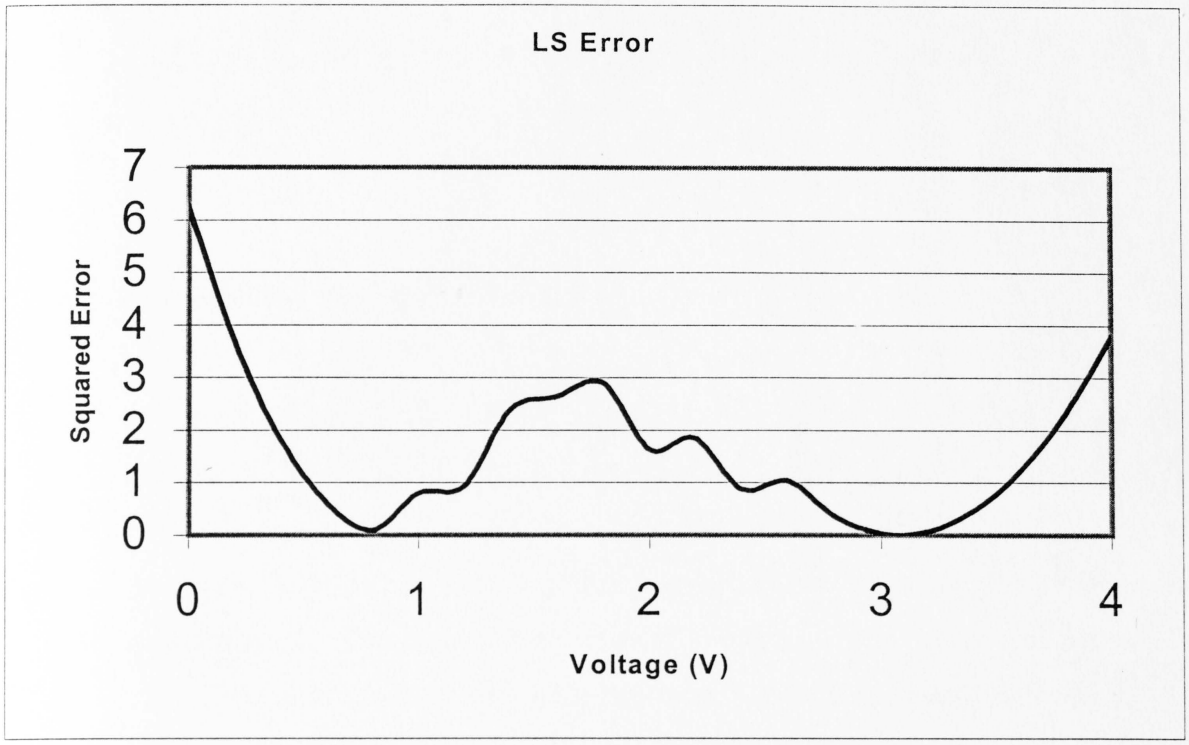


Figure C-2 LS Error on Pump Characteristic

As can be seen from the above plot, the points where the squared error is approximately zero indicate the intersection of the LS fit to the actual pump characteristic. Since, the error between the LS fit and the actual pump characteristic is low, the following linear equation (from the LS estimation) can be used to determine the pump characteristic (over the input voltage range of 0 – 4V).

$$y(cc/min) = mx(V) + b \quad (\pm error)$$

The "LINEST" function in Microsoft EXCEL was used to obtain the gradient, m and point of intersection, b of the LS linear equation.

$$\therefore y = 7.11x + 3.50 \quad (\pm 2.50)$$

Discussion:

The flow meter is accurate to 0.5 [cc/min]. As a result, the observed flow rate values are not as accurate as they could possibly be. The flow rates were recorded when steady state had been reached. The pump is nonlinear, as can be seen from its characteristic.

The pump should not be operated without water. Therefore, there should always be enough water in the bath (at least 2/3 full) before operation, and the level switch should at all times be free to move vertically.

C.2 Depth Sensor Characteristics

Calibration of the sensors is required in order to determine the characteristics of the depth sensors. The method of calibration follows. The tanks were filled up till they were full (with the tap closed). The voltages were taken from the depth sensors of both tanks. The tap was opened till the water level fell by 1cm, and then closed again. The voltage was then taken again at this water level. The process was repeated till the level of water was beneath the level of the tap. The following table shows the data obtained from the pump calibration.

Table C-2 Sensor Calibration Data

Height (cm)	*Sensor 1 Voltage (V)*	*Sensor 2 Voltage (V)*
28	9.05	5.25
27	5.67	4.32
26	3.06	3.53
25	1.38	3.16
24	0.00	2.80
23	-1.04	2.49
22	-1.97	2.12
21	-2.63	1.83
20	-3.18	1.57
19	-3.64	1.30
18	-4.05	1.06
17	-4.41	0.84
16	-4.67	0.66
15	-4.92	0.49
14	-5.17	0.32
13	-5.37	0.17
12	-5.55	0.03
11	-5.73	-0.11
10	-5.87	-0.23
9	-6.02	-0.36
8	-6.15	-0.47
7	-6.21	-0.54
6	-6.36	-0.66
5	-6.50	-0.79
4	-6.66	-0.98
3	-6.93	-1.42

From the data obtained, the relationships between the water level heights and the sensor voltages are shown in Figure C-3.

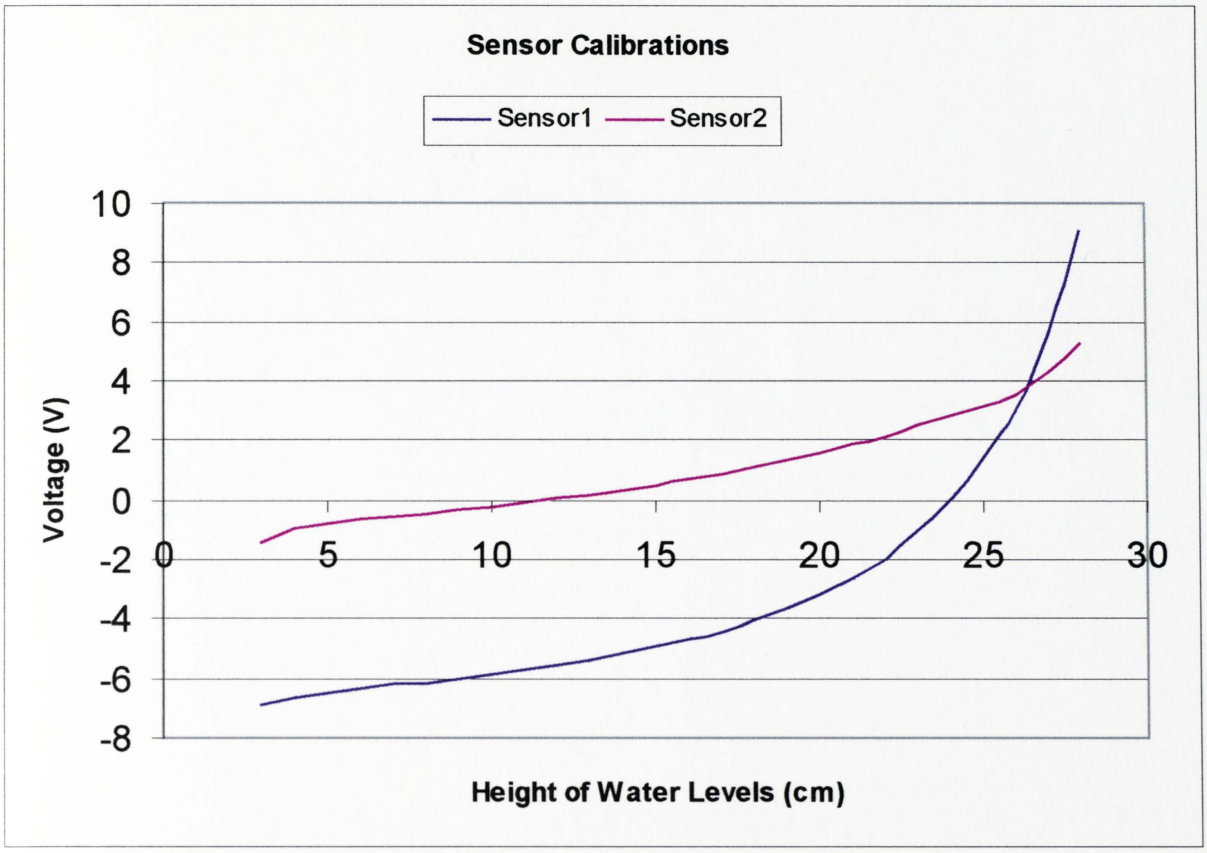


Figure C-3 Sensor Characteristics

It can be noticed that the characteristic for Sensor 1 is fairly well spread out over the operating range ±10V. However, the characteristic for Sensor 2 is over a small range of 7V. If the characteristic for Sensor 2 were used, there would be a decrease in the sensor sensitivity. The gains of the filters and zero offsets were then adjusted to make the characteristics of the sensors similar, with as wide an operating range as possible. The following table shows the data used to obtain the new depth sensor characteristics. Included are the LS estimations and the squared error values.

Table C-3 Sensor Characteristics with LS Data and Errors

	Sensor 1			Sensor 2		
Height (cm)	Voltage (V)	LS Data	Squared Error	Flow Rate (cc/min)	Voltage (V)	Squared Error
28	7.84	4.53	10.97	7.31	5.72	2.54
27	6.02	4.29	2.98	6.48	5.46	1.05
26	4.74	4.06	0.46	5.78	5.19	0.34
25	3.83	3.83	0.00	5.16	4.93	0.05
24	3.18	3.59	0.17	4.64	4.67	0.00
23	2.63	3.36	0.54	4.17	4.41	0.06
22	2.23	3.13	0.81	3.77	4.15	0.14
21	1.87	2.90	1.05	3.39	3.89	0.25
20	1.60	2.66	1.13	3.06	3.63	0.32
19	1.34	2.43	1.19	2.72	3.37	0.42
18	1.15	2.20	1.09	2.45	3.10	0.43
17	0.98	1.96	0.96	2.20	2.84	0.41
16	0.83	1.73	0.81	1.97	2.58	0.37
15	0.70	1.50	0.64	1.76	2.32	0.31
14	0.57	1.26	0.48	1.58	2.06	0.23
13	0.47	1.03	0.32	1.40	1.80	0.16
12	0.37	0.80	0.18	1.24	1.54	0.09
11	0.29	0.56	0.08	1.08	1.27	0.04
10	0.22	0.33	0.01	0.96	1.01	0.00
9	0.15	0.10	0.00	0.83	0.75	0.01
8	0.08	-0.14	0.05	0.69	0.49	0.04
7	0.02	-0.37	0.15	0.59	0.23	0.13
6	-0.03	-0.60	0.33	0.46	-0.03	0.24
5	-0.10	-0.83	0.54	0.30	-0.29	0.35
4	-0.18	-1.07	0.79	0.11	-0.55	0.44
3	-0.31	-1.30	0.98	-0.37	-0.82	0.20
Mean Squared Error			1.03			0.33

The LS error graphs for each sensor are shown after the respective sensor characteristic.

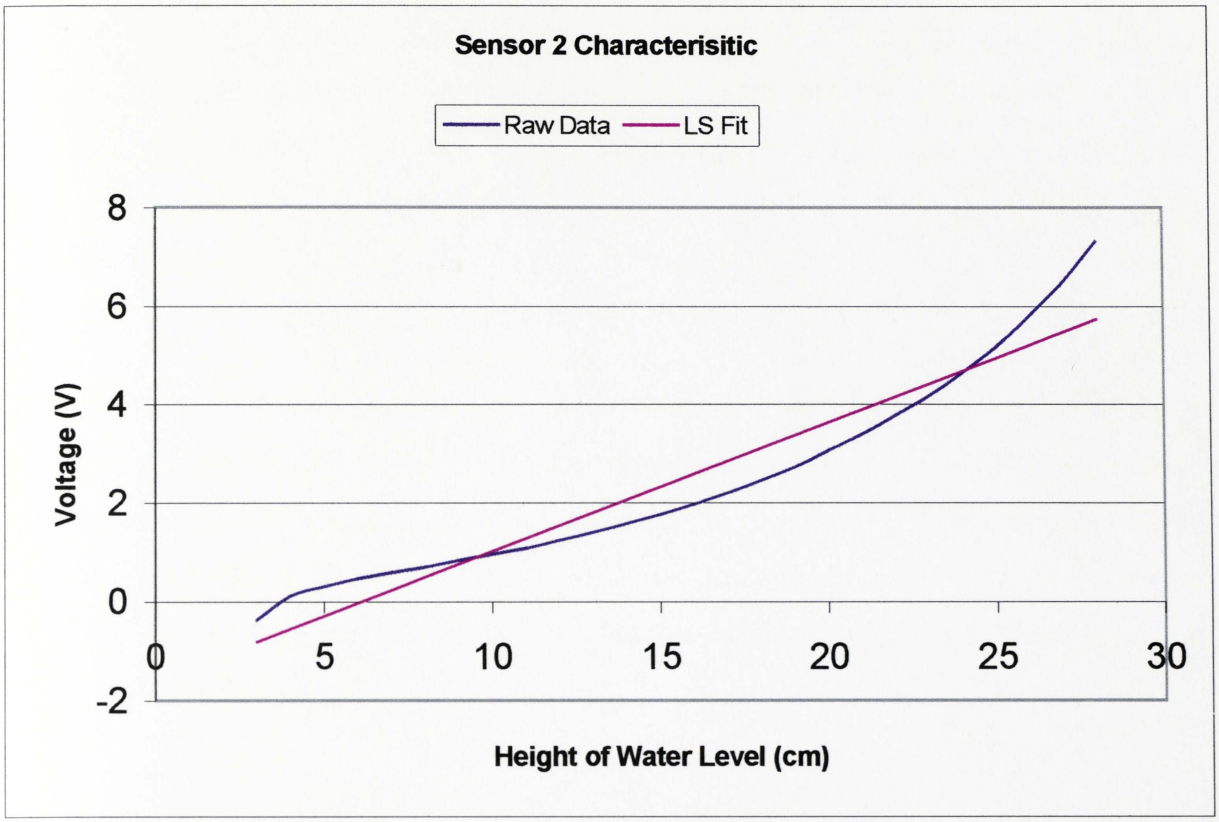


Figure C-6 Sensor 2 Characteristic

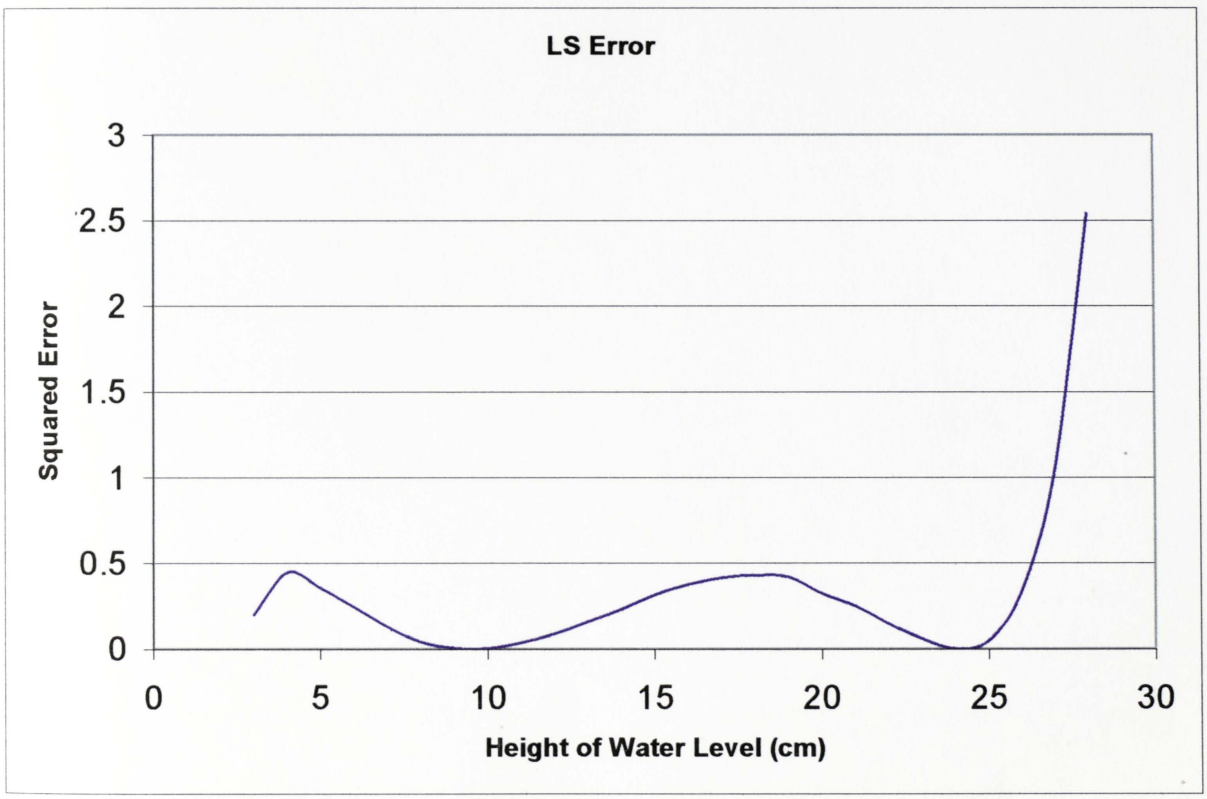


Figure C-7 LS Error on Sensor 2 Characteristic

As can be seen from the Sensor characteristics, the whole operating range of the sensors has not been used. This was due to the sensitivity of the filters to the sensors being too high; therefore any minor adjustment resulted in differing characteristics. The results that have been given, show that the characteristics are similar, and have a linear range approximately between 5 – 21 cm height of water. However, there is a loss in sensor sensitivity by not using the full operating range of the sensors (±10V).

The LS fit to the data of Sensor 1 is not as good as the fit to Sensor 2. However both fits are not good and better LS fits are required as the changes in the gradients of the characteristics cannot be accurately estimated by one linear equation. Furthermore, it can be seen that the LS error is fairly small until the height of the water in the tanks is greater than 26cm. The reason why the LS fits are not good is due to the fact that the sensors are highly nonlinear, especially above a water level height of 26 cm.

However, the following linear equations are used to describe the sensor characteristics. Given the general equation for the sensor characteristic:

$$y(V) = mx(cm) + b \quad (\pm error)$$

For Sensor 1:

$$y = 0.23x - 2.00 \quad (\pm 1.03)$$

For Sensor 2:

$$y = 0.26x - 1.60 \quad (\pm 0.33)$$

In order to obtain these accurate fits, three LS equations were used to estimate each characteristic, split into the following regions (water level heights): 3 – 4 cm, 5 – 21 cm and 22 – 28 cm. There are regions between 4 – 5 cm and 21 – 22 cm where the LS fits are nonlinear. These regions represent the transition regions between the three LS fits. The revised LS fits are shown in the following table, together with the squared error values.

Table C-4 Sensor Characteristics with LS Data and Errors

	Sensor 1			Sensor 2		
Height (cm)	Voltage (V)	LS Data	Squared Error	Flow Rate (cc/min)	Voltage (V)	Squared Error
28	7.84	7.07	0.59	7.31	6.98	0.11
27	6.02	6.17	0.02	6.48	6.39	0.01
26	4.74	5.27	0.28	5.78	5.81	0.00
25	3.83	4.37	0.30	5.16	5.22	0.00
24	3.18	3.47	0.09	4.64	4.64	0.00
23	2.63	2.58	0.00	4.17	4.05	0.01
22	2.23	1.68	0.31	3.77	3.47	0.09
21	1.87	1.65	0.05	3.39	3.15	0.06
20	1.60	1.53	0.00	3.06	2.96	0.01
19	1.34	1.41	0.01	2.72	2.77	0.00
18	1.15	1.29	0.02	2.45	2.59	0.02
17	0.98	1.17	0.04	2.20	2.40	0.04
16	0.83	1.06	0.05	1.97	2.21	0.06
15	0.70	0.94	0.06	1.76	2.02	0.07
14	0.57	0.82	0.06	1.58	1.83	0.06
13	0.47	0.70	0.05	1.40	1.64	0.06
12	0.37	0.58	0.04	1.24	1.46	0.05
11	0.29	0.46	0.03	1.08	1.27	0.04
10	0.22	0.34	0.02	0.96	1.08	0.02
9	0.15	0.22	0.01	0.83	0.89	0.00
8	0.08	0.10	0.00	0.69	0.70	0.00
7	0.02	-0.02	0.00	0.59	0.52	0.00
6	-0.03	-0.14	0.01	0.46	0.33	0.02
5	-0.10	-0.25	0.02	0.30	0.14	0.03
4	-0.18	-0.17	0.00	0.11	0.11	0.00
3	-0.31	-0.30	0.00	-0.37	-0.36	0.00
Mean Squared Error			0.08			0.03

The graphs of the revised LS estimations and the respective error graphs for each sensor follow.

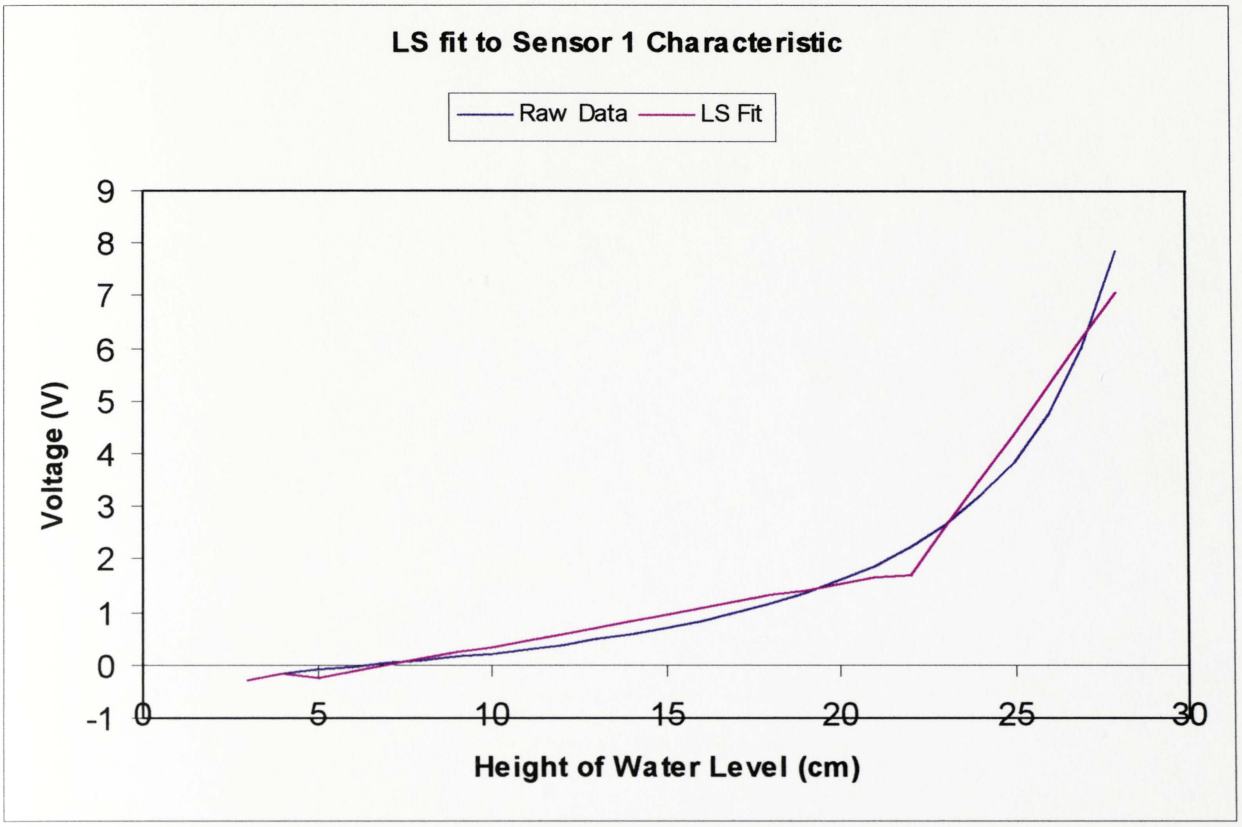


Figure C-8 LS Fit to Sensor 1 Characteristic

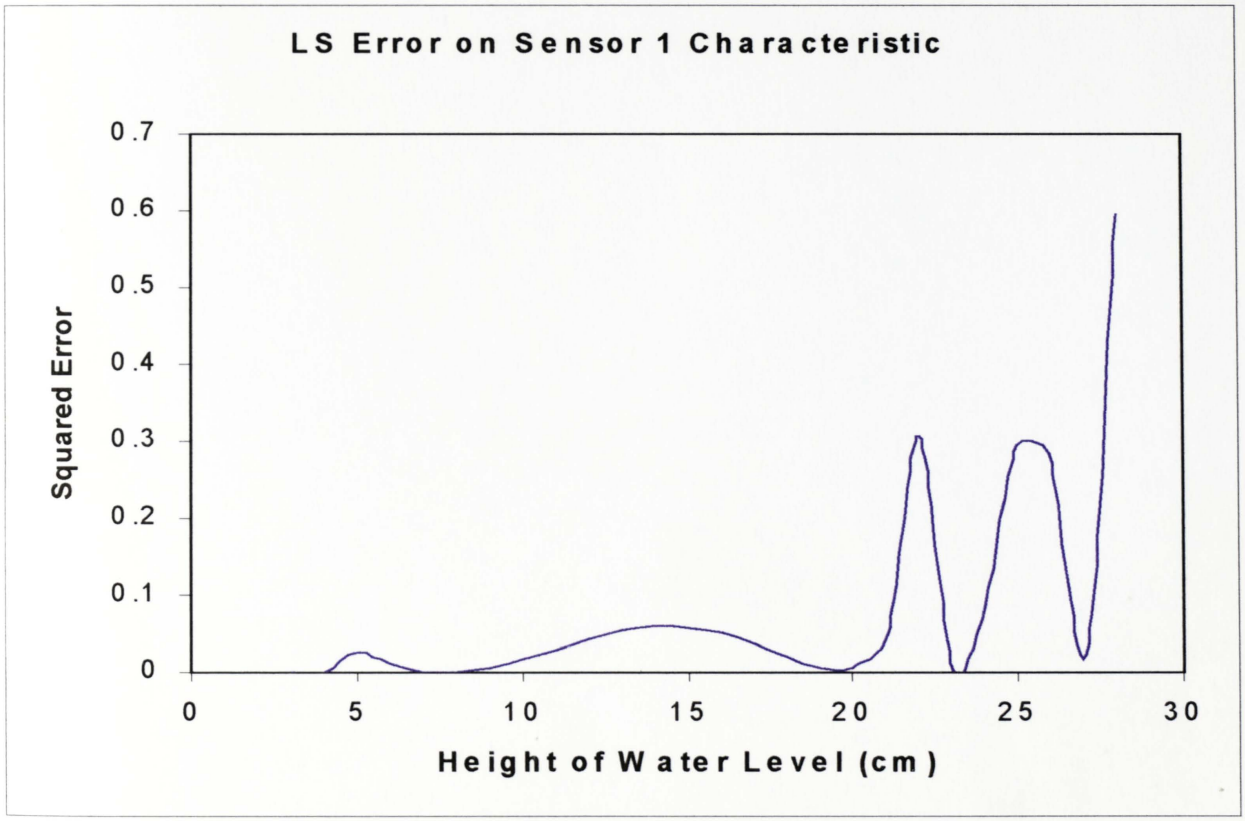


Figure C-9 Revised LS Error on Sensor 1 Characteristic

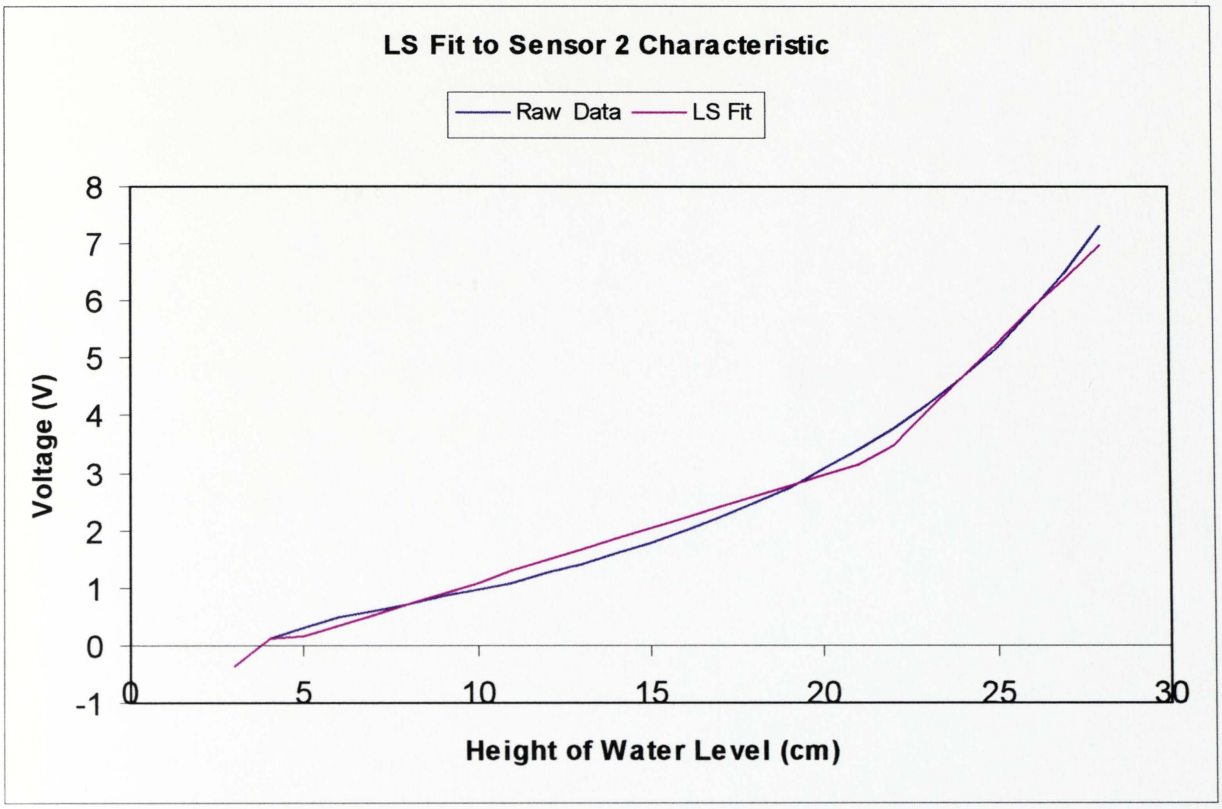


Figure C-10 LS Fit to Sensor 2 Characteristic

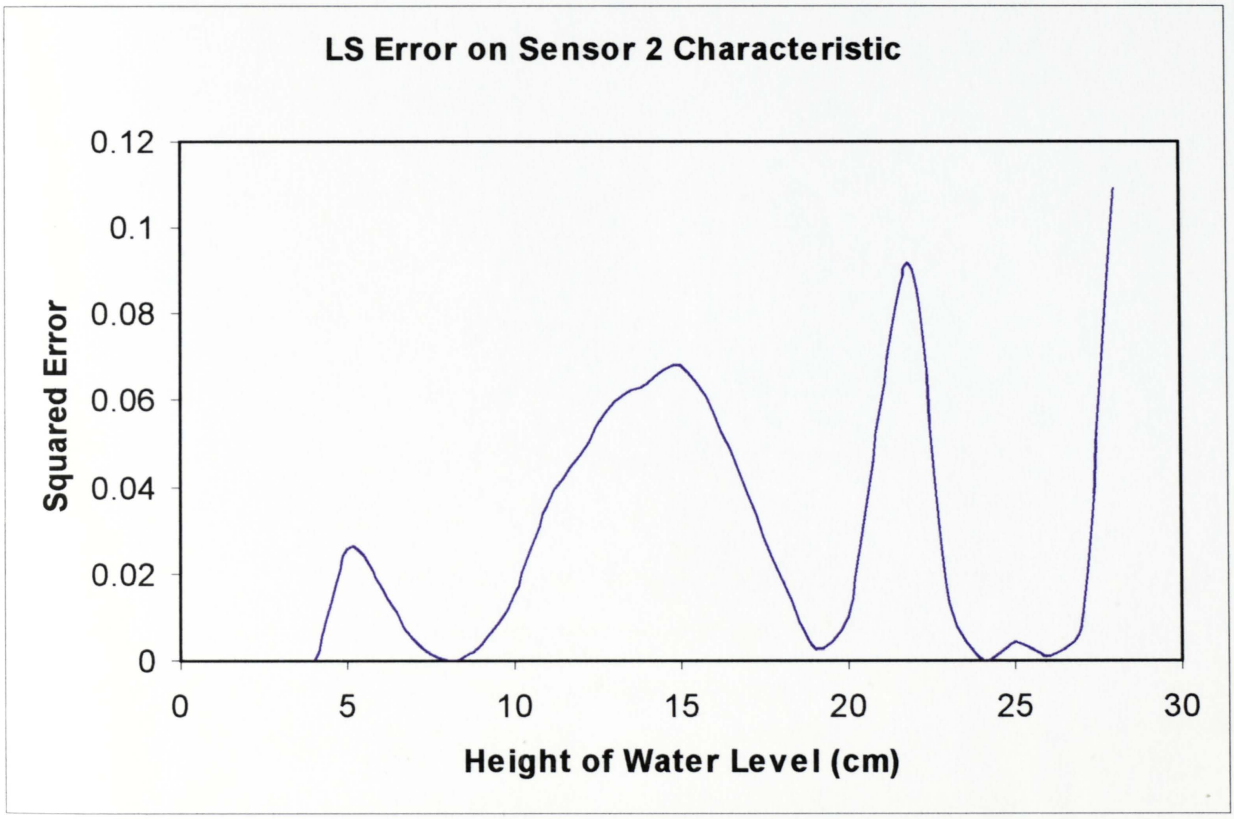


Figure C-11 LS Error on Sensor 2 Characteristic

From the Figures C-8 and C-10, it can be seen that the LS fits fairly accurately represent both sensor characteristics. From Figures C-9 and C-11, it can be seen that the LS error is very small, and that the LS fits to the sensor characteristics are more accurate. The LS error on the Sensor 2 characteristic is lower than the LS error on the Sensor 1 characteristic because Sensor 1 is more nonlinear than Sensor 2, resulting in a LS approximation to its characteristic which is not as accurate as the LS approximation to Sensor 2. Since the LS fits are fairly accurate, the following linear equations are used to describe the sensor characteristics. Given the general equation for the sensor characteristic:

$$y(V) = mx(cm) + b \quad (\pm error)$$

For Sensor 1:

$$y = \begin{cases} 0.13x - 0.70 \ (\pm 0.00) & for \ 3 \le x \le 4 \\ 0.12x - 0.85 \ (\pm 0.25) & for \ 5 \le x \le 21 \\ 0.90x - 18.10 \ (\pm 0.77) & for \ 22 \le x \le 28 \end{cases}$$

For Sensor 2:

$$y = \begin{cases} 0.98x - 0.00 \ (\pm 0.00) & for \ 3 \le x \le 4 \\ 0.19x - 0.80 \ (\pm 0.26) & for \ 5 \le x \le 21 \\ 0.56x - 9.40 \ (\pm 0.33) & for \ 22 \le x \le 28 \end{cases}$$

Discussion:

When the calibration was performed, there was noticeable drift in the output voltages of the sensors. The sensor output voltages were recorded once steady state had been reached. The accuracy of these values was to 0.005V.

The sensors are very sensitive, as wiping the sensors would result in significant change in the output voltages. Displacement of the sensors would also cause a disparity in the output voltages. Distilled water should not be used in the system, as it does not contain the impurities that the sensors use to function. The sensors use a change in capacitance between the two plates (in each sensor) to generate an output voltage. While air is present, the output voltage is zero. The capacitance changes with water level. The reason distilled water cannot be used is because it has a different dielectric constant to normal water (which is the media that TecQuipment have designed their product to work on).

APPENDIX D: LS & RLS THEORY

This appendix is based closely on the textbook, "Adaptive Control" by Åström and Wittenmark, 1989.

D.1 Least Squares

The Least Squares estimation equation: $\theta = \{U^T U\}^{-1}\{U^T y\}$ for the plant parameters of θ is an explicit formulation, which provides the best parameter values for a given experiment in a one-shot or one-step method. Implementation of this equation requires inversion of the matrix $\{U^T U\}$. Since the model parameters of the servomotor vary continuously, the parameters have to be updated at each sample interval using the latest measurements of the plant input and output.

Plant outputs y_t are usually corrupted by noise n_t. The Least Squares Estimators operate on these measurements. It is assumed that the input signal u_t, is not corrupted by noise. A general system is illustrated in the following figure.

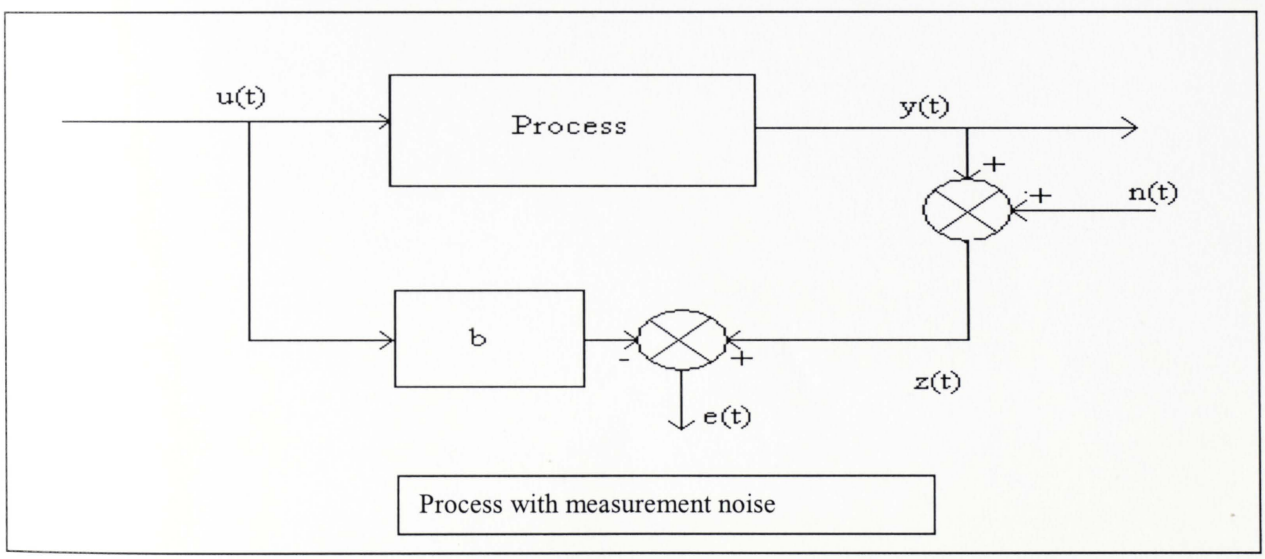


Figure D-1 Process with measurement noise

Measurement noise causes a bias in the values estimated for the model parameters. When (zero-mean) measurement noise is introduced into the outputs, the Least Squares Estimator (LSE) gives an incorrect estimate. The difference between the estimated and the actual parameters is called the estimator bias and is dependant on the statistical properties of the noise signal n_t. In a system the measurement noise results in a LSE equation based on the component matrix and vector $\{U^TU\}$ and $\{U^Ty\}$. Thus, there is a bias in the estimates. The bias term can be zero only if the noise signal n_t is derived from white noise w_t that is filtered by the characteristic function of the model.

D.1.1 Practical Considerations

- In practice, it is vital that $\{U^TU\}$ remains nonsingular at all times. This implies that the actual plant input should never become constant. Failure to do so leads to problems.
- The summations that make up the elements of the matrix and vector of the LS formula can become large, unless normalised.

D.2 Recursive Least Squares

Least Squares algorithms are usually formulated as sets of recursive equations. This enables estimates of the parameters of θ, to be available at each sampling interval. The difference between the predicted process output (model), obtained using the latest updated estimate of the parameter vector, and the actual output measured from the process can be attributed to measurement noise. These estimates of the measurement noise can be utilised for updating matrices in a Least Squares Estimation of the extended plant model.

In the recursive algorithm an inverse matrix P_N^{-1} that is defined by: $P_N^{-1} = U_N^T U_N$ is updated at each sample instant. This inverse matrix defines the relationship between the plant signals after N samples have been measured. P_N is known as the covariance matrix since it is proportional to the covariance matrix for the plant parameters (assuming that noise n_t is a white noise signal with zero mean and standard deviation σ).

Matrix P is defined as: $P^{-1} = U^TU$ so the expression for the model parameters after N samples is:

$$\theta_N = P_N\{U^T y\}$$

Thus, the matrix inversion associated with θ, is replaced by a simple scalar inversion in the recursive formulation. A recursive equation giving the latest parameter values at each sample instant can be developed from the recursive evaluation of the covariance matrix P_N.

$$K_{N+1} = P_N\begin{bmatrix} -y_N \\ u_N \end{bmatrix}\left\{1+\begin{bmatrix} -y_N & u_N \end{bmatrix}P_N\begin{bmatrix} -y_N \\ u_N \end{bmatrix}\right\}^{-1} = P_N\begin{bmatrix} -y_N \\ u_N \end{bmatrix}\Gamma_N$$

$$\theta_{N+1} = \theta_N + K_{N+1}\{y_{N+1} - \begin{bmatrix} -y_N & u_N \end{bmatrix}\theta_N\} = \theta_N + K_{N+1}e_{N+1}$$

$$P_{N+1} = P_N - K_{N+1}\begin{bmatrix} -y_N & u_N \end{bmatrix}P_N$$

These equations are the basis for the Recursive Least Squares Estimation Algorithm.

D.2.1 Initial Conditions

The recursive least squares estimator requires certain initial conditions for the parameter vector, θ_0 and an initial covariance matrix that is set to a diagonal matrix $P_0 = Diag[\sigma^2] = \sigma^2$, Where σ is the standard deviation assumed for the parameters. Initially, the elements of θ_0 are set to sensible estimates of probable parameter values based on a priori knowledge of the plant behaviour. In the absence of such estimates θ_0 can be started with all the elements at unity or at zero. The algorithm should be started with a very large (greater than 10^6) value for σ in defining P_0. Plant data can also be collected for a one-shot estimation of initial values for θ_0 and P_0 before starting the recursive algorithm

D.2.2 Practical Considerations

The practical considerations include:

- A more sophisticated recursive equation matrix for P_{N+1} (shown below) to ensure that the inverse covariance matrix remains symmetric ($P_N = P_{N+1}^T$).

  $$P_{N+1} = \{I - K_{N+1}\begin{bmatrix} -y_N & u_N \end{bmatrix}\}P_N\{I - K_{N+1}\begin{bmatrix} -y_N & u_N \end{bmatrix}\}^T + K_{N+1}K_{N+1}^T$$

  The recursive equation could also be implemented in the U-D covariance factorisation algorithm. This method is based on a factorisation of P as $P = UDU^T$ where D is diagonal and U is an upper-triangular matrix. The UD factorisation does not include square root calculations and is therefore well suited to real-time applications like the servomotor.

- The inverse matrix P^{-1} can become singular should the values of the signals u_t and y_t remain at constant values for an extended period of time. This would result in large discrepancies in the least squares estimation due to the instability of the plant input.
- Measurement noise on the plant output will bias the parameters obtained by the least squares algorithm; sometimes the bias is so severe that the actual plant does not resemble the estimated model at all. Estimating more parameters than required as described by the extended Least Squares Method can alleviate the problem. The reason for the failure of the normal least squares method can be attributed to the least squares algorithm simultaneously estimating both the parameters of the plant model and those of a hypothetical model of the noise characteristics.

D.2.3 Time Varying Systems

As recursive least squares methods are designed for plants in which non-linear effects result in a time varying behaviour, it is conventional to forget old data in the favour of new. This is accomplished by introducing a scalar forgetting factor λ, in the covariance updating equation. This can be seen in the following equation.

$$P_{N+1} = P_N - K_{N+1}[-y_N \quad u_N]P_N / \lambda$$

Where λ is a scalar constant chosen in the range: $0.9 < \lambda < 1.0$ according to. When λ is 1.0 the old data is not discarded. In practice a λ of 0.98 ensures that old data is discarded very rapidly. The rate at which old data is rejected should be matched to the rate at which the plant is likely to change. For most engineering applications the selection of a plant's speed to change its characteristics and select an appropriate λ, is important. Lambda is related to a pole in the z-plane at $z = \lambda$. The old data is weighted by a factor of less than 0.05 if it is more than $-3T/\ln[\lambda]$ time units old and by a factor of less than 0.02 if it is more than $-4T/\ln[\lambda]$, where T is the sample interval.

The forgetting factor λ is a tool for rejecting old data in favour of new data. The estimation algorithm may require a jump-start into revising its current estimates for a particular parameter or set of parameters in vector θ. This selective re-evaluation of particular sections of θ can be done by means of a random walk. This is where the covariance matrix is suddenly increased by the addition: $P = P + R$. Where matrix R is a positive definite matrix (often a

diagonal for ease of specification). As matrix R is a very selective method of dealing with parameter estimates, it should be used with caution.

APPENDIX E: KALMAN FILTER

The values of the covariance matrix, R and matrix, Q were determined by using the Matlab programming language.

E.1 Calculation of R matrix

The value of R is obtained from calculating the covariance of the output measurements, *z(k)* in a fault-free environment with random noise of 0.05V maximum amplitude. The covariance equation is:

$$R = E[z(k) \cdot z(k)^T]$$

The value of R was obtained from averaging the results of the individual elements of covariance matrices calculated from three simulations.

Table E-I Covariance Matrices for different Simulations

Simulation	*R* Matrix
1	$\begin{pmatrix} 0.02577 & 0.00812 \\ 0.00812 & 0.01507 \end{pmatrix}$
2	$\begin{pmatrix} 0.02637 & 0.00909 \\ 0.00909 & 0.01541 \end{pmatrix}$
3	$\begin{pmatrix} 0.02541 & 0.00784 \\ 0.00784 & 0.01498 \end{pmatrix}$
Average	$\begin{pmatrix} 0.02585 & 0.00834 \\ 0.00834 & 0.01518 \end{pmatrix}$

E.2 Calculation of Q matrix

The Q matrix (which is diagonal) was computed by running a Matlab optimisation program, "Kalopt.m". The optimisation program uses an initial guess for Q to generate the observer gains and the observer response. A cost function is then generated by calculating the difference between the absolute values of the real data and the model data. The eigenvalues of the observer are controlled by a parameter called "ROOT". Setting this parameter to a value of 10 for example, will imply that an additional large cost will be incurred if the observer eigenvalues should stray towards a value of less than –10[dB/s]. The program then evaluates the values of Q for different values of "ROOT" so as to allow a comparison of the Q matrix, and thus choose an appropriate Q matrix. The following table shows the calculated diagonal elements of the Q matrices with varying "ROOT" parameters, for different simulations, where:

$$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}$$

Table E-II Diagonal elements of the Q matrices with different "rootsize", for three simulations

	Simulation 1		Simulation 2		Simulation 3		Average	
"root" value	Q_1	Q_2	Q_1	Q_2	Q_1	Q_2	Q_1	Q_2
0	2.40E-08	2.50E-10	-8.31E-08	1.05E-10	1.30E-07	1.39E-11	2.35E-08	1.23E-10
1	2.72E+01	1.10E-11	2.81E+01	-6.05E-03	2.59E+01	2.94E-03	2.71E+01	-1.04E-03
2	2.97E+03	-9.52E+00	4.43E+04	-1.55E+00	8.46E+03	8.15E+00	1.86E+04	-9.73E-01
3	9.21E+04	-2.07E-07	1.11E+05	1.57E+01	1.24E+04	2.34E+01	7.18E+04	1.30E+01
4	1.63E+05	5.04E+00	2.90E+01	1.56E+02	6.56E+04	3.48E+01	7.62E+04	6.51E+01
5	1.48E+01	7.94E+01	1.26E+04	-2.45E+02	1.43E+01	7.65E+01	4.22E+03	-2.98E+01
6	3.70E+04	-1.10E+02	4.54E+05	-1.09E+02	3.24E+05	2.78E+01	2.72E+05	-6.36E+01
7	2.95E+01	1.59E+02	9.26E+01	4.97E+02	8.13E+03	4.16E+01	2.75E+03	2.32E+02
8	3.88E+01	2.08E+02	3.17E+03	-6.52E+02	1.50E+05	1.75E+02	5.11E+04	-8.95E+01
9	4.92E+01	2.64E+02	1.02E+06	2.32E+02	1.90E+05	1.66E+02	4.04E+05	2.21E+02
10	6.09E+01	3.27E+02	2.23E+04	1.02E+03	5.87E+01	3.15E+02	7.48E+03	5.54E+02

The average values for Q_1 and Q_2 for the different values of "ROOT" > 0 are:

$Q_1 = 90800 \pm 136881$

$Q_2 = 90 \pm 196$

$$\therefore Q = \begin{pmatrix} 90800 & 0 \\ 0 & 90 \end{pmatrix}$$

The question then arises as to whether it is practical to implement the resultant Q matrix, which has elements with such large standard deviations.

E.3 Applied results using the Kalman Filter

The following graphs show the residuals obtained from the parity space method with the use of a Kalman filter.

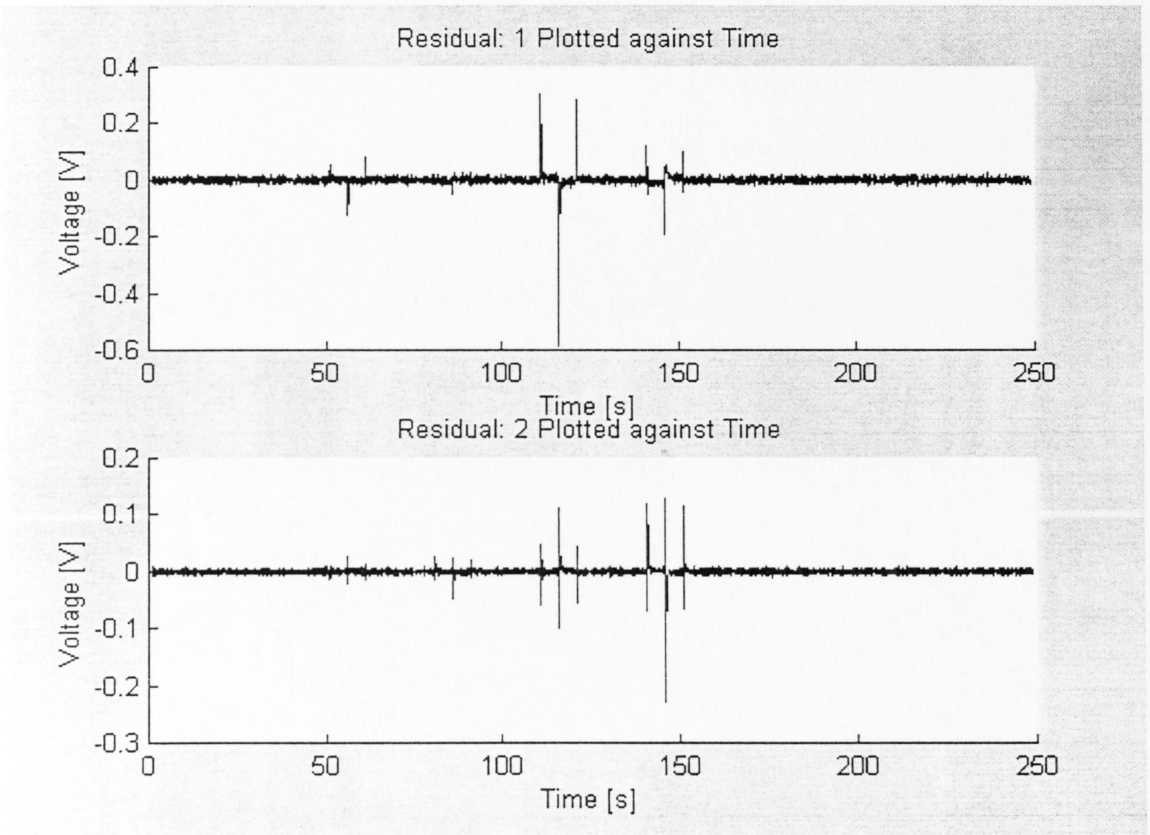


Figure E-1 Parity Space Residuals 1 & 2 with Kalman filter, using Q1

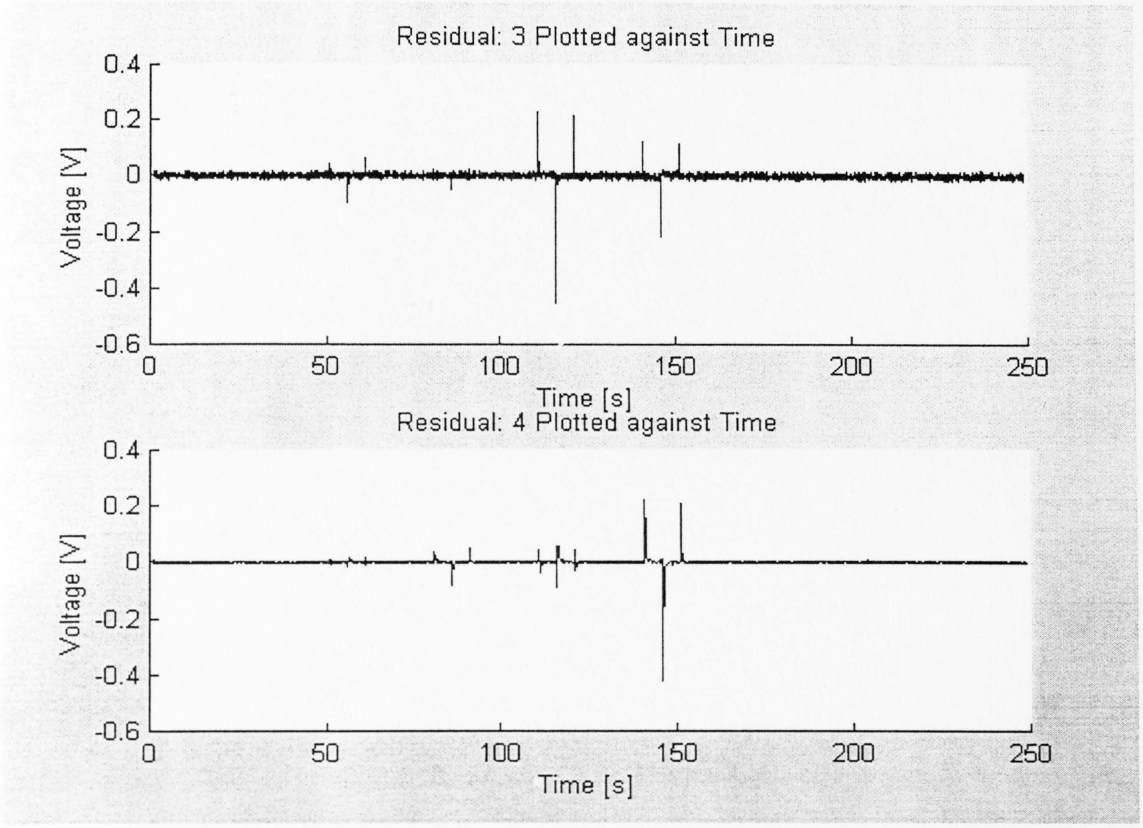


Figure E-2 Parity Space Residuals 3 & 4 with Kalman filter, using Q1

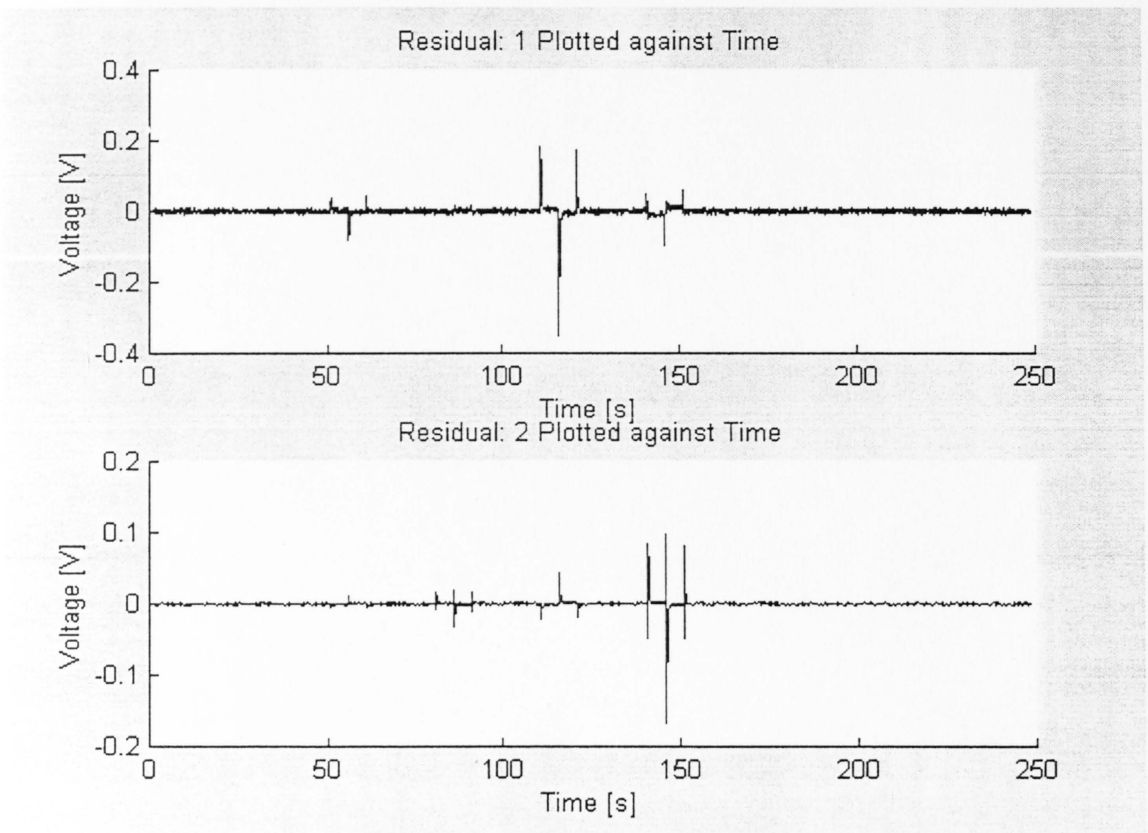


Figure E-3 Parity Space Residuals 1 & 2 with Kalman filter, using Q2

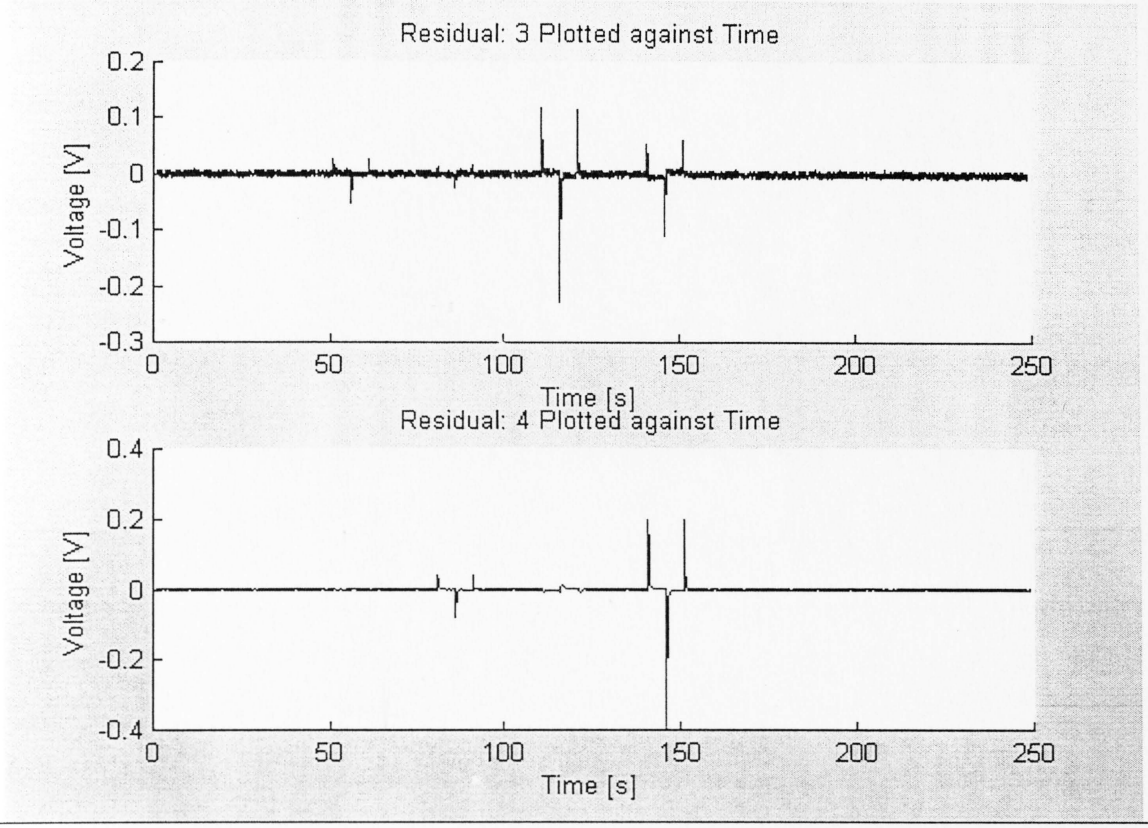


Figure E-4 Parity Space Residuals 3 & 4 with Kalman filter, using Q2

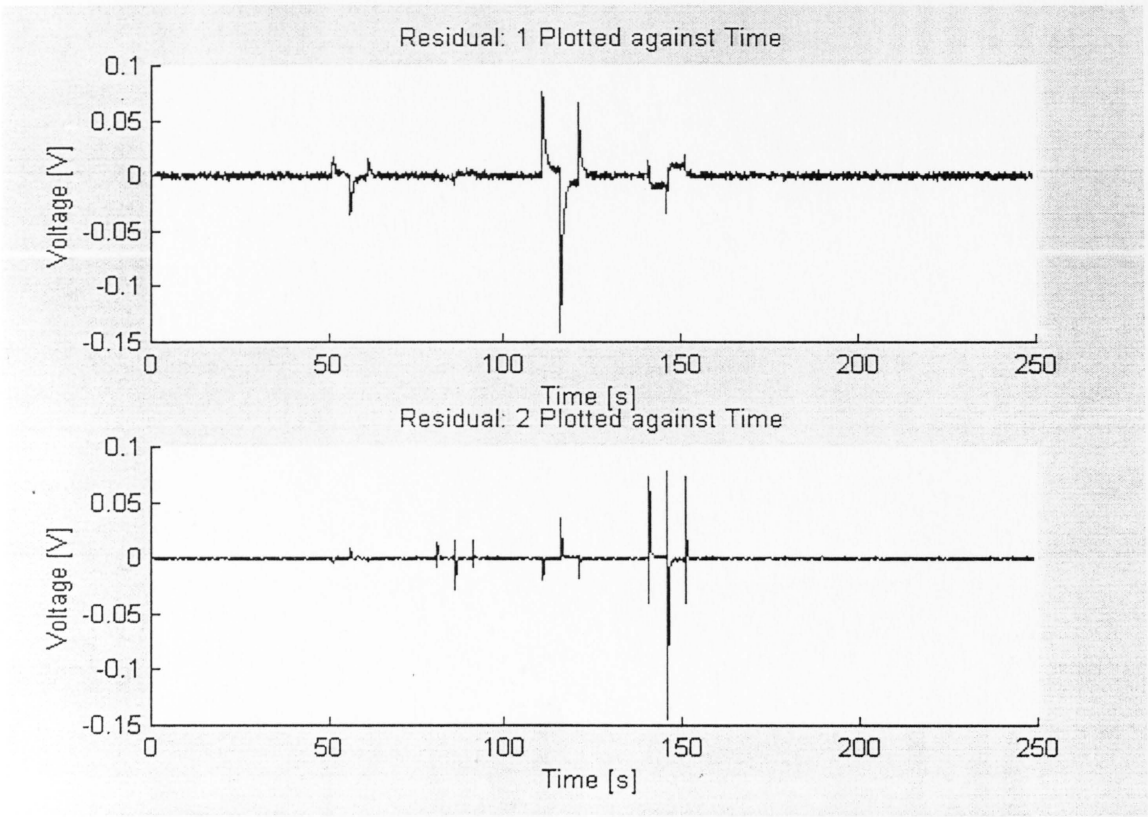


Figure E-5 Parity Space Residuals 1 & 2 with Kalman filter, using Q3

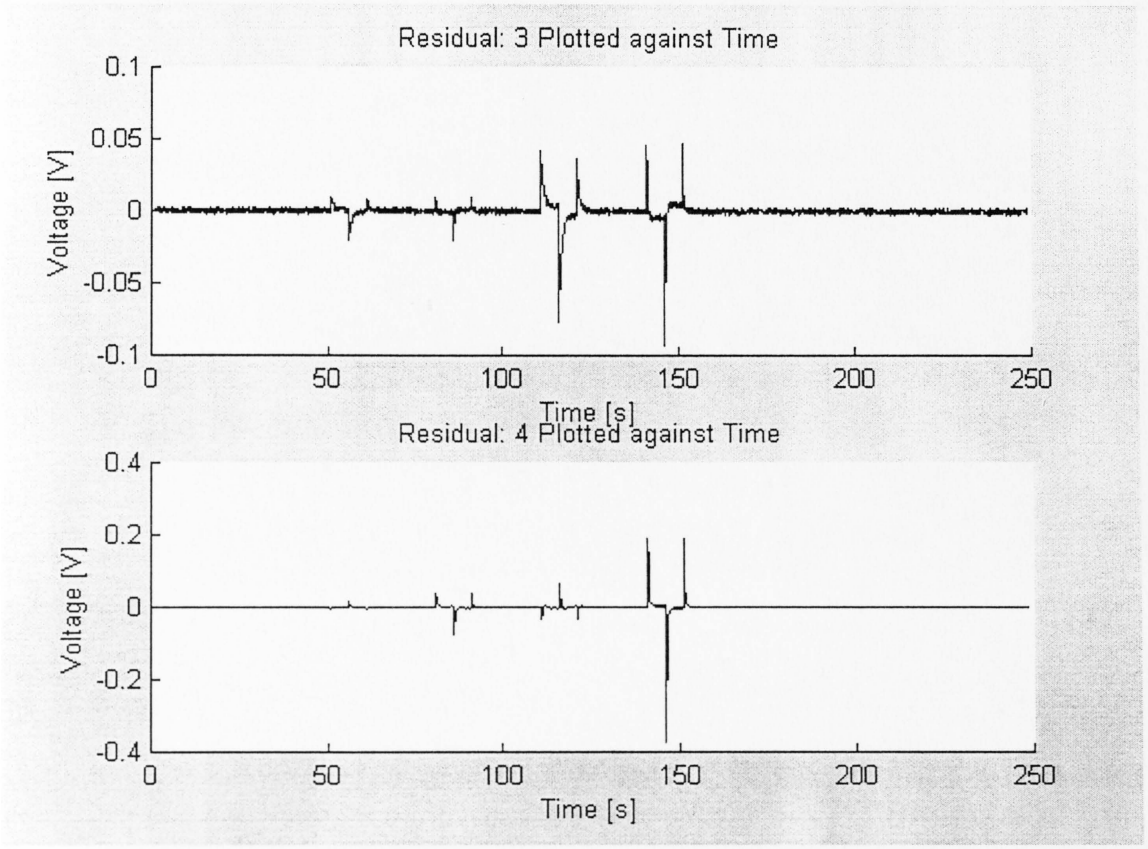


Figure E-6 Parity Space Residuals 3 & 4 with Kalman filter, using Q3

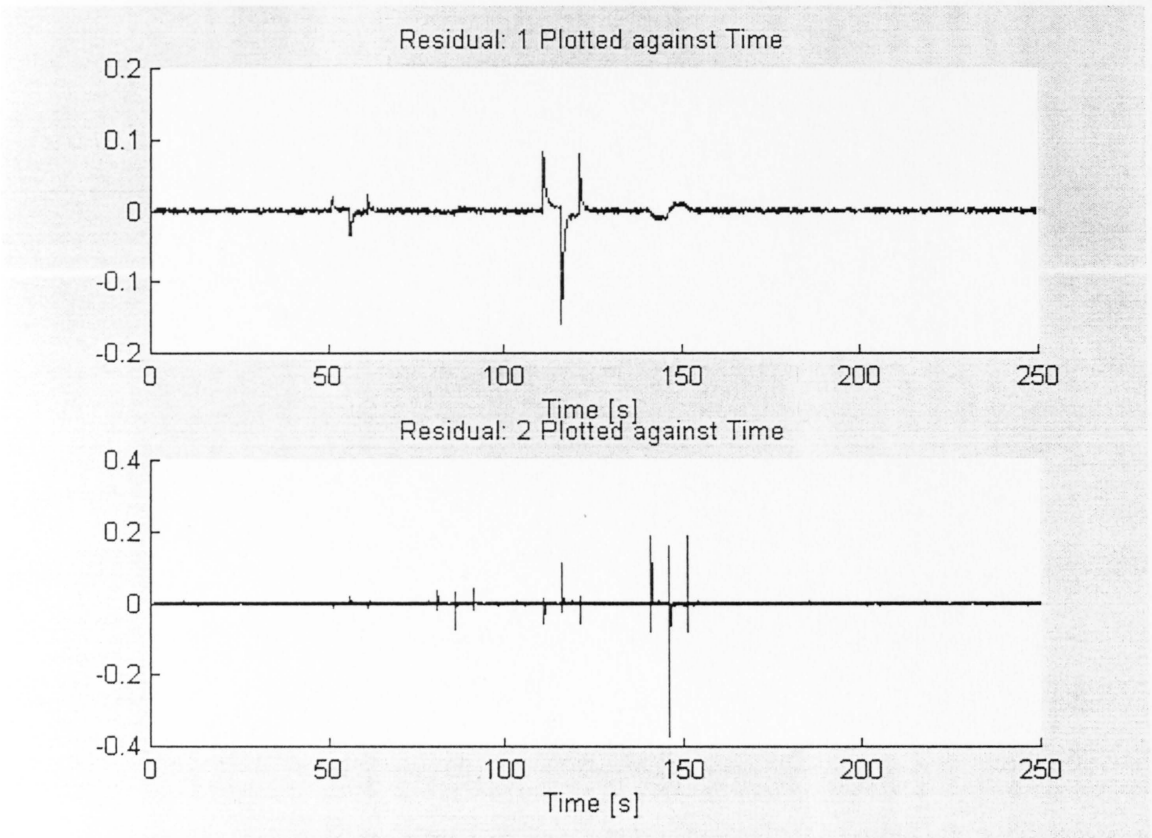


Figure E-7 Parity Space Residuals 1 & 2 with Kalman filter, using Q4

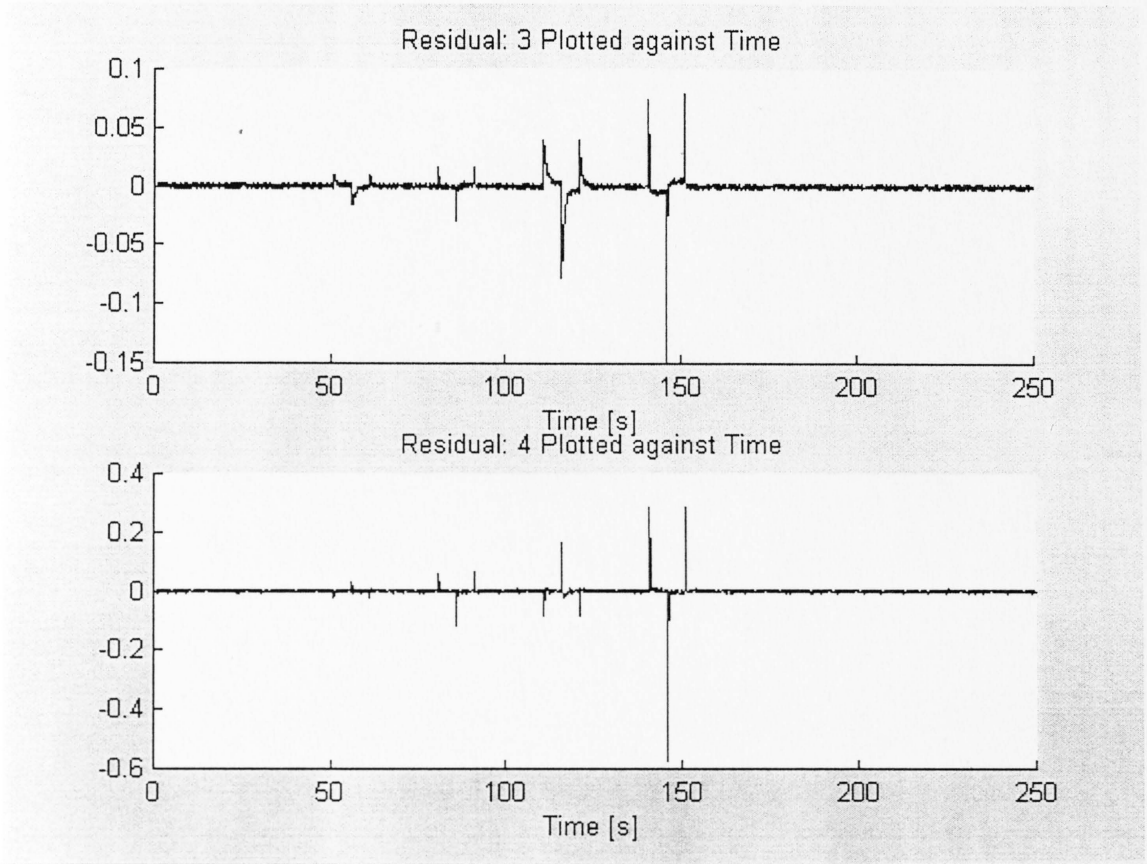


Figure E-8 Parity Space Residuals 3 & 4 with Kalman filter, using Q4

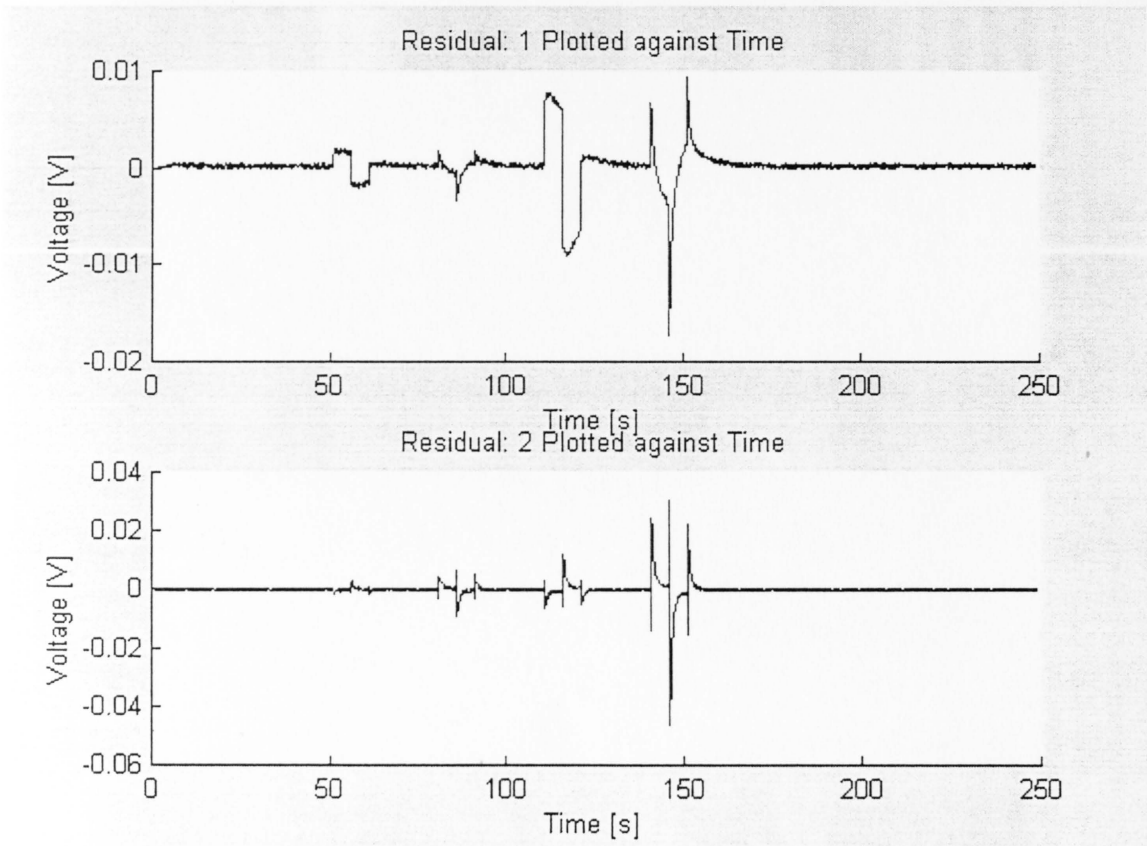


Figure E-9 Parity Space Residuals 1 & 2 with Kalman filter, using Q5

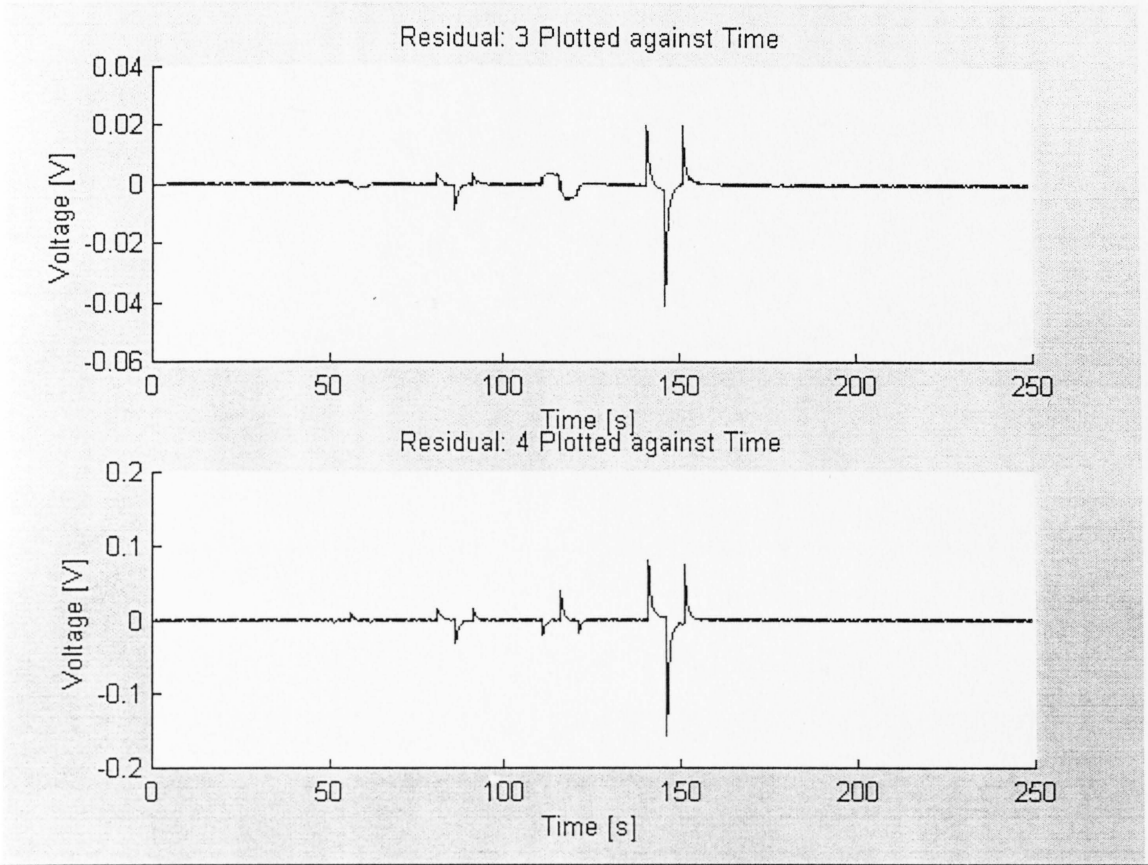


Figure E-10 Parity Space Residuals 3 & 4 with Kalman filter, using Q5

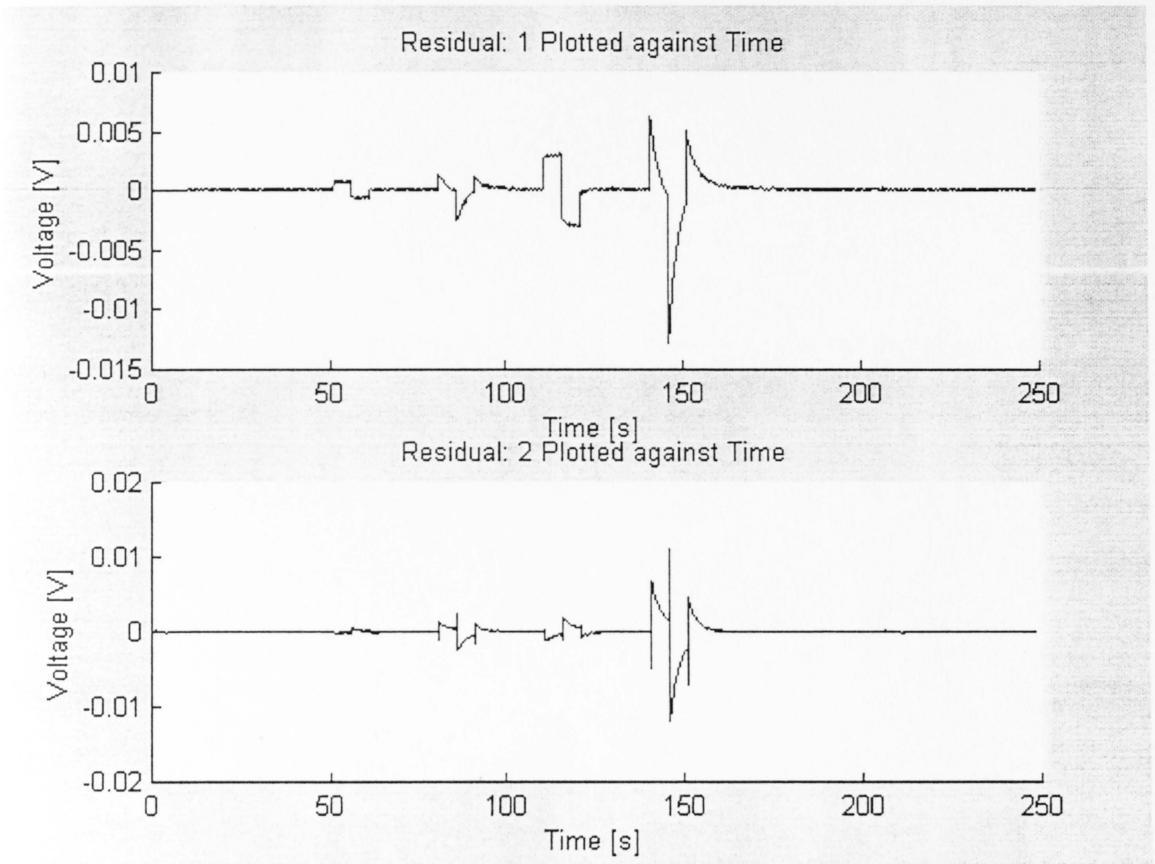


Figure E-11 Parity Space Residuals 1 & 2 with Kalman filter, using Q6

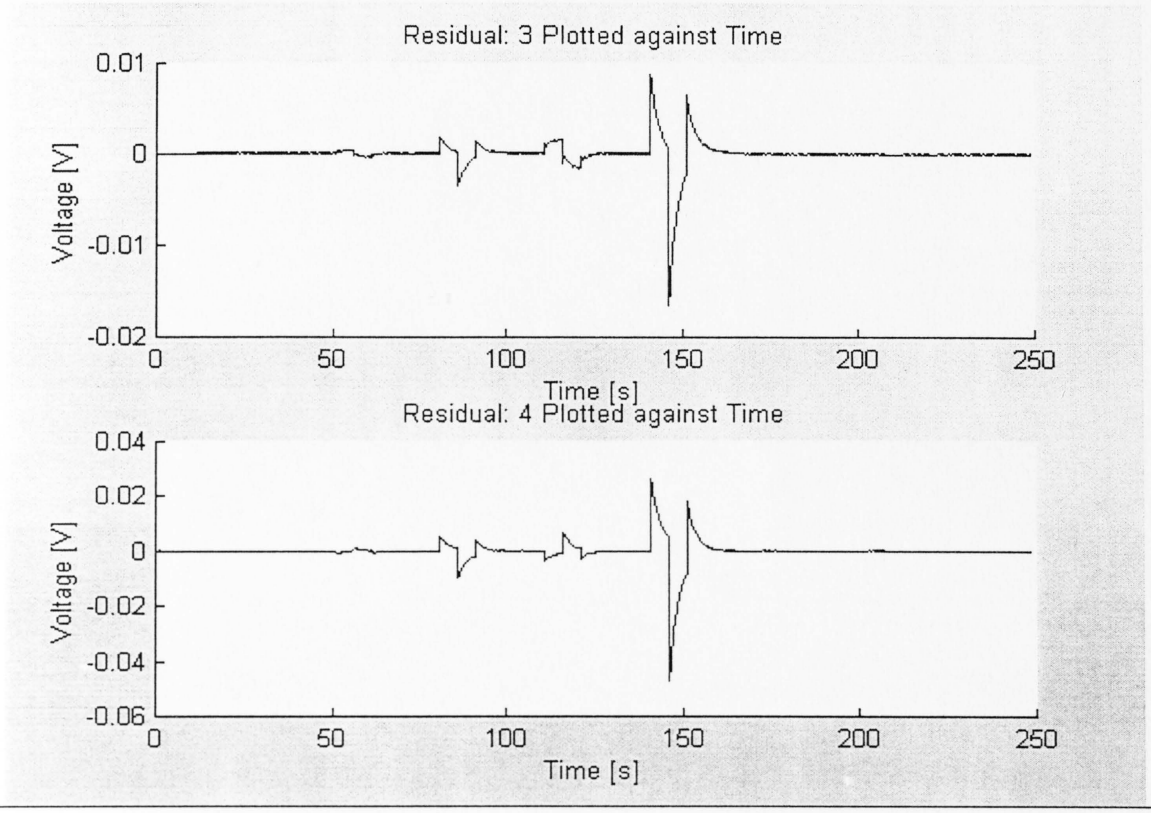


Figure E-12 Parity Space Residuals 3 & 4 with Kalman filter, using Q6

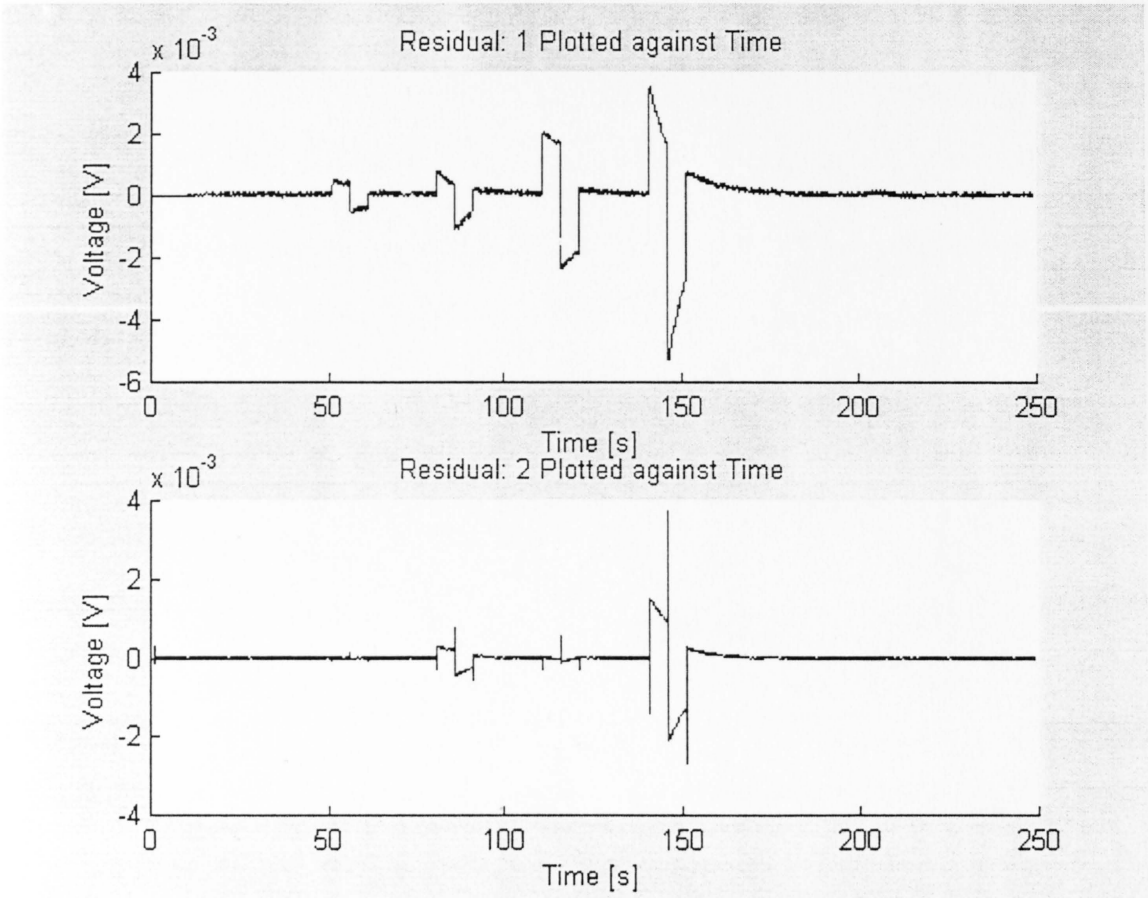


Figure E-13 Parity Space Residuals 1 & 2 with Kalman filter, using Q7

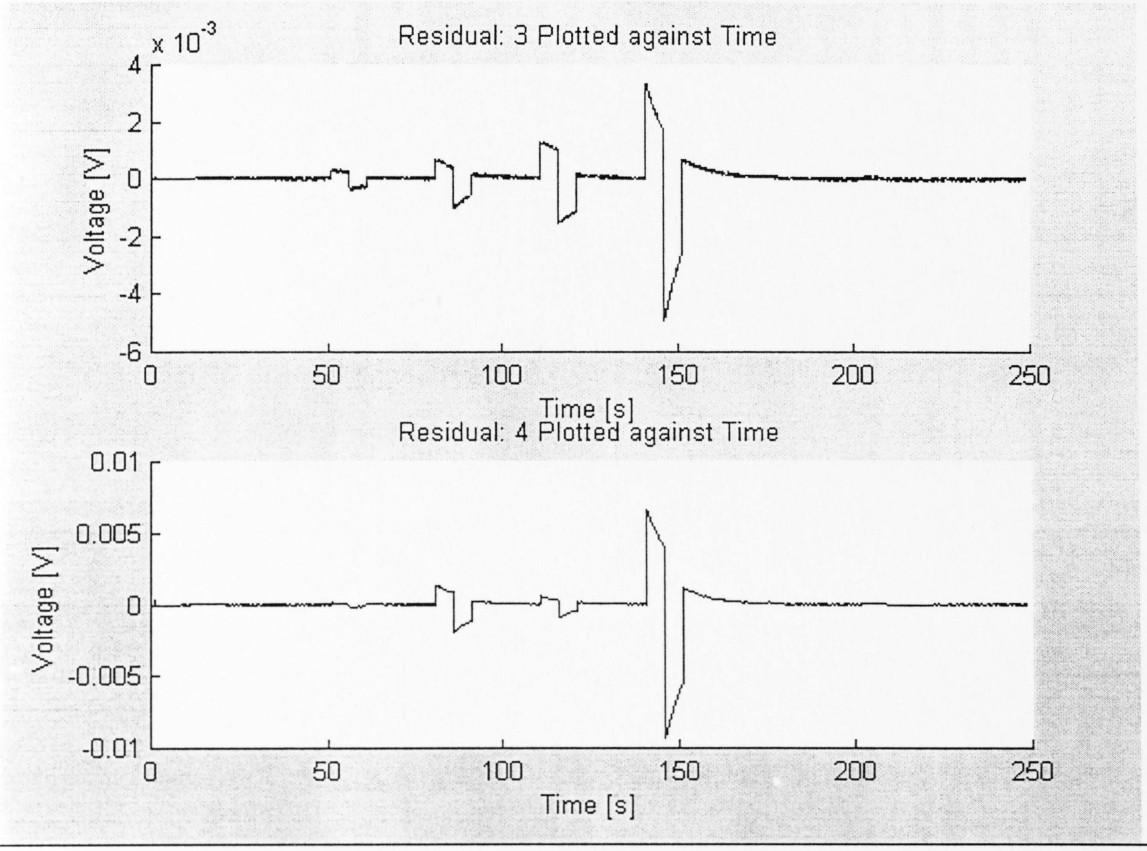


Figure E-14 Parity Space Residuals 3 & 4 with Kalman filter, using Q7

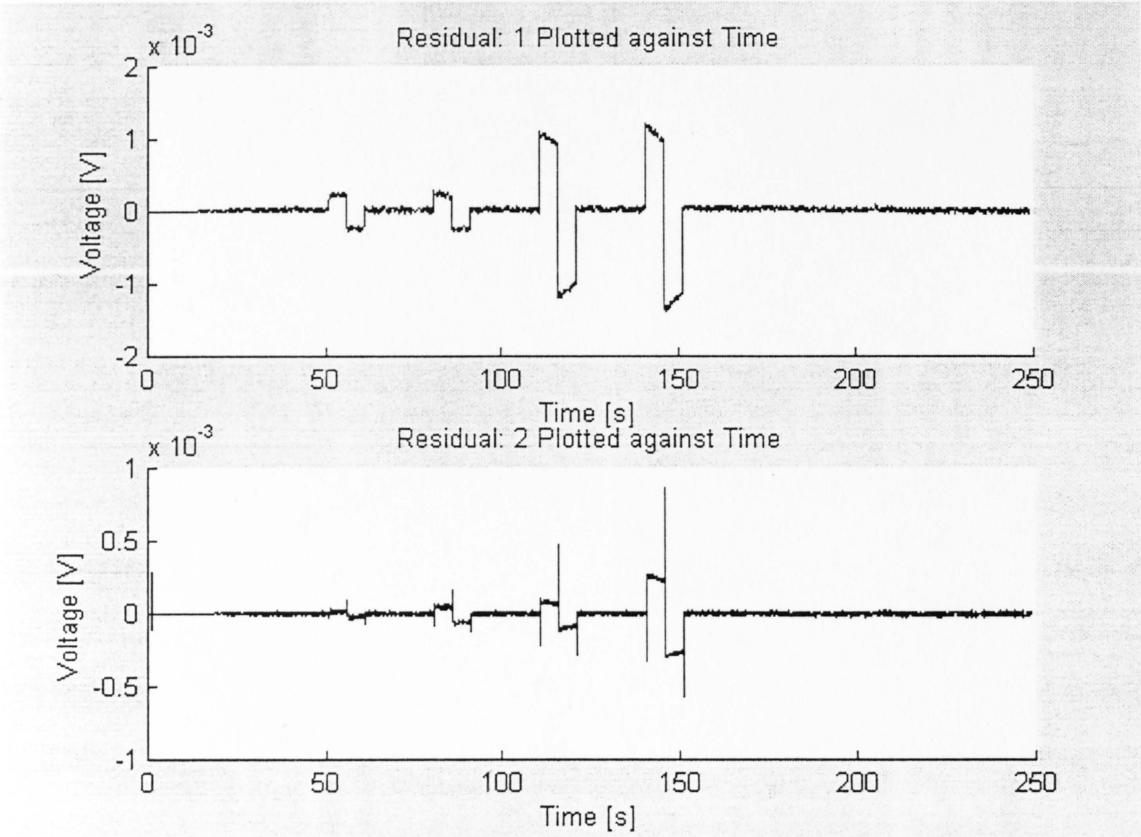


Figure E-15 Parity Space Residuals 1 & 2 with Kalman filter, using Q7

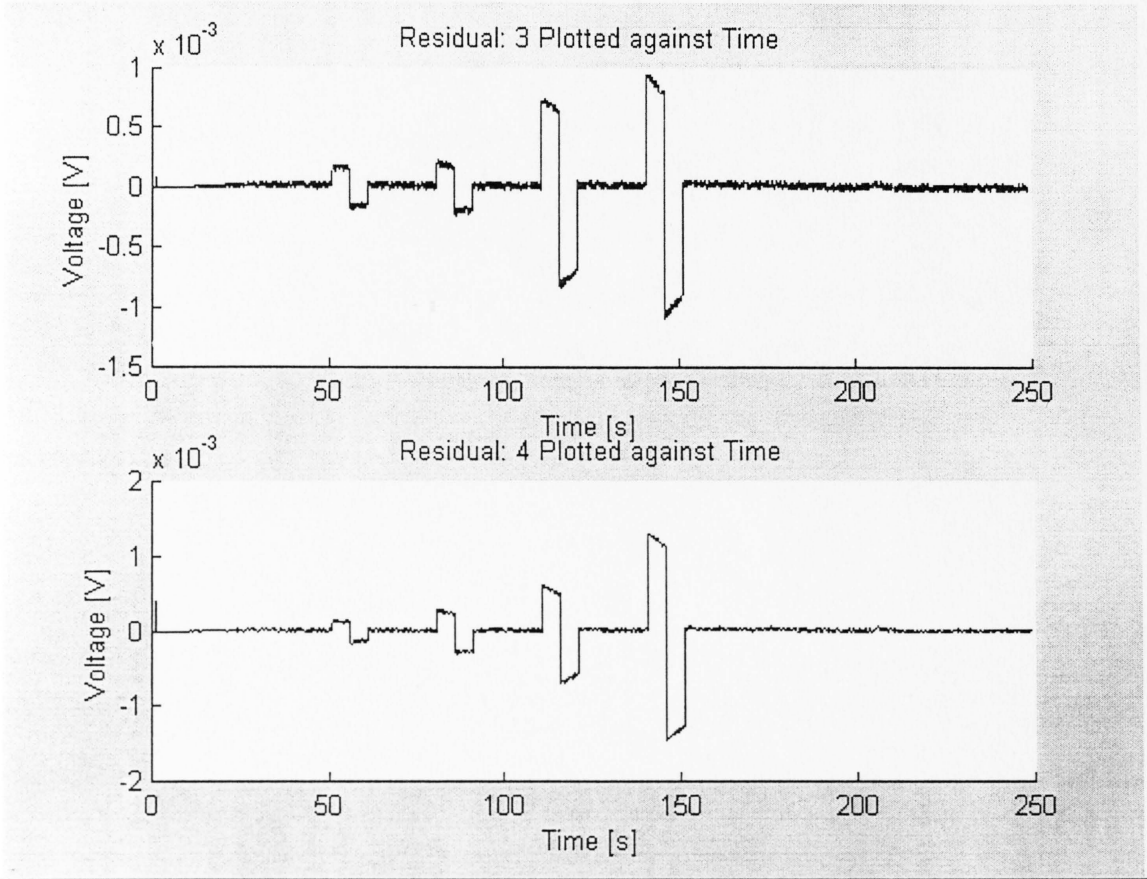


Figure E-16 Parity Space Residuals 3 & 4 with Kalman filter, using Q8

Legend	Q_1	Q_2
Q1	90800	90
Q2	9000	90
Q3	900	90
Q4	1000	1000
Q5	10	10
Q6	1	1
Q7	0.1	0.1
Q8	0.01	0.01

Comment:

From the graphs shown, it is seen that as the value of the Q matrix decreases, the time constant of the filter increases, therefore the decay of the residual deviations increases. Also

the noise on the residuals also decreases, with a decreasing Q matrix. Both these observations are predicted by theory. The optimum value of the Q matrix should be at the point where the noise on the residuals is low, and the time constant of the filter high, so as to obtain better fault tracking. As the Q matrix increases, the residuals exhibit initial "spikes" (due to stepping the process), as can be seen from the residual Graphs in Figures E-11 to E-16. These "spikes" if large enough, will be detected as faults, by the fault detection method, therefore it is necessary to choose a Q value which does not have large "spikes". As a result the Q6 matrix was chosen to be the optimal matrix, which has low noise, a fairly long time constant and shows "spikes", which are not significant enough to be detected as faults.

APPENDIX F: EFFECT & DETERMINATION OF STABILISING & FORGETTING FACTORS ON THE APS FAULT DETECTION

This appendix explains the effect that the stabilising and forgetting factors have on the recursive least squares estimation of the residuals for the Adaptive Parity Space (APS) method of fault detection.

F.1 Forgetting Factor and Stabilising Factor values

The optimal values for the forgetting factor, λ and stabilising factor, α were found through trial and error. Given that old data is weighted by a factor of less than 0.05 if it is more than $-3\Delta t/\ln[\lambda]$ time units old, the forgetting factor must be dependent on the sampling time, Δt (refer to Appendix D for more details). The process uses a sampling time of 0.1[s], and in practice, $0.9<\lambda<1.0$ with a sampling rate of 1[s]. Therefore, to get the appropriate forgetting factor for the process, the following equation must be solved:

$$\frac{-3\Delta t_{1s}}{ln\,\lambda_{1s}} = \frac{-3\Delta t_{0.1s}}{ln\,\lambda_{0.1s}}$$

$$\therefore \frac{1}{ln\,\lambda_{1s}} = \frac{0.1}{ln\,\lambda_{0.1s}}$$

$$\therefore ln\,\lambda_{0.1s} = 0.1 \cdot ln\,\lambda_{1s}$$

$$\therefore \lambda_{0.1s} = \lambda_{1s}^{\,0.1}$$

Where λ_{1s} and $\lambda_{0.1s}$ are the forgetting factors for sampling times of 1[s] and 0.1[s] respectively, and Δt_{1s} is a 1[s] sampling time and $\Delta t_{0.1s}$ is a 0.1[s] sampling time. Then, $\lambda_{0.1s}$ is the corrected forgetting factor that should be used for a 0.1[s] sampling time, on the RLS estimation. For consistency, the stabilising factor was also scaled by the same value as that used for the forgetting factor. The optimum values for λ and α are:

$$\lambda = 6.64 \times 10^{-1}$$

$$\alpha = 5.75 \times 10^{-9}$$

F.2 Determination of Forgetting Factor

The determination of the forgetting factor was performed through trial and error. Since the LS estimation of the residuals resulted in a fairly accurate fault detection scheme, it is expected that the old data should be forgotten quickly. Although a range of forgetting factors were investigated, the residuals obtained for three different forgetting factors are shown in the following diagrams.

Table F-1 Fault Parameters used for the Simulations

Simulation Number	λ_{1s}	$\lambda_{0.1s}$
1	0.300	0.887
2	0.017	0.664
3	0.002	0.537

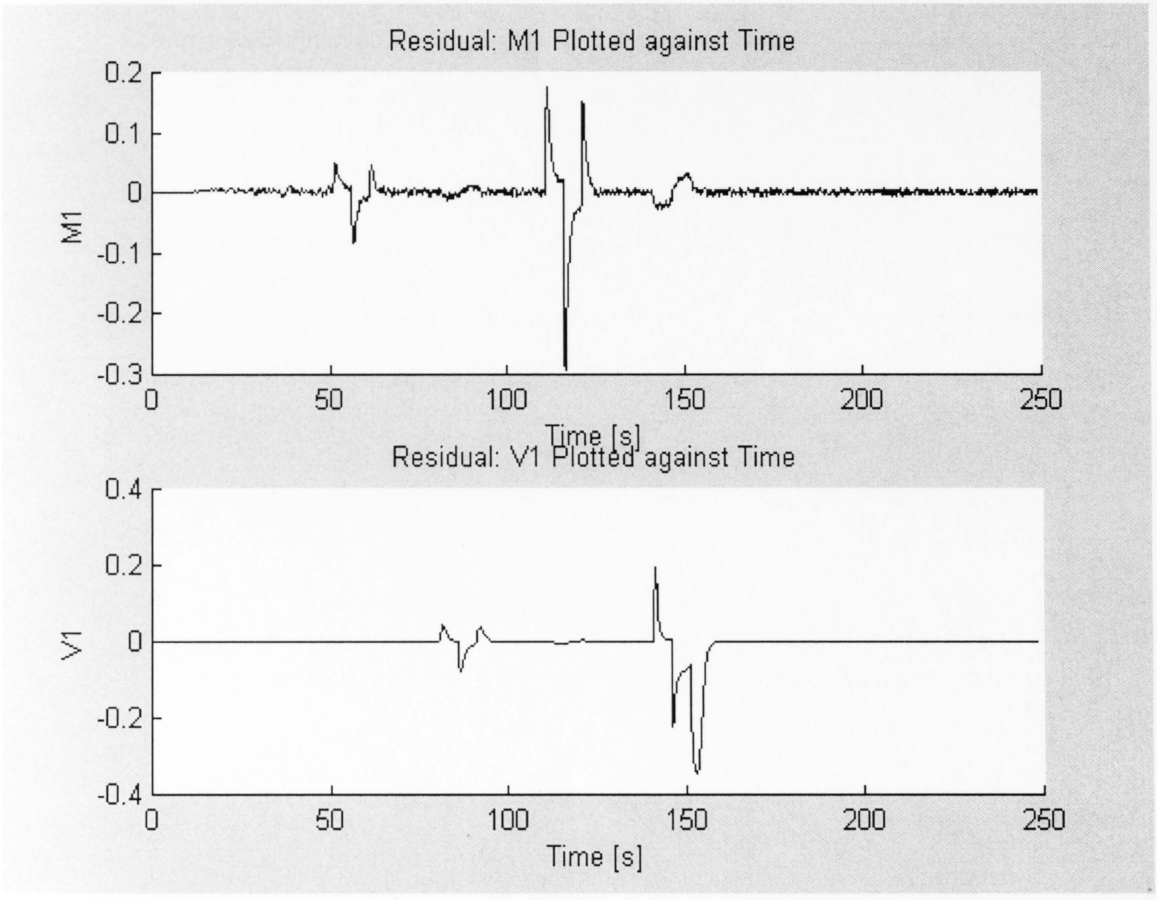


Figure F-1 APS Residuals M1 & V1 for Simulation1

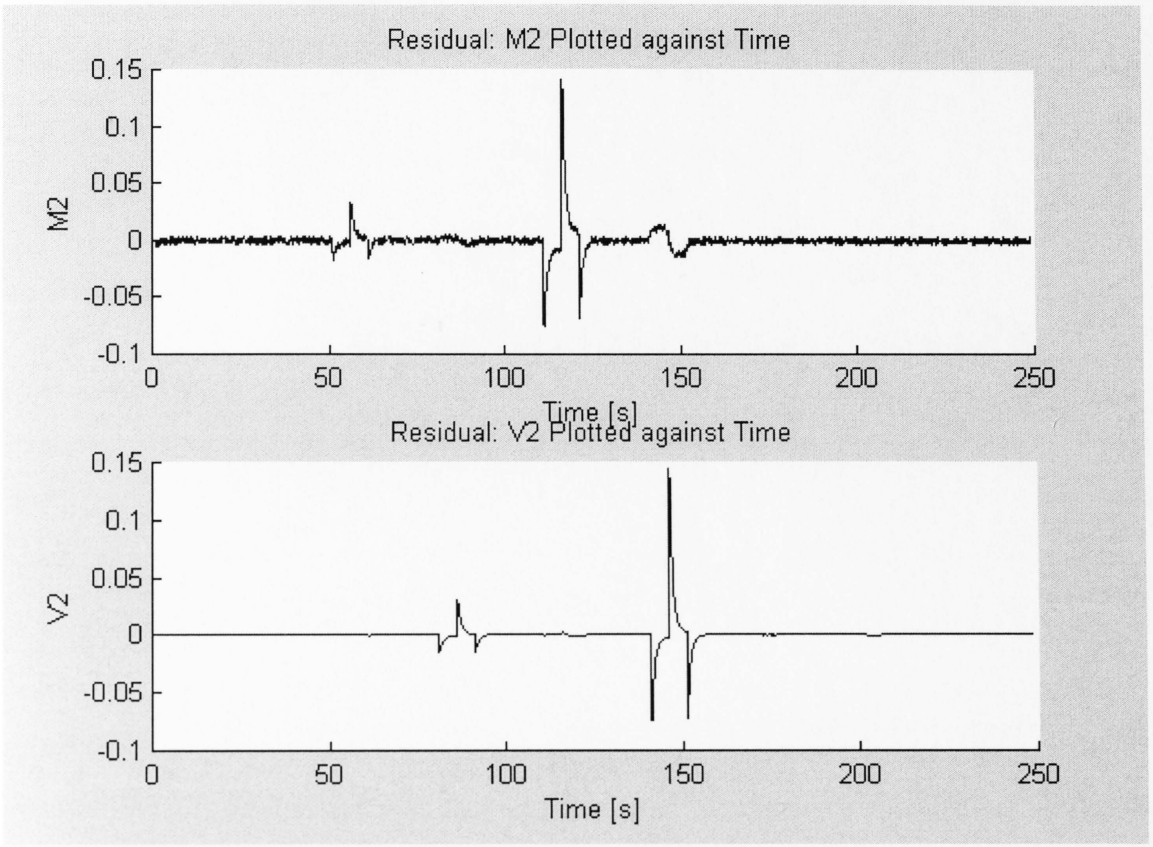


Figure F-2 APS Residuals M2 & V2 for Simulation 1

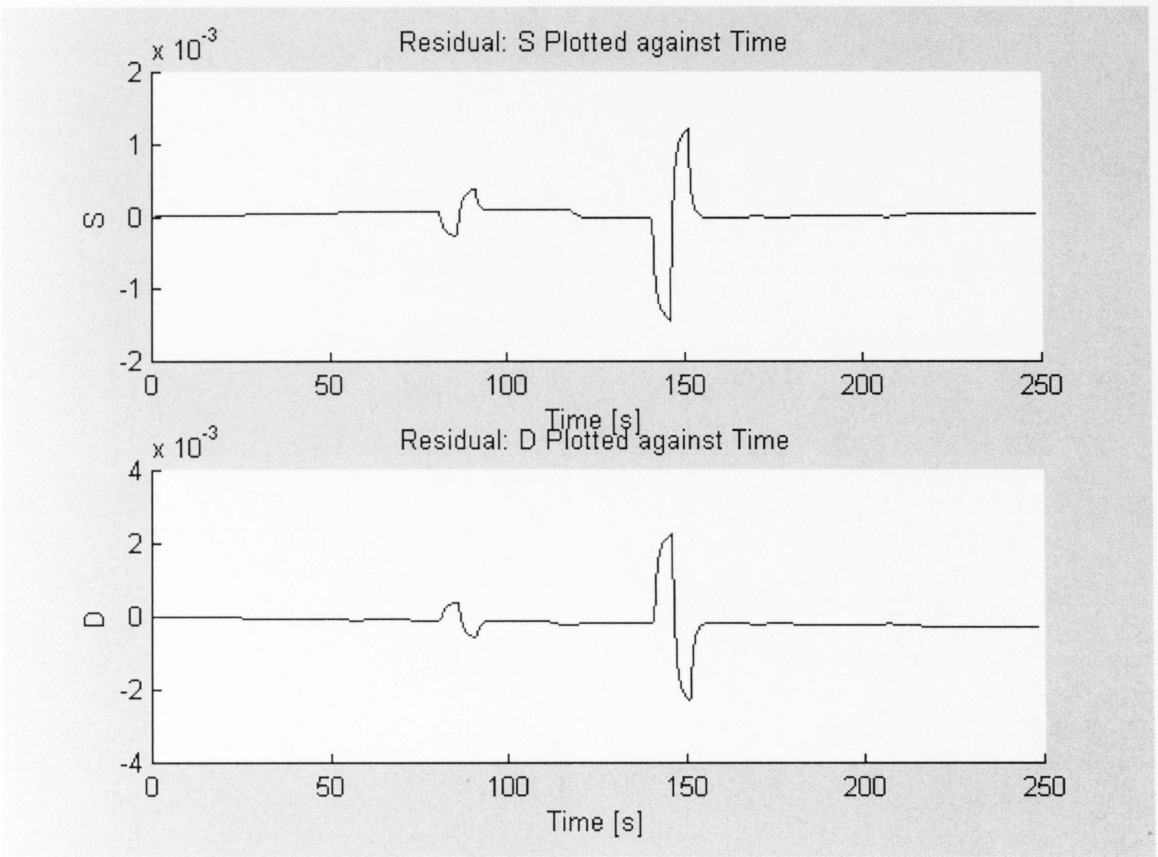


Figure F-3 APS Residuals S & D for Simulation 1

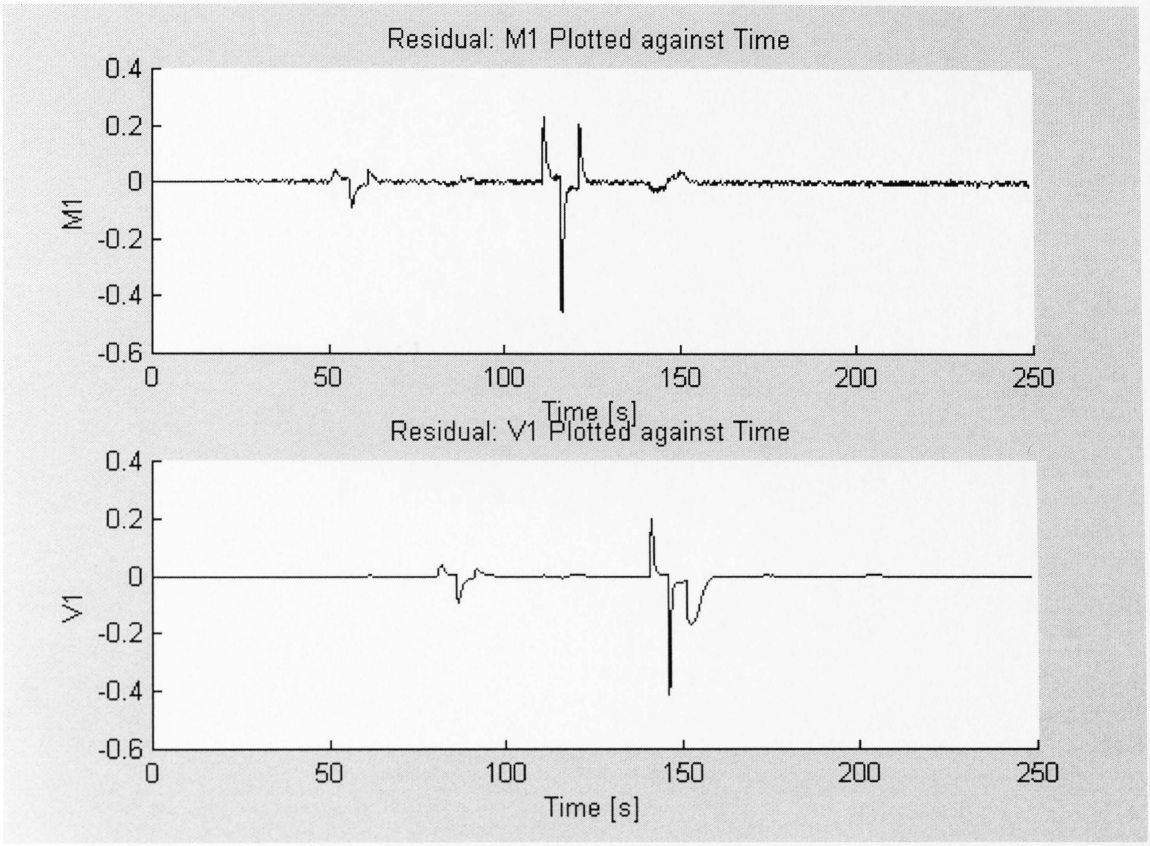


Figure F-4 APS Residuals M1 & V1 for Simulation 2

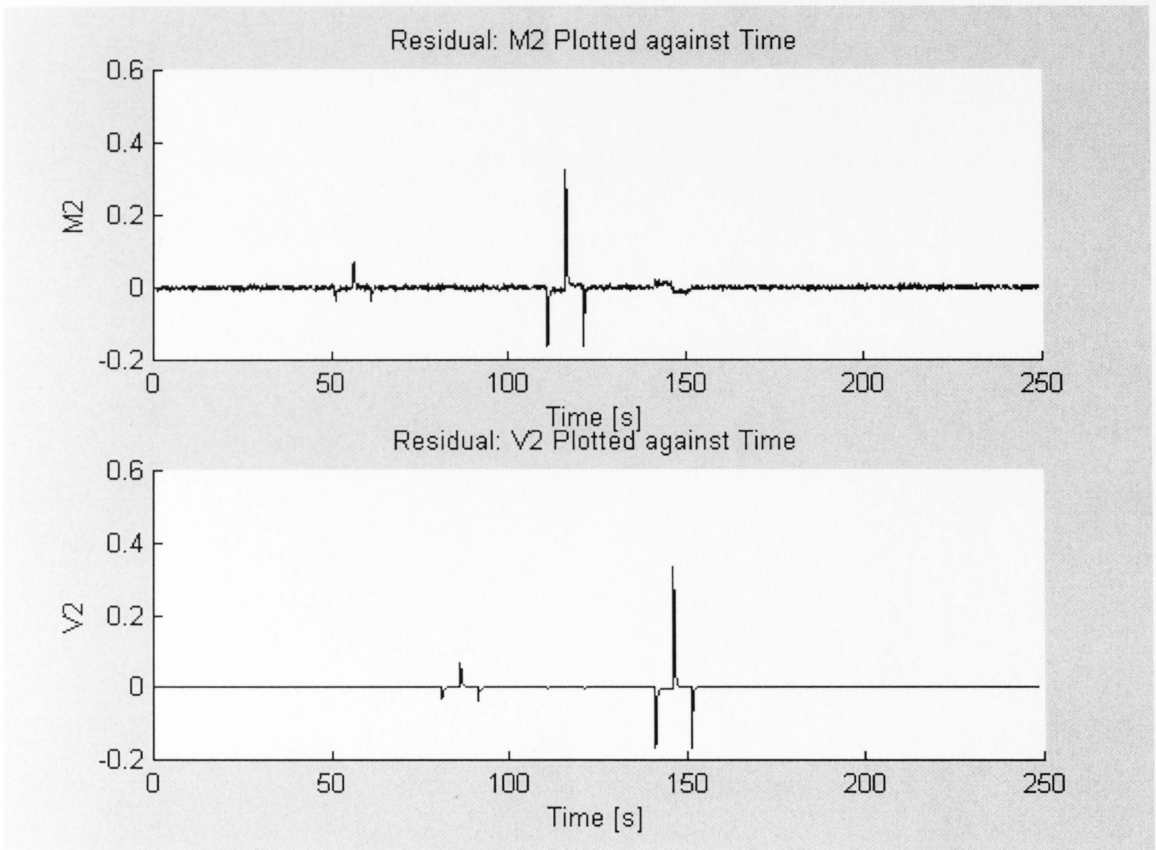


Figure F-5 APS Residuals M2 & V2 for Simulation 2

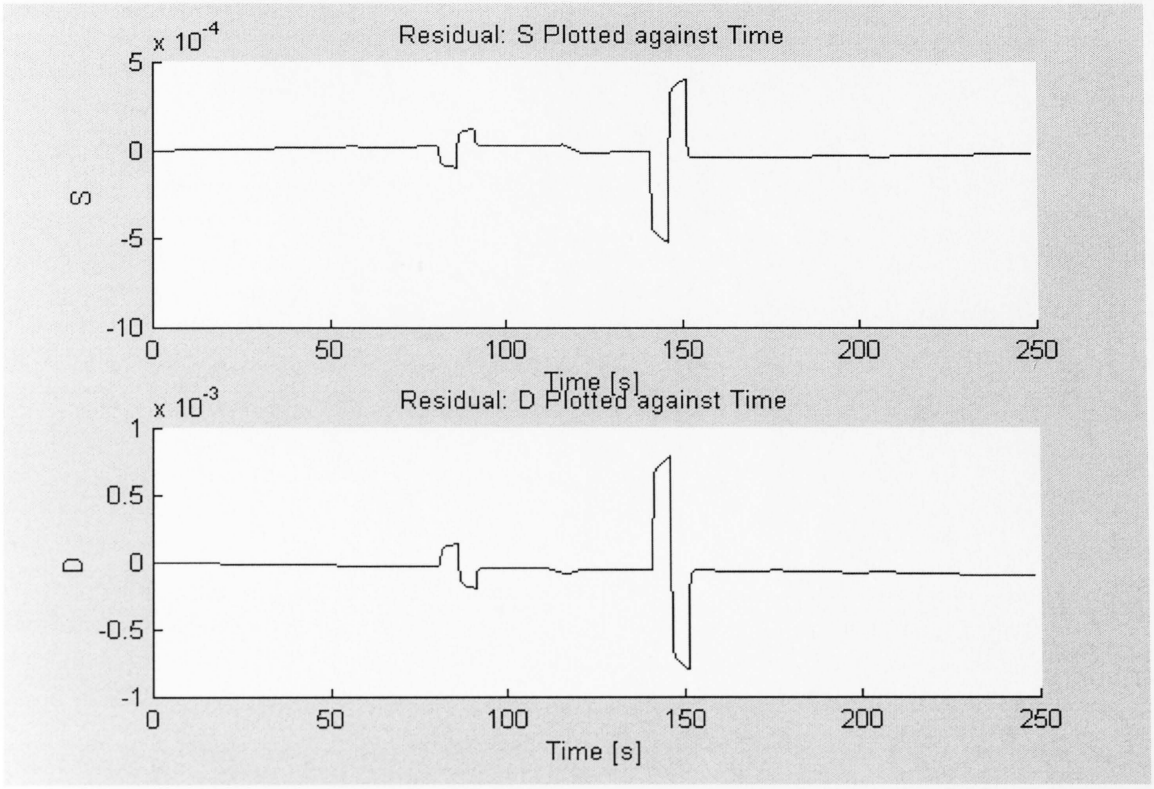


Figure F-6 APS Residuals S & D for Simulation 2

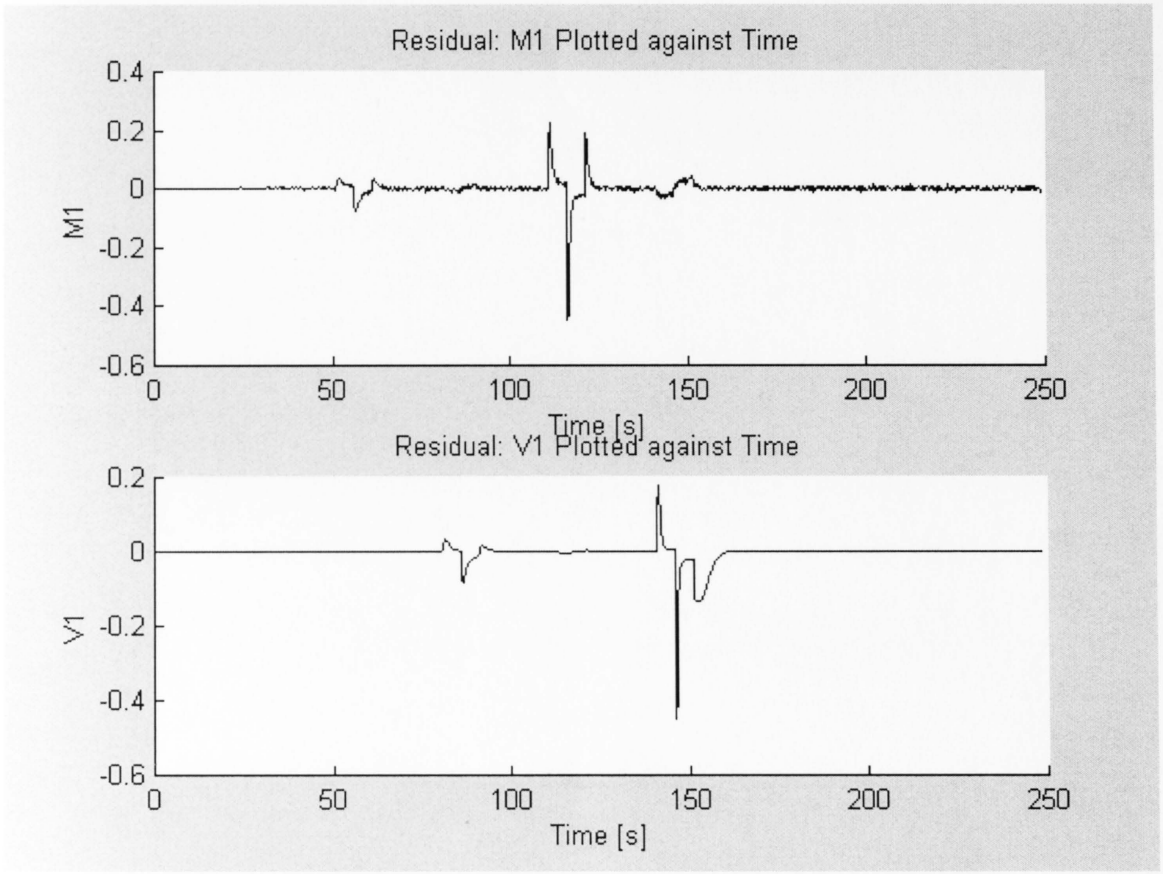


Figure F-7 APS Residuals M1 &V1 for Simulation 3

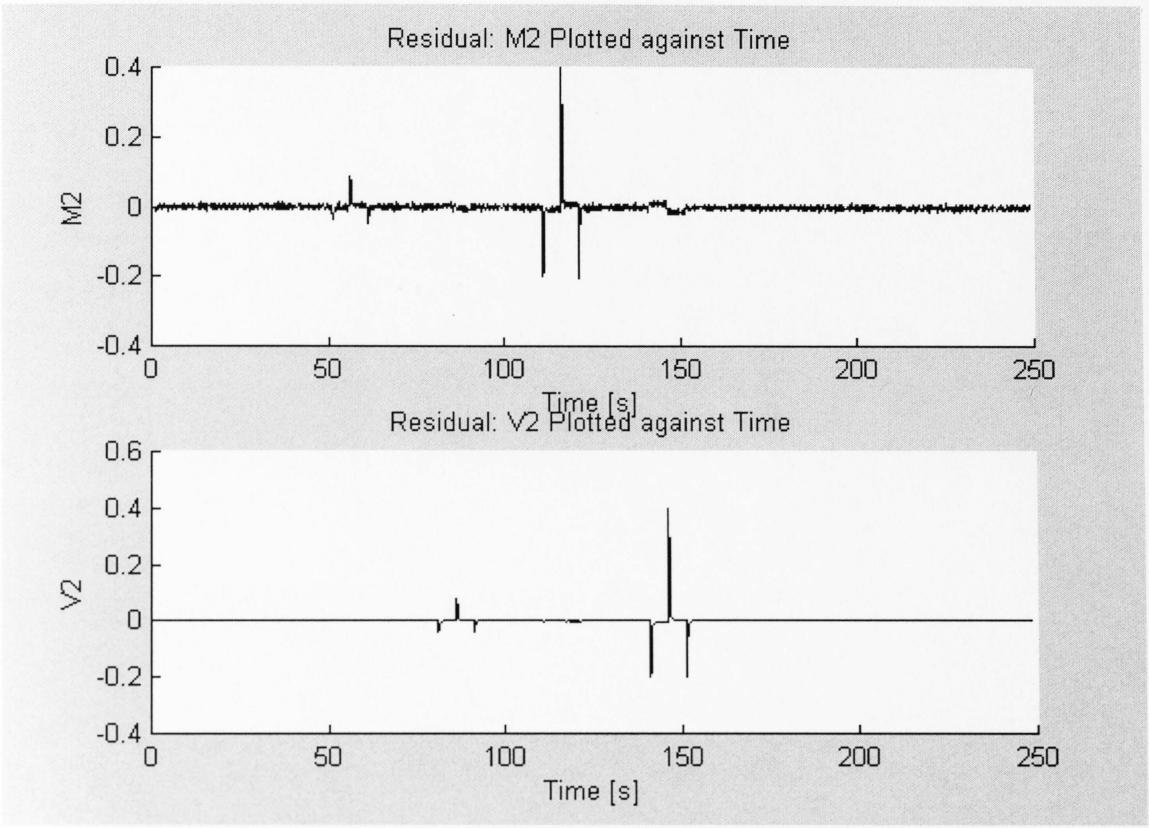


Figure F-8 APS Residuals M2 & V2 for Simulation 3

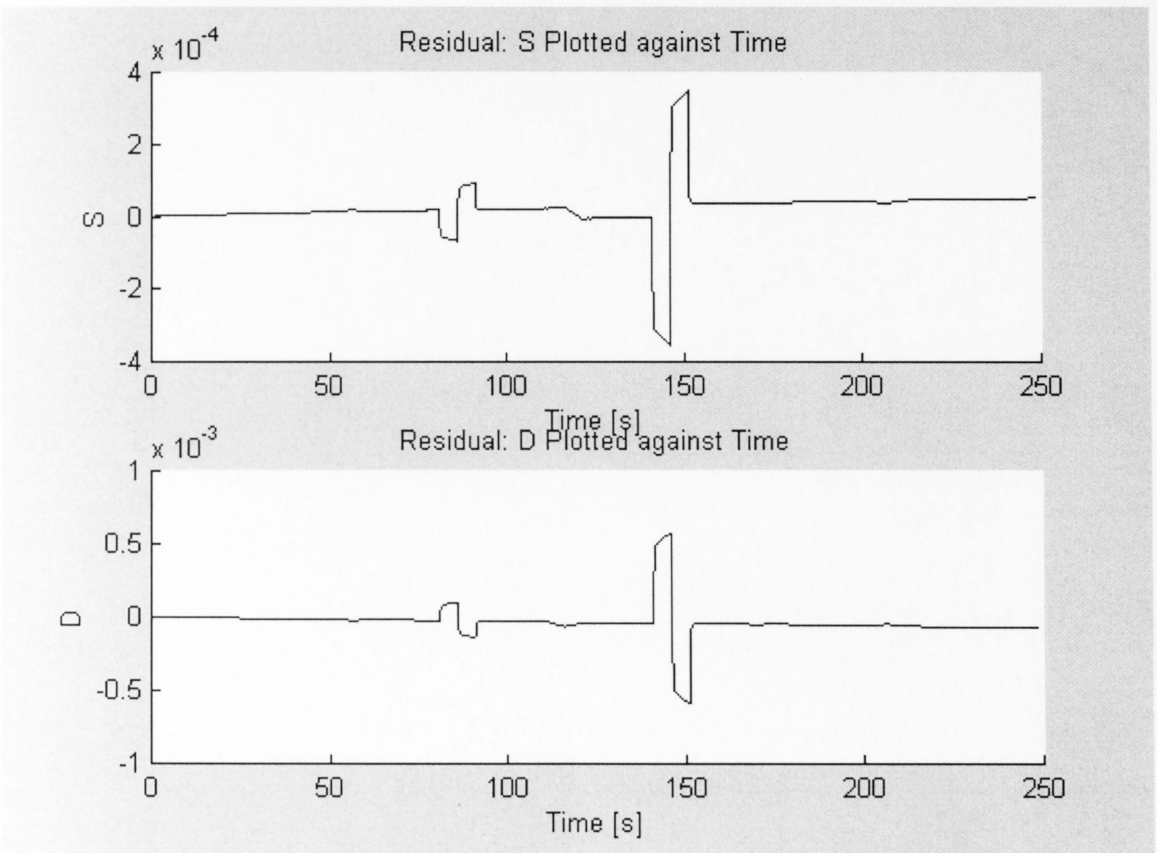


Figure F-9 APS Residuals S & D for Simulation 3

Comment:

As the value of $\lambda_{0.1s}$ decreases, the amplitudes of the deviations for Residuals M1, V1, M2 and V2 decrease (where faults have occurred), but the deviations of Residuals S and D begin to closely resemble the fault shape. The introduction of the forgetting factor also reduced the overshoot on Residuals S and D, from the normal RLS estimation. As a result, the optimum value for the forgetting factor was found to be 0.664. At this value, the deviations associated with Residuals M1, V1, M2 and V2 were not compromised significantly, and that Residuals S and D showed deviations to all the simulated faults, which was previously not the case in the RLS estimation, or with a large forgetting factor.

F.3 Determination of Stabilising Factor

The determination of the stabilising factor was also performed through trial and error, as with the determination of the forgetting factor. Although a range of stabilising factors were investigated, the residuals obtained for three different stabilising factors are shown in the following diagrams, the forgetting factor of 0.664 was used in order to obtain the simulation results.

Table F-2 Fault Parameters used for the Simulations

Simulation Number	α_{1s}	$\alpha_{0.1s}$
4	1.00×10^{-70}	1.00×10^{-7}
5	4.00×10^{-83}	5.75×10^{-9}
6	1.00×10^{-100}	1.00×10^{-10}

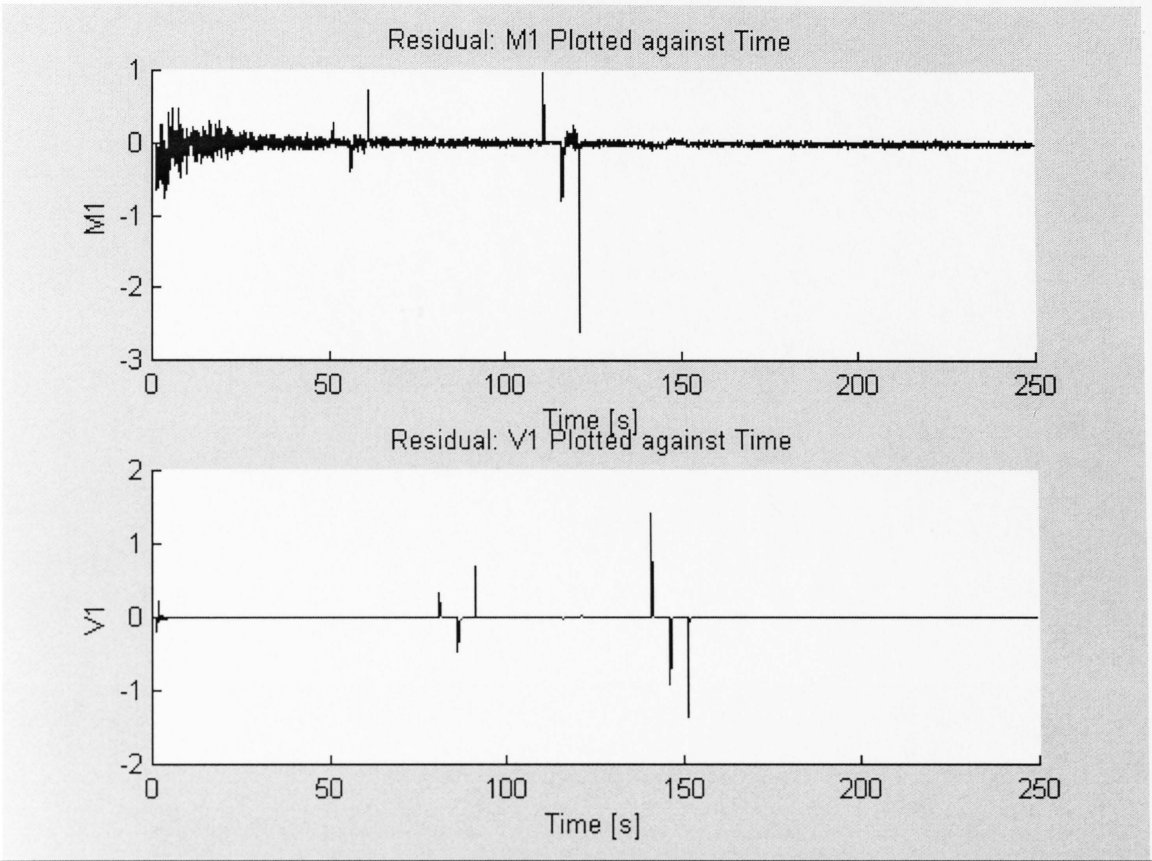


Figure F-10 APS Residuals M1 & V1 for Simulation 4

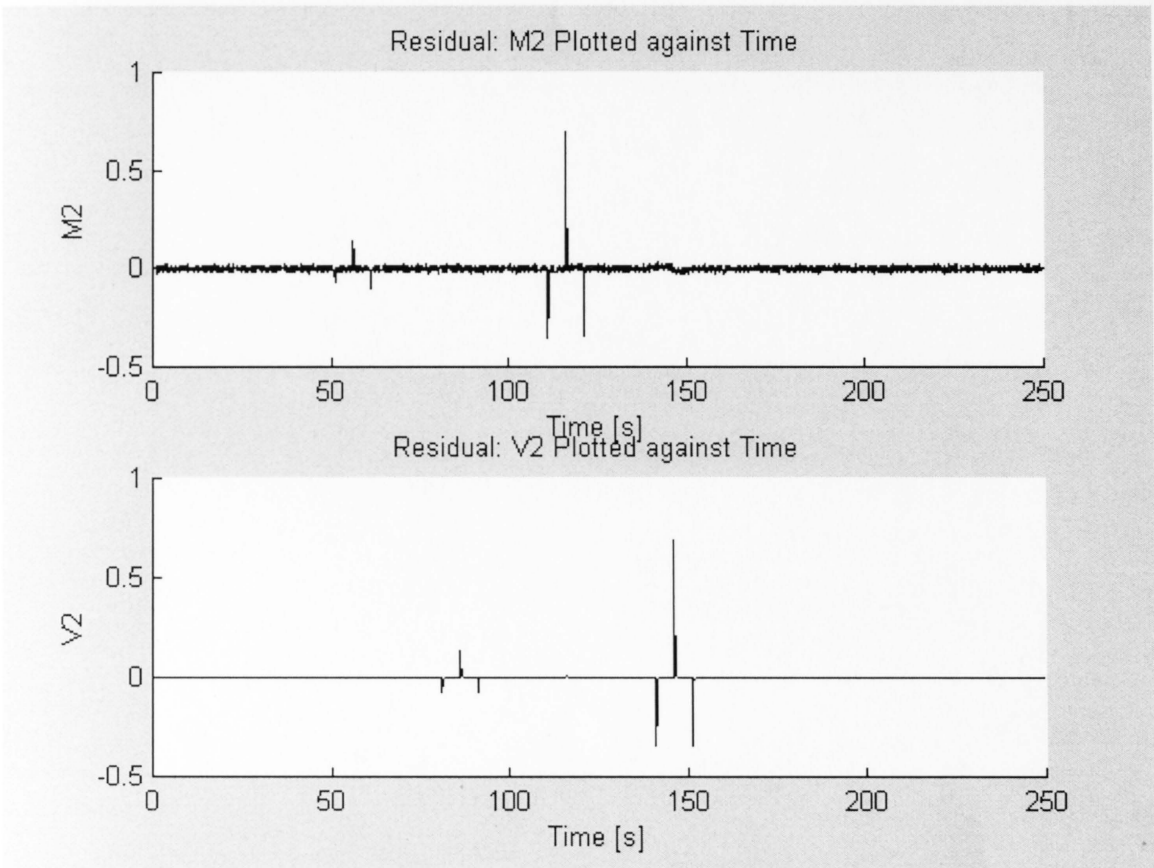


Figure F-11 APS Residuals M2 & V2 for Simulation 4

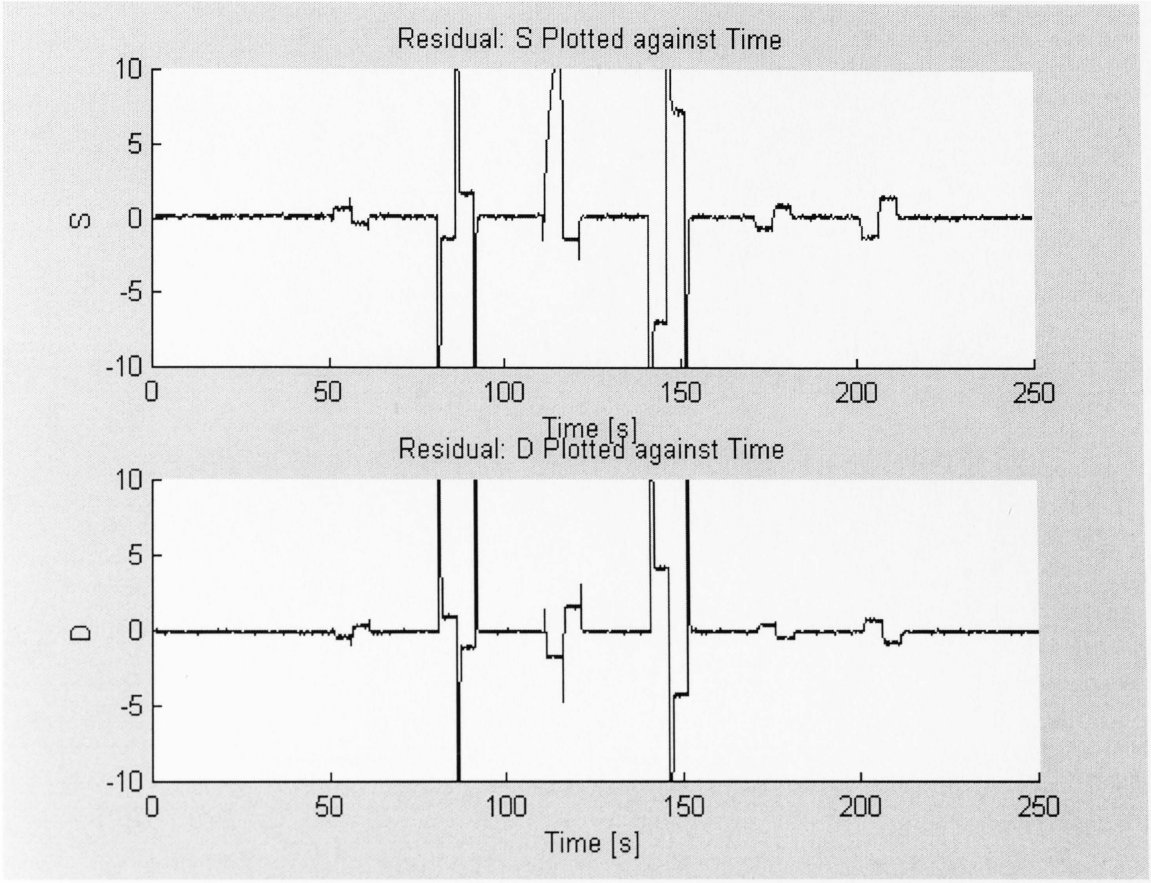


Figure F-12 APS Residuals D & S for Simulation 4

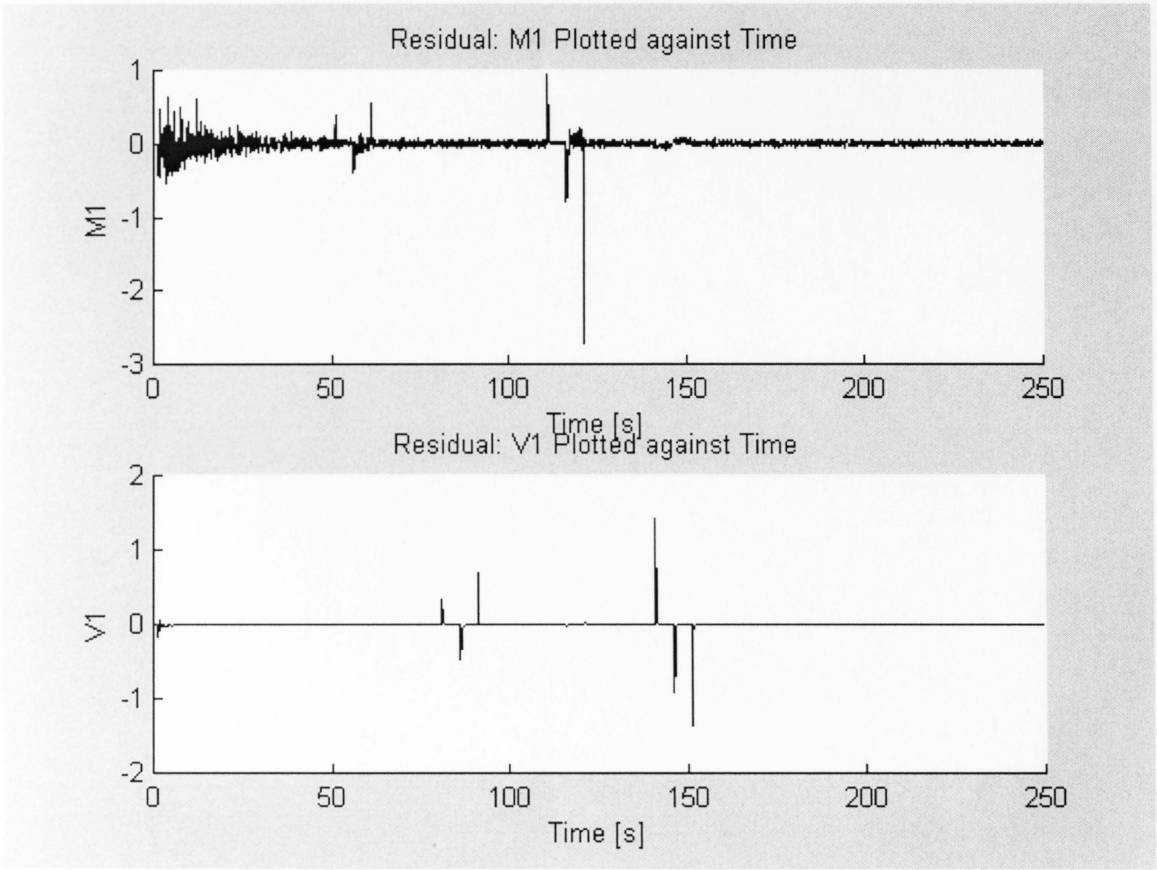


Figure F-13 APS Residuals M1 & V1 for Simulation 5

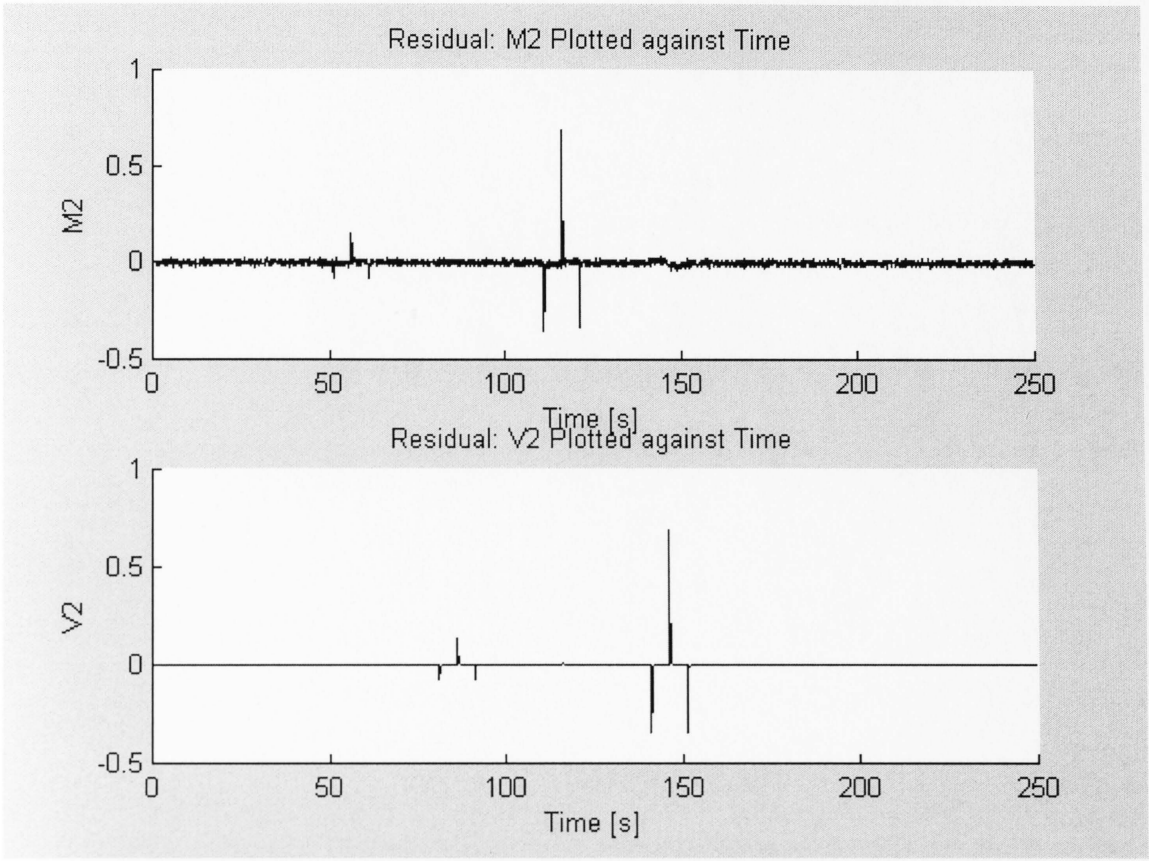


Figure F-14 APS Residuals M2 & V2 for Simulation 5

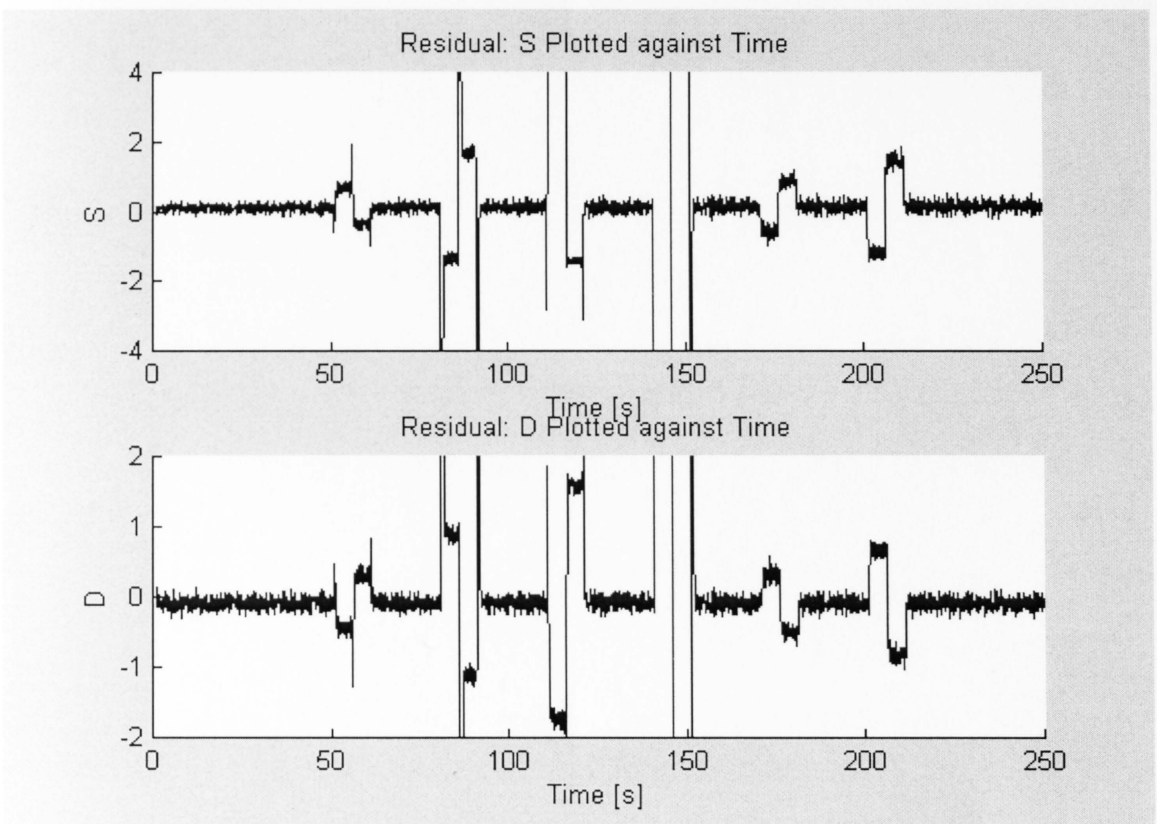


Figure F-15 APS Residuals D & S for Simulation 5

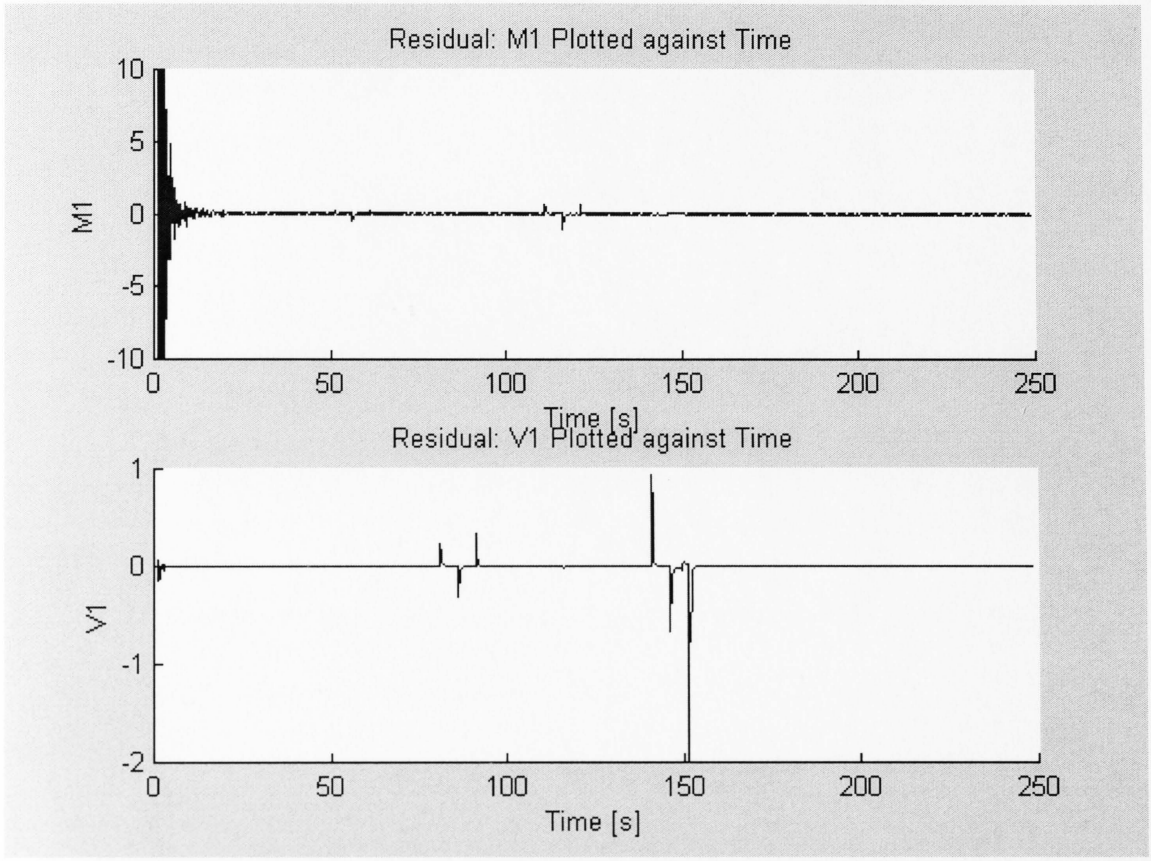


Figure F-16 APS Residuals M1 & V1 for Simulation 6

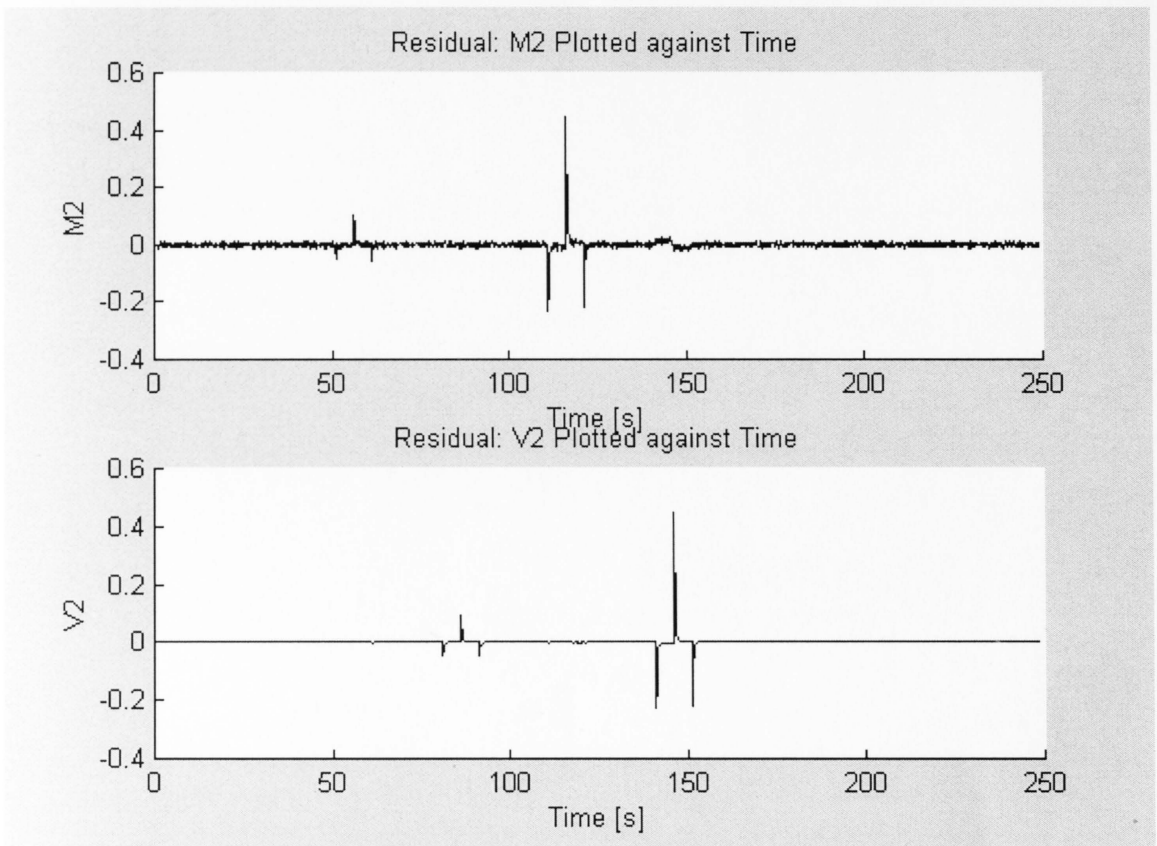


Figure F-17 APS Residuals M2 & V2 for Simulation 6

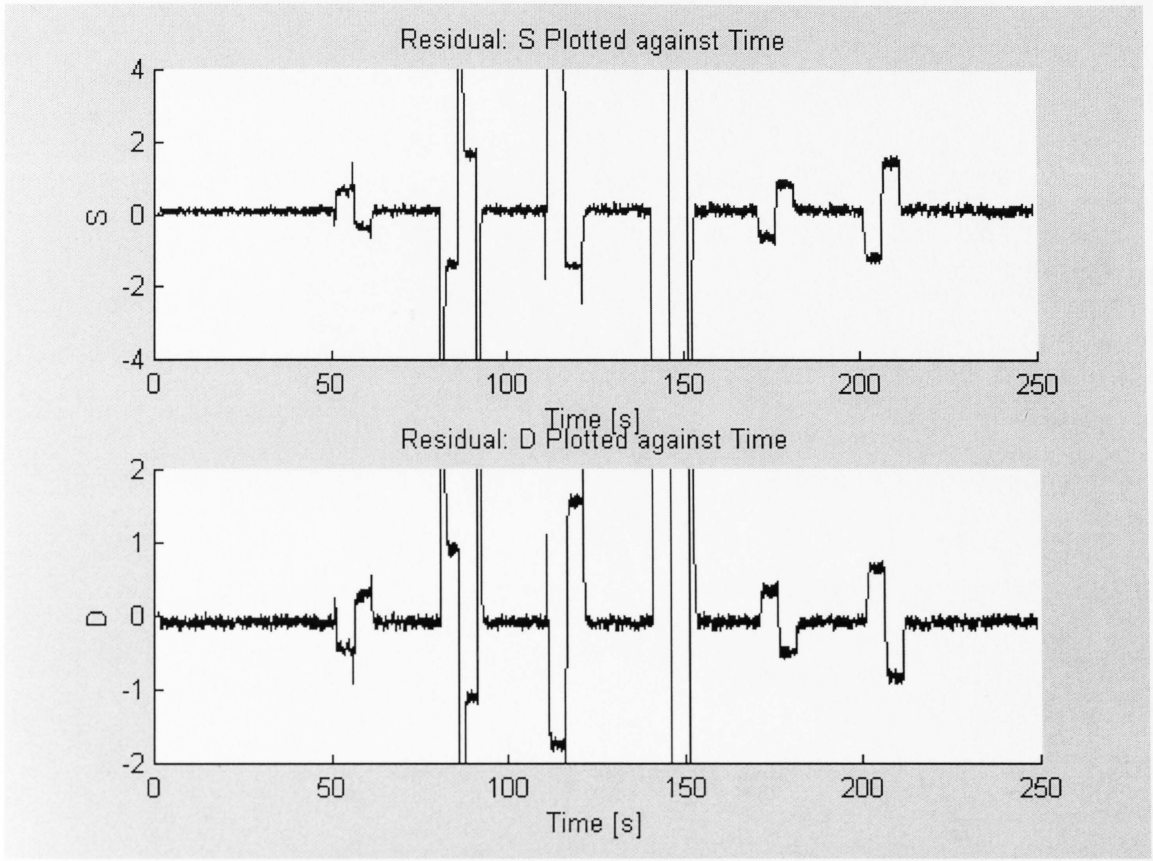


Figure F-18 APS Residuals D & S for Simulation 6

Comment:

As the value of $\alpha_{0.1s}$ decreases, it can be seen that there is an increase in the noise at the start of Residuals M1 and V1. Also, the amplitudes of the deviations for Residuals M1, V1, M2 and V2 decrease (where faults have occurred), but the deviations of Residuals S and D begin to closely resemble the fault shape. The introduction of the stabilising factor also reduced the overshoot on Residuals S and D, from the RLS estimation, with λ. As a result, the optimum value for the stabilising factor was found to be 5.75×10^{-83}. At this value, the deviations associated with Residuals M1, V1, M2 and V2 were not compromised significantly, and Residuals S and D showed good deviations to all the simulated faults.

F.4 Effect of Forgetting and Stabilising Factors on APS Performance

As seen form the previous sections, the performance of the adaptive parity space method of fault detection is extremely dependent on the values of λ and α. It is thus important to view the effect that changing the values of λ and α has on the performance of the fault detection method. The simulation and fault parameters used for the following simulations are described in Section 5.1. The thresholds used on the residuals are those indicated in Section 5.3.4. In the following diagram, the values for λ and α are varied between 0 and 1.

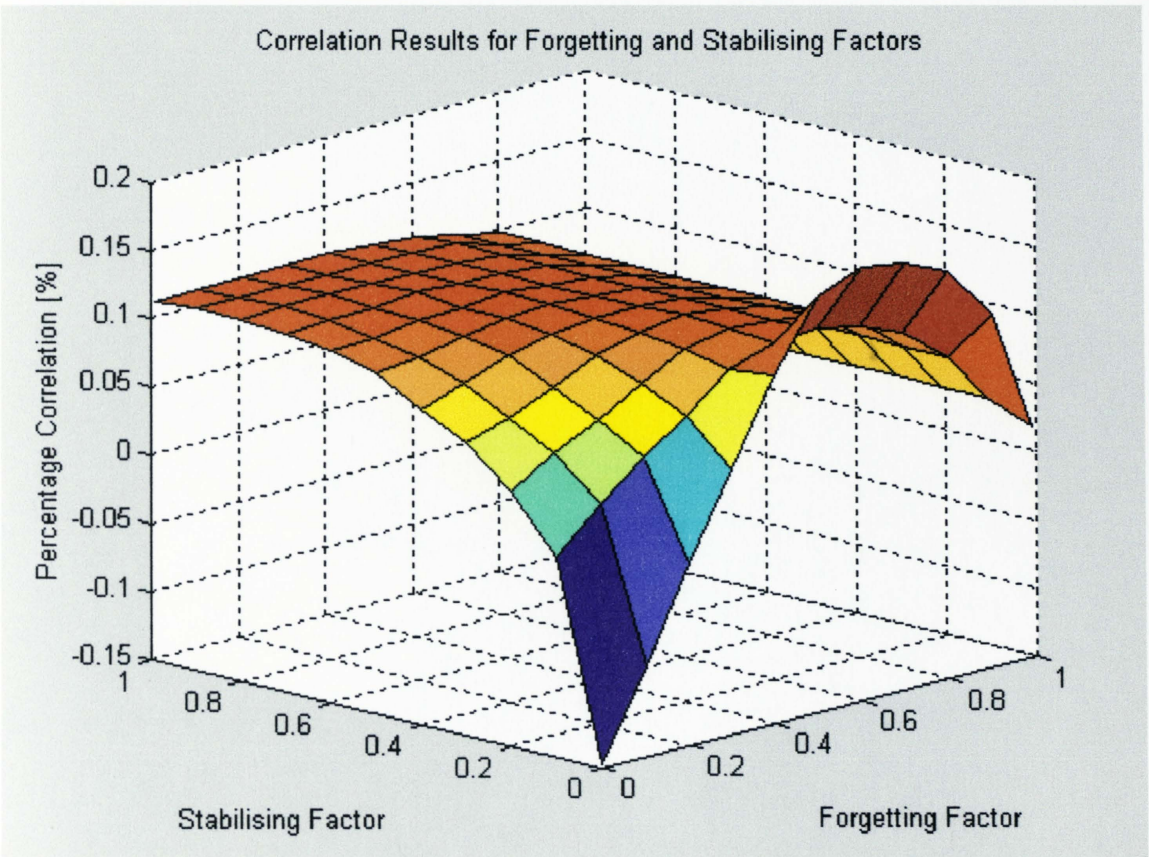


Figure F-19 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation

Comment:

The graph shows that the best correlation result is obtained at a forgetting factor of 0.6 and stabilising factor value of 0. This proves that the choice of λ is correct, but disproves the choice of α being 5.75×10^{-9}. It is necessary to analyse the effect of α in a closer data range. The following graph shows the percentage of correlation obtained for $0\leq\lambda\leq1$, and $0\leq\alpha\leq0.1$.

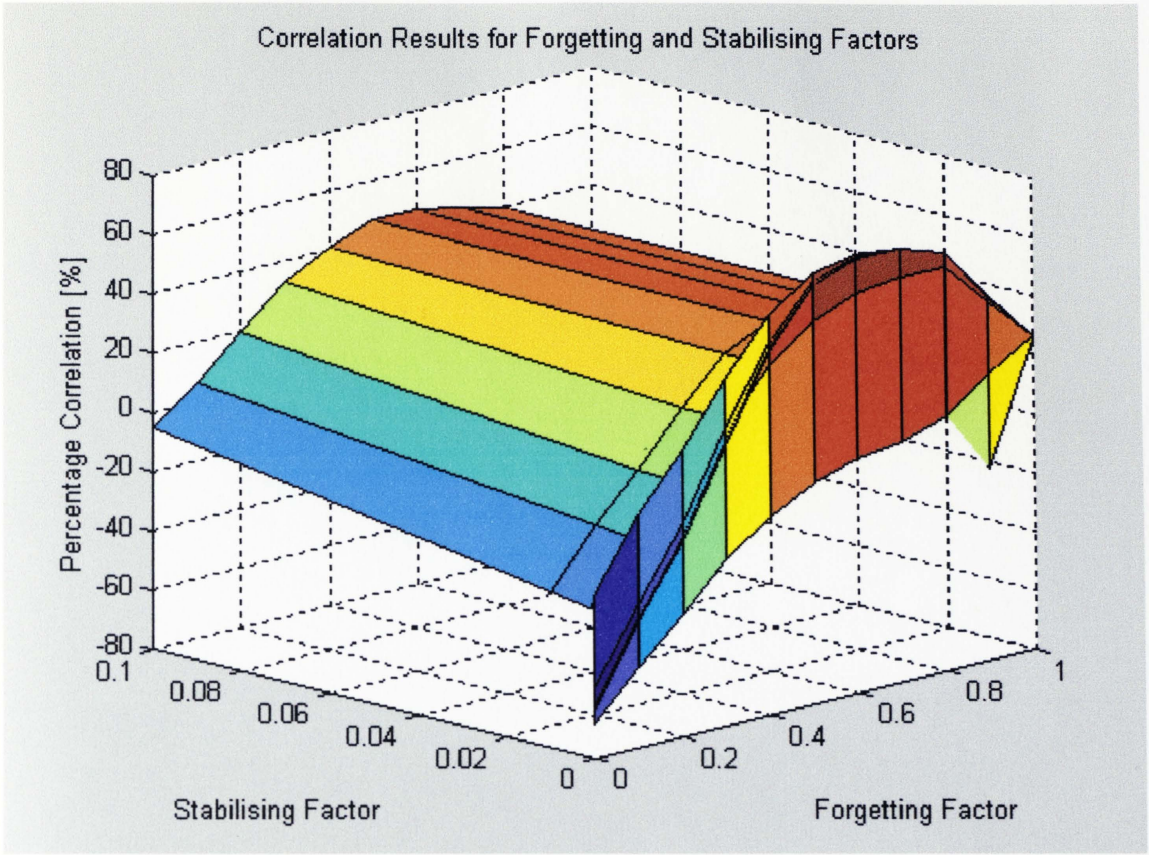


Figure F-20 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (zoomed)

From the above diagram, it is seen that there are a lot of initial variations in the percentage of correlation, for varied stabilising factors. Also, Figure F-20 shows that the relationship between the stabilising and forgetting factors remains the same. Furthermore The following graph is a contour map, showing the percentage of correlation for $0 \leq \alpha \leq 0.0001$. Following the contour map is another scaled correlation graph. Both graphs vary with the forgetting factor in the region, $0 \leq \lambda \leq 1$.

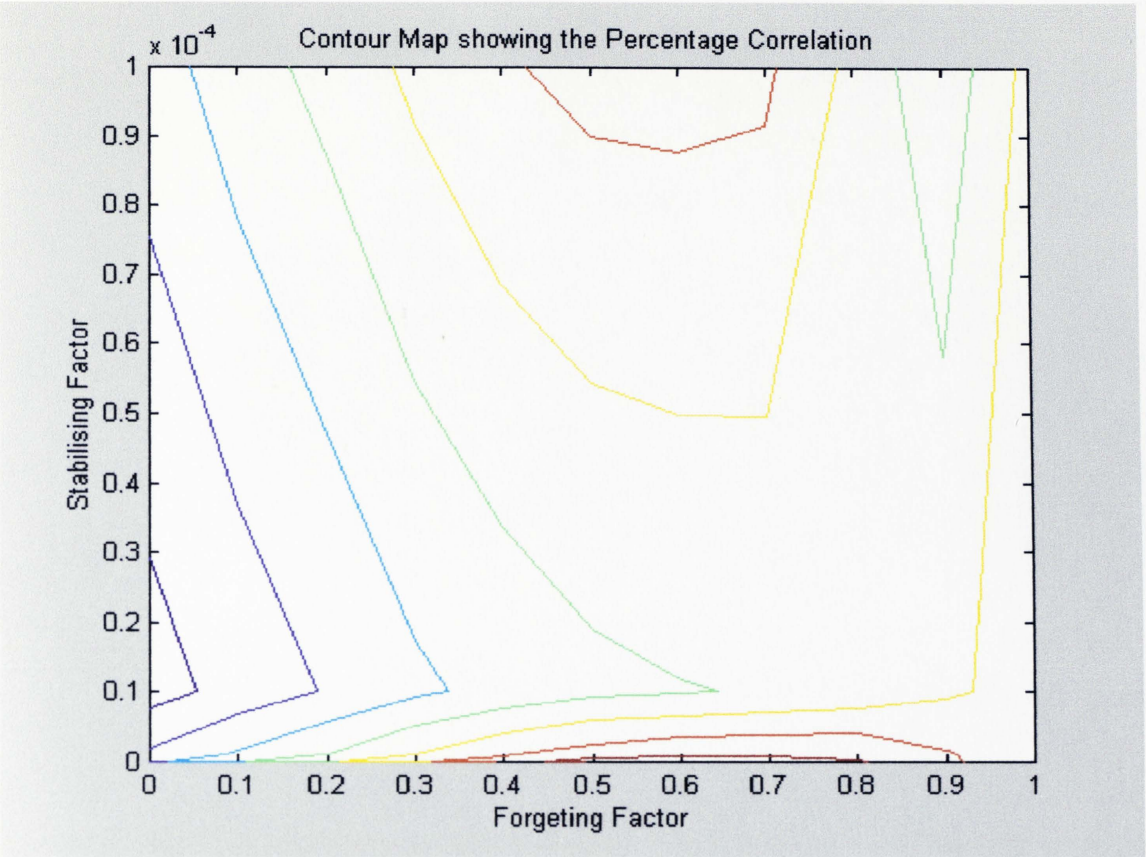


Figure F-21 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (contour map)

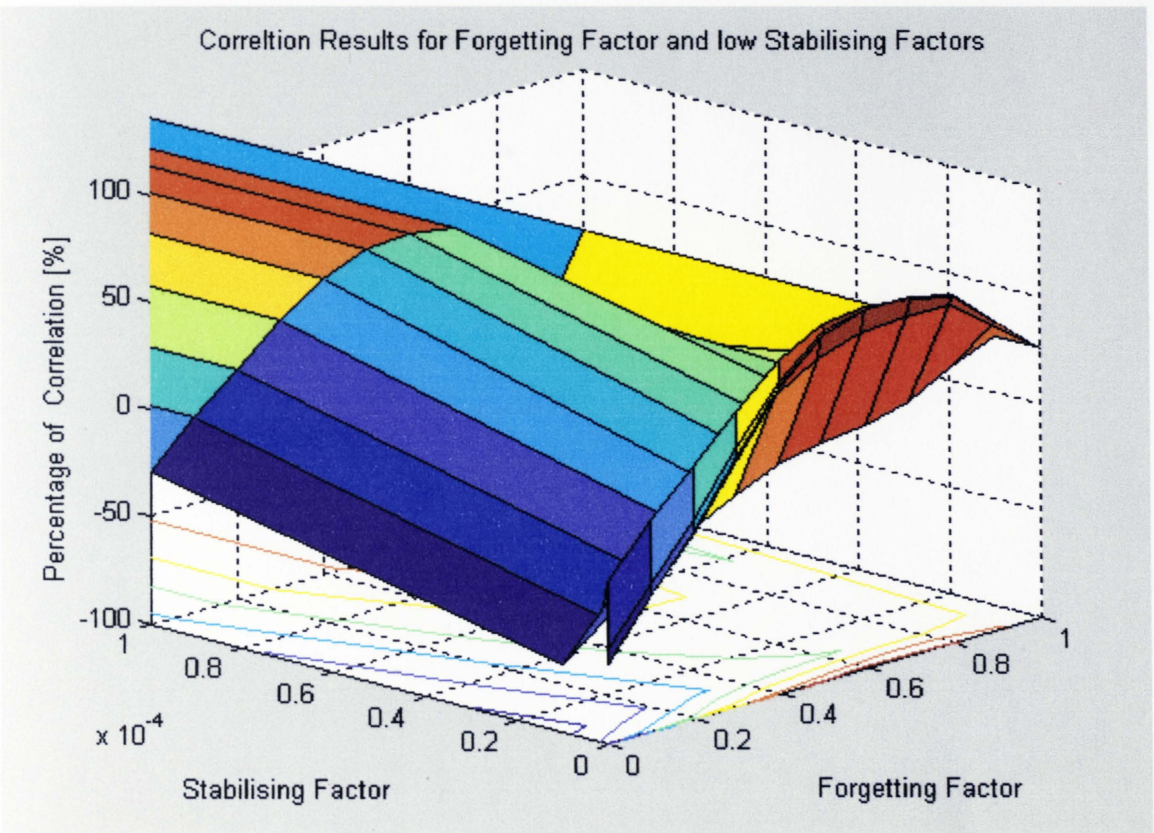


Figure F-22 Effect of Stabilising and Forgetting Factors on the Percentage of Correlation (zoomed with contours)

Comment:

From the previous two graphs, it can be seen that the percentage of correlation is low for α equal to zero for almost all λ, except at the peak, where the correlation values seem to be similar. For $\alpha \geq 0$, the percentages of correlation increase initially, then decrease, then increase, and finally the values become steady. For $\lambda=1$, a higher correlation percentage is seen than for $0.7 \leq \lambda \leq 1$. These observations cannot be explained, and could be subject matter for future research.

APPENDIX G: CORRELATION RESULTS FOR VARYING FAULT PARAMETERS FOR THE FAULT DETECTION METHODS

The following percentages of correlation for the different fault detection methods are shown below, where:

A = Fault Amplitude [V]

W = Fault Duration [10*s]

The shapes of the waveforms are highlighted.

Ramp:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 79.7222
Percentage Correlation Parity Space: 37.2222
Percentage Correlation Observer: 65.9444
W =
200
Percentage Correlation Adaptive Parity Space: 71
Percentage Correlation Parity Space: 39
Percentage Correlation Observer: 25
W =
100
Percentage Correlation Adaptive Parity Space: 53.8333
Percentage Correlation Parity Space: 38.1667
Percentage Correlation Observer: -72.5
W =
50
Percentage Correlation Adaptive Parity Space: 19
Percentage Correlation Parity Space: 24.6667
Percentage Correlation Observer: -215
W =
30
Percentage Correlation Adaptive Parity Space: -29.4444
Percentage Correlation Parity Space: 10
Percentage Correlation Observer: -347.2222

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 78.6111
Percentage Correlation Parity Space: 23.7778

```
Percentage Correlation Observer: 64.3333
W =
  200
Percentage Correlation Adaptive Parity Space: 72.1667
Percentage Correlation Parity Space: 25.6667
Percentage Correlation Observer: 21.9167
W =
  100
Percentage Correlation Adaptive Parity Space: 52.6667
Percentage Correlation Parity Space: 29.5
Percentage Correlation Observer: -57.1667
W =
   50
Percentage Correlation Adaptive Parity Space: 17
Percentage Correlation Parity Space: 25.3333
Percentage Correlation Observer: -145.3333
W =
   30
Percentage Correlation Adaptive Parity Space: -27.7778
Percentage Correlation Parity Space: 10
Percentage Correlation Observer: -240.5556

A =
    1
W =
  300
Percentage Correlation Adaptive Parity Space: 70.1111
Percentage Correlation Parity Space: 11.5556
Percentage Correlation Observer: 56.6111
W =
  200
Percentage Correlation Adaptive Parity Space: 63.25
Percentage Correlation Parity Space: 14.3333
Percentage Correlation Observer: 26
W =
  100
Percentage Correlation Adaptive Parity Space: 44.6667
Percentage Correlation Parity Space: 15
Percentage Correlation Observer: -18.3333
W =
   50
Percentage Correlation Adaptive Parity Space: 14
Percentage Correlation Parity Space: 11.6667
Percentage Correlation Observer: -86.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -22.2222
Percentage Correlation Parity Space: 7.2222
Percentage Correlation Observer: -137.2222

A =
   0.5000
W =
  300
Percentage Correlation Adaptive Parity Space: 56.8889
Percentage Correlation Parity Space: 2.6667
Percentage Correlation Observer: 48.3333
W =
  200
Percentage Correlation Adaptive Parity Space: 50.0833
```

Percentage Correlation Parity Space: 4.9167
Percentage Correlation Observer: 24.1667
W =
  100
Percentage Correlation Adaptive Parity Space: 35.5
Percentage Correlation Parity Space: 10.5
Percentage Correlation Observer: -6.1667
W =
  50
Percentage Correlation Adaptive Parity Space: 10.3333
Percentage Correlation Parity Space: 8.6667
Percentage Correlation Observer: -48
W =
  30
Percentage Correlation Adaptive Parity Space: -20.5556
Percentage Correlation Parity Space: 1.6667
Percentage Correlation Observer: -88.8889

A =
  0.2500
W =
  300
Percentage Correlation Adaptive Parity Space: 41.8889
Percentage Correlation Parity Space: 0
Percentage Correlation Observer: 29.3333
W =
  200
Percentage Correlation Adaptive Parity Space: 38.4167
Percentage Correlation Parity Space: 0
Percentage Correlation Observer: 22.1667
W =
  100
Percentage Correlation Adaptive Parity Space: 28.6667
Percentage Correlation Parity Space: 3
Percentage Correlation Observer: 3.1667
W =
  50
Percentage Correlation Adaptive Parity Space: 8.3333
Percentage Correlation Parity Space: 5
Percentage Correlation Observer: -25
W =
  30
Percentage Correlation Adaptive Parity Space: -13.3333
Percentage Correlation Parity Space: 1.1111
Percentage Correlation Observer: -55.5556

Saw:

A =
  5
W =
  300
Percentage Correlation Adaptive Parity Space: 84.4444
Percentage Correlation Parity Space: 38.6667
Percentage Correlation Observer: 75.1667
W =
  200
Percentage Correlation Adaptive Parity Space: 78.1667

Percentage Correlation Parity Space: 41.8333
Percentage Correlation Observer: 35.8333
W =
  100
Percentage Correlation Adaptive Parity Space: 67.5
Percentage Correlation Parity Space: 41
Percentage Correlation Observer: -54
W =
  50
Percentage Correlation Adaptive Parity Space: 35
Percentage Correlation Parity Space: 30.3333
Percentage Correlation Observer: -193
W =
  30
Percentage Correlation Adaptive Parity Space: -8.8889
Percentage Correlation Parity Space: 16.1111
Percentage Correlation Observer: -325.5556

A =
 2.5000e+000
W =
  300
Percentage Correlation Adaptive Parity Space: 85.9444
Percentage Correlation Parity Space: 25.2222
Percentage Correlation Observer: 74.2778
W =
  200
Percentage Correlation Adaptive Parity Space: 81.8333
Percentage Correlation Parity Space: 29.4167
Percentage Correlation Observer: 40.4167
W =
  100
Percentage Correlation Adaptive Parity Space: 66.6667
Percentage Correlation Parity Space: 37.3333
Percentage Correlation Observer: -31.5
W =
  50
Percentage Correlation Adaptive Parity Space: 36
Percentage Correlation Parity Space: 29
Percentage Correlation Observer: -120.6667
W =
  30
Percentage Correlation Adaptive Parity Space: -7.7778
Percentage Correlation Parity Space: 13.8889
Percentage Correlation Observer: -209.4444

A =
   1
W =
  300
Percentage Correlation Adaptive Parity Space: 77
Percentage Correlation Parity Space: 14.9444
Percentage Correlation Observer: 65.3333
W =
  200
Percentage Correlation Adaptive Parity Space: 72
Percentage Correlation Parity Space: 17.1667
Percentage Correlation Observer: 45
W =
  100

Percentage Correlation Adaptive Parity Space: 59.3333
Percentage Correlation Parity Space: 20.5
Percentage Correlation Observer: 8.8333
W =
50
Percentage Correlation Adaptive Parity Space: 32.6667
Percentage Correlation Parity Space: 24.3333
Percentage Correlation Observer: -57.3333
W =
30
Percentage Correlation Adaptive Parity Space: -4.4444
Percentage Correlation Parity Space: 12.7778
Percentage Correlation Observer: -126.1111

A =
5.0000e-001
W =
300
Percentage Correlation Adaptive Parity Space: 62.1667
Percentage Correlation Parity Space: 5.9444
Percentage Correlation Observer: 56.4444
W =
200
Percentage Correlation Adaptive Parity Space: 57.9167
Percentage Correlation Parity Space: 10.75
Percentage Correlation Observer: 40.4167
W =
100
Percentage Correlation Adaptive Parity Space: 49.5
Percentage Correlation Parity Space: 13.5
Percentage Correlation Observer: 10.1667
W =
50
Percentage Correlation Adaptive Parity Space: 31.3333
Percentage Correlation Parity Space: 12.6667
Percentage Correlation Observer: -36.6667
W =
30
Percentage Correlation Adaptive Parity Space: 1.1111
Percentage Correlation Parity Space: 12.7778
Percentage Correlation Observer: -77.7778

A =
2.5000e-001
W =
300
Percentage Correlation Adaptive Parity Space: 47.0556
Percentage Correlation Parity Space: 1.3333
Percentage Correlation Observer: 37.8889
W =
200
Percentage Correlation Adaptive Parity Space: 44.75
Percentage Correlation Parity Space: 3
Percentage Correlation Observer: 30.6667
W =
100
Percentage Correlation Adaptive Parity Space: 37.8333
Percentage Correlation Parity Space: 9.6667
Percentage Correlation Observer: 12
W =

50
Percentage Correlation Adaptive Parity Space: 21.6667
Percentage Correlation Parity Space: 10.6667
Percentage Correlation Observer: -16.3333
W =
30
Percentage Correlation Adaptive Parity Space: 0
Percentage Correlation Parity Space: 7.2222
Percentage Correlation Observer: -45.5556

Sine:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 87.8889
Percentage Correlation Parity Space: 45.8889
Percentage Correlation Observer: 71.6111
W =
200
Percentage Correlation Adaptive Parity Space: 82.25
Percentage Correlation Parity Space: 44.5
Percentage Correlation Observer: 35.5
W =
100
Percentage Correlation Adaptive Parity Space: 66
Percentage Correlation Parity Space: 39.5
Percentage Correlation Observer: -45.5
W =
50
Percentage Correlation Adaptive Parity Space: 33.6667
Percentage Correlation Parity Space: 28.3333
Percentage Correlation Observer: -99.3333
W =
30
Percentage Correlation Adaptive Parity Space: -15
Percentage Correlation Parity Space: 15
Percentage Correlation Observer: -145.5556

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 86.5556
Percentage Correlation Parity Space: 37.2778
Percentage Correlation Observer: 73.0556
W =
200
Percentage Correlation Adaptive Parity Space: 80.4167
Percentage Correlation Parity Space: 38.8333
Percentage Correlation Observer: 36.8333
W =
100
Percentage Correlation Adaptive Parity Space: 63.8333
Percentage Correlation Parity Space: 36.5
Percentage Correlation Observer: -23.1667
W =

```
   50
Percentage Correlation Adaptive Parity Space: 29.3333
Percentage Correlation Parity Space: 27
Percentage Correlation Observer: -58.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -9.4444
Percentage Correlation Parity Space: 14.4444
Percentage Correlation Observer: -102.2222

A =
    1
W =
  300
Percentage Correlation Adaptive Parity Space: 79.2778
Percentage Correlation Parity Space: 19.5556
Percentage Correlation Observer: 65.8333
W =
  200
Percentage Correlation Adaptive Parity Space: 71
Percentage Correlation Parity Space: 24.4167
Percentage Correlation Observer: 37.25
W =
  100
Percentage Correlation Adaptive Parity Space: 55
Percentage Correlation Parity Space: 29.8333
Percentage Correlation Observer: -0.16667
W =
   50
Percentage Correlation Adaptive Parity Space: 28.3333
Percentage Correlation Parity Space: 23.3333
Percentage Correlation Observer: -37
W =
   30
Percentage Correlation Adaptive Parity Space: -6.1111
Percentage Correlation Parity Space: 13.8889
Percentage Correlation Observer: -71.6667

A =
   0.5000
W =
  300
Percentage Correlation Adaptive Parity Space: 67.3333
Percentage Correlation Parity Space: 12.7222
Percentage Correlation Observer: 52.1667
W =
  200
Percentage Correlation Adaptive Parity Space: 65
Percentage Correlation Parity Space: 13.8333
Percentage Correlation Observer: 33.4167
W =
  100
Percentage Correlation Adaptive Parity Space: 52.3333
Percentage Correlation Parity Space: 18.6667
Percentage Correlation Observer: 1.6667
W =
   50
Percentage Correlation Adaptive Parity Space: 25.6667
Percentage Correlation Parity Space: 20.3333
Percentage Correlation Observer: -26.3333
```

W =
30
Percentage Correlation Adaptive Parity Space: -8.8889
Percentage Correlation Parity Space: 12.2222
Percentage Correlation Observer: -47.7778

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 56.1667
Percentage Correlation Parity Space: 6.2778
Percentage Correlation Observer: 42.0556
W =
200
Percentage Correlation Adaptive Parity Space: 53.5833
Percentage Correlation Parity Space: 9.4167
Percentage Correlation Observer: 21.5833
W =
100
Percentage Correlation Adaptive Parity Space: 44.1667
Percentage Correlation Parity Space: 11.5
Percentage Correlation Observer: 10.5
W =
50
Percentage Correlation Adaptive Parity Space: 22.6667
Percentage Correlation Parity Space: 12.3333
Percentage Correlation Observer: -7.6667
W =
30
Percentage Correlation Adaptive Parity Space: -9.4444
Percentage Correlation Parity Space: 10.5556
Percentage Correlation Observer: -20

Square:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 87.7778
Percentage Correlation Parity Space: 44.2222
Percentage Correlation Observer: 70.7778
W =
200
Percentage Correlation Adaptive Parity Space: 81.9167
Percentage Correlation Parity Space: 42.1667
Percentage Correlation Observer: 27.6667
W =
100
Percentage Correlation Adaptive Parity Space: 67.1667
Percentage Correlation Parity Space: 36
Percentage Correlation Observer: -59.1667
W =
50
Percentage Correlation Adaptive Parity Space: 29.6667
Percentage Correlation Parity Space: 27
Percentage Correlation Observer: -137
W =

30
Percentage Correlation Adaptive Parity Space: -16.6667
Percentage Correlation Parity Space: 13.3333
Percentage Correlation Observer: -182.2222

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 88.5
Percentage Correlation Parity Space: 29.9444
Percentage Correlation Observer: 71.3889
W =
200
Percentage Correlation Adaptive Parity Space: 82.8333
Percentage Correlation Parity Space: 28.6667
Percentage Correlation Observer: 35.4167
W =
100
Percentage Correlation Adaptive Parity Space: 68.3333
Percentage Correlation Parity Space: 24.8333
Percentage Correlation Observer: -36.6667
W =
50
Percentage Correlation Adaptive Parity Space: 29.3333
Percentage Correlation Parity Space: 20.3333
Percentage Correlation Observer: -79.6667
W =
30
Percentage Correlation Adaptive Parity Space: -8.8889
Percentage Correlation Parity Space: 11.6667
Percentage Correlation Observer: -123.3333

A =
1
W =
300
Percentage Correlation Adaptive Parity Space: 85.2778
Percentage Correlation Parity Space: 10.9444
Percentage Correlation Observer: 69.9444
W =
200
Percentage Correlation Adaptive Parity Space: 75.75
Percentage Correlation Parity Space: 11.3333
Percentage Correlation Observer: 33.75
W =
100
Percentage Correlation Adaptive Parity Space: 59.5
Percentage Correlation Parity Space: 9.8333
Percentage Correlation Observer: -3
W =
50
Percentage Correlation Adaptive Parity Space: 28.3333
Percentage Correlation Parity Space: 10.6667
Percentage Correlation Observer: -42
W =
30
Percentage Correlation Adaptive Parity Space: -6.1111
Percentage Correlation Parity Space: 6.6667
Percentage Correlation Observer: -86.1111

A =
0.5000
W =
300
Percentage Correlation Adaptive Parity Space: 75.6667
Percentage Correlation Parity Space: 3
Percentage Correlation Observer: 56.2778
W =
200
Percentage Correlation Adaptive Parity Space: 70.25
Percentage Correlation Parity Space: 2.6667
Percentage Correlation Observer: 32
W =
100
Percentage Correlation Adaptive Parity Space: 58.3333
Percentage Correlation Parity Space: 3.5
Percentage Correlation Observer: 1.6667
W =
50
Percentage Correlation Adaptive Parity Space: 29.6667
Percentage Correlation Parity Space: 6
Percentage Correlation Observer: -36.6667
W =
30
Percentage Correlation Adaptive Parity Space: -8.3333
Percentage Correlation Parity Space: 5.5556
Percentage Correlation Observer: -63.3333

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 65.6667
Percentage Correlation Parity Space: 0.88889
Percentage Correlation Observer: 40.2222
W =
200
Percentage Correlation Adaptive Parity Space: 61
Percentage Correlation Parity Space: 0.83333
Percentage Correlation Observer: 20.25
W =
100
Percentage Correlation Adaptive Parity Space: 50.6667
Percentage Correlation Parity Space: 1
Percentage Correlation Observer: 6.5
W =
50
Percentage Correlation Adaptive Parity Space: 21.3333
Percentage Correlation Parity Space: 2.6667
Percentage Correlation Observer: -20.6667
W =
30
Percentage Correlation Adaptive Parity Space: -12.2222
Percentage Correlation Parity Space: 3.8889
Percentage Correlation Observer: -39.4444

Step:

```
A =
     5
W =
   300
Percentage Correlation Adaptive Parity Space: 87.3333
Percentage Correlation Parity Space: 42.1111
Percentage Correlation Observer: 70.3333
W =
   200
Percentage Correlation Adaptive Parity Space: 67.9167
Percentage Correlation Parity Space: 39.8333
Percentage Correlation Observer: 31
W =
   100
Percentage Correlation Adaptive Parity Space: 49.5
Percentage Correlation Parity Space: 34.6667
Percentage Correlation Observer: -78.1667
W =
    50
Percentage Correlation Adaptive Parity Space: 27.3333
Percentage Correlation Parity Space: 25
Percentage Correlation Observer: -250
W =
    30
Percentage Correlation Adaptive Parity Space: -17.2222
Percentage Correlation Parity Space: 11.1111
Percentage Correlation Observer: -435.5556

A =
  2.5000e+000
W =
   300
Percentage Correlation Adaptive Parity Space: 84.8889
Percentage Correlation Parity Space: 29.5556
Percentage Correlation Observer: 72.2778
W =
   200
Percentage Correlation Adaptive Parity Space: 78
Percentage Correlation Parity Space: 28.4167
Percentage Correlation Observer: 33.3333
W =
   100
Percentage Correlation Adaptive Parity Space: 64.6667
Percentage Correlation Parity Space: 25.1667
Percentage Correlation Observer: -54.5
W =
    50
Percentage Correlation Adaptive Parity Space: 31.3333
Percentage Correlation Parity Space: 18
Percentage Correlation Observer: -189
W =
    30
Percentage Correlation Adaptive Parity Space: -13.3333
Percentage Correlation Parity Space: 8.8889
Percentage Correlation Observer: -317.2222

A =
```

```
    1
W =
  300
Percentage Correlation Adaptive Parity Space: 86.0556
Percentage Correlation Parity Space: 9.3333
Percentage Correlation Observer: 69.3333
W =
  200
Percentage Correlation Adaptive Parity Space: 79.1667
Percentage Correlation Parity Space: 9
Percentage Correlation Observer: 36.1667
W =
  100
Percentage Correlation Adaptive Parity Space: 59.5
Percentage Correlation Parity Space: 7.6667
Percentage Correlation Observer: -23.6667
W =
   50
Percentage Correlation Adaptive Parity Space: 28
Percentage Correlation Parity Space: 7
Percentage Correlation Observer: -96.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -10
Percentage Correlation Parity Space: 3.3333
Percentage Correlation Observer: -187.2222

A =
 5.0000e-001
W =
  300
Percentage Correlation Adaptive Parity Space: 81.8333
Percentage Correlation Parity Space: 2.4444
Percentage Correlation Observer: 62.6667
W =
  200
Percentage Correlation Adaptive Parity Space: 73.8333
Percentage Correlation Parity Space: 2.4167
Percentage Correlation Observer: 37.75
W =
  100
Percentage Correlation Adaptive Parity Space: 58
Percentage Correlation Parity Space: 1.5
Percentage Correlation Observer: -2.6667
W =
   50
Percentage Correlation Adaptive Parity Space: 30.6667
Percentage Correlation Parity Space: 2.3333
Percentage Correlation Observer: -59.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -7.2222
Percentage Correlation Parity Space: 0
Percentage Correlation Observer: -131.6667

A =
 2.5000e-001
W =
  300
Percentage Correlation Adaptive Parity Space: 60.6667
```

```
Percentage Correlation Parity Space: 0.33333
Percentage Correlation Observer: 41.5556
W =
  200
Percentage Correlation Adaptive Parity Space: 56.25
Percentage Correlation Parity Space: 0.41667
Percentage Correlation Observer: 21.1667
W =
  100
Percentage Correlation Adaptive Parity Space: 44.6667
Percentage Correlation Parity Space: 0
Percentage Correlation Observer: -1.1667
W =
   50
Percentage Correlation Adaptive Parity Space: 22.6667
Percentage Correlation Parity Space: 0.33333
Percentage Correlation Observer: -37.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -6.6667
Percentage Correlation Parity Space: -0.55556
Percentage Correlation Observer: -80.5556
```

APPENDIX H: RESIDUAL AND FAULT INDICATION GRAPHS FOR VARIED FAULT PARAMETERS

This appendix includes the residual and fault indication graphs of the fault detection methods with varied fault duration and amplitude. Firstly, the parity space method results are shown, followed by the adaptive parity space and observer methods. The following are the static simulation parameters for all the simulation results given in this appendix:

- The initial operating conditions, x_0, u_0 and y_0 were set to zero.
- The process was required to track a setpoint, which was a step waveform of 1V amplitude from a steady state value of 0V.
- The duration of the simulation was 250[s] which shows the transient response of the system (simulated, steady state settling time = 304[s] of the closed loop response from Section 3.4). The reason that the simulation investigations were performed in the transient state was due to requirements set by the minerals extraction industry. The sampling time was chosen to be 0.1[s] (for accurate observation of the residuals), and the initial delay of the setpoint was 1[s].
- Noise injected into the system was 5% of the setpoint amplitude and uniformly distributed for the duration of the simulation. Thus the amplitude of the noise at each sample was chosen randomly from a uniform distribution between 0.05V.
- The six faults simulated were:
    1. Sensor 1 Scale fault
    2. Sensor 2 Scale fault
    3. Sensor 1 Bias fault
    4. Sensor 2 Bias fault
    5. Actuator Scale fault
    6. Actuator Bias fault

The scale faults were simulated with on offset of 1V, which is the final value of the step setpoint. This is necessary since the scale faults are multiplicative faults and therefore have to be initiated at a voltage offset equal to the amplitude of the setpoint in order for the fault to be correctly implemented. All the faults were initiated at different times, with no fault occurring

at the same time as another. The following table describes the dynamic fault parameters used for the simulations:

Table H-1 Fault Parameters used for the Simulations

Simulation Number	*Fault Waveform*	*Fault Amplitude [V]*	*Fault Duration [s]*
1	Ramp	0.25	3
2	Saw	0.50	5
3	Sine	1.00	10
4	Square	2.50	20
5	Step	5.00	30

H.1 Parity Space Method

The following graphs show the residuals and fault indication graphs for the various simulation parameters described above, for the parity space method of fault detection (as described in Section 5.2.2).

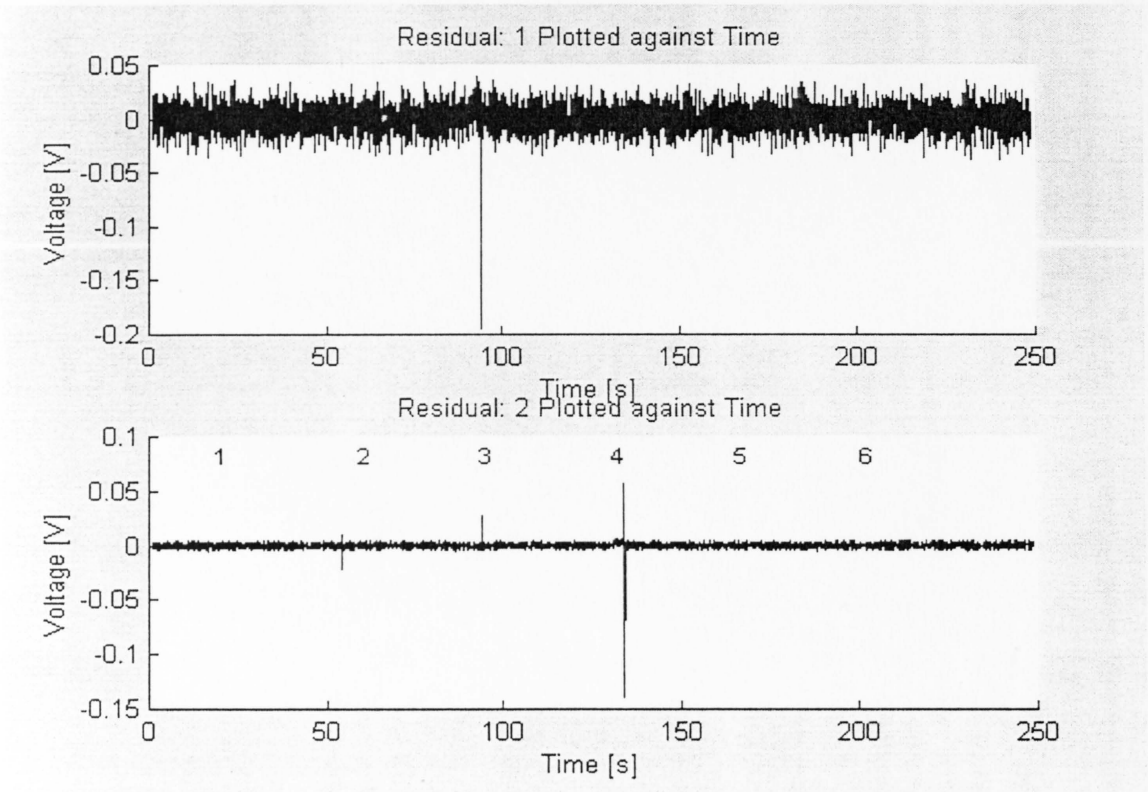


Figure H-1 Parity Space Residuals 1 & 2 for Simulation 1

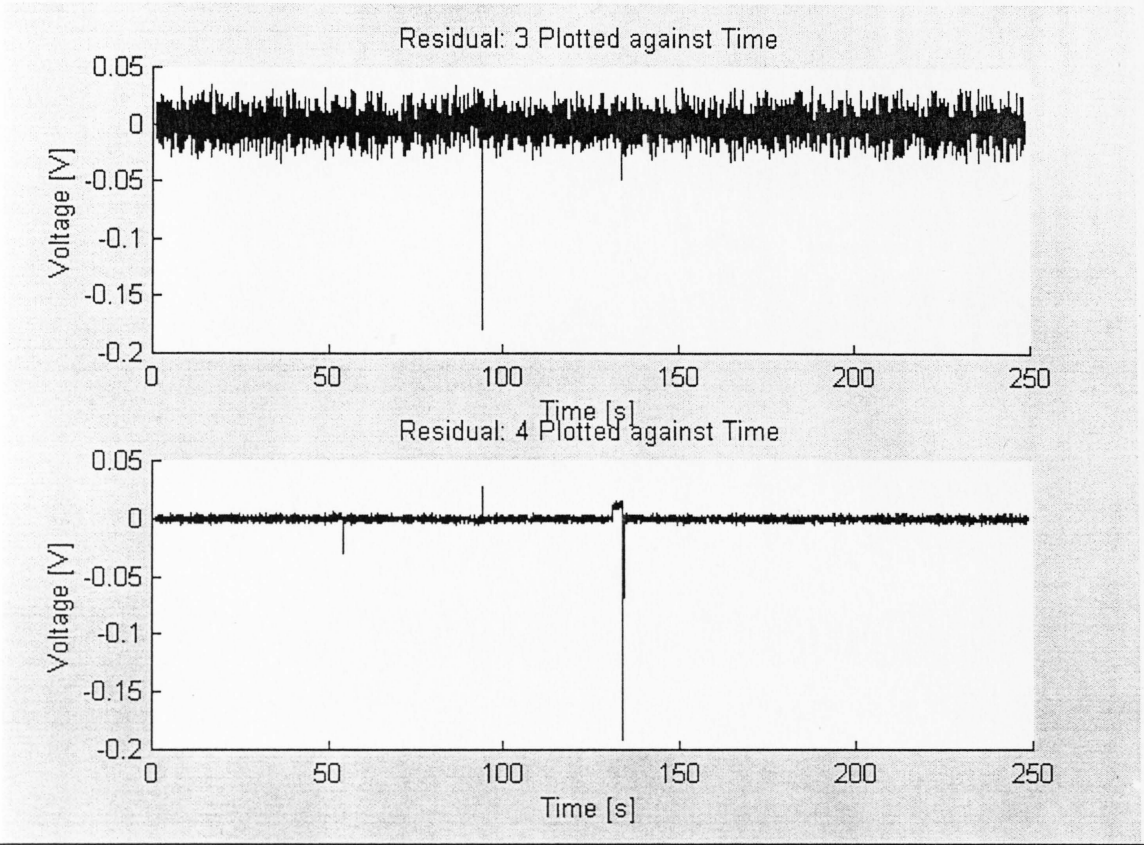


Figure H-2 Parity Space Residuals 3 & 4 for Simulation 1

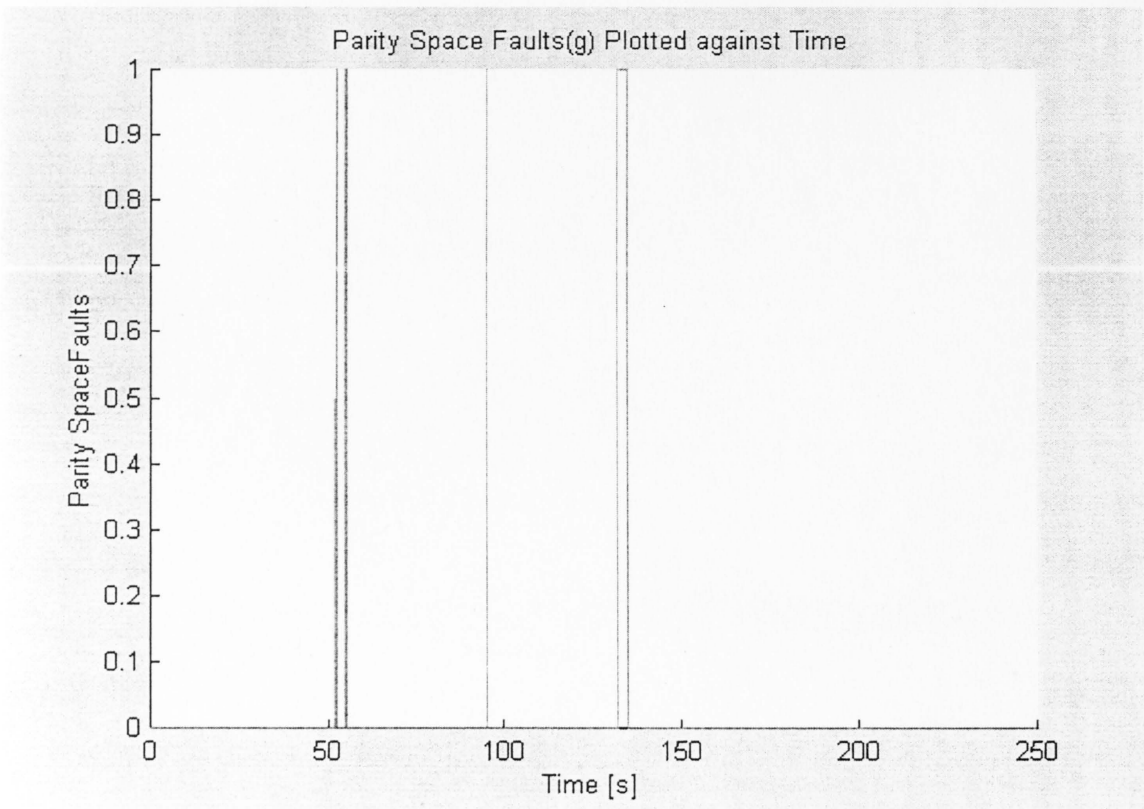


Figure H-3 Parity Space Fault Indication for Simulation 1

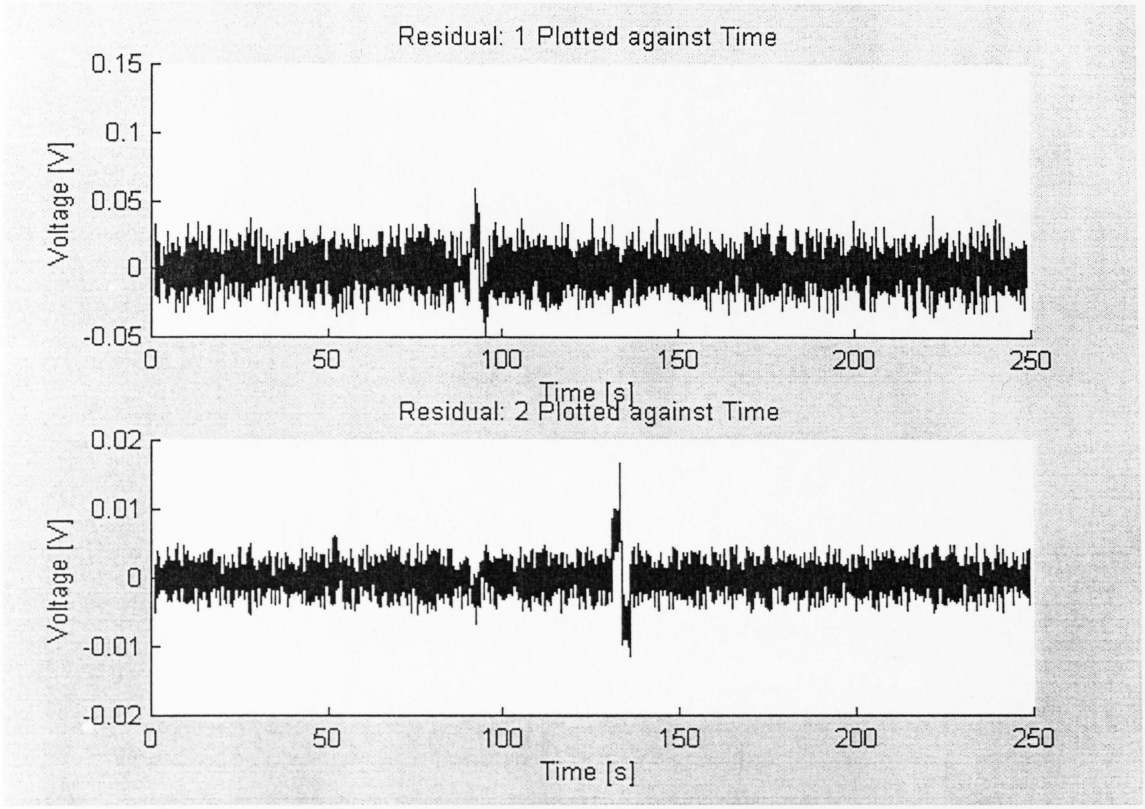


Figure H-4 Parity Space Residuals 1 & 2 for Simulation 2

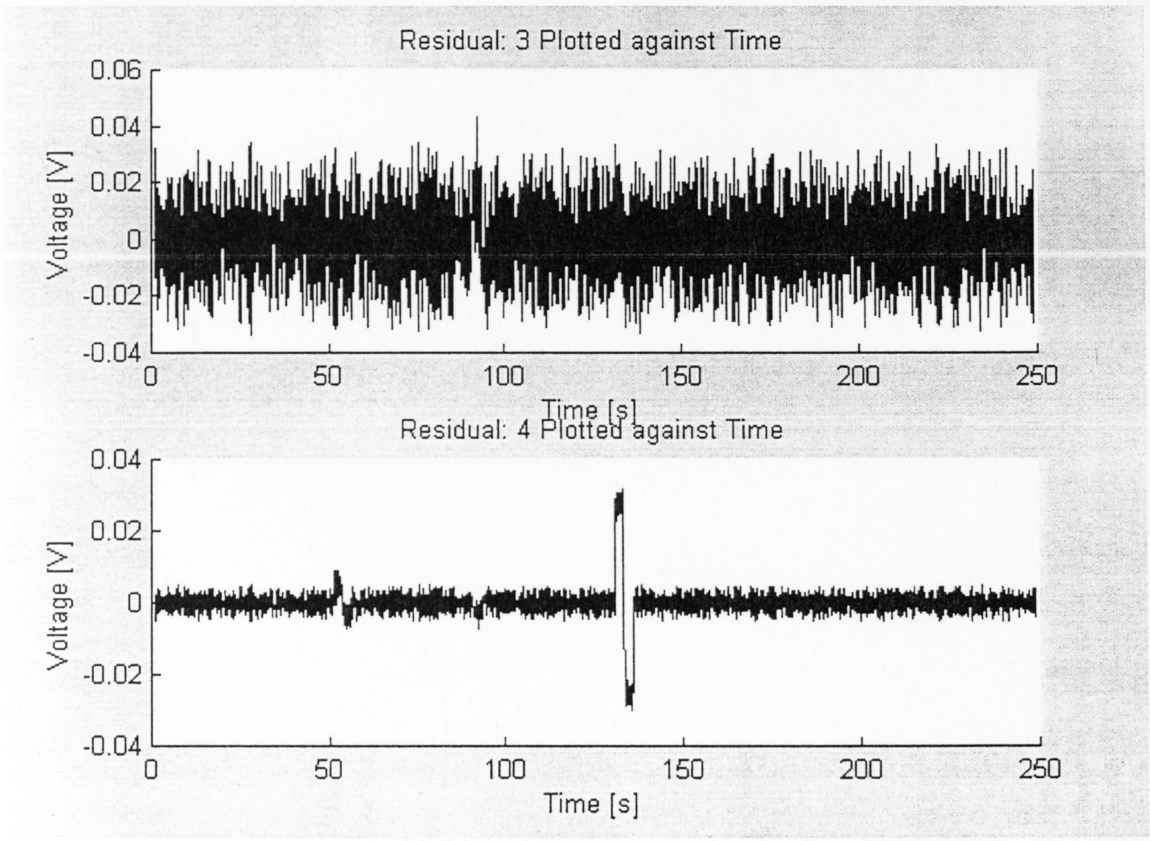


Figure H-5 Parity Space Residuals 3 & 4 for Simulation 2

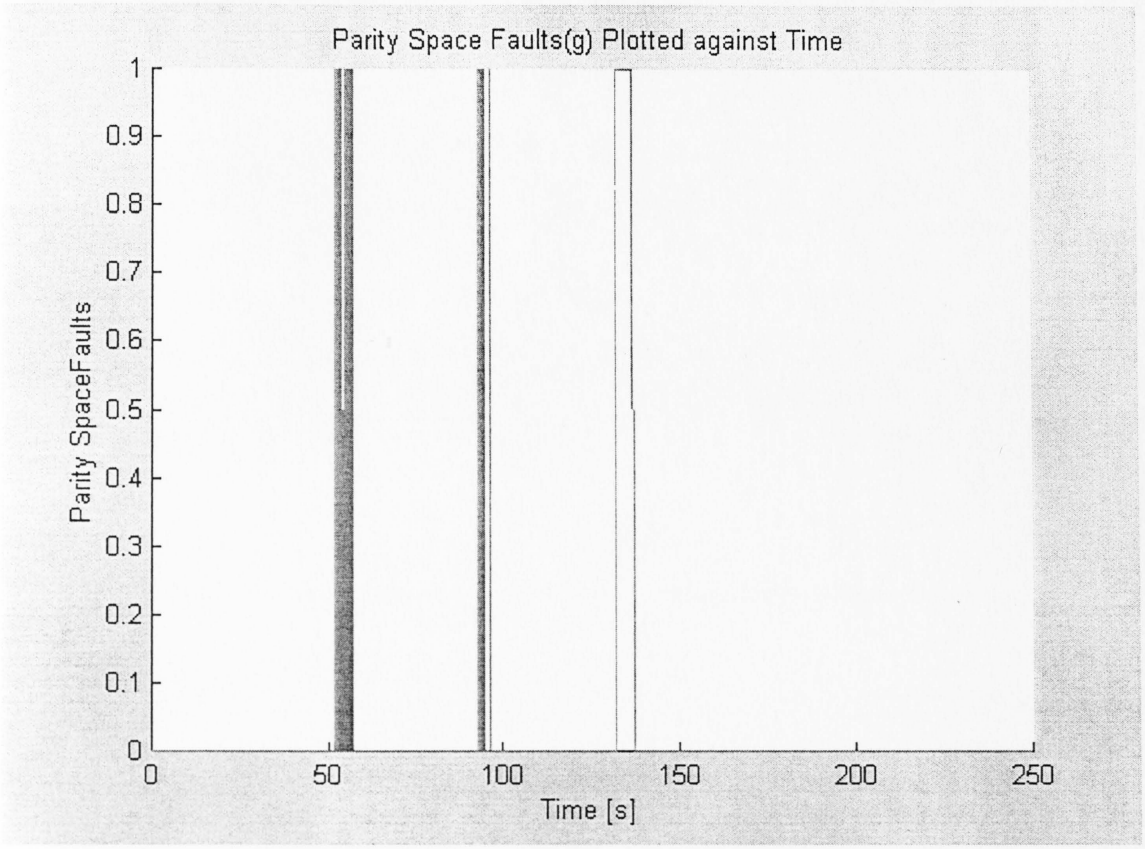


Figure H-6 Parity Space Fault Indication for Simulation 2

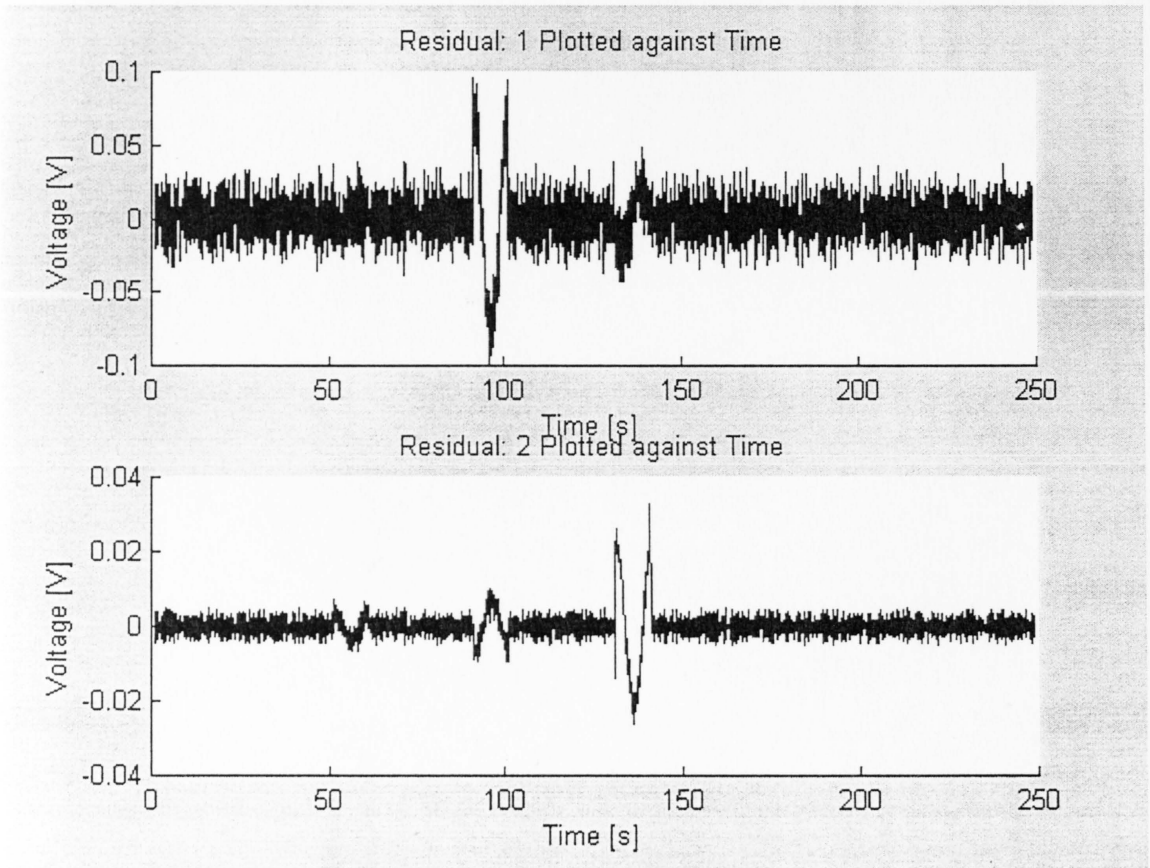


Figure H-7 Parity Space Residuals 1 & 2 for Simulation 3

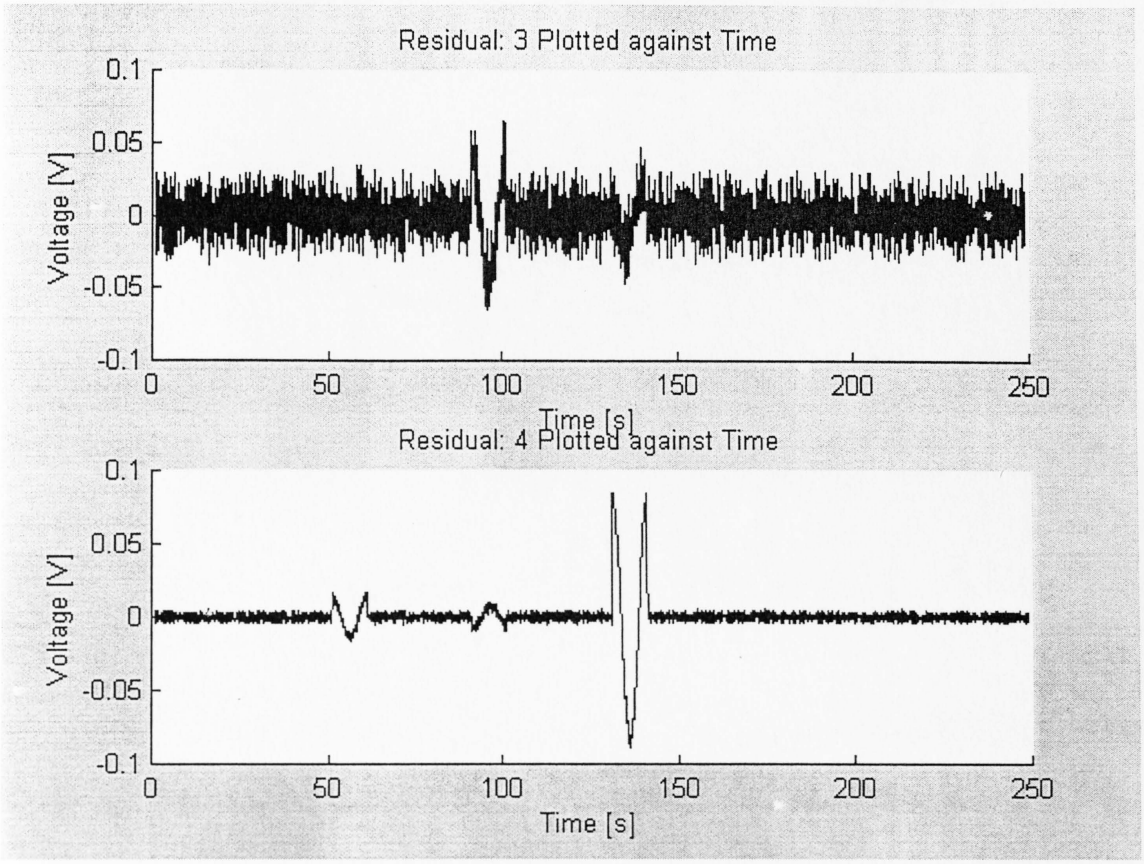


Figure H-8 Parity Space Residuals 3 & 4 for Simulation 3

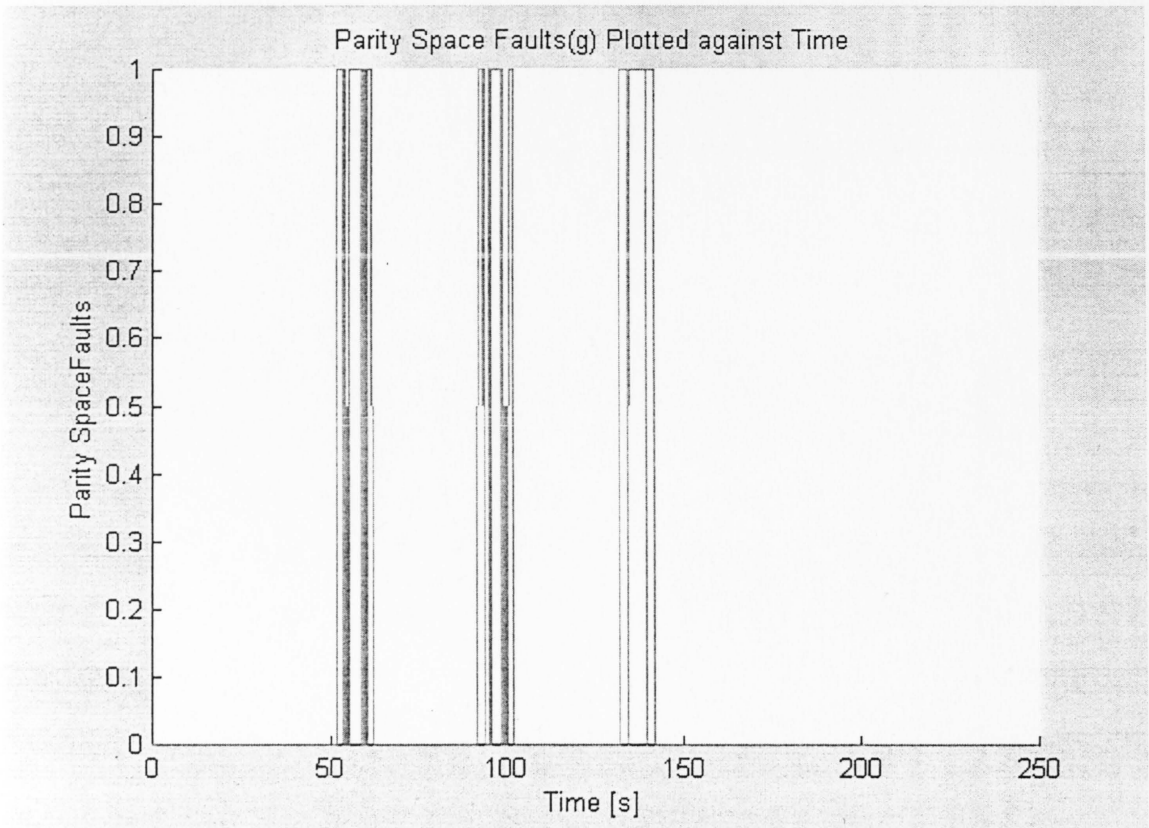


Figure H-9 Parity Space Fault Detection for Simulation 3

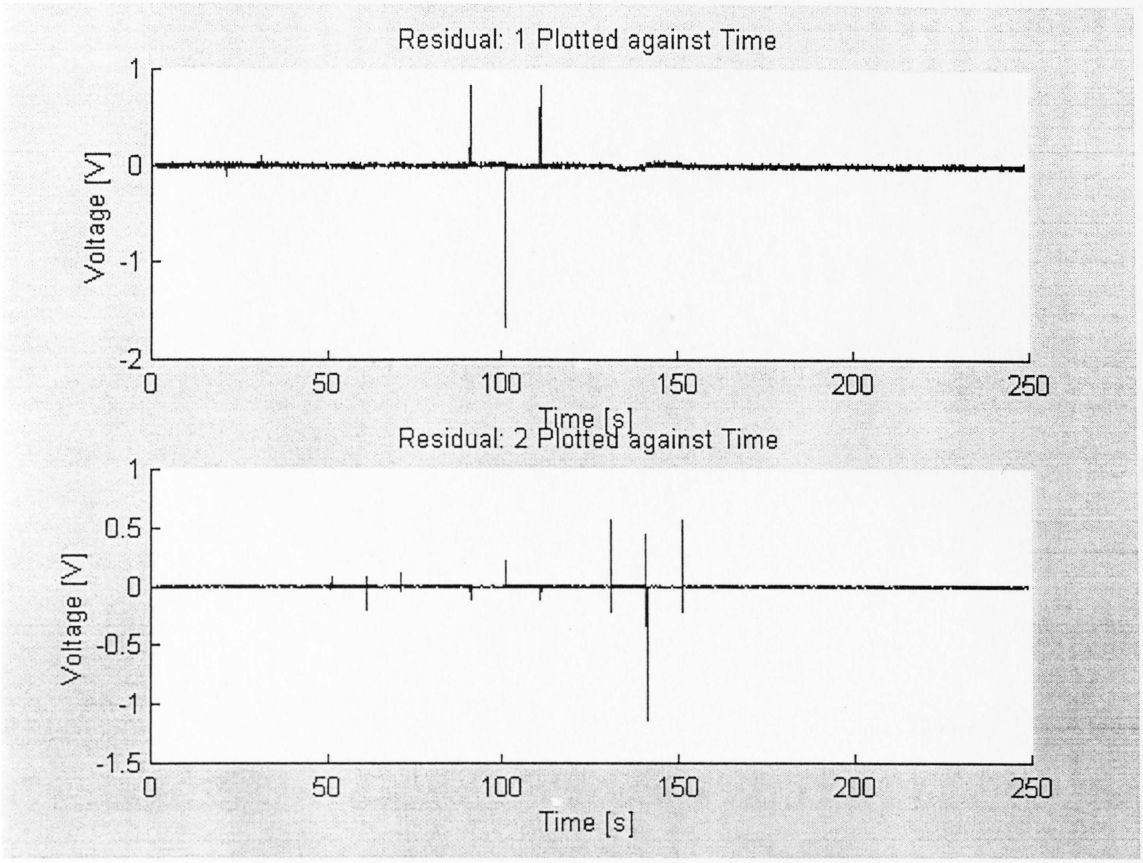


Figure H-10 Parity Space Residuals 1 & 2 for Simulation 4

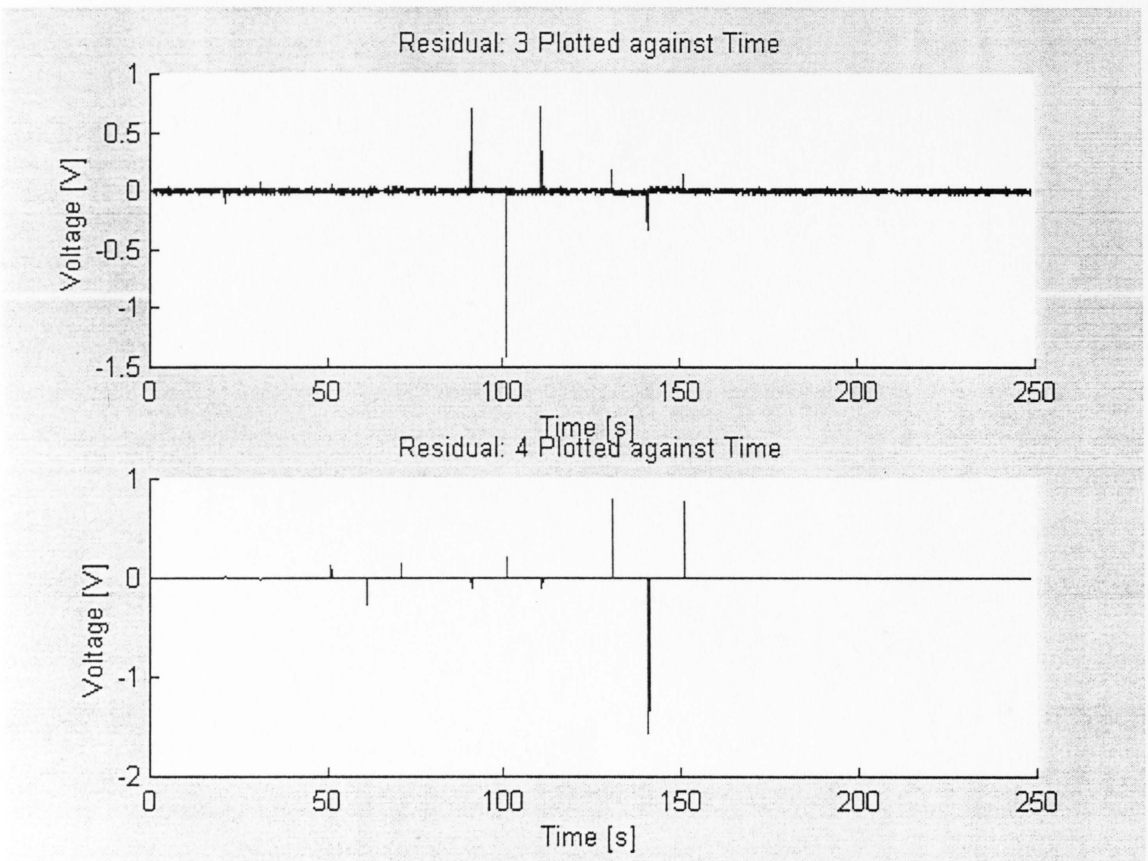


Figure H-11 Parity Space Residuals 3 & 4 for Simulation 4

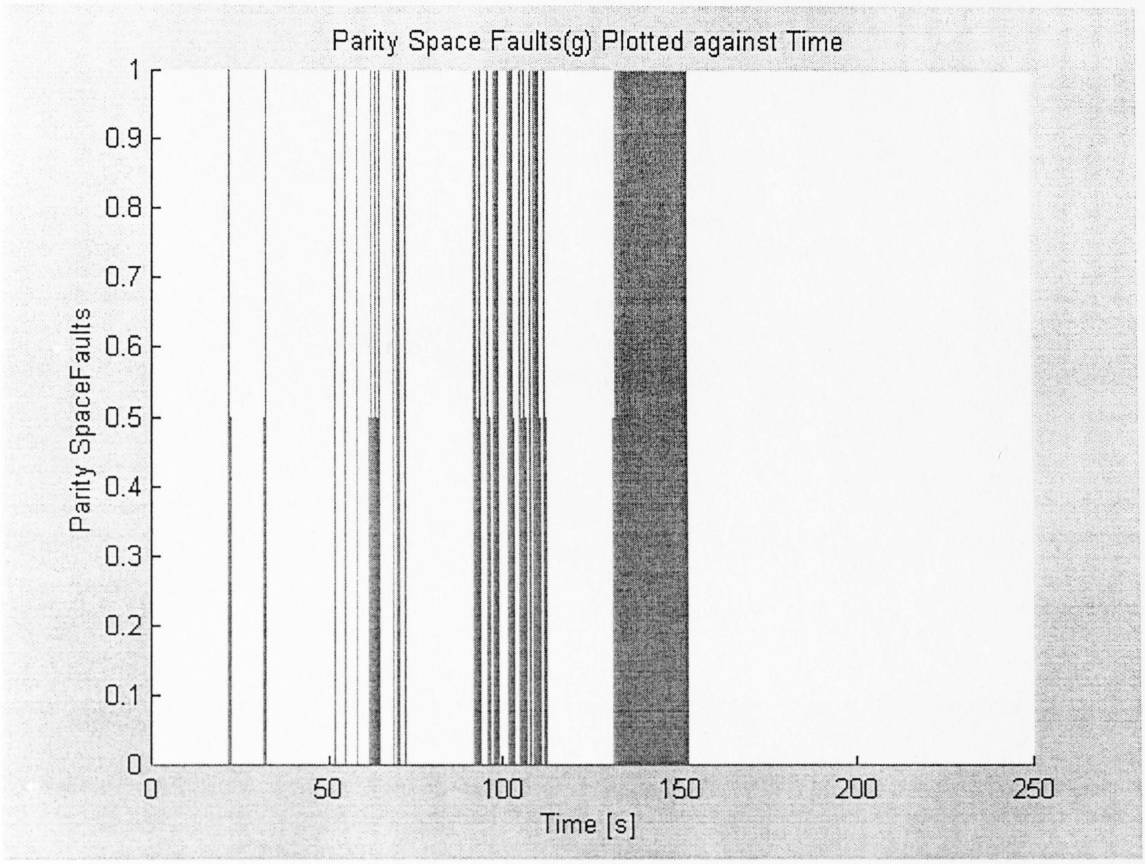


Figure H-12 Parity Space Fault Detection for Simulation 4

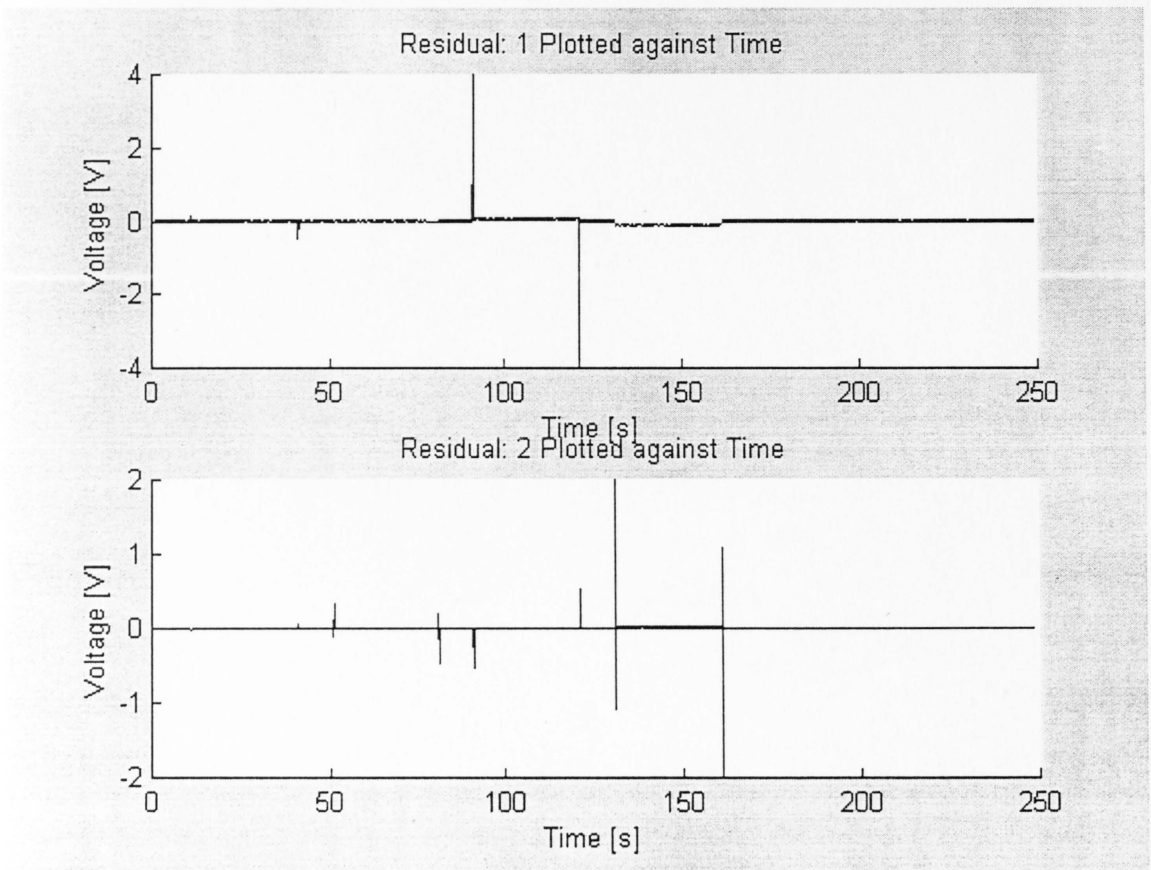


Figure H-13 Parity Space Residuals 1 & 2 for Simulation 5

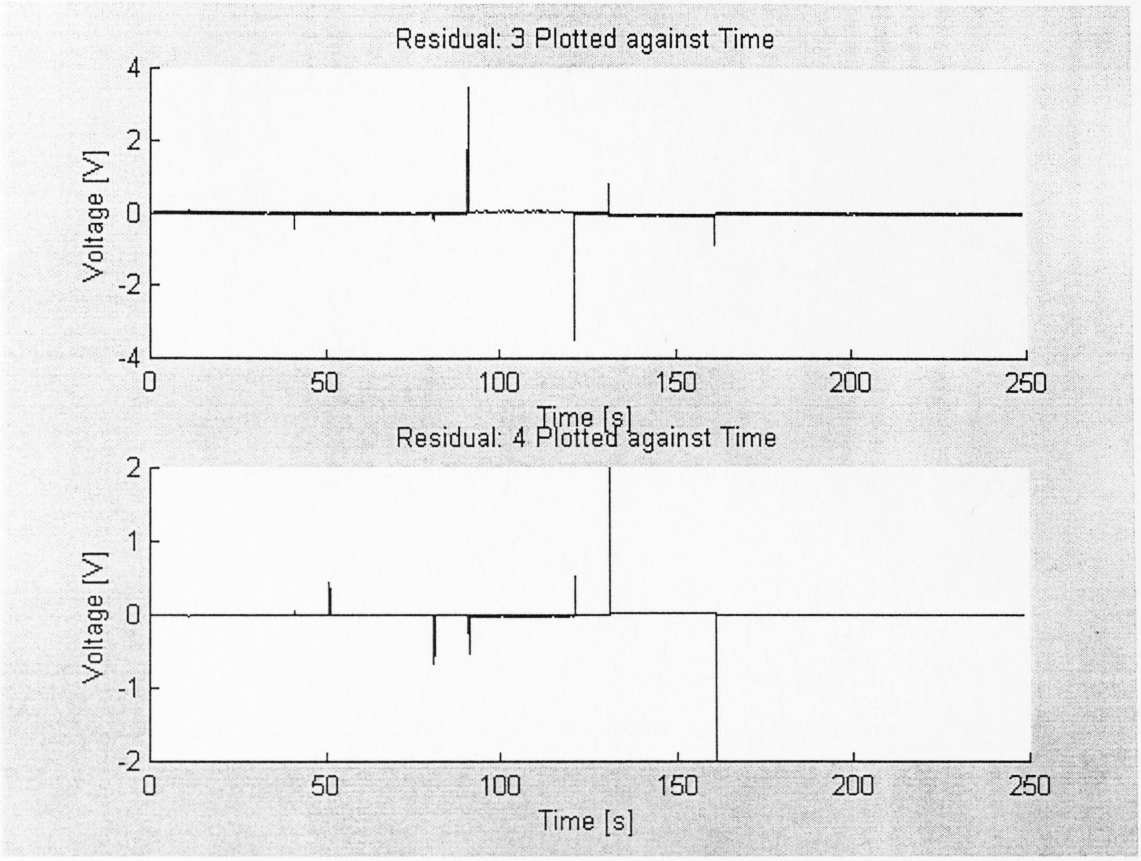


Figure H-14 Parity Space Residuals 3 & 4 for Simulation 5

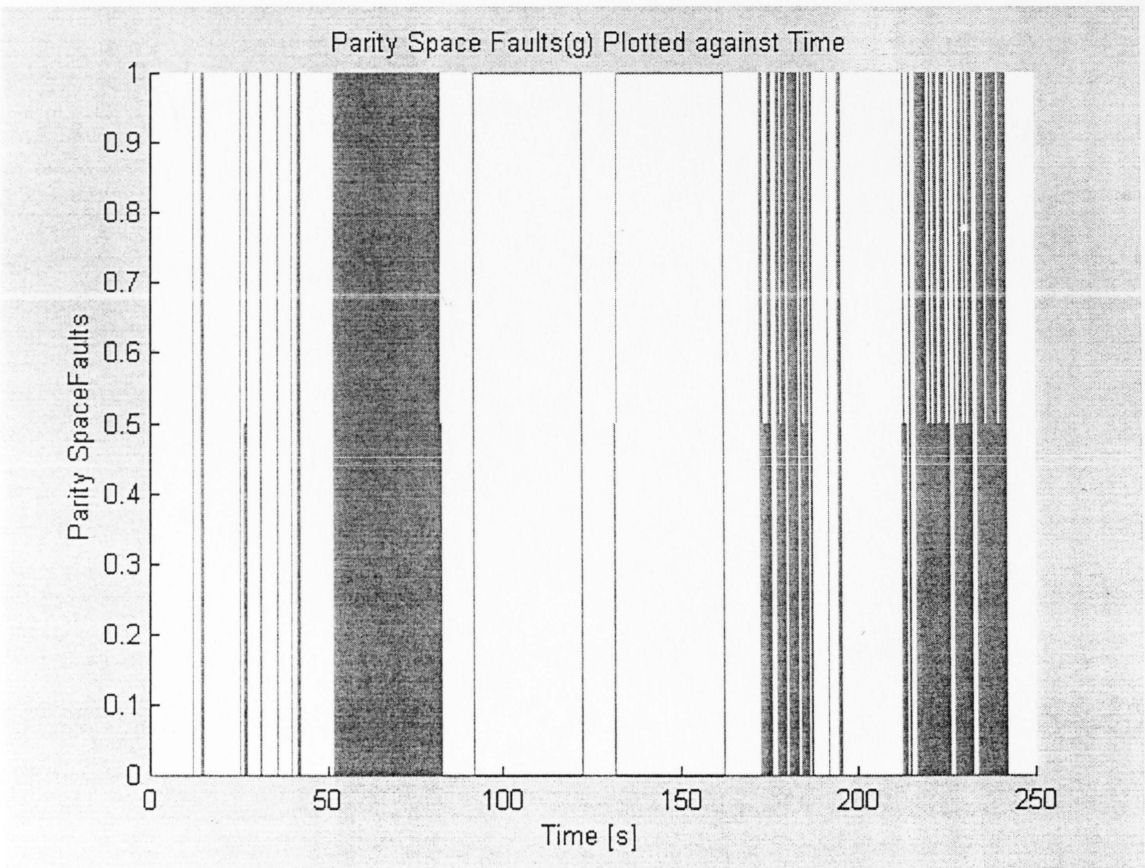


Figure H-15 Parity Space Fault Indication for Simulation 5

Comment:

From analysis of the fault indication graphs, it can be seen that the fault detection performance of the parity space method improves significantly with fault amplitude and fault duration. Further observations show that the method has difficulty in detecting the Sensor 1 scale fault and both actuator faults (the Sensor 1 scale fault is only detected in Simulation 3, while the actuator faults are only detected in Simulation 4). The fault detection performance with small fault amplitudes and a short duration is not good at all.

H.2 Adaptive Parity Space Method

The following graphs show the residuals and fault indication graphs for the various simulation parameters described above, for the adaptive parity space method of fault detection (as described in Section 5.3.4).

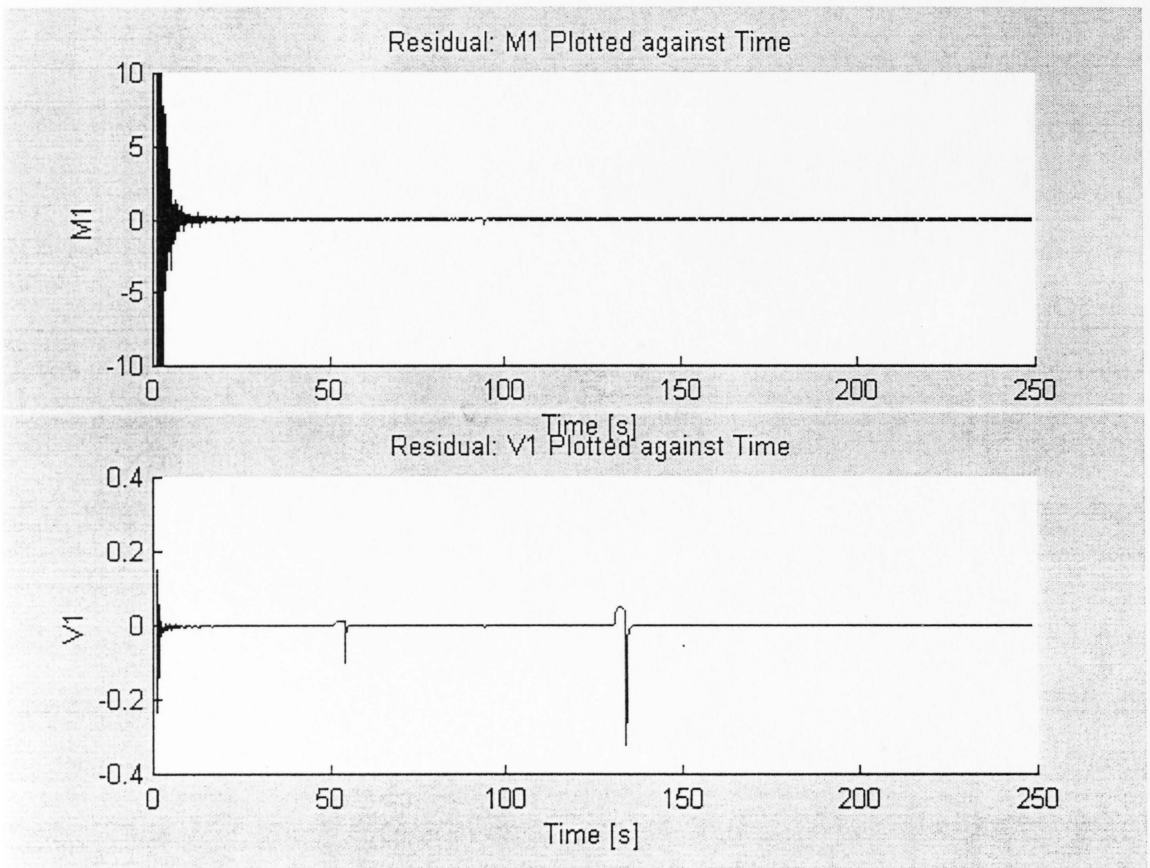


Figure H-16 Adaptive Parity Space Residuals M1 & V1 for Simulation 1

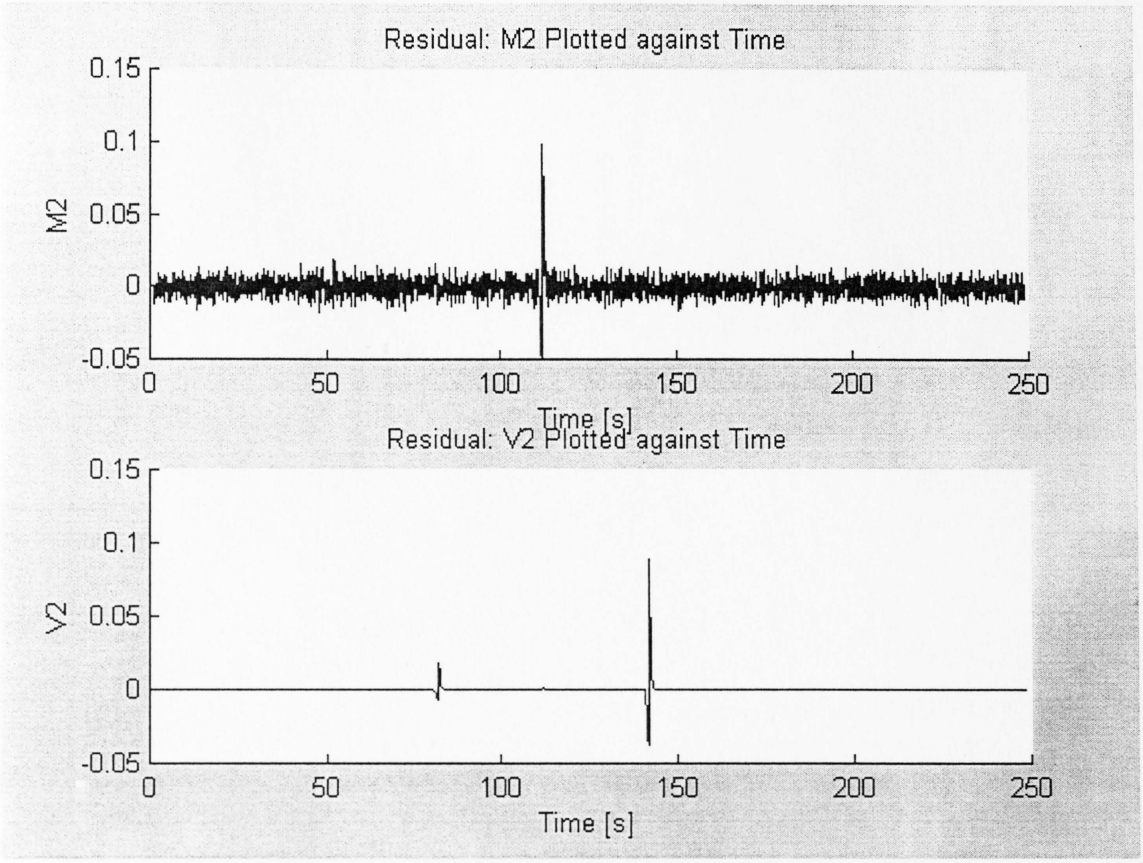


Figure H-17 Adaptive Parity Space Residuals M2 & V2 for Simulation 1

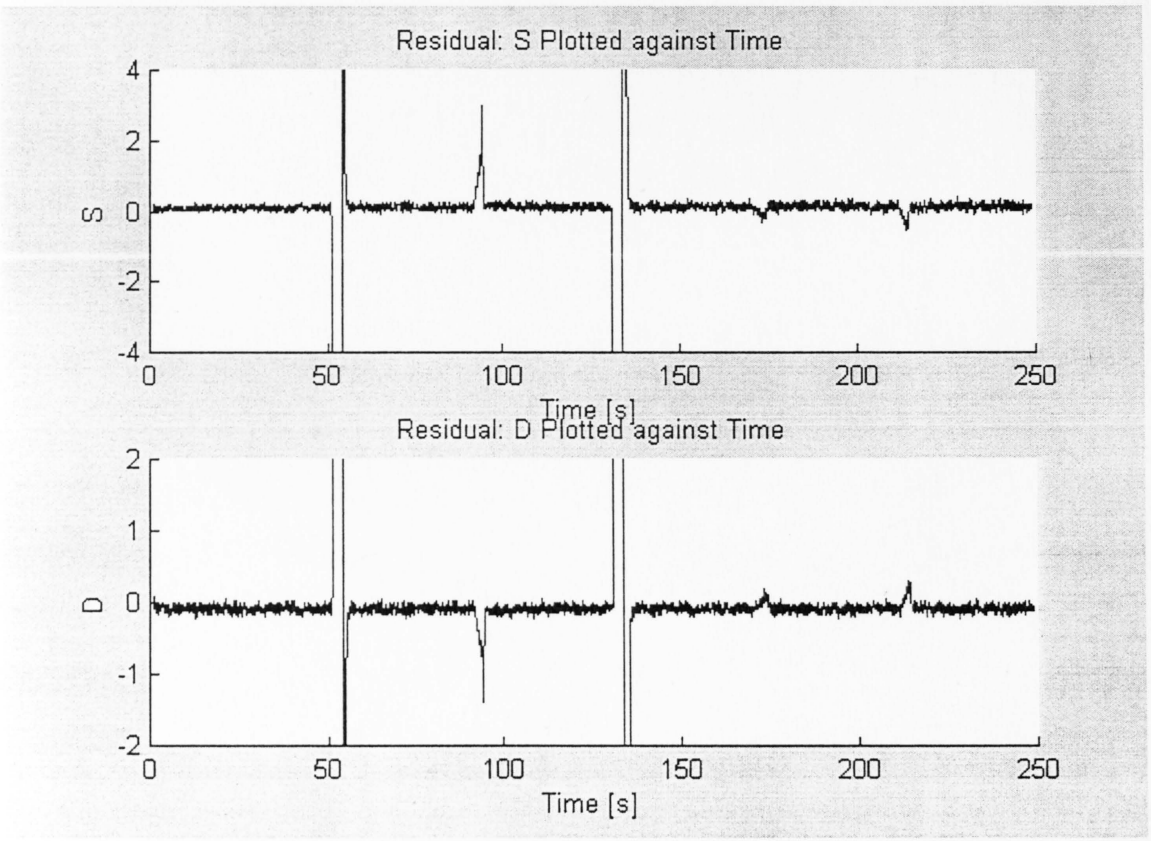


Figure H-18 Adaptive Parity Space Residuals S & D for Simulation 1

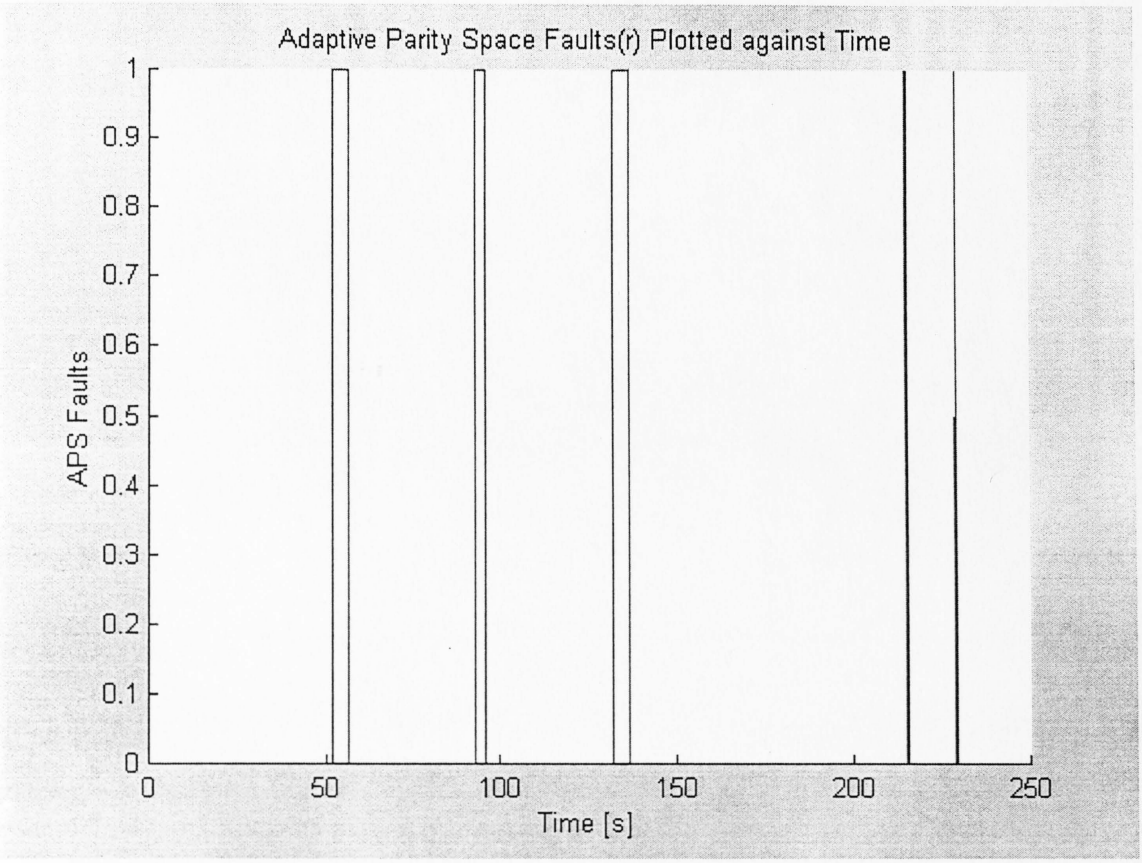


Figure H-19 Adaptive Parity Space Fault Indication for Simulation 1

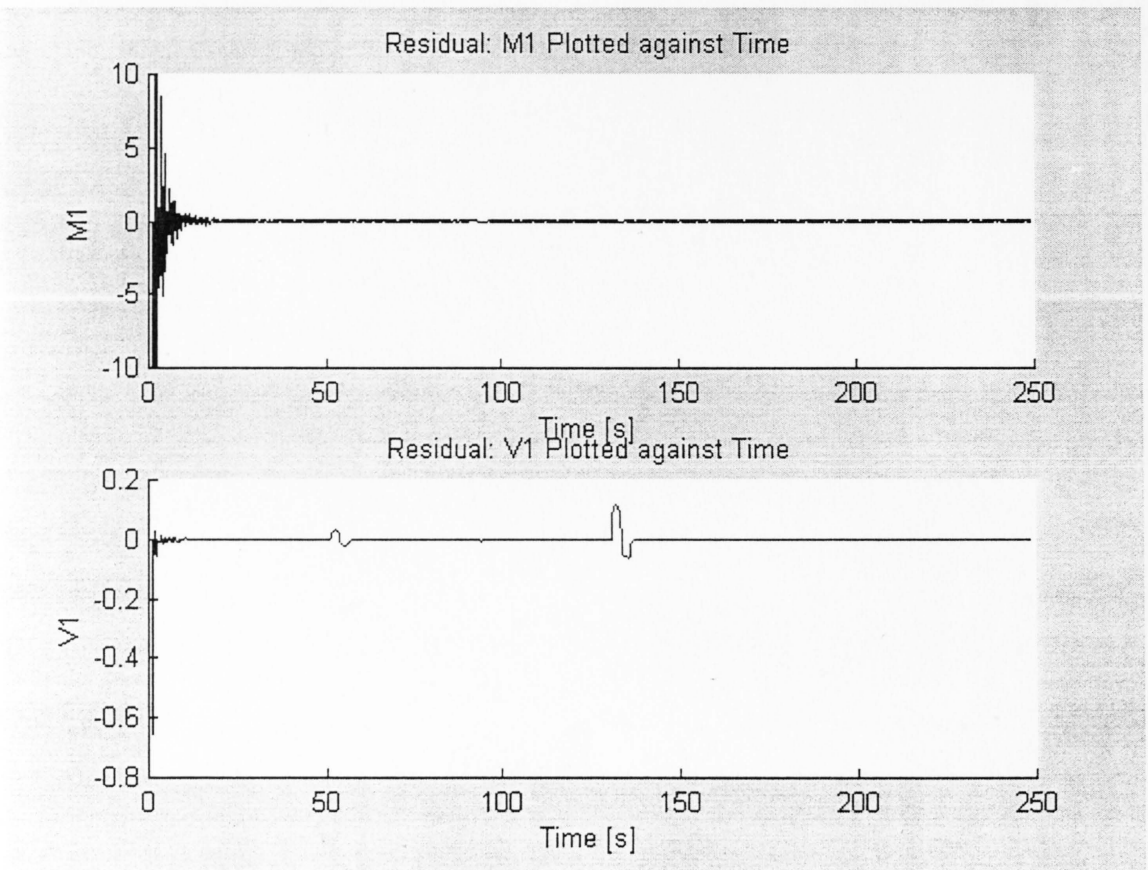


Figure H-20 Adaptive Parity Space Residuals M1 & V1 for Simulation 2

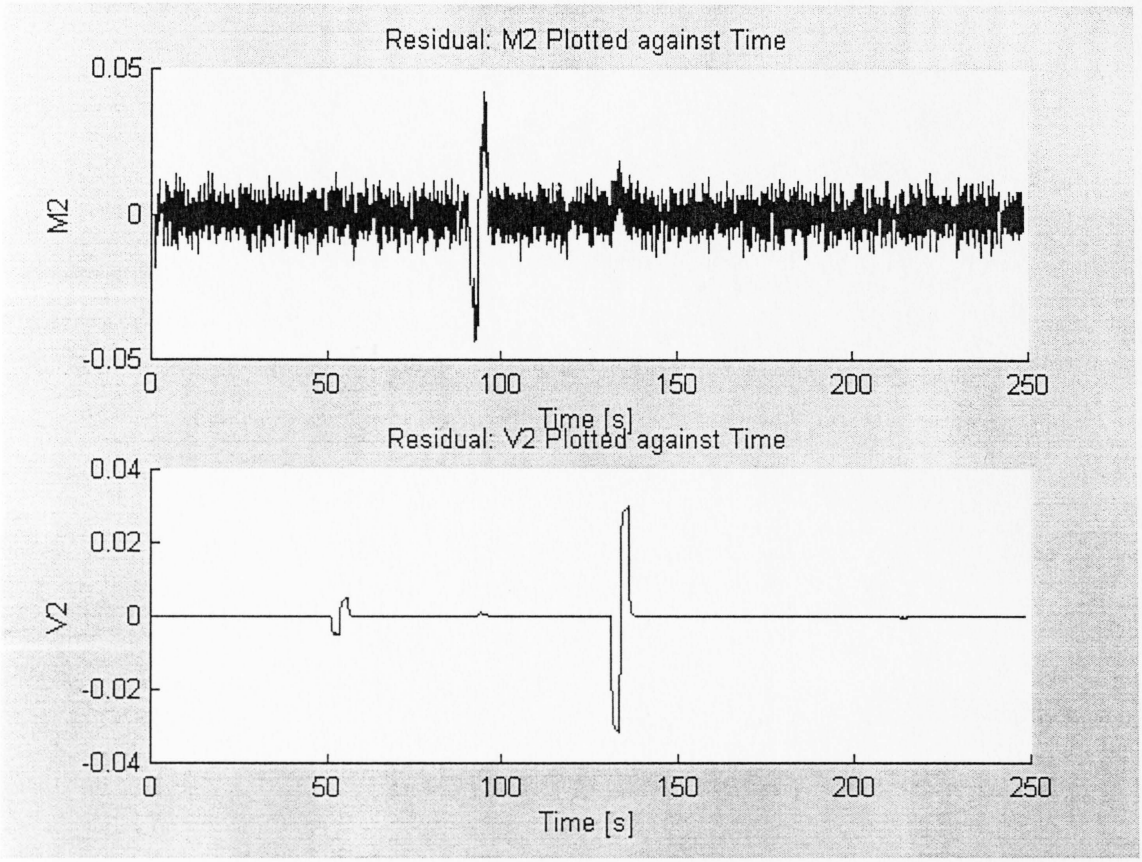


Figure H-21 Adaptive Parity Space Residuals M2 & V2 for Simulation 2

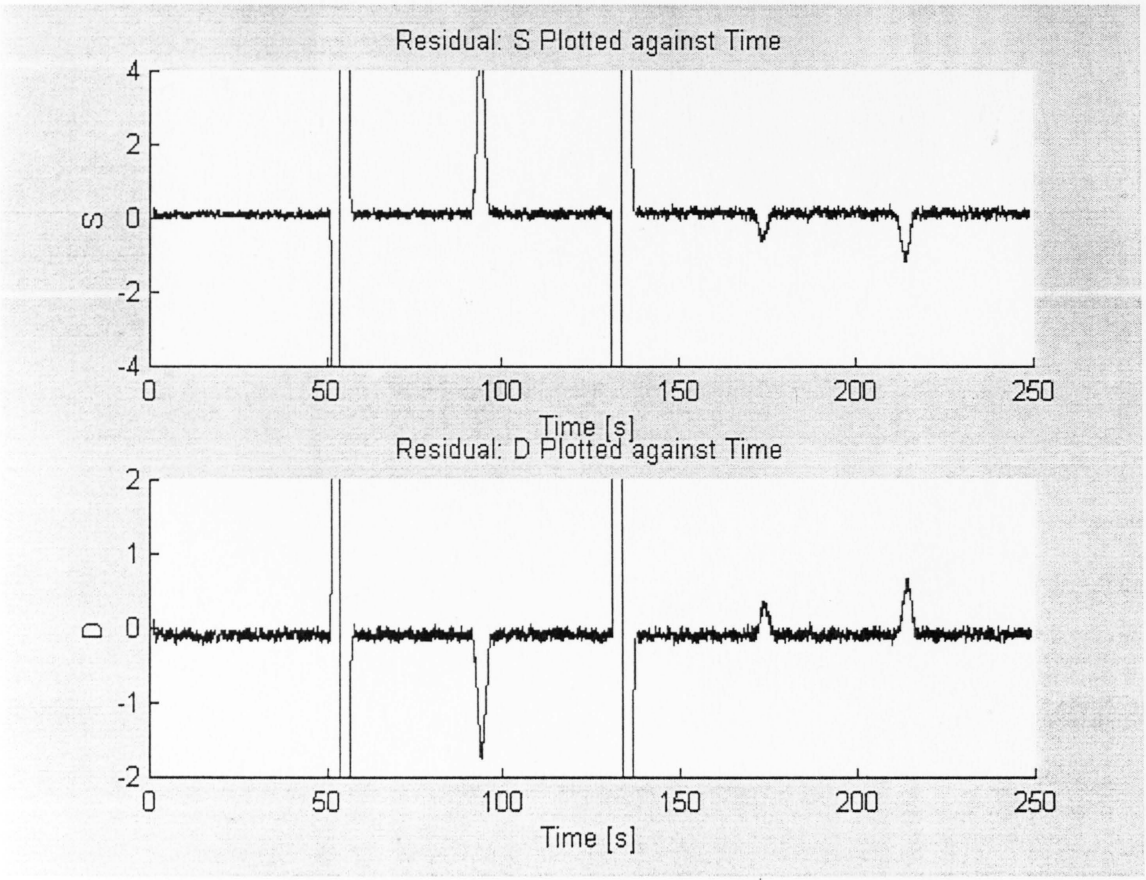


Figure H-22 Adaptive Parity Space Residuals S & D for Simulation 2

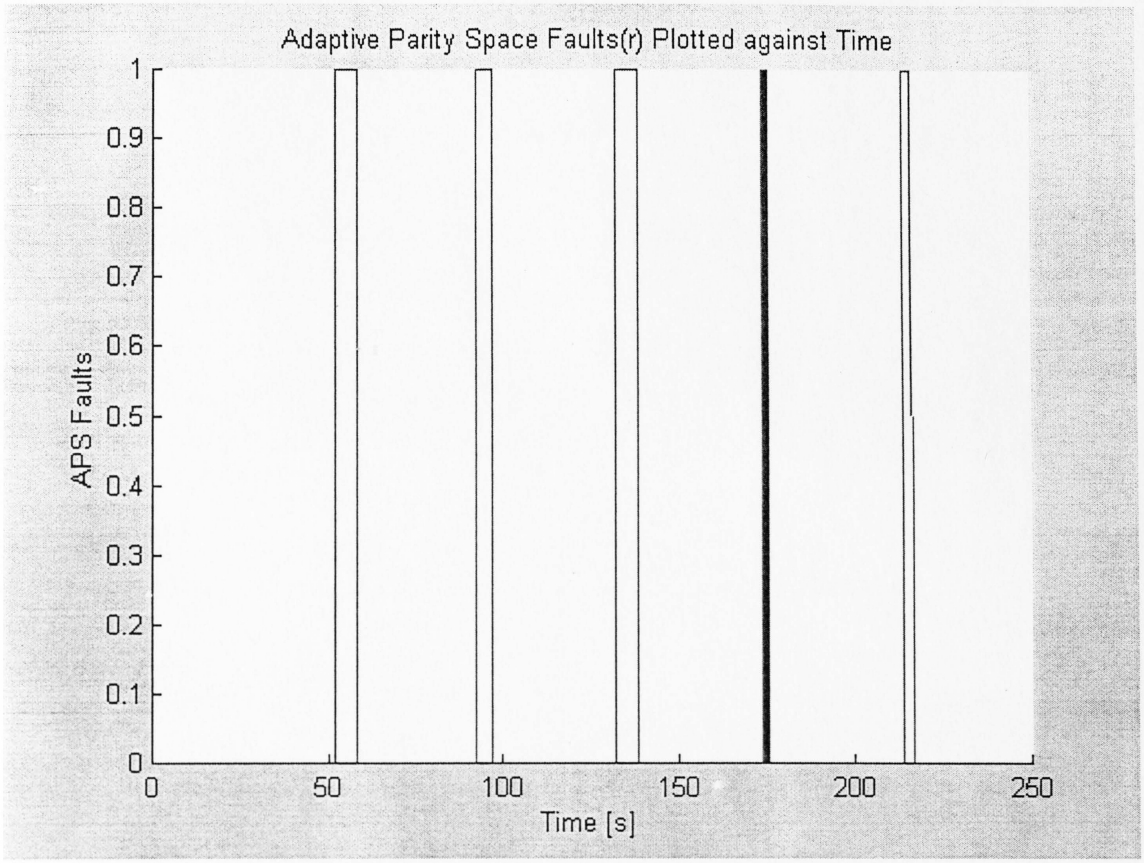


Figure H-23 Adaptive Parity Space Fault Indication for Simulation 2

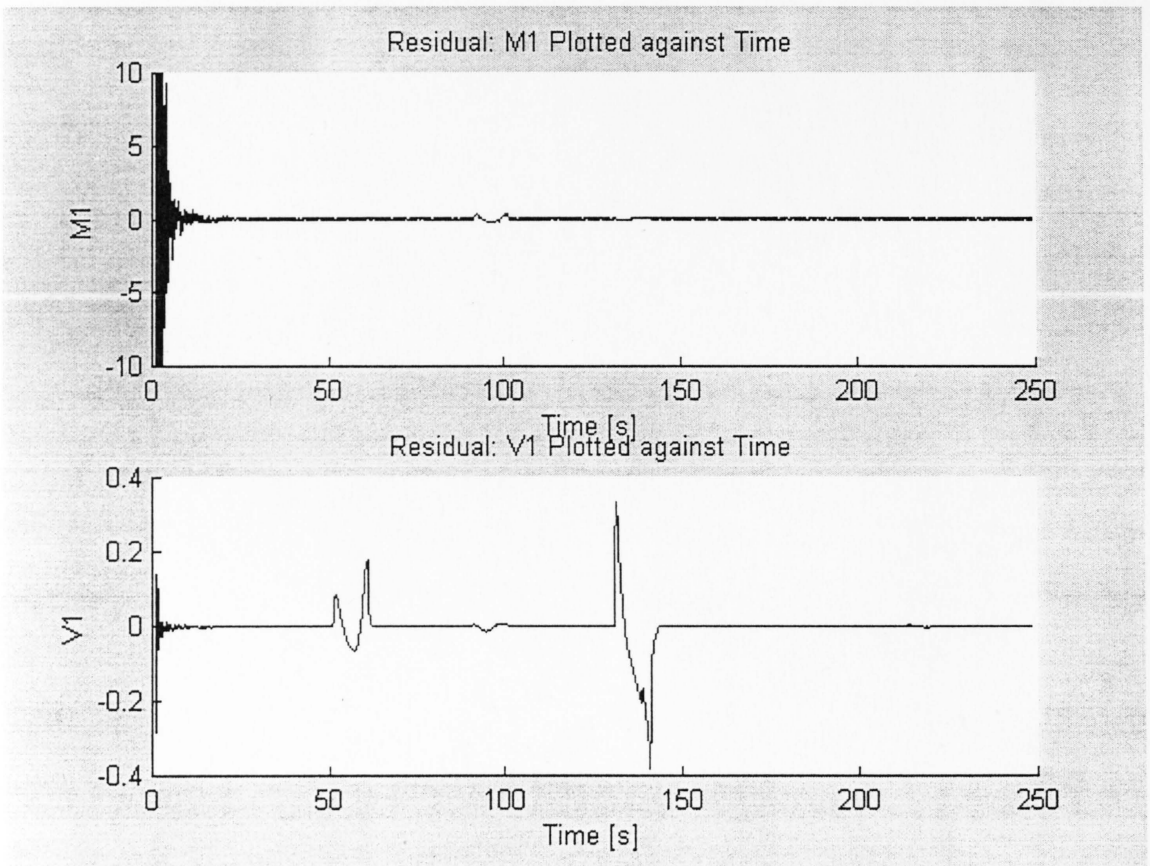


Figure H-24 Adaptive Parity Space Residuals M1 & V1 for Simulation 3

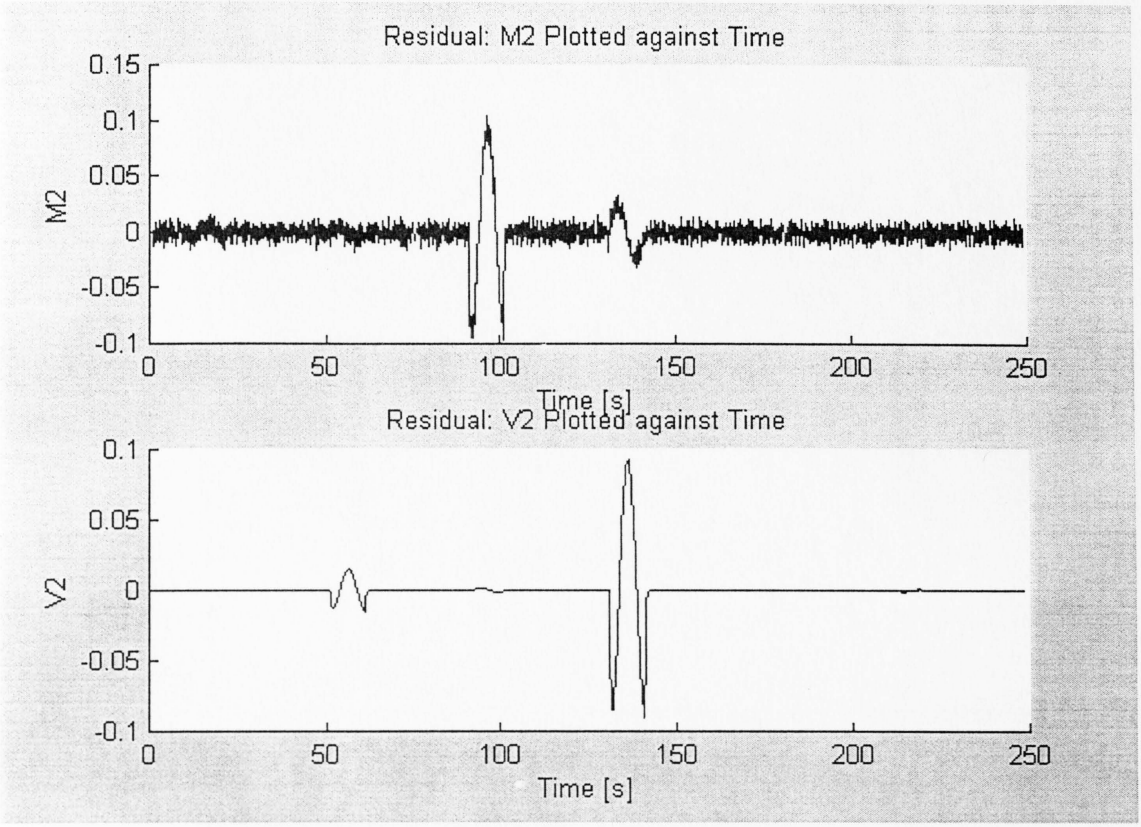


Figure H-25 Adaptive Parity Space Residuals M2 & V2 for Simulation 3

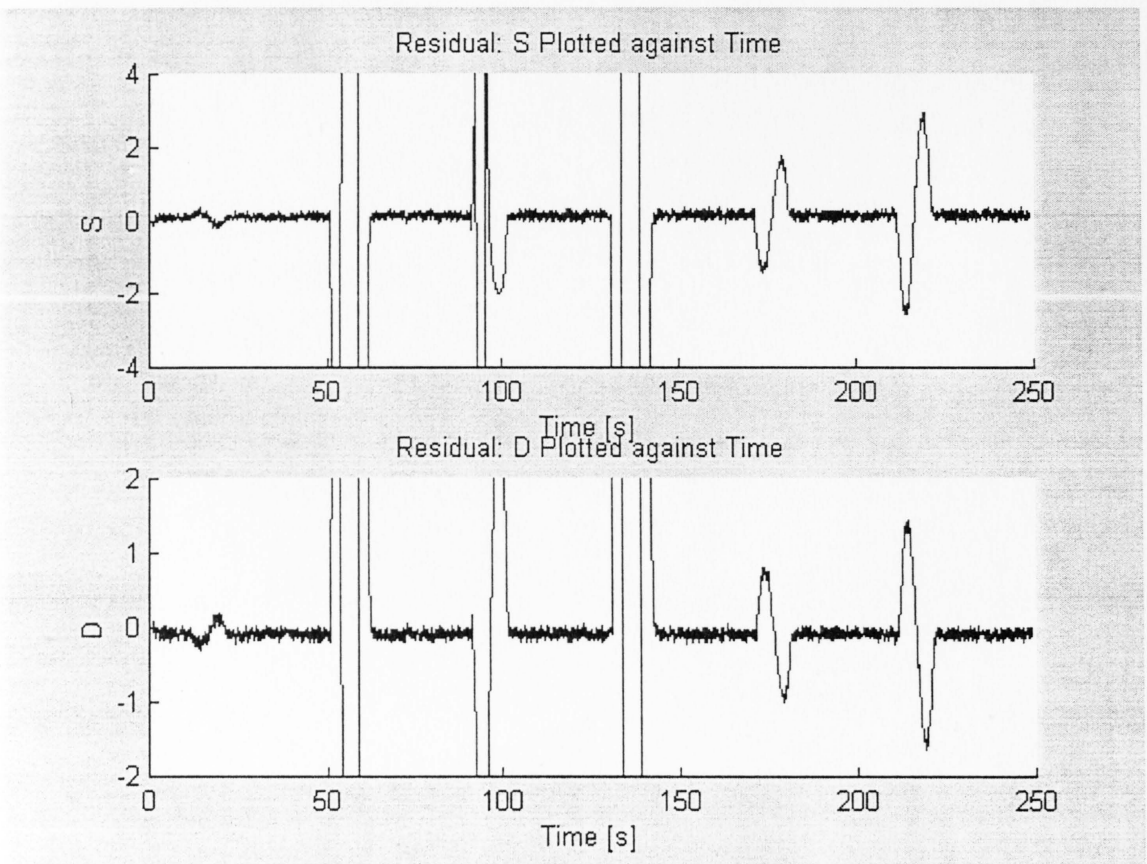


Figure H-26 Adaptive Parity Space Residuals S & D for Simulation 3

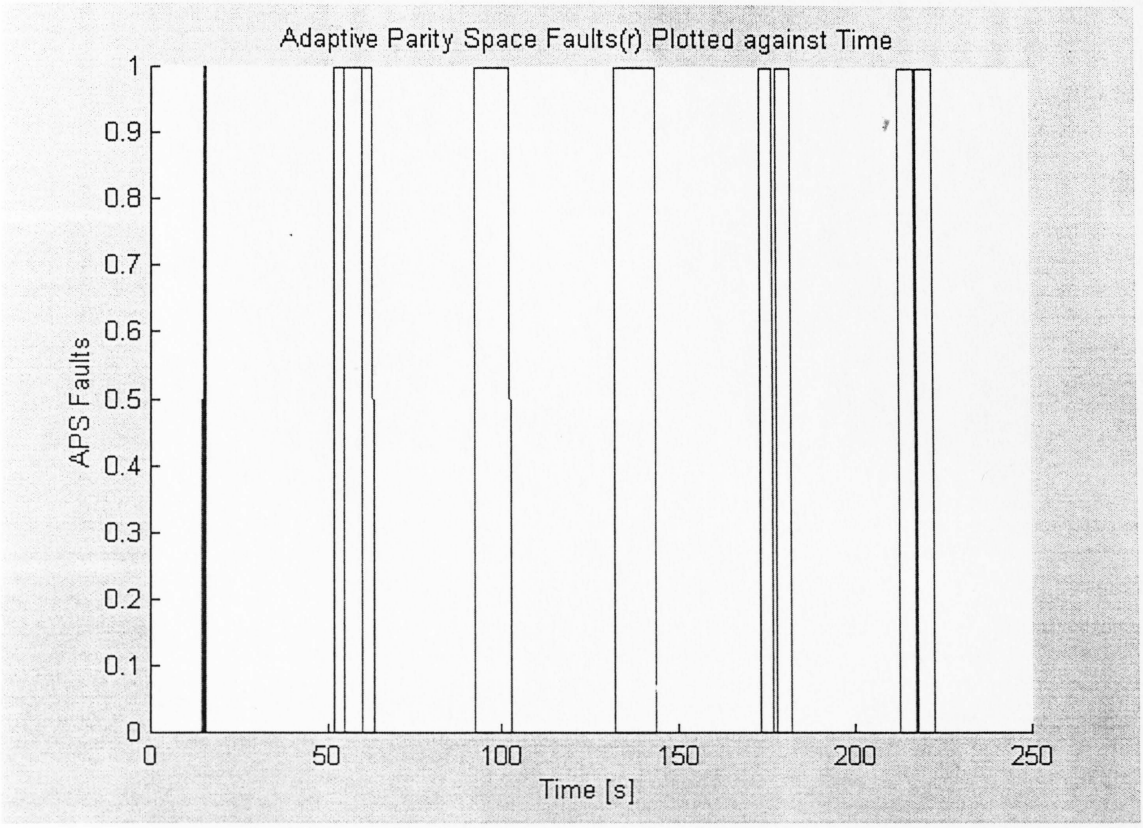


Figure H-27 Adaptive Parity Space Fault Indication for Simulation 3

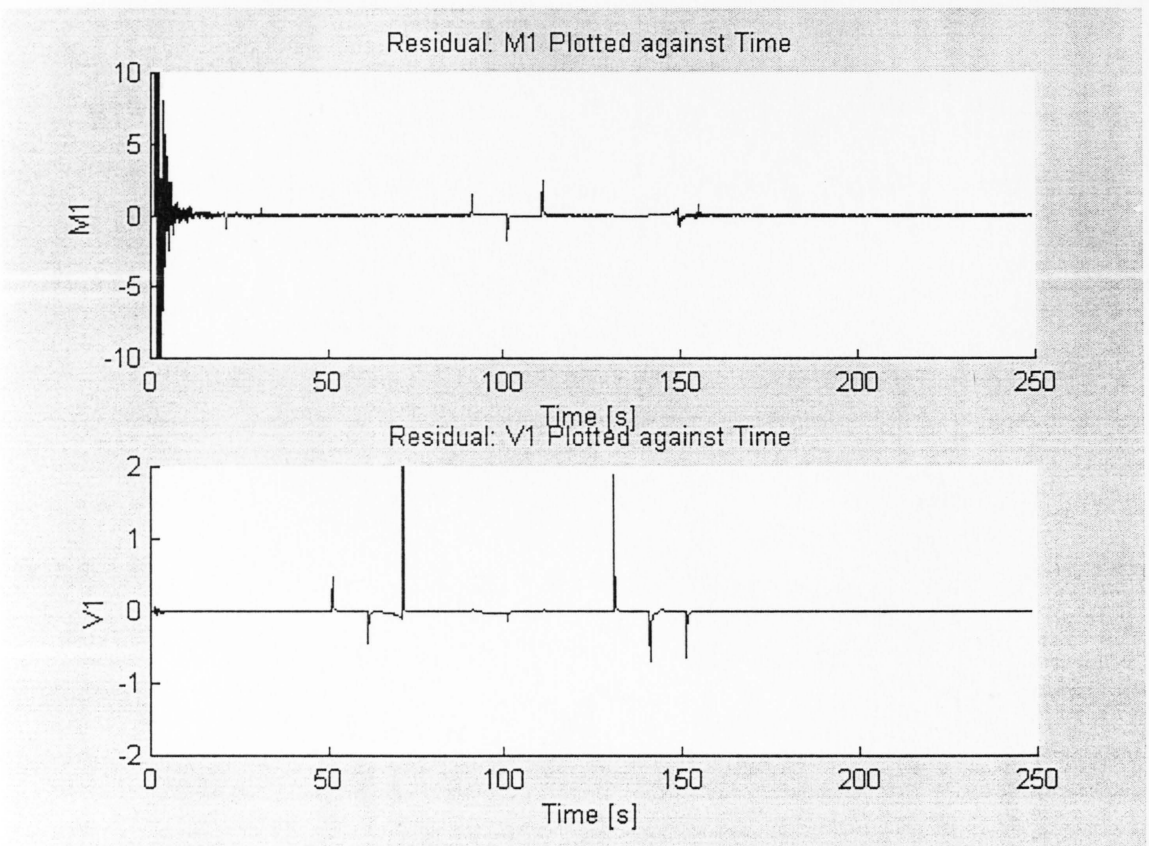


Figure H-28 Adaptive Parity Space Residuals M1 & V1 for Simulation 4

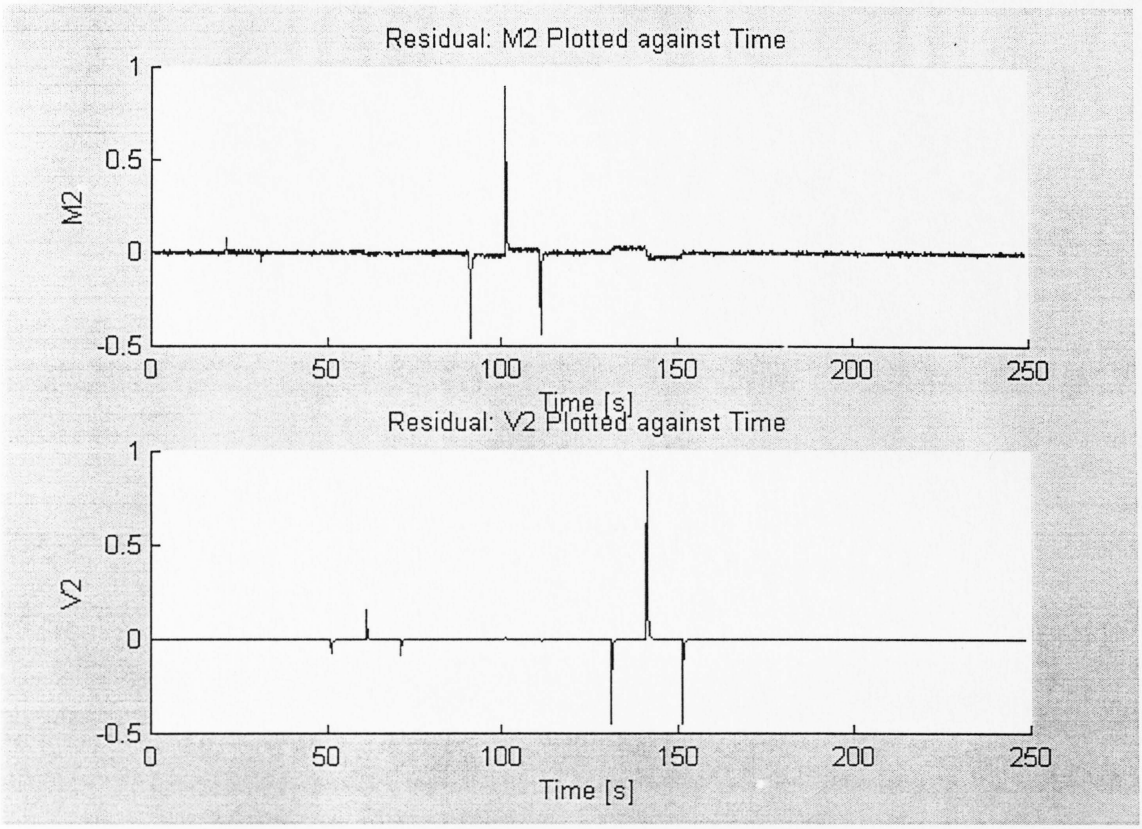


Figure H-29 Adaptive Parity Space Residuals M2 & V2 for Simulation 4

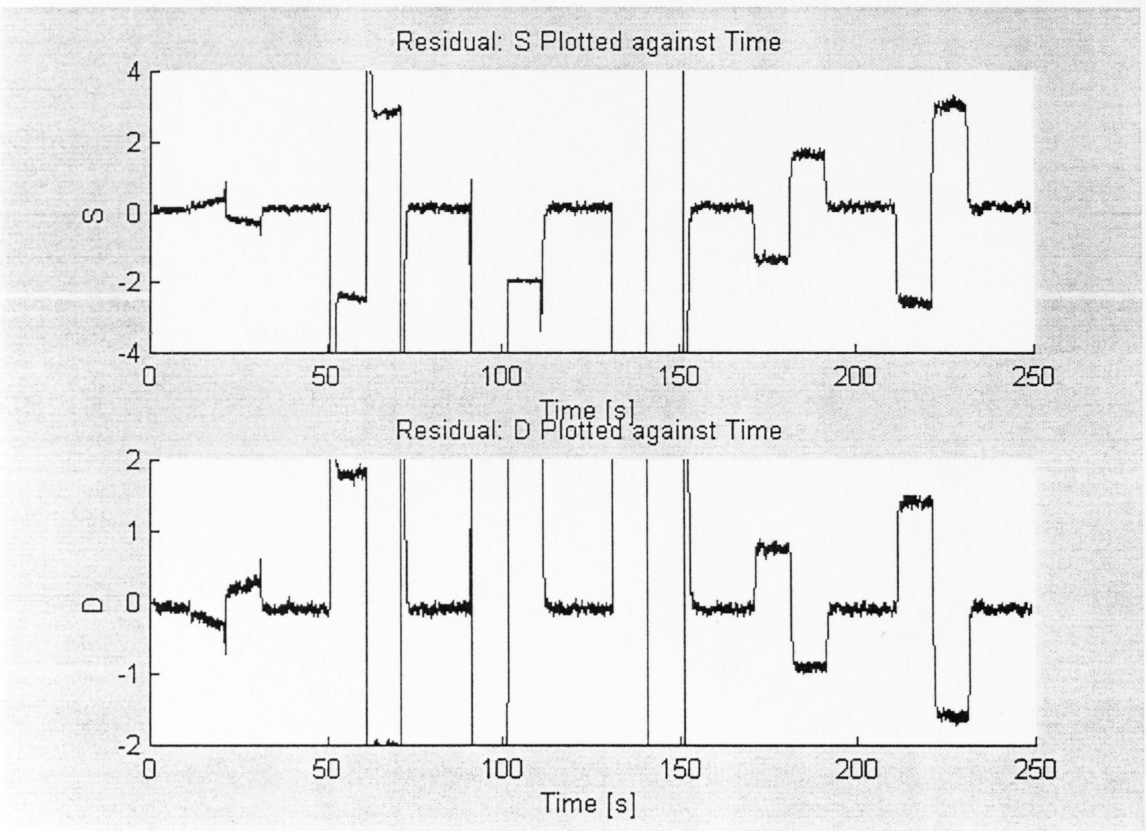


Figure H-30 Adaptive Parity Space Residuals S & D for Simulation 4

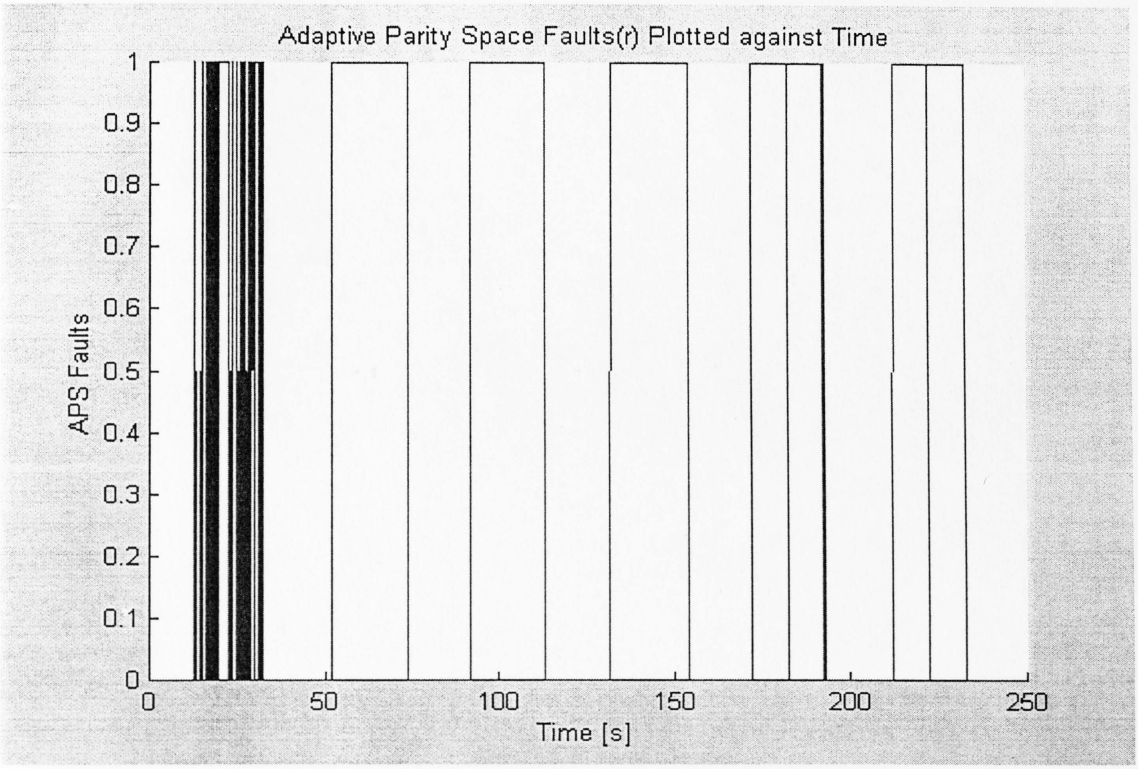


Figure H-31 Adaptive Parity Space Fault Indication for Simulation 4

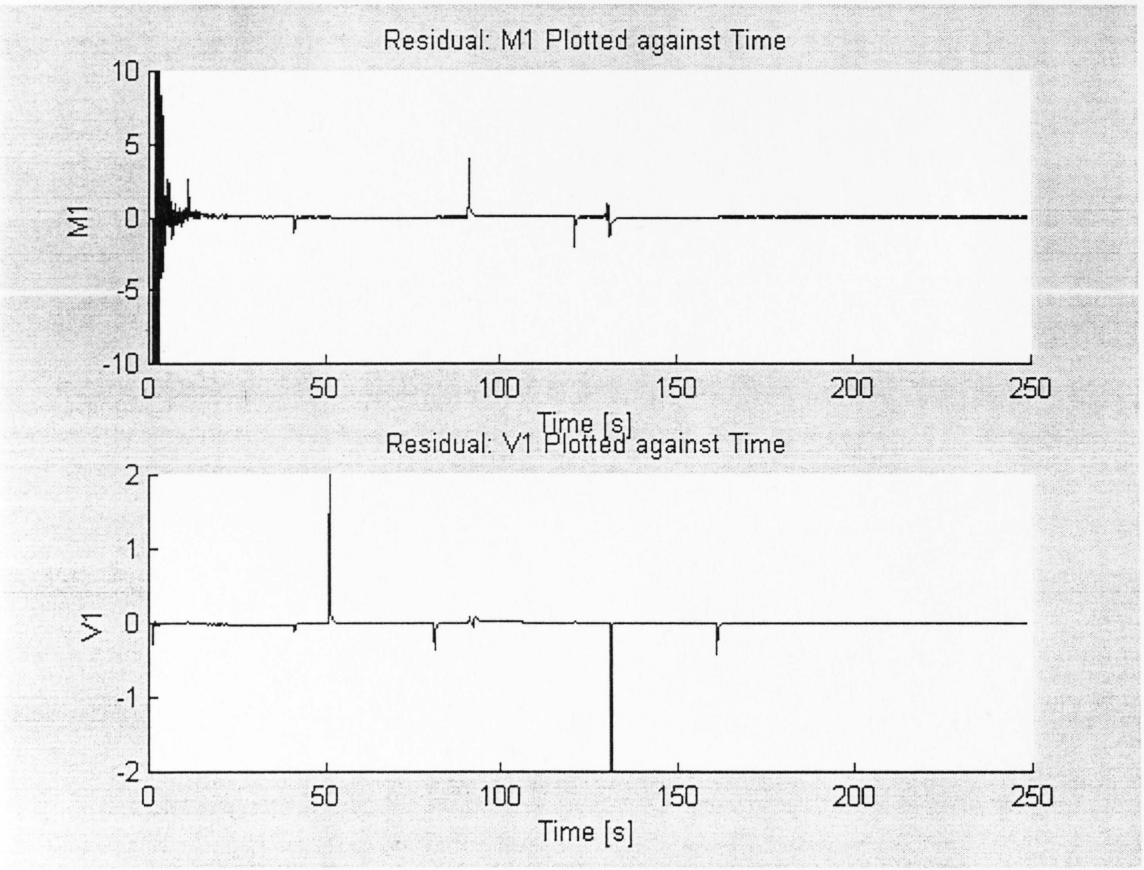


Figure H-32 Adaptive Parity Space Residuals M1 & V1 for Simulation 5

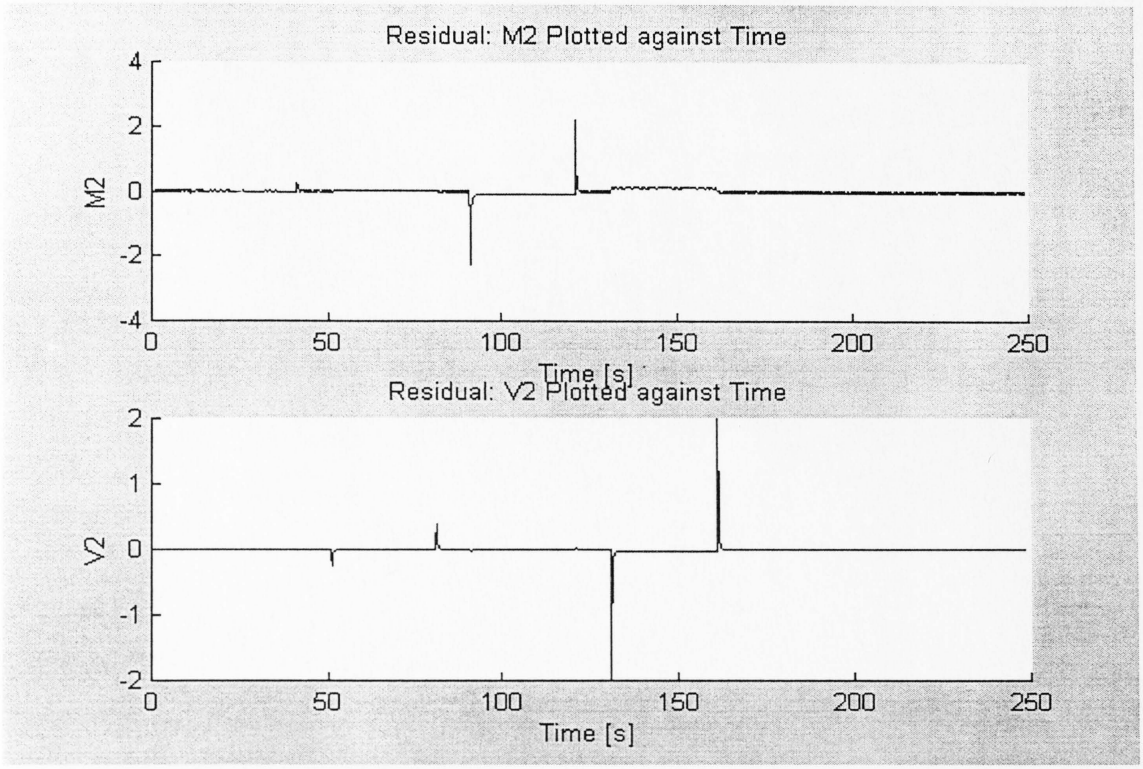


Figure H-33 Adaptive Parity Space Residuals M2 & V2 for Simulation 5

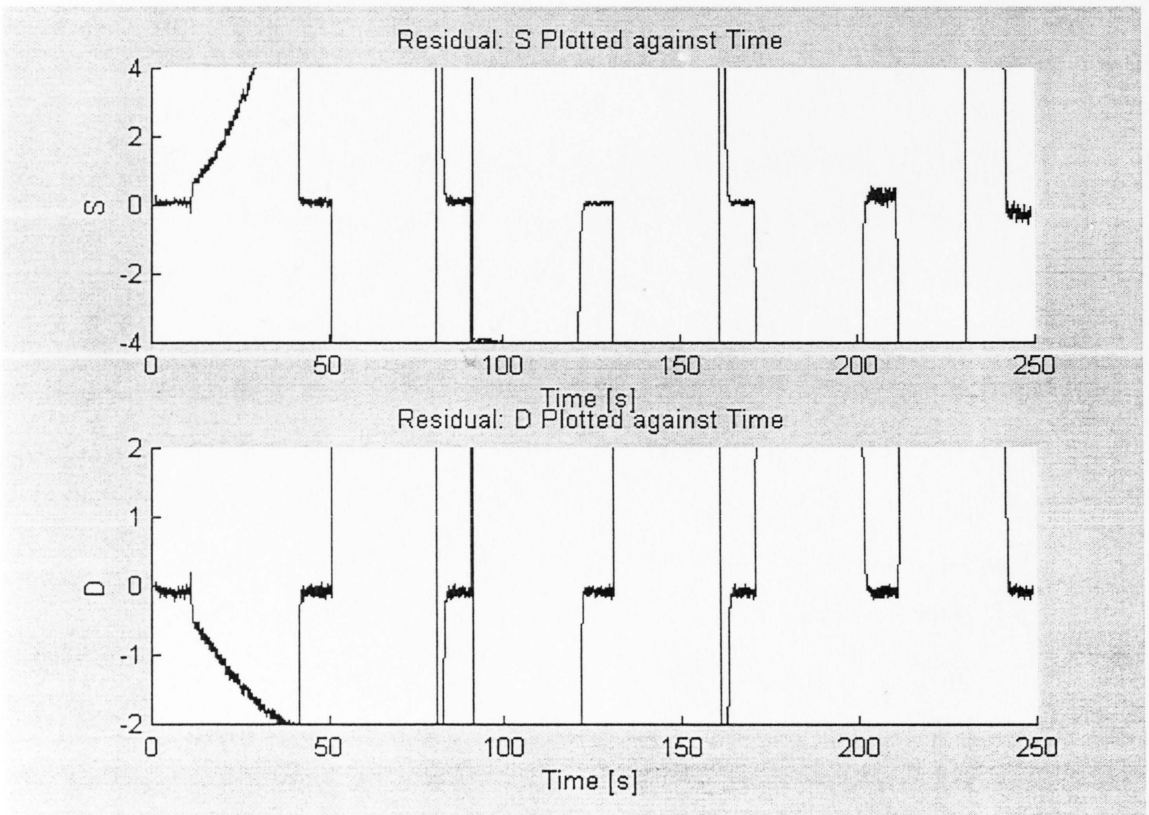


Figure H-34 Adaptive Parity Space Residuals S & D for Simulation 5

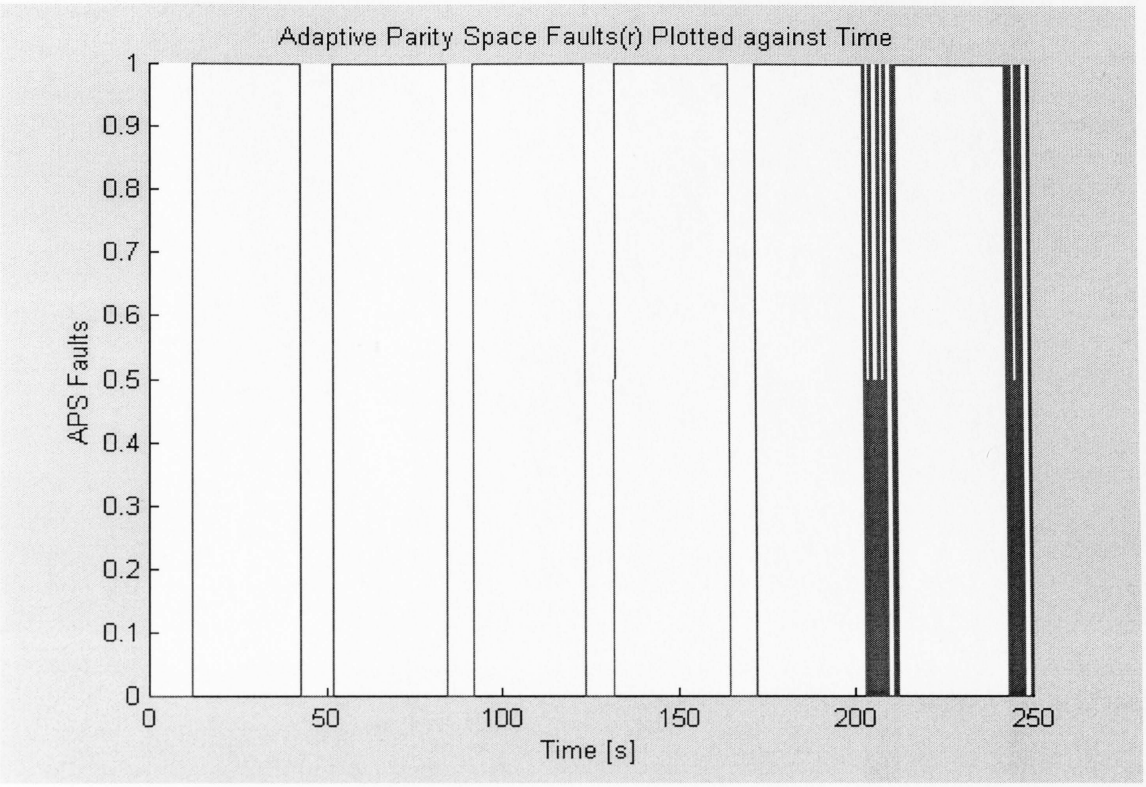


Figure H-35 Adaptive Parity Space Fault Indication for Simulation 5

Comment:

From the fault indication graphs of the adaptive parity space method of fault detection, it can be seen that the performance of the method is fairly good, in that it is able to detect most of the faults, irrespective of fault amplitude, duration, or waveform. The only exception being the for the Sensor 1 scale fault, which is only seen to be detected in Simulation 2. However, as the fault amplitude and duration become large, spurious faults are detected, which cause the performance of the method to be undermined. This problem is easily overcome by setting the residual thresholds at a slightly higher value, which would overcome the problem of spurious fault detection, but decrease the performance of the correct fault detection. The duration and amplitude of the faults that need to be detected thus determine the values of the threshold settings. By inspection of the residual graphs, it was found that only Residual S caused the spurious faults to be detected. So with the threshold setting increased from 0.4 to 0.8 for Residual S, the following fault indication graph was obtained.

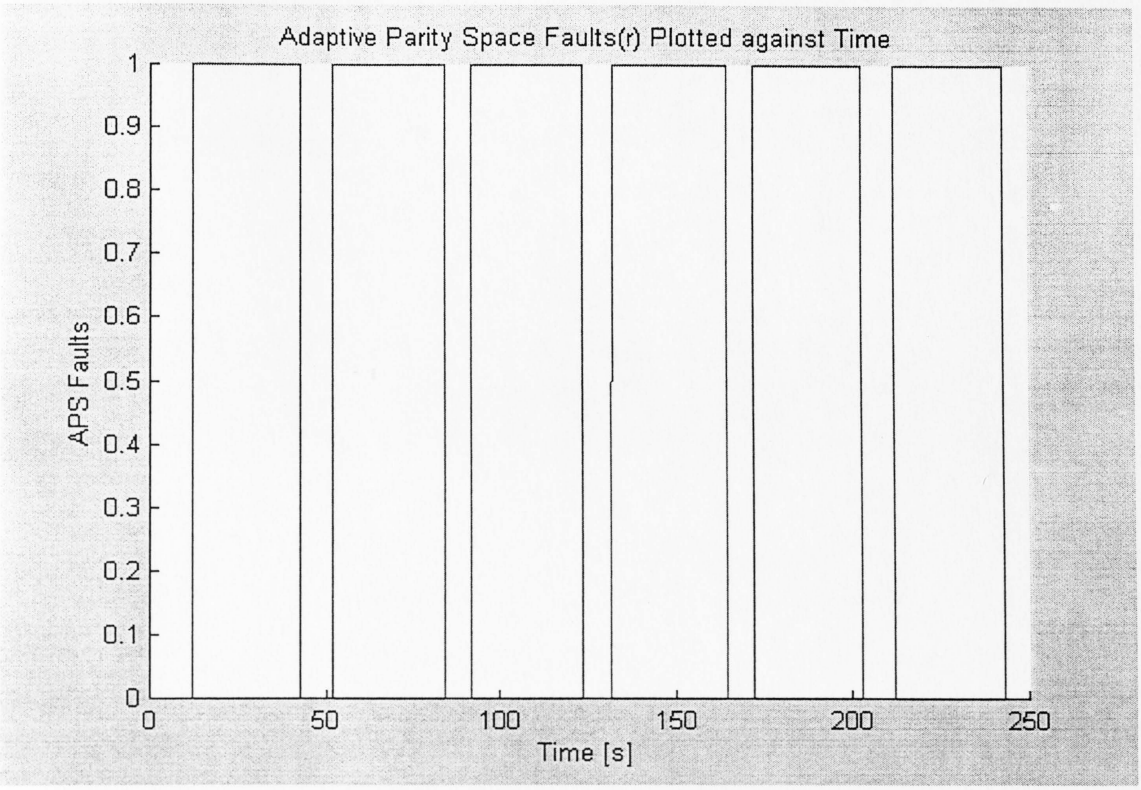


Figure H-36 Adaptive Parity Space new Fault Indication for Simulation 5

Comment:

The new fault indication graph shows that most of the spurious faults have disappeared (the spurious faults that are present are due to the delay in the residual decay). The fault detection performance shows:

Percentage of Correct Fault Detection: 95.88%

Percentage of Spurious Fault Detection: 14.93%

Percentage of Correlation: 86.12%

The percentage of correlation obtained is the highest when compared to the percentages of correlation for all the fault detection methods. This implies that the new threshold setting should be used.

H.3 Observer Method

The following graphs show the residuals and fault indication graphs for the various simulation parameters described above, for the observer method of fault detection (as described in Section 5.4.1).

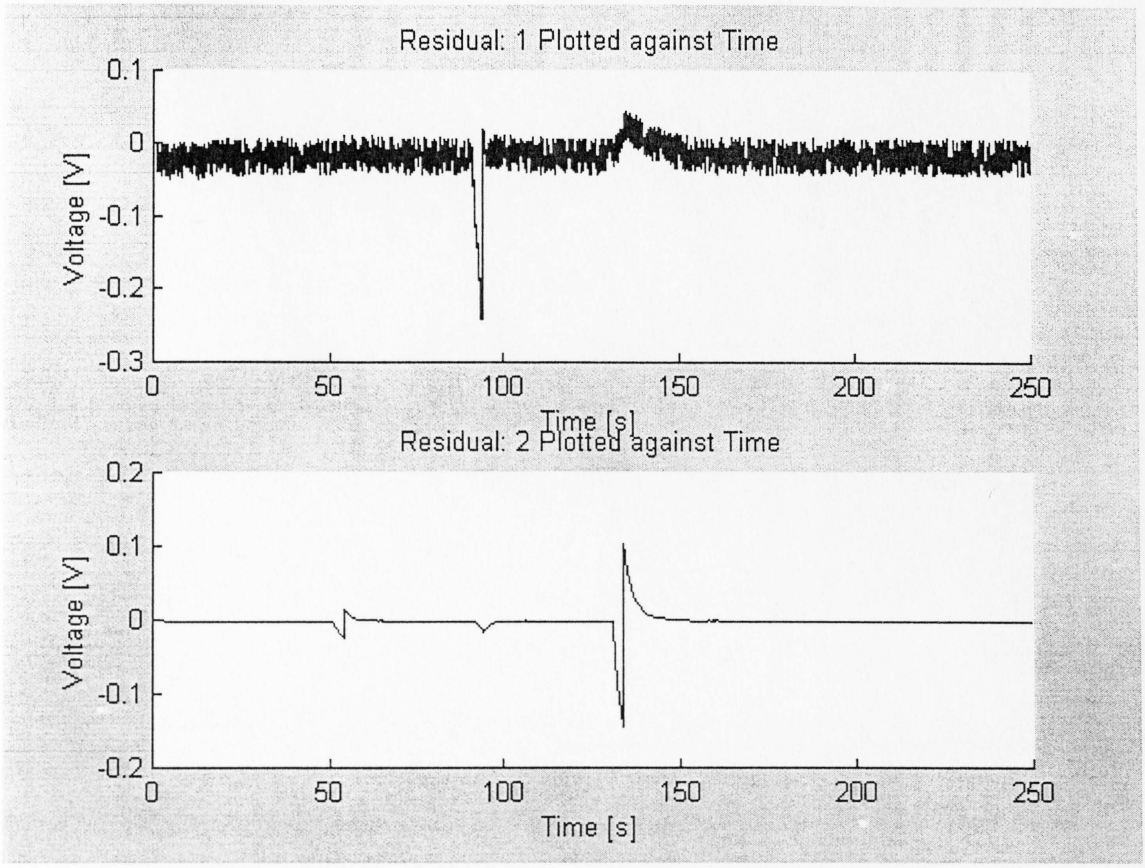


Figure H-37 Observer Residuals 1 & 2 for Simulation 1

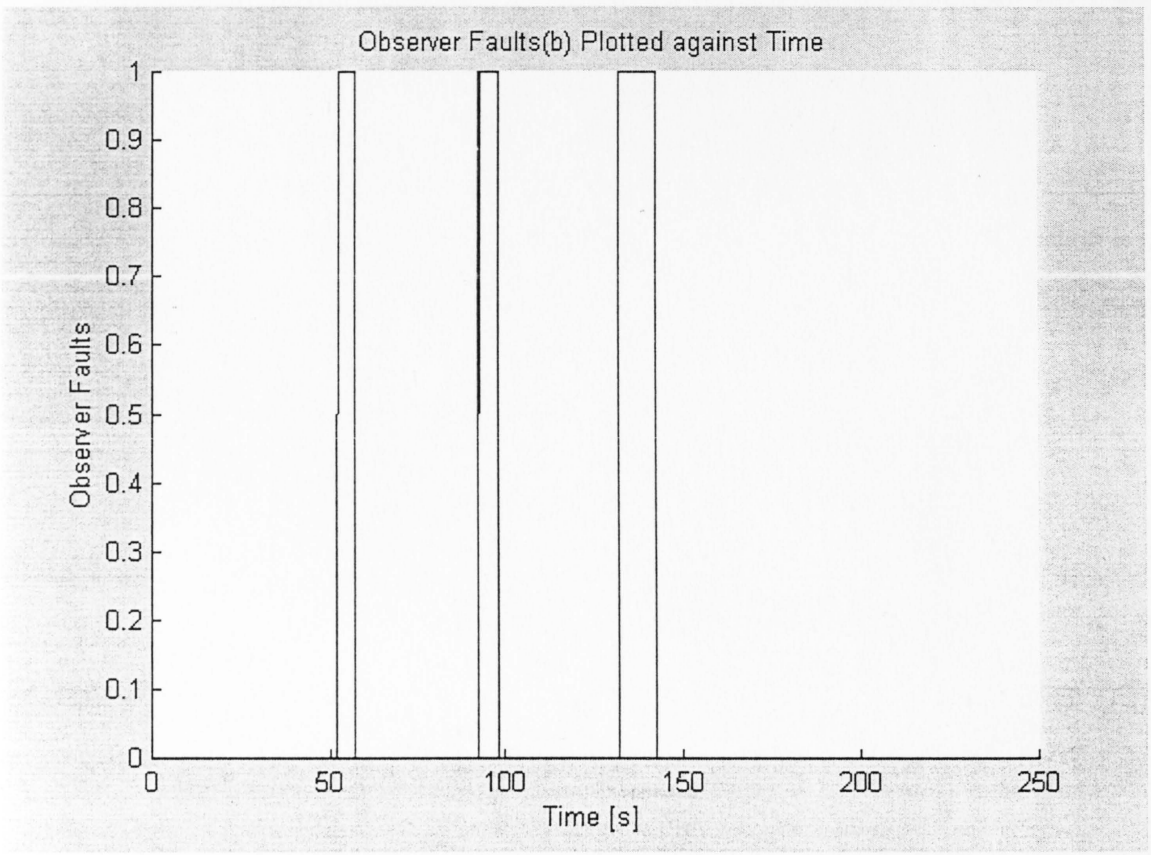


Figure H-38 Observer Fault Detection for Simulation 1

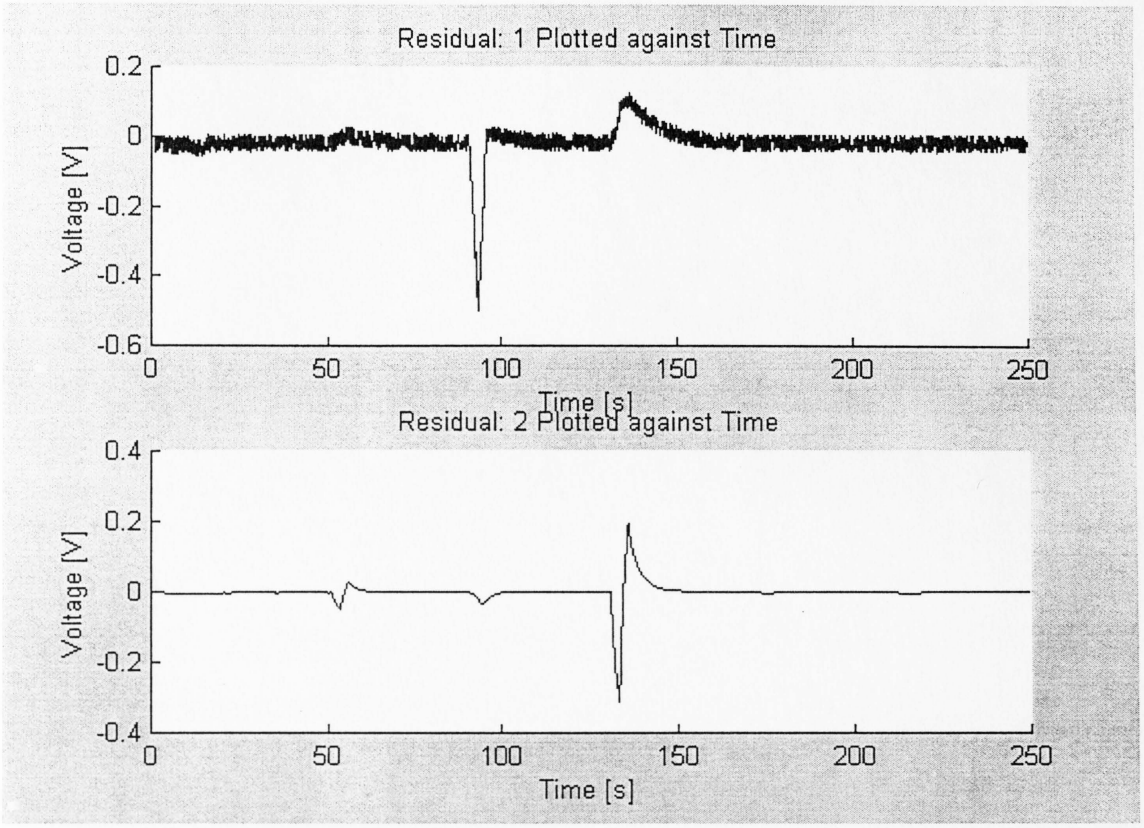


Figure H-39 Observer Residuals 1 & 2 for Simulation 2

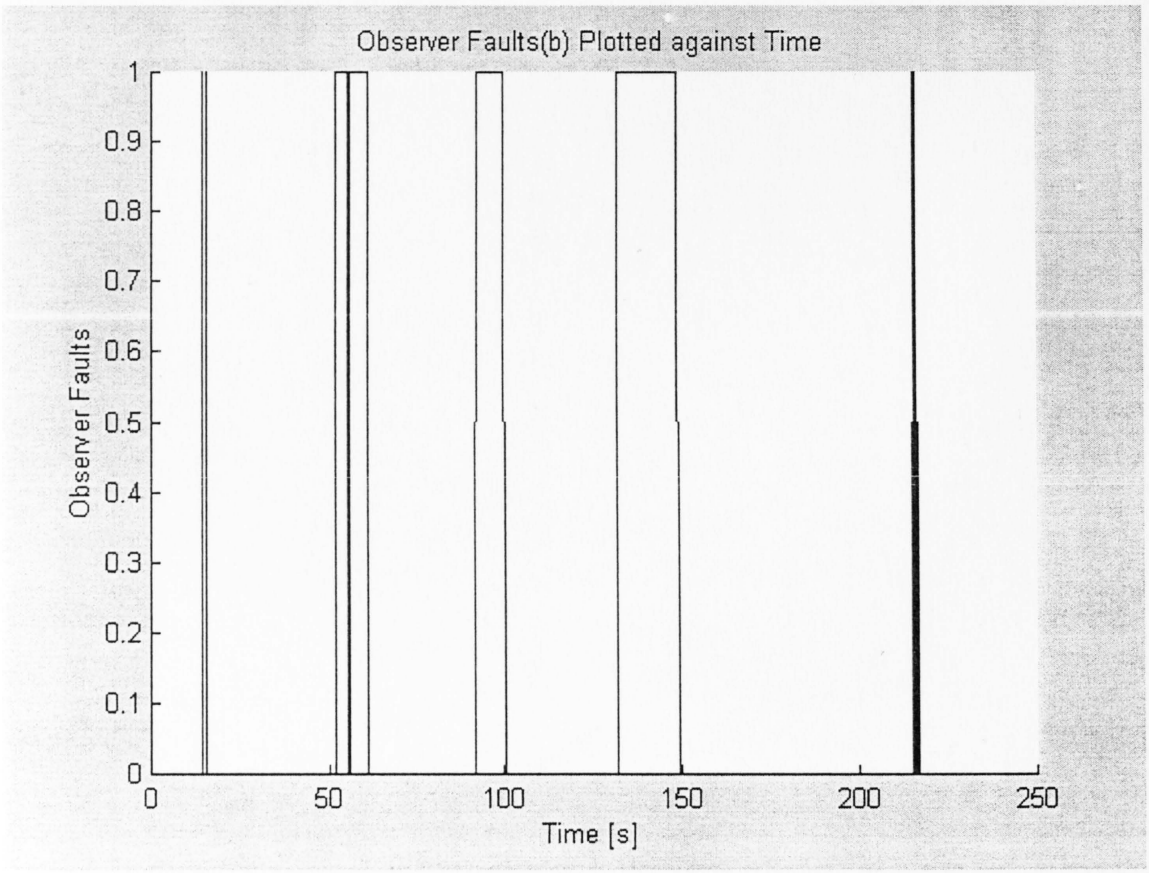


Figure H-40 Observer Fault Detection for Simulation 2

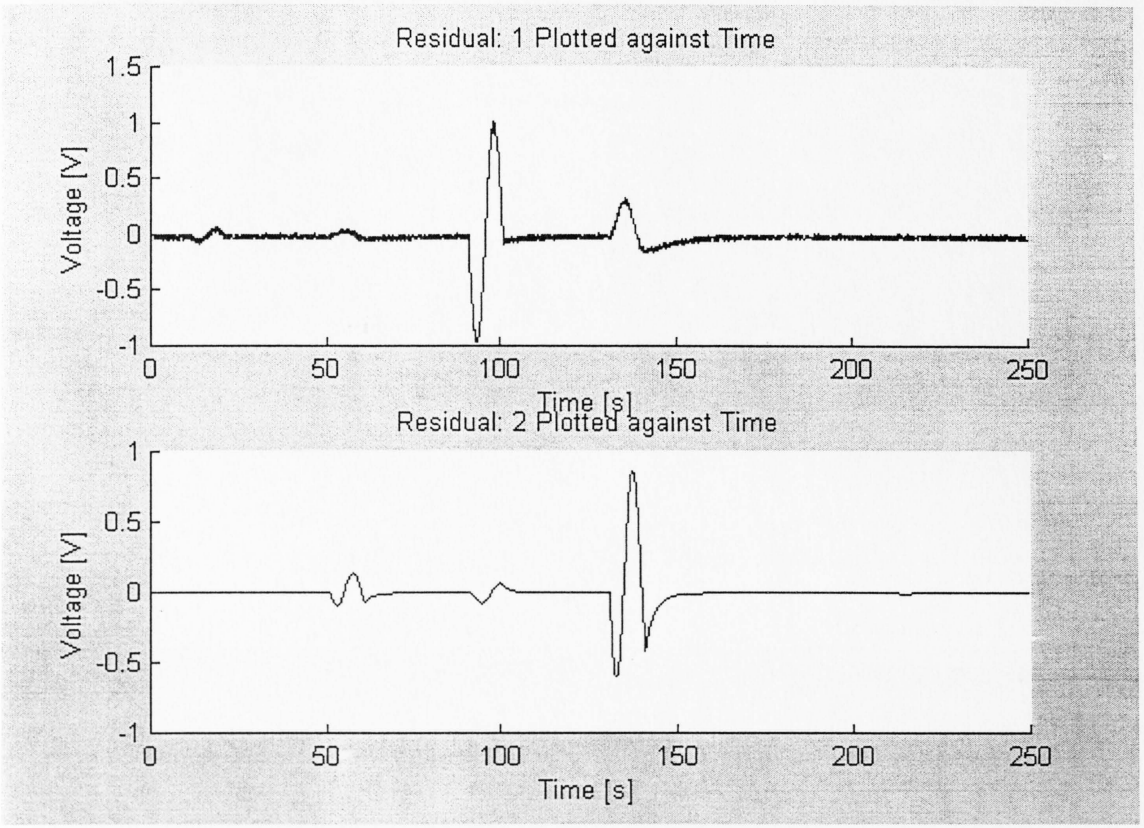


Figure H-41 Observer Residuals 1 & 2 for Simulation 3

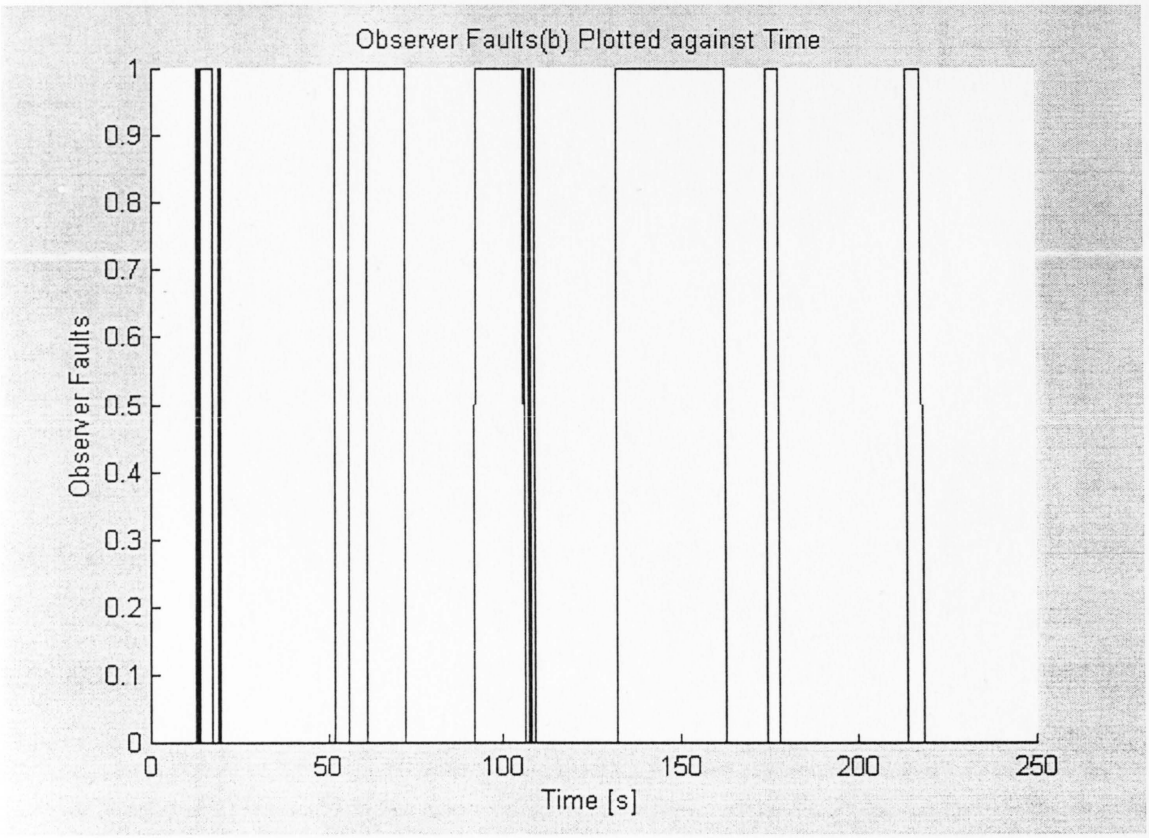


Figure H-42 Observer Fault Detection for Simulation 3

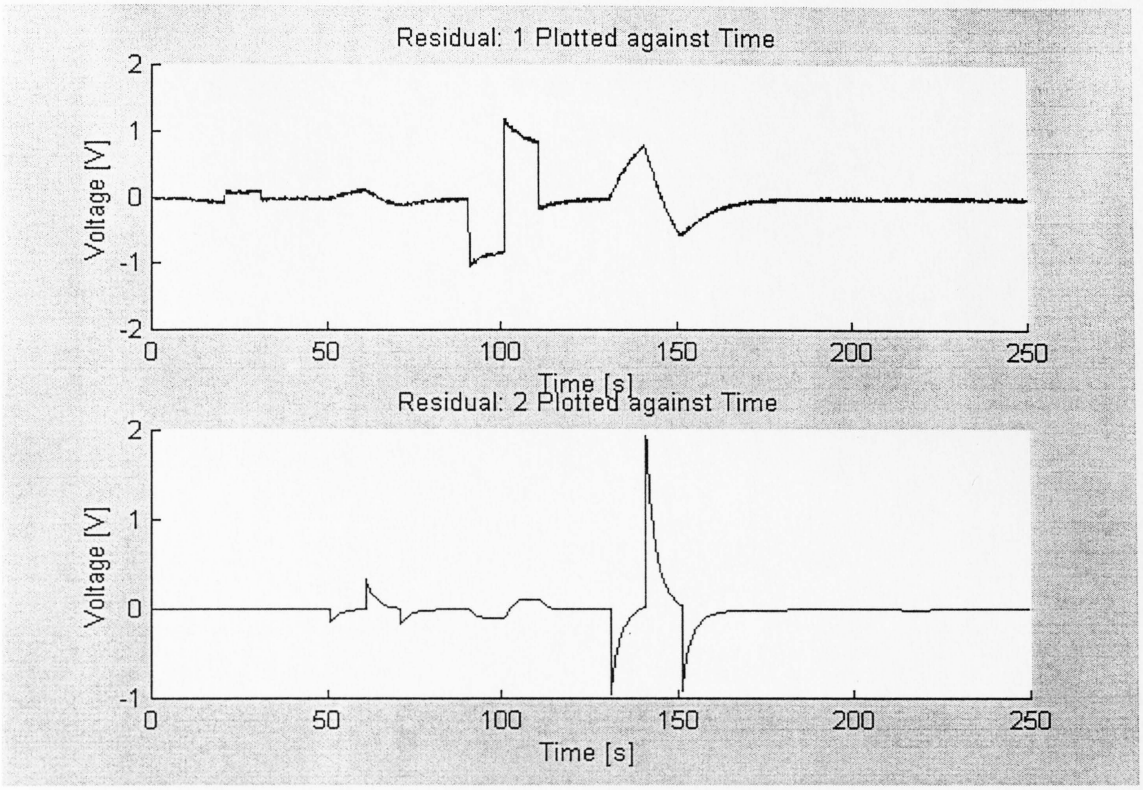


Figure H-43 Observer Residuals 1 & 2 for Simulation 4

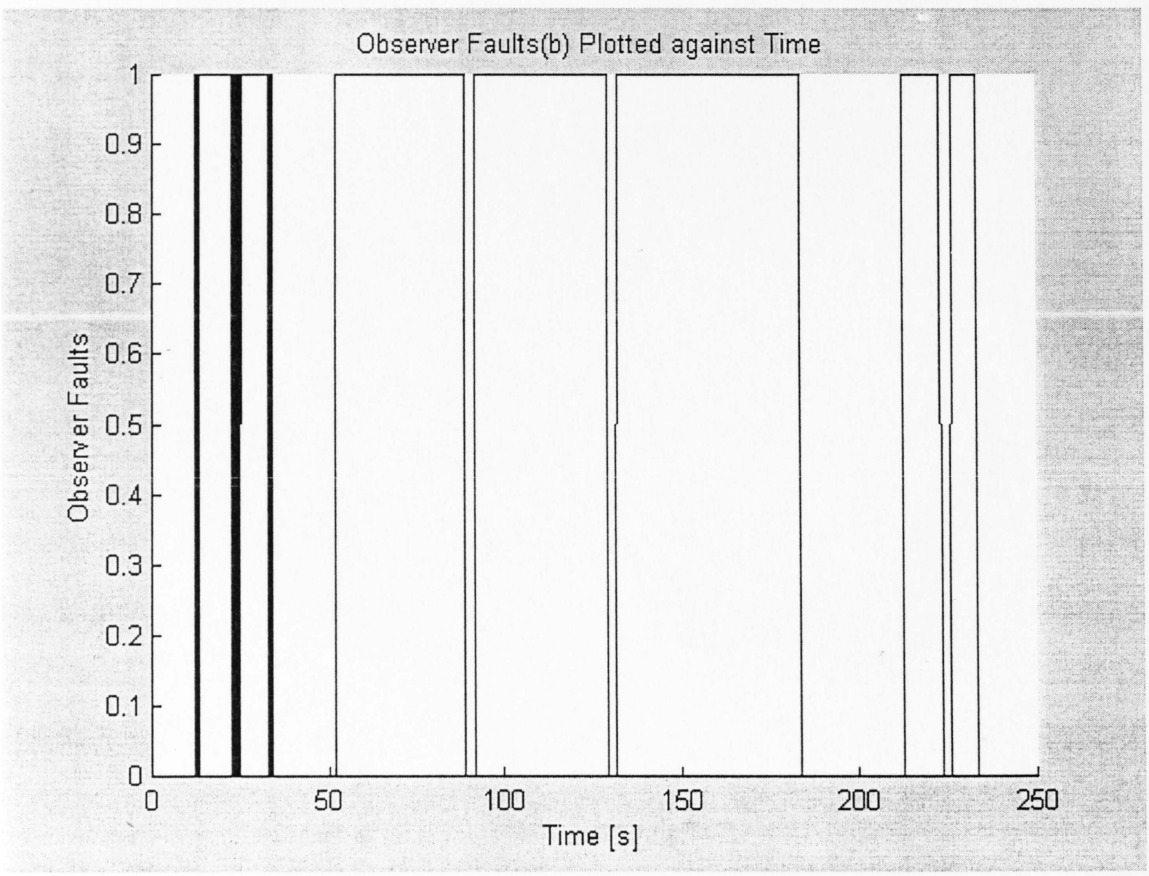


Figure H-44 Observer Fault Detection for Simulation

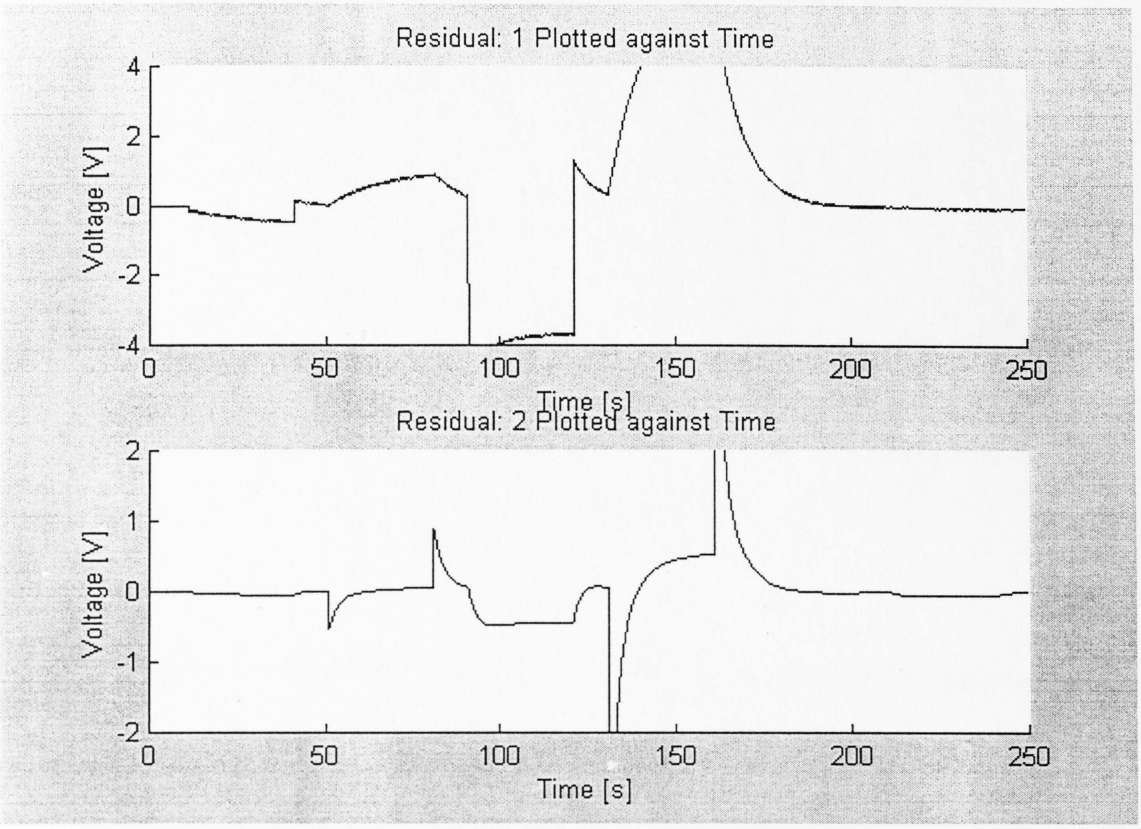


Figure H-45 Observer Residuals 1 & 2 for Simulation 5

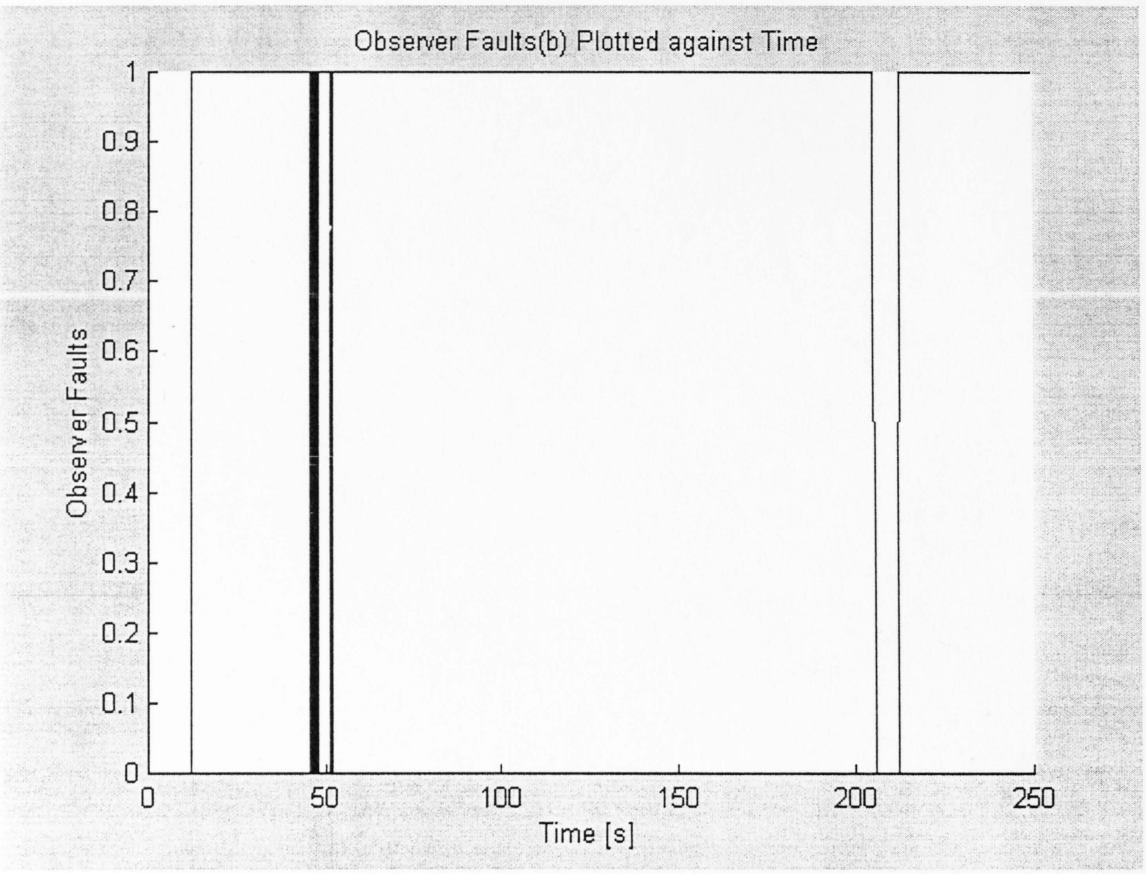


Figure H-46 Observer Fault Detection for Simulation 5

Comment:

The delay in the residual decay causes the observer method of fault detection to detect many spurious faults. As the fault duration increases, the percentage of correct faults increases and the spurious fault detection decreases, since the faults are close together. As a result, the percentage of correlation becomes higher for faults with a large duration. As the fault amplitude increases, the otherwise difficult to detect actuator faults are now detected, causing an increase in the percentage of correct faults detected, and thus an increase in the percentage correlation.

APPENDIX I: REVISED CORRELATION GRAPHS AND DATA FOR THE APS FAULT DETECTION METHOD

This appendix shows the percentages of correlation obtained for the adaptive parity space method of fault detection with the threshold setting increased from 0.4 to 0.8 for Residual S. The revised correlation graphs plotted against fault amplitude and duration, for the ramp, saw, sine, square and step waveforms follow. The corresponding percentages of correlation, are given at the end of the appendix.

I.1 Correlation Graphs

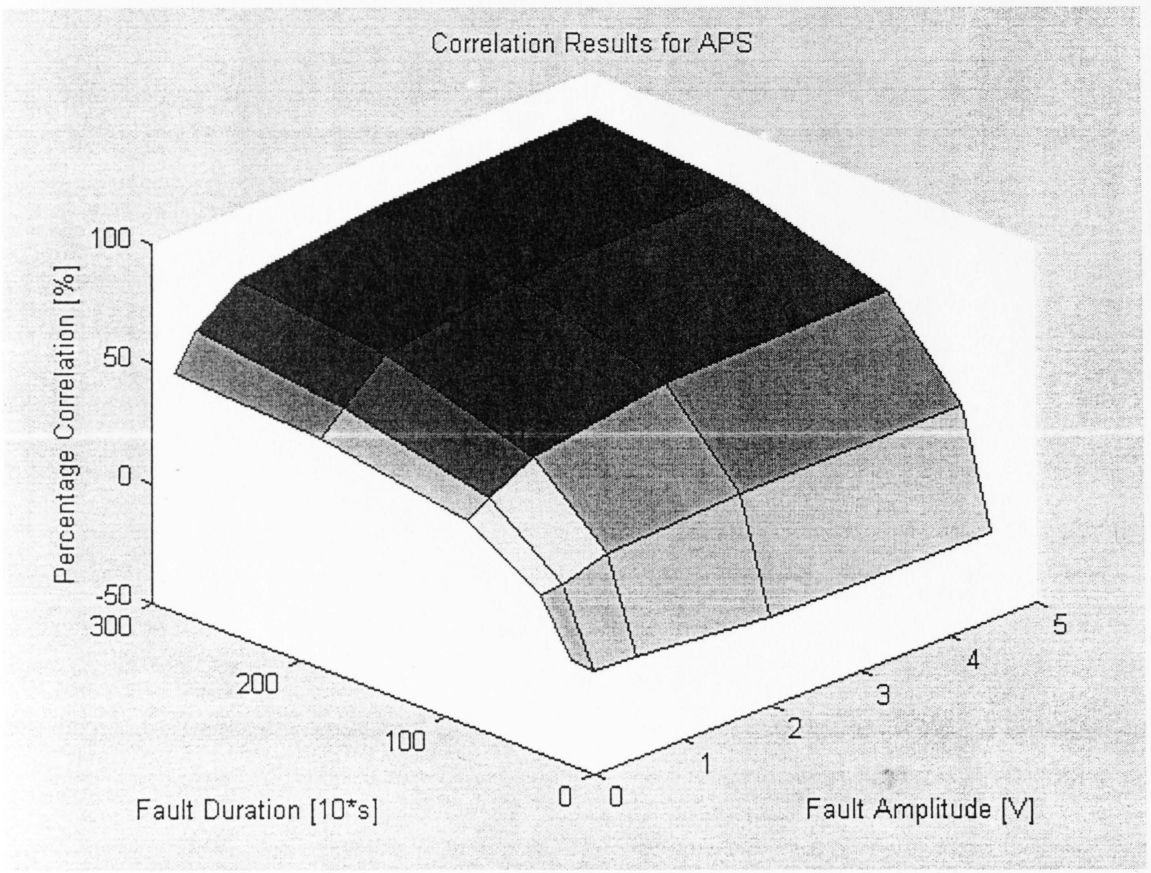


Figure I-1 New APS Correlation Graph for Ramp waveform

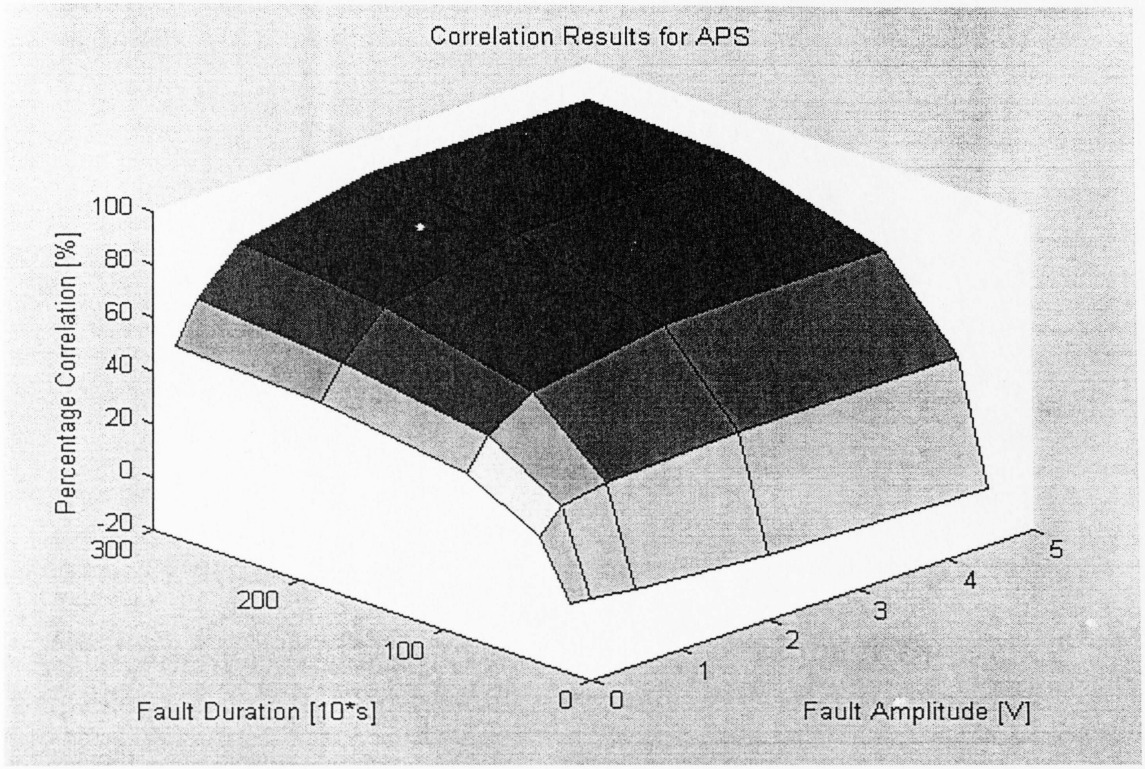


Figure I-2 New APS Correlation Graph for Saw waveform

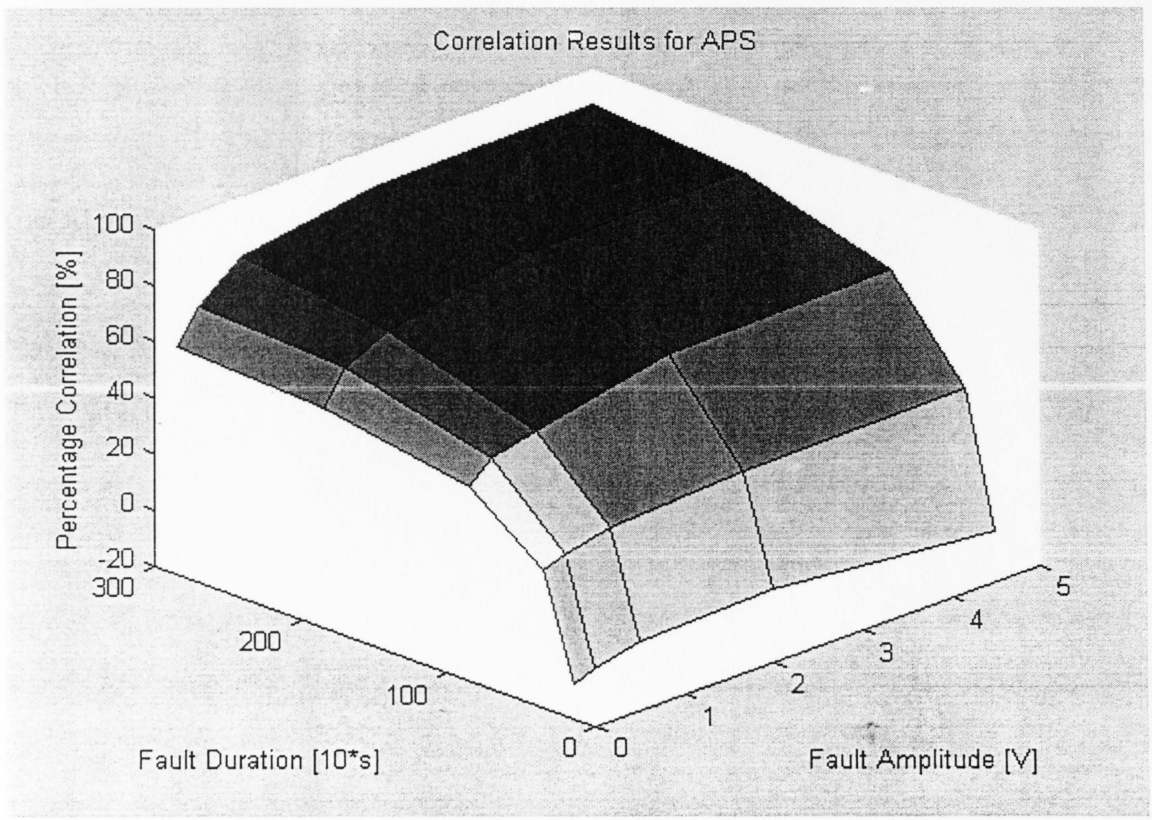


Figure I-3 New APS Correlation Graph for Sine waveform

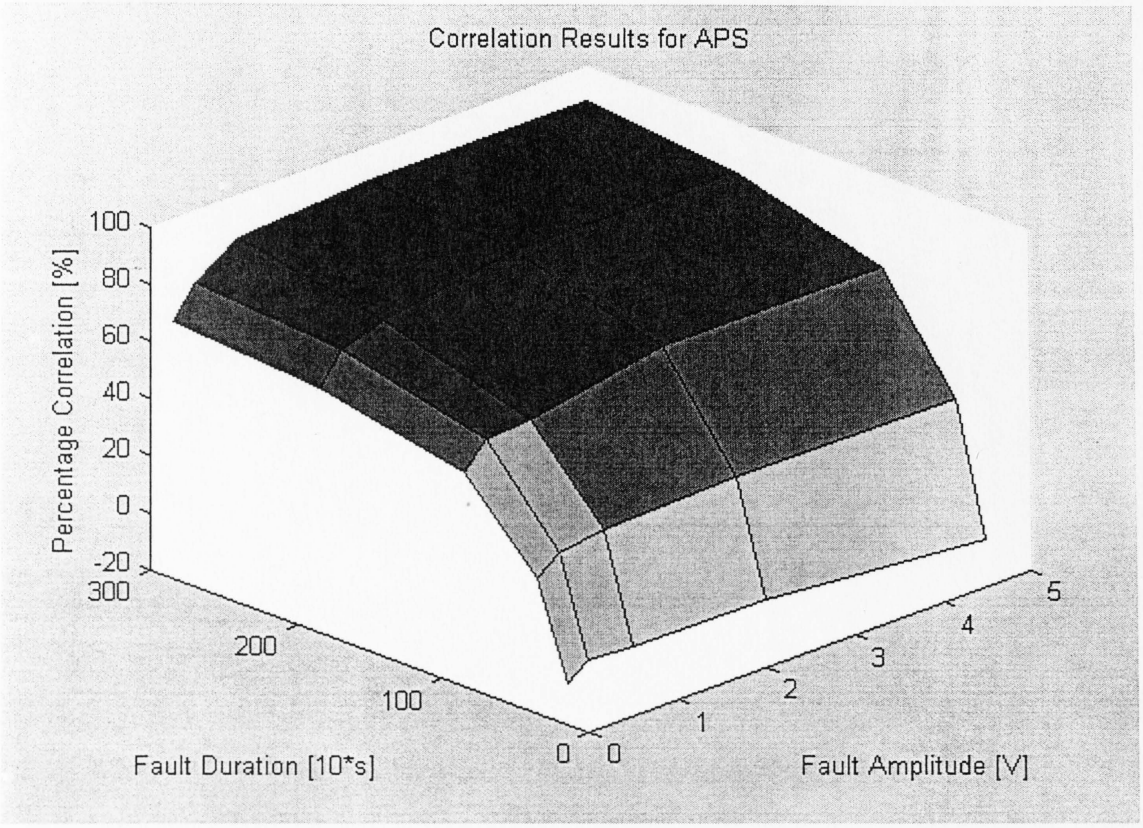


Figure I-4 New APS Correlation Graph for Square waveform

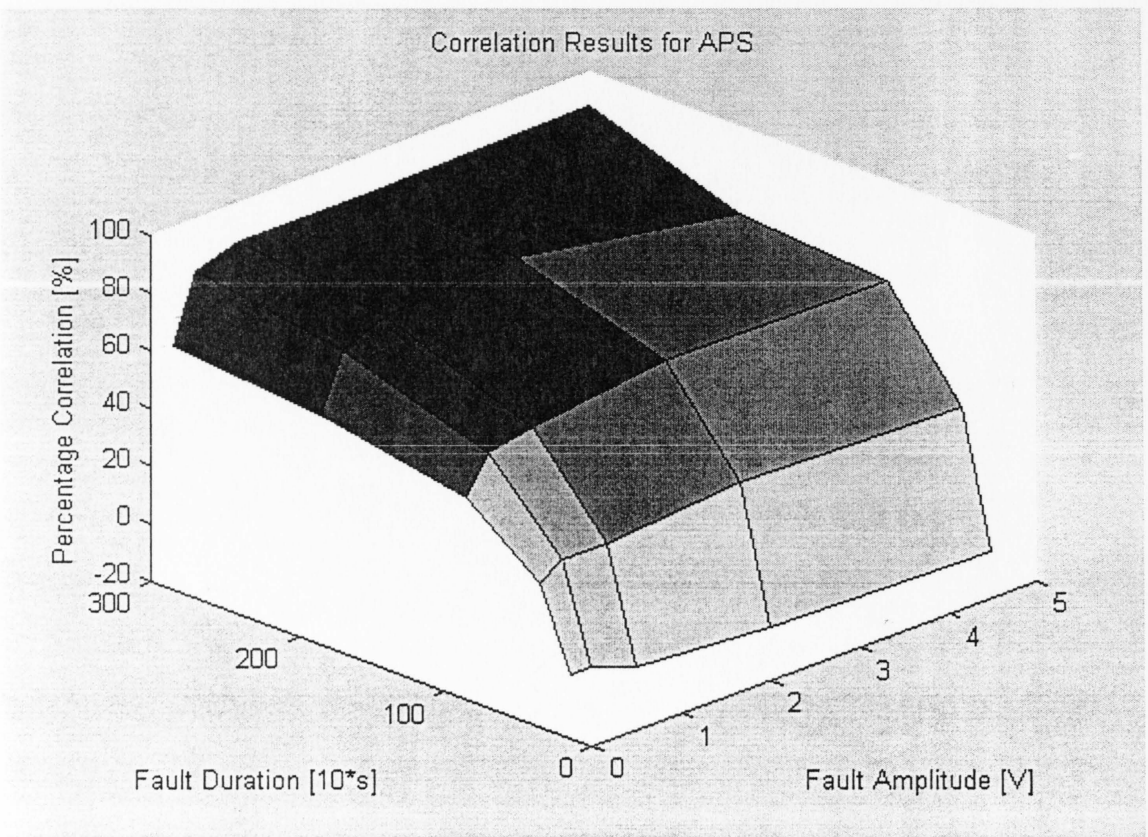


Figure I-5 New APS Correlation Graph for Step waveform

I.2 Correlation Results

The following percentages of correlation for the APS fault detection method are shown below, where:

A = Fault Amplitude [V]

W = Fault Duration [10*s]

The shapes of the waveforms are highlighted.

Ramp:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 81.8333
W =
200
Percentage Correlation Adaptive Parity Space: 75.8333
W =
100
Percentage Correlation Adaptive Parity Space: 56.6667
W =
50
Percentage Correlation Adaptive Parity Space: 20
W =
30
Percentage Correlation Adaptive Parity Space: -28.3333

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 79.1111
W =
200
Percentage Correlation Adaptive Parity Space: 72.5
W =
100
Percentage Correlation Adaptive Parity Space: 54
W =
50
Percentage Correlation Adaptive Parity Space: 19.3333
W =
30
Percentage Correlation Adaptive Parity Space: -27.2222

A =
1
W =
300
Percentage Correlation Adaptive Parity Space: 70.0556
W =

200
Percentage Correlation Adaptive Parity Space: 63.0833
W =
100
Percentage Correlation Adaptive Parity Space: 43.8333
W =
50
Percentage Correlation Adaptive Parity Space: 15.6667
W =
30
Percentage Correlation Adaptive Parity Space: -22.7778

A =
0.5000
W =
300
Percentage Correlation Adaptive Parity Space: 56.5556
W =
200
Percentage Correlation Adaptive Parity Space: 48
W =
100
Percentage Correlation Adaptive Parity Space: 33.8333
W =
50
Percentage Correlation Adaptive Parity Space: 9.6667
W =
30
Percentage Correlation Adaptive Parity Space: -21.6667

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 42.8889
W =
200
Percentage Correlation Adaptive Parity Space: 38.5833
W =
100
Percentage Correlation Adaptive Parity Space: 28.1667
W =
50
Percentage Correlation Adaptive Parity Space: 9.3333
W =
30
Percentage Correlation Adaptive Parity Space: -13.8889

Saw:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 86.4444
W =
200
Percentage Correlation Adaptive Parity Space: 83.4167
W =

```
   100
Percentage Correlation Adaptive Parity Space: 67.6667
W =
    50
Percentage Correlation Adaptive Parity Space: 36
W =
    30
Percentage Correlation Adaptive Parity Space: -10

A =
    2.5000
W =
   300
Percentage Correlation Adaptive Parity Space: 86.4444
W =
   200
Percentage Correlation Adaptive Parity Space: 82.5
W =
   100
Percentage Correlation Adaptive Parity Space: 67
W =
    50
Percentage Correlation Adaptive Parity Space: 36.3333
W =
    30
Percentage Correlation Adaptive Parity Space: -6.6667

A =
     1
W =
   300
Percentage Correlation Adaptive Parity Space: 76.7222
W =
   200
Percentage Correlation Adaptive Parity Space: 71.8333
W =
   100
Percentage Correlation Adaptive Parity Space: 58.6667
W =
    50
Percentage Correlation Adaptive Parity Space: 34
W =
    30
Percentage Correlation Adaptive Parity Space: -2.7778

A =
    0.5000
W =
   300
Percentage Correlation Adaptive Parity Space: 61.3889
W =
   200
Percentage Correlation Adaptive Parity Space: 56.5
W =
   100
Percentage Correlation Adaptive Parity Space: 48
W =
    50
Percentage Correlation Adaptive Parity Space: 31
W =
```

30
Percentage Correlation Adaptive Parity Space: -0.55556

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 46.3333
W =
200
Percentage Correlation Adaptive Parity Space: 44.0833
W =
100
Percentage Correlation Adaptive Parity Space: 36.3333
W =
50
Percentage Correlation Adaptive Parity Space: 22
W =
30
Percentage Correlation Adaptive Parity Space: 0

Sine:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 87.5
W =
200
Percentage Correlation Adaptive Parity Space: 82.5833
W =
100
Percentage Correlation Adaptive Parity Space: 66.3333
W =
50
Percentage Correlation Adaptive Parity Space: 33.3333
W =
30
Percentage Correlation Adaptive Parity Space: -13.3333

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 86.3889
W =
200
Percentage Correlation Adaptive Parity Space: 81.25
W =
100
Percentage Correlation Adaptive Parity Space: 65.1667
W =
50
Percentage Correlation Adaptive Parity Space: 32.6667
W =
30
Percentage Correlation Adaptive Parity Space: -5.5556

A =
1
W =
300
Percentage Correlation Adaptive Parity Space: 79.0556
W =
200
Percentage Correlation Adaptive Parity Space: 71.4167
W =
100
Percentage Correlation Adaptive Parity Space: 55
W =
50
Percentage Correlation Adaptive Parity Space: 29.6667
W =
30
Percentage Correlation Adaptive Parity Space: -6.6667

A =
0.5000
W =
300
Percentage Correlation Adaptive Parity Space: 66.7778
W =
200
Percentage Correlation Adaptive Parity Space: 63.75
W =
100
Percentage Correlation Adaptive Parity Space: 50.8333
W =
50
Percentage Correlation Adaptive Parity Space: 25.6667
W =
30
Percentage Correlation Adaptive Parity Space: -10.5556

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 55.0556
W =
200
Percentage Correlation Adaptive Parity Space: 52.6667
W =
100
Percentage Correlation Adaptive Parity Space: 43.1667
W =
50
Percentage Correlation Adaptive Parity Space: 23
W =
30
Percentage Correlation Adaptive Parity Space: -13.3333

Square:

A =
5
W =

```
  300
Percentage Correlation Adaptive Parity Space: 88.1667
W =
  200
Percentage Correlation Adaptive Parity Space: 82.5833
W =
  100
Percentage Correlation Adaptive Parity Space: 68
W =
   50
Percentage Correlation Adaptive Parity Space: 30.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -15

A =
   2.5000
W =
  300
Percentage Correlation Adaptive Parity Space: 88.6667
W =
  200
Percentage Correlation Adaptive Parity Space: 83.1667
W =
  100
Percentage Correlation Adaptive Parity Space: 68.3333
W =
   50
Percentage Correlation Adaptive Parity Space: 32
W =
   30
Percentage Correlation Adaptive Parity Space: -7.7778

A =
    1
W =
  300
Percentage Correlation Adaptive Parity Space: 85.6111
W =
  200
Percentage Correlation Adaptive Parity Space: 75.75
W =
  100
Percentage Correlation Adaptive Parity Space: 59.5
W =
   50
Percentage Correlation Adaptive Parity Space: 30.3333
W =
   30
Percentage Correlation Adaptive Parity Space: -7.2222

A =
   0.5000
W =
  300
Percentage Correlation Adaptive Parity Space: 75.1111
W =
  200
Percentage Correlation Adaptive Parity Space: 70.5833
W =
```

100
Percentage Correlation Adaptive Parity Space: 58.6667
W =
50
Percentage Correlation Adaptive Parity Space: 28
W =
30
Percentage Correlation Adaptive Parity Space: -6.1111

A =
0.2500
W =
300
Percentage Correlation Adaptive Parity Space: 64.1667
W =
200
Percentage Correlation Adaptive Parity Space: 60.0833
W =
100
Percentage Correlation Adaptive Parity Space: 49.6667
W =
50
Percentage Correlation Adaptive Parity Space: 22
W =
30
Percentage Correlation Adaptive Parity Space: -11.1111

Step:

A =
5
W =
300
Percentage Correlation Adaptive Parity Space: 88.4444
W =
200
Percentage Correlation Adaptive Parity Space: 69.3333
W =
100
Percentage Correlation Adaptive Parity Space: 65.1667
W =
50
Percentage Correlation Adaptive Parity Space: 30.6667
W =
30
Percentage Correlation Adaptive Parity Space: -15.5556

A =
2.5000
W =
300
Percentage Correlation Adaptive Parity Space: 86.7222
W =
200
Percentage Correlation Adaptive Parity Space: 83
W =
100
Percentage Correlation Adaptive Parity Space: 66.1667

```
W =
   50
Percentage Correlation Adaptive Parity Space: 33
W =
   30
Percentage Correlation Adaptive Parity Space: -13.3333

A =
    1
W =
  300
Percentage Correlation Adaptive Parity Space: 86.5556
W =
  200
Percentage Correlation Adaptive Parity Space: 79.5833
W =
  100
Percentage Correlation Adaptive Parity Space: 60.3333
W =
   50
Percentage Correlation Adaptive Parity Space: 29
W =
   30
Percentage Correlation Adaptive Parity Space: -10

A =
   0.5000
W =
  300
Percentage Correlation Adaptive Parity Space: 81.9444
W =
  200
Percentage Correlation Adaptive Parity Space: 73.75
W =
  100
Percentage Correlation Adaptive Parity Space: 57
W =
   50
Percentage Correlation Adaptive Parity Space: 29.3333
W =
   30
Percentage Correlation Adaptive Parity Space: -3.8889

A =
   0.2500
W =
  300
Percentage Correlation Adaptive Parity Space: 58.6111
W =
  200
Percentage Correlation Adaptive Parity Space: 53.9167
W =
  100
Percentage Correlation Adaptive Parity Space: 44.1667
W =
   50
Percentage Correlation Adaptive Parity Space: 23.6667
W =
   30
Percentage Correlation Adaptive Parity Space: -3.8889
```

APPENDIX J: CORRELATION RESULTS FOR VARIED NOISE LEVELS

This appendix shows the percentages of correlation obtained for the parity space, adaptive parity space and observer methods of fault detection with the noise parameter, A, varied between 0.01V and 1.00V (maximum amplitude). The static simulation and fault parameters are described in Section 5.1. The results are:

A =
0.0100
Percentage Correlation Adaptive Parity Space: 72.0963
Percentage Correlation Parity Space: 0.56657
Percentage Correlation Observer: 25.0708

A =
0.0250
Percentage Correlation Adaptive Parity Space: 72.0963
Percentage Correlation Parity Space: 0.56657
Percentage Correlation Observer: 27.6204

A =
0.0500
Percentage Correlation Adaptive Parity Space: 70.255
Percentage Correlation Parity Space: 2.4079
Percentage Correlation Observer: 30.8782

A =
0.0750
Percentage Correlation Adaptive Parity Space: 52.5496
Percentage Correlation Parity Space: -19.5467
Percentage Correlation Observer: -6.9405

A =
0.1000
Percentage Correlation Adaptive Parity Space: -2.1246
Percentage Correlation Parity Space: -68.9802
Percentage Correlation Observer: -59.6317

A =
0.2500
Percentage Correlation Adaptive Parity Space: -126.0623
Percentage Correlation Parity Space: -132.153
Percentage Correlation Observer: -151.983

A =
0.5000
Percentage Correlation Adaptive Parity Space: -146.3173
Percentage Correlation Parity Space: -147.0255
Percentage Correlation Observer: -152.2663

A =

0.7500

Percentage Correlation Adaptive Parity Space: -150.1416
Percentage Correlation Parity Space: -150.5666
Percentage Correlation Observer: -152.9745

A =

1

Percentage Correlation Adaptive Parity Space: -152.5496
Percentage Correlation Parity Space: -151.1331
Percentage Correlation Observer: -153.2578

APPENDIX K: APS WITH RLS RESIDUAL GRAPHS

This appendix shows the residual graphs obtained for the Adaptive Parity Space (APS) method of fault detection, using a Recursive Least Squares (RLS) estimate of the residuals. The first three figures show the residuals obtained for a RLS estimation for a simulation duration of 1000[s]. Figure K-4 shows Residuals S and D using a RLS estimation with forgetting factor, λ, for a simulation duration of 5000[s].

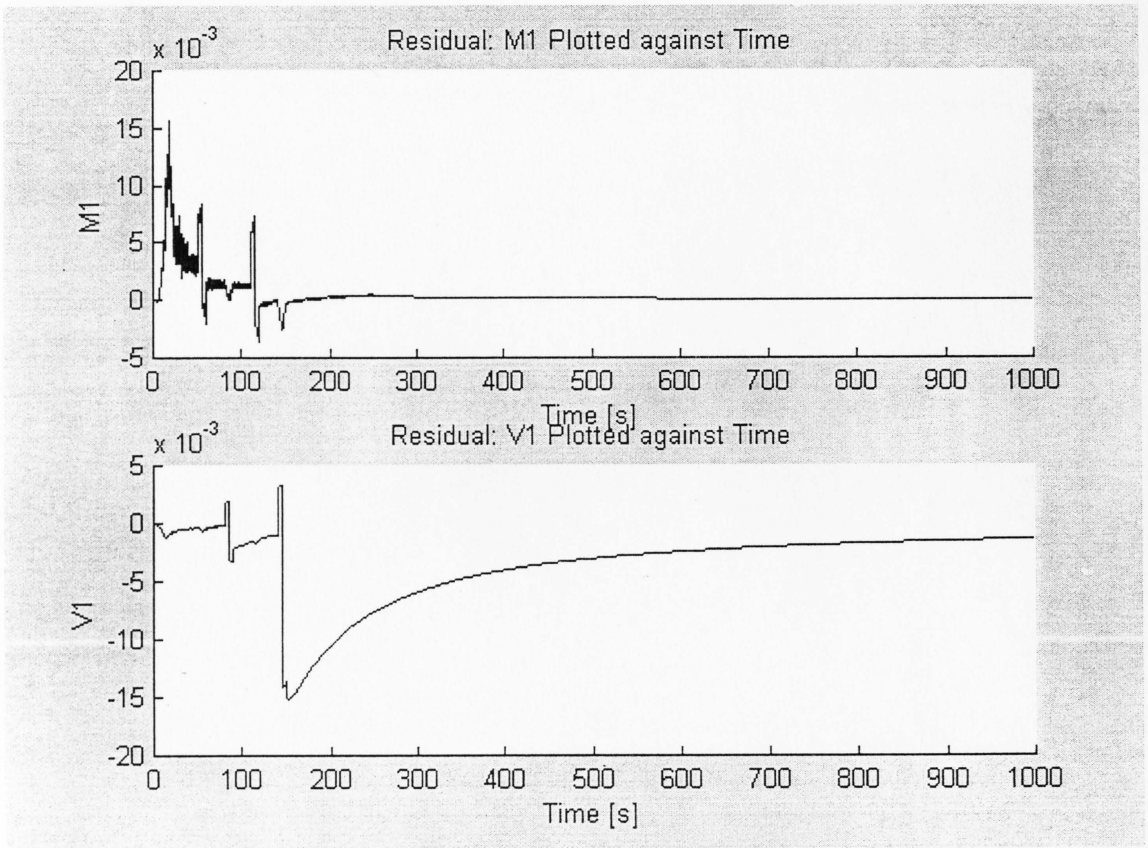


Figure K-1 APS Residuals M1 and V1 using RLS

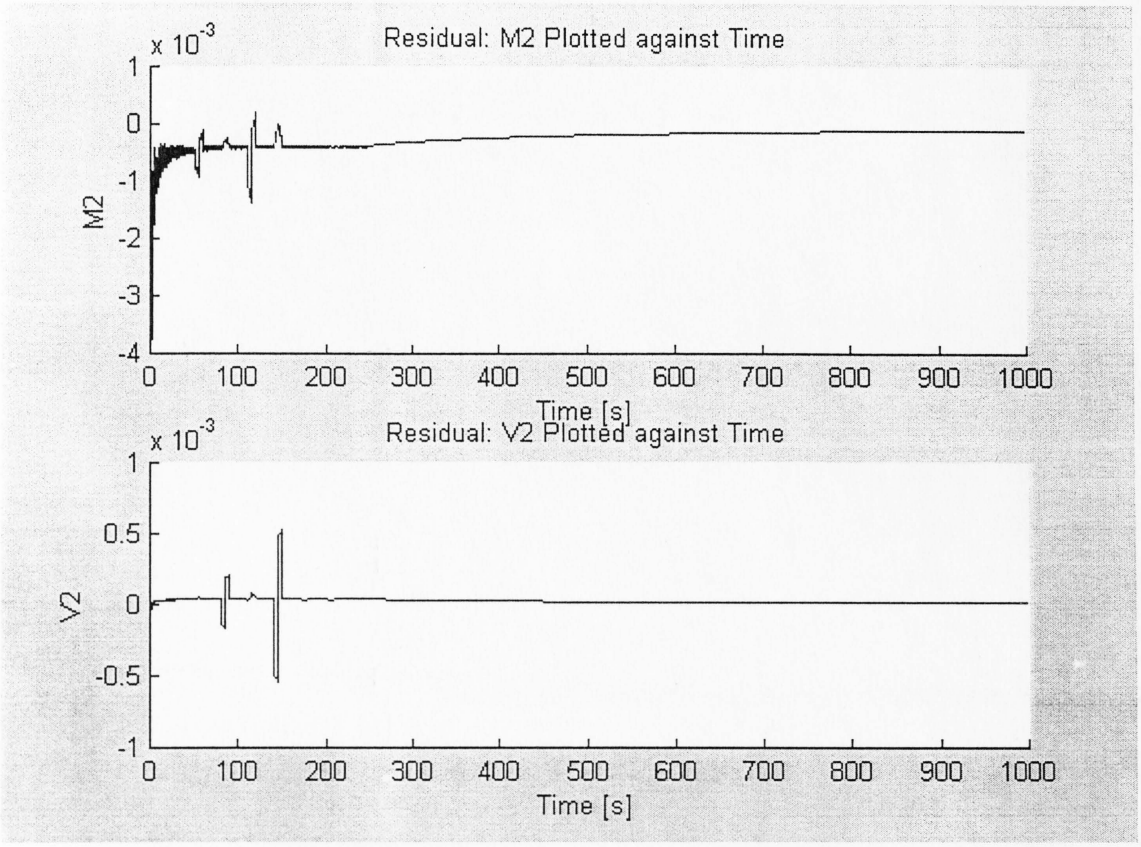


Figure K-2 APS Residuals M2 and V2 using RLS

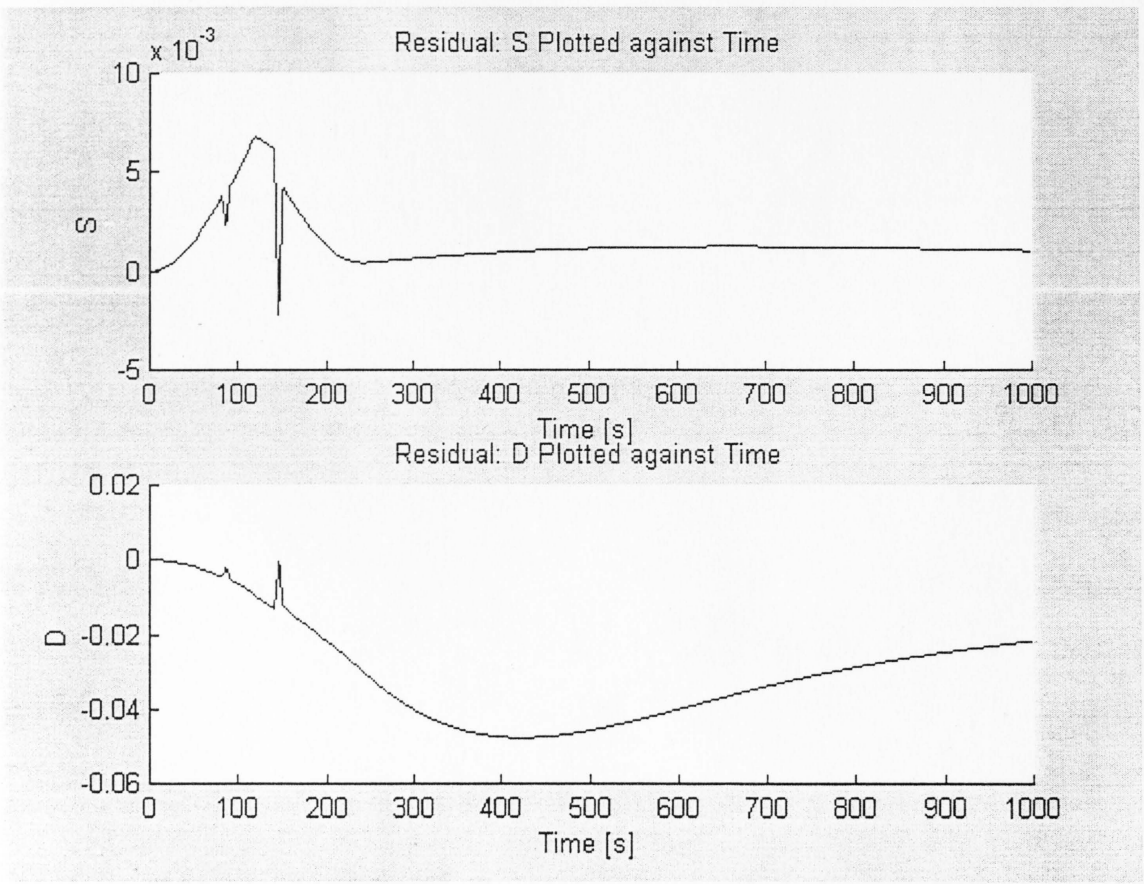


Figure K-3 APS Residuals S and D using RLS

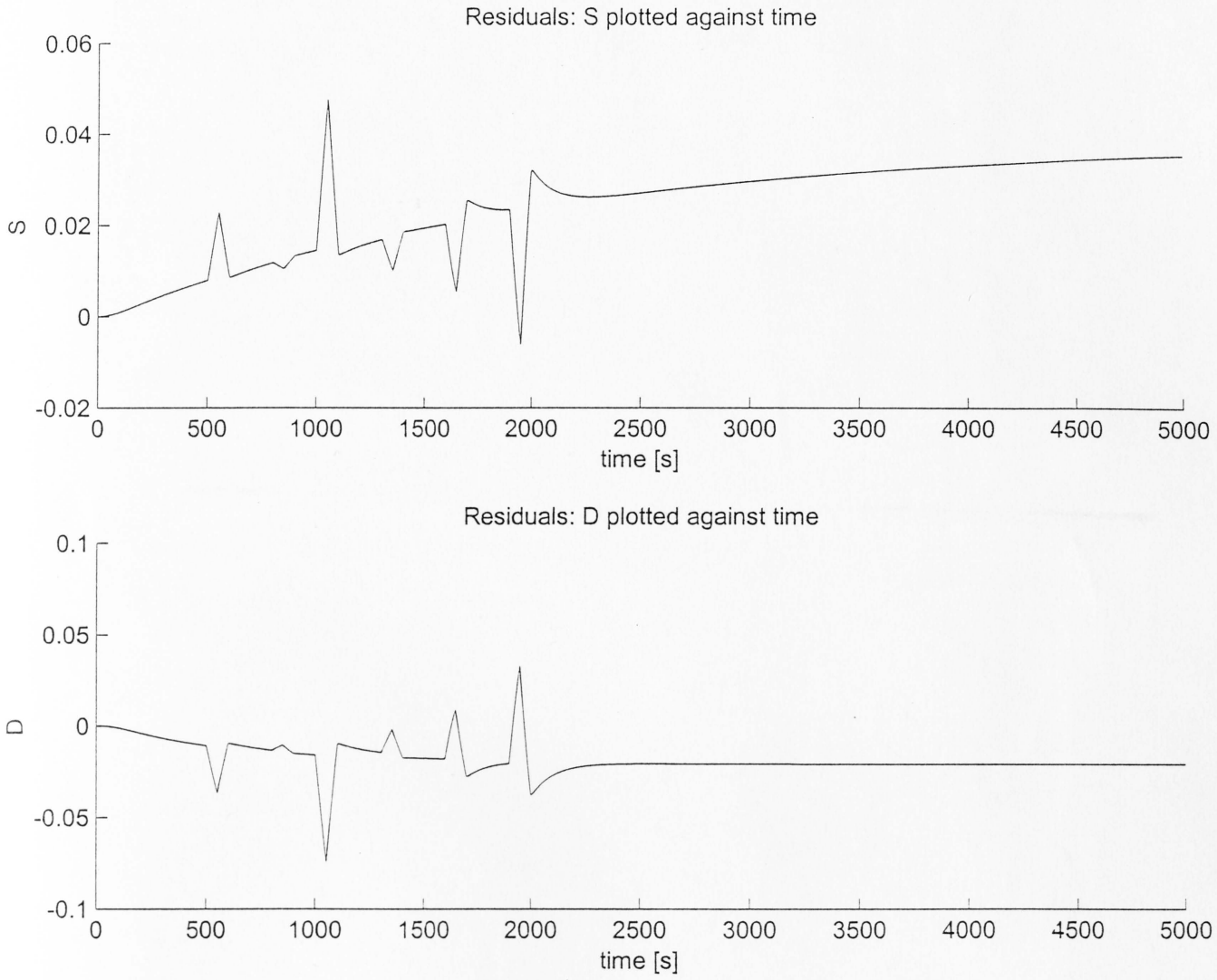


Figure K-4 APS Residuals S and D using RLS with λ

Comment:

The graphs show that the residuals do not become unstable, but have a large overshoot, settling after a lengthy duration, which is dependent on the residual concerned.