①

MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED TO THE VISTA F-16 WITH ACTUATOR AND SENSOR FAILURES

THESIS

Timothy E. Menke

AFIT/GA/ENG/92J-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

①

AFIT/GA/ENG/92J-01

MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED TO THE VISTA F-16 WITH ACTUATOR AND SENSOR FAILURES

THESIS

Timothy E. Menke

AFIT/GA/ENG/92J-01

Approved for public release; distribution unlimited

AFIT/GA/ENG/92J-01

MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED TO THE VISTA F-16 WITH ACTUATOR AND SENSOR FAILURES

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fullfillment of the
Requirements for the Degree of
Master of Science in Aerospace Engineering

Timothy E. Menke, B.S.A.E.

June 1992

Acknowledgements

Throughout my studies I have had the pleasure of working with many outstanding instructors, but these experiences pale in comparison to the lessons learned by working with my thesis advisor, Dr. Peter S. Maybeck. Dr. Maybeck provides a challenging, dynamic, and inspiring learning environment. I firmly believe I became an engineer in this thesis effort, under his guidance. I will never forget his optimism, kindness, and patience. It has been an honor to work with him. Also, I wish to thank the other two members of my thesis committee, Major Riggins and Captain Ridgely. Their comments and questions significantly improved the quality of the final report. I want to thank my sponsor, Captain Stuart Sheldon and WL/FIGL. Also, I'd like to thank Capt Danny "the computer wizard" Shoop for his help in those rough coding spots early on in the development (I didn't forget you). Finally, I want to thank my "home" crew for holding down the fort while I was "goofing off" (working). They are responsible for maintaining my fighting spirit. Thank you *Heidi and Gunnar.*

Timothy E Menke

TABLE OF CONTENTS

Acknowledgements ii
List of Figures v
List of Tables xii
Abstract xiii

CHAPTER 1 Introduction

1.1 General 1
1.2 Problem Statement 1
1.3 Overview of Thesis Objectives 2
1.4 A Brief Overview of the MMAC Algorithm 2
1.5 Assignment of Probabilities 4
1.6 MMAE-Based Control 6
1.7 Research Objectives 7

1.8 Research Questions

  1.8.1 Probability Convergence 8
  1.8.2 Residual Monitoring 9
  1.8.3 Hierarchical Modeling 10
  1.8.4 Time Sequencing Between Failures 10
  1.8.5 Cross-Axis Coupling 12

1.9 Scope 12
1.10 Limitations 13
1.11 Report Format 14
1.12 Summary 14

CHAPTER 2 Multiple Model Adaptive Estimation

2.1 Algorithm Development 15
2.2 Modified Bayesian Form 19
2.3 Beta Dominance 20
2.4 Scalar Residual Monitoring 21
2.5 Hierarchical Modeling 22
2.6 Summary 24

CHAPTER 3 Model Methodology and Development

3.1 Aircraft Model 25

  3.1.1 Evaluation of Measurement Noise Covariance Matrix *R* 28
  3.1.2 Actuator Dynamics Model 29
  3.1.3 Dryden Wind Model 31
  3.1.4 Discrete Gust Model 34

3.2 Truth Model 35
3.3 The Design Model 41
3.4 The Flight Control System 46
3.5 Actuator Failures 48

3.6 Sensor Failures 48
3.7 Multiple Failures 49
3.8 Dither Signal Design 49
3.9 Step-by-Step Outline 51
3.10 Summary 54

CHAPTER 4 Results

4.1 General 55
4.2 Single Hard Failures 56

  4.2.1 Subliminal Pulsed Dither Signals 57
  4.2.2 Non-Subliminal Dither Signals 62
  4.2.3 Purposeful Commands 88
  4.2.4 Subliminal Sinusoidal Dither Signals 94
  4.2.5 Residual Monitoring 104

4.3 Single Soft Failures

  4.3.1 Subliminal Dither Signals (actuators/increased sensor noise) 113
  4.3.2 Sinusoidal Dither (sensor bias) 141

4.4 Multiple Failures 145

  4.4.1 Subliminal Dither Signals 148
  4.4.2 Sinusoidal Dither Signals 162
  4.4.3 Residual Monitoring 167
  4.4.4 Increased Dither Signal Strength 174
  4.4.5 Simultaneous Dual Failures 180

4.5 Summary 195

CHAPTER 5 Conclusions and Recommendations

5.1 General 196
5.2 Response to Chapter 1 196
5.3 Conclusions and Trends 201

6.0 Bibliography 205

Appendices

Appendix A Results for multiple failures A-1

Appendix B VISTA F-16 Simulation Verification Results B-1

Appendix C MMAESIM Computer Code with Block Diagram C-1

Appendix D Matrix$_X$ Macros D-1

Appendix E Data Files E-1

Appendix F Naecon Paper (Menke/Maybeck) F-1

List of Figures

Figure 1.1 Multiple model adaptive controller 3

Figure 1.2 Maximum aposteriori technique 5

Figure 1.3 MMAE-Based Control 7

Figure 1.4 "Moving Bank" hierarchial structure 11

Figure 2.1 Multiple model adaptive estimation algorithm 17

Figure 2.2 Filter lower probability bounds 20

Figure 2.3 Hierarchical "Moving Bank" Model 23

Figure 3.1 F-16 functional block diagram 45

Figure 3.2 Dither signal wave forms 50

Figure 4.1 Probabilities for no failure scenario using subliminal dither signal 1 63

Figure 4.2 States for no failure scenario using subliminal dither signal 1 64

Figure 4.3 Probabilities for a left stabilator failure using subliminal dither signal 1 65

Figure 4.4 States for a left stabilator failure using subliminal dither signal 1 66

Figure 4.5 Probabilities for a right stabilator failure using subliminal dither signal 1 67

Figure 4.6 States for a right stabilator failure using subliminal dither signal 1 68

Figure 4.7 Probabilities for a left flaperon failure using a subliminal dither signal 1 69

Figure 4.8 States for a left flaperon failure using a subliminal dither signal 1 70

Figure 4.9 Probabilities for a right flaperon failure using a subliminal dither signal 1 71

Figure 4.10 Probabilities for a rudder failure using a subliminal dither signal 1 72

Figure 4.11 States for a rudder failure using a subliminal dither signal 1 73

Figure 4.12 Probabilities for a velocity sensor failure using a subliminal dither signal 1 74

Figure 4.13 Probabilities for an angle of attack sensor failure using a subliminal dither signal 75

Figure 4.14 Probabilities for a pitch rate sensor failure using a subliminal dither signal 1 76

Figure 4.15 Probabilities for a normal acceleration sensor failure using a subliminal dither signal 1 77

Figure 4.16 Probabilities for a roll rate sensor failure using a subliminal dither signal 1 78

Figure 4.17 Probabilities for a yaw rate sensor failure using a subliminal dither signal 1 79

Figure 4.18 Probabilities for a lateral acceleration sensor failure using a subliminal dither signal 1 80

Figure 4.19 Probabilities for no failure scenario using subliminal dither signal 2 81

Figure 4.20 States for no failure scenario using subliminal dither signal 2 82

Figure 4.21 Probabilities for a left stabilator failure using subliminal dither signal 2 83

Figure 4.22 Probabilities for a right stabilator failure using subliminal dither signal 2 84

Figure 4.23 Probabilities for a left flaperon failure using a subliminal dither signal 2 85

Figure 4.24 Probabilities for a right flaperon failure using a subliminal dither signal 2 86

Figure 4.25 Probabilities for a rudder failure using a subliminal dither signal 2 87

Figure 4.26 Probabilities for a left stabilator failure using a purposeful roll command 89

Figure 4.27 Probabilities for a right stabilator failure using a purposeful roll command 90

Figure 4.28 Probabilities for a left flaperon failure using a purposeful roll command 91

Figure 4.29 Probabilities for a right flaperon failure using a purposeful roll command 92

Figure 4.30 Probabilities for a rudder failure using a purposeful roll command 93

Figure 4.31 Probabilities for no failure scenario using a subliminal sinusoidal dither 97

Figure 4.32 States for no failure scenario using a subliminal sinusoidal dither 98

Figure 4.33 Probabilities for a left stabilator failure using a subliminal sinusoidal dither 99

Figure 4.34 Probabilities for a right stabilator failure using a subliminal sinusoidal dither 100

Figure 4.35 Probabilities for a left flaperon failure using a subliminal sinusoidal dither 101

Figure 4.36 Probabilities for a right flaperon failure using a subliminal sinusoidal dither 102

Figure 4.37 Probabilities for a rudder failure using a subliminal sinusoidal dither 103

Fig 4.38(a) Single scalar velocity residual for the fully functional filter given a left stabilator failure 106

Fig 4.38(b) Single scalar velocity residual for the left stabilator filter given a left stabilator failure 106

Fig 4.39(a) Single scalar angle of attack residual for the fully functional filter given a left stabilator failure 107

Fig 4.39(b) Single scalar angle of attack residual for the left stabilator filter given a left stabilator failure 107

Fig 4.40(a) Single scalar pitch rate residual for the fully functional filter given a left stabilator failure 108

Fig 4.40(b) Single scalar pitch rate residual for the left stabilator filter given a left stabilator failure 108

Fig 4.41(a) Single scalar normal acceleration residual for the fully functional filter given a left stabilator failure 109

Fig 4.41(b) Single scalar normal acceleration residual for the left stabilator filter given a left stabilator failure 109

Fig 4.42(a) Single scalar roll rate residual for the fully functional filter given a left stabilator failure 110

Fig 4.42(b) Single scalar roll rate residual for the left stabilator filter given a left stabilator failure 110

Fig 4.43(a) Single scalar yaw rate residual for the fully functional filter given a left stabilator failure 111

Fig 4.43(b) Single scalar yaw rate residual for the left stabilator filter given a left stabilator failure 111

Fig 4.44(a) Single scalar lateral acceleration residual for the fully functional filter given a left stabilator failure 112

Fig 4.44(b) Single scalar lateral acceleration residual for the left stabilator filter given a left stabilator failure 112

Figure 4.45 Probabilities for a 50% reduced effectiveness left stabilator failure using a subliminal dither pulse 117

Figure 4.46 Probabilities for a 75% reduced effectiveness left stabilator failure using a subliminal dither pulse 118

Figure 4.47 Probabilities for a 50% reduced effectiveness right stabilator failure using a subliminal dither pulse 119

Figure 4.48 Probabilities for a 75% reduced effectiveness right stabilator failure using a subliminal dither pulse 120

Figure 4.49 Probabilities for a 50% reduced effectiveness left flaperon failure using a subliminal dither pulse 121

Figure 4.50 Probabilities for a 75% reduced effectiveness left flaperon failure using a subliminal dither pulse 122

Figure 4.51 Probabilities for a 50% reduced effectiveness right flaperon failure using a subliminal dither pulse 123

Figure 4.52 Probabilities for a 75% reduced effectiveness right flaperon failure using a subliminal dither pulse 124

Figure 4.53 Probabilities for a 50% reduced effectiveness rudder failure using a subliminal dither pulse 125

Figure 4.54 Probabilities for a 75% reduced effectiveness rudder failure using a subliminal dither pulse 126

Figure 4.55 Probabilities for a 1σ increased sensor noise failure for the velocity sensor 127

Figure 4.56 Probabilities for a 3.16σ increased sensor noise failure for the velocity sensor 128

Figure 4.57 Probabilities for a 1σ increased sensor noise failure for the angle of attack sensor 129

Figure 4.58 Probabilities for a 3.16σ increased sensor noise failure for the angle of attack sensor 130

Figure 4.59 Probabilities for a 1σ increased sensor noise failure for the pitch rate sensor 131

Figure 4.60 Probabilities for a 3.16σ increased sensor noise failure for the pitch rate sensor 132

Figure 4.61 Probabilities for a 1σ increased sensor noise failure for the normal acceleration sensor 133

Figure 4.62 Probabilities for a 3.16σ increased sensor noise failure for the normal acceleration sensor 134

Figure 4.63 Probabilities for a 1σ increased sensor noise failure for the roll rate sensor 135

Figure 4.64 Probabilities for a 3.16σ increased sensor noise failure for the roll rate sensor 136

Figure 4.65 Probabilities for a 1σ increased sensor noise failure for the yaw rate sensor 137

Figure 4.66 Probabilities for a 3.16σ increased sensor noise failure for the yaw rate sensor 138

Figure 4.67 Probabilities for a 1σ increased sensor noise failure for the lateral acceleration sensor 139

Figure 4.68 Probabilities for a 3.16σ increased sensor noise failure for the lateral acceleration sensor 140

Figure 4.69 Probabilities for a roll rate bias of 0.1 deg/sec using a subliminal dither pulse 143

Figure 4.70 Probabilities for a yaw rate bias of 0.01 deg/sec using a subliminal dither pulse 144

Figure 4.71 Probabilities for a left stabilator failure followed by a right stabilator failure using a subliminal dither pulse 152

Figure 4.72 Probabilities for a right flaperon failure followed by a rudder failure using a subliminal dither pulse 153

Figure 4.73 Probabilities for a rudder failure followed by an angle of attack sensor failure using a subliminal dither pulse 154

Figure 4.74 Probabilities for a velocity sensor failure followed by a left stabilator failure using a subliminal dither pulse 155

Figure 4.75 Probabilities for a pitch rate sensor failure followed by a left stabilator failure using a subliminal dither pulse 156

Figure 4.76 Probabilities for a pitch rate sensor failure followed by a rudder failure using a subliminal dither pulse 157

Figure 4.77 Probabilities for a pitch rate sensor failure followed by a roll rate sensor failure using a subliminal dither pulse 158

Figure 4.78 Probabilities for a normal acceleration sensor failure followed by pitch rate sensor failure using a subliminal dither pulse 159

Figure 4.79 Probabilities for a yaw rate sensor failure followed by a lateral acceleration sensor failure using a subliminal dither pulse 160

Figure 4.80 Probabilities for a lateral acceleration sensor failure followed by a yaw rate sensor failure using a subliminal dither pulse 161

Figure 4.81 Probabilities for a left stabilator failure followed by a right stabilator failure using a sinusoidal dither signal 164

Figure 4.82 Probabilities for a right flaperon failure followed by a rudder failure using a sinusoidal dither signal 165

Figure 4.83 Probabilities for a velocity sensor failure followed by a left stabilator failure using a sinusoidal dither signal 166

Figure 4.84 Velocity residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure 169

Figure 4.85 Pitch rate residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure 169

Figure 4.86 Normal acceleration residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure 170

Figure 4.87 Velocity residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure 170

Figure 4.88 Pitch rate residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure 171

Figure 4.89 Normal acceleration residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure 171

Figure 4.90 Velocity residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure 172

Figure 4.91 Pitch rate residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure 172

Figure 4.92 Normal acceleration residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure 173

Figure 4.93 Probabilities for a right flaperon failure followed by a pitch rate sensor failure using an increased strength dither signal 173

Figure 4.94 States for a right flaperon failure followed by a pitch rate sensor failure using an increased strength dither signal 176

Figure 4.95 Probabilities for a right flaperon failure followed by a pitch rate sensor failure using a further increased strength dither signal 177

Figure 4.96 States for a right flaperon failure followed by a pitch rate sensor failure using a further increased strength dither signal 178

Figure 4.97 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 3.0 seconds 183

Figure 4.98 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 0.5 seconds 184

Figure 4.99 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 0.1 seconds 185

Fig 4.100 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 3.0 seconds 186

Fig 4.101 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 0.5 seconds 187

Fig 4.102 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 0.1 seconds 188

Fig 4.103 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 3.0 seconds 189

Fig 4.104 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 0.5 seconds 190

Fig 4.105 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 0.1 seconds 191

Fig 4.106 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 3.0 seconds 192

Fig 4.107 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 0.5 seconds 193

Fig 4.108 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 0.1 seconds 194

List of Tables

Table 3.1 Aircraft State and Control Variables 25

Table 3.2 Sensor Noise RMS Values 28

Table 3.3 Actuator Position and Rate Limits 31

Table 3.4 Zero Order White Noise Strength 43

Table 4.1 Multiple Hard Failure Summary Matrix 146

Table 4.2 Revised Multiple Hard Failure Summary Matrix 179

Abstract

A Multiple Model Adaptive Estimation (MMAE) algorithm is applied to the Variable Stability In-flight Simulator Test Aircraft (VISTA) F-16 at a low dynamic pressure flight condition (0.4 Mach at 20000 ft). A complete F-16 flight control system is modeled containing the longitudinal and lateral-directional axes. Single and dual actuator and sensor failures are simulated including: complete actuator failures, partial actuator failures, complete sensor failures, increased sensor noise, sensor biases, dual complete actuator failures, dual complete sensor failures, and combinations of actuator and sensor failures. Failure scenarios are examined in both maneuvering and straight and level flight conditions. The system performance is characterized when excited by purposeful commands and dither signals. Single scalar residual monitoring techniques are evaluated with suggestions for improved performance. A Kalman filter is designed for each hypothesized failure condition. In this thesis, thirteen elemental Kalman filters are designed encompassing: a no failure filter, left stabilator failure filter, a right stabilator failure filter, a left flaperon failure filter, a right flaperon failure filter, a rudder failure filter, a velocity sensor failure filter, an angle of attack sensor failure filter, a pitch rate sensor failure filter, a normal acceleration sensor failure filter, a roll rate sensor failure filter, a yaw rate sensor failure filter, and a lateral acceleration sensor failure filter. The Bayesian Multiple Model Adaptive Estimator (MMAE) algorithm blends the state estimates from each of the filters, representing a hypothesized failure, multiplied by the filters computed probability. The blended state estimates are sent to the VISTA F-16 flight control system. A hierarchical "moving bank" structure is utilized for multiple failure scenarios. Simultaneous dual failures are included within the study. White Gaussian noise is included to simulate the effects of atmospheric disturbances, and white Gaussian noise is added to the measurements to simulate the effects of sensor noise. Each elemental Kalman filter is compared to the truth model with a selected failure. Filters with residuals that have mean square values most in consonance with their internally computed covariance are assigned the higher probabilities.

MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED TO THE VISTA F-16 WITH ACTUATOR AND SENSOR FAILURES

I. INTRODUCTION

1.1 General

From the advent of the mechanical flight control system of the 1903 Wright Flyer to the modern F-16 digital fly-by-wire flight control system, engineers have applied advancing technologies in an effort to improve performance, reliability, maintainability, robustness, and survivability. The evolution of failure and isolation techniques [7] integrated within the flight control system continues with the application of the Multiple Model Adaptive Estimator (MMAE) [2,5,6,11,12] and Multiple Model Adaptive Controller (MMAC) [5,6,8,10].

While the MMAC algorithm was developed in the early seventies [7], application of the algorithm wasn't practical at the time. The MMAC algorithm uses multiple elemental controllers, each of which can represent a specific failure condition. Inherent to the algorithm's performance is the evaluation of each controller simultaneously, yielding faster and better detection of failures (or other hypothesized conditions) than would be possible without parallel computations. Application of the algorithm wasn't feasible until the development of parallel processing digital computers. A significant characteristic of this algorithm is the ability to detect and isolate failures and properly reconfigure the control system to operate without failed sensors and/or actuators.

The complexity and inherently nonlinear nature of the multiple model adaptive controller prevents complete theoretical analysis of its closed loop performance. However, the algorithm lends itself to computer modelling and simulation in order to characterize its performance attributes.

1.2 Problem Statement

Previous research efforts have addressed a number of interesting modifications to the MMAE/MMAC algorithms for a variety of different aircraft, often only in the longitudinal axis with no cross-axis coupling [8,10]. We will investigate the ability of the multiple model adaptive estimator-based controller to monitor the current state of a VISTA F-16 [Appendix B] in both longitudinal and lateral-directional axes, at a low dynamic pressure flight condition (0.4 Mach at 20,000 ft). A low dynamic pressure test case provides an environment in which control surfaces are less effective and dynamic damping is poor, so this scenario will demonstrate the algorithm's characteristics under unfavorable circumstances. We will induce various types of single and dual failures, including partial and complete actuator and sensor failures. Of particular interest will be the ability of the MMAC to detect and isolate the failure in a timely manner, in order to maintain stable control of the vehicle throughout the transition period after the onset of a failure.

1.3 Overview of Thesis Objectives

As a starting point, we will briefly review the MMAC and MMAE-based controller methodologies, followed with an overview of three techniques used to construct the MMAE-based controller algorithm, and conclude with a discussion of the objectives of the thesis effort. The reader is assumed to have a good knowledge of basic Kalman filter design and analysis techniques [5,6,7].

1.4 A Brief Overview of the MMAC Algorithm

The implementation of the multiple model adaptive controller for this application requires the design of a series of individual elemental controllers corresponding to a set of failure modes at a given flight condition. The controllers can be gain-scheduled to form an integrated flight control system capable of functioning anywhere within the specified flight envelope.

Figure 1-1 illustrates the MMAC structure with its bank of k elemental controllers, each of which corresponds to a given failure mode. The failure modes can be developed by analyzing the flight control system within its proposed operating environment. As an example, a fighter is designed to operate in a hostile environment. The failure modes for a fighter might include the partial or complete loss of an actuator or control surface. One would also include the biasing or complete loss of any sensor, etc. In its simplest implementation using a constant-gain controller, each of the elemental controllers consists of a Kalman filter based upon the aircraft plant corresponding to the failed condition, cascaded with a deterministic optimal controller gain, both

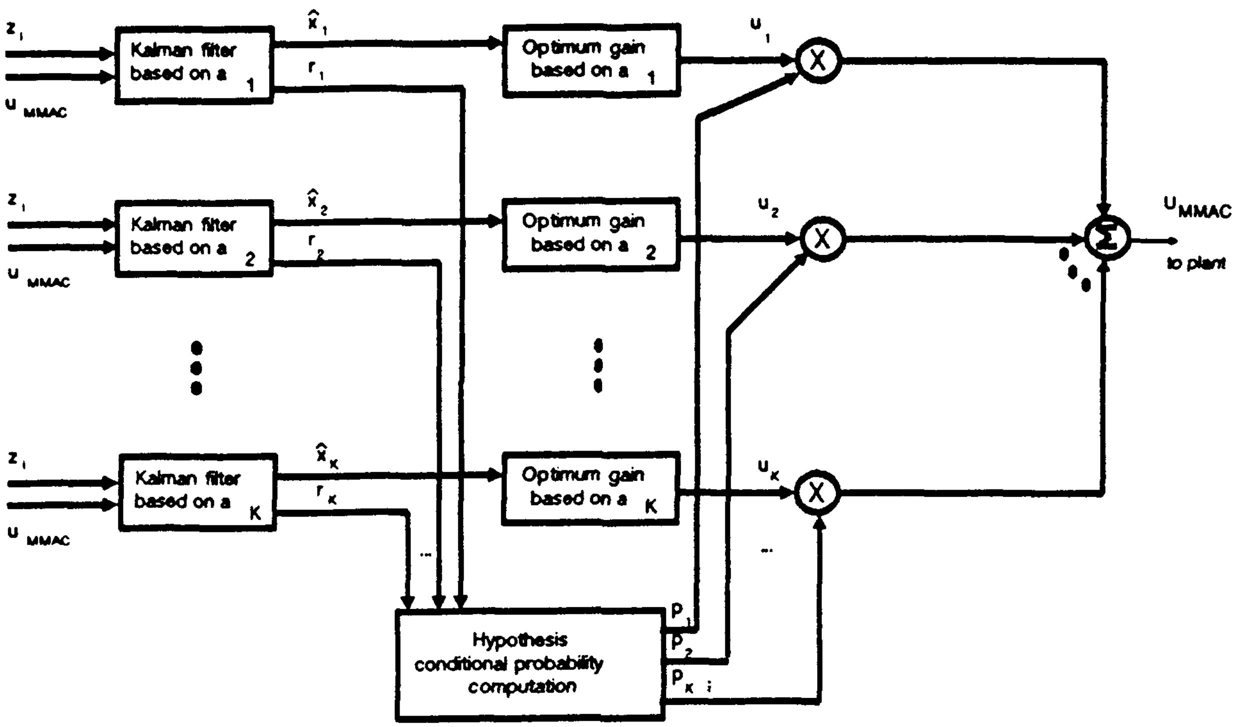

Figure 1.1 Multiple Model Adaptive Controller

based on an assumed parameter value, a_k. If a is a parameter representing a failure space, then a_k is one of the possible discrete realizations of a, in this case one possible failure mode. Each of the elemental Kalman filters utilize the same measurement realization vector

$$z(t_i) = z_i \tag{1-1}$$

and generate state estimates $\hat{x}(t_i^+)$, and a measurement residual vector $r_k(t_i)$, where

$$r_k(t_i) = z_i - H_k(t_i)\hat{x}_k(t_i^-) \tag{1-2}$$

and the subscript k corresponds the parameter value a_k assumed by that particular elemental filter. The residual vector is the difference between the observed measurement vector and the predicted value from the filter. $H_k(t_i)$ is the measurement matrix value from the kth filter. $\hat{x}_k(t_i^-)$ is the Kalman filter estimate of the state vector before the incorporation of the measurement at time t_i.

If the *kth* filter represents the current plant configuration (i.e. the failure condition), the magnitude of the residual from the filter will be small, on the order of the size of the residual's standard deviation as computed within the *kth* filter itself. The magnitudes of the residuals from the other "mismatched" filters, representing other failure scenarios, will be larger than anticipated by the filter-computed standard deviation.

In summary, the elemental controller with the most accurate representation of the current plant configuration will provide the smallest residual to the MMAC algorithm. The algorithm will assign a probability to each of the filters (i.e., assumed failure scenarios) based upon the magnitude of the associated residual.

1.5 Assignment of Probabilities

From *Figure 1.1*, after the construction of the residuals, a number of techniques can be used to assign probabilities to each of the elemental controllers and generate an adaptive control vector, u_{MMAC}. This thesis will examine three techniques. Some previous research efforts have addressed the formulation of the control vector u_{MMAC} by using a maximum a posteriori (MAP) approach, a Bayesian approach, and a modified Bayesian approach [10].

Using a MAP technique, the single u_k corresponding to the highest probability p_k (i.e., with residuals having the smallest magnitude, relative to the filter-computed standard deviation) would be selected as the control vector to apply to the plant (*Figure 1.2*). Past research efforts demonstrated the ability of the algorithm to select a controller in a timely fashion; however, the algorithm cannot adapt well to any failure scenario not explicitly modelled within the MMAC hierarchy. Specifically, it cannot blend together the outputs of two or more elemental controllers that each have large probabilities p_k, such as controllers based on no failure and a single hard failure, to be blended together to address a partial failure of that same sensor or actuator. The algorithm is discussed in detail in Chapter 2.

The Bayesian form of the MMAC algorithm produces a probability-weighted average of the form:

$$u_{MMAC}(t_i) = \sum_{k=1}^{K} u_k[\hat{x}_k(t_i^+),t_i]\, p_k(t_i) \tag{1-3}$$

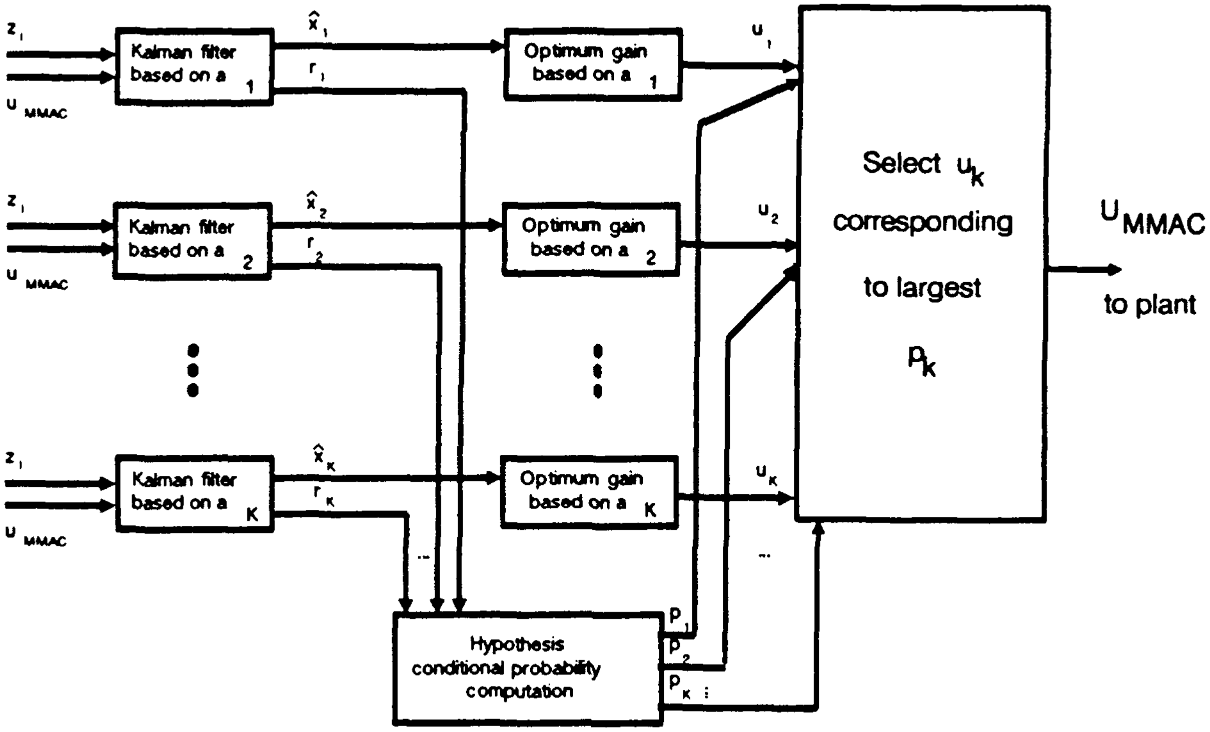

Figure 1.2 Maximum A Posteriori (MAP) technique

where K is the total number of elemental filter-controller combinations, $u_k[\hat{x}_k(t_i^+),t_i]$ is a deterministic optimal full-state feedback control law, and $\hat{x}_k(t_i^+)$ is the state estimate generated by a Kalman filter for the *kth* failure mode modelled within the multiple model adaptive controller. The control applied to the plant is the probability-weighted average of all of the individual elemental controller outputs. The Bayesian form is ideally suited to address failures that may not be explicitly modelled within the MMAC hierarchy. As such, it is capable of blending the output of two or more elemental controllers to adapt to an unmodelled failure condition. The Bayesian form is discussed in detail in Chapter 2.

While the Bayesian form of the MMAC algorithm is a powerful control technique, it requires some modification for practical implementation. The blending of the outputs of each of the elemental controllers allows for the adaptation of the algorithm for failure scenarios not explicitly modelled within the MMAC hierarchy; however, an optimum solution would suggest that only the output of models with hypotheses that "bound" the actual failure be included in the probability-weighted control vector, u_{MMAC}. For instance, control of an aircraft with a 50% loss of stabilator authority might be accomplished by blending the outputs of controllers designed for a fully functional vehicle and for a totally failed stabilator. In fact, by including the output of models unrelated to the failure, one is reducing the robustness of the algorithm, since these contributions to u_{MMAC} may yield destabilizing control to an aircraft with a failure mode other than that for which the elemental controller was designed. The *modified Bayesian* algorithm accounts for this phenomena by establishing a lower probability bound, below which control vectors are excluded from the calculation of the control vector, u_{MMAC}, via the probability weighted sum. An additional, even lower p_k bound prevents the algorithm form ever assigning a p_k value of exactly (or essentially) zero to any elemental controller, since that would preclude the iterative computation of p_k from ever yielding a nonzero value for that probability thereafter. Use of such artificial lower bounding on p_k further motivates the use of *modified Bayesian* vs. Bayesian algorithms, to preclude a nonzero weighting of a completely erroneous elemental controller in the formation of u_{MMAC}. An explanation of this effect can be found in Chapter 2. In this thesis, the MMAE algorithm is implemented as explained in Section 1.6.

1.6 MMAE-Based Control

MMAE-based control was selected over MMAC based control for many reasons. In this research effort, the control system was selected from existing control systems. Previous efforts designed control systems with the objective of reconfiguring the estimator and controller portions of the system upon the isolation of a failure. The purpose of this effort is to demonstrate the algorithm's robustness and emerging maturity by applying the algorithm to a full-scale control system in use with the United States Air Force. In addition, the primary focus of this effort is the detection and isolation of failures, not the reconfigurability of the control laws (we anticipate the possibility of loss of control for some failure combinations). Another important consideration for the selection of MMAE-based control is the demonstration of this technique on an Air Force test vehicle. In the event of good performance from the research effort, opportunities may exist to demonstrate the algorithm on this or another test vehicle. A flight test would be another maturation step for the algorithm and provide valuable information to assess the real world implications of applying the algorithm to a flight vehicle.

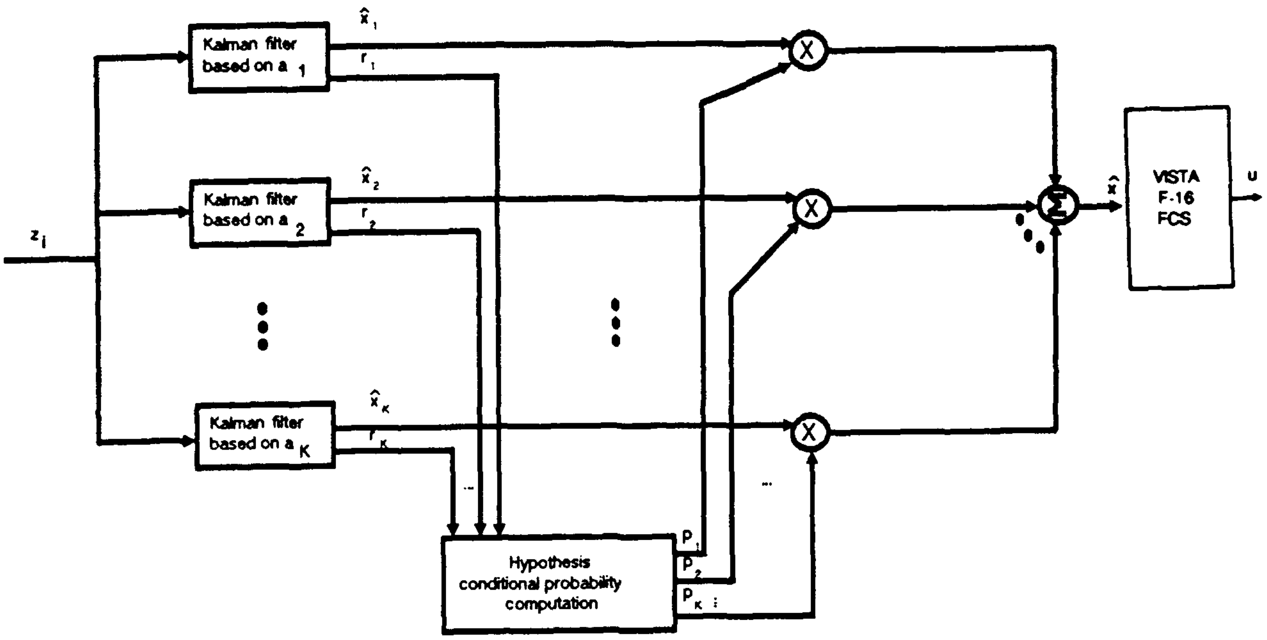

Figure 1.3 MMAE-Based Control

As previously stated, the MMAE algorithm is ideally suited for failure detection and isolation. *Figure 1.3* illustrates the MMAE algorithm structure. The residuals from each of the Kalman filters, representing a different hypothesized failure, determine the appropriate magnitude of the probability assigned to each filter. In this manner, failures are detected and identified according to the probability assigned to the filter.

1.7 Research Objectives

Previous research investigated the use of the MMAC algorithm with elemental controllers designed for a corresponding set of failure conditions [8,10]. This research effort will follow previous work and apply MMAE-based control techniques to the VISTA F-16 aircraft using a MMAE cascaded with a FORTRAN model of the F-16 flight control system.

The primary issue throughout this effort will be failure detection and isolation. If the current VISTA F-16 flight control system is unable to maintain stability for a given failure, once the failure mode is correctly identified, we will assume that a controller could be developed to fulfill that requirement. By evaluating different MMAE-based control algorithms and their ability to detect and isolate failures, we can validate previous efforts while simultaneously advancing the state of the art by applying these techniques to a longitudinal-and-lateral-directional flight control system in service with the United States Air Force.

1.8 Research Questions

1.8.1 Probability Convergence

Some questions of interest include:

Do the elemental probabilities p_k *converge to a solution? Is the solution the correct one?*

For dual failures, is the convergence path-dependent (i.e., is the order of failure occurrence important for convergence properties)?

Are the convergence rates quick enough to prevent large transients or loss of control?

Are the convergence rates dependent on the failure type?

When inducing a failure into the system, we would expect the residuals of the elemental controller corresponding to the true system failure status to be small as compared to the residuals of other "mismatched" filters. For a failure not explicitly modeled in the bank of elemental controllers, we would anticipate a blending of the control vectors from the "bounding" elemental controllers.

While it may be obvious that the proper detection and identification of failures not explicitly modelled in the bank of elemental controllers will characterize the capabilities of the MMAC algorithm, both cases (explicitly modelled and not) are of interest. One attribute we can use to measure performance is how quickly the algorithm converges to a solution. A comparison of convergence time between hard (complete) and soft (partial) actuator or sensor failures provides insight into future enhancements to the calculation and assignment of probability weightings. Expanding the thought process, we can compare convergence properties between failures explicitly modeled in the bank of elemental controllers and those which are not. For dual failures, we can identify any ambiguities which may arise due to the order in which the failures occur.

If the MMAE or MMAC algorithm only had to address failures explicitly modelled in the bank of elemental controllers, the Kalman filters within each elemental controller could be tightly tuned to provide optimum performance at design conditions. From a probabilistic viewpoint, an infinite number of flight control system failure variations are possible. The nature of this problem forces the consideration of failures not explicitly modelled within the bank of elemental controllers. Tuning a Kalman filter in any of the elemental controllers to optimize its performance for a single failure mode may improve performance for that failure condition but may cause significant degradation for those cases not explicitly modelled. However, one cannot afford to add too much dynamic pseudonoise during the tuning process, because it masks inadequacies of assumed models, and thus incapacitates adaptation. The best combination is a tightly tuned filter combined with an enhanced-robustness full-state feedback controller. One would obtain good adaptation to failure status and a robust controller if some degree of misidentification occurs.

The convergence rate of the probability weightings not only indicates the MMAE/MMAC algorithm's performance, but is also indicative to how well the elemental controllers model the true system state for a given failure condition. For an explicitly modelled failure, a rapid convergence rate directly corresponds to a good match between the true system failure status and the elemental controller. For the "matching" filter/controller, the residuals from that filter will be small compared to the residuals of other "mismatched" filters.

1.8.2 Residual Monitoring

By monitoring the residuals generated by each of the Kalman filters within the bank of elemental controllers, one can identify which filter model represents the best indication of the current system status. The "best" model will have a residual mean squared value consistent with its own internally computed covariance.

Questions of interest include:

Are there additional techniques which may enhance the algorithm's performance?

Can additional voting (beyond the u_{MMAC} probability calculations) improve the performance of the algorithm? Is it useful to monitor the scalar components of the residual vector, rather than the entire residual vector as done by the MMAE/MMAC algorithm itself?

Are there situations in which residual monitoring breaks down?

In what situations does "Beta Dominance" become important (for an explanation of this effect, see Chapter 2), and what are the viable solutions to this problem?

1.8.3 Hierarchical Modeling

The MMAE/MMAC algorithm provides flexibility in its application to single and multiple failures. Given a relatively small finite set of failure conditions, it might be feasible to design an elemental controller for every failure condition within the set. For a flight control system application, this is not practical. Modeling the single and dual failures of K sensors and/or actuators would require $1+K+K!/((K-2)!2!)$ elemental filters, with one no-failure filter, K single-failure filters, and $K!/((K-2)!2!)$ dual-failure filters [10,11].

An alternative method used in previous flight control research involves a "moving bank" of $1+K$ filters [10,11,12]. This technique significantly reduces the computational burden by requiring fewer elemental filters to be "on line" at any given time.

Figure 1.4 identifies the hierarchical structure of the "moving bank" of filters. At level 0, filters a_1 through a_K correspond to the single-failure filters. Filter a_0 is the no-failure filter. The algorithm continuously monitors the system with the $(1+K)$ Level 0 filters. At the isolation of a single failure, the algorithm will activate the appropriate Level 1 bank of $(1+K)$ elemental filters designed for the isolated failure alone and any other possible second failure. The addition of a no-failure filter allows the algorithm to "back out" of the decision tree structure and return to Level 0 in the event of a first misidentified failure, intermittent failure, etc.

Questions of interest include:

Will hierarchical modeling provide good multiple failure performance for the current application?

In the event of simultaneous dual failures, will the algorithm's performance be path-dependent?

1.8.4 Time Sequencing Between Failures

Preliminary investigations of dual failure scenarios will concentrate on the ability of the algorithm to identify two failures separated in time. After the system performance has been characterized, the time between failures

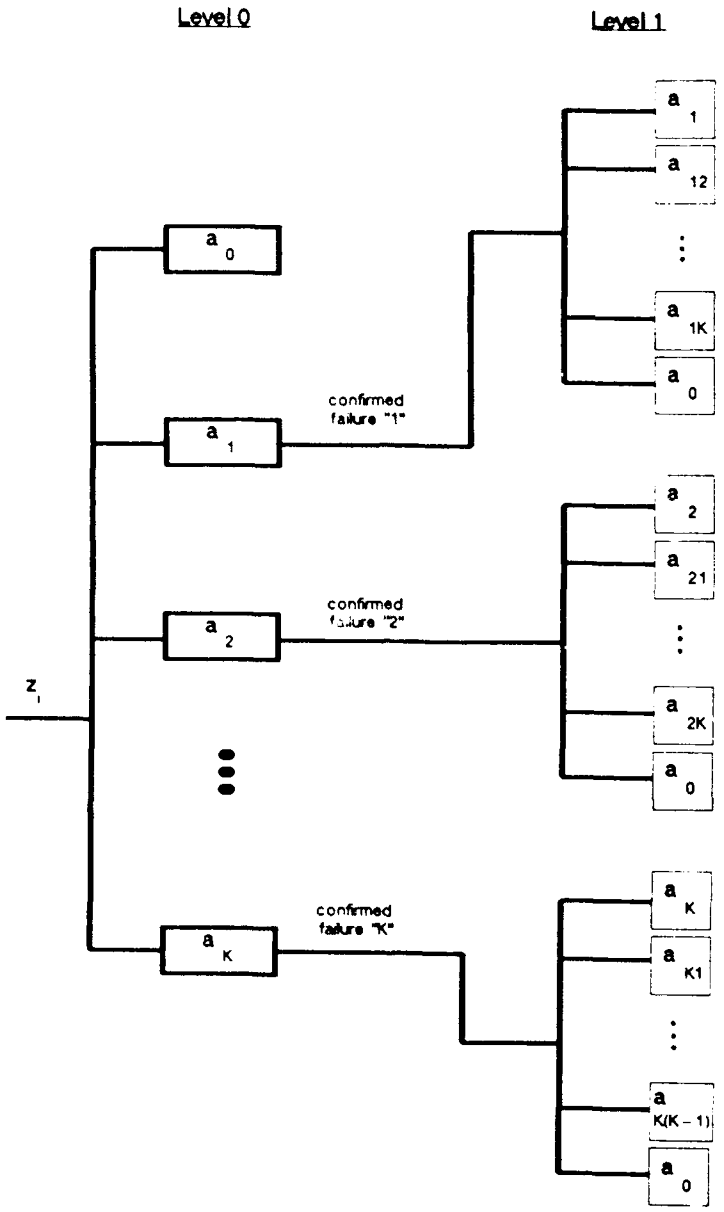

Figure 1.4 Hierarchical Modeling

will be reduced. Eventually, simultaneous failures will be investigated. Of particular interest will be the ability of the algorithm to identify two simultaneous failures correctly.

1.8.5 Cross-Axis Coupling

The VISTA F-16 uses stabilators for both pitch and roll control. Cross-axis coupling may occur, for instance, if a failure in one axis is propagated through the stabilators and identified by the MMAC algorithm as a failure in another axis.

As an example, assume that a stabilator failure is introduced. The failure will propagate through the flight control system and will be summed with the pitch axis commands at the stabilator mixers. The output of the stabilator contains the unfailed pitch axis signal and a signal from the roll axis containing the subsystem failure effects. The MMAC algorithm may identify the roll axis failure from the residuals of one of the elemental filters within the bank or it may not. The MMAC algorithm may have a difficult time resolving whether the roll axis, the pitch axis, or both contain the failure.

Questions of interest include:

Are there any misidentification of failures due to the cross-axis coupling phenomena?

Can additional voters be used to improve the algorithm's performance?

Are filters specifically designed to address a failure that may affect states in both axes ("cross coupling" filters) necessary?

Can the algorithm properly identify multiple failures in different axes?

1.9 Scope

The research effort constructs a multiple model adaptive estimator-based controller using a computer simulation of a VISTA F-16 flight control system. The flight control system differs from the full VISTA flight control system [Appendix B] by the following aspects:

1) *Dynamics above 40 rad/sec are ignored (secondary dynamic effects)*

2) *Fourth order actuator models are replaced by first order models (1st order actuators simplify the model and provide a good frequency response match in the frequencies of interest)*

3) The flight control system is modeled as a continuous system using Euler integration techniques instead of a 40 Hz digital controller. Eventually, the sampled data nature of the controller will be captured. (Simulation results indicate minimal differences between the digital controller and the fast-sampling Euler technique. The flight control system was coded using an analog representation of a block diagram. The digital version became available after preliminary checkout had begun. Given the limited time available for testing, a decision was made to proceed with the checkout and begin data runs.)

The flight control system does include both the longitudinal and lateral-directional axes, gain scheduling, position and rate limits (in the controller software as well as in the physical hardware implementation), angle of attack limiting, the leading edge flap controller, trim effects, command and position breakouts, surface biases, and the yaw structural filter, as in the current full VISTA F-16 flight control system.

The aerodynamic model is linearized about a flight condition of 0.4 Mach, 20000 ft. The model includes the pitch attitude, velocity, angle of attack, pitch rate, sideslip angle, roll angle, roll rate, and yaw rate. The control vector includes the left and right stabilator positions, the left and right flaperon positions, the rudder, and the leading edge flap. Since the model is linearized about a selected flight condition, the thrust is constant in this perturbation model.

The performance analysis of the MMAE-based control algorithm begins by inducing hard modeled failures, soft modeled failures, and finally unmodelled failures. Following the evaluation of single-failure performance, dual-failure scenarios will be investigated. Initially, the time between the first failure and the induction of the second failure will be large. As confidence in the ability of the system to isolate both failures increases, the time between the failures will be reduced. Eventually, simultaneous dual failures will be studied.

1.10 Limitations

Undoubtedly, the most obvious limitation throughout this study is the use a linearized VISTA F-16 aerodynamic model. However, since our interest lies within the area of failure detection and isolation, as opposed to control, this is not considered a restrictive limitation. If time permits, we may elect to test against the full nonlinear simulation model. Another limitation is the use of reduced order actuators which allow a simpler implementation in the model. Frequency response characteristics for first vs. fourth order actuators are essentially equivalent for the frequencies of interest. We simulate the Dryden wind model [8,9,10,11,13] using the appropriate shaping filters driven by white Gaussian noise of the appropriate strength. Within the elemental filters, we represent sensor dynamics as essentially instantaneous, with the sensed output being corrupted by discrete-time white Gaussian noise of appropriate covariances. One of the strengths of this study is its realistic models, particularly of the VISTA F-16 flight control system (as opposed to a strictly "paper" study). The limitations listed within this study do not seriously bound the validity of the results.

1.11 Report Format

Chapter 2 discusses the MMAE/MMAC development in depth, including a complete description of the modified Bayesian algorithm. Chapter 3 addresses the models used within the study, provides an overview of the VISTA F-16 flight control system, discusses the integration of the models, and outlines the data collection process. Chapter 4 presents the results in a "walk through" fashion. The data is exhibited in chronological order with the appropriate comments, observations, and conclusions included. Chapter 5 summarizes the data and provides a comprehensive list of the significant conclusions and recommendations for future research.

1.12 Summary

The goal of this chapter has been to present the MMAC and MMAE-based control algorithms and provide motivation for their implementation. Based upon the projected cost of future military aircraft, sufficient motivation exists within the Air Force to increase the survivability of our present and future aircraft. The MMAE/MMAC algorithm may fulfill that need. The application of this algorithm to a sophisticated flight control system currently in service with the USAF provides a realistic and valuable test.

This chapter discusses the construction of the MMAE/MMAC algorithm and demonstrates the hierarchical structure utilized in addressing multiple failures. The assignment of elemental controller probabilities using a modified Bayesian form allows the algorithm to blend control outputs in an effort to address unmodelled failures. Additional voting techniques may be necessary to resolve ambiguities in the detection and isolation of failures.

A large number of questions remain to be answered concerning the algorithm's ability to isolate single and multiple failures, distinguish between simultaneous failures, and differentiate between longitudinal and lateral directional failures. Investigating these questions forms the basis of this research.

II. MULTIPLE MODEL ADAPTIVE ESTIMATION

2.1 *Algorithm Development*

Let a denote a vector of uncertain parameters in a given linear stochastic state model for a dynamic system. For this application a resides in the space of actuator and/or sensor failures within a flight control system. The a parameters can affect the model structure or the noises entering it. The continuous range of a is discretized to provide a computationally feasible solution. By defining the hypothesis conditional probability $p_k(t_i)$ as the probability that a assumes a value a_k *(for* $k = 1, 2, \dots K$*)*, conditioned on the observed measurement history to time t_i:

$$p_k(t_i) = Prob(a = a_k \,|\, Z(t_i) = Z_i) \tag{2-1}$$

then we can show that $p_k(t_i)$ can be evaluated recursively for all k via the iteration:

$$p_k(t_i) = \frac{f_{z(t_i)|a,Z(t_{i-1})}(z_i|a_k,Z_{i-1})\cdot p_k(t_{i-1})}{\sum_{j=1}^{K} f_{z(t_i)|a,Z_{(t_{i-1})}}(z_i|a_j,Z_{i-1})\cdot p_j(t_{i-1})} \tag{2-2}$$

The measurement history random vector $Z(t_i)$ is made up of partitions $z(t_1) \dots z(t_i)$ which correspond to the measurement vectors available at the sample times $t_1 \dots t_i$. The realization Z_i of the measurement history vector has partitions $z_1 \dots z_i$. The Bayesian minimum mean square error estimate of the state is the probabilistically weighted average:

$$\hat{x}(t_i^+) = E[x(t_i)|Z(t_i) = Z_i] = \sum_{k=1}^{K} \hat{x}_k(t_i^+)\cdot p_k(t_i) \tag{2-3}$$

where $\hat{x}_k(t_i^+)$ is the state estimate generated by a Kalman filter based on the assumption that the parameter vector equals a_k. More explicitly, let the model corresponding to a_k be described by (an "equivalent discrete-time model" [5,7] for a continuous-time system with sampled data measurements):

$$x_k(t_{i+1}) = \Phi_k(t_{i+1},t_i)x_k(t_i) + B_k(t_i)u(t_i) + G_k(t_i)w_k(t_i) \tag{2-4}$$

$$z(t_i) = H_k(t_i)x_k(t_i) + v_k(t_i) \tag{2-5}$$

where x_k is the state, u is a control unit, w_k is a discrete-time zero-mean white Gaussian dynamic noise of covariance $Q_k(t_i)$ at each t_i, z is the measurement vector, and v_k is discrete-time zero-mean white Gaussian measurement noise of covariance $R_k(t_i)$ at t_i, assumed independent (and thus uncorrelated) of w_k. The initial condition $x(t_o)$ is modeled as Gaussian, with mean x_{ko} and covariance P_{ko} and is assumed independent of w_k and v_k. Based on this set of models, the Kalman filter is specified by the measurement update:

$$A_k(t_i) = H_k(t_i)P_k(t_i^-)H_k^T(t_i) + R_k(t_i) \tag{2-6}$$

$$K_k(t_i) = P_k(t_i^-)H_k^T(t_i)A_k^{-1}(t_i) \tag{2-7}$$

$$\hat{x}_k(t_i^+) = \hat{x}_k(t_i^-) + K_k(t_i)\,[z_i - H_k(t_i)\hat{x}_k(t_i^-)] \tag{2-8}$$

$$P_k(t_i^+) = P_k(t_i^-) - K_k(t_i)H_k(t_i)P_k(t_i^-) \tag{2-9}$$

and the time propagation relations:

$$\hat{x}_k(t_{i+1}^-) = \Phi_k(t_{i+1},t_i)\hat{x}_k(t_i^+) + B_k(t_i)u(t_i)$$

(2-10)

$$P_k(t_{i+1}^-) = \Phi_k(t_{i+1},t_i)P_k(t_i^+)\Phi_k^T(t_{i+1},t_i) + G_k(t_i)Q_k(t_i)G_k^T(t_i)$$

(2-11)

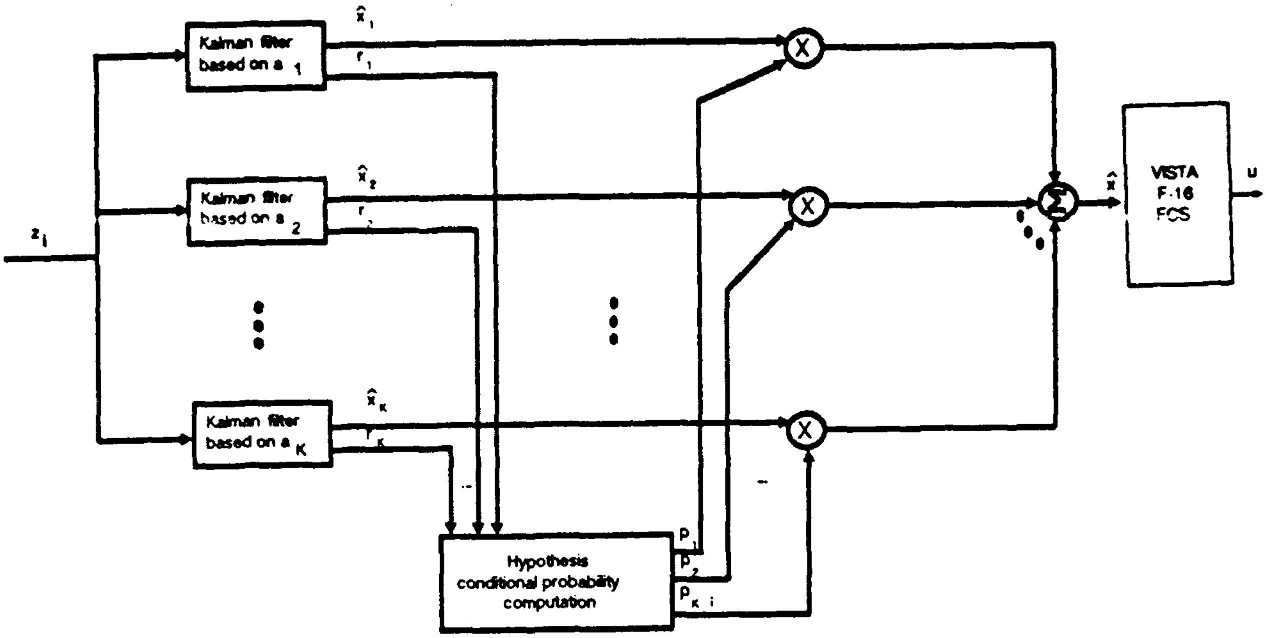

Figure 2.1 Multiple Model Adaptive Estimation Algorithm

As such, the multiple model adaptive filtering algorithm is composed of a bank of K separate Kalman filters, each based on a particular value $a_1 \dots a_K$ of the parameter vector a, as depicted in *Figure 2.1*. When the measurement z_i becomes available at t_i, the residuals $r_1(t_i) \dots r_K(t_i)$ are generated in the K filters as shown by the bracketed term in Eq (2-8), and used to compute $p_1(t_i) \dots p_K(t_i)$ via Eq.(2-2). Each numerator density function in Eq.(2-2) is given by:

$$f_{z(t_i)|a,Z(t_{i-1})}(z_i|a_k,Z_{i-1}) = \frac{1}{(2\pi)^{m/2}\,|A_k(t_i)|^{1/2}}\exp(\cdot) \tag{2-12}$$

$$(\cdot) = [-\frac{1}{2}r_k^T(t_i)A_k^{-1}(t_i)r_k(t_i)] \tag{2-13}$$

where m is the measurement dimension and $A_k(t_i)$ is the residual covariance calculated in the k^{th} Kalman filter as in Eq (2-6). The denominator in Eq (2-2) is simply the sum of all the computed numerator terms and thus is the scale factor required to ensure that the $p_k(t_i)$'s sum to one [5,7,11,17]. Thus the filter probabilities must satisfy two constraints:

$$p_k(t_i) \geq 0 \tag{2-14}$$

$$\sum_{k=1}^{K} p_k(t_i) = 1 \tag{2-15}$$

In operation, one expects the residuals of the Kalman filter with the hypothesis that best matches the true system state to have mean squared value most in consonance with its internally computed covariance, $A_k(t_i)$. The performance of the algorithm depends on the existence of significant differences in the characteristics of "correct" vs "mismatched" filters' residuals. Each filter should be tuned for best performance when the true *values of the uncertain parameters match the assumed value for these parameters.* The addition of substantial amounts of dynamics pseudonoise to the dynamics model, often used to guard against filter divergence, should be avoided as it tends to mask the differences between good and bad models [5,7].

Although the multiple model estimation algorithm appears similar to the multiple model adaptive controller, fundamental differences exist between the two algorithms. The intent of the multiple model estimation algorithm is to generate a composite $\hat{x}_{MMAE}(t_i^+)$ from each of the K elemental filter estimates $\hat{x}_k(t_i^+)$ as shown in ***Figure** 2.1*. In this thesis, we are compensating for parameter uncertainties in B and H representing actuator and sensor failures; a hard actuator failure is modeled with a column of B going to zero, and a sensor hard failure is represented by a row of H being zeroed. The multiple model adaptive estimation algorithm produces a composite state estimate vector used by the flight control system, as shown in ***Figure 1.3***.

2.2 Modified Bayesian Form

The final probability-weighted average of the state estimates, computed as shown in *Figure 2.1*, is produced by a modified Bayesian form of the MMAE algorithm. A Bayesian form of the algorithm allows for a blending of filters designed for hard failures and those designed for no failures to address partial or soft failures. Practical implementation requires a lower bound when computing the probabilities according to Eq (2-2). The addition of a lower bound prevents the algorithm from assigning any single $p_k(t_i)$ a value of zero, which would prevent it from being considered in future probability computations. From the iterative nature of Eq. (2-2), if $p_k(t_{i-1})$ were assigned a value of zero for one of the filters, subsequent probability calculations for that filter would also assign a probability of zero (i.e. $p_k(t_i) = 0$). The addition of a lower bound provides another favorable characteristic. The number of iterations required to increase a very small, but nonzero, p_k significantly increases for smaller magnitudes of p_k values. By providing a lower bound we allow p_k values, previously not important to the combined state estimate, to increase in a timely manner if the system state changes such that the parameter value a_k becomes the best hypothesized value.

A second lower bound excludes the contributions of filters whose hypothesis has nothing to do with the true system failure status from being included in the formulation of the control vector u_{MMAC}. This bound improves performance by ignoring unrelated filters whose probabilities are small. This is important in view of the artificial lower bounds presented in the previous paragraph. Such artificial bounds could cause nonzero contributions from totally erroneous filters to be included in the weighted sum of Eq (2-3).

The combination of these two bounds demonstrates the transition from a theoretical Bayesian form to the application of the modified Bayesian form. *Figure 2.2* graphically displays the bounds developed for use in this research effort. Of particular interest is the selection of the bound magnitudes. The smallest bound should be selected to prevent any value of $p_k(t_i)$ from being assigned an essentially zero value while allowing good dynamic growth of p_k performance should that particular parameter value a_k become the best new parameter value due to a change in the true system failure conditions. The second bound prevents filters whose hypotheses are unrelated to the true failure from being considered in the formulation of the control vector. This bound must not be so large as to interfere with the MMAE algorithm's adaptation performance. Too small a bound would

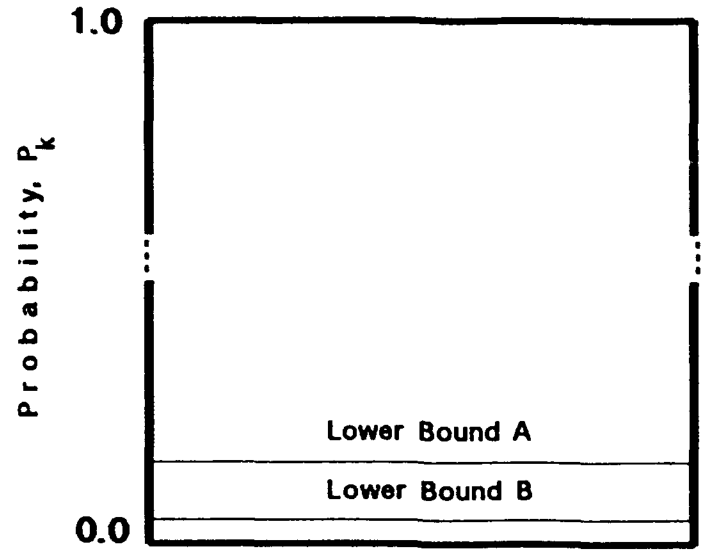

Figure 2.2 Modified Bayesian Form - Lower bounds

cause unrelated filters to contribute to the control vector and reduce system performance. As is often the case in research efforts, such threshold values must be established empirically. Experience shows a value of 0.0001 for the lower probability bound and 0.003 for the upper probability bound produces the desired effect [10,11].

2.3 "Beta Dominance"

As discussed earlier in Chapter 2, the hypothesis probabilities $p_k(t_i)$ are calculated according to Eq. (2-2). Earlier efforts [2,4,6,10] noted that the leading coefficient preceding the exponential term in Eq (2-12) does not provide any useful information in the identification of the failure. The likelihood quotient,

$$L_k = r_k^T(t_i) A_k^{-1} r_k(t_i) \tag{2-16}$$

which appears within the exponential of Eq (2-13), compares the squared residual with the hypothesized filter's internally computed residual covariance. Filters with residuals that have mean square values most in consonance with their internally computed covariance are assigned the higher probabilities by the MMAE algorithm. However, if the likelihood quotients were nearly identical in magnitude for all k, the probability computations of Eqs. (2-2), (2-12), and (2-13) would be based upon the magnitude of the determinants of the $A_k(t_i)$ matrices, resulting in an incorrect assignment of the probabilities that would be independent of the real-world measurements. This effect is known as "Beta Dominance". Because sensor failures, as simulated by zeroing out a row of H, yield smaller $A_k(t_i)$ determinant values and thus larger p_k values, "Beta Dominance" produces a tendency to generate false alarms about sensor failures.

Previous efforts removed the term preceding the exponential in Eq. (2-12) [10,11,12], with significant enhancements of performance and reduction of sensor failure false alarms. Since the denominator of Eq. (2-2) contains the summation of all numerator terms, excluding the terms preceding the exponentials in the calculation of the probabilities does not alter the fact that the computed probabilities sum to one.

2.4 Scalar Residual Monitoring

Incorrect or ambiguous failure identification may be resolved through the use of scalar residual monitoring. Eqs. (2-2), (2-13), and (2-14) demonstrate the relationship of the probability calculations, the probability density function, and the likelihood quotient. These three equations demonstrate the dependency of the probability calculations on the magnitude of the likelihood quotient, Eq. (2-14). The likelihood quotient is merely the sum of scalar terms relating the product of any two scalar components of the residual vector and the filter's internally computed covariance for those two components. If a sensor failure occurs, the single scalar term associated solely with that sensor should have a residual value with a magnitude that is much larger than the associated variance in all of the elemental filters except for the filter designed to "look" for that sensor failure. Scalar residual monitoring can be used as an additional vote when attempting to reduce or eliminate failure identification ambiguities. Scalar residual monitoring is particularly useful when attempting to identify sensor failures. There is a direct relationship between filters designed for sensor failures and the generation of single scalar residuals corresponding to a sensor failure. Failed-actuator elemental filters do not exhibit such direct relationships and often provide a more challenging test for the MMAE algorithm. This thesis will attempt to demonstrate the effectiveness of previous methods to identify actuator failures and provide additional techniques and insight to reduce failure detection ambiguities. In characterizing the performance of the algorithm and the ability of scalar residual monitoring to improve that performance, conditions in which scalar residual monitoring breaks down will be documented.

2.5 Hierarchical Modeling

Previous discussion in Chapter 2 restricted the application of the MMAE algorithm to single failure scenarios. The research effort used a building block approach to addressing system failures. Chapter 4 presents the results for single complete sensor or actuator (hard) failures and then single partial actuator failures or increased sensor noise (soft) failures. The next logical step in this progression is the inclusion of multiple failures. ***Figure*** *2.3* depicts the hierarchical model.

For many practical complex systems, an infinite number of failure scenarios can be hypothesized. As an example, consider an aircraft. In this example, the failure set would include: surface or actuator failures, sensor failures, wiring failures, software failures, etc, and both hard and varying degrees of soft failures. In more precise terms, the failure space would *include an infinite number of possible failure* scenarios. Designing filters for each of these scenarios is impractical and could not physically be implemented. Additionally, performance from such a system would be unacceptable since many filters would be virtually indistinguishable from each other in residual characteristics. By discretizing the failure space, an acceptable set of failure scenarios from which to build filters can be selected. This set spans the failure space since all failures under consideration are bounded by these filters (e.g., a 50% failure in actuator authority is bounded by a fully functional filter hypothesized with no failures and a complete actuator failure). The blended output of these bounding filters can be used to approximate the output of a hypothesized 50% actuator failure not included in the filter "bank" to generate a control vector u_{MMAC}. This "blending" approach allows for a finite number of filters to be used in the design of the MMAE algorithm. Given the assumption of a specific failure having occurred, a new bank of filters is created based upon the first failure and including any other possible second failure within the failure space. The hypothesis of no failure is also included, to allow subsequent measurements to reverse the decision that the first failure had, in fact, occurred. The banks of filters are completely precomputable. This allows for reduced computational loading since at any given time only $1+K$ filters are on line (where K represents the total number of hypothesized failures and the additional filter assumes no failure). This technique is known as a "moving bank" of filters. *Figure 2.3* demonstrates the hierarchical structure of the "moving bank" of filters. The algorithm continuously computes the residuals for each of the $1+K$ filters and assigns probabilities based upon those residuals. Level 0 represents the no failure "bank" of elemental filters. Filter a_0 is the no-failure filter. Filters based on hypothesized values $a_1 ... a_K$ represent each of the single hard failures. When a single

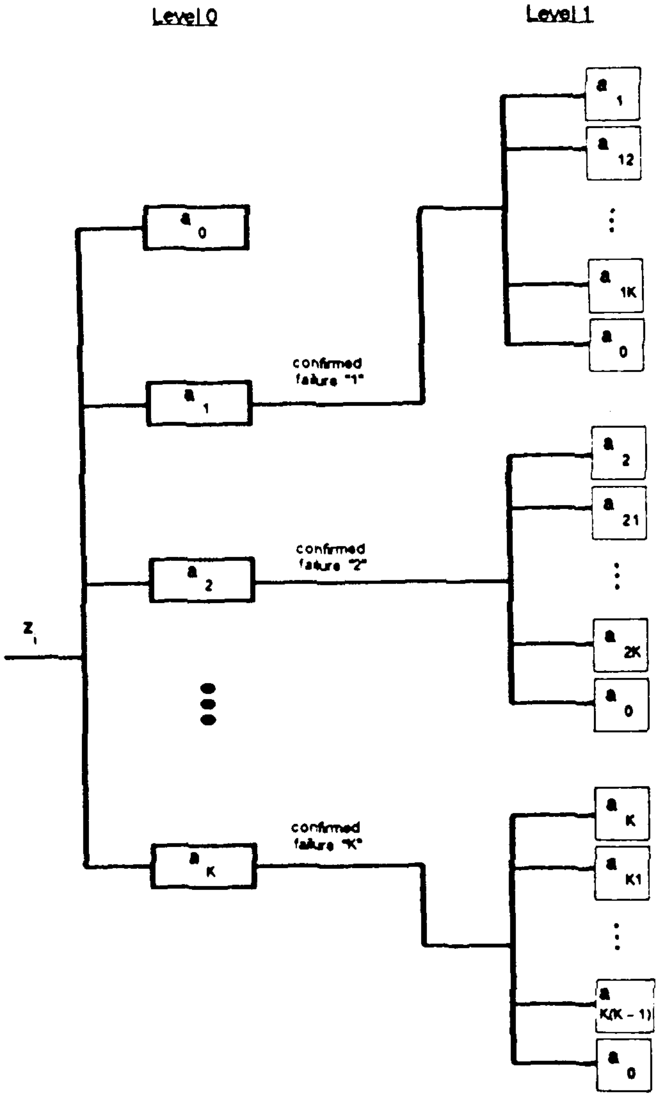

Figure 2.3 Hierarchical "Moving Bank" Model

failure is isolated by the algorithm, the algorithm will activate the appropriate Level 1 bank of $(I+K)$ filters designed for the isolated failure alone and any other possible second failure. The addition of a no-failure filter allows the algorithm to "back out" of the structure and return to Level 0 in the event of a first misidentified filter, intermittent failure, etc.

Dual failures will concentrate not only on convergence criteria but also on the ability of the algorithm to address simultaneous failures. Initially, failures will be separated in time to allow transients to settle to steady state and the algorithm to identify the first failure. After the system performance has been characterized under these conditions, simultaneous failures will be addressed. The ability of the algorithm to address multiple axes failures, simultaneous failures, and combination hard and soft failures will be investigated.

2.6 Summary

This chapter presented the development of the MMAE algorithm. The Modified Bayesian form of the MMAE algorithm is introduced and a detailed description is provided. Two probability lower bounds necessary for practical implementation are described. The "Beta Dominance" effect is associated with the fact that the likelihood quotient contains the useful information when calculating the probabilities; the appropriate tendency to signal sensor failure false alarms due to such "Beta Dominance" is readily compensated by removing the leading coefficient from Eq. (2-12). Scalar residual monitoring provides an additional voting algorithm to resolve failure ambiguities. Section 2.5 introduces the hierarchical modeling structure used for multiple failure identification.

III. MODEL METHODOLOGY AND DEVELOPMENT

3.1 Aircraft Model

The aircraft model is represented as a linear continuous-time dynamics equation given by

$$\dot{x}_{AC}(t) = A_{AC}\, x_{AC}(t) + B_{AC}\, u_{AC}(t) \tag{3-1}$$

with an output equation

$$y(t) = C_{AC}\, x_{AC}(t) + D_{AC}\, u_{AC}(t) \tag{3-2}$$

and a discrete-time measurement equation given by

$$z(t_i) = H_{AC}\, x_{AC}(t_i) + D_{za}\, u_{AC}(t_i) + v(t_i) \tag{3-3}$$

where $y(t)$, $z(t_i)$, $v(t_i)$ are identical in dimension and physical interpretation of components for the truth and filter models. Generally the truth models are of higher state dimensionality than the filter models due to higher order actuator models, discrete gust models, wind models, etc. Additionally, the truth model contains actuator rate and position limiting functions. The noise vector $v(t_i)$ is a zero-mean white Gaussian noise of covariance R. The covariance matrix represents measurement inaccuracies caused by sensor noise. The high frequency sensor dynamics are omitted from the aircraft dynamics model since these are secondary effects.

Table 3.1 Aircraft State and Control Variables

State Variables	Units	Control Variables	Units
Θ pitch angle	rad	δS_L Left Stabilator	rad
u forward velocity	ft/sec	δS_R Right Stabilator	rad
α angle of attack	rad	δF_L Left Flaperon	rad
q pitch rate	rad/sec	δF_R Right Flaperon	rad
ϕ bank angle	rad	δR Rudder	rad
β sideslip angle	rad	δLEF Leading Edge Flap	rad
p roll rate	rad/sec		
r yaw rate	rad/sec		

Table 3.1 lists the eight state and six control variables used in the study. Each of the state variables can be expressed in terms of dimensional derivatives [8,9,10,11]. The aircraft velocity can be expressed as

$$u(t_i) = X'_\theta \theta + X'_u u + X'_\alpha \alpha + X'_q q + X'_{\delta S} \delta S + X'_{\delta F} \delta F + X'_{LEF} \delta LEF \tag{3-4}$$

with $X'_i = X_i / (m I_{xx})$, where m is the aircraft mass in *slugs/ft*^{2} and I_{xx} is the moment of inertia about the aircraft x-axis expressed in units of *slug-ft*^{2}. The stabilator and flaperon inputs are expressed as $0.5X'_{\delta S}\,(\delta S_L + \delta S_R)$ and $0.5X'_{\delta F}\,(\delta F_L + \delta F_R)$ respectively. Eq.(3-5) represents the aircraft equations of motion in a state space format:

$$\begin{bmatrix} \dot{\theta} \\ u \\ \alpha \\ q \\ \dot{\phi} \\ \beta \\ p \\ r \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X'_\theta & X'_u & X'_\alpha & X'_q & 0 & 0 & 0 & 0 \\ Z'_\theta & Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 & 0 \\ M'_\theta & M'_u & M'_\alpha & M'_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & Y'_\phi & Y'_\beta & Y'_p & Y'_r \\ 0 & 0 & 0 & 0 & 0 & L'_\beta & L'_p & L'_r \\ 0 & 0 & 0 & 0 & 0 & N'_\beta & N'_p & N'_r \end{bmatrix} \begin{bmatrix} \theta \\ u \\ \alpha \\ q \\ \phi \\ \beta \\ p \\ r \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5X'_{\delta S} & 0.5X'_{\delta S} & 0.5X'_{\delta F} & 0.5X'_{\delta F} & 0 & X'_{\delta LEF} \\ 0.5Z'_{\delta S} & 0.5Z'_{\delta S} & 0.5Z'_{\delta F} & 0.5Z'_{\delta F} & 0 & Z_{\delta LEF} \\ 0.5M'_{\delta S} & 0.5M'_{\delta S} & 0.5M'_{\delta F} & 0.5M'_{\delta F} & 0 & M'_{\delta LEF} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5Y'_{\delta S} & -0.5Y'_{\delta S} & 0.5Y'_{\delta F} & -0.5Y'_{\delta F} & Y'_{\delta R} & 0 \\ 0.5L'_{\delta S} & -0.5L'_{\delta S} & 0.5L'_{\delta F} & -0.5L'_{\delta F} & L'_{\delta R} & 0 \\ 0.5N'_{\delta S} & -0.5N'_{\delta S} & 0.5N'_{\delta F} & -0.5N'_{\delta F} & N'_{\delta R} & 0 \end{bmatrix} \begin{bmatrix} \delta S_L \\ \delta S_R \\ \delta F_L \\ \delta F_R \\ \delta R \\ \delta LEF \end{bmatrix} \tag{3-5}$$

Positive deflections are defined as stabilator trailing edge down, left flaperon trailing edge down, and rudder trailing edge right (deflections which produce positive moments). The dimensional derivatives are expressed in the body axes.

The dimensional derivatives of *Equation (3-5)* were developed using the GENESIS simulation [13, Appendix

B] residing at the Flight Dynamics Laboratory at Wright-Patterson AFB. GENESIS contains the VISTA F-16 flight control system as well as a complete nonlinear aerodynamic data base. The code allows the user the ability to linearize the model about a trim condition. In this thesis, the trim condition is 0.4 Mach at 20000 ft. As mentioned earlier, the thrust remains constant for this application. Another useful option within GENESIS is the ability to run simulations and produce hard output for a given user command sequence. This option was used to check out the FORTRAN flight control system coded for the MMAE simulation with the linearized model against the full VISTA F-16 flight control system with the nonlinear aerodynamic data base.

The normal and lateral accelerations at the center of gravity are computed by

$$A_{n_{cg}} = -(Z_\alpha \alpha + Z_q q + Z_u u + Z_{\delta S} \delta S + Z_{\delta F} \delta F + Z_{\delta LEF} \delta LEF) \qquad (ft/\sec^2) \qquad (3\text{-}6)$$

$$A_{y_{cg}} = -(Y_\beta \beta + Y_p p + Y_r r + Y_{(\delta S_L - \delta S_R)}(\delta S_L - \delta S_R) + Y_{\delta F} \delta F + Y_{\delta R} \delta R) \qquad (ft/\sec^2) \quad (3\text{-}7)$$

where A_{ncg} and A_{ycg} are the longitudinal and lateral accelerations at the center of gravity and the dimensional derivatives are shown in the unprimed body axes for convenience. The measurement equation is given as

$$z(t_i) = \begin{bmatrix} u \\ \alpha \\ q \\ A_{n_{cg}} \\ p \\ r \\ A_{y_{cg}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & z_{n1} & z_{n2} & z_{n3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & z_{y1} & z_{y2} & z_{y3} \end{bmatrix} \begin{bmatrix} \theta \\ u \\ \alpha \\ q \\ \phi \\ \beta \\ p \\ r \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ z_{n4} & z_{n5} & z_{n6} & z_{n7} & 0 & z_{n8} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ z_{y4} & z_{y5} & z_{y6} & z_{y7} & z_{y8} & 0 \end{bmatrix} \begin{bmatrix} \delta S_L \\ \delta S_R \\ \delta F_L \\ \delta F_R \\ \delta R \\ \delta_{LEF} \end{bmatrix} + v(t_i) \qquad (3\text{-}8)$$

$$\begin{bmatrix} z_{n1} & z_{n2} & z_{n3} \end{bmatrix} = \begin{bmatrix} -Z_u & -Z_\alpha & -Z_q \end{bmatrix} \tag{3-9}$$

$$\begin{bmatrix} z_{y1} & z_{y2} & z_{y3} \end{bmatrix} = \begin{bmatrix} -Y_\beta & -Y_p & -Y_r \end{bmatrix} \tag{3-10}$$

$$\begin{bmatrix} z_{n4} & z_{n5} & z_{n6} & z_{n7} & z_{n8} \end{bmatrix} = \begin{bmatrix} -0.5Z_{\delta S} & -0.5Z_{\delta S} & -0.5Z_{\delta F} & -0.5Z_{\delta F} & -Z_{\delta LEF} \end{bmatrix} \tag{3-11}$$

$$\begin{bmatrix} z_{y4} & z_{y5} & z_{y6} & z_{y7} & z_{y8} \end{bmatrix} = \begin{bmatrix} -0.5Y_{\delta S} & 0.5Y_{\delta S} & -0.5Y_{\delta F} & 0.5Y_{\delta F} & -Y_{\delta R} \end{bmatrix} \tag{3-12}$$

where the subscripts n and y represent normal and lateral axes, respectively.

3.1.1 Evaluation of Measurement Noise Covariance R

The sensor noise variances were determined by using flight test data for NASA's F-8. This approach was utilized in previous efforts [10,11,18]. Based upon the noise correlation time constants ("0.02 sec for angular rates, 0.01 sec for accelerations, and 0.005 sec for the angle of attack" [18]), the bandwidth of the sensor noise is very large as compared to the frequency response of an aircraft. We conclude the sensor noise is essentially white over the bandwidth of the aircraft. The root mean square (RMS) noise values are listed in *Table 3.2*, followed by the values (in appropriate units) used within this thesis. While one might expect the VISTA F-16 sensors to be significantly better than those used in the NASA F-8 experiments, the lack of data to substantiate those claims forced the use of the conservative values listed in *Table 3.2* [12,13,18].

Table 3.2 Sensor Noise RMS values

Variable	RMS Noise	Units	RMS Noise	Units
u	0.1	ft/sec	0.1	ft/sec
α	0.10 to 0.30	deg	0.0017 to 0.0052	rad
q	0.15 to 0.50	deg/sec	0.0026 to 0.0087	rad/sec
A_{ncg}	0.02 to 0.06	g's	0.6400 to 1.9000	ft/sec^2
p	0.75 to 2.00	deg/sec	0.0130 to 0.0350	rad/sec
r	0.15 to 0.50	deg/sec	0.0026 to 0.0087	rad/sec
A_{ycg}	0.003 to 0.04	g's	0.0966 to 1.3000	ft/sec^2

Conservative estimates, relative to anticipated VISTA F-16 sensor performance, for σ values are established as 0.1 *ft/sec*, 0.004 *rad*, 0.006 *rad*, 1 *ft/sec*^{2}, 0.02 *rad/sec*, 0.006 *rad/sec*, and 0.6 *ft/sec*^{2}, respectively. The covariance matrix R is given by Eq. (3-13); note that a listing such as 1.0^{-2} in the table is a shorthand representation for 1.0×10^{-2}, etc.

$$R = \begin{bmatrix} 1.0^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.6^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.6^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.0^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.6^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.6^{-1} \end{bmatrix} \begin{bmatrix} ft/\mathrm{sec} \\ rad \\ rad/\mathrm{sec} \\ ft/\mathrm{sec}^2 \\ rad/\mathrm{sec} \\ rad/\mathrm{sec} \\ ft/\mathrm{sec}^2 \end{bmatrix}^2 \tag{3-13}$$

3.1.2 Actuator Dynamics Model

The VISTA F-16 control surface actuators are modelled as fourth order actuators. The transfer function used to describe each of the actuators is given by [Appendix B]

$$T(s) = \frac{\delta C}{\delta C_{CMD}} = \frac{(20.2)(141.4)(5214.5)}{(s+20.2)(s+141.4)(s^2+107.0s+5214.5)} \tag{3-14}$$

where δC is the actuator (control surface) position and δC_{CMD} is the command surface position. The left and right stabilators, flaperons, and the rudder use this fourth order model to represent the surface dynamics. The leading edge flap is represented as a first order lag given by Eq. (3-15):

$$T_{LEF}(s) = \frac{16.0}{s+16.0} \tag{3-15}$$

The actuator/control surface transfer functions can be written is state-space phase variable form as

$$\begin{bmatrix} \delta\dot{C} \\ \delta\ddot{C} \\ \delta\dddot{C} \\ \delta\ddddot{C} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -d1 & -d2 & -d3 & -d4 \end{bmatrix} \begin{bmatrix} \delta C \\ \delta\dot{C} \\ \delta\ddot{C} \\ \delta\dddot{C} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \delta C_{CMD} \tag{3-16}$$

with the output equation expressed by

$$Y_C(t) = \begin{bmatrix} d5 & 0 & 0 & 0 \end{bmatrix} \tag{3-17}$$

where δC represents the different control surfaces (δS, δF, and δR). The MMAE software allows these control surfaces to be modelled as either a first order, second order, or fourth order representation. The first order approximation, implemented in the filters, allows for a realistic simulation without the complexity of a higher order state-space model:

$$T_C(s) = \frac{20.2}{s + 20.2} \tag{3-18}$$

Previous efforts encountered robustness difficulties associated with reduced-order actuator models [16]. The results from that effort provided additional motivation for not using a zero-order actuator model in the elemental filters models. As mentioned earlier, the leading edge flap is always modelled as a first order lag. For most implementation purposes, the first order model approximation of Eq. (3-18) is used in *both* the truth model and the filters. Some cases were run with the fourth order representation in the truth model and the first order representation in the filters. Results indicated some minor performance degradation. Since the actual model in the aircraft is effectively fourth order, actual implementation would require the filters to contain the fourth order actuator model to maximize the algorithm's performance capabilities; final implementable controllers would most likely incorporate first order filter design models, and the truth model repeats this first order model in order to keep computational loading down without any severe misrepresentations of performance attributes.

Actuator position and rate limiting are included in the truth model. Position and rate limits for the VISTA F-16 are included in *Table 3.3*. Recall that the flight control system contains other limiting functions, including internal signal and rate limiting and pilot command limiting.

Table 3.3 Actuator Position and Rate Limits

Control Surface	Limits	Units
δS - Stabilator	+/- 21.0 +/- 60.0	deg deg/sec
δF - Flaperons	+ 20.0 / - 23.0 +/- 61.0	deg deg/sec
δR - Rudder	+/- 30.0 +/- 120.0	deg deg/sec

3.1.3 Dryden Wind Model

This section is based in part on the development presented by Pogoda [8,9]. The development of the Dryden wind model is taken from MIL-STD-1797A [14] and is presented in state space representation by Eq. (3-19). The gust states are represented using the subscript g. The gust states are given as: u_g, α'_g, α_g, q_g, p_g, β'_g, β_g, r_g. The gust variable u_g is the gust or air mass velocity effect on the aircraft forward velocity; α'_g is the air mass velocity effect on the aircraft angle of attack; α_g is the air mass velocity effect on the angle of attack; q_g is the air mass velocity effect on the aircraft pitch rate; p_g is the air mass velocity effect on the aircraft roll rate; β'_g is the air mass velocity effect on the aircraft sideslip angle; β_g is the air mass velocity effect on the aircraft sideslip angle; and r_g is the air mass velocity effect on the aircraft yaw rate.

$$\begin{bmatrix} \dot{u}_g \\ \dot{\alpha}'_g \\ \dot{\alpha}_g \\ \dot{q}_g \\ \dot{p}_g \\ \dot{\beta}'_g \\ \dot{\beta}_g \\ \dot{r}_g \end{bmatrix} = \begin{bmatrix} \frac{-V_T}{L_u} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-V_T}{2L_w} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_1 & \frac{-V_T}{2L_w} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & \frac{-\pi V_T^2}{8bL_w} & \frac{-\pi V_T}{4b} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-\pi V_T}{4b} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-V_T}{2L_v} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_3 & \frac{-V_T}{2L_v} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_4 & \frac{\pi V_T^2}{6bL_v} & \frac{-\pi V_T}{3b} \end{bmatrix} \begin{bmatrix} u_g \\ \alpha'_g \\ \alpha_g \\ q_g \\ p_g \\ \beta'_g \\ \beta_g \\ r_g \end{bmatrix} +$$

$$+ \begin{bmatrix} \sigma_u \sqrt{\frac{2V_T}{L_u}} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \sigma_w \sqrt{\frac{3}{2L_w V_T}} & 0 & 0 \\ 0 & \frac{\sigma_w \pi}{b} \sqrt{\frac{3V_T}{32L_w}} & 0 & 0 \\ 0 & 0 & \sigma_w \left(\frac{\pi^{10} V_T^3}{128{,}000\, b^7 L_w^2} \right)^{1/6} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \sigma_v \sqrt{\frac{3}{2L_v V_T}} \\ 0 & 0 & 0 & \frac{-\sigma_v \pi}{b} \sqrt{\frac{V_T}{6L_w}} \end{bmatrix} \begin{bmatrix} w_u \\ w_w \\ w_p \\ w_v \end{bmatrix} \qquad (3\text{-}19)$$

with *d1*, *d2*, *d3*, and *d4* given by:

$$\begin{bmatrix} d1 \\ d2 \end{bmatrix} = \begin{bmatrix} \sigma_w(1-\sqrt{3})\sqrt{\dfrac{V_T}{8L_w^3}} \\ \dfrac{\sigma_w(1-\sqrt{3})\pi}{b}\sqrt{\dfrac{V_T^3}{128L_w^3}} \end{bmatrix} \qquad \begin{bmatrix} d3 \\ d4 \end{bmatrix} = \begin{bmatrix} \sigma_v(1-\sqrt{3})\sqrt{\dfrac{V_T}{8L_v^3}} \\ \dfrac{\sigma_v(1-\sqrt{3})\pi}{b}\sqrt{\dfrac{V_T^3}{128L_v^3}} \end{bmatrix}$$

where w_i are independent white noise sources of unit strength, V_T is the aircraft velocity, b is the wing span, L_i are the appropriate scale lengths, and σ_i are the turbulence RMS values. An altitude of 20000 ft is selected for this thesis. Scale lengths are chosen for the medium/high altitude values, specifically $L_u = 2L_v = 2L_w =$ 1750 ft and $\sigma_u = \sigma_v = \sigma_w = \sigma$ [14], where σ is based on light, moderate, or severe turbulence categories from Figure 262, page 652 of MIL-STD-1797A [14]. The design models computed in this thesis assume the value of $\sigma = 1$ (light turbulence). Pilot perception of turbulence is often significantly different from the categories listed in MIL-STD-1797A. Sufficient evidence exists to suggest a value of $\sigma = 1$ may more accurately describe light to moderate turbulence [14]. By basing the wind disturbance calculations on $\sigma = 1$, evaluations can be made of higher turbulence levels by multiplying the design model equivalent discrete time dynamics noise covariance matrix Q_{DM} by $(\sigma')^2$.

The addition of the Dryden wind model modifies the aircraft dynamics model of Eq. (3-5). Relating aircraft motion to air mass motion (wind) in an inertial reference frame yields

$$v_{a/i} = v_{a/m} + v_{m/i} \tag{3-20}$$

where the subscripts *a*, *i*, *m* are aircraft, inertial, and air mass respectively. Then the *u* from Equation (3-5) may be expressed as

$$u = X'_\theta(\theta-\theta_g) + X'_u(u-u_g) + X'_\alpha(\alpha-\alpha_g) + X'_q(q-q_g) + X'_{\delta S}\delta S + X'_{\delta F}\delta F + X'_{LEF}\delta LEF \tag{3-21}$$

where Θ_g, u_g, α_g represents the gust or air mass velocity effect on the state variable. Augmenting the air mass velocities from the wind model with the truth model states yields terms to be added to Eq.(3-5) as shown in Eq. (3-22).

$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \alpha \\ q \\ \dot{\phi} \\ \beta \\ \dot{p} \\ \dot{r} \\ \vdots \end{bmatrix} = \begin{bmatrix} \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdots & -X'_u & 0 & -X'_\alpha & -X'_q & 0 & 0 & 0 & 0 \\ \cdots & -Z'_u & 0 & -Z'_\alpha & -Z'_q & 0 & 0 & 0 & 0 \\ \cdots & -M'_u & 0 & -M'_\alpha & -M'_q & 0 & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 & -Y'_p & 0 & -Y'_\beta & -Y'_r \\ \cdots & 0 & 0 & 0 & 0 & -L'_p & 0 & -L'_\beta & -L'_r \\ \cdots & 0 & 0 & 0 & 0 & -N'_p & 0 & -N'_\beta & -N'_r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ u_g \\ \alpha'_g \\ \alpha_g \\ q_g \\ p_g \\ \beta'_g \\ \beta_g \\ r_g \end{bmatrix} \qquad (3\text{-}22)$$

3.1.4 Discrete Gust Model

This section is based in part from the development presented by Pogoda [9]. From MIL-STD-1797A [14], the magnitude of wind shear is computed using Figure 265. In equation form,

$$v_x = k\,\sigma_u, \qquad v_y = k\,\sigma_v, \qquad v_z = k\,\sigma_w \qquad (3\text{-}23)$$

where k varies from 0.4 to 2.6 and is tuned for each gust to excite the natural frequencies of the aircraft. The rms intensities of forward, lateral, and vertical turbulence are described by σ_u, σ_v, and σ_w, respectively. The magnitude of the wind sheer in each component direction are described by v_x, v_y, and v_z. After calculating the velocities for a given disturbance level, the gust variables are converted into the state variables of the Dryden wind model. The discrete gust velocities are defined as

$$v_x = \Delta u_g, \qquad v_y = \Delta v_g, \qquad v_z = \Delta w_g \qquad (3\text{-}24)$$

and using the equalities

$$\Delta\alpha_g = \frac{1}{V_T}\Delta w_g \qquad \Delta\beta_g = \frac{1}{V_T}\Delta v_g \qquad (3\text{-}25)$$

provides the following relationships:

$$\Delta u_g = v_x \qquad (3\text{-}26)$$

Substitution of the values Δu_g, $\Delta\alpha_g$, $\Delta\beta_g$ into the Dryden wind model equations provides additional terms with

$$\Delta\alpha_g = \frac{v_z}{V_T} \tag{3-27}$$

$$\Delta\beta_g = \frac{v_y}{V_T} \tag{3-38}$$

which to augment the wind disturbance model, modifying Eqs. (3-19) and (3-22) with an additional term $d_g(t)$, as:

$$\dot{x}_g(t) = A_g x_g(t) + G_g w_g(t) + d_g(t) \tag{3-29}$$

where $d_g(t)$ is given by Equation (3-30):

$$d_g(t) = \begin{bmatrix} \frac{-V_T}{L_u} v_x & 0 & \frac{-1}{2L_w} v_z & \frac{-\pi V_T}{8bL_w} & 0 & 0 & \frac{-1}{2L_v} v_y & \frac{\pi V_T}{6bL_v} v_y & \underline{0}^T \end{bmatrix}^T \tag{3-30}$$

3.2 Truth Model

This section is based in part by the development presented in Martin [18]. The aircraft truth model is represented as a continuous-time dynamic system

$$\dot{x}_{TM}(t) = A_{TM} x_{TM}(t) + B_{TM} u_{TM}(t) + G_{TM} w_{TM}(t) + d_{TM}(t) \tag{3-31}$$

with an output equation given by

$$y(t) = C_{TM} x_{TM}(t) + D_{TM} u_{TM}(t) \tag{3-31}$$

and the discrete-time measurement equation

$$z(t_i) = H_{TM} x_{TM}(t) + D_{zT} u_{TM}(t_i) + v(t_i) \tag{3-32}$$

where $w_{TM}(t)$ is a zero-mean white Gaussian noise of strength Q_{TM}, statistically representing aircraft disturbances generated by random wind buffet. For the 4th-order actuator representations with state elements δS_L, δS_R, etc, $u_{AC}(t)$ becomes δS_{Lcmd}, δS_{Rcmd}, etc. The $\delta\{\}_{cmd}$ represents the input command to the actuator model, and $\delta\{\}$ is the output of the actuator which is provided as input to the aircraft model. The aircraft state model, Eq. (3-5), is augmented to incorporate the actuator and wind disturbance models. In this thesis, the wind models are the only source of disturbances to the aircraft model. Explicitly, the nonzero components in the augmented truth model matrices G_{TM} and d_{TM} come from the wind model matrices G_g and d_g.

The remainder of this section presents the aircraft truth model with fourth-order actuators and the complete

Dryden wind model. Results presented in Chapter 4 include cases for both fourth-order actuators and first-order actuators in the truth model. The design models were created with first-order actuator representations. The truth model didn't incorporate either the Dryden or the discrete gust models, and it was divided into longitudinal and lateral directional components. The full-scale wind model is replaced with the "appropriately scaled" white noise. The white noise used in the truth model is scaled relative to the strength levels used in the design models. An empirical approach is taken to provide white noise that is noticeable but not offensive from a pilot's perspective. The noise strength levels were chosen to resemble light turbulence. Also, the flight control system incorporates cross-channel effects. Limitations in the size and scope of the thesis effort and the unnecessary computational loading precluded the addition of these models. The full representation of the truth model is included in the next four pages:

$$\begin{bmatrix} \theta \\ \dot{u} \\ \alpha \\ q \\ \dot{\phi} \\ \beta \\ \dot{p} \\ \dot{r} \\ \delta\dot{S}_L \\ \delta\ddot{S}_L \\ \delta\dddot{S}_L \\ \delta\ddddot{S}_L \\ \delta\dot{S}_R \\ \delta\ddot{S}_R \\ \delta\dddot{S}_R \\ \delta\ddddot{S}_R \\ \delta\dot{F}_L \\ \delta\ddot{F}_L \\ \delta\dddot{F}_L \\ \delta\ddddot{F}_L \\ \delta\dot{F}_R \\ \delta\ddot{F}_R \\ \delta\dddot{F}_R \\ \delta\ddddot{F}_R \\ \delta\dot{R} \\ \delta\ddot{R} \\ \delta\dddot{R} \\ \delta\ddddot{R} \\ \delta LE\dot{F} \\ \dot{u}_g \\ \alpha'_g \\ \alpha_g \\ \dot{q}_g \\ \dot{p}_g \\ \beta'_g \\ \beta_g \\ \dot{r}_g \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & b_{11} & 0 & 0 & 0 & - \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & b_{21} & 0 & 0 & 0 & - \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & b_{31} & 0 & 0 & 0 & - \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & b_{41} & 0 & 0 & 0 & - \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & b_{51} & 0 & 0 & 0 & - \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & b_{61} & 0 & 0 & 0 & - \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & b_{71} & 0 & 0 & 0 & - \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & b_{81} & 0 & 0 & 0 & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{11} & s_{12} & s_{13} & s_{14} & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{21} & s_{22} & s_{23} & s_{24} & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{31} & s_{32} & s_{33} & s_{34} & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{41} & s_{42} & s_{43} & s_{44} & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - \end{bmatrix}$$

Truth Model, page 1

$$
\begin{bmatrix}
b_{12} & 0 & 0 & 0 & b_{13} & 0 & 0 & 0 & b_{14} & 0 & 0 & 0 \\
b_{22} & 0 & 0 & 0 & b_{23} & 0 & 0 & 0 & b_{24} & 0 & 0 & 0 \\
b_{32} & 0 & 0 & 0 & b_{33} & 0 & 0 & 0 & b_{34} & 0 & 0 & 0 \\
b_{42} & 0 & 0 & 0 & b_{43} & 0 & 0 & 0 & b_{44} & 0 & 0 & 0 \\
b_{52} & 0 & 0 & 0 & b_{53} & 0 & 0 & 0 & b_{54} & 0 & 0 & 0 \\
b_{62} & 0 & 0 & 0 & b_{63} & 0 & 0 & 0 & b_{64} & 0 & 0 & 0 \\
b_{72} & 0 & 0 & 0 & b_{73} & 0 & 0 & 0 & b_{74} & 0 & 0 & 0 \\
b_{82} & 0 & 0 & 0 & b_{83} & 0 & 0 & 0 & b_{84} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{11} & s_{12} & s_{13} & s_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{21} & s_{22} & s_{23} & s_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{31} & s_{32} & s_{33} & s_{34} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{41} & s_{42} & s_{43} & s_{44} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_{11} & s_{12} & s_{13} & s_{14} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_{21} & s_{22} & s_{23} & s_{24} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_{31} & s_{32} & s_{33} & s_{34} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_{41} & s_{42} & s_{43} & s_{44} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{11} & s_{12} & s_{13} & s_{14} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{21} & s_{22} & s_{23} & s_{24} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{31} & s_{32} & s_{33} & s_{34} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{41} & s_{42} & s_{43} & s_{44} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Truth Model, page 2

$$
\left.\begin{array}{cccccccccccccc}
- & b_{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & b_{25} & 0 & 0 & 0 & 0 & -X'_u & 0 & -X'_\alpha & -X'_q & 0 & 0 & 0 & 0 \\
- & b_{35} & 0 & 0 & 0 & 0 & -Z'_u & 0 & -Z'_\alpha & -Z'_q & 0 & 0 & 0 & 0 \\
- & b_{45} & 0 & 0 & 0 & 0 & -M'_u & 0 & -M'_\alpha & -M'_q & 0 & 0 & 0 & 0 \\
- & b_{55} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & b_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Y'_p & 0 & -Y'_\beta & -Y'_R \\
- & b_{75} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -L'_p & 0 & -L'_\beta & -L'_R \\
- & b_{85} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -N'_p & 0 & -N'_\beta & -N'_R \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & s_{11} & s_{12} & s_{13} & s_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & s_{21} & s_{22} & s_{23} & s_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & s_{31} & s_{32} & s_{33} & s_{34} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & s_{41} & s_{42} & s_{43} & s_{44} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & s_{55} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 0 & w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} & w_{17} & w_{18} \\
- & 0 & 0 & 0 & 0 & 0 & w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26} & w_{27} & w_{28} \\
- & 0 & 0 & 0 & 0 & 0 & w_{31} & w_{32} & w_{33} & w_{34} & w_{35} & w_{36} & w_{37} & w_{38} \\
- & 0 & 0 & 0 & 0 & 0 & w_{41} & w_{42} & w_{43} & w_{44} & w_{45} & w_{46} & w_{47} & w_{48} \\
- & 0 & 0 & 0 & 0 & 0 & w_{51} & w_{52} & w_{53} & w_{54} & w_{55} & w_{56} & w_{57} & w_{58} \\
- & 0 & 0 & 0 & 0 & 0 & w_{61} & w_{62} & w_{63} & w_{64} & w_{65} & w_{66} & w_{67} & w_{68} \\
- & 0 & 0 & 0 & 0 & 0 & w_{71} & w_{72} & w_{73} & w_{74} & w_{75} & w_{76} & w_{77} & w_{78} \\
- & 0 & 0 & 0 & 0 & 0 & w_{81} & w_{82} & w_{83} & w_{84} & w_{85} & w_{86} & w_{87} & w_{88}
\end{array}\right]
\begin{bmatrix}
\theta \\ u \\ \alpha \\ q \\ \phi \\ \beta \\ p \\ r \\
\delta S_L \\ \delta \dot{S}_L \\ \delta \ddot{S}_L \\ \delta \dddot{S}_L \\
\delta S_R \\ \delta \dot{S}_R \\ \delta \ddot{S}_R \\ \delta \dddot{S}_R \\
\delta F_L \\ \delta \dot{F}_L \\ \delta \ddot{F}_L \\ \delta \dddot{F}_L \\
\delta F_R \\ \delta \dot{F}_R \\ \delta \ddot{F}_R \\ \delta \dddot{F}_R \\
\delta R \\ \delta \dot{R} \\ \delta \ddot{R} \\ \delta \dddot{R} \\
\delta LEF \\
u_g \\ \alpha'_g \\ \alpha_g \\ q_g \\ p_g \\ \beta'_g \\ \beta_g \\ r_g
\end{bmatrix}
+ -
$$

Truth model, page 3

$$
- + \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
b_{s1} & 0 & 0 & 0 & 0 & 0 \\
b_{s2} & 0 & 0 & 0 & 0 & 0 \\
b_{s3} & 0 & 0 & 0 & 0 & 0 \\
b_{s4} & 0 & 0 & 0 & 0 & 0 \\
0 & b_{s1} & 0 & 0 & 0 & 0 \\
0 & b_{s2} & 0 & 0 & 0 & 0 \\
0 & b_{s3} & 0 & 0 & 0 & 0 \\
0 & b_{s4} & 0 & 0 & 0 & 0 \\
0 & 0 & b_{s1} & 0 & 0 & 0 \\
0 & 0 & b_{s2} & 0 & 0 & 0 \\
0 & 0 & b_{s3} & 0 & 0 & 0 \\
0 & 0 & b_{s4} & 0 & 0 & 0 \\
0 & 0 & 0 & b_{s1} & 0 & 0 \\
0 & 0 & 0 & b_{s2} & 0 & 0 \\
0 & 0 & 0 & b_{s3} & 0 & 0 \\
0 & 0 & 0 & b_{s4} & 0 & 0 \\
0 & 0 & 0 & 0 & b_{s1} & 0 \\
0 & 0 & 0 & 0 & b_{s2} & 0 \\
0 & 0 & 0 & 0 & b_{s3} & 0 \\
0 & 0 & 0 & 0 & b_{s4} & 0 \\
0 & 0 & 0 & 0 & 0 & b_{s5} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta S_L \\ \delta S_R \\ \delta F_L \\ \delta F_R \\ \delta R \\ \delta LEF
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
g_{w11} & g_{w12} & g_{w13} & g_{w14} \\
g_{w21} & g_{w22} & g_{w23} & g_{w24} \\
g_{w31} & g_{w32} & g_{w33} & g_{w34} \\
g_{w41} & g_{w42} & g_{w43} & g_{w44} \\
g_{w51} & g_{w52} & g_{w53} & g_{w54} \\
g_{w61} & g_{w62} & g_{w63} & g_{w64} \\
g_{w71} & g_{w72} & g_{w73} & g_{w74} \\
g_{w81} & g_{w82} & g_{w83} & g_{w84}
\end{bmatrix}
\begin{bmatrix}
w_u \\ w_w \\ w_p \\ w_v
\end{bmatrix}
+ \begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8
\end{bmatrix}
\qquad (3\text{-}33)
$$

Truth model, page 4

where a_{ii} and b_{ii} are elements from the aircraft dynamics model, s_{ii} and b_{si} are elements from the actuator dynamics model, w_{ii} and g_{wii} are elements from the Dryden wind model, and d_i is the wind shear disturbance input [18]. The w_i disturbance inputs are independent white Gaussian noises of strength

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} ft^2/(rad\text{-sec}) \\ ft^2/(rad\text{-sec}) \\ rad/\text{sec} \\ ft^2/(rad\text{-sec}) \end{bmatrix} \tag{3-34}$$

where w_u, w_v, w_w, and w_p are given in the units $[ft/sec]^2/[rad/sec]$, $[ft/sec]^2/[rad/sec]$, $[ft/sec]^2/[rad/sec]$, and $[rad/sec]^2/[rad/sec]$, respectively [9,10,18].

Recall this model is implemented as a perturbation model so that the initial conditions on the aircraft states are zero ($x_{TM}(t_o) = 0$). The covariance model at the initial conditions, $P_{TM}(t_o)$, is set to the steady state covariance value, $P_{TM}(t)$ since the simulation begins with the aircraft in steady state trim flight.

3.3 *The Design Model*

This section is based in part on work presented by Martin [18]. The design model is similar to the truth model in form. However, the design model is generally of lower order to facilitate faster on-line computations necessary in flight control applications. In this thesis, the design model incorporates a 0-order disturbance model and a 1st-order actuator model.

The design model is given by the form

$$\dot{x}_{DM}(t) = A_{DM}x_{DM}(t) + B_{DM}u_{DM}(t) + G_{DM}w_{DM}(t) \tag{3-35}$$

$$y(t) = C_{DM}x_{DM}(t) + D_{DM}u_{DM}(t) \tag{3-36}$$

$$z(t_i) = H_{DM}x_{DM}(t_i) + D_{zD}u_{DM}(t) + v(t_i) \tag{3-38}$$

with $y(t)$, $z(t_i)$, $v(t_i)$ being equivalent in both the aircraft and truth models.

The augmentation of the aircraft model found in Section 3.1 with first-order actuators yields a 14th-order design model. The augmentation of the aircraft model with the 1st-order actuator representation for δS, δF, *and* δR

$$S_1(s) = \frac{20.2}{s + 20.2} \tag{3-39}$$

and for the leading edge flap

$$S_2(s) = \frac{16.0}{s + 16.0} \tag{3-39}$$

is as done with the incorporation of 4th-order actuators in the truth model. In this thesis, 1st-order actuators were chosen over 0-order to improve control response [16]. Nesline and Zarchan [16] highlighted robustness issues associated with the use of reduced order actuator models in the design models. This finding provided motivation for not using zero-order actuators in the design models. Initial results using zero-order actuators in the design model demonstrated poor algorithm failure detection and identification performance.

The 0-order wind model is generated by approximating the 6 time-correlated wind disturbances $u_g(t)$, $\alpha_g(t)$, etc. from the Dryden wind model with the 6 white noises

$$w_{DM}(t) = \begin{bmatrix} w_{u_g}(t) & w_{\alpha_g}(t) & w_{q_g}(t) & w_{p_g}(t) & w_{\beta_g}(t) & w_{r_g}(t) \end{bmatrix} \tag{3-40}$$

where the white noise strengths are calculated by averaging the wind model power spectral density over the appropriate frequency range [18]. The strengths of the white noise approximations are generated by using the appropriate relationships from MIL-STD-1797A and generating the power spectral densities. The strength is calculated as the average magnitude of the power spectral density from 0 rad/sec to the aircraft bandwidth for the variable of interest. The Q_{DM} matrix given by

$$Q_{DM} = \begin{bmatrix} Q_{u_g} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{\alpha_g} & Q_{\alpha_g q_g} & 0 & 0 & 0 \\ 0 & Q_{\alpha_g q_g} & Q_{q_g} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{p_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{\beta_g} & Q_{\beta_g r_g} \\ 0 & 0 & 0 & 0 & Q_{\beta_g r_g} & Q_{r_g} \end{bmatrix} \tag{3-41}$$

Table 3.4 Zero Order White Noise Strength

Variable	Aircraft Bandwidth	Units	Average Noise Strength	Units
u	0.25	rad/sec	4.5^{-1}	ft^2/rad-sec
α	4	rad/sec	3.0^{-6}	rad-sec
q	20	rad/sec	1.5^{-6}	rad/sec
α *vs* q	4	rad/sec	1.1^{-8}	rad^2
p	15	rad/sec	6.0^{-6}	rad/sec
β	3.5	rad/sec	3.0^{-6}	rad-sec
r	7	rad/sec	2.4^{-6}	rad/sec
β *vs* r	3.5	rad/sec	6.3^{-9}	rad^2

Since $\alpha_g(t)$ and $q_g(t)$ are both functions of $w_{\alpha g}(t)$, and $\beta_g(t)$ and $r_g(t)$ are functions of $w_{\beta g}(t)$, the cross spectral densities $\Phi_{\alpha g q g}(s)$ and $\Phi_{\beta g r g}(s)$ must be calculated to form the off-diagonal elements of Eq.(3-41). A description of these calculations is given by Martin [18].

The bandwidths for the aircraft states and their associated average noise strengths based on a wind turbulence RMS value of $\sigma = 1$ *ft/sec* are given in *Table 3.4*. Since each power spectral density equation contains a σ^2 term, these average noise values can simply be multiplied by $(\sigma')^2$ ($= k\sigma^2$ where k is a multiple) for increased RMS wind turbulence values. The white noise strengths approximations are combined to form the Q matrix:

$$Q_{DM} = \begin{bmatrix} 4.5^{-2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.0^{-6} & 1.1^{-8} & 0 & 0 & 0 \\ 0 & 1.1^{-8} & 1.5^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.0^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.0^{-6} & 6.3^{-9} \\ 0 & 0 & 0 & 0 & 6.3^{-9} & 2.4^{-6} \end{bmatrix} \tag{3-42}$$

where the cross spectral densities yield the nonzero off-diagonal terms of Q_{DM}. The off-diagonal terms indicate that these white noise terms are not independent of each other. Note the reversal of two rows in the G matrix. The order of the rows within the state matrix differs from the order of columns within the wind disturbance model, w_{DM}, necessitating the switching of two rows in the G matrix.

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-43}$$

These white noises are incorporated into the aircraft equations by multiplying the G matrix by the primed aircraft dimensional derivatives, as shown in Equation (3-22), to convert the $w_{ig}(t)$ random disturbances into the aircraft derivative states $u(t)$, $\alpha(t)$, etc. This yields the design model G_{DM} matrix

$$G_{DM} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ X'_u & X'_\alpha & X'_q & 0 & 0 & 0 \\ Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 \\ M'_u & M'_\alpha & M'_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y'_p & Y'_\beta & Y'_r \\ 0 & 0 & 0 & L'_p & L'_\beta & L'_r \\ 0 & 0 & 0 & N'_p & N'_\beta & N'_r \end{bmatrix} \tag{3-44}$$

where X', Z', M', Y', L', and N' are the dimensional derivatives as defined in Section 3.1. Higher order design models are created by augmenting Eq. (3-44) with additional rows of zeros.

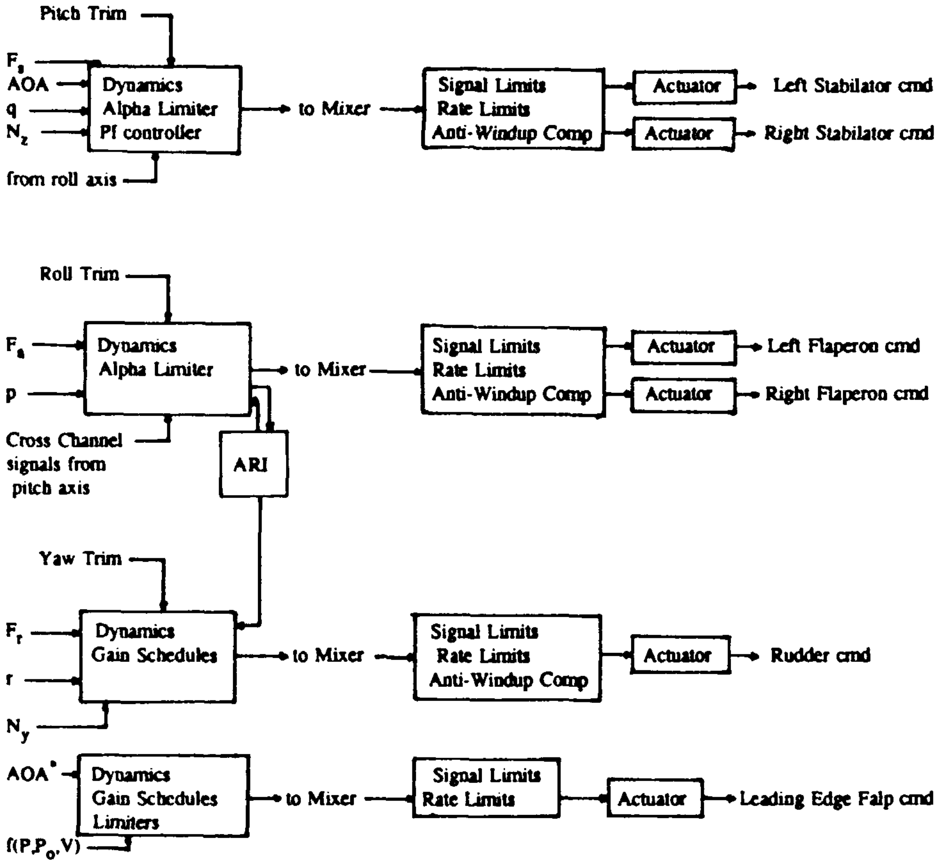

Figure 3.1 VISTA F-16 Functional Block Diagram

3.4 Flight Control System

The MMAE algorithm is applied to the VISTA F-16 flight control system in this research application. The VISTA F-16 flight control system is the Block 40 digital fly-by-wire F-16 system [not included]. This flight control system incorporates seven input sensors, namely, angle of attack, velocity, pitch rate, normal acceleration, roll rate, yaw rate, and lateral acceleration. These sensor inputs are processed as shown in *Figure 3.1*. In this effort six actuators are modeled, including: the left and right stabilator, the left and right flaperon, the rudder, and the leading edge flap. General features of the system include: pilot command limiting, angle of attack limiting, and normal acceleration limiting, anti-windup compensation, pitch and roll command mixers, gain scheduling, autonomous leading edge flap actuation, and numerous autopilot functions.

The input signals are processed, combined if required, limited, and output. Thirteen primary control system inputs are of interest in this effort: the seven sensor inputs mentioned, the pilot commands in three axes, and the trim inputs in three axes. *Figure 3.1* depicts the flight control internal operations in a block diagram format. The full block diagram is not shown to protect proprietary rights.

The pitch control system converts the pilot commanded stick force into a pitch command gradient which is summed with the trim inputs. This command is limited and filtered and then summed with autopilot inputs and sensor feedbacks. The feedback includes a combination of angle of attack, pitch rate, and normal load factor and an angle of attack limiting function which is based upon the flight condition and inputs from the roll axis. The summed signal is scaled and limited prior to entry into the elevator command proportional plus integral (PI) control loop. The output of the PI controller is sent to the control surface mixer.

The roll control system converts the roll command in lbs to a roll command in g's via a roll command gradient. This signal is filtered and summed with the roll trim command, autopilot command, and the filtered roll rate feedback. The summed signal is limited and sent to the aileron rudder interconnect (ARI) and the control surface mixer.

The ARI uses the signal from the roll axis and the angle of attack to develop a coordinating input for the yaw axis. The block diagram routes a signal through the ARI to produce a differential tail command sent to the control surface mixer.

The yaw axis produces a command conversion similar to the pitch and roll axes. This command is filtered and scaled according to the flight condition, angle of attack, and limited angle of attack. This process is not explicitly shown in *Figure 3.1* due to proprietary restrictions. The scaled input is summed with the yaw trim command and summed functions of the yaw rate, lateral acceleration, and ARI output. The signal is passed through a yaw structural filter and sent to the control surface mixer.

The leading edge flap is an autonomous control surface with no differential actuation between left and right wings possible. The leading edge flap command is generated by taking the output of the sensor and sending it through a limiter, filtering it, scaling the signal, and summing it with a bias (omitted from *Figure 3.1*) and a function based upon dynamic pressure, static pressure, and total pressure. The signal passes through a function designed to select the larger of the signal or another generated function. The larger signal is selected, limited, and sent to the control surface mixer.

The control surface mixer provides software rate and position limiting functions. This function is often called "anti-windup" compensation [6,7] (Anti-windup compensation is essential for PI controllers). The stabilator (elevator) portion of the control surface mixer accepts inputs from the pitch axis, the ARI (not explicitly shown in *Figure 3.1*) , and the roll axis. The ARI and roll signals are added to the pitch signals in the right stabilator path and subtracted from the pitch signal in the left stabilator path. After the signal summation, each path is limited, differentiated, rate limited, integrated, and sent through the actuator. The actuators have position and rate limits given in *Table 3.3*. The roll axis contains a summing scheme (not explicitly shown in *Figure 3.1* to protect proprietary rights) to control the roll inputs at the higher command levels. The signal at the roll command portion of the surface command mixer limits the command, differentiates, rate limits, integrates and sends the signal to the actuator. The signal sent to the yaw axis portion of the control surface mixer is limited, differentiated, rate limited, integrated, and sent to the rudder actuator. The signal sent to the leading edge flap control surface mixer is summed with an error feedback (not explicitly shown in *Figure 3.1* to protect proprietary rights), scaled, limited, filtered, rate limited, scaled, mechanically limited, and sent to the actuator.

3.5 Actuator Failures

Actuator failures are grouped into two categories: hard and soft. Hard failures are represented as the complete loss of actuator control with the surface trailing edge positioned at zero degrees of deflection. The leading edge flap is not included in the failure scenarios. This surface is not flight-critical and its inclusion only complicates the ambiguity problems with the other flight-critical surfaces. While the leading edge flap's omission simplifies the ambiguity issues, a filter could be designed to detect this failure if desired. As mentioned in Chapter 1, hard actuator failures are modeled by zeroing out columns of the B matrix.

Soft failures are represented as a partial loss of actuator control authority (as opposed to modelling a soft failure as a partial loss of actuator rate capability). Arbitrary percentages are selected to represent the loss of control authority. The critical issue is whether the algorithm will be able to blend the outputs of the fully functional filter with the appropriate hard failure filter to achieve the proper control for the soft failure scenario. The soft failures are introduced by multiplying the appropriate column of the B matrix in the truth model with the percentage of control authority available.

3.6 Sensor Failures

Sensor failures are grouped into two categories: hard and soft. Hard sensor failures are modelled by assuming the complete loss of the sensor's output. All of the flight-critical sensors are included in the model. A hard sensor failure is represented by zeroing out the appropriate row of the H matrix. Unlike actuator failures in which one surface can affect many different residuals, sensor failures have a direct relationship to the associated scalar residuals and are easily detected.

Soft sensor failures are modelled by assuming increased sensor noise (as opposed to a bias). Soft sensor failures are represented by increasing the appropriate sensor variance in the sensor noise covariance matrix, R, within the truth model. Detection of a soft sensor failure is considerably harder when the elemental variance increment, from the sensor noise covariance matrix, is small.

3.7 Multiple Failures

Multiple failures include all of the two-failure combinations: dual hard actuator failures, dual hard sensor failures, hard actuator and sensor failures; dual soft actuator failures, dual soft sensor failures, soft actuator and sensor failures; combinations of hard actuator with soft sensor failures, and combinations of soft actuator with hard sensor failures. Multiple failures are implemented using the "moving bank" hierarchical structure described in Chapter 2. Multiple failures are implemented according to a delay sequence, with the time between failures selected by the user. Multiple failures require the generation of a bank of filters designed for the first failure and any second failure including a no-failure filter. In actuality, the MMAE algorithm being tested accounts for all hard failure combination scenarios; it is the full, implementable algorithm vs. a "stripped down" version with less than the full complement of level-one bank filters. Partial failures provide an example of the robustness of the MMAE algorithm, to be handled by blending the estimates from the appropriate hard-failure elemental filter and the fully-functional aircraft elemental filter. Chapter 2 provides detailed insight into the philosophy of the "moving bank" structure and questions of interest.

3.8 Dither Signal Design

A dither signal is defined in this thesis as an intentional stimulus imposed on the system with the expressed purpose of improving the MMAE algorithm's ability to identify a failure. A variety of dither signals are explored in this research effort including: sine wave, modified sine wave, square waves, triangle waves, and modified pulse trains. Two philosophies are employed throughout the dither signal process. There exist situations in which one prefers the dither signal to be of a subliminal nature, such as nominal in-flight conditions for long periods of time. Additionally, there exist emergency situations in which a failure must be positively identified. In this scenario, the requirement for a subliminal dither is secondary to the necessity for complete and correct failure characterization. Automated failure identification requires no pilot intervention. By automating the identification process, an excitation signal can be optimized to provide good failure detection performance. To preserve steady state flight conditions and provide good excitation, "pulse train" signals were investigated. The single pulse and inverted pulse combinations, properly tuned, provide good excitation and good performance but have a basic flaw. The signals repeat every 3.0 seconds. Thus a failure could exists

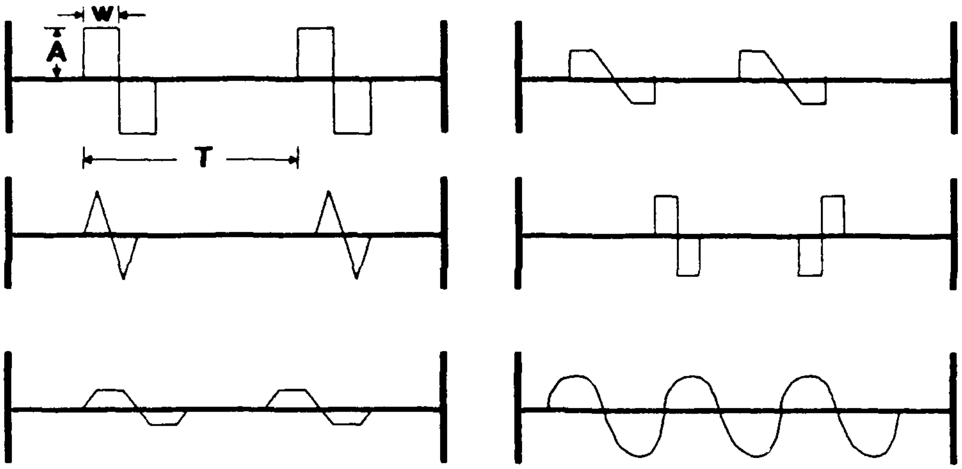

Figure 3.2 Dither Signal Wave Forms

between pulses for up to 3.0 seconds. This may be unacceptable in some flight conditions. In the other extreme, bringing the pulses closer together produces the same effect as a continuous sine wave. Variations between these two signals demonstrate the designer's attempt to minimize disruption of steady state response while still providing sufficient excitation necessary for good algorithm performance. *Figure 3.2* presents the dither signal wave forms investigated in this thesis.

The generation of subliminal dither signals implies some constraints on the magnitude and frequency of acceptable dither signals. In this thesis, intuitive reasoning and previous handling qualities experience set the magnitude of the dither signal. The dither signal is not allowed to produce more than +/- 0.1 g's in the longitudinal axis or 0.2 g's in the lateral directional axis. The frequency requirement is not as clear. In the pulse train, the frequency requirement is dictated by identification characteristics. It did not appear that the frequency for good identification resulted in a noticeable increase in RMS g between pulses. In contrast, the sine wave form did produce changes in the RMS g levels since the wave is continuous. The frequency is selected by good identification criteria, minimal deviations from steady state, and the occurrence of any deviations in state variables that might be considered objectionable by a pilot.

Non-subliminal dither signals could be used in an emergency situation to identify the current system's configuration and capabilities. The criteria for the selection of the dither signal in this application include unconstrained accelerations at the pilot station (up to a reasonable physical limit). The signal frequency and strength are determined by good identification capabilities. These signals would certainly be noticeable to a pilot but will provide excellent identification performance. Chapter 4 provides evidence for the observations listed in this section.

3.9 Step-by-Step Outline

The analysis process is described below in a step-by-step process. Within each step, the appropriate computer codes are identified. This section does not attempt to identify every detail necessary to design and run the MMAE simulation, but rather to give the user an overview of what steps are necessary to execute the design and simulation and the accompanying thought process.

GENERAL: SIMULATION CODES OF INTEREST:

MMAE SIMULATION - This code runs the MMAE computer simulation of the VISTA flight control system. It uses many subroutines (See App. C - Code Description). The VISTA subroutine can be removed and replaced with any other flight control subroutine. Changing the subroutine will require a change to the MMAE code since it is unlikely that the number of sensors and actuators would be equivalent.

STAND ALONE VISTA F-16 FLIGHT CONTROL SYSTEM - This code computes the flight control system's trim conditions for a desired flight condition. The code stores the flight control system's trim values for every internal and external variable of interest. This is a desirable feature since the alternative is to run the MMAE simulation code for an extra period of time to find the trim state. The MMAE simulation code is computationally intensive, and adding additional run time is undesirable.

GENESIS - The GENESIS simulation is a six-degree-of-freedom nonlinear simulation with a full set of VISTA F-16 flight control laws [not included]. The simulation allows for the examination of internal variables within the flight control system. It produces trim states and linearized models. Linearized models are necessary for the implementation of the truth and design models within the MMAE simulation. Implementing the full non-linear data base would be difficult and computationally intensive as well as unnecessary. A reasonable linearized model provides excellent results for the short periods of simulation run time. The simulation contains output options as well. The simulation can produce high quality laser plots of virtually any desired control sequence or any state variable trace with time.

MATRIXx - This code [15] resides at both AFIT and Wright Laboratories. MATRIXx provides the user with the capability to design and manipulate high-order state space models. The elemental filters are constructed using MATRIXx. Additionally, MATRIXx allows for high quality laser plotting of results.

MATRIXx FILTER DESIGN MACROS: KFEVAL - This code calculates the steady state measurement noise strength matrix Q, and state covariance matrix, for the elemental filters. This code was written by Dr Peter Maybeck. Only part of the code is used within the filter creation sequence.

SETUPBX (where X is 1,2,... # of banks) - This code is used as a subroutine. This code sets up the filters in banks.

MATRIXx PLOTTING ROUTINES: GENPLTXX (where XX is A1, A2, A3, A4, A5, S1, S2, S3, S4, S5, S6, or S7) - This code creates hard copies of the plots in MATRIXx. It is executed within MATRIXx and produces probability strip charts.

GENRESID - This routine generates residual plots within MATRIXx and is similar in operation to GENPLTXX.

STEP 1 Using GENESIS or any other applicable code, generate a linearized aerodynamic model of the vehicle of interest. A code that allows the user to generate model responses to control stimuli should be selected. This will allow the user to check his model and control system against the full nonlinear system prior to incorporation into the MMAE simulation.

STEP 2 Design or simulate a control system to couple with the linearized model. The system should be a stand-alone system to check against the full nonlinear system and to generate a trim file for the MMAE simulation controller.

STEP 3 Construct the measurement noise covariance matrix R by evaluating the RMS noise of the control system's sensors.

STEP 4 Compute the strength of the white noise approximations for the Dryden wind disturbance model, Q, by evaluating the average noise strength for u, α, q, p, β, r, and couplings of these variables using PSD plots [8,9,18]. Construct the Q matrix.

STEP 5 Design the filters by running the MATRIXx MACRO FILECREATE.MXX. This routine will construct the filters using the Q matrix in local memory and the R matrix also in local memory. The linearized model must be placed into the local (temporary) memory as well. After the macro executes, the filters can be found in the directory. Step 7 explains the filter naming convention. Perform an "fsave *variable names* filename.dat" in MATRIXx before exiting to save the linearized model, the initial design state and noise covariances and disturbance models to a data set.

STEP 6 The MMAE simulation needs 6 data files to allow proper execution. (1) Data file DECLARR.TXT provides a common block structure used within every routine and subroutine in the MMAE simulation code. A FORTRAN "include DECLARR.TXT" within every routine and subroutine allows a single data file to exist with all the variable declarations. This provides for fast and simple variable additions, deletions, common block structure, and variable declaration changes. (2) Elemental filter data files. (3) Data file REALS.DAT contains all of the real data variable values used within the simulation. (4) Data file FLAGS.DAT groups all of the logical variables into a single file. (5) Data file TRIM DAT contains the trim variables for the flight control system generated by the stand-alone flight control system. (6) NAMES.DAT contains all of the six-digit codes used by subroutine REDMAT.FOR to identify each elemental filter's name and bank number (e.g., F01B1 is interpreted as F̲ilter number 0̲1̲ in B̲ank 1̲.).

STEP 7 Executing the MMAE code requires the user to input the command "RUN MMAESIM". Prior to code execution, the user should select a failure by setting the appropriate variables within the data files and copying the elemental filter selected for a failure to data file F02B1. F01B1, F02B1, and F03B1 are always the truth models after creation from the design macro. The code uses F01B1.dat as the no-failure condition for run intialization and will use this model until the first single failure occurs, upon which the code uses data file F02B1.dat as the system model. If a second failure occurs, data file F03B1.dat is used to specify the appropriate system model. Execution macros exist for both on-line execution and batch execution.

STEP 8 Post-execution options allow the user to evaluate the system performance quickly by looking at standard formatted data files or by transferring the data file MXX.DAT to MATRIXx and plotting using the GENPLTXX routines or the GENPLT MACRO (generates the entire series of plots for all single hard failures).

STEP 9 After executing the MATRIXx plotting routines, the output files must be converted into the appropriate printer/plotter format. MATRIXx supports a variety of output options. This thesis used the LN03 laser printer for plots. The MACROs PLOTTIE and PLOTRESID contain the appropriate commands. Appendix D contains the Matrixx Code. Appendix E contains the data files used for Matrixx and the simulation.

3.10 Summary

This chapter has described the development of the aircraft truth model, the covariance model, the actuator model, the Dryden wind model, and the discrete gust model. The design model development was presented assuming 1st-order actuators and a 0-order wind model. A functional description of the VISTA F-16 flight control system was outlined and the actuator/sensor failure methodology developed. A section on dither signals and the criteria in designing and applying them was included. The chapter concludes with a brief step-by-step general outline of the design and execution process necessary to run the code. Additionally, short descriptions of the codes and their associated names have been included as a reference.

IV. RESULTS

4.1 General

This chapter presents data for single and multiple failures. Single failures include: complete (hard) and partial (soft) actuator failures, and complete (hard) and increased-noise (soft) sensor failures. Multiple failures are generated by selecting two of the single failures. The data is presented for single failures followed by multiple failures. Single failure data is arranged according to failure attributes in a hierarchical order. Hard failures are presented before soft failures. Within the hard failure section, subliminal dither signals are presented before non-subliminal dither signals. The dither signal results are followed by purposeful commands (i.e. stick pull and hold). A subsection demonstrating sinusoidal dither commands, in contrast to pulse repetition dithers, is included. The final subsection illustrates residual monitoring characteristics. Within the soft failure section, subliminal and non-subliminal dithering is presented using the same dither signals evaluated in the hard failure section. In the event that the established dither signals are ineffective in providing good identification performance, a dither signal designed specifically for enhancing the identifiability of the soft failure will demonstrate the algorithm's performance characteristics (whether acceptable or not). The soft actuator failures are modelled by reducing the surface's effectiveness by a percentage value. The soft sensor failures are modelled by increasing the sensor noise variance by a multiple factor. Multiple failures require the creation of additional filter "banks" to allow for two failure scenarios (Chapter 2,3). The first bank (level 0) includes all of the single failure filters. Each of the filters in the second series of banks (level 1) is designed assuming a first failure and any other second failure. As an example, the left stabilator level 1 bank consists of 11 filters designed for a left stabilator failure and any other second failure, a left stabilator failure alone, and a no-failure filter. The no-failure filter allows the algorithm to back up to level 0 if a left stabilator misidentification had occurred.

The data within the sections are presented by displaying the computed p_k probabilities for each of the 13 filters in the filter bank. For single failure scenarios, only one bank is of interest. In some cases, the corresponding state time histories are included to demonstrate relevant characteristics or to establish traits from which to base conclusions. The following filter abbreviations are used: **FF** - fully functional, **A1** - left stabilator failure, **A2** - right stabilator failure, **A3** - left flaperon failure, **A4** - right flaperon failure, **A5** - rudder failure, **S1** - velocity sensor failure, **S2** - angle of attack sensor failure, **S3** - pitch rate failure, **S4** - normal acceleration failure, **S5** - roll rate failure, **S6** - yaw rate failure, and **S7** - lateral acceleration failure. The state variables are represented by fairly standard abbreviations: **theta** - pitch attitude angle in degrees, **u** - velocity in ft/sec, **alpha** - angle of attack in degrees, **q** - pitch rate in deg/sec, **Ancg** - normal acceleration in g's, **phi** - roll angle in degrees, **beta** - sideslip angle in degrees, **p** - roll rate in deg/sec, **r** - yaw rate in deg/sec, and **Aycg** - lateral acceleration in g's. Also of interest in some cases are the residual values. (See chapter 2 for a detailed explanation of residual values.) Seven residuals are used within the Kalman filter calculations: velocity, angle of attack, pitch rate, normal acceleration, roll rate, yaw rate, and lateral acceleration. Throughout the chapter, the text for a section will be presented followed by the appropriate figures.

4.2 Single Hard Failures

Hard failures are characterized by the complete loss of effectiveness of an actuator (and in this application a control surface) or sensor signal. As previously mentioned in Chapter 2, hard actuator failures are modelled by zeroing out a column of the control matrix, $\boldsymbol{B}$. Hard sensor failures are modelled by zeroing out the appropriate row of the $\boldsymbol{H}$ matrix. The data presented within this section encompasses results associated with either dither signals or purposeful commands used to enhance the identification of hard failures. Two "subliminal" dither signals are presented within this section. The author admits that there is the potential debate over the definition of subliminal. In the context presented in this thesis, subliminal is defined in terms of the state parameters, particularly the normal and lateral accelerations. Dither signals which produce accelerations of less than +/- 0.1 g's of normal acceleration and +/- 0.2 g's lateral acceleration are considered subliminal by definition. Subliminal dither signals have a number of potential applications for failure detection and identification applications. If a pilot perceives a change in handling qualities, a nonsubliminal pulsed or continuous dither signal can be used to identify a failure positively. During routine flight (flight in which a failure is not perceived by the pilot), a subliminal dither signal can be used. Subliminal dither signals provide excellent results for most applications. A few difficult situations require a nonsubliminal dither signal for identification (dual failures in which the first failure is an actuator or surface). This effect is discussed later in the chapter. If the pulse is truly subliminal, the system can be engaged during the entire flight to provide continuous protection. This prevents a departure from controlled flight should the pilot attempt a maneuver without being aware of the system's true configuration (i.e. failure status). In such situations, the algorithm will identify the failure prior to the insertion of the command. If not, purposeful commands provide positive identification of failures in most situations. These effects are discussed within this section.

4.2.1 Subliminal Pulsed Dither Signals

The pulsed dither signals presented in this section are pulse trains recurring every 3.0 seconds. The signals are characterized by a positive pulse held for a period of 0.125 seconds followed by a negative pulse held for a period of 0.125 seconds. The pulses occur at 0, 3, and 6 seconds. The pulse durations were determined empirically. The rationale for dither pulse design was to provide a pulse of enough duration to excite the system, yet not too long as to violate the subliminal constraints. The pulse application times were chosen relative to the eight second run time. A pulse at zero starts the identification process immediately and places the probability in the no-failure filter (all of the runs throughout the thesis begin with the probability in the no-failure filter). The pulses at three and six seconds into the simulation are evenly spaced to demonstrate the probability characteristics before and after the failures.

Figure 4.1 displays the p_k probabilities for the bank of 13 elemental filters for a no-failure scenario. This scenario establishes a baseline for comparison with other failure scenarios. Figure 4.2 contains the corresponding states. From Figure 4.1, the algorithm correctly identifies the no-failure filter from the starting probability of 0.75 within 0.25 seconds. The probability grows from 0.75 to 0.988 (maximum) within this time. The excitation of the pulse forced the residuals, corresponding to each of the filters with incorrect hypotheses, to grow large as compared to their internally computed covariances. In contrast, the residuals of the filter with the correct hypothesis fall within the 3σ bounds. The MMAE algorithm assigns probabilities based upon the residuals characteristics. Except for a minor probability drop out at 5.95 seconds, the probability is locked on the fully functional (no-failure) filter. The minor probability dropout occurs near the six second pulse, a time at which ambiguity arises due to the application of the pulse. The pulse excites the system; however, the pulse also provides a transient. The drop out at 5.95 seconds is picked up by the angle of attack failure filter. This phenomena is consistent throughout the hard- failure scenarios. It is an anomaly caused by a 3σ noise spike which violates the 3σ bounds representing the internally computed covariance and causes a momentary loss of probability, which happens to be picked up by the angle of attack failure filter. Note the probability spike is momentary, consistent between cases, and the algorithm instantly returns the probability to the correct filter, namely the fully functional aircraft filter. From Figure 4.2, the pulse dither signal can clearly be observed at 0, 3, and 6 seconds. From the acceleration values within Figure 4.2, the dither meets the subliminal criteria discussed in Section 4.2.1. In Figure 4.2, all of the variables are centered around zero with the exception of the normal acceleration (centered at 1 g). These variables are relative to the trim conditions. The trim angle of attack is 10.06 degrees, as is the trim pitch angle. The trim velocity is 414.8 ft/sec. All other trim states relative to Figure 4.2 are zero with the exception of the trim normal acceleration which is 1g. The reader may be disturbed by the apparent ramping of many of the state variables after the application of each successive dither pulse. A small change in the pulse amplitude can alter the ramping with no change in performance, and the ramping was easily controlled. Since the emphasis of this thesis is not perfect dither signals and the ramping is easily controlled, the author didn't invest any additional time in "perfecting" the dither signal to eliminate the small ramping effect.

Figure 4.3 demonstrates a left stabilator (A1) failure. The failure is introduced at 3.0 seconds. The failure is introduced at this time to coincide with the pulse application. This should produce the best algorithm performance for this pulse. Inserting the failure before the pulse application would result in an undetected failure until the application of the pulse. Inserting the failure after the pulse, and resulting transient, will result in an undetected failure until the next pulse application. At 3.15 seconds, the algorithm correctly identifies the left stabilator as the failure (filter A1). At 3.8 seconds, the rudder, the right stabilator, the fully functional aircraft, and the yaw rate sensor elemental filters share the probability. This indicates four filters with relatively equivalent residuals and a left stabilator with a poor residual value. At 4.0 seconds, the right stabilator is misidentified as the failure. At 4.4 seconds, the left stabilator is correctly and positively identified. Ambiguities between left and right surfaces are common throughout the results. Whether left or right, a stabilator failure was correctly identified throughout the simulation. Moreover, left and right ambiguities can be resolved with purposeful commands. Also, ambiguities can be resolved through residual monitoring which will be demonstrated in Section 4.2.1.4. For the VISTA F-16, left and right stabilators are used for both roll and pitch. Based upon the dither signals characteristics (pulse polarities, strengths, and relative timing), the left and right stabilator command magnitudes may be different. They will certainly be small (due to the subliminal dither criteria). Upon initiating a failure in the truth model, one of the signals will be assigned a value of zero in the truth model. This leaves a small signal to "shake up" the system. Often the signal is not of sufficient strength to provide essentially instant failure detection and identification. Practical application would require an increased magnitude dither signal on the remaining surface to reestablish the desired system excitation. Ambiguities are usually the result of an insufficient excitation of the system or a delay in proper identification due to shaking up a system too much. The first point we've already discussed. Excessive system excitation can result in either ambiguities or a delay in identification. When the transients, generated by a pulse, approach steady state, failure identification occurs. The pitch rate in Figure 4.4 demonstrates the reduction in the system excitation caused by the induction of a left stabilator failure.

Comparison of Figures 4.5 and 4.3 demonstrate the differences between "sister" actuator failures. The right stabilator (A2) failure scenario displays some of the same characteristics as the left stabilator failure scenario. Ambiguities exist between the two surfaces. At 3.1 seconds, the right stabilator is identified as the correct failure. However, at 3.93 seconds, the probability is shared between the left flaperon (A3), the rudder (A5), the right flaperon (A4), and the left stabilator (A1). The left and right stabilator compete for the probability from 4.0 to 4.5 seconds. At 4.5 seconds, the right stabilator failure is correctly and positively identified. Here some ambiguity exists primarily between the left and right stabilator actuators. The differences between the left and right failure scenarios can be attributed to the dither signal characteristics and the flight control system. Comparison of the pitch rate signals between Figure 4.6 and Figure 4.4 highlights the differences between dither signals generated under different failure scenarios. Even with the relatively small signal strengths, pitch rate signal characteristics are obvious. A non-subliminal dither signal or a purposeful command would be required to provide positive identification without ambiguity (See Appendix F).

Figure 4.7 displays a left flaperon (A3) failure induced at 3.0 seconds. The failure is detected and identified within 0.15 seconds. A drop out in the probability occurs at 4.2 to 4.7 seconds. During this time period, the yaw rate failure filter picks up the probability. Again, a slightly larger lateral dither pulse would provide better results and reduce or eliminate the ambiguities. Figure 4.8 displays the state values for this failure. The lateral acceleration demonstrates the effect of the failure upon the excitation of the system (see Figure 4.2 for comparison).

Figure 4.9 presents a right flaperon (A4) failure induced at 3.0 seconds. The right flaperon failure is detected and identified within 0.15 seconds. A drop in probability occurs at 4.2 to 4.7 seconds. During this time period, the yaw rate failure filter picks up the probability. As with the left flaperon failure, a larger lateral dither pulse would provide better results in reducing or eliminating the ambiguities. It is of no surprise that the left and right flaperon failures exhibit similar probability characteristics. The flaperons provide roll control. The states are not included since they are similar to the left flaperon scenario.

The rudder (A5) failure scenario is presented by Figures 4.10 and 4.11. From Figure 4.10, the rudder failure is induced at 3.0 seconds and the inappropriateness of the fully functional (A4) aircraft hypothesis is initially detected at 3.15 seconds. The rudder failure is strongly identified at 3.35 seconds. A small drop out unrelated to the true failure occurs at 5.9 seconds and is picked up the by angle of attack failure filter. This phenomenon was discussed earlier, and it is corrected by the onset of the next dither pulse at 6.0 seconds. Figure 4.11 displays the states for the rudder failure scenario.

Figure 4.12 demonstrates the detection and identification of a velocity sensor (S1) failure. The failure is detected essentially instantaneously at 3.01 seconds. Figure 4.13 demonstrates the detection and identification of an angle of attack sensor (S2) failure; the failure is detected and identified at 3.04 seconds. The detection and identification of a pitch rate sensor (S3) failure is presented in Figure 4.14; the failure is induced at 3.0 seconds and detected and identified at 3.18 seconds. Figure 4.15 displays the detection and identification of a normal acceleration sensor (S4) failure. Here, the failure is induced at 3.0 seconds and identified at 3.02 seconds. Figure 4.16 demonstrates a roll rate sensor (S5) failure, and the failure is detected and isolated at 3.2 and 3.26 seconds respectively. A larger number of dropouts occur for this failure scenario. The largest spike occurs just before 6 seconds, as noted previously. Figure 4.17 presents a yaw rate sensor (S6) failure. The failure is induced at 3.0 seconds and detected and identified at 4.6 and 4.7 seconds. The yaw rate sensor demonstrates different characteristics than the other sensors in that the there is a slow buildup of p_k. This is due to a bias in the residual which gradually decays when the true failure condition is a yaw rate sensor failure.

A proposed solution to this phenomena is discussed later in the chapter. Also, a large spike occurs just before the pulse at 6 seconds, as noted previously. Figure 4.18 presents the lateral acceleration failure. The failure is detected and identified at 3.01 seconds.

In general, the sensor failures display rapid and good identification characteristics. The probability traces are clean and exhibit very little ambiguity when compared to actuator failures. The actuators tend to demonstrate more ambiguity, especially between left and right similar surfaces (stabilators or flaperons), and probability convergence is always slower. The convergence properties of the two classes of failures is related to their residual characteristics. The sensor failures appear directly and instantaneously on single scalar residuals, while actuator failures appear through a number of scalar residuals, and only after the actuator failure effects become evident through the system dynamics (with inherent delays).

A second subliminal dither signal is included within this section to demonstrate the relationship between dither signal characteristics (signal polarity, strength, and relative timing) and failure detection characteristics. The relative timings between the two dithers are identical. The modification to the dither signal occurs in the signal strengths. The longitudinal pulse strength was reduced by 2 lbs while the lateral and directional pulse strengths were increased by 0.5 lbs and 2 lbs respectively. Previous experience with the dither pulses motivated the change. Often when ambiguity is present between a left and right surface, a small increase in the lateral and directional pulse strengths can help resolve the ambiguity. Too large an increase will result in a longer delay period before proper identification. The reduction in the longitudinal pulse prevents any violation of the subliminal dither signal criteria. Also, the longitudinal pulse strength reduction from very large values reduced the amount of system excitation and resulted in cleaner probability time histories. Figure 4.19 establishes the fully functional (no-failure) filter. Note the probability traces are very similar to the first subliminal dither scenario. Figure 4.20 displays the corresponding state variables. Of particular interest is the magnitude of the dither pulses reflected in the lateral acceleration, as compared to Figure 4.2. The first subliminal dither had lateral acceleration values between +/- 0.1 g's. The lateral acceleration from Figure 4.20 is between +0.1 and -0.16. The rationale for the increases in lateral pulse strength will become obvious.

Figure 4.21 presents the data for a left stabilator (A1) failure with the second subliminal dither signal. It is similar to Figure 4.3. The failure is initially detected and identified at 3.15 seconds. Ambiguities exist between the left and right stabilator from 4.0 to 4.7 seconds. The left stabilator failure is detected and identified at 4.7 seconds.

Figure 4.22 displays the right stabilator (A2) failure. The failure is induced at 3.0 seconds and detected and identified at 3.18 seconds. Comparison with Figure 4.5 indicates significantly improved performance.

Figure 4.23 presents the left flaperon (A3) failure. The ambiguity between the left flaperon and the yaw rate failure filters in Figure 4.7 is eliminated. Ambiguity is usually caused by insufficient system excitation. By increasing the lateral excitation, the ambiguity is eliminated.

Figure 4.24 presents the right flaperon (A4) failure. Again, the ambiguity is eliminated.

The rudder (A5) failure is nearly identical to Figure 4.10. It is not included.

All of the sensor failures are nearly identical to those presented in Figures 4.12 to 4.20 with the exception of the yaw rate failure. Figure 4.25 presents the yaw rate (S6) failure. The failure is detected and identified at 4.25 seconds, as compared to 4.6 sec and 4.7 sec, respectively, for the first subliminal dither signal presented in Figure 4.17.

The comparison between the two subliminal dither signals is designed to demonstrate differences in dither signals. A dither signal can be designed to provide good performance for any failure scenario. The second dither signal was designed to provide good failure identification for all failures and excellent identification for the flaperon actuator failures. A combination of pulses designed to provide excellent performance identification for each actuator could be utilized in a failure identification mode. The pilot could select this mode and positively identify the failure.

4.2.2 Non-Subliminal Dither Signals

Non-subliminal dither signals are considered purposeful commands within this thesis. Appendix F includes a paper by Menke and Maybeck which provides a direct comparison between subliminal and non-subliminal dither signals. The results indicate failures can be positively identified if no restrictions are placed on the dither strength. This is basically a purposeful command.

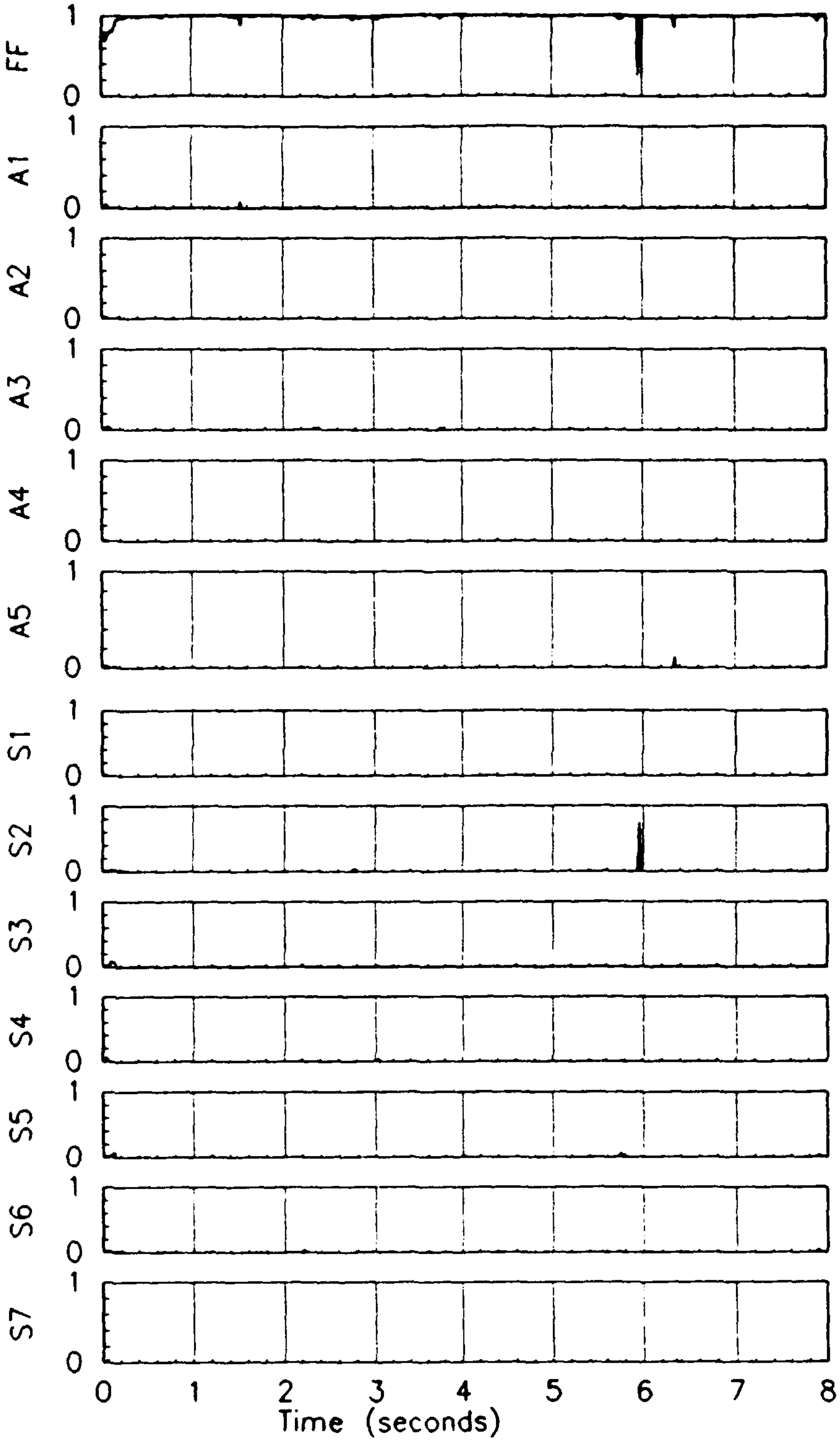

Figure 4.1 Probabilities for no failure scenario using subliminal dither signal 1

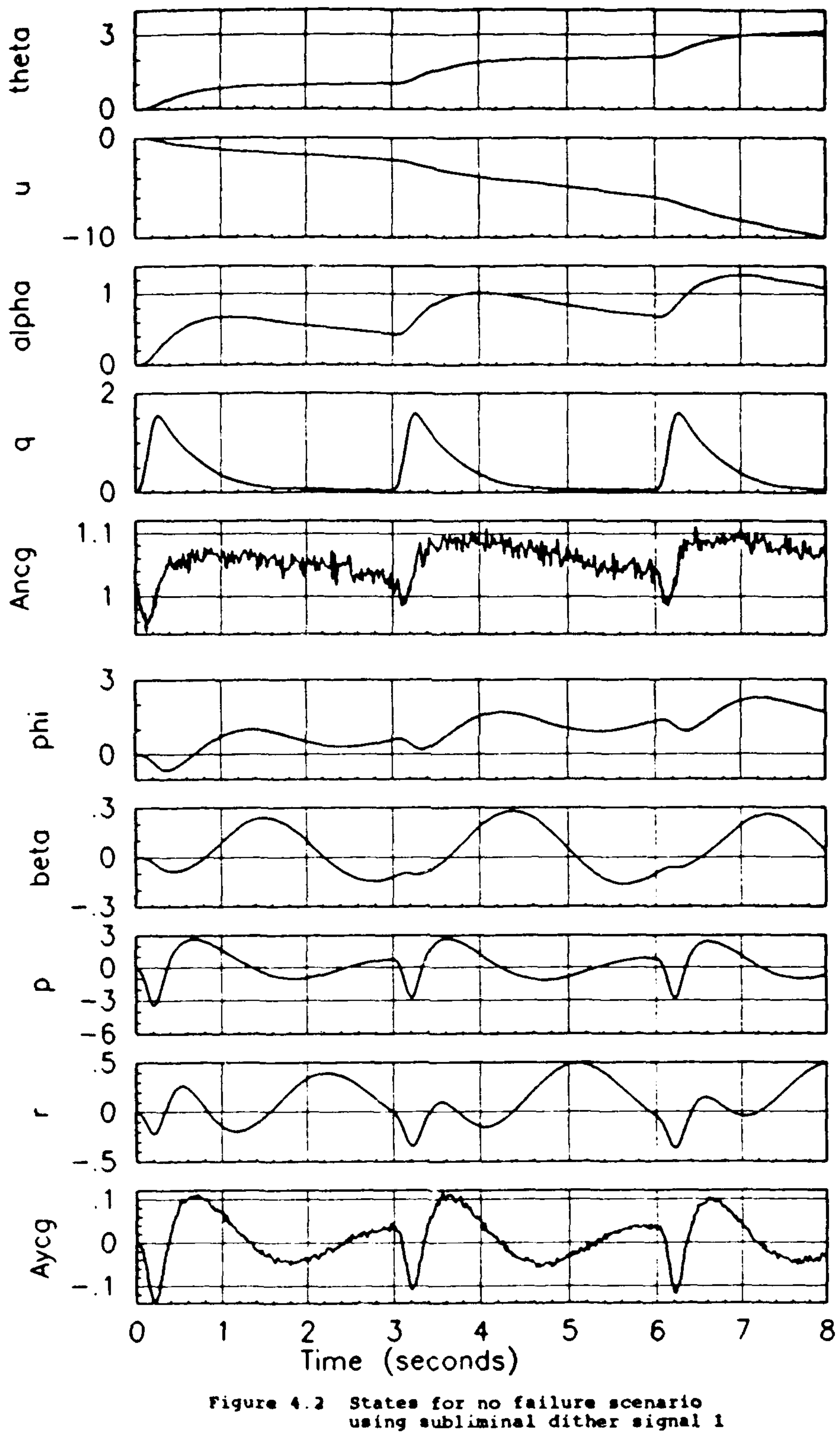

Figure 4.2 States for no failure scenario using subliminal dither signal 1

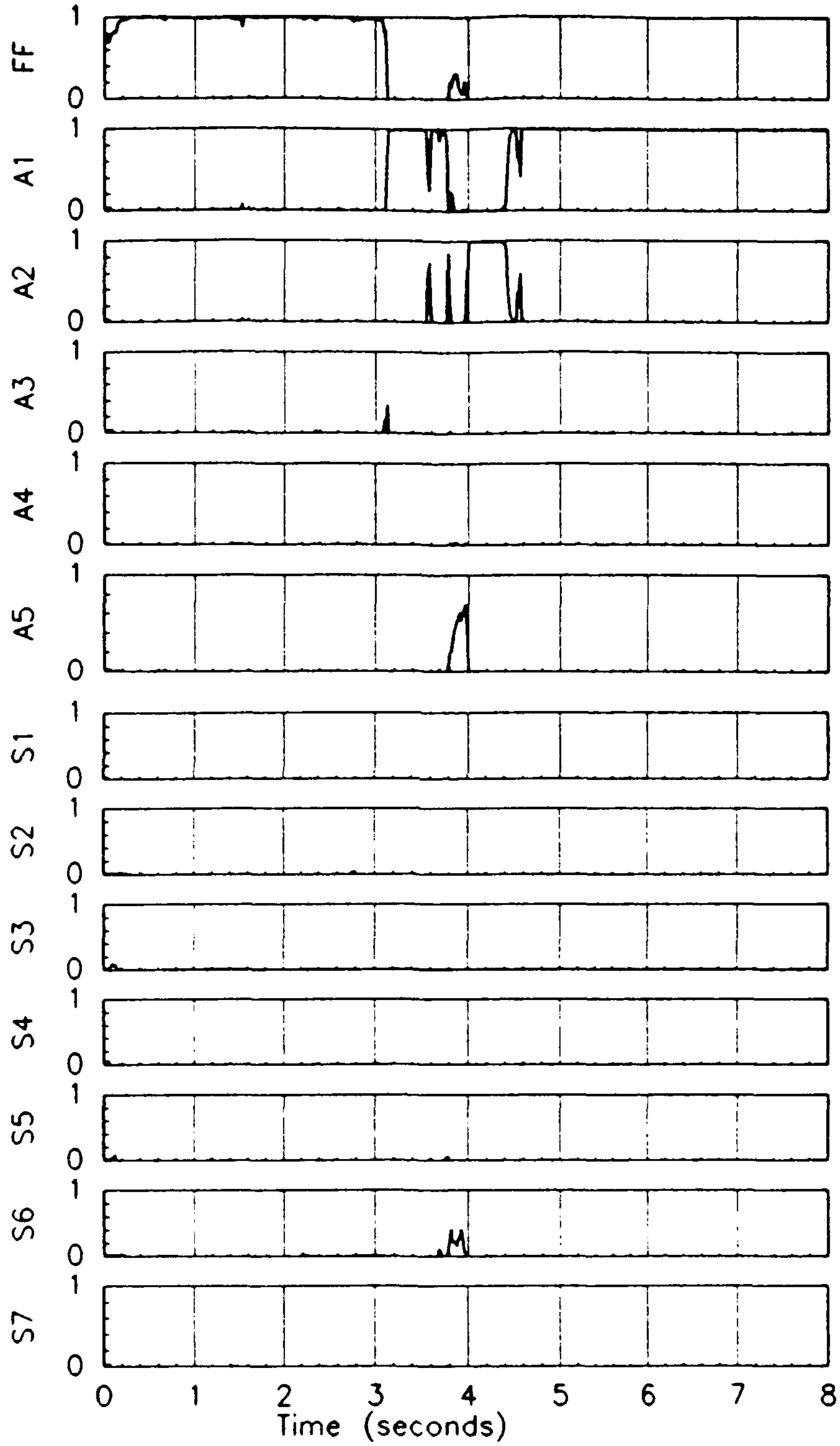

Figure 4.3 Probabilities for a left stabilator failure using subliminal dither signal 1

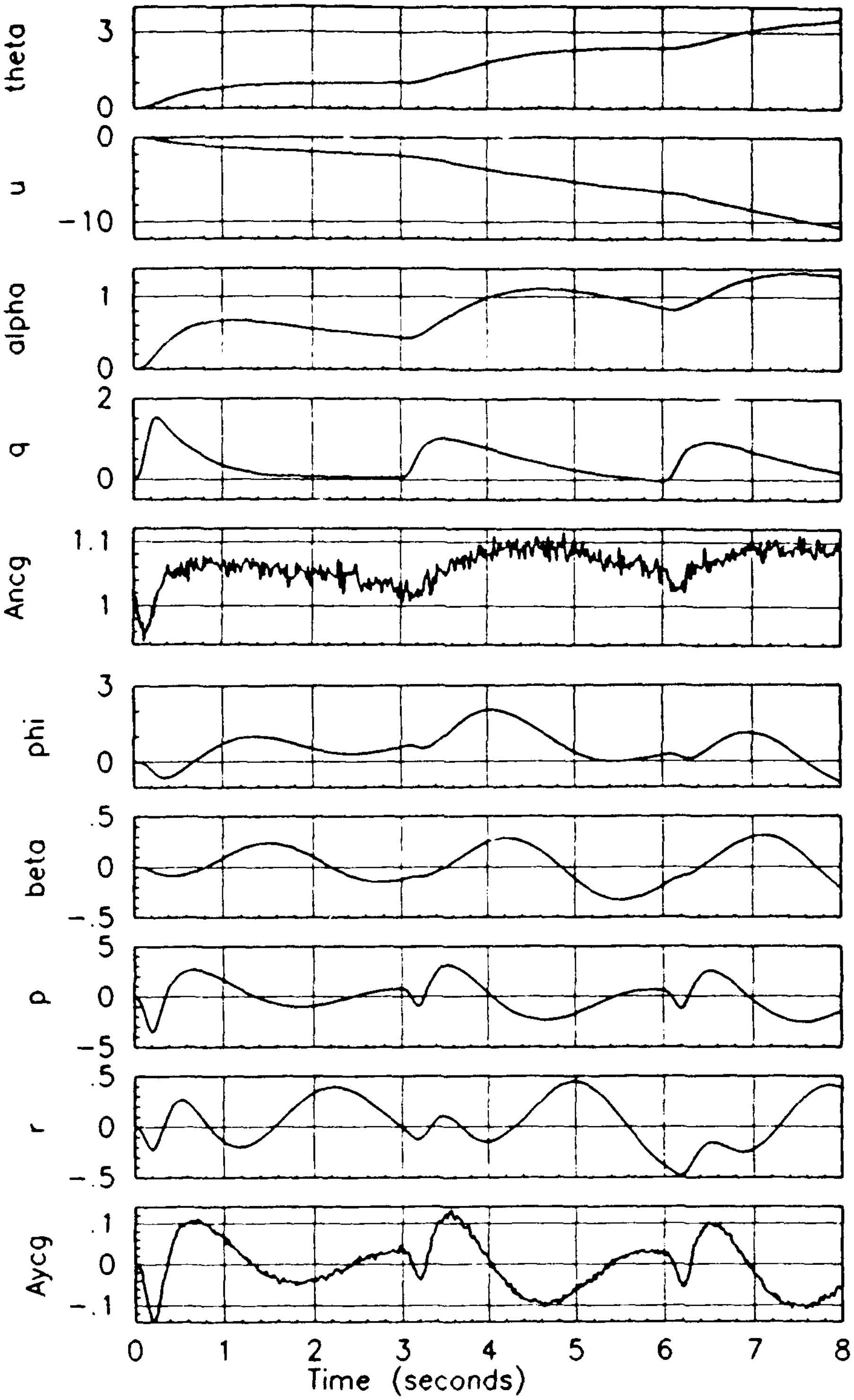

Figure 4.4 States for a left stabilator failure using subliminal dither signal 1

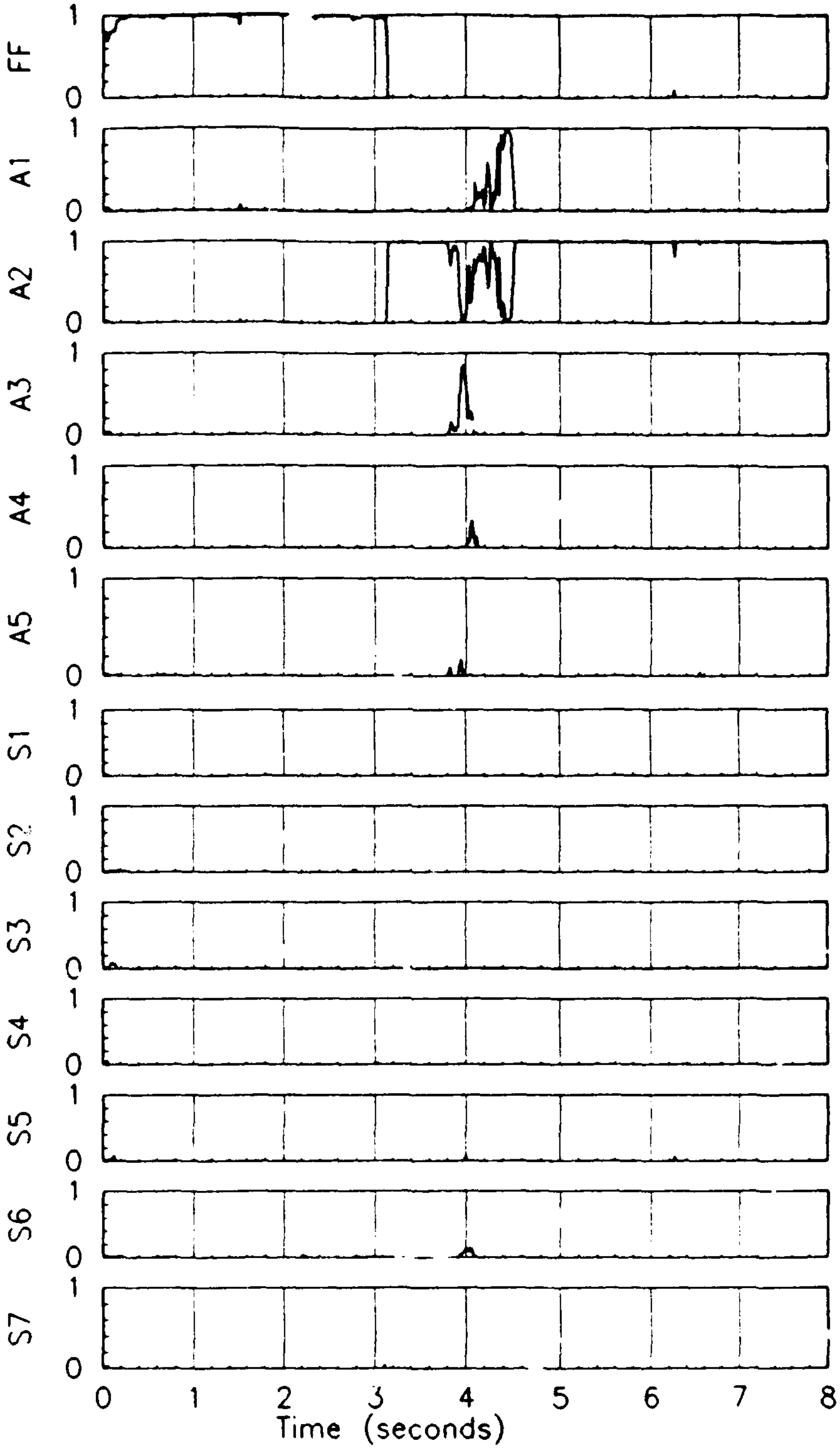

Figure 4.5 Probabilities for a right stabilator failure using subliminal dither signal 1

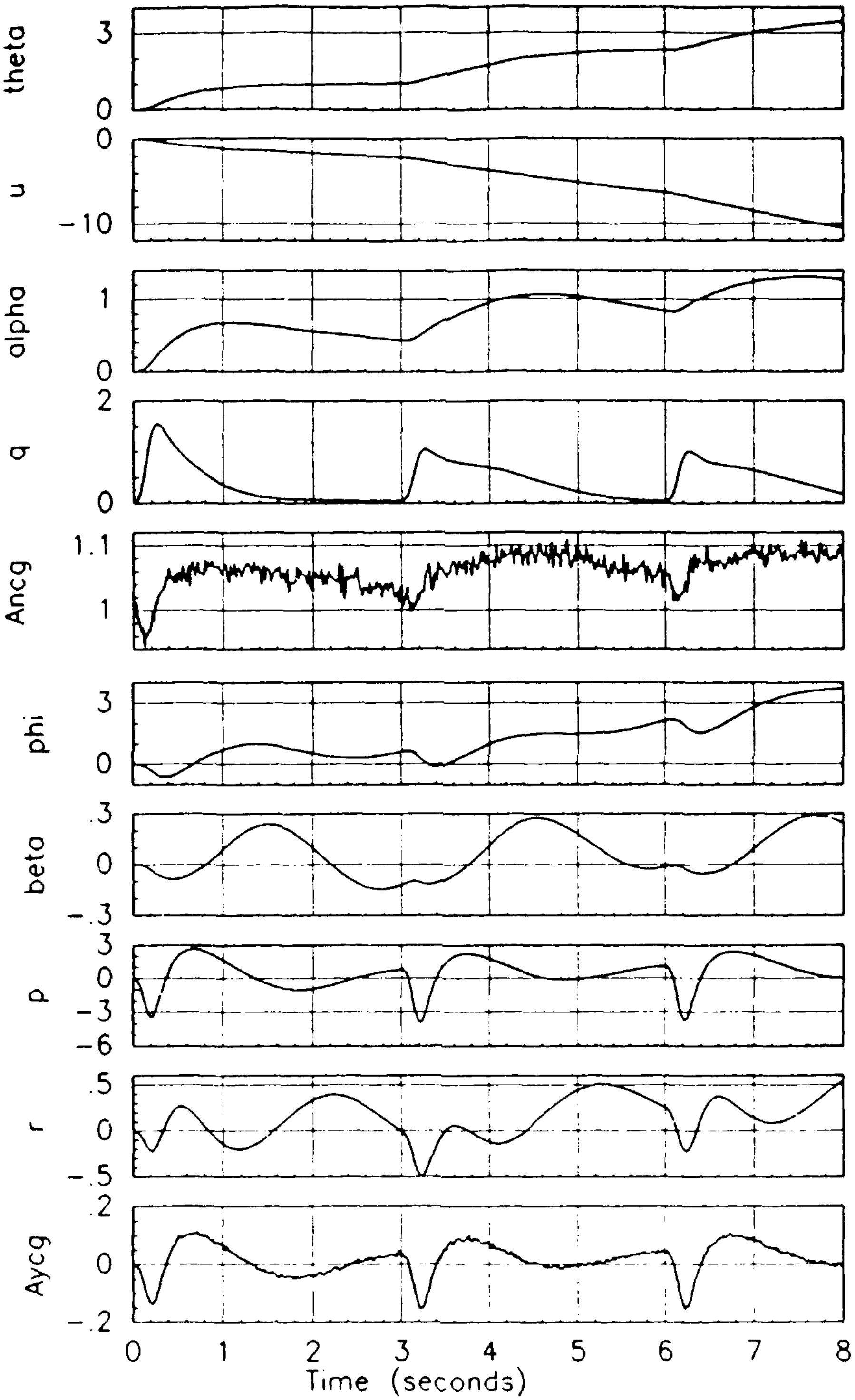

Figure 4.6 States for a right stabilator failure using subliminal dither signal 1

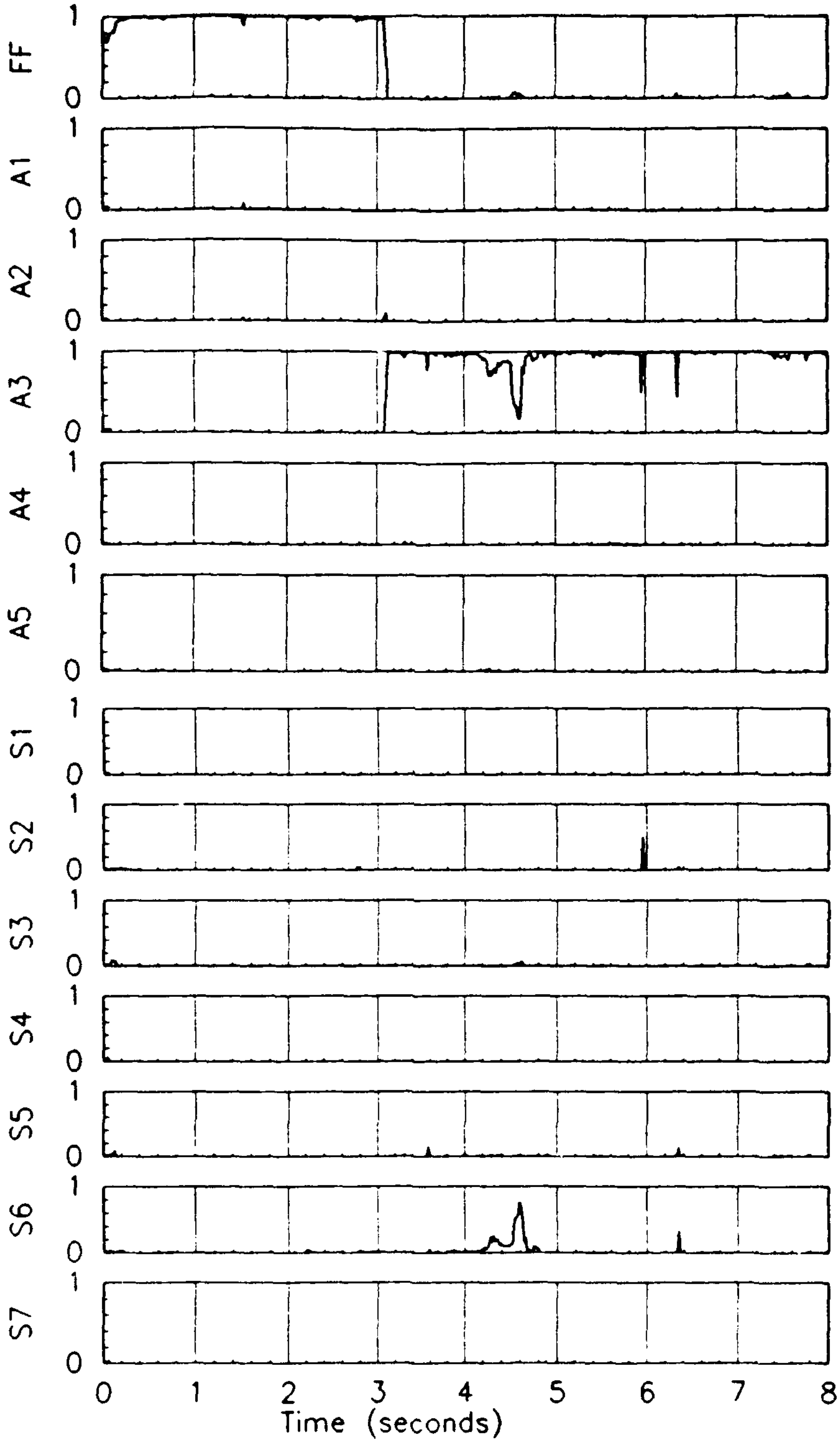

Figure 4.7 Probabilities for a left flaperon failure using a subliminal dither signal 1

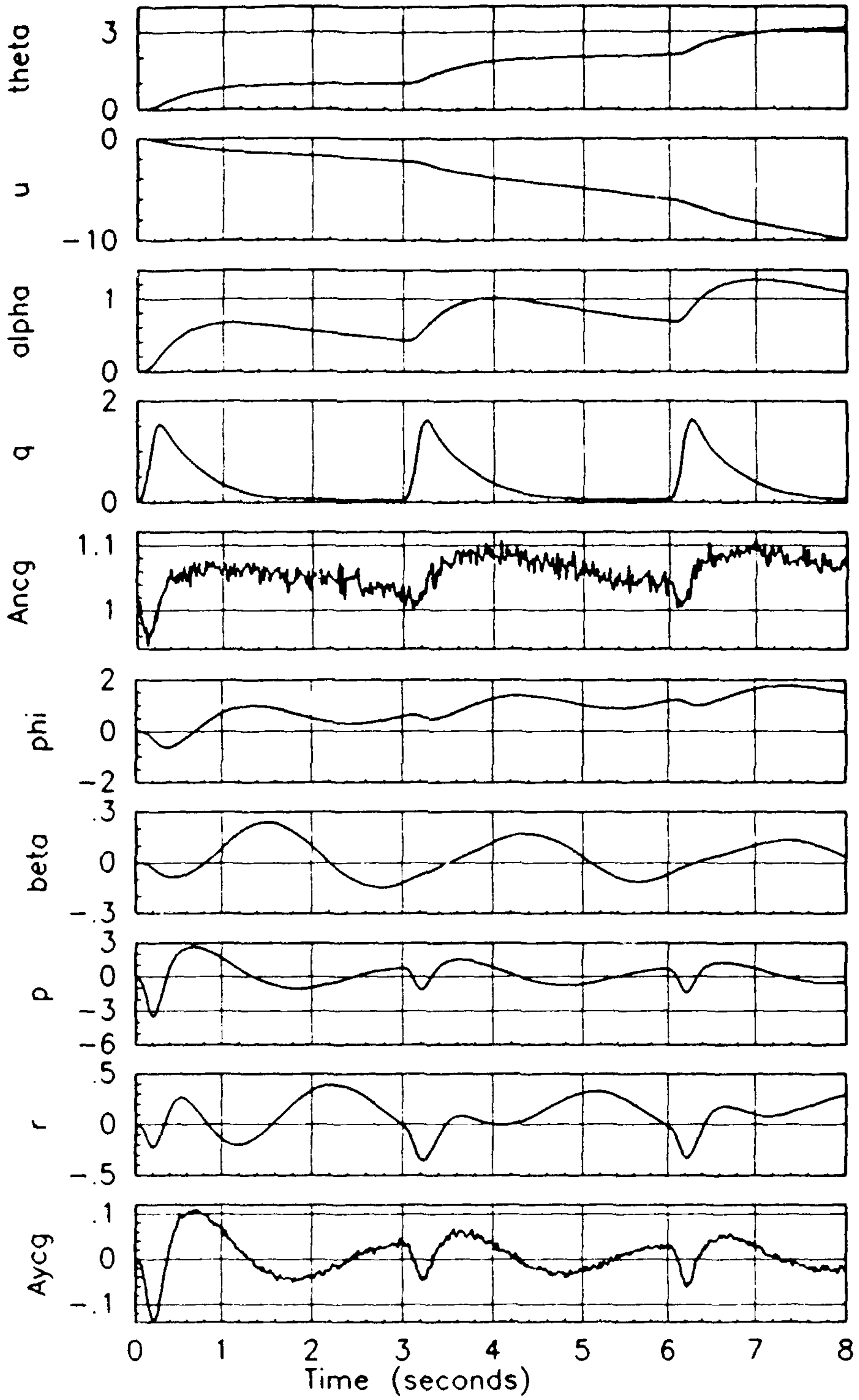

Figure 4.8 States for a left flaperon failure using a subliminal dither signal 1

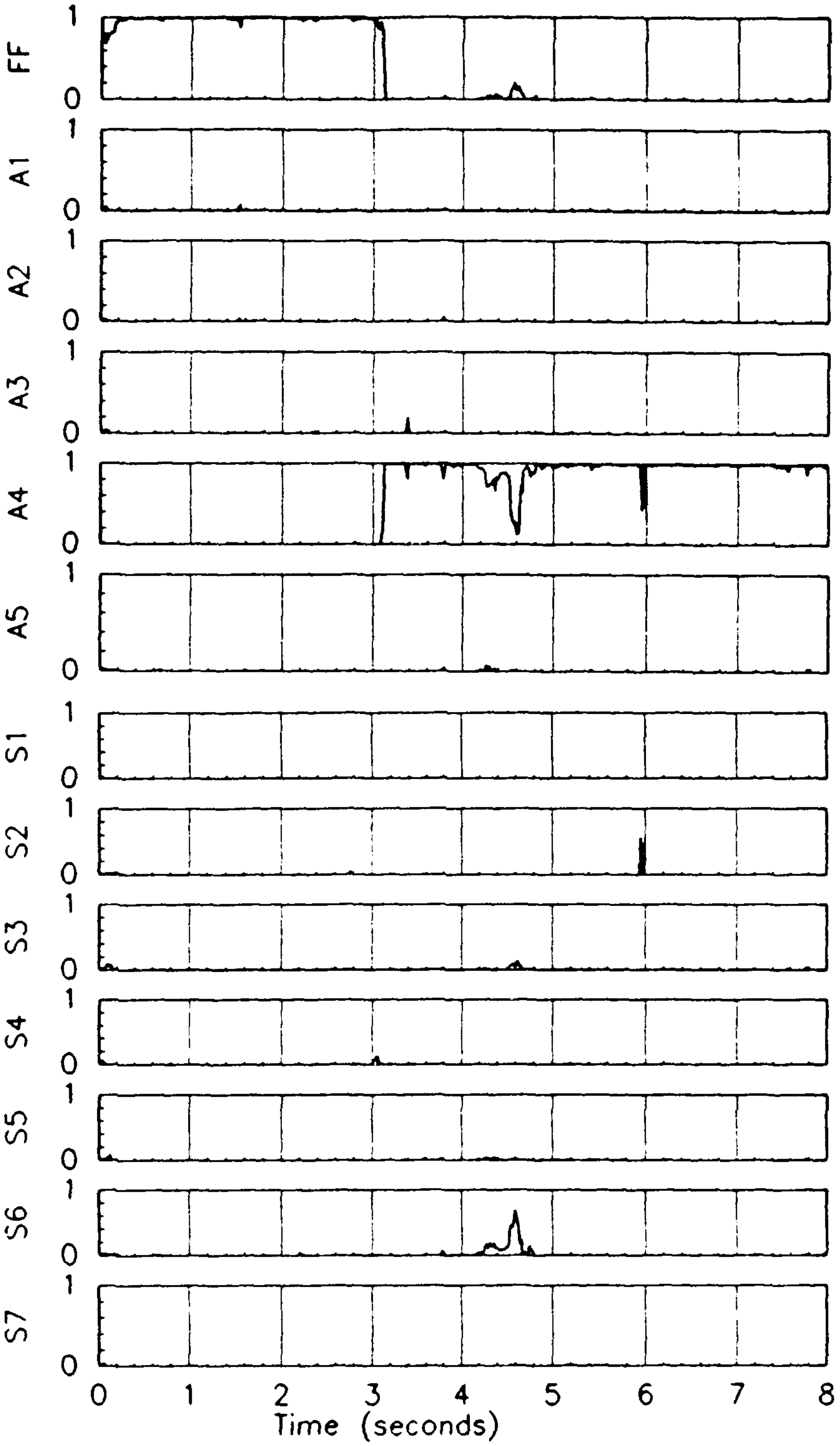

Figure 4.9 Probabilities for a right flaperon failure using a subliminal dither signal 1

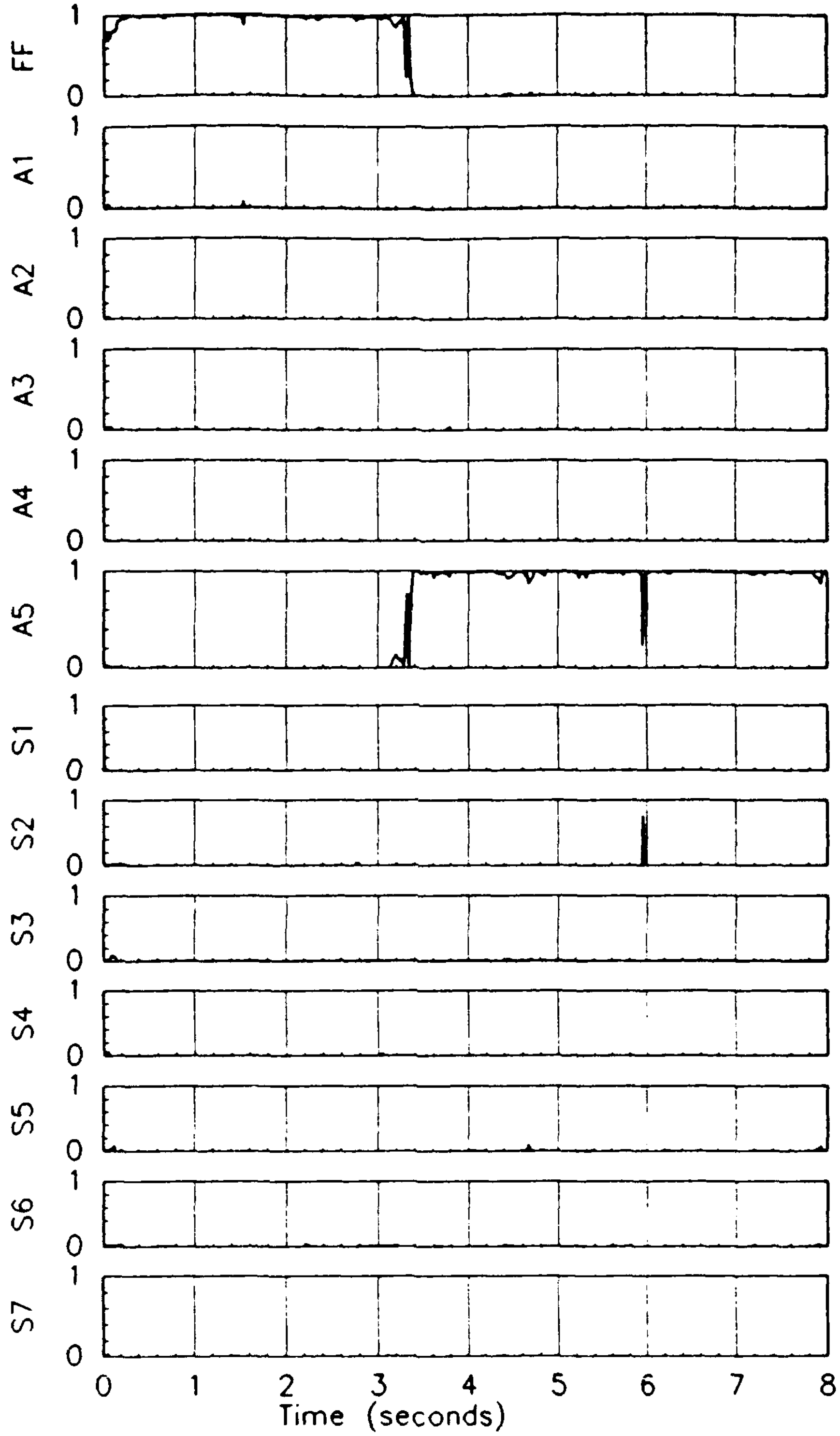

Figure 4.10 Probabilities for a rudder failure using a subliminal dither signal 1

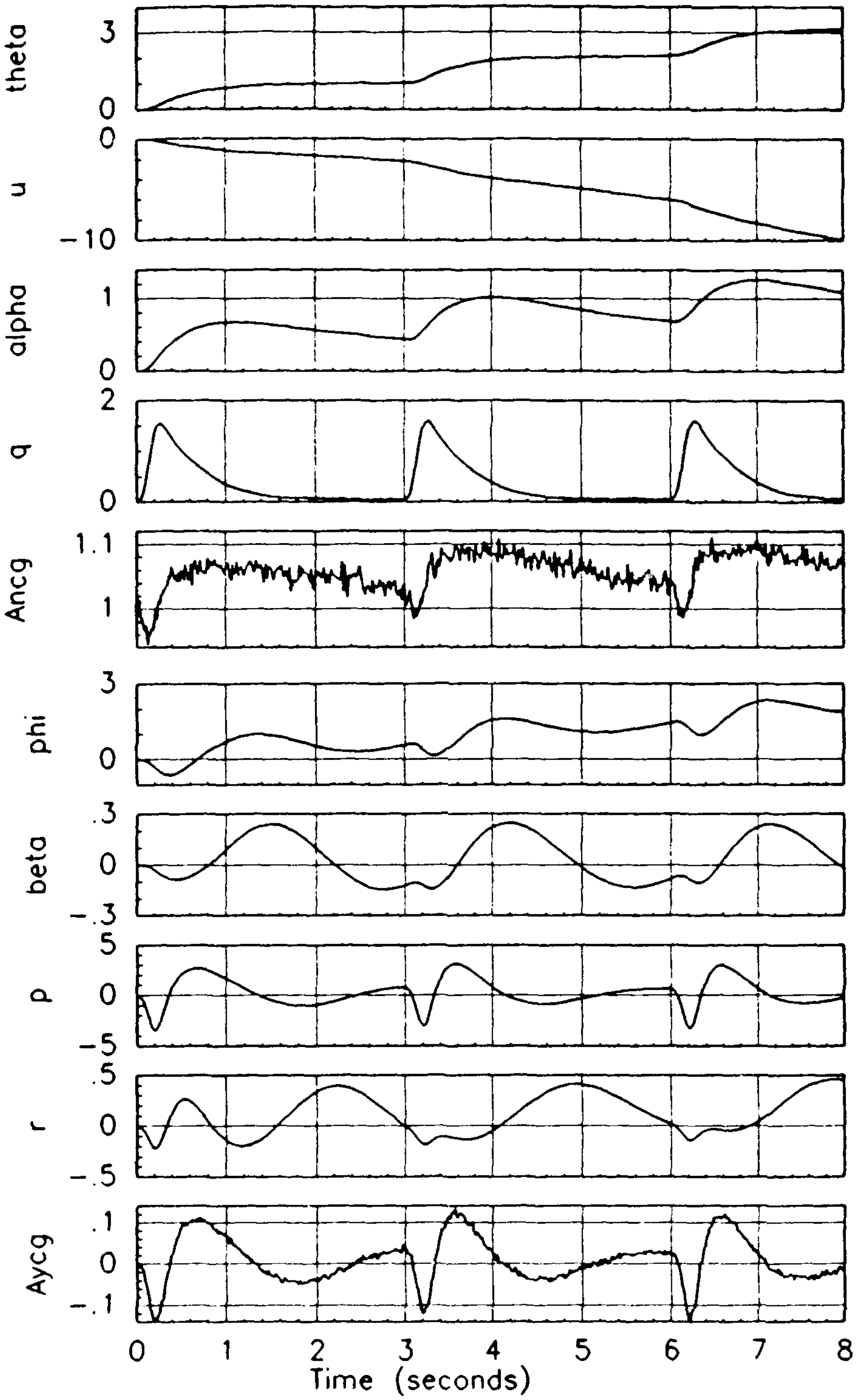

Figure 4.11 States for a rudder failure using a subliminal dither signal 1

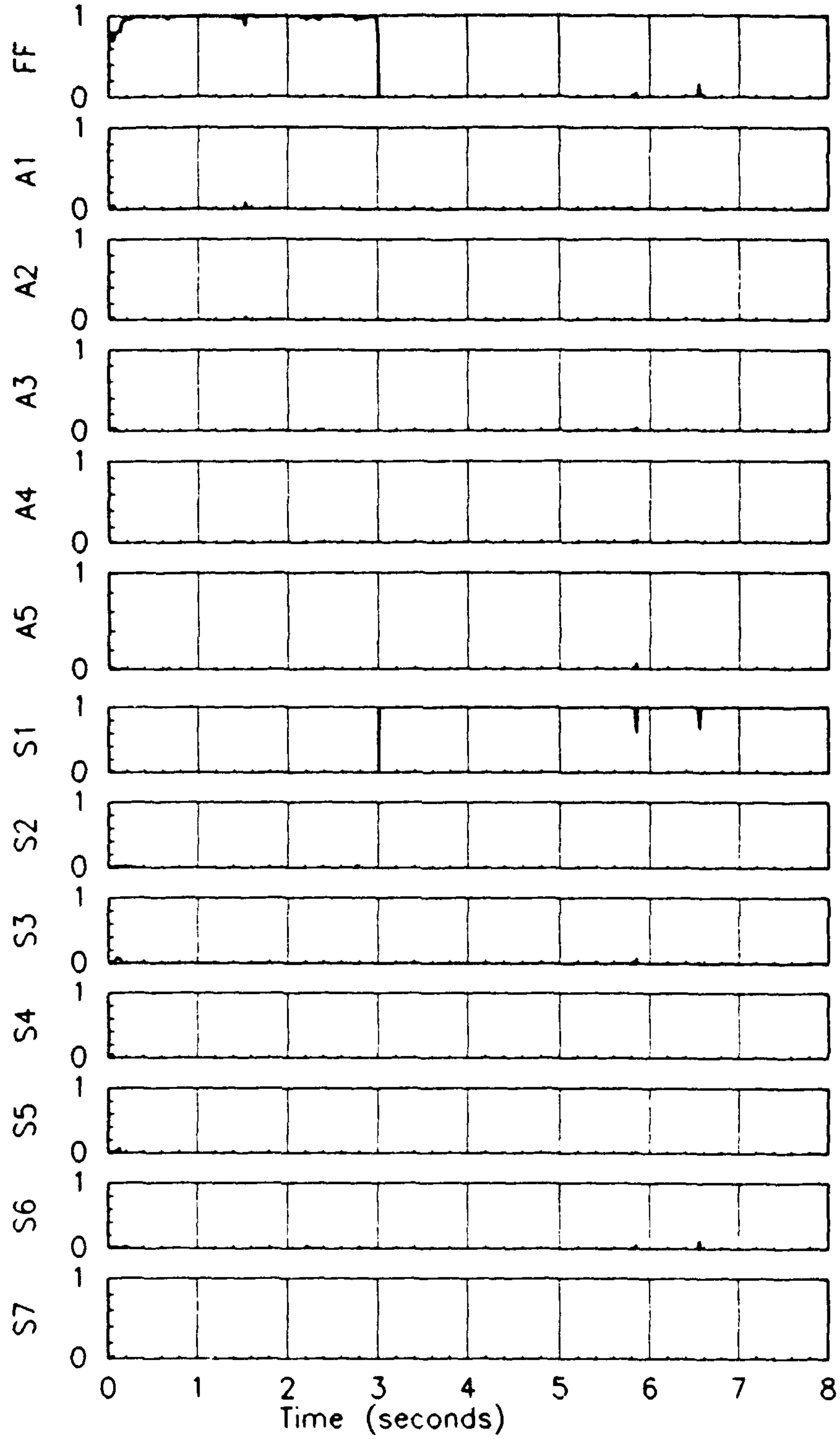

Figure 4.12 Probabilities for a velocity sensor failure using a subliminal dither signal 1

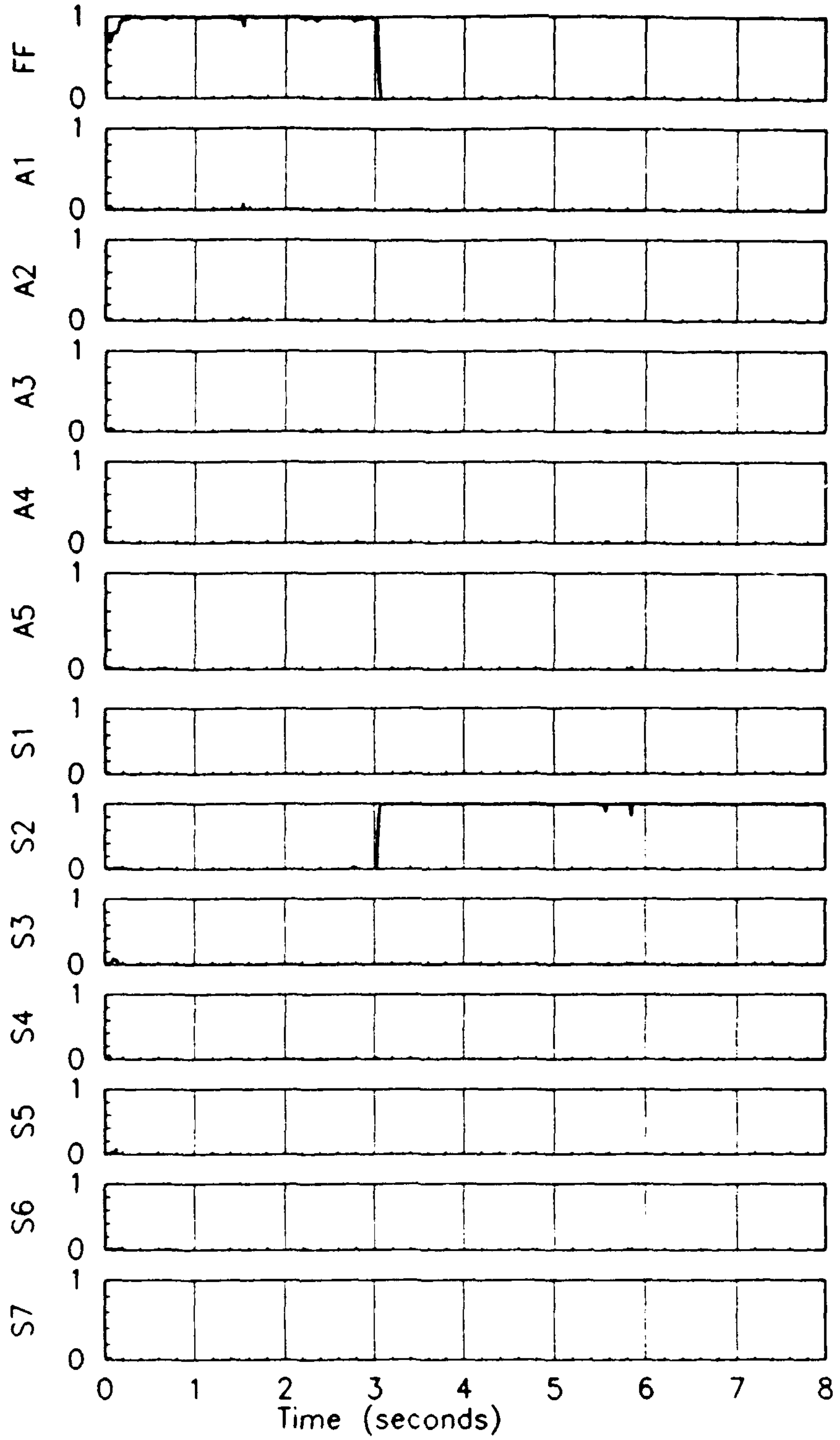

Figure 4.13 Probabilities for an angle of attack sensor failure using a subliminal dither signal 1

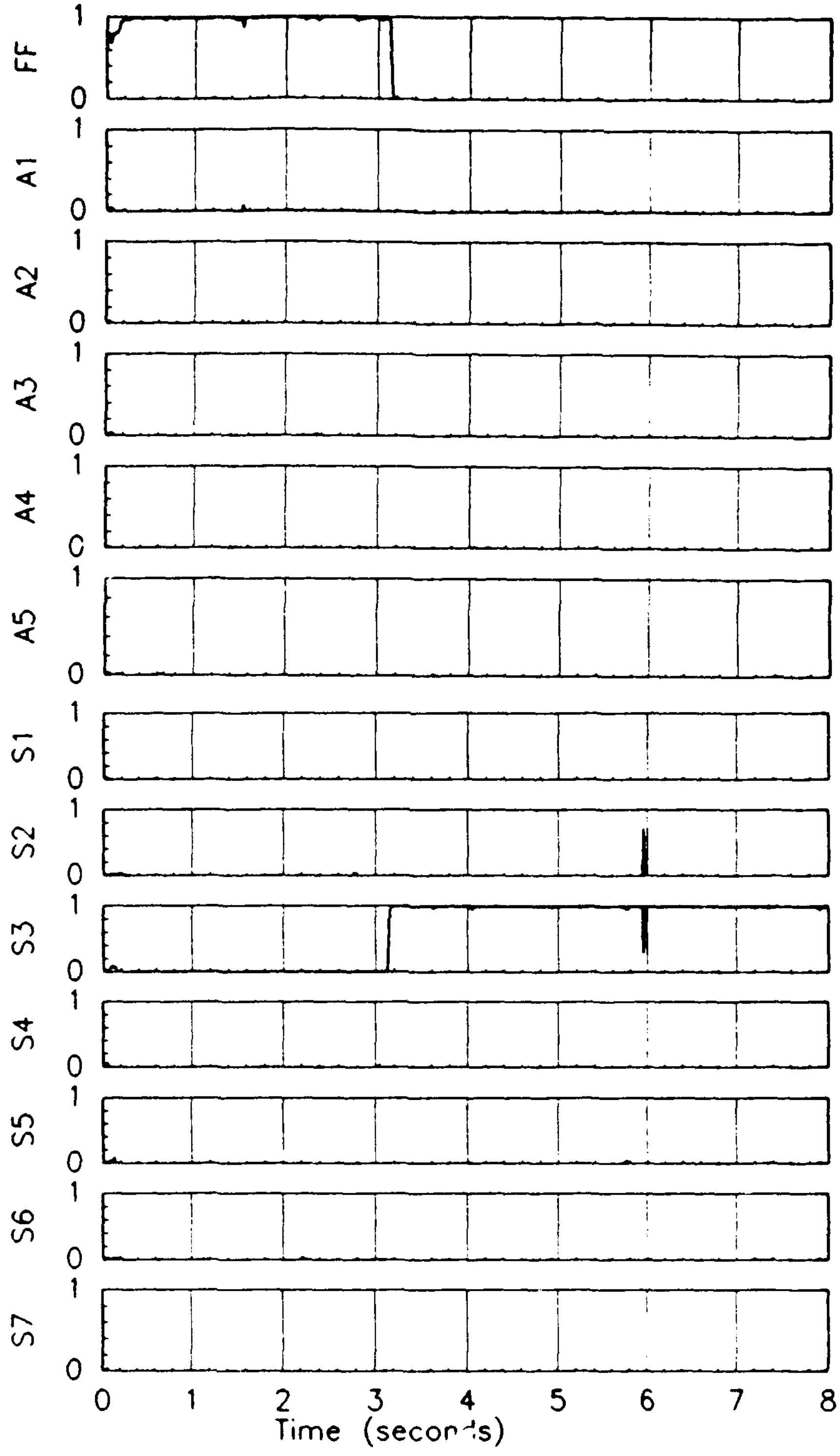

Figure 4.14 Probabilities for a pitch rate sensor failure using a subliminal dither signal 1

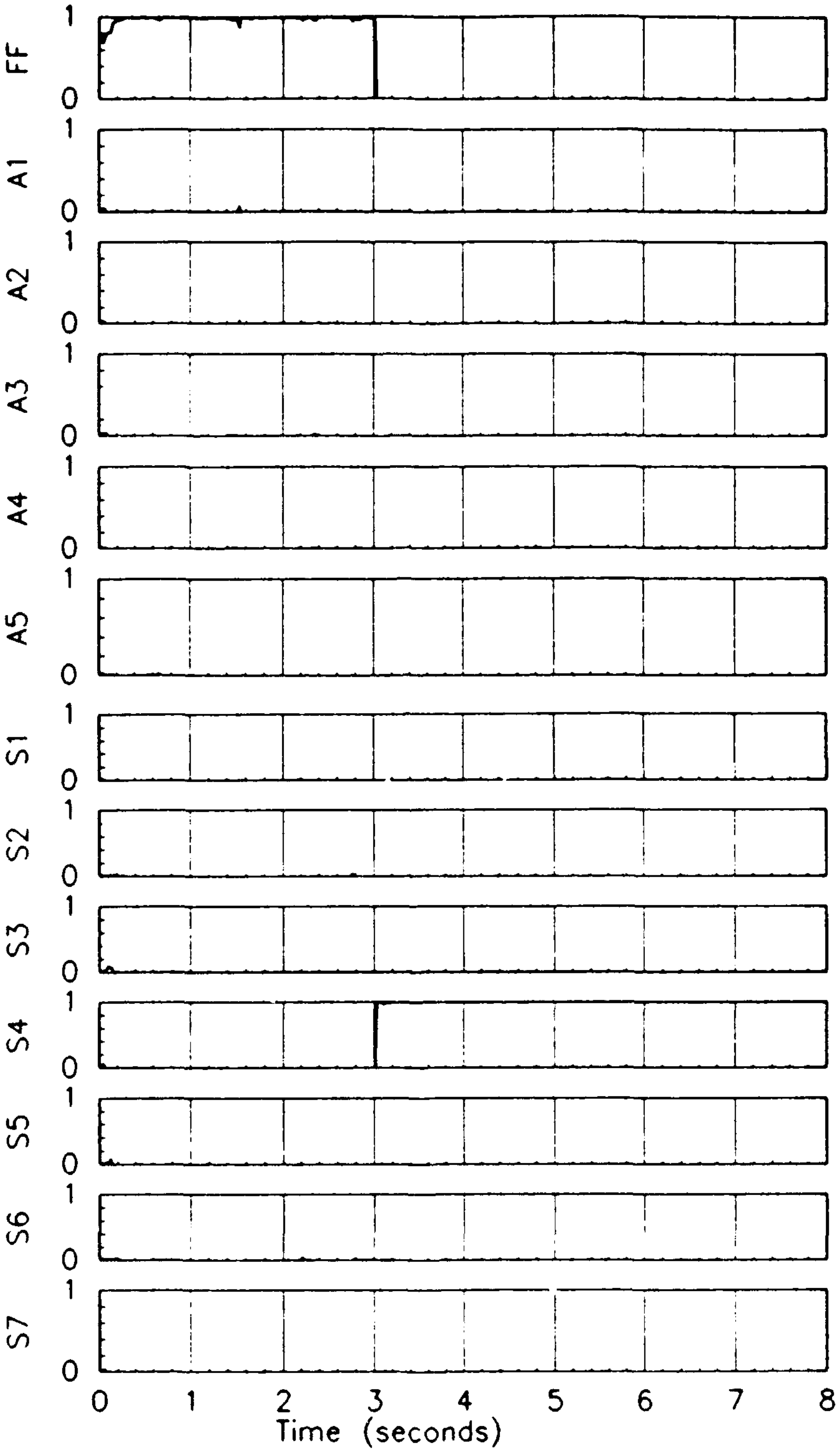

Figure 4.15 Probabilities for a normal acceleration sensor failure using a subliminal dither signal 1

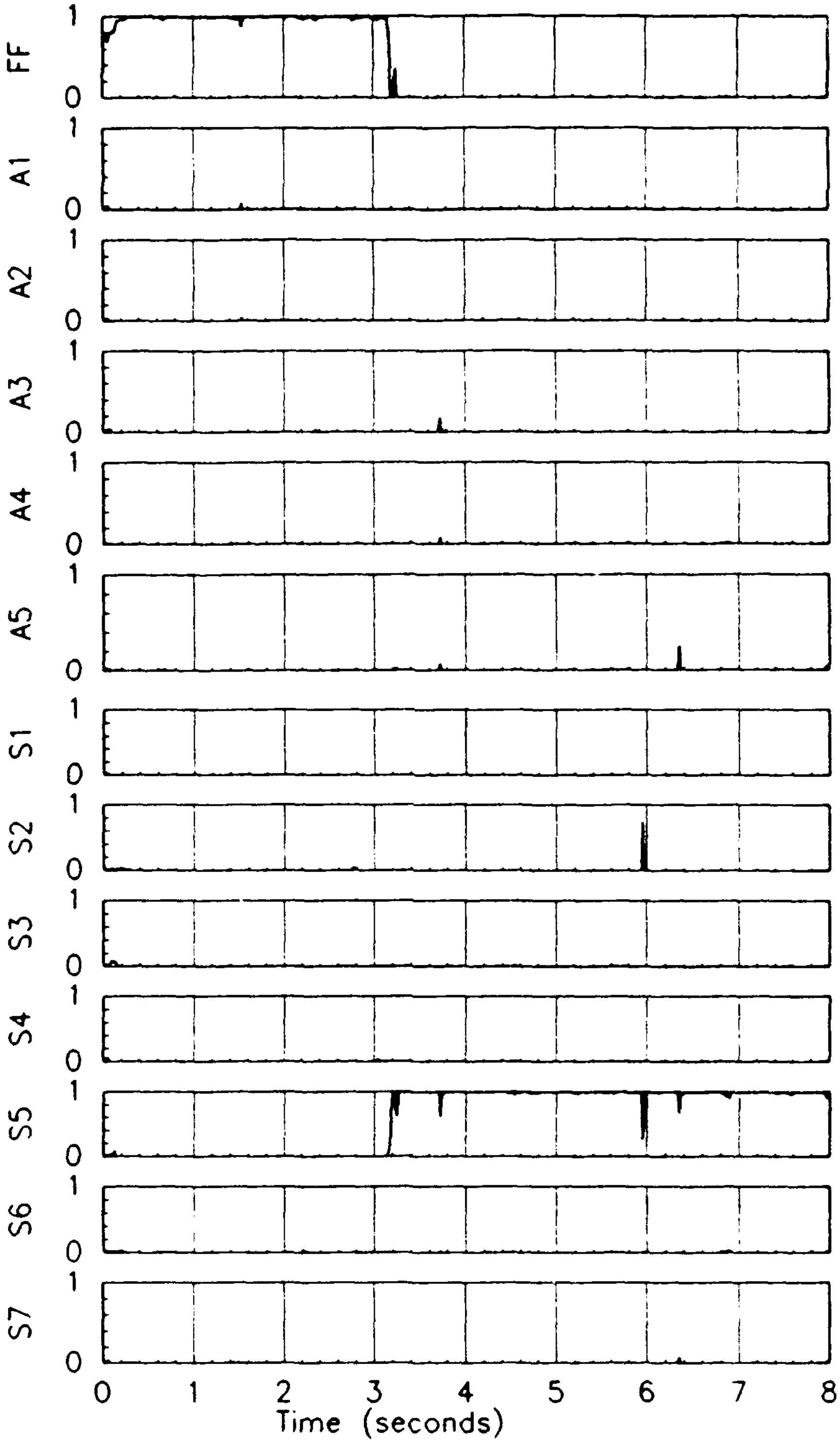

Figure 4.16 Probabilities for a roll rate sensor failure using a subliminal dither signal 1

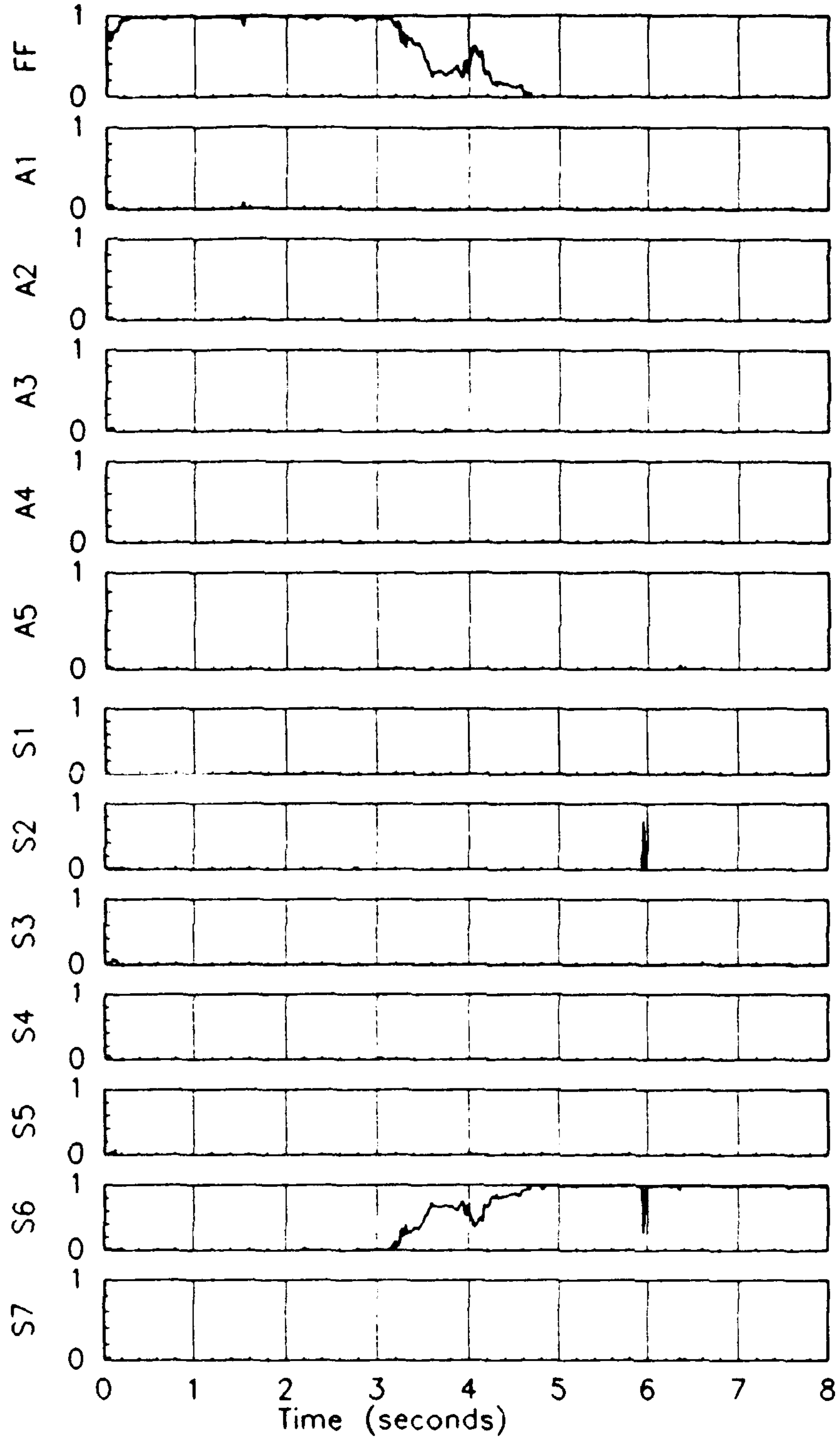

Figure 4.17 Probabilities for a yaw rate sensor failure using a subliminal dither signal 1

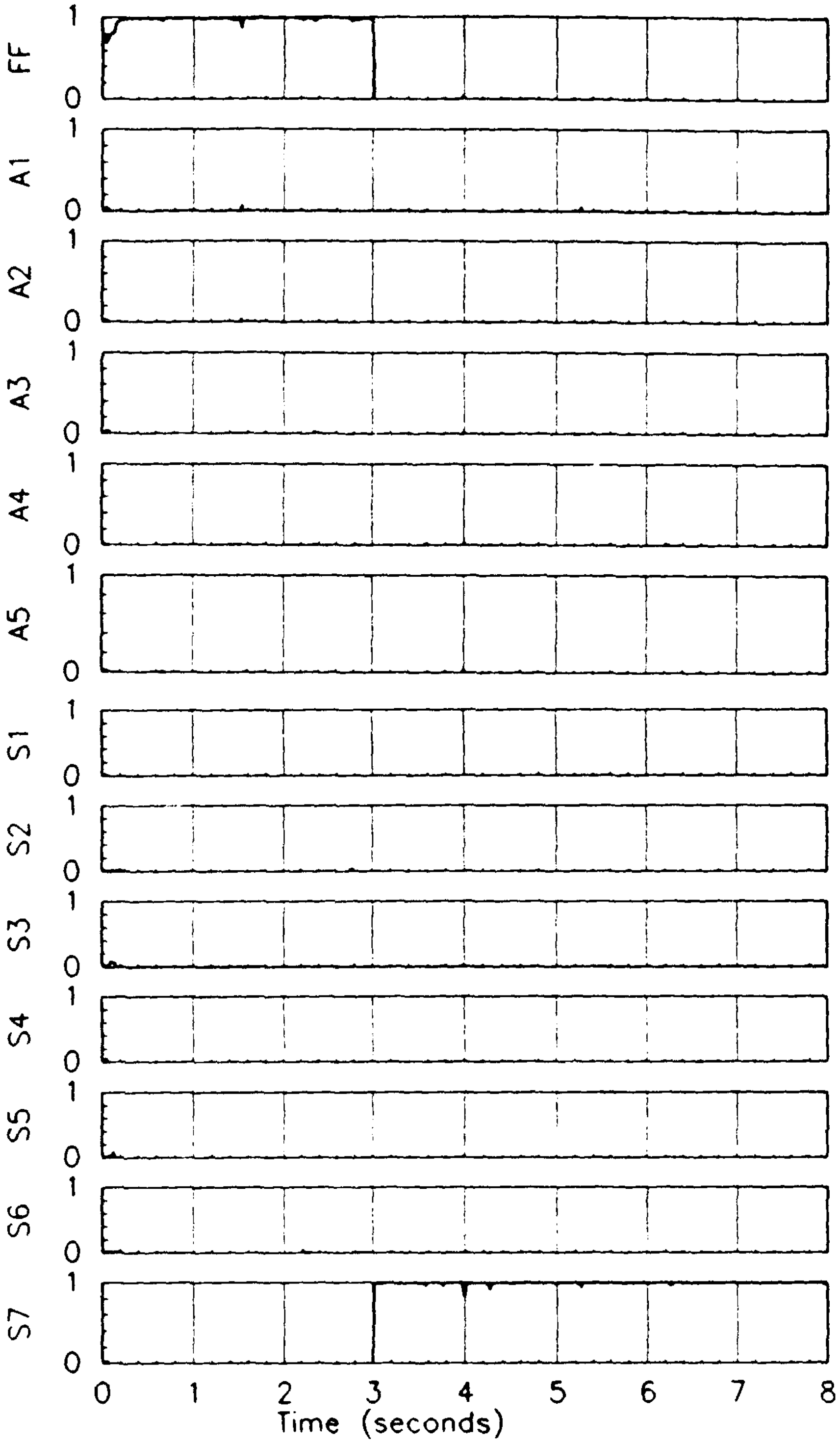

Figure 4.18 Probabilities for a lateral acceleration sensor failure using a subliminal dither signal 1

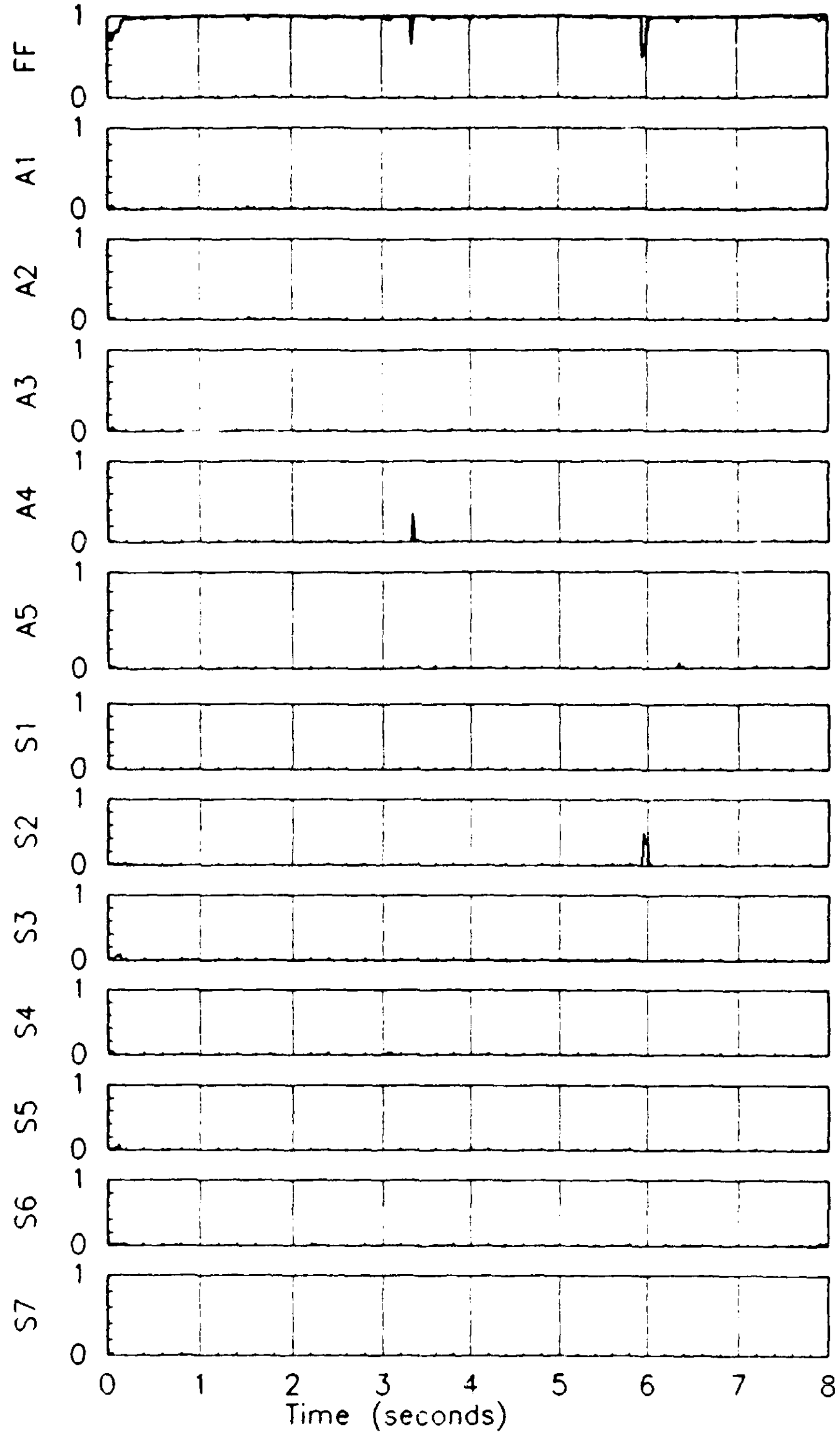

Figure 4.19 Probabilities for no failure scenario using subliminal dither signal 2

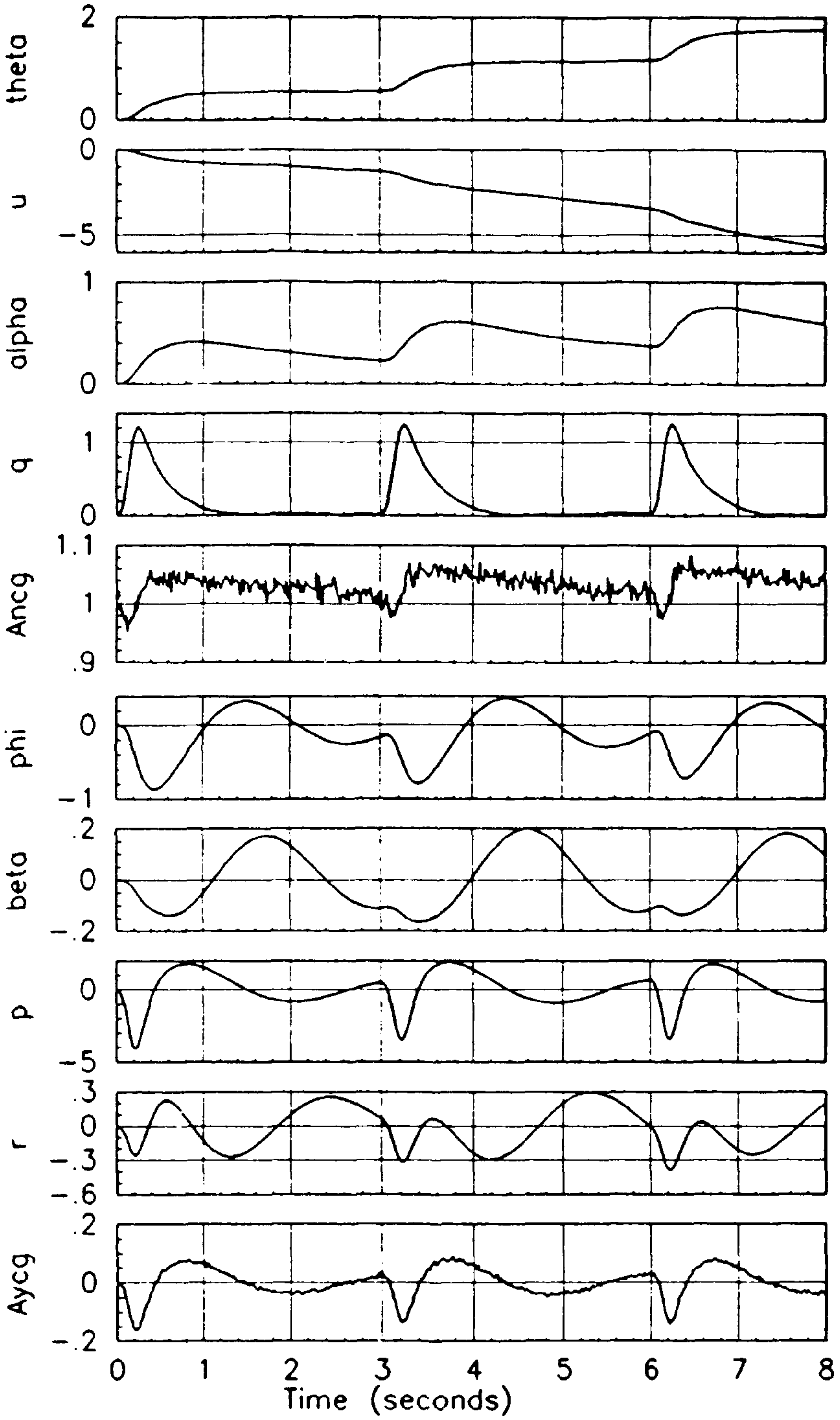

Figure 4.20 States for no failure scenario using subliminal dither signal 2

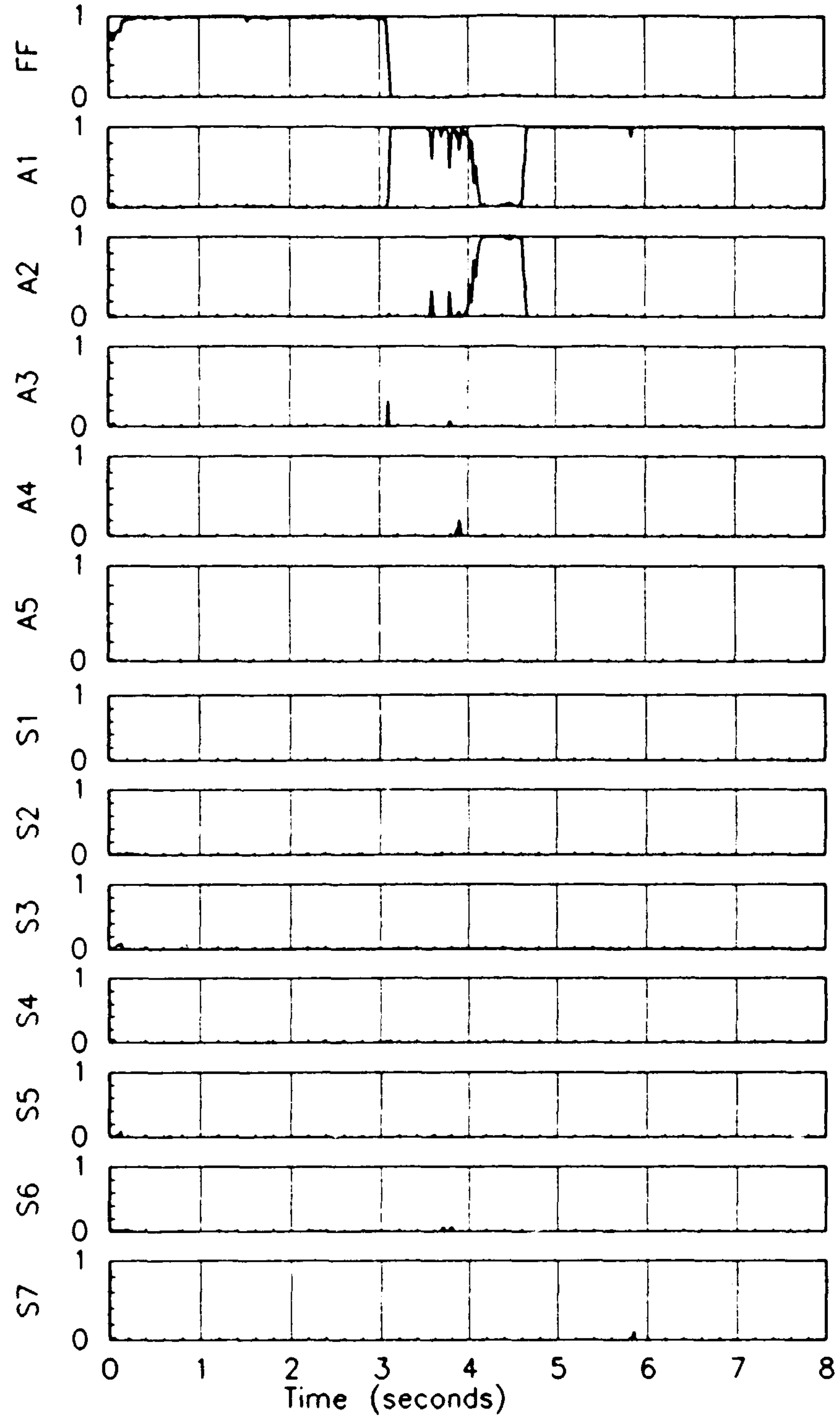

Figure 4.21 Probabilities for a left stabilator failure using subliminal dither signal 2

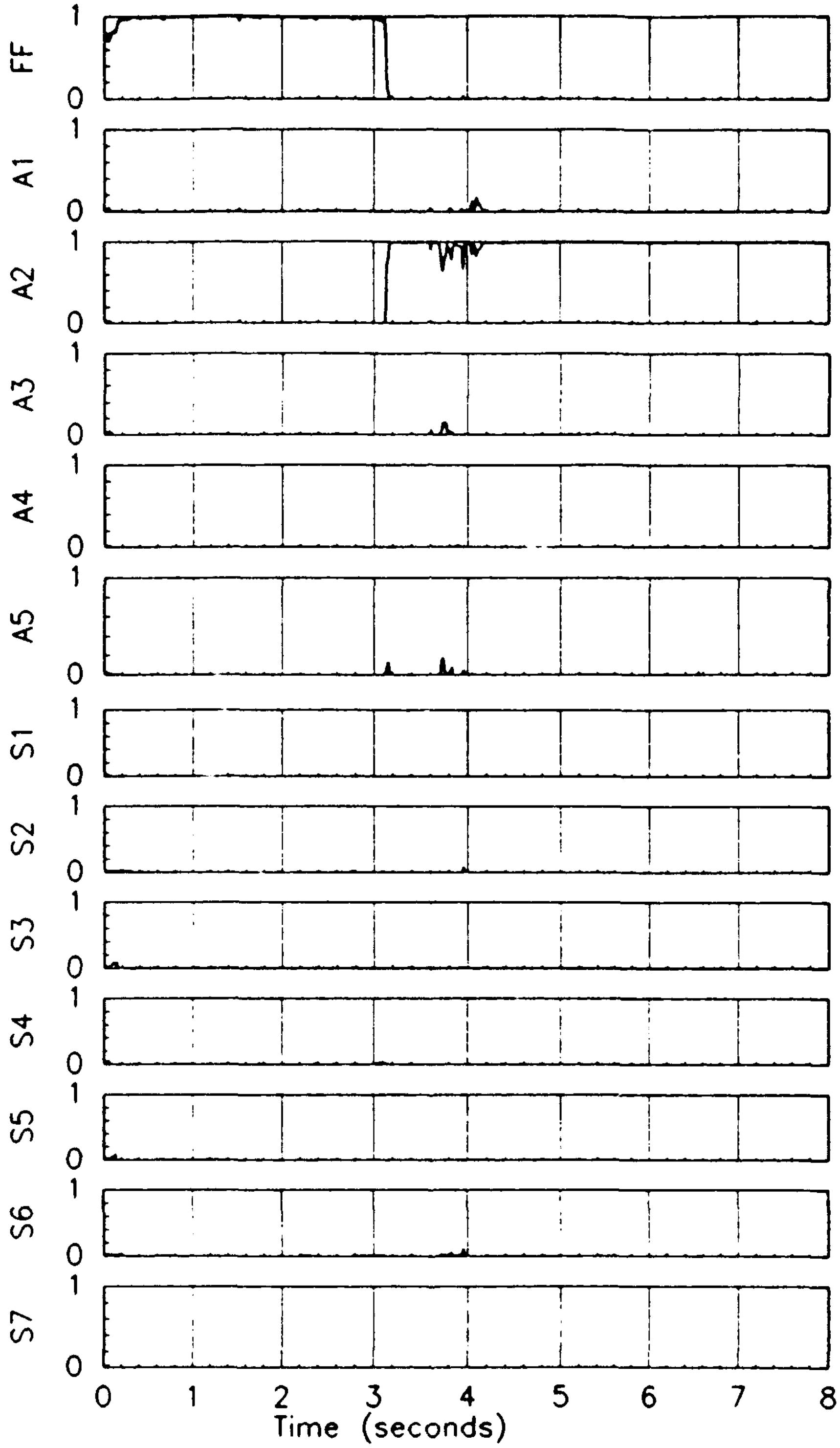

Figure 4.22 Probabilities for a right stabilator failure using subliminal dither signal 2

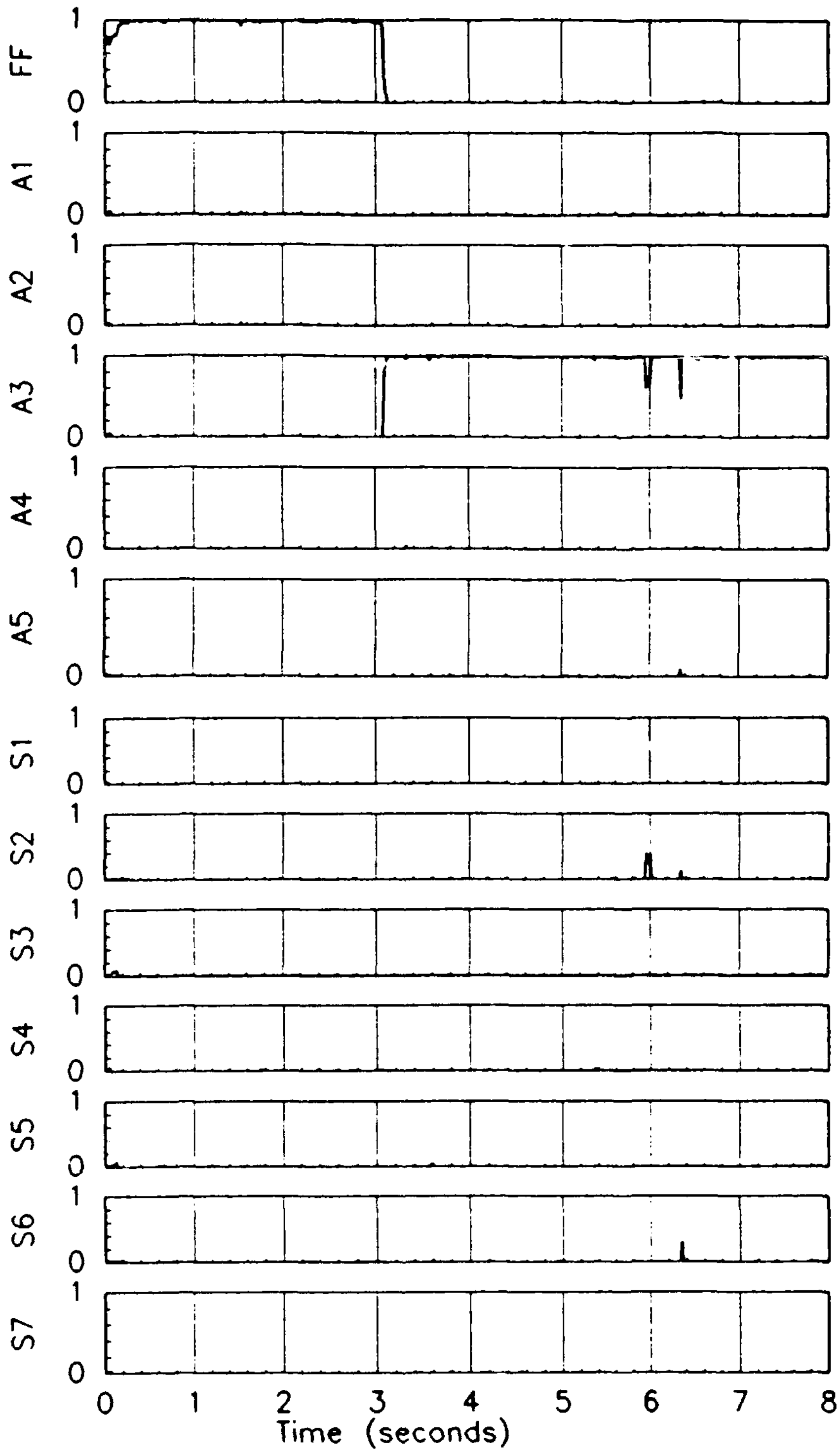

Figure 4.23 Probabilities for a left flaperon failure using a subliminal dither signal 2

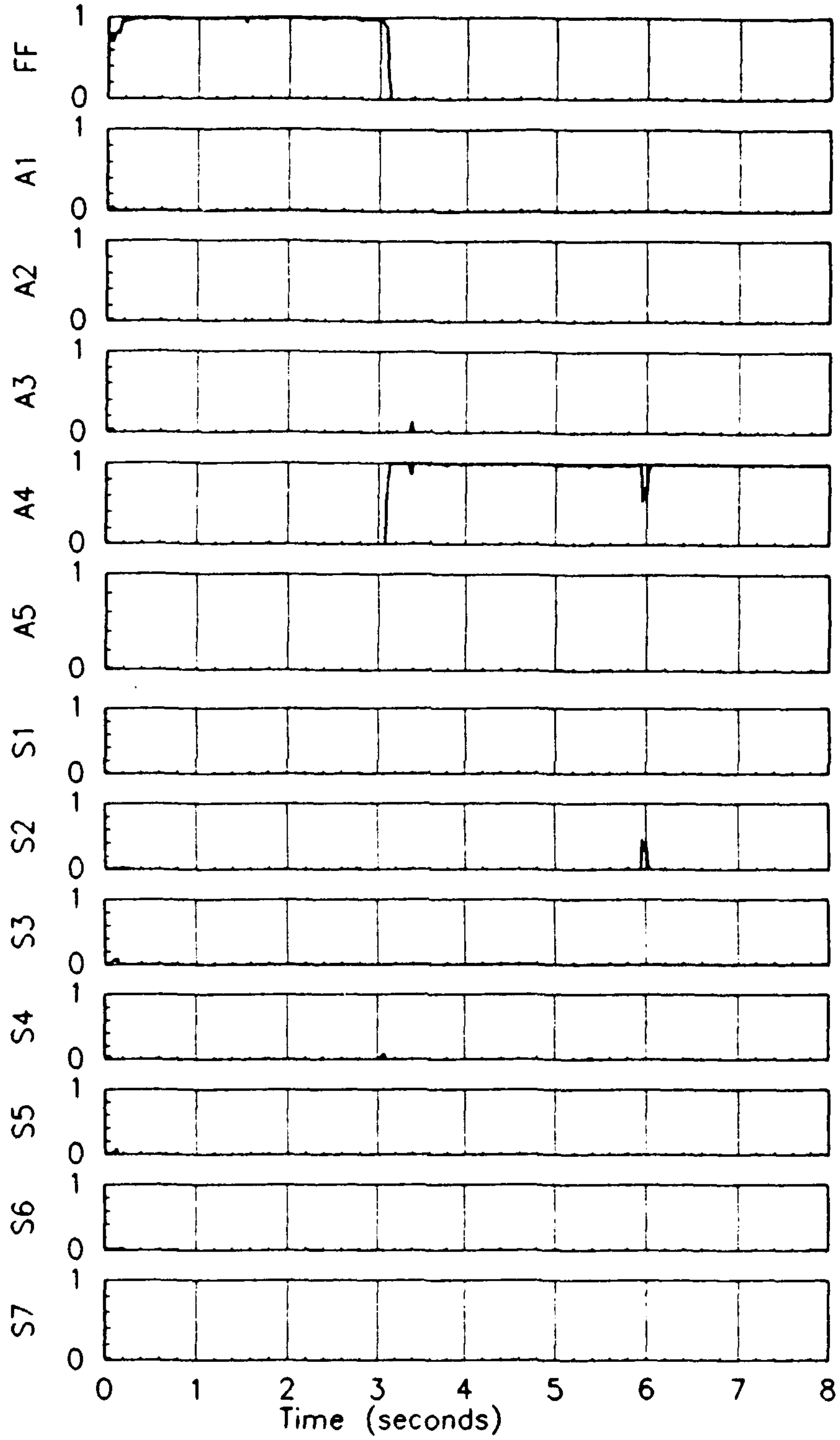

Figure 4.24 Probabilities for a right flaperon failure using a subliminal dither signal 2

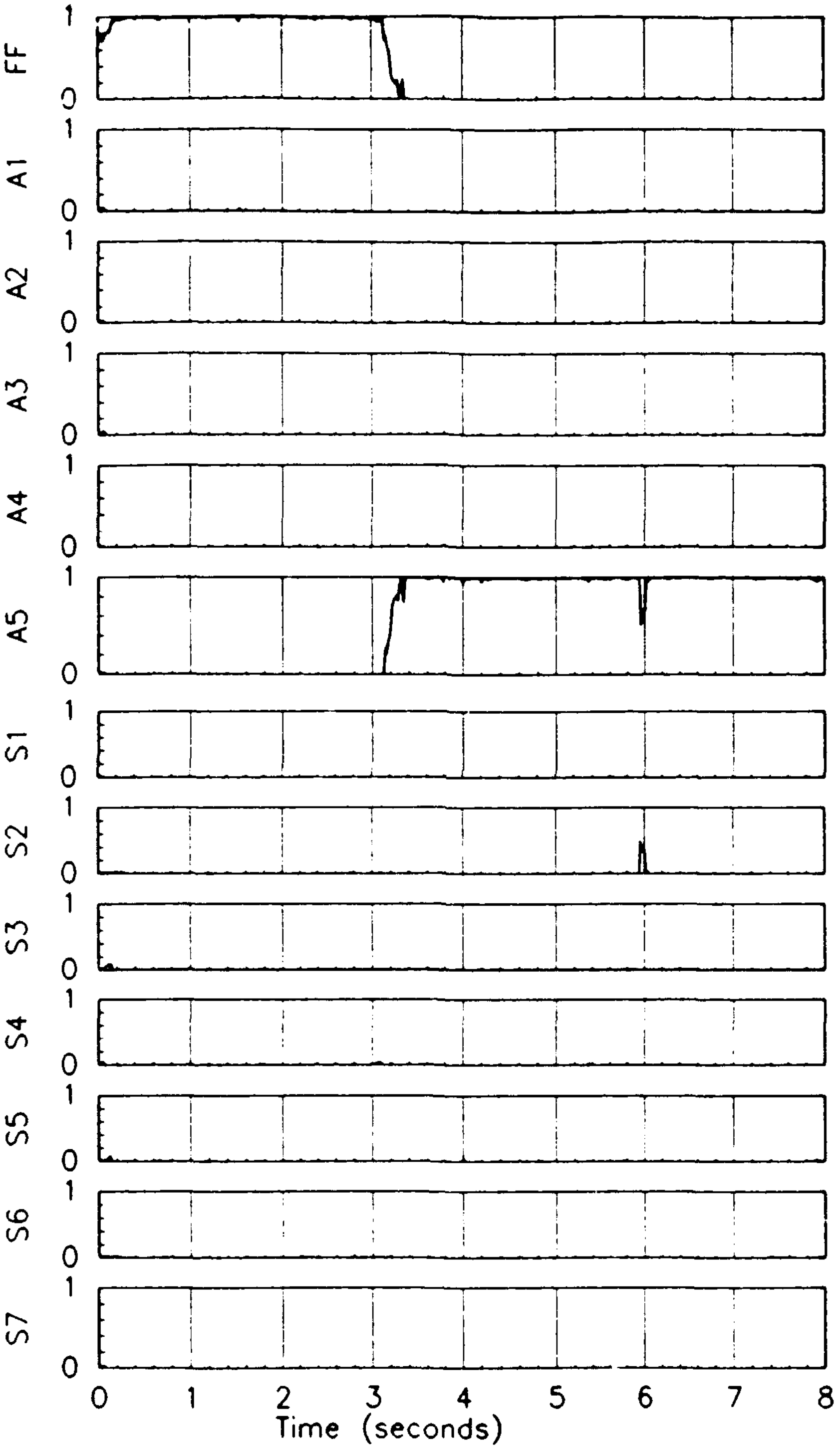

Figure 4.25 Probabilities for a rudder failure using a subliminal dither signal 2

4.2.3 Purposeful Commands

Purposeful commands are presented to demonstrate the MMAE algorithm's ability to detect and isolate failures during maneuvering flight. Figures 4.26 - 4.30 demonstrate the detection and isolation during a roll command. The results indicate excellent performance. Sensor failure results are not included since these results are excellent as well, as they were with no purposeful commands. During a purposeful command, the dither pulses in that channel, as presented previously, are suspended. The roll command used to generate Figures 4.26 - 4.30 is a roll and hold of 13.5 lateral stick lbs from 3.0 to 4.55 seconds. The side stick application times were selected to coincide with the failure insertion at 3.0 seconds and to demonstrate the detection capability with the command in and with the command released. A pull and roll command provided additional results. The aircraft was rolled by applying 13.5 lbs of lateral stick from 2.95 to 3.15 seconds. At 3.15 seconds, the lateral force was reduced to 10.0 lbs and a longitudinal command of 13.5 lbs was introduced. All commands were removed at 4.55 seconds. The results are very similar to Figures 4.26 - 4.30 with the exception of the rudder failure detection scenario. During the roll and pull maneuver, the isolation of the rudder failure required an additional 2.4 seconds (time of failure isolation was 6.15 seconds). During a multi-axis maneuver, the system may be excited too much, producing a larger delay for failure detection and isolation.

The application of a rudder kick and hold resulted in a misidentification of the rudder failure for a yaw rate sensor failure. The rudder input is applied at 2.95 seconds with a magnitude of 40 lbs. The probability strip charts can be found in Appendix A, plots A.30 through A.43. The misidentification occurs as long as the rudder is held. Releasing the rudder produces the correct failure identification at the application of the dither pulse. To date, over 1000 data runs have been conducted, nearly 2000 in fact, with only one misidentification. Some non-identifications have occurred due to insufficient signal strength of the input or too active a signal resulting in large transients, which delay the identification process.

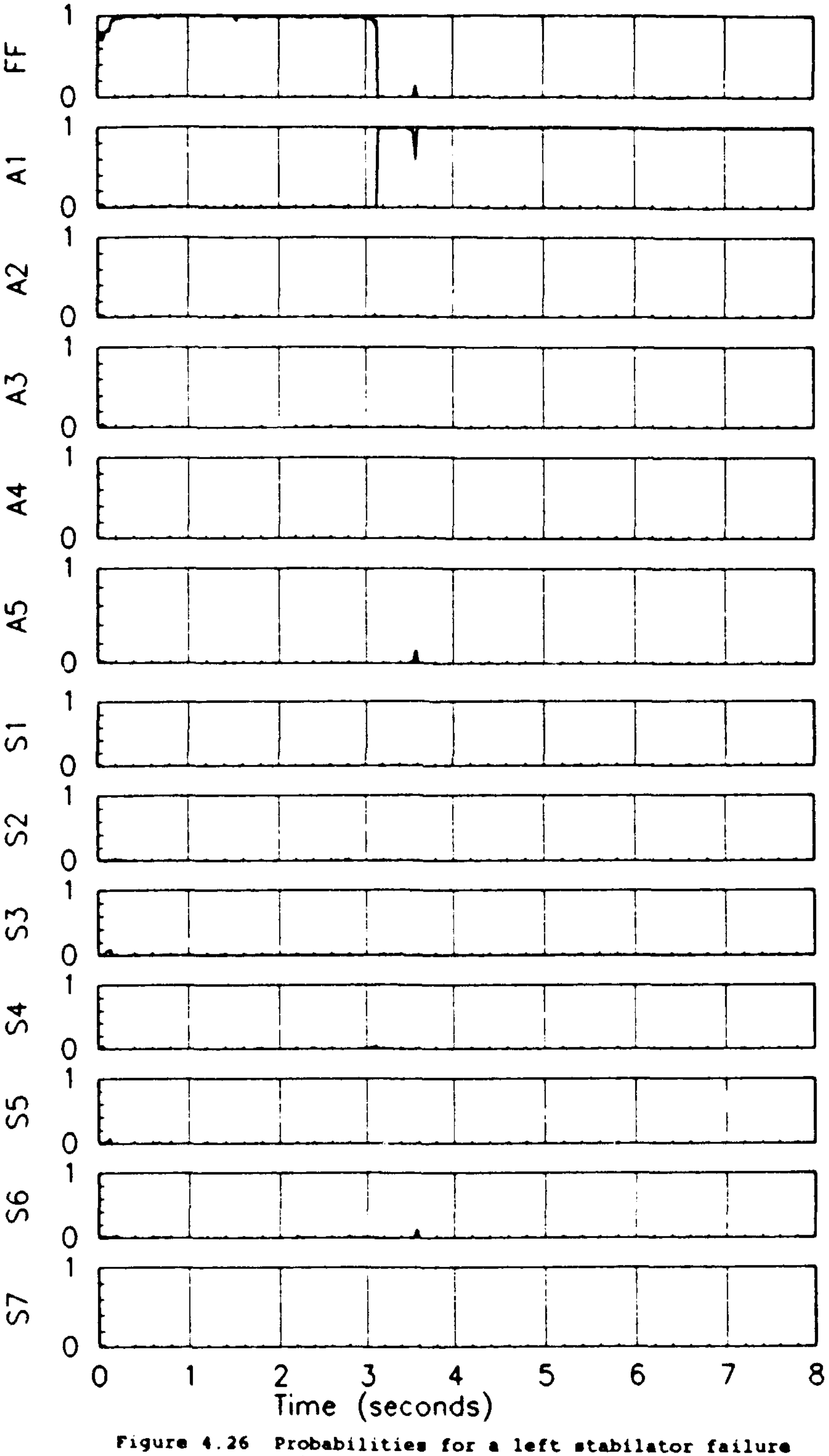

Figure 4.26 Probabilities for a left stabilator failure using a purposeful roll command

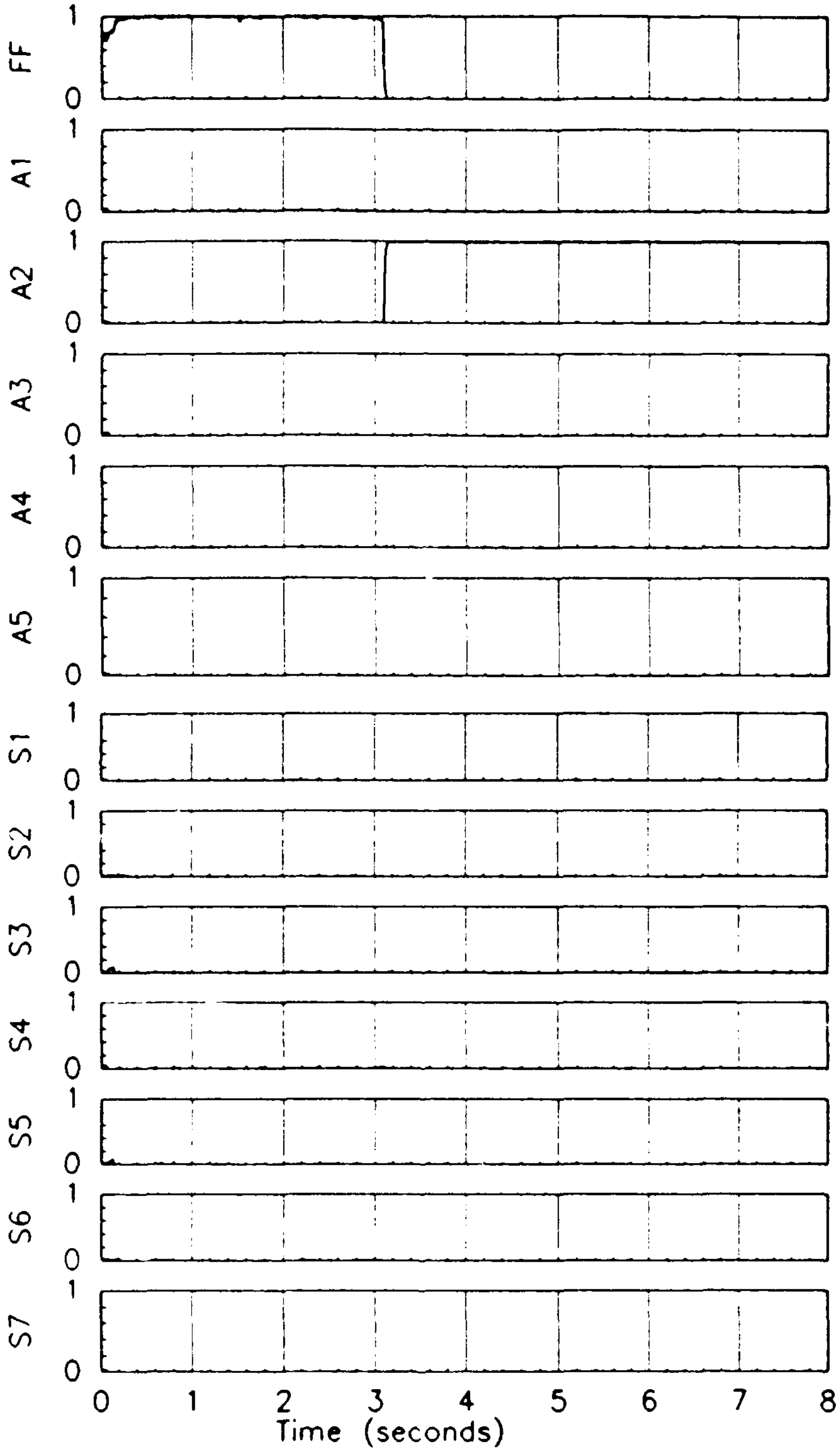

Figure 4.27 Probabilities for a right stabilator failure using a purposeful roll command

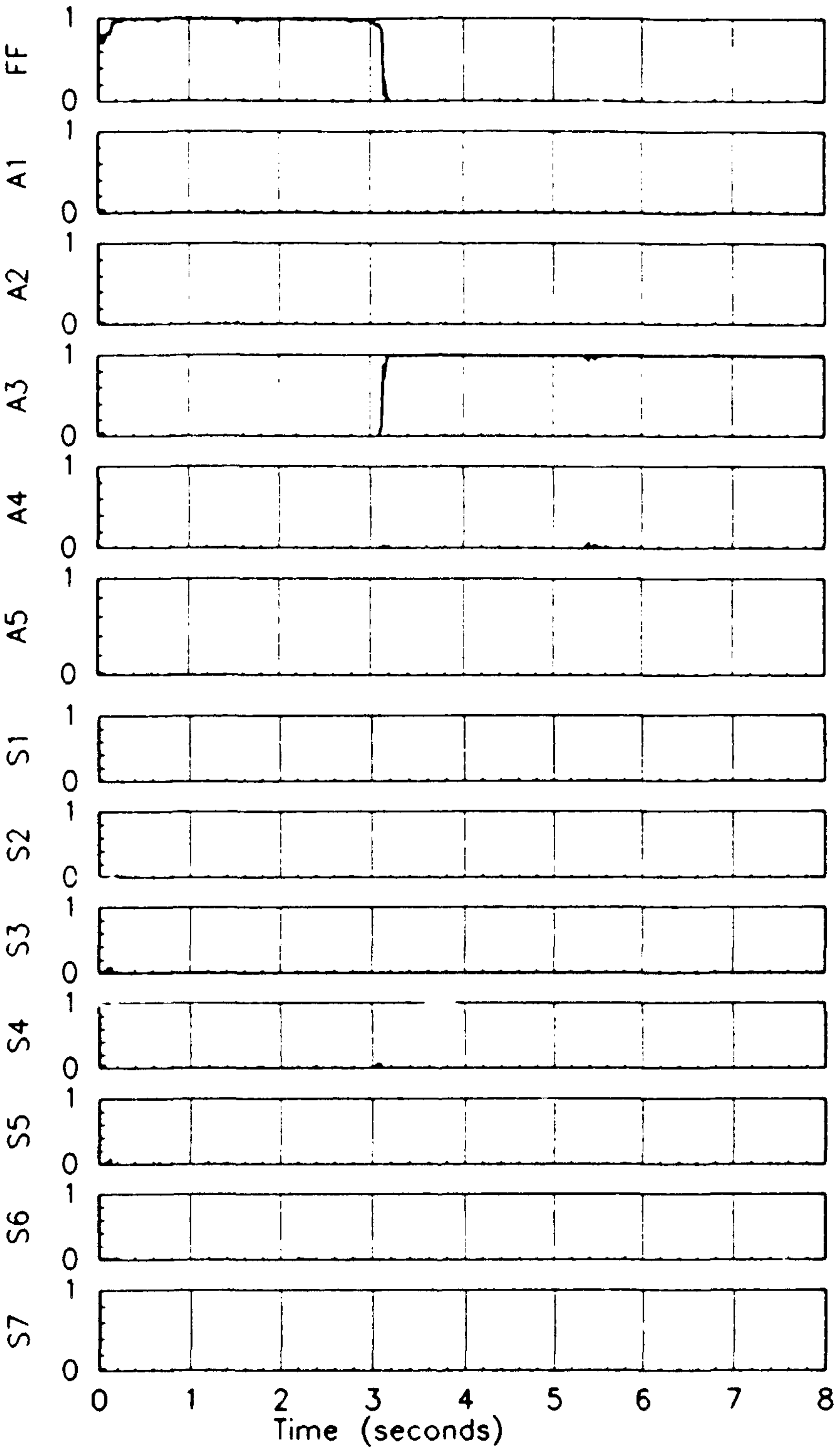

Figure 4.28 Probabilities for a left flaperon failure using a purposeful roll command

THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT

page 92 Fig. 4.29

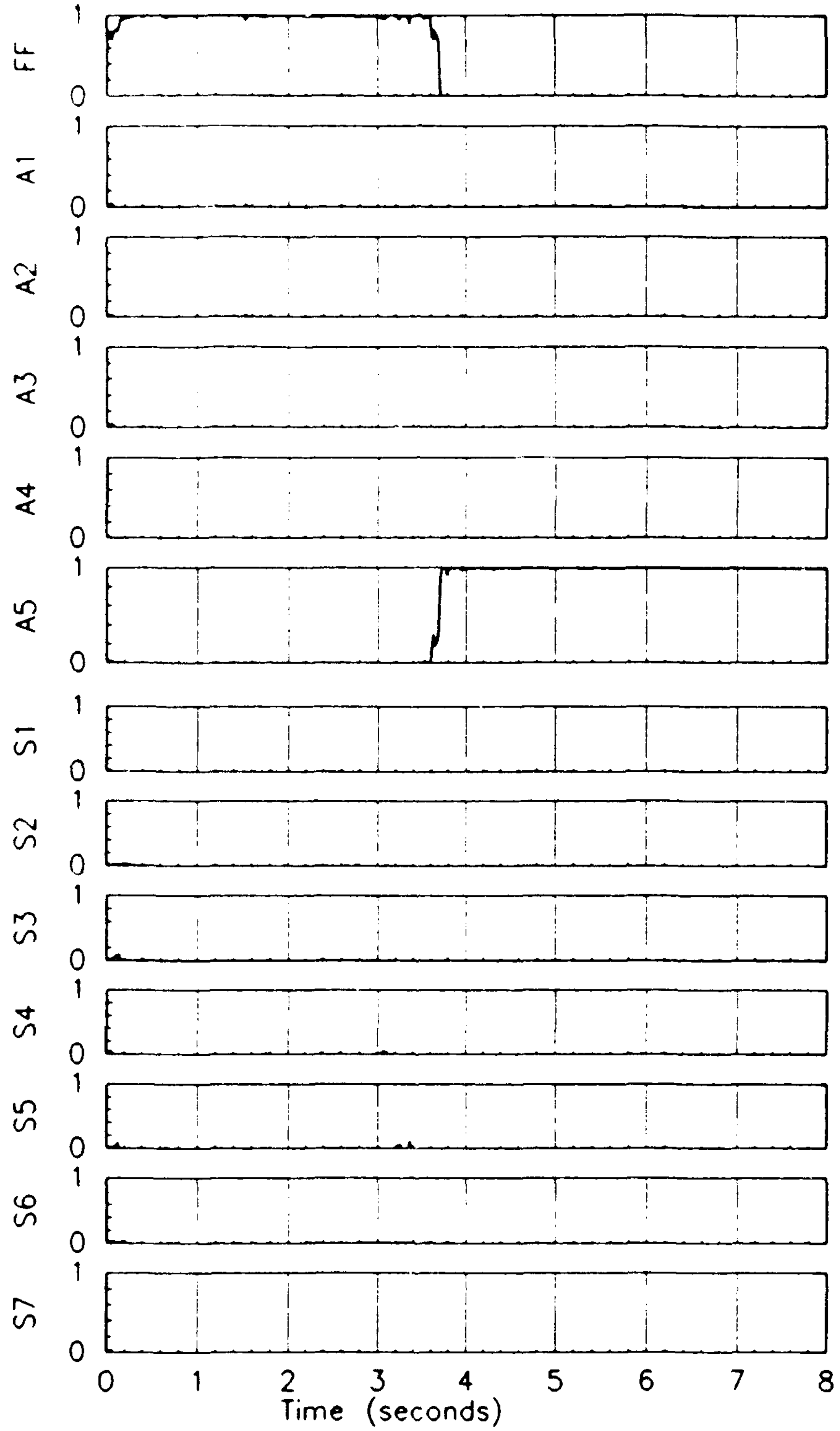

Figure 4.30 Probabilities for a rudder failure using a purposeful roll command

4.2.4 Subliminal Sinusoidal Dither Signals

The subliminal sinusoidal dither signal is a continuous dither signal in a sine wave form. The frequency, established by empirical methods, is 2.39 Hz. The magnitudes of the three pulses are chosen based upon empirical data. The frequency and magnitudes were established by attempting to provide sufficient excitation for good failure detection and identification while minimizing longitudinal and lateral directional accelerations and attempting to maintain a steady state trim condition. The advantage of a continuous pulse is that the failure can be detected at any time. A repeating pulse train with 3 second intervals between pulses cannot detect failures for all time between pulses, particularly just before the subsequent pulse. The MMAE algorithm is based upon large relative differences between residuals of elemental filters based upon different hypotheses. This requires a system excitation. A continuous dither signal continuously excites the system and provides constant failure detection capability.

Figure 4.31 demonstrates the no-failure (FF - fully functional) scenario for a continuous sinusoidal dither signal. The correct hypothesis is detected and identified within 0.4 seconds from the initial conditions. Figure 4.32 presents the corresponding state values. Note after the initial transients, the normal acceleration oscillates between 0.95 and 1.05 g's. The lateral acceleration oscillates between -0.15 and + 0.15 g's. The apparent growth in amplitude with time is of no concern as the lateral acceleration achieves a steady state value.

Figure 4.33 presents a left stabilator (A1) failure induced at 3.0 seconds. The failure is detected and locked at 3.2 seconds. A few dropouts occur in the left stabilator probability trace during the run; however, the dropouts are of very short duration and do not seriously affect the system performance. The ambiguity with the right stabilator, left flaperon, and roll rate sensor make physical sense. A left stabilator failure would reduce the pitch rate, introduce a roll rate, reduce the normal acceleration, modify the yaw rate, and change the lateral acceleration and change the velocity. These effects, coupled with the reduced excitation inputs from the stabilators, produce insufficient residual growth. By simply increasing the signal to the remaining stabilator, the ambiguities can be resolved. Figures 4.3 and 4.21 display the same ambiguity characteristics for a left stabilator failure for pulse dithering signals.

Figure 4.34 shows a right stabilator (A2) failure induced at 3.0 seconds. The right stabilator probability "chatters" during the first 2.0 seconds from the failure insertion and then locks. The "chattering" effect demonstrates the differences between the left and right stabilator failures. The longitudinal states are of slightly lower magnitude for the right stabilator failure and the lateral directional state traces display more dynamic effects. The result is less excitation in the longitudinal plane and more excitation in the lateral directional plane. Too much excitation in the lateral directional plane tends to delay the onset of probability convergence. In this failure scenario, ambiguity exists between the left stabilator, the rudder, and the normal acceleration sensor filters. This scenario is entirely different from the left stabilator failure; note, for instance, there is no ambiguity with the flaperons here. This indicates the differences in the control system's use of the stabilators. By failing the right stabilator, a different transient is produced, resulting in different residual characteristics. Comparison with previous pulse dither results (Figures 4.5 and 4.22) demonstrate the undesirable characteristics of a continuous sinusoidal dither signal. Continuous excitation can provide more dropouts than a pulse every 3.0 seconds. Figures 4.5 and 4.22 provide "cleaner" probability time histories than Figure 4.34.

Figure 4.35 demonstrates a left flaperon (A3) failure induced at 3.0 seconds. The failure is first detected at 3.35 seconds, locked at 3.4 seconds, breaks lock at 3.7 seconds, and finally locked at 4.0 seconds. The rudder filter contains a portion of the probability during the 3.6 to 4.0 time frame. Comparison with Figures 4.7 and 4.23 demonstrate "cleaner" probability traces than Figure 4.35. This characteristic is consistent through out the sinusoidal continuous dither scenarios.

The right flaperon (A4) failure is very similar to the left with the exception of a larger portion of probability being contained within the rudder filter during the 3.6 to 4.0 time frame. Figure 4.36 demonstrates this failure scenario. Figures 4.9 and 4.24 display right flaperon failures for pulse dither signals. A portion of the probability is contained in the yaw rate filter in Figure 4.9. Virtually no ambiguity exists in Figure 4.24, while in Figure 4.36 the rudder picks up part of the probability for a short period of time. Again, the sinusoidal dither, by virtue of its constant excitation, produces a probability time history with more drop outs than the pulse dither scenarios.

Figure 4.37 presents the rudder (A5) failure induced at 3.0 seconds. The failure is detected at 3.8 seconds, and finally locked at 5.0 seconds. Performance is good from 4.0 to 8.0 seconds. Comparison to Figure 4.10 highlights the sinusoidal dither's characteristics. The continuous excitation produces a probability time history with more dropouts. The delay time for the first pulse dither was 0.3 seconds. The delay for the sinusoidal dither is 1.0 sec to first lock and 2.0 sec to the final lock. The difference in delay times are two-fold. First, the magnitude used in the sinusoidal dither is smaller. Secondly, continuous excitation doesn't produce the best results as soon as the failure has been identified. Since the failure always occurred at 3.0 seconds, we timed the pulse also to occur at 3.0 seconds. This should produce the best results; however, it is not a realistic application by which to measure "real world" performance. The sinusoidal dither signal is not tied to any failure time. It will produce consistent results regardless of the failure time. A pulse dither signal with smaller times between the pulses would produce better results for a "real world" application. The closer the pulses are constructed together, the closer the signal is to a continuous dither signal.

All sensor failures resulted in excellent performance characteristics, as in the case with pulse dithers. Due to space constraints, sensor performance data for the sinusoidal dither signal is not included but can be found in Appendix A (Figures A.7 through A.13).

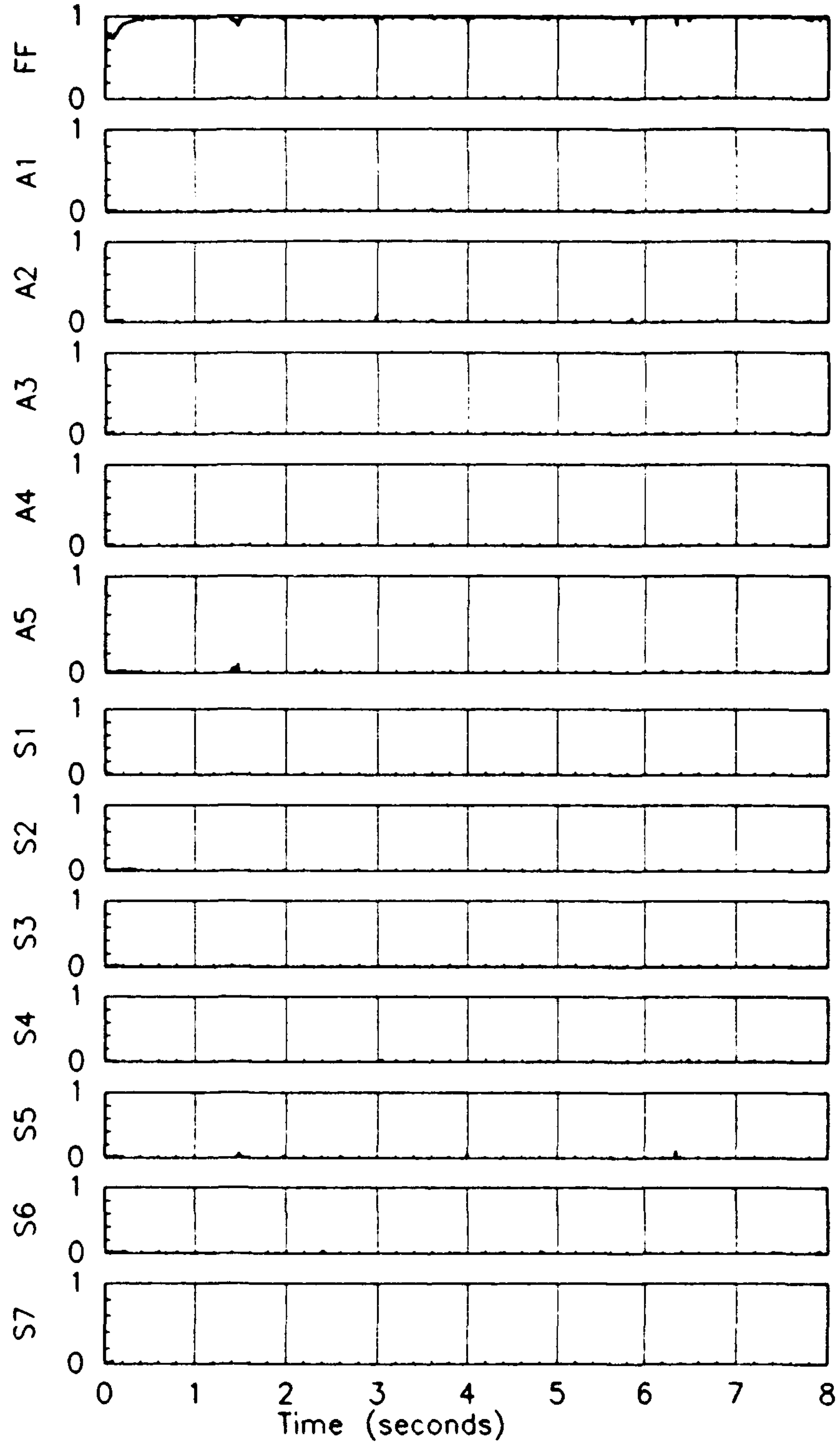

Figure 4.31 Probabilities for no failure scenario using a subliminal sinusoidal dither

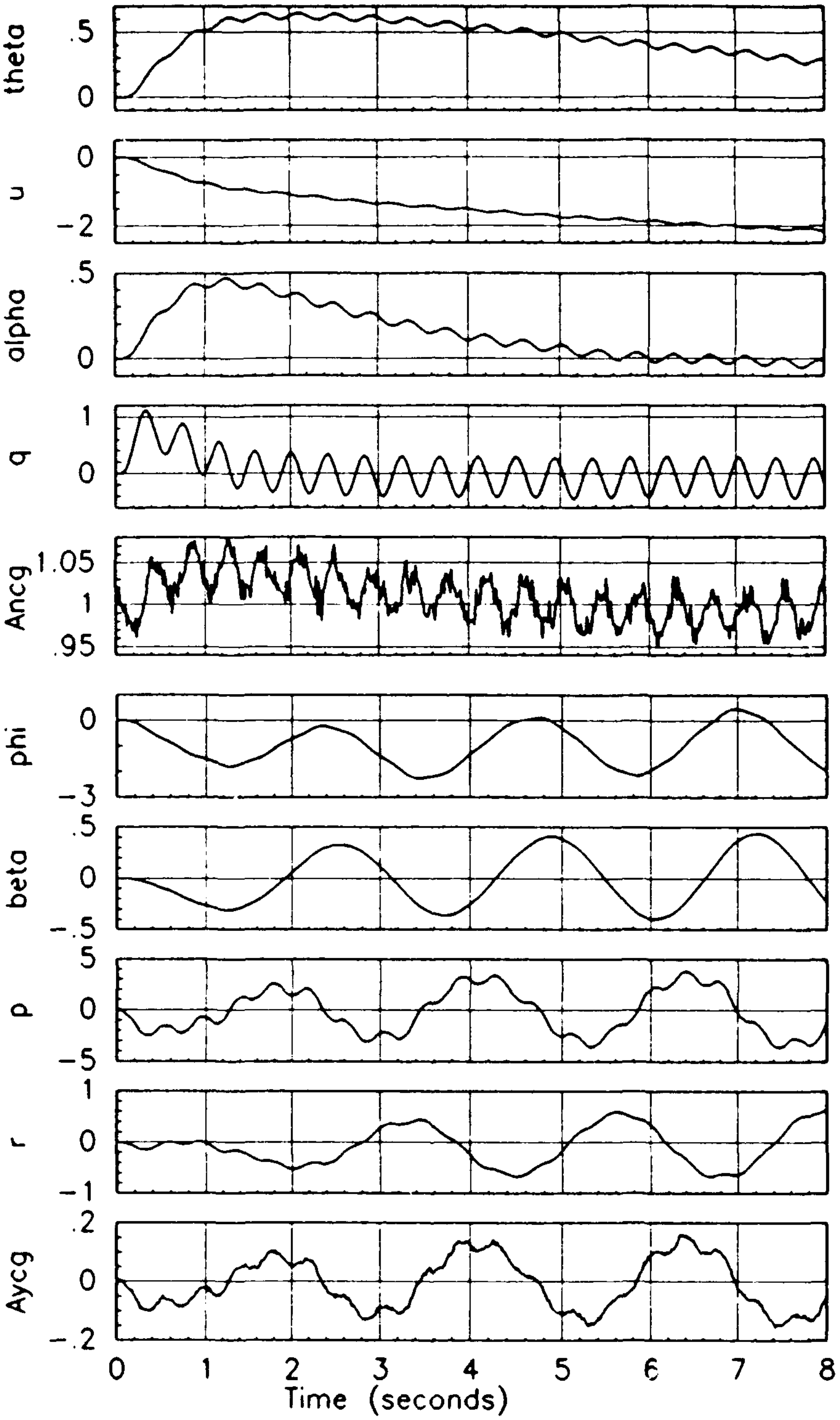

Figure 4.32 States for no failure scenario using a subliminal sinusoidal dither

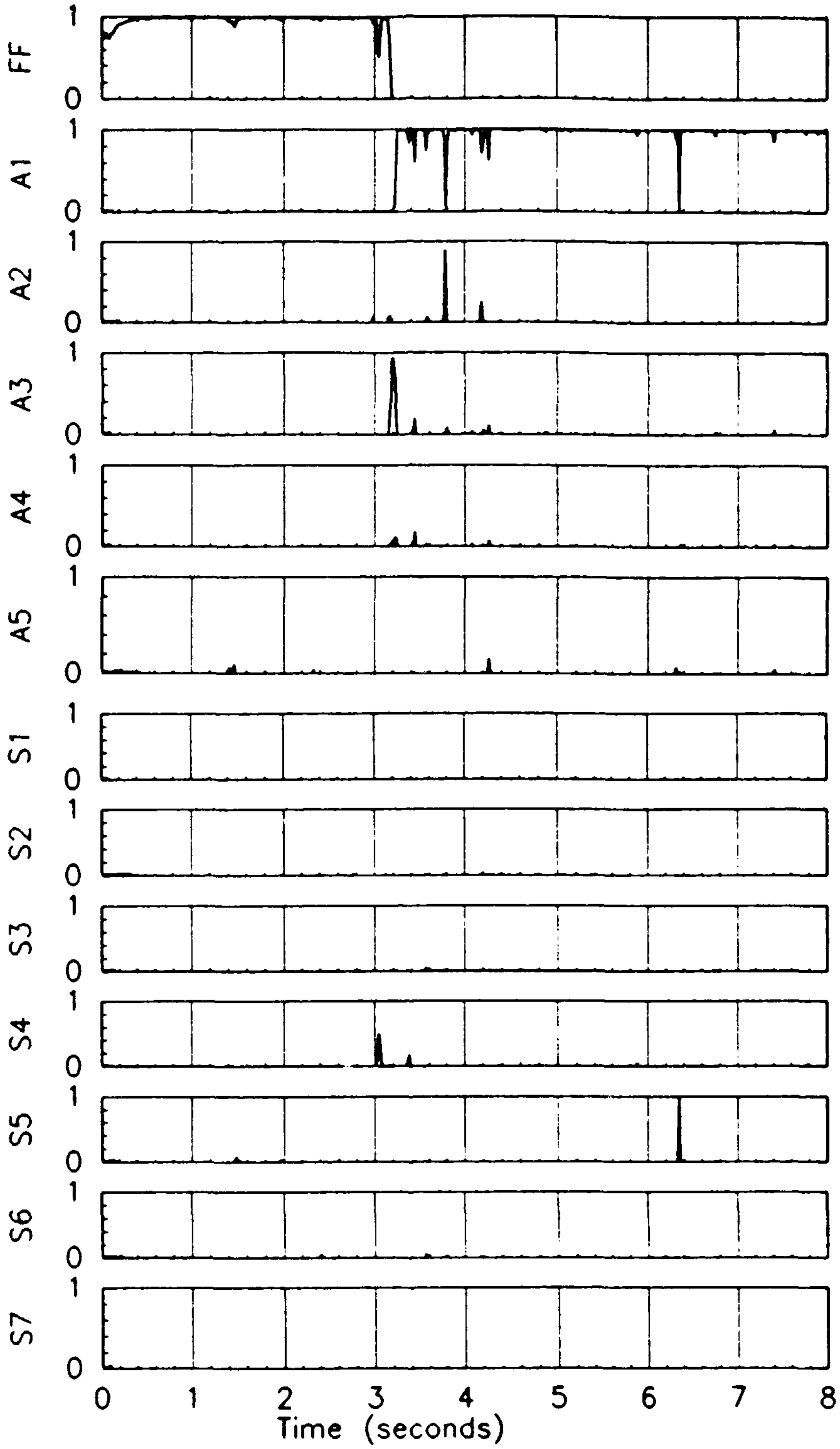

Figure 4.33 Probabilities for a left stabilator failure using a subliminal sinusoidal dither

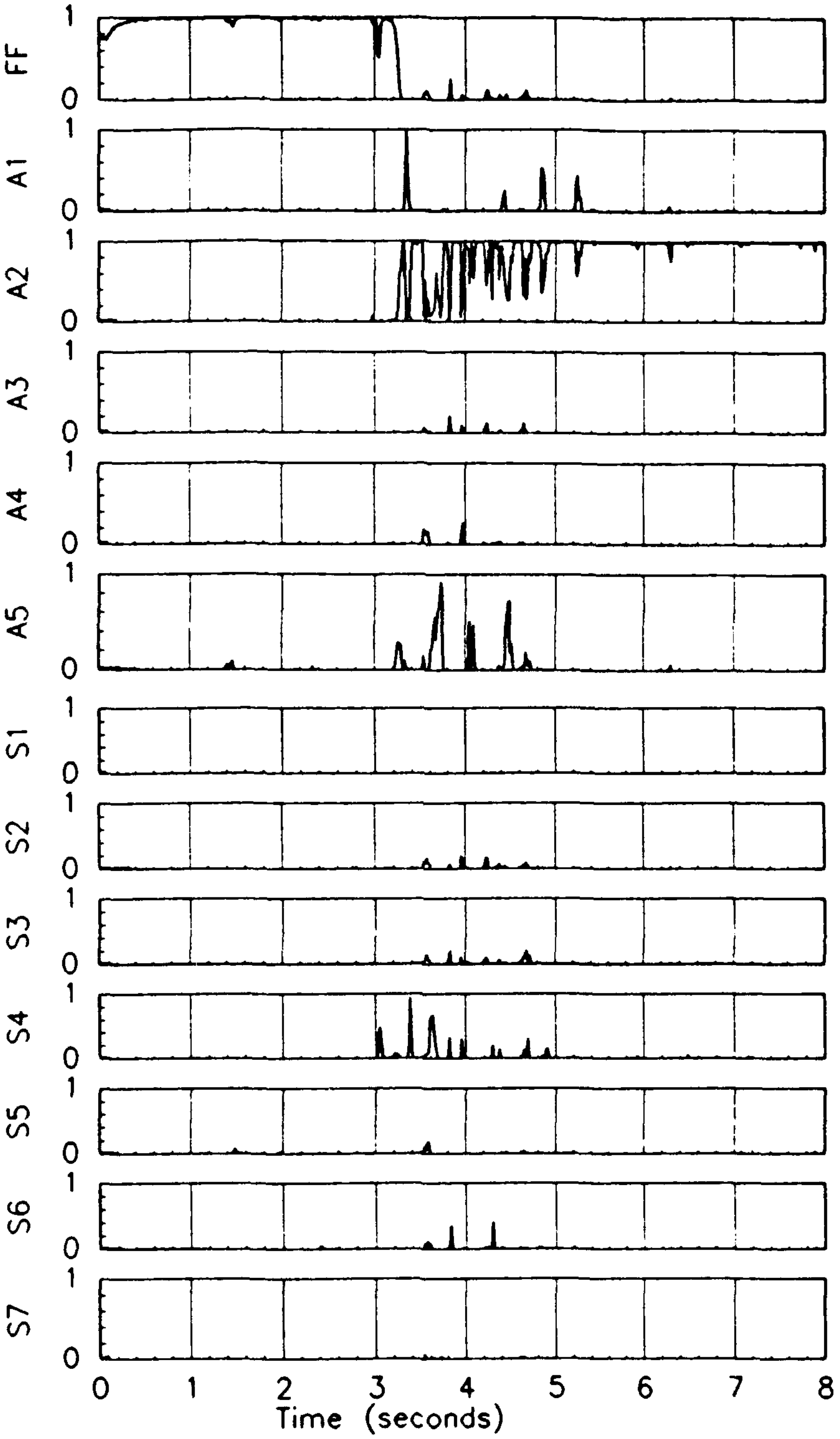

Figure 4.34 Probabilities for a right stabilator failure using a subliminal sinusoidal dither

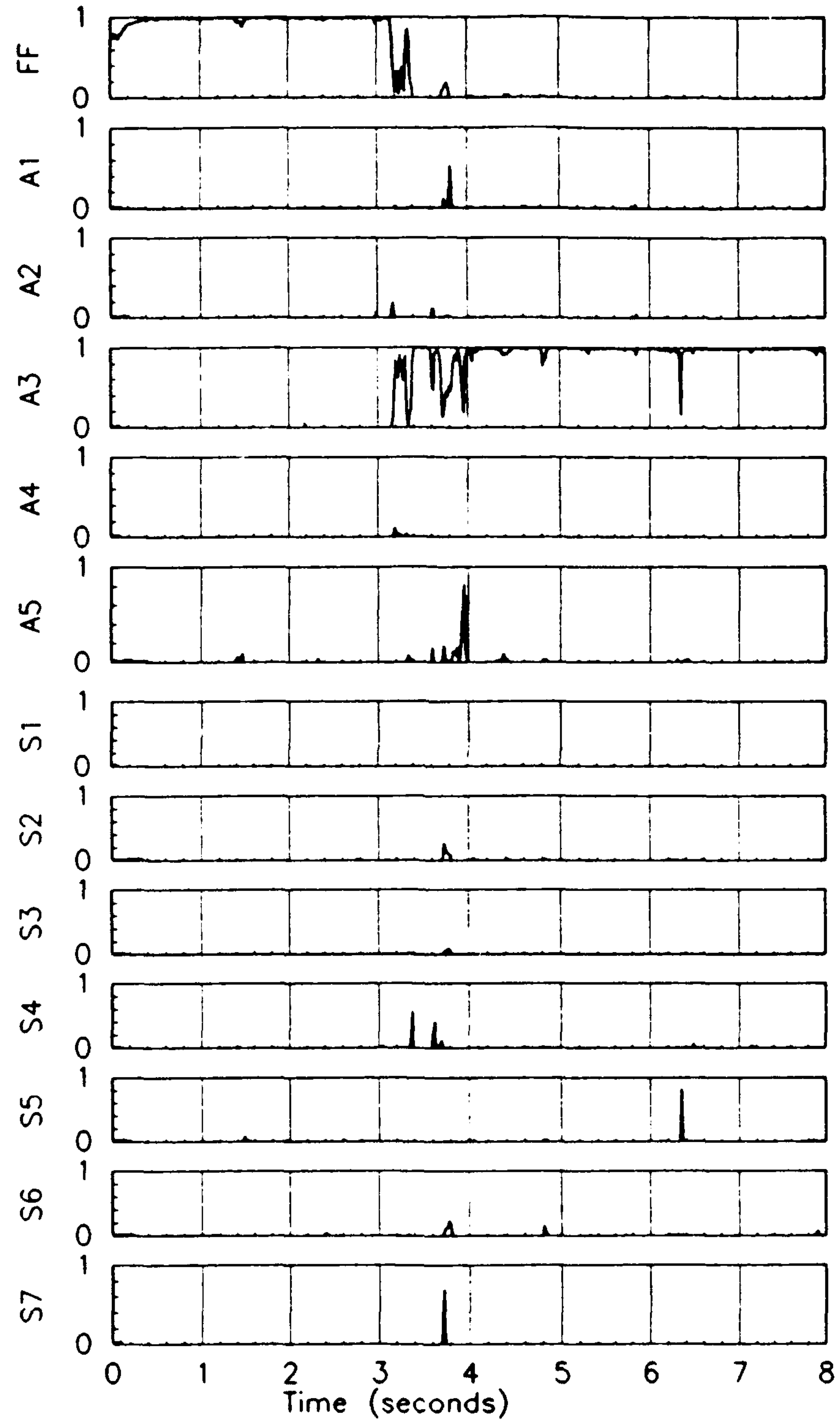

Figure 4.35 Probabilities for a left flaperon failure using a subliminal sinusoidal dither

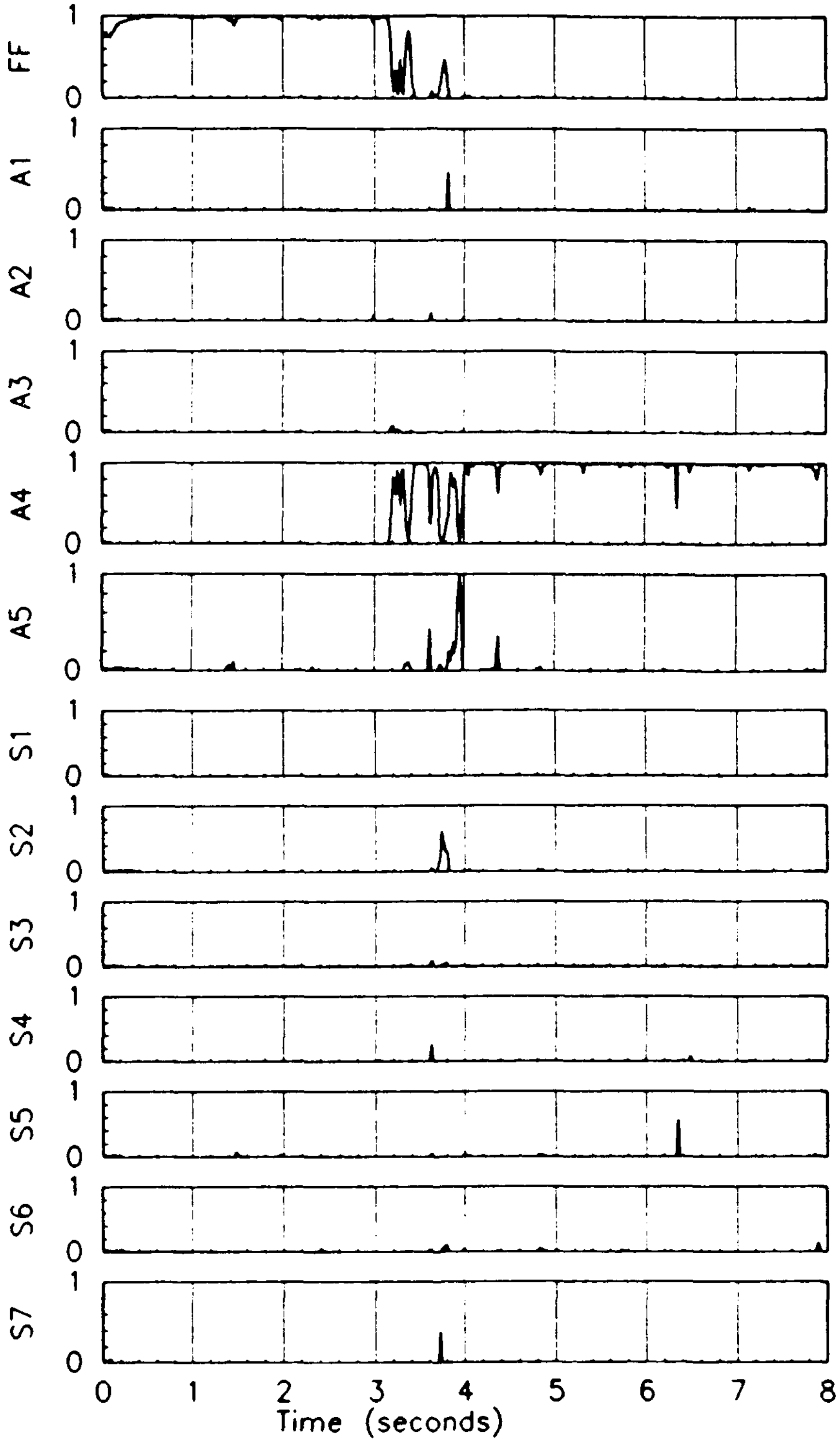

Figure 4.36 Probabilities for a right flaperon failure using a subliminal sinusoidal dither

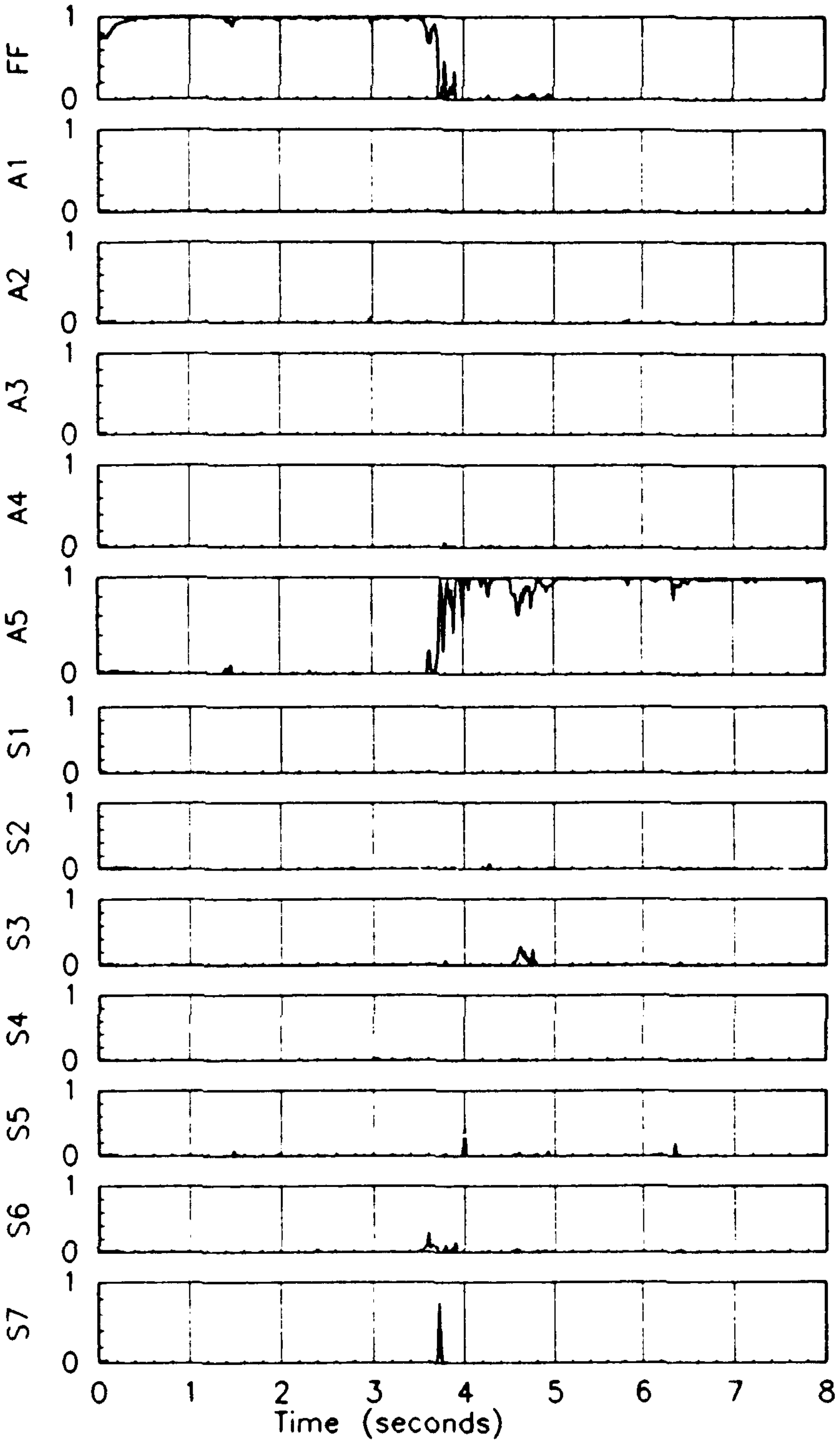

Figure 4.37 Probabilities for a rudder failure using a subliminal sinusoidal dither

4.2.5 Residual Monitoring

Residual monitoring can provide an additional vote in the declaration of a failure, as discussed in Section 2.4. This additional vote is particularly useful when resolving ambiguities. Seven scalar residuals are available for use in the determination of a failure. They are given as follows: velocity, angle of attack, pitch rate, normal acceleration, roll rate, yaw rate, and the lateral acceleration. Figures 4.38 - 4.44 demonstrate the scalar residuals' usefulness. The residuals presented are for the fully functional aircraft filter and the left stabilator elemental filter, given a left stabilator failure at 3.0 seconds. In each plot, +/- 3σ bounds are superimposed; these are derived from the filter-computed residual covariance $[H\ P^{-}\ H^{T} + R]$.

Figure 4.38a is the single scalar velocity residual for the fully functional aircraft filter given a left stabilator failure at 3.0 seconds. The velocity residual appears white and is within the +/- 3σ bounds prior to the induction of the failure at 3.0 seconds. At 3.2 seconds, the velocity residual is no longer white, appears sinusoidal with a frequency matching that of the sinusoidal dither, and violates the +/- 3σ bounds. The residual shows that the no-failure hypothesis is correct up to 3.0 seconds. Beyond 3.0 seconds, the no-failure hypothesis is incorrect. Figure 4.38b is the single scalar velocity residual for the left stabilator filter given a left stabilator failure at 3.0 seconds. A sinusoidal subliminal dither of 2.39 Hz is employed in a continuous fashion. From 0 to 3.0 seconds, the dither signal frequency is reflected in the residual (2.39 Hz). Its appearance in the residual, rather than a zero-mean white characteristic being exhibited, is a clear indication that the hypothesis of that elemental filter is wrong prior to 3.0 seconds into the simulation. At 3.0 seconds, the correct hypothesis is reflected by the left stabilator elemental filter. The velocity residual shifts within the 3σ bounds, a white zero-mean form is displayed, and the probability soon follows.

Figures 4.39a, 4.39b, 4.40a, and 4.40b do not exhibit such clear indications. Figures 4.41a and 4.41b demonstrates some of the frequency characteristics but not as cleanly as the velocity residual. Figures 4.42a, 4.42b, 4.43a, and 4.43b do not demonstrate clear indications of a correct hypothesis in this case.

Figure 4.44a is the lateral acceleration residual for the fully functional aircraft filter. This figure shows the lateral acceleration residual is white and within the +/- 3σ bounds prior to 3.0 seconds. After the induction of the failure at 3.0 seconds, the lateral acceleration residual is not white, has a frequency matching the sinusoidal dither signal, and violates the +/- 3σ bounds. The residual shows that prior to 3.0 seconds, the no-failure hypothesis is correct. After 3.0 seconds, the no-failure hypothesis is incorrect. Figure 4.44b, the lateral acceleration residual, clearly indicates the failure. From 0 to 3.0 seconds, the residual violates the 3σ bounds and has a frequency of 2.39 Hz. It is clearly not white: an attribute a residual signal should possess if the filter's assumed model is correct. From 3.0 to 8.0 seconds, the lateral acceleration residual looks white and falls within the 3σ bounds. Notice that a small bias prevents the signal from completely falling within the bounds.

To date, analysis indicates the velocity, normal acceleration, and lateral acceleration residuals produce the best indications of failures. A detection algorithm could be developed which could determine the "whiteness" of a residual simply by counting the zero crossings during a time interval. If residual biases are significant, as in Figure 4.44b, then an estimate of the bias (as accomplished by a simple finite-memory averaging) could be used to compensate the residual to a zero-mean signal before counting the zero crossings; such compensation would also allow residuals such as in Figure 4.44b to fall within the 3σ bounds. A large number of crossings indicating a white signal and the correctness of the filter-assumed hypothesis. A relatively small number of zero crossings would indicate a non-random signal or incorrect hypothesis. This technique would only work for a sinusoidal dither signal. Pulse dither traces do not demonstrate these attributes.

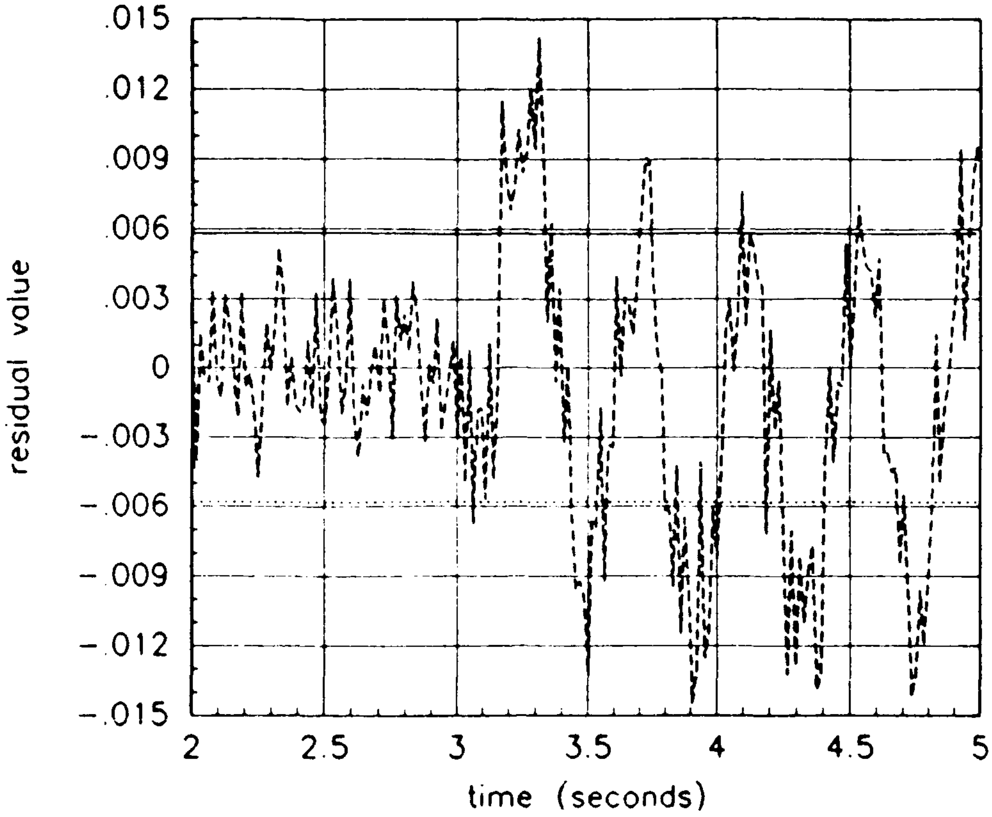

Figure 4.38a Single scalar velocity residual for the fully-functional filter given a left stabilator failure

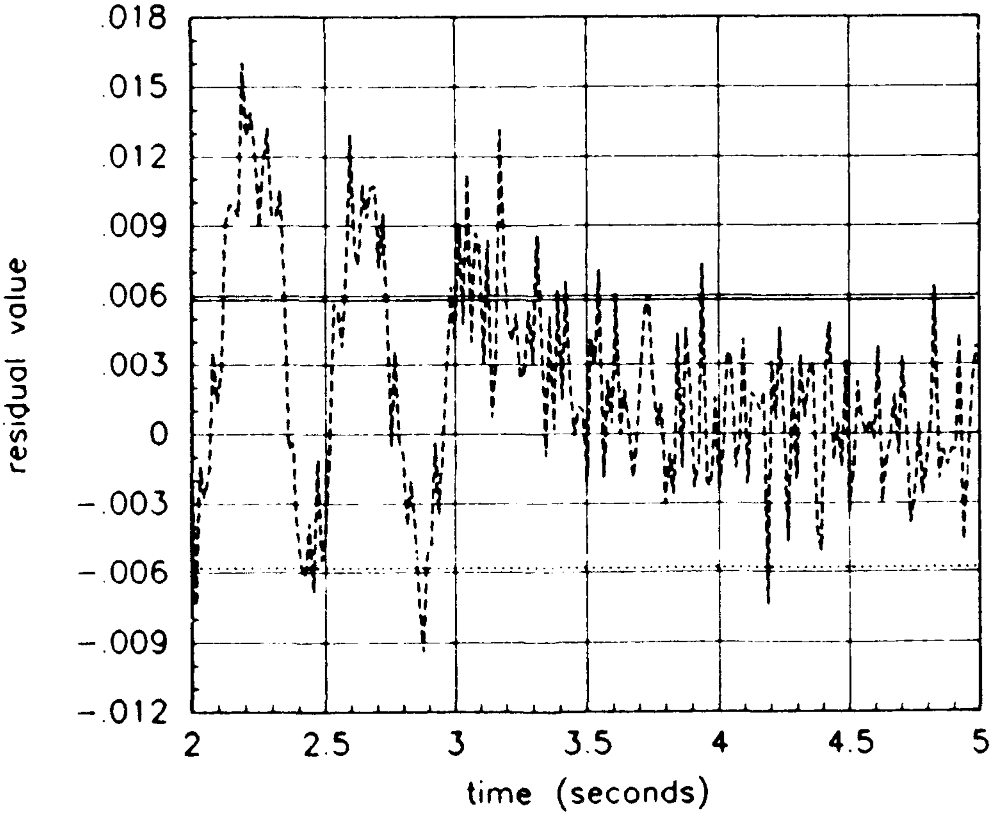

Figure 4.38b Single scalar velocity residual for the left stabilator filter given a left stabilator failure

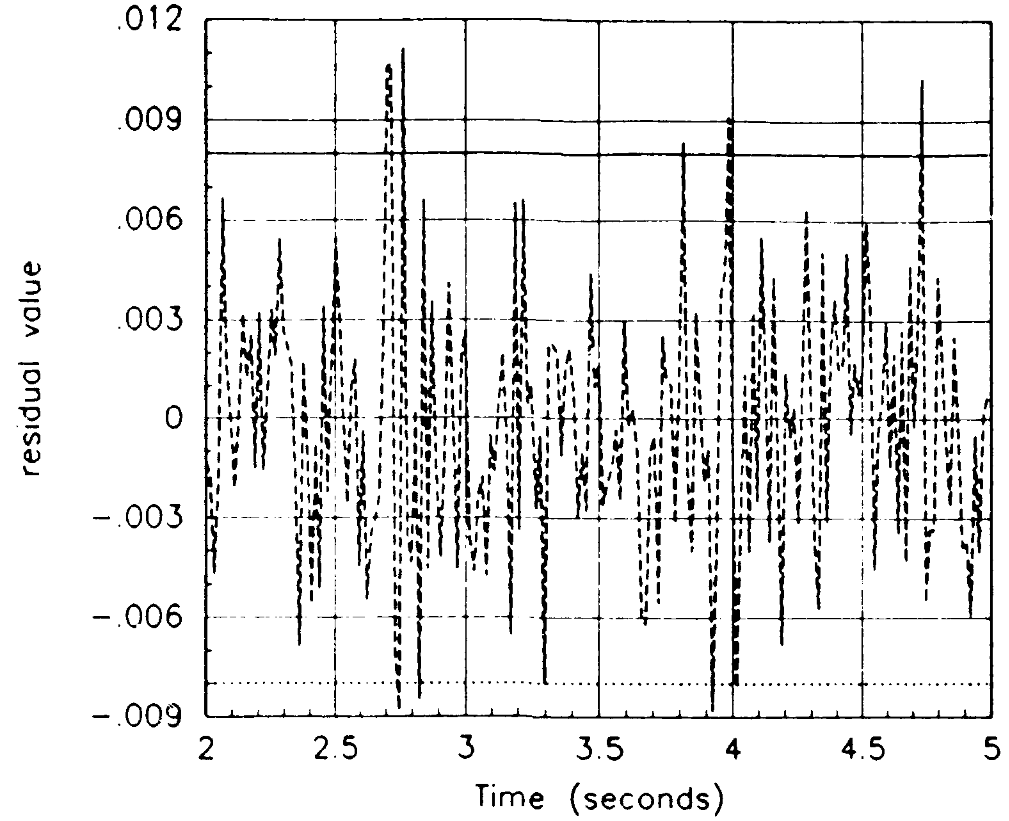

Figure 4.39a Single scalar angle of attack residual for the fully-functional filter given a left stabilator failure

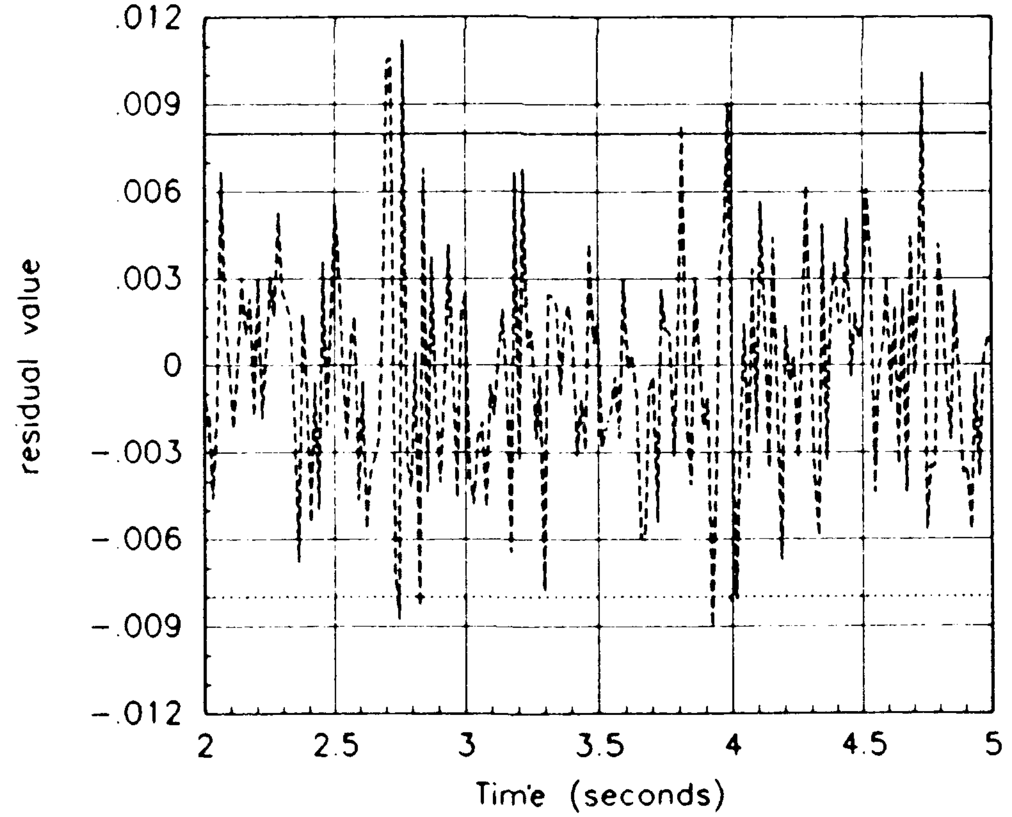

Figure 4.39b Single scalar angle of attack residual for the left stabilator filter given a left stabilator failure

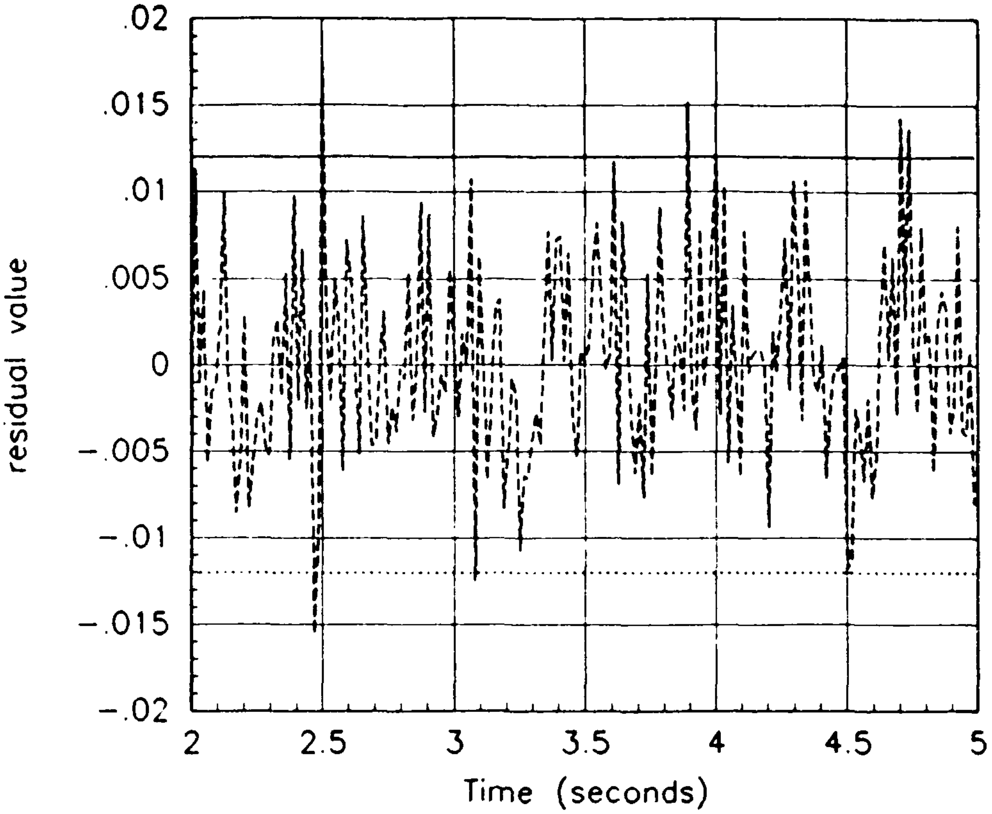

Figure 4.40a Single scalar pitch rate residual for the fully-functional filter given a left stabilator failure

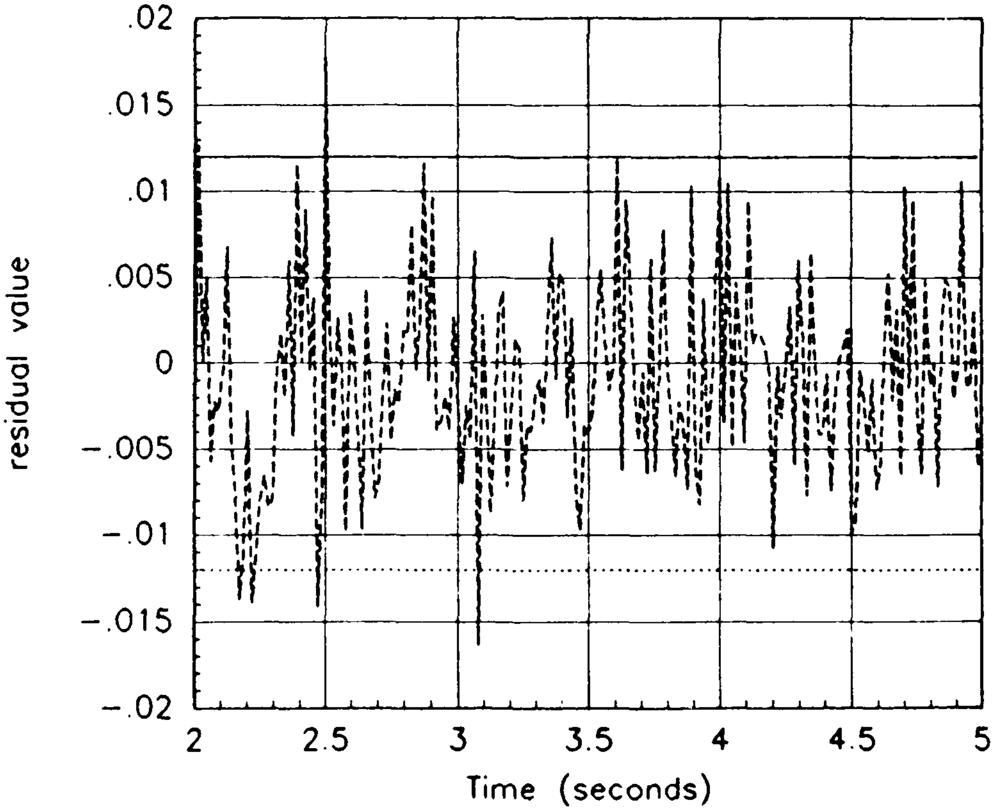

Figure 4.40b Single scalar pitch rate residual for the left stabilator filter given a left stabilator failure

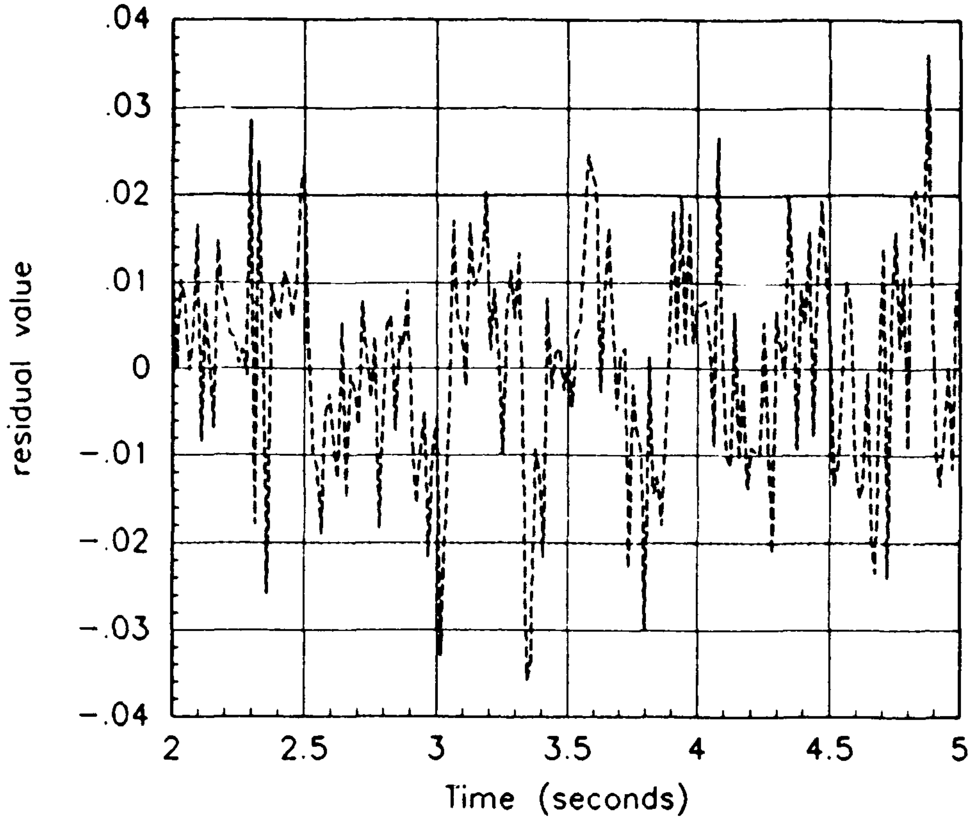

Figure 4.41a Single scalar normal acceleration residual for the fully-functional filter given a left stabilator failure

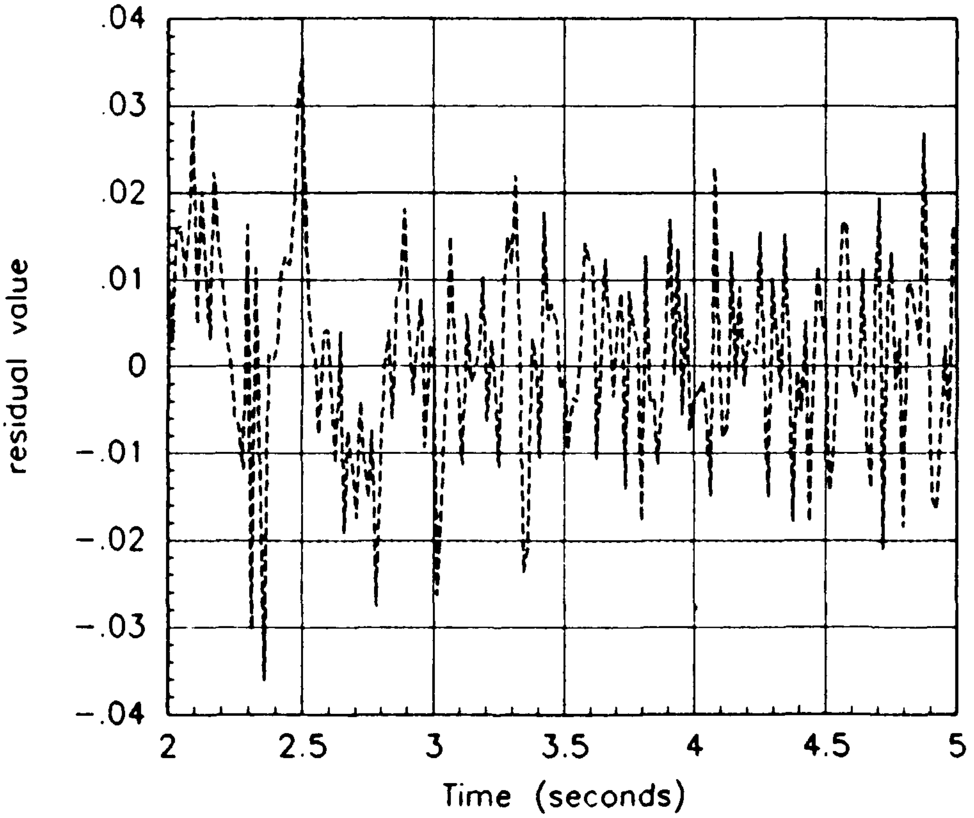

Figure 4.41b Single scalar normal acceleration residual for the left stabilator filter given a left stabilator failure

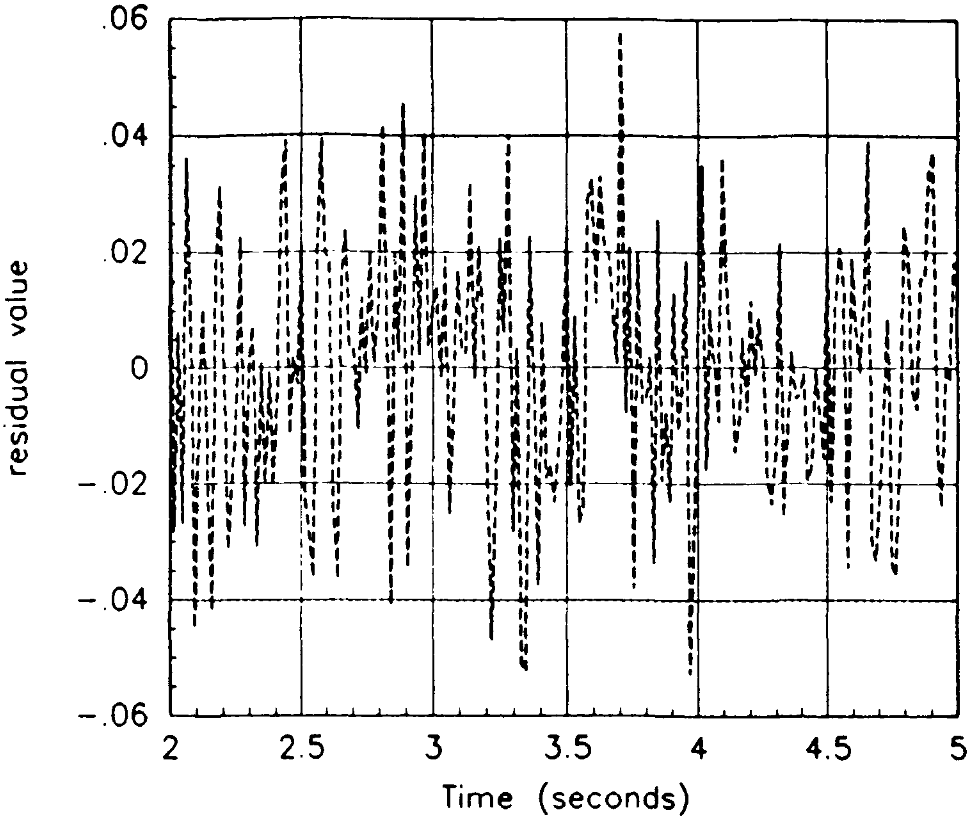

Figure 4.42a Single scalar roll rate residual for the fully-functional filter given a left stabilator failure

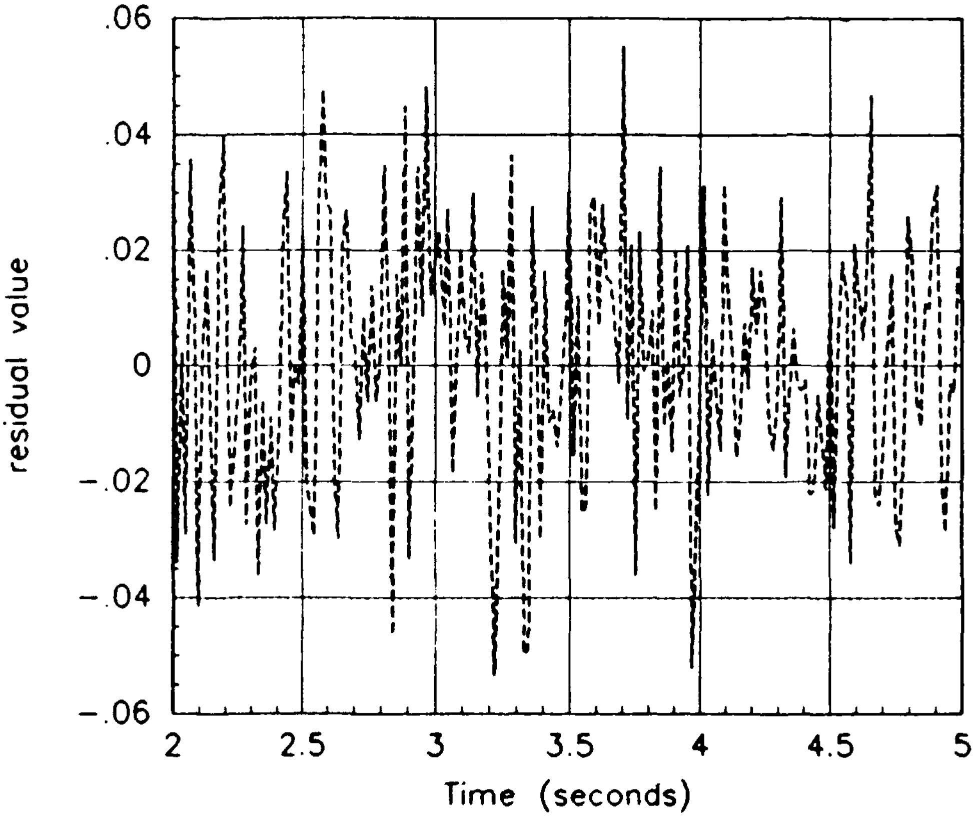

Figure 4.42b Single scalar roll rate residual for the left stabilator filter given a left stabilator failure

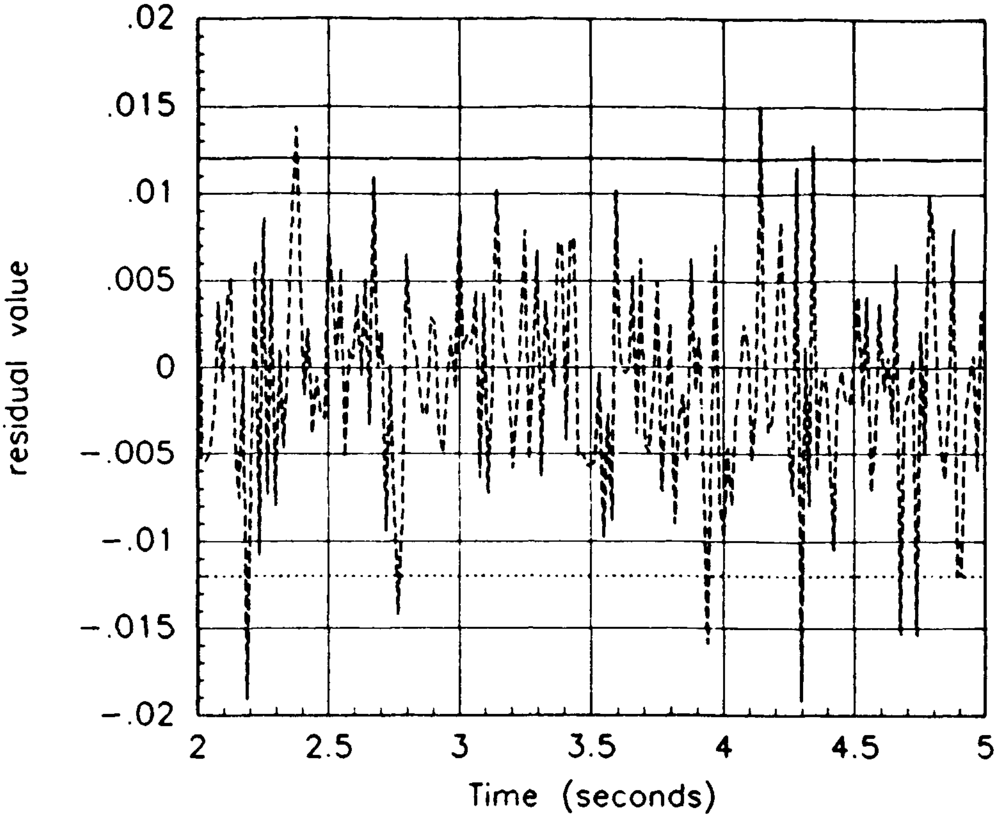

Figure 4.43a Single scalar yaw rate residual for the fully-functional filter given a left stabilator failure

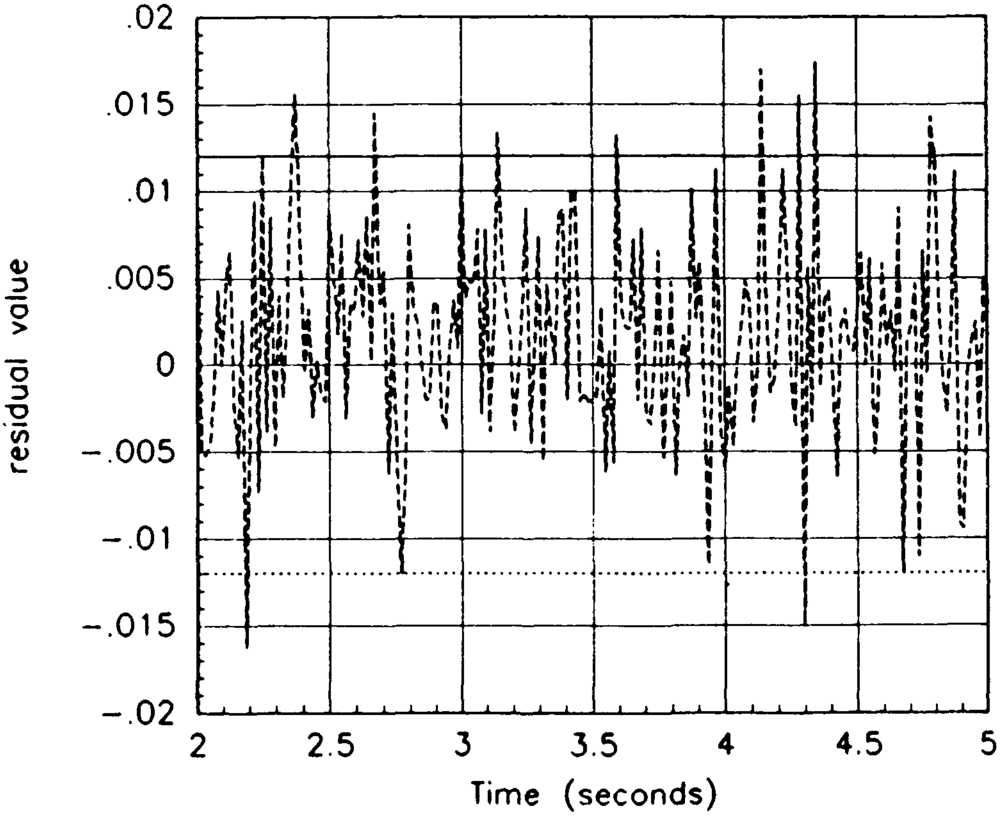

Figure 4.43b Single scalar yaw rate residual for the left stabilator filter given a left stabilator failure

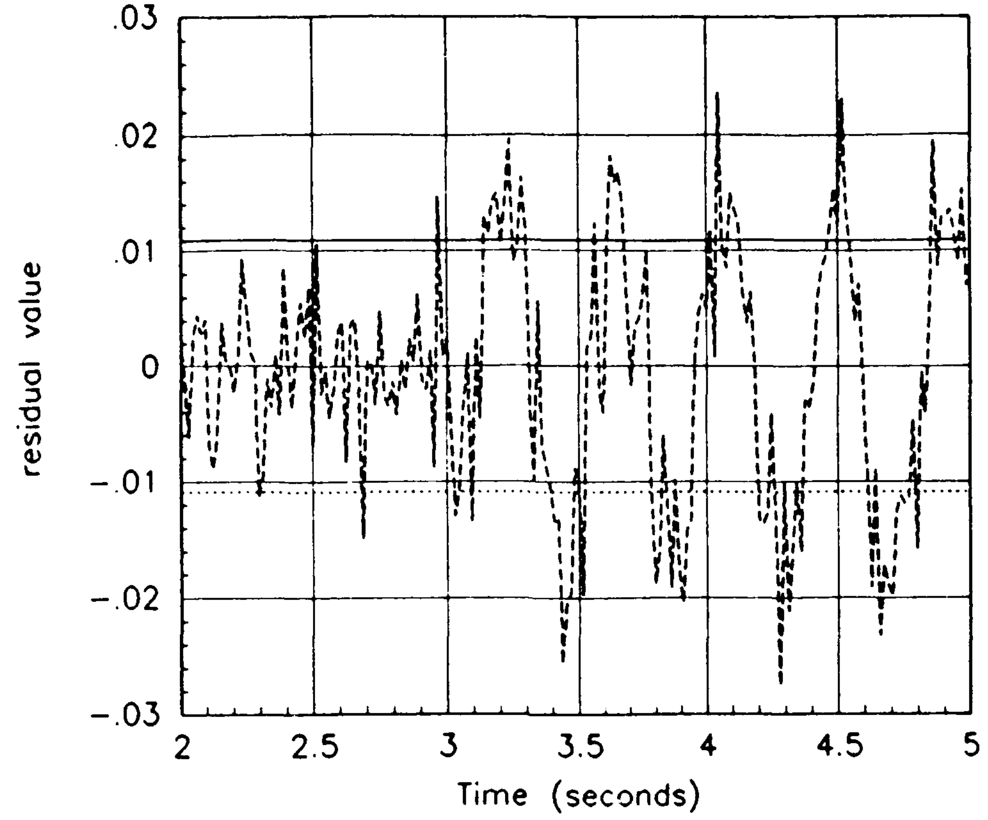

Figure 4.44a Single scalar lateral acceleration residual for the fully-functional filter given a left stabilator failure

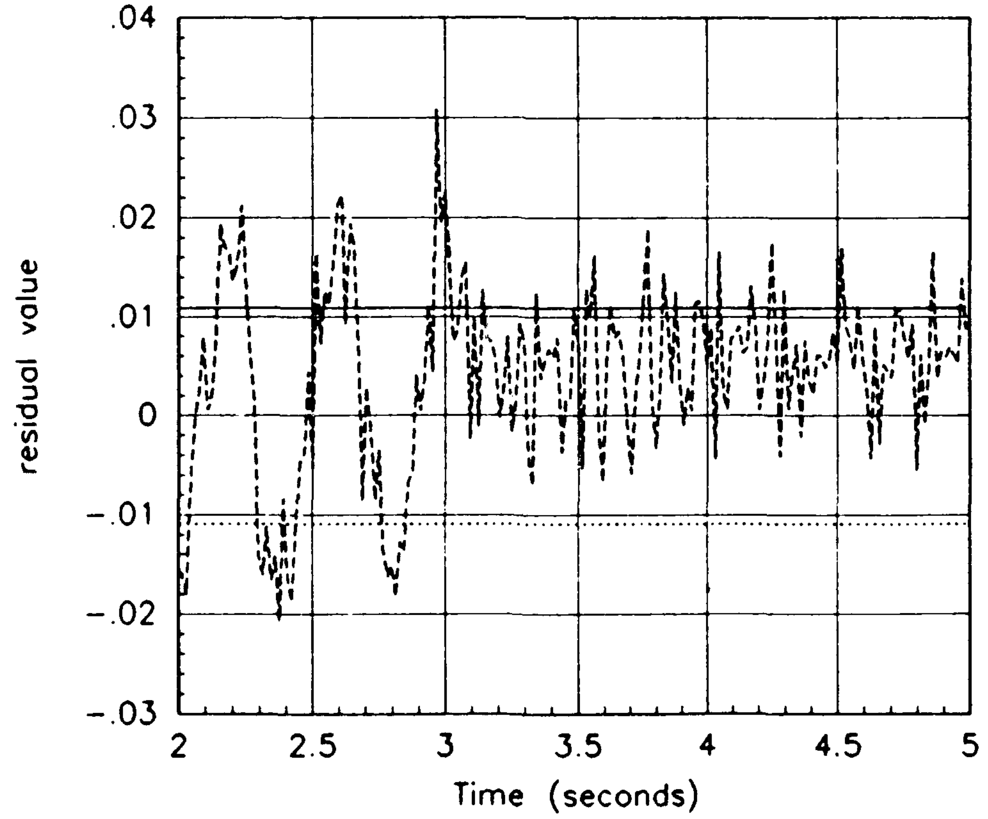

Figure 4.44b Single scalar lateral acceleration residual for the left stabilator filter given a left stabilator failure

4.3 Single Soft Failures

4.3.1 Subliminal Dither Signals (actuators/increased sensor noise)

Two soft failure scenarios were developed for this thesis effort: a 50% reduction in actuator effectiveness (obtained by multiplying individual columns of the ***B*** matrix by 0.5) and a 75% reduction in actuator effectiveness. For the sensors, soft failures were simulated by a doubling in the sensor noise standard deviation (σ) and a multiplier of 3.16 in the sensor noise σ (i.e. a multiplier of 10 on the σ^2). The goal of the soft failure study was to see if a proportionate combination of the fully-functional aircraft elemental filter and the appropriate hard-failure elemental filter could be used to indicate a soft failure.

Figure 4.45 demonstrates a 50% reduced effectiveness left stabilator failure induced at 3.0 seconds. The probability is spread throughout the actuators and sensors rather than just between FF and A1 as desired. Figure 4.46 presents the 75% reduced effectiveness left stabilator failure. The left stabilator filter contains the majority of the probability. From 3.8 to 4.4 seconds, the right stabilator is misidentified by the algorithm. Hypothetically, the fully functional and left stabilator should equally share probability given a 50% left stabilator effectiveness, and proportionately share the probability for the 75% effectiveness cases. The actual value of probability associated with A1 in the 75% effectiveness cases is between 50 and 75% and may best be measured with temporal averaging, but the remainder is not confined to FF. The filters used in this section were tuned for hard failure detection performance, and the tuning was not readjusted to enhance soft failure detection performance. Comparison of Figures 4.45, 4.46, and 4.21 demonstrate the progression of the failure from 50% to 75% to 100% reduced effectiveness. While it is difficult to discern any useful information from Figure 4.45 alone, comparison of Figures 4.46 and 4.21 yield similar characteristics. The ambiguity with the right stabilator present in Figure 4.21 is also present in Figure 4.46.

Figure 4.47 demonstrates a 50% reduced effectiveness right stabilator failure induced at 3.0 seconds. Again, the probability is shared between the actuator and sensor filters. The right stabilator filter displays a number of probability spikes indicative of improving residual characteristics. Figure 4.48 presents the 75% reduction in right stabilator effectiveness. The right stabilator filter contains the majority of the probability trace, as desired. Comparison of Figures 4.47, 4.48, and 4.22 demonstrate the progression of the failure as the actuator effectiveness is reduced from 50% to 75% to 100%. Again, Figure 4.47 provides little validation information. Comparison of Figures 4.48 and 4.22 yield similar characteristics. The spikes present in Figure 4.22 can be found in Figure 4.48.

Figure 4.49 presents a 50% reduced effectiveness left flaperon failure. The flaperon failure produces results that are consistent with theoretical expectations, a sharing of probability data. This sharing of probability produces the desired blending of control outputs. Although the sharing is not equal, the bounding value is close to 50% reduced effectiveness. Figure 4.50 demonstrates the 75% reduction in left flaperon effectiveness. The left flaperon filter contains the probability with some minor ambiguity with the right flaperon.

Figure 4.51 demonstrates a 50% reduced effectiveness right flaperon failure. The right flaperon failure presents improved results over those given in Figure 4.49. Figure 4.52 presents the 75% reduction. The probability trace is very clean for the right stabilator, with the exception of ambiguity at 3.4 to 3.6 seconds and the small spike in the angle of attack elemental filter prior to the 6 second pulse (as previously noted).

Figure 4.53 presents a 50% reduced effectiveness rudder failure. The rudder and fully functional filter share the probability after the insertion of the soft rudder failure. Figure 4.54 demonstrates a 75% reduction. The rudder trace in this case is very similar to the hard rudder failure.

Figure 4.55 demonstrates the 1σ increased noise failure for the velocity sensor. The results indicate the velocity sensor was not identified as the failed sensor. Probability spikes exist in many of the actuator filters. The pitch rate filter also contains a small but consistent portion of the probability. Increasing the noise content to 3.16σ as shown in Figure 4.56 does not improve the failure identifiability. The velocity sensor filter's scalar velocity residual indicates a divergent filter. At this point, a few options are available. First, we could improve the velocity model to match the real world better. A second option might be to reset or restart the filter at a regular interval in an attempt to control its divergence. The interval at which the filter must be restarted depends upon the divergence rate and the performance degradation. Restarting a filter implies reinitializing the divergent elemental Kalman filters to agree with the fully functional Kalman filter values (if it is nondivergent), or with a suitable linear combination of nondivergent filter state estimates. This technique forces the filter residuals to nearly zero at each reintialization. Such a modification to the algorithm is not included in this thesis effort.

The angle of attack soft failure provides better results. Figure 4.57 demonstrates the 1σ increased noise scenario. Other than the fully functional filter, the angle of attack filter contains a large portion of the probability. Figure 4.58 presents the 3.16σ scenario. The flaperon, rudder, pitch rate, roll rate, and yaw rate contain small but consistent portions of the probability trace. The angle of attack filter contains a larger portion of the probability and more probability spikes.

Figure 4.59 demonstrates the 1σ increased noise for the pitch rate sensor. Two small probability spikes occur in the pitch rate elemental filter. Increasing the noise to 3.16σ, Figure 4.60, slightly improves the probability trace. The aircraft is controllable throughout the increased noise scenario.

Figure 4.61 presents the 1σ increased noise for the normal acceleration sensor. While the probability is spread throughout the flaperons, rudder, angle of attack, pitch rate, normal acceleration, roll rate, and yaw rate filters, the normal acceleration has an appropriate spiking phenomenon. This spiking phenomenon occurs throughout the failure set and experimental results have shown that this phenomenon is indicative of a filter with the correct hypothesized failure. Figure 4.62 presents the 3.16σ increased noise for the normal acceleration sensor. All of the filters experience increased probabilities except the velocity sensor. The normal acceleration contains more probability spikes, indicating improved failure detection capability from the 1σ case.

Figure 4.63 demonstrates the 1σ increased sensor noise for the roll rate sensor. The roll rate probability trace indicates some small spiking in the roll rate filter. Increasing the noise to 3.16σ, Figure 4.64, yields more spiking in the roll rate filter probability trace. A small amount of probability occurs in the pitch rate filter.

Figure 4.65 presents the 1σ increased sensor noise for the yaw rate sensor. The yaw rate elemental filter picks up the probability in the last 0.4 seconds of the run. Figure 4.66 demonstrates the 3.16σ sensor noise increase. A significantly larger portion of probability is evident within the filter probability trace.

Figure 4.67 demonstrates the 1σ increased sensor noise for the lateral acceleration sensor. All of the actuator elemental filters contain a portion of the probability value. A few small spikes occur in the lateral acceleration elemental filter. Figure 4.68 presents the 3.16σ increased sensor noise scenario. All of the actuators contain portions of the probability. The sensor traces reveal small portions of the probability in the angle of attack, pitch rate, normal acceleration, roll rate, yaw rate, and lateral acceleration. The lateral acceleration sensor contains the spiking phenomena discussed earlier. The algorithm has great difficulty in identifying this failure scenario, possibly due to the fact that all surfaces can produce a lateral acceleration.

Trying to detect a 1σ increase in sensor noise is consistently difficult with the filters tuned as originally for hard failure detection performance, but using only fully-functional aircraft and hard failure hypotheses for the basis of designing elemental filters does yield capacity to detect "soft failures" of the form of an order of magnitude increase in sensor noise variance, or 50 - 75% loss of effectiveness of actuators.

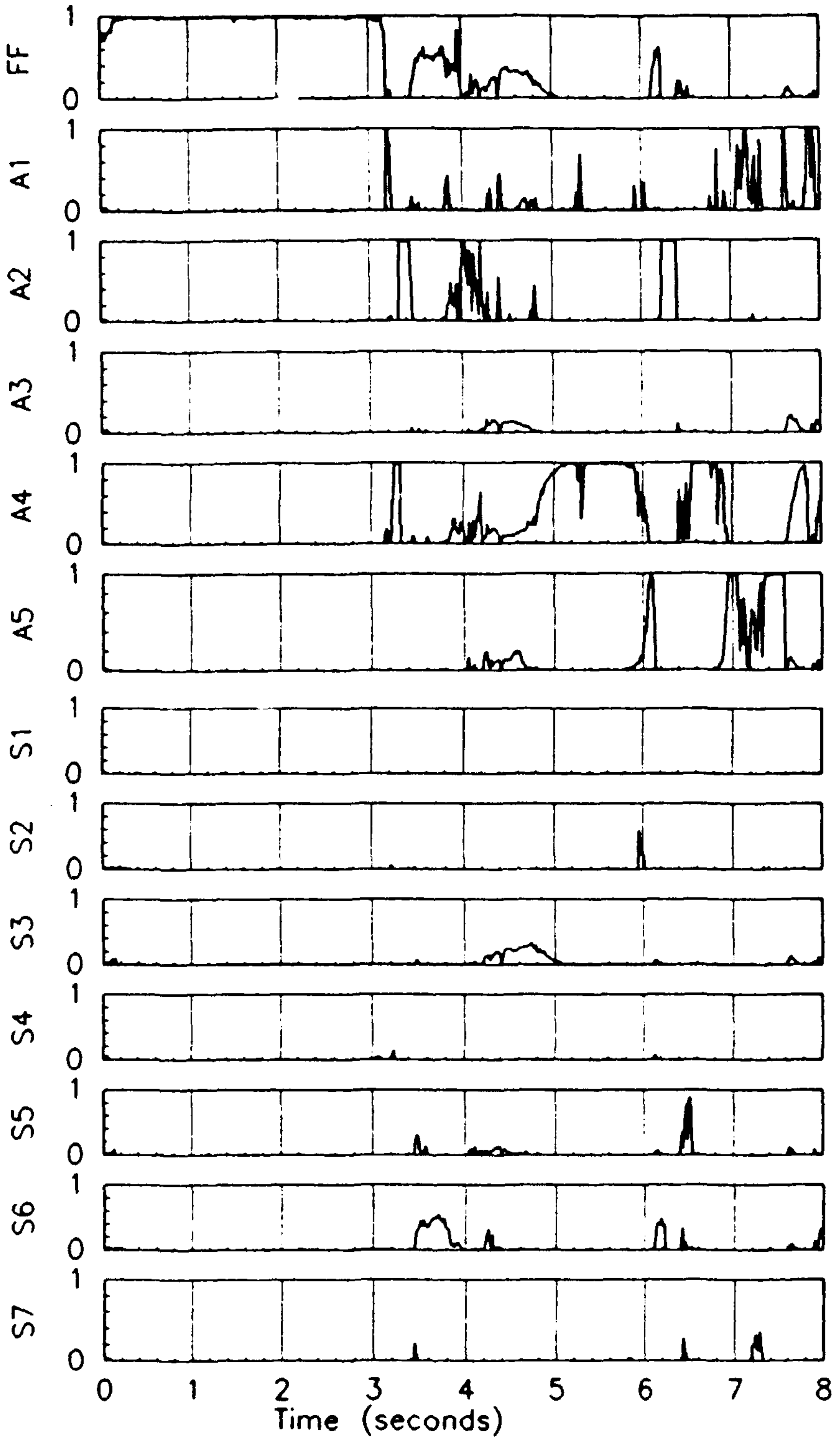

Figure 4.45 Probabilities for a 50% reduced effectiveness left stabilator failure using a subliminal dither pulse

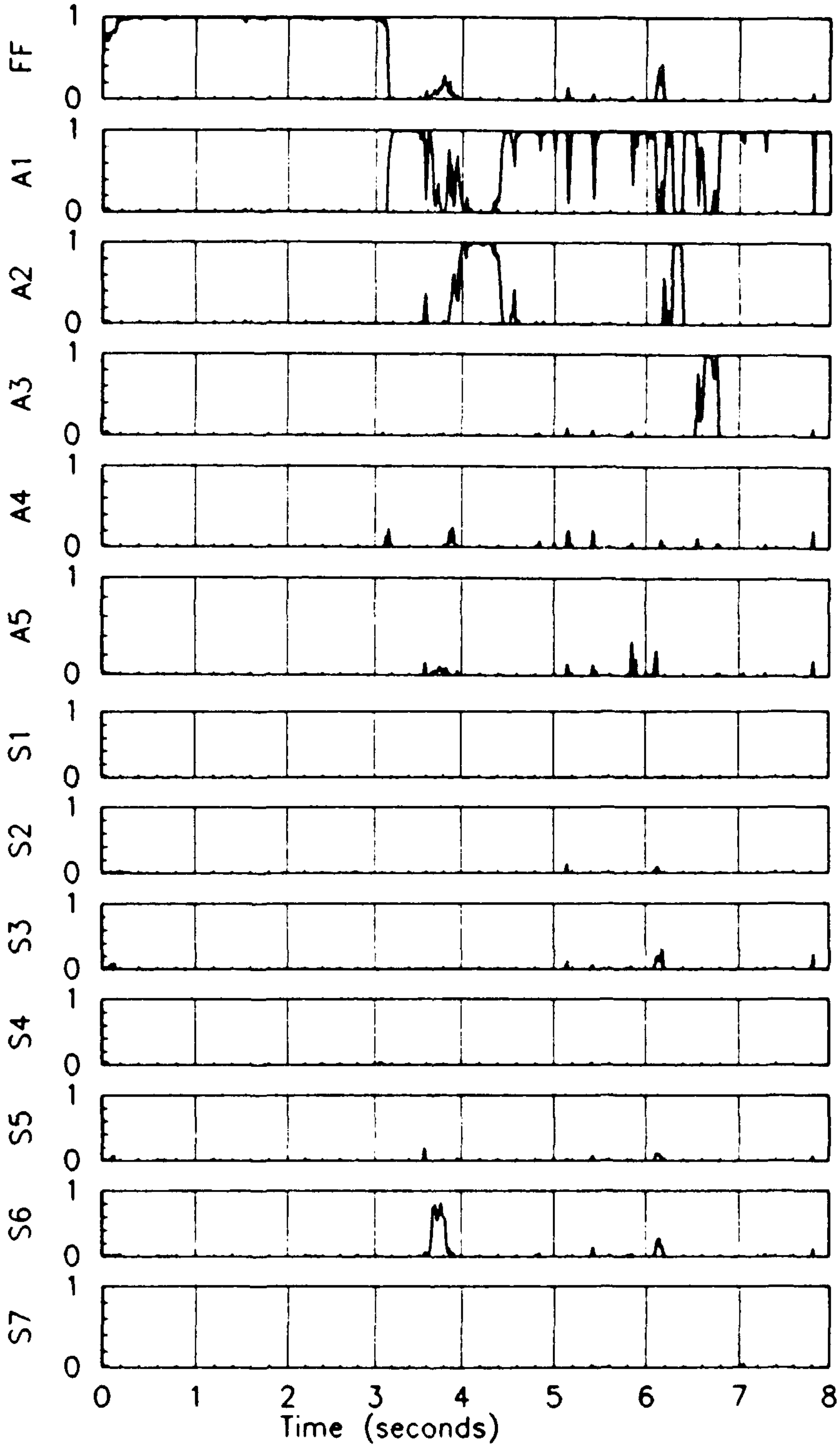

Figure 4.46 Probabilities for a 75% reduced effectiveness left stabilator failure using a subliminal dither pulse

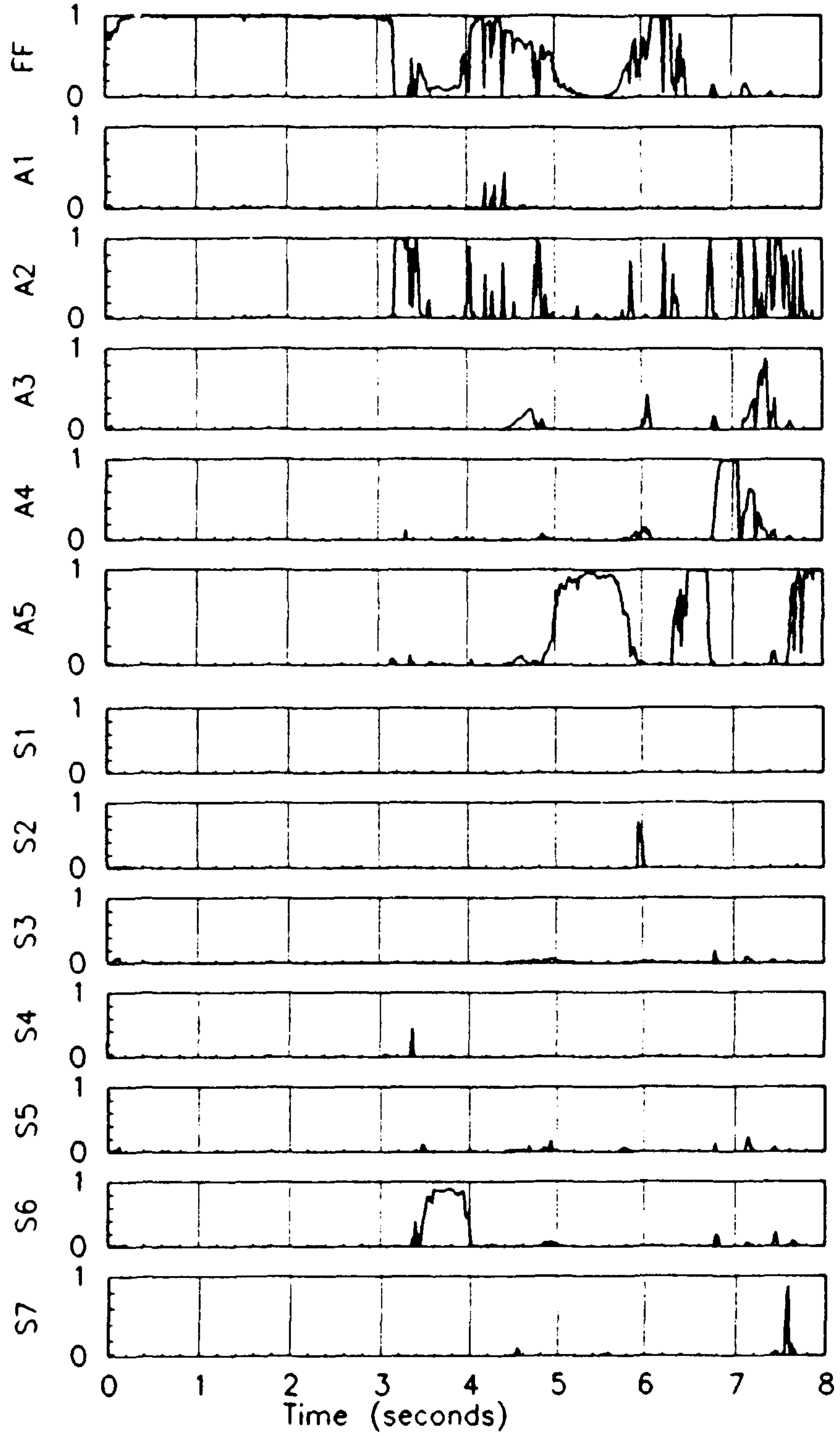

Figure 4.47 Probabilities for a 50% reduced effectiveness right stabilator failure using a subliminal dither pulse

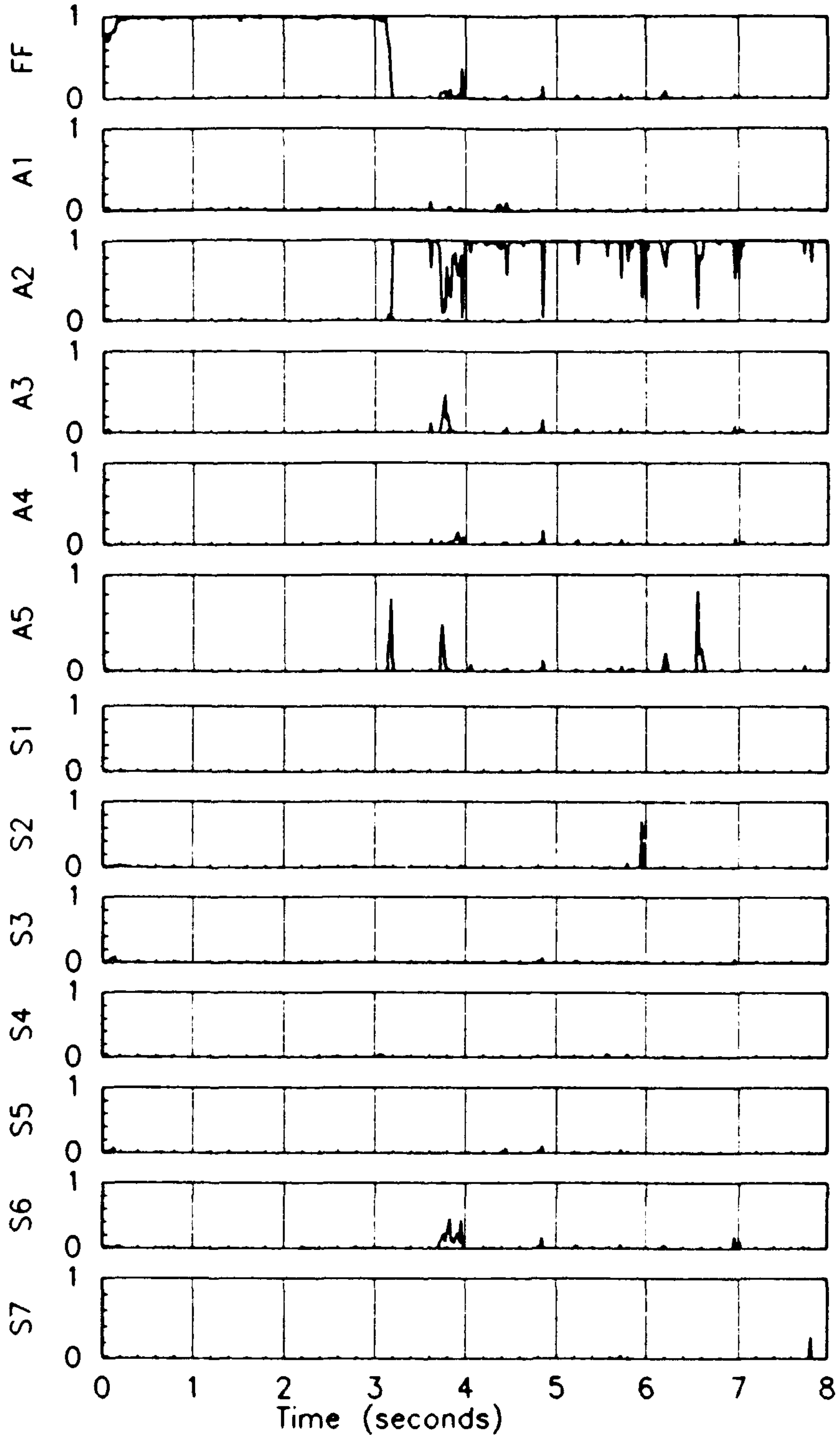

Figure 4.48 Probabilities for a 75% reduced effectiveness right stabilator failure using a subliminal dither pulse

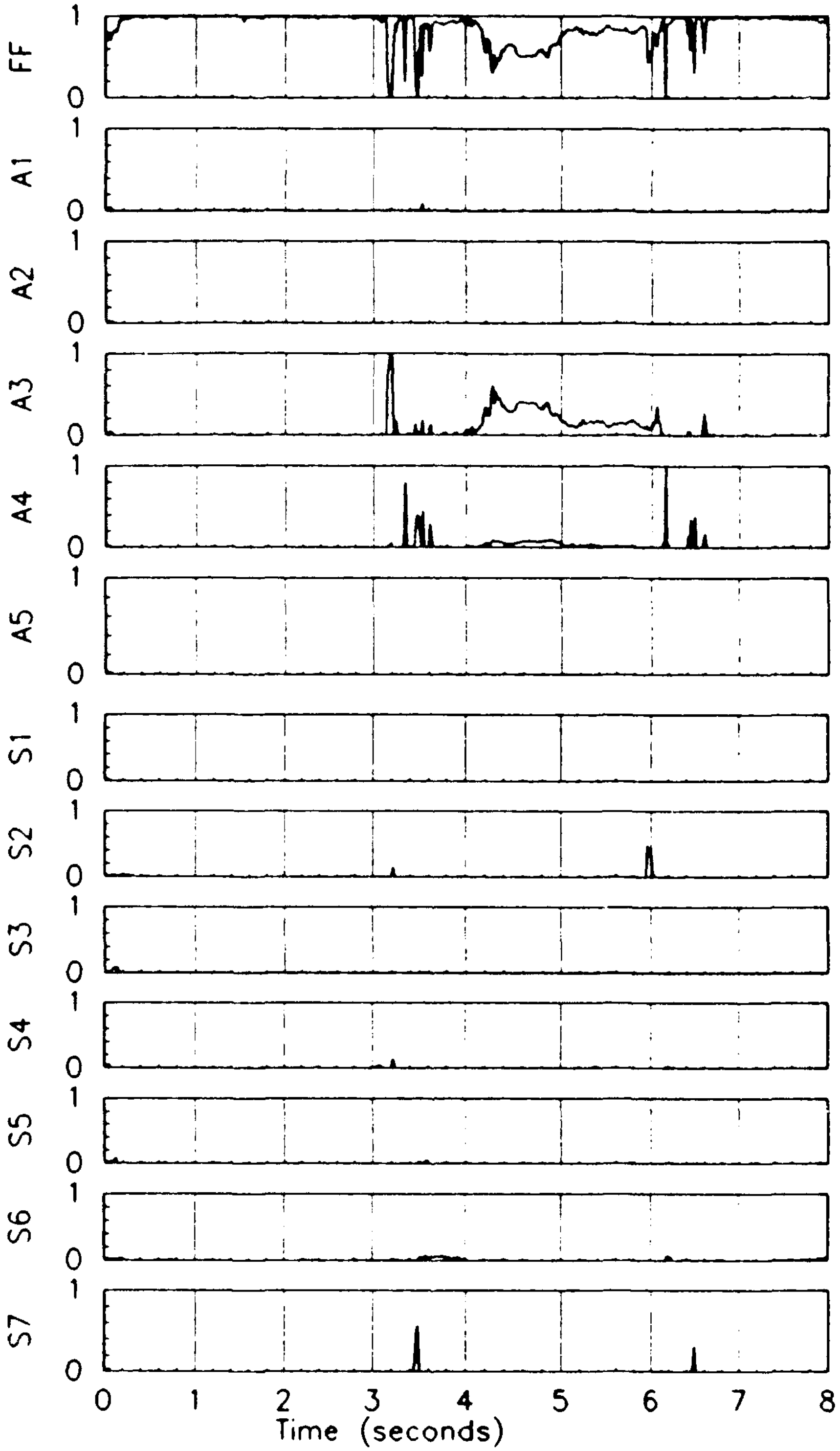

Figure 4.49 Probabilities for a 50% reduced effectiveness left flaperon failure using a subliminal dither pulse

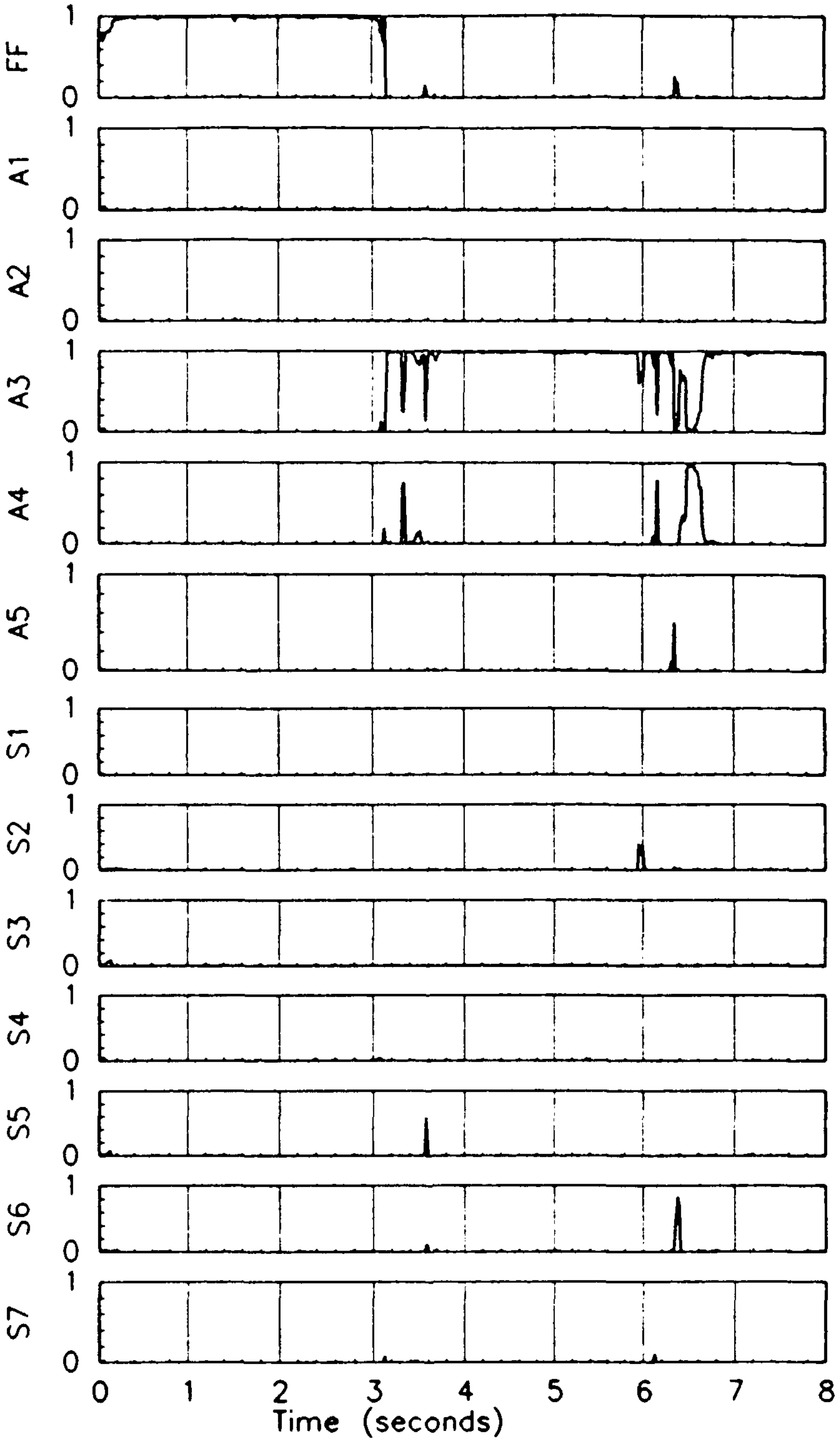

Figure 4.50 Probabilities for a 75% reduced effectiveness left flaperon failure using a subliminal dither pulse

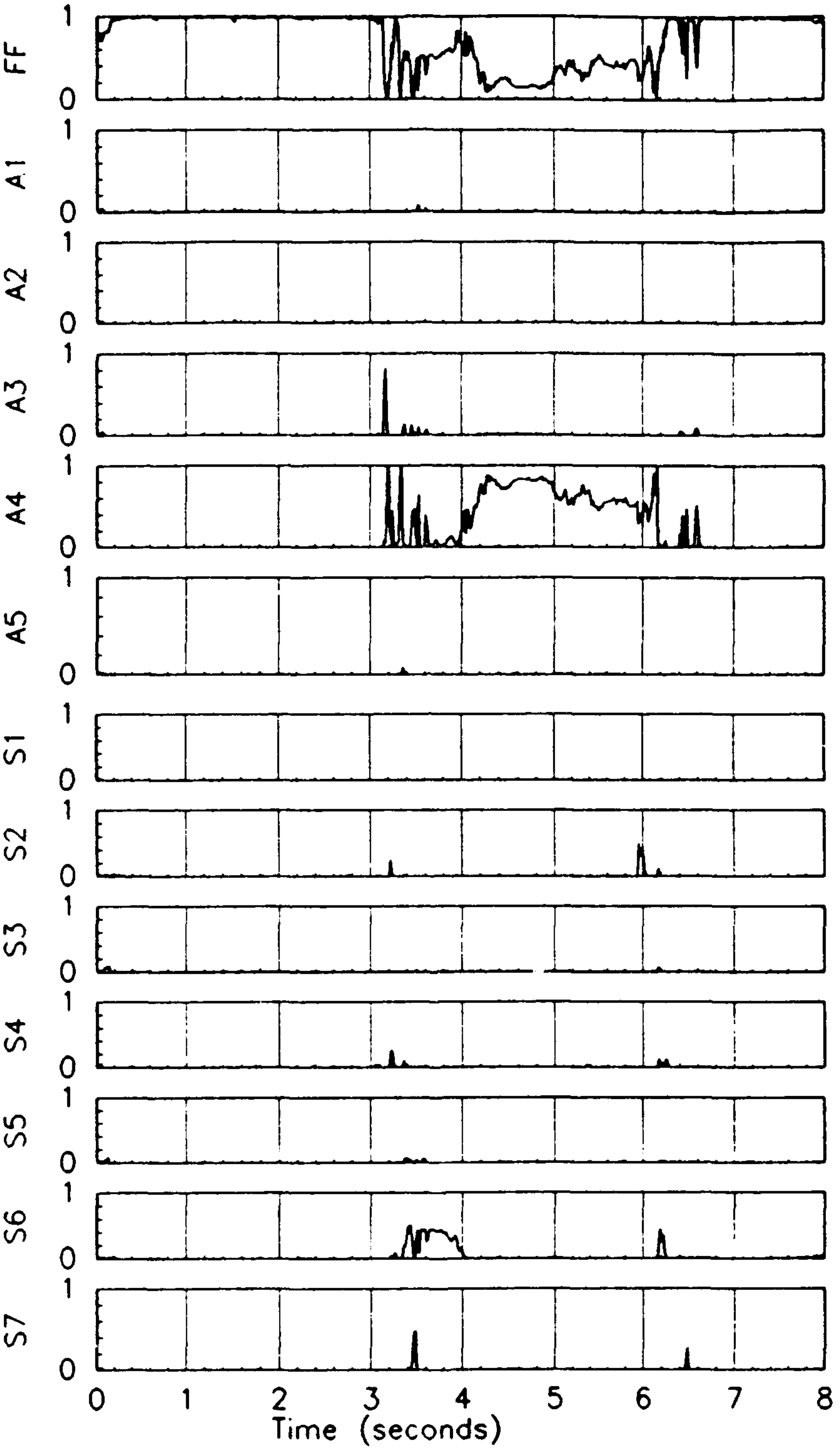

Figure 4.51 Probabilities for a 50% reduced effectiveness right flaperon failure using a subliminal dither pulse

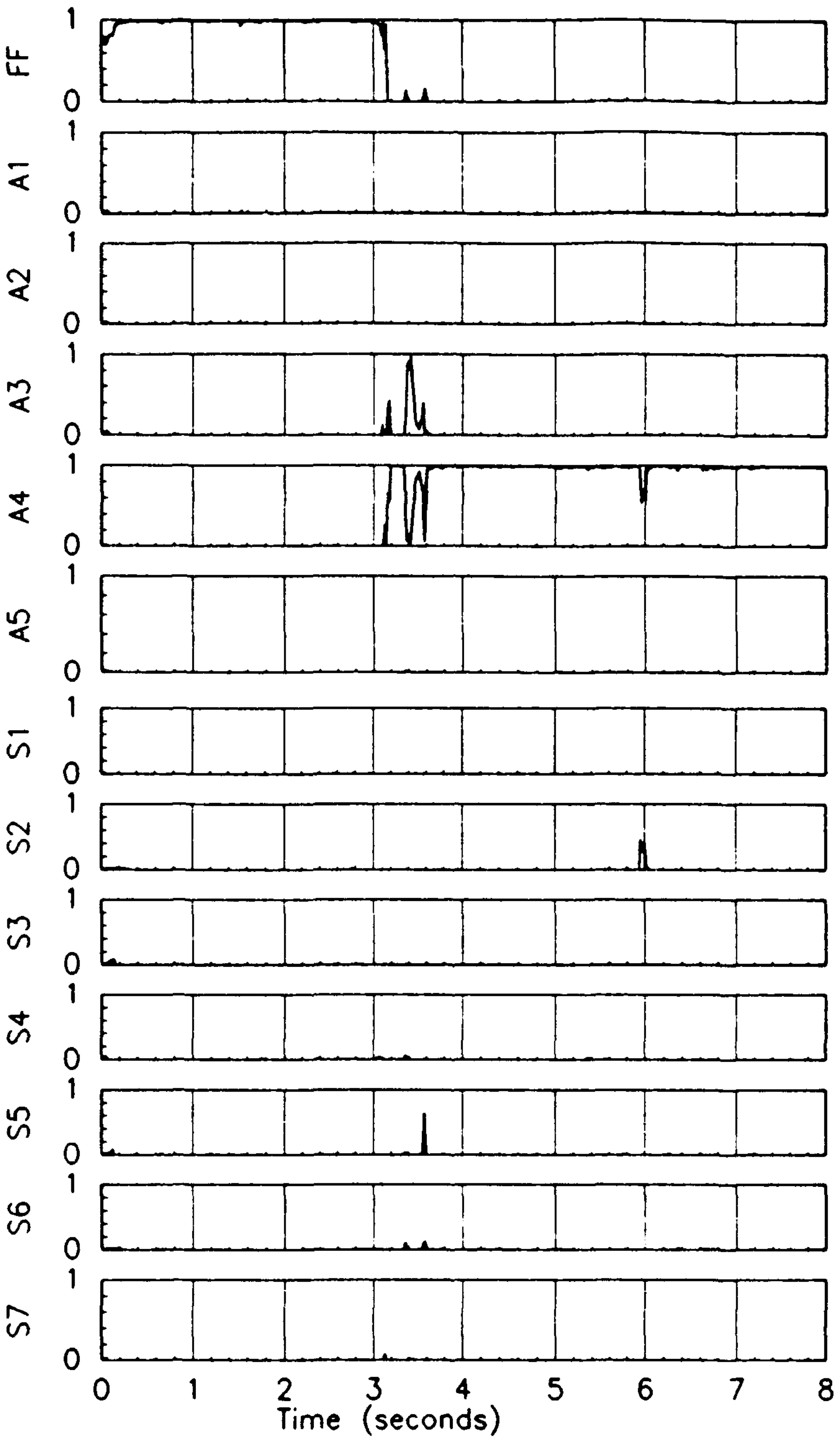

Figure 4.52 Probabilities for a 75% reduced effectiveness right flaperon failure using a subliminal dither pulse

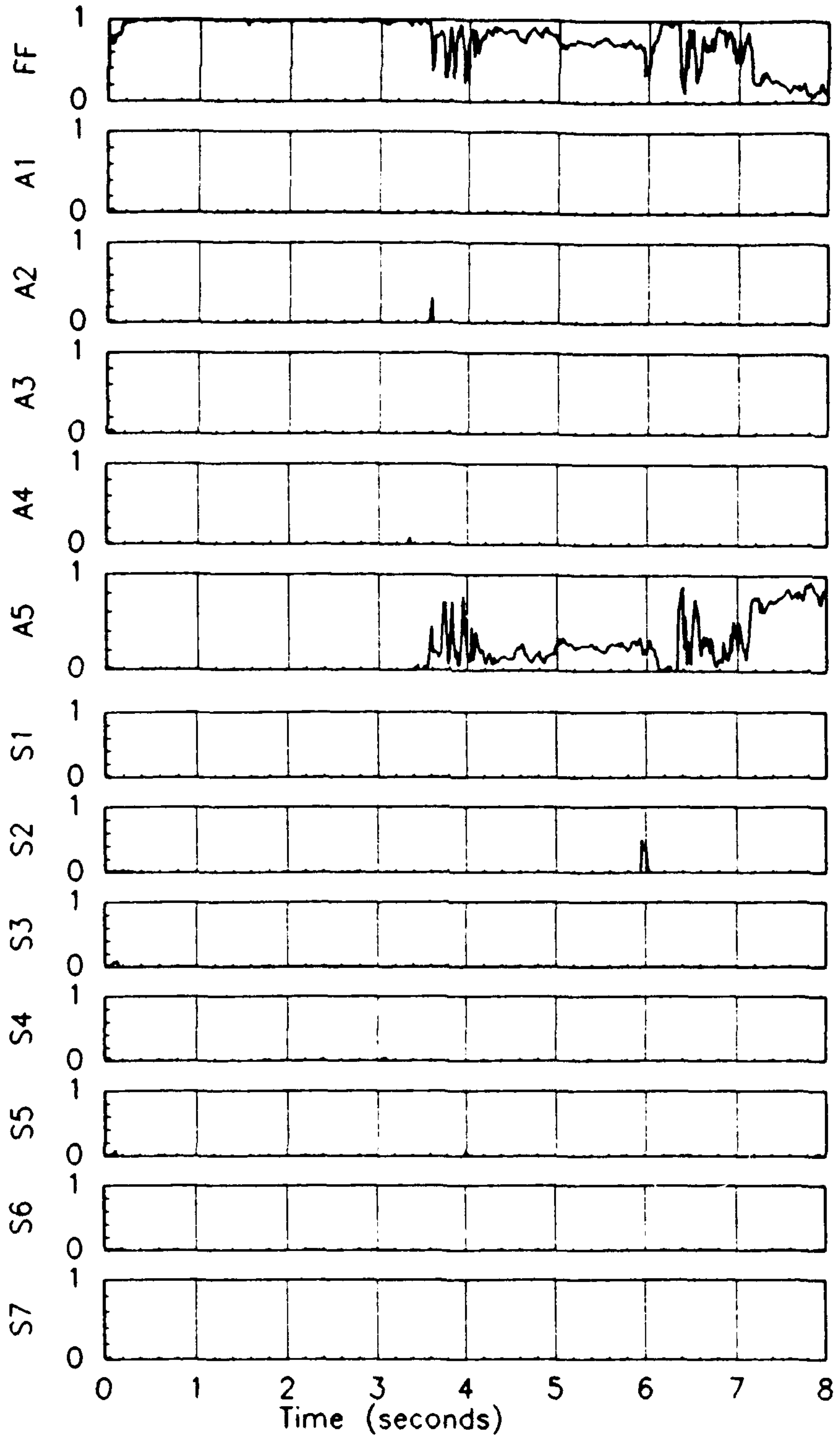

Figure 4.53 Probabilities for a 50% reduced effectiveness rudder failure using a subliminal dither pulse

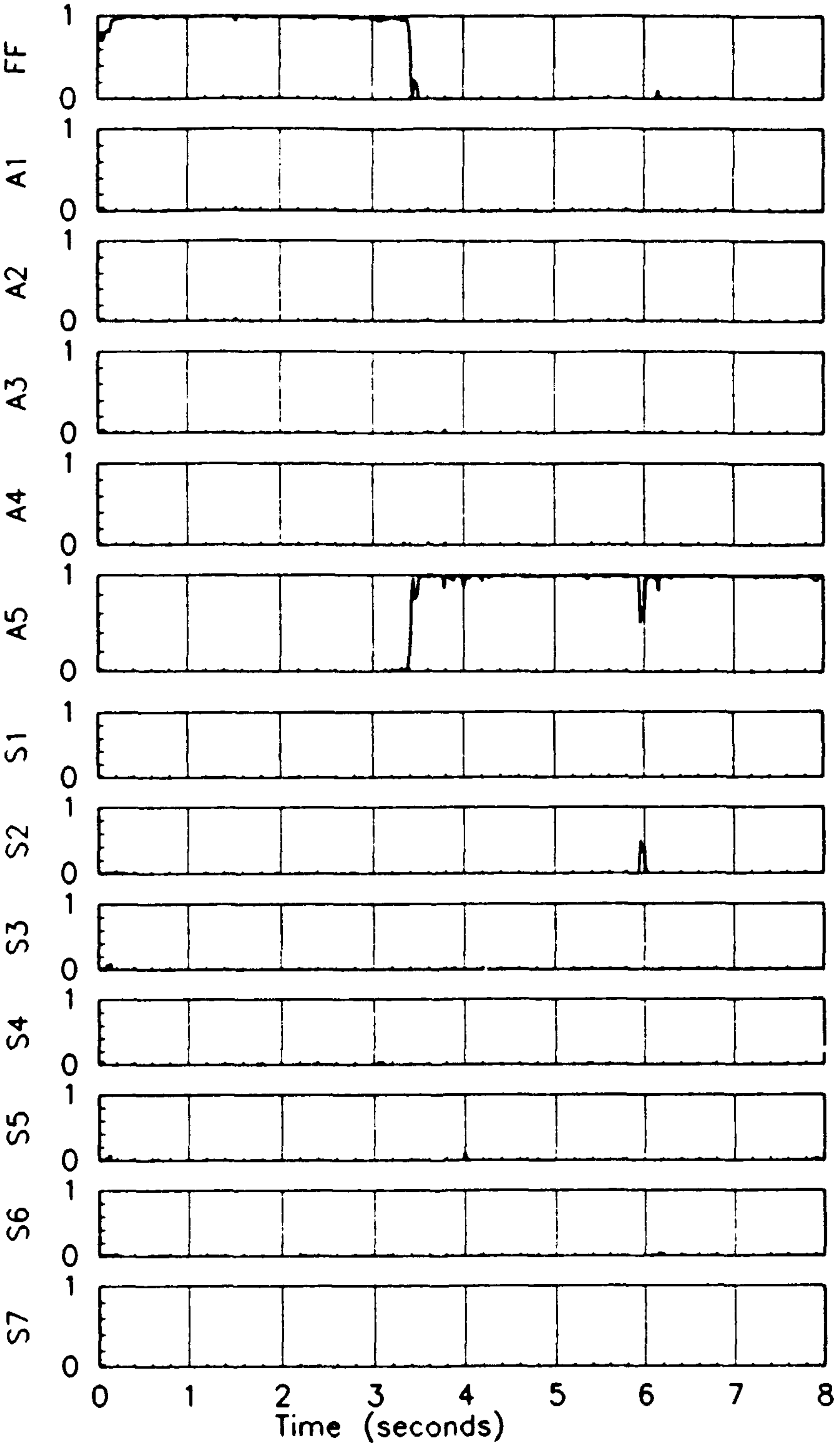

Figure 4.54 Probabilities for a 75% reduced effectiveness rudder failure using a subliminal dither pulse

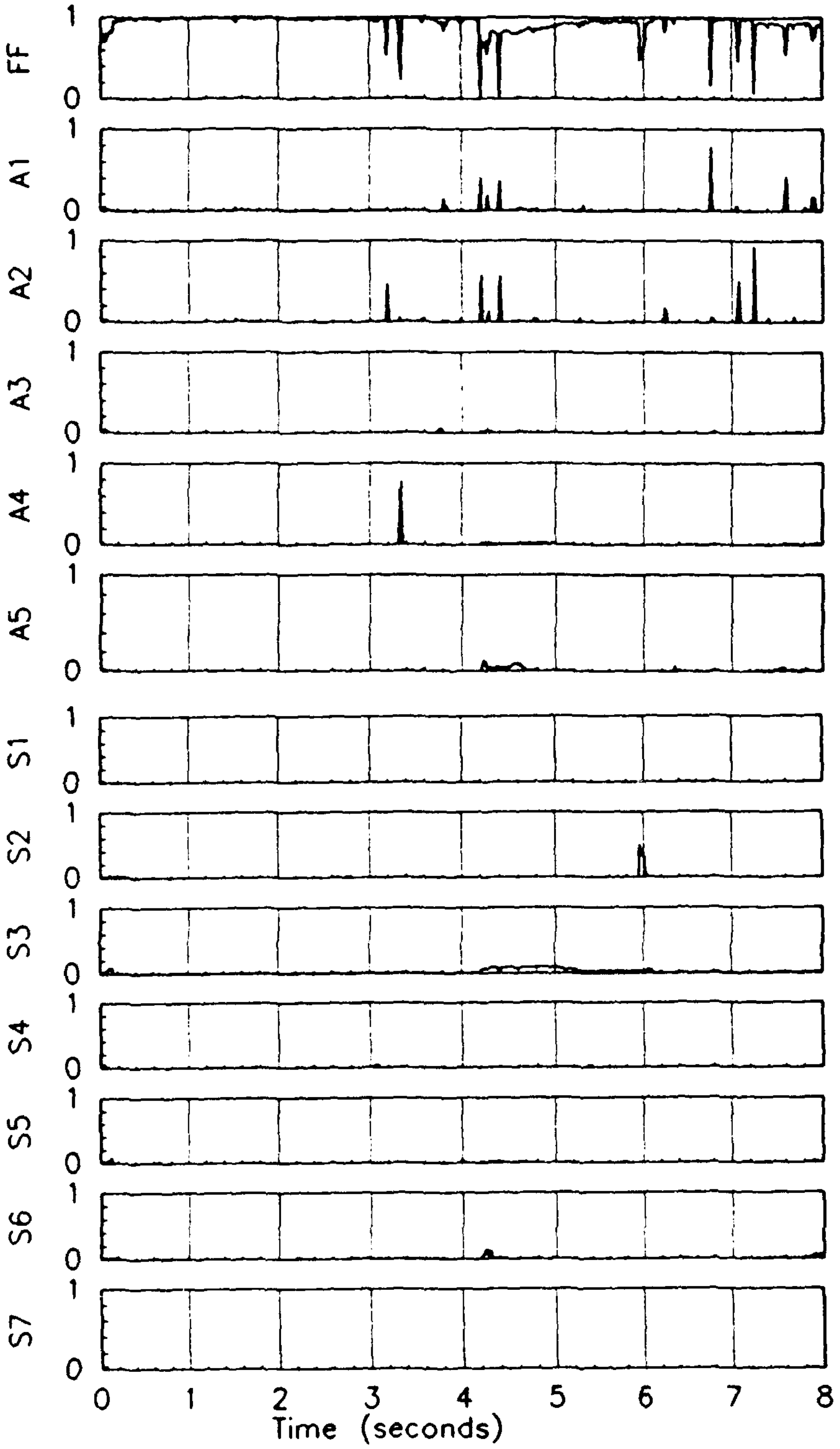

Figure 4.55 Probabilities for a 1σ increased sensor noise failure for the velocity sensor

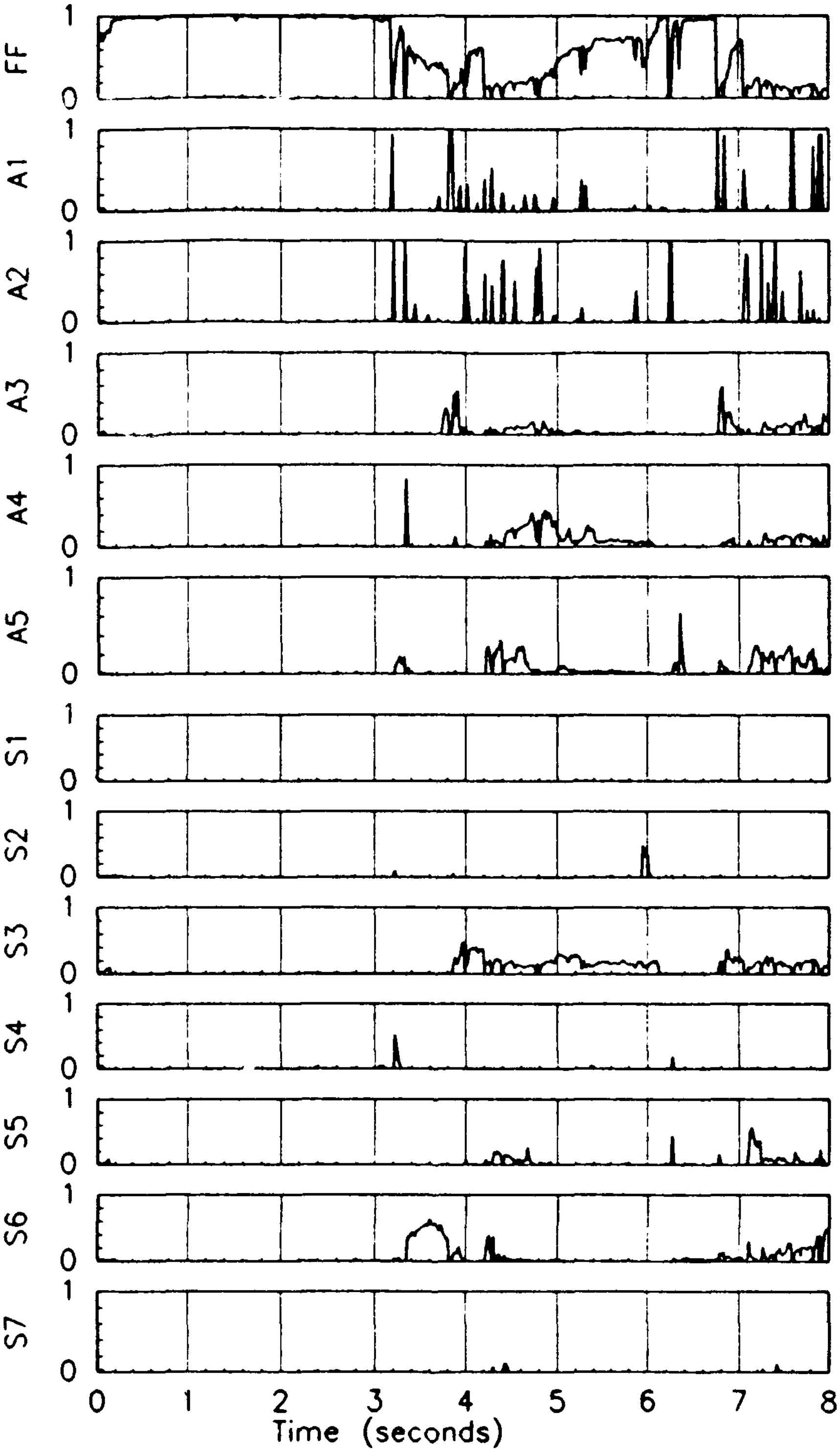

Figure 4.56 Probabilities for a 3.16σ increased sensor noise failure for the velocity sensor

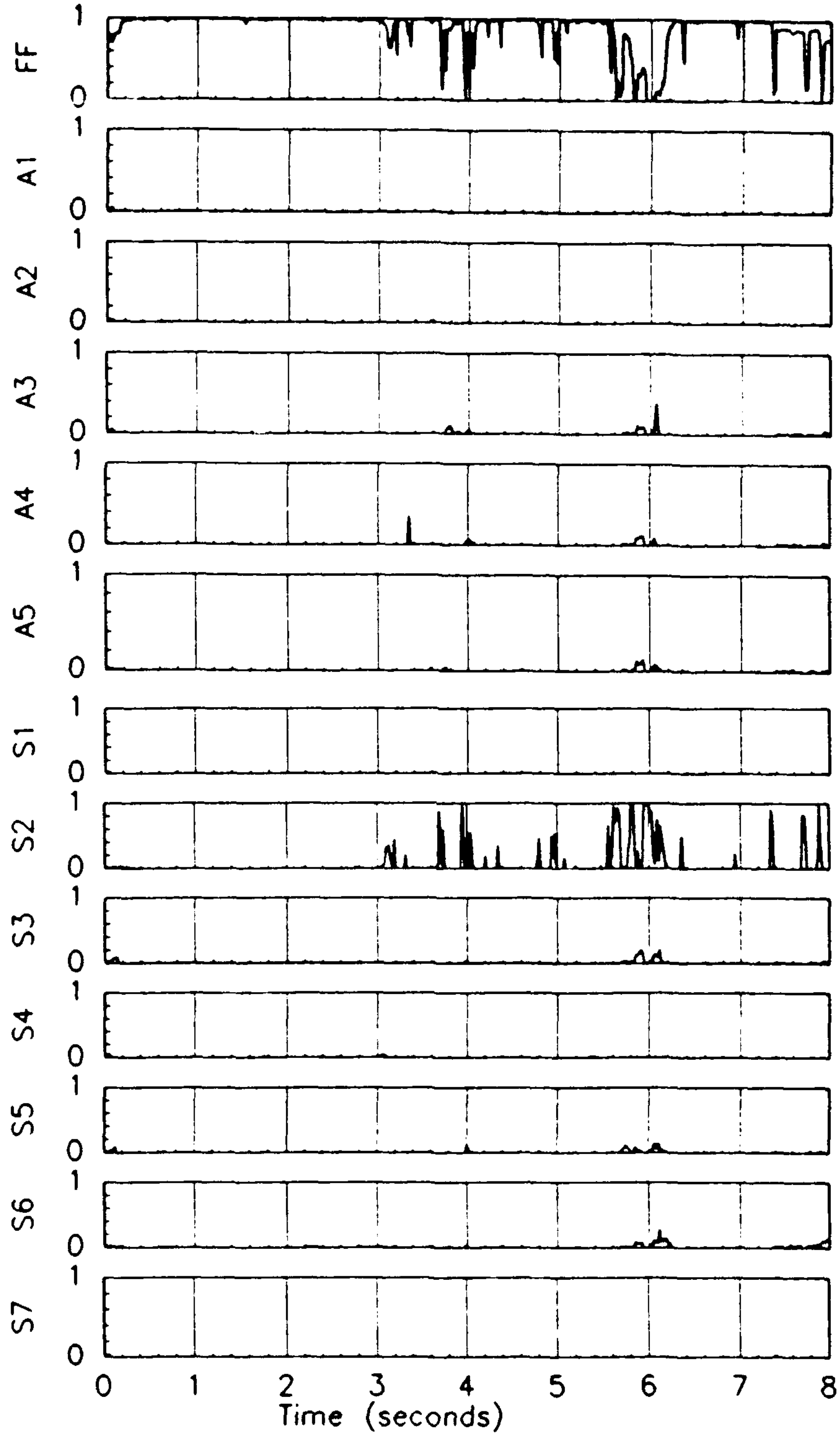

Figure 4.57 Probabilities for a 1σ increased sensor noise failure for the angle of attack sensor

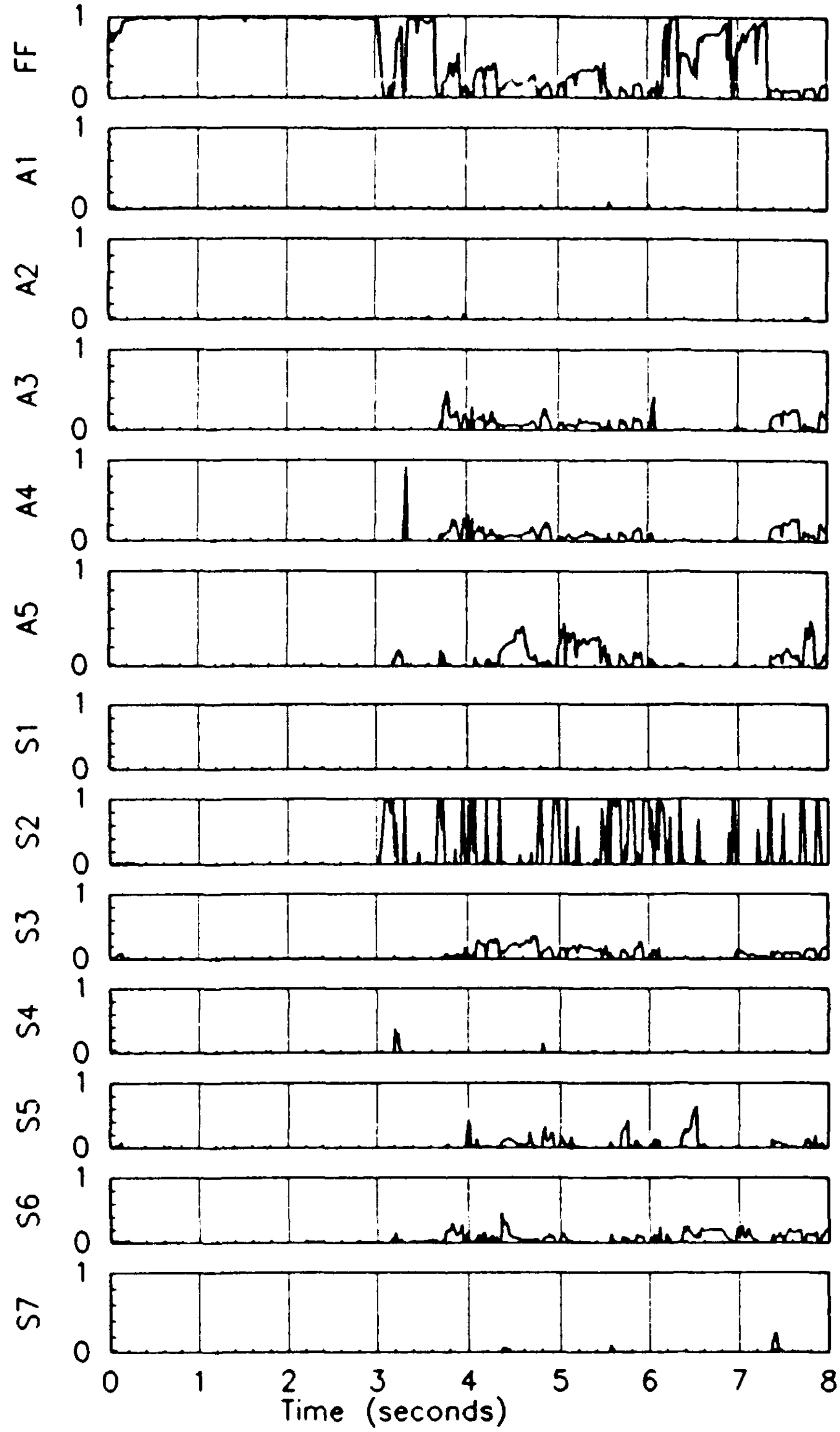

Figure 4.58 Probabilities for a 3.16σ increased sensor noise failure for the angle of attack sensor

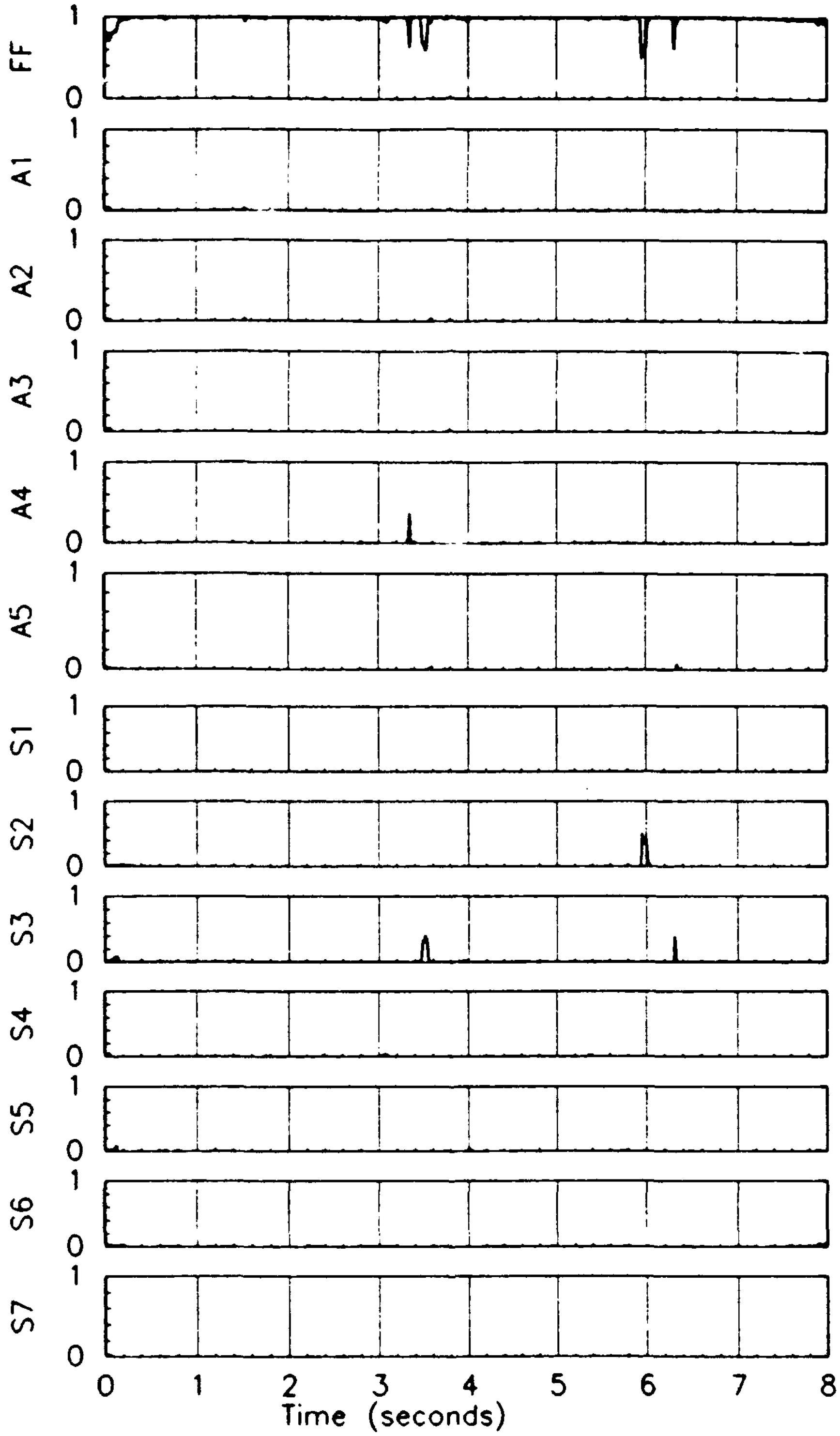

Figure 4.59 Probabilities for a 1σ increased sensor noise failure for the pitch rate sensor

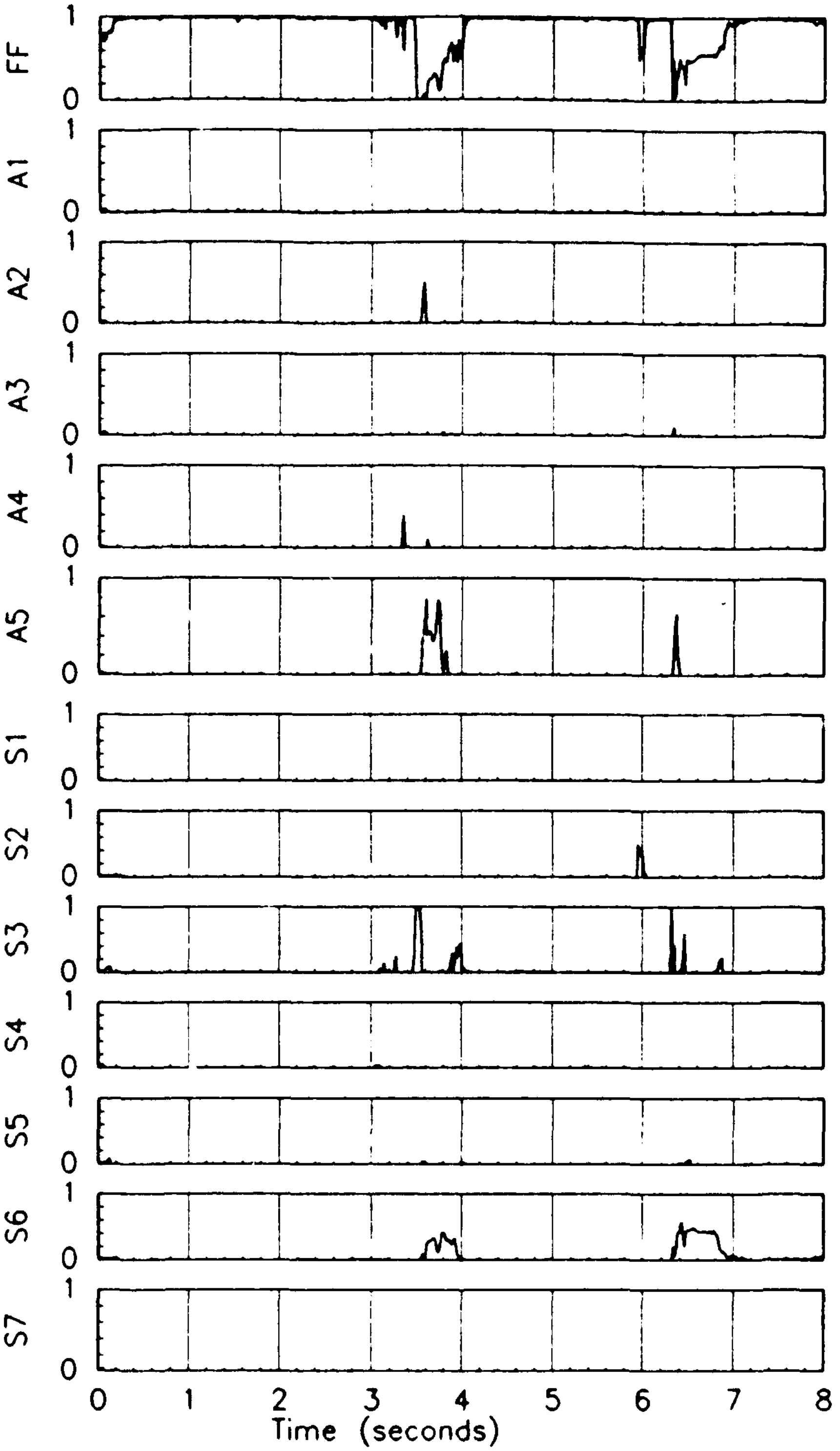

Figure 4.60 Probabilities for a 3.16σ increased sensor noise failure for the pitch rate sensor

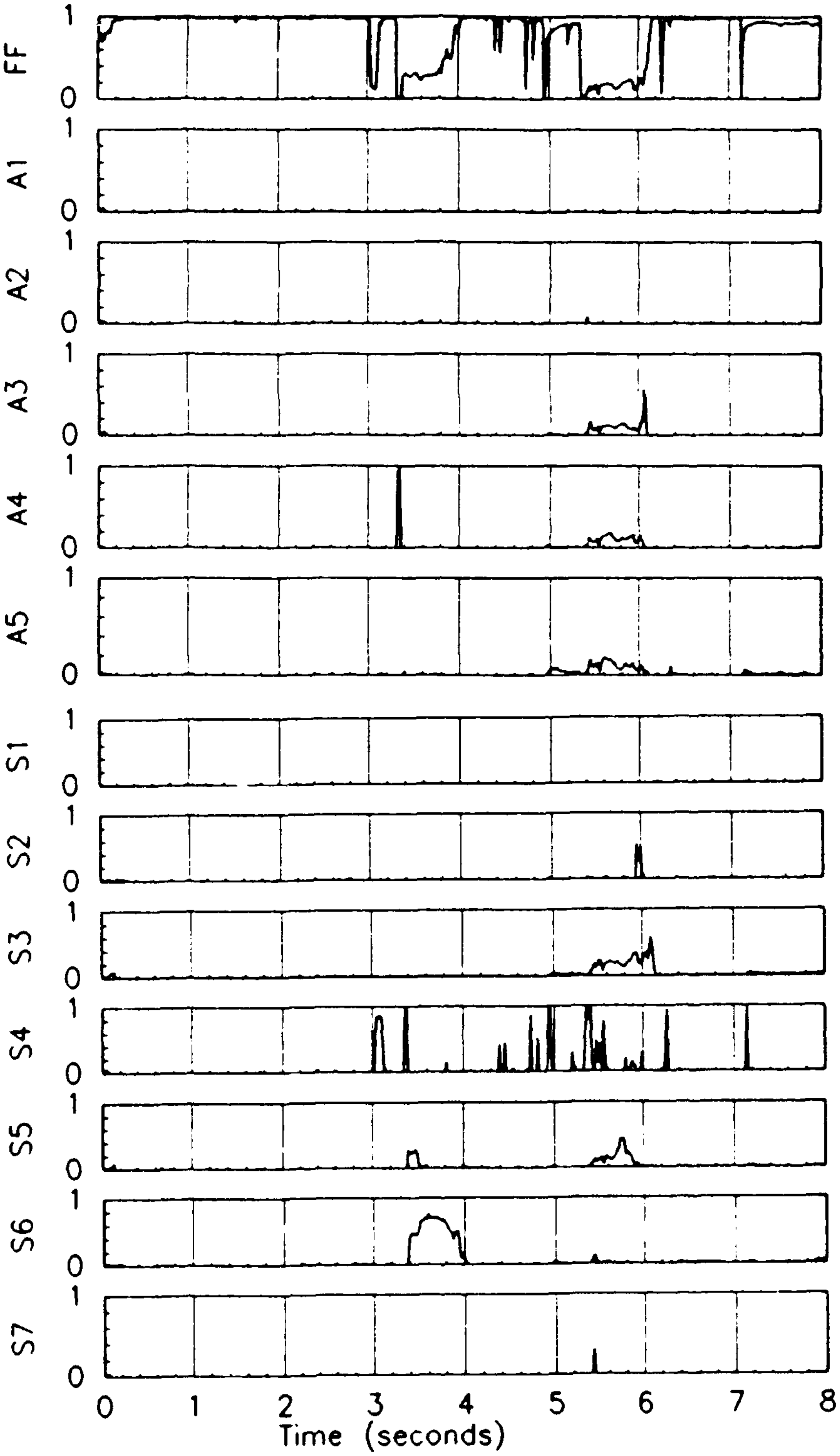

Figure 4.61 Probabilities for a 1σ increased sensor noise failure for the normal acceleration sensor

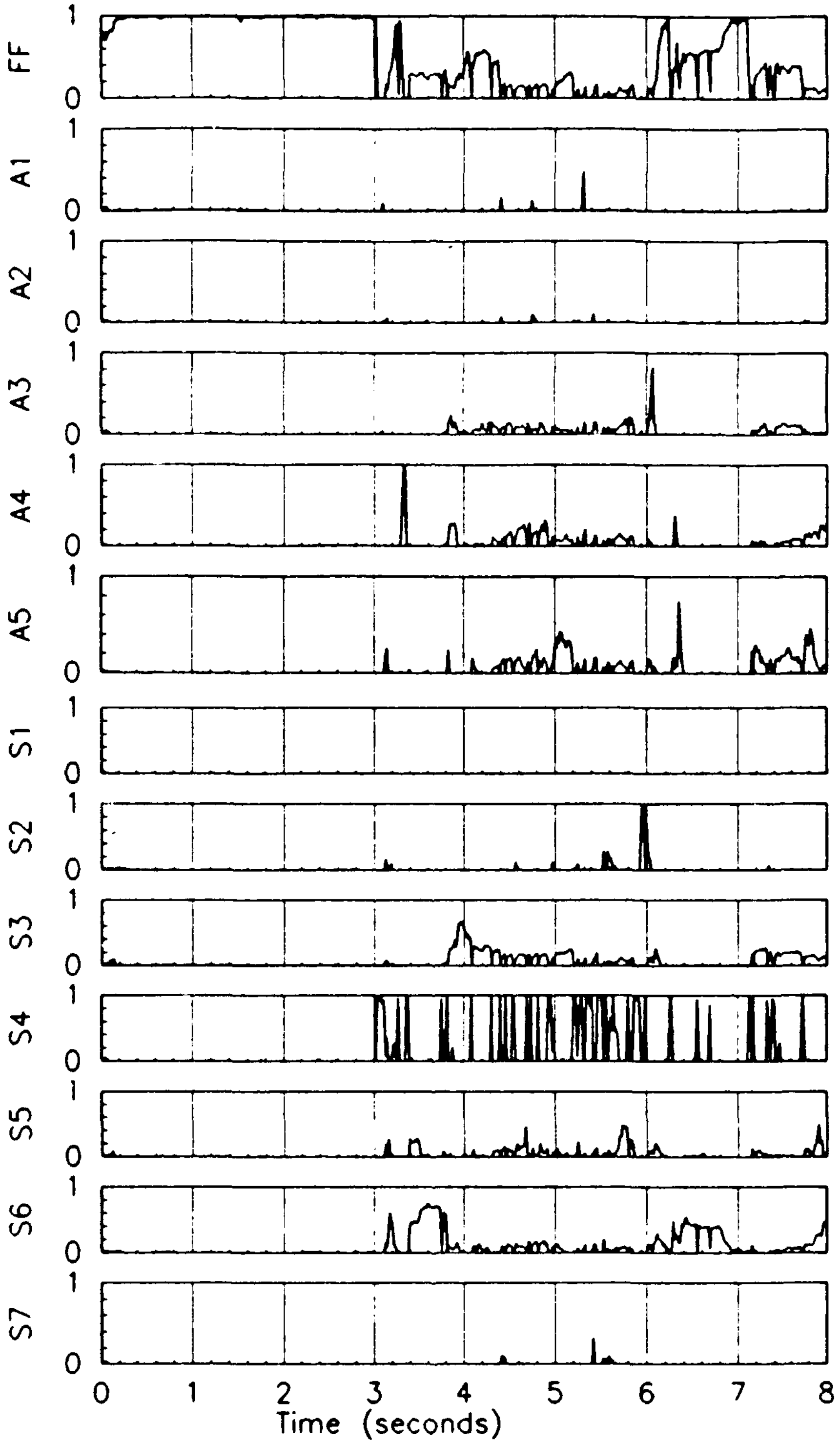

Figure 4.62 Probabilities for a 3.16σ increased sensor noise failure for the normal acceleration sensor

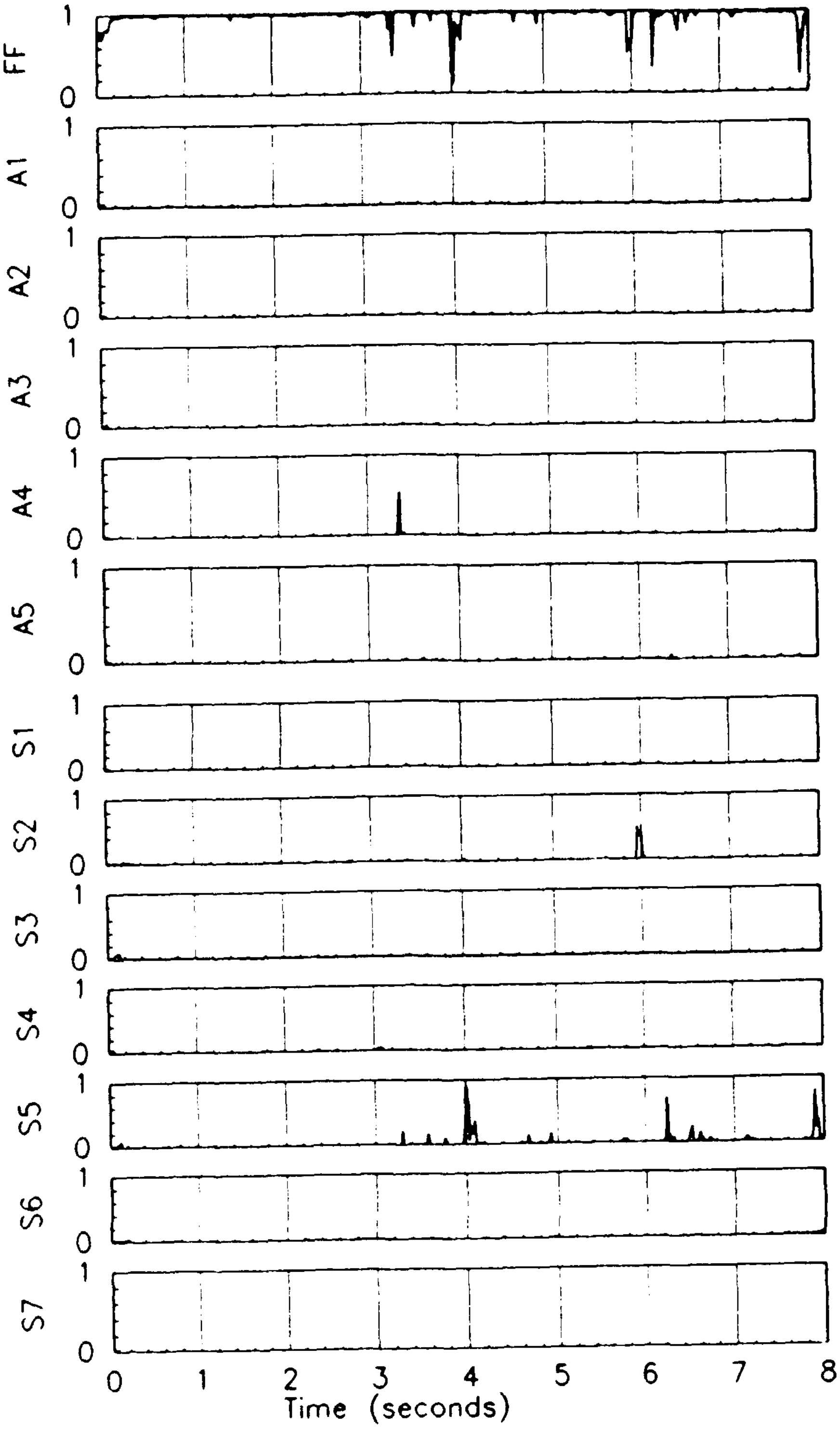

Figure 4.63 Probabilities for a 1σ increased sensor noise failure for the roll rate sensor

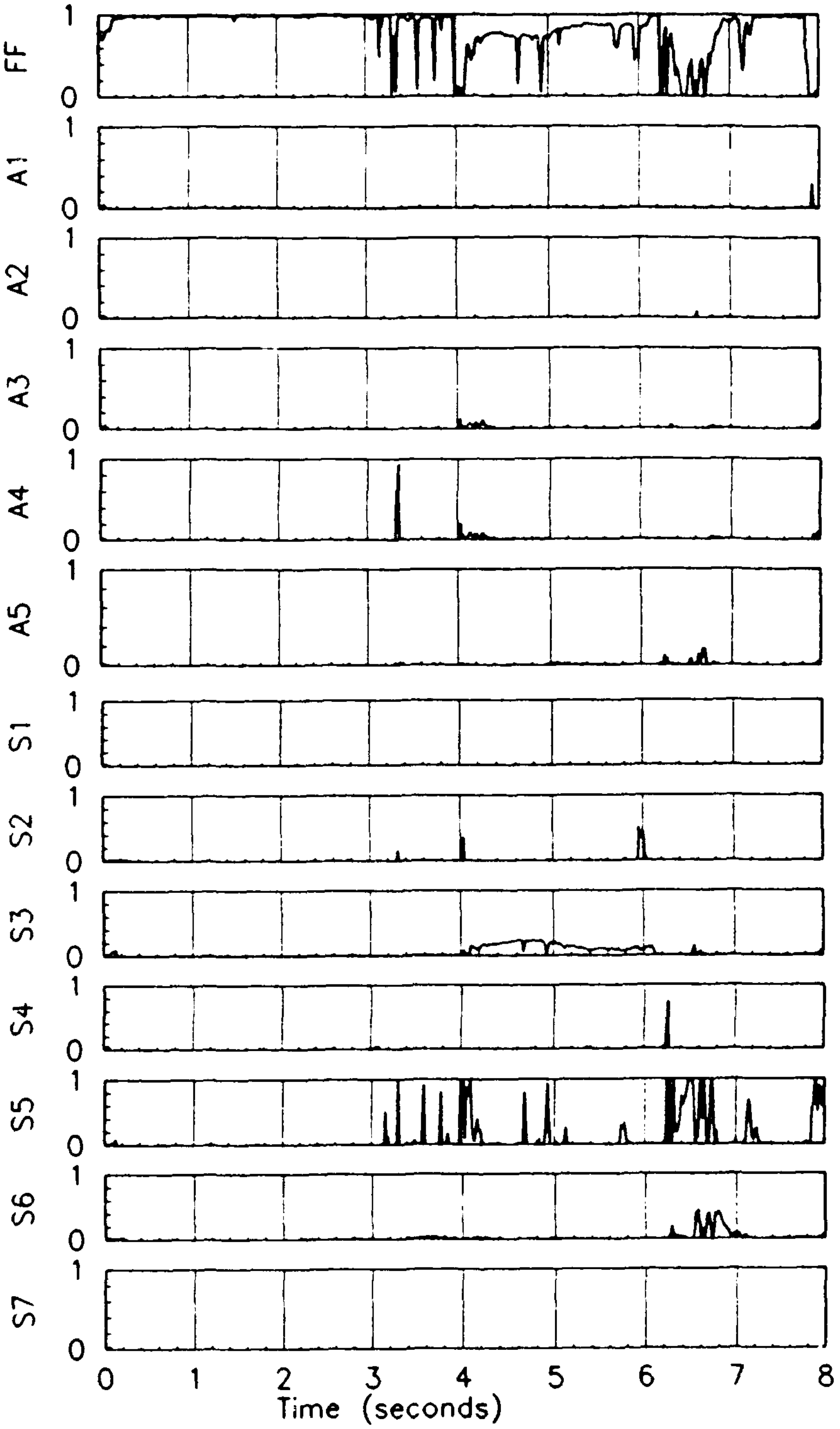

Figure 4.64 Probabilities for a 3.16σ increased sensor noise failure for the roll rate sensor

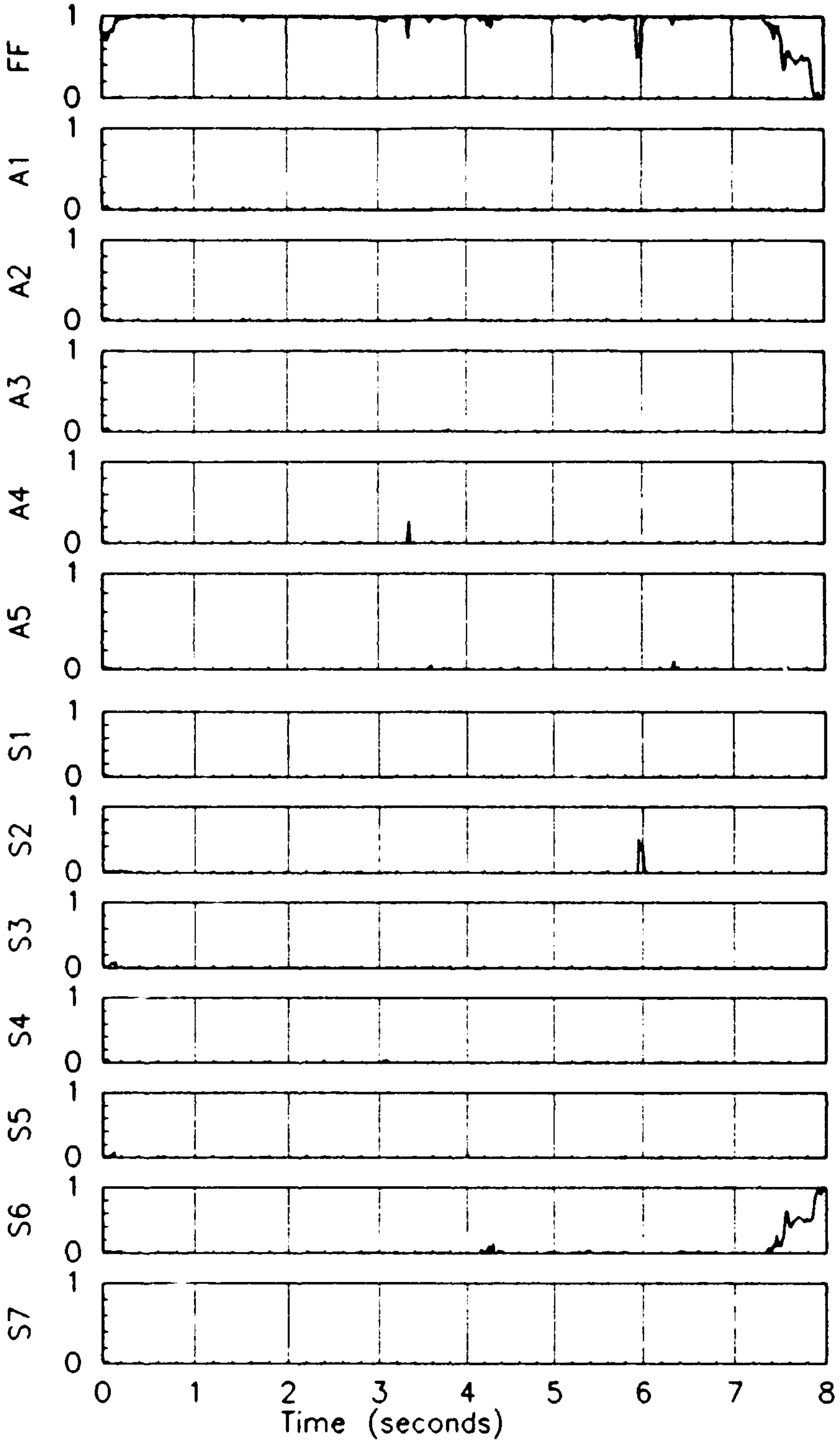

Figure 4.65 Probabilities for a 1σ increased sensor noise failure for the yaw rate sensor

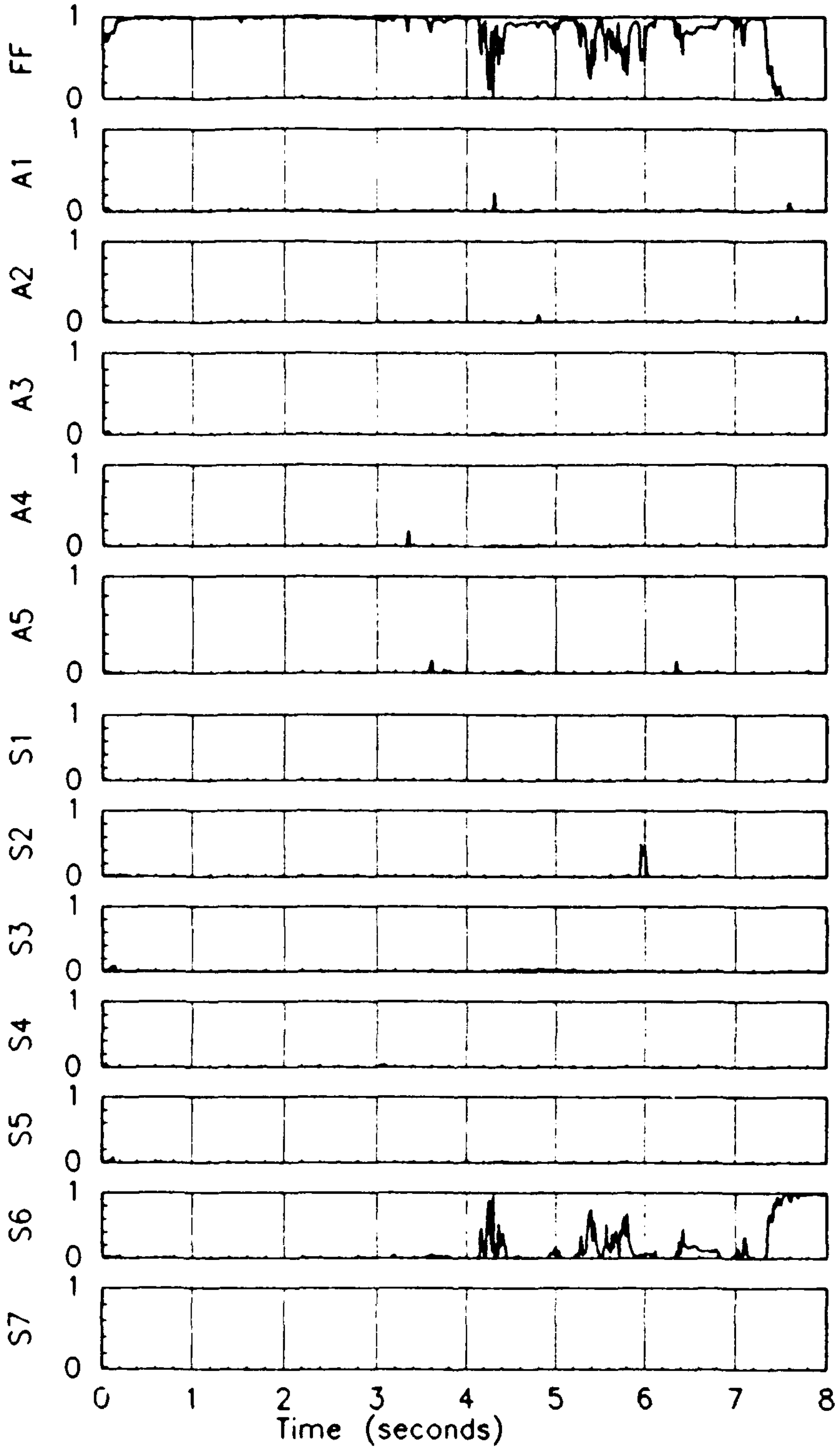

Figure 4.66 Probabilities for a 3.16σ increased sensor noise failure for the yaw rate sensor

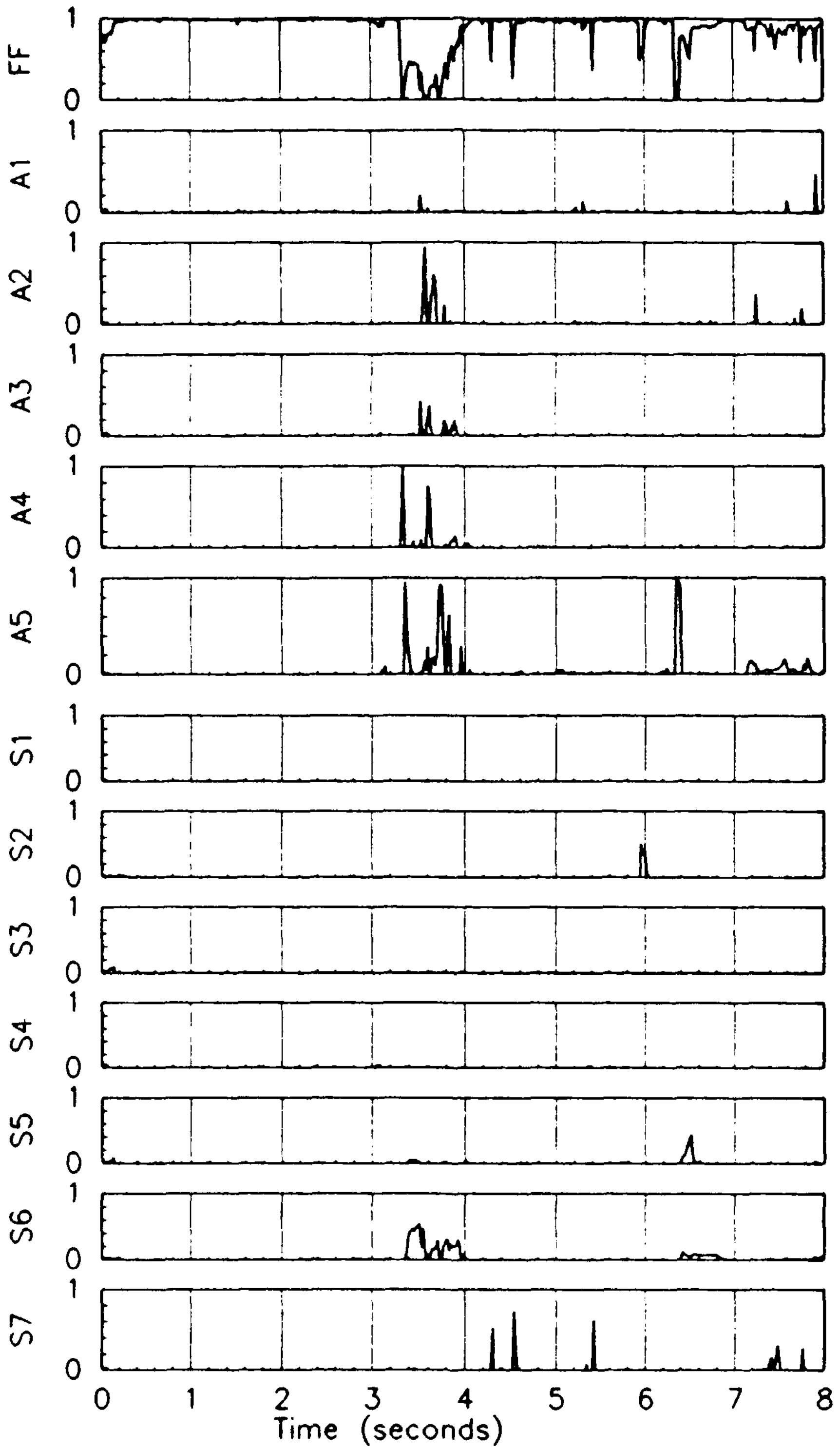

Figure 4.67 Probabilities for a 1σ increased sensor noise failure for the lateral acceleration sensor

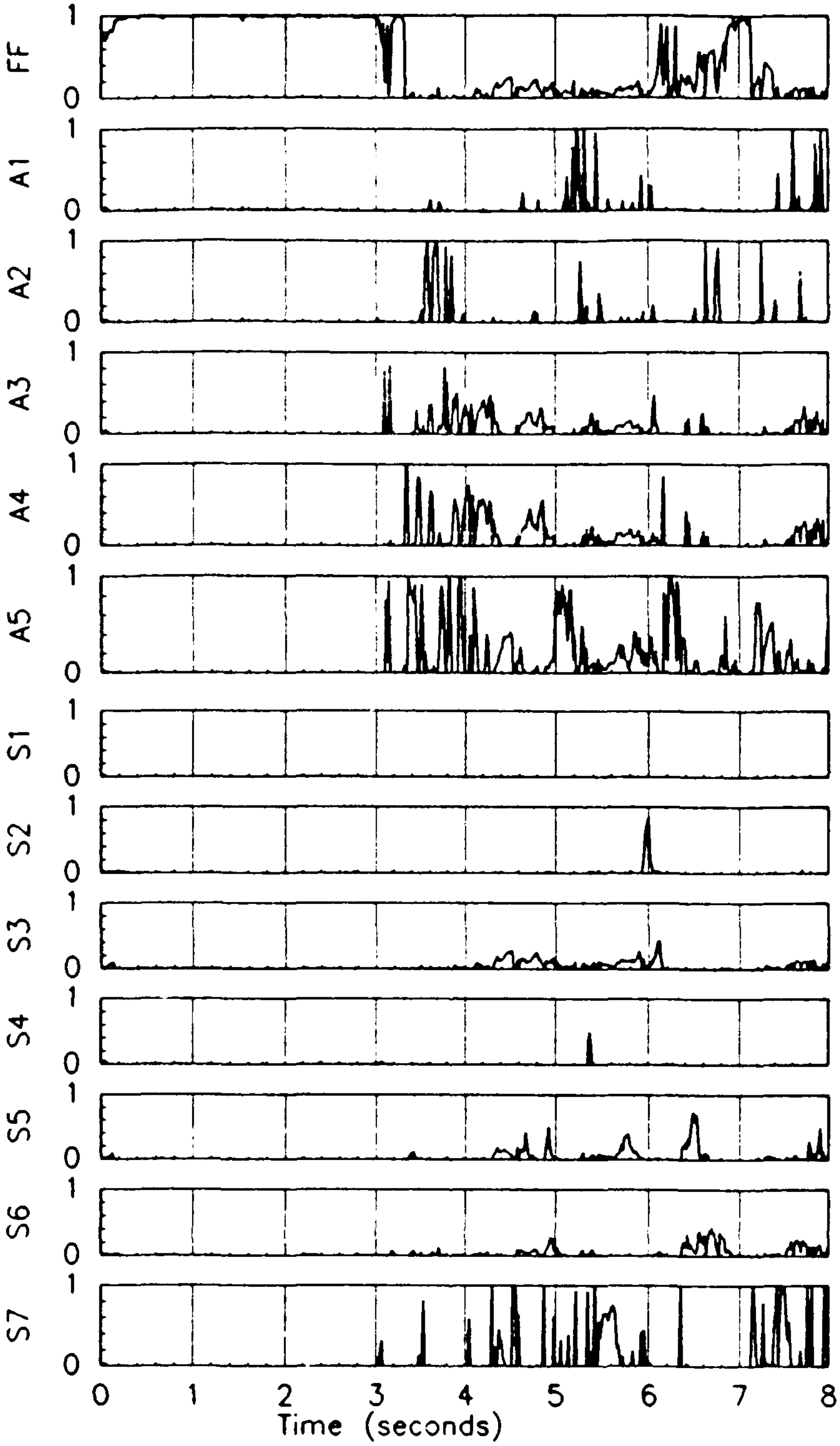

Figure 4.68 Probabilities for a 3.16σ increased sensor noise failure for the lateral acceleration sensor

4.3.2 Sinusoidal Dither (sensor bias)

During the investigation of single failures, an interesting situation arose. Whenever a yaw rate failure was inserted into the simulation, the probability would ramp slowly to the maximum value in the yaw rate failure (S6) filter (Figure 4.17). A study of the yaw rate residual within that filter identified a bias that ramped up until the failure occurred and then proceeded to ramp to zero. The bias is believed to be caused by the integration routine; however, this conjecture has not been proven conclusively. A question arose because of this bias effect. *If a failure has been declared by the algorithm, how much of a bias can exist in the residual before the failure is masked by the bias?* The thesis attempted to address this question by studying two sensor bias cases. The purpose of this study was to investigate the effect of single residual biases on the MMAE algorithm performance. Figure 4.69 demonstrates the effect of a roll rate bias of 0.1 deg/sec applied to the roll rate residual at 3.0 seconds (also the insertion of a roll rate failure). Figure 4.69 shows that the algorithm has difficulty in determining which filter contains the correct hypothesis. At 3.1 to 3.5 seconds, the roll rate failure elemental filter is identified as the correct filter. At 3.5 seconds, the probability leaves the roll rate failure filter and is captured by the fully functional aircraft filter until 4.6 seconds. Notice this pattern occurs again after the second pulse application. At 4.5 seconds, the roll rate failure filter is determined to be the correct filter. Just prior to the application of the second pulse, the probability is scattered between the fully functional aircraft filter, the flaperon failure filters, and the pitch rate failure filter. The pulse pattern is repeated from 6.0 to 6.6 seconds. At 6.6 seconds, the roll rate filter is identified as the correct failure. The evidence suggests that very little additional bias could be tolerated without the complete inability of the algorithm to identify the correct failure. An additional run was performed for a bias of 1.0 deg/sec (not shown). The results indicated no detection by the roll rate filter of a failure.

Figure 4.70 demonstrates a yaw rate bias of 0.01 deg/sec. The bias is inserted with the yaw rate sensor failure at 3.0 seconds. Again, the purpose of this study is to determine how much of a bias can be tolerated and still provide adequate failure protection. Figure 4.70 indicates the identification of the yaw rate failure at 3.2 seconds. At 4.85 seconds, the yaw rate filter loses the probability to the fully functional aircraft elemental filter, the flaperon failure filters, and the pitch rate failure filter. The application of a second pulse restores the probability to the yaw rate failure filter for a nearly identical time period following the first pulse (or 1.65

seconds). An additional run with the bias set at 0.08 deg/sec revealed that the algorithm could no longer identify the correct filter.

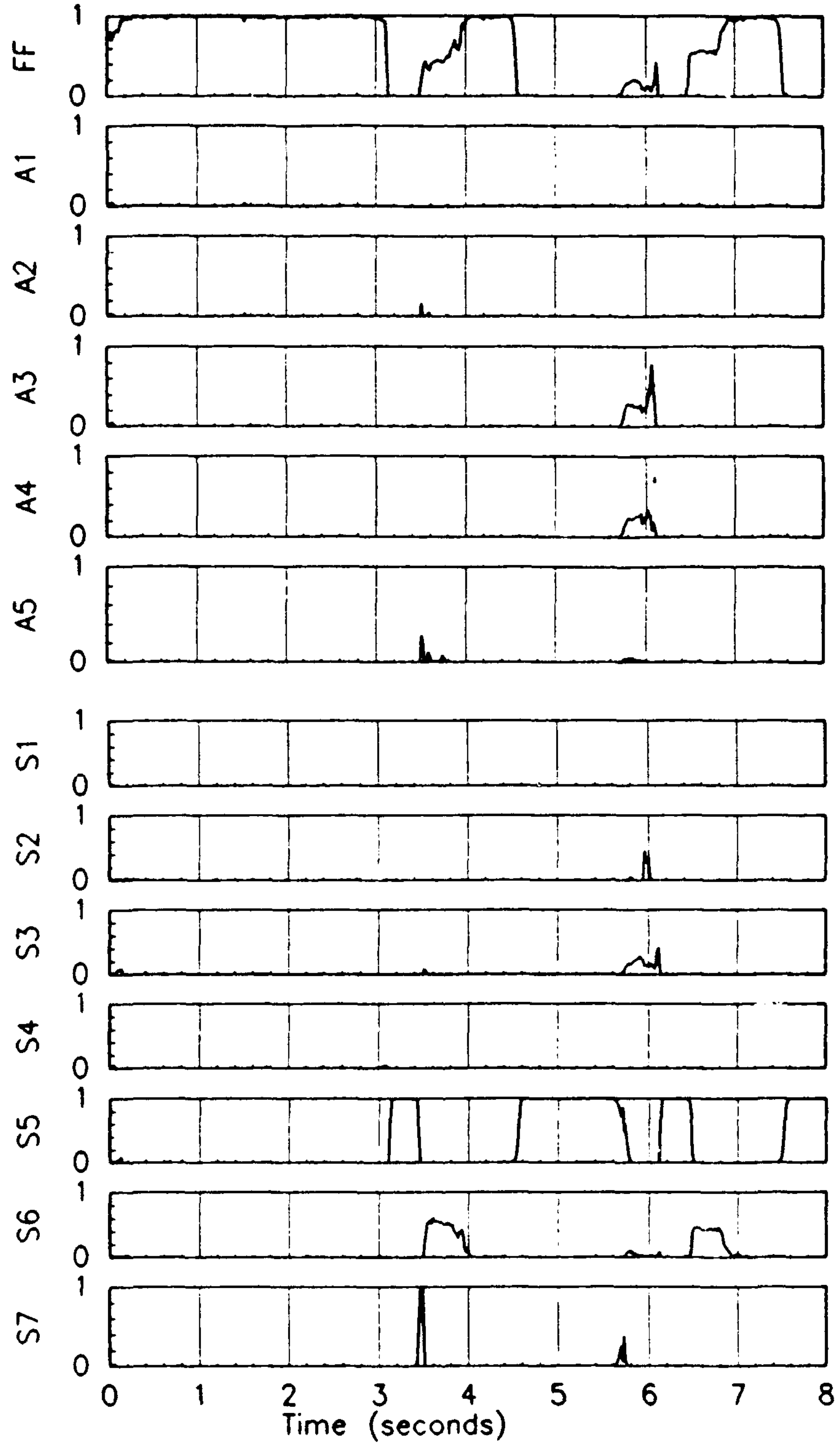

Figure 4.69 Probabilities for a roll rate bias of 0.1 deg/sec using a subliminal dither pulse

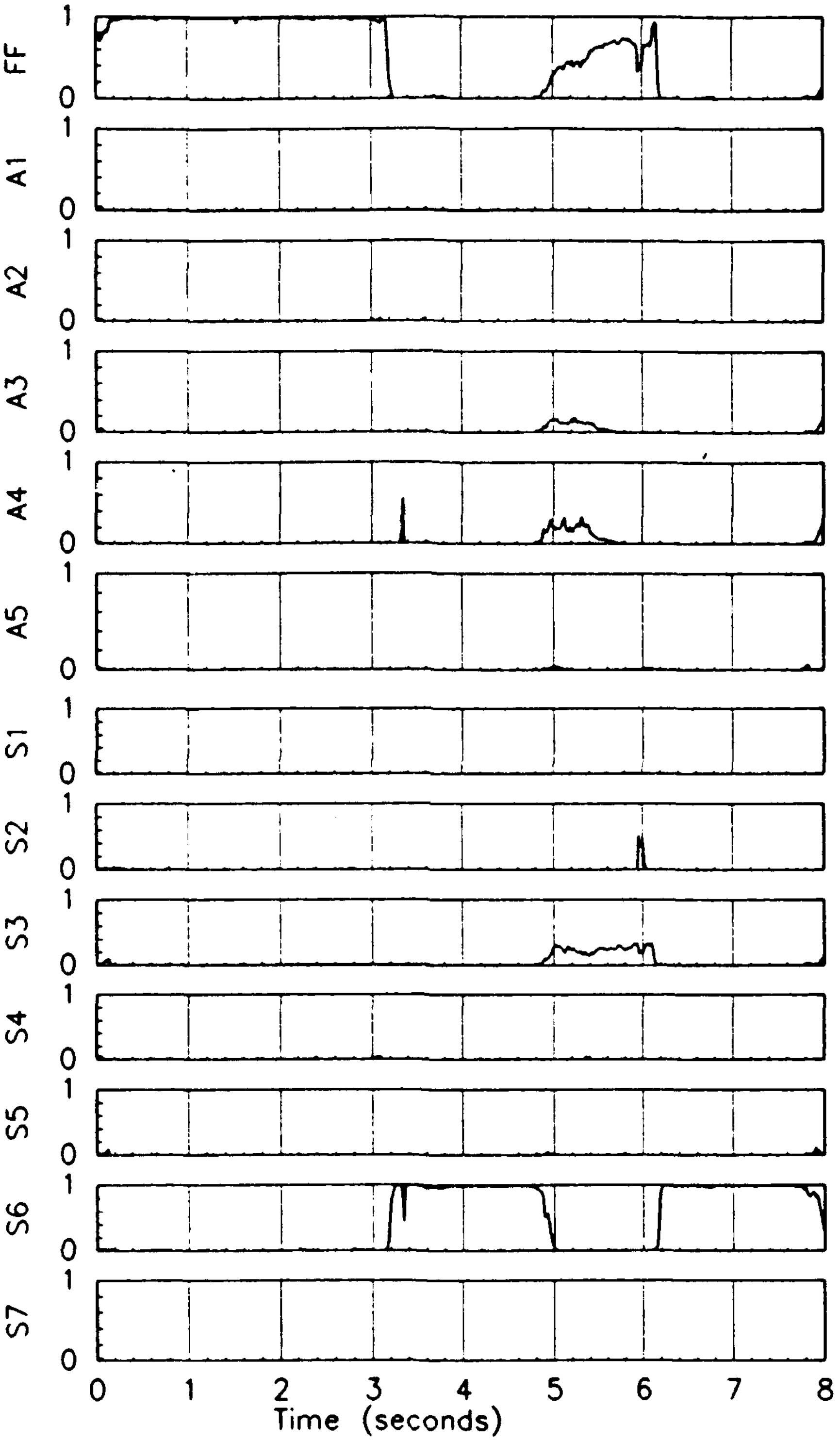

Figure 4.70 Probabilities for a yaw rate bias of 0.01 deg/sec using a subliminal dither pulse

4.4 Multiple Failures

The term "multiple failures" in this thesis is defined as two failures. The only multiple failures analyzed within this thesis are dual hard failures. The first failure is induced at 3.0 seconds. The second failure is induced at 6.0 seconds, coinciding with the second pulse application. The best system performance will occur when the pulse and failure occur simultaneously. After the characterization of the dual failure performance with the failures separated by 3.0 seconds, the time between failures was reduced until the two failures occurred simultaneously. Table 4.1 presents the results of the multiple failure matrix. The column header lists the left stabilator (LS), the right flaperon (RF), the rudder (RD), the velocity sensor (VS), the angle of attack sensor (AOA), the pitch rate sensor (PR), the normal acceleration sensor (NZ), the roll rate sensor (RR), the yaw rate sensor (YR), and the lateral acceleration sensor (NY). The right stabilator and left flaperon were not included since no additional insight would be gained by running these failures. The first column lists all of the second failures. Any dual failure scenario can be evaluated by finding the intersection of the column and row. The first failure is listed in the column heading and the second failure is the intersecting row.

The terminology within the table is in some sense subjective and is defined as follows:

ND - No Detection - The signal was not detected in the appropriate failed filter

Poor - Some spiking - Some probability spiking occurred in the appropriate failed filter

Fair - Prob. Lock - The probability in the appropriate filter was at 0.988 (max) for some appreciable period of time

Good - Lock & Hold - The probability in the appropriate filter was at 0.988 (max) for some appreciable period of time and was consistently 0.988 through the end of the simulation run (8.0 seconds).

	LS	RF	RD	VS	AOA	PR	NZ	RR	YR	NY
LS		Good	Good	Poor	Good	Good	Good	Good	Good	Good
RS	ND	Good	Good	Poor	Good	Good	Good	Good	Good	Good
LF	Poor	Good	Poor	Fair	Good	Good	Good	Good	Good	Good
RF	Poor		Poor	Fair	Good	Good	Good	Good	Good	Good
RD	ND	ND		Fair	Good	Good	Good	Good	Good	Good
VS	Good	Good	Good		Good	Good	Good	Good	Good	Good
AOA	ND	Good	Good	Good		Good	Good	Good	Good	Good
PR	ND	Poor	Poor	Fair	Good		Good	Good	Good	Good
NZ	Good	Good	Good	Good	Good	Good		Good	Good	Good
RR	ND	ND	Poor	Fair	Good	Good	Good		Good	Good
YR	ND	ND	ND	Poor	Good	Good	Good	Good		Poor
NY	Fair	Good	Good	Fair	Good	Good	Good	Good	Good	

Table 4.1 Multiple Hard Failure Summary Matrix

The "ND" and "Poor" results are shaded to accentuate the problem areas. Particularly evident from this shading is that there is much less difficulty if the first failure is a sensor rather than an actuator. In many cases, this is due to the fact that the first failure of an actuator reduces the ability to apply significant dither excitation (because of the failed actuator) to allow desirable detection of the second failure. This can be addressed in final implementation by modifying the appropriate dither once a first actuator failure has been declared. This is explored later in Section 4.4.4.

The complete set of results can be found in Appendix A. A portion of those results will be presented within this section. The criteria for presentation within this section is to demonstrate dual actuator failures, dual sensor failures, combinations of actuator and sensor failures; to present examples of the subjective rating system for each category (ND, poor, fair, and good); to demonstrate differences between the order of two failures; and finally, to show examples of improved excitation and the effect on algorithm identification performance. All of the results, generated with the subliminal dither pulse, presented in Table 4.1 are based upon a pulse train dither occurring every 3.0 seconds (the same pulse used in the single failure section).

Multiple failures, in this case dual failures, were represented conceptually in Section 2.5. The level 0 designation represents the first no-failure condition. The filters within this bank consists of a fully-functional aircraft filter, five single-actuator-failure filters, and seven single-sensor-failure filters. The next level is assigned the designation of level 1. Each level 1 filter bank assumes a first failure has already occurred. The filters within these banks consists of a no-failure filter (to allow the algorithm the ability to back out of the structure to level 0 in the case of a misidentification), five actuator filters designed for an assumed failure and any other second failure, and seven sensor failures designed for an assumed failure and any other second failure. (The elemental filter that duplicates the single assumed failure is the only exception - it assumes only that one single failure.) As an example for a left stabilator failure, from 0 to 3.0 seconds the probability would be in the level 0 fully-functional aircraft filter. Following the detection of a left stabilator failure, sometime after the insertion of that failure at 3.0 seconds into the simulation, the probability would collect in the level 0 left stabilator filter (A1) until the probability was above 0.9 for more than 10 sample periods (this switching criterion is used to reduce erroneous bank switching compared to what would result from a probability threshold less than 0.9 or froma time threshold less than 10 sample periods). After the bank switching criterion was met, the algorithm would switch to the left stabilator (A1) level 1 bank. This bank contains 13 filters: the no-failure (FF) filter, a left stabilator failure alone (A1) filter, four actuator filters each designed for a left stabilator failure and the appropriate actuator failure (i.e., left and right stabilator (A2) failures, left stabilator and left flaperon (A3) failures , left stabilator and right flaperon (A4) failures, and left stabilator and rudder (A5) failures), and seven sensor filters each designed for a left stabilator and any other sensor failure (i.e., left stabilator and velocity sensor (S1) failures, left stabilator and angle of attack sensor (S2) failures, left stabilator and pitch rate sensor

(S3) failures, left stabilator and normal acceleration sensor (S4) failures, left stabilator and roll rate sensor (S5) failures, left stabilator and yaw rate sensor (S6) failures, and the left stabilator and lateral acceleration sensor (S7) failures). The level 1 bank is not brought "on-line" until after a filter, other than the fully-functional filter, has met the switching criterion discussed previously. At this point the bank is switched to the appropriate level 1 bank. To conserve space, the probability time histories for the level 0 and level 1 banks are displayed on the same strip charts. To identify quickly whether the algorithm has switched banks, apply the following criteria: (1) the fully functional filter must not contain any appreciable (more than 10%) portion of the probability, (2) the first filter to have more than 90% of the probability for greater than 10 sample periods (0.15625 seconds) will be the level 1 bank the filter will bring "on-line". Also, a "Bank Switch" time designation is provided at the top of the first strip chart for convenience. At that point, the designations for each of the filters change meaning. The designation A1 no longer means a single left stabilator failure. It implies the first failure, that the algorithm identified as meeting the necessary conditions to switch the bank, and a left stabilator failure. This is true for all of the filter designations. Obviously, one filter in the twelve failure filters will be a single failure alone filter (i.e., A1 would still be a left stabilator only filter if the level 1 bank is the left stabilator failure bank).

4.4.1 Subliminal Dither Signals

Figure 4.71 presents the dual stabilator failure. At 3.0 seconds, a left stabilator failure is induced. The failure is detected, the probability is greater than 0.9 for more than ten sample periods (0.15625 seconds), and the level 1 (A1) bank is brought "on-line". Within the level 1 (A1) bank, the failure is locked for .9 seconds at which point ambiguity arises. The dual failure (A2) filter is incorrectly identified as the correct failure until 4.6 seconds. At 4.7 seconds, the left stabilator (A1) is properly identified as the correct failure. The induction of the right stabilator failure at 6.0 seconds is not detected within the 8.0 second simulation run. In this scenario, without a proper and necessary excitation of the pitch axis, a failure can not be detected. The only control surfaces which can provide this necessary excitation are the stabilators, both of which have failed. After a first failure, a different type of dither signal could be enabled to provide better performance. A higher magnitude sinusoidal dither signal may provide improved response. This hypothesis is demonstrated for a dual stabilator failure and a right flaperon followed by a pitch rate sensor failure. These figures will be presented later in this chapter. Additional surfaces provide additional options. Upon the isolation of a stabilator failure, the flaperons could be reconfigured to provide symmetrical deflections. This augmentation to the longitudinal dither signal might be enough to produce the necessary level of excitation for good detection performance. Since no failure detection occurred within the 6.0 to 8.0 time frame, the rating system assigns a "ND" for this failure scenario.

Figure 4.72 demonstrates the right flaperon and rudder failure. The right flaperon failure is induced at 3.0 seconds and is detected immediately. The level 1 right flaperon failure (A4) bank is brought "on-line". The induction of the rudder failure at 6.0 seconds is not detected by the algorithm. The majority of the probability remains with the right flaperon (only) filter. A small portion is in the fully functional "no-failure" filter. A small spike occurs in the AOA sensor failure filter and a small spike at 6.4 seconds in the yaw rate sensor failure filter is present. This scenario is assigned an "ND".

Figure 4.73 demonstrates a rudder and angle of attack sensor failure. The rudder failure is induced at 3.0 seconds and is locked by 3.3 seconds. The level 1 rudder failure (A5) bank is brought "on-line" at approximately 3.3 seconds. The induction of the angle of attack failure at 6.0 seconds is detected and locked by 6.1 seconds in the rudder and angle of attack sensor failure (S2) filter. A single probability spike occurs at 6.4 seconds. After the spike, the probability remains in the rudder and angle of attack sensor failure filter until the end of the simulation. This scenario is assigned a "Good" by the rating system.

Figure 4.74 presents the velocity sensor and left stabilator failure. The velocity sensor failure is induced at 3.0 seconds and is locked almost immediately. The level 1 velocity sensor failure (S1) bank is brought "on-line" at approximately 3.2 seconds. The left stabilator is induced at 6.0 seconds and some probability spiking occurs in the "no-failure" (FF) filter, the velocity sensor and left stabilator (A1) filter, the velocity sensor and right stabilator (A2) filter, and the velocity sensor and left flaperon (A3) filter. Some small spiking occurs in a few of the sensors. Since only probability spiking occurs in the velocity sensor and left stabilator (A1) failure filter, the rating system assigns a value of "Poor" to this scenario.

Figure 4.75 illustrates a pitch rate sensor and left stabilator actuator failure combination. The pitch rate sensor failure is detected and isolated within 0.2 seconds. The level 1 pitch rate sensor failure (S3) bank is brought "on-line" at approximately 3.3 seconds. The insertion of a left stabilator failure at 6.0 seconds is initially captured by the pitch rate and left stabilator failure (A1) filter, lost to the pitch rate failure only (S3) filter, split between the pitch rate and left stabilator failure (A1) filter and the pitch rate and right stabilator failure (A2) filter, and finally locked by the pitch rate sensor and left stabilator failure (A1) filter at 7.6 seconds. Note that, even during this time of ambiguity, the pitch rate failure and a stabilator failure were detected (more than just a pitch rate), and it takes some finite amount of time to resolve which stabilator failed (a typical left/right ambiguity). This ambiguity is acceptable in that a the correct "class" of second failure is identified as opposed to ambiguities with other actuators. While not optimum by any criteria, the correct filter was identified and locked 1.6 seconds after the second failure. The rating system assigns this scenario a rating of "Good".

Figure 4.76 demonstrates a pitch rate sensor and rudder actuator failure. The probability is captured by the pitch rate sensor failure filter at 3.2 seconds (0.2 seconds after failure insertion). The level 1 pitch rate sensor failure (S3) bank is brought "on-line". The second failure is inserted at 6.0 seconds, after which the pitch rate sensor and rudder actuator (A5) filter begins to collect probability at 6.4 seconds but shares the probability with the pitch rate failure only (S3) filter until 6.9 seconds. At this time, all of the probability is contained within the pitch rate sensor and rudder actuator failure (A5) filter. The subjective rating system assigns this scenario a rating of "Good".

Figure 4.77 presents a pitch rate sensor and roll rate sensor combination failure. The pitch rate sensor is identified as the correct failure at 3.2 seconds (0.2 seconds after failure insertion). The level 1 pitch rate sensor failure (S3) bank is brought "on-line" at approximately 3.3 seconds. The pitch rate and roll rate failure (S5) filter is identified as the proper filter at 6.2 seconds (0.2 seconds after the second failure insertion). Some small probability spiking occurs within the pitch rate and right flaperon failure (A4) filter and the pitch rate and rudder failure (A5) filter. As is the case with nearly all of the double sensor failures, the detection is swift, clean, and correct. The rating system assigns this scenario a rating of "Good".

Figure 4.78 illustrates a normal acceleration and pitch rate failure. The normal acceleration failure is inserted at 3.0 seconds, and it is detected and locked by 3.15 seconds. The level 1 normal acceleration failure (S4) bank is brought "on-line" at approximately 3.3 seconds. The pitch rate failure is inserted at 6.0 seconds.

The normal acceleration and pitch rate failure (S3) filter captures the probability within 0.15 seconds and remains locked. The rating system assigns this scenario a rating of "Good". This failure scenario is included to demonstrate additional double sensor failure characteristics.

Figure 4.79 presents the data for a yaw rate sensor and lateral acceleration sensor dual failure. The yaw rate sensor failure is initially detected at 3.3 seconds but not locked until 4.2 seconds. This delay is due to a known phenomena within the yaw rate residual in the yaw rate filter. A bias is present in the yaw rate residual. The bias is believed to be due to numerical integration problems. The solution to this nuisance is to restart the filter at regular intervals. As previously discussed, restarting the filters will remove the bias from the residual calculations. By removing the bias, the yaw residuals will fall well within the 3σ bounds and will result in good performance. The level 1 yaw rate sensor failure (S6) bank is brought "on-line" at approximately 4.3 seconds. The insertion of the second failure is detected immediately. The rating system assigns this scenario a rating of "Good". The next scenario reverses the order of these two failures to demonstrate differences in failure detection characteristics.

Figure 4.80 demonstrates a lateral acceleration sensor and yaw rate sensor failure. The lateral acceleration failure is detected almost immediately after failure insertion at 3.0 seconds. The level 1 lateral acceleration failure (S7) bank is brought "on-line" at approximately 3.2 seconds. The insertion of the second failure, the yaw rate sensor, is only partially detected by the lateral acceleration and yaw rate sensor failure (S6) filter. The majority of the probability is retained by the lateral acceleration only (S7) filter. This scenario is assigned a rating of "Poor" by the subjective rating system. This scenario produces the same end result as in Figure 4.78 yet the probability traces are markedly different. Restarting the yaw rate filter may significantly improve this scenario's rating.

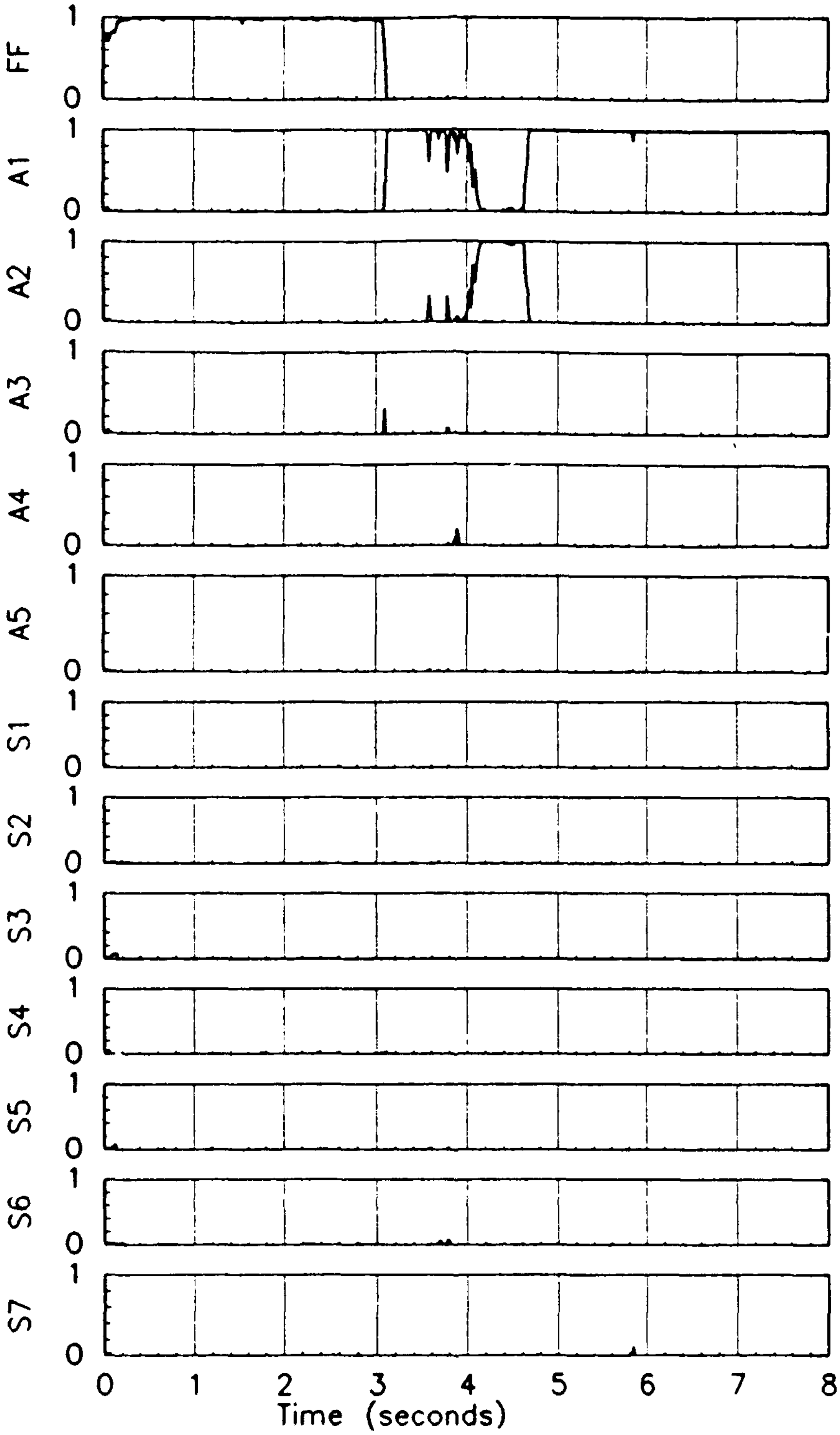

Figure 4.71 Probabilities for a left stabilator failure followed by a right stabilator failure using a subliminal dither pulse

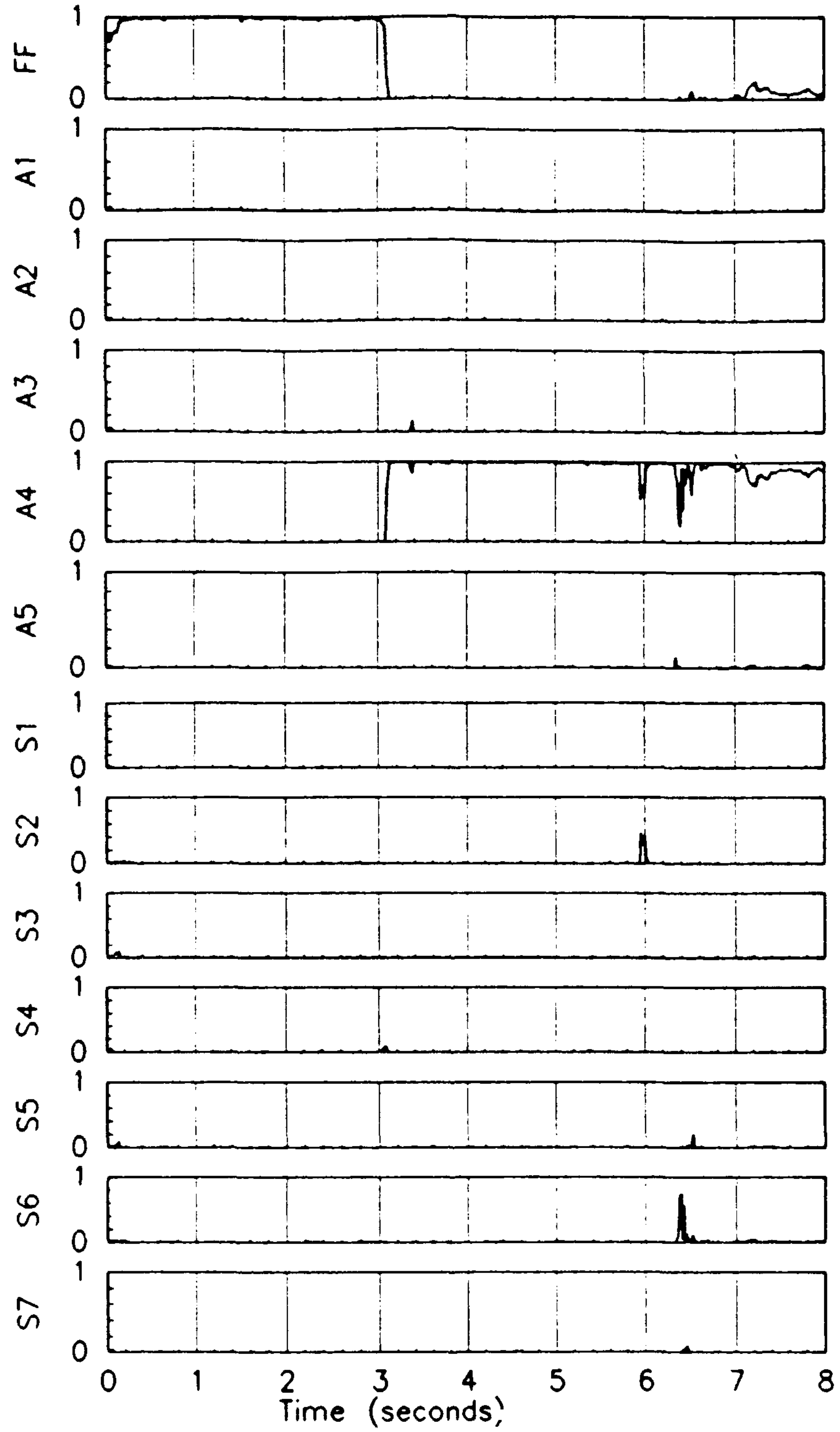

Figure 4.72 Probabilities for a right flaperon failure followed by a rudder failure using a subliminal dither pulse

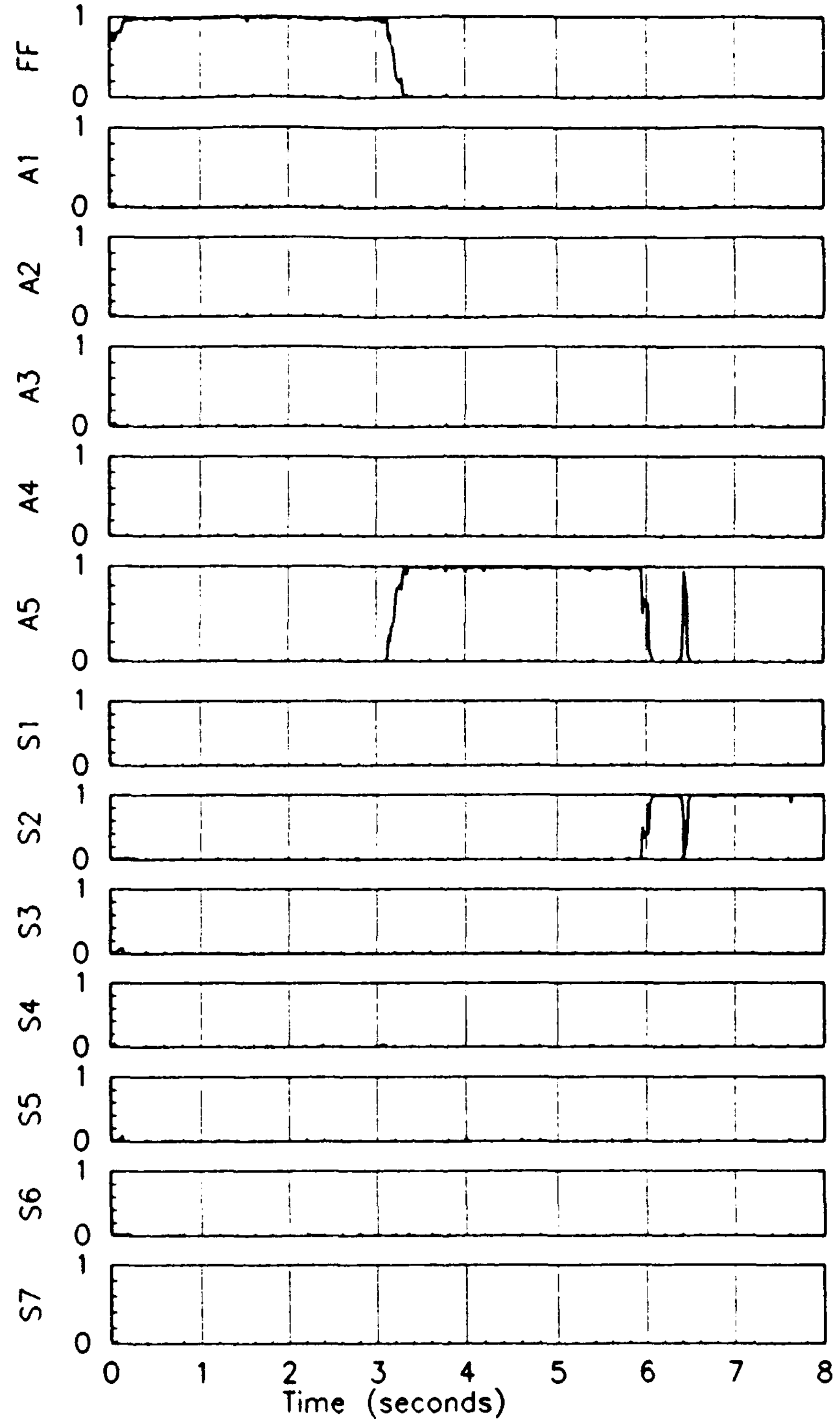

Figure 4.73 Probabilities for a rudder failure followed by an angle of attack sensor failure using a subliminal dither pulse

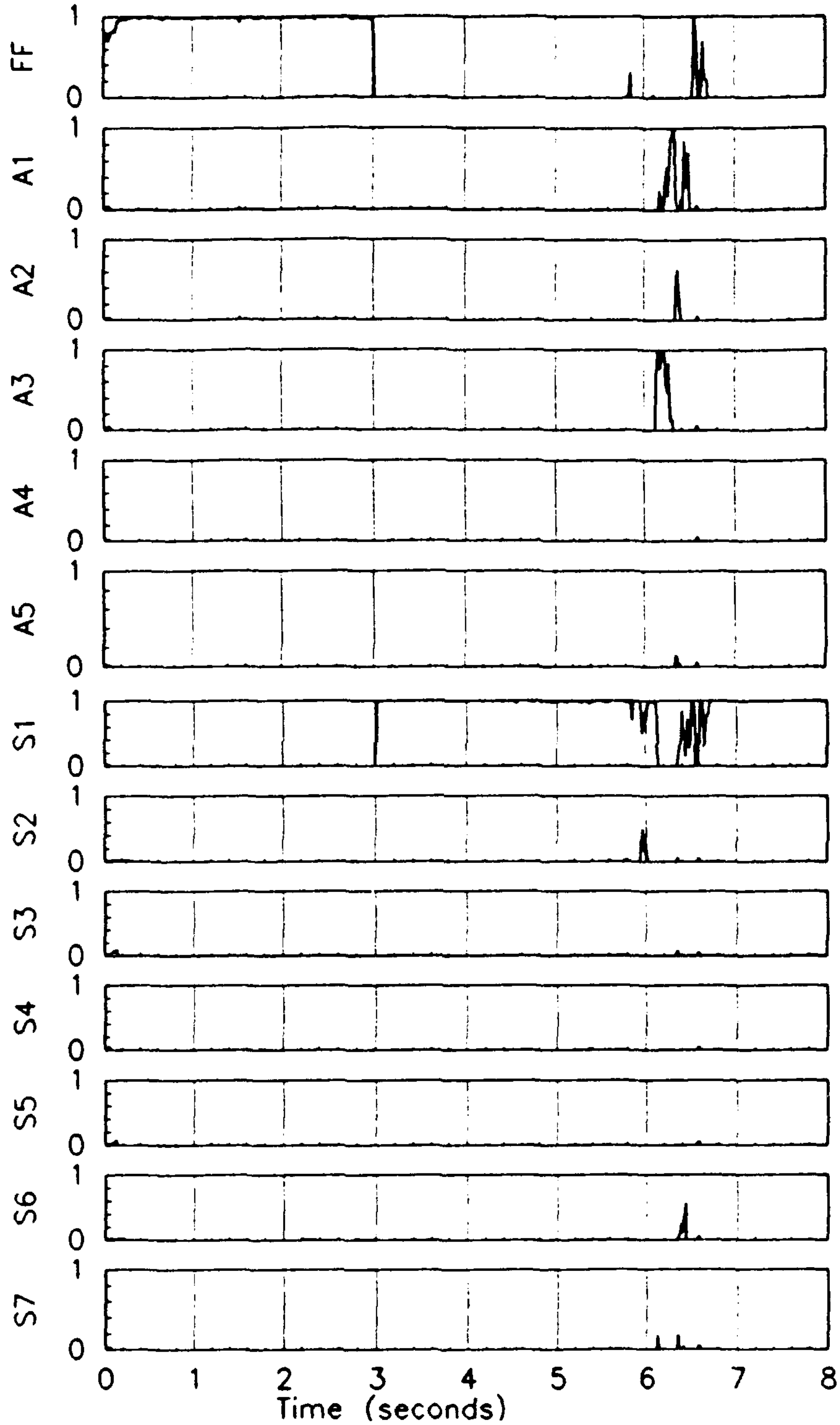

Figure 4.74 Probabilities for a velocity sensor failure followed by a left stabilator failure using a subliminal dither pulse

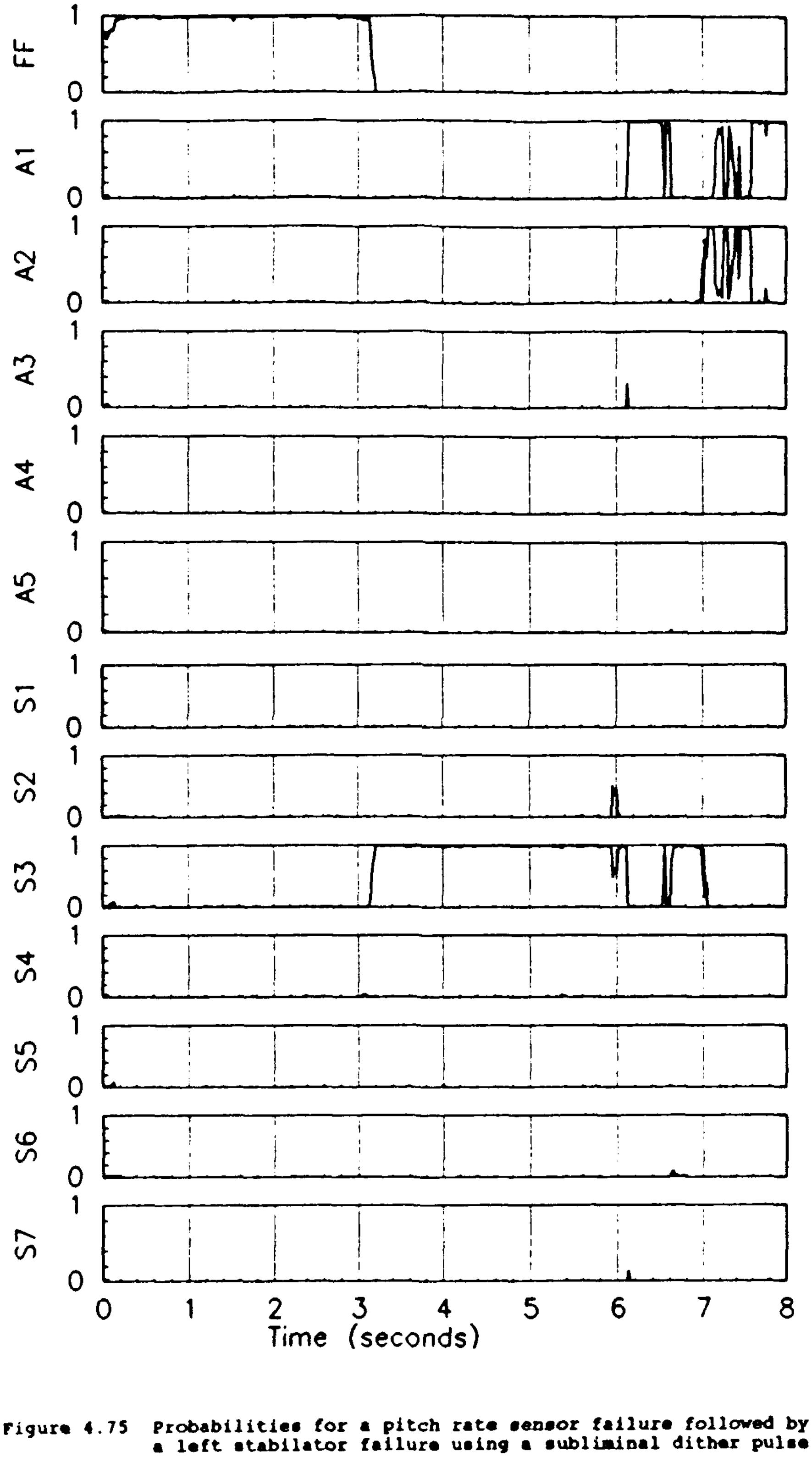

Figure 4.75 Probabilities for a pitch rate sensor failure followed by a left stabilator failure using a subliminal dither pulse

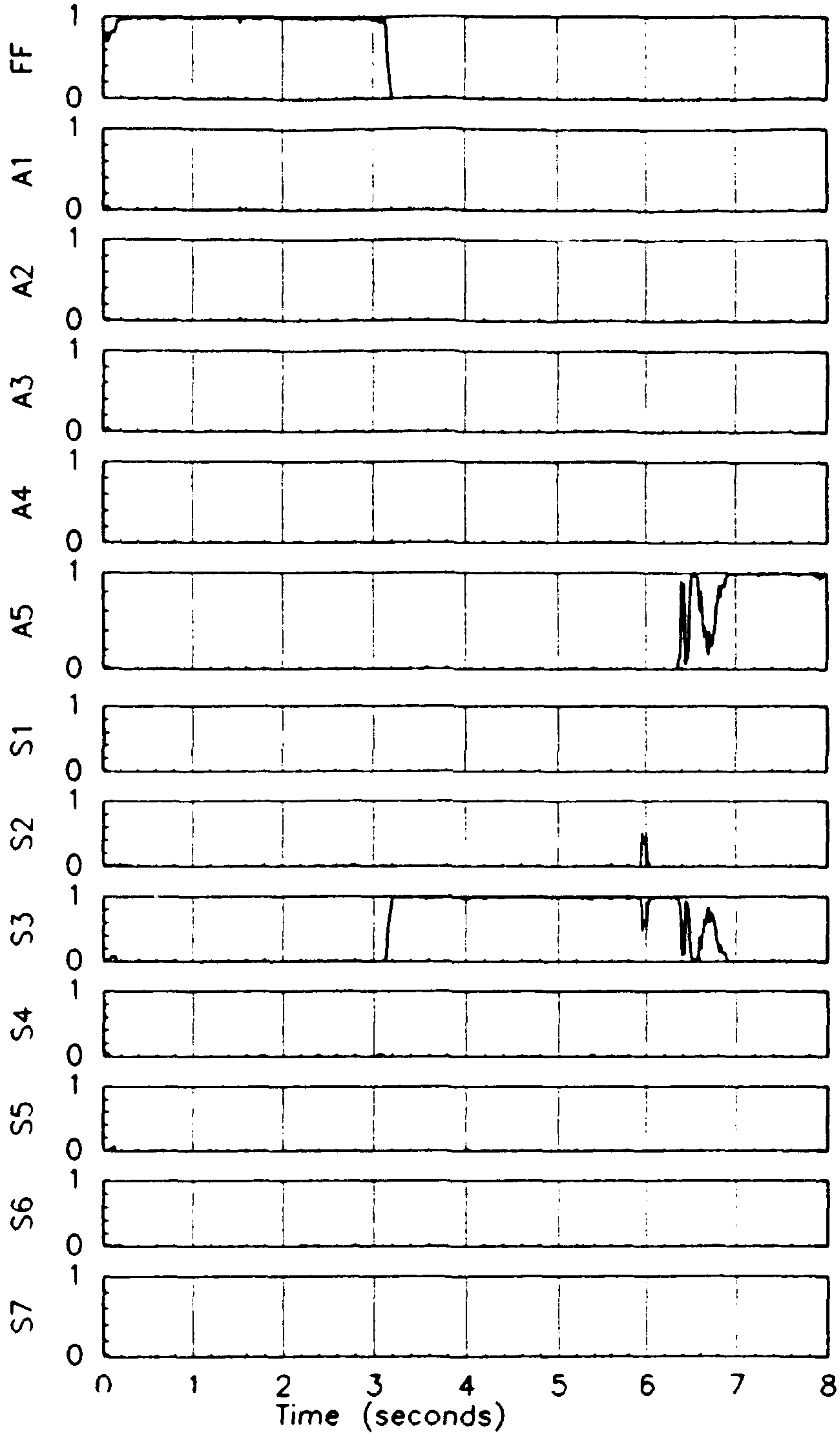

Figure 4.76 Probabilities for a pitch rate sensor failure followed by a rudder failure using a subliminal dither pulse

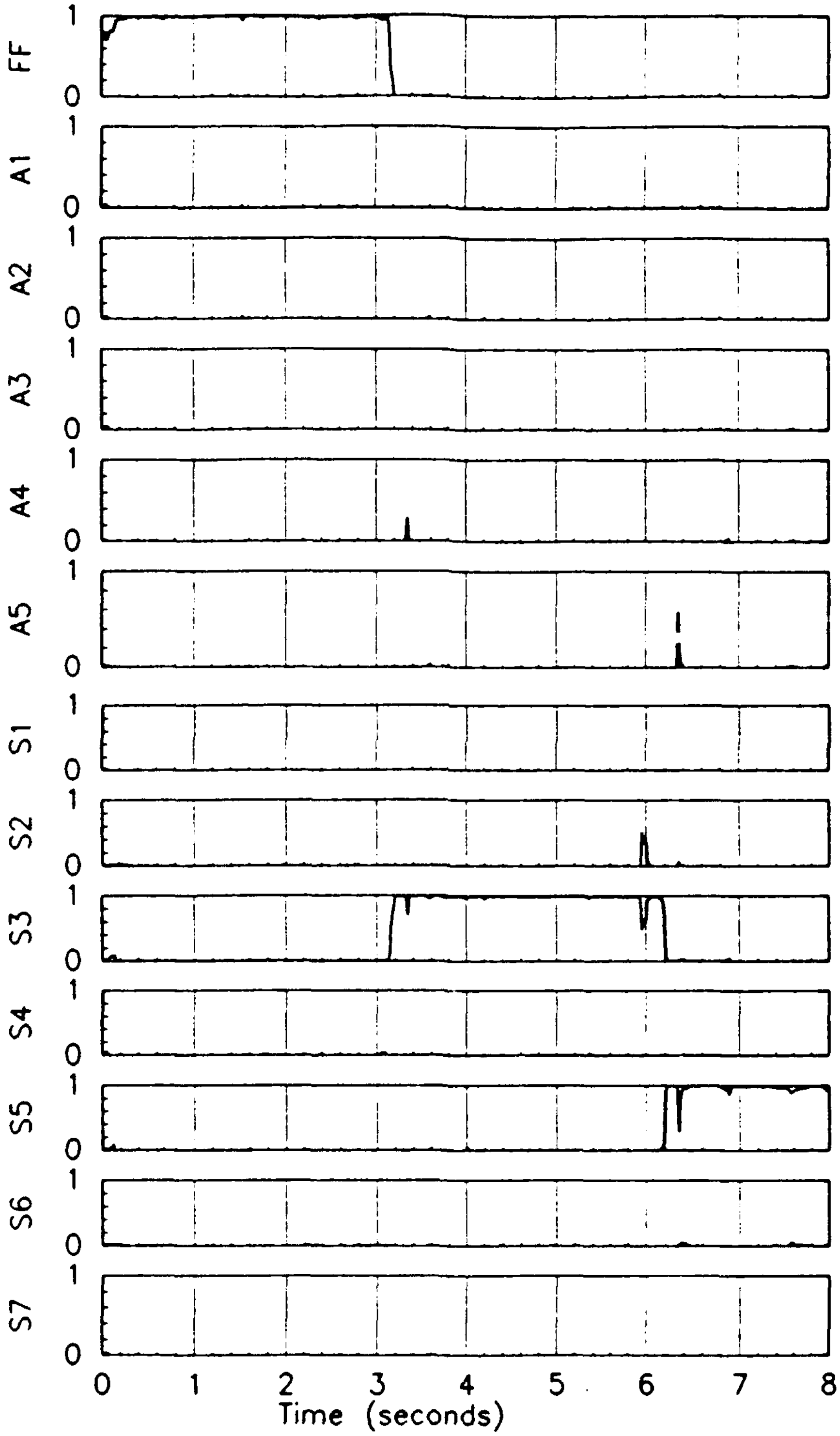

Figure 4.77 Probabilities for a pitch rate sensor failure followed by a roll rate sensor failure using a subliminal dither pulse

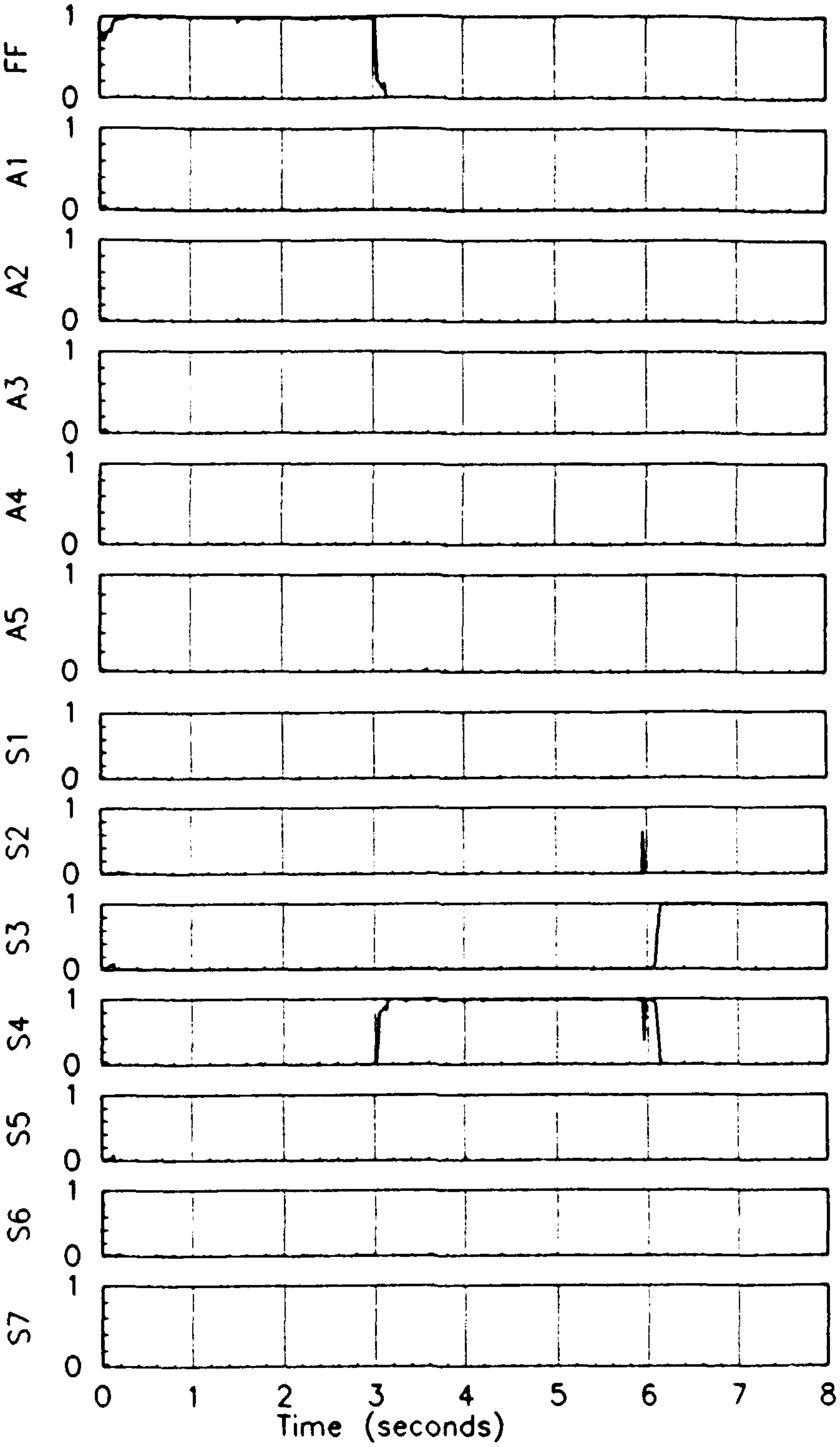

Figure 4.78 P obabilities for a normal acceleration sensor failure followed by pitch rate sensor failure using a subliminal dither pulse

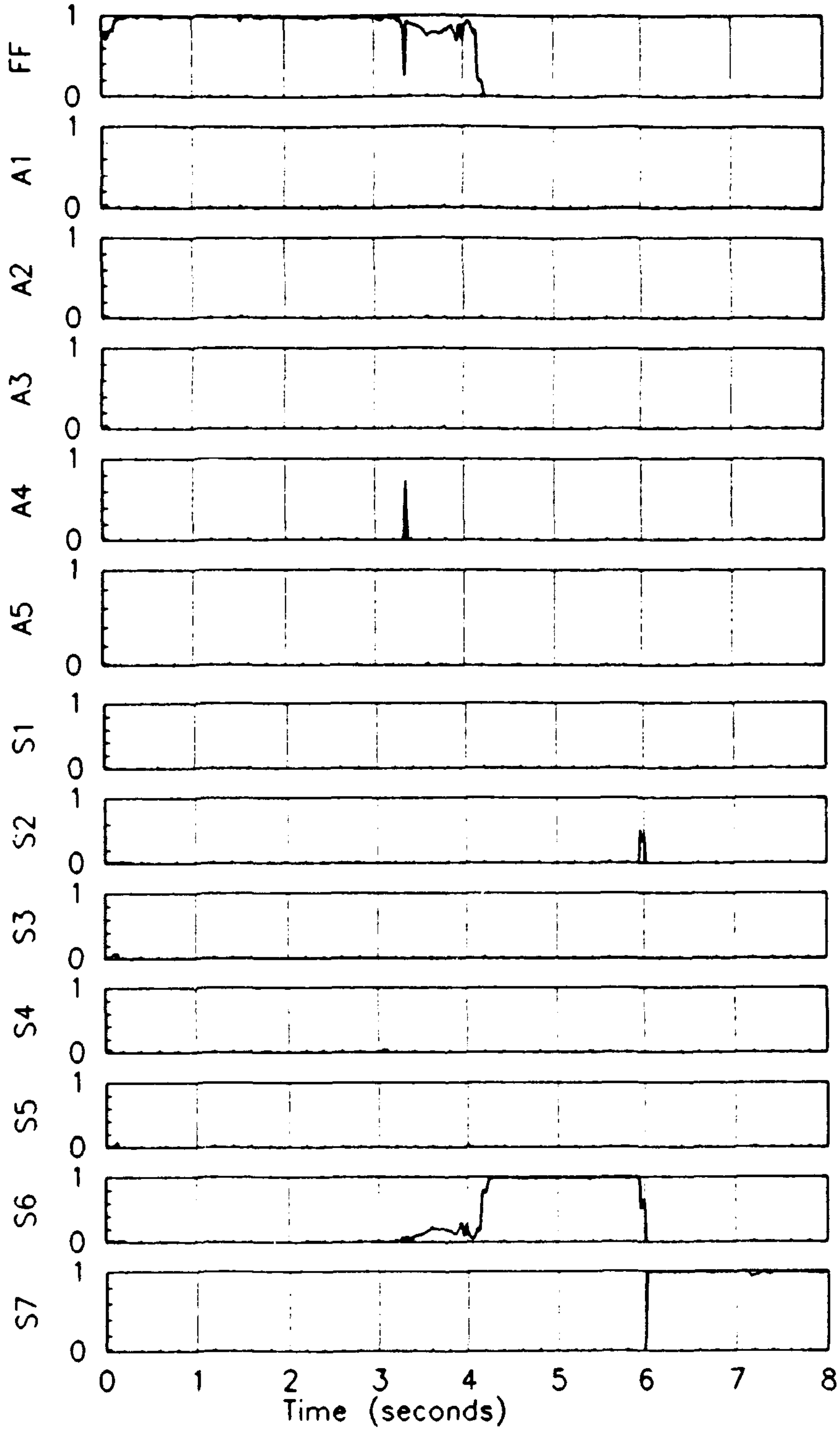

Figure 4.79 Probabilities for a yaw rate sensor failure followed by a lateral acceleration sensor failure using a subliminal dither pulse

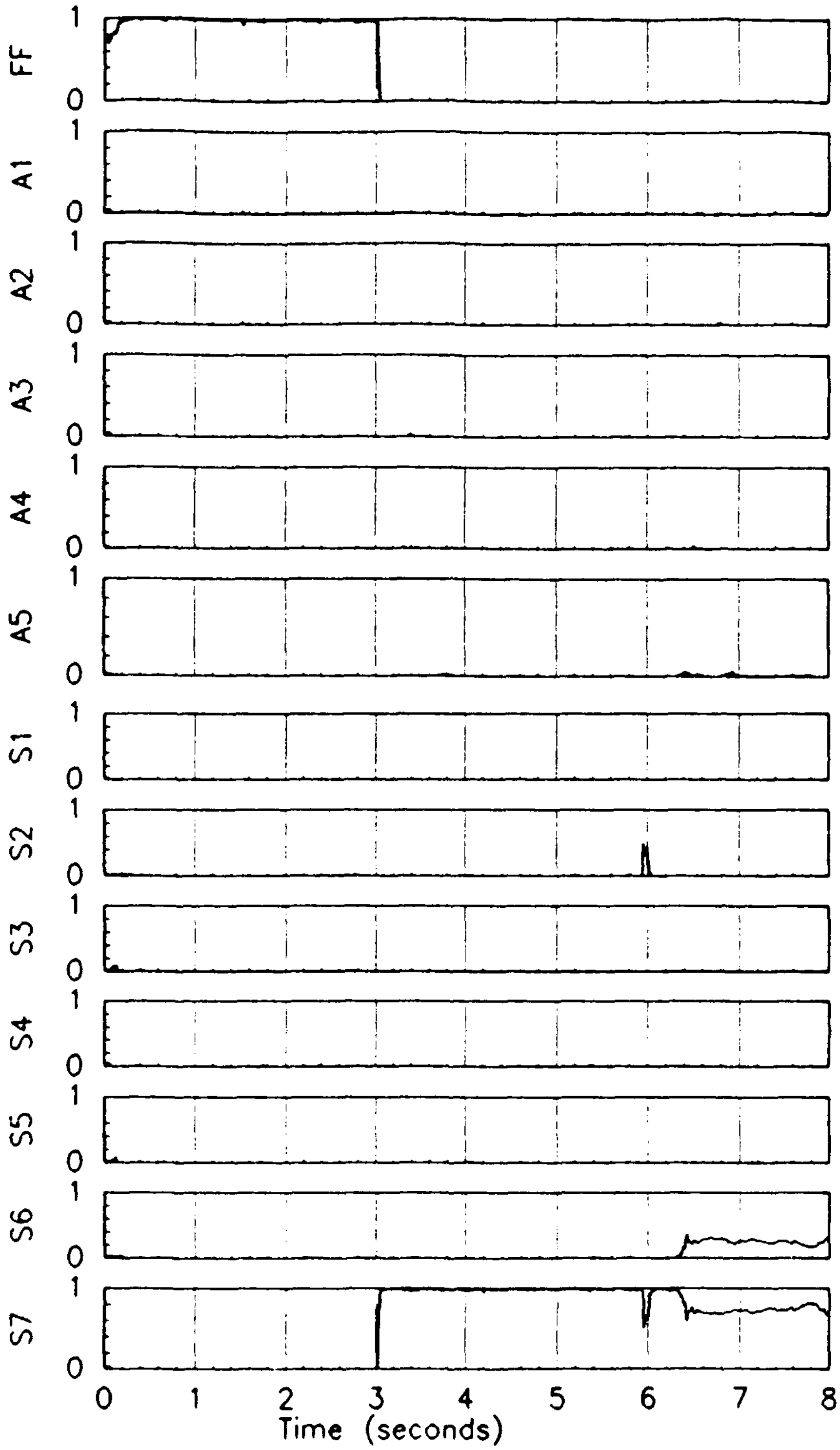

Figure 4.80 Probabilities for a lateral acceleration sensor failure followed by a yaw rate sensor failure using a subliminal dither pulse

4.4.2 Sinusoidal Dither Signals

Figure 4.81 presents data for a dual stabilator failure using a sinusoidal dither. At 3.0 seconds, the left stabilator failure is induced. The stabilator is detected and locked for small periods of time from 3.2 to 6.0 seconds. The level 1 left stabilator failure (A1) bank is brought "on-line" at approximately 3.65 seconds. The induction of a right stabilator failure at 6.0 seconds is indicated by the probability spiking that occurs between the left stabilator failure only (A1) filter and the dual stabilator failure (A2) filter. While this scenario demonstrates that the excitation dither signal is not of sufficient strength to isolate the failure positively, it is a significant improvement over Figure 4.71 in which proper failure detection failed to occur. The subjective rating system assigns a rating of "Fair" to this scenario.

Figure 4.82 presents a right flaperon and rudder failure using a sinusoidal dither signal. The insertion of a right flaperon failure at 3.0 seconds is initially detected by the right flaperon failure filter at 3.2 seconds. The failure is not locked by the right flaperon failure filter until 4.0 seconds. The level 1 right flaperon failure (A4) bank is brought "on-line" at 3.6 seconds. Some spiking occurs in other filters during the 3.0 to 4.0 time frame. At 6.0 seconds, the rudder failure is inserted. The majority of the probability is shared between the right flaperon failure only (A4) and the right flaperon and rudder failure (A5) filters. The dual failure combination is detected by the right flaperon and rudder failure (A5) filter at 6.2 to 6.5 seconds, lost until 7.2 seconds, lost again at 7.5 seconds, and finally locked at 7.55 seconds. The signal is locked through 8.0 seconds. Comparison with Figure 4.72 demonstrates significant differences. Figure 4.72 was assigned an "ND" by the rating system. Applying the previously defined criteria to this scenario, a rating of "Good" is assigned with a delay between failure insertion and probability lock of 1.5 seconds.

Figure 4.83 presents data for a velocity sensor failure and a left stabilator failure using a sinusoidal dither. The velocity sensor failure is induced at 3.0 seconds, and it is detected and locked immediately. The level 1 velocity sensor failure (B7) bank is brought "on-line" at approximately 3.2 seconds. The insertion of the second failure at 6.0 seconds demonstrates that the probability is initially shared between the velocity sensor and left stabilator failure (A1) filter, the velocity sensor and left flaperon failure (A3) filter, the velocity sensor and right flaperon failure (A4) filter, the velocity sensor and rudder failure (A5) filter, the velocity sensor failure only (S1) filter, the velocity sensor and angle of attack filter (S2), the velocity sensor and normal acceleration failure (S4) filter, and the velocity sensor and roll rate failure (S5) filter. The velocity sensor and left stabilator failure (A1) filter and the velocity sensor and left flaperon failure (A3) filter compete for the majority of the probability. By 7.2 seconds, the velocity sensor and left stabilator failure (A1) filter and the velocity sensor failure only (S1) filter fight for the probability. From 7.0 to 7.8, the velocity sensor and left stabilator failure (A1) filter contains the majority of the probability with only two spikes of probability showing up in the velocity sensor failure only (S1) filter. At 7.8 seconds, it appears that the probability is contained within the velocity sensor and left stabilator failure (A1) filter. By the established rating system, this scenario is assigned a "Good" with a delay between failure insertion and probability lock of 1.8 seconds. Comparison with Figure 4.74 illustrates the differences for this dual failure combination using a single pulse every three seconds versus continuous excitation.

Based upon the data presented, it would seem possible to provide good algorithm performance to all of the failure conditions listed in Table 4.1 by increasing the excitation signals. Recall the excitation signals were "sized" for single failures, i.e., assuming that all actuators were available to apply needed system excitation until the first declared failure. Dual actuator failures reduced the amount of excitation in many scenarios, resulting in insufficient excitation for proper failure identification. Note that "resizing" could be accomplished for each declared first failure; however, time constraints did not allow the investigation of this hypothesis. Also, "resized" dither signals might still be able to meet the established subliminal dither criteria.

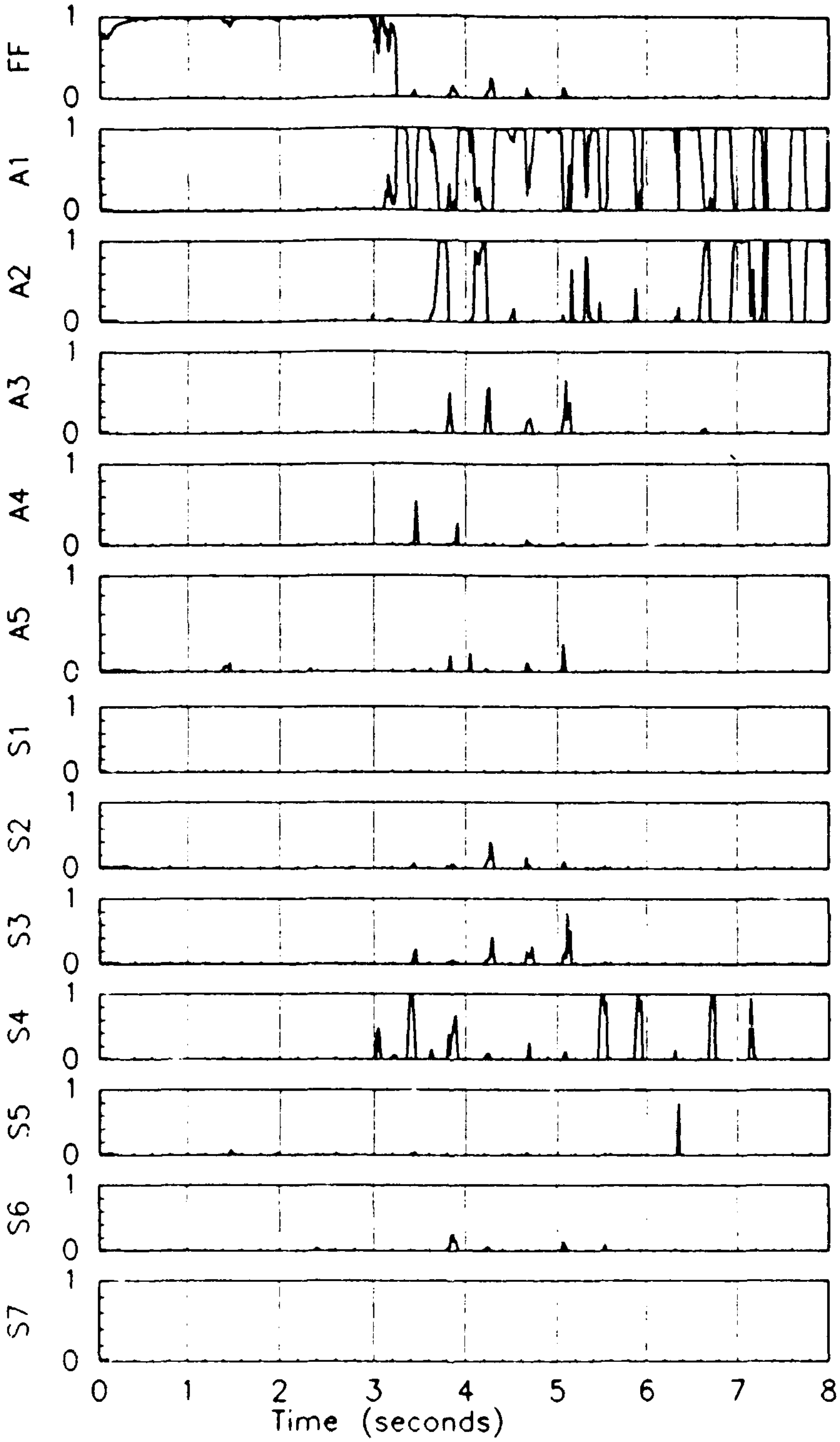

Figure 4.81 Probabilities for a left stabilator failure followed by a right stabilator failure using a sinusoidal dither signal

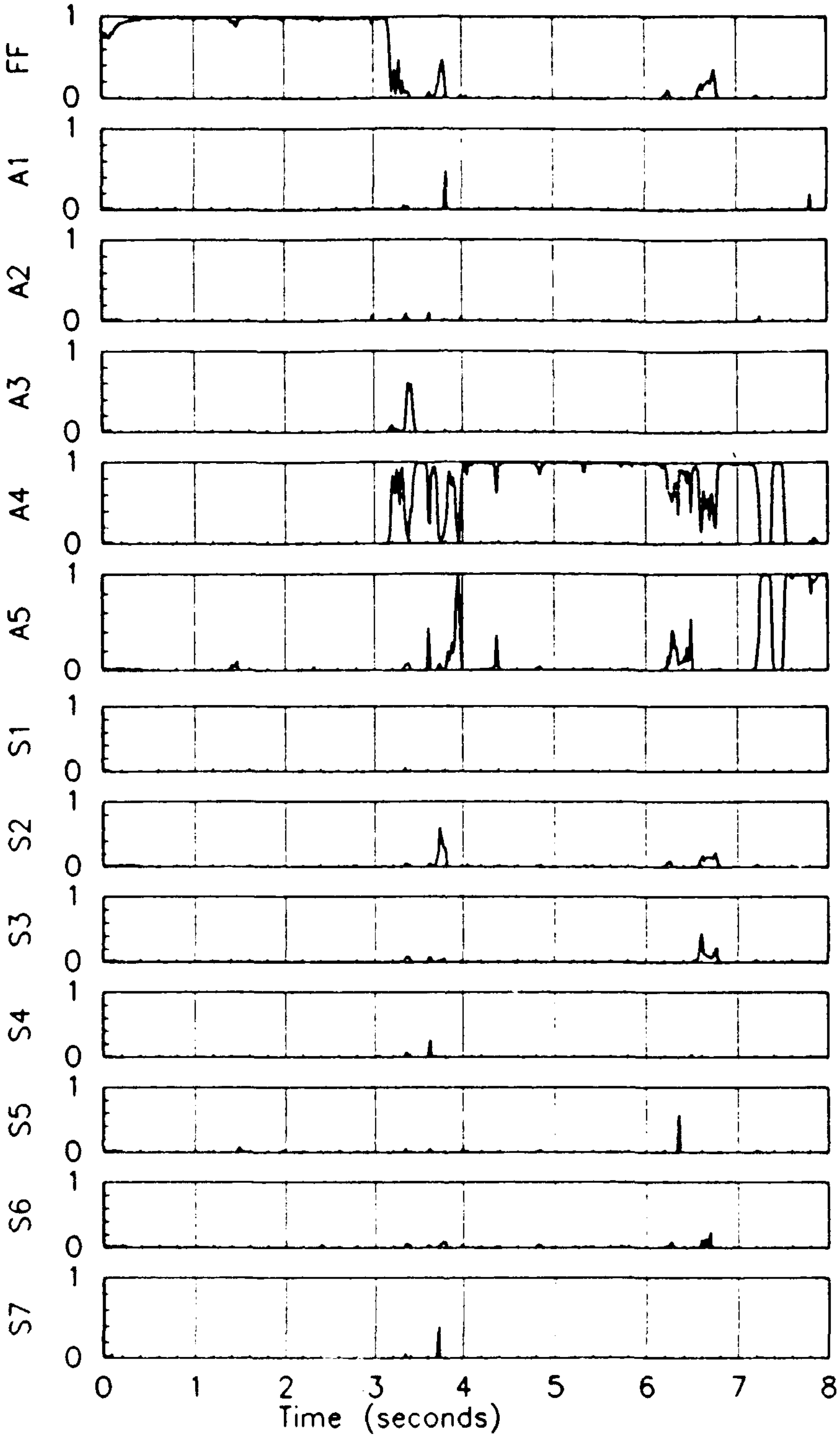

Figure 4.82 Probabilities for a right flaperon failure followed by a rudder failure using a sinusoidal dither signal

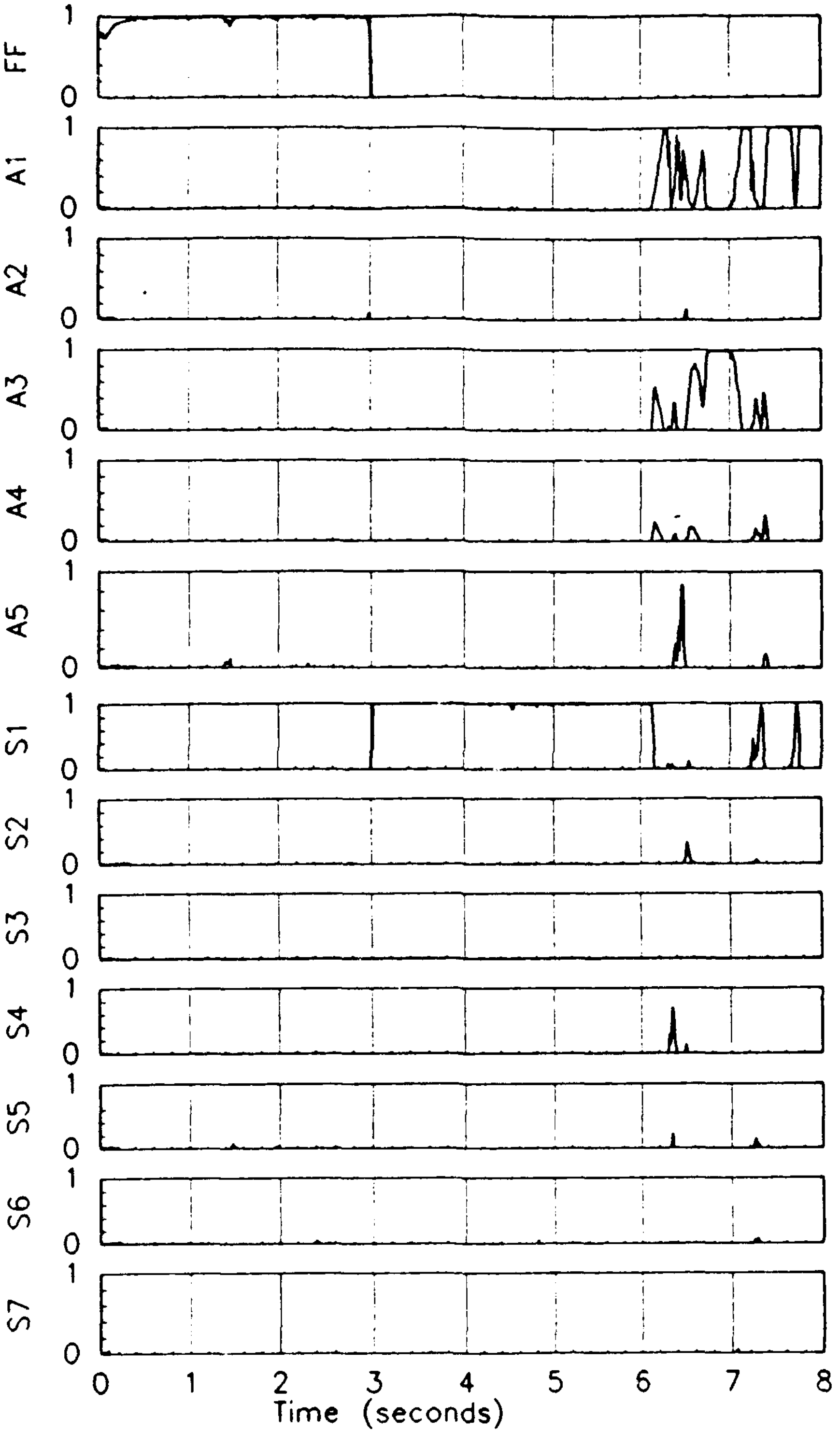

Figure 4.83 Probabilities for a velocity sensor failure followed by a left stabilator failure using a sinusoidal dither signal

4.4.3 Residual Monitoring

Figures 4.84 - 4.92 demonstrate the usefulness of residual monitoring. The failure scenario is a multiple failure of the left stabilator and the velocity sensor, Figure 4.74. The left stabilator failure is induced at 3.0 seconds, and the velocity sensor failure is induced at 6.0 seconds. Due to file space limitations, the residuals are shown only from 4.0 to 7.0 seconds. Three filters were chosen to demonstrate the residuals' characteristics in a multiple failure: the fully functional filter, the left stabilator (only) failure filter, and the left stabilator and velocity sensor failure filter. For each of the filters, three residuals are presented that portray the residuals' characteristics: the velocity residual, the pitch rate residual, and the normal acceleration residual. Other residuals provide additional voting in this case as well (angle of attack, yaw rate, and lateral acceleration). To conserve space, these residuals are not presented in this section. Figures 4.84 - 4.86 present the residuals for the fully functional filter. Figure 4.84 is the velocity residual, Figure 4.85 is the pitch rate residual, and Figure 4.86 is the normal acceleration residual. It is clear that, from 6.0 seconds forward, all of these residuals are well outside of the 3σ bounds.

Figures 4.87 - 4.89 present the residuals for the left stabilator (only) failure filter. Figures 4.87 - 4.89 are the velocity, pitch rate, and normal acceleration residuals, respectively. Again, it is clear that these residuals violate the 3σ bounds beyond 6.0 seconds. Comparison of these figures with Figures 4.84 - 4.86 indicate the residuals are close, except for a few differences, most notably in the pitch rate residual. Figure 4.85 violates the 3σ bound from 4.65 to approximately 5.6 seconds. Figure 4.88 contains only one small spike that violates the 3σ bounds during this time frame. If these were the only three residuals in the probability computation for the filters, one might deduce that the filter corresponding to Figures 4.87 - 4.89 would capture more of the probability than the filter containing Figures 4.84 - 4.86. This is exactly the case. The left stabilator failure (only) filter contains a large portion of the probability until 6.0 seconds. A direct relationship exists between the residuals and the probabilities as shown mathematically in Chapters 2 and 3.

Figures 4.90 - 4.92 present the velocity, pitch rate, and normal acceleration residuals, respectively, for the left stabilator and velocity sensor failure filter. From Figure 4.90, it is obvious that this filter has the incorrect hypothesis until 6.0 seconds, and the correct hypothesis thereafter. The residual moves back within the 3σ bounds at 6.0 seconds. Figure 4.91 demonstrates that the pitch rate residual violates the 3σ bounds at 4.7

seconds and continues to do so until 6.0 seconds. At 6.0 seconds, the residual moves within the 3σ bounds and remains within the bounds. This is another clear signal that this filter's hypothesis is correct. Figure 4.92 provides additional confirmation of the correct hypothesis from 6 seconds and beyond. The normal acceleration residual clearly violates the 3σ bounds and moves back within the bounds at 6.5 seconds. It is difficult to determine whether the residual will violate the bounds after 7.0 seconds, but based upon the probability trace, it is doubtful.

This section was intended to provide the reader with insight into the relationship between residuals and probabilities and to demonstrate the usefulness of residual monitoring. Filters based on an incorrect hypothesis can provide a clear "poor hypothesis" vote through residual monitoring, while filters whose residuals are within the 3σ bounds can provide a clear "correct hypothesis" vote. All of the filters together provide a complete picture of the correct failure. This technique is very useful for sinusoidal dithering, but has not been as useful for single dither pulses. Sinusoidal dithering tends to differentiate clearly between good and poor filter hypotheses. Residuals generated by the application of dither pulses don't provide sufficient residual signal characteristics from which to draw conclusions.

Practical implementation would warrant the use of a mixture of dither signals to exploit the "good" characteristics of each dither wave form. Several versions of each wave form could be developed. Within each wave form, the user could tune the dither signal to "look" for a specific failure. An algorithm designed to identify failures could implement each of the specifically tuned forms to enhance its ability to identify the failure(s). Single dither pulses are useful in providing clean strong pulses which have good identification characteristics. However, the associated filter residuals are of very little use in resolving ambiguities. Sine waves provide good identification characteristics but may not be the best wave form in every flight regime/scenario. However, residuals generated by a sine wave dither form have demonstrated their usefulness in resolving ambiguities. A fault detection algorithm should combine these forms to produce a synergistic effect, good failure detection, isolation, and the ability to resolve any ambiguities that may arise. The use of subliminal dither signals provide failure detection without interfering with the handling qualities of the aircraft. This is highly desirable when a subliminal dither signal can provide good performance. For situations in which the performance is reduced, the pilot could enable (at his discretion) a nonsubliminal signal to identify the failure.

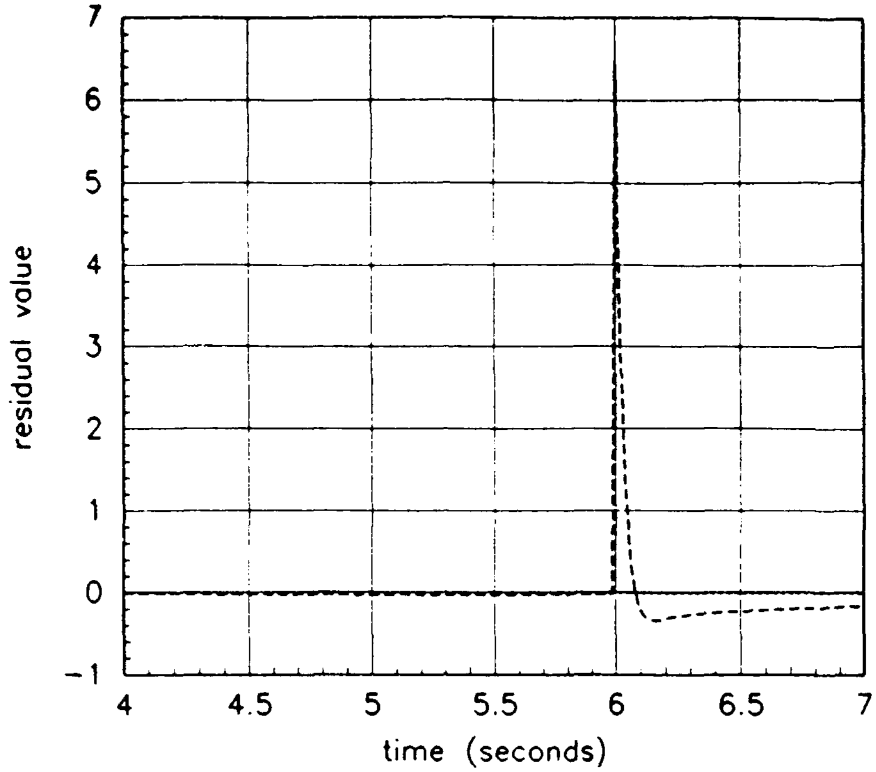

Figure 4.84 Velocity residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure

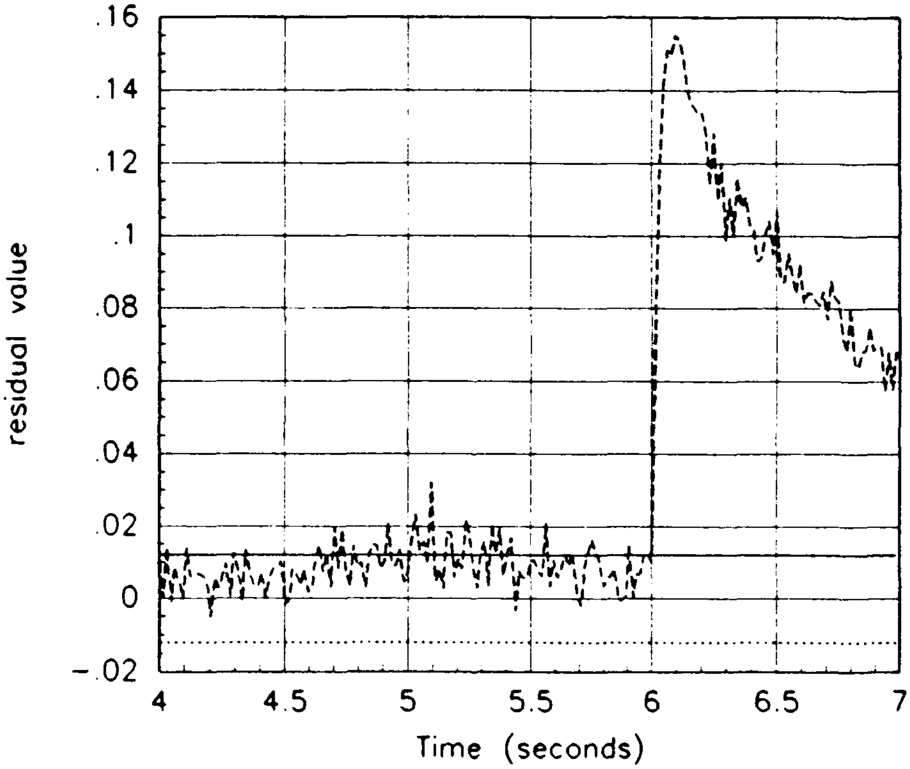

Figure 4.85 Pitch rate residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure

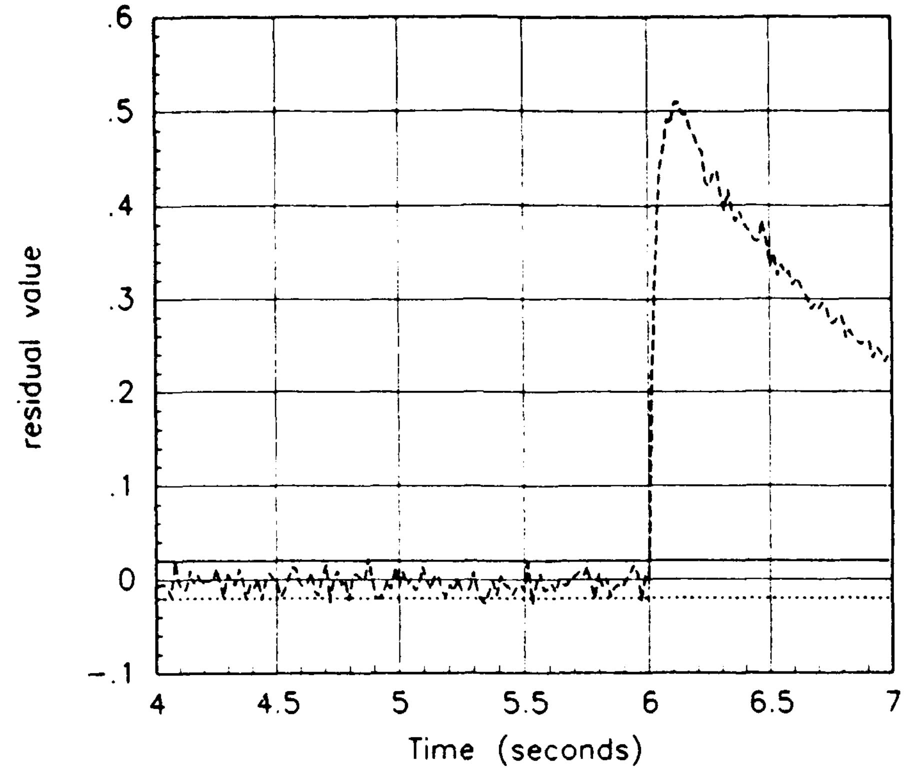

Figure 4.86 Normal acceleration residual for the no failure filter for a left stabilator failure followed by a velocity sensor failure

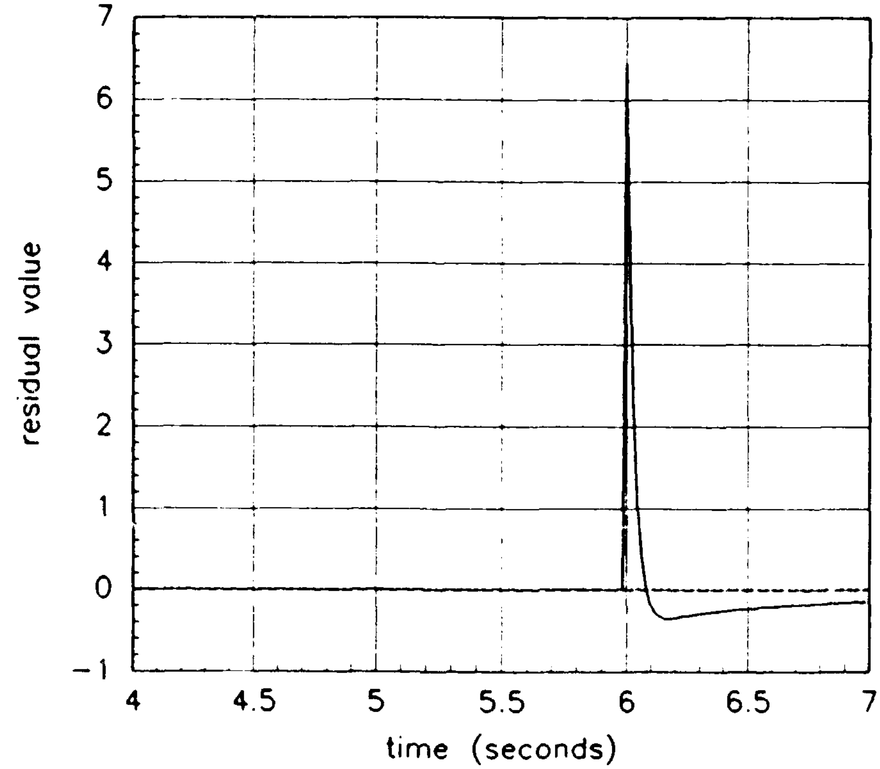

Figure 4.87 Velocity residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure

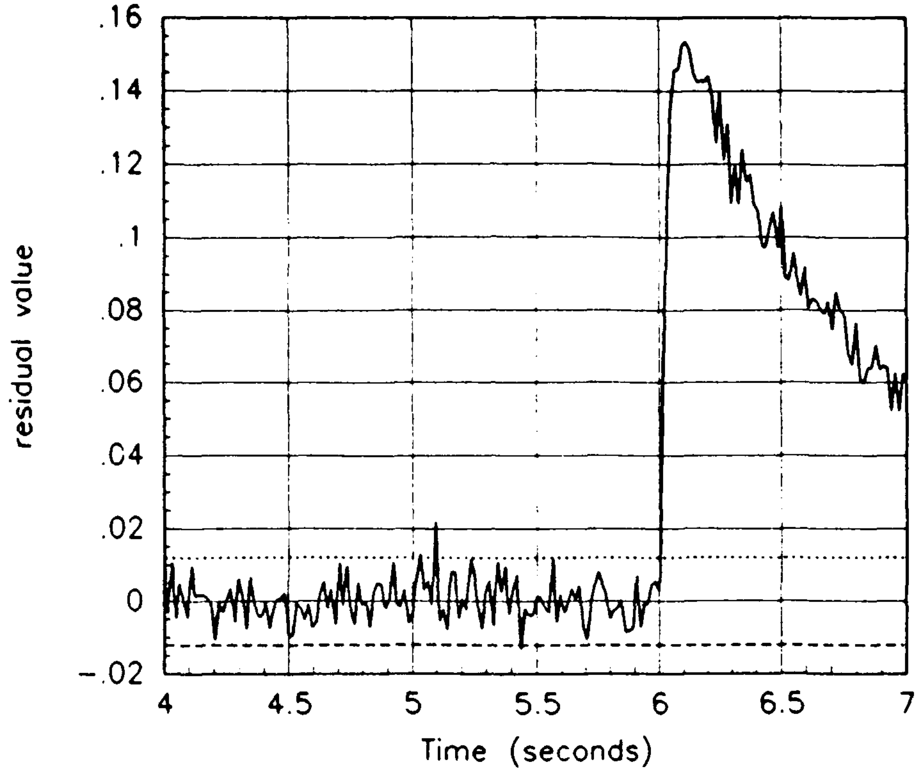

Figure 4.88 Pitch rate residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure

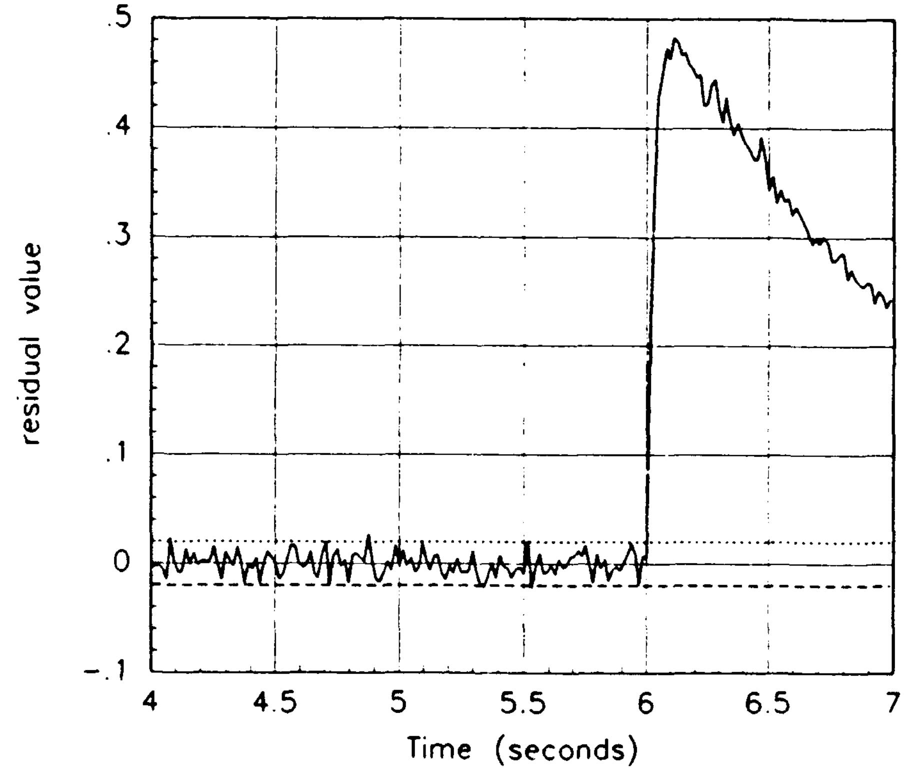

Figure 4.89 Normal acceleration residual for the left stabilator filter for a left stabilator failure followed by a velocity sensor failure

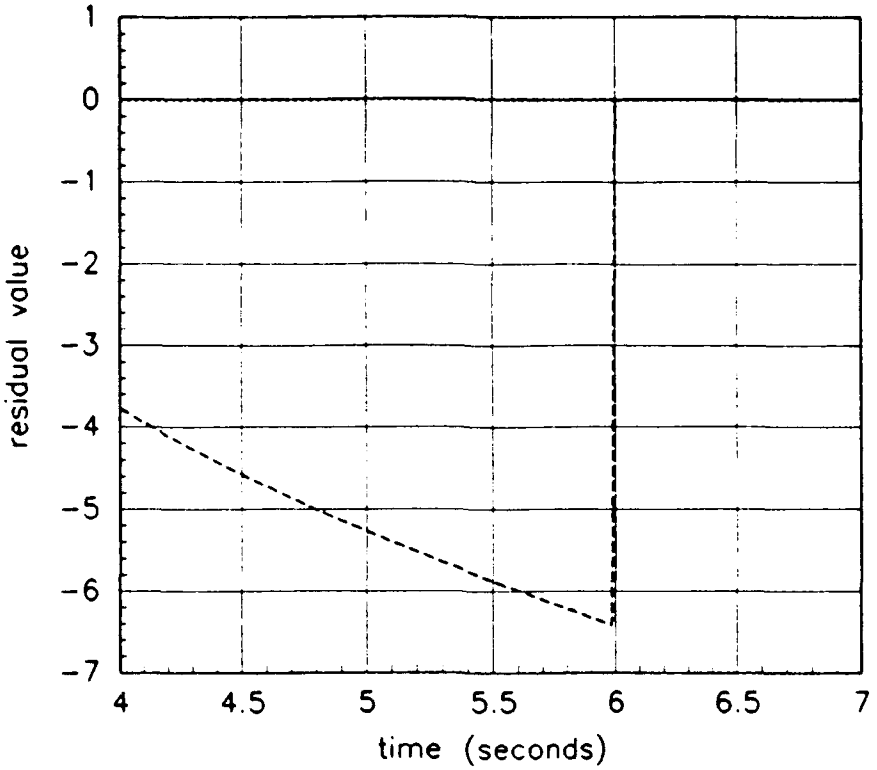

Figure 4.90 Velocity residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure

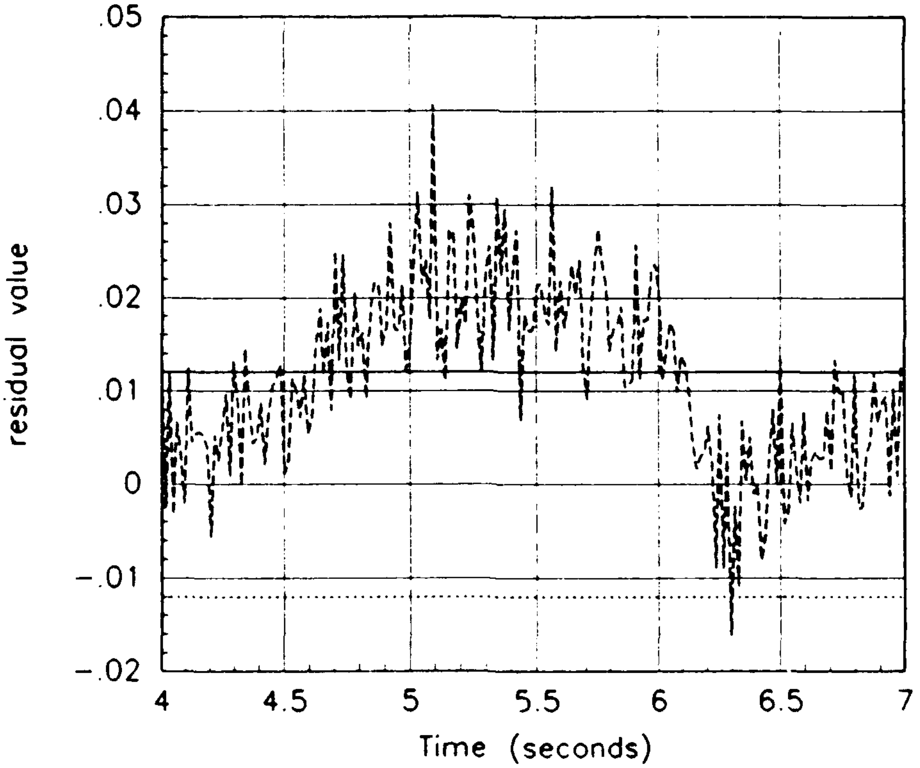

Figure 4.91 Pitch rate residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure

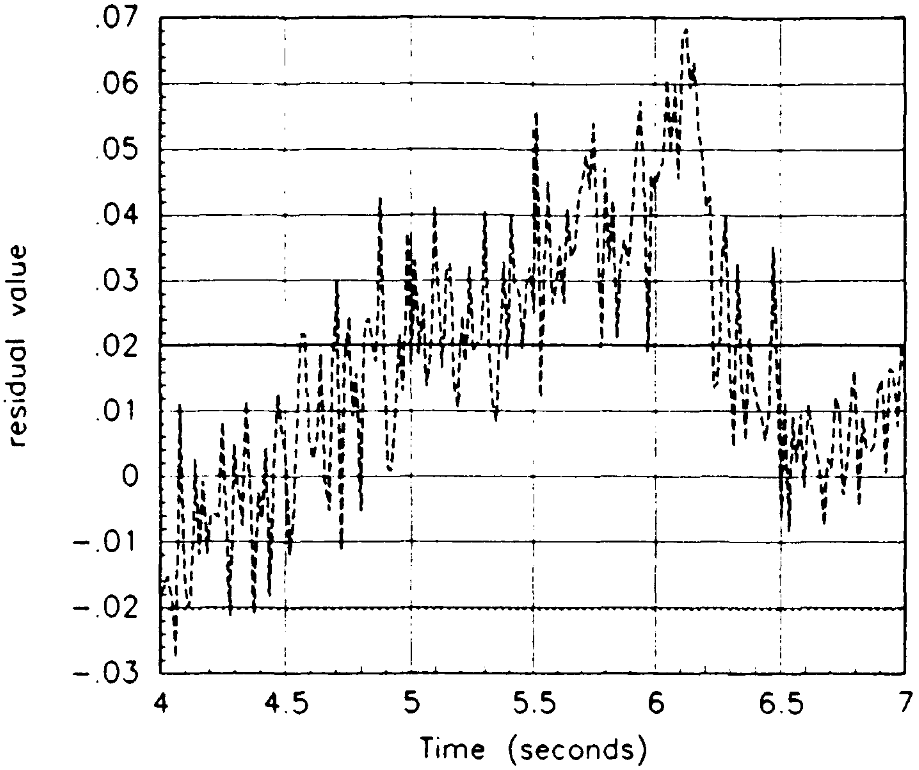

Figure 4.92 Normal acceleration residual for the left stabilator and velocity sensor failure filter for a left stabilator failure followed by a velocity sensor failure

4.4.4 Increased Dither Signal Strength

Recall from Table 4.1, the right flaperon and pitch rate sensor failure performance rating ("Poor"). By increasing the strength of the dither signal, the performance can be significantly improved. Figure 4.93 demonstrates the system performance for a right flaperon failure inserted at 3.0 seconds followed by a pitch rate sensor failure induced at 6.0 seconds. The level 1 right flaperon bank was brought "on-line" at approximately 3.25 seconds. Based upon Figure 4.93, the subjective rating system would assign this result as "Fair". Figure 4.94 allows the reader to view the effects of the dither signal on the state variables. At 3.0 seconds, a sinusoid is superimposed on the dither signal. This provides increased system excitation, resulting in improved performance; however, the performance is still not "Good". Increasing the dither signal a second time results in "Good" system performance as demonstrated by Figure 4.95. Both failures in Figure 4.95 are identified and locked immediately. Figure 4.96 displays the effect of the increased dither signal on the state variables. It is clear that this dither signal is not subliminal; note the A_{ncg} magnitude here. The results from this section are included in Table 4.2. The purpose of this section has been to demonstrate that an increased strength dither signal could increase the subjective rating from "ND" or "Poor" to "Good". This increase in performance is directly related to the type and strength of the dither signal.

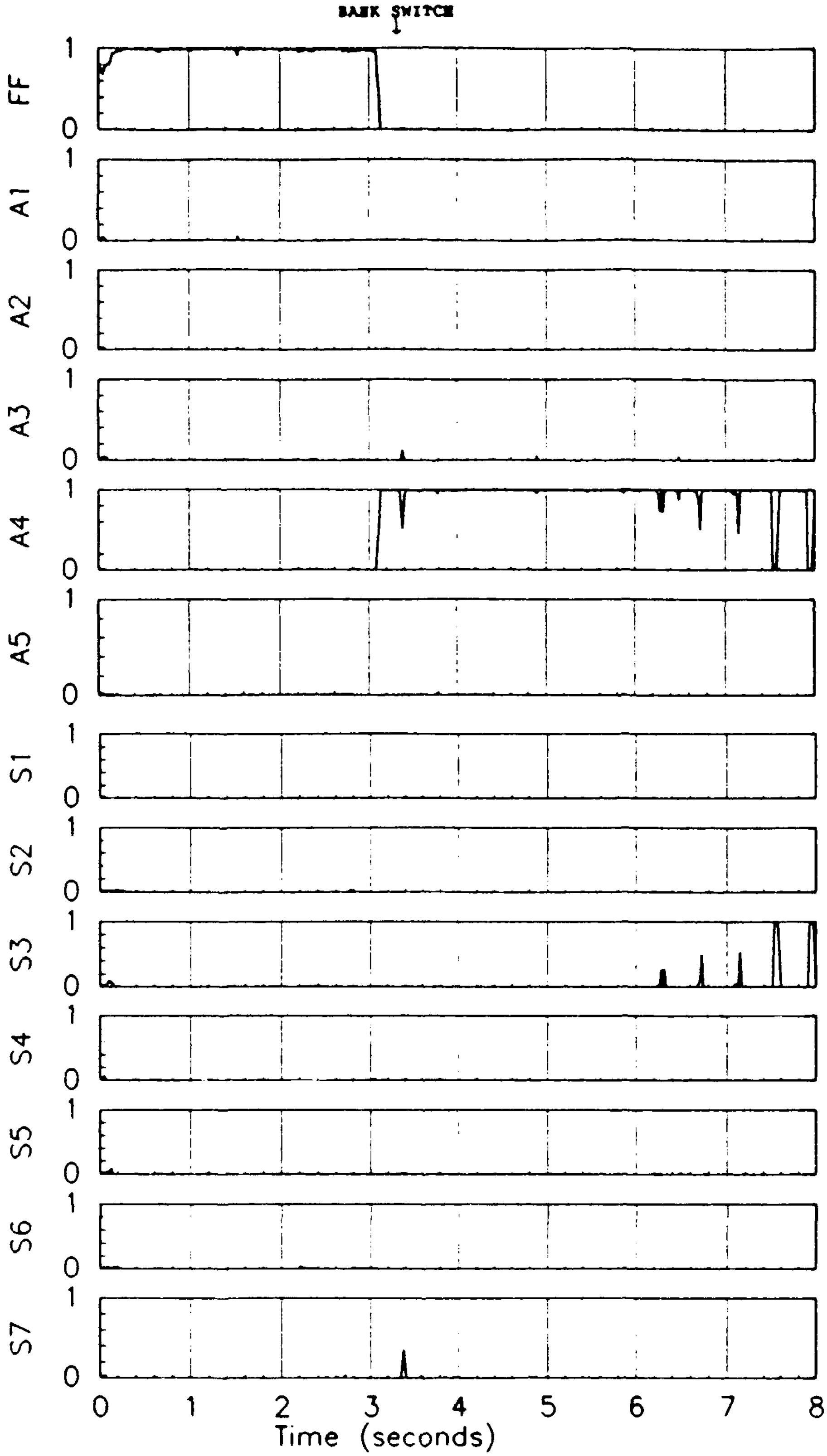

Figure 4.93 Probabilities for a right flaperon failure followed by a pitch rate sensor failure using an increased strength dither signal

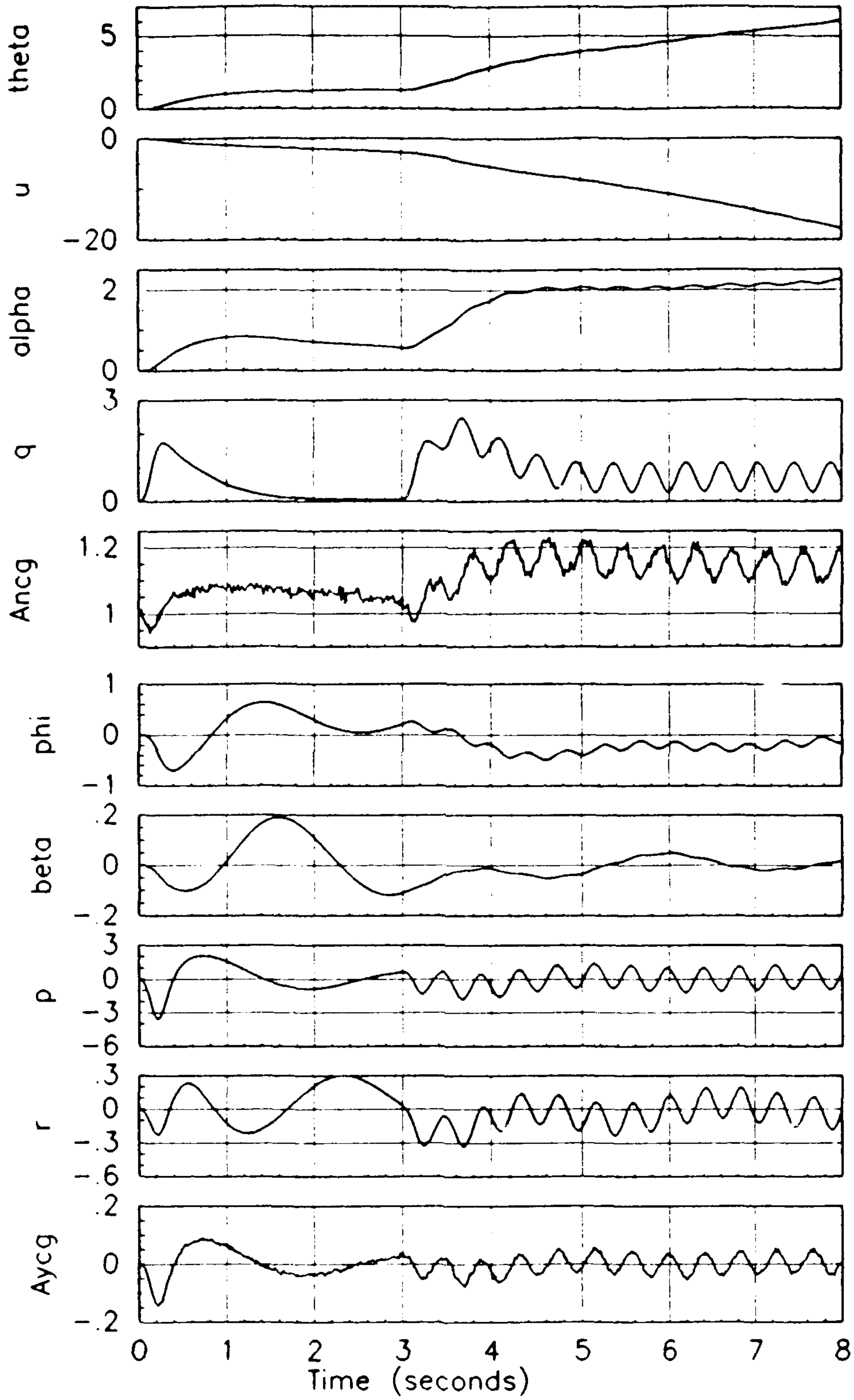

Figure 4.94 States for a right flaperon failure followed by a pitch rate sensor failure using an increased strength dither signal

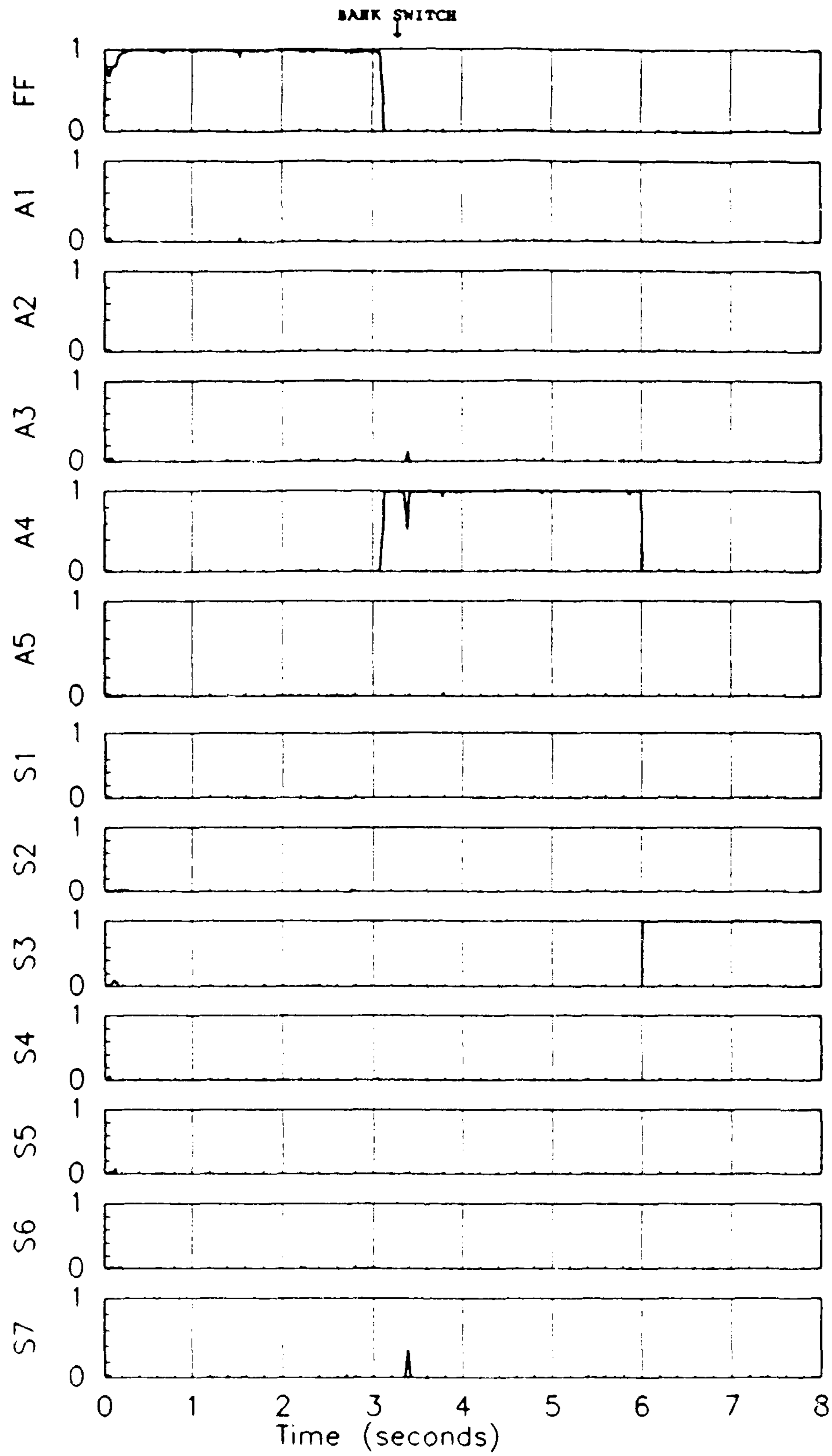

Figure 4.95 Probabilities for a right flaperon failure followed by a pitch rate sensor failure using a further increased strength dither signal

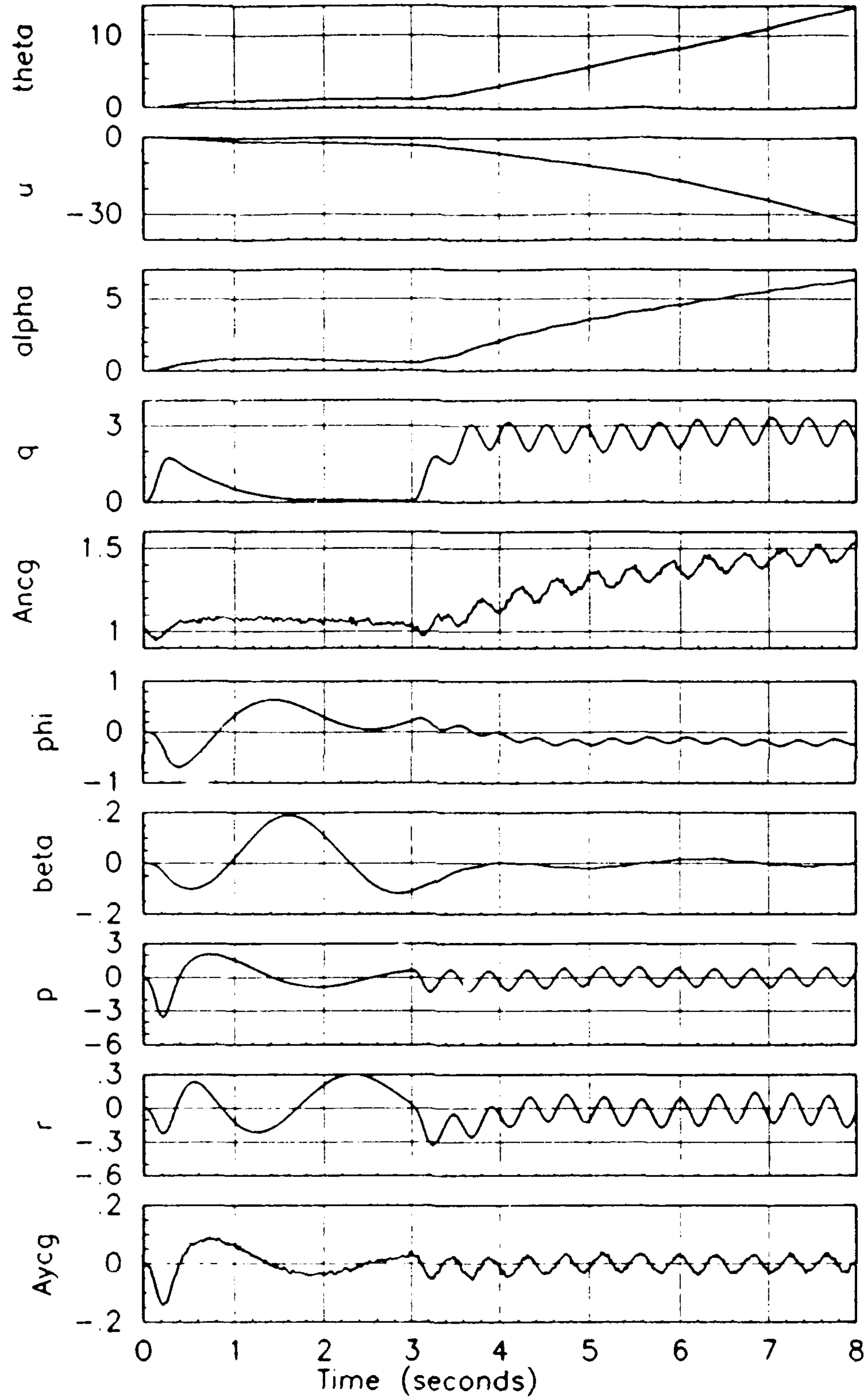

Figure 4.96 States for a right flaperon failure followed by a pitch rate sensor failure using a further increased strength dither signal

	LS	RF	RD	VS	AOA	PR	NZ	RR	YR	NY
LS		Good	Good	Good	Good	Good	Good	Good	Good	Good
RS	Fair	Good	Good	Poor	Good	Good	Good	Good	Good	Good
LF	Poor	Good	Poor	Fair	Good	Good	Good	Good	Good	Good
RF	Poor		Poor	Fair	Good	Good	Good	Good	Good	Good
RD	ND	Good		Fair	Good	Good	Good	Good	Good	Good
VS	Good	Good	Good		Good	Good	Good	Good	Good	Good
AOA	ND	Good	Good	Good		Good	Good	Good	Good	Good
PR	ND	Good	Poor	Fair	Good		Good	Good	Good	Good
NZ	Good	Good	Good	Good	Good	Good		Good	Good	Good
RR	ND	ND	Poor	Fair	Good	Good	Good		Good	Good
YR	ND	ND	ND	Poor	Good	Good	Good	Good		Poor
NY	Fair	Good	Good	Fair	Good	Good	Good	Good	Good	

Table 4.2 Revised Multiple Hard Failure Summary Matrix

4.4.5 Simultaneous Dual Failures

Simultaneous failures were developed for failures separated by 3.0, 0.5, and finally 0.1 seconds. For this thesis, failures separated by 0.1 seconds are defined as simultaneous. Four failure scenarios were investigated: a right flaperon failure and left stabilator failure, a normal acceleration sensor failure and pitch rate sensor failure, a roll rate sensor failure followed by a lateral acceleration sensor failure, and a roll rate sensor failure followed by a rudder failure. Figure 4.97 presents data for a right flaperon failure followed 3.0 seconds later by a left stabilator failure. The dither signal implemented is the sinusoidal dither signal used in the previous section. At 3.6 seconds, the level 1 right flaperon failure (A4) bank is brought "on-line". The second failure is induced at 6.0 seconds. The left stabilator failure is detected and locked within 0.2 seconds. This dual failure was assigned a rating of "Good" in Table 4.2. Figure 4.98 presents the data for the same dual failure with the failures separated by 0.5 seconds instead of 3.0 seconds. The right flaperon is identified and the appropriate level 1 bank is brought "on-line". The left stabilator failure is identified within 0.2 seconds. This dual failure is assigned a rating of "Good". Figure 4.99 presents the simultaneous failure of the right flaperon and the left stabilator. In this dual failure, the left stabilator is identified within 0.25 seconds. The level 1 left stabilator failure (A1) bank is brought "on-line". Both flaperons experience some probability spiking during the remainder of the simulation run. This dual failure scenario is assigned a rating of "Poor" by the subjective rating system.

Figure 4.100 portrays a normal acceleration sensor failure followed 3.0 seconds later by a pitch rate sensor failure. The normal acceleration sensor failure is inserted at 3.0 seconds and identified immediately. The level 1 normal acceleration sensor failure (S4) bank is brought "on-line". The pitch rate failure is induced at 6.0 seconds and identified at 6.5 seconds. The subjective rating system assigned this failure scenario a rating of "Good". Figure 4.101 presents the same failure scenario with the failures separated by 0.5 seconds instead of 3.0 seconds. The first failure is identified within 0.1 seconds, and the appropriate bank is brought "on-line". The second failure is identified 1.0 seconds after the insertion of the second failure at 3.5 seconds. This failure scenario is assigned a rating of "Good". Figure 4.102 demonstrates the same failure scenario with the failures induced simultaneously. Since the normal acceleration sensor failure has a much smaller identification delay, the normal acceleration sensor failure is identified first, and the appropriate bank is brought "on-line". The normal acceleration and pitch rate failure filter contains all of the probability at 4.4 seconds. The second failure requires 1.4 seconds to be properly identified. This failure scenario is assigned a rating of "Good", since both failures were identified and locked for the duration of the simulation run.

Figure 4.103 presents a roll rate sensor failure followed 3.0 seconds later by a lateral acceleration sensor failure. The roll rate sensor failure is induced at 3.0 seconds and identified almost immediately. The level 1 roll rate sensor failure (S5) bank is brought "on-line". The lateral acceleration sensor failure is induced at 6.0 seconds and identified by the roll rate sensor failure and lateral acceleration sensor failure filter almost immediately. This failure scenario is assigned a rating of "Good". Figure 4.104 presents the dual failure data with a 0.5 second separation between failures. Again, the results are good. The first failure is identified immediately after failure insertion at 3.0 seconds, and the appropriate level 1 bank is brought "on-line". The second failure is inserted at 3.5 seconds and identified by the roll rate sensor failure and lateral acceleration sensor failure filter (S7) at 3.7 seconds. This failure scenario is assigned a rating of "Good". Figure 4.105 presents the simultaneous failure scenario. Although not obvious from the figure, the roll rate sensor is identified almost immediately after failure insertion and the level 1 roll rate sensor failure (S5) bank is brought "on-line". The second failure is identified shortly thereafter by the roll rate sensor failure and lateral acceleration sensor failure (S7) filter. This scenario is assigned a rating of "Good".

Figure 4.106 presents the roll rate sensor failure followed 3.0 seconds later by a rudder failure. As is often the case with sensor failures, the roll rate sensor failure is identified almost immediately after failure insertion at 3.0 seconds. The level 1 roll rate sensor failure (S5) bank is brought "on-line". The rudder failure is not identified until 7.2 seconds (1.2 seconds after failure insertion). The roll rate sensor failure and rudder failure (A5) filter holds the probability from 7.2 seconds until the end of the simulation run. This scenario is assigned a rating of "Good". Figure 4.107 presents the same failure scenario with the failures separated by 0.5 seconds instead of 3.0 seconds. The first failure is identified immediately after insertion and the proper bank is brought "on-line". The insertion of the rudder failure at 3.5 seconds is not detected until 5.15 seconds. Some ambiguity with the roll rate sensor failure and left flaperon failure (A3) filter and the roll rate sensor and right stabilator failure (A2) filter is evident from the figure at 4.9 to 5.1 seconds. At 5.15 seconds, the roll rate sensor failure and rudder failure (A5) filter holds most of the probability, until approximately 6.4 seconds, at which point more ambiguity arises. The roll rate sensor failure and angle of attack sensor failure (S2) filter, the roll rate sensor failure only (S5) filter, and the roll rate sensor failure and yaw rate sensor failure (S6) filter all contain some portion of the probability from 6.2 seconds to 7.2 seconds. At 7.2 seconds, the roll rate sensor failure and rudder failure (A5) filter contains all of the probability until the end of the simulation run. This scenario is assigned a rating of "Good". Figure 4.108 presents the simultaneous failure case for the roll rate sensor failure and the rudder failure. The roll rate sensor failure is identified almost immediately after the insertion of the failure at 3.0 seconds. The level 1 roll rate sensor failure (S5) bank is brought "on-line". The roll rate sensor failure and rudder failure (A5) filter contains some probability spikes from 3.6 to 4.5 seconds. A small spike occurs in the roll rate sensor failure and lateral acceleration sensor failure (S7) filter at 3.7 seconds. A small amount of the probability can be found in the roll rate sensor failure and yaw rate sensor failure (S6) filter from 3.5 to 5.0 seconds. The roll rate sensor failure and rudder sensor failure (A5) filter contains the majority of the probability from 5.1 seconds until the end of the simulation. This simultaneous failure is assigned a rating of "Good".

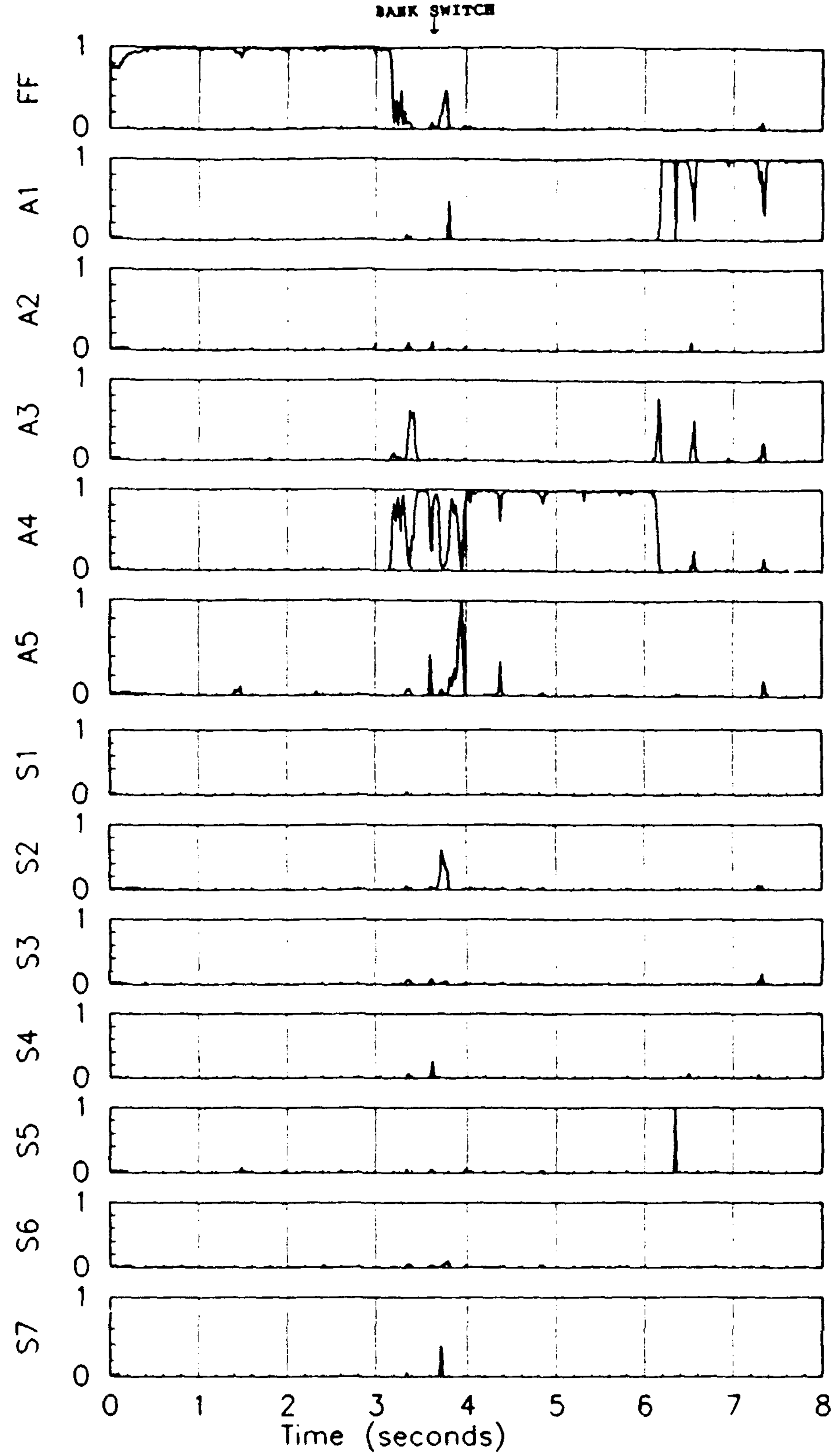

Figure 4.97 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 3.0 seconds.

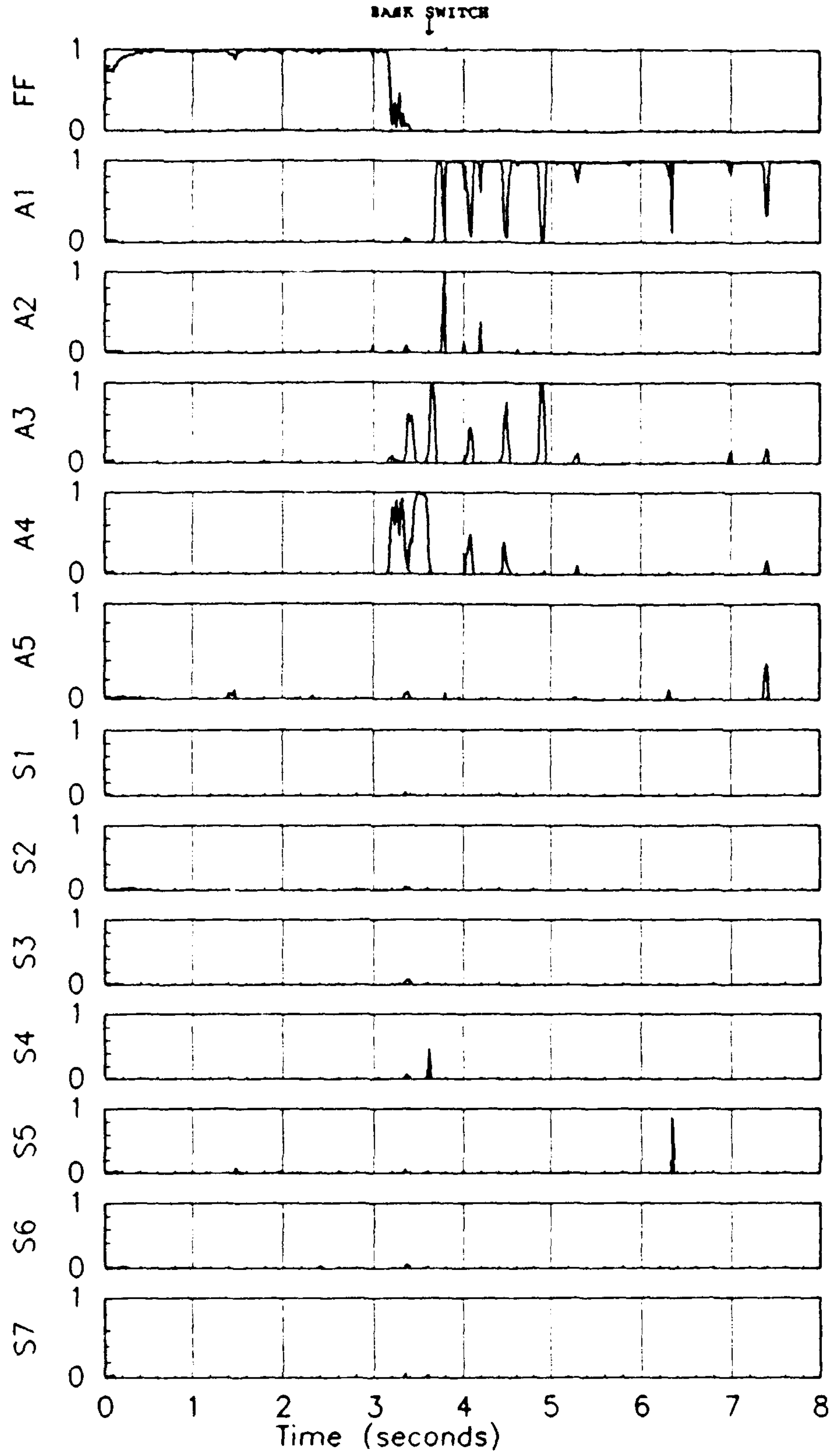

Figure 4.98 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 0.5 seconds.

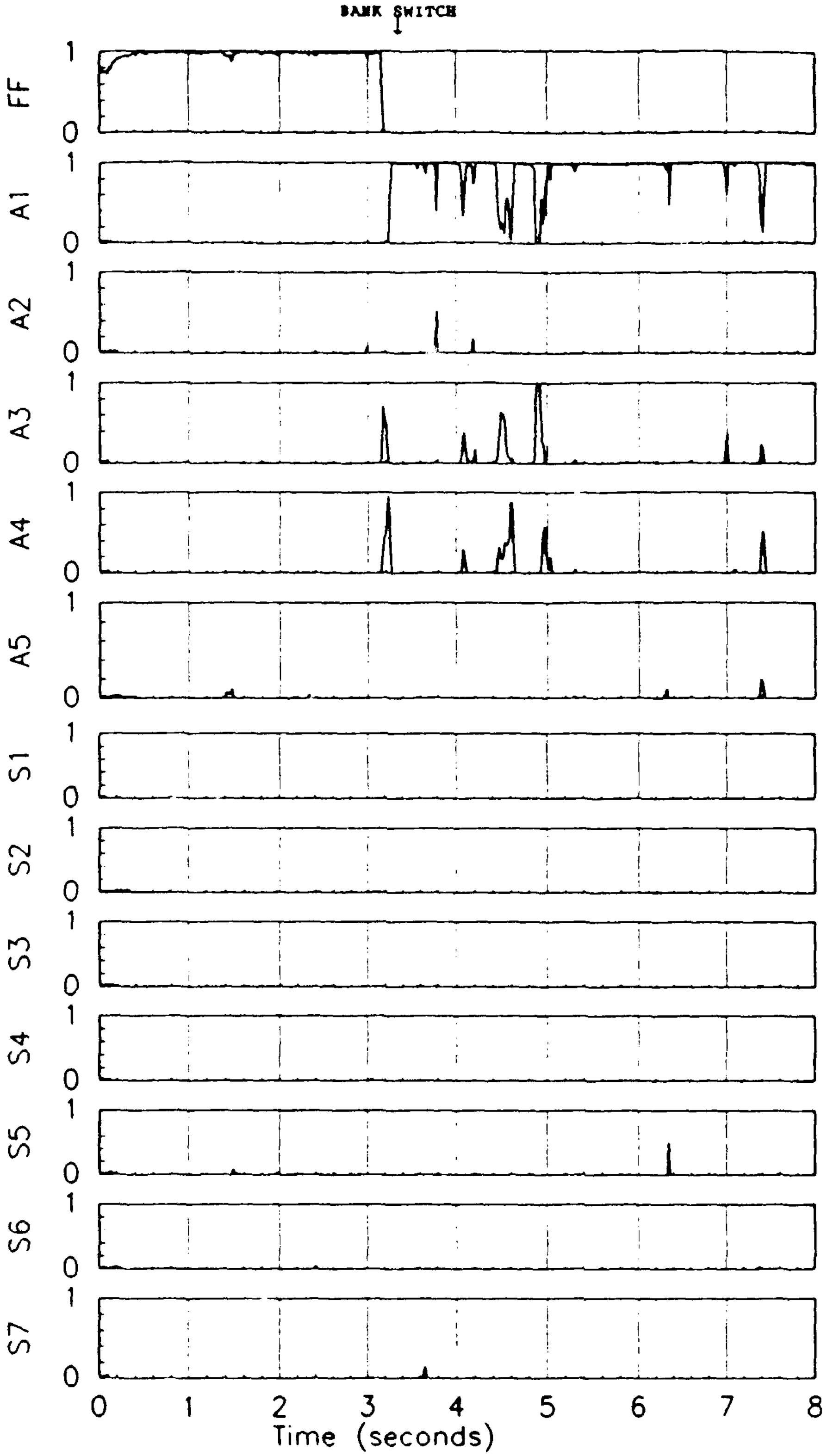

Figure 4.99 Probabilities for a right flaperon failure followed by a left stabilator failure separated by 0.1 seconds.

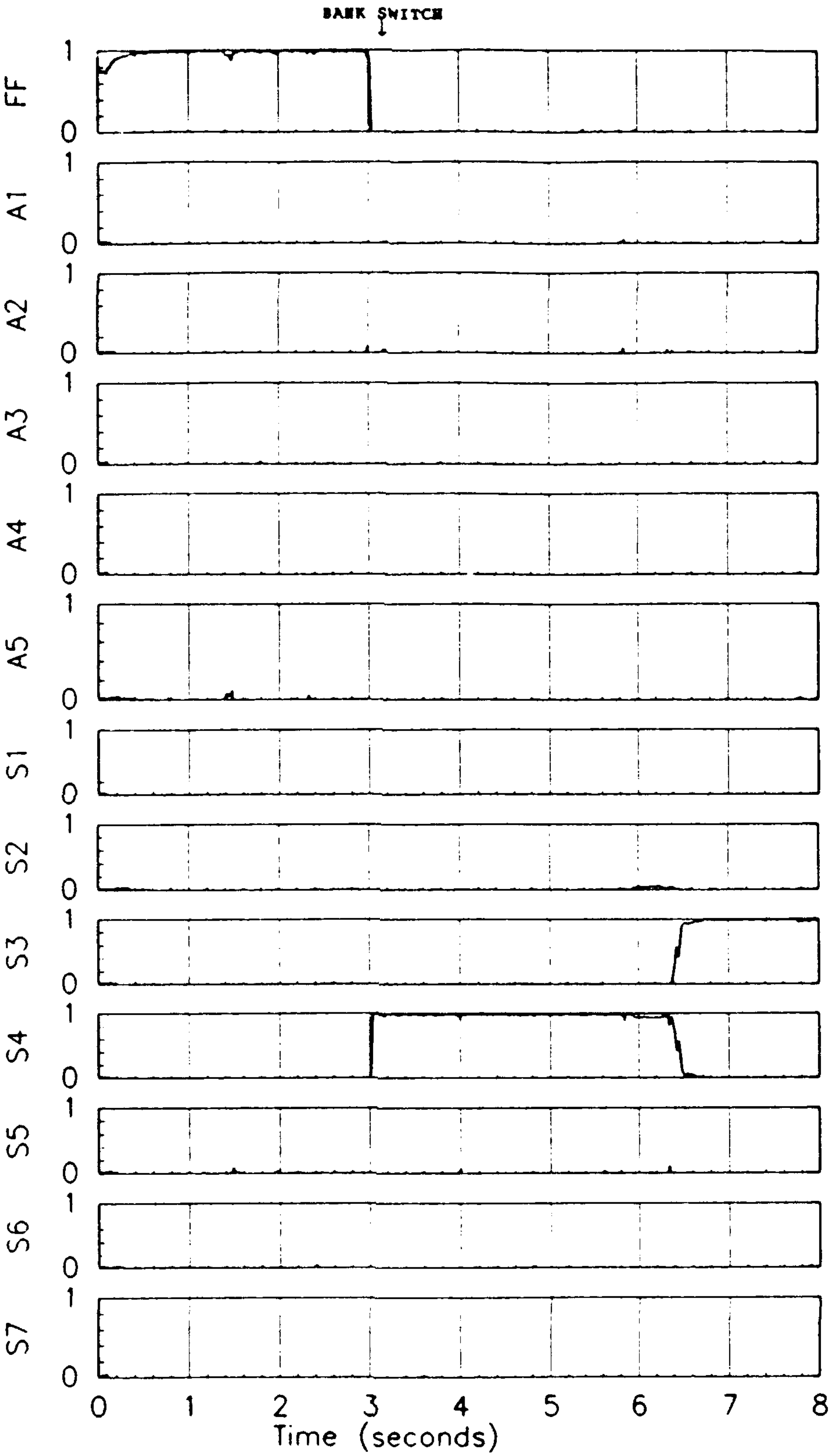

Figure 4.100 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 3.0 seconds.

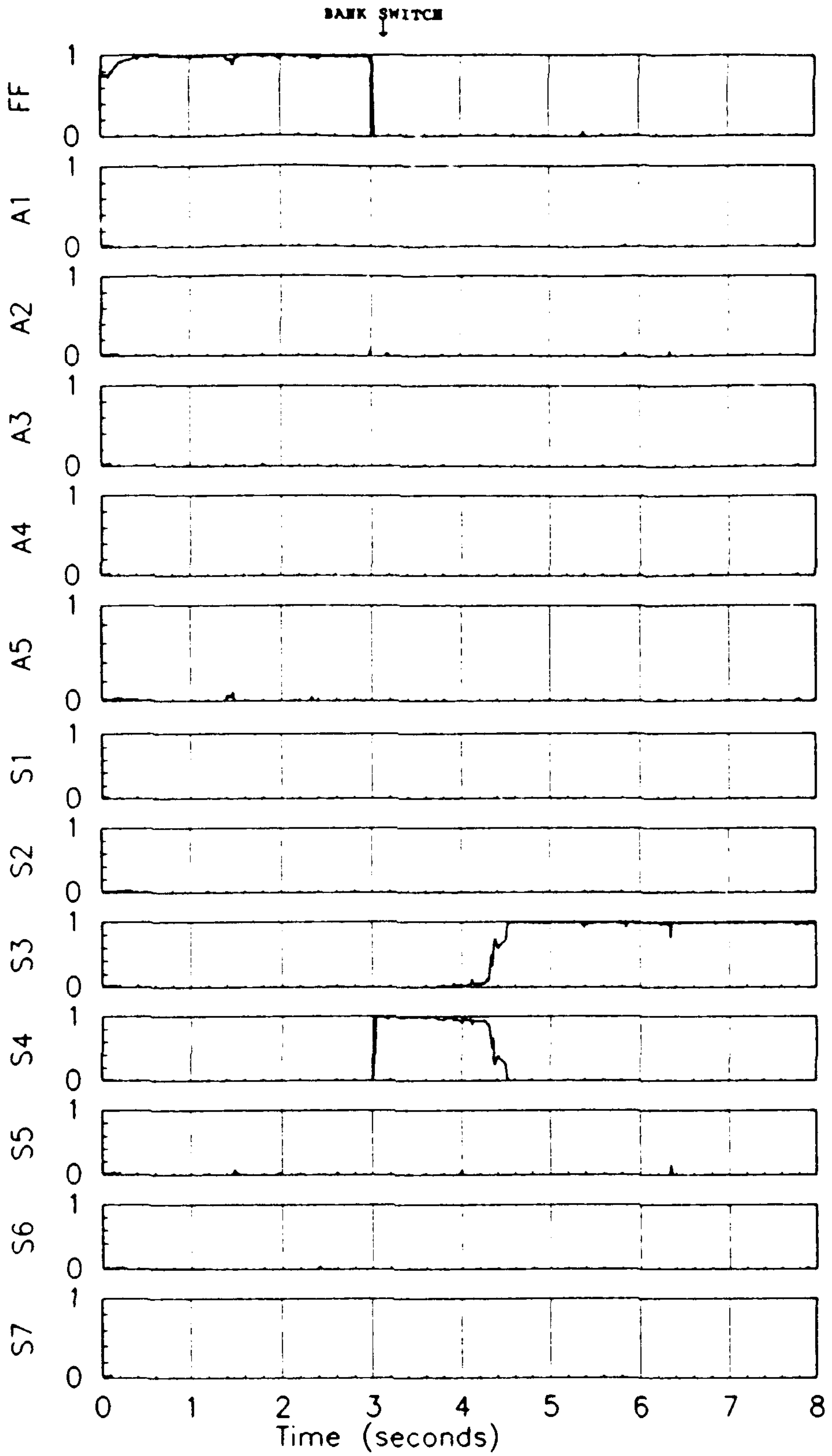

Figure 4.101 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 0.5 seconds.

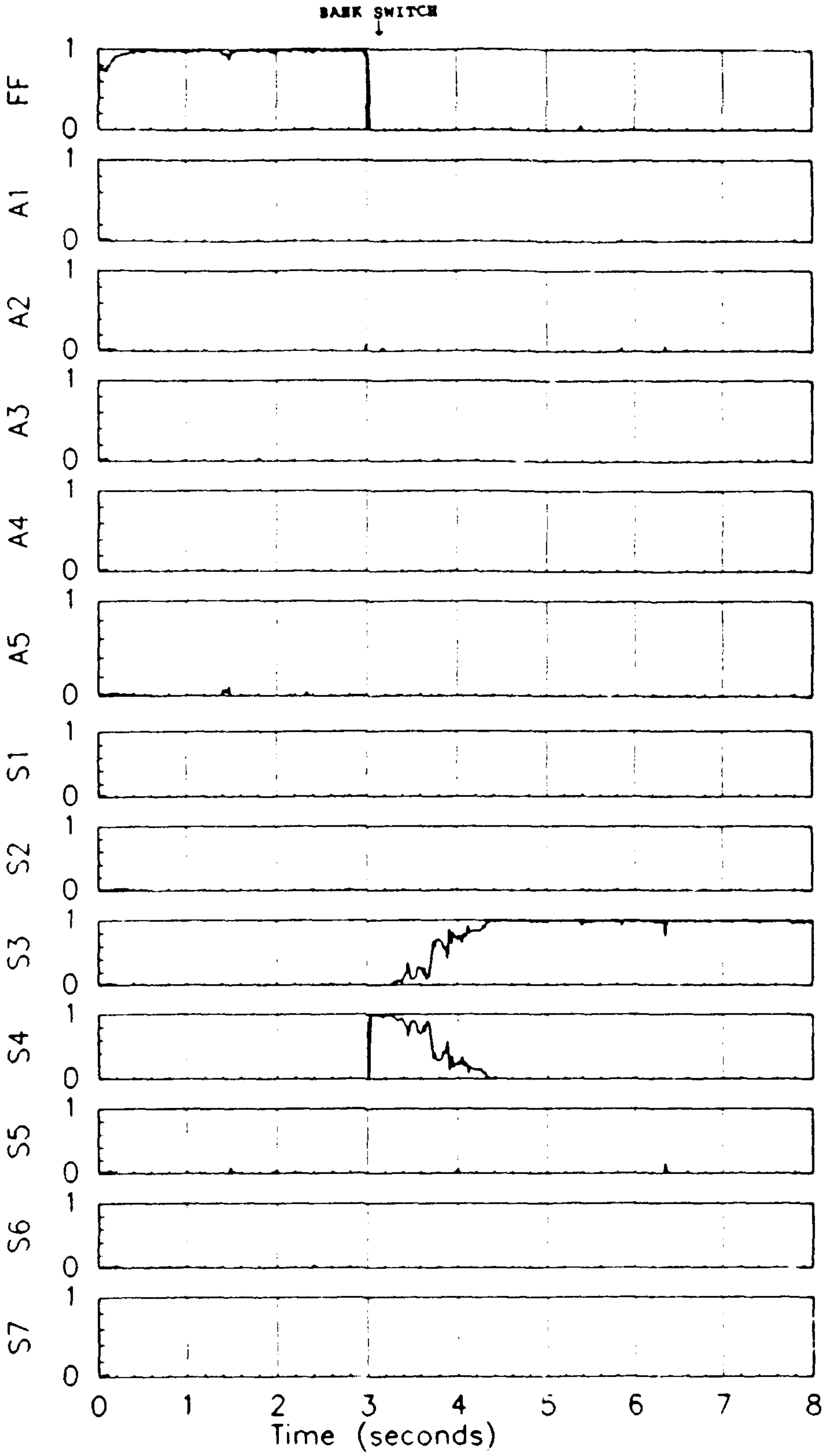

Figure 4.102 Probabilities for a normal acceleration sensor failure followed by a pitch rate sensor failure separated by 0.1 seconds.

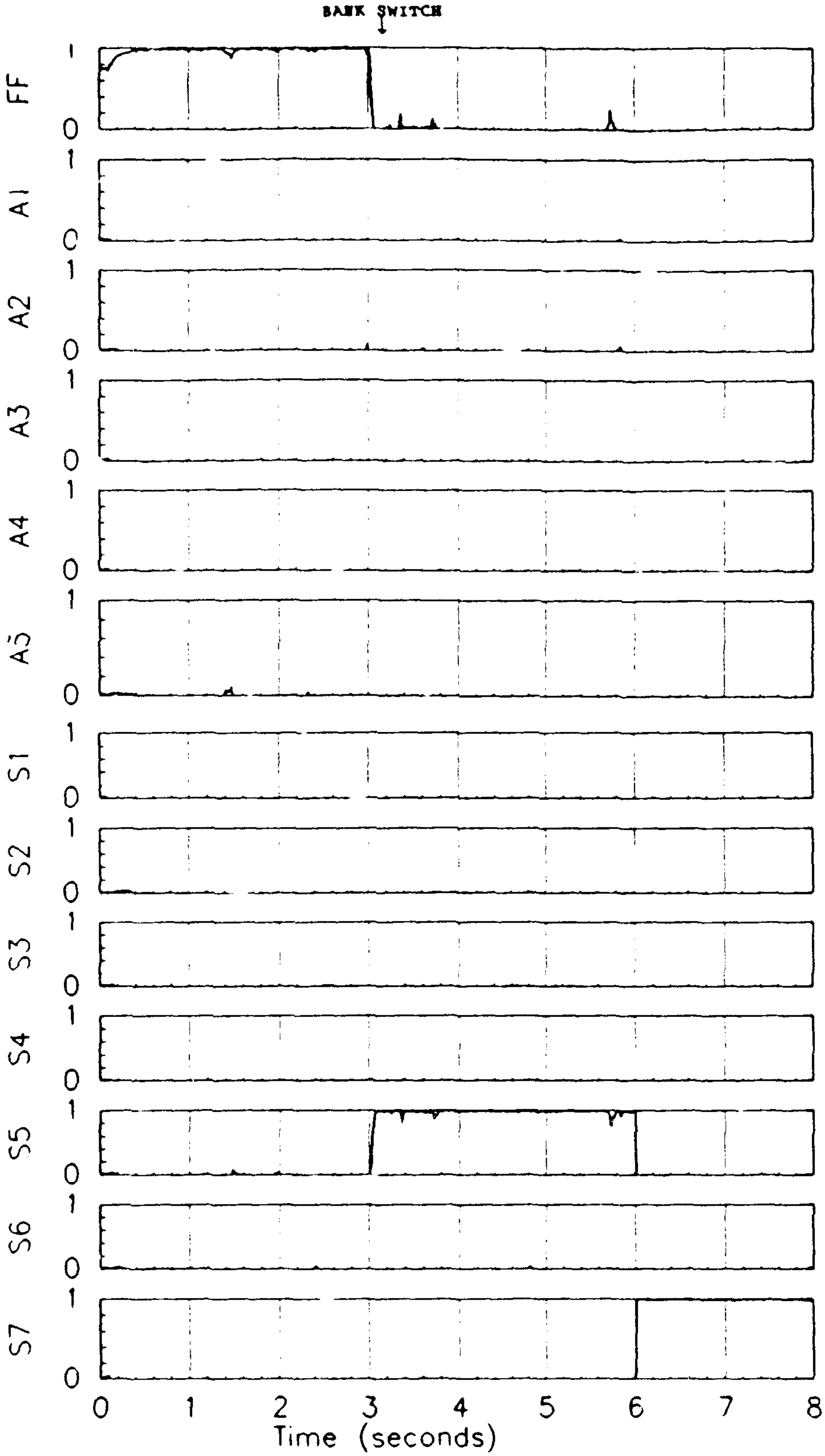

Figure 4.103 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 3.0 seconds.

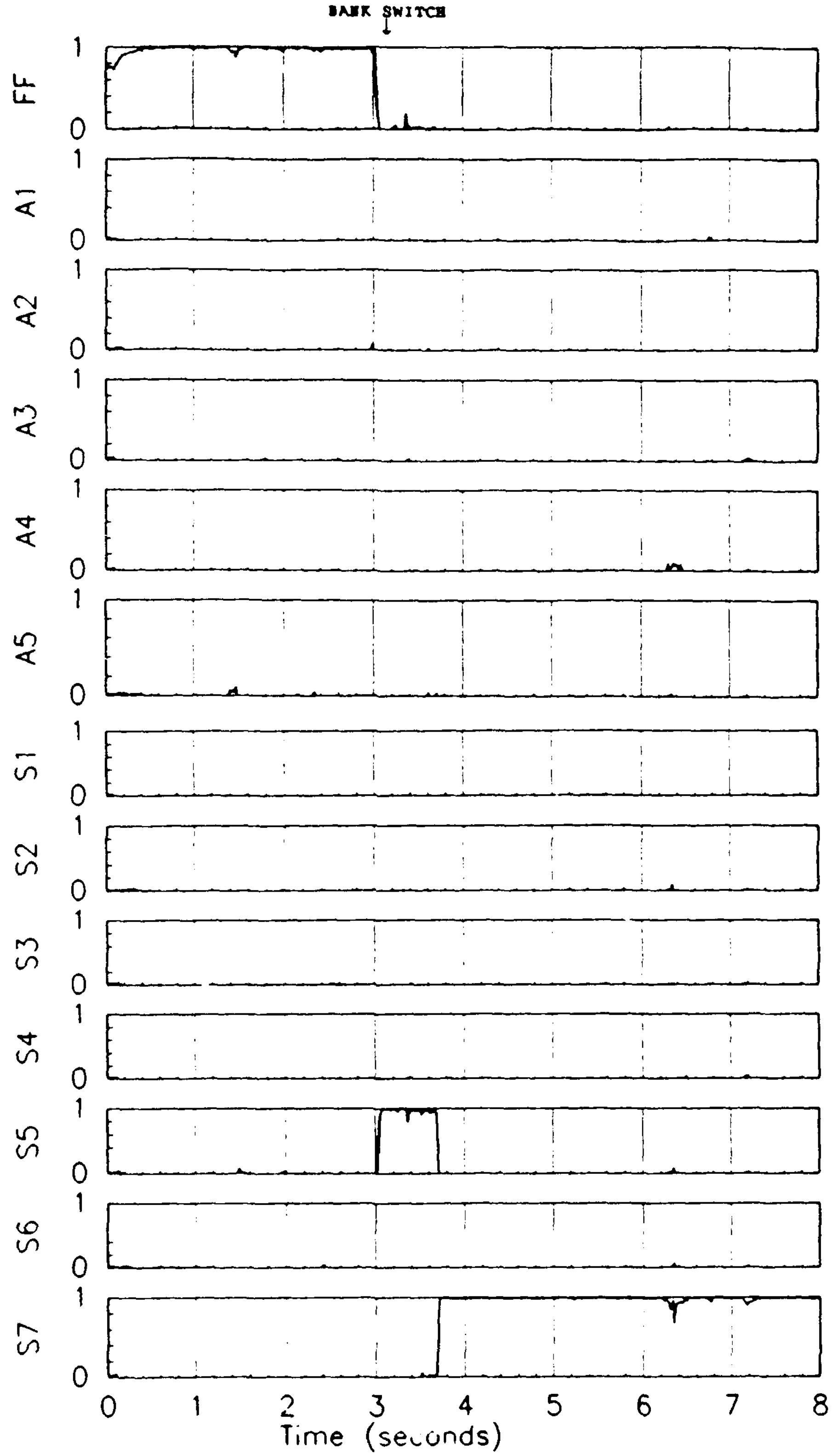

Figure 4.104 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 0.5 seconds.

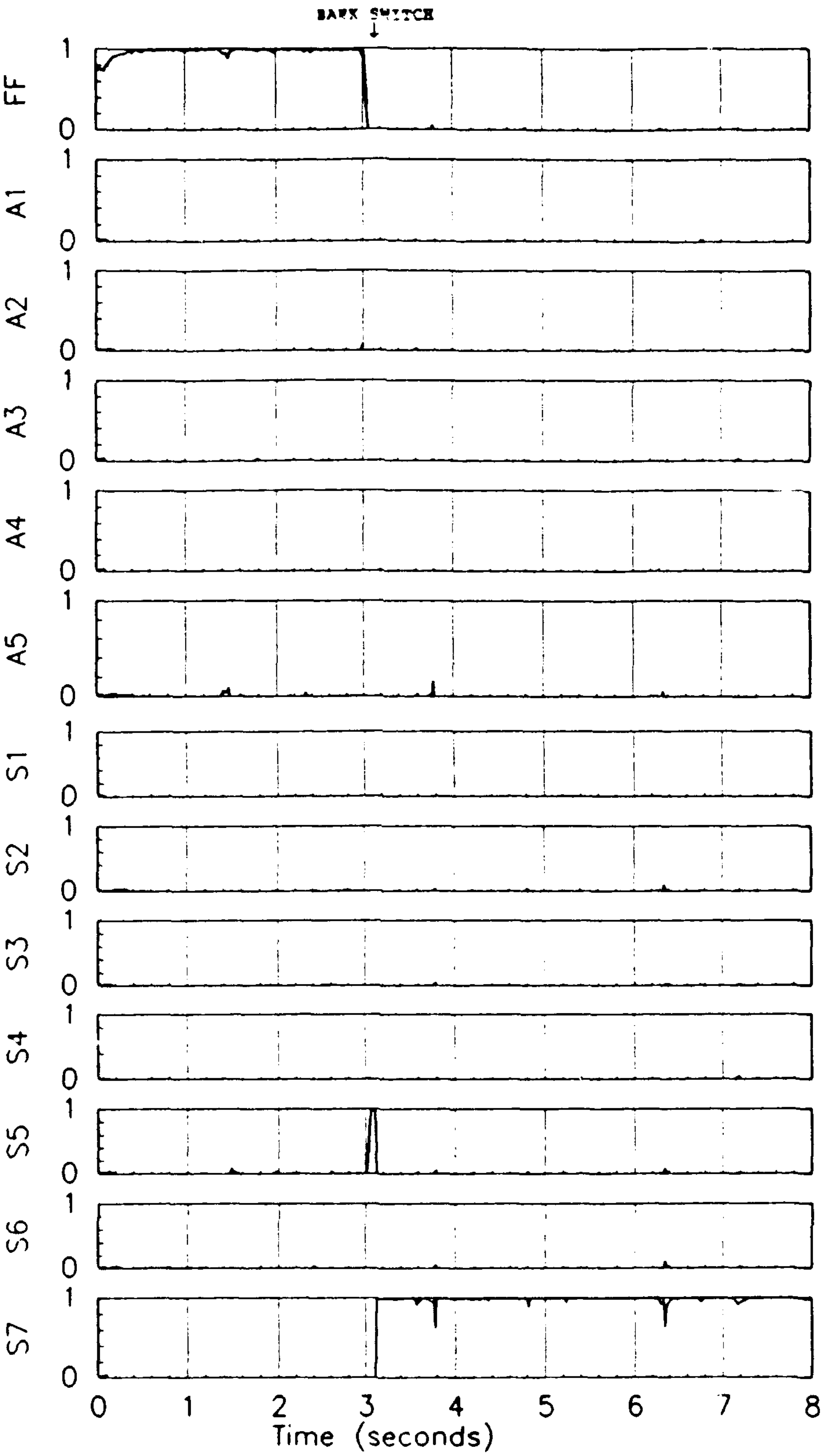

Figure 4.105 Probabilities for a roll rate sensor failure followed by a lateral acceleration sensor failure separated by 0.1 seconds.

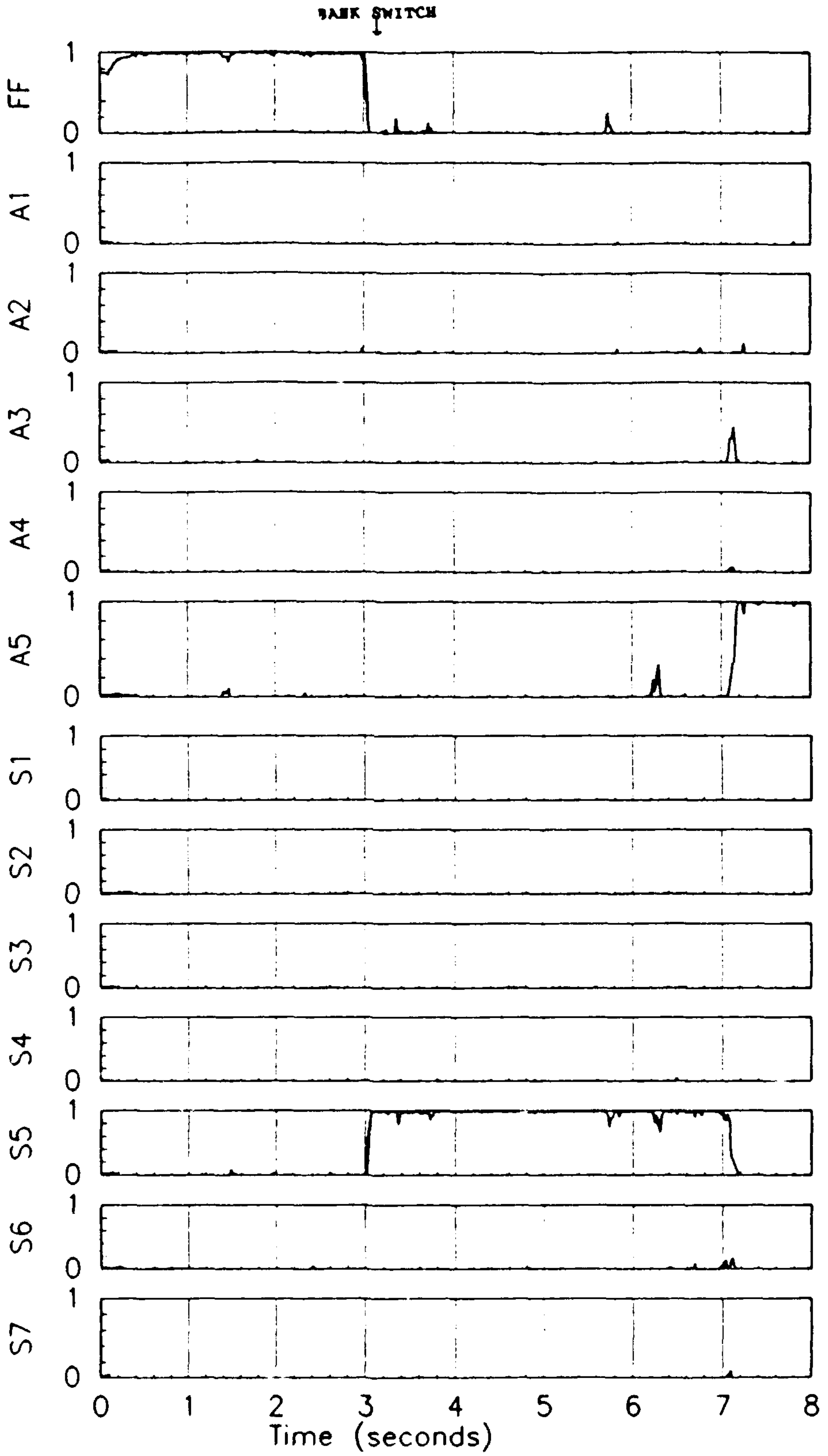

Figure 4.106 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 3.0 seconds.

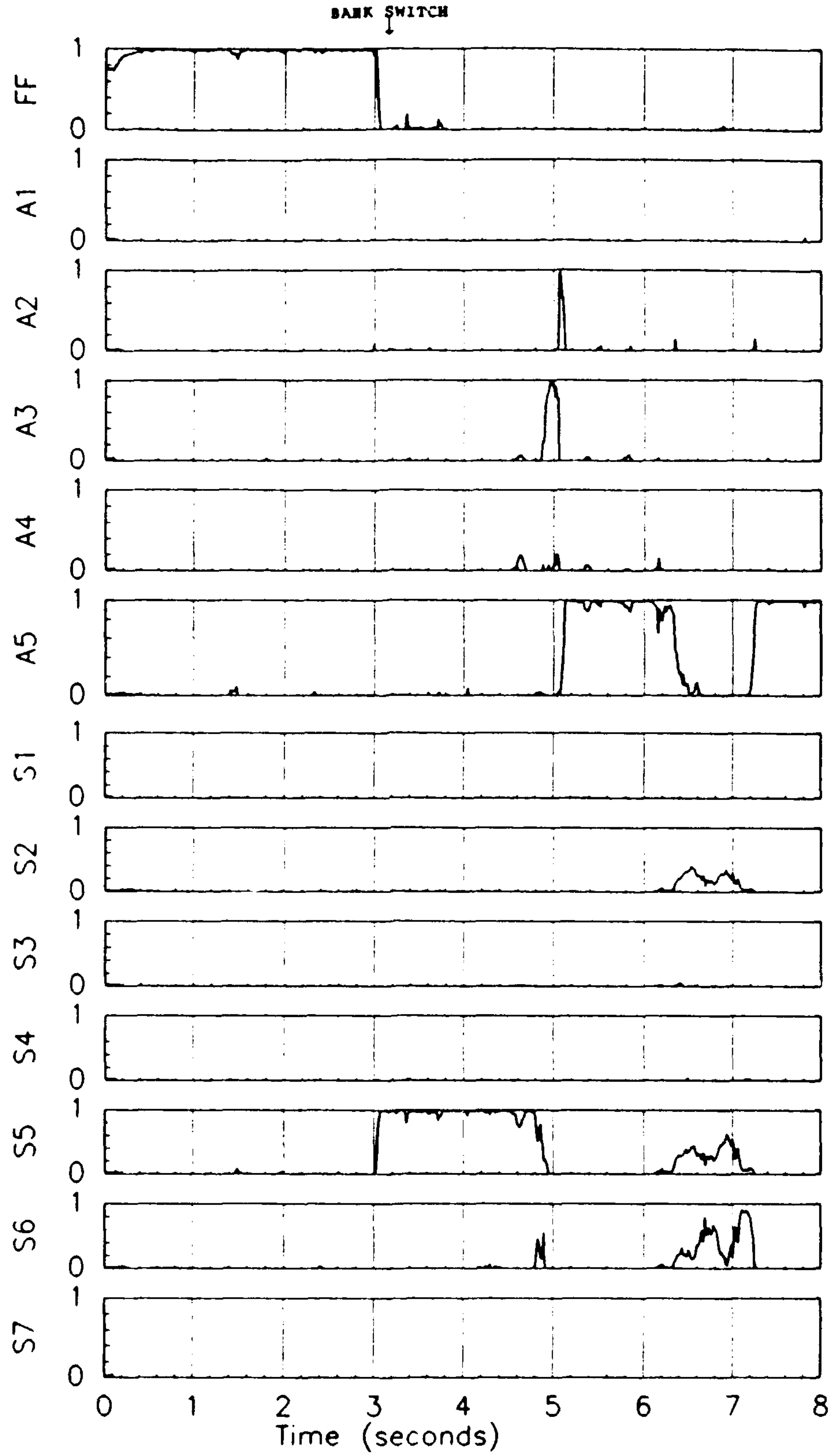

Figure 4.107 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 0.5 seconds.

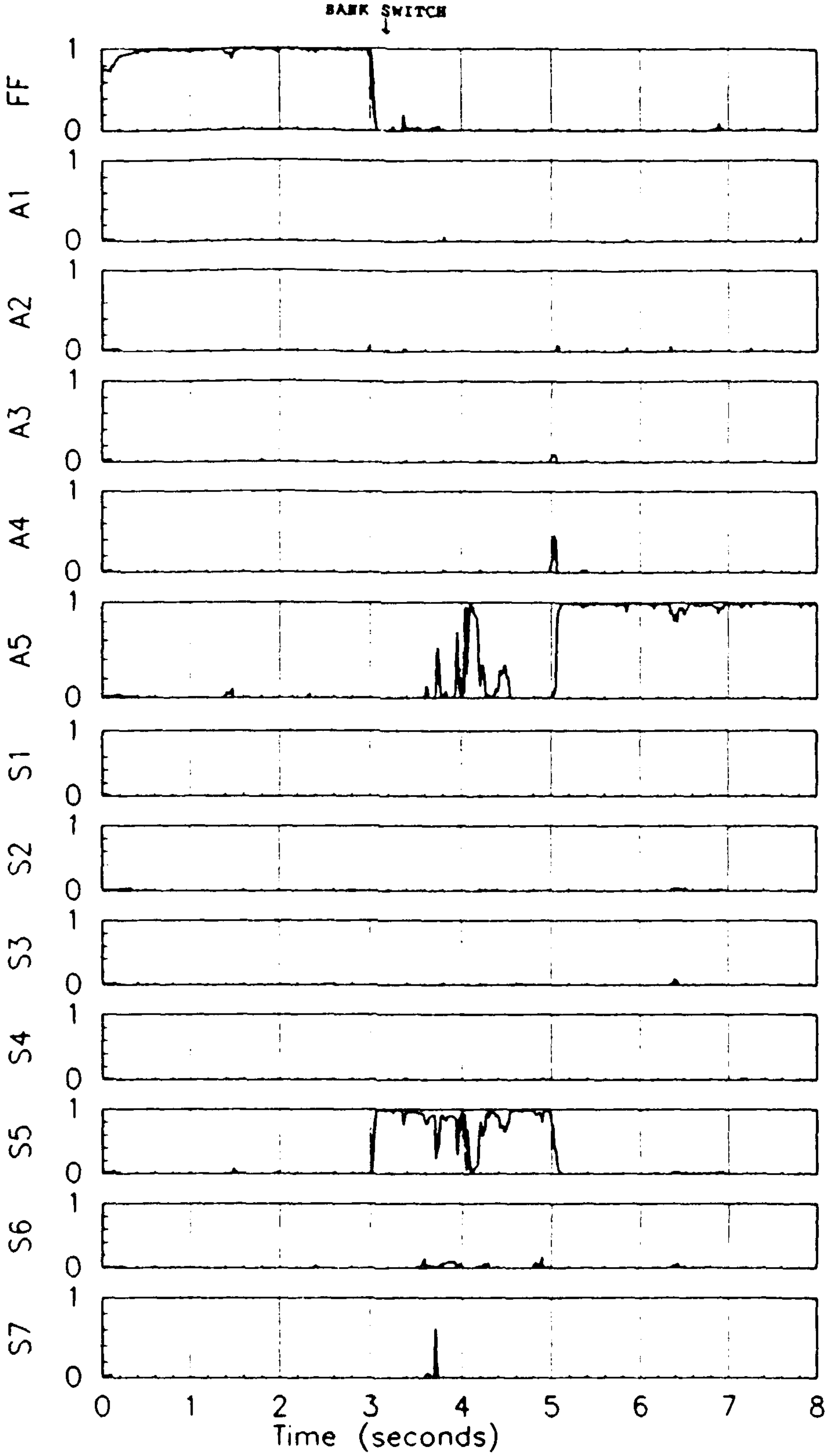

Figure 4.108 Probabilities for a roll rate sensor failure followed by a rudder failure separated by 0.1 seconds.

4.5 Summary

Chapter 4 has provided an overview of the results of this investigation. The intent of this chapter was to present the results of the research effort, demonstrate its usefulness, provide insight into failure detection phenomena, and explain trends in the data. Chapter 5 will present overall conclusions and recommendations for future studies and implementation.

This chapter began with an explanation of the symbology used in the presentation of the results and a general discussion of hard, soft, and multiple failures. Different types of excitation commands were presented, including: purposeful commands, subliminal dithering, and nonsubliminal dithering. A discussion of pulsed dither signals and sinusoidal dither signals was included. A section discussing residual monitoring and its value as an additional voter was provided. Multiple failures were summarized in tabular form. A subset of the total dual failure matrix was presented and discussed and related to residual monitoring.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 General

This chapter begins by addressing each question posed in Chapter 1. Relevant results from Chapter 4 are included as supporting evidence where appropriate. A review of results presented within Chapter 4 is included for completeness. Trends are identified, as are areas of concern requiring additional research. The presentation of any subjective opinions will be labeled as such.

5.2 Response to Chapter 1

The first array of questions presented in Chapter 1 were from Section ***1.6.2.1 Probability Convergence.***

Do the elemental probabilities p_k converge to a solution? Is the solution the correct one?

For dual failures, is the convergence path-dependent (i.e., is the order of failure occurrence important for convergence properties)?

Are the convergence rates quick enough to prevent large transients or loss of control?

Are the convergence rates dependent on the failure type?

All of the data produced in this thesis effort indicates that the elemental probabilities converge to a solution. While some failures have been undetected, due to lack of sufficient residual excitation, only one misidentification was discovered out of nearly 2000 tests conducted. This case was mentioned in Chapter 4: the application of a rudder kick and hold when attempting to identify a rudder failure. In that scenario, the algorithm misidentified the yaw rate filter with a probability of 0.988 (maximum). The result was not surprising since a rudder failure and a yaw rate failure appear similar in the state variables. Since the rudder failed, the lack of a rudder input produced very little yaw rate, resulting in a situation in which the algorithm could not distinguish between a rudder failure of a yaw rate sensor failure. The rudder kick and hold provides a large yaw rate until the application of the rudder failure in which no yaw rate is produced. In this scenario, yaw rate could be generated by applying differential stabilator and flaperon inputs to assist in resolving the ambiguity.

For dual failures, some differences exist between cases in which the orders of the failures are reversed. This is the case for dual failures involving an actuator and a sensor. If the sensor failure occurs first, both failures are identified promptly and correctly. However, if the actuator failure occurs first, a larger excitation signal is required to produce good identification performance. Table 4.1 demonstrates this phenomena. The gray regions, indicating substandard performance, are primarily located within the actuator first failure region. While many of the failure ratings within this region were shown to be "good" by increasing the signal excitation level (strength), an increased signal strength was unnecessary for the sensor-first failure region. Also of interest, the yaw rate and lateral acceleration dual sensor failure demonstrated differences based upon the order of the failures. While it is believed this phenomena is explained by a biased yaw rate residual, a difference does exist.

For the VISTA F-16, no failures resulted in loss of control (LOC). This is due to several factors. It is clear the F-16 was designed to be fail-safe for single failures (probably except for hard-over flight critical surfaces). In this research effort, hard failures were modelled by failing the surface trailing edge to zero. This failure scenario is well within the design limitations imposed by fail-safe failure constraints. For failed sensors, the flight control system used filter estimates in place of the lost sensor inputs. In this implementation, the flight control system never completely lost a sensor input, even if the quality of the input was degraded. By only experiencing a slight degradation in a sensor signal's quality, the flight control system never completely lost a sensor signal and its corresponding feedback path (the loss of a critical feedback path could produce severe consequences, including loss of control). A dual stabilator failure may result in LOC, but not within 2.0 seconds in a benign flight condition. (The second failure was typically introduced at 6.0 seconds, whereas the simulation was not run beyond 8.0 seconds.).

Convergence rates are dependent on the failure type. Actuator failures may be detected immediately, as in the case of purposeful commands, or they may be detected at a much slower rate. Sensor failures almost always converge very quickly (within 0.1 seconds). Some sensor failures are delayed by a second or two, but the convergence rate properties are the same (very quick) once identification is initiated. An exception to this rule exists. Often, the yaw rate filter demonstrates slow convergence properties. A bias exists within this filter, and the convergence is directly related to the dissipation of the bias. A recurring restart on the yaw rate filter would probably eliminate this problem.

From Section *1.6.2.2* ***Residual Monitoring***:

Are there additional voting techniques which may enhance the algorithm's performance?

Can additional voting (beyond the u_{MMAC} probability calculations) improve the performance of the algorithm? Is it useful to monitor the scalar components of the residual vector, rather than the entire residual vector as done by the MMAE/MMAC algorithm itself?

Are there situations in which residual monitoring breaks down?

In what situations does "Beta Dominance" become important and what are the viable solutions to this problem?

Residual monitoring has demonstrated its usefulness in resolving ambiguities. Residual monitoring was extremely effective using a sinusoidal dither signal. For this scenario, the technique clearly demonstrated failure indications in the velocity, angle of attack, normal acceleration, and lateral acceleration single scalar residuals. By developing a "whiteness" check, an algorithm could count the number of zero crossings during a time interval and provide a vote as to whether the hypothesis within a specific elemental filter was "correct" or "incorrect". A "correct" filter's residuals would have many zero crossings and look essentially like "white" noise. An incorrect filter would demonstrate a significant reduction in the residual zero crossings and might nearly match the frequency of the dither signal in at least some of the single scalar residuals for that filter. Residual monitoring has its limitations. It is not as useful for pulse dither signals in this application due to the short duration of the pulses and the limited magnitude of the pulse signal strengths imposed by the subliminal dither signal requirement. Monitoring is useful for continuous sinusoidal signals and for larger amplitude and duration purposeful commands. "Beta Dominance" was demonstrated by Stevens [11,12]. Stevens' results indicated that, by stripping off the leading coefficient in the numerator density function (Eq. (2.12)), improved performance could be obtained. This thesis effort assumed Stevens' results were correct and implemented his recommendation. The results from this thesis provide indirect proof of that assumption.

From Section *1.6.2.3 Hierarchical Modeling*:

Will hierarchical modeling provide good multiple failure performance for the current application?

In the event of simultaneous dual failures, will the algorithm's performance be path-dependent?

The results from Chapter 4 indicate good overall performance for multiple failures. The hierarchical modeling approach is sound and implementable. One improvement to the hierarchical structure should be added. When using a sinusoidal dither, a probability spike can send the algorithm to the incorrect filter bank. Once within the bank, the only route back to the level 0 (single failure) bank is through the no-failure filter. By waiting for the probability to converge to the no-failure filter within the incorrect bank (which may never happen), the overall performance will be degraded. The solution to an incorrect bank switch, usually caused by probability spiking, is to include the most likely single failure filter within the filters of the incorrectly identified bank. This modification will allow an alternate route out of an incorrect bank and enhance performance for spiking probability. As an example, assume a left stabilator failure is modeled. After the insertion of the failure, a probability spike can occur in the right stabilator filter. The right stabilator failure bank (level 1) will be identified and brought "on-line". The probability may well settle into the dual stabilator failure filter because that filter has a hypothesis closer to a left stabilator failure than any other filter within the right stabilator failure bank (incidentally, this is an actual case result). After the probability settles into the dual stabilator filter, add the left stabilator failure (only) filter. If the probability converges to this filter for a period of time, assume you misidentified the first failure and transfer back into the level 0 bank (or transfer to the left stabilator failure level 1 bank directly) . The algorithm will transfer into the left stabilator failure bank (level 1) and performance is significantly enhanced.

For simultaneous failures, there is some evidence to suggest that the algorithm's performance is path-dependent. This is particularly true for actuator and sensor failure combinations. From Table 4.1, the majority of identification difficulties were the result of inducing a hard actuator failure as the first failure. In cases where the sensor failures were induced prior to the actuator failures, the results were very good with very few ratings below "Good". While these performance differences have been attributed to the loss of residual excitation due to the loss of an actuator, this characteristic should be listed as part of the path dependence results. Additionally, for sensor failures, the probability convergence rate of a particular sensor failure will determine which path the algorithm will follow to the double failure filter. If sensor A has a faster convergence rate than sensor B, the algorithm will bring the bank corresponding to sensor A "on-line" and then identify the second failure within that bank.

From Section *1.6.2.5 "Cross Axis Coupling"*:

Are there any misidentifications of failures due to the cross-axis coupling phenomena?

Can additional voters be used to improve the algorithm's performance?

Are filters specifically designed to address a failure that may affect states in both axes ("cross coupling" filters) necessary?

Can the algorithm properly identify multiple failures in different axes?

The results didn't demonstrate any cross-axis coupling issues. One might speculate that, as the number of failure levels increase (as to allow for 3 failures), the level of ambiguity increases, and coupling phenomena may occur. The addition of a nonlinear aerodynamic data base may introduce coupling phenomena. The argument against that premise is that the failure detection is of such short duration (a few seconds at most), any reasonable aerodynamic model could be considered linear over that time period (provided the inputs driving the model are not too large).

The only additional voter used within this study was residual monitoring. A "whiteness" check was introduced to improve monitoring effectiveness but no other new useful voting technique became apparent during the evaluation process. A commonly used (versus new) technique of counting the violations of the 3σ bounds by the individual scalar residual magnitudes was still very useful. The addition of specially designed filters for "cross axes coupling" was not warranted in this effort.

The algorithm has proven itself to be robust, accurate, and reasonably quick. One misidentification has occurred. Some "No Detections" occurred, but every class of "No Detection" or "Poor" detection was addressed and shown to be correctable by using larger excitation signals. The dual stabilator failure will be the most difficult to detect since there are no surfaces available to provide large state excitations in the pitch axis, unless one is allowed to actuate flaperons in tandem rather than only differentially. A number of mixed-axis failures are included in Table 4.1 with overall good results. Table 4.2 is an update of Table 4.1 created to demonstrate the performance impact of dither signals specifically designed to address four failure scenarios rated "ND" or "Poor".

5.3 Conclusions and Trends

In general, the performance of the MMAE algorithm in this application can be characterized as good. This is true for single and multiple hard failures. The results are good for all single hard failures of actuators and sensors using a pulse train dither signal. Also, single hard failure results for purposeful commands are good. Three purposeful commands were evaluated: a roll command, a pitch and roll command, and a rudder kick and hold. One misidentification occurred for the rudder kick and hold command. For a rudder failure, a yaw rate sensor failure was identified as the "correct" failure (see Section 5.1.1). This is the only misidentification discovered within this research effort. The results obtained by using sinusoidal dither signals are also good. The use of a sinusoid dither provides constant failure protection because of the continuous nature of the signal. Residual monitoring, using a sinusoidal dither signal, provides clear indications of failures through the "whiteness", frequency, and magnitude (in relation to the 3σ bounds) of the residual.

Single soft actuator failures can be detected in level flight when the actuator effectiveness is between 50% and 75% of full authority. In many cases the two "bounding" filters (i.e., the fully functional aircraft filter and the corresponding hard failure filter) share the probability, as the theory predicts. The performance can be characterized as fair to good. For the single soft sensor failures (modeled as 1σ or 3.16σ-increases in sensor noise), the results are poor. No scenarios received a good rating for the soft sensor failure scenarios. Many of the "correct" filters exhibited a spiking phenomena during the simulation. This characteristic could be used to identify increased sensor noise failures in some cases. Although the soft sensor failure performance is poor, the algorithm is much more sensitive to sensor biases. Two biases cases were run for the roll and yaw rate sensor filters with good results (Section 4.2.4.2).

Overall, the multiple hard failure scenarios demonstrated good results (Table 4.2). For a dither signal, designed for single failure identification, the order and type of failures affect failure performance. If the first failure is an actuator, the reduction in the excitation signal affects the second failure identification. The solution to improve performance is to increase the dither signal magnitude after the first failure. In general, for simultaneous failures, the performance can be rated as good. Overall, the multiple hard failure performance can be rated as good. Some notable exceptions exist within the multiple failure section. Dual stabilator failures are difficult to detect and isolate properly. The left stabilator failure and pitch rate sensor failure combination has proven difficult to detect and isolate. This may also be true of the right flaperon and roll rate sensor failure combination and the rudder and yaw rate sensor failure combination. Other failures appear to require a stronger amplitude signal to provide good results.

For partial or "soft" actuator failures, the detection threshold is usually between 50 and 75 percent failed on any individual surface. A 50 percent failed left stabilator is detected but not isolated (locked), while a 75 percent failed left stabilator is detected and isolated. This failure test occurred in a benign flight condition (straight and level). An argument could be made that these thresholds would be improved in maneuvering flight, since the aircraft system would experience greater excitation. However, the criticality of these thresholds in straight-and-level flight appears to be a moot issue. In straight-and-level flight. the probability for a loss of control situation is small.

In general, the sensor noise failures, modeled by increasing the sensor noise by 1σ and 3.16σ, did not provide encouraging results. In some cases, the sensor failures are evident by the increased probability spiking in the appropriate filter. However, the isolation of a sensor failure (defined in the soft sensor failure context, as the appropriate sharing of the probability by the two "proper" filters) rarely, if ever, occurs. Often the probability is distributed among many elemental filters besides the two "proper" filters. An argument could be made that modeling a sensor with increased noise may not be appropriate as an important failure mode in most scenarios. In an air data or inertial sensor environment, a sensor bias may be a more realistic manifestation of a soft failure, rather than increased noise. Preliminary investigations of two sensor bias failures were conducted (roll and yaw rate sensor biases). The results indicate the algorithm is much more sensitive to sensor biases than increased sensor noise, at least up to the 3.16σ-increased sensor noise level tested. While the increased sensor noise seems large, the aircraft didn't experience any loss of control or uncommanded transients during the simulation. Again, the flight criticality of this failure would suggest the lack of failure identification or detection is a moot issue. It would be interesting to investigate the level of noise required to disrupt controlled flight and correlate that level with the sensor thresholds for increased sensor noise failure detection and isolation. One might surmise, based upon the available data (the probability spiking at the relatively low levels of increased sensor noise), that the thresholds would occur prior to loss of control. This conjecture remains to be addressed.

Residual monitoring provides additional voting to resolve ambiguities. Residual monitoring is especially helpful when a sinusoidal dither signal is applied. Single scalar residuals have demonstrated clear indications of failures. A correct hypothesis is reflected by the residual falling between the 3σ bounds and appearing "white" (uncorrelated, with a large number of zero crossings, perhaps after any residual bias is estimated and compensated). An incorrect hypothesis within the filter is reflected by a (possibly) biased residual violating the 3σ bounds and possessing a frequency nearly matching that of the sinusoidal dither (also the number of zero crossings is greatly reduced). These indications could be implemented in an algorithm that counts the zero crossings to indicate "whiteness" and thereby the "correctness" of a hypothesis during situations where ambiguities must be resolved.

While many dither wave forms were explored, the sine wave dither form yielded the best performance. The signal is continuous, arguably subliminal, and demonstrates good performance. It also is extremely useful in residual monitoring to resolve ambiguities. The continuous nature of the wave form provides constant failure detection coverage and can easily be modulated in amplitude or frequency to provide the best detection signal for a given failure scenario. A continuously alternating frequency may provide the "optimal" signal for good algorithm performance at all flight conditions. Amplitude modulation accounts for varying dynamic pressures, atmospheric disturbances, or aircraft status (a stabilator failure may exist, which requires a higher amplitude signal on the remaining stabilator to provide sufficient excitation for good second failure performance). In this thesis effort, we did not consider a wideband dither signal, but sinusoidal dither forms did provide better overall performance than pulse train forms. Good results are also obtained during maneuvering flight. Maneuvering flight provides sufficient excitation to ensure good algorithm performance. Large amplitude dynamic maneuvers may require a settling time prior to failure detection and isolation. In general, as soon as a quasi-steady state is achieved, failure detection is swift.

This chapter answered the research questions posed in Chapter 1. While the answers are important, they are not nearly as important as the trends demonstrated by the data of Chapter 4. Since the writing of Chapter 1 at the beginning of the research effort, additional questions and concerns have arisen. The conclusions and trends section of this chapter has attempted to summarize the data and provide additional insight to the reader. Some conjecture is included throughout this chapter (often with supporting evidence found in Chapter 4). While Chapter 4 provides clues, there is no substitute for experience with the algorithm in this application. The results are promising and offer a possible window into the future of fault detection and isolation within flight control systems.

Some additional questions remain to be addressed in future studies.

Are there any other additional voting techniques which may be useful in this application?

Will restarting the filters at regular intervals improve performance?

For increased sensor noise, do the thresholds for detection and isolation occur prior to loss of control?

Does the algorithm provide good performance for sensor biases?

Does the incorporation of a nonlinear aircraft model introduce coupling issues?

Will a sinusoidal dither signal, modulated in frequency and amplitude, produce good algorithm performance throughout the flight control regime, in atmospheric disturbances, and after a first failure has occurred?

VI. REFERENCES

1. Magill, D.T., "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Trans AC*, Vol. AC-10, No. 5, pp. 434-439, Oct. 1965.

2. Athans, M., and C.B. Chang, "Adaptive Estimation and Parameter Identification Using Multiple Model Estimation Algorithm", Technical Note 1976-28, ESD-TR-76-184, Lincoln Laboratory, Lexington, Mass., June 1976.

3. Athans, M., et. al., "Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part 1: Equilibrium Flight," *IEEE Trans. AC*, Vol AC-22, No. 5, pp. 768-780, Oct 1977.

4. D'Azzo, J.J., and C.H. Houpis, "Linear Control System Analysis and Design (2nd Edition)," McGraw Hill, New York, 1981.

5. Maybeck, P.S., Stochastic Models, Estimation and Control, Vol. 1., Academic Press, New York, 1979.

6. Maybeck, P.S., Stochastic Models, Estimation and Control, Vol. 2., Academic Press, New York, 1982.

7. Maybeck, P.S., Stochastic Models, Estimation and Control, Vol. 3., Academic Press, New York, 1982.

8. Greene, C.S., and A.S. Willsky, "An Analysis of the Multiple Model Adaptive Control Algorithm," *Proc IEEE Conf. Dec. and Cont.*, Albuquerque, New Mexico, pp. 1142-1145, Dec. 1980.

9. Pogoda, D.L., "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," M.S. thesis, A.F. Inst. of Tech., Wright-Patterson AFB, Ohio, Dec. 1988.

10. Maybeck, P.S., and D.L. Pogoda, "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," *Proc. IEEE Conf. Dec. and Cont.*, Tampa, Florida, pp. 1566-1572, Dec 1989.

11. Stevens, R.D., "Characterization of a Reconfigurable Multiple Model Adaptive Controller Using a STOL F-15 Model," M.S.E.E. thesis, A.F. Inst. of Tech., Wright-Patterson AFB, Ohio, Dec. 1989.

12. Maybeck, P.S., and R.D. Stevens, "Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-27, No. 3., pp. 470-480, May 1991.

13. Stratton, G.L., "Actuator and Sensor Failure Detection Using a Multiple Model Adaptive Technique for the VISTA/F-16", M.S.E.E. thesis, A.F. Inst. of Tech., Wright-Patterson AFB, Ohio, Dec. 1991.

14. Military Specification - Flying Qualities of Piloted Airplanes (MIL-STD-1797A), Government Printing Office, Washington D.C., 30 January 1990.

15. Integrated Systems Incorporated, "MATRIXx Users Manual, Version 7.0", Palo Alto, CA, 1986.

16. Nesline, F.W., and D. Zarchan, "A Classical Look at Modern Control for Missle Autopilot Design", Proceedings from the 1982 Guidance and Control Conference", AIAA paper no. 82-1512 (1982).

17. Maybeck, P.S., and K. Hentz (1987), "Investigation of Moving-Bank Multiple Model Adaptive Algorithms", *AIAA Journal of Guidance, Control, and Dynamics*, 10,1 (Jan-Feb 1987), 90-96.

18. Martin, R.M., "LQG Synthesis of Elemental Controllers for AFTI/F-16 Adaptive Flight Control, M.S.E.E. thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec, 1990.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE 1 Jun 92

3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Multiple Model Adaptive Estimation Applied to
the VISTA F-16 with sensor and actuator failures

5. FUNDING NUMBERS

6. AUTHOR(S)
Timothy E. Menke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Wright-Patterson AFB, Ohio

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GA/ENG/92J-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
WL/FIGL
Wright-Patterson AFB, Ohio

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited distribution

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A Multiple Model Adaptive Estimation (MMAE) algorithm is applied to the Variable Stability In-flight Simulator Test Aircraft (VISTA) F-16 at a low dynamic pressure flight condition (0.4 Mach at 20000 ft). A complete F-16 flight control system is modeled containing the longitudinal and lateral-directional axes. Single and dual actuator and sensor failures are simulated including: complete actuator failures, partial actuator failures, complete sensor failures, increased sensor noise, sensor biases, dual complete actuator failures, dual complete sensor failures, and combinations of actuator and sensor failures. Failure scenarios are examined in both maneuvering and straight and level flight conditions. Single scalar residual monitoring techniques are evaluated with suggestions for improved performance. A hierarchical "moving bank" structure is utilized for multiple failure scenarios. Simultaneous dual failures are included within the study. White Gaussian noise is included to simulate the effects of atmospheric disturbances, and white Gaussian noise is added to the measurements to simulate the effects of sensor noise.

14. SUBJECT TERMS
Kalman Filtering, Multiple Model Adaptive Estimation,
MMAE, Failure detection and isolation, Bayesian,
MMAC, Multiple Model Adaptive Control

15. NUMBER OF PAGES 219

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED

20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102