12-1992

Failure Identification using Multiple Model Adaptive Estimation for the LAMBDA Flight Vehicle

Peter D. Hanlon

AFIT/GE/ENG/92D-19

FAILURE IDENTIFICATION USING MULTIPLE MODEL ADAPTIVE ESTIMATION FOR THE LAMBDA FLIGHT VEHICLE

THESIS

Peter D. Hanlon
Captain, USAF

AFIT/GE/ENG/92D-19

Approved for public release; distribution unlimited

AFIT/GE/ENG/92D-19

FAILURE IDENTIFICATION

USING

MULTIPLE MODEL ADAPTIVE ESTIMATION

FOR THE LAMBDA FLIGHT VEHICLE

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In partial fulfillment of the

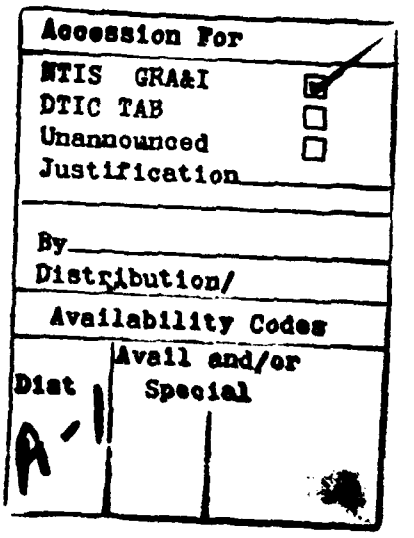
requirements for the Degree of

Master of Science in Electrical Engineering

Peter D. Hanlon, B.S.E.

Captain, USAF

December 1992



PREFACE

This study has been a fascinating, challenging, at times exhausting, and always an exhilarating experience. Fascinating because it opened up an area of engineering that I formerly viewed as "black magic." Challenging because my weakest areas, prior to this assignment, were control theory and statistics, both of which were foundational for this research. Exhausting because of the many late hours that were required to keep up with the "fire hose." However, it was always exhilarating to learn, understand, and even apply concepts that I formerly viewed as well beyond my capacity to comprehend.

There are of course many, many people to whom I am indebted for support and encouragement throughout this research. First and foremost I want to thank my wife, Kathy, for putting up with my highly volatile and unpredictable schedule, for listening to my engineering banter and even pretending to be interested, and for reorganizing my priorities when they really needed it. Next are my girls, Amy, Rachel, and Becka, who put up with a daddy that would stare at the computer screen and say "uh huh" or "go watch TV" whenever they asked any questions, and who would occasionally get me to read something other than a textbook (although textbooks work great as nighttime stories, puts them right out). I want to thank my thesis advisor Dr. Peter Maybeck for the prayers (Lord knows I needed them), confidence in me, guidance, understanding, slaps on the back, kicks in the pants, etc., etc. ... well you get the idea. Mostly, I want to thank him for his enthusiasm for teaching, without his animated lectures I would never have understood the foundational material for this research, and his violation of the Principle of Conservation of Misery (if I had to go through this misery, then, by God, you will too) that seems all too prevalent in engineering education.

Finally, I want to thank the my fellow students that suffered through the controls sequence with me. The cohesiveness and mutual assistance that we built during our 18 month internment changed this experience from something to be tolerated, into something that we can look back on (thank goodness) and actually have fond memories.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Figure Page

ABSTRACT

This study develops and investigates the performance of a Multiple Model Adaptive Estimator (MMAE) to detect and identify control surface and sensor failures on the LAMBDA flight vehicle (a URV developed by Wright Laboratories). The MMAE uses a bank of Kalman filters that predict the aircraft response to a given input, with each filter model based on a different failure hypothesis, and then forms the residual difference between the prediction and sensor measurements for each filter. The MMAE uses these residuals to determine the probabilities of the failures that are modeled by the Kalman filters. Initially the MMAE identified all these failures within 4 seconds of onset. Various performance improvement techniques were researched and the identification time was reduced to less than 2 seconds after failure onset. This improvement was mostly due to an increase in the penalty for measurement differences, and through retuning of the Kalman filters. The MMAE performance was tested at the boundaries of the LAMBDA flight envelope, with good performance found at points close to the design flight condition. The performance at points that were far from the design flight condition indicates that gain scheduling is required to provide adequate performance across the entire envelope.

FAILURE IDENTIFICATION USING MULTIPLE MODEL ADAPTIVE ESTIMATION FOR THE LAMBDA FLIGHT VEHICLE

Chapter I: Introduction

1.1 Chapter Overview

In this chapter we introduce the development of a Multiple Model Adaptive Estimator (MMAE) for the LAMBDA flight vehicle. In Section 1.2 we present the motivation for developing the MMAE, and show the relationship of this research with contemporary research efforts in this area. In Sections 1.3 through 1.7, we define the details of the research problem by first defining the problem statement, then describing the assumptions made to accomplish this research, followed by the specific research questions that will be studied, and then a description of the scope and limits of this research. Finally, in Section 1.8 we present an overview of the structure of this thesis.

1.2 Problem Perspective

Throughout the history of aircraft development, there has been a continuous push for higher performance in all measures of aircraft capabilities (speed, altitude, maneuverability, acceleration etc.). To obtain dramatic increases in performance, designers have had to create

increasingly complex flight control systems. For example, the Wright brother's major contribution to flight was the design of an aircraft that was just slightly unstable but controllable by a human pilot. Since the instability took several seconds to grow, the pilot could easily countermand this growing instability, thus producing stable flight [2:1-2]. Prior attempts at powered flight were made with aircraft that were inherently stable, which proved to be unresponsive and impossible for a human pilot to fly. Subsequent years saw remarkable improvements in the flight control area, primarily in structural and mechanical fields, as the fundamentals of flight were investigated and exploited.

Eventually, improvements in aircraft performance were limited by the capability of a human pilot to control the instabilities that were necessary to produce the desired improvement in performance. At first, pilots were screened and improved through flight training, and the best pilots were assigned the highest performance aircraft, usually fighter or test aircraft. The mystique of these elite pilots is mostly due to their capability to adjust their reactions appropriately to the aircraft and flight conditions that they encountered. In particular, test pilots were legendary for their ability to control high performance aircraft despite difficult and dangerous handling qualities or various faults and failures that they encountered during flight testing. The demand for even higher performance aircraft, such as the F-16, that have instabilities well beyond the capabilities of a human pilot to control without a control augmentation system, has produced a demand for even more capable flight control systems.

Fortunately, the explosive improvement in computer technology has enabled the development of these sophisticated flight control systems. The computer capabilities of the current flight control systems are the dreams of the flight control designers just a decade ago. In particular, the development of the microprocessor has brought the concept of parallel

processing to fruition. This architecture allows each avionics system, and even subsystem, to accomplish its own processing without relying on a central computer, which is the primary performance limitation in non-parallel avionics suites. While these improvements in the hardware of flight control systems have enabled a tremendous increase in flight control capabilities, further research and development of the software algorithms that employ parallel architecture is required to realize the full potential of parallel processing.

The Multiple Model Adaptive Estimation (MMAE) algorithm [1, 21, 22, 24, 47] naturally exploits the capabilities of parallel processing. Its structure is shown in Figure 1. It is composed of a bank of Kalman filters, each imbued with its own model, and is by nature a parallel processing algorithm. Each elemental Kalman filter uses its own model, along with a given input (u), to develop an estimation of the current aircraft states ($\hat{x}_i$), independent of the other filters. The filter then uses this estimate, along with the current measurement of those states (z), to form the residual (r_i), which is the difference between the measurement and the filter's prediction of the measurements before they arrive. The residuals from the filters are used by the hypothesis testing algorithm as a relative indication of how close each of the filter models are to the true model. The smaller the residual, the closer the filter model matches the true model. The hypothesis testing algorithm first scales the residuals to account for various uncertainties and noises in the measurements, and then computes the conditional probability for each of the hypotheses modeled in the bank of Kalman filters (p_i). These probabilities are then used to weight the individual Kalman filter state estimates to produce a blended estimate of the true states ($\hat{x}_{MMAE}$), which can then be used as the optimal estimate of the states by a control system. When used for failure identification, each of the Kalman filters would model a different failure condition and the residuals from each filter would indicate how

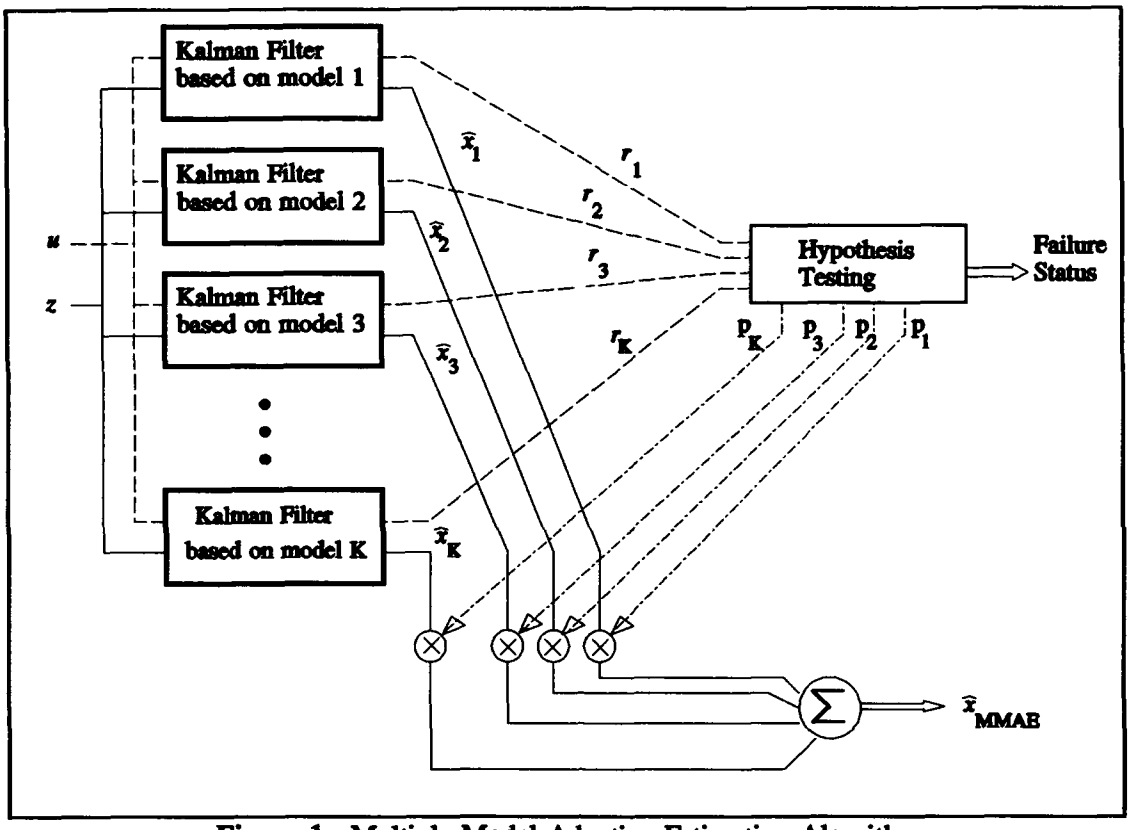


Figure 1. Multiple Model Adaptive Estimation Algorithm

close that filter's model is to the actual failure condition. By monitoring these residuals, the hypothesis tester can determine the current failure status of the aircraft.

The capabilities of the MMAE have been sporadically investigated since the early seventies [1, 15, 22, 23, 24, 36, 37], with a significant increase in research since the development of microprocessors and distributed computation [6, 7, 10, 21, 30, 31, 32, 33, 37, 39, 40, 41, 42, 44, 45, 46].

One particular capability that has produced significant research effort is the ability of the MMAE to identify various flight control failures [28, 29, 33, 34, 35, 38, 41, 42]. Much of this effort has been through computer modeling and simulation, since the adaptation and

complexity of the MMAE does not lend itself to simple, straightforward analysis. Also, while various implementations have been studied, none have been implemented and tested in actual aircraft systems.

This research advances the investigation of Multiple Model Adaptive Estimation by developing a MMAE that operates onboard the LAMBDA flight test vehicle as a flight control failure detector, in preparation for flight testing. The development and flight testing of a MMAE for the LAMBDA will provide realistic and valuable information on the capabilities of the MMAE to detect failures in flight control system actuators and/or sensors, and provide the groundwork for future investigations that broaden the applications of the MMAE.

1.3 Problem Statement

It is intended to develop a MMAE algorithm to detect a single failure of a flight surface or flight sensor, for flight testing on the LAMBDA flight vehicle. The LAMBDA is an unmanned research vehicle (URV) developed by the Flight Control Division of the Flight Dynamics Directorate, Wright Laboratory, as an affordable, flexible research vehicle for testing and demonstrating flight control concepts, devices, and systems [43]. Further, it is desired to investigate the MMAE's ability, through computer simulation, to identify failures in both lateral and longitudinal-directional axes at various flight conditions.

The Flight Dynamics Laboratory plans to flight test the MMAE design on the LAMBDA flight vehicle, investigating its ability to detect an aileron actuator failure at a flight condition that is normally encountered in the average flight profile for the LAMBDA. The flight condition of the LAMBDA is determined by five parameters; the aircraft weight, forward velocity, dynamic pressure, center of gravity, and trim angle. The average flight condition parameters are a weight of 200 lbs, a speed of 160 ft/sec, a dynamic pressure of

30.43 lb/ft^2, a center of gravity located at 46.8 inches from the aircraft nose, and a trim angle of zero degrees. The LAMBDA usually flies at low altitude; therefore the dynamic pressure ($\frac{1}{2}\rho v^2$) is directly related to forward velocity (v) since the air density (ρ) is essentially constant. The trim angle is usually very small for a well trimmed aircraft; therefore we assume a trim angle of zero. Since the altitude and trim angle are considered constant, we end up with a three dimensional flight envelope for the LAMBDA as shown in Figure 2, which also shows the design flight condition. This flight condition is considered average because the LAMBDA is usually flown close to maximum velocity, at low altitude, with the variation in weight primarily dependent on the amount of fuel consumed. Therefore the design flight condition is a median value of the normal operating points.

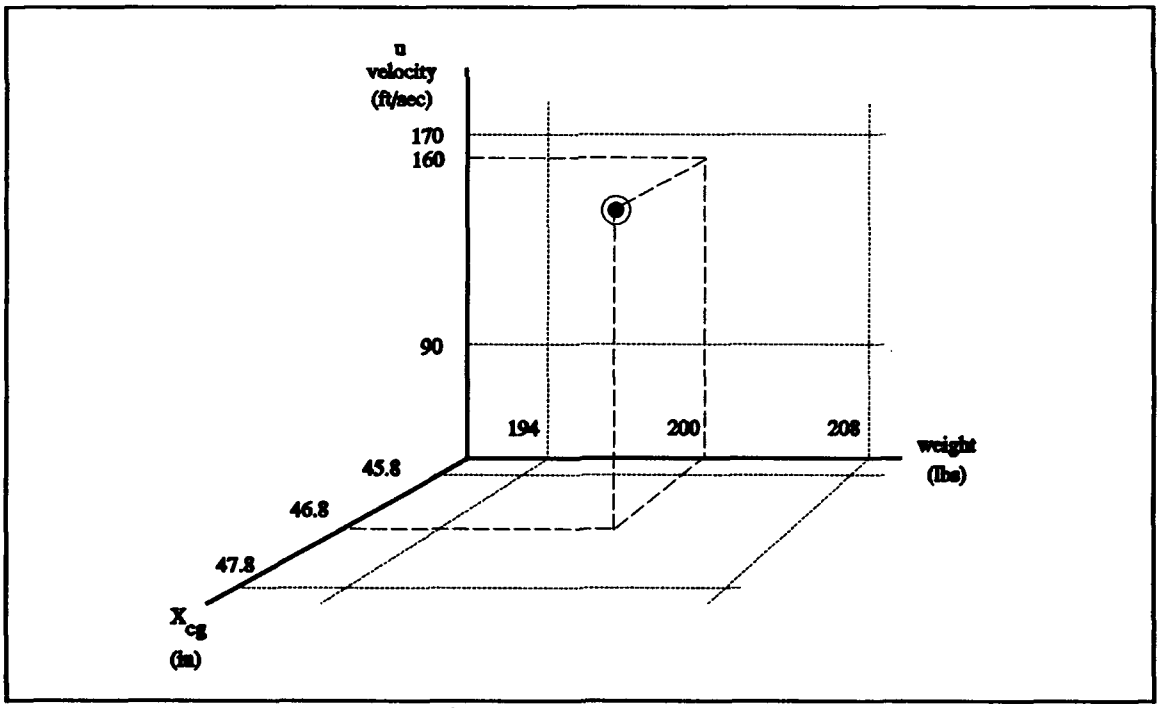


Figure 2. LAMBDA Flight Envelope

1.4 General Assumptions

The assumptions used for this research are based on those of previous investigations into the use of Multiple Model Adaptive Control (MMAC) to provide flight control during various failure conditions in the STOL F-15 [28, 38, 29, 41] and MMAE to detect and identify various failure conditions in the VISTA F-16 [33, 34, 35, 42]. The primary task for this thesis is failure detection and isolation for the LAMBDA flight vehicle. We assume that once the failure is properly identified, a flight controller with sufficient robustness will continue to provide stable flight control. Therefore, we will design a MMAE to accomplish the failure detection and identification task, rather than a MMAC with its associated elemental controller designs.

We further assume that the linearized model for the LAMBDA is sufficient for the development of the MMAE algorithm. Embedded within this linearized model are several other assumptions that were used to develop the model [43], namely lateral and longitudinal axis decoupling and first order actuator dynamics. The linearized model was developed by estimating flight coefficients from geometric data and then adjusting these estimates using good engineering judgment and flight test data. For a fully functional aircraft, the cross axis coupling terms are so small that they produce a negligible aircraft response. Currently, all test flights have been conducted using a fully functional aircraft, and therefore, the flight test data does not contain enough information to estimate the cross axis terms. The assumption of first order actuator dynamics compares quite well with the flight test data.

To model single actuator failures, we have assumed that a single flight control surface failure will produce half of the expected response from dual control surfaces. For instance, a certain elevator input might produce a 10 degree pitch-up response. We have assumed that,

for this same input, a failure of only the right elevator actuator would produce half of this response (5 degree pitch-up) since only the left elevator actuator would be functional, thereby decreasing the actuator surface area by one half.

As mentioned above, we have assumed that the cross-axis coupling terms are negligible even for single surface failures, primarily because there is insufficient data to estimate these terms [43]. Using our example, a right elevator failure would produce a small yawing and rolling response, which we assume to be negligible. An exception to this assumption is the cross-coupling between the ailerons and the yaw axis because the long yaw moment arm of the ailerons could easily produce a significant yaw. Fortunately, there is sufficient data to estimate the aileron's effect on the yaw rate (r) and the rudder's effect on the roll rate (p). At the design point, the contribution of the aileron to a yaw rate is 10.5% and the contribution of the rudder to a roll rate is only 0.3%. This supports our assumption that the cross-axis terms are small. However, Swift found noticeable roll/pitch coupling dynamics for the LAMBDA, but was unable to estimate the cross coupling terms [43:3.50]. A flight test with single actuator failures should provide the much needed data for estimating these cross-axis coupling terms, and the aircraft model can then be corrected for further tests.

1.5 Specific Research Questions

In this section we present the specific questions that will be used to guide this research. These research questions were grouped into three areas, decision convergence, decision switching, and unmodeled effects. This same structure will be used in Chapter IV, Results, and Chapter V, Conclusions and Recommendations. For each of these areas, we present the specific research questions (in *italics*), followed by our rationale for investigating the answers to these questions.

1.5.1 Decision Convergence

Does the MMAE probability-based failure detection algorithm converge to a specific failure hypothesis?

Does the MMAE probability-based failure detection algorithm converge to the correct failure hypothesis?

The answers to these questions will characterize the failure detection performance for the MMAE. We are attempting to develop an algorithm that not only identifies the failure status of the LAMBDA flight vehicle, but does not exhibit fluctuations in its identification. For this implementation to be useful, it must converge to the correct failure status, and maintain that correct identification.

1.5.2 Decision Switching

Are the decision convergence rates fast enough to prevent lengthy (or even momentary) incorrect failure hypotheses declarations by MMAE failure detection algorithm?

Are the decision convergence rates dependent on the flight control failure?

Can the MMAE correctly identify the failure status despite an initial or momentary incorrect identification of the failure status?

We need to develop an algorithm that not only identifies the correct failure status, but also identifies it quickly in order to minimize the time lag until the appropriate flight control is applied.

We also will lay the groundwork for future investigations into multiple failure detection and identification for the LAMBDA. If the MMAE can correctly identify

the failure status, independent of the initial status, then it has the potential of switching between hierarchical banks of parallel Kalman filters to provide identification of multiple failures. This potential was investigated in previous research [29, 33, 34, 35, 41, 42], but we must verify this property for this specific implementation of the MMAE.

1.5.3 Unmodeled Effects

Are there any misidentifications of failure status due to unmodeled phenomena? Will constant coefficients in the parallel Kalman filter models provide sufficient performance for failure identification, or will gain scheduling of these coefficients to the current flight condition (speed, altitude, vehicle weight, etc.) be required?

Previous research [40] has indicated that the MMAE is capable of operating over a large variation of model parameters, such as a flight envelope, by using a moving parameter window approach shown in Figure 3. With this approach a few Kalman filter models are used to bracket the system's operating point in the parameter space. These Kalman filter models are then adjusted as the system's operating point moves in the parameter space, such as a change in the aircraft's flight condition. This allows the MMAE to detect failures over a much larger portion of the flight envelope. We are designing our MMAE at one specific design point (the median flight condition), but we need to find the boundaries over the flight envelope where the MMAE performance deteriorates to the point where a shift in the design point is required to maintain the desired failure identification performance.

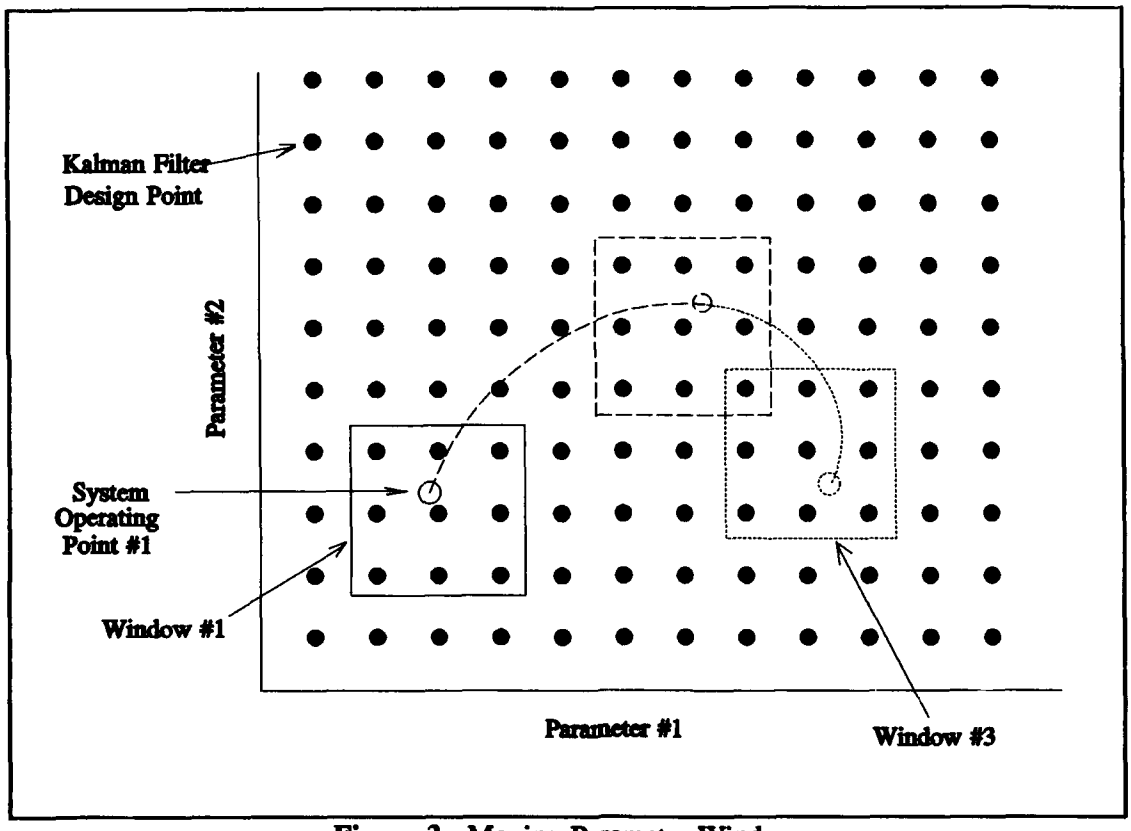


Figure 3. Moving Parameter Window.

1.6 Scope of Research

This research is primarily directed at implementation and testing of the MMAE algorithm for the LAMBDA flight vehicle. Our goal is to develop an algorithm that can identify various single failure modes, i.e, failures of surface actuators or sensors of the LAMBDA flight vehicle.

There are certain aspects of MMAE that we are not investigating. We are not implementing a MMAE that will identify multiple failures. Therefore, we will not be developing a hierarchy of Kalman filter designs to identify these multiple failures. However, we will investigate the MMAE performance under various initial conditions to verify that it

correctly identifies the failure, independent of the initial conditions, thus laying a foundation for future research of multiple failure detection for the LAMBDA flight vehicle.

1.7 Research Limitations

The most significant limitation of this research is the adequacy of the linearized model for the LAMBDA. This model was developed using geometric measurements, mass estimates, and some adjustments to the parameters to match previously recorded flight data [43]. This development assumed that the axes were completely decoupled, that the actuator dynamics were first order, and that control surfaces worked in tandem to produce the various observed responses. We are interested in detecting and identifying a single actuator failure, so we must modify the model to predict these single actuator responses to the given inputs. We will use this linearized model to build our various Kalman filter models by representing actuator failures as a zeroing out of the appropriate column of the input matrix (B_d), and sensor failures are represented by zeroing out the appropriate row of the measurement matrix (H). Other failure modes are not modeled in this thesis.

To detect a single actuator failure, we require a model that accounts for the effect of an input to each of the individual surfaces. For example, our model needs to account for an input to the right aileron, independent of what the left aileron is doing, and vice versa. To develop this split actuator model, we will assume that the response from a single actuator is one half of the expected response of the case of both actuators being used. Note that a single failed actuator will cross couple its response into the other axes, but we have not included these effects in the linearized aircraft model because we don't have enough data available to estimate these cross coupling terms. This area definitely needs to be researched, and as more

data becomes available on the effects of these cross coupling terms, corrections to the model should be incorporated.

1.8 Thesis Overview

In the subsequent chapters we develop this research in more detail. Chapter 2 gives a brief background on the development of Multiple Model Adaptive Estimation and/or Control (MMAE/MMAC) and discusses some of the properties found in previous research. Chapter 3 presents our research methodology by discussing the development of the MMAE algorithm for this implementation, the software tools used to test this implementation, and a description of the various performance measures used to characterize the capabilities of the MMAE. Chapter 4 presents the findings from our attempt to enhance the MMAE's performance through various techniques, and to answer the specific research questions posed in Section 1.5. Chapter 5 summarizes these findings and concludes with our recommendations for further study.

Chapter II: Background

2.1 Chapter Overview

In this Chapter we present the background of multiple model adaptation. We start with a description of the development of multiple model adaptation estimation, followed by the development of multiple model adaptive control. We close with a brief catalog of applications of multiple model adaptive estimation.

2.2 Multiple Model Adaptation Development

2.2.1 Multiple Model Adaptive Estimation Development

The use of multiple filters in a parallel structure to generate adaptive estimation algorithms was first developed by D. T. Magill [23]. He arranged a number of Kalman filters, each with different time invariant plant models, and used the residuals from these filters to form an appropriately weighted sum of the Kalman filter estimates, as shown in Figure 4. He showed that this adaptive estimation algorithm produced the optimal estimate in the minimum mean square error sense for a Gauss-Markov process. This work was extended to handle discrete systems by C. B. Chang and M. Athans [3, 4].

The properties of this multiple model adaptive estimation concept were further developed by Lainiotis and others. They investigated the properties of this concept when using a discrete number of models to represent a continuous domain of plant models [21, 22]. With others, he showed that good performance was attained with tightly tuned Kalman filters, but these filters produced erratic behavior if the filter with the correct model was not in the filter bank [6, 13].

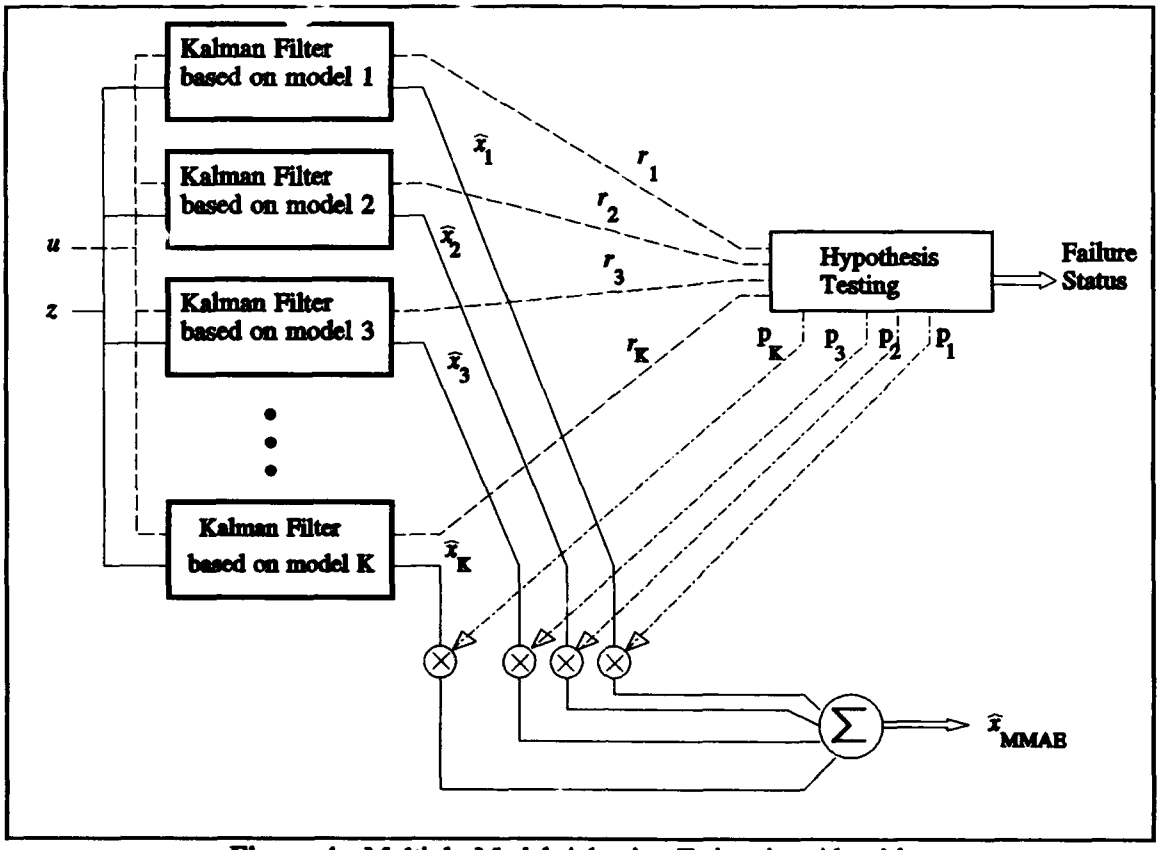


Figure 4. Multiple Model Adaptive Estimation Algorithm

2.2.2 Multiple Model Adaptive Control Development

M. Athans and others extended this adaptive estimation concept to adaptive control developed for the flight control system of NASA's F-8C flight test aircraft [1]. They weighted the optimal control signals generated by a bank of Linear system-Quadratic cost function-Gaussian noise distribution (LQG) controllers, each with an embedded Kalman filter using a different plant model, and then summed these weighted control signals together, as shown in Figure 5. They found that this algorithm provided good control at flight conditions that were close to the design conditions of the Kalman filter models, as long as the system was

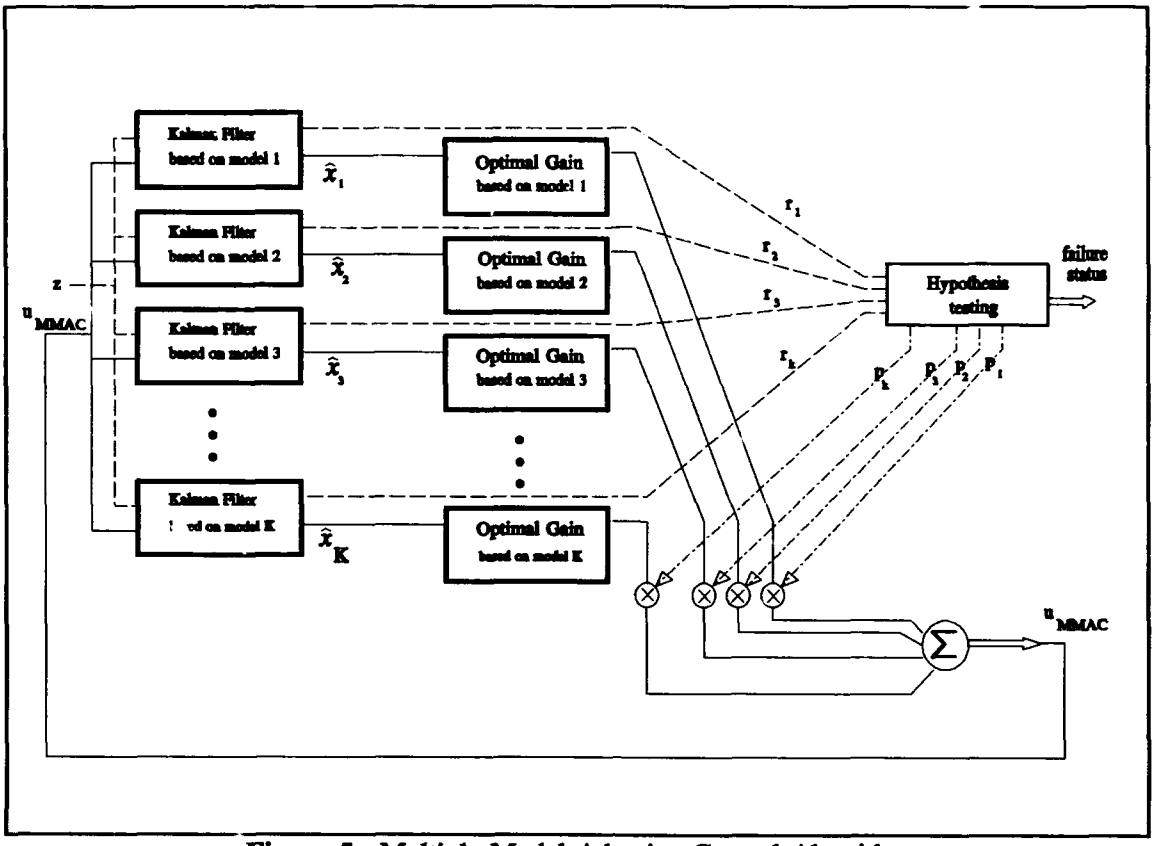


Figure 5. Multiple Model Adaptive Control Algorithm

appropriately excited, using a test signal or dither, to attain good failure identification. They also found that this algorithm was sensitive to high frequency noise, specifically strong wind gusts, and required low pass filtering to attain good performance.

Greene and Willsky [11] examined and defined stability regions where the multiple model adaptive control (MMAC) algorithm yields non-oscillatory responses. They found that the stability of the MMAC was determined by the relation between the growth rate of the most unstable mode of a Kalman filter with a mismatched model, when compared to the truth model, and the rate of decay of the slowest stable mode of a Kalman filter with a matching model. They presented a method of computing the borders where the algorithm is neutrally

stable, which then defines the "domain of attraction," or the region where the MMAC's response does not oscillate.

2.3 Multiple Model Adaptation Applications

2.3.1 Failure Detection

A. S. Willsky [47] surveyed a number of failure detection methods, using a performance index as an empirical measurement of the capabilities of the various methods. This performance index included the types of failure modes that can be detected, implementation complexity, various performance measures (false alarms, detection delays, repeatability), and robustness in the presence of modeling errors. He found that the multiple model adaptive controller will yield the best performance over the widest class of failures.

Longitudinal control of the Short Take-Off and Landing (STOL) F-15 using Multiple Model Adaptive Control (MMAC) was investigated by Pogoda [28, 38] and Stevens [29, 41]. Pogoda developed an algorithm for the landing phase of the flight regime, which would reconfigure the flight control in the presence of a single failed control surface or sensor. He designed Kalman filter models for a fully functional aircraft, a failed stabilator, a failed "pseudo-surface," and a failed pitch rate sensor. The "pseudo-surface" was a combination of the canards, ailerons, and flaps, to allow a reduction in the number of aircraft states. Stevens extended this research further by including failed reverser vanes, a failed velocity sensor, and a failed flight path sensor. He first investigated "soft" or partial failures of these sensors and surfaces, with the soft failures modeled as either partial power to a flight surface, an increase in the sensor noise, or an increase in the sensor bias. He then investigated the performance of a hierarchical structure of multiple model adaptive controllers to detect the presence of double

failures, such as a stabilator surface failure followed by a flight path angle sensor failure. This hierarchical structure of controllers started with a bank of controllers, with different single failure models, that would detect the first failure and then switch to another bank of controllers that had models that assumed both the detected failure and a second failure. Each of the banks also had a controller with a failure model that assumed that the detected failure had not occurred after all, which allowed the structure to correct any misidentifications. Pogoda and Stevens both found that the MMAC structure was able to identify the failures properly and reconfigure the control law to maintain stable flight control. Stevens found that the MMAC would blend the appropriate controller commands in the presence of soft failures, and that the hierarchical structure would properly detect multiple failures.

In a similar manner, Stratton [42] and Menke [33, 34, 35] have developed a MMAE algorithm for both the longitudinal and lateral axes of the VISTA F-16 aircraft. The emphasis of their design was to develop a failure detector that would work with an existing flight controller. Therefore, they developed a MMAE that produced the best estimate of the failure status of the aircraft and the aircraft states, which was then used by the aircraft's controller to provide appropriate flight control in the presence of the failure. Stratton used a flight condition of Mach 0.8 and altitude 10,000 feet, while Menke designed for a flight condition of Mach 0.4 and altitude 20,000 feet, the latter involving low dynamic pressures and thus presenting a more difficult failure detection problem. They investigated both single and multiple, hard and soft, actuator and sensor failures. Included in their study was the effect of a test signal or commanded dither to aid in identifying the failure during benign straight and level flight conditions. Several dither signals were tested, including sine waves, square waves, triangular waves, and pulse trains, at levels that were deemed either subliminal (up to $\pm$ 0.1 g's

in the longitudinal axis and ± 0.2 g's in the lateral axis) or non-subliminal (reasonable physical acceleration limits at the pilot's station). They found that the MMAE identified these failures correctly as long as an appropriate dither signal was present that excited all failure modes in both axes.

2.3.2 Other Applications

Multiple model adaptive algorithms have been successfully developed for a number of other applications. It was investigated for the detection and tracking of maneuvering targets [5, 9, 10, 24, 30, 36, 37, 44, 45, 46], flexible space structure control [8, 16, 17, 19, 20, 40], multiple hypotheses testing [1, 31], and to prevent the initial divergence of extended Kalman filters due to large initial uncertainties [15, 32]. It has been studied for use in diverse applications such as instrument failure detection in a pressurized water fusion reactor [7], autonomous monitoring of cardiac patients [12], adaptive signal processing of seismic data [39], and detection of incidents on freeways [48].

Chapter III: Methodology

3.1 Chapter Overview

In this chapter we develop the MMAE algorithm and the software simulator that we used to test the MMAE. We start with an overview of the MMAE structure and then develop the Kalman filter and hypothesis testing algorithms. We then describe the computation of the steady state Kalman filter gains. Then we develop a truth model of the LAMBDA that is used to provide the MMAE with a simulation of noise-corrupted measurements and commanded inputs during performance testing. Lastly, we describe the various outputs that were used as diagnostics during the software development and we present the format of the results of the performance testing that will be used in Chapter IV.

3.2 Multiple Model Adaptive Estimation Algorithm

3.2.1 Overview

A MMAE consists of a bank of parallel Kalman filters, each with a different internal model, and a hypothesis conditional probability computation as shown in Figure 6. The Kalman filters are provided a measurement vector (z) and the input vector (u), and produce a state estimate ($\hat{x}_i$) and a residual (r_i). Each Kalman filter has a different failure model that it uses to form the state estimate and the residual, so the sizes of the residuals give a relative indication of how wrong each of these models are. The residuals are used by the hypothesis testing algorithm to assign relative probabilities (p_i) to each of the hypotheses that were used to form the Kalman filter models. The individual probabilities indicate how correct each of the Kalman filter models are, and can be used to weight the individual state estimates

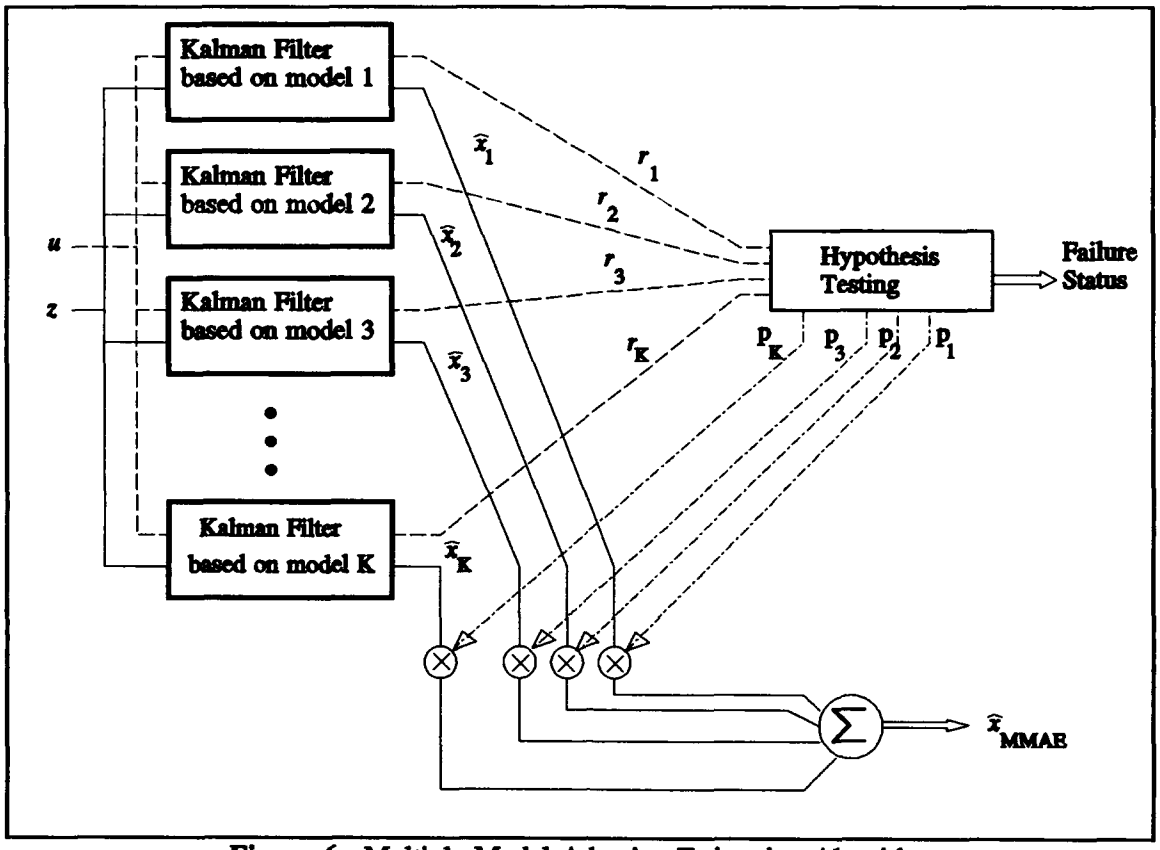


Figure 6. Multiple Model Adaptive Estimation Algorithm

appropriately, to form a probability-weighted average.

An example would demonstrate the workings of the MMAE more clearly. Let us assign Kalman filter 1 the fully functional aircraft model, and Kalman filter 2 the left elevator failure. These filters would use the commanded input and compute an estimate of what the state measurements should be. Included in this computation is a model of the noise in these measurements, which the filters use to calculate an expected value and standard deviation of the measurement. The actual measurement is then used to form the residual, which is the difference between the actual measurement and what the filter expected for the measurement. If the aircraft has no failures, then the residual from filter 1 would be much smaller, relative to

its own internally computed standard deviation, than the residual in filter 2. The hypothesis testing algorithm would take all of the residuals and assign the highest probability to filter 1 since it has the smallest residual. Now, let us assume that a left elevator failure occurs, thus the residual in filter 2 would become quite small and the residual in filter 1 would grow. The hypothesis testing algorithm would then assign less probability to filter 1 and more to filter 2. The specific workings of these blocks are explained in the following sections.

3.2.2 Kalman Filters

In this section we assume that the reader is familiar with Kalman filtering and present the specific Kalman filter equations used in the MMAE. Maybeck [24] presents an excellent development of Kalman filtering theory, if further Kalman filtering background is needed.

We are implementing this algorithm on a digital computer, which requires a discrete-time system model, and we also need to minimize the computational loading. Previous research [43] developed a continuous time system model for the LAMBDA, which we will convert to a discrete-time equivalent model. To minimize computational loading, we assume that the system is time invariant, giving us constant, precomputable coefficients for the system model (Φ, B_d, G_d, H), and we use a linear, steady state Kalman filter, which gives us constant Kalman filter gains (K).

In Section 3.3.1 we will develop a discrete time equivalent system model of the form

$$\begin{aligned} x(t_i) &= \Phi\, x(t_{i-1}) + B_d\, u(t_{i-1}) + G_d\, w_d(t_{i-1}) \\ z(t_i) &= H\, x(t_i) + v(t_i) \end{aligned} \tag{1}$$

where x is the system state vector

Φ is the state transition matrix, the discrete equivalent of the system dynamics matrix

B_d is the discrete equivalent of the system control input matrix

u is the system input vector

G_d is the discrete equivalent noise input matrix

w_d is an additive white dynamics noise input with zero mean and

$$E\{w_d(t_i)\, w_d^T(t_j)\} = \begin{cases} Q_d(t_i) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases} \tag{2}$$

z is the measurement vector

H is the system output matrix

v is an additive white measurement noise input, independent of w_d, with zero mean and

$$E\{v(t_i)\, v^T(t_j)\} = \begin{cases} R(t_i) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases} \tag{3}$$

Using this model we get the Kalman filter state estimate propagation equation:

$$\hat{x}(t_i^-) = \Phi\, \hat{x}(t_{i-1}^+) + B_d u(t_{i-1}) \tag{4}$$

where t_i^- is the time just before the ith time sample, and

t_{i-1}^+ is the time just after the i - 1 time sample,

and the Kalman filter update equation:

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K[z(t_i^-) - H\hat{x}(t_i^-)] \tag{5}$$

This model also gives us the Kalman filter residual:

$$r(t_i) \triangleq z(t_i) - H\hat{x}(t_i^-) \tag{6}$$

which the hypothesis testing algorithm uses as a relative measure of how much the Kalman filter model differs from the true model. The residual is the difference between the true measurements (z), and the Kalman filter estimates of those measurements before they are taken ($H\hat{x}(t_i^-)$), which are based on its model. Thus, if the Kalman filter model is correct, the residual will be close to zero, otherwise the residual will be larger than anticipated when compared to the filter-computed residual covariance matrix, which is shown in (7) below.

3.2.3 Hypothesis Testing

Maybeck shows [24:228-229] that the Kalman filter residual is a white Gaussian sequence of mean zero and covariance

$$H(t_i)P(t_i^-)H^T(t_i) + R(t_i) \tag{7}$$

For our application, which assumes a time invariant system, this covariance is constant and precomputable. For the k-th elemental Kalman filter the covariance is

$$A_k = H_k P_k H_k^T + R_k \tag{8}$$

Since we know that the residual is a Gaussian vector with zero mean and covariance A_k, we substitute these values into the known expression for a Gaussian conditional density function. Therefore we get that the conditional density function of the measurement (z) at t_i , given the Kalman filter model (a_k) and the measurement history ($Z(t_{i-1}) = [\, z^T(t_i) \vdots \cdots \vdots z^T(t_{i-1})]$), is

$$f_{z(t_i)|a,Z(t_{i-1})}(z_i|a_p,Z_{i-1}) = \beta \exp\{\cdot\}$$

$$\beta = \frac{1}{(2\pi)^{m/2}|A_k|^{1/2}} \tag{9}$$

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

We desire the conditional probabilities of the various hypotheses, $p_k(t_i)$, which we can compute using (9) and the previous conditional probabilities $p_k(t_{i-1})$:

$$p_k(t_i) = \frac{f_{z(t_i)|a,Z(t_{i-1})}(z_i|a_p,Z_{i-1})p_k(t_{i-1})}{\sum_{j=1}^{K} f_{z(t_i)|a,Z(t_{i-1})}(z_i|a_p,Z_{i-1})p_j(t_{i-1})} \tag{10}$$

Stevens [29, 41] found that certain performance problems could be reduced by modifying (9). He altered the conditional density function in (9) by removing the β term, which was used to make the area under the density function equal to 1. If all the exponential terms, $\{\cdot\}$ in (9), were approximately the same size for all elemental filters, (9) and (10) would put the highest probability on the elemental filters with the smallest $|A_k|$ value. This is an inappropriate weighting since the size of $|A_k|$ has nothing to do with the correctness of the hypothesis in matching the current real-world failure status. Since sensor failures exhibit themselves as a row of H going to zero, filters based on the hypothesis of a failed sensor tend to have smaller $|A_k|$ values, and thus the MMAE will be prone to false alarms on sensor failures. The algorithm functions properly with the β term removed because the denominator in (10) is the sum of all possible numerators, so the probabilities (p_k) will still sum to 1 even if the area under each of the modified "densities," the density function in (9) with β removed, is no longer unity.

Previous research [3, 28, 29, 38, 41] also found that a lower bound needed to be placed on the probabilities. The purpose of the MMAE is to make quick and accurate failure identifications, and it was found that if some of the probabilities were allowed to get too small, it took a long time for the probability in the correct filter to build up when a change in the failure status occurred. This was due to the fact that the previous conditional probability, $p_k(t_{i-1})$, was so small for the new correct filter model and so large for the old correct model, that (10) had to be iterated several times before the values would change significantly. Menke and Stratton [33, 34, 35, 42] found that 0.001 was a good lower bound on the conditional probabilities. This is the implementation of the hypothesis testing algorithm used for this application of the MMAE.

3.3 Multiple Model Adaptive Estimator Simulation

Now that we have a MMAE algorithm, we need to develop a simulation that will characterize the performance of the MMAE for the LAMBDA flight vehicle. First, we will determine the precomputable coefficients for the system model (Φ, $\boldsymbol{B_d}$, $\boldsymbol{G_d}$, $\boldsymbol{H}$), and the constant Kalman filter gains ($\boldsymbol{K}$). Then we will develop the truth model that will be used to generate the measurements of the aircraft states for a given input. Lastly, we outline the variables that were used as diagnostic tools and measures of performance for the MMAE.

3.3.1 Steady State Kalman Filter Coefficients Computation

The MMAE algorithm requires several precomputable coefficients for each Kalman filter model. To obtain these coefficients, we start with a continuous-time model of the form

$$\dot{x} = Fx + Bu + Gw \tag{11}$$

This continuous-time model for the LAMBDA was developed by Swift [43].

We then find the discrete-time equivalent system model shown in (1) and define the following variables from these equivalent system models:

H, the measurement matrix in (1) is defined by the measurements of the Kalman filter states that are available,

G, the white noise input matrix in (11) is defined by the states that are affected by the white noise input ***w***,

G_d, the discrete time equivalent matrix to ***G*** shown in (1), which is defined to be I for the discrete time equivalent model,

Q, the strength of the additive white noise (***w***) in (11), which is defined in the truth model development (Section 3.3.2) and modified if the Kalman filters need tuning to attain good performance,

R, the measurement noise covariance, defined by estimates of the noise in the various measurements (i.e., sensor precision) and modified if the Kalman filters need tuning to attain good performance.

The estimates of ***R*** were empirically determined using the estimated standard deviation of aircraft measurement data taken when the aircraft was quiescent (sitting on the ground with no movement).

We used the continuous-time state model (11) to compute the state transition matrix (Φ). We start with the given flight condition, compute the state space model of the continuous-time system, zero out the appropriate column or row to simulate the desired failure

model, and then use a MATRIX$_x$ intrinsic function (SPLIT) to find the discrete-time equivalent Φ for this system. This method was verified by propagating x for several sample times using the equation

$$x(t_i) = \int_{t_0}^{t_i} Fx(\tau)\, d\tau + x(t_0) \tag{12}$$

and comparing its value against the discrete-time equivalent equation

$$x(t_i) = \Phi^i x(t_0) \tag{13}$$

With Φ evaluated for one sample period ($\Phi(t_i, t_{i+1})$), we can find second order approximations to B_d and Q_d where

$$\begin{aligned} B_d &= \int_{t_i}^{t_{i+1}} \Phi(t_{i+1} - \tau) B d\tau \approx \frac{1}{2}(\Phi B + B)\Delta t \\ Q_d &= \int_{t_i}^{t_{i+1}} \Phi(t_{i+1} - \tau) G Q G^T \Phi^T(t_{i+1} - \tau)\, d\tau \\ &\approx \frac{1}{2}(\Phi G Q G^T \Phi^T + G Q G^T)\Delta t \end{aligned} \tag{14}$$

To find the steady state Kalman filter gains, we propagated and updated the filter covariance using

$$P(t_i^-) = \Phi P(t^+_{i-1}) \Phi^T + G_d Q_d G_d^T \tag{15}$$

for the propagation and

$$\begin{aligned} \boldsymbol{K}(t_i) &= \boldsymbol{P}(t_i^-)\boldsymbol{H}^T\left[\boldsymbol{H}\boldsymbol{P}(t_i^-)\boldsymbol{H}^T + \boldsymbol{R}\right]^{-1} \\ \boldsymbol{P}(t_i^+) &= \boldsymbol{P}(t_i^-) - \boldsymbol{K}(t_i)\boldsymbol{H}\boldsymbol{P}(t_i^-) \end{aligned} \tag{16}$$

for the update, until the value of $\boldsymbol{K}$ reached steady state. This gave us $\boldsymbol{K}_{ss}$ and $\boldsymbol{P}_{ss}^-$, which we used to compute $\boldsymbol{A}_k$ and its inverse. This gave us all the precomputable coefficients for the various Kalman filter models, which were then stored in data files for use by the MMAE.

3.3.2 Truth Model Development

The truth model was used to generate the simulation of the measurements that the LAMBDA aircraft would normally provide the MMAE. The truth model also started with the continuous time system model developed by Swift [43], but we augmented it with an eight-state Dryden wind model used in previous research [33, 38, 41, 42], with some adjustment for altitude and speed. The software module used in previous research to integrate the truth model proved to be inadequate for this application. This application involves an open-loop aircraft (no flight controller) which yielded a stiff differential equation (widely separate eigenvalues) that had to be integrated. We ended up using MATRIX$_x$ to integrate this differential equation because of this deficiency. The integrated values were sampled at the measurement sampling frequency and stored in a data file, along with the aircraft input vector, that was used by the MMAE as the source of its measurement and input vectors.

3.3.3 Diagnostics and Performance Measures

We used several variables from the preceding equations as diagnostic tools to verify the operation of the MMAE. The individual scalar elements of the residual vector ($\boldsymbol{r}$) were used to verify the operation of the elemental Kalman filters. The residuals in a well tuned

Kalman filter with the correct model will be substantially smaller, relative to the filter-computed covariance, than the ones in the incorrect filters. Thus, the elements of the residual vector gave a general indication of how well the Kalman filters were tuned. The $\{\cdot\}$ term and conditional probability density in (9) were used to verify the operation of the hypothesis testing algorithm. The incorrect Kalman filters should have much larger values for these terms (i.e., larger values of $\boldsymbol{r}^T \boldsymbol{A}_k^{-1} \boldsymbol{r}$), and when a failure occurred, the $\{\cdot\}$ term gives an indication of how fast the conditional probabilities will change. All these terms aided in diagnosing and verifying the operation of the MMAE.

The best indicator of the MMAE performance was a smoothed version of the individual filter conditional probability, $p_k(t_i)$ in (10), that is generated by the hypothesis testing algorithm. Previous research [33, 42] has shown that some smoothing of the filter conditional probabilities is needed to prevent false declarations. For our implementation, we smoothed the probabilities over a variable length data window (PWINSIZ), that was usually set at 10 samples, and then tested for excursions above the threshold level. If the probability, averaged over the data window, exceeded the threshold level, a failure was declared that corresponded to the Kalman filter model that had the highest probability.

The conditional probabilities for each of the Kalman filters were plotted versus time for various failure scenarios to visualize the MMAE performance. A typical plot is shown in Figure 7, where a right elevator failure is simulated at 2 seconds. Note that the probability in the fully functional filter is high while all the other filters are essentially zero until the failure occurs, then the probability in the right elevator failure filter grows as the probability in the fully functional filter decreases. This behavior was observed to occur for most of the failure scenarios, so the information was combined into one plot, shown in Figure 8. This plot shows

how the probabilities for each of the filters grows when a failure that matches its own model occurs. We found that, in all cases except an aileron failure, the probability in the fully functional filter was an inverse of the probabilities shown in each plot (i.e., one minus that probability value) and the probabilities assigned to the other filters were minimal. For instance, the right elevator failure shown in Figure 7 is represented by the plot of the right elevator filter in Figure 8. You can see that the probability in the right elevator failure matches in both plots, while the fully functional probability is simply the inverse of the right elevator probability and all the other probabilities are at their minimums. The plotting format in Figure 8 will be used throughout the rest of this document, except as noted. Also, to prevent discontinuities in the text, we will group full page figures, such as Figure 7 and Figure 8, at the end of the section in which they are referenced.

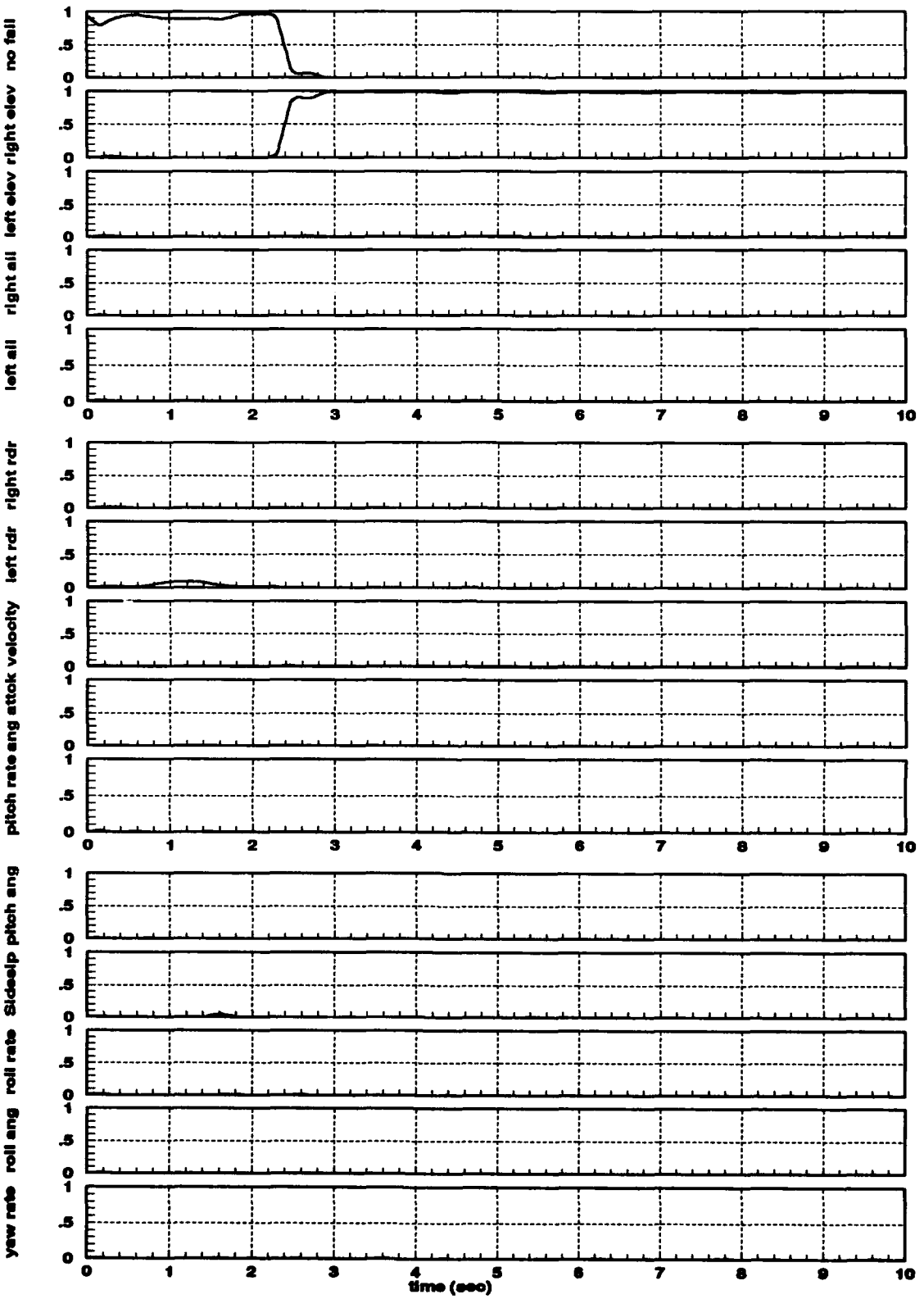


Figure 7. Failure #2 - Right Elevator Actuator

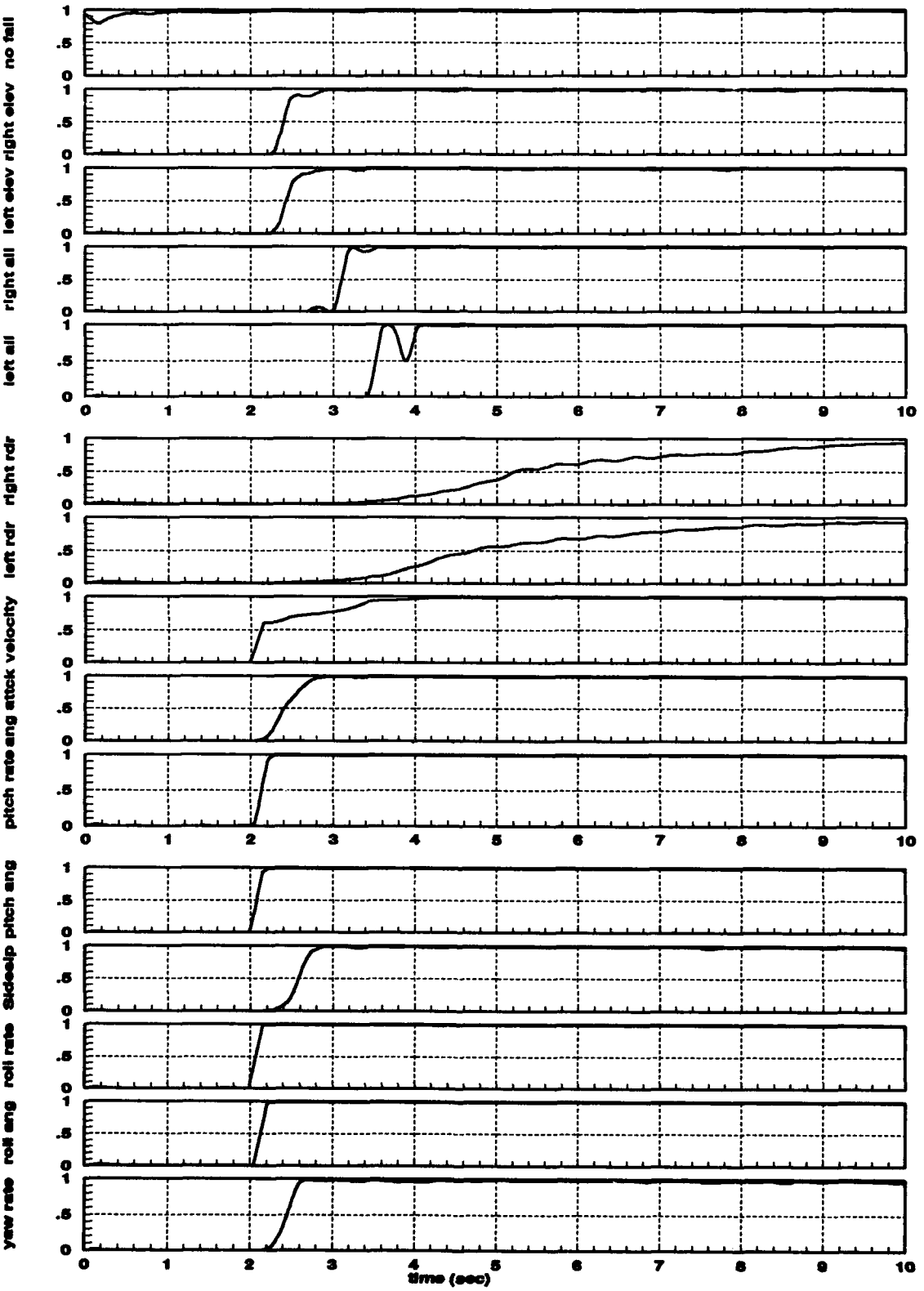


Figure 8. MMAE Plotting Format.

Chapter IV: Results

4.1 Chapter Overview

This chapter presents the results of our investigation into the failure detection capabilities of the MMAE for the LAMBDA flight vehicle. These results are organized to correspond with the research questions presented in Section 1.5, Research Questions. In Section 4.2, Decision Convergence, we present the MMAE's baseline decision convergence performance. Section 4.3, Decision Switching, presents the results of various attempts to improve this performance and the effect that either an incorrect decision or improper initialization has on the MMAE's performance. Finally, in Section 4.4, Unmodeled Effects, we present the MMAE's performance at various flight conditions located on the boundaries of the flight envelope. Throughout this chapter we have tried to present a modest number of results to illustrate our findings. An exhaustive set of results is on file with Dr. Peter Maybeck, Air Force Institute of Technology, School of Engineering, Wright-Patterson Air Force Base, Ohio.

4.2 Decision Convergence

The baseline performance of the MMAE developed in Chapter III is shown in Figure 9. In Section 3.3.3 we described the format of this plot, where each strip plot shows the probabilities of the correct filter for a separate simulation run. For instance, the no-fail strip plot shows the probabilities assigned to the no-failure hypothesis during a simulation run in which there were no failures induced. Likewise, the right elevator strip plot shows the probabilities assigned to the right elevator failure hypothesis during a simulation run in which

a right elevator failure was induced at 2 seconds, and so forth for each of the failure hypotheses that were modeled. Note that the no-failure hypothesis results are presented first, followed by all the actuator failure hypotheses results, and then the sensor failure hypotheses results. Also, each plot is the average probabilities over a 10-run Monte Carlo simulation. Normally a $\pm 1\sigma$ plot is superimposed on a multiple run simulation to indicate how variable each run is compared to the average run. In most cases the $\pm 1\sigma$ plot and the 10-run average were so close that it was impossible to distinguish them. This shows that most single run performances are very close to the 10-run average performance, which was verified by plotting some single run simulations. The plots that are presented in this thesis are 10-run Monte Carlo averages without the $\pm 1\sigma$ plots, unless otherwise noted.

We consider convergence to be when one of the smoothed filter probabilities remains above the decision threshold, while the other probabilities remain below the decision threshold. We used a decision threshold of 0.5 (50%), which will be shown as a dotted line on the various plots of the MMAE performance throughout this document. The baseline performance in Figure 9 shows that the MMAE converges to the correct failure condition within 3.2 seconds for all failure conditions.

In following sections we experiment with various techniques to improve the MMAE performance. To identify these techniques clearly, the values of the parameters that are adjusted for these various techniques are listed in the caption of the figure. We will test the MMAE performance using three different Kalman filter bank designs, which will be identified by the Design # in the figure caption. Then we will experiment with different values of the $\{\cdot\}$ term in (9) on Page 25. That value will be identified as the "Dot" term in the figure caption. We also experiment with different size data windows over which the probabilities are

smoothed. The window size will be indicated by "PWINSIZ" (probability window size) in the caption of Figure 9. Lastly, we will try propagating the Kalman filter estimates a few sample periods without updating. The number of sample periods that the estimates are propagated is indicated by "NPROP" (number of propagations) in the caption of Figure 9.

Some experimentation with various commanded inputs was required to attain this baseline performance. We found that the commanded input had to properly excite all the aircraft modes for the MMAE to converge well. For example, a 5 Hertz dither input in all channels does not excite any of the aircraft modes and Figure 10 shows that the MMAE cannot converge to the proper decision. Note that the probabilities in the filter with the wrong hypothesis are not at their minimums (0.001). Since the probabilities are forced to sum to 1, their combined effect is to lower the probability of the filter with the correct hypothesis so that it is below or at the decision threshold, thus the MMAE does not converge. This occurs when the Kalman filter residuals are all about equal in size (relative to the filter-computed residual standard deviations), so all of the filter hypotheses look equally likely, and the MMAE must allow some probability for the wrong hypotheses. To prevent this behavior, the commanded input had to have the correct frequency, strength, and lag between right and left actuators.

To find the frequency of an input that would provide good failure identification performance, we started with the eigenvalues of the continuous-time model and then experimented with various combinations of the natural frequencies. For the design point in the flight envelope these eigenvalues were $-2.928 \pm j\ 6.68$ and $-1.516 \pm j\ 0.260$ rad/sec in the longitudinal axis, and 0, -6.724, $-0.9424 \pm j\ 3.625$ rad/sec in the longitudinal axis.

Experimenting with inputs at the natural frequencies of 6.68 and 3.625 rad/sec, we found that the magnitude of the inputs had to exceed a certain value to produce good MMAE

performance. If the input was below this threshold, the aircraft modes were not excited enough and the performance was similar to Figure 10. Once this threshold was exceeded, the MMAE performed quite well. Also, increasing the magnitude above this threshold did not enhance the MMAE performance. We found that a ±10 degree roll axis input at 3.625 rad/sec, ±15 degree pitch axis input at 6.678 rad/sec, and a ±20 degree yaw axis input at 6.678 rad/sec produced good results. While these inputs seem rather large, the aircraft response was actually quite small because the inputs were almost exactly out of phase with each other, due to reasons that are explained in the next paragraph.

Not only does the input need to excite the aircraft modes, but a differential input (Figure 11) had to be applied for the MMAE to distinguish between a right and left actuator failure. A differential input commands one actuator with a positive command and the other actuator with a negative command, with positive and negative defined by the sign of the aircraft response to that command. For example, if the left elevator received a positive command, it would turn upwards in an attempt to produce a positive pitch, and if the right elevator received a negative command, it would turn downwards in an attempt to produce a negative pitch. Note that the aileron's response to the differential input differs from the rudder and elevator responses. If the left aileron received a positive command, it would turn downwards in an attempt to produce a positive roll, and if the right aileron received a negative command, it too would turn downwards in an attempt to produce a negative roll.

If both commands are equal, and both actuators are working properly, then the net response is zero according to our model for the LAMBDA. Good engineering judgment tells us that, if the elevators are used as described above, there would be some rolling motion induced by the elevators. However, the LAMBDA model does not predict this response due to

the lack of cross-axis terms that was noted in Section 1.5. We have assumed that this cross-coupling between the axes is small due to the small moment arm of the cross-axis flight surface (i.e., the elevator roll moment arm is small compared to the aileron roll moment arm).

When a non-differential input is applied (i.e., when the right and left actuator commands are the same), there is insufficient information to determine the difference between a right or left actuator failure. We found that the MMAE divides the probability equally between both hypotheses, when either actuator fails. To identify right vs left, we used a differential input so that if one actuator fails, the net response is no longer zero, and the failure becomes quite evident. We also found that the differential inputs had to be slightly shifted relative to each other (about 0.05 wavelengths) to provide some output so that the sensors were excited and a sensor failure could be detected. After experimenting with a few different inputs, we found that the input shown in Figure 11 provided good failure detection performance. This plot shows the commanded input to each actuator starting with the right and left elevators (rt elev and lt elev), the right outboard, right inboard, left inboard, left outboard flaps (rof, rif, lif, and lof), the right and left ailerons (rt ail and lt ail), and finally, the right and left rudders (rt rdr and lt rdr).

The MMAE seemed to be quite sensitive to the magnitude and frequency of the inputs. Figure 10 shows the performance of the MMAE when the frequency was not at the natural frequencies, and similar results were obtained when the magnitude was too low. Also, if the input was not shifted enough to excite the sensors, the performance would degrade rapidly. This seems to indicate that the input has to be rather precise in its magnitude, frequency, and phase relationship, to provide good failure identification. This would indicate that normal commanded inputs from a pilot will not provide good MMAE performance, so a purposeful

dither must be applied. This dither must excite *all* the failure modes that are modeled in the MMAE so the MMAE can converge to the correct failure condition.

The input that was used for this thesis is rather large, but because the inputs are almost exactly out of phase with each other, the aircraft response is actually quite small, as shown in Figure 12. This shows that the forward velocity (u) changes only 3 ft/sec, which is quite small compared to design point velocity of 180 ft/sec. Also, the angle of attack (alpha) varies only ± 0.5°, the pitch rate (q) varies ±5°/sec, and the pitch angle varies about ±1°. In the lateral axis the sideslip angle (beta) varies only ±0.8°, the roll rate (p) varies ±2.5°/sec, the roll angle (phi) varies ±0.8°, the yaw rate (r) varies ±4°/sec, and the yaw angle (psi) varies ±1°. The design of an *optimal* input is beyond the scope of this thesis, so other inputs were not tried once a good input was found.

Figure 9 shows that the MMAE does converge to the proper failure detection decision in less than 1 second for most of the failure conditions. However, a right aileron failure produces some confusion with the sideslip and yaw rate sensors for about 1 second before it finally converges to the correct failure decision, as shown in Figure 13. Note that Figure 13 is a plot of a single 10-run Monte Carlo simulation where the right aileron actuator is failed at 2 seconds; as opposed to results such as Figure 9, in which each time history is derived from a separate 10-run Monte Carlo simulation. Similarly, a left aileron failure produced some confusion for almost 2 seconds, as shown in Figure 14. The difference between these results might possibly be due to the phasing of their respective inputs.

None of the other failure conditions produced the confusion observed for an aileron failure, which compelled us to investigate the cause for this behavior. The noise measurements that were used to determine the values for $\boldsymbol{R}$ in (3) on Page 23, showed that the

sideslip and yaw rate sensors were the least noisy of all the sensors. The standard deviations for the residuals that are formed using these sensor measurements would be small compared to the other sensors. Therefore, any residuals that are larger than expected from these sensors would show up very quickly, and this would be interpreted as a failure of one of these sensors. Eventually, the residuals from the other sensors would become large enough to correct any erroneous declarations, but the end result is that the MMAE is more likely to declare a failure in these sensors, if there are residuals that are larger than expected. In Section 4.3.1 we present an attempt to decrease the amount of time that an aileron failure causes confusion.

Figure 9 also shows that a rudder failure condition took substantially longer to converge to the proper decision than any of the other failure conditions. Note that the rudder failures took about 3 seconds to cross the 50% probability threshold that appears as the dashed line in the plots, while most of the other failures took less than 1 second. In Sections 4.3.2, 4.3.3, and 4.3.4, we present various attempts to decrease the amount of time it took for the MMAE to converge for a rudder failure.

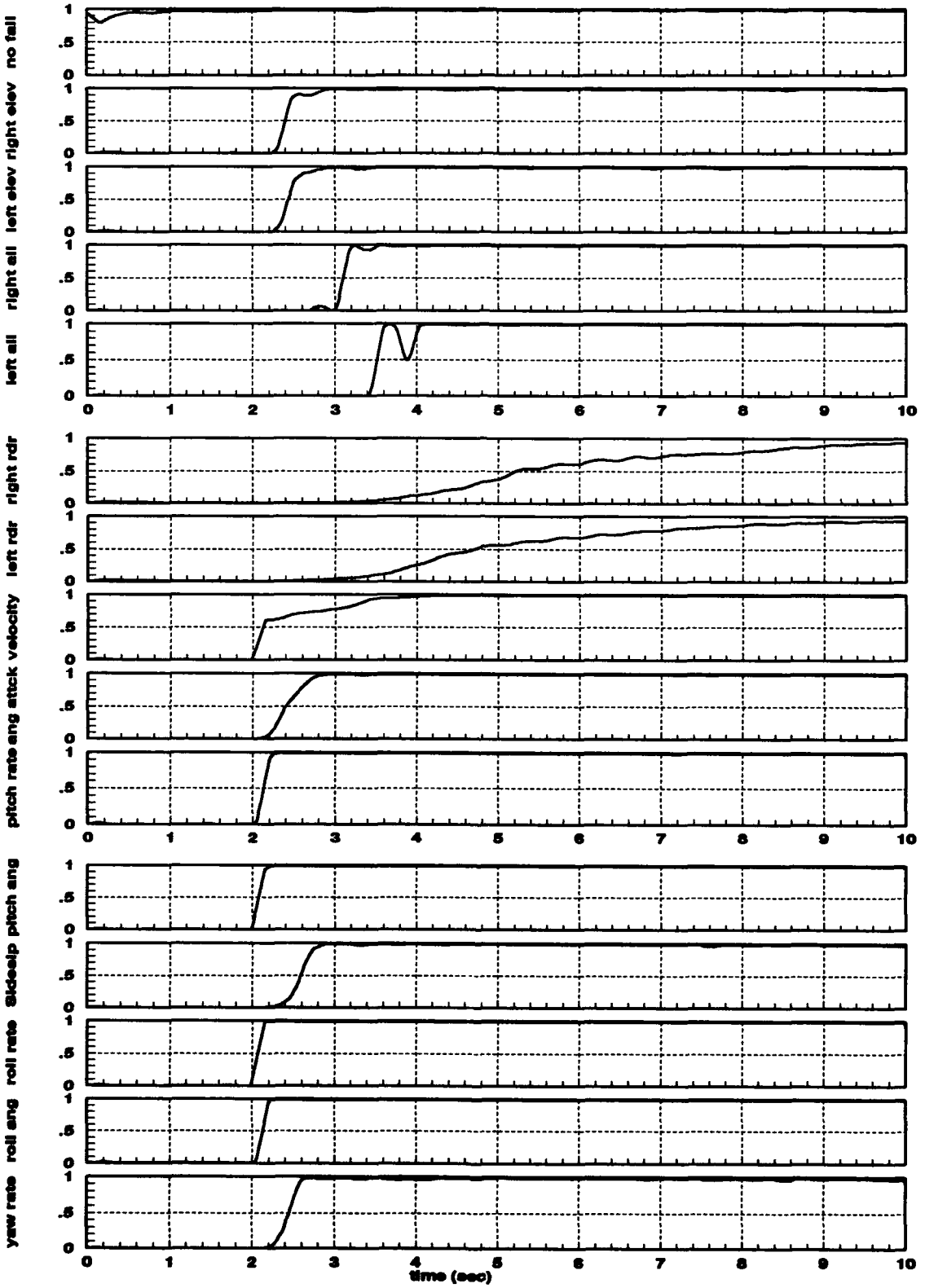


Figure 9. Design #1, Dot = -0.5, PWINSIZ = 10, NPROP = 1

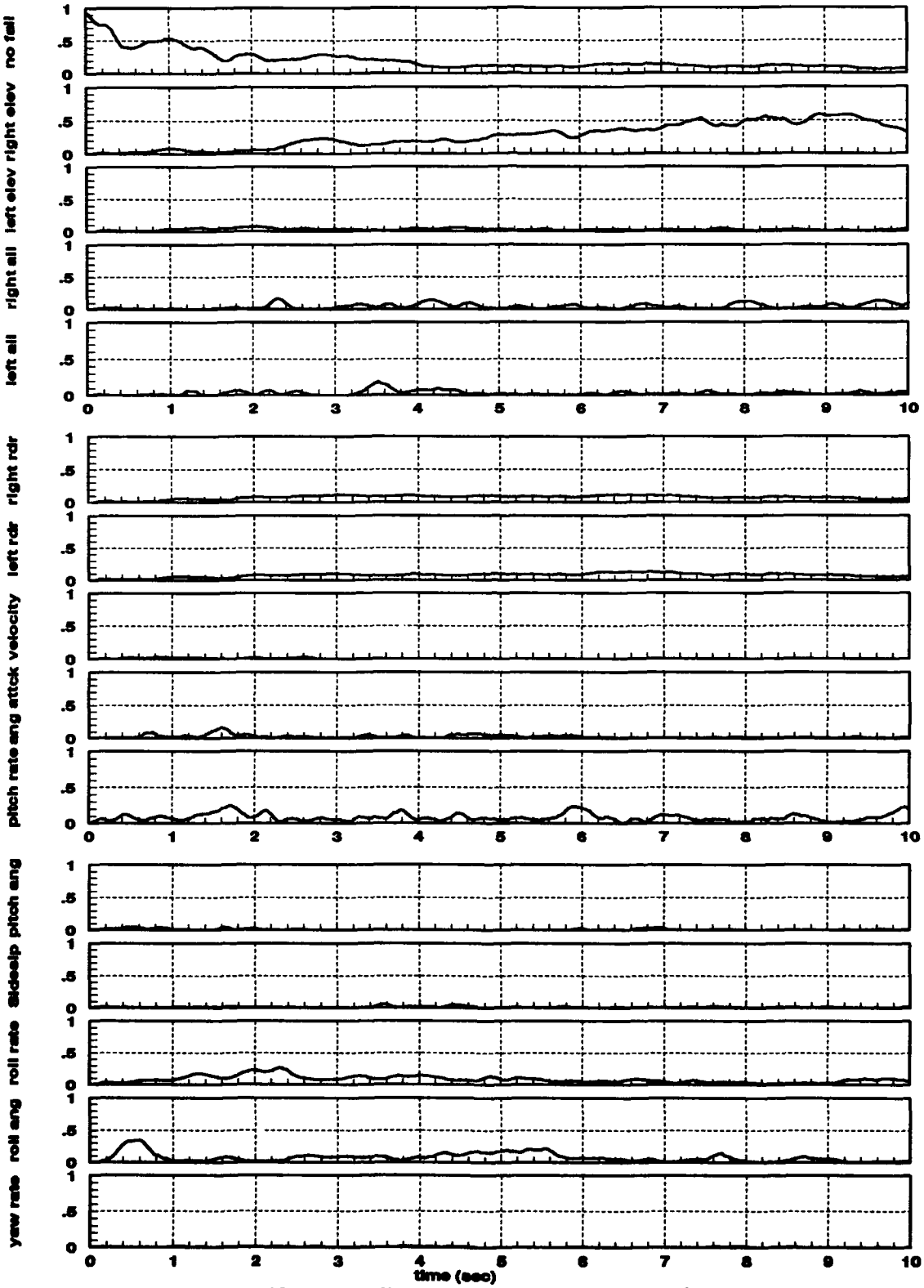


Figure 10. Right Elevator Failure with a 5 Hertz input.

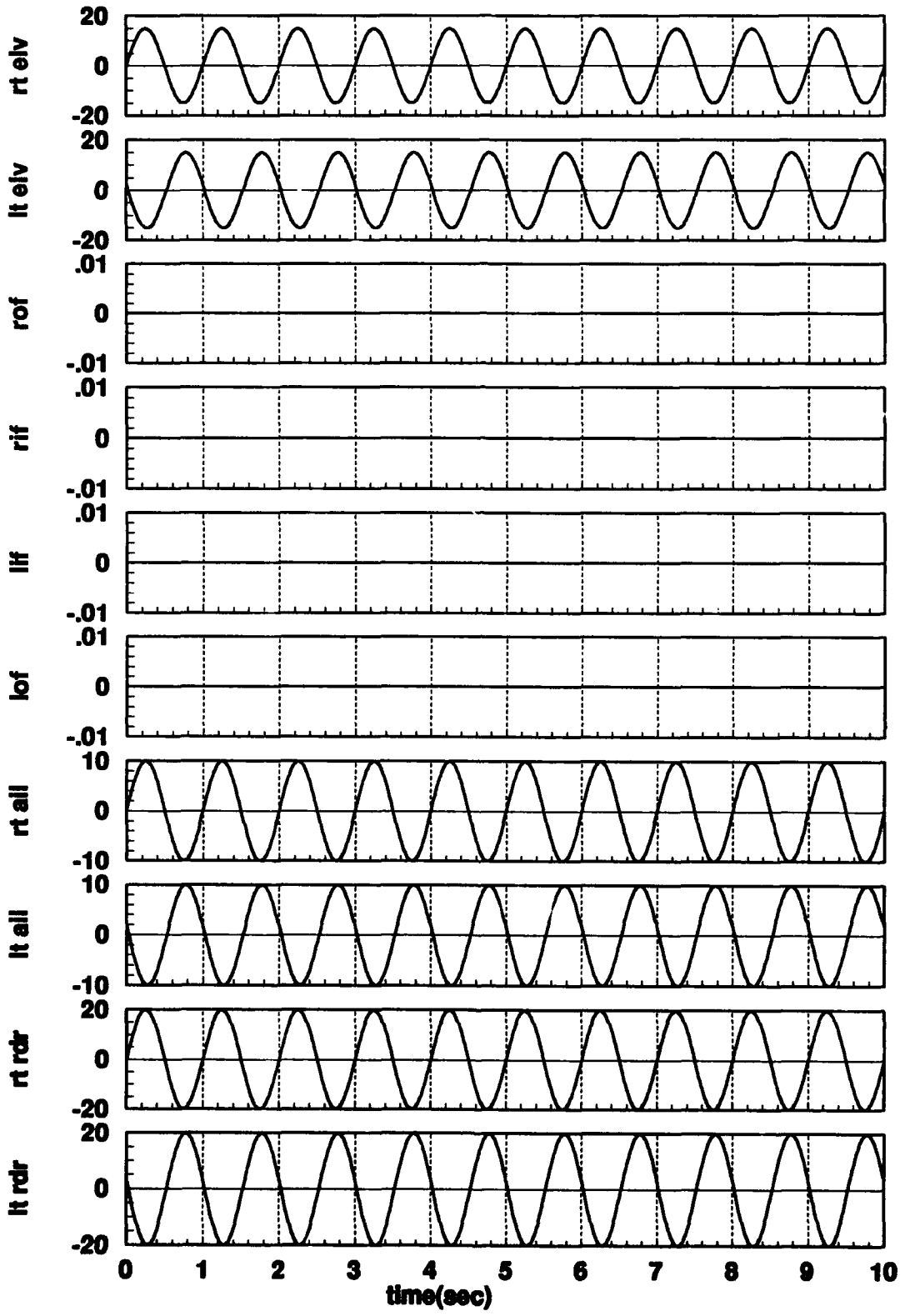


Figure 11. Differential Input.

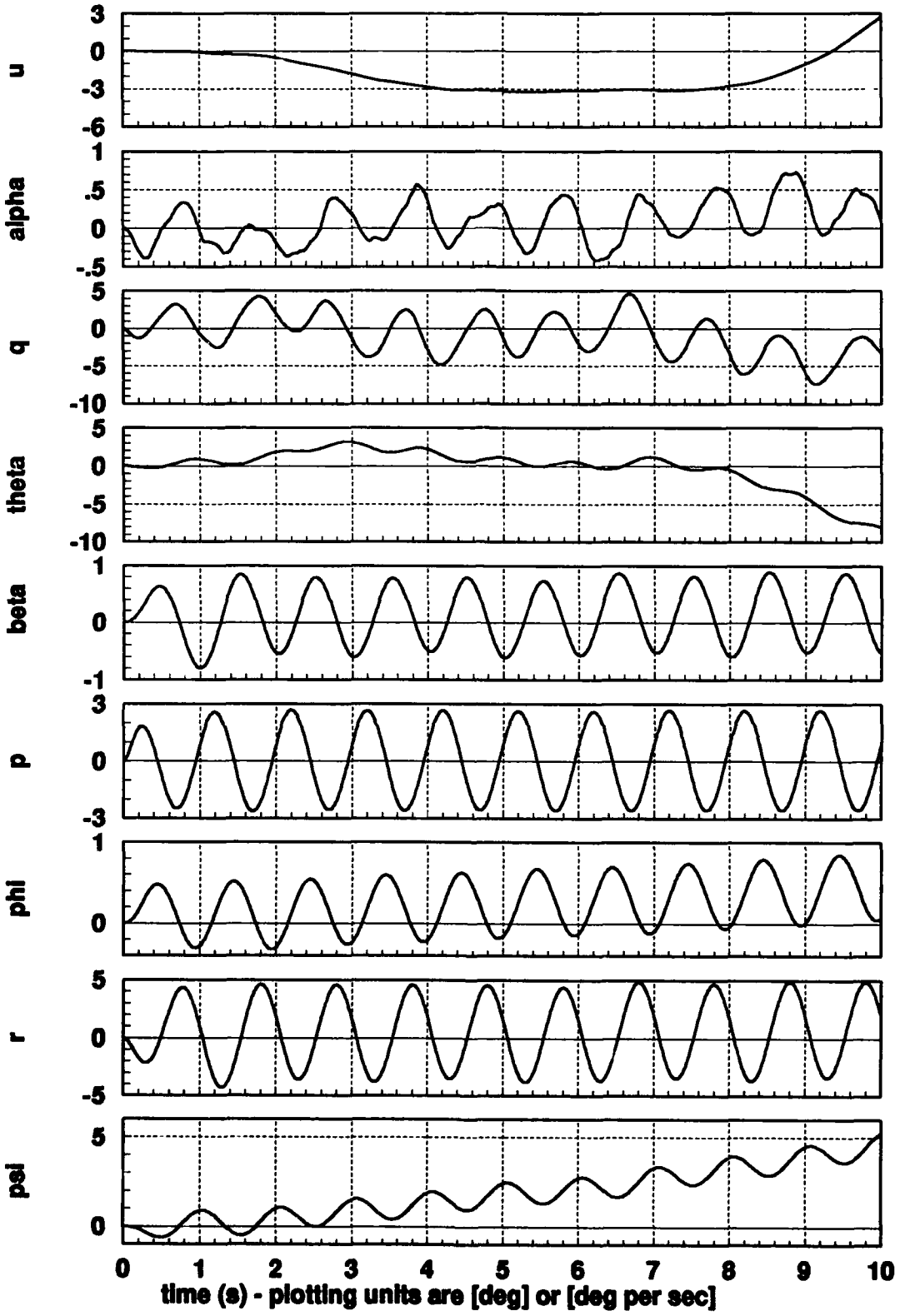


Figure 12. Aircraft response to differential input.

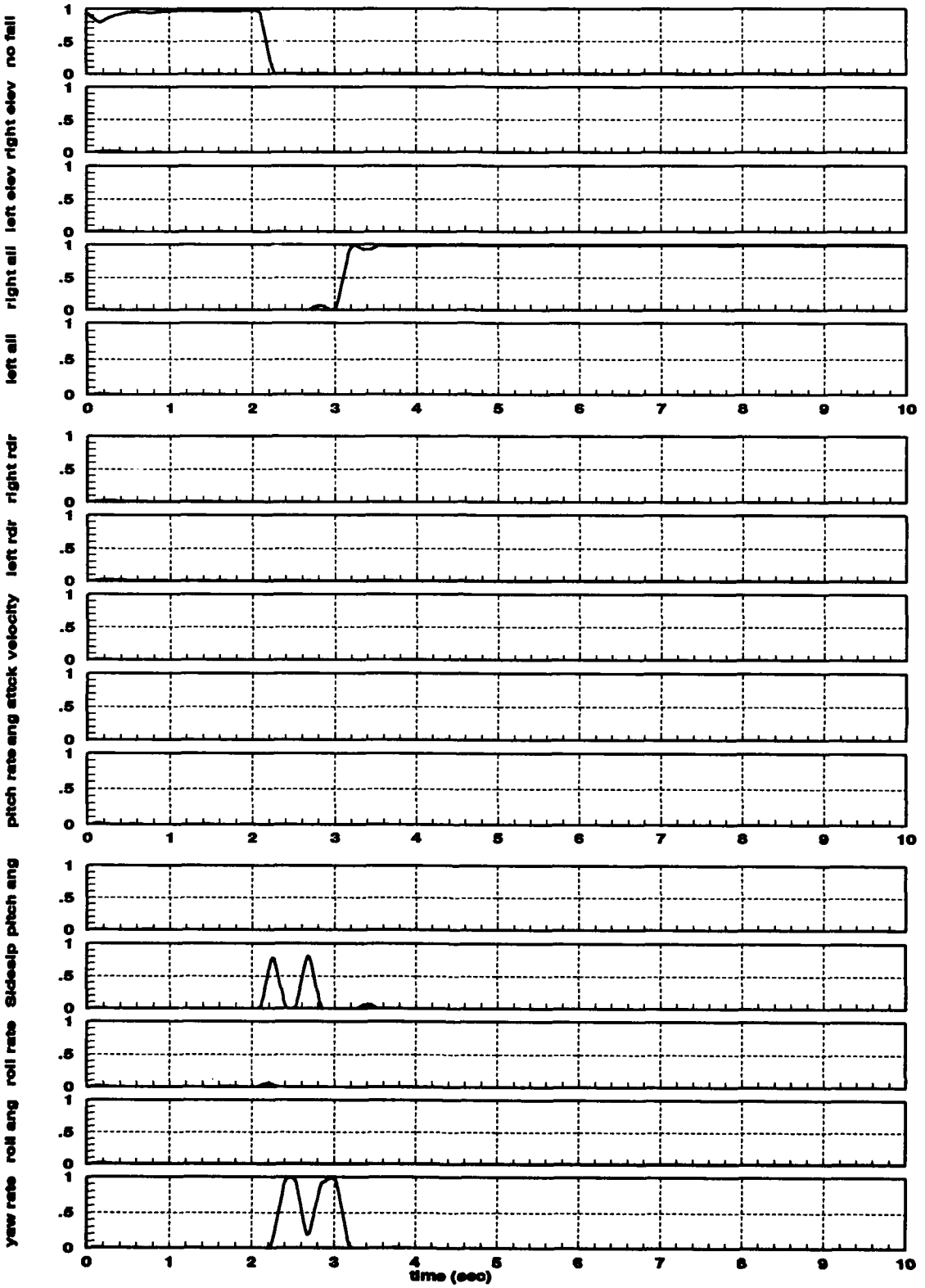


Figure 13. Right Elevator Failure, Design #1, Dot = -0.5, PWINSIZ = 10, NPROP = 1

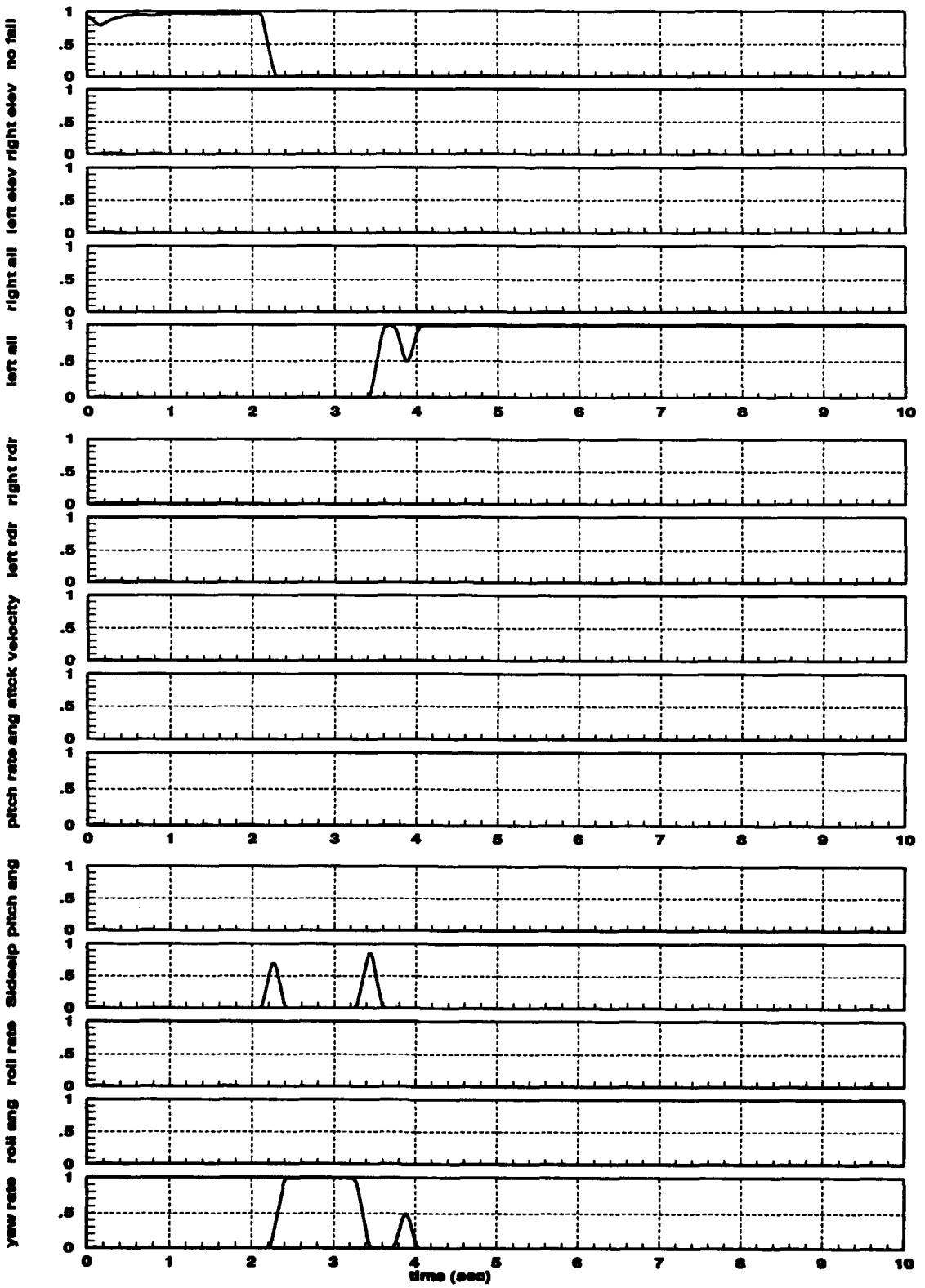


Figure 14. Left Elevator Failure, Design #1, Dot = -0.5, PWINSIZ = 10, NPROP = 1

4.3 Decision Switching

In this section we present the results of various attempts to decrease the amount of time that the MMAE took to switch to the correct failure condition when a failure occurred. We start by tuning the Kalman filters to minimize the confusion caused by an aileron failure. We then increase the exponential penalty for a larger than expected residual. The MMAE design in Chapter III smoothed the probabilities to minimize false alarm declarations, which introduces a delay in the decision convergence. This smoothing is decreased in an attempt to reduce the time it takes for the MMAE to converge to the correct failure condition. Lastly, the residuals from the Kalman filters are enlarged by propagating the state estimates a few sample periods without updating. The residuals in the Kalman filters with the wrong hypotheses become even larger, which causes the MMAE to converge even faster.

4.3.1 Kalman Filter Tuning

We experimented with increasing the measurement variances in the sideslip and yaw rate sensors, in an attempt to decrease the confusion (ambiguous and/or false declarations of the failure of these two sensors) during an aileron failure that was noted in the previous section. We developed three filter designs: design one used the sensor noise variances that were estimated from sensor measurement data, design two used nine times the sensor noise strength estimates (i.e., increasing the noise standard deviations by a factor of three), and design three used 25 times those estimates (multiplying the standard deviations by a factor of five). The MMAE performance using design one was presented in the previous section as the baseline performance (Figure 9), along with the right aileron performance (Figure 13) and the left aileron performance (Figure 14). The MMAE performance using design two is shown in

Figure 15, with the right aileron failure performance shown in Figure 16 and the left aileron failure performance shown in Figure 17. Likewise, the MMAE performance using design three is shown in Figure 18, with the right aileron failure performance shown in Figure 19 and the left aileron failure performance shown in Figure 20.

The best performance was obtained using design two. Figure 15 shows that the convergence time for an aileron failure using design two was half of that for design one (Figure 9). Figures 16 and 17 show that this improvement in convergence time is primarily due to a significant decrease in the amount of confusion caused by an aileron failure. Comparing the results from design one (Figure 13 and Figure 14) with those from design two (Figure 16 and Figure 17), we see that design two has eliminated the second hump in the probabilities for a sideslip and yaw rate sensor failure. This caused the MMAE to converge much faster to the correct failure condition.

Figures 18, 19, and 20 show that the performance for design three is almost exactly the same as the performance for design two, but the probabilities showed an increase in fluctuations. We noted before that these results are an average of 10 Monte Carlo simulations, therefore fluctuations in these plots would indicate even more severe fluctuations during flight testing, which would result in a larger number of false alarms, which are deemed unacceptable.

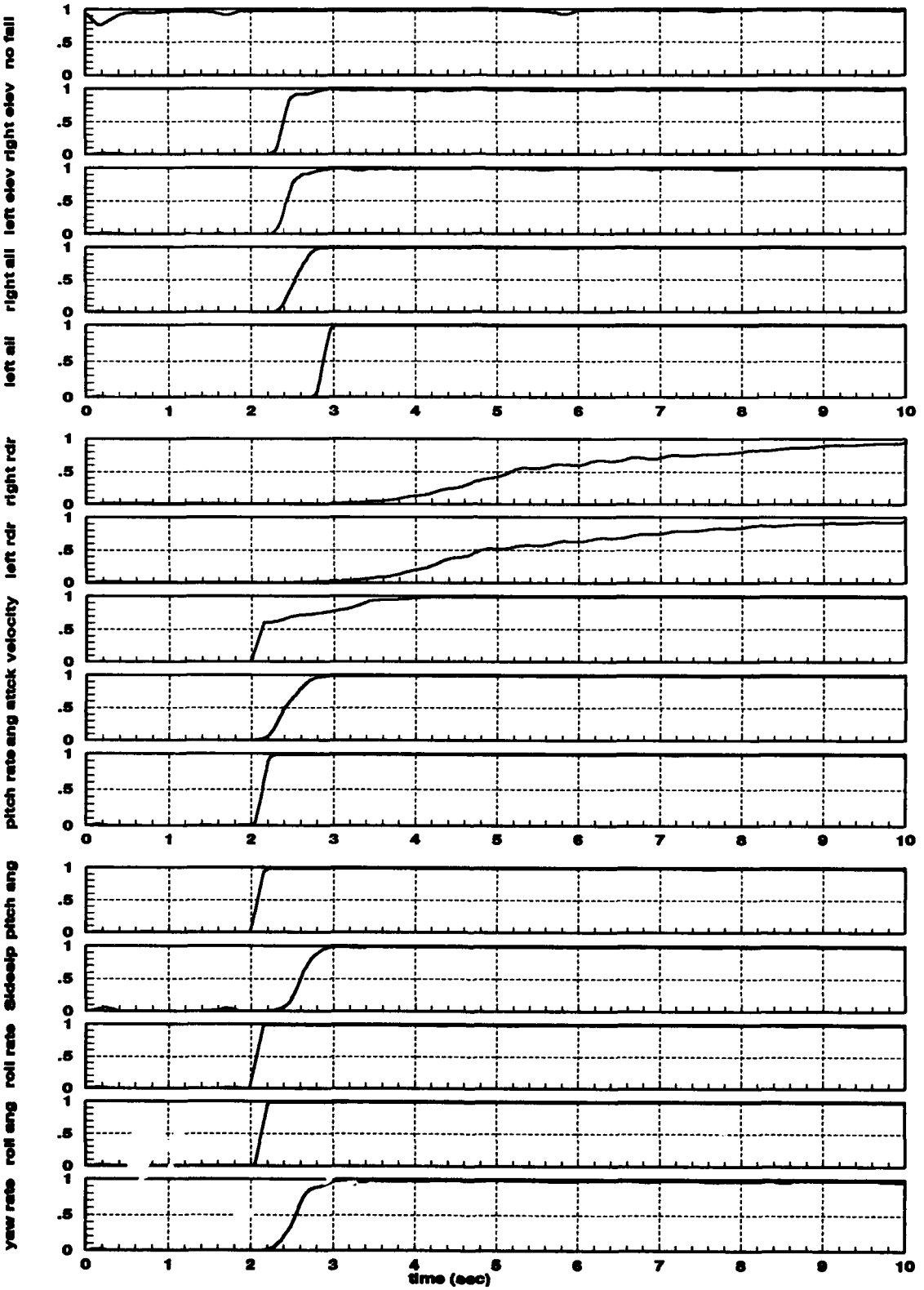


Figure 15. Design #2, Dot = -0.5, PWINSIZ = 10, NPROP = 1

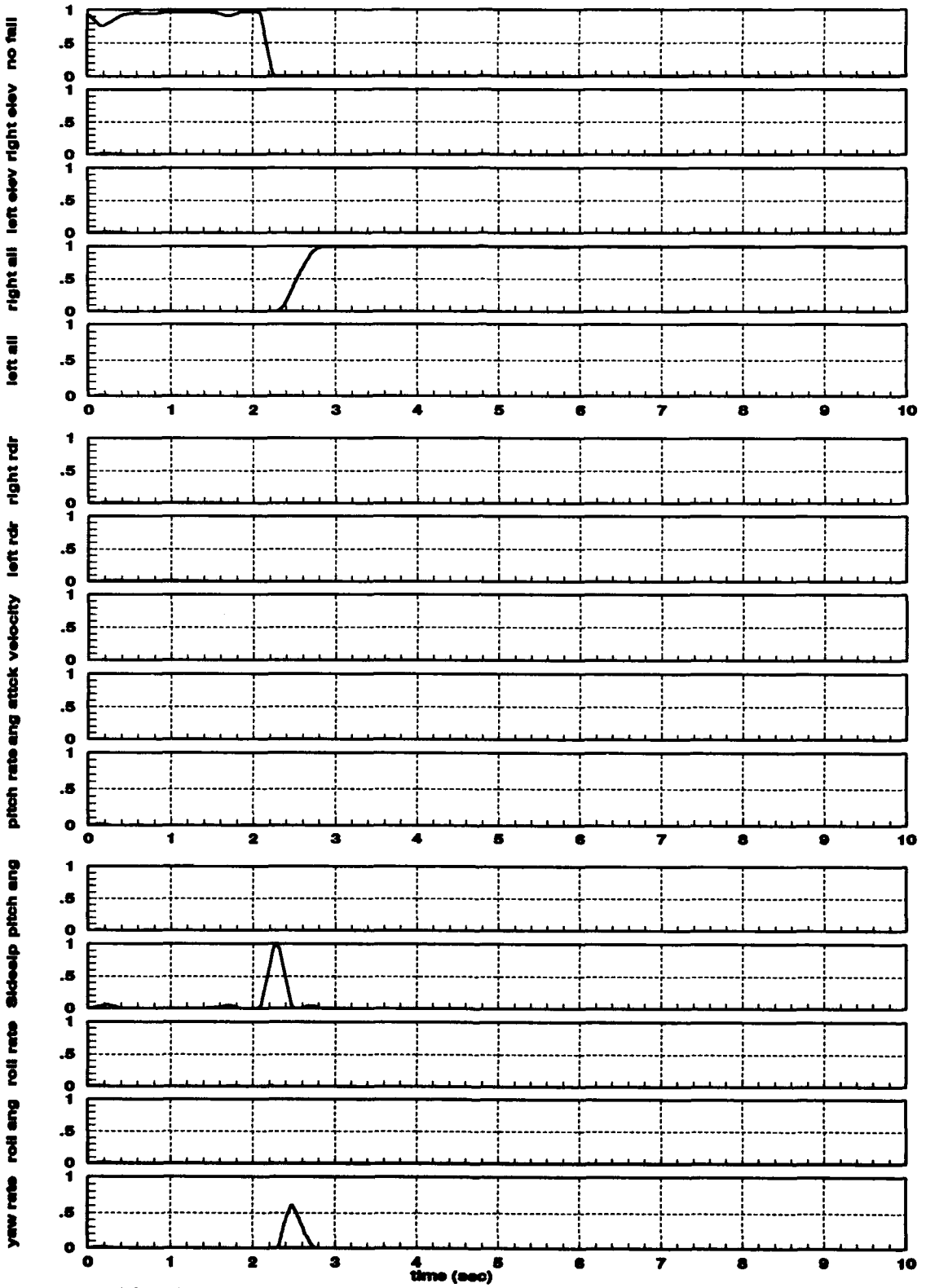


Figure 16. Right Aileron Failure, Design #2, Dot = -0.5, PWINSIZ = 10, NPROP = 1

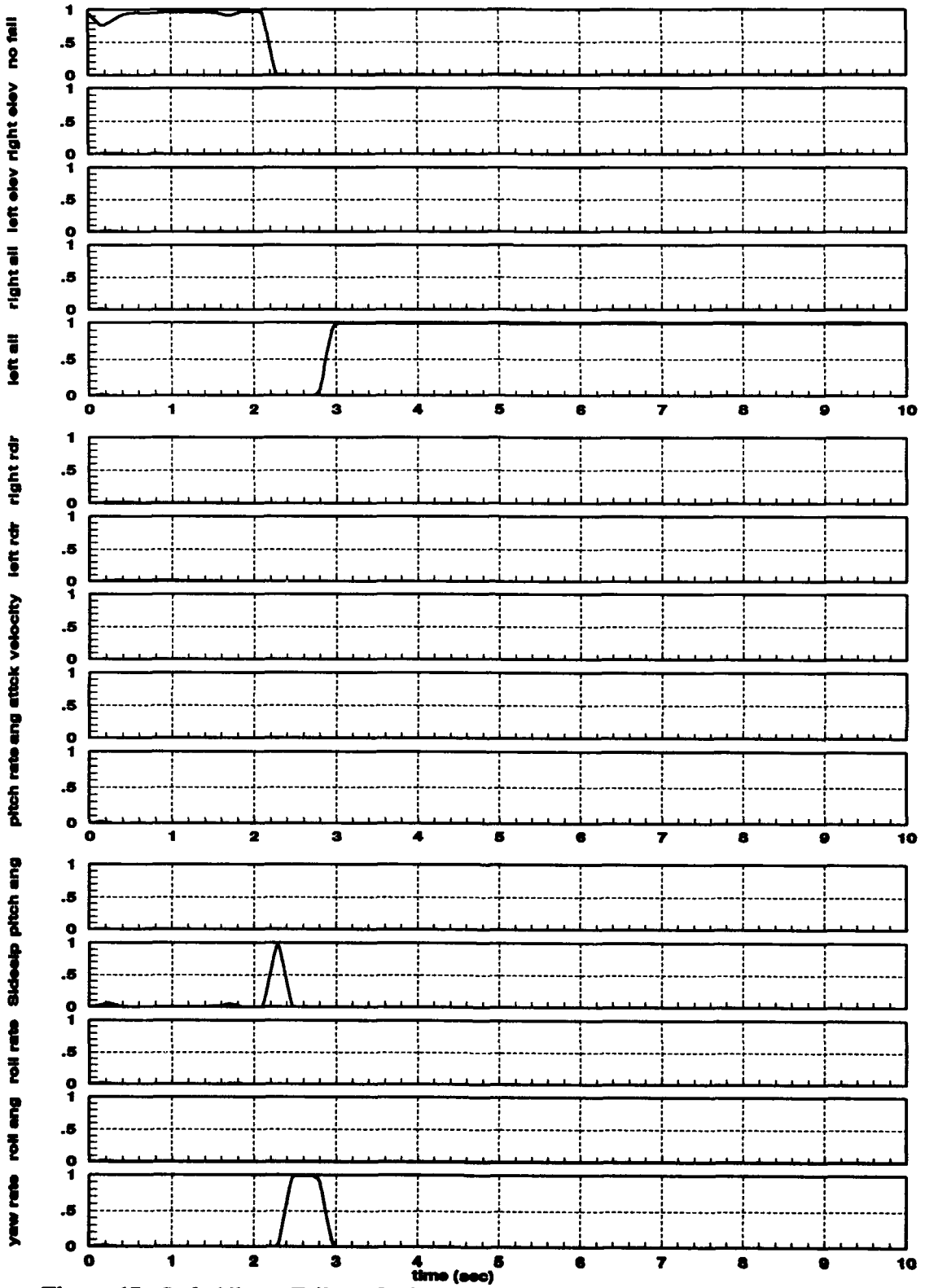


Figure 17. Left Aileron Failure, Design #2, Dot = -0.5, PWINSIZ = 10, NPROP = 1

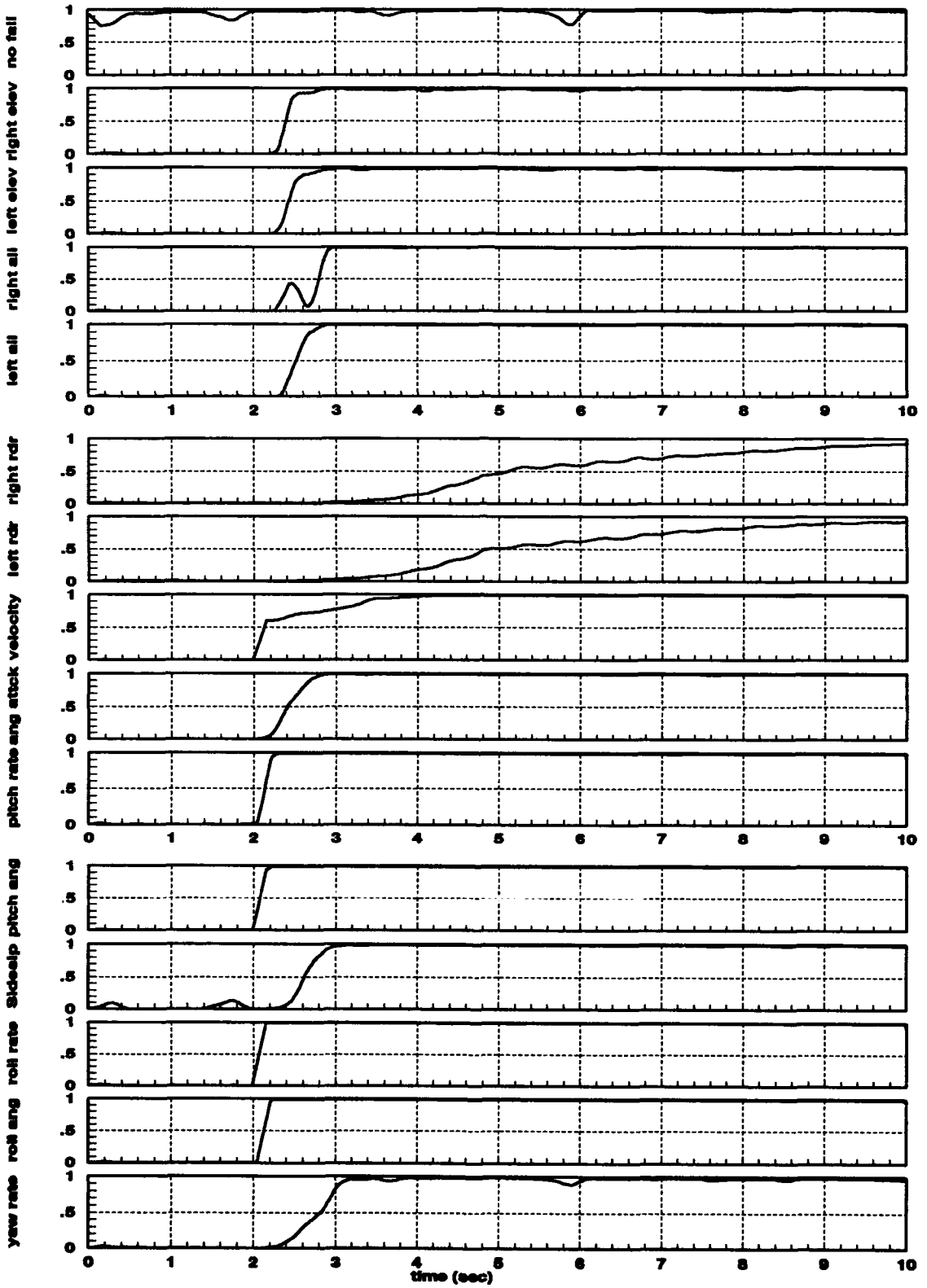


Figure 18. Design #3, Dot = -0.5, PWINSIZ = 10, NPROP = 1

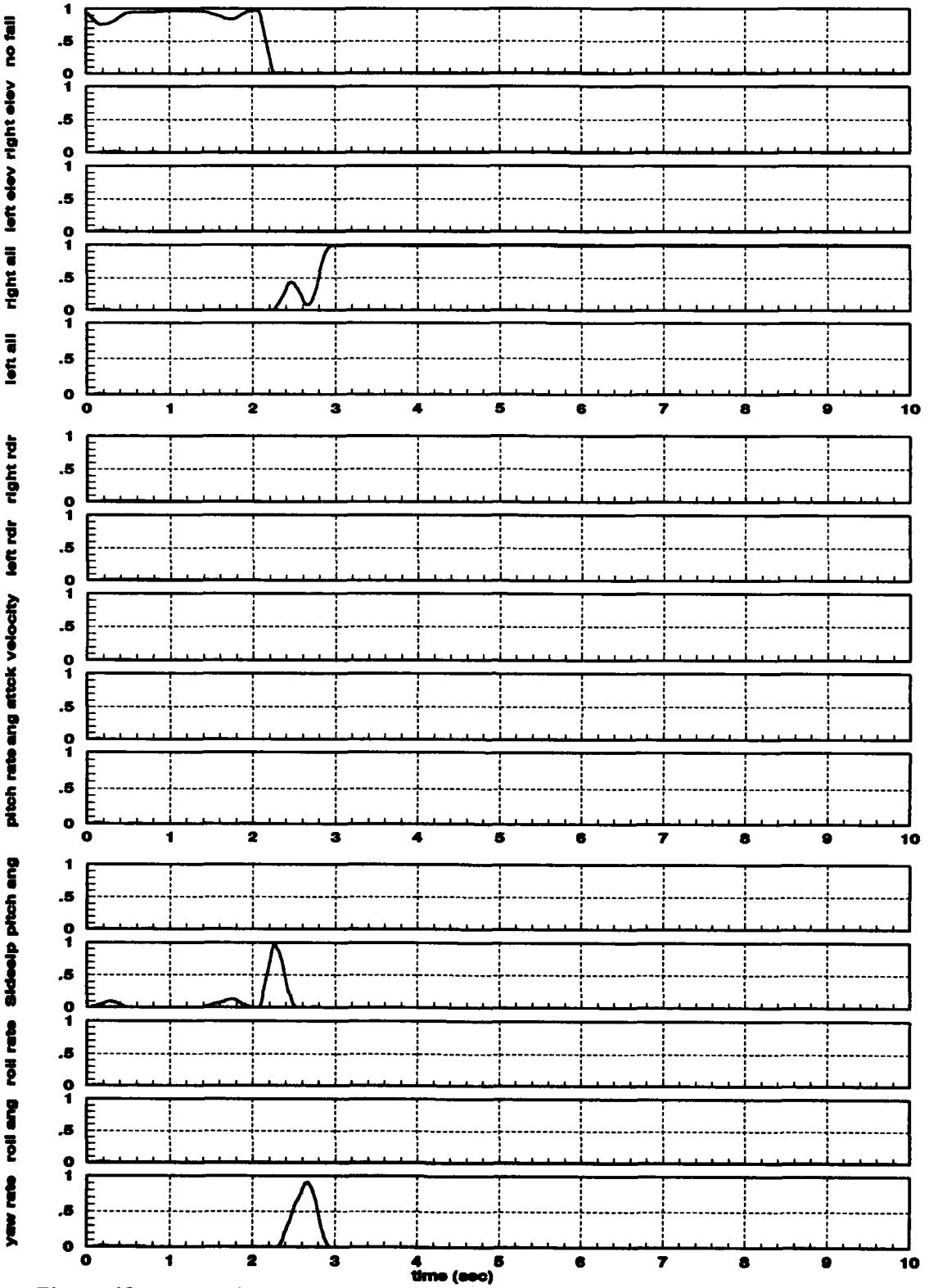


Figure 19. Right Aileron Failure, Design #3, Dot = -0.5, PWINSIZ = 10, NPROP = 1

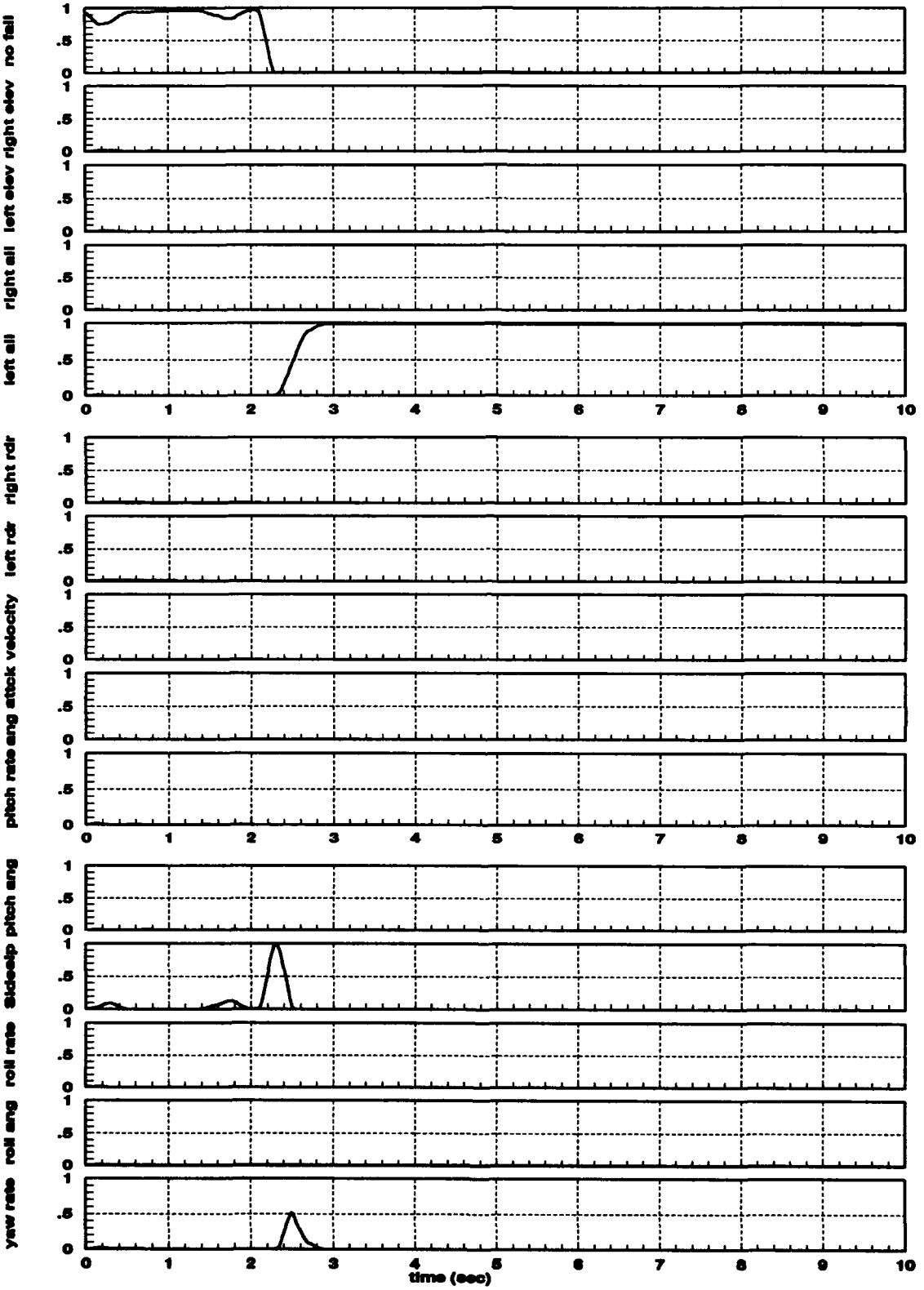


Figure 20. Left Aileron Failure, Design #3, Dot = -0.5, PWINSIZ = 10, NPROP = 1

4.3.2 Exponential Penalty Increase

Previously we noted that the rudder failure identification required a substantially longer time to converge than all other failure conditions. To decrease this convergence time, we noted that the - ½ in the $\{\cdot\}$ term in the equation for the conditional density function,

$$f_{z(t_i)|a,z(t_{i-1})}(z_i|a_k,Z_{i-1}) = \beta \exp\{\cdot\}$$

$$\beta = \frac{1}{(2\pi)^{m/2}|A_k|^{1/2}} \tag{9}$$

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

can be viewed as a penalty for having a larger than expected residual. By increasing this term, the same residual will produce higher penalties and cause the MMAE to change its probabilities faster when a failure occurs, thus acting as a decision convergence gain.

We experimented with values of -1 and -2 to replace -½ in the "Dot" term of (9). Using this strategy with Design #1, we found that the convergence time (when the probability was above the 50% probability threshold) for a rudder failure decreased from 3.2 seconds for a "Dot" term of -½ (shown in Figure 9), to 1.8 seconds for a "Dot" term of -1 (shown in Figure 21), and finally to 0.8 seconds for a "Dot" term of -2 (shown in Figure 22). We found that the fastest convergence times were obtained with Design #2 and a "Dot" term of -1, as shown in Figure 23. However, this increase in the "Dot" term also produced much larger fluctuations in the probabilities. The performance for a *single* Monte Carlo run is shown in Figure 24, which indicates that false alarms would sporadically occur during actual flight testing, which is considered to be unacceptable performance. To avoid these false alarms, we used Design #1 with a "Dot" term of -1 (Figure 21). Note that, by using Design #1, the aileron failures take up to one second longer than they do for Design #2, but the advantage is the lack of false

alarms. For this design the convergence to the correct failure condition occurs in less than 2 seconds for all the failure conditions, without any false alarms once convergence is attained. Other values of the "Dot" term and the sensor noise strengths would probably improve the performance even more, but time constraints prevented further experimentation.

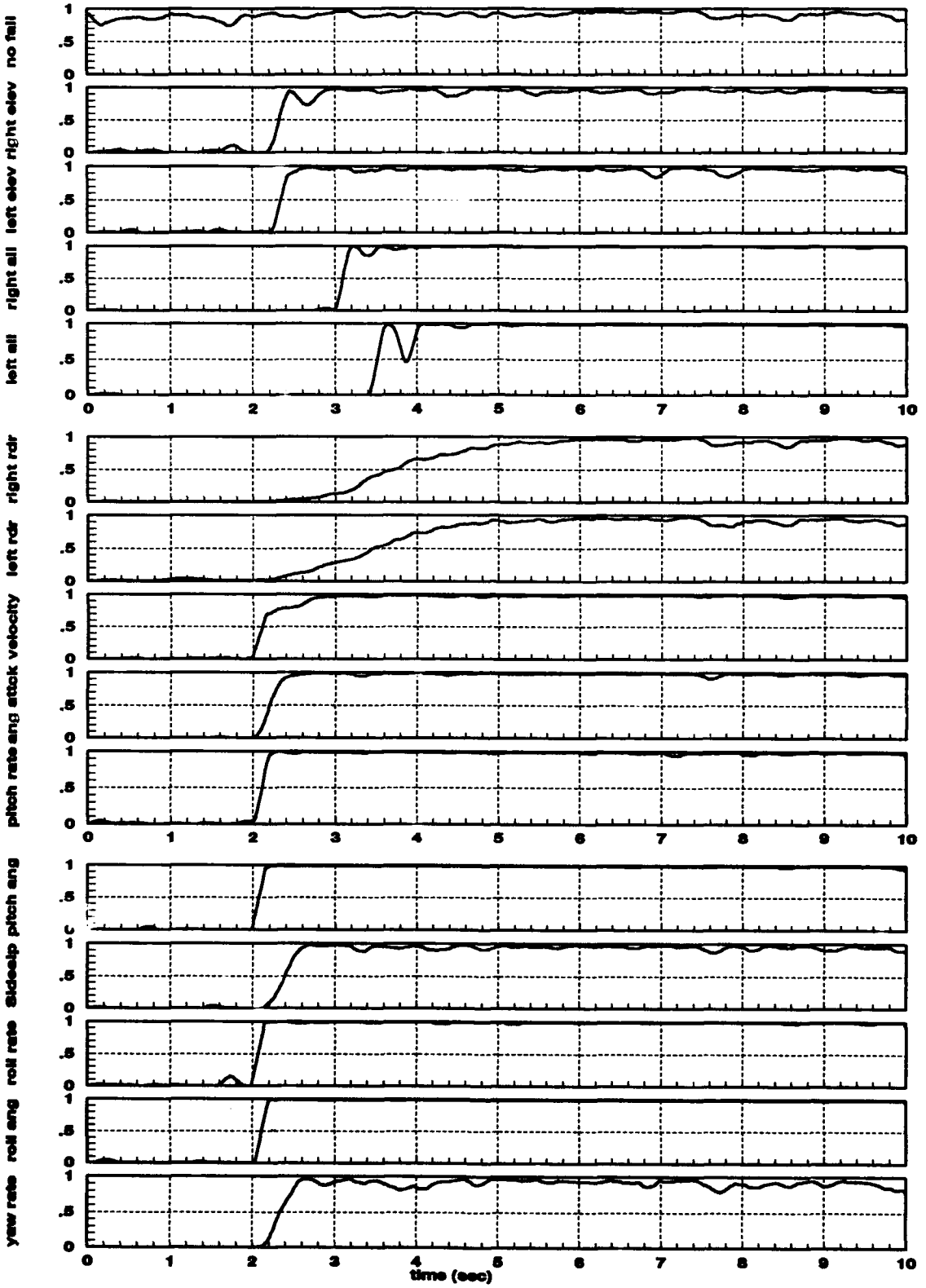


Figure 21. Design #1, Dot = -1, PWINSIZ = 10, NPROP = 1

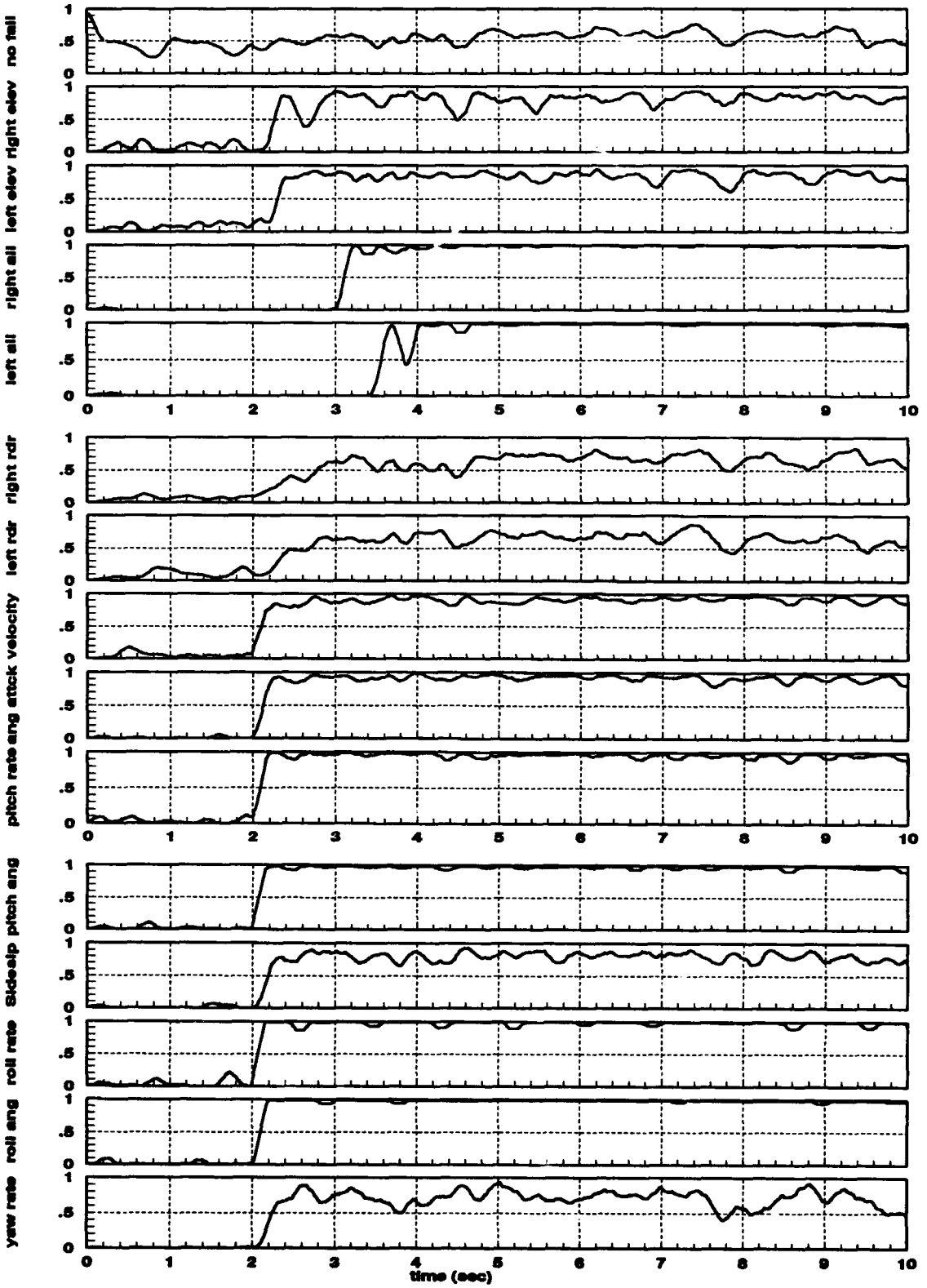


Figure 22. Design #1, Dot = -2, PWINSIZ = 10, NPROP = 1

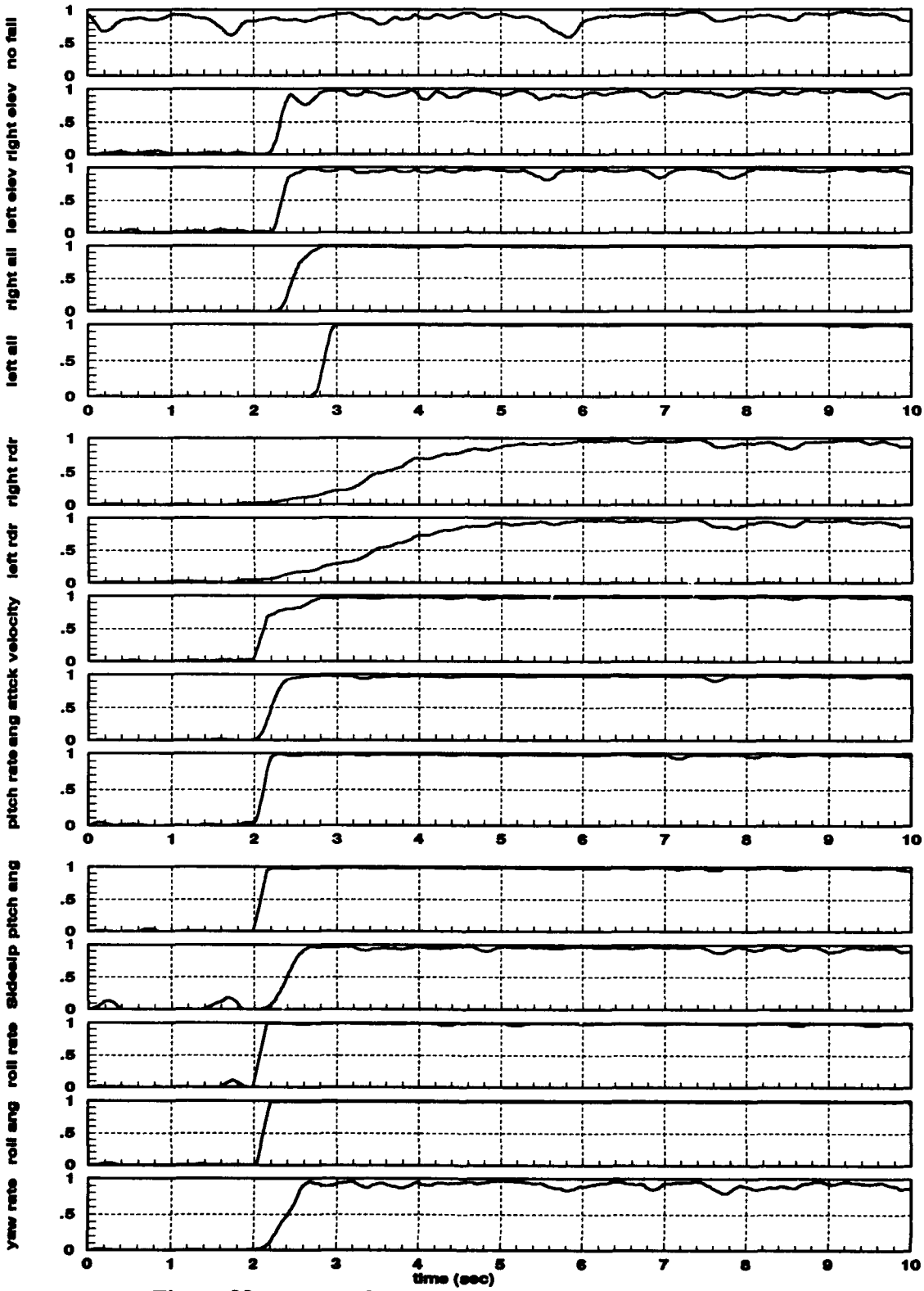


Figure 23. Design #2, Dot = -1, PWINSIZ = 10, NPROP = 1

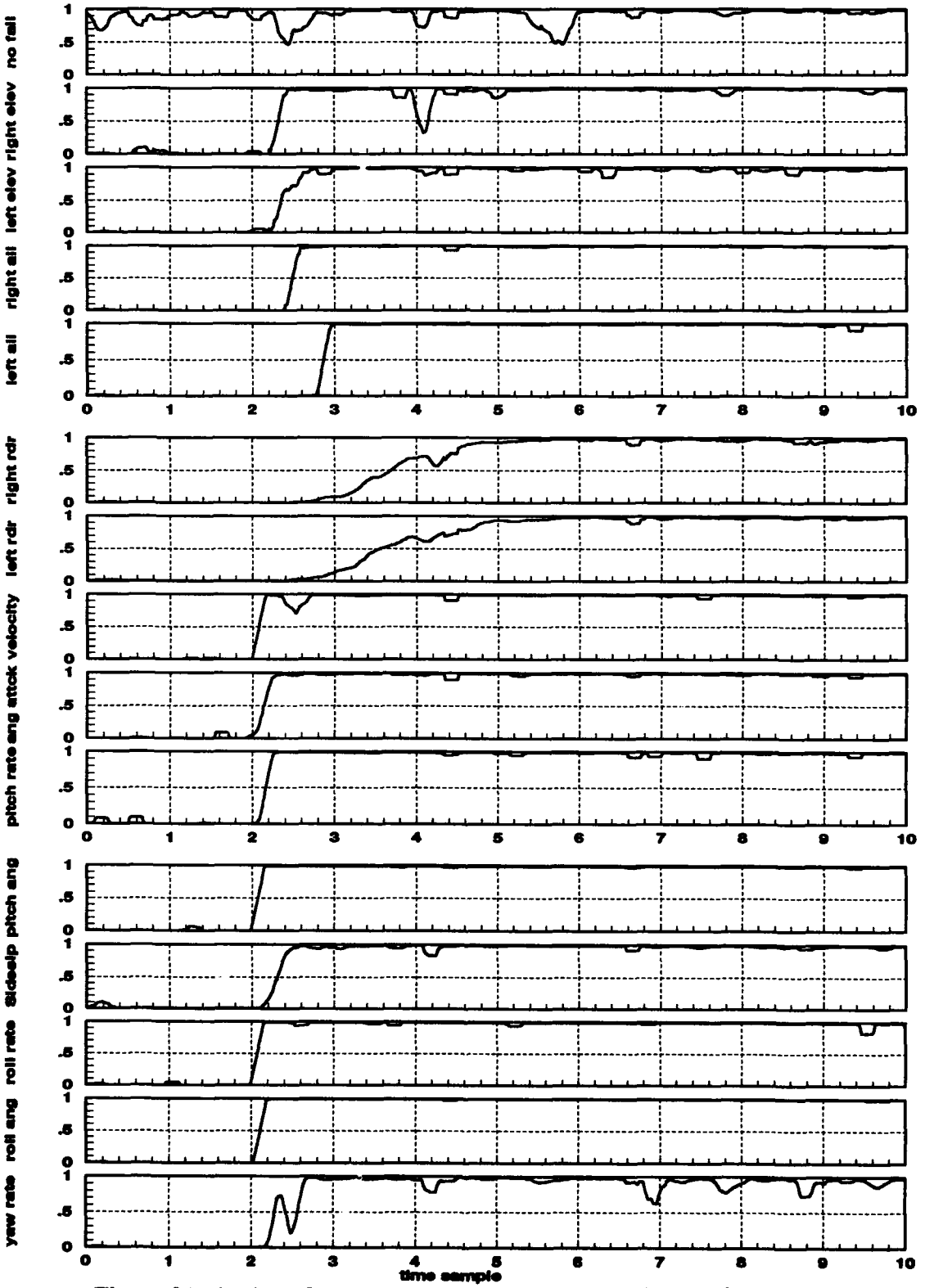


Figure 24. Design #2, Dot = -1, PWINSIZ = 10, NPROP = 1, Single Run

4.3.3 Decreased Probability Smoothing

We experimented with decreasing the size of the data window over which the probabilities are smoothed (PWINSIZ) as another possible means of decreasing the convergence time. Figure 25 shows the results of decreasing the window size from 10 to 5. Note that the performance, shown in Figure 25, does not improve significantly over the performance shown in Figure 21, while the fluctuations in the probabilities increase, which indicates an increase in false alarms would occur during flight testing. Decreasing the window size increases the false alarms because high frequency fluctuations in the probabilities are no longer smoothed over several time periods. Since decreasing the smoothing does not improve the performance significantly, but it does increase the number of false alarms, we retained the original data window size (PWINSIZ = 10).

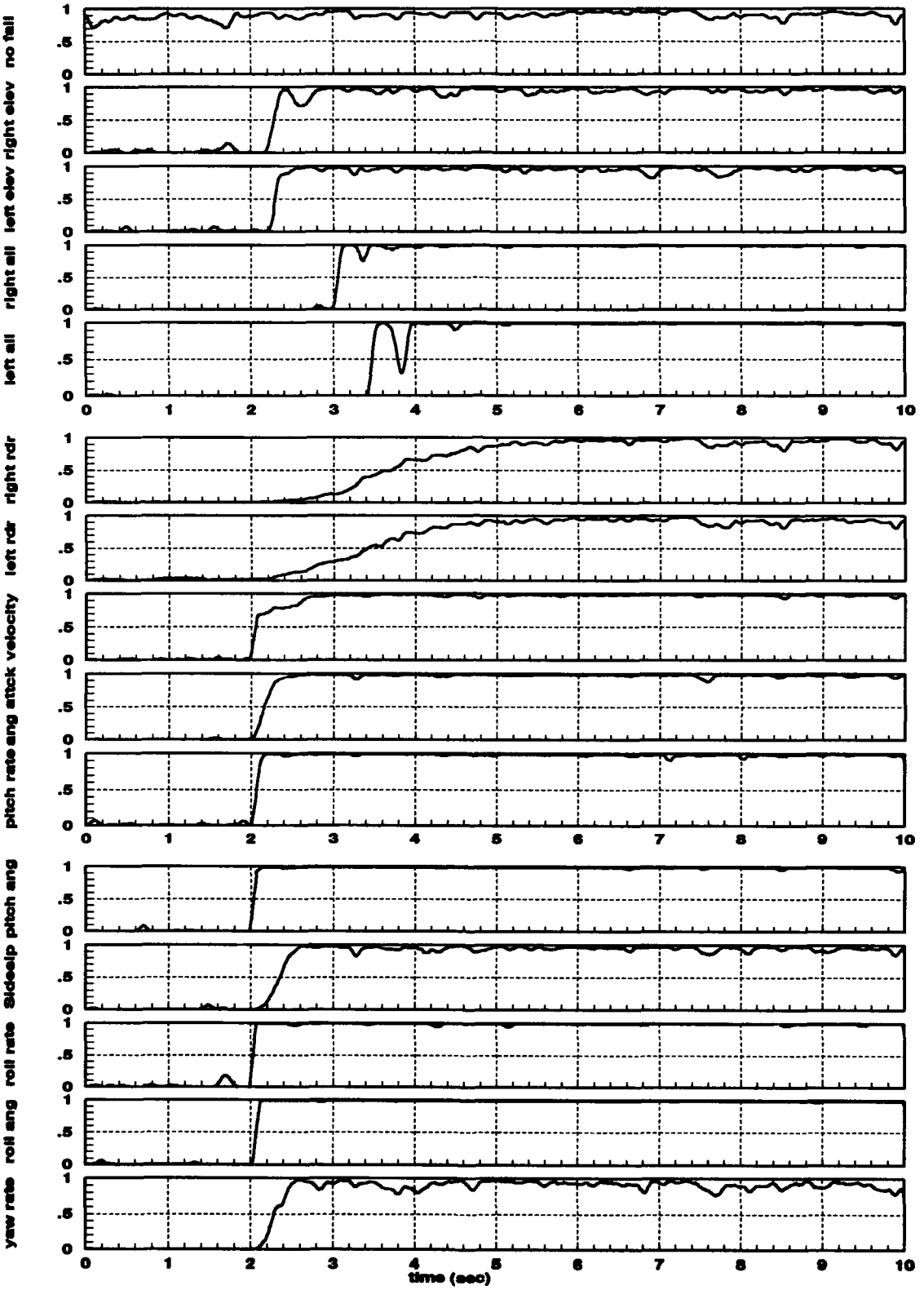


Figure 25. Design #1, Dot = -1, PWINSIZ = 5, NPROP = 1

4.3.4 Enlarged Residuals

Another method of possibly decreasing the decision convergence time would be to propagate the Kalman filter state estimates, without updating, for a few sample times while still monitoring the residuals. This allows the residuals in all the filters to grow much larger since the usual measurement updates are no longer correcting the state estimates toward the actual measurements, thereby masking the impact of an incorrect hypothesis. The results for propagating two sample periods (NPROP = 2) before updating again are shown in Figure 26. These results show no significant improvement in the MMAE performance using this strategy, and a slight increase in the fluctuations of the probabilities.

Figure 27 shows the results of propagating 5 sample periods before updating. These results are quite poor, which could be due to a poor input for identifying these failures. This method essentially changes the measurement sampling time because the measurements are used only when the estimates are updated, which is every NPROP sample periods. A new sampling time has a profound impact on the system modes, thus a new input might be needed to excite the failure modes.

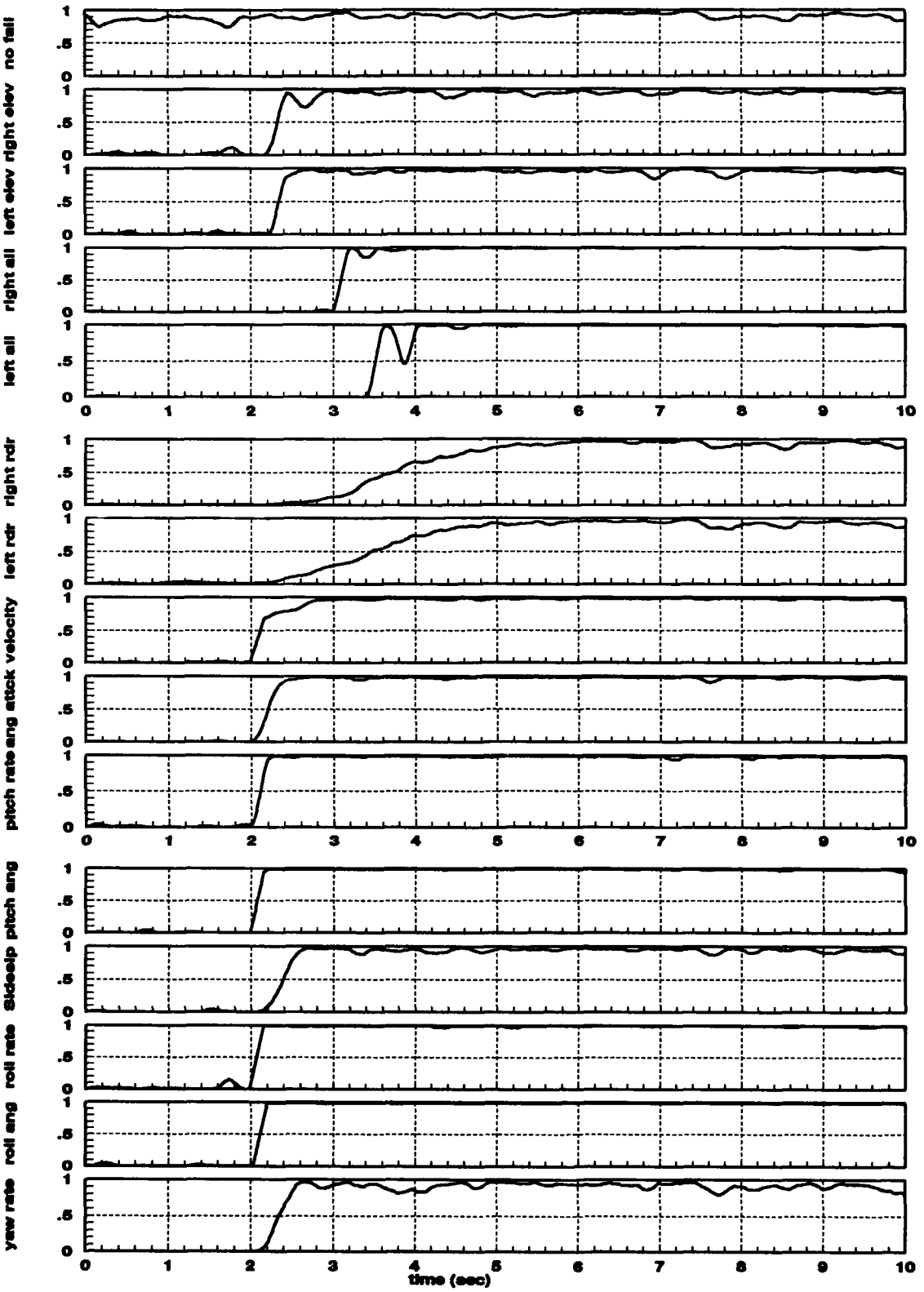


Figure 26. Design #1, Dot = -1, PWINSIZ = 10, NPROP = 2

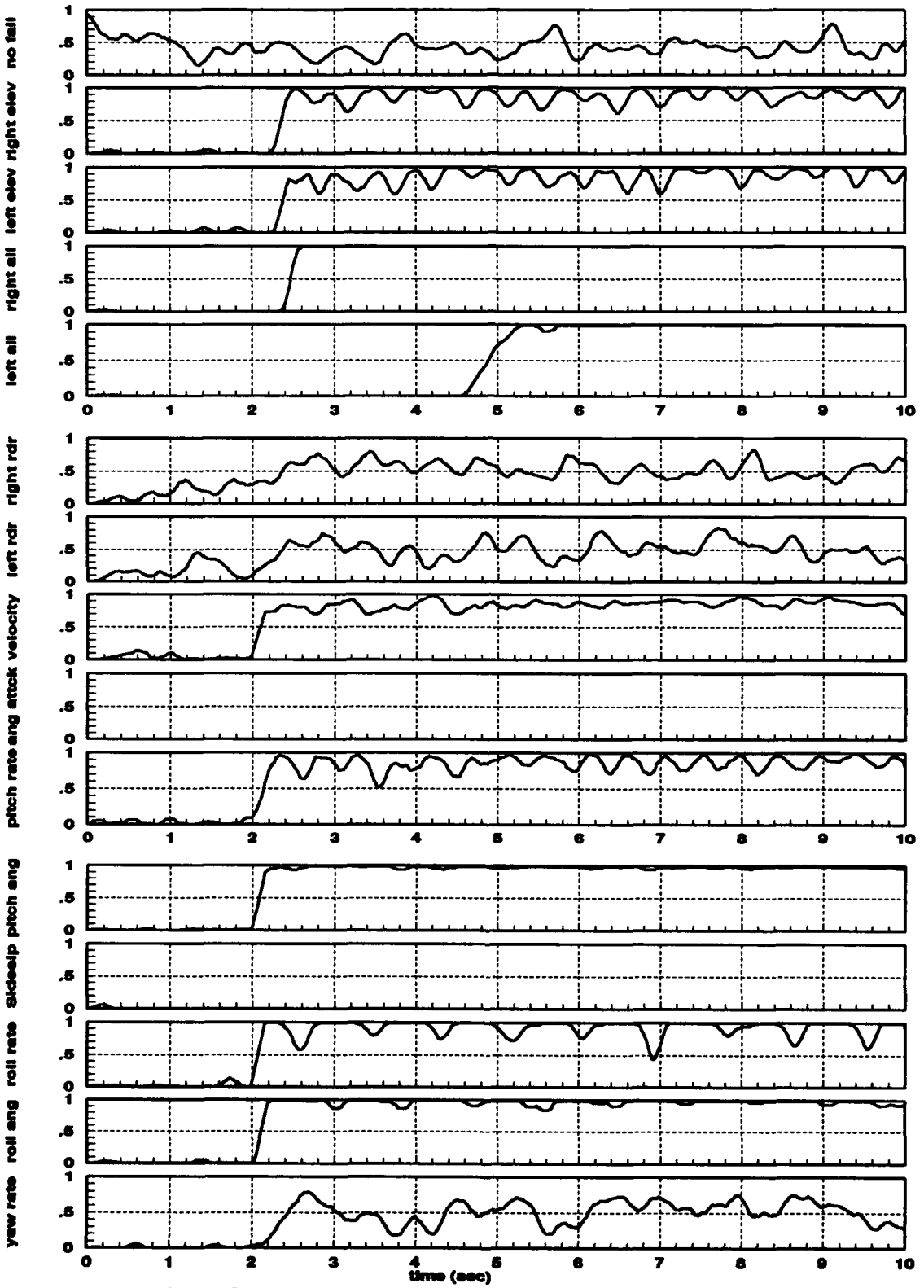


Figure 27. Design #1, Dot = -1, PWINSIZ = 10, NPROP = 5

4.3.5 Initialization Effects

The probabilities in the MMAE need to be initialized because they are used to calculate the current hypothesis probability. Of all the failure hypotheses, the fully functional hypothesis is most likely to be the correct initial hypothesis for a well maintained aircraft. Therefore, the MMAE should be initialized such that the Kalman filter with the fully functional hypothesis has very high probability and the other Kalman filters equally divide the remaining probability. In previous simulations, we initialized the MMAE such that the fully functional hypothesis Kalman filter has a probability of 0.75 and the other Kalman filters have a probability of 0.0179, so that the total probability sums to one.

To investigate the performance of an MMAE that has either been initialized poorly or has momentarily converged on the wrong hypothesis, several simulations were performed with the MMAE intentionally initialized incorrectly. For each of the simulations, one of the Kalman filters was initialized with a probability of 0.95 and the others with a probability of 0.0036. Several simulations, one for each failure condition, were run to observe the amount of time it took for the MMAE to converge to the correct failure condition. Figure 28 shows the performance of the MMAE when each filter is initialized incorrectly and a no failure simulation is performed. Note that this plot differs significantly from previous plots. In Figure 28 the plots show only the probabilities for the no-fail filter (which is the correct failure hypothesis). The filters that are listed along the y-axis are the filters that are initialized to 0.95. For instance, the right elevator plot shows the no-failure probability when the right elevator filter hypothesis is initialized to 0.95. The MMAE converges to the no-failure hypothesis usually within one second, except for the rudder failure hypotheses where it takes about two seconds. The same behavior was found for other failure simulations.

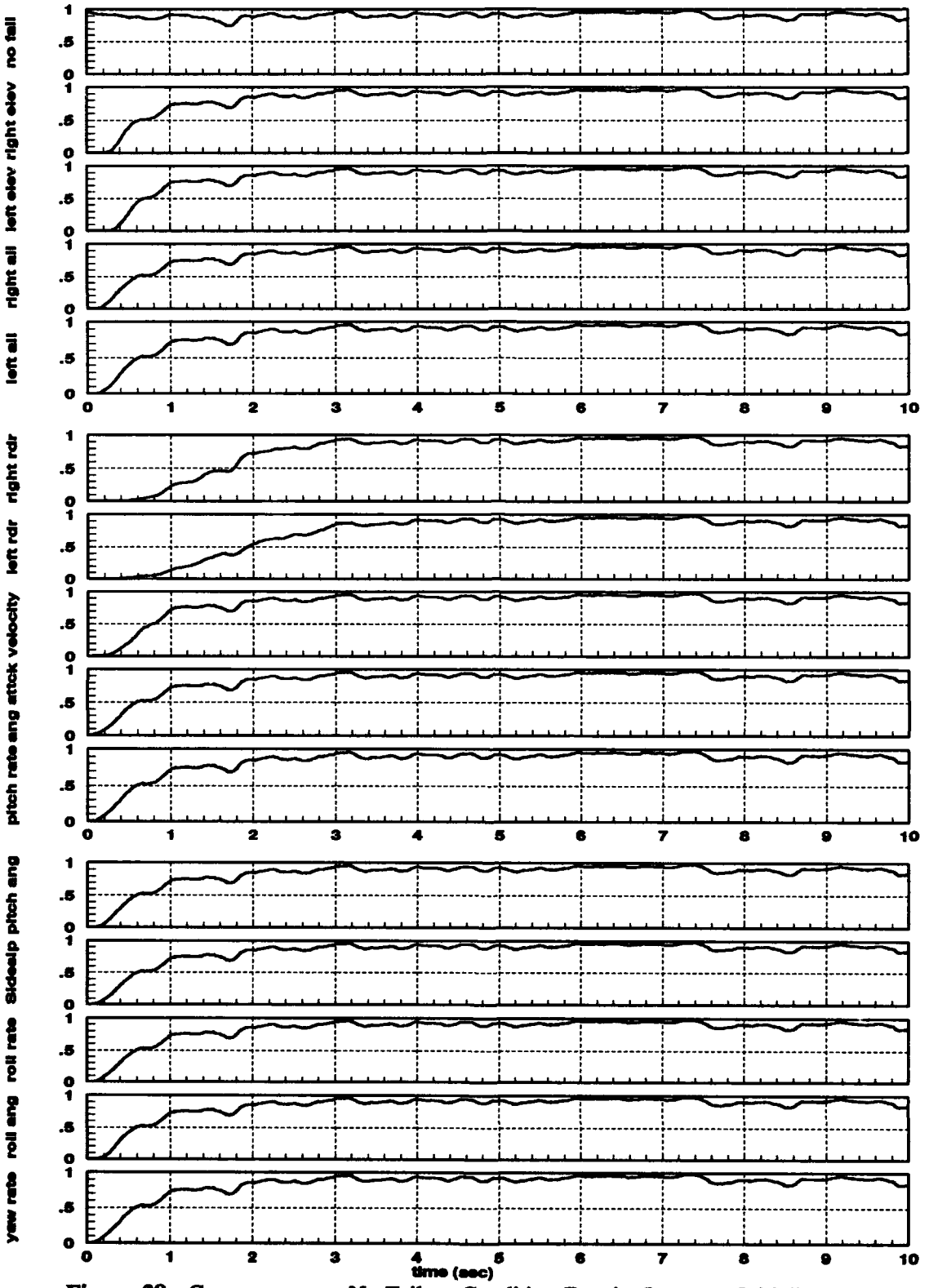


Figure 28. Convergence to No Failure Condition Despite Incorrect Initialization.

4.4 Unmodeled Effects

We investigated the failure detection performance of the MMAE against unmodeled effects by simulating its performance at the corners of the flight envelope shown in Figure 29, with the MMAE design using the nominal conditions (shown by the circled dot in Figure 29).

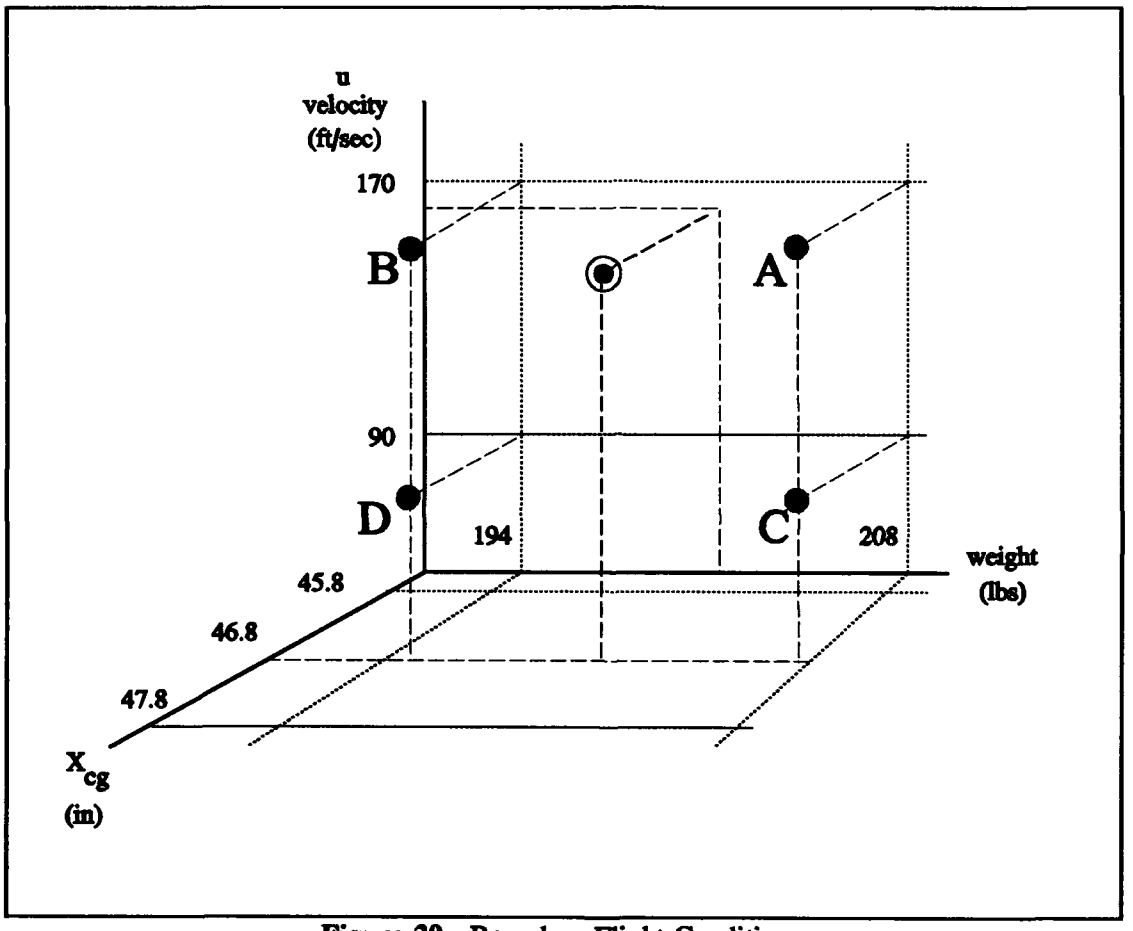


Figure 29. Boundary Flight Conditions.

At point A (high velocity, high weight) and point B (high velocity, low weight), we found that the MMAE performance (Figure 30) was almost exactly the same. Figure 30 shows that the MMAE still converges to the correct failure condition in less than 1.5 seconds for all failures

conditions, which is better than at the design point (2 seconds). However, there are significantly higher fluctuations in the probabilities, which indicate that quite a few momentary false alarms will occur during actual flight testing. This is particularly true for the aileron-failure Kalman filters, which show even higher fluctuations than any of the other filters. This is commensurate with our earlier observations that these filters have the most trouble converging to the correct hypothesis.

We also looked at the MMAE performance at the low end (with respect to velocity and dynamic pressure) of the flight envelope. Figure 31 shows extremely poor MMAE performance at point C, and comparable performance was found for point D. The performance in Figure 31 is totally unacceptable; therefore, some modification of the MMAE, such as gain scheduling, is required to provide the desired performance at these points. Such gain scheduling would not be difficult to implement and it is highly recommended for a final system configuration.

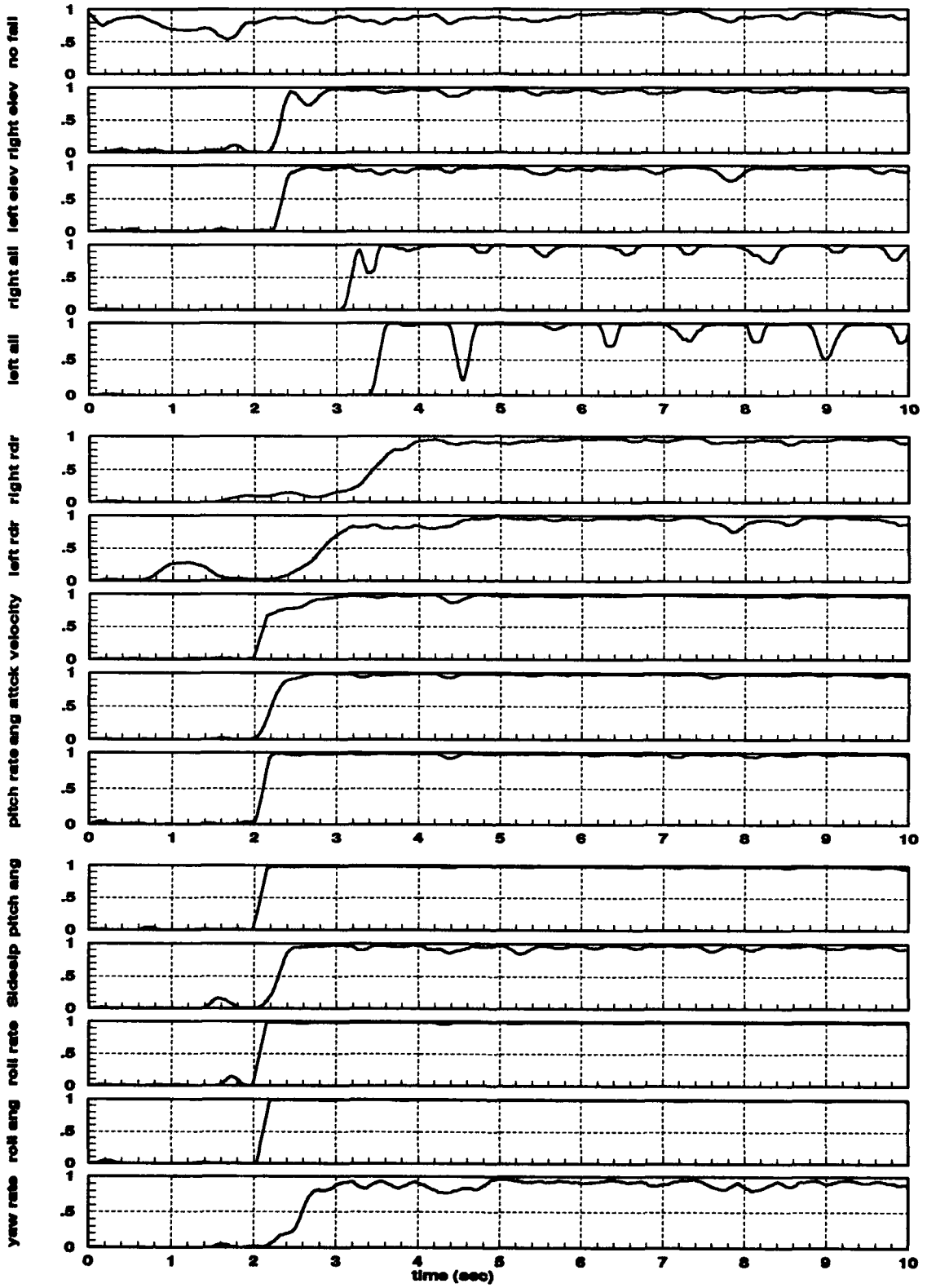


Figure 30. MMAE Performance at Flight Condition Point A.

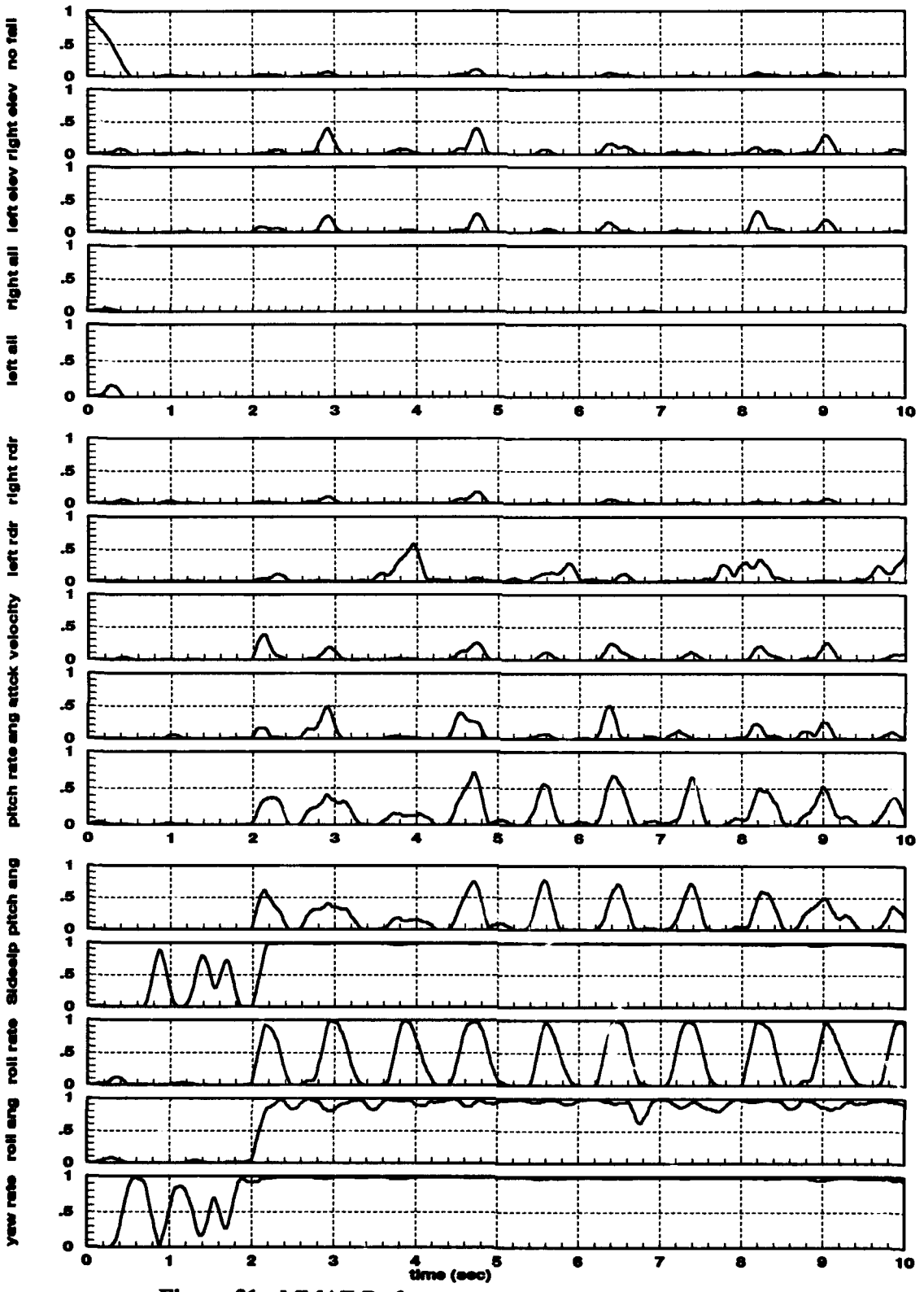


Figure 31. MMAE Performance at Flight Condition Point C.

Chapter V: Conclusions

5.1 Chapter Overview

In this chapter we present the conclusions that can be drawn from our findings that were presented in Chapter IV. The conclusions are organized to correspond with the research questions that were presented in Section 1.5. In Section 5.2 we look at the ability of the MMAE to converge to the correct failure condition. In Section 5.3 we evaluate the various methods that were tried to improve the rate of convergence to the correct failure condition. In Section 5.4 we draw conclusions about some aspects of the robustness of the MMAE based on its performance at the boundaries of the flight envelope. Finally, we close with our recommendations for future research in Section 5.4.

5.2 Decision Convergence

We found that the MMAE did converge to the correct failure condition if a command input (either a purposeful command or a dither to enhance detection) is provided that excites *all* the failure modes correctly. We found such a dither by using the eigenvalues of the continuous-time plant model and then experimenting until good results were achieved. This dither had different characteristics (magnitude and frequency) in each axis, and had to be a differential input (when right actuator is positive, the left actuator is negative) that was slightly shifted in phase to provide enough excitement for the MMAE to detect a sensor failure. We also found that there were thr[illegible]s in magnitude, frequency, and differential phase that had to be exceeded for the MMAE to converge. Once the thresholds were crossed, additional magnitude or phase did not enhance convergence. The frequency of the dither probably has to

be within a certain bandwidth centered around the natural frequencies. This points to the possibility of using spectral analysis to design optimal inputs.

5.3 Decision Switching

We found that the decision convergence rates differed dramatically depending on the failure condition, and various methods did improve the convergence rate, particularly for the failure conditions that suffered from slow detection rates. All of these methods involved trading an increase in false alarms for an increase in convergence performance. The methods that we tried were tuning of the Kalman filters, increasing the exponential penalty for larger than expected residuals, decreasing the data smoothing, and propagating without updating the state estimates to enlarge the residuals, particularly the ones associated with an incorrect hypothesis.

We made a cursory examination of the eigenvalues of the aircraft dynamics matrix, F in (11), and the Kalman filter-computed residual covariance matrix, A_k in (8), to determine why our detection rates were longer than those of previous studies [28, 29, 33, 34, 35, 38, 41, 42]. We found that the Kalman filters had to be propagated up to 100 time samples before steady state was reached, when computing the steady state Kalman filter gains. The aircraft eigenvalues were widely separated because we were using a model that included all the aircraft modes. This initially caused some problems with the software integration package, but by using a lower order integration routine and subintervals for integration, we circumvented this problem. An aircraft model based on a short period approximation might have prevented this problem, and it would have reduced the number of aircraft states, along with possibly improving the convergence rate. It would seem that the difference in convergence rates is most likely due to the open-loop model of the aircraft that was used for this research.

We tried tuning the Kalman filter models to decrease the ambiguities that were observed when an aileron failure was modeled. This method was quite effective in decreasing the confusion, as presented in Section 4.3.1. We found that an aileron failure was identified in about half the time it took to converge prior to the filters being tuned. Some experimentation was required to attain this performance. If the values in the measurement noise covariance matrix were increased too far, fluctuations in the probabilities occurred and the false alarm rate would increase. The fluctuations caused the correct hypothesis filter probabilities to dip below the threshold, but the other filters rarely jumped above the threshold. This behavior either caused a misidentification of "no failures," or simply confusion since none of the hypotheses probabilities were above the threshold. We could compensate for this increase in false alarms by lowering the decision threshold, as long as the fluctuations were not too severe. Also, this method worked well only up to a certain point, after which the performance did not improve. The precomputed values for the measurement noise variances were used by the Kalman filters to determine how large the residuals should be, so larger values for the measurement noise variances would cause the filters to expect larger residuals, which would mask the sensor measurement residuals and make it more difficult to detect a failure.

We found that a rudder failure took substantially longer to converge than all other failure conditions. This could be due to the shorter moment arm for the rudders when compared to other flight control surfaces. A rudder input will take longer to produce a substantial output due to this shorter moment arm, thus the Kalman filter residuals will take longer to build and the MMAE will take longer to detect this failure. Some improvement was attained by increasing the "Dot" term in (9), as shown in Section 4.3.2. Increasing the "Dot" term had the effect of turning up the gain on the decision convergence, since it not only

caused the MMAE to converge faster, but it increased the "noise" (fluctuations) in the probabilities which forced us to set the probability decision threshold to 50% to avoid false alarms and ambiguous failure detections. This method works well as long as the false alarm rate is tolerable.

We tried to decrease the convergence time even further by decreasing the data smoothing. The probabilities were smoothed over 10 time periods, which introduced a delay in the decision convergence. We found that this delay was minimal since it would be at most 10 sample periods (about 1/6 th of a second), which is quite small compared to the decision convergence times on the order of one second. As this smoothing was decreased, the false alarm rate increased, so we kept the smoothing set at 10 sample periods.

Lastly, we tried propagating the Kalman filter state estimates a few sample periods without measurement updating, which would cause the residuals to grow larger than when the updates do occur, which would create even larger residuals in filters based on incorrect hypotheses about the failure status. This technique did not improve the performance of the MMAE very much before the false alarm rate grew too large. This method essentially changes the measurement sampling rate, which in turn changes the system dynamics as seen over a single measurement sample period, which then requires a new input to excite the failure modes optimally. This method might have worked quite well if a new input could have been developed, but time did not allow a full exploration of this approach.

5.4 Unmodeled Effects

We found that the MMAE was robust enough to provide fairly good performance at the upper boundaries of the flight envelope, but the performance at the lower boundaries was inadequate. The robustness of the MMAE would be improved by discretizing the flight

envelope, creating MMAE designs at those points, and gain scheduling the MMAE for the current flight condition. The MMAE blended state estimate would provide the best estimate of the current flight condition and, by using this estimate with gain scheduling, feedback is accomplished which would increase the robustness of the MMAE. A similar technique would involve accomplishing a sensitivity study to find the parameters to which the MMAE gains are most sensitive, and gain scheduling using those parameters. We found that the LAMBDA model was most sensitive to a change in velocity, but we were not able to determine the effect of these changes on the MMAE.

5.5 Recommendations for Further Investigation

There are many areas that require further investigation. The most significant deficiency in this MMAE is the lack of cross axis coupling terms in the LAMBDA model. If these terms were defined in future research, then a more reasonable command input could be designed since the effects of a failure in one axis could also be found in the other axis. Another area that would be quite promising is the use of reduced order models. If a model was developed using a short period approximation for the LAMBDA, then a MMAE design using this model would reduce the computational loading and might have even faster convergence rates. Various gain scheduling techniques need to be considered to enhance the robustness of the MMAE across the whole flight envelope. If detection of multiple failures on the LAMBDA is desired, then a hierarchical structure of filter designs needs to be developed, as in previous research into multiple failure detection [29, 33, 34, 35, 41, 42]. The MMAE capabilities of detecting partial failures on the LAMBDA needs to studied as in previous research for other aircraft [29, 33, 34, 35, 41, 42]. A much more general study, which would be applicable to all MMAE designs, would be to determine the relationship, if any, between

the eigenvalues of the plant and/or the filters and the convergence rates. If a relationship exists and is predictable, then the MMAE performance could be determined prior to the actual design and simulation. Lastly, a study on the design of optimal inputs for enhancing failure identification and/or detection is sorely needed, particularly if a different optimal input is required at various points of the flight envelope.

Bibliography

1. Athans, M. *et al.* "The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part I: Equilibrium Flight," *IEEE Transactions on Automatic Control*, AC-22(5): 768 - 780, (October 1977).

2. Blakelock, John H. *Automatic Control of Aircraft and Missiles.* New York: John Wiley & Sons, Inc., 1991

3. Chang, C. B., and M. Athans. "Hypothesis Testing and State Estimation for Discrete Systems with Finite-Valued Switching Parameters," MIT Electronic Systems Laboratory publication ESL-P-758, Lexington, MA, (June 1977).

4. Chang, C. B. "State Estimation for Discrete Systems with Switching Parameters," *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(4): 418 - 425, (May 1978).

5. Chang, C. B., and J. A. Tabaczynski. "Application of State Estimation to Target Tracking," *IEEE Transactions on Automatic Control*, AC-29(2): 98 - 109, (February 1984).

6. Dusgapta, S., and L. C. Westphal. "Convergence of Partitioned Adaptive Filters for Systems with Unknown Biases," *IEEE Transactions on Automatic Control*, AC-28(5): 614 - 615, (May 1983).

7. Eckert, S. J., and K. A. Loparo. "An Application of Nonlinear Filtering to Instrument Failure Detection in a Pressurized Water Reactor," *Nuclear Technology*, 74: 139 - 150, (August 1986).

8. Filios, P. G. *Moving-Bank Multiple Model Adaptive Algorithms Applied to Flexible Spacecraft Control.* MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.

9. Flynn, 2Lt Patrick M. *Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker.* MS Thesis, AFIT/GE/EE/81D-21. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981.

10. Gholson, N. H., and R. L. Moose, "Maneuvering Target Tracking Using Adaptive State Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, AES-13(3): 310 - 317, (May 1977).

11. Greene, C. S., and A. S. Willsky. "An Analysis of the Multiple Model Adaptive Control Algorithm," *Proceedings of the IEEE Conference on Decision and Control*, Albuquerque, New Mexico: pp. 1142 - 1145 (Dec. 1980).

12. Gustafson, D. E., A. S. Willsky, and J.-Y. Wang. "Final Report: Cardiac Arrhythmia Detection and Classification Through Signal Analysis," The Charles Stark Draper Laboratory, Report Number R-920, Cambridge, MA, July 1975.

13. Hawkes, R. M., and J. B. Moore. "Performance Bounds for Adaptive Estimation," *Proceedings of the IEEE*, 64: 1143 - 1150, (August 1976).

14. Hentz, K. P. *Feasibility Analysis of Moving Bank Muliple Model Adaptive Estimation and Control Algorithms*. MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984.

15. Hostetler, L. D., and R. D. Andreas, "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," *IEEE Transactions on Automatic Control*, AC-28(3): 315 - 323, (March 1983).

16. Karnick, D. A. *Moving-Bank Muliple Model Adaptive Estimation Applied to Flexible Space-Structure Control*. MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.

17. Karnick, D. A., and P. S. Maybeck. "Moving Bank Muliple Model Adaptive Estimation Applied to Flexible Spacestructure Control," *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, California: pp 1249 - 1257, (December 1987).

18. Korn, J., and L. Beean, "Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation," Technical Report TR-152, Alphatech, Inc., Burlington, Mass., June 1983.

19. Lashlee, R. W., Jr. *Moving-Bank Muliple Model Adaptive Estimation Applied to Flexible Space-Structure Control*. MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

20. Lashlee, R. W., Jr., and P. S. Maybeck. "Spacestructure Control Using Moving Bank Multiple Model Adaptive Estimation," *Proceedings of the IEEE Conference on Decision and Control*, San Antonio, Texas, (December 1988).

21. Lainiotis, D. G. "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation," *Proceedings of the IEEE*, 64: 1126 - 1142, (August 1976).

22. Lainiotis, D. G. "Partitioning: A Unifying Framework for Adaptive Systems, II: Control," *Proceedings of the IEEE*, 64: 1182 - 1197, (August 1976).

23. Magill, D. T. "Optimal Adaptive Estimation of Sample Stochastic Processes," *IEEE Transactions on Automatic Control*, AC-10(4): 434 - 439, (October 1965).

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

25. Maybeck, P. S. *Stochastic Models, Estimation, and Control*, Vol. 2. New York: Academic Press, 1979.

26. Maybeck, P. S. "Moving-Bank Multiple Model Adaptive Estimation and Control Algorithms: An Evaluation," *Control and Dynamic Systems*, pp. 1 - 31, edited by C. T. Leondes, Academic Press, San Diego, California, 1989.

27. Maybeck, P. S., and K. P. Hentz, "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *AIAA Journal of Guidance, Control, and Dynamics*, Volume 10, Number 1: pp. 90 - 96, (January-February 1987).

28. Maybeck, P. S., and D. L. Pogoda, "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," *Proceedings of the IEEE Conference on Decision and Control*, Tampa, Florida: pp. 1566 - 1572, (December 1989).

29. Maybeck, P. S., and R. D. Stevens, "Reconfigurable Flight Control via Multiple Model Adaptive Control Methods," *IEEE Transactions on Aerospace and Electronic Systems*, AES-27(3): 470 - 480, (May 1991).

30. Maybeck, P. S., and R. I. Suizu, "Adaptive Tracker Field-of-View Variation Via Multiple Model Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(4): 529 - 539, (July 1985).

31. Maybeck, P. S., and W. L. Zicker, "MMAE-Based Control with Space-Time Point Process Observations," *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3): 292 - 300, (May 1985).

32. Mealy, G. L., and W. Tang, "Application of Multiple Model Estimation to a Recursive Terrain Height Correlation System," *IEEE Transactions on Automatic Control*, AC-28(3): 323 - 331, (March 1983).

33. Menke, Timothy E. *Multiple Model Adaptive Estimation Applied to the VISTA F-16 with Actuator and Sensor Failures.* MS Thesis, AFIT/GA/ENG/92J-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1992.

34. Menke, Timothy E., and P. S. Maybeck, "Multiple Model Adaptive Estimation Applied to the VISTA F-16 Flight Control System with Actuator and Sensor Failures," *Proceedings of the IEEE of the National Avioncs and Electronics Conference (NAECON)*, Dayton, Ohio: pp 441 - 448 (May 1992).

35. Menke, Timothy E., and P. S. Maybeck, "Sensor/Actuator Failure Detection in the VISTA F-16 by Multiple Model Adaptive Estimation," to be presented at the *American Control Conference*, San Francisco, California (June 1993).

36. Moose, R. L. "An Adaptive State Estimation Solution to the Maneuvering Target Problem," *IEEE Transactions on Automatic Control*, AC-20(3): 359 - 362 (June 1975).

37. Moose, R. L., H. F. Van Landingham, and D. H. McCabe, "Modeling and Estimation for Tracking Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(3): 448 - 456, (May 1979).

38. Pogoda, Capt Donald L. *Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures.* MS Thesis, AFIT/GE/ENG/88D-37. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.

39. Sims, C. S., and M. R. D'Mello, "Adaptive Deconvolution of Seismic Signals," *IEEE Transactions on Geoscience Electronics*, GE-16: 99 - 103, (April 1978).

40. Schore, M. R. *Robustness of a Moving-Bank Multiple Model Adaptive Controller for a Large Space Structure.* MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.

41. Stevens, Capt Richard D. *Characterization of a Reconfigurable Multiple Model Adaptive Controller Using A STOL F-15 Model.* MS Thesis, AFIT/GE/ENG/89D-52. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.

42. Stratton, Capt Gregory L. *Actuator and Sensor Failure Identification using a Multiple Model Adaptive Technique for the VISTA/F-16.* MS Thesis, AFIT/GE/ENG/91D-53. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991.

43. Swift, 1Lt Gerald A. *Model Identification and Control System Design for the LAMBDA Unmanned Research Vehicle.* MS Thesis, AFIT/GAE/ENY/91S-4. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1991.

44. Tenney, R. R., R. S. Hebbert, and N. R. Sandell, Jr., "A Tracking Filter for Maneuvering Sources," *IEEE Transactions on Automatic Control*, AC-22(2): 246 - 261, (March 1977).

45. Thorp, J. S. "Optimal Tracking of Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, AES-9(4): 512 - 519, (July 1973).

46. Tobin, D. M., and P. S. Maybeck, "Substantial Enhancements to a Multiple Model Adaptive Estimator for Target Image Tracking," *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, CA: 2002 - 2011, (December 1987).

47. Willsky, A. S. "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, 12: 601 - 611, (December 1976).

48. Willsky, A. S. *et al.* "Dynamic Model-Based Techniques for the Detection of Incidents on Freeways," *IEEE Transactions on Automatic Controls*, AC-25(3): 347 - 359, (June 1980).

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	4 December 1992	Master's Thesis

4. TITLE AND SUBTITLE
Failure Identification Using Multiple Model Adaptive Estimation for the LAMBDA Flight Vehicle

5. FUNDING NUMBERS

6. AUTHOR(S)
Peter D. Hanlon
Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/92D-19

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
WL/FIGS
WPAFB AFB, OH 45433-6553

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This study develops and investigates the performance of a Multiple Model Adaptive Estimator (MMAE) to detect and identify control surface and sensor failures on the LAMBDA flight vehicle (a URV developed by Wright Laboatories). The MMAE uses a bank of Kalman filters that predict the aircraft response to a given input, with each filter model based on a different failure hypothesis, and then forms the residual difference between the prediction and sensor measurements for each filter. The MMAE uses these residuals to determine the probabilities of the failures that are modeled by the Kalman filters. Initially the MMAE identified all these failures within 4 seconds of onset. Various performance improvement techniques were researched and the identification time was reduced to less than 2 seconds after failure onset. This improvement was mostly due to an increase in the penalty for measurement differences, and through retuning of the Kalmn filters. The MMAE performance was tested at the boundaries of the LAMBDA flight envelope, with good performance found at points close to the design flight condition. The performance at points that were far from the design flight condition indicates that gain scheduling is required to provide adequate performance across the entire envelope.

14. SUBJECT TERMS
Kalman Filter, Multiple Model Adaptation, Failure Identification

15. NUMBER OF PAGES
93

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	**Unclassified**	**Unclassified**	**UL**