MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION FOR THE VISTA F-16

THESIS

Robert Weston Lewis Senior
Captain, USAF

AFIT/GE/ENG/96D-29

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/96D-29

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION FOR THE VISTA F-16

THESIS

Robert Weston Lewis Senior
Captain, USAF

AFIT/GE/ENG/96D-29

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

AFIT/GE/ENG/96D-29

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION FOR THE VISTA F-16

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

Robert Weston Lewis, B.S.E.E.
Captain, USAF

December, 1996

Acknowledgments

This thesis will undoubtedly be the source of much pride and sense of accomplishment for the rest of my life. However it would not have been possible without the assistance of my family, friends, and professors. At the top of this list is my family. My wife Elizabeth deserves as much, or more, credit for this accomplishment as I do, for her unending encouragement is all that kept me motivated throughout the many low points of this experience. No one, including her, will never know how much she has helped me. Thank you Ely. My three boys, Robert, Christopher, and David, were extremely understanding throughout our stay at AFIT. There constant smiles, and forgiveness for my absences were an extremely important factor throughout this experience, and I couldn't be prouder of each one of them. There were many friends at AFIT. Like all difficult experiences, the bonds with your peers come fast, and strong. In particular, my two classmates, Jason McKay, and Nathan White, were invaluable. Their friendship and assistance will not be forgotten. My professors at AFIT were fantastic. Major Schneider provided an excellent introduction to several topics which I had absolutely no background. Dr. Pachter was outstanding in flight control, bringing not only an in-depth knowledge, but also a wealth of experience. Don Smith, while technically not a professor, played a vital roll by providing countless hours of assistance with the SIMSTAR and SUN workstations. Finally, my advisor, Dr. Peter Maybeck, who supported me from the very beginning, allowing me to take on the thesis topic of my choice, and guiding me to its completion. Without a doubt, you sir, are the most outstanding professor at this institution, and one of the finest men I have ever had the pleasure of knowing. It will always be with great pride that I will respond to the question, "Who was your advisor at AFIT?"; Dr. Peter Maybeck.

Table of Contents

List of Figures

List of Tables

Abstract

Performance characteristics of a Multiple Model Adaptive Estimation and Control Redistribution (MMAE/CR) algorithm are evaluated against single and double actuator and sensor failures. MMAE alone can compensate for sensor failures, whereas Control Redistribution compensates for actuator failures by redistributing commands, initially intended for failed actuators, to the unfailed actuators in such a way that the desired system response is achieved. Both failure detection and compensation capabilities are developed and analyzed through an extensive amount of simulation data, particularly addressing multiple failures. Simulations are performed utilizing the high-fidelity, non-linear six-degree-of-freedom Simulation Rapid-Prototyping Facility for the VISTA F-16, for both benign and maneuvering scenarios. Methods utilized to incorporate the MMAE/CR techniques are examined and modifications required to enhance performance are also presented. Results are presented which indicate the techniques incorporated provide an excellent means of both failure detection and compensation for the failures of both actuators and sensors. Approximately 98 percent of all secondary failures were successfully detected, and the majority of these detections are shown to occur in less than .5 seconds. The techniques of Multiple Model Adaptive Estimation and Control Redistribution are shown to complement each other well by providing improved failure detection in the face of actuator failures through the redistribution of the dither signal (used to enhance identifiability when there are no maneuvering commands to excite the system), and improved control authority through enhanced state variable estimation in the face of sensor failures. Control Redistribution is also shown to be an effective tool for modifying the dither to be applied once some actuators have already failed.

MULTIPLE MODEL ADAPTIVE ESTIMATION

AND

CONTROL REDISTRIBUTION

FOR THE VISTA F-16

1. Introduction

1.1 Chapter Overview

In this thesis the design of a Multiple Model Adaptive Estimator and Control Redistribution algorithm for the VISTA F-16 is presented. In Section 1.2 a brief motivation for this project will be provided. The problem statement and scope of the research will be presented Sections 1.3 and 1.4 respectively and in Section 1.5 the thesis format will be outlined.

1.2 Motivation

Reliability has always been a crucial element in development of flight control systems. The ability of an aircraft to withstand unexpected degradations in the performance of its equipment, as well as changes in its surrounding environment, has been the topic of extensive flight control research. The ability of a flight control system to survive abrupt, severe equipment degradations, such as sensor and actuator failures is an especially challenging, yet crucial requirement. This requirement for survivability is particularly critical in modern military aircraft which may be required to fly in hostile environments, making critical flight control hardware more susceptible to damage and or failure. To make matters worse, the desired maneuverability of modern fighter jets results in aircraft designs

which have very little, if any, inherent flight stability. It is this instability that makes the flight control system, and subsequently the rapid detection and compensation of hardware failures, absolutely critical.

Current state-of-the-art flight control systems rely heavily on physical redundancy to provide acceptable levels of reliability and survivability. This physical redundancy not only increases the expense of the flight control system, but also requires extra space and, perhaps more importantly, contributes additional weight, which limits the aircraft's range and payload capabilities. Another approach is to design a robust flight control system which provides *some* level of control despite changes in flight conditions or degradations in equipment. This approach has been widely utilized; however, any increase in robustness is balanced with a corresponding decrease in performance at design conditions, making it inappropriate for many high performance aircraft. These constraints have resulted in the desire for *reconfigurable* flight control systems for high performance aircraft that are both survivable and reliable without requiring excessive physical redundancies. This type of flight control system actually adapts to the changes mentioned above, resulting in very little, if any, performance loss as with the robust system. In this research reconfigurable flight control is obtained by utilizing Multiple Model Adaptive Estimation (MMAE) [28] and Control Redistribution [47].

MMAE has been shown to provide accurate estimates of the system's state variables despite single or even multiple failures of the system's sensors and actuators. By replacing the flight control system's raw sensor measurement inputs with a set of estimated measurements, the MMAE provides the desired adaptability to sensor failures. This technique also provides an accurate means of detecting and isolating the faulty components. Adaptability to actuator and control surface failures is obtained by a new technique known as Control Redistribution [47]. Control Redistribution uses the failure information provided by the MMAE to exploit the functional redundancies which *already* exist in the

various control effectors to provide the required dynamic stability and control of the aircraft. Essentially, Control Redistribution asks the question: If a given control surface has failed, how can the remaining control equipment be utilized in such a manner as to provide the same, or at least acceptable, performance?

1.3 Problem Statement

Previous research has demonstrated the capability of Multiple Model Adaptive Estimation to provide accurate state estimations despite actuator and sensor failures. This technique has also been shown to provide excellent detection and isolation of these failures [9, 10, 11, 38, 47, 48]. Recent research investigating Multiple Model Adaptive Control has resulted in a new technique entitled Control Redistribution [47]. Preliminary investigations into Control Redistribution have indicated a tremendous potential to enhance the survivability of an aircraft control system in the event of actuator failures. This research provides further insights to this technique. Specifically, the MMAE and Control Redistribution is shown to be capable of providing appropriate aircraft system responses to pilot input despite the failure of one, or multiple sensors and/or actuators. As with the previous research, the truth model used to confirm these capabilities is the Simulation Rapid-Prototyping Facility (SRF) VISTA F-16 simulation [9, 10,11, 47]. The SRF provides a state-of-the-art, nonlinear, 6-degree-of-freedom model of the VISTA F-16. Simulations were carefully constructed to provide both a realistic and challenging environment for this enhanced flight control system.

1.4 Assumptions

Because the verification of the SRF as a valid truth model of the VISTA F-16 is beyond the scope of this thesis, its accuracy is assumed to be acceptable. To evaluate the capabilities of the MMAE and the Control Redistribution system, sensor and actuator

failure models have been developed based on previous research, and enhancements as suggested by the thesis advisor and sponsor. It is assumed that these failure models will accurately represent actual failure modes of the VISTA F-16.

1.5 Thesis Format

This thesis is presented in five chapters. The motivation for developing an MMAE Control Redistribution system is presented in Chapter 1, along with the problem statement and assumptions. Chapter 2 provides a brief history of MMAE and Control Redistribution. Also in Chapter 2, the concepts of MMAE and Control Redistribution are explained, along with the modifications necessary for this application. The details of the truth, design and error models are presented in Chapter 3, as well as the development of the MMAE models and Control Redistribution algorithm for this specific application. The results of this research are reported in Chapter 4, and conclusions and recommendations are found in Chapter 5. Additional information is listed in the appendices to facilitate future research.

2. Algorithm Development

2.1 Chapter Overview

This chapter presents the basic background required to understand the various techniques utilized in this research. Section 2.2 presents a brief history of the development of the MMAE and Control Redistribution concepts. The details of the two concepts are presented in Sections 2.3 and 2.4, and the modifications required for this specific application are presented in Section 2.5.

2.2 Concept History

This research examines the combination of a concept that has been under development over the past thirty years with one that has only recently been proposed. The concept of utilizing Multiple Model techniques to provide state estimations in systems containing time varying parameters has its origins as far back as 1965. At that time, the usefulness of the technique was limited by the computational and memory capabilities available. With current technology, computer capabilities have improved to the point where the Multiple Model techniques can now be implemented in applications requiring real time processing. In contrast, the Control Redistribution technique has resulted from very recent efforts in research performed at the Air Force Institute of Technology (AFIT).

Multiple Model Adaptive Estimation was first presented in a paper by Magill in 1965 [22]. Magill does not actually use the term MMAE; however, the concept of providing separate estimations from filters based on a parameter discretization, and blending these estimations to form the "optimal adaptive estimate" is clearly the beginning of MMAE as it is currently used in practice. Subsequent efforts by Lainiotis [21] focused

on appending each elemental filter in the MMAE with a separate controller, forming the structure known as Multiple Model Adaptive Control (MMAC). While this research does not utilize the MMAC approach directly, the developments in this area have defined several key aspects of the MMAE. One particularly noteworthy effort was that of Athans et. al. [2]. The research presented by Athans develops an MMAC to handle parameter variations due to changing flight conditions. The Athans work introduces several issues which must be considered in an MMAE development, such as the research presented here. These issues include the use of a lower bound on the probability calculations to improve the responsiveness of the system, the effect of a phenomenon referred to as "beta dominance" (developed further in Section 2.3), and the use of dither input signals to assist in the identification of failures in benign conditions. Also developed in this paper, and in subsequent work by Chang and Athans [8], is the realization that adaptive techniques of this nature do *not,* in general, provide optimal estimations or control calculations. While proofs do exist, in the discretized case, that show the MMAE converges to a point estimate closest to the true parameter value in the Baram distance measure sense [3, 4, 5], optimality can only be shown to exist if the unknown parameters are exactly matched in one of the elemental filters, and if those parameters remain constant. Realistically, this exact match can never actually occur, and, with the exception of initial parameter value acquisition, the case of constant parameters eliminates the need of adaptive techniques. The fact that an MMAE performs a Bayesian type blending of outputs, which in turn requires that a lower bound be placed on the probability of correctness assigned to each filter in order to handle time-varying parameter values (discussed further in Section 2.5.1) precludes it from ever being considered truly optimum, even if an exact replication of the parameters were available. However, it is this blending of several hypothesis which allows the designer of an MMAE to perform a relatively coarse discretization of the time varying parameter space, thus making the MMAE a feasible approach. Athans continued beyond the issue of optimality by showing that, with various (albeit ad-hoc) considerations, this heuristic

method had the potential to provide quick, accurate estimations despite time-varying parameters.

Even with the increased computational capabilities of present computer technology, a single-level approach to the discretization of a given parameter space can quickly result in an overwhelming number of elemental Kalman filters. For example, it will be shown that the number of possible single and double hard failure hypotheses for the aircraft considered in this research is 67. While the Bayesian blending mentioned above eliminates the need to define the infinite possible hypotheses for partial failures, a limit will still exist concerning the coarseness of the discretization. For this reason, hierarchical structures, first introduced by Fry and Sage [12] are used. The research by Fry and Sage used MMAE for system identification and broke the problem into separate subsystems, each with its own parameter. This technique reduced the number of filters which had to be on-line at any given time. Hierarchical structures have been extensively developed in AFIT research [9, 10, 11, 35, 37, 47, 48]. This research will also use a hierarchical approach. Here, the system examines the possibility of only one failure at any given time. The multiple model algorithm will rely on a bank of filters, each of which hypothesizes a single failure (one of eleven possible) or no failure. Once a failure occurs, the system switches to a second bank which contains filters hypothesizing two failures (the first of which has just been declared), a filter hypothesizing only the recently declared failure, and a filter hypothesizing no failures. The last, no failure, hypothesis allows the algorithm to "back out" of the second bank of filters and return to the first bank if subsequent measurements indicate that the first failure declaration was erroneous. Only one bank of filters will be on line at any given time, which reduces the number of on-line filters from 67 (a no-failure hypothesis plus all possible single and double combinations considered simultaneously) to 12. This technique will be explained in more detail in Section 2.4.

Several significant contributions to the development of the MMAE/MMAC concept have been made by the Air Force Institute of Technology (AFIT). Under the guidance of Dr. Peter Maybeck, several significant contributions in this area have been made in the utilization of these techniques. Many applications have been, and continue to be examined, including flight control [9, 10, 11, 19, 20, 31, 32, 44, 47], control of flexible space structures [16, 34, 46], response enhancements to virtual reality simulators [18], target tracking with infrared sensors and lasers [33], and the detection of jamming and spoofing signals in navigation systems utilizing the Global Positioning System, or GPS [51].

In 1985 Maybeck and Suizu [36] examined the beta dominance phenomena and were able to alleviate the situation by eliminating the beta term in their calculations. The term "beta dominance" refers to the false failure declarations which occur due to the β_k term in the probability weighting calculation for each filter (explained further in Sections 2.3.3 and 2.5.2). This development has been extremely useful in increasing the accuracy of failure declarations in subsequent AFIT MMAC/MMAE research [9, 10, 11, 17, 19, 20, 29, 35, 38, 39, 47, 48, 50].

The concept of a "moving bank" hierarchical technique was developed to reduce the number of on-line filters and has been used extensively in AFIT research [15, 16, 25, 30, 34, 46]. This concept has been used to control the vibration and bending of a large, flexible space structure and also in several flight control applications, including that which is presented here.

Flight control applications of MMAC/MMAE at AFIT have been quite numerous. Pogoda's research [31, 32, 44] provided the first MMAC/MMAE flight control application in 1988. The work by Pogoda examined failure detection on the STOL F-15. This research with the STOL F-15 was continued by Stevens [35, 48] in 1989. Stevens utilized the MMAC to compensate for both partial and multiple failures. Stevens utilized the

Bayesian blending technique mentioned previously in this section (developed in more detail in Section 2.4) to allow for partial failure modes which were not specifically hypothesized by one of the elemental filters. Stevens developed the concept of scalar residual monitoring to reduce false failure declarations and also presented more evidence that the removal of the beta term will also alleviate this problem. Stevens also was the first to establish the specific form of hierarchical technique for multiple-failure detection.

In 1990, Martin [23] developed an MMAC system for the AFTI F-16. Martin's research led to the design of several elemental Linear Quadratic Gaussian (LQG) controllers for an MMAC for both the longitudinal and lateral directional channels. At this point Multiple Model techniques for the Variable-Stability In-flight Simulator Test Aircraft (VISTA) F-16 were initiated. Both Stratton [50] and Menke [38, 39] developed MMAE-based controllers for the VISTA F-16 in 1991 and 1993, respectively. This research provided several developments including the use of dither signals to enhance observability, and scalar residual monitoring as a means of corroborating the detection of failures. Also in 1993, MMAE-based control research was performed by Lane [19, 20] and Hanlon [17, 29] on the LAMBDA unmanned research vehicle (URV). This LAMBDA URV research examined several issues including probability smoothing to reduce the occurrence of false or improper failure declarations and enhancing failure detection through residual propagation. Residual propagation allows the Kalman filters to propagate several sample periods before measurement updating, allowing residuals in filters based on wrong hypotheses to grow larger than they would with updates every sample period. This process makes these residuals more distinguishable from the residuals in the filter which is based on the correct hypothesis. The LAMBDA research also examined the effects of increasing the scalar penalty on measurement residuals to enhance the response time of the algorithm.

In 1994 and 1995 the VISTA F-16 research continued with the both Eide and Stepaniak [9, 10, 11, 47]. This research utilized the nonlinear six degree-of-freedom Simulation Rapid-Prototyping Facility (SRF) to provide a more realistic truth model for the development and analysis of the design than the linearized simulation models used previously. In Eide's research, an MMAE-based control system was developed. This research developed the elemental filters to detect both single and double hard failures, and is the basis for the MMAE which will be utilized in the research presented here. In addition to the development and evaluation of the MMAE-based controller, Eide also presents several key observations in the area of residual monitoring. An important example of these observations is that the residuals of a filter based on the correct hypothesis of the system's condition are shown to display the basic traits of zero-mean, white, Gaussian noise, while the residuals of the filter with an incorrect hypotheses often contain the obvious characteristics of the applied dither signal (dither is often applied to increase the observability of the system and will be discussed further in Section 2.5.4). Stepaniak attempted to develop an MMAC for the VISTA F-16 (the differences between the MMAE-based controller and MMAC will be further developed in Section 2.4). Stepaniak utilized the work of Menke for the MMAE portion of the design, then developed individual elemental LQG controllers for the output of each filter. Several unresolved numerical and modeling difficulties led to an unsatisfactory design in this case. Undaunted, Stepaniak developed a new technique to take advantage of the MMAE output. This technique, labeled MMAE-Based Control with Control Redistribution, utilizes failure information from the MMAE to "redistribute" the control inputs away from failed or unhealthy surfaces, to the proper combination of those which will provide the same aircraft system response. This technique takes advantage of inherent redundancies present in the control surfaces of the aircraft. Stepaniak had very little time to verify this concept, but his preliminary efforts show excellent promise.

The research presented herein picks up at the end of Stepaniak's work. The concept of Control Redistribution is thoroughly tested for many types of failure scenarios, and several modifications and conclusions are also presented.

2.3 Multiple Model Adaptive Estimation

2.3.1 Overview

The control of complex systems such as aircraft can only be accomplished by first estimating the various state variables that depict the properties or conditions of the system and which govern the desired response. These "states" are typically estimated based on a mathematical model of the system, previously applied control inputs, and sensed measurements of the actual conditions. The accuracy of the mathematical model is based, in part, on various parameters within its structure. Because some or all of these parameter values may vary over time, the need exists for an estimator that will adapt to these variations. In this research, a technique referred to as Multiple Model Adaptive Estimation (MMAE) is used to provide this capability.

The basic concept behind MMAE is actually quite simple. Rather than developing one state estimator, or system model, based on a single value for each parameter, several models are constructed, each depicting a different hypothesized set of values of the parameter vector within the appropriate parameter space. A filter based on each of these models is then subjected to the same set of control and sensor inputs, while the estimation performance (as indicated by the filter residuals) of each is monitored to determine which is the best hypothesis at any given time. The basic structure of the Multiple Model Adaptive Estimator is depicted in Figure 2-1.

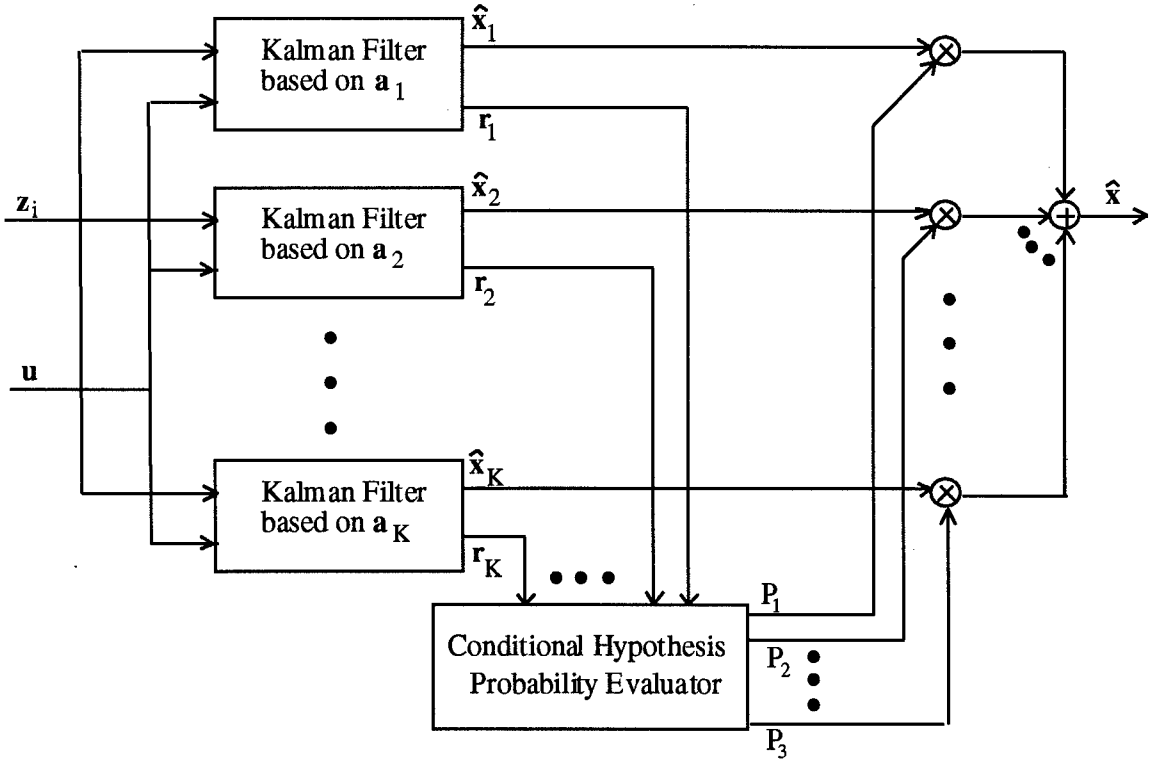

Figure 2-1 Multiple Model Adaptive Estimator

In order to design each of the models discussed above, the extent of the parameter variations, or parameter space, must be determined. Once this has been accomplished, a representative discretization of this parameter space must be determined based on a trade-off between computational burden and the fidelity of the representation. The values chosen for this discretized parameter space are typically depicted by the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K$.

Performance monitoring of each filter is accomplished by comparing its predicted measurement values with those actually provided by the system sensors. This difference, or residual, is then used by the Conditional Hypothesis Probability Evaluator to determine the probability that the model within that filter is the model that best describes the current behavior of the system. Once this probability has been determined, two approaches can be

taken to determine the final state estimation. The first method simply outputs the estimates from the filter which has been assigned the highest probability. This technique, referred to as maximum a posteriori (MAP), is appropriate for applications in which all possible parameter values can be depicted in the discretization discussed above. The second method applies a statistical weighting to each model output equivalent to the probability of its correctness. Each of the weighted filter outputs is then summed, resulting in a Bayesian blending of each elemental state estimator. This technique provides estimates which are somewhere between those of the actual elemental models, thus allowing a continuous parameter space, with infinite possible variations, to be depicted with a discrete number of models. This Bayesian form of MMAE is what is depicted in Figure 2-1.

In this research the parameter space used to generate the various estimators in the MMAE depicts all of the possible failure conditions of the sensors and control surface actuators of the VISTA F-16. Initially it might appear as though all of these failure modes could be easily represented by a finite number of parameter values, thus allowing the MAP version of the MMAE discussed above. This approach does not consider the infinitely possible partial failures which could occur in any of the modeled sensors or actuators. Fortunately, the Bayesian blending approach does account for this partial failure mode by blending the estimates of the models which depict either no failures, or fully failed components. The ability of the Bayesian blended MMAE to provide adequate performance against partial failures, despite the lack of elemental filters explicitly depicting any such hypotheses, has been extensively demonstrated in previous research [9, 10, 11, 39, 35, 38, 47, 48].

2.3.2 Kalman Filters

The state estimators used in the MMAE of this research are Kalman filters. Each filter provides a state estimate which is based on an internal model of the system (including

a particular hypothesized failure), the previously applied control inputs, and the sensor measurements. Since the sensor measurements of the VISTA F-16 are sampled (at 64 Hz), the Kalman filters for the MMAE are designed in the discrete-time domain. It is assumed that the reader has a basic understanding of the Kalman filter, and therefore only the equations are presented here. The complete derivation of these equations can be found in Maybeck [26].

The dynamics model of the system is assumed to be a linear, time-invariant, discrete-time system as described by Equation (2.1):

$$\mathbf{x}(t_i) = \Phi\, \mathbf{x}(t_i) + \mathbf{B}_d\mathbf{u}(t_i) + \mathbf{G}_d\mathbf{w}_d(t_i)$$

$$\mathbf{z}(t_i) = \mathbf{H}\mathbf{x}(t_i) + \mathbf{D}_z\mathbf{u}(t_i) + \mathbf{v}(t_i) \qquad (2.1)$$

where $\mathbf{x}$ represents the vector of system states, $\mathbf{u}$ the control input vector, and $\mathbf{z}$ the measurement vector. The dynamics driving noise $\mathbf{w}_d$ and measurement corruption noise $\mathbf{v}$ are both discrete-time, white Gaussian noises, each having the following statistics:

$$E[\mathbf{w}_d(t_i)] = \mathbf{0} \quad E[\mathbf{w}_d(t_i)\,\mathbf{w}_d^T(t_i)] = \mathbf{Q}_d$$

$$E[\mathbf{v}(t_i)] = \mathbf{0} \quad E[\mathbf{v}(t_i)\mathbf{v}^T(t_i)] = \mathbf{R}$$

$$E[\mathbf{w}_d(t_i)\mathbf{v}^T(t_i)] = \mathbf{0} \qquad (2.2)$$

The propagation equations for each filter are given below in Equations (2.3) and (2.4):

$$\hat{\mathbf{x}}_k(t_{i+1}^-) = \Phi_k \hat{\mathbf{x}}_k(t_i^+) + \mathbf{B}_{dk}\,\mathbf{u}(t_i) \tag{2.3}$$

$$\mathbf{P}_k(t_{i+1}^-) = \Phi_k \mathbf{P}_k(t_i^+)\Phi_k^T + \mathbf{G}_{dk}\,\mathbf{Q}_{dk}\,\mathbf{G}_{dk}^T \tag{2.4}$$

where $\hat{\mathbf{x}}_k(t_{i+1}^-)$ represents the k^{th} elemental filter estimate of the system states just before the measurement update at time t_{i+1}, and $\mathbf{P}_k(t_{i+1}^-)$ represents the state error covariance matrix for that elemental filter. Finally, the update equations which incorporate the sensor measurements are given below in Equations (2.5) and (2.6):

$$\hat{\mathbf{x}}_k(t_i^+) = \hat{\mathbf{x}}_k(t_i^-) + \mathbf{K}_k(t_i)\,\mathbf{r}_k(t_i) \tag{2.5}$$

$$\mathbf{P}_k(t_i^+) = \mathbf{P}_k(t_i^-) - \mathbf{K}_k(t_i)\mathbf{H}_k\mathbf{P}_k(t_i^-) \tag{2.6}$$

where

$$\mathbf{r}_k(t_i) = \mathbf{z}(t_i) - \mathbf{H}_k\,\hat{\mathbf{x}}_k(t_i^-) - \mathbf{D}_z\mathbf{u}(t_i) \tag{2.7}$$

$$\mathbf{K}_k(t_i) = \mathbf{P}_k(t_i^-)\,\mathbf{H}_k^T\mathbf{A}_k^{-1}(t_i) \tag{2.8}$$

$$\mathbf{A}_k(t_i) = \mathbf{H}_k\,\mathbf{P}_k(t_i^-)\mathbf{H}_k^T + \mathbf{R}_k \tag{2.9}$$

In these equations the times (t_i^-) and (t_i^+) represent the times just before and just after the sensor measurements are provided. The residual vector discussed in Section 2.3.1 is defined by Equation (2.7) and is utilized in Equation (2.5) for the update of the state estimates. These residuals are zero-mean and their covariance is represented by $\mathbf{A}_k(t_i)$, as shown by Equation (2.9). The actual Kalman filters used in this research are steady-state constant-gain filters, which eliminate the need for the time arguments shown on $\mathbf{P}_k$, $\mathbf{K}_k$, and $\mathbf{A}_k$.

2.3.3 Conditional Hypothesis Probability Evaluator

As discussed in Section 2.3.1, and depicted in Figure 2-1, the Conditional Hypothesis Probability Evaluator utilizes the residual information to assign a probability of correctness weighting to each filter output. This probability can be expressed as

$$p_k(t_i) = \text{prob}\{\mathbf{a}=\mathbf{a}_k \mid \mathbf{Z}(t_i)=\mathbf{Z}_i\} \tag{2.10}$$

where $\mathbf{Z}(t_i)$ represents the history of sensor measurements up to an including time t_i, $\mathbf{a}$ is the parameter random variable representing the failure condition, and $\mathbf{a}_k$ is the realization of the failure assumed by the k^{th} filter [47]. This probability is shown in [27] to be calculated by

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) \cdot p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_j, \mathbf{Z}_{i-1}) \cdot p_j(t_{i-1})} \tag{2.11}$$

where

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) = \beta_k(t_i) \exp\{-\frac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\} \tag{2.12}$$

$$\beta_k(t_i) = \frac{1}{(2\pi)^{m/2} |\mathbf{A}_k(t_i)|^{1/2}} \tag{2.13}$$

The actual probability information is depicted in the numerator of Equation (2.11), and the denominator scales the calculation to ensure that the summation of the probabilities from each filter equals one:

$$p_k(t_i) \geq 0 \quad \text{for all } k \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1 \tag{2.14}$$

Equation (2.11) is the density function of the k^{th} Kalman filter's residuals, which were shown in Section 2.3.2 to be zero-mean, white, Gaussian, with covariance $\mathbf{A}_k(t_i)$.

The quadratic term depicted in Equation (2.12) provides a good insight to the method by which the Conditional Hypothesis Probability Evaluator determines how each filter should be weighted. To examine this term, let it first be defined as the likelihood quotient:

$$L_k(t_i) = \mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i) \tag{2.15}$$

Close examination of Equation (2.15) reveals that, in the scalar measurement case, it is indeed a quotient between the square of the residual from the k^{th} filter to the predicted variance computed by the *same* filter. If this filter's hypothesized failure does indeed match the actual conditions of the system, the value of $L_k(t_i)$ will be on the order of m, the dimension of the residual vector $\mathbf{r}_k(t_i)$, and the measurement vector $\mathbf{z}_k(t_i)$. If, however, the hypothesis of the particular filter does not match the actual system, then the residual vector,

$\mathbf{r}_k$, will increase in magnitude, resulting in an increase in the likelihood quotient. Since this quotient is multiplied by a negative scalar, then used as the exponential term in Equation (2.12), it can be seen that the result will be a **decrease** in probability density value if the system deviates from the hypothesis of a particular filter. Using this information, Equation (2.11) calculates the appropriate weighting for each filter.

2.4 Control Redistribution

2.4.1 Overview

As mentioned in Section 1.2, this research examines the combination of a MMAE with a technique known as Control Redistribution to provide the VISTA-F16 with acceptable flight control in the presence of failed components. Figure 2-2 shows how this structure has been implemented in past research [47].

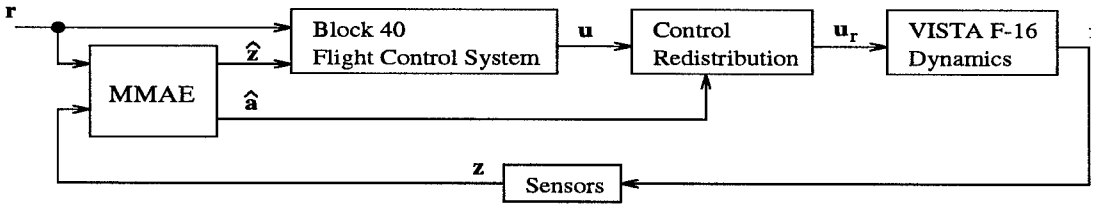

Figure 2-2 MMAE with Control Redistribution

Figure 2-2 shows that the MMAE, which was described in Section 2.3.3, provides estimates of both the parameter value and the system measurements (the latter to be used to replace raw sensor measurements, to protect against sensor failures). This is accomplished by using the probability weightings from the Conditional Hypothesis Probability Evaluator on the outputs of the Kalman filters (for the measurement estimates) as well as the sets of parameters for which each filter is designed. This structure is shown in Figure 2-3, where

$\hat{\mathbf{x}}$ is shown being passed to the controller; with this $\hat{\mathbf{x}}$, $\hat{\mathbf{z}}$ and can be generated as $[\mathbf{H}\hat{\mathbf{x}} + \mathbf{Du}]$. Note also that the controller in Figure 2-3 is the cascade combination of the "Block 40 Flight Control System" and the "Control Redistribution" blocks shown in Figure 2-2. By utilizing the *existing* Block 40 Flight Control System for the VISTA F-16, the reliability and survivability of the aircraft can be enhanced, while simultaneously guaranteeing that the desirable characteristics of the current flight control system are maintained. Since the existing flight control system remains intact, this proposed structure would also be relatively easy and inexpensive to implement and test.

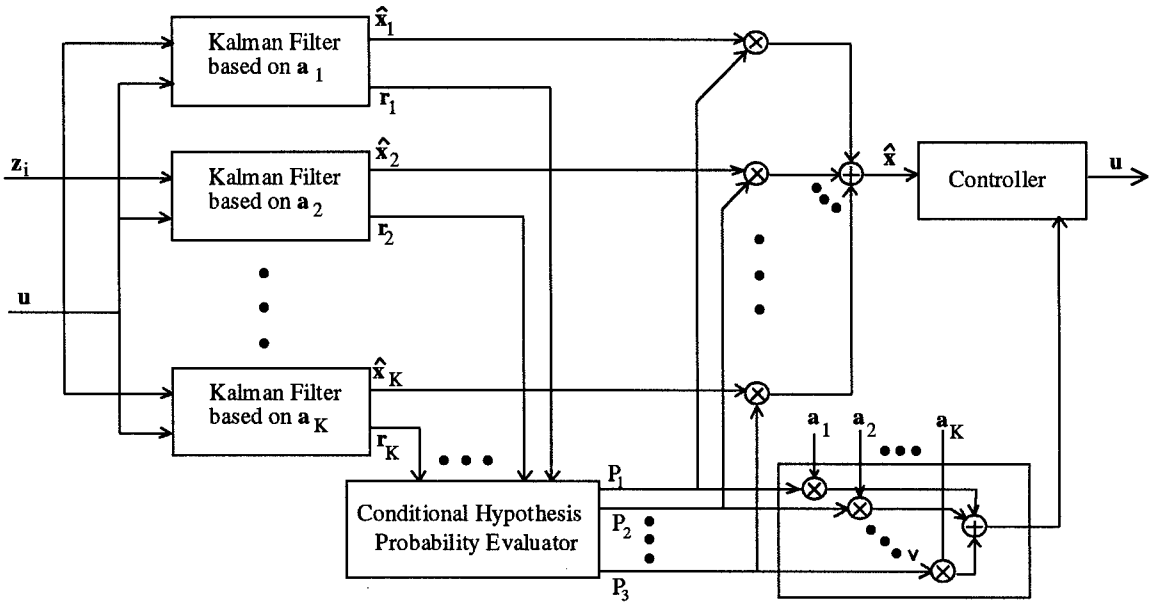

Figure 2-3 MMAE with Parameter Estimation

Based on the parameter estimation from the MMAE, the Control Redistribution algorithm manipulates the original control vector, $\mathbf{u}$, to form a new control vector $\mathbf{u}_r$. If the MMAE is estimating that no failure exists in the aircraft's actuators, then $\mathbf{u}_r$ will be identical to $\mathbf{u}$, i.e. the control inputs from the Block 40 flight control system will pass directly to their intended actuators. However, if the MMAE has detected an actuator failure, the

Control Redistribution algorithms output $\mathbf{u}_r$ will effectively reroute the intended command signals away from failed actuators to the combination of those which can provide the same desired aircraft system response.

2.4.2 Algorithm Development

Stepaniak's development of the Control Redistribution technique [47] is based on equating the *redistributed* control input of a failed system, to the control input of a fully functional system as shown in Equation (2.16):

$$\mathbf{B}_{fail}\mathbf{u}_r = \mathbf{B}\mathbf{u} \tag{2.16}$$

where the control input distribution matrix for the failed and fully functional system are represented by $\mathbf{B}_{fail}$ and $\mathbf{B}$, respectively. Note that $\mathbf{B}_{fail}$ is simply $\mathbf{B}$ with a single column (corresponding to the specific failed actuator) zeroed out. The redistributed and normal control inputs are represented by $\mathbf{u}_r$ and $\mathbf{u}$, respectively. To obtain the vector $\mathbf{u}_r$, $\mathbf{u}$ is multiplied by a *redistribution* matrix $\mathbf{D}_{ai}$, which corresponds to the complete failure of the i^{th} actuator (thus the subscript "*ai*").

$$\mathbf{u}_r = \mathbf{D}_{ai}\mathbf{u} \tag{2.17}$$

It is exactly this matrix, $\mathbf{D}_{ai}$, for $i = 1, 2, \ldots$, [number of scalar control variables], which must be computed a priori for the algorithm to accomplish the required objective.

To simulate the failure of a particular actuator, the corresponding column of control input distribution matrix $\mathbf{B}$, is multiplied by a constant, $0 \leq \varepsilon \leq 1$ (where $\varepsilon = 0$ corresponds

to a total failure). This is accomplished by forming the matrix $\mathbf{F}_{ai}$ (where the subscript *ai* again denotes the i^{th} *a*ctuator) which results in the following relationship:

$$\mathbf{B}_{fail} = \mathbf{B}\mathbf{F}_{ai} \tag{2.18}$$

Substituting equations (2.17) and (2.18) into (2.16), the following relationship is obtained:

$$\mathbf{B}\mathbf{F}_{ai}\mathbf{D}_{ai} = \mathbf{B} \tag{2.19}$$

To the casual observer, it would appear that Equation (2.19) indicates that the distribution matrix, $\mathbf{D}_{ai}$, is simply equal to the inverse of the failure matrix $\mathbf{F}_{ai}$. A more careful analysis would reveal that a definite problem exists with this solution if $\mathbf{F}_{ai}$ is rank deficient. This certainly will be the case if the value of e, which indicates the failure in a row of $\mathbf{F}_{ai}$, is zero, denoting a total failure of a particular actuator. Therefore the pseudoinverse of $\mathbf{B}\mathbf{F}_{ai}$ is required to obtain an *approximation* of $\mathbf{D}_{ai}$:

$$\mathbf{D}_{ai} \approx (\mathbf{B}\mathbf{F}_{ai})^{\dagger}\,\mathbf{B} \tag{2.20}$$

where the superscript † indicates the Moore-Penrose pseudoinverse. The results of Equation (2.20) will be the identity matrix with the exception of the i^{th} column, which will be replaced with a column that will redistribute the control inputs, away from the failed i^{th} actuator, to an appropriate combination of the remaining, healthy, actuators. A typical example of a Control Redistribution matrix, $\mathbf{D}_{ai}$, is provided in Equation (2.21). Here, $\mathbf{D}_{ai}$ corresponds to the first failure hypothesis (i = 1), which corresponds to a failed left stabilator.

$$\mathbf{D}_{a1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0.9060 & 0 & 1 & 0 & 0 \\ -0.9060 & 0 & 0 & 1 & 0 \\ -0.7802 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.21)$$

Because the (1,1) position of the matrix in Equation (2.21) is zero, no command will be sent to the left stabilator. Any requirement for this actuator will instead be *redistributed* to the other control surfaces as indicated by the remainder of the first column. If the failure indicated by the matrix $\mathbf{F}_{ai}$ indicates a partial failure, Equation (2.20) will yield the exact relationship. While it might be argued that the occurrence of a partial failure is more likely than a total failure (the latter resulting in the approximate solution), some level of consolation is obtained from knowing that the pseudoinverse does give the closest approximation in the least squares sense [49]. This research investigates the Control Redistribution algorithm for single and multiple fully failed ($\varepsilon = 0$) components.

2.5 Required Modifications

As discussed in Section 2.2, past research in the Multiple Model concept has resulted in several modifications to improve the performance characteristics of the technique. Several of these general modifications, as well some specific to this application, are incorporated into this research. The Control Redistribution, while being a relatively new concept, has also required a few modifications to enhance the performance of this design.

2.5.1 *Lower Bounding*

Because the probability weighting calculation shown in Equation (2.11) is recursive (i.e. the calculation of $p_k(t_i)$ is based on the result of $p_k(t_{i-1})$), the result can never be allowed to equal zero. This occurrence would result in any future calculation for that particular weighting to be forced to zero, thereby locking out the k^{th} filter's hypothesis. Since the MMAE relies more heavily on the relative weighting between the elemental filters, this situation is resolved by equally limiting each p_k to a lower bound. As mentioned in Section 2.2, this concept was first introduced by Athans in his F-8C research [2]. The necessary adjustments are made so that the summation of the probabilities, as depicted in Equation (2.13), still equals one. While increasing the lower bound enhances the responsiveness of the MMAE, it also makes it more susceptible to false alarms and other undesirable behavior. Based on research performed by Stevens [35, 48], a bound of 0.001 was found to be satisfactory for this application.

2.5.2 *Beta Dominance*

Returning once again to the work of Athans [2], the issue of beta dominance is now addressed. Recall that the β_k term is also related to the probability weighting calculation $p_k(t_i)$ shown in Equation (2.11), and first appeared in the density calculation of Equation (2.12) and is defined in Equation (2.13). These equations are repeated here for convenience:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) \cdot p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_j, \mathbf{Z}_{i-1}) \cdot p_j(t_{i-1})} \qquad (2.11)$$

where

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \,|\mathbf{a}_k, \mathbf{Z}_{i-1}) = \beta_k(t_i) \exp\{-\frac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\} \tag{2.12}$$

$$\beta_k(t_i) = \frac{1}{(2\pi)^{m/2}\,|\mathbf{A}_k(t_i)|^{1/2}} \tag{2.13}$$

The actual term "Beta Dominance" refers to the tendency of the MMAE to make erroneous failure declarations due to the effect of the beta term on filters with similar $\mathbf{r}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{r}$ terms. If this quadratic form of two filters are nearly equal, they should, hopefully, receive a correspondingly nearly equal weighting from Equation (2.11). However, because of the beta term, the filter estimates with the lower magnitude determinant of filter-predicted covariance, A_k, will incorrectly be weighted higher. Since the covariance calculation relies heavily on the values contained in **H** (see Equation (2.5)), those filters which have hypothesized a sensor failure (by zeroing out a row of **H**) will tend to be disproportionately weighted, leading to false alarms on sensors.

Previous AFIT research into this problem has revealed that an effective method of dealing with this issue is to simply remove the beta term from the density calculation of Equation (2.11) [35, 36, 48], yielding the following replacement:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \,|\mathbf{a}_k, \mathbf{Z}_{i-1}) = \exp\{-\frac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\} \tag{2.22}$$

Because of this change, Equation (2.21) can no longer be considered a *true* density calculation since the area under this function is no longer one. For the MMAE, this issue is

of little importance, since (2.21) is utilized only to determine the probability weightings in Equation (2.11), and the denominator of that equation will continue to ensure that the probabilities sum to one, as stated in Equation (2.14).

2.5.3 Scalar Penalty

As with previous research in this area [9, 10, 11, 39, 38, 47] the magnitude of the scalar coefficient (-1/2) found in the quadratic term of Equation (2.12) was utilized to adjust the sensitivity of the MMAE. The magnitude of this term has been found to have a direct relationship with the responsiveness of the of the adaptation process, increasing the speed at which the probability weightings of mismatched filters are forced toward zero. This improvement is met with a corresponding increase in false alarms, thereby limiting its usefulness. Therefore, a value of (-1/2) has been used throughout this research.

2.5.4 Dither

Despite the exceptional failure detection efforts of the MMAE, certain failure conditions are still somewhat challenging. This is due to the fact that, under certain conditions, one or more of the parameters could lack sufficient identifiability. The identifiability problem is better understood by considering the fact that the filters which compose the MMAE rely on control inputs (to excite the system in order to enhance identifiability) as well as sensor measurements to make their estimations. If an actuator fails while its corresponding control surface is in free steam, and an aircraft is in steady-level flight (i.e. no control input is being applied to the actuator), then the MMAE will have no chance of detecting the problem. One might argue that, under these conditions, the detection of the failure is not required; however, the objective of any system such as that proposed by this research is to detect the failure and provide the necessary compensation before it becomes a serious problem. One method of enhancing failure detection in such a

benign environment is to apply dither signals to actuators of the aircraft. These signals provide the actuators with continually applied, low magnitude, control signals which excite the surfaces and allow the sensors to provide the necessary feedback so that the MMAE can adequately detect any failures. An extensive amount of research has been performed to determine the most effective magnitude and type of dither [14, 45]. The problem has two opposing constraints: first, the signals must be strong enough to provide increased identifiability of actuator failures, and second, the signals must be low enough so as to be subliminal to the pilot. Past research efforts [9, 10, 11, 39, 38, 47] at AFIT with the VISTA F-16 have utilized sinusoidal signals with a frequency of 15 radians/second and limited the magnitude to below + 0.1 g's in the longitudinal direction, and + 0.2 g's in the lateral direction.

Despite an exceptional amount of effort in this area, problems still exist. This is especially true for the case of multiple failure situations in which the first declared failure is an actuator. When the first actuator failure occurs, the decrease in control authority directly affects the ability to detect any future failures. This problem is easily understood when one considers the fact that the dither signal discussed above is no longer having any effect on the control surface corresponding to the failed actuator. Therefore, the system is not excited enough to enhance parameter identifiability as much as in the unfailed-actuator case. In this research, the dither signal is applied via the Control Redistribution portion of the system so that the dither will continue to add full benefit to the detection of multiple failures. The results of this approach are reported in Chapter 4.

2.5.5 Control Redistribution

In Chapter 4, the results of routing the dither signal through the Control Redistribution portion of the system show that this technique does assist in the detection of failures, especially in the case of multiple failures in which the first failure is an actuator.

Unfortunately, this technique also resulted in some difficulties. Because the dither signals used were optimized to provide rapid single failure detection, they fluctuate the actuators at rates which sometimes reach the saturation limits. This is not a problem until Control Redistribution reroutes the commands intended for failed actuators, to those which are behaving normally. For actual pilot commanded inputs, this is not a problem, since the dither signals are only required during steady level flight, and are therefore turned off during any input from the pilot. If, however, one or more actuators fail and there is no pilot commanded input, the dither signal will be compounded on the remaining healthy actuators, in an attempt to provide the same observability and failure detection capability as for a fully functional aircraft. If this new, compounded dither signal surpasses the rate limits of any given actuator, the corresponding motion of that surface will no longer follow the linear model upon which the Kalman filters were based. The result of this departure from linear response will be unpredictable fluctuations in the residuals from the Kalman filters based on each hypothesis model, which, in-turn, will result in erratic and false failure declarations. To remedy this situation, the redistributed dither signals were scaled so that they would not result in the rate saturation of any actuators. This problem, its solution, and the results are all included in Chapter 4.

2.5.6 Hierarchical Structure

Due to the number of failure possibilities that exist for this problem, and the relatively rapid sampling rate with which the bank of elemental filters within the MMAE must be updated (64 Hz), the simple, one-level, structure of the MMAE in Figure 2-1 is clearly not an option. In the general case, if there are K separate single failure combinations, the number of required filter hypotheses would be K+1 (including the fully functional model). If the double failure is considered, this number increases to $1+K+\frac{K!}{(K-2)!\cdot 2!}$. The MMAE used in this research considers the failure of 5 sensors and

6 actuators, making K=11. This results in 67 filters being required to hypothesize all failure possibilities. Clearly the computational requirements to implement 67 Kalman filters (each having 11 states) and the corresponding Conditional Hypothesis Probability Evaluator, all of which would be required to perform the necessary calculations presented in Section 2.3, would be enormous.

In an effort to bring this computational requirement down to a more manageable level, Stevens [35, 48] developed the hierarchical structure depicted in Figure 2-4. This structure requires the MMAE first to examine the possibility of a single failure (12-filter) case. Once a single failure has been detected, the filters used in the single failure case (level 0) are replaced by the bank of filters which considers a second failure. This second bank will contain one filter that models just the currently declared failure, 10 filters that model the currently declared failure plus the failure of one of the remaining components, plus one

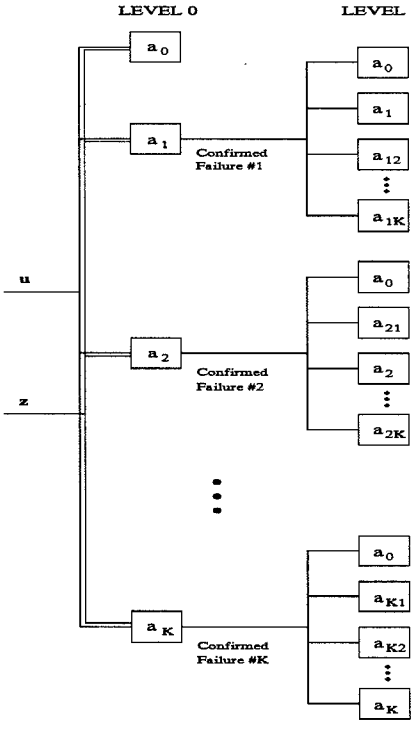

Figure 2-4 Hierarchical Structure

filter based on a fully functional aircraft model, which will allow the system to go back to level 0. By requiring the MMAE to observe only the next possible failure, the required number of on-line filters at any time is reduced from 67 to 12.

2.6 Chapter Summary

This chapter has presented the background required to understand this research effort. A brief history of Multiple Model Adaptive Estimation and Control Redistribution techniques was provided, concentrating on research that has been performed at AFIT. This background was followed by the specific algorithm development of each technique. Starting with the MMAE, both the Kalman filter and the Conditional Hypothesis Probability Evaluator were developed. The Control Redistribution concept was then presented, showing the development of the approximation for the Control Redistribution matrix. The Modifications required for this research are presented in Section 2.5. This section discusses lower bounding, beta dominance, scalar penalty modifications, dither, and the hierarchical structure of the MMAE for this application. Chapter 3 will build on these concepts, presenting the details of the design utilized for this research.

3. *Aircraft Models*

3.1 Chapter Overview

In this chapter the aircraft models used to develop and test the MMAE and Control Redistribution algorithms are presented. First, a brief introduction to the aircraft is provided, together with the justifications for its choice in this research. The following sections provide the details concerning the truth model, design model, and failure models. These models are used with the concepts developed in Chapter 2 to construct and evaluate the Multiple Model Adaptive Estimator and Control Redistribution algorithm.

3.2 The Aircraft

This research is performed on the Variable-Stability In-Flight Simulator Test Aircraft (VISTA) F-16. This aircraft was developed to provide an in-flight simulator for various high performance military aircraft. The control system of the VISTA F-16 can be configured so that the aircraft will emulate the characteristics of other aircraft. This capability provides an extremely flexible tool to develop and evaluate aircraft control systems. This capability also provides a safe and realistic means to train pilots for these systems. The VISTA F-16 was built using a modified F-16D airframe by Calspan and General Dynamics [52].

The principle reason for using the VISTA F-16 in this research is to facilitate the continuation of previous research which has been performed at AFIT [9, 47]. Originally, the primary reason for choosing this platform was based on the availability of an accurate truth model and advanced simulation tools at the Flight Dynamics Directorate of Wright Laboratory.

3.3 Truth Model

In order to evaluate the performance of the MMAE and Control Redistribution algorithm, a *truth model* is needed which will provide an *accurate simulation of the real world environment and a proper statistical portrayal of estimation errors committed by each filter and the MMAE in that "real world" environment* [26]. As mentioned in Section 3.2, the truth model was provided by the Flight Dynamics Directorate at Wright Laboratory.

The VISTA simulation used for the truth model runs as part of the Simulation Rapid-Prototyping Facility (SRF) and incorporates General Dynamics VISTA F-16 simulation software with variable stability flight control system software provided by Calspan. The SRF VISTA F-16 simulation provides a full six-degree-of-freedom truth model which incorporates the aircraft's *nonlinear* equations of motion, advanced actuator modeling, the complete Block 40 controller, and the aileron to rudder interconnect which is used to provide coordinated turns. The Transportable Applications Executive (TAE) provides the interface to allow the user to configure the F-16, select a flight condition, and command the pilot inputs in non-real time [52].

Throughout this sequence of research, several modifications have been made to the FORTRAN source code of the SRF VISTA F-16 simulation [9, 10, 11, 47]. These changes were made to accommodate the multiple model architecture and to provide a more accurate depiction of the *real world* environment. These changes include the incorporation of a more precise wind model based on a zero-order Dryden wind model [9, 10, 11, 31, 32, 44], the incorporation of code which allows the simulation of simultaneous dual failures [9, 10, 11], the incorporation of sensor noise [9, 10, 11], and the replacement of the lateral acceleration measurement computation by a linear model [47].

3.4 Design Model

To generate the Kalman filters for the MMAE, a different, lower dimensioned, system model, called the *design model,* is needed [26]. For this research the design model is a *linear time-invariant, discrete-time* model which is based on the continuous-time model provided by the SRF VISTA simulation. In this section, the development of the *continuous-time* model is presented, followed by its conversion to the equivalent *discrete-time* model.

The equations for this *continuous-time* model can be written as a system of first order differential equations. These equations are often referred to as *state space* equations, and are represented mathematically as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \qquad (3.1)$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control vector, and $\mathbf{w}(t)$ is the white dynamics driving noise. The componants of the vectors $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are provided in Table (3-1):

Vector Component	Symbol	Description	Units
$\mathbf{x}_1$	θ	Pitch Angle	rad
$\mathbf{x}_2$	u	Forward Velocity	ft/sec
$\mathbf{x}_3$	α	Angle of Attack	radians
$\mathbf{x}_4$	q	Pitch Rate	rad/sec
$\mathbf{x}_5$	ϕ	Bank Angle	rad
$\mathbf{x}_6$	β	Sideslip Angle	rad
$\mathbf{x}_7$	p	Roll Rate	rad/sec
$\mathbf{x}_8$	r	Yaw Rate	rad/sec
$\mathbf{u}_1$	δ_e	Elevator Position	rad
$\mathbf{u}_2$	δ_{dt}	Differential Tail Position	rad
$\mathbf{u}_3$	δ_f	Flap Position	rad
$\mathbf{u}_4$	δ_a	Aileron Position	rad
$\mathbf{u}_5$	δ_r	Rudder Position	rad

Table 3-1 State and Input Vectors

Matrices **A** and **B** in Equation (3.1) contain the aircraft's dimensional stability derivatives [40] and **G** is the noise injection matrix. For the VISTA F-16, the matrix **A** is represented as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X'_\theta & X'_u & X'_\alpha & X'_q & 0 & 0 & 0 & 0 \\ Z'_\theta & Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 & 0 \\ M'_\theta & M'_u & M'_\alpha & M'_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \phi_r \\ 0 & 0 & 0 & 0 & Y'_\phi & Y'_\beta & Y'_p & Y'_r \\ 0 & 0 & 0 & 0 & 0 & L'_\beta & L'_p & L'_r \\ 0 & 0 & 0 & 0 & 0 & N'_\beta & N'_p & N'_r \end{bmatrix} \qquad (3.2)$$

Here, the longitudinal derivatives (upper left) are augmented with the lateral derivatives (lower right). The full size variables *X'*, *Y'*, *Z'*, represent the aerodynamic forces in the x, y, and z aircraft body axes, respectively. These *body axes* follow the traditional convention that the directions of the nose, right wing, and bottom of the the aircraft will *always* represent the positive directions of the x, y, and z axes, respectively. Because the axes are said to be *fixed* to the aircraft, these directions will hold true regardless of its position relative to the surface of the earth. The variables *L'*, *M'*, and *N'* represent the moments about the x, y, and z axes, respectively. The prime notation on these variables indicate that the state vector, **x**, includes the angle of attack, α, and sideslip angle β, instead of the velocities v and w. The subscripts on the variables indicate the particular reference variable for each derivative. In the longitudinal direction, θ and q represent the pitch angle and pitch rate respectively, α represents the angle of attack, and u represents the

translational velocity in the x (aircraft body axis) direction. In the lateral direction, ϕ and p represent the roll angle and roll rate respectively, β represents the sideslip angle, and r represents the yaw rate. An example of this notation, entry (2,1) of **A** in Equation (3.2), X'_θ, should be interpreted as the change in aerodynamic force in the aircraft body axis x, resulting from a change in the pitch angle θ.

The entries of the **B** matrix, shown in Equation (3.3), should be interpreted in the same manner as described for the **A** matrix. Here, the changes in force and momentum are in reference to changes in the various control surfaces. In the longitudinal direction, the subscripts δ_e and δ_f represent a change in the elevator and flaps, respectively. In the lateral direction, δ_{dt}, δ_a, and δ_r, represent changes in the differential tail, aileron, and rudder positions, respectively. An example of this notation, entry (2,1) of **B** in Equation (3.3), X'_{δ_e}, should be interpreted as the change in aerodynamic force in the aircraft body axis x, resulting from a change in the elevator position, δ_e.

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ X'_{\delta_e} & 0 & X'_{\delta_f} & 0 & 0 \\ Z'_{\delta_e} & 0 & Z'_{\delta_f} & 0 & 0 \\ M'_{\delta_e} & 0 & M'_{\delta_f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & Y'_{\delta_{dt}} & 0 & Y'_{\delta_a} & Y'_{\delta_r} \\ 0 & L'_{\delta_{dt}} & 0 & L'_{\delta_a} & L'_{\delta_r} \\ 0 & N'_{\delta_{dt}} & 0 & N'_{\delta_a} & N'_{\delta_r} \end{bmatrix} \qquad (3.3)$$

Each of the subscripts in (3.2) and (3.3) is also an indication of the particular component of the state and input vectors **x** and **u** (Table 3-1).

The dynamics driving noise, $\mathbf{w}(t)$, provides the design model with a means to represent uncertainties caused by both wind gusts and inadequacies of the system model. The dynamics driving noise is assumed to be Gaussian and essentially white over the frequency bandwidth of the actual system. The strength of the vector $\mathbf{w}(t)$ is given by the matrix **Q**, the non-zero entries of which are listed below in Table 3-2 [9, 10, 11, 31, 32, 35, 38, 39, 47, 48]:

Q	Parameter	Average Noise Strength
Q(1,1)	u	4.5×10^{-2} *ft²/rad sec*
Q(2,2)	α	3.0×10^{-6} *rad sec*
Q(2,3)	α vs. q	1.1×10^{-8} *rad*
Q(3,3)	q	1.5×10^{-6} *rad/sec*
Q(4,4)	p	6.0×10^{-6} *rad/sec*
Q(5,5)	β	3.0×10^{-6} *rad sec*
Q(5,6)	β vs. r	6.3×10^{-9} *rad*
Q(6,6)	r	2.4×10^{-6} *rad/sec*

Table 3-2 Dynamics Driving Noise Strength

To incorporate the dynamics noise into the continuous-time model, the noise injection matrix, **G**, is used. This matrix is taken directly from previous research [9, 10, 11, 47], and has been modified (fourth and fifth columns interchanged) to reflect the difference in order of β and p in **x** (Table 3-1) and **Q** (Table 3-2).

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ X'_u & X'_\alpha & X'_q & 0 & 0 & 0 \\ Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 \\ M'_u & M'_\alpha & M'_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y'_p & Y'_\beta & Y'_r \\ 0 & 0 & 0 & L'_p & L'_\beta & L'_r \\ 0 & 0 & 0 & N'_p & N'_\beta & N'_r \end{bmatrix} \tag{3.4}$$

This Dynamics noise model was originally derived by Pogoda [31, 32, 44] from the Dryden wind model. This model was adapted for use with the VISTA F-16 by Menke [39, 38], and later incorporated into the SRF VISTA simulation by Eide [9, 10, 11]. Eide was also responsible for decreasing the dynamics noise strength for velocity by two orders of magnitude, to reflect that the original noise models incorporated a heavier turbulence than originally specified.

3.4.1 Input Modifications

While the control surfaces described in Table 3-1 are typical of most aircraft, the VISTA F-16 has some important differences. The VISTA F-16 combines the flap and aileron control surface into a single set of surfaces referred to as the *flaperon.* These surfaces span much of the trailing edge of the wing and can be commanded symmetrically to provide a change in the pitching moment (i.e., as would a typical flap), or differentially to provide a change in the roll moment (i.e., as would a typical aileron). There are also leading edge flaps which are used primarily for take-off and landing. These leading edge flaps have very little control authority, and have not been included in the design model. The entire *stabilizer* portion of the tail can be pivoted to function as the aircraft's *elevator*, and is therefore referred to as the *stabilator.* As with the flaperons, this surface can also be

commanded symmetrically to provide a typical elevator response, or differentially, to produce both a roll and yaw moment. The VISTA F-16 has a single vertical fin with a typical rudder surface to provide yawing moment.

The ability to command the left and right portions of the flaperons and stabilator separately plays a crucial role in the development of the Control Redistribution algorithm discussed in Section 2.4. To exploit this capability, the input vector, **u**, is modified as shown in Table 3-3.

Vector Component	Symbol	Description	Units
u_{mod1}	δ_{ls}	Left Stabilator Position	rad
u_{mod2}	δ_{rs}	Right Stabilator Position	rad
u_{mod3}	δ_{lf}	Left Flaperon Position	rad
u_{mod4}	δ_{rf}	Right Flaperon Position	rad
u_{mod5}	δ_{rud}	Rudder Position	rad

Table 3-3 Modified Input Vector

To accommodate these changes, the input matrix **B** is restructured as follows:

$$\mathbf{B}_{mod} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}X'_{\delta_e} & \frac{1}{2}X'_{\delta_e} & \frac{1}{2}X'_{\delta_f} & \frac{1}{2}X'_{\delta_f} & 0 \\ \frac{1}{2}Z'_{\delta_e} & \frac{1}{2}Z'_{\delta_e} & \frac{1}{2}Z'_{\delta_f} & \frac{1}{2}Z'_{\delta_f} & 0 \\ \frac{1}{2}M'_{\delta_e} & \frac{1}{2}M'_{\delta_e} & \frac{1}{2}M'_{\delta_f} & \frac{1}{2}M'_{\delta_f} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}Y'_{\delta_{dt}} & \frac{1}{2}Y'_{\delta_{dt}} & \frac{1}{2}Y'_{\delta_a} & \frac{1}{2}Y'_{\delta_a} & Y'_{\delta_r} \\ \frac{1}{2}L'_{\delta_{dt}} & \frac{1}{2}L'_{\delta_{dt}} & \frac{1}{2}L'_{\delta_a} & \frac{1}{2}L'_{\delta_a} & L'_{\delta_r} \\ \frac{1}{2}N'_{\delta_{dt}} & \frac{1}{2}N'_{\delta_{dt}} & \frac{1}{2}N'_{\delta_a} & \frac{1}{2}N'_{\delta_a} & N'_{\delta_r} \end{bmatrix} \qquad (3.5)$$

3.4.2 *Measurement and Output Models*

In order to provide estimation of the system's state variables, and proper control of the aircraft, mathematical relationships depicting the relationships of the state variables and control inputs to both the sensor measurements and the outputs are required in the design model. Models which depict these relationships are given in Equations (3.6) and (3.7).

$$\mathbf{z}(t_i) = \mathbf{H}\mathbf{x}(t_i) + \mathbf{D}_z\mathbf{u}(t_i) + \mathbf{v}(t_i) \tag{3.6}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}_y\mathbf{u}(t) \tag{3.7}$$

The measurement model (Eq. (3.6)) depicts the relationship between the sampled-data measurements, $\mathbf{z}(t_i)$, and the state and control vectors, $\mathbf{x}(t_i)$ and $\mathbf{u}(t_i)$, at sample time t_i. This is accomplished through the use of the measurement matrices, $\mathbf{H}$ and $\mathbf{D}_z$ as shown in Equation (3.6). Similarly, the output model (Eq. (3.7)) depicts the relationship between the output variables contained in the vector $\mathbf{y}(t)$, and the state and control vectors, $\mathbf{x}(t)$ and $\mathbf{u}(t)$. This relationship is formed using the output matrices $\mathbf{C}$ and $\mathbf{D}_y$. The particular components of the measurement vector, $\mathbf{z}(t_i)$, are identical to those used in Stepaniak's research [47]. Stepaniak deleted the velocity component of this vector (as formed earlier by Eide [9, 10, 11] and Menke [38, 39]) due to the fact that it is not used by the Block 40 flight control system of the VISTA F-16. The components of the output vector, $\mathbf{y}(t)$, are taken directly from the system state vector, $\mathbf{x}$ (Table 3-1). The components of both the measurement and output vectors are given in Table 3.4:

Vector Component	Symbol	Description	Units
$\mathbf{z}_1$	α	Angle of Attack	rad
$\mathbf{z}_2$	q	Pitch Rate	rad/sec
$\mathbf{z}_3$	a_n	Normal Acceleration	g's
$\mathbf{z}_4$	p	Roll Rate	rad/sec
$\mathbf{z}_5$	r	Yaw Rate	rad/sec
$\mathbf{z}_6$	a_y	Lateral Acceleration	g's
$\mathbf{y}_1$	θ	Pitch Angle	radians
$\mathbf{y}_2$	u	Forward Velocity	ft/sec
$\mathbf{y}_3$	α	Angle of Attack	radians
$\mathbf{y}_4$	q	Pitch Rate	rad/sec
$\mathbf{y}_5$	ϕ	Bank Angle	rad
$\mathbf{y}_6$	β	Sideslip Angle	rad
$\mathbf{y}_7$	p	Roll Rate	rad/sec
$\mathbf{y}_8$	r	Yaw Rate	rad/sec

Table 3-4 Measurement and Output Vectors

A comparison between Table 3-4 and Table 3-1 reveals that several of the measurements and all of the output variables are available as states in the design model depicted by Equation (3.1). The measurement variables not available as states, i.e., the normal and lateral accelerations (a_n and a_y), must be appropriately modeled. It should be noted that these accelerations are expressed in g's (the ratio of acceleration caused by aerodynamic forces to the acceleration caused by gravity) felt by the pilot.

The acceleration models are developed by modifying the equations for acceleration at the center of gravity [6] to include only aerodynamic forces (i.e., remove the forces due to gravity) and translated from the center of gravity to the pilot's station where the actual sensors are located. The resulting equations are divided by the acceleration due to gravity so that the results will be in terms of g's. The final equations are provided in Equations (3.8) and (3.9):

$$a_n = -\frac{u}{g}(\dot{\alpha} - q) + \frac{l_x}{g}\dot{q} \tag{3.8}$$

$$a_y = -\frac{u}{g}\left(\dot{\beta} + r\right) - \phi + \frac{l_x}{g}\dot{r} \tag{3.9}$$

where g represents the acceleration due to gravity and l_x is the distance (in the x direction) from the center of gravity to the pilot's station. The SRF neglects the small contribution of l_z, and calculates l_x to be 14.219 *ft* using the equation:

$$l_x = 9.988 + \bar{c} \cdot x_{cg} \cdot .01 \tag{3.10}$$

where $\bar{c}$ is the mean aerodynamic chord (11.32 *ft*) and x_{cg} is the *average* location of the center of gravity along the x axis (37.376 *ft*). Attempts by Stepaniak [47] to verify these acceleration models show differences between the linear design model and the nonlinear truth model of the SRF. The normal acceleration, a_n, is somewhat biased in comparison to the nonlinear truth model. This bias is attributed to first order approximations in resolving the gravity contributions. Due to the slowly varying nature of α in this research, this bias is deemed to have a negligible effect on the perturbation states. The lateral acceleration, a_y, shows significant difference in both phase and magnitude. Due to the inability to resolve these differences through the incorporation of pseudonoise (acknowledging a poor sensor reading), and the questionable results from the *nonlinear* truth model, the *linear* model for a_y has been incorporated into the truth model, as done by Stepaniak [47].

3.4.3 Actuator Models

In order to provide a more realistic (and plausible) design model, both Eide [9, 10, 11] and Stepaniak [47] incorporated models of the system's actuators. The actuator model used is a first order lag which has a break point at $\omega = 14$, and is given by Equation (3.11):

$$\dot{\delta}_{act} = -14\delta_{act} + 14\delta_{cmd} \tag{3.11}$$

This model is based on the fourth order servo-actuator model which is given for *all* of the actuators in the Block 40 functional diagram [13]:

$$\frac{\delta_{act}}{\delta_{cmd}} = \frac{(20.2)(144.8)(71.4)^2}{(s+20.2)(s+144.8)(s^2+2(0.736)(71.4)s+71.4^2)} \tag{3.12}$$

Note that in Equation (3.11) the break point of 14 is chosen instead of the value 20.2 as shown in Equation (3.12). This choice is based upon empirical results from Eide's research [9] which shows that the lower value provides a better match to the SRF VISTA actuator simulation.

To reflect the difference between the commanded input, $\boldsymbol{u}$, and the *resulting* actuator positions, the actuator positions are augmented to the state vector, $\mathbf{x}$, as shown in Table 3-5:

Vector Component	Symbol	Description	Units
$\mathbf{x}_{aug1}$	θ	Pitch Angle	rad
$\mathbf{x}_{aug2}$	u	Forward Velocity	ft/sec
$\mathbf{x}_{aug3}$	α	Angle of Attack	rad
$\mathbf{x}_{aug4}$	q	Pitch Rate	rad/sec
$\mathbf{x}_{aug5}$	ϕ	Bank Angle	rad
$\mathbf{x}_{aug6}$	β	Sideslip Angle	rad
$\mathbf{x}_{aug7}$	p	Roll Rate	rad/sec
$\mathbf{x}_{aug8}$	r	Yaw Rate	rad/sec
$\mathbf{x}_{aug9}$	δ_{ls}	Left Stabilator Position	rad
$\mathbf{x}_{aug10}$	δ_{rs}	Right Stabilator Position	rad
$\mathbf{x}_{aug11}$	δ_{lf}	Left Flaperon Position	rad
$\mathbf{x}_{aug12}$	δ_{rf}	Right Flaperon Position	rad
$\mathbf{x}_{aug13}$	δ_{rud}	Rudder Position	rad
$\mathbf{u}_{aug1}$	$\delta_{ls\ comm}$	Left Stabilator Command	rad
$\mathbf{u}_{aug2}$	$\delta_{rs\ comm}$	Right Stabilator Command	rad
$\mathbf{u}_{aug3}$	$\delta_{lf\ comm}$	Left Flaperon Command	rad
$\mathbf{u}_{aug4}$	$\delta_{rf\ comm}$	Right Flaperon Command	rad
$\mathbf{u}_{aug5}$	$\delta_{rud\ comm}$	Rudder Command	rad

Table 3-5 Augmented State and Input Vectors

Note that the command input vector, **u**, has not actually been augmented; rather, only its description was changed to reflect the fact that it is truly a command rather than the actual position as depicted in Table 3-3. With this *augmented* state vector, Equation (3.11) can now be incorporated into the design model by augmenting the matrices of the state space equation as shown in Equation (3.13). In a similar manner, the matrices of the measurement and output models are appropriately augmented to form Equations (3.14) and (3.15). Note that augmentation to the output equation can be easily utilized to extract desired actuator position data from any of the newly augmented states, specifically $\mathbf{x}_{aug9}$ through $\mathbf{x}_{aug13}$.

$$\dot{\mathbf{x}}_{\text{aug}}(t) = \begin{bmatrix} \mathbf{A} & \mathbf{B}_{\text{mod}} \\ \mathbf{0} & -14 \cdot \mathbf{I} \end{bmatrix} \cdot \mathbf{x}_{\text{aug}}(t) + \begin{bmatrix} \mathbf{0} \\ 14 \cdot \mathbf{I} \end{bmatrix} \cdot \mathbf{u}_{\text{aug}}(t) + \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \mathbf{w}(t) \qquad (3\text{-}13)$$

$$\mathbf{z}(\mathrm{t_i}) = \begin{bmatrix} \mathbf{H} & \mathbf{D}_z \end{bmatrix} \cdot \mathbf{x}_{\text{aug}}(t_i) + \mathbf{v}(\mathrm{t_i}) \qquad (3\text{-}14)$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{C} & \mathbf{D}_y \end{bmatrix} \cdot \mathbf{x}_{aug}(t) \qquad (3\text{-}15)$$

These augmented matrices will be referred to as $\mathbf{A}_{aug}$, $\mathbf{B}_{aug}$, $\mathbf{G}_{aug}$ for Equation (3.13), $\mathbf{H}_{aug}$ for Equation (3-14), and $\mathbf{C}_{aug}$ for Equation (3.15).

3.4.4 Model Discretization

As with many estimation problems, the MMAE used in this research will be implemented on a digital computer. This results in the need either to discretize the results obtained from the continuous-time filters, based on the continuous-time design model, or to determine the equivalent *discrete-time* design model and then generate the discrete-time filters. Maybeck [26] shows that the first approach provides only an *approximation* to the optimal discrete-time filters, and that this approach is considerably more difficult to implement (requiring the integration of Riccati differential equations). The second approach, which has been implemented for this research, involves no such approximations, and requires recursion formulas which behave significantly better than those required by the first approach.

The complete development of the equivalent discrete-time design model is given in Maybeck [26] and results in the following equations:

$$\mathbf{x}_{aug}(t_{i+1}) = \mathbf{\Phi}\mathbf{x}_{aug}(t_i) + \mathbf{B}_d\mathbf{u}_{aug} + \mathbf{w}_d(t_i) \qquad (3\text{-}16)$$

$$\mathbf{z}(t_i) = \mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \qquad (3\text{-}17)$$

$$\mathbf{y}(t_i) = \mathbf{C}_{aug}\mathbf{x}_{aug}(t_i) \qquad (3.18)$$

where the state transition matrix, $\mathbf{\Phi}$, and the matrix $\mathbf{B}_d$ are given by:

$$\mathbf{\Phi} = e^{\mathbf{A}_{aug}\Delta T} \qquad (3.19)$$

$$\mathbf{B}_d = \left(\int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau} d\tau\right) \cdot \mathbf{B}_{aug} \qquad (3.20)$$

where ΔT is the sample period. The appropriate *discrete-time* white noise has the following statistics:

$$E\{\mathbf{w}_d(t_i)\} = \mathbf{0} \qquad (3.21)$$

$$E\{\mathbf{w}_d(t_i)\,\mathbf{w}_d^T(t_i)\} = \mathbf{Q}_d = \int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau}\mathbf{G}\mathbf{Q}\mathbf{G}^T e^{\mathbf{A}_{aug}^T\tau} d\tau \qquad (3.22)$$

$$E\{\mathbf{w}_d(t_i)\,\mathbf{w}_d^T(t_j)\} = \mathbf{0}, \quad t_i \neq t_j \qquad (3.23)$$

3.5 Failure Models

In order to design filters for the MMAE which correctly hypothesize failures of actuators and/or sensors, models of these modes have to be incorporated into the design model. To accomplish this, the linear design model is once again modified to incorporate a failure matrix ($\mathbf{F}_{ai}$ for the i^{th} actuator, and $\mathbf{F}_{sj}$ for the j^{th} sensor) which can be used to multiply the appropriate column of the input matrix $\mathbf{B}_{aug}$ (for actuator failures) or the appropriate row of the measurement matrix $\mathbf{H}_{aug}$ (for sensor failures) by a scalar, ε, which can range between 0 (representing a complete failure), and 1 (representing a fully functional component). The failure modes are incorporated into the *continuous-time* dynamics model, *before* the conversion to discrete time, to reflect the fact that the failures are inserted into the *continuous-time* truth model. Once again, the approach of generating an equivalent discrete time model leads to the design of *optimum* Kalman filters for the MMAE, rather than approximations of the same. The resulting dynamics and measurement models are as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_{aug}\mathbf{x}(t) + \mathbf{B}_{aug}\mathbf{F}_{ai}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \qquad (3.24)$$

$$\mathbf{z}(t_i) = \mathbf{F}_{sj}\mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \qquad (3.25)$$

Both of the failure matrices, $\mathbf{F}_{ai}$ and $\mathbf{F}_{sj}$, will be appropriately dimensioned identity matrices with the exception that locations representing failed components will be $0 \leq \varepsilon < 1$, as discussed above. Note that even when an actuator or sensor has completely failed ($\varepsilon = 0$), the dynamics driving noise and sensor noise are still incorporated into the model.

Comparison of the measurement model depicted in Equation (3.25) with the measurement model of Equation (3.6) reveals the absence of the measurement matrix and control input term $\mathbf{D}_z\mathbf{u}(t_i)$. This is due the change which occurred in Section 3.4.3 where the state vector $\mathbf{x}$ was augmented to include the actual actuator positions and $\mathbf{H}$ was augmented by $\mathbf{D}_z$ to form $\mathbf{H}_{aug}$ as depicted in Equation (3.14).

As in previous research [9, 10, 11, 47], the actuator failures are incorporated into the truth model by commanding them to their trim position. This technique accomplishes two objectives. First, it approximates the failure as an *actuator failure to free stream*, in which the surface simply floats in the relative wind (as accomplished by well designed actuators undergoing failures). Second, this approach is a good approximation to that used in the design model, where the *input commanded perturbation position* is zeroed out, leaving the *nominal* position unaffected, i.e. the failed position of the actuator on the design model will indeed be the trim position.

In addition to reaccomplishing the complete, single-failure analysis which was performed by Stepaniak [47], this research also examines the algorithm's ability to detect and compensate for multiple failures in the system. This multiple failure analysis utilizes the same failure models as for the single failure research (i.e., actuators and sensors which have fully failed as described above). Other failure modes, such as biased sensors, stuck actuators, and others are also of great importance and should be seriously considered in future research.

3.6 Chapter Summary

This chapter has presented a detailed overview of the various mathematical models which are used to generate and test the MMAE with Control Redistribution algorithm. The

details concerning the VISTA F-16 were presented, along with the rationale for selecting this aircraft for this research. The various aspects of the SRF were discussed in relation to the generation of the truth model and components necessary for the design model. The design model was presented in various stages, starting with a simplified, linear, continuous-time, state space model, and incorporating several necessary modifications which resulted in an equivalent *discrete-time* model which incorporates sensor measurements, appropriate outputs, actuator models, and failure models. In the next chapter, the results are presented of the extensive testing which has been performed using the models developed in this chapter.

4. Simulation Results

4.1 Chapter Overview

In this chapter the results of extensive testing of the Multiple Model Adaptive Estimation with Control Redistribution (MMAE/CR) algorithm are presented. To establish the validity of the Control Redistribution algorithm, and to introduce the reader to the simulation and data presentation techniques used throughout the remainder of the chapter, Stepaniak's *single* failure analysis [47] are recreated and presented. This data is compared to the case of using the same Multiple Model Adaptive Estimator with *no* Control Redistribution.

The remainder of this chapter is an analysis of the algorithm's ability to detect and compensate for *double* failures of both actuators and sensors. As a baseline, the abilities of the Multiple Model architecture *without* Control Redistribution are presented. Since Stepaniak [47] developed an improved method for dithering the actuators, a quick comparison is made in this section to the double failure analysis of previous work accomplished by Eide [9, 10, 11] prior to the incorporation of improved dithering. To provide adequate double failure detection and compensation, several modifications were made to the Control Redistribution algorithms. These modifications were the result of extensive testing and analysis, all of which is presented, along with the rationale for any changes which may have been required to improve the algorithm's performance.

4.2 Single Failure Analysis

In an effort to develop a firm understanding SRF, FORTRAN and Matlab software [41, 24, 52] required to perform a simulation with the VISTA F-16, the single-failure results of the VISTA simulation with Multiple Model Adaptive Estimation (MMAE) and Control Redistribution as presented in Stepaniak's thesis [47], were recreated. These results are presented in this chapter to establish the validity of the Control Redistribution algorithm and to introduce the reader to the techniques used to present the data resulting from each simulation.

4.2.1 Probability Plots

As discussed in Chapter 2, the MMAE incorporates a Conditional Hypothesis Probability Evaluator which utilizes the residual information from each Kalman filter to assign an appropriate *probability of correctness* weighting to each output state estimate. The MMAE utilizes these probability weightings in two ways. First, the weightings are used as depicted in Figure 2-1, to combine the state estimates of the Kalman filters in a manner which allows the finite discretization of the possible failure scenarios. Second, the MMAE algorithm utilizes the probability calculation to *declare* any hard failures which may occur. These declarations are the key to the moving bank structure depicted in Figure 2-4. Once the probability weighting of a given filter, based on a particular failure (a_k), has surpassed an established threshold for a certain amount of time, the failure a_k is declared. Several values of threshold levels and times were tested, the final result being a level of .95 and a time of 10 sample periods (.156 seconds). These threshold levels were incorporated in such a manner as to permit their easy adjustment for future research. Once the first failure has been declared, a new bank of filters is implemented. This new bank consists of the filter depicting the first failure (with no second failure), a filter based on the fully functional aircraft (to permit a return to the first bank), and 10 additional filters each based

on the hypothesis of the first failure in combination with one additional failure. Because of this, and the fact that the Control Redistribution algorithm utilizes these failure declarations to determine the proper redistribution matrices, plots of the probability allocations for each filter are essential to the analysis of the system.

To remain consistent with previous research [9, 38, 47], all simulations are the result of 10 Monte Carlo runs, each of which is conducted for a length of 8 seconds and each of which is based on a particular failure scenario. The results of each simulation are presented by showing the mean ± one standard deviation of the probability of each filter in the active bank of the MMAE. A typical example of such a probability plot is shown in Figure 4-1. Here it can be seen that a probability *subplot* is provided for a filter based on no failure (a fully functional aircraft) as well as for each of the eleven filters which are based on hypotheses of a different single failure scenario. Recall that the eleven possible single failures are each based on either one of the five actuators, or one of the 6 sensors having failed. The particular failure depicted in each subplot is abbreviated on the Y-axis of each subplot, and defined in Table 4-1.

Abbreviation	Hypothesis	Abbreviation	Hypothesis
ff	Fully Functional	aoa	Angle-of-Attack Sensor
ls	Left Stabilator Failure	q	Pitch Rate Sensor
rs	Right Stabilator Failure	a_n	Normal Acceleration Sensor
lf	Left Flaperon Failure	p	Roll Rate Sensor
rf	Right Flaperon Failure	r	Yaw Rate Sensor
rud	Rudder Failure	a_y	Lateral Acceleration Sensor

Table 4-1 Failure Abbreviations

Noting that the simulated failure of the rudder was inserted into the truth model at three seconds into the simulation, a thorough comprehension of the information presented in Figure 4-1 is easily obtained. Before the three-second insertion point, the MMAE is correctly depicting the fully functional hypothesis as correct. At the three-second insertion

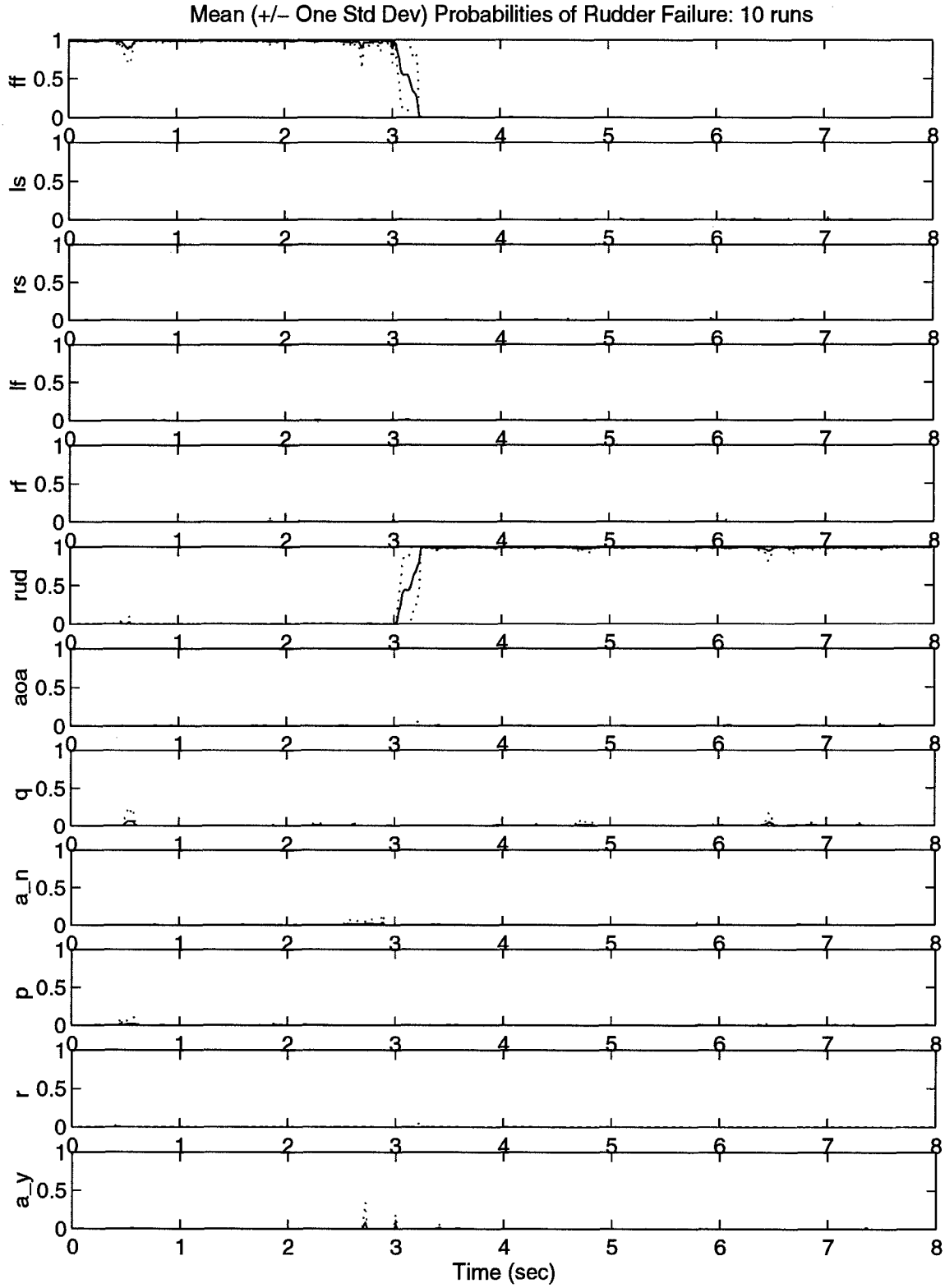

Figure 4-1 Probability Plot for Failed Rudder

point, probability correctly shifts from the fully functional hypothesis to that of the failed rudder. At this point the truth model ceases to change for the remainder of the 8-second simulation. During this time, the MMAE correctly maintains a lock on the correct hypothesis (rs) and only small variations (possibly due to unexpected strong wind gusts) are shown.

It is interesting to note (but not surprising) that the majority of information portrayed in Figure 4-1 is contained in the two subplots depicting the fully functional aircraft (ff) and that of the actual simulated failure (rudder - rud). Because of extensive Kalman filter tuning, truth model adjustments, and dither enhancements performed during previous research [9, 38, 47], it will be assumed that the majority of the simulations will show this characteristic of a *well tuned MMAE*, i.e. where the majority of the probability (.95 or greater) is contained in either the fully functional hypothesis or the correct failure hypothesis. Based on this assumption, a consolidation of the probability plots is formed, in which only the subplot of the particular failure scenario being simulated is provided. An example of such a consolidated, or *summary*, plot is shown in Figure 4-2. This summary plot shows the result of each single failure simulation (10 Monte Carlo runs each). This summary plot technique is used throughout the remainder of the thesis, with a slight modification in the double failure case. At any point where the particular subplot decreases substantially from its lock (probability close to 1), a more detailed analysis similar to Figure 4-1 can be performed.

Analysis of Figure 4-2 clearly shows that Stepaniak's [47] MMAE with Control Redistribution does an excellent job of detecting and locking onto single failures. Note that this analysis was performed without the use of bank swapping. The method of bank swapping, which is incorporated for double failures, actually transfers the filter used to detect the first failure into the second bank, giving it the same excellent detection and lock capabilities shown here. Figure 4-3 is the result of Eide's [9] results for the same single

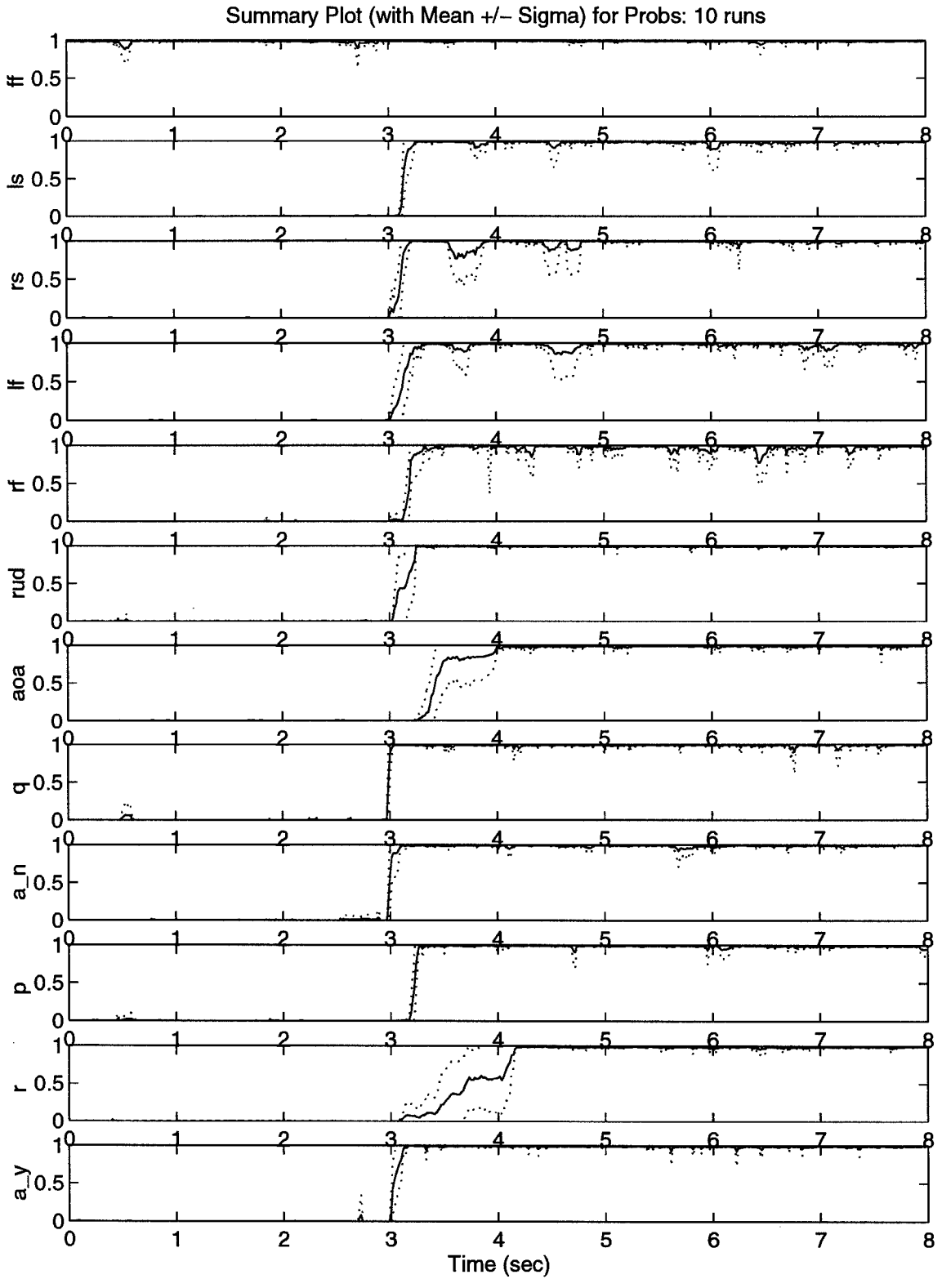

Figure 4-2 Single Failure Probability Summary Plot - Stepaniak [47]

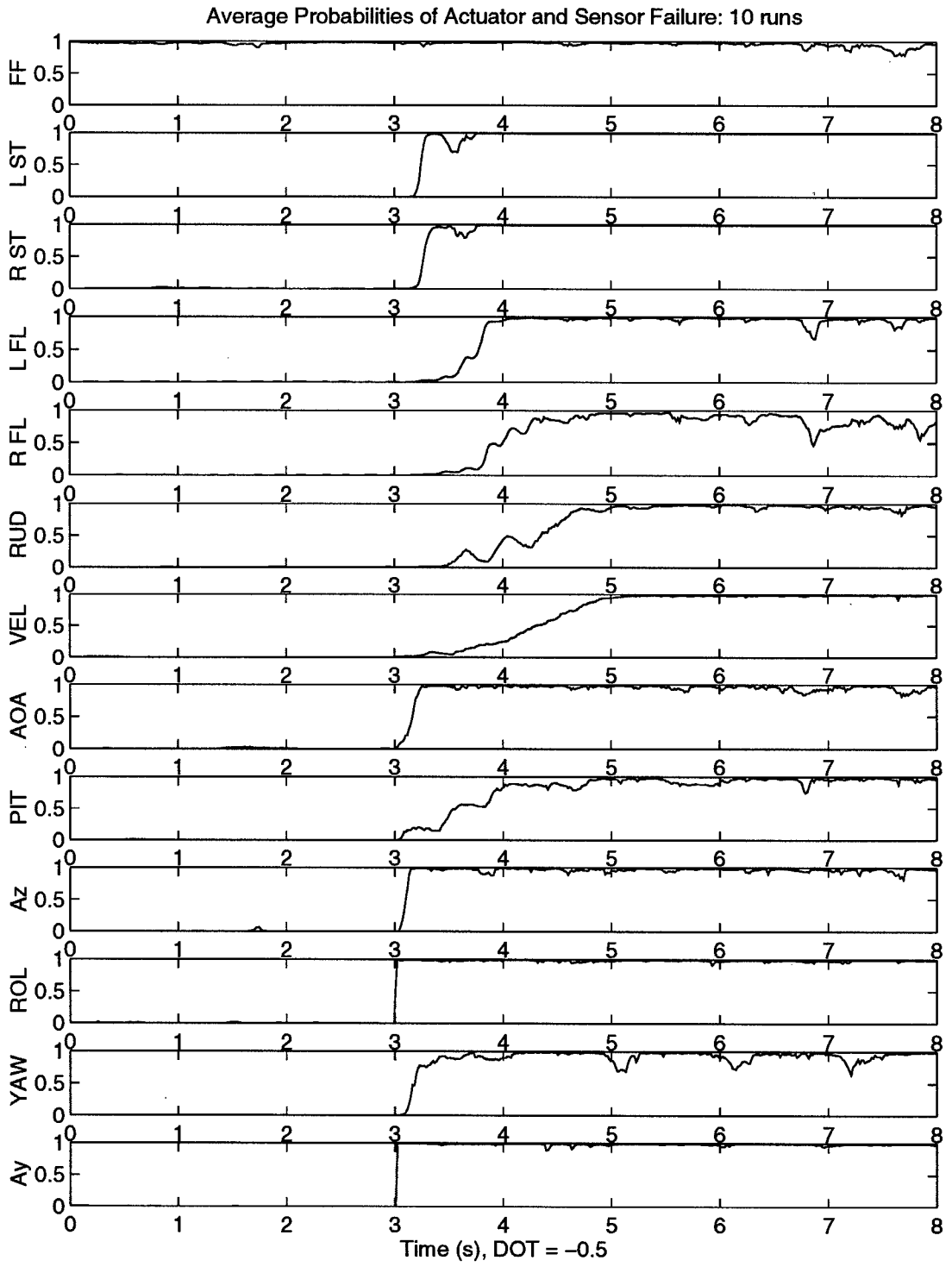

Figure 4-3 Single Failure Probability Summary Plot - Eide [9]

failure analysis. Comparison of these two results clearly demonstrates an improvement in the ability of Stepaniak's algorithm to detect a failure quickly, and remain locked onto that hypothesis despite variations in strong wind gusts. Careful comparison of Figures 4-2 and 4-3 reveals that the velocity sensor failure hypothesis was dropped by Stepaniak. This change was due to the fact that velocity, u, is not used in the Block 40 flight control system.

Because Control Redistribution is only incorporated *after* a given actuator failure has been declared, it is easy to understand that these improvements obtained by Stepaniak [47] must be the result of other changes (*not* Control Redistribution). The principle change responsible for this enhancement involves the aircraft's dither signal. Stepaniak noted that the dither scheme utilized by Eide [9] on the *nonlinear* VISTA truth model was actually developed by Menke [38] for a *linear* model. This problem was compounded by the fact that a dither signal for either of these models cannot be independently applied to each actuator, but rather must be the result of combined inputs to the three dynamic channels of the aircraft (i.e., longitudinal, lateral, and directional). These inputs are applied via the pitch stick, roll stick, and pedal inputs, respectively, and are summarized in Table 4-2. Each of these inputs must command a unique combination of control surfaces in order to achieve the desired commanded response and avoid problems such as adverse yaw. While the

Dynamic Channel	Associated Input	Command Issued	Frequency (rad/sec)	Magnitude (lbs)	Phase (deg)
Longitudinal	Pitch Stick	Normal Acceleration	15	+12/-12.5	0
Lateral	Roll Stick	Roll Rate	15	11	180
Directional	Pedals	Sideslip Angle	15	30	0

Table 4-2 Dither Signals Used by Menke[38] and Eide [9]

signals in Table 4-2 were chosen in an attempt to replicate certain waveforms on each actuator (i.e., to enhance identification of failures), Stapaniak discovered that they actually interfered with each other in a manner which seriously limited the detection of certain failures. To illustrate this *constructive interference,* Stepaniak showed the actual commanded actuator positions resulting from this dither. These plots have been reproduced and are shown below in Figure 4-4. While the offset bias shown in this figure was produced intentionally to keep the aircraft in a trimmed flight condition, the reduced magnitude on the right stabilator and rudder are clearly unintentional. This reduced dither signal (resulting from constructive interference) directly contributed to the failure detection problems encountered by both Menke and Eide.

In an effort to remedy this situation, Stepaniak [47] made three distinct changes to the dither scheme. First, the 180° phase shift was removed from the lateral dither. This

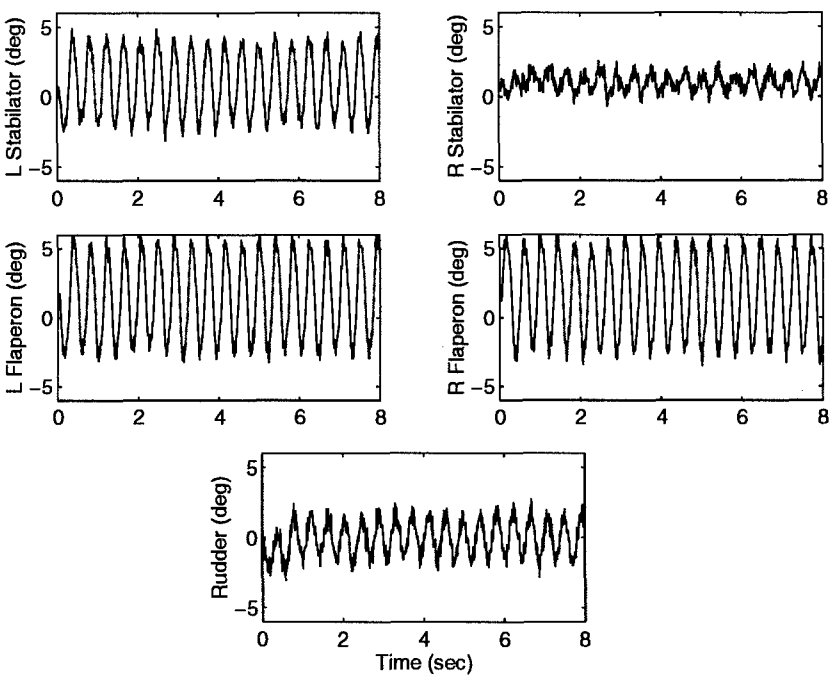

Figure 4-4 Commanded Actuator Signals Resulting Form Menke/Eide Dither Signal[47]

was incorporated based on empirical evidence which showed it resulted in improved rudder excitation. Second, a 90° phase shift was placed on the longitudinal channel. This change was incorporated to achieve identical peak magnitudes between the left and right stabilators. Third, the frequency of the longitudinal channel was reduced by half. This reduction had the effect of increasing the range of motion to increase failure detection, without increasing the magnitude of the actual signal. These signals are summarized in Table 4-3 and the resulting waveforms are depicted in Figure 4-5.

Dynamic Channel	Associated Input	Command Issued	Frequency (rad/sec)	Magnitude (lbs)	Phase (deg)
Longitudinal	Pitch Stick	Normal Acceleration	7.5	+12/-12.4	90
Lateral	Roll Stick	Roll Rate	15	11	0
Directional	Pedals	Sideslip Angle	15	30	0

Table 4-3 Modified Dither Signals Used Developed by Stepaniak [47]

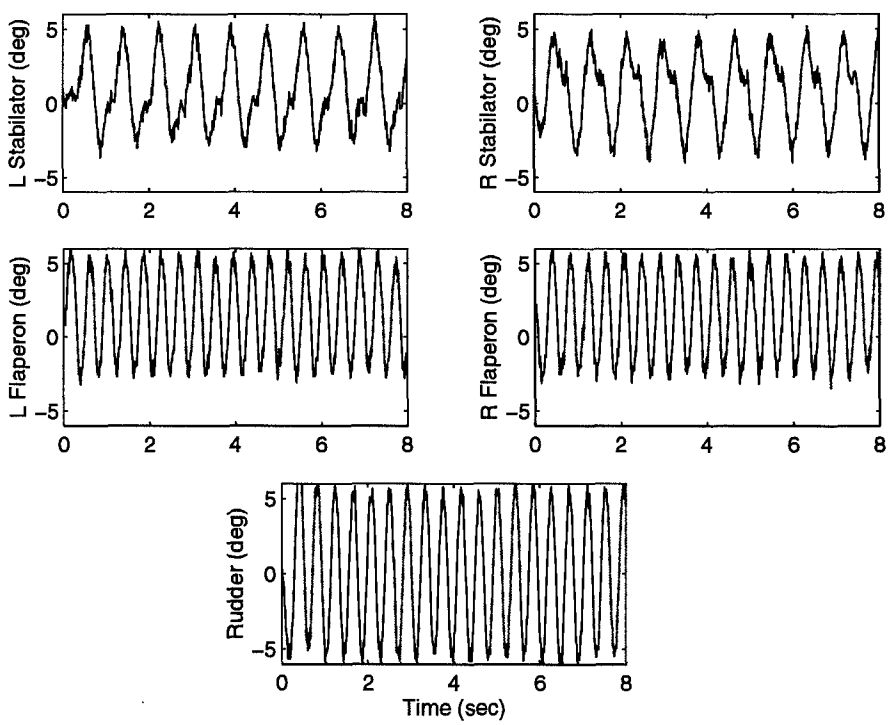

Figure 4-5 Commanded Actuator Positions Resulting From Modified Dither

Stepaniak [47] noted that, while this modified dither did improve failure detection (compare Figs. 4-2 and 4-3), it also resulted in a few new dilemmas. First, the lower frequency of the dither might lead one to question the true subliminality of the signal to the pilot. While the resulting normal and lateral accelerations were shown to be within the specified tolerance of $\pm.2$ and $\pm.1$ g's, respectively, these specified limits were reportedly hard to obtain, and a more in-depth research of the subject would need to be executed before implementation is considered. Second, the question of surpassing the rate limits of the actuators arises. The rates induced by this modified dither were shown by Stepaniak to be barely within the limits imposed by the actuators. Conceivably, this near-saturation condition could limit the amount of remaining control authority left to the pilot. However, the nature of the dither signal requires it to be utilized only during a steady level flight condition, when the absence of control input from the pilot might obscure an actuator or sensor failure. This being the case, a software switch can easily be (and has been) implemented which disables the dither signal during an actual pilot commanded input. This lack of dither will be apparent in the plots of the next section. The rate saturation could pose a problem if the proposed Control Redistribution algorithm is employed. As explained in Section 2.4, Control Redistribution utilizes the remaining "healthy" actuators to compensate for those which have failed. This results in an actuator having an additional burden which, in-turn, may result in rate saturation. During single actuator failures, this was not a serious problem, but when the failure of two actuators was tested it became obvious (though not immediately) that the actuators were indeed operating in a nonlinear region. This resulted in severe degradation of the Kalman filters' ability to estimate the values of the state variables accurately, making the MMAE useless for determining the correct failure condition. This problem was addressed by scaling the amount of additional command sent to the actuators, and will be discussed in more detail in Section 4.3.

The modified dither signal also created some drift in the trimmed flight condition for which the Kalman Filters had been precisely tuned. This problem is also further addressed in Section 4.3.

4.2.2 Output Plots

Once the given algorithm has shown its ability to detect failures with the probability plots, the true benefits of Control Redistribution can be analyzed via the output plots. An example of such an output plot is presented in Figure 4-6. Unlike the probability plots in which the results of 12 separate failure scenarios are consolidated into a single set of subplots, each output plot will represent a single failure scenario. Each plot will contain 15 subplots, each representing one of the various output variables of the aircraft. Table 4-4 gives a breakdown of each output plotted. The output plots are conveniently laid out into three distinct columns. The first column contains the longitudinal states and normal acceleration (a_n), the second column shows the lateral states and lateral acceleration (a_y), and the third column shows the achieved (vs. commanded) actuator angles. Figure 4-6 is the result of simulating the MMAE-based Block 40 Flight Control System (FCS) *without* Control Redistribution algorithm. As with Stepaniak's research [47], the response of the fully functional aircraft in this simulation is used as a baseline to compare the remaining failure responses. During this particular simulation a yaw doublet has been applied between 2 and 3 seconds. This commanded yaw doublet is incorporated via the SRF and consists of a one second yaw doublet of ±54.9 pounds. Three separate simulations are performed, each with a different commanded input doublet. The magnitude of the each doublet (yaw, pitch, and roll) is based on 90% of maximum value. This value was set for each doublet by determining the magnitude boundaries over which the MMAE-based controller could maintain lock on the correct (fully functional) hypothesis. A summary of each doublet and their respective magnitudes are provided in Table 4-5.

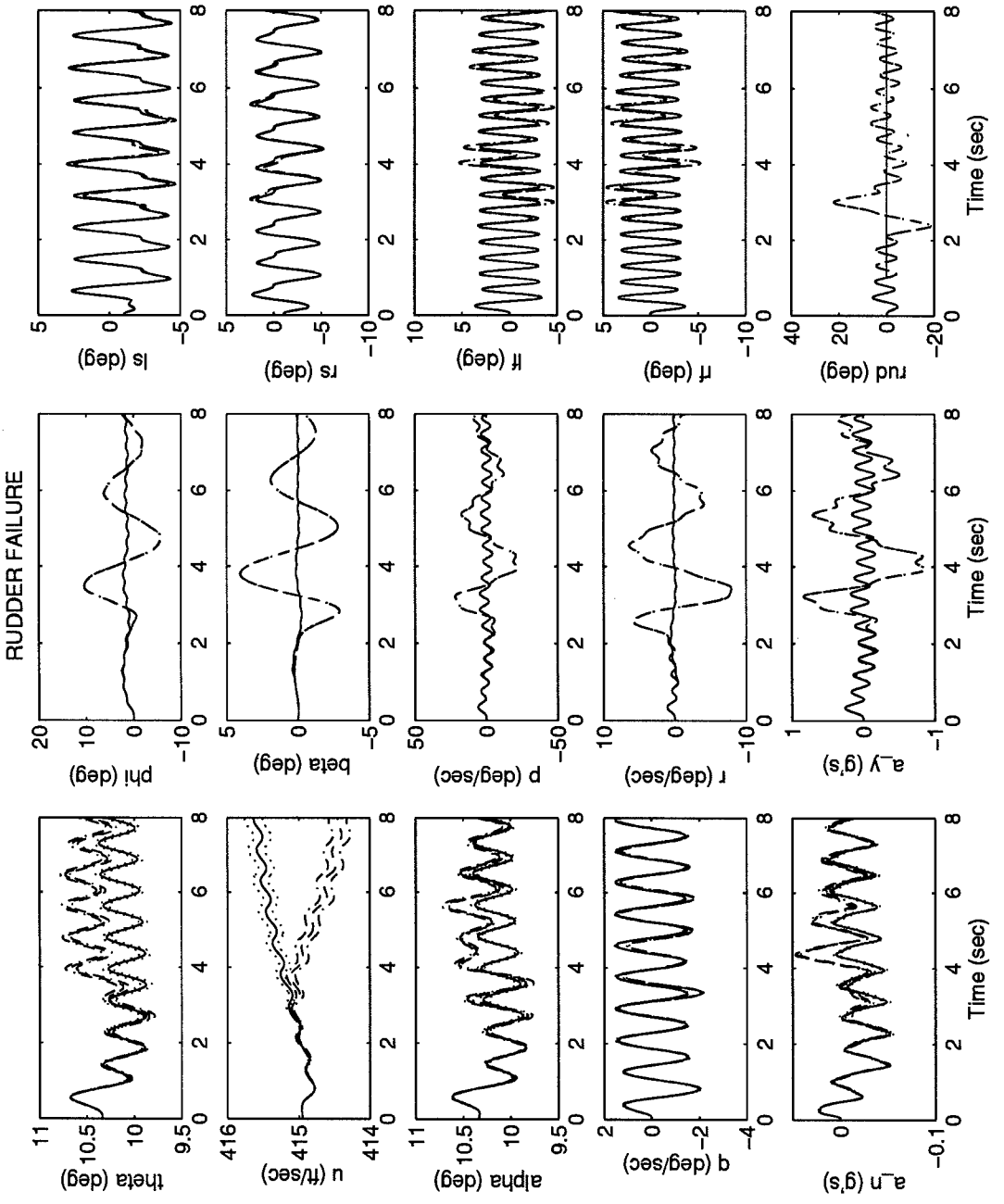

Figure 4-6 Output Plot - Yaw Doublet Input - Rudder Failure - Block 40 with MMAE (No Control Redistribution)

Symbol	Y-Axis	Description	Units
θ	theta	Pitch Angle	degrees
u	u	Forward Velocity	ft/sec
α	alpha	Angle of Attack	degrees
q	q	Pitch Rate	deg/sec
a_n	a_n	Normal Acceleration	g's
ϕ	phi	Bank Angle	degrees
β	beta	Sideslip Angle	degrees
p	p	Roll Rate	deg/sec
r	r	Yaw Rate	deg/sec
a_y	a_y	Lateral Acceleration	g's
δ_{ls}	ls	Left Stabilator Position	degrees
δ_{rs}	rs	Right Stabilator Position	degrees
δ_{lf}	lf	Left Flaperon Position	degrees
δ_{rf}	rf	Right Flaperon Position	degrees
δ_{rud}	rud	Rudder Position	degrees

Table 4-4 Output Plot Definitions

Dynamic Channel	Associated Input	Command Magnitude	Commanded Variable	Converted Magnitude
Longitudinal	Pitch Stick	9 lb	a_n	1.64231 g's
Lateral	Roll Stick	2.7 lb	p	8.50 deg/sec
Directional	Pedals	54.9 lb	β	15.578 deg

Table 4-5 Test Doublets

Each of the subplots in Figure 4-6 shows the response (mean ± one standard deviation) of the fully functional aircraft (dashed/dashed-dotted) as well as that of the impaired aircraft (solid line/dotted). An output plot for each failure scenario is generated to examine the effects of the particular commanded input (in this case a yaw doublet). To complete the single failure analysis output plots were also generated for the pitch and roll doublets. Again each of these inputs results in 12 separate plots, each containing 15 separate subplots. All of these plots have been included in Appendix A. Figure 4-6 was

chosen to represent this analysis because it clearly demonstrates the effects of the given failure. In this particular example, the rudder actuator has failed at 1 second, as indicated by the "rud" subplot. A quick comparison between the fully-functional aircraft response and the failed-rudder aircraft response (specifically the subplot of a_y) clearly shows the lack of command authority in this failure scenario.

Figure 4-7 shows the identical failure analysis with the exception that Control Redistribution is now being implemented. Note that, while the response of the lateral acceleration is not exactly the same as for the fully functional case, there is a significant amount of lateral control authority. The reader should note that this control is not the result of the rudder, as can be seen by the 'rud' subplot, but rather the result of the Control Redistribution algorithm which reroutes the rudder commands to the remaining actuators. This is evident in both the stabilator and flaperon subplots, where both sets of control surfaces are now being utilized in differential manner to obtain the desired command in the lateral channel. These figures clearly indicate that Control Redistribution can provide greatly enhanced control authority during the single failure of a control surface actuator. While this is especially evident during the rudder failure scenario (this failure provides the most challenging failure scenario because of the lack of control redundancy with the rudder), similar conclusions can also be derived from the results of the remaining single failure scenarios included in Appendix A.

Appendix A has been separated into sections containing output plots from each doublet input (pitch, roll, and yaw). Each section includes a complete set of plots from the basis simulation (Block 40, no Control Redistribution), and a set of plots from a simulation which employed Control Redistribution. The analysis of each doublet set is provided below.

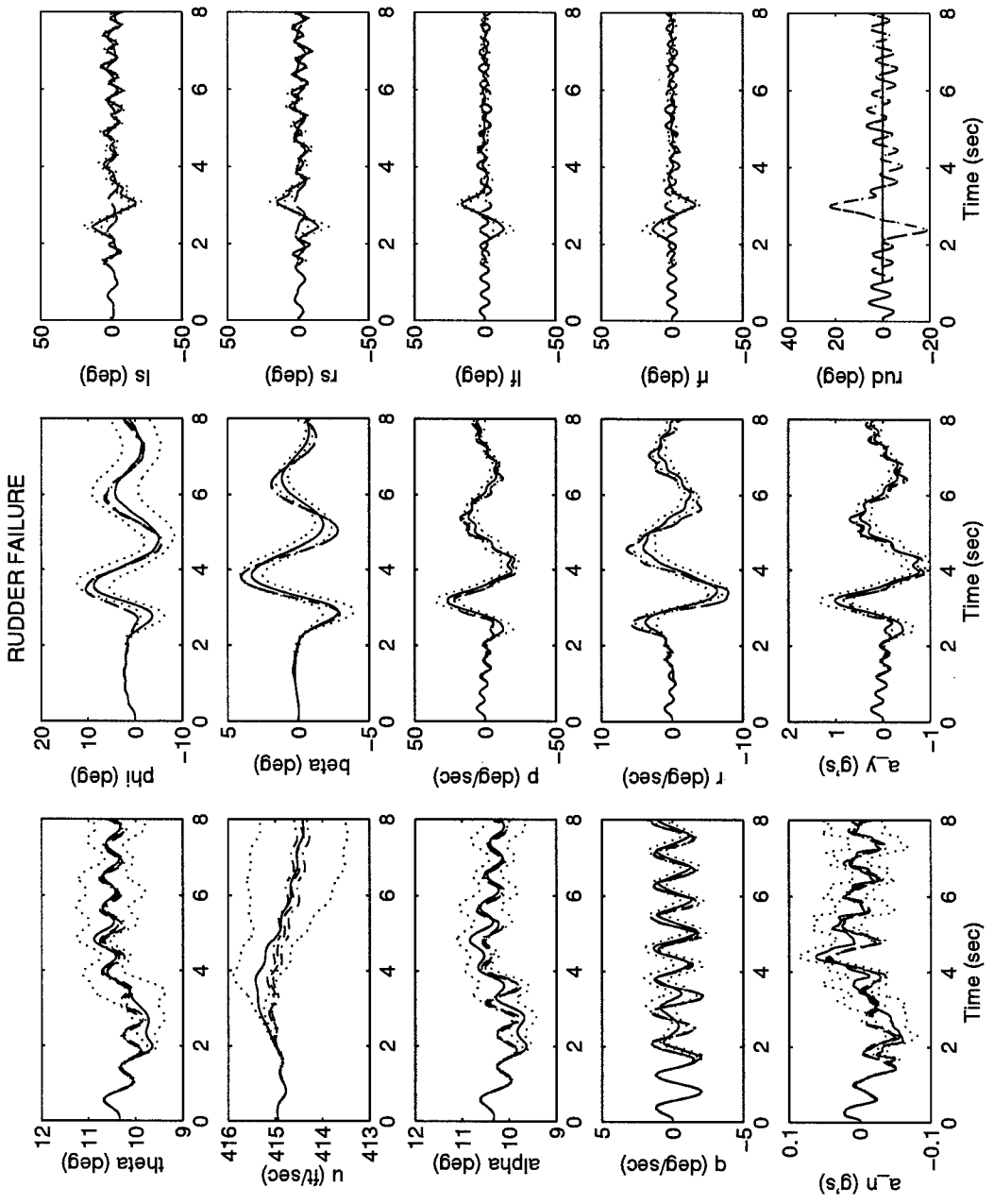

Figure 4-7 Output Plot - Yaw Doublet Input - Rudder Failure - Block 40 with MMAE/CR (with Control Redistribution)

4.2.2.1 Pitch Doublet

Pages A-2 through A-11 (Appendix A) contain the output plots for the pitch doublet. Because the pitch doublet command is incorporated almost exclusively by the stabilators (see actuators in third column of Figures A-1 through A-10), failures of these actuators provide the greatest difficulty for the control system. Comparison of Figure A-1 with A-6 (left stabilator failure) and Figure A-2 with Figure A-7 (right stabilator failure) shows the difference between the simulation with and without Control Redistribution. In the longitudinal channel (first column), it can be seen that, when a stabilator fails, Control Redistribution provides an improvement in the angle of attack (alpha), pitch angle (q), and normal acceleration (a_n). In the lateral channel (second column) the most notable effect resulting from the failure of a stabilator is the shift in the response of the sideslip angle (ß). This shift is the result of only one stabilator responding to the pitch doublet, which creates an imbalance in the lateral channel. Note that the simulation utilizing Control Redistribution has eliminated this delay, indicating that the imbalance has been compensated by the remaining actuators. Note that, in the simulation employing Control Redistribution, the failure of a stabilator does result in the slight separation of roll angle (phi) and sideslip angle (ß). These deviations are attributed to the differences between the nonlinear simulation and the linear design model (from which the redistribution matrices were derived). The deviations were deemed by Stepaniak to be insignificant and easily correctable by the pilot. Stepaniak's conclusions for these deviations were also confirmed by Pachter [42].

Another observation made in the pitch doublet simulation which employed Control Redistribution is the effect of a failed flaperon on the remaining healthy flaperon. Note in Figures A-8 and A-9 that when either flaperon fails, the remaining *healthy* flaperon is also essentially void of any dither input. This result can be understood by recalling that the dither, which is applied to both flaperons of the fully functional aircraft, must be generated

with a 180° phase shift in order to cancel any induced *pitch* which would result otherwise. The same holds true in this situation, with the exception that we must now prevent any induced *roll* moment which would obviously result if only one flaperon were to be dithered. The Control Redistribution algorithm is automatically compensating for this problem by zeroing out the dither to the remaining healthy actuator. The redistribution matrix for a failed left flaperon is given in Equation (4.1):

$$\mathbf{D}_{ai} = \begin{bmatrix} 1 & 0 & +1.1037 & 0 & 0 \\ 0 & 1 & -1.1037 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0.8678 & 0 & 1 \end{bmatrix} \qquad (4.1)$$

Recall from Equation (2.17) in Section 2.4.2 that this matrix is post-multiplied by the control input vector **u**, to obtain the redistributed input vector $\mathbf{u}_r$. From Table 3.3 in Section 3.4.1 it is shown that the third element of this vector, which corresponds to the third column of the redistribution matrix, represents the command to the left flaperon. Note from the third column of Equation (4.1) that the command intended for the left flaperon will be sent in its entirety (multiplied by one) to the right flaperon. Since the dither signal which was originally intended for the left flaperon is 180° out of phase from the original signal from the right flaperon, they cancel each other, resulting in zero dither input. The observant reader will also note a slight bias in the healthy flaperon's position in Figures A-8 and A-9. This is thought to be an attempt to compensate for the slightly increasing roll angle which was discussed above.

4.2.2.2 Roll Doublet

Pages A-12 through A-21 contain the output plots for the roll doublet. For both the left and right stabilator failures, very little effect is seen during the simulation (i.e. these control surfaces are not critical to this doublet). The comparison between the first simulation (no Control Redistribution) and the second simulation (with Control Redistribution) for these failures does show some interesting differences. Stepaniak [47] noted that there was a slight, steadily increasing, separation in the pitch (theta) and roll angles (phi) angles (see page A-17). This change will result in an increase in the flight path angle, which, in-turn, results in a decrease in forward velocity (see the subplot for u). For the flaperon failures, note that the simulation not utilizing Control Redistribution shows a slight delay in sideslip angle. This delay was nonexistent in the simulation utilizing Control Redistribution, however a slight bias does exist due to modeling differences. Note also the same separation is present in the forward velocity, pitch angle, and roll angle as discussed for the stabilator failures. Interestingly, the most noticeable improvement in for this doublet came during the rudder failure. Note that in Figure A-15, the simulation which did not utilize Control Redistribution suffered both a time shift and a decrease in magnitude in the sideslip angle, β, when comparing the failed rudder (solid line) to that with no failures (dashed line). Curiously, close examination of the time shift in the sideslip angle of Figure A-15 reveals that the aircraft with the failed rudder, actually seems to be leading the response of the fully functional aircraft from approximately 2.5 seconds into the simulation until the end. To understand this phenomenon, it is helpful to understand some of the other changes which are occurring during this simulation. Examine the subplot for yaw rate (r) in Figure A-15 and note that it too has deviated significantly from that of the fully functional aircraft (dotted line). In the fully functional case, the yaw rate shows an abrupt change at approximately 2.5 seconds, which corresponds exactly to a shift in the rudder position (see *rud* subplot). The function of the rudder at this point is to prevent an

undesired effect referred to as adverse yaw. Adverse yaw occurs during a roll maneuver and results in the nose of the aircraft turning up. This effect results from torque in the yaw axis caused by changes in both lift and drag of each wing. These changes occur when the roll angle (phi) is non-zero (e.g., in a roll maneuver). The rudder compensates for adverse yaw by creating an equal, but opposite moment in the same axis. This compensation results in what is commonly referred to as a *coordinated turn* and can be incorporated by the pilot, or, as in the F-16, automatically by the flight control system. It is exactly this lack of adverse yaw compensation, or coordinated turn, which results in the apparent lead of the sideslip angle of the failed rudder simulation, over that of the fully functional. Recall that sideslip angle is *not* an angle which results from the aircraft's position relative to some fixed reference frame, but rather an expression which shows the *relationship* between linear velocities in the roll axis (***u*** - out of the nose), pitch axis, (***v*** - out of the right wing), and yaw axis, (***w*** - out of the bottom). When the aircraft experiences adverse yaw resulting from a positive change in phi, both u and v are decreased in magnitude from what they would be during a coordinated turn. In the simulation, these changes result in the premature decrease (at approximately 2.5 seconds), and increase (at approximately 3.75 seconds) of the sideslip angle. In summary, the failure of the rudder does not cause the system to respond any faster to the roll doublet, as may be perceived by the sideslip angle, but rather results in adverse yaw, which alters the velocity components of the aircraft, which, in turn, are represented by the *relationship* ß.

On page A-21, the same rudder failure is simulated utilizing Control Redistribution. Note that here, both β and r vary only slightly from that of the fully functional aircraft, suggesting that the adverse yaw which was being compensated by the rudder, is now compensated by the remaining *healthy* actuators, most notably the flaperons.

4.2.2.3 Yaw Doublet

Pages A-22 through A-31 contain the output plots for the yaw doublet. As noted with the roll doublet, very little difficulty arises from the single failure of either a stabilator or flaperon failure. Note that the same minor separations appear in the roll angle (phi) and forward velocity for the simulation using Control Redistribution as were seen in the previous section. As noted in Section 4.2.2, the rudder failure in this simulation presents the most dramatic indication of Control Redistribution enhancement. Note on page A-26 that a rudder failure in the simulation with no Control Redistribution results in the complete failure of the aircraft to follow the commanded yaw doublet input. This lack of response is easily viewed in every subplot for the lateral channel (middle column). Page A-31 reveals that the output of the aircraft which does utilize Control Redistribution very effectively compensates for this failure, resulting in an output which very closely resembles that of the *fully functional* aircraft.

4.3 Double Failure Analysis

This section examines the results of testing the MMAE/CR against double failures of actuators and/or sensors. In order to implement the MMAE/CR algorithm against multiple failures, several changes had to be made in both the hierarchical structure of the MMAE and the development of the redistribution matrices of the Control Redistribution algorithm. These changes are outlined in Section 4.3.1 and are followed by a comparison of Eide's [9] double failure detection performance to that of the MMAE with the improved dither scheme discussed in Section 4.2. Once this baseline failure detection performance has been established, the results of the double failure analysis of the MMAE/CR are examined. These results have been divided into failure detection performance, shown with

probability plots similar to those in Section 4.2.1, and the dynamic performance of the aircraft, shown with the output plots as in Section 4.2.2.

4.3.1 Required Changes

4.3.1.1 MMAE

As discussed in Section 2.5.5, a hierarchical structure is utilized to reduce the number of required on-line Kalman filters from 67 to 12. Also noted in Section 2.5.5 was the fact that the *total* number of filters required to represent each double failure hypothesis could be represented by $1 + K + \frac{K!}{(K-2)! \cdot 2!}$, where K represents the number of components which could possibly fail. In this research, where K = 11 (i.e., 5 actuators and 6 sensors), 67 filters are required. The astute reader will note that, with 11 components, there are actually 122 double failure combinations possible (including the fully functional and no second failure cases). It should be noted, however, the 1 + (11x11) matrix which makes up these combinations (see Table 4-5) will include several redundant filters. For example, a filter which has been designed assuming that the right stabilator has failed first, and the left stabilator has failed second, will be identical to the filter which assumes the same two failures, in the opposite order. Referring to Figure 2-4, this is equivalent to stating that the filter representing hypothesis $a_{1,2}$ is identical to that which represents hypothesis $a_{2,1}$. While this reduction was not actually necessary for this simulation, it was felt that the eventual implementation of this algorithm could benefit from the reduction in memory required to store the Kalman filters. Table 4-6 summarizes this reduction in filters. Note in Table 4-6 that the asterisk (*) represents the impossible assumption of a double failure of the same actuator, and the X's represent those filters which have already been created for a different column.

First Failure Assumed

Second Failure Assumed											
$a_{0,0}$	1	2	3	4	5	6	7	8	9	10	11
0	$a_{1,0}$	$a_{2,0}$	$a_{3,0}$	$a_{4,0}$	$a_{5,0}$	$a_{6,0}$	$a_{7,0}$	$a_{8,0}$	$a_{9,0}$	$a_{10,0}$	$a_{11,0}$
1	*	X	X	X	X	X	X	X	X	X	X
2	$a_{1,2}$	*	X	X	X	X	X	X	X	X	X
3	$a_{1,3}$	$a_{2,3}$	*	X	X	X	X	X	X	X	X
4	$a_{1,4}$	$a_{2,4}$	$a_{3,4}$	*	X	X	X	X	X	X	X
5	$a_{1,5}$	$a_{2,5}$	$a_{3,5}$	$a_{4,5}$	*	X	X	X	X	X	X
6	$a_{1,6}$	$a_{2,6}$	$a_{3,6}$	$a_{4,6}$	$a_{5,6}$	*	X	X	X	X	X
7	$a_{1,7}$	$a_{2,7}$	$a_{3,7}$	$a_{4,7}$	$a_{5,7}$	$a_{6,7}$	*	X	X	X	X
8	$a_{1,8}$	$a_{2,8}$	$a_{3,8}$	$a_{4,8}$	$a_{5,8}$	$a_{6,8}$	$a_{7,8}$	*	X	X	X
9	$a_{1,9}$	$a_{2,9}$	$a_{3,9}$	$a_{4,9}$	$a_{5,9}$	$a_{6,9}$	$a_{7,9}$	$a_{8,9}$	*	X	X
10	$a_{1,10}$	$a_{2,10}$	$a_{3,10}$	$a_{4,10}$	$a_{5,10}$	$a_{6,10}$	$a_{7,10}$	$a_{8,10}$	$a_{9,10}$	*	X
11	$a_{1,11}$	$a_{2,11}$	$a_{3,11}$	$a_{4,11}$	$a_{5,11}$	$a_{6,11}$	$a_{7,11}$	$a_{8,11}$	$a_{9,11}$	$a_{10,11}$	*

Table 4-6 Reduction of Kalman Filters Required Due to Redundancy

4.3.1.2 Control Redistribution

Recall from Section 2.4 that the principle behind Control Redistribution requires that the healthy actuators compensate for those which have failed. This is accomplished by *redistributing* any command which would normally be directed to a failed actuator to the proper combination of healthy actuators. Recall that this redistribution is obtained by multiplying the original control input vector, $\mathbf{u}$, by a redistribution matrix $\mathbf{D}_{ai}$, which is developed for the failure of the i^{th} actuator. Equations (2.16), (2.17), and (2.20) represent these relationships and are repeated below for convenience:

$$\mathbf{B}_{fail}\mathbf{u}_r = \mathbf{B}\mathbf{u} \tag{4.2}$$

$$\mathbf{u}_r = \mathbf{D}_{ai}\mathbf{u} \tag{4.3}$$

$$\mathbf{D}_{ai} \approx (\mathbf{B}\mathbf{F}_{ai})^{\dagger}\,\mathbf{B} \tag{4.4}$$

Also recall that the failure matrix $\mathbf{F}_{ai}$ is simply the identity matrix with the i^{th} column zeroed out to represent the failure of the i^{th} actuator.

To obtain the failure matrix representing a double failure, $\mathbf{F}_{ai,j}$, a simple matrix multiply of both $\mathbf{F}_{ai}$ and $\mathbf{F}_{aj}$ is performed, or, even more simply, it is set equal to an identity matrix with both failed columns zeroed out. This new double failure matrix is used in Equation (4.5) to obtain the double failure redistribution matrix, $\mathbf{D}_{ai,j}$. Once again, this results in an identity matrix with the exception that the columns representing the failed actuators (now two) contain values which redistribute the command away from the failed components, to those which remain healthy. A typical example of a redistribution matrix for a double failure, $\mathbf{D}_{ai,j}$, is provided in Equation (4.5). Here, $\mathbf{D}_{ai,j}$, corresponds to the failure of the left stabilator ($i = 1$), and the failure of the right flapperon ($j = 4$):

$$\mathbf{D}_{a1,4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.235 & 1 & 0 & -.2595 & 0 \\ 2.014 & 0 & 1 & -1.223 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ .8586 & 0 & 0 & -.0799 & 1 \end{bmatrix} \tag{4.5}$$

In the single failure case, only five redistribution matrices existed (one for each possible actuator failure), and each had only one column which actually redistributed the

command input (see Equation (2.21)). In fact, the actual code used to implement the Control Redistribution algorithm uses only this one column (since the remaining columns have no effect on the command). This allowed Stepaniak [47] to create a *"packed"* matrix, which contained all pertinent columns of each redistribution matrix in one single matrix, which is given in Equation (4.6):

$$\mathbf{D}_{packed} = \begin{bmatrix} 0 & 1 & +1.1037 & -1.1037 & -1.2791 \\ 1 & 0 & -1.1037 & +1.1037 & +1.2719 \\ +0.9060 & -0.9060 & 0 & 1 & +1.1524 \\ -0.9060 & +0.9060 & 1 & 0 & -1.1524 \\ -0.7862 & +0.7862 & +0.8678 & -0.8678 & 0 \end{bmatrix} \tag{4.6}$$

In the case of the double failure, this issue becomes a bit more complicated. As shown in Equation (4.4), each double failure now requires two columns of redistribution as well as an indication for which columns are being represented (1 and 4 for Equation (4.5)). Note also that each column is extremely dependent upon the other, e.g. column one (representing the failure of the right stabilator) will be different for each second failure. This eliminated the consideration of a packed matrix as developed for Stepaniak's research. Instead, a single *three dimensional array* was developed. This array placed the two pertinent columns from each double failure combination on a separate page. The array was structured such that the page number (represented by the third dimension) represents the type of double failure (i.e., which two columns of the failure matrix are being represented). This structure permits the a priori construction and ease of implementation of a single redistribution matrix. As with the Kalman filters, an algorithm was developed to take advantage of the fact that each redistribution matrix, $\mathbf{D}_{ai,j}$, would also represent the redistribution matrix for same two failures $\mathbf{D}_{aj,i}$. All of this resulted in the requirement for a redistribution matrix of 5 rows (number of actuators), 2 columns (number of failures), and

15 pages (possible double failure combinations, not including redundancies). It should also be noted that the 15 pages are utilized to represent the 10 possible double failure combinations as well as the 5 possible single failure cases. It could be argued that a more efficient implementation would be to utilize only 10 pages for each double failure, and Stepaniak's single packed matrix for the remaining single-failure combinations. This approach, while saving a minimal amount of memory, would require two separate algorithms (one for single failures, the other for double failures). To avoid this inefficiency, each single failure case was treated as a double failure, with the second failure represented by 0, allowing it to be represented on a separate page of the 5x2x15 array.

4.3.1.3 Probability Plots

In order to demonstrate the failure detection of the MMAE and MMAE/CR against double failures, the structure of the probability plots requires new consideration. To begin, it must be realized that, unlike the single failure analysis where the same twelve on-line filters were utilized throughout the simulation, the double failure analysis utilizes the hierarchical structure discussed in Section 2.5.5. This structure *swaps out* the original twelve filters for a new set of filters, based on the declared first failure. This would require one to consider two of the simple probability plots as shown in Figure 4-1, one to represent the set of filters used until the first failure is declared, and another set of subplots to represent the set of filters used thereafter. Following this structure would result in no less than 210 plots (2520 subplots) to be examined. In an effort to reduce this monstrous amount of data, the following steps were taken. First, it was assumed, based upon the single failure analysis, that the first failure detection could be adequately represented by a subplot representing the fully functional aircraft hypothesis, and the hypothesis of that particular failure. Second, it was assumed that each second failure probability analysis could be adequately represented by the subplot representing that particular *double* failure.

Utilizing this structure, each double failure summary plot represents a single first failure with all possible second failures. This structure is perhaps better explained by examining an actual example. Figure 4-7 is the double failure summary plot for the consideration of the first failure (left stabilator) occurring at one second, and the second failures occurring at two seconds. Note that the first bank of filters is sufficiently represented by the subplots for the fully functional hypothesis, and that of the failed left stabilator. These two subplots provide sufficient information to show that the correct failure was detected after it was inserted at one second into the simulation. At this point the MMAE algorithm swaps out the 12 original on-line filters for a set based on this first failure. Thus the subplots labeled rs, lf, rf, etc., are actually representing the filters which consider the left stabilator as the first failure (i.e. $\mathbf{a}_{1,2}$, $\mathbf{a}_{1,3}$, $\mathbf{a}_{1,4}$, etc., from Table 4-6) from that time point forward. Because the algorithm used eliminates any redundancies in the required filters, the filters representing the fully functional aircraft and that which represents the first failure (i.e., no second failure) are carried over from the first bank to the second. This not only provides continuity in the two subplots shown, but also eliminates any transients which may occur from bringing a new filter on-line.

It should also be noted that the subplots representing the second failures are each the result of a separate Monte Carlo simulation. This resulted in the need to decide which of these simulations should provide the subplot of the filter representing the first failure. Assuming that the second failure is eventually detected, this was a purely arbitrary choice. The convention used here utilizes the simulation which generates the subplot immediately following that of the first failure. For example, in Figure 4-7, the simulation used to provide the subplot of the probability for $\mathbf{a}_{1,0}$ (failed left stabilator, no second failure) is that of the following subplot representing $\mathbf{a}_{1,2}$ (failed left stabilator, failed right stabilator). There are two noted exceptions to this convention. Because of difficulties with the plotting algorithm, when the first failure is the rudder or the lateral acceleration sensor, the

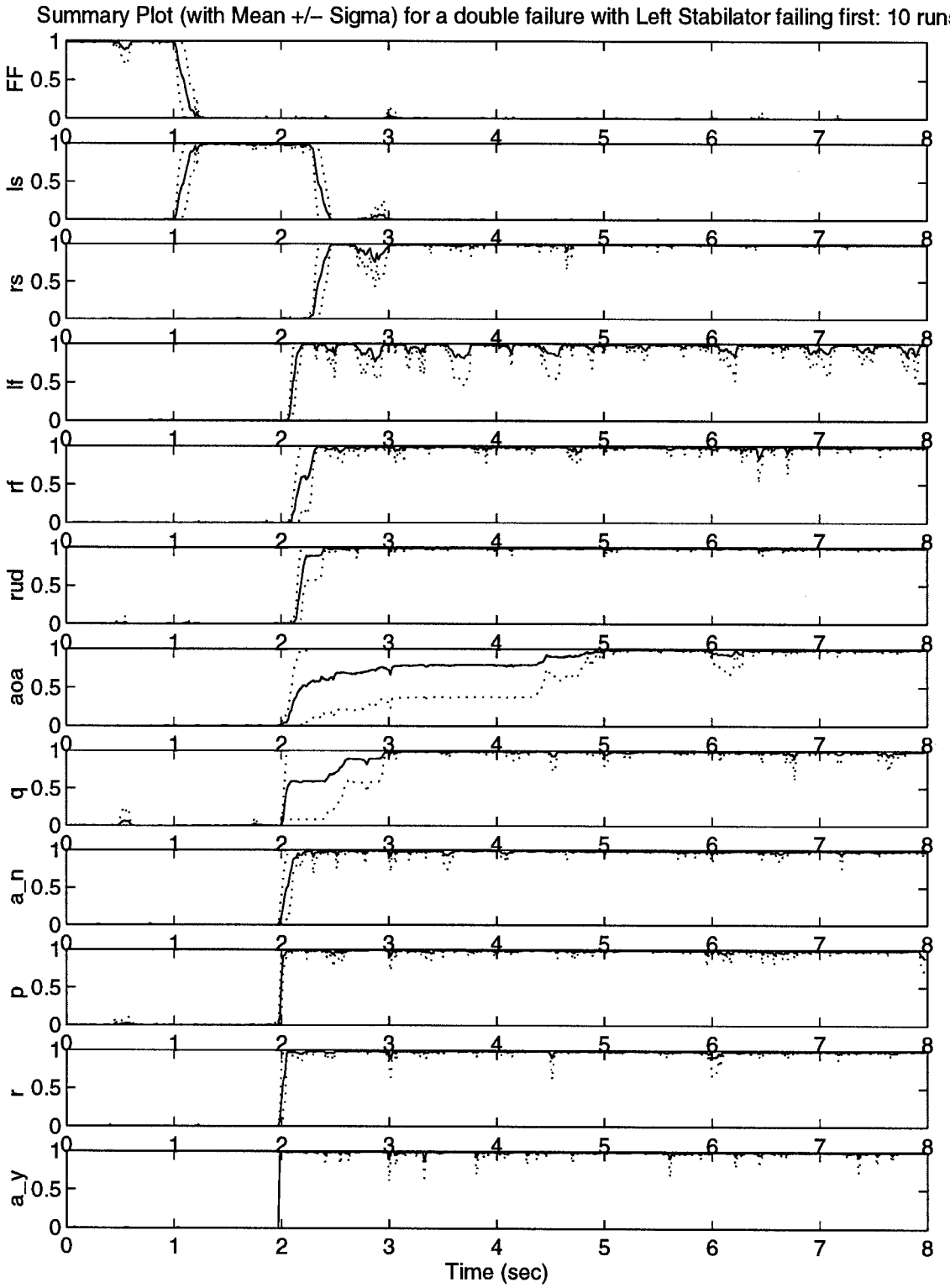

Figure 4-7 Double Failure Summary Plot - Left Stabilator Failing First

simulations used to represent the first failure are the rudder/a_y and a_y/q simulations, respectively. In retrospect, perhaps these difficulties should have been resolved so that these two simulations followed the convention of the other simulations; however, as mentioned above, the choice is purely arbitrary. With this convention in mind (along with the two noted exceptions), the reader should not be alarmed when the probabilities of the filters representing the first-failure-only hypothesis, and that representing the second failure do not sum to one (i.e., during the transition). Of course, if a particular simulation warrants further analysis, (e.g., if the correct hypothesis is reached with difficulty), then the individual plots of that simulation can be easily generated.

4.3.2 Basis Simulation

Once the algorithms had been developed to generate and analyze each double failure case, a simulation was performed to provide a basis upon which to compare any significant changes resulting from the implementation of Control Redistribution. The basis simulation was performed using the MMAE with the standard Block 40 controller, and Stepaniak's [47] improved dither scheme. No maneuvering was incorporated into this simulation, for it was desired to establish the systems ability to detect and lock onto failures correctly in a benign flight condition. Later, in Section 4.3.3, simulations are performed with commanded pitch, yaw and roll doublets, to determine the system's control authority capabilities, as well as any changes in failure detection/lock which may result. The resulting double failure summary plots are located in Appendix B. Since no double failure analysis had been performed with this configuration, a comparison was made to the double failure results obtained by Eide [9]. Eide qualitatively summarized the double failure performance in a chart which has been reproduced in Table 4-7. The key to this chart (Table 4-8) indicates that the assessment was made utilizing Eide's single failure results as a basis. Rather than include all of Eide's single and double failure probability plots, a

comparison of the simulation performed with Stepaniak's dither was made and a new matrix was developed. This matrix is shown in Table 4-9 and provides a performance indication of both detection (first indicator) and probability lock (second indicator). Each indicator includes a subscript which gives an indication of improvement (if any) over the Eide simulation. This matrix was made by comparing the actual probability summary plots located in Appendix B, with the summary plots from Eide's simulation [9]. The interested reader can compare each *row* of Table 4-9 with the corresponding summary plot in Appendix B.

Second Failure

First Failure	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS		Good	**ND**	**Poor**	Good	Good	**Poor**	Good	**Fair**	**Fair**	**Fair**
RS	Good		**Poor**	**ND**	Good	Good	**Poor**	Good	**Fair**	**Fair**	**Fair**
LF	Good	Good		Good	Good	Good	Good	Good	Good	Good	Good
RF	Good	Good	Good		**Fair**	**Fair**	Good	Good	Good	Good	Good
RUD	Good	Good	Good	Good		Good	Good	Good	Good	**ND**	Good
AOA	Good	Good	Good	Good	Good		Good	**Fair**	Good	Good	Good
Q	Good	Good	**Fair**	**Fair**	**Fair**	Good		Good	Good	Good	Good
A_n	Good	Good	**Poor**	**Fair**	Good	Good	Good		Good	Good	Good
P	Good	Good	Good	Good	Good	Good	Good	Good		Good	Good
R	Good	Good	**Fair**	Good	**Poor**	Good	Good	Good	Good		Good
A_y	Good	Good	**Fair**	**Fair**	Good	Good	Good	Good	Good	Good	

Table 4-7 Dual Failure Evaluation - Old Dither, Eide [9]

Rating	Qualitative Description
Good	Second failure shows performance commensurate with or better than for lone single failure. Look for probability lock and hold.
Fair	Second failure shows performance somewhat degraded from that of single failure case. Look for significant dropouts from probability lock.
Poor	Second failure shows performance drastically degraded from that of single failure case. Look for spiking of probability, but no lock.
ND	No Detect. Second failure completely missed or falsely declared.

Table 4-8 Performance Definitions for Table 4-7, Eide [9]

For the first set of simulations shown in Figure B.1, it is seen that the probability of the filter portraying the failure of the left stabilator correctly converges to one in less than .25 seconds. This is not surprising, for at this point, the simulation will behave exactly as it did during the single failure analysis (see Figure 4-2). With the exception of the angle of attack and pitch sensors, all of the second failures converge to a probability of one in less than .5 seconds after the actual failure insertion at 2 seconds into the simulation. The angle of attack and pitch sensor failures eventually converge after three and one seconds, respectively. Note from the AOA (angle of attack sensor) *column* in Table 4-9 that the detection of the failure of this sensor was slow for all cases except that of the right stabilator failing first. In fact, inspection of Figure 4-2 reveals that even as a single failure, the probability of the filter portraying this failure took an entire second to reach a probability near one (relatively slow when compared to the other failures), which suggests that even with the improved dither, the observability of this device is somewhat limited. This trend is somewhat puzzling, for the failure of the left stabilator does not exhibit this problem. It should be noted that this actuator has a history of being slow to detect in previous research as well, suggesting that perhaps the SRF's nonlinear response to its fluctuations, or lack thereof in the advent of failure, is not quite accurate, or that dither phasing and its correlation to the particular aircraft motion in this simulation still yields a certain amount of asymmetry between the left and right stabilator-induced responses of the aircraft. Examination of the plots for the individual simulations (not included) revealed that the missing probability for the two slow sensors (AOA and q) were contained in the filter hypothesizing no secondary failure. Thus, the delay was not due to a case of mistaken identity, but rather poor observability as previously hypothesized. Utilizing the criteria given in the third column of Table 4-10, it can be seen that the probability lock is good for all of the secondary failures in this scenario of a first failure being the left stabilator.

Second Failure

First Failure	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS		G_3 G_3	G_1 G_1	G_1 G_1	G_3 G_3	$P_4(3)$ G_3	$F_1(1)$ G_1	G_3 G_3	G_3 G_2	G_1 G_2	G_3 G_2
RS	G_3 G_3		G_2 G_1	$F_1(.8)$ G_1	$F_2(.7)$ G_3	G_4 G_3	$F_1(1)$ G_1	G_3 G_3	G_1 G_2	$F_1(.8)$ G_1	G_2 G_2
LF	G_2 G_3	G_2 G_4		G_1 G_3	$F_2(.6)$ G_3	$F_4(1.7)$ G_2	G_1 G_1	G_3 G_2	$F_4(.8)$ G_3	$F_4(1)$ G_3	G_3 G_3
RF	G_3 G_3	G_2 G_3	G_2 G_2		G_2 G_1	$F_4(1.7)$ G_2	G_1 G_1	G_2 G_2	$F_4(.8)$ G_3	$P_4(2.2)$ G_3	G_3 G_3
RUD	G_1 G_4	G_1 F_4	G_1 G_1	G_1 G_1		$F_3(1)$ G_2	G_1 G_3	G_2 G_2	G_1 G_3	G_1 G_1	G_1 G_3
AOA	G_3 G_3	G_2 F_4	G_1 G_1	$F_1(.7)$ F_1	G_1 G_2		G_1 G_1	G_4 G_1	G_4 G_3	G_2 G_2	G_3 G_3
Q	$F_3(.8)$ F_4	$F_4(1.1)$ F_4	G_1 G_1	G_1 G_1	G_1 G_1	$F_4(.7)$ G_2		G_3 G_3	G_4 G_3	G_2 G_2	G_3 G_3
A_n	$F_4(1.2)$ F_4	$F_4(.6)$ F_4	N_3 N_3	N_3 N_3	G_1 G_1	$F_4(1.5)$ G_1	G_1 G_1		G_2 G_3	G_2 G_1	G_3 G_3
P	G_3 G_3	G_2 G_4	G_1 G_3	$F_1(.7)$ F_1	G_1 G_2	$F_4(.8)$ G_2	G_1 G_1	G_3 G_3		G_3 G_1	G_3 G_2
R	G_3 G_3	G_3 F_4	G_1 P_1	G_1 P_1	G_1 G_1	$F_4(1.5)$ G_2	G_1 P_4	G_3 P_4	G_4 G_3		G_3 P_4
A_y	G_3 G_3	G_2 G_3	G_1 F_1	$F_1(.8)$ F_1	N_4 N_4	$F_4(1.5)$ G_2	G_1 G_2	G_3 G_3	$F_4(.8)$ F_2	G_3 G_2	

Table 4-9 Dual Failure Evaluation - New Dither

Rating	First Indicator (Detection Speed)	Second Indicator (Detection Lock)
G	Good: Average second failure detected in less than .5 seconds.	Strong Lock - very few, if any dropouts
F	Fair: Average second failure detected in less than 2 seconds (average time indicated in parenthesis)	Few Dropouts detected before 6 seconds (Dropout = probability less than .9)
P	Poor: Eventual Detection of second failure (average time indicated in parentheses)	Extensive Dropouts before 6 seconds (Dropout = probability less than .9)
N	Non-detection: Average probability never reaches 1	Average probability never reaches 1
Subscripts - Used for comparison to old dither scheme		
1	Significant Improvement (one second or more)	Second simulation rated 2 or more levels higher than first
2	Some Improvement shown	Second simulation rated 1 level higher than first
3	No Change	No Change
4	Performance Degraded	Performance Degraded

Table 4-10 Performance Definitions for Table 4-9

The second row in Table 4-9 and Figure B.2 shows that the failure of the right stabilator (first failure) was somewhat similar to that of the left stabilator (first failure), with the notable exception being that the angle of attack sensor failure was detected fairly quickly (the only set of simulations where this occurred). The reader who is solely examining Table 4-9 (without referring to Appendix B) should note that, while this scenario did generate four ratings of "Fair" ($.5s \leq t \leq 2s$ for detection), each of these secondary failures is detected in one second or less.

First failures of the left and right flaperons and the rudder are depicted by Figures B.3, B.4, and B.5 respectively, as well as the third, fourth and fifth rows of Table 4-9. The delay in the angle of attack sensor, as discussed above, is present in all three of these scenarios (though not as bad for the case of the rudder failing first). Note that the detection of failures in the roll and yaw rate sensors are also somewhat delayed. This delay is to be expected when a flaperon or the rudder fails first, for these surfaces greatly effect both the roll and yaw parameters. The application of dither to these actuators is crucial to the detection of failures in either sensor. These delays were examined using individual simulation plots, which indicated that the only filter which shared the probability during the transition was that which hypothesized no secondary failure. Detection lock for all three of these scenarios was found to be quite good.

With a few notable exceptions, detection and lock of secondary failures when a sensor fails first (Figures B.6 - B.11 and rows 6-11 in Table 4-9) were quite good. The first of these exceptions is for the first failure of the normal acceleration sensor (a_n). Note from Figure B.9 that this scenario results in the difficult detection of a failure in one of the flaperons. In fact, the mean of the ten Monte Carlo runs for the filter hypothesizing a failure of the normal acceleration sensor and the correct flaperon never does reach a probability of 0.9 (at which point a detection would be declared). Again this difficulty was expected. The failure of a flaperon would result in an unexpected change in normal acceleration, and

indeed, it is this unique change in a_n (unique to a flaperon failure) which should cause the residuals of the Kalman filter hypothesizing a flaperon failure to decrease, resulting in its probability increasing to nearly one. The removal of the normal acceleration information (i.e., the failure of the a_n sensor) will greatly inhibit the correct declaration. Another notable exception occurs when the yaw rate sensor (r) fails first. This scenario is depicted in Figure B.10, where it can be observed that, for several of the simulations, the probability of the filter which depicts the failure of the yaw sensor with no secondary failure takes a sudden (and unexpected) jump in probability at approximately 3 or 3.5 seconds. This anomaly can be seen in both the flapperon simulations, and the pitch (q), normal acceleration (a_n), and lateral acceleration (a_y) sensors. No solid explanation was obtained for this problem. One hypothesis is that perhaps these failure combinations caused the aircraft to deviate form the design point in the flight envelope (speed and altitude). This hypothesis was explored by Eide [9] who found that the constant gain Kalman filters were severely effected by small changes in the aircraft's velocity or altitude. The final exception is that of the lateral acceleration sensor failing first, and the rudder failing second. This scenario, as with the first, is to be expected. The detection of the failure of the rudder is understandably very dependent upon inputs from this sensor. Again, all of these difficulties were found to have their remaining probabilities contained almost exclusively in the filter depicting no second failure.

It should be noted that, while the results of the comparison shown in Table 4-9 will give an overall indication of the improvements obtained with the new dither, a truly adequate comparison would require that the failure insertion times be made identical. In Eide's analysis, the insertion times for the first and second failures were at 3 and 5 seconds into the simulation, respectively, whereas this simulation inserts the failures at 1 and 3 seconds. These relatively early insertions were required for the analysis of the Control Redistribution algorithm. As noted by Eide, the failure detection performance of the

MMAE is somewhat dependent upon the time of the failure insertion point. Since the main purpose of this simulation was to provide a basis upon which to evaluate the Control Redistribution algorithm, the decision was made not to adjust the times to match those of Eide's research. Examination of Table 4-9 reveals that over half (51%) of the 110 simulations resulted in an improvement in failure detection time. The same percentage of improvement was found to occur for detection lock. With one exception, all failures in the second simulation were detected. Note that this one *no detection* case is for the $a_{11,5}$ failure (lateral acceleration sensor and rudder) and is to be expected. These two components are very tightly coupled, and somewhat independent of the other components. This is especially true when the lateral acceleration sensor (a_y) fails first, for the detection of a rudder failure is singularly dependent upon this sensor. In fact, past research has indicated that this aircraft could benefit from the incorporation of a rudder position sensor to assist in the event of the a_y sensor failure [37]. While the one ND (no detection) rating in the second simulation may not seem to be a tremendous improvement over the three "ND's" indicated for Eide's simulation in Table 4-7, if the same criteria used to develop Table 4-9 were applied to Eide's simulation, the result would be no fewer than 14 "ND's". Only 20 simulations (18%) showed any degradation in failure detection. While this might seem significant, it should be noted that of these 20, 17 were still classified as "Good" or "Fair" (i.e., the degradations were rather slight). For failure lock, 14 simulations (13%) showed some degradation, and of these 10 were still "Good" or "Fair". In summary, as with the single failure analysis, the ability of the MMAE to detect and maintain lock on double failures has been significantly improved by the incorporation of Stepaniak's [47] new dither algorithm.

4.3.3 Double Failures with Control Redistribution

4.3.3.1 Failure Detection

Once the methods of implementing Control Redistribution for double failures had been fully developed, and a basis simulation completed, simulations were performed which incorporated two failures on an MMAE-based Block 40 controller which utilized Control Redistribution. Initial results were mixed. While Control Redistribution seemed to help tremendously in the detection of a sensor failure once an actuator had failed first, many double failure simulations were unable to maintain correct detection lock after the second failure. Both of these trends can be readily observed in the probability summary plot of Figure 4-8. In this plot, the results of the left stabilator failing first are plotted. Notice that, while the initial detection of the second failures is exceptional, detection lock is extremely poor for the actuators while excellent for the sensors.

The improvement in sensor detection was somewhat expected. The problem which always exists after the failure of an actuator is that of impaired identifiability of a failure due to lowered system excitation. When an actuator fails to respond to the dither input command, a second component failure can become difficult to detect. This problem can be readily seen in the basis simulation double failure plots found in Appendix B. When Control Redistribution is employed, the dither signal for the failed actuator is *redistributed* to the healthy actuators in an attempt to create the same identifiability that was present with the fully functional aircraft. Comparison of Figure 4-8 with Figure B.1 shows this improvement in the aoa, q, and a_n sensors.

The degradation in detection lock which came after the failure of two actuators was not expected. After an extensive investigation into possible problems with the MMAE and Control Redistribution code, a relatively simple explanation became quite evident. As

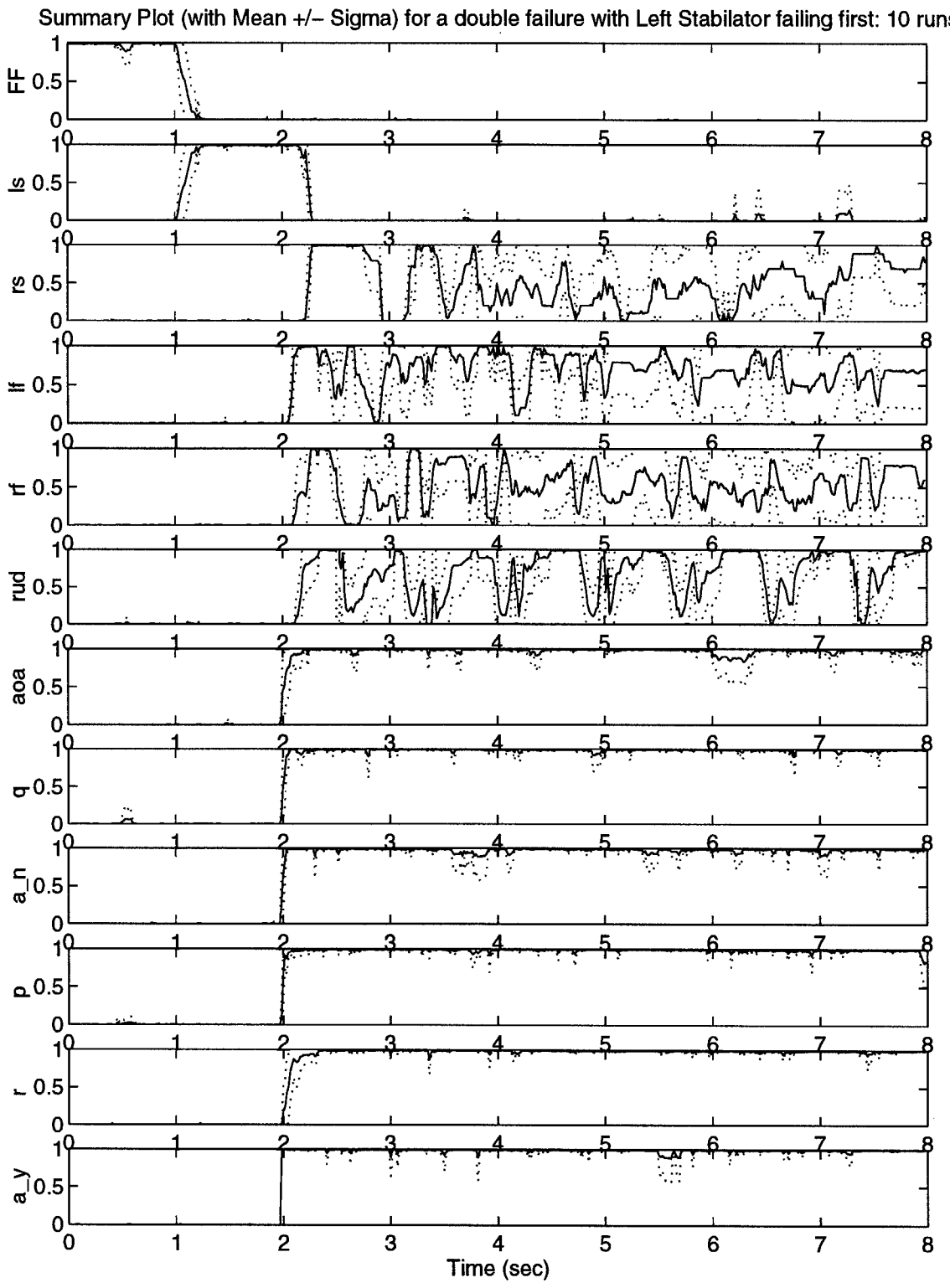

Figure 4-8 Double Failure Summary Plot Showing Detection lock Degradations due to Control Redistribution

discussed in Section 4.1, during a single failure, the dither fluctuates the actuators at nearly their physical rate saturation points. When a double actuator failure occurs, the increased load which is placed on the remaining healthy actuators will force them to their rate saturation limits, making their response to the dither nonlinear. When this occurs, the residuals of the Kalman filters, which were designed based upon a linear response, will no longer provide a good indication of the correct hypothesis, and the probabilities, which result from these residuals, will be in error.

This hypothesis was supported by the observation that the healthy actuators, which received the most redistributed command, seemed to suffer the greatest amount of degradation. Also, when one of the two failures was a sensor, the probability dropouts were usually less noticeable. This also supported the rate saturation hypothesis, since no Control Redistribution is incorporated for sensor failures.

To verify the hypothesis, the amount of commanded input from the Control Redistribution algorithm was scaled to various degrees and the results were examined. This test showed a definite decrease in the anomaly as the amount of scaling was increased (i.e., as the magnitude of the redistributed control was *decreased*). With this trend firmly established, various levels of scaling were tested in an effort to minimize the problem. It should be realized that there is a tradeoff in this approach. While scaling back the amount of redistributed control resulted in a better detection lock, it also reduces the amount of control authority and system excitation gained from utilizing Control Redistribution in the first place. Thus the objective was to scale back the redistributed command just enough to provide an acceptable level of detection lock (i.e., not saturate the unfailed actuators), while continuing to allow for an improvement in commanded control authority, system excitation, and failure detection.

Because the failure lock problem was not a problem in the single failure case, it was thought that a viable solution could be to normalize each *individual* redistribution matrix (i.e., the particular page of the *packed* redistribution matrix which corresponds to the particular failure scenario declared) based upon the maximum amount of redistribution used in the single failure case. A quick examination of the packed matrix used for the single failure simulations (see Equation (4.6)) shows that the maximum factor of redistribution to any one actuator to be one. This being the case, an algorithm was developed that summed the two redistribution factors of each row of the individual redistribution matrix, and determined the greatest value to be used. This value was then used to normalize actual command to each actuator. While this solution worked well for several double failure scenarios, it failed to consider that while some of the actuators can handle this maximum redistribution factor of one, others were apparently more sensitive. In an effort to determine what the minimum limits were for each sensor, an analysis was performed by multiplying the normalizing factor found for each simulation by varying degrees. Thus, after the second failure has been declared, the command for the i^{th} healthy actuator will be its original command ($Hcom_i$ - note: H = healthy), plus the redistributed command from each of the two failed components ($Rcom_1$ and $Rcom_2$ - note: R = Redistributed), all of which is divided by the scaled normalizing factor consisting of the maximum sum from the redistribution matrix (R_{max}), multiplied by the scale factor (SF). This algorithm is depicted in Equation (4.7):

$$Hcom_i = (Hcom_i + Rcom_1 + Rcom_2)/(R_{max} * SF) \qquad (4.7)$$

The scale factors (i.e., SF values) used for this analysis were 1, 1.2, and 1.5. These values were each used to generate a set of double failure probability plots, which were then evaluated to determine the best scaling factor for each double failure scenario (110 in all). The scale factor choice was made based on the *minimum* value which resulted in *acceptable*

results. To be considered acceptable, the resulting detection lock was required to be judged as "FAIR" as rated in Table 4-10 ("Few dropouts detected before 6 seconds"). It was deemed more important to select a minimum scale factor in lieu of improved performance, in the hopes that increased Control Redistribution would pay off during the pilot-commanded doublet analysis subsequently.

The resulting scale factor choices for implementation are shown in Table 4-11. In this table, it can be seen that for the majority (81%) of the double failure scenarios, a scale factor of one (no scaling) is sufficient. Note that the most difficulty is found in the scenarios in which the right stabilator or rudder is inserted as one of the failures. As mentioned previously, the right stabilator has also been troublesome in past research, and its difference in behavior from its counterpart on the left side of the aircraft suggests that perhaps there is still a phasing problem in the dither, or perhaps that the VISTA simulation's incorporation of the effects of this surface is somewhat flawed. The difficulty associated with the rudder was somewhat expected, as the failure of the non-redundant

Second Failure

First Failure	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS		1	1	1	1	1	1	1	1	1	1
RS	1		1.5	1.5	1.5	1	1	1.2	1.2	1	1
LF	1	1		1	1	1	1	1	1	1	
RF	1	1.2	1.2		1	1	1	1	1	1.2	1
RUD	1	1	1	1		1.2	1.2	1.2	1.2	1.5	1
AOA	1	1.2	1	1	1		1	1	1	1	1
Q	1	1.2	1	1.2	1.2	1		1	1	1	1
A_n	1	1.2	1	1	1.2	1	1		1	1	1
P	1	1.2	1	1	1.2	1	1	1		1	1
R	1	1	1	1	1	1	1	1	1		1
A_y	1	1	1	1	1	1	1	1	1	1	

Table 4-11 Control Redistribution Scale Factor Choices

rudder will result in the highest amount of redistributed command to both the stabilators and the flaperons. This is evident in the last column of the single failure packed redistribution matrix shown in Equation (4.6).

It is interesting to note from Table 4-11 that many of the double failure scenarios which required a redistribution scale factor involved sensor failures. This is true, despite the fact that no added redistribution will result from the failure of a sensor. It is thought that this anomaly is present because the increased difficulty in failure detection which occurs when a particular sensor fails could push an already marginally rate-saturated situation into the trouble zone. If, for example, the single failure of the rudder creates a minor amount of rate saturation in the remaining actuators, the resulting probabilities of the MMAE algorithm might still be acceptable provided that good sensor information is still available. If however, a critical sensor fails (in this case, the yaw rate sensor - R), the results could be as bad as those seen for the double actuator failure in Figure 4-8.

Once the choices shown in Table 4-11 were made, a simple logical filter was devised to determine what scale factor to use based on which components had failed. The results of this implementation can be found in Appendix C. Comparison of these probability plots with those generated for the basis set in Section 4.3.2 are mixed. In most scenarios (i.e. groupings of first failures), a definite improvement can be seen in the detection of sensor failures after an actuator has failed first. Comparison scenarios where the actuator fails second tend to show some degradation in the area of detection lock. Again, this is to be expected since the scale factor was chosen to satisfy minimally acceptable constraints in the hopes that the resulting level of Control Redistribution would provide more control authority and system excitation in these scenarios. Thus it is hoped that the benefits of failure detection, due to increased excitation and improved control authority, will outweigh any degradations seen in probability lock.

4.3.3.2 Doublet Response

To test this system for control authority in the face of double failures, the same purposefully commanded pitch, roll, and yaw doublet inputs used in the single failure analysis are applied one second *after* the insertion of the second failure. The reader should note that this doublet analysis *only* examines the control authority issue, i.e., the failure *detections* are assumed to occur before the application of the doublet command, exactly as demonstrated in the previous section. A recommendation is made in Chapter 5 that future research efforts examine the capability of the system to *detect* failures during the commanded doublet input (or any other maneuvering commands). For comparison purposes, the doublets are applied to a system which does not incorporate Control Redistribution, as well as one which does. Due to the enormous amount of data involved in a complete evaluation, only double failures involving two actuator failures were tested. This results in 10 separate double failure scenarios for each of the three doublets, which are performed for both the systems with and without Control Redistribution. This resulted in 60 simulations, each consisting of 10 Monte Carlo Runs and a set of 15 output subplots. While a more complete analysis (i.e., allowing failures in both actuators and sensors) would have been preferred, the time and effort required to process the 240 simulations was deemed to be unfeasible.

All of the output plots for this section are listed in Appendix D. This appendix is constructed in the same manner as Appendix A, i.e., divided into three separate sections, one for each doublet. Each section contains both the plots for the basis simulations (no Control Redistribution) and those plots for the test simulation (with Control Redistribution). For each simulation, the first and second failures are inserted at one and two seconds into the simulation, respectively, and the one-second doublet is initiated at three seconds. While it could be argued that a later doublet insertion time would insure that *all* second failures are properly detected before the doublet insertion, the departure from the

flight envelope design point (discussed in Section 4.3.2), and the fact that *most* secondary failures *are* properly declared at this point, made the choice for early insertion (3 seconds) more logical.

Some of the initial results of the doublet simulations were rather disappointing. In particular, during the yaw doublet, the incorporation of Control Redistribution in a simulation in which the failure of any actuator occurred in conjunction with the failure of the rudder, resulted in practically no improvement in control authority. This was disturbing, for it was thought that in the event of a rudder failure, the flaperons and stabilators could be utilized to provide, at the very least, a yaw doublet response that could be helpful to the pilot. Certainly in the single failure analysis, this yaw doublet was produced for the rudder failure (see Section 4.2.2.3). But this was not the case with the double failure analysis, where the yaw response invoked by Control Redistribution during *any* double failure combination which included a rudder failure was, to say the least, unsatisfactory. This lack of performance can be seen in Figure 4-9. In this figure, it can be seen that the left flaperon (lf) and rudder (rud) fail at one and two seconds respectively. The lack of yaw response can be seen in every subplot of the center column (i.e., the lateral channel). While there does appear to be a small amount of oscillating response in each subplot, each is approximately 180° out of phase from the normal response (dashed line). A close analysis of the actuator responses in the third column reveals the source of the problem. Recall that in Section 4.2.1 it was stated that the dither signal was not to be incorporated in conjunction with *any* commanded input from the pilot. Obviously, in Figure 4-9 the dither signal is applied throughout the simulation, including during the yaw doublet input which occurs between three and four seconds. Reexamination of the FORTRAN code used to generate the VISTA simulation revealed that the software switch utilized to stop the dither in the event of commanded input was only set for the pitch stick input. Indeed, this reexamination of the single failures plots in Appendix A reveals that this

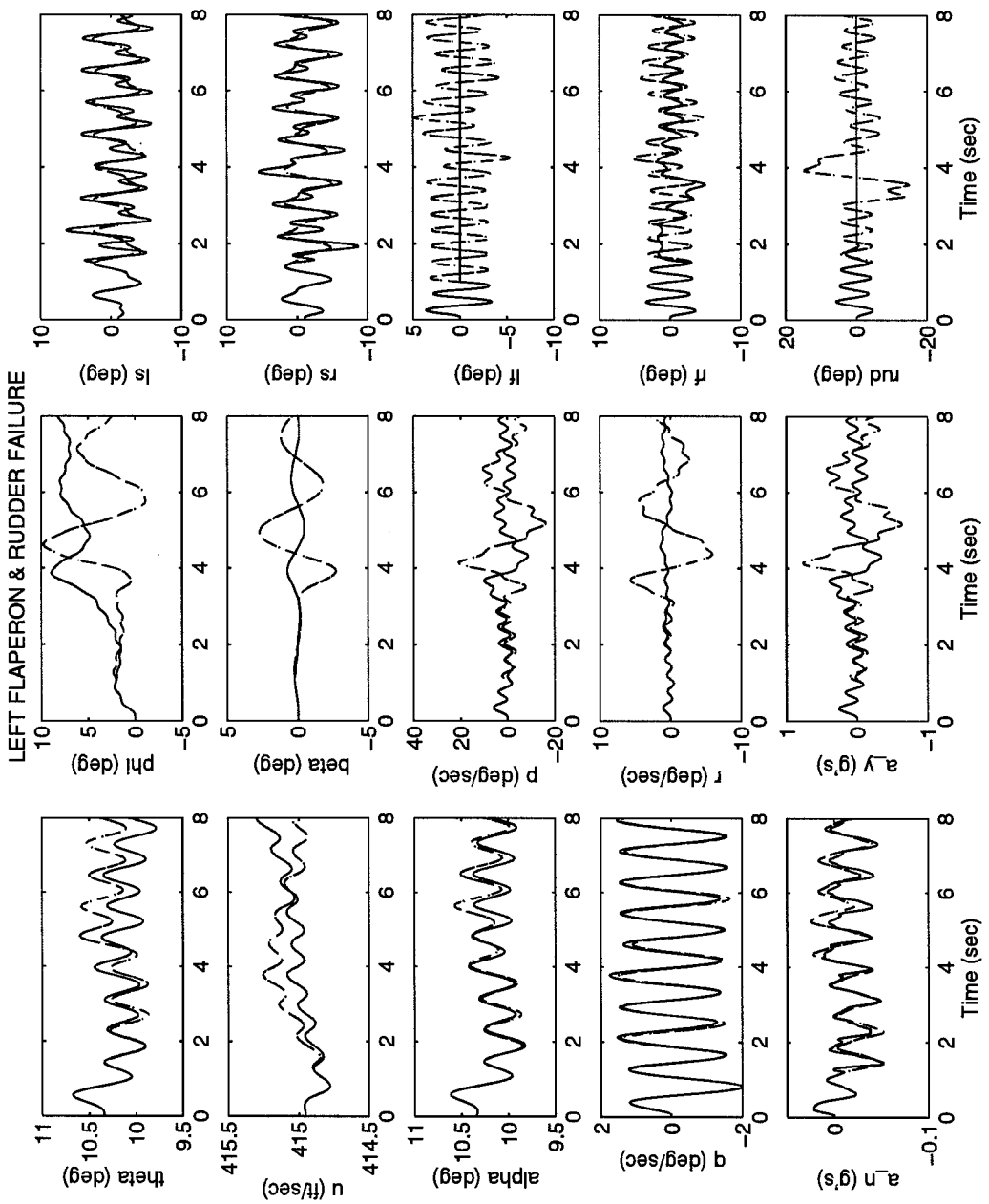

Figure 4-9 Initial Yaw Doublet Response - with Control Redistribution

Left Flaperon & Rudder Failure

trend is present. While the impact of this error is not thought to affect the results of the single failure analysis seriously, the double failure simulations were re-accomplished with the incorporation of the proper code. The results of these simulations are provided below.

Before presenting the separate analysis of each set of doublet simulations, it should be noted that there was one failure scenario which produced a diverging effect in all cases (i.e., irrespective of doublet type or the incorporation of Control Redistribution). The failure of both stabilators resulted in the divergence of both the pitch angle (theta) and the forward linear velocity (u). Recalling that the failures of these actuators places them at their nominal positions (i.e., the positions required to keep the aircraft in trim), one might think that their failures should not result in any change of these two variables. Close examination of Figures D-1 and D-11 reveals that these two failures do *not* actually cause the divergence, for the problem does not commence until the insertion of the doublet at three seconds. At this point it is felt that a disturbance in the pitch axis starts the departure from the nominal position, and with no stabilators to counter this torque, it quickly diverges at what appears to be an exponential rate. While a simulation with no input doublet was not performed, it is strongly felt that even small pitch variations due to wind gusts, and the dithering of the remaining actuators would cause the same results. Note that, in the simulations performed with Control Redistribution, each doublet performed in the presence of the double failure of both stabilators (see Figures D-11, D-31, or D-51) also resulted in an increasing (magnitude) bias on both the left and right flaperon. It is thought that this is an attempt by the flaperons to nullify the diverging pitch. Because of the relatively close position of the flaperons to the aircraft's center of gravity, this attempt is futile; in fact, close examination reveals that pitch angle (theta) and velocity (u) actually diverged more in the simulation with Control Redistribution than in the simulation without it. Perhaps this deterioration is due to the increased activity of the remaining healthy actuators.

4.3.3.2.1 Pitch Doublet

As with the single failures, the nature of the pitch doublet motivates the analysis of both the pitch angle, theta, and the pitch rate, q. This doublet will also be most affected by failures of the actuators which control the pitch maneuver, i.e., the stabilators. Because of the problems discussed above, the double failure of both stabilators resulted in an unstable aircraft for both simulations (i.e., with, and without Control Redistribution) of this doublet input (see Figures D-1 and D-11). Comparison of Figures D-2 through D-7 to D-12 through D-17 shows that double failures which include a *single* stabilator failure along with a flaperon or rudder failure do show some improvement in both the pitch angle, theta, and pitch rate, q. As with the single failures, the pitch doublet does result in a slight departure in the roll angle, phi, which was deemed to be insignificant. The remaining failures (i.e. those which do not include the failure of a stabilator) show very little degradation in doublet response in either the system with, or the system without Control Redistribution.

4.3.3.2.2 Roll Doublet

The roll doublet simulations resulted in perhaps the most significant amount of improvement from the use of Control Redistribution. For this doublet, failures of the flaperons and rudder create the greatest amount of difficulty. This difficulty stems from the fact that the flaperons are utilized differentially in conjunction with the rudder to generate the roll manuever, while the stabilators, which are normally operated only in unison to create a pitching moment, are not.

All seven of the simulations which include flaperon failures show at least some improvement from Control Redistribution in the lateral channel (center column of subplots). Comparison of Figures D-22 and 23 with D-32 and 33 (left stabilator with either flaperon) reveals that, while the simulation which does not utilize Control Redistribution shows very little deviation in the roll rate, p, yaw rate, r, and lateral

acceleration, a_y, the simulation utilizing Control Redistribution is even closer to the no-failure case. Sideslip angle, ß shows a tremendous improvement from Control Redistribution, which can be accounted for by the lack of adverse yaw as discussed in Section 4.2.2.2. In this case, the adverse yaw is not being created by a rudder failure, however, the rudder is now being utilized to help compensate for the other failures, making its original job in the roll doublet more difficult. The roll angle of the simulation utilizing Control Redistribution, while showing the characteristic departure, shows much more fidelity to the original, no-failure, response. Either flaperon failing with the right stabilator (compare D-25 and 26 with D-35 and 36) show similar results.

Perhaps the most challenging scenario for the roll doublet input, is when both flaperons have failed. This result was quite expected, for as stated previously, the flaperons are the primary surfaces utilized to generate the roll manuever. These plots can be seen in Figures D-28 and D-38. In this case it is helpful to examine the dither response of each simulation as well as the doublet. Comparison of the stabilator responses of the impaired simulation with the basis simulation (no failures) shows that Control Redistribution has these actuators working very hard to try and produce the same doublet, as well as dither response. This is evident in the plots of roll rate, p, yaw rate, r, and lateral acceleration, a_y, where it can be seen that the simulation utilizing Control Redistribution not only better approximates the doublet response, but also has a very high level of fidelity to the dither response. The exception to this is the initial doublet response in the yaw rate, r. Here it is seen that r takes a large jump in the simulation using Control Redistribution. While no explanation for this jump was obtained, it should be noted that the simulation converges very quickly to the unimpaired response. Perhaps connected with this anomaly is the strong rise seen in the sideslip angle, ß, at approximately two seconds (the time of the second failure insertion). This jump also corresponds to a brief rise in the right flaperon response, which occurs just prior to its failure at two seconds. It is thought

that perhaps the left flaperon is responding to the failure of the right flaperon at this point, for as was seen in the single failure case, these actuators are effectively nulled out when their counterpart fails. This action prevents the dither from causing any unwanted roll and yaw from acting on a single flaperon. Since the nullification is not instantaneous, the right flaperon *does* receive a small positive dither pulse just before its two second failure point. This pulse could explain the sideslip and yaw jumps seen in this simulation. Note that, as with the yaw angle, the sideslip angle quickly converges to the response generated by the basis simulation.

4.3.3.2.3 Yaw Doublet

The simulations resulting from the input of a yaw doublet to both systems with and without Control Redistribution can be found in Figures D-41 through D-60. Examination of the no-fail simulation, represented by the dashed line on all of these figures, reveals that the majority of this doublet is created with the rudder, and that a small part is also played by the flaperons. Thus it is not surprising to see very little difficulty encountered by either system when the two induced failures do not include the failure of the rudder. The first simulations which combine a stabilator failure with a rudder failure are shown in Figures D-44, D-57 (no Control Redistribution), D-54 and D-57 (with Control Redistribution). The results of these simulations were somewhat surprising, a total lack of command response from either system. As discussed previously, it was thought that Control Redistribution would cause the flaperons to generate a yaw response. In this scenario (i.e., a stabilator and rudder failure) it is thought that the majority of the flaperon's control authority is being used to assist the remaining stabilator in maintaining the desired pitch angle, theta, thus preventing their use to generate the yaw response. In Figures D-49, D-50, D-59, and D-60, the yaw response of each system with a flaperon and rudder failure is shown. In these scenarios, it can be seen that, while the system utilizing Control

Redistribution does not reproduce the same amplitude as the no-fail response, there is, at least, a response, which is more than what can be said for the system which does not use Control Redistribution (Figure D-49). It is thought that the reduced amplitude is a direct result of the scaling which has been incorporated to prevent actuator rate saturation (see Section 4.3.3.1). It is hoped that this small amount of control authority will at least assist the pilot in the landing of an aircraft with this class of damage.

The simultaneous failures of *both* flaperons also provided interesting results for the yaw doublet. Figures D-48 and D-58 show that, while the system with no control Redistribution does provide a response to the yaw doublet, this response appears to diverge in every lateral channel variable shown in the center column. In contrast, the system using Control Redistribution is almost identical to the no-fail simulation. This is perhaps the most notable improvement attributable to Control Redistribution.

4.4 Chapter Summary

This chapter presents the data and subsequent analysis resulting from computer simulations incorporating the algorithms discussed in the first three chapters. The chapter begins by introducing the reader to the various forms of data presentation (i.e., probability and output plots) and proceeds to recreate the single failure analysis which was performed by Stepaniak [47]. The dither algorithm developed by Stepaniak is presented, and the results are compared to the single failure analysis of Eide [9].

The effects of Control Redistribution are examined in the single failure simulation, and output plots are generated and compared to the same simulations on a system which does not incorporate Control Redistribution. These output plots have been placed in Appendix A, and are the result of applying separate yaw, pitch and roll doublets.

Various aspects of the double failure analysis are then presented. First, the many changes to the MMAE and Control Redistribution algorithms, as well as changes in the probability plots, are presented. Next, a basis simulation is performed using Stepaniak's new dither and *no Control Redistribution*. The results of this simulation are included in Appendix B and are compared to those obtained by Eide for double failures.

With the basis simulation established, the simulations which tested double failures against a system using Control Redistribution are then presented. Problems which arose due to rate saturation are presented, along with the scaling technique developed to compensate for these problems. The resulting probability plots are included in Appendix C. The output plots comparing the doublet responses to double failures in both the systems which did and did not incorporate Control Redistribution are presented in Appendix D. An analysis of these simulations is presented which shows the benefits provided by Control Redistribution in the face of double actuator failures.

5. Conclusions and Recommendations

5.1 Chapter Overview

In this chapter the conclusions are presented which pertain to the performance of the Multiple Model Adaptive Estimation with Control Redistribution (MMAE/CR) algorithm in failure detection and compensation for the VISTA F-16 aircraft. These conclusions are based upon the results which have been presented in Chapter 4, which, in-turn, are the results of the theories and concepts presented in Chapters 1 through 3. Section 5.2.1 presents the conclusions which can be made from the analysis of the MMAE-based Block 40 controller against single failures of both actuators and sensors. Conclusions reached from the utilization of Control Redistribution against the same set of single failures is then presented in Section 5.2.2. A summary of the conclusions reached in the double failure analysis is then presented in Section 5.2.3.

Section 5.3 presents some of the many possible recommendations for further research in this area. Initially, Section 5.3.1 discusses the need to explore gain scheduling, in an effort to develop an algorithm that is not so sensitive to changes in flight conditions. This section is followed by recommendations for real-time simulations of this algorithm, and suggests current AFIT Graduate Computer Science research projects which could quite feasibly be utilized to accomplish this task. Section 5.3.3 presents suggestions for alternative failure mode investigations. The chapter ends with the ultimate recommendation for in-flight evaluation of these concepts.

5.2 *Conclusions*

5.2.1 *Single Failures - MMAE*

The compensation for single failures using MMAE techniques has been thoroughly researched [9, 10, 11, 19, 20, 31, 32, 44, 47]. Chapter 2 presented some of the history behind this development, as well as recent efforts at AFIT to incorporate the technique into high performance aircraft such as the VISTA F-16. The results of MMAE against single failures in this research are readily apparent in the data presented in Chapter 4. In that chapter it can be concluded that a properly tuned MMAE can do an excellent job of detecting failures and properly estimating the system's state variables in the presence of these failures. As presented in Chapter 2, the MMAE design accomplishes this by incorporating a bank of Kalman filters, each designed based upon a specific failure hypothesis. In the case of a sensor failure, the particular filter brought on line to estimate the various states of the system can do so without the failed sensors measurements, because it was designed to utilize the information provided by the remaining *healthy* sensors. In the event of an actuator failure, the filter which is based upon that failure is *expecting* the type of abnormal response which may result. The purpose of sensor information is to provide the conventional Block 40 Flight Control System with the necessary information to determine the appropriate controls to apply to the aircraft. Normally, if the raw information from one or more sensors is not present, the control system cannot generate the commanded controls completely or effectively. Because the MMAE *can* provide accurate estimates of these normally measured variables *without* the information of one or more sensors, it can be said that it effectively *compensates* for sensor failures. The failure of an actuator will usually cause certain difficulties in the estimation of the system's state variables in a nonadaptive filter, for changes in the system's dynamic response may not be interpreted properly. In this sense the MMAE also provides some

compensation for actuator failures, i.e. state variable estimates which would normally be corrupted by the failure of an actuator can now be estimated rather precisely. This form of actuator compensation should not be confused with the type of *control authority* compensation which results from Control Redistribution, for while the MMAE can provide accurate state *information* in the presence of actuator failures, it does nothing to redirect, or otherwise change, the commands being sent to the aircraft's actuators. Figure 4-2 shows that the MMAE can provide detection of most single failures within .25 seconds, and all within 1.25 seconds. This figure also shows the excellent ability of the algorithm to maintain these declarations.

It can also be concluded that the capabilities of the MMAE are extremely dependent upon two distinct factors. The first of these factors relates to how well the nonlinear system is modeled. This pertains not only to the truth model used to evaluate the algorithms, but also to the linear approximation which is used to design the linearized Kalman Filters of the MMAE and to develop the redistribution matrices for the Control Redistribution algorithm. The second factor concerns the identifiability of the system. With this system, many problems have been resolved by altering the dither signals used to assist in the detection of failures. These improvements are readily observed by comparing the results of Stepaniak's dither [47] in Figure 4-2, with those of Eide [9] in Figure 4-3. Differences in the ability of the algorithm to detect failures of similar components (e.g., left flaperon vs. right flaperon) leads to the conclusion that improvements may still be necessary in future research.

5.2.2 Single Failures - Control Redistribution

Incorporation of Control Redistribution into the MMAE provides yet another level of performance improvement in the face of single failures. Results in Chapter 4 show that, because of Control Redistribution, the algorithm now actually has the ability to *compensate*

for failed actuators. This provides an excellent complement to the MMAE, which in effect compensates for failed sensors by replacing raw measurements as inputs to the Block 40 Flight Control System with optimal adaptive estimates of those measurements. Now Control Redistribution compensates for actuator failures by *redistributing* the commands intended for a failed component to those which remain healthy. The results of this concept against single failures (presented in Section 4.2) show that Control Redistribution does indeed provide a significant level of control authority that would otherwise not be available. The added benefit of Control Redistribution is that it also redistributes the dither signal to the remaining healthy actuators. This has the effect of exciting the system with the single failure in much the same way as the system was when it was fully functional. The end result is that the system is now better prepared to detect and respond to a second failure.

5.2.3 Double Failures - MMAE/CR

The advancement of the MMAE/CR algorithms to allow the detection and compensation of double failures provided some very promising results. These results required several modifications to the algorithms, all of which are presented in Chapter 4. To provide a basis upon which to judge the effects of Control Redistribution against double actuator failures, simulations using MMAE-based Block 40 controller without Control Redistribution were performed. These simulations were extremely exhaustive, since simulations examining each failure (11 total) in combination with all other possible secondary failures (10 total) had to be performed, and each case required several different Monte Carlo runs (at least 10). While these results served mainly to develop and analyze the Control Redistribution portion of the algorithm, they were also evaluated and compared to the results obtained by Eide [9]. This comparison, detailed in Table 4-9 and Section 4.3.2, showed a significant improvement in the ability of the MMAE to detect and maintain

probability lock on secondary failures. This improvement is directly attributed to the improved dither algorithm developed by Stepaniak [47].

Once the basis simulation was complete, an attempt at incorporating the Control Redistribution was performed. Initially the results were disastrous. Failure to heed the warnings of Stepaniak concerning possible rate saturation of actuators resulted in extremely poor secondary failure probability lock when both failures were actuators. This problem is quite obvious in Figure 4-8. On the bright side, these simulations did show a much improved detection time of the second failures. This was especially true for secondary sensor failures. The improvement was expected, and attributed to the fact that the dither was now being re-routed from the failed actuator to the remaining healthy actuators, providing a level of excitation closer to that of the original fully functional system, thereby maintaining identifiability of the second failure. After considerable investigation into the probability lock problem, it was determined that the problem was indeed rate saturation of the actuators resulting from the load of *two* failed actuators being redistributed to the remaining healthy actuators. To remedy this situation, various levels of command scaling were incorporated. Analysis of these levels showed that a definite tradeoff exists for any attempt to reduce the amount of redistribution (i.e., decreasing Control Redistribution helped the rate saturation problem, but reduced the improvements sought in controllability). Once an acceptable set of redistribution levels was obtained (see Table 4-11), simulations were performed to evaluate the failure detection *and* compensation capabilities of the algorithm. These results, placed in Appendices C and D and detailed in Sections 4.3.3.1 and 4.3.3.2, show that, while the double failures seriously challenge the MMAE/CR algorithm, the improvements over the system utilizing only MMAE are quite noteworthy. As noted in Chapter 4, there are certain double failures for which no amount of compensation will be satisfactory. This was definitely the case for the double failure of both stabilators, which resulted in an unstable aircraft for all command inputs, and both

system configurations. Along with this realization, there were also some pleasing discoveries. The most notable of these was the simulation showing the yaw doublet response in the case of the failure of both flaperons. This simulation resulted in an unstable aircraft for the system which did not incorporate Control Redistribution, and a very responsive aircraft for the system that did. Several other simulations demonstrated that Control redistribution can provide a limited amount of control authority in the presence of most double actuator failures.

5.3 Recommendations

5.3.1 Gain Scheduling

Throughout this research the limitations resulting from the single constant gain of each Kalman filter were quite apparent. In Eide's research [9], a study of the MMAE's robustness showed that it could not perform as expected if small changes occurred in the aircraft's flight condition. These effects are most apparent in probability plots such as Figure C-10. Here it can be seen that, as the simulation progresses, the ability of the MMAE to maintain the correct probability lock is degraded more and more. This is thought to be occurring as a result of the Control Redistribution algorithm's attempt to compensate for the specific failure. Considering the example of a rudder failure of Figure C-10, the Control Redistribution algorithm will attempt to compensate by utilizing the flaperons and stabilators. While this compensation will result in a compensating effect, it will also cause the aircraft to change its position within the flight envelope, i.e. changes in both speed and altitude will occur. If these changes deviate significantly from the point at which the Kalman filter was designed, the ability of the MMAE will be impaired. The development of a gain look-up table for various points within the flight envelope could possibly remedy

this situation: the gain of each elemental filter in the MMAE algorithm could be gain-scheduled, just as the controller gains of the Block 40 Flight Control System are.

5.3.2 Identifiability Issues

As discussed in Section 5.2.1, the identifiability of the system will greatly affect the failure identification capabilities of this algorithm. It has been demonstrated in this, and previous research, that in a benign flight condition (i.e., trimmed, with no pilot commanded input), accurate identification of sensor and actuator failures can be difficult. This problem is intensified when more than one failure occurs. It has also been shown that a carefully constructed dither signal, applied to the actuators to "shake up" the system can improve this situation tremendously. But problems still exist. Data presented in this research shows that improvements in the dither signal are still needed. Particularly, it can be seen that the failures of the two flaperons are detected differently, perhaps suggesting that the *phase* relationship between the signals sent to these two actuators is not quite optimal.

Another identifiability issue worthy of future research concerns commanded inputs. Eide [9] questioned the necessity of the dither signal, speculating that a periodic, pilot-commanded doublet could provide the necessary excitement to the system. If this were indeed true, the questions concerning the true subliminality, and effectiveness, of the dither signals could be of reduced concern. The fact that this method of system excitement would require a deliberate pilot comanded input in order to detect failures during steady level flight is also a valid point. However, it can also be argued that the most crucial time for adequate failure detection (and subsequent compensation) is *not* during benign flight, but during a pilot-commanded maneuver. Before this debate can continue, investigations into the failure detection capability of the system *during* the commanded doublets should be performed. If the system is able to detect component failures adequately*without* a dither input, in the

presence of certain maneuvers, then the issue of deciding *when* dither is actually required, and when it can be shut off can be further investigated.

5.3.3 Real-Time Simulation

A logical next step would be to integrate the algorithm into a real time simulation. This would provide a means of qualitatively evaluating the performance of MMAE/CR at a higher degree of accuracy. This is not to say that the SRF VISTA does not do an excellent job of providing a nonlinear truth model for the aircraft, but a real-time simulation would allow the incorporation of an actual "man-in-the-loop", and perhaps even a human who has had some actual experience in flying this aircraft. If successful, this type of research could provide an enormous amount of momentum in the implementation of this system in an actual aircraft.

Research into this area has revealed that an extensive amount of work is being performed in the Department of Electrical and Computer Engineering at AFIT in the development of a "virtual cockpit" designed specifically for the F-16. Adams [1] has developed this simulation with an impressive amount of detail and fidelity, and has been able to solicit the recommendations and feedback of actual F-16 pilots. While the truth model used for this simulation is not the SRF VISTA, and it utilizes the C programming language rather than FORTRAN, future research efforts could be made to resolve these differences, resulting in a product of truly exceptional proportions.

5.3.4 Alternative Failure Modes

The results of this research in its investigation of single and double failures has by no means completed the analysis of this algorithm. In this research, only fully failed

actuators and sensors were investigated. What remains to be accomplished is the investigation into the detection and compensation of partial failures.

Eide [9] investigated the performance of the MMAE against partial sensor failures, modeling them as increases in measurement noise, or as the appearance of a measurement bias. This analysis resulted in a relatively a limited amount of success. It has been suggested by Maybeck [37] that perhaps a way to model these failures to ensure good MMAE performance would be to multiply a row of the **H** matrix (see Equation (2-35)) by a scalar between 0 and 1, resulting in a failure model which maintains the measurement noise upon which the filters were based, and yeilding a system status hypothesis that can truly lie within the parameter space of the MMAE. This would yield models very analogous to those used to model partial failures of actuators, with which MMAE identification has been consistently superior.

Partial failures of actuators can be modeled in much the same way, by multiplying a row of the **B** by a similar scalar. This concept is discussed in Section 3.5. Other considerations for the actuator failures are also worth considering. These include the actuator that becomes stuck in a non-free-stream position, actuators which fail intermittently, and actuators which are completely or partially obliterated from the aircraft (and thus associated with different stability derivatives and trim values than those depicted in this research).

5.3.5 In-Flight Evaluation

As emphasized by Stepaniak [**47**], the concept of implementing the MMAE/CR algorithm on an actual aircraft is extremely attractive for two very important reasons. First, because the aircraft's original Block 40 control system is maintained in its entirety, the algorithm would be relatively easy and safe to implement. A full-scale Multiple Model

Adaptive Controller (MMAC) algorithm, in comparison, would require validation of the flight-worthiness of each of the separate control designs incorporated into its structure. Second, because the results of Control Redistribution are transparent to the fully functional aircraft, performance during the no-fail condition would not be affected in any way. During a failure, the Control Redistribution attempts to provide the same degree of performance and response as was seen with the fully functional Block 40 aircraft, thus providing a familiar feel to the pilot, and not trying to change or enhance any aspect other than controllability in the event of failure.

Appendix A

Output Plots of Single Failures

This appendix contains output plots for the analysis of single failures of the VISTA F-16 using a standard Block 40 controller preceded by an MMAE algorithm, both with and without control redistribution. Plots represent simulations which were performed using separate pitch, roll and yaw doublet inputs. Specific analysis of these plots can be found in Section 4.2.2.

This appendix is structured into three sections, each representing the response of the system to a either a pitch, roll or yaw doublet. Each section is subdivided into a simulation which does not utilize Control Redistribution, and one that does. Each of these subdivisions contains five plots, one for the failure of each actuator, and each plot contains 15 subplots representing the various longitudinal (first column) and lateral (second column) output variables of the system, along with the output positions of the five actuators (third column).

The subplots are once again further subdivided into two separate traces. The first trace shows the mean of the responses from 10 Monte Carlo runs of the simulation against a fully functional aircraft. This trace is represented by a dashed line, and its ± one sigma value is shown with a dashed dotted line. Note that if the variance is small, these lines will coincide and appear as solid. The second trace represents the mean of the responses from 10 Monte Carlo runs during the particular failure. This trace is represented by a solid line, with the ± one sigma value shown by a dotted line.

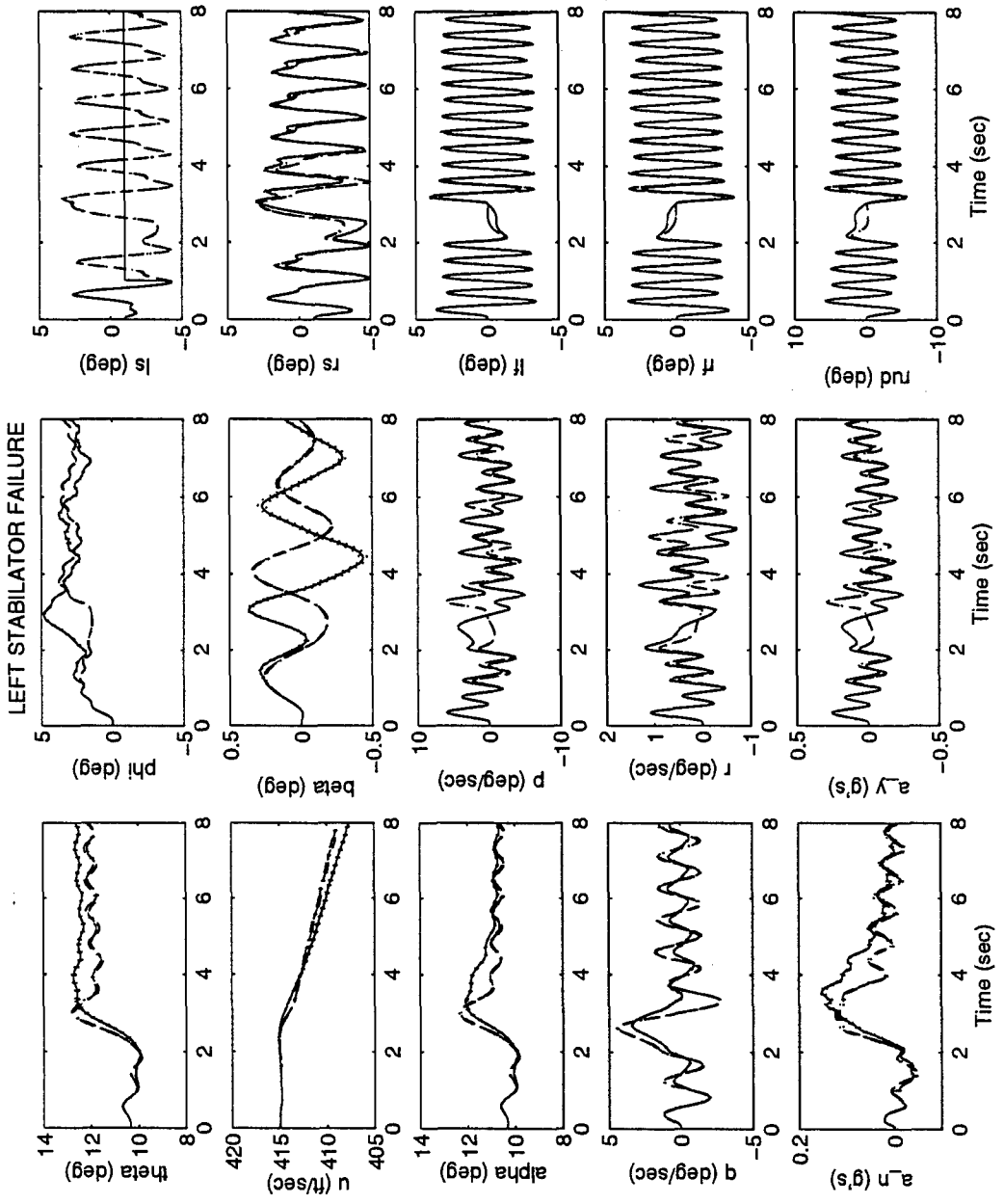

Figure A-1 MMAE-Based - Pitch Doublet - No Control Redistribution

Left Stabilator Failure

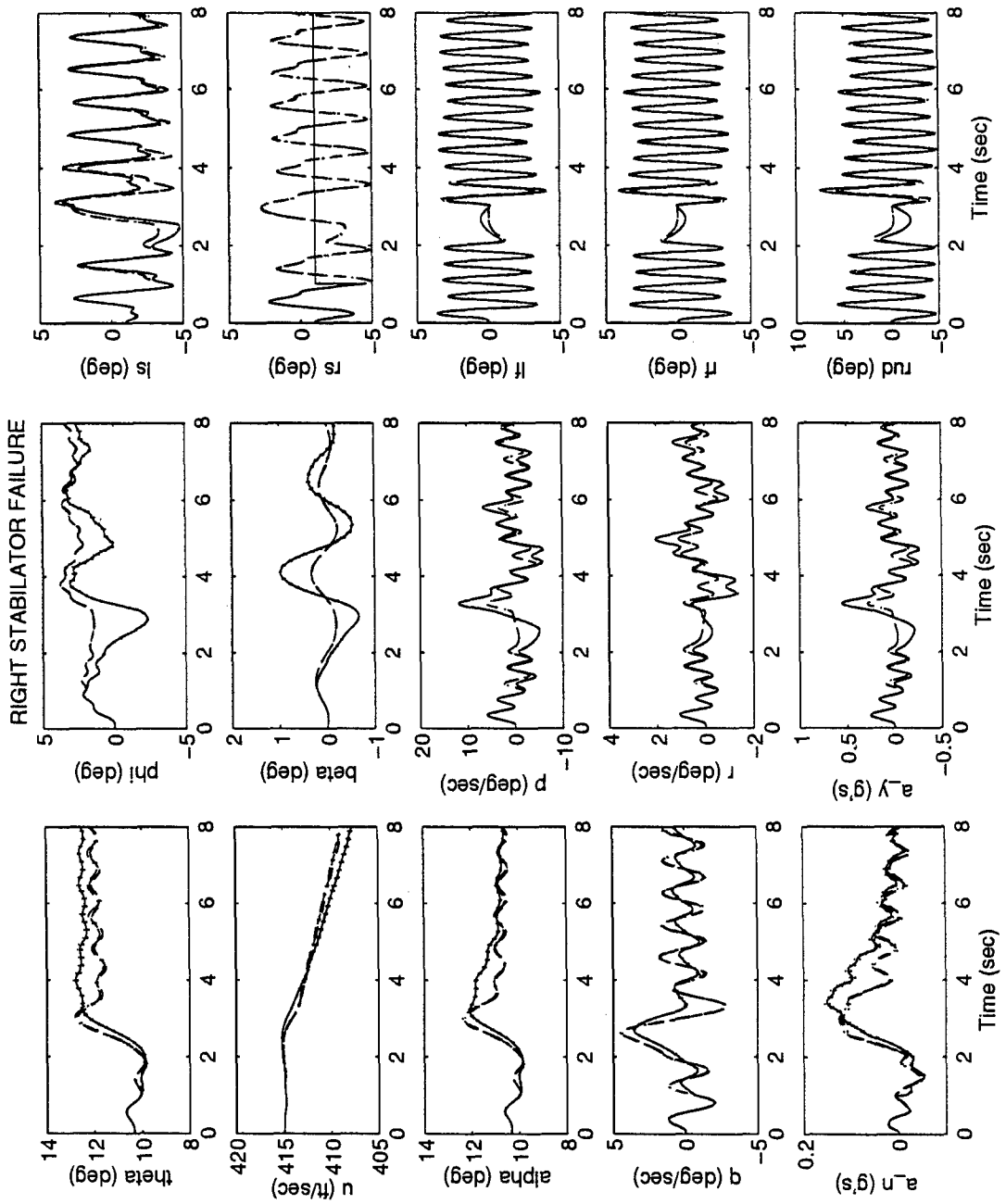

Figure A-2 MMAE-Based - Pitch Doublet - No Control Redistribution

Right Stabilator Failure

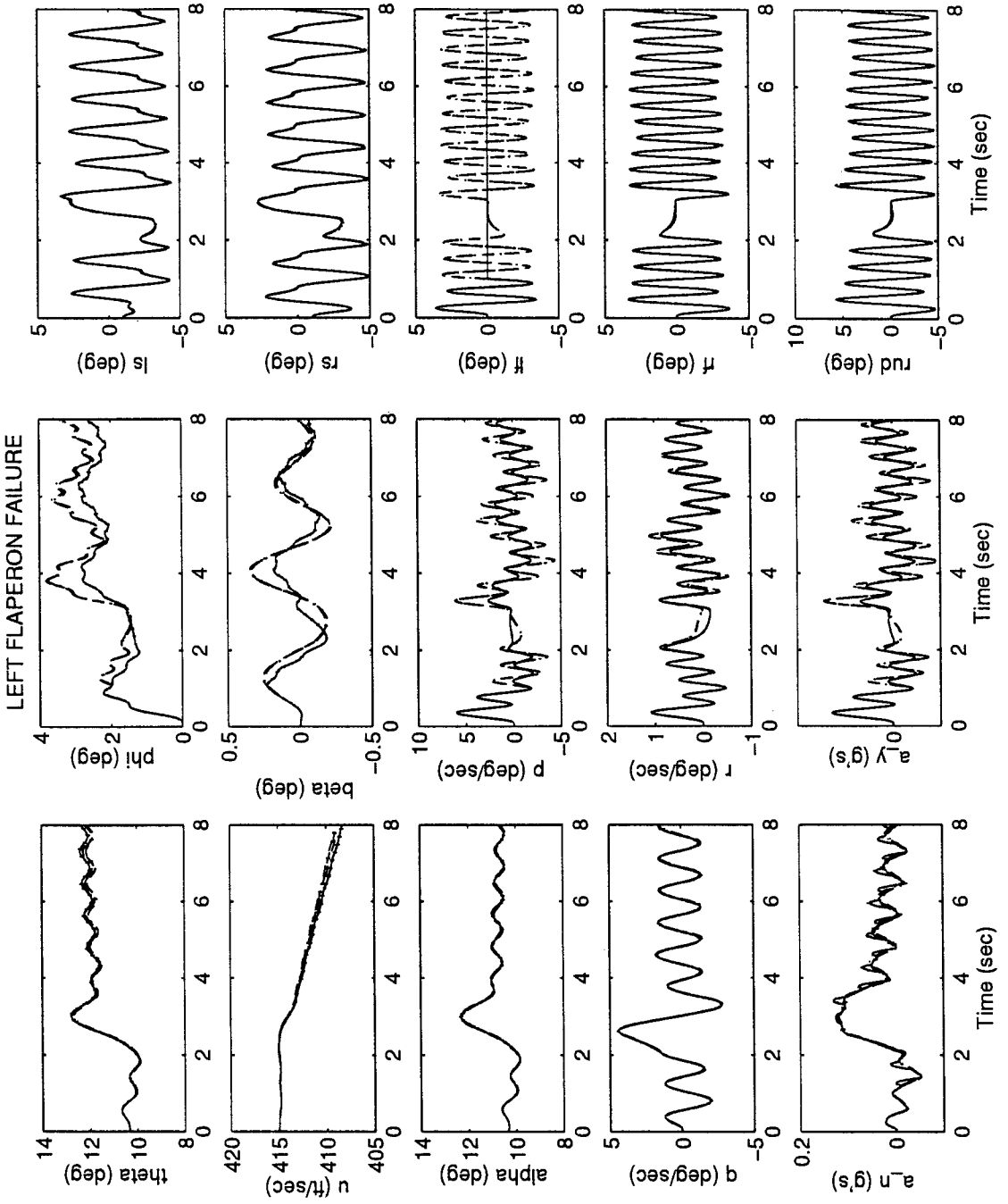

Figure A-3 MMAE-Based - Pitch Doublet - No Control Redistribution

Left Flaperon Failure

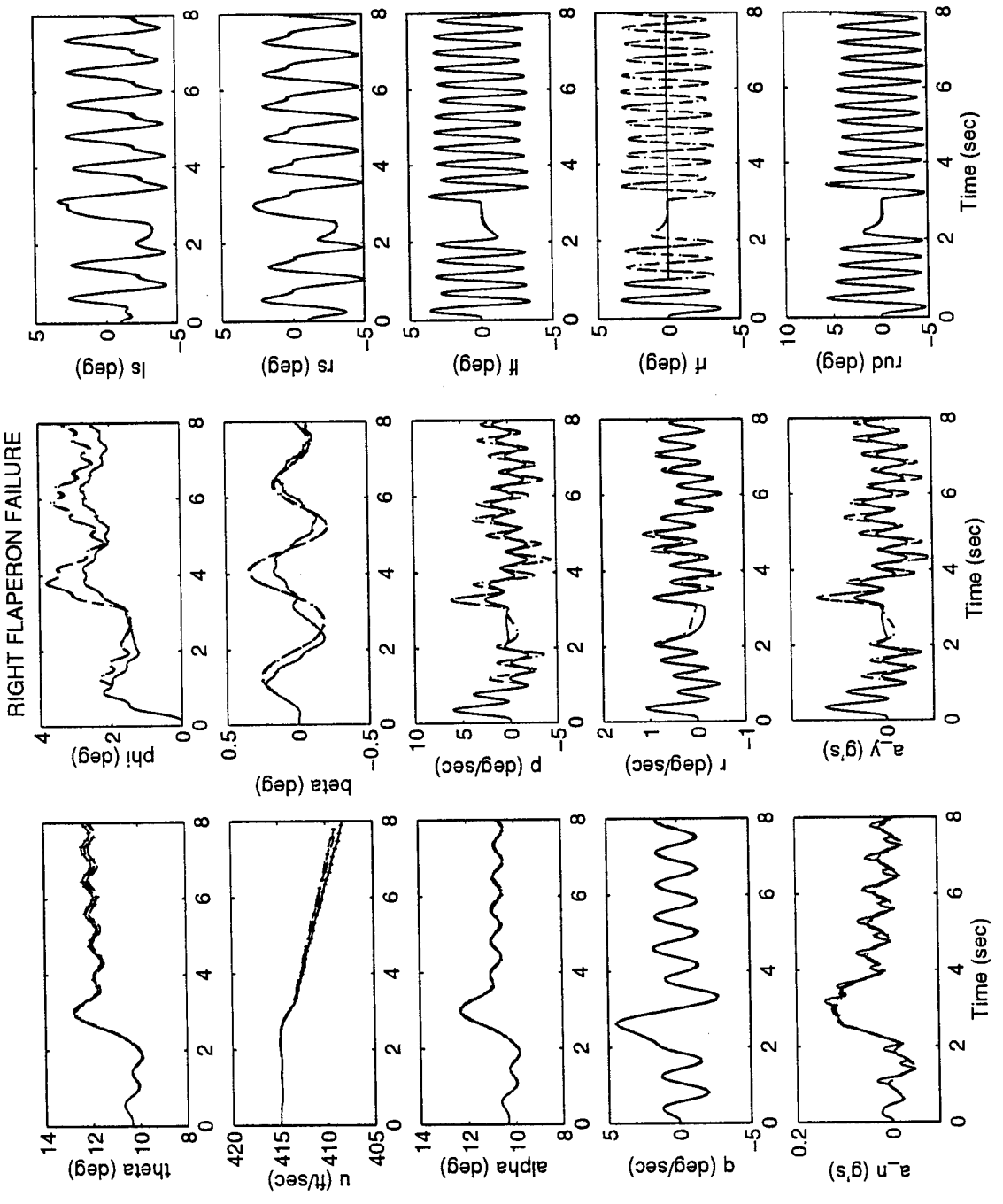

Figure A-4 MMAE-Based - Pitch Doublet - No Control Redistribution

Right Flaperon Failure

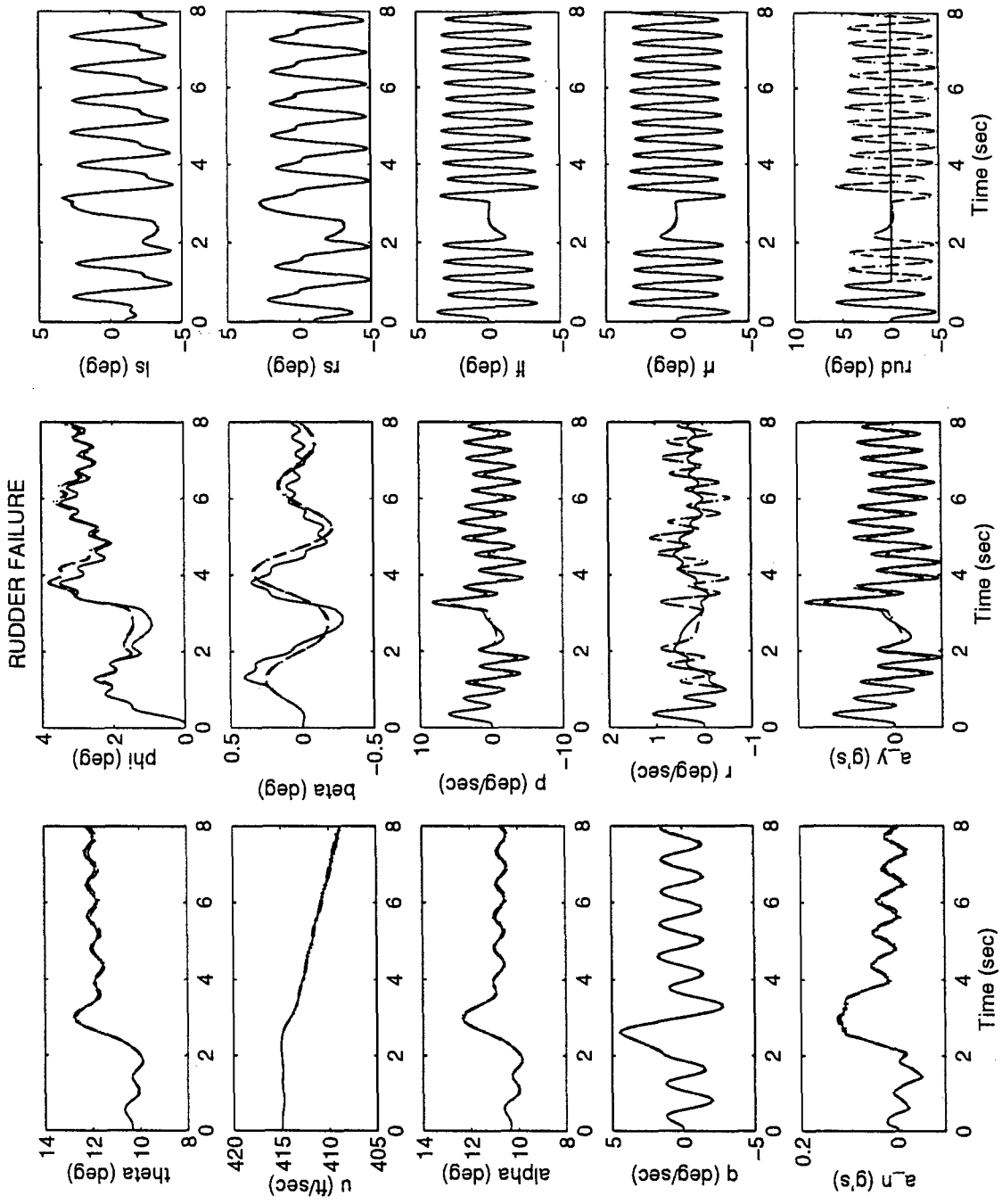

Figure A-5 MMAE-Based - Pitch Doublet - No Control Redistribution

Rudder Failure

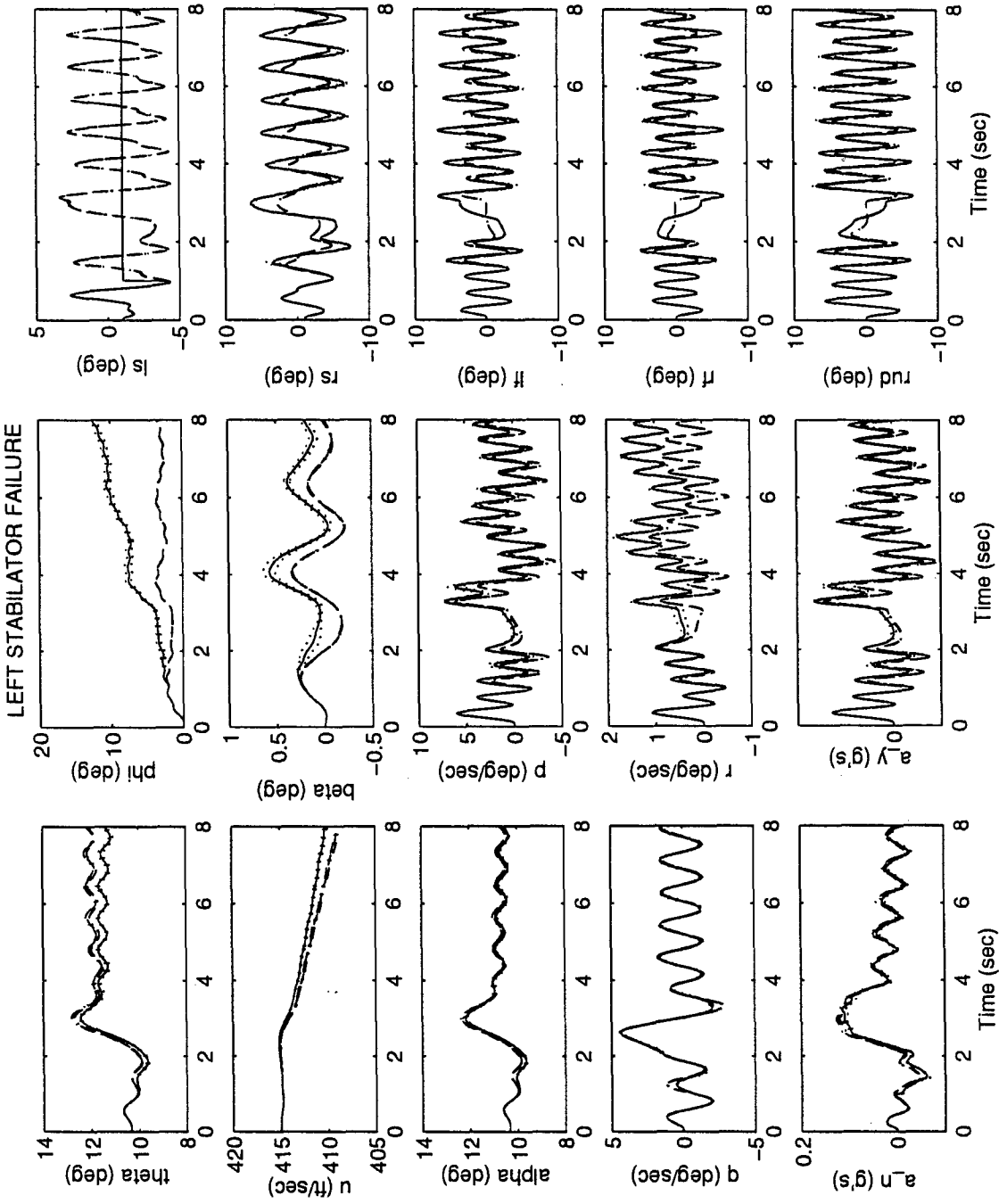

Figure A.6 MMAE-Based - Pitch Doublet - With Control Redistribution

Left Stabilator Failure

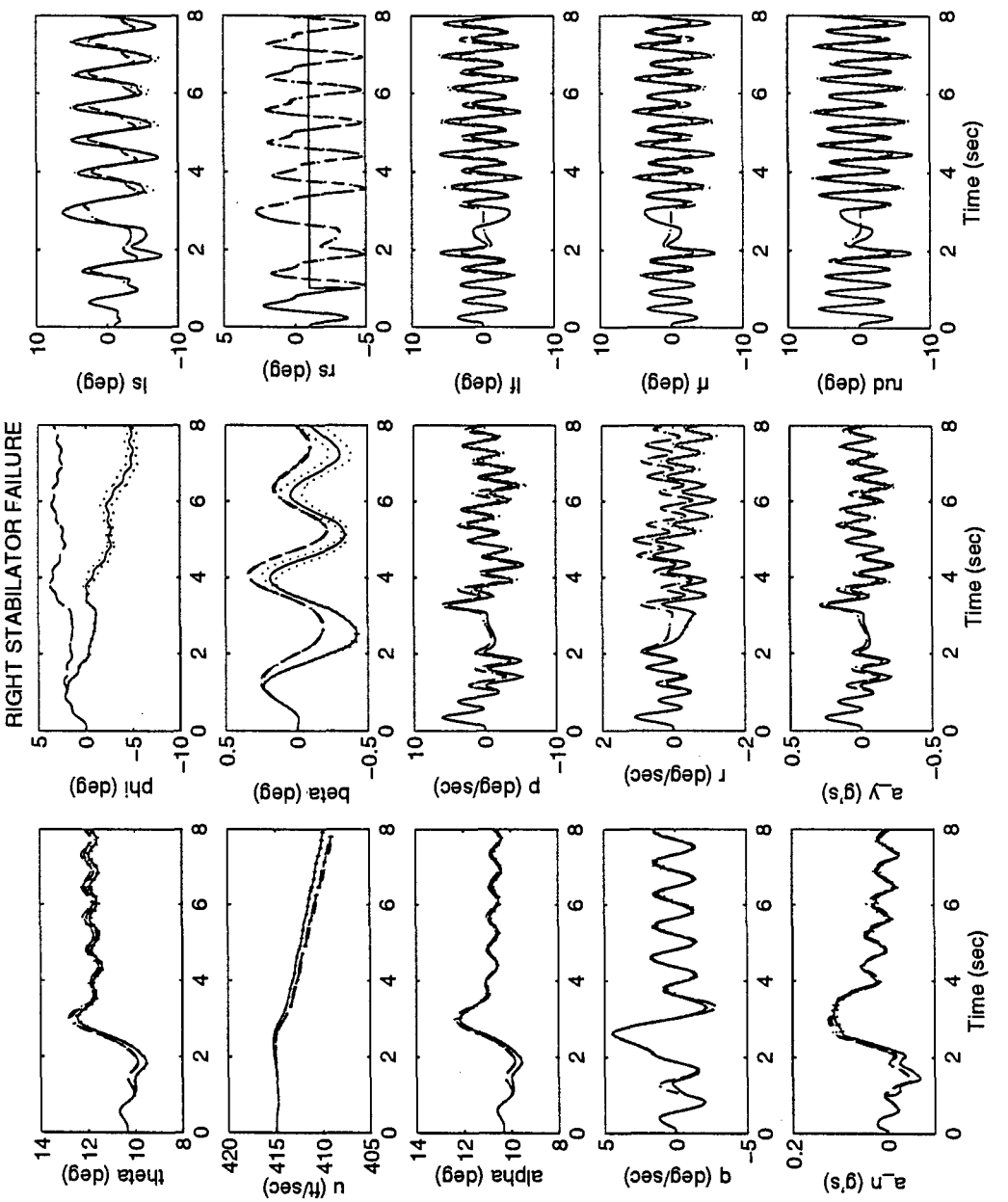

Figure A.7 MMAE-Based - Pitch Doublet - With Control Redistribution

Right Stabilator Failure

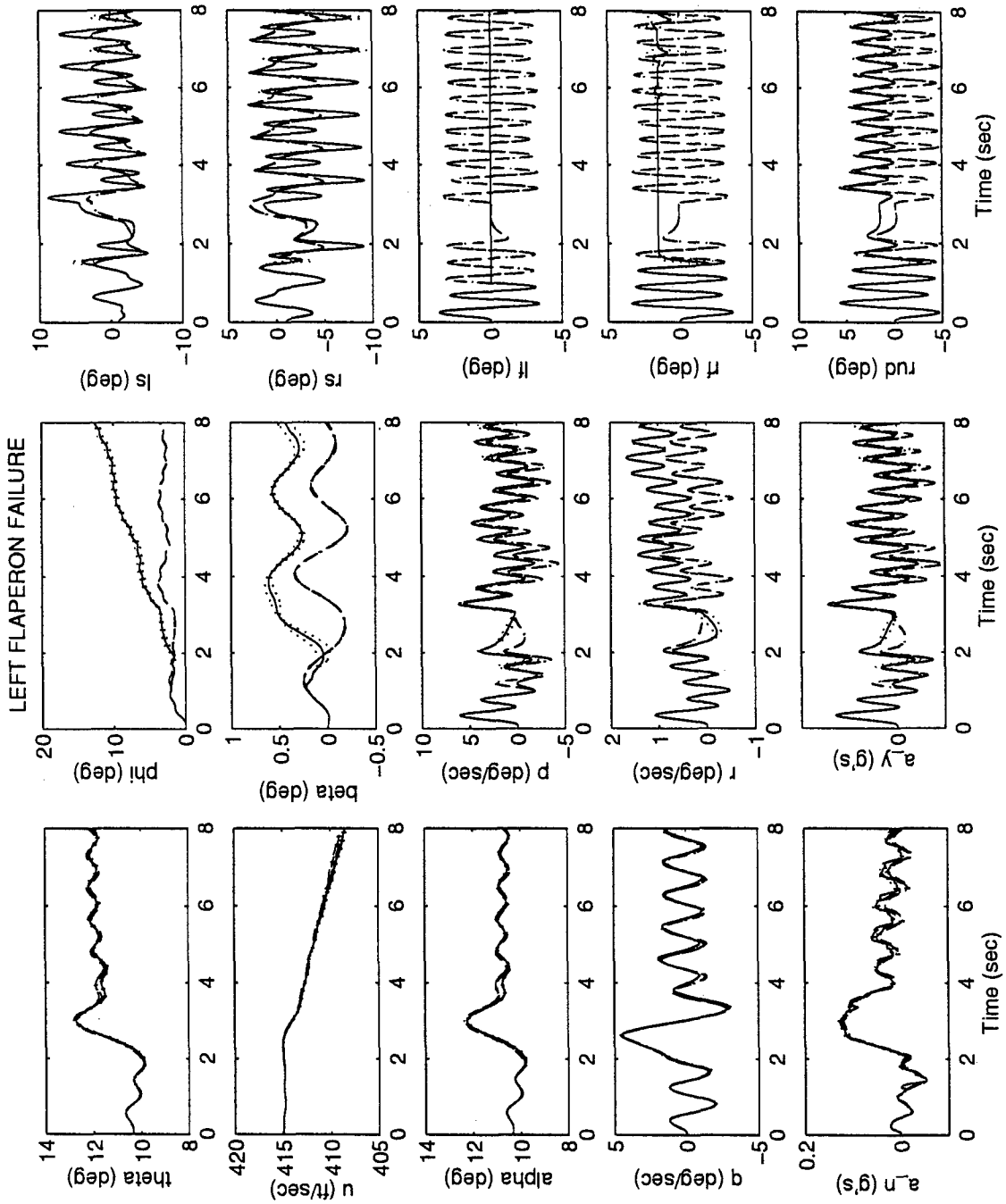

Figure A.8 MMAE-Based - Pitch Doublet - With Control Redistribution

Left Flaperon Failure

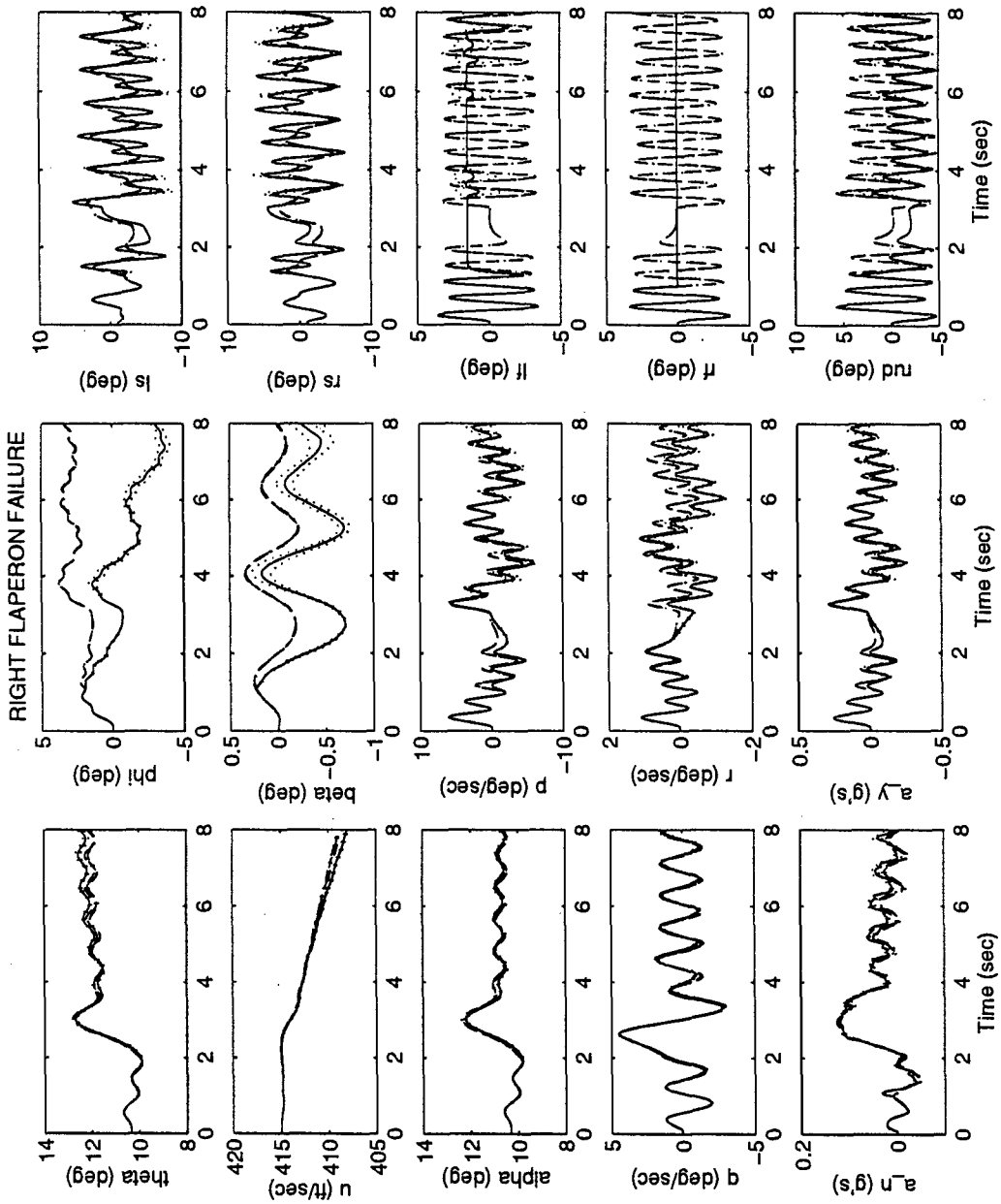

Figure A.9 MMAE-Based - Pitch Doublet - With Control Redistribution

Right Flaperon Failure

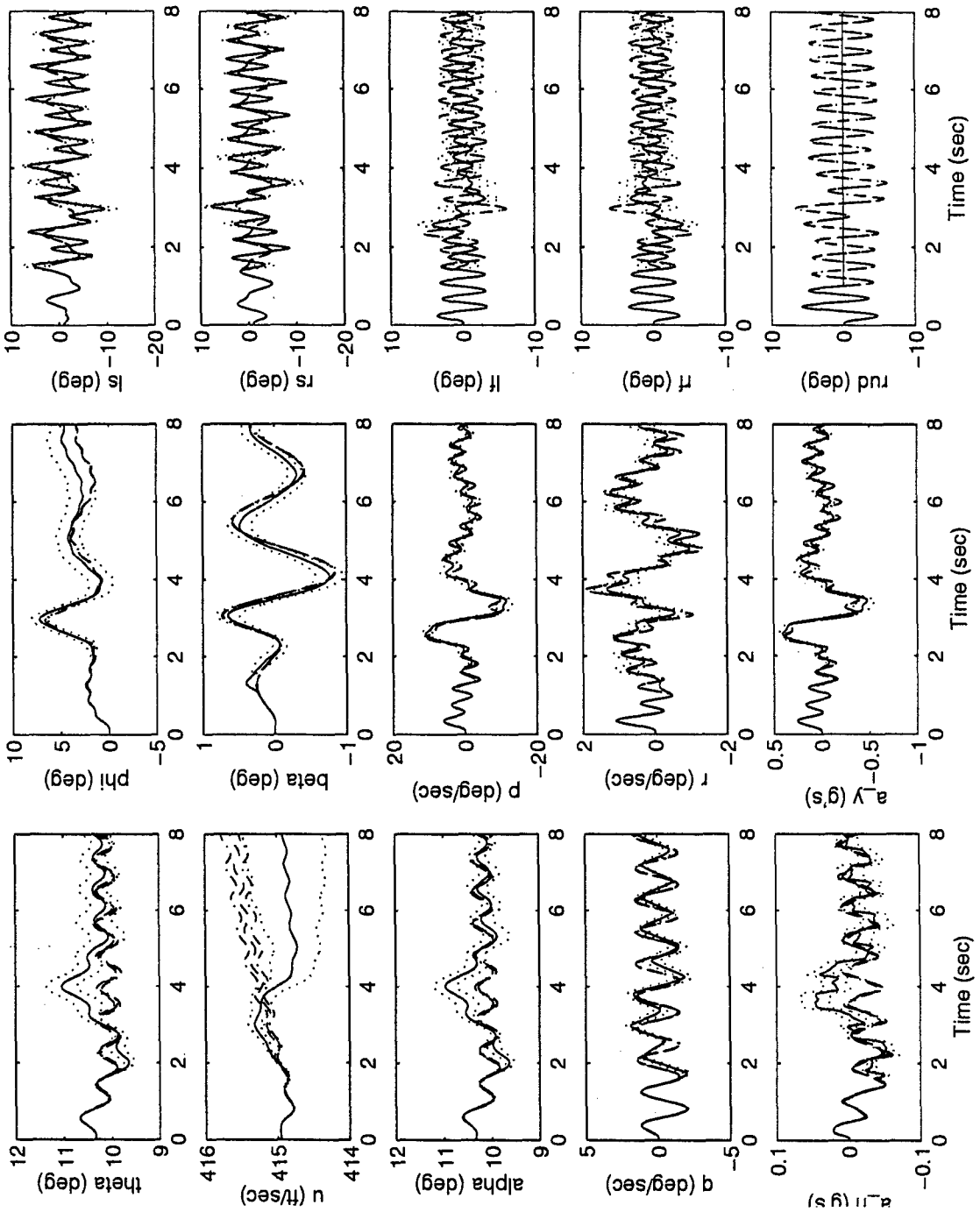

Figure A.10 MMAE-Based - Pitch Doublet - With Control Redistribution

Rudder Failure

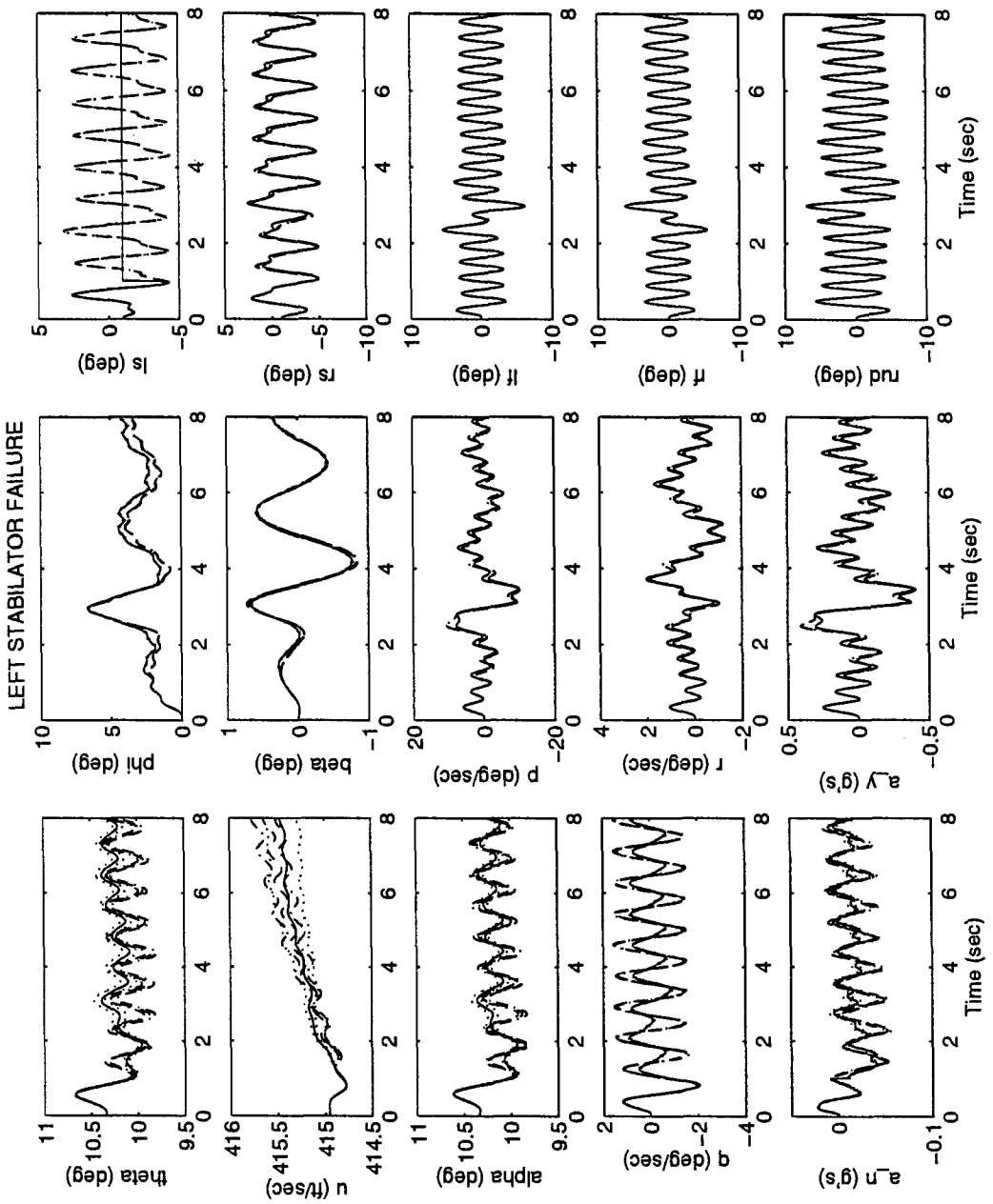

Figure A.11 MMAE-Based - Roll Doublet - No Control Redistribution

Left Stabilator Failure

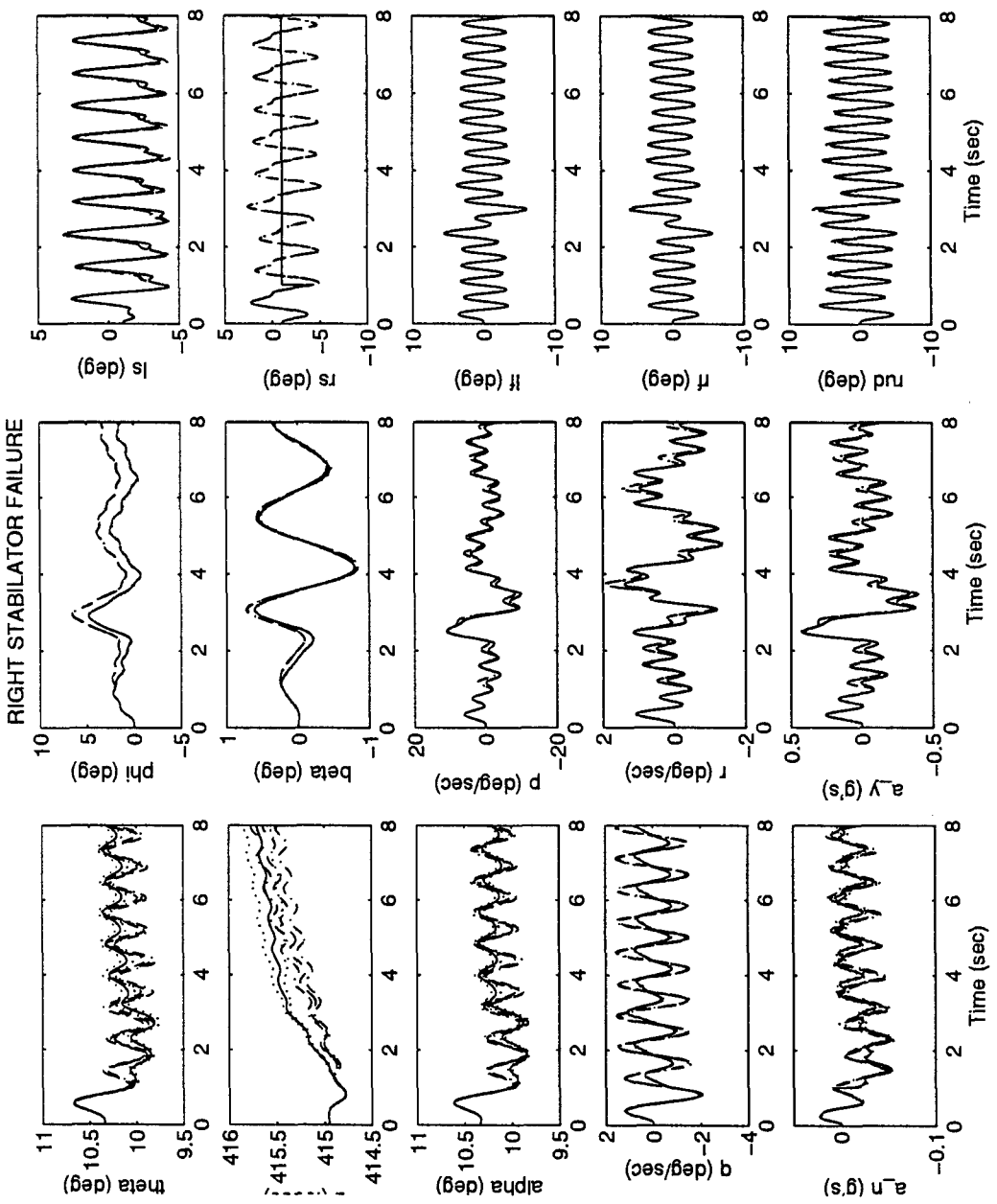

Figure A.12 MMAE-Based - Roll Doublet - No Control Redistribution

Right Stabilator Failure

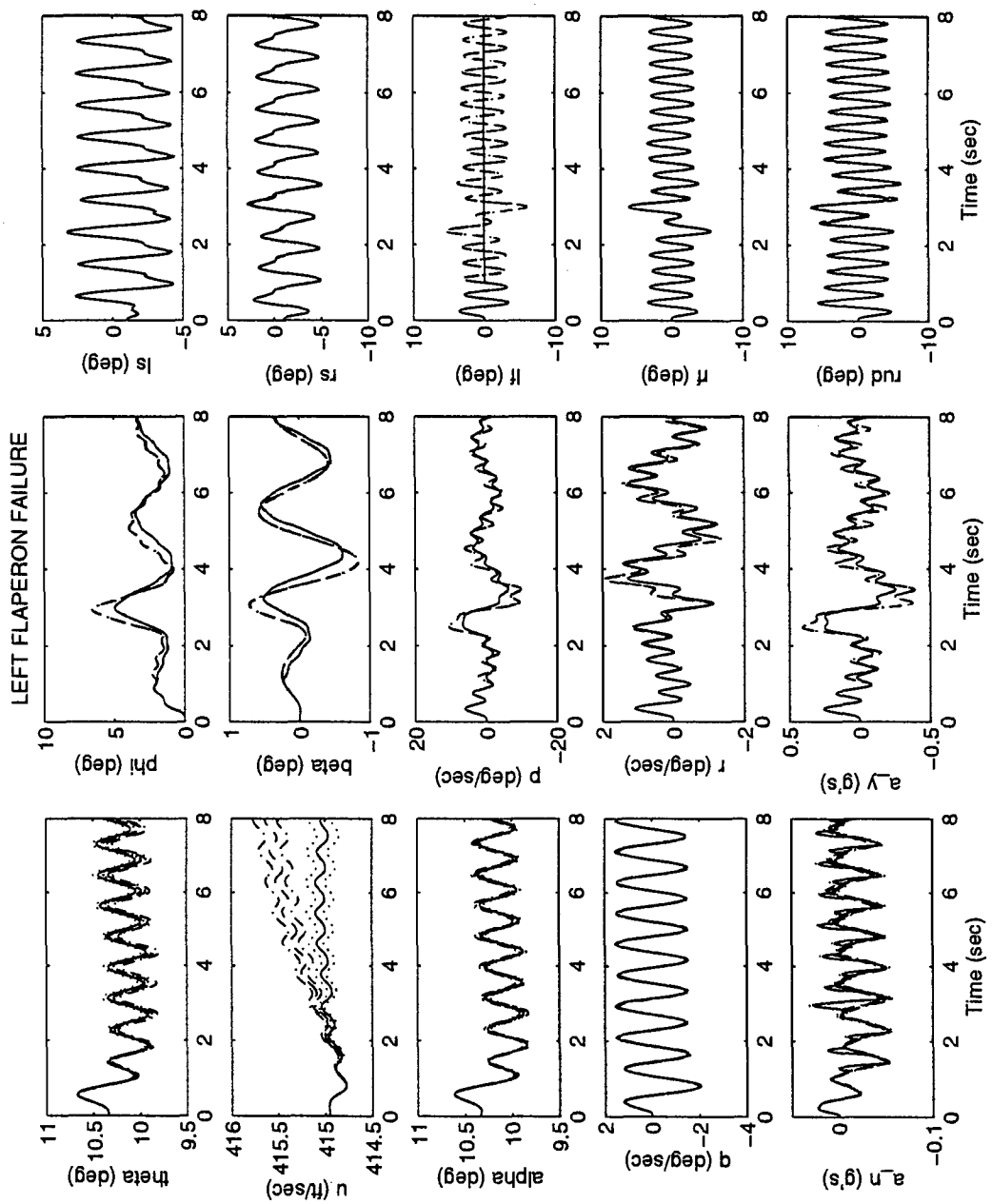

Figure A.13 MMAE-Based - Roll Doublet - No Control Redistribution

Left Flaperon Failure

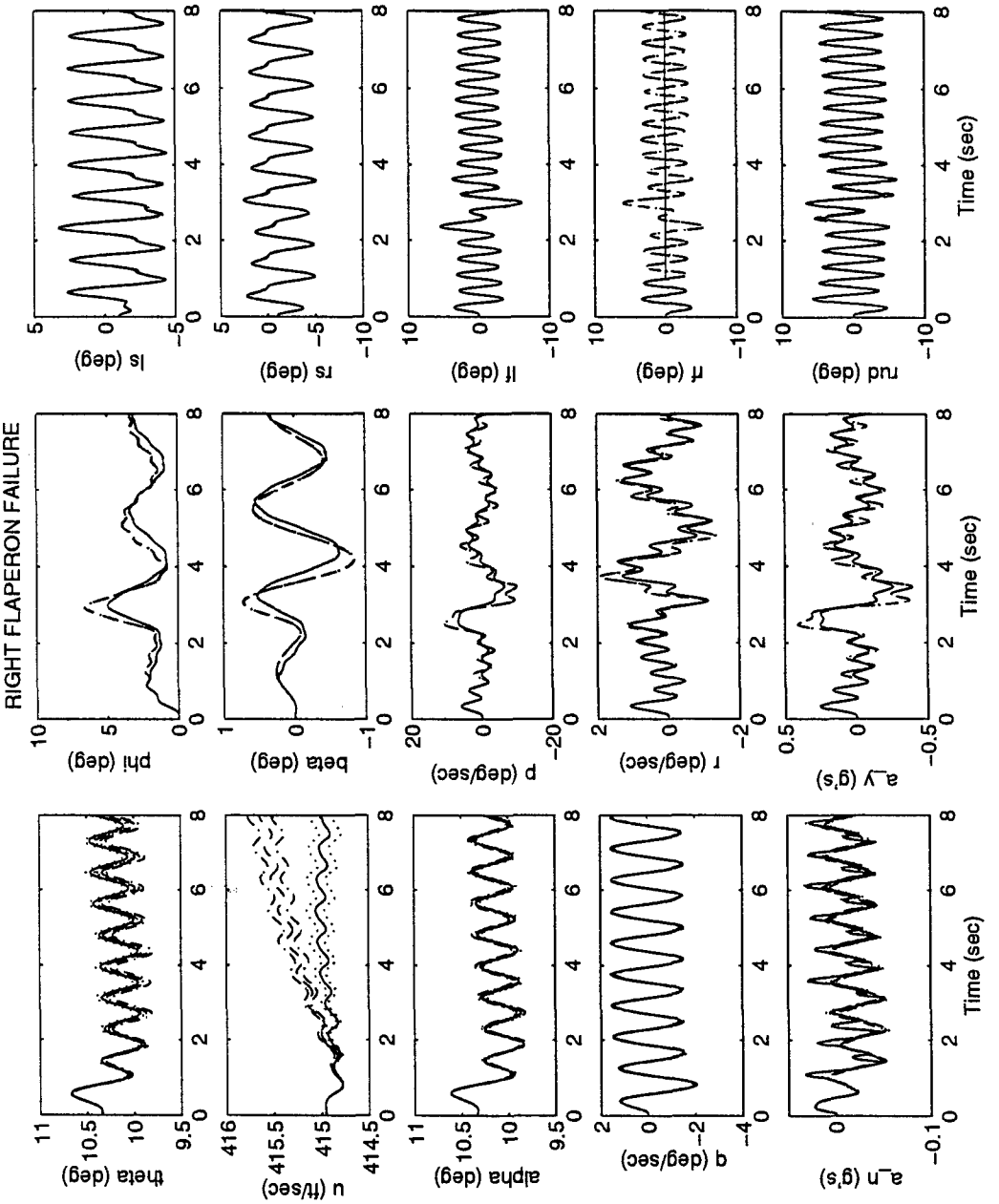

Figure A.14 MMAE-Based - Roll Doublet - No Control Redistribution

Right Flaperon Failure

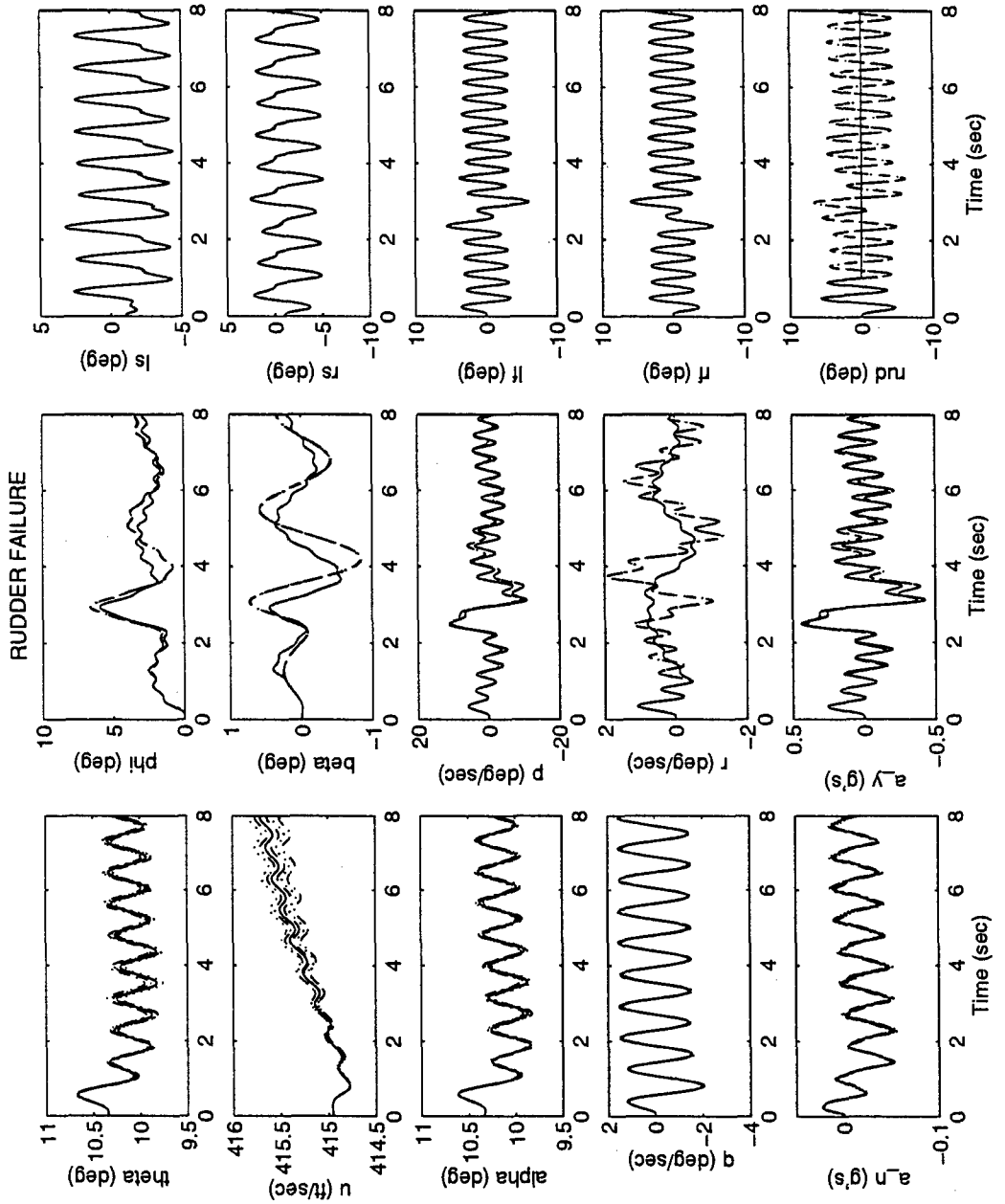

Figure A.15 MMAE-Based - Roll Doublet - No Control Redistribution

Rudder Failure

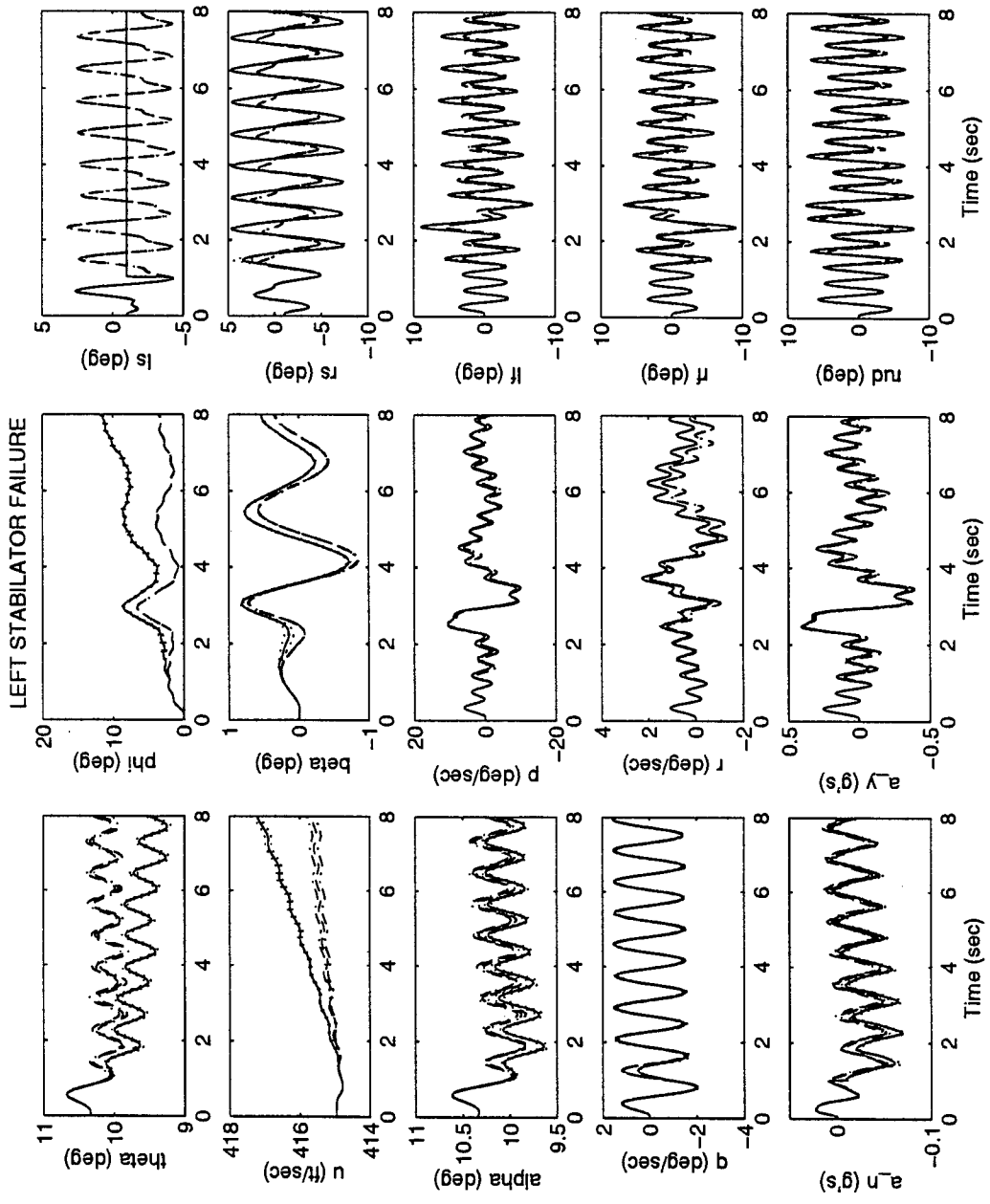

Figure A.16 MMAE-Based - Roll Doublet - With Control Redistribution

Left Stabilator Failure

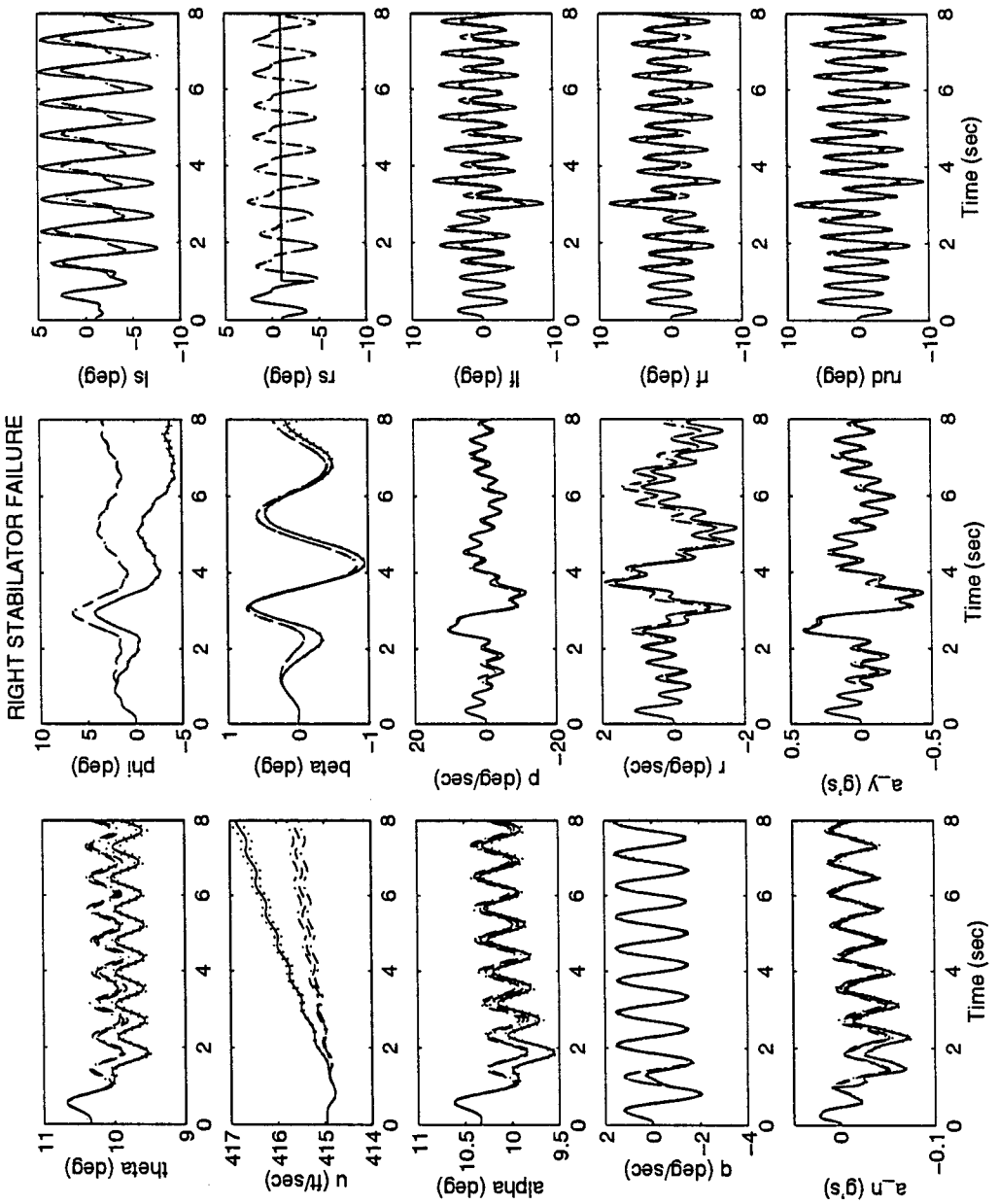

Figure A.17 MMAE-Based - Roll Doublet - With Control Redistribution

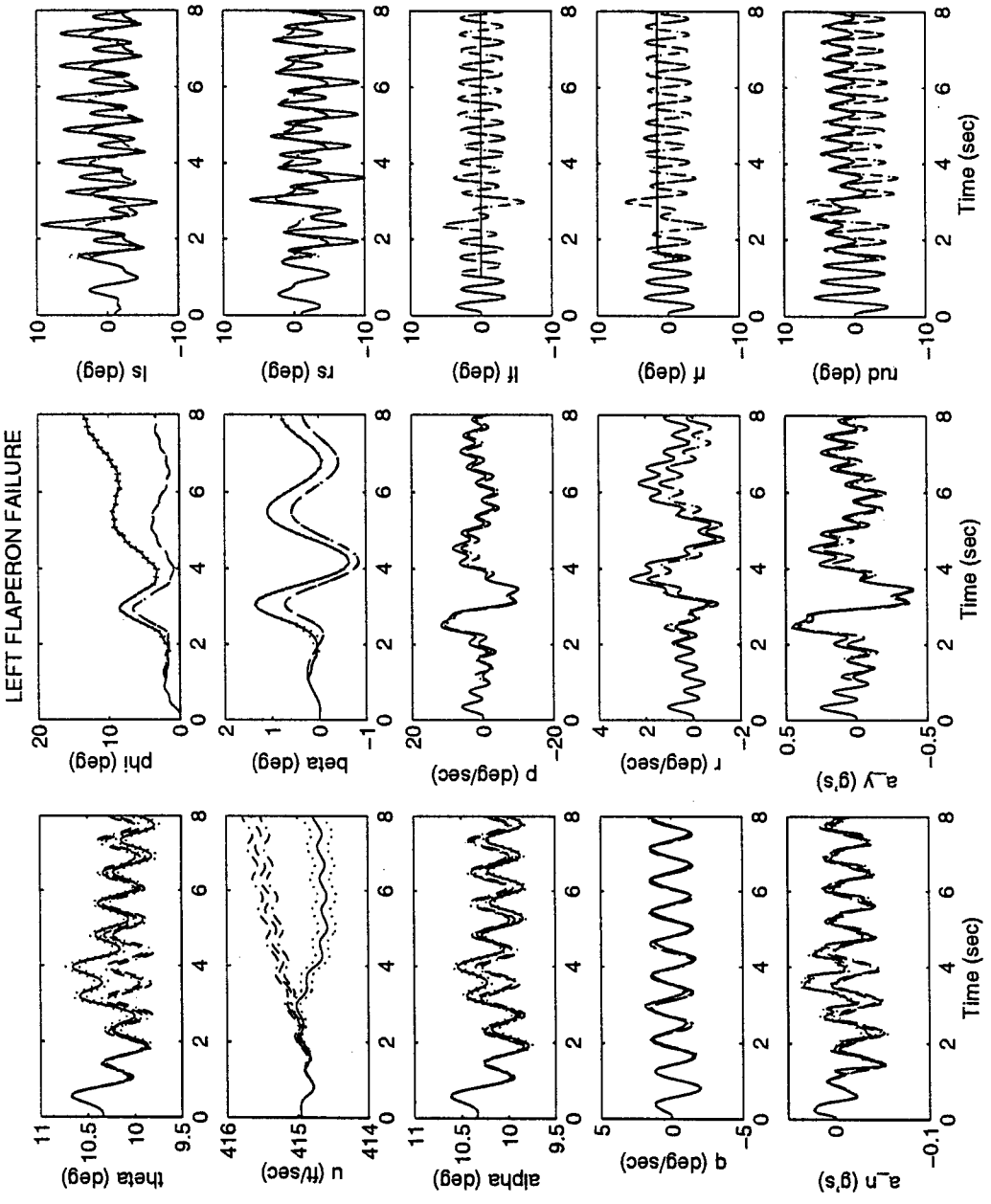

Figure A.18 MMAE-Based - Roll Doublet - With Control Redistribution

Left Flapperon Failure

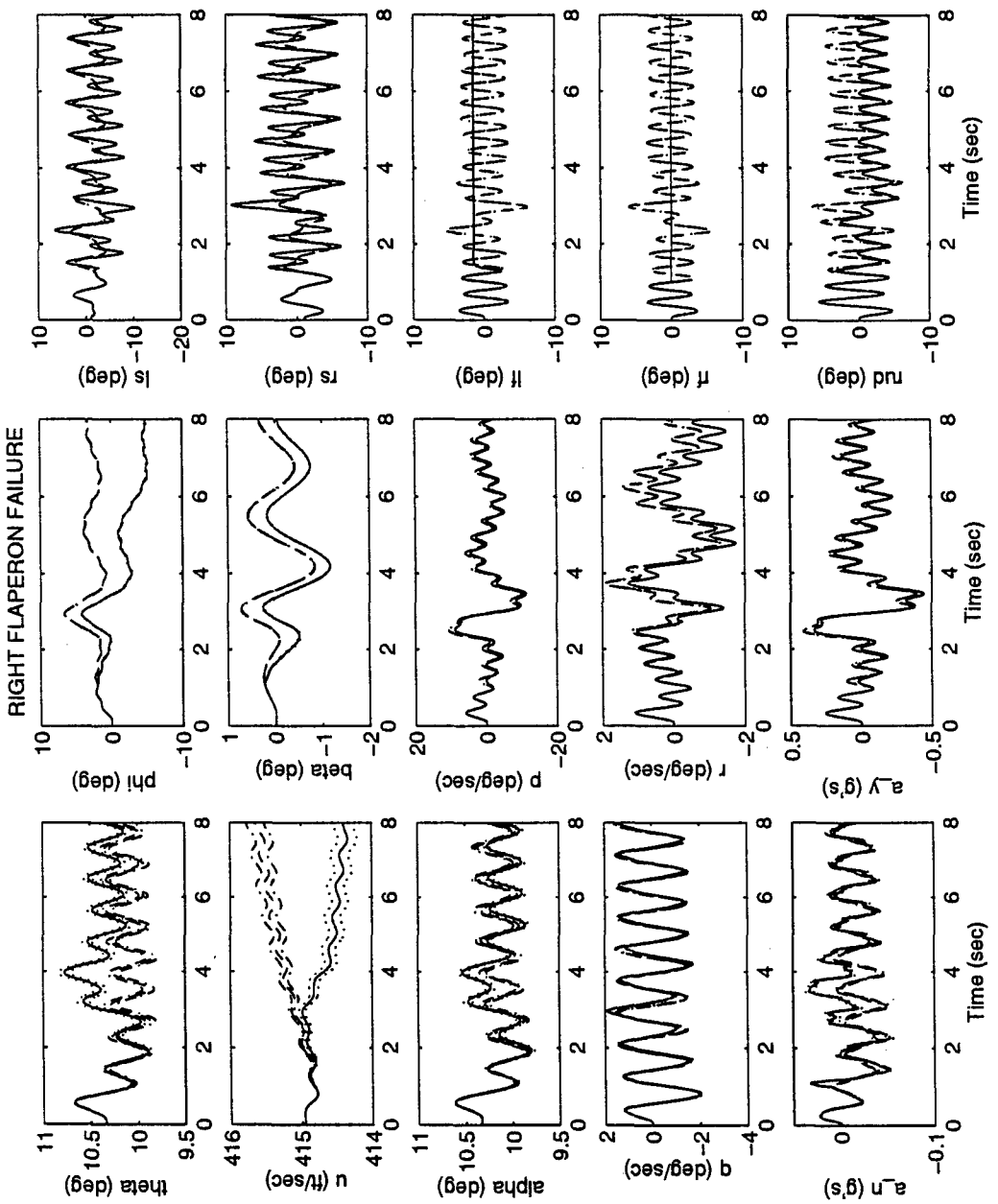

Figure A.19 MMAE-Based - Roll Doublet - With Control Redistribution

Right Flaperon Failure

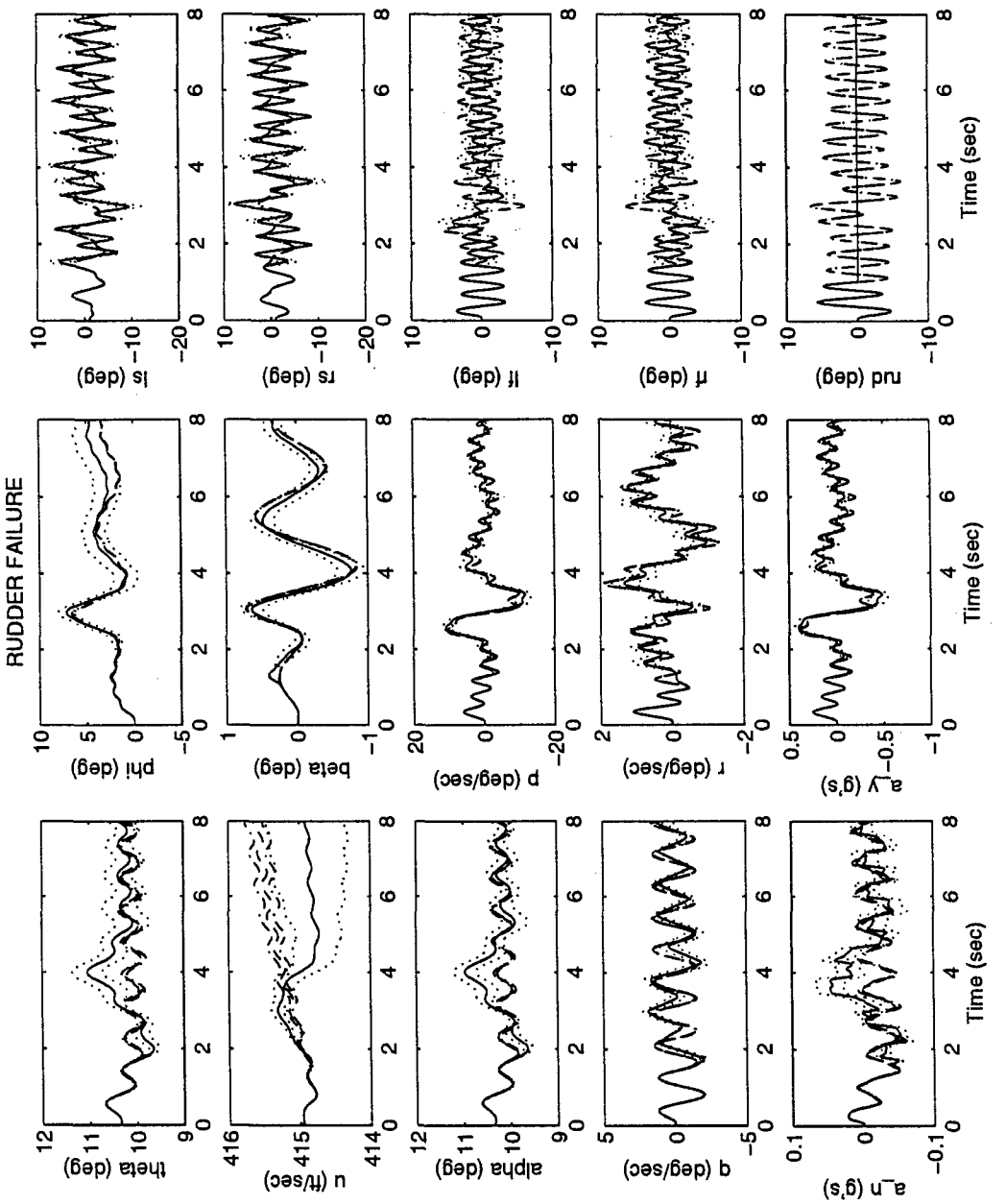

Figure A.20 MMAE-Based - Roll Doublet - With Control Redistribution

Rudder Failure

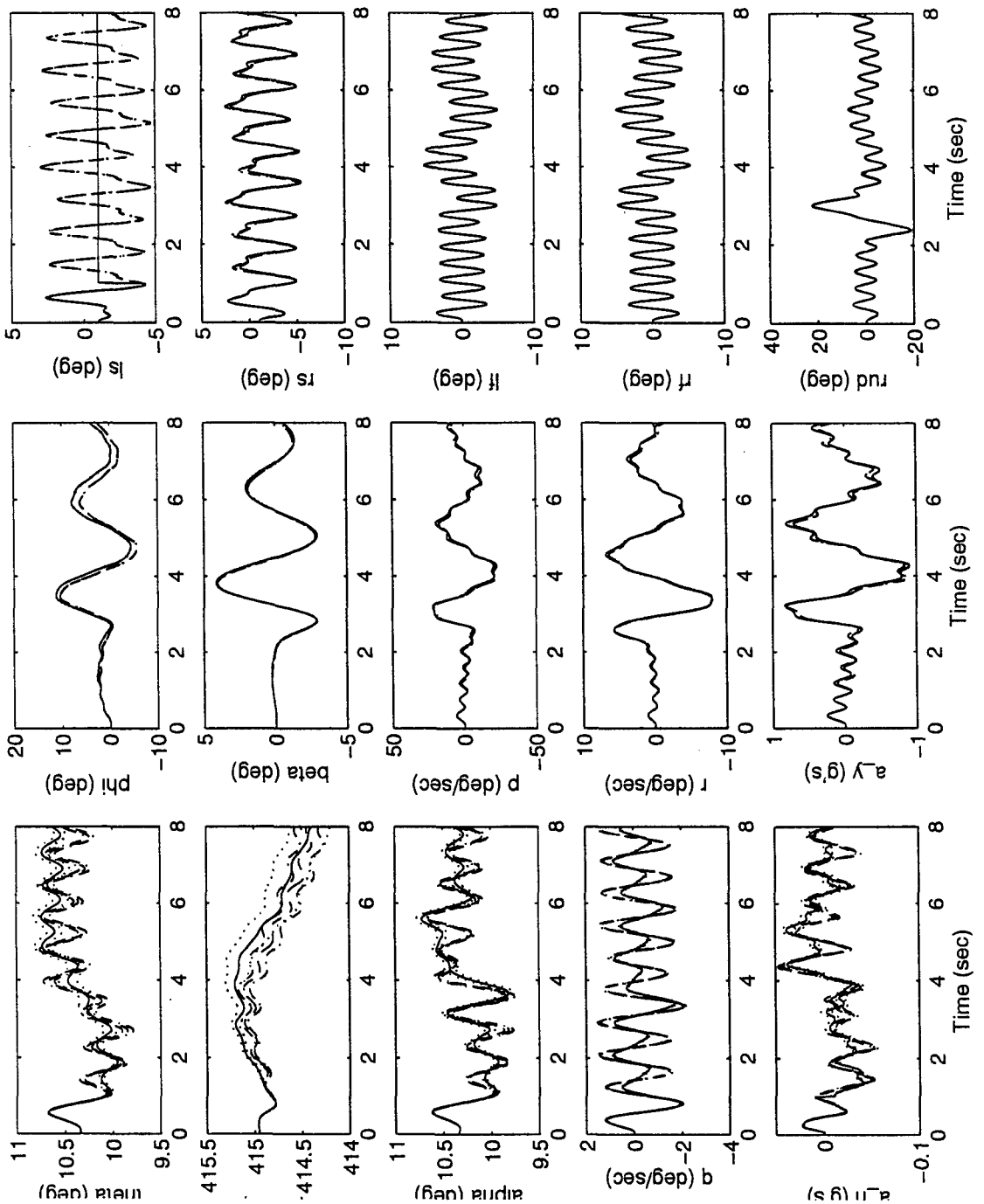

Figure A.21 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Stabilator Failure

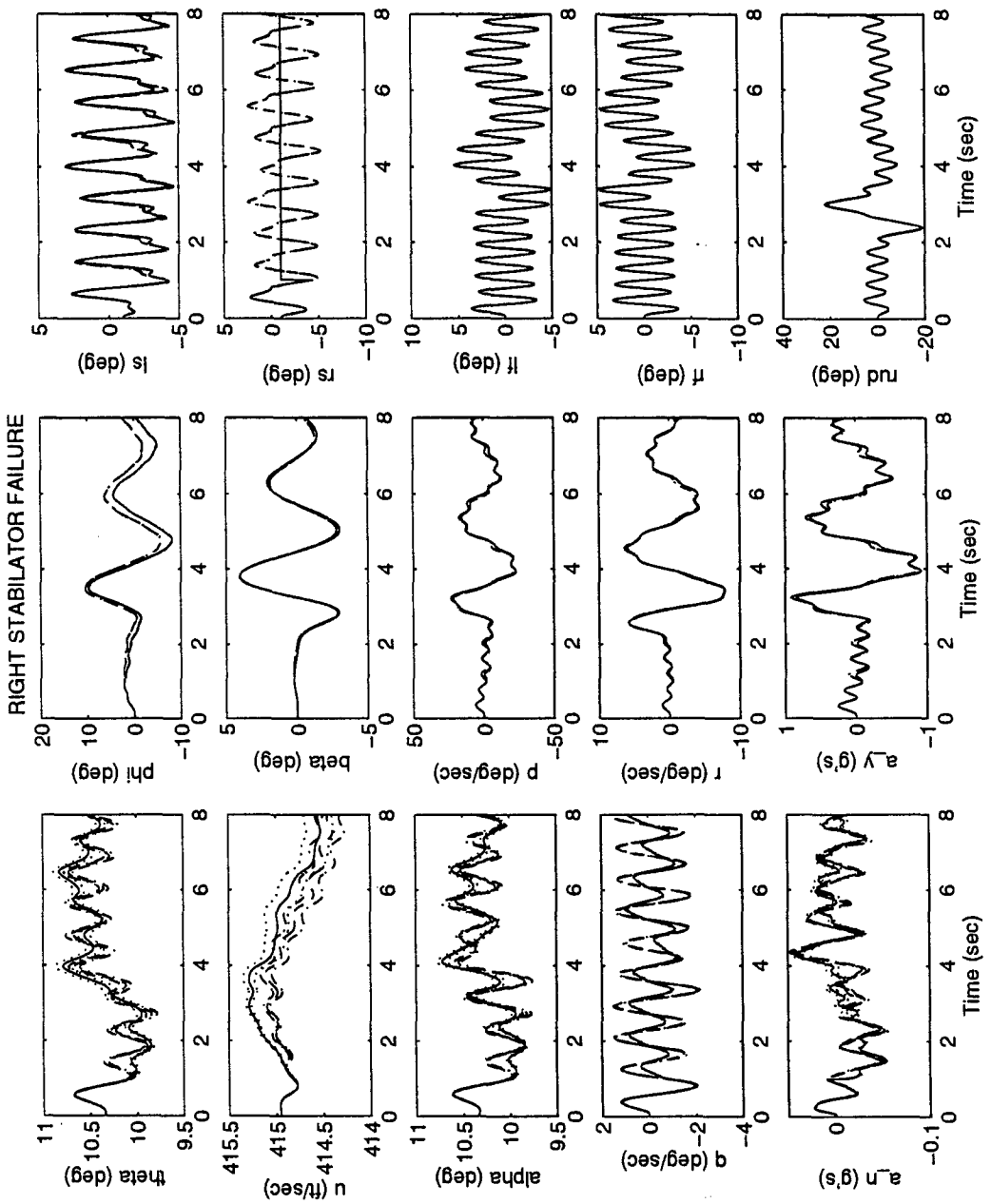

Figure A.22 MMAE-Based - Yaw Doublet - No Control Redistribution

Right Stabilator Failure

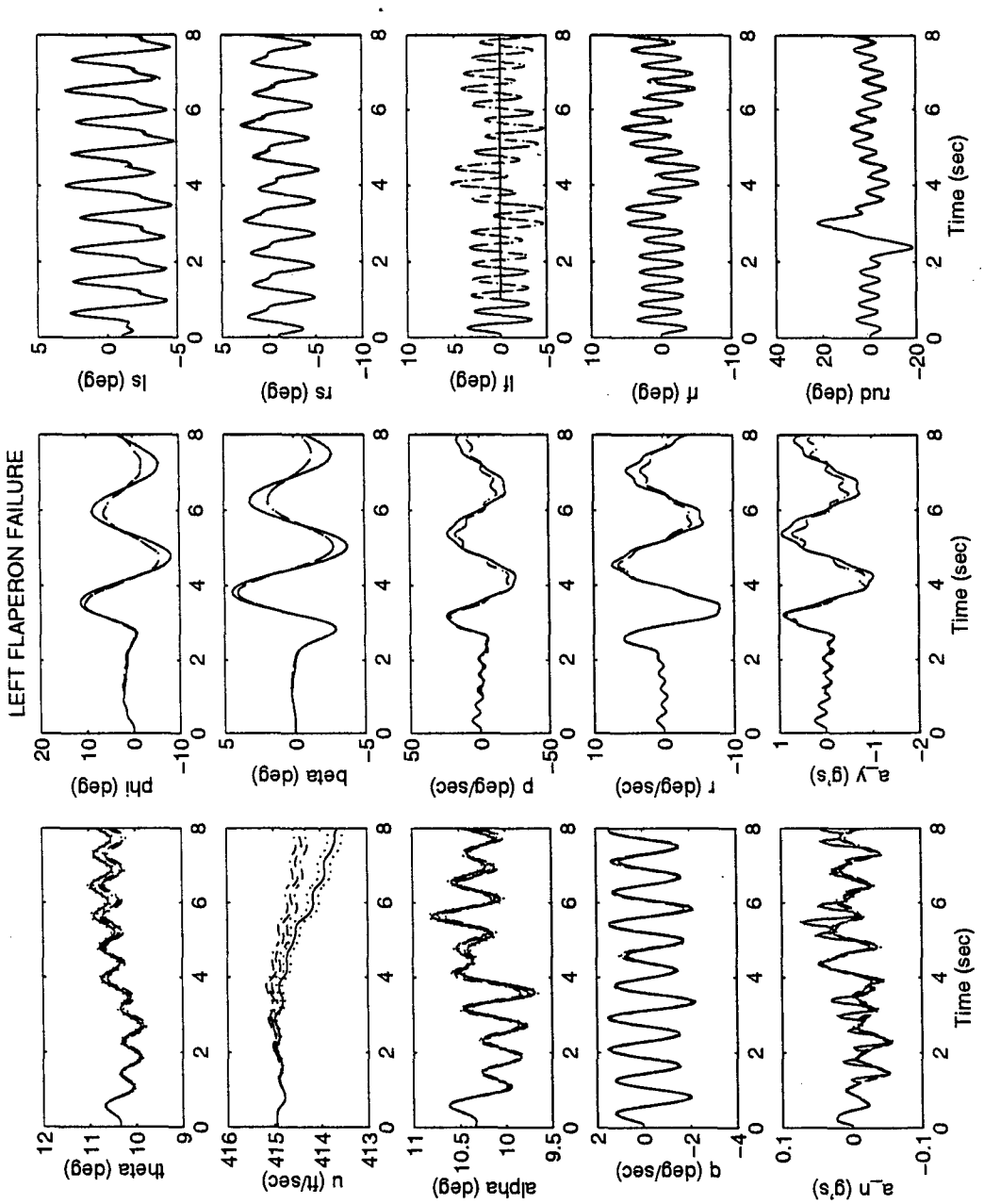

Figure A.23 MMAE-Based - Yaw Doublet - No Control Redistribution

Left flaperon Failure

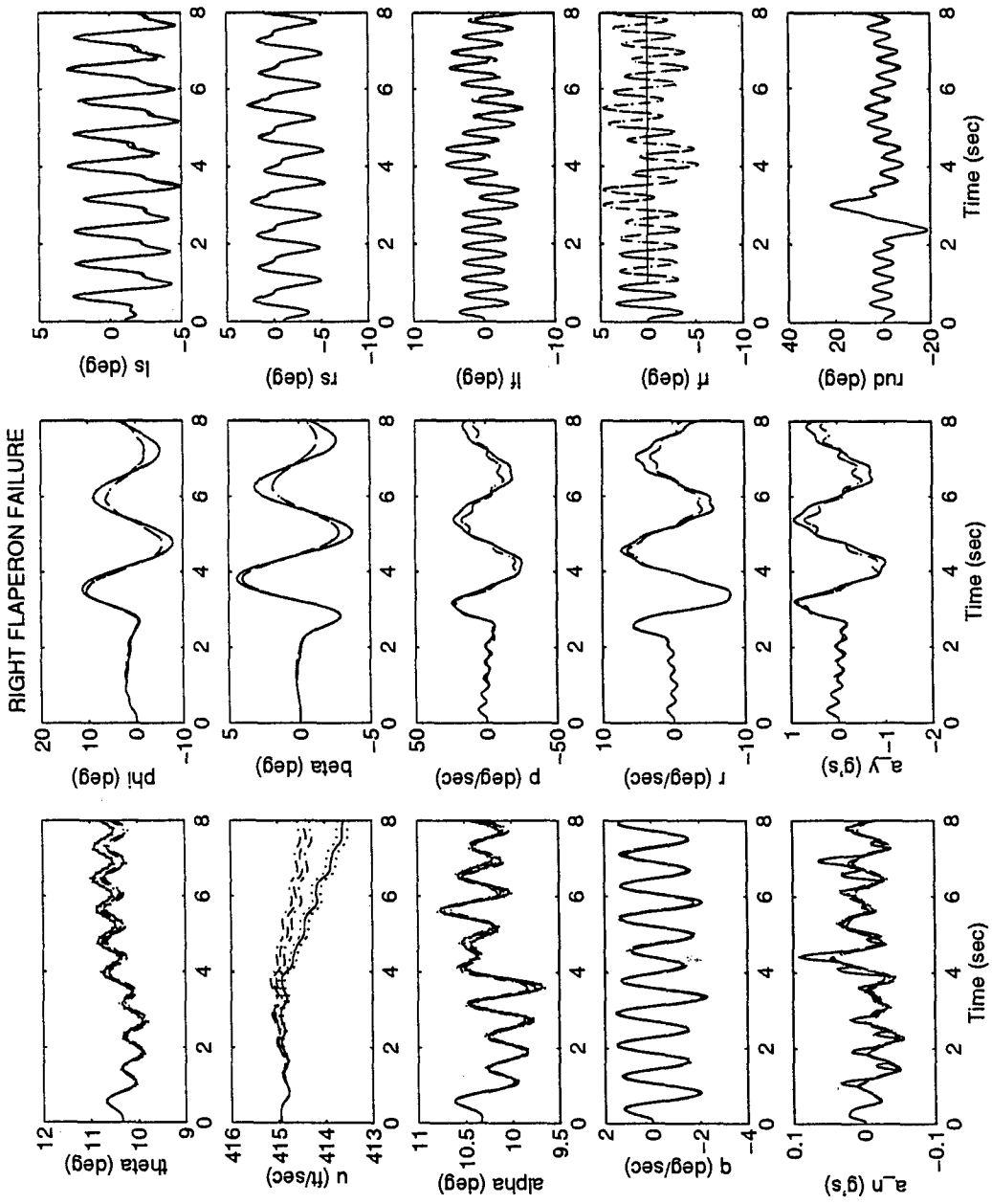

Figure A.24 MMAE-Based - Yaw Doublet - No Control Redistribution

Right Flaperon Failure

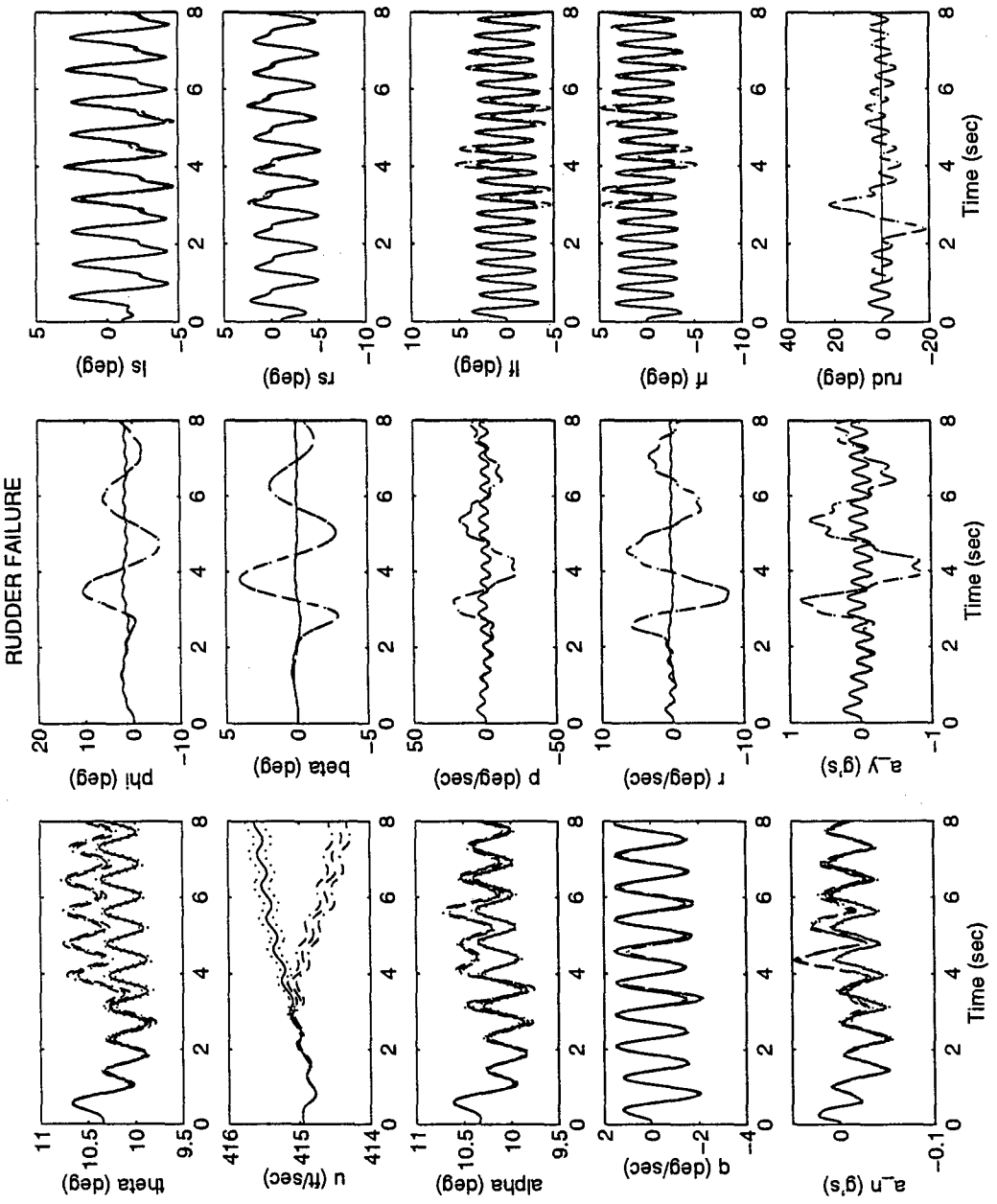

Figure A.25 MMAE-Based - Yaw Doublet - No Control Redistribution

Rudder Failure

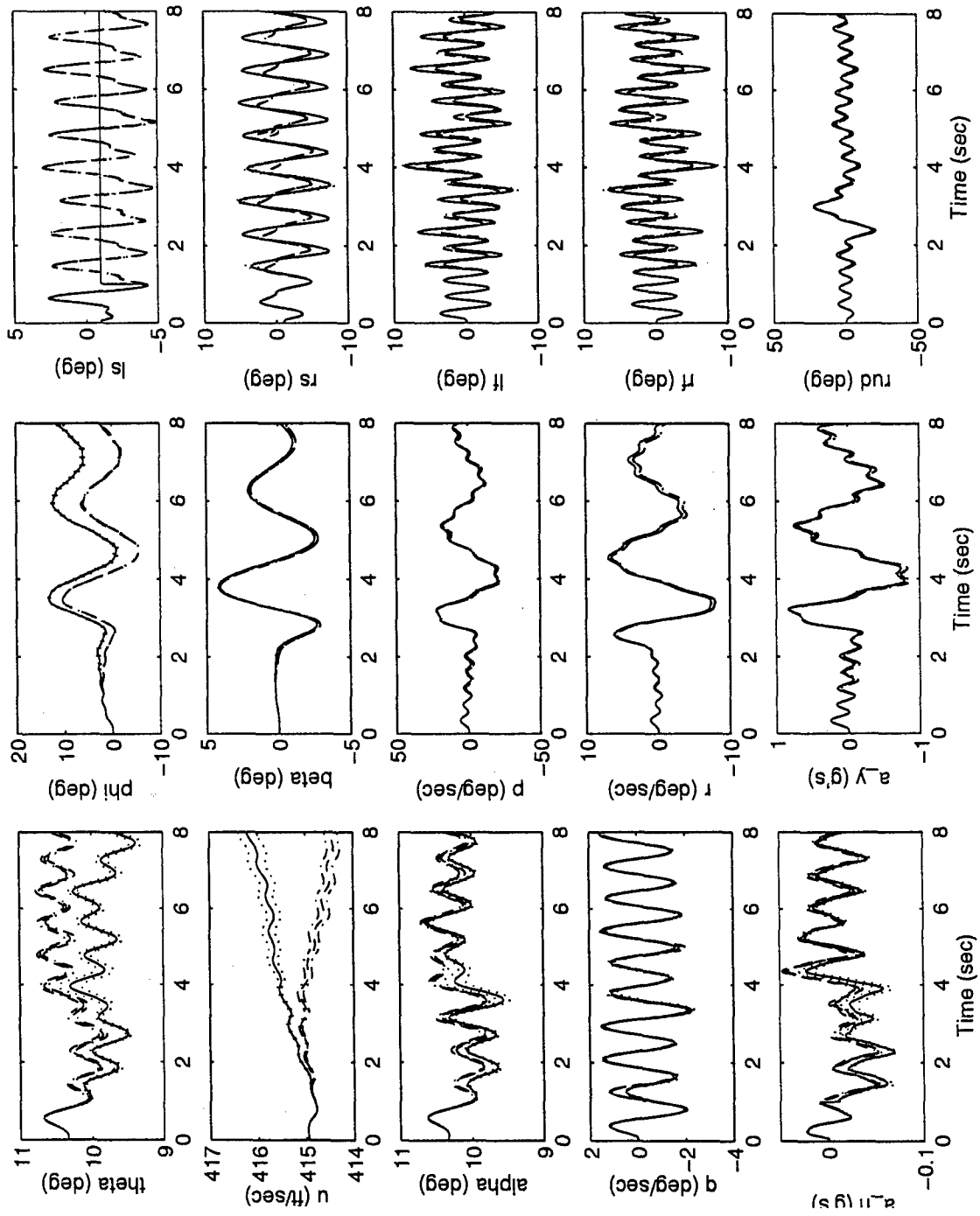

Figure A-26 MMAE-Based - Yaw Doublet - With Control Redistribution

Left Stabilator Failure

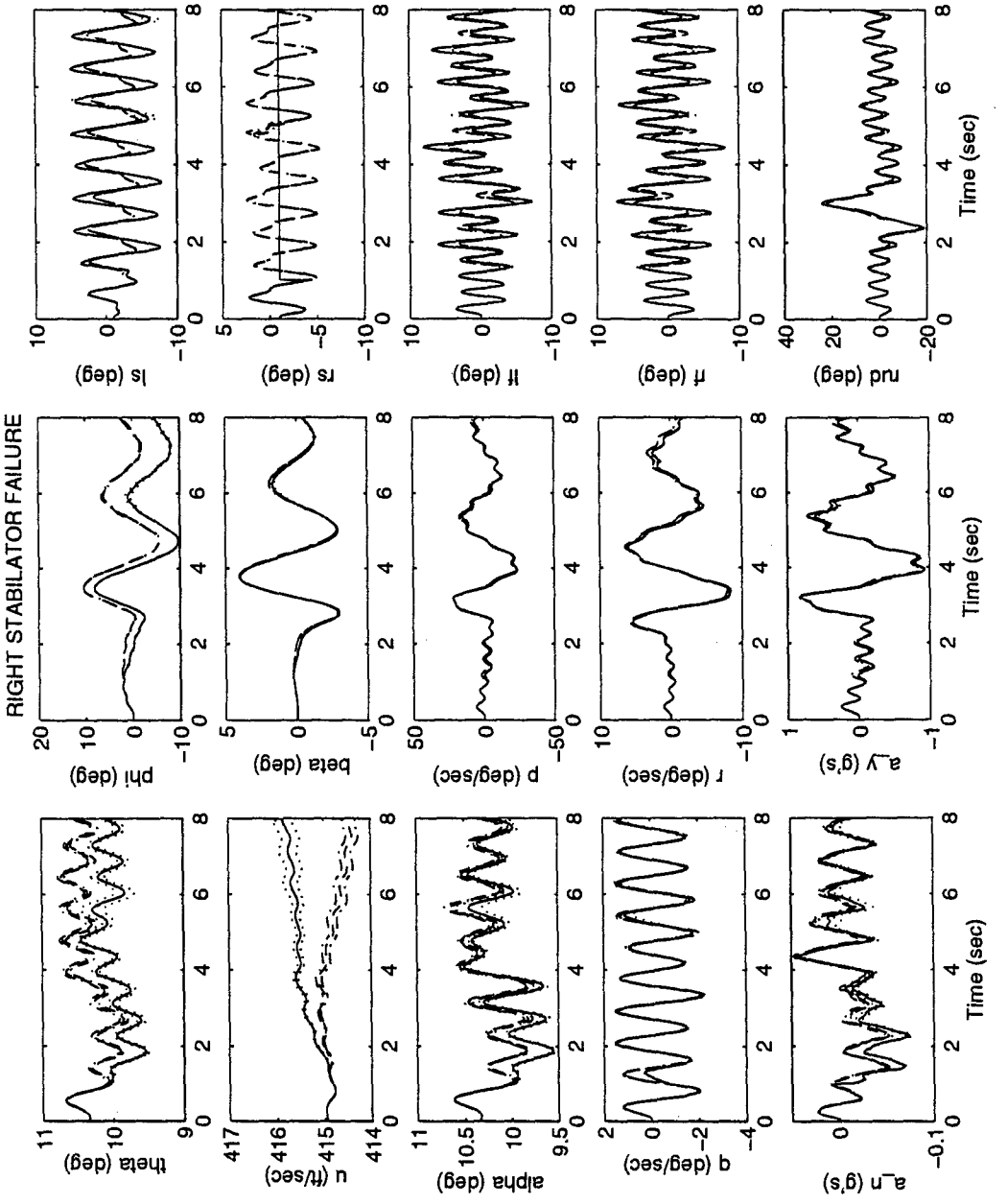

Figure A-27 MMAE-Based - Yaw Doublet - With Control Redistribution

Right Stabilator Failure

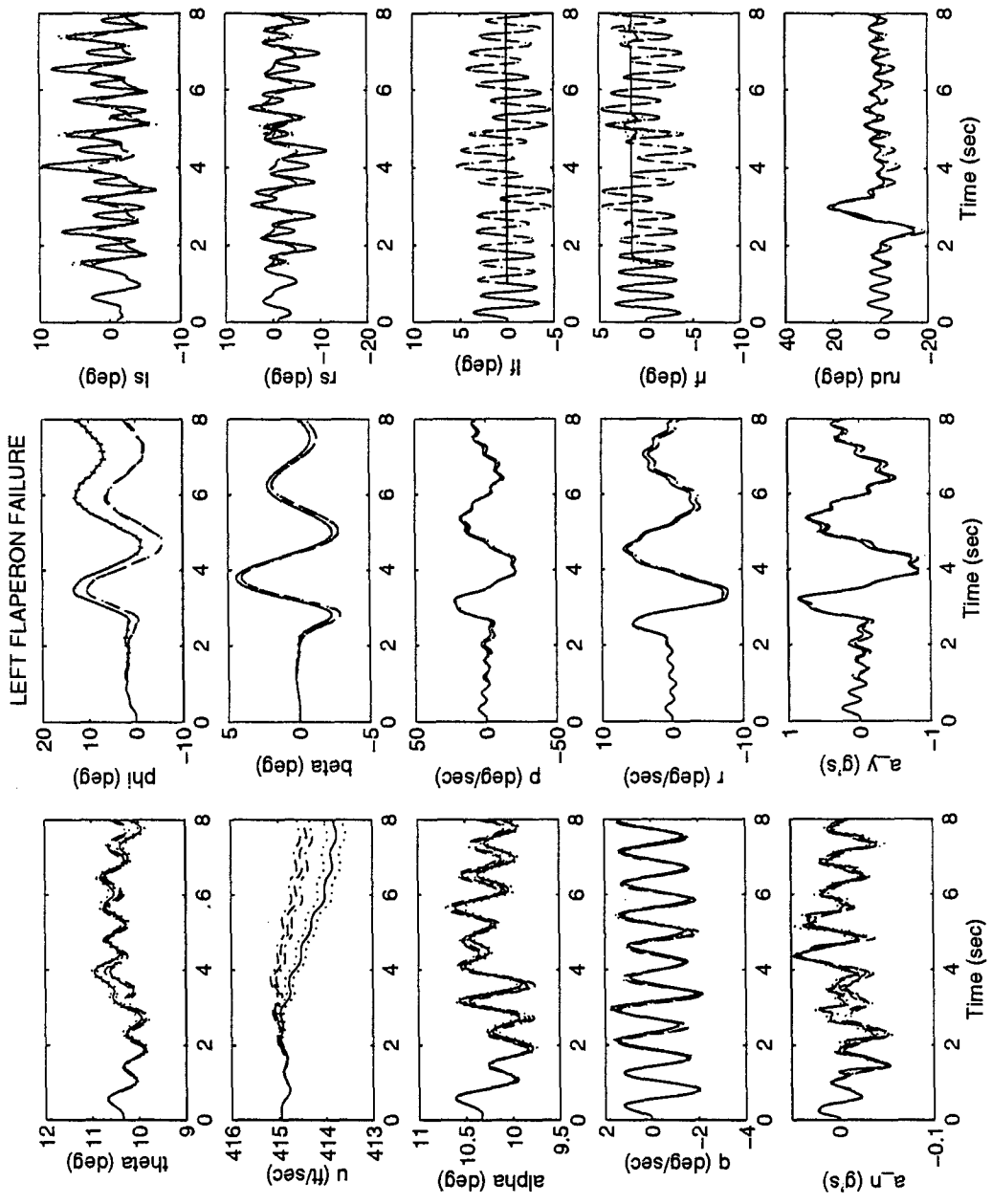

Figure A-28 MMAE-Based - Yaw Doublet - With Control Redistribution

Left Flaperon Failure

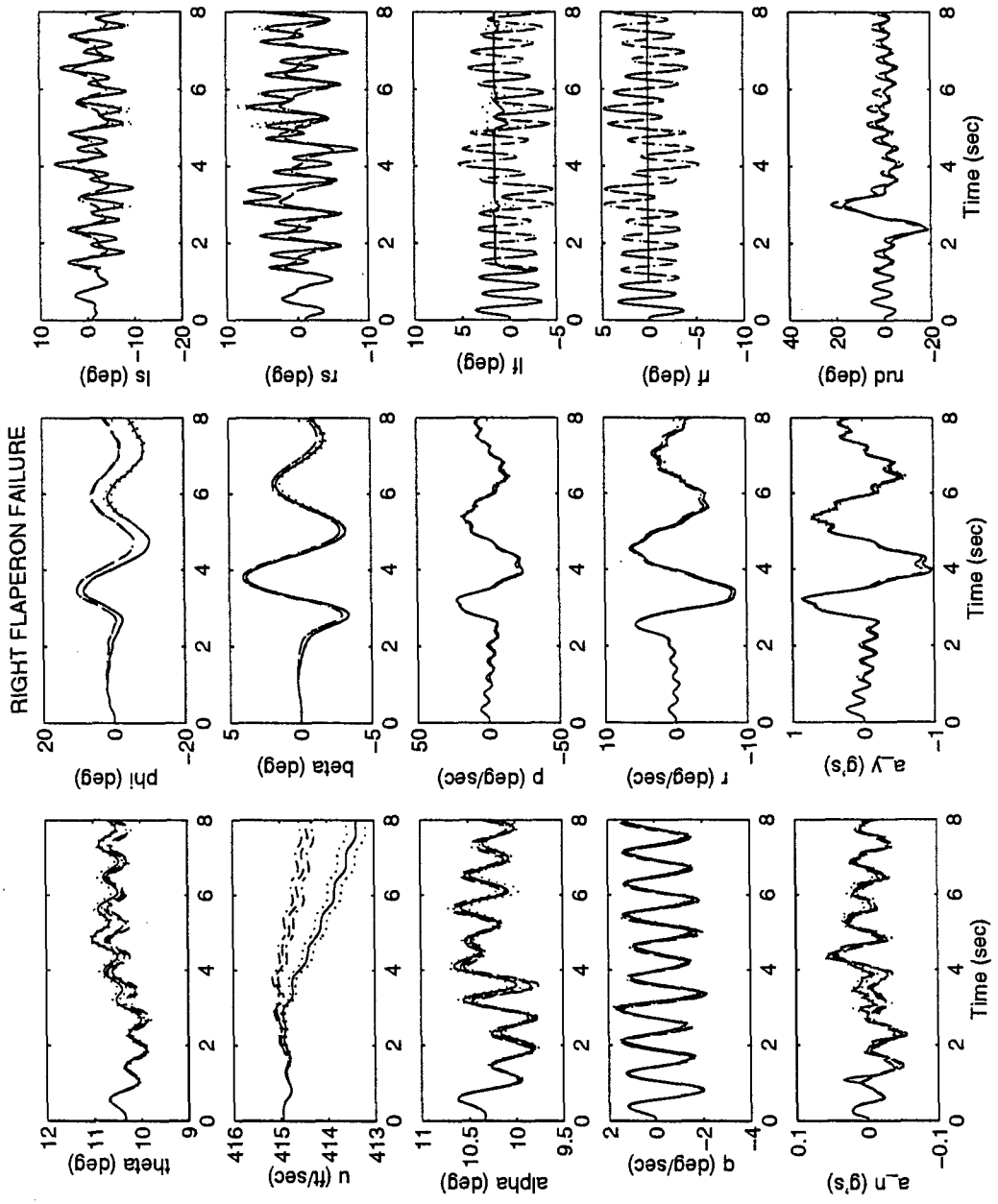

Figure A-29 MMAE-Based - Yaw Doublet - With Control Redistribution

Right Flaperon Failure

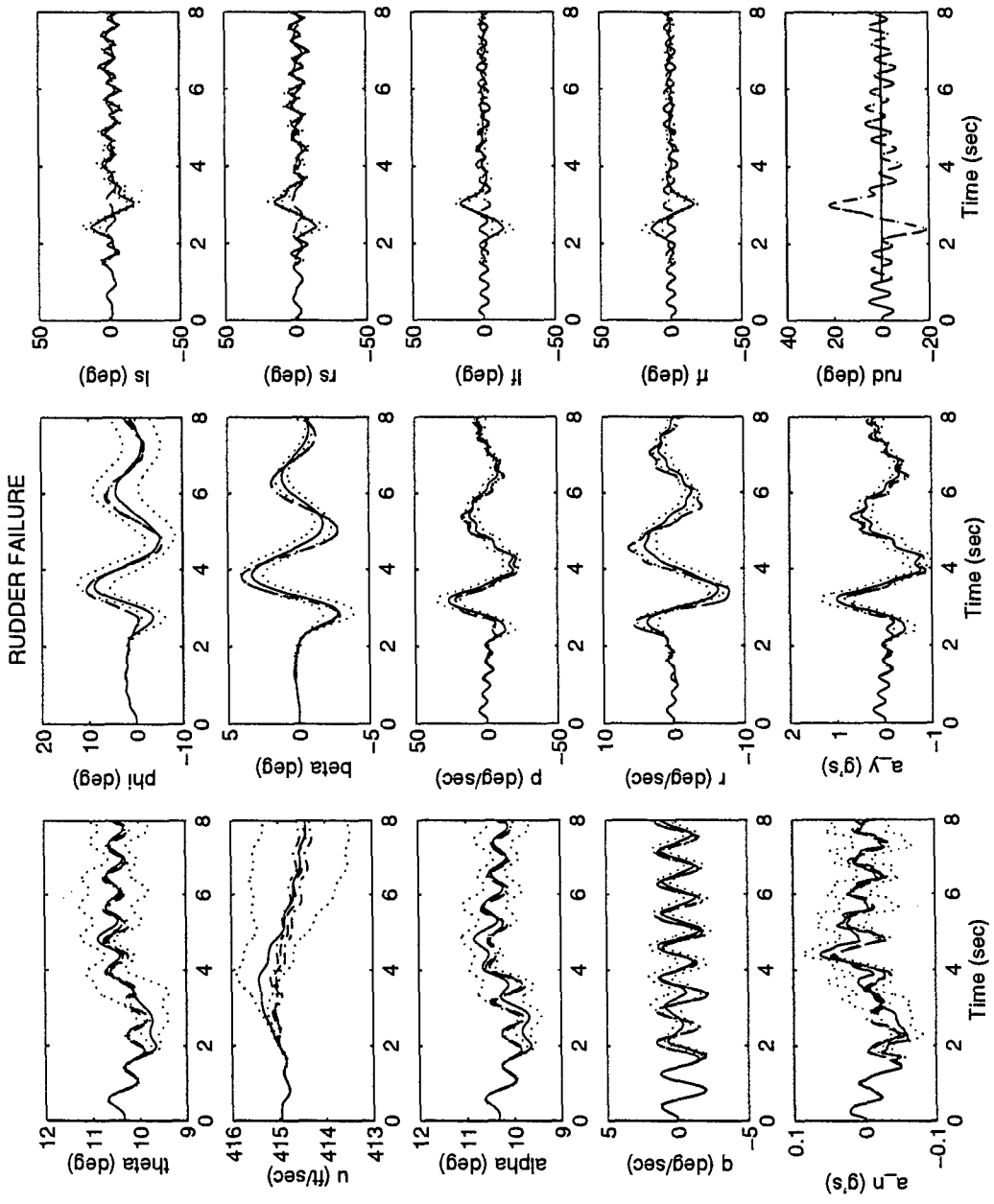

Figure A-30 MMAE-Based - Yaw Doublet - With Control Redistribution

Rudder Failure

Appendix B

Probability Plots for MMAE-based Block 40 with No Control Redistribution

This appendix contains the probability plots of a simulation which was run without Control Redistribution. The purpose of this simulation was to compare Stepaniak's [46] dither scheme with that used by Eide [8], and to provide a basis with which to compare the simulations performed with control redistribution. These two analyses are both provided in Section 4.3.

Each of the following probability summary plots is the result of 11 separate simulations. Each simulation is shown with a separate subplot, and each subplot shows the mean of 10 Monte Carlo runs (solid line) and the ± one sigma value (dotted line). The first subplot shown represents the probability of the Kalman filter which hypothesizes a fully functional (FF) aircraft, and therefore should drop from one to zero soon after the injection of the first failure at one second. The remaining 11 represent simulations in which the first failure (injected at one second) is combined with a particular secondary failure (injected at two seconds). The exception to this is the location which, by this convention, would be showing the double failure of the same component (an impossible scenario). In this location, the probability plot for the Kalman filter which represents the first failure, with no secondary failure, is presented. The simulation for this subplot is usually chosen as the simulation shown in the next consecutive subplot, with the exception of the first failure being the rudder ("rud" = last actuator), which utilizes the simulation for the lateral acceleration sensor (a_y) as the second failure, and the first failure being the lateral acceleration sensor (a_y = last sensor), which utilizes the simulation for the pitch sensor (q) as the second failure.

Note that because of relatively long titles, Figures B-8 and B-12 are scaled slightly smaller than the others. The plots in these figures are no different from the others, they've just been proportionally scaled so that their title would fit on the page.

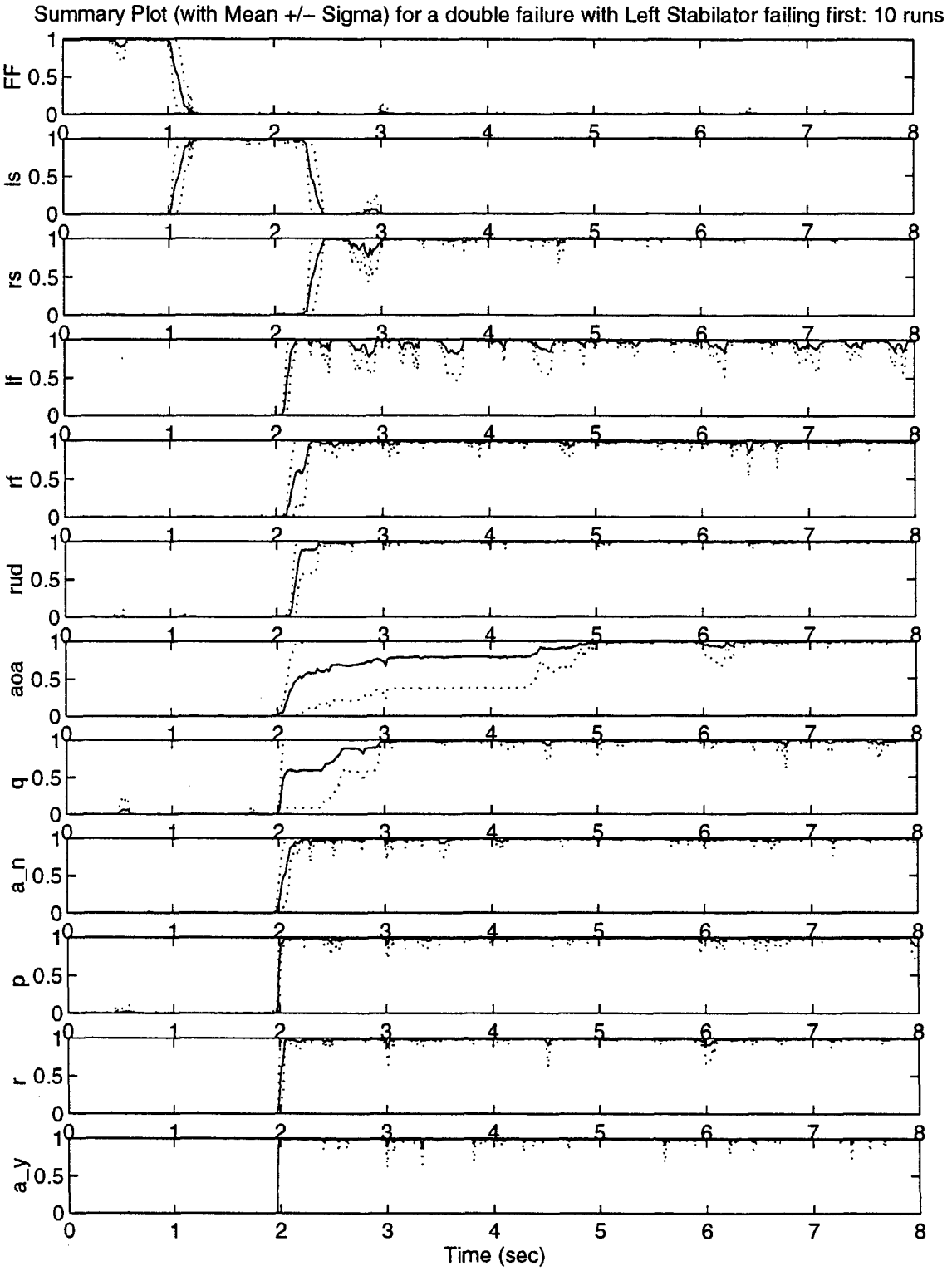

Figure B-1 Double Failure Summary Plot - No Control Redistribution

Left Stabilator Failing First

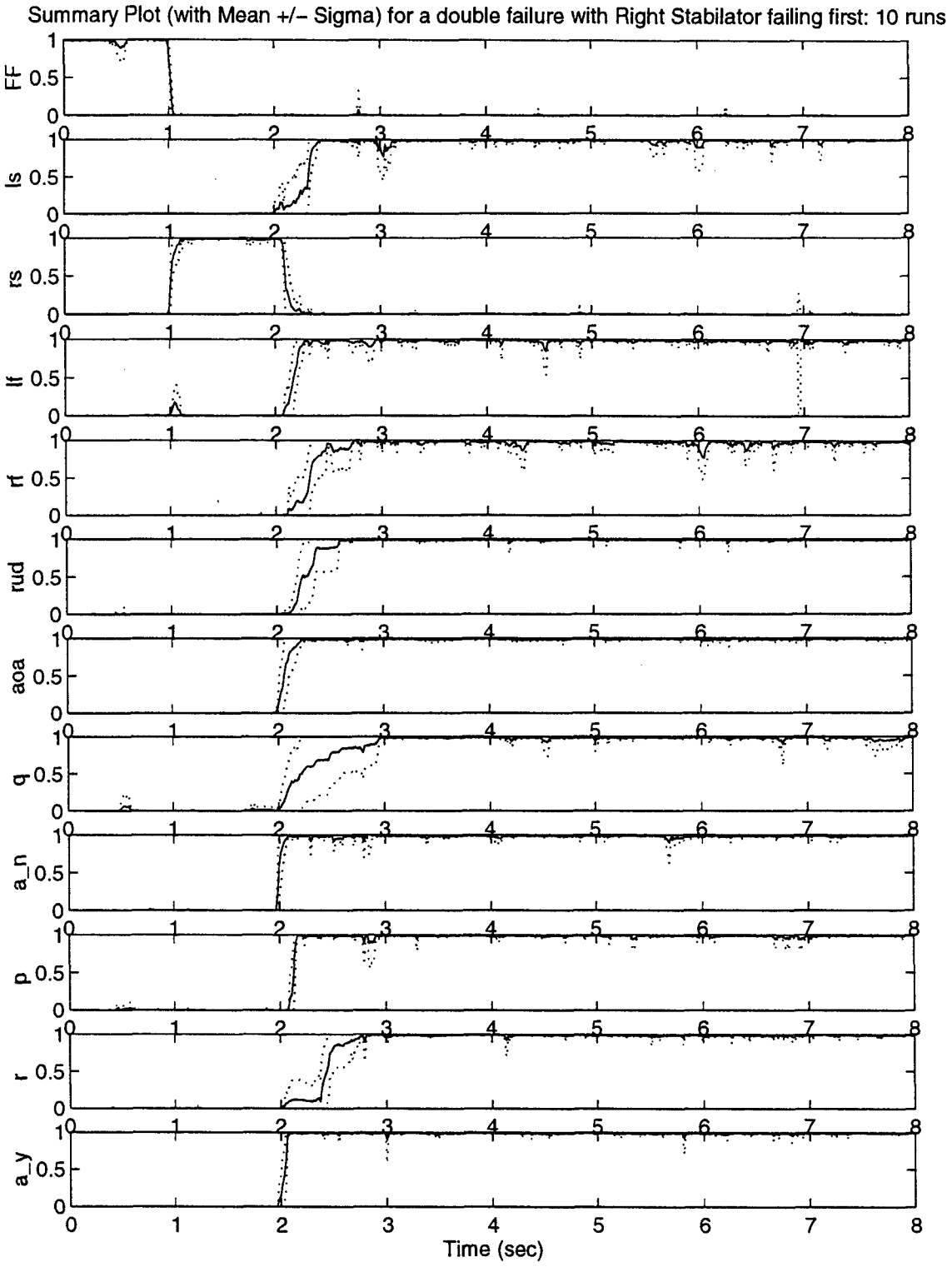

Figure B-2 Double Failure Summary Plot - No Control Redistribution

Right Stabilator Failing First

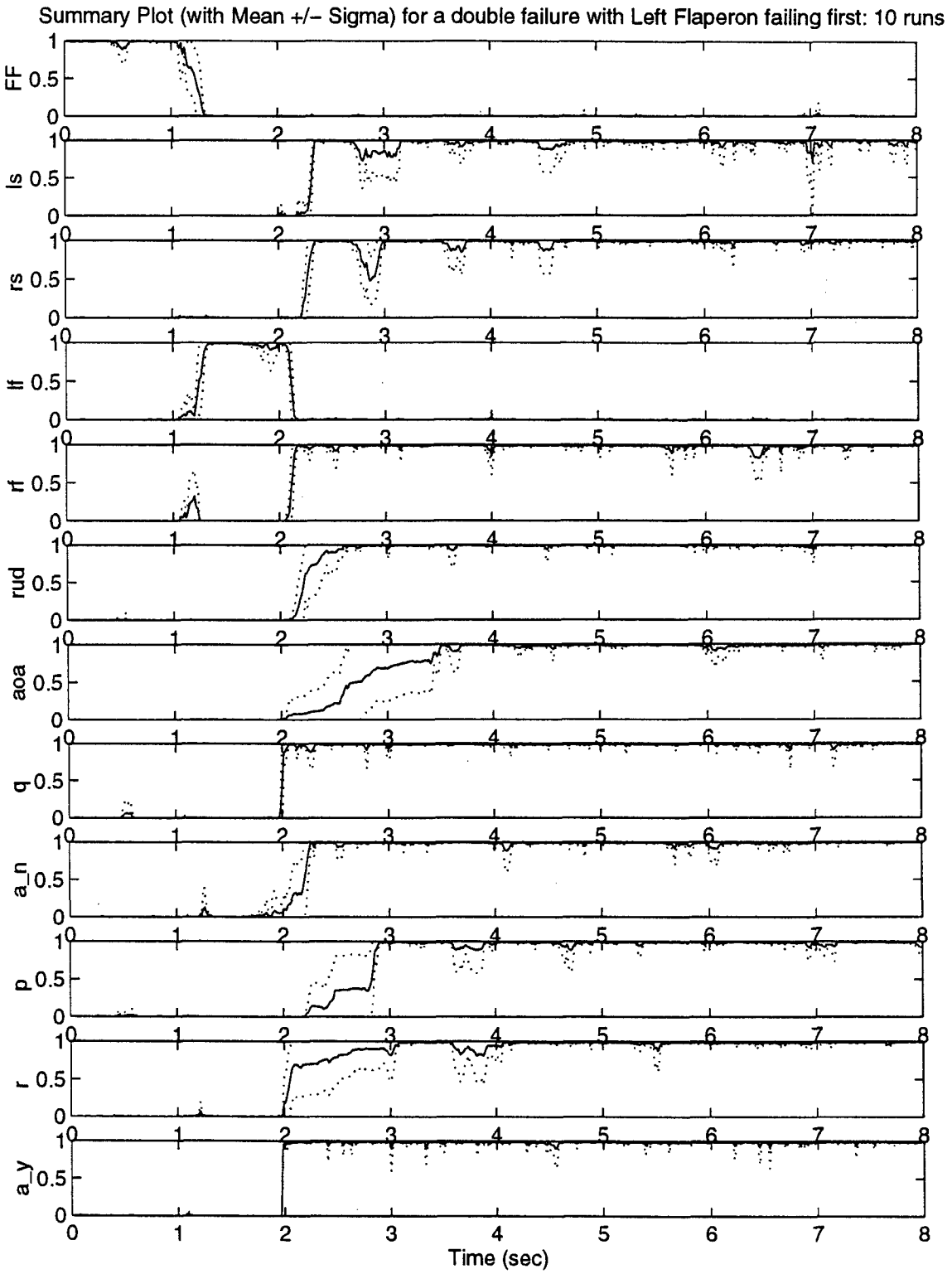

Figure B-3 Double Failure Summary Plot - No Control Redistribution

Left Flaperon Failing First

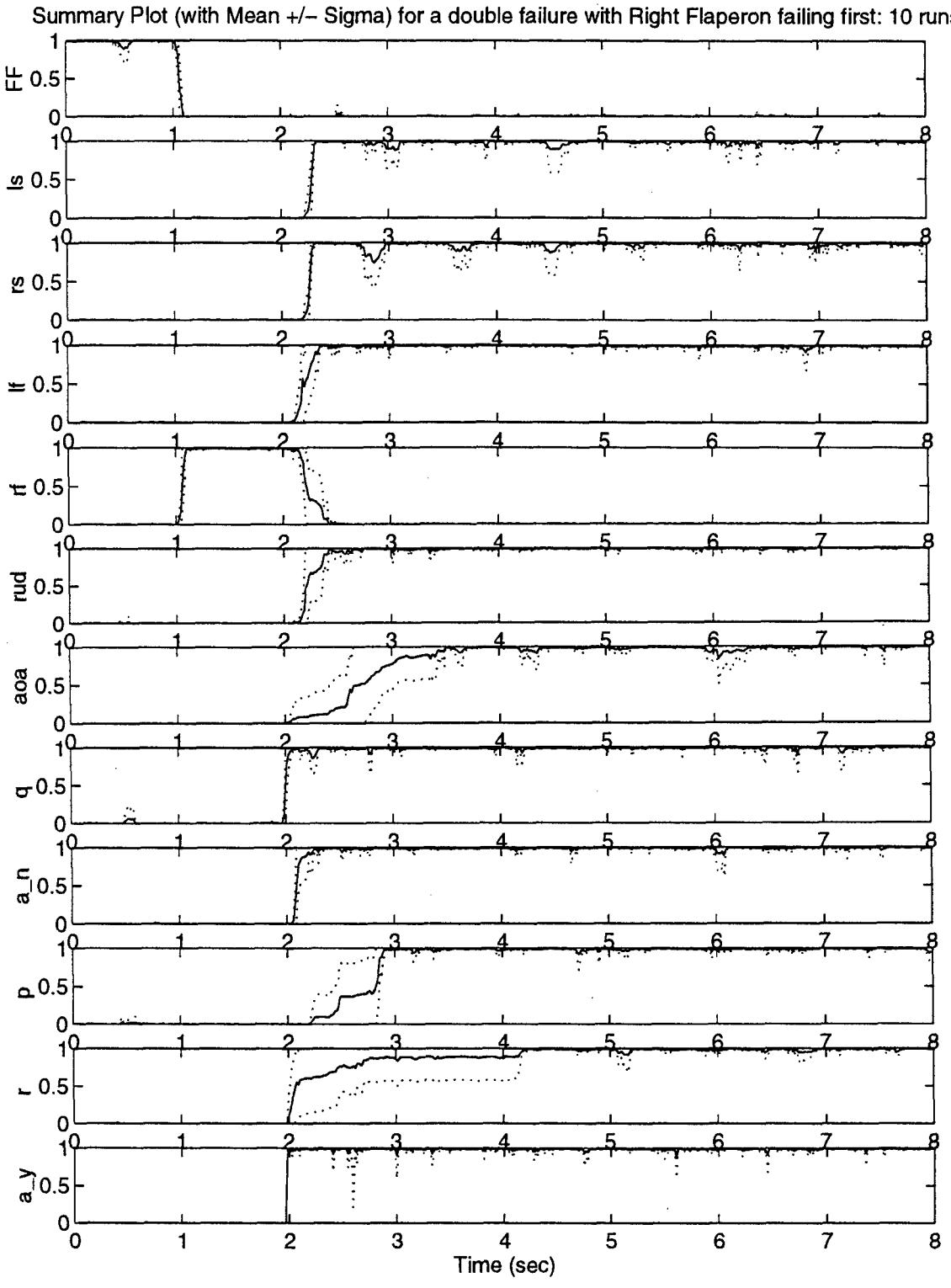

Figure B-4 Double Failure Summary Plot - No Control Redistribution

Right Flaperon Failing First

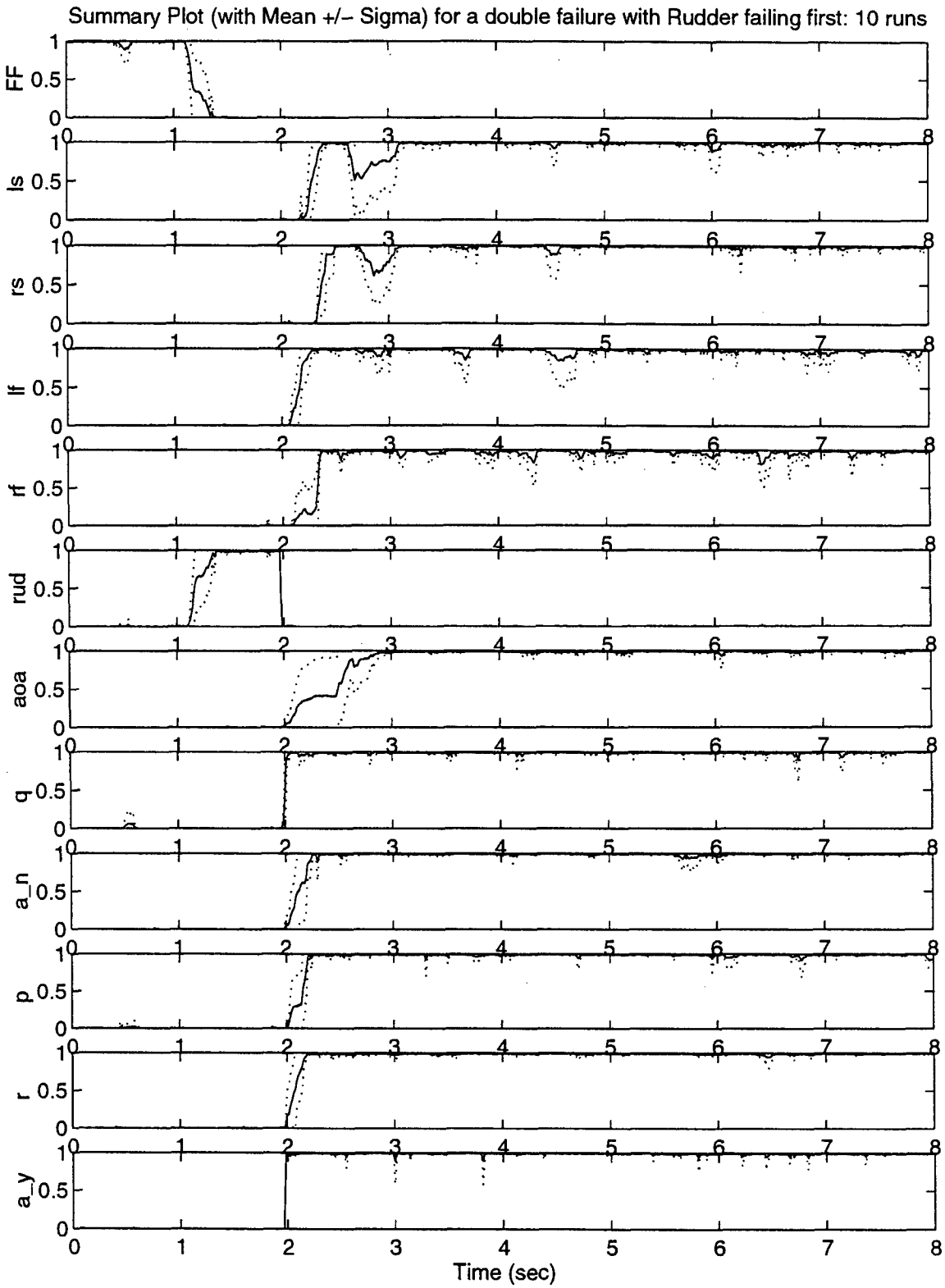

Figure B-5 Double Failure Summary Plot - No Control Redistribution

Rudder Failing First

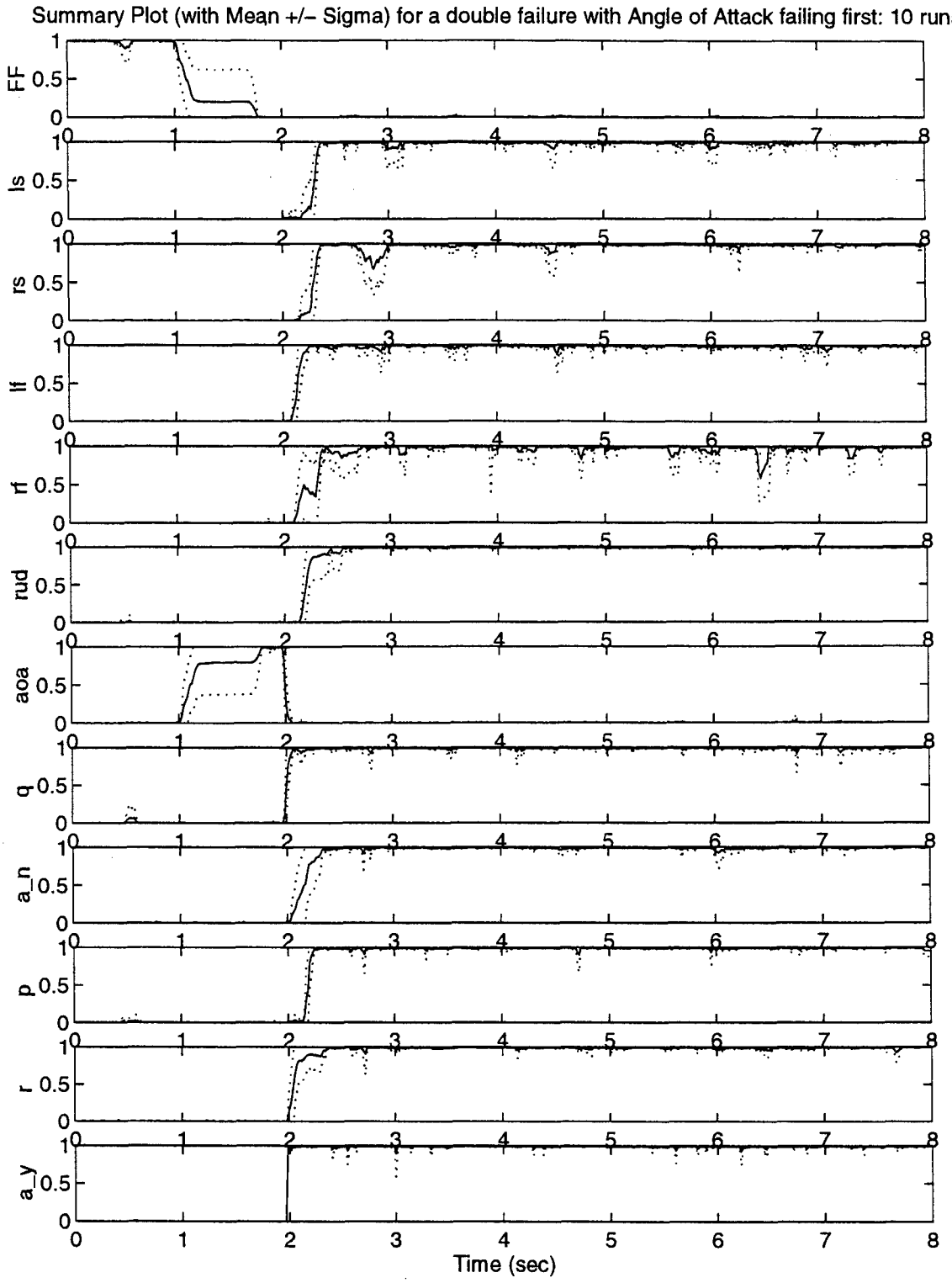

Figure B-6 Double Failure Summary Plot - No Control Redistribution

Angle of Attack Sensor Failing First

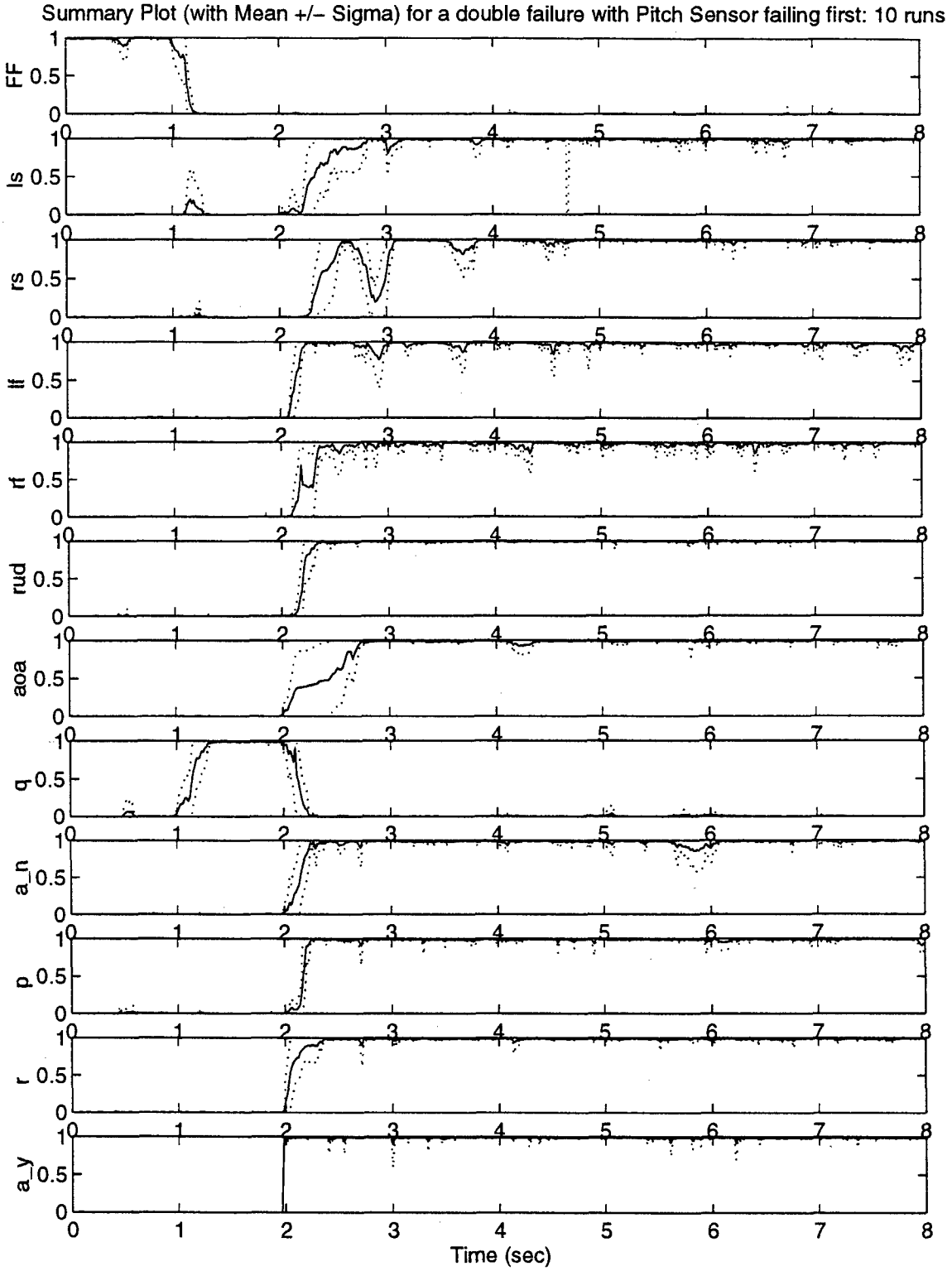

Figure B-7 Double Failure Summary Plot - No Control Redistribution

Pitch Rate Sensor Failing First

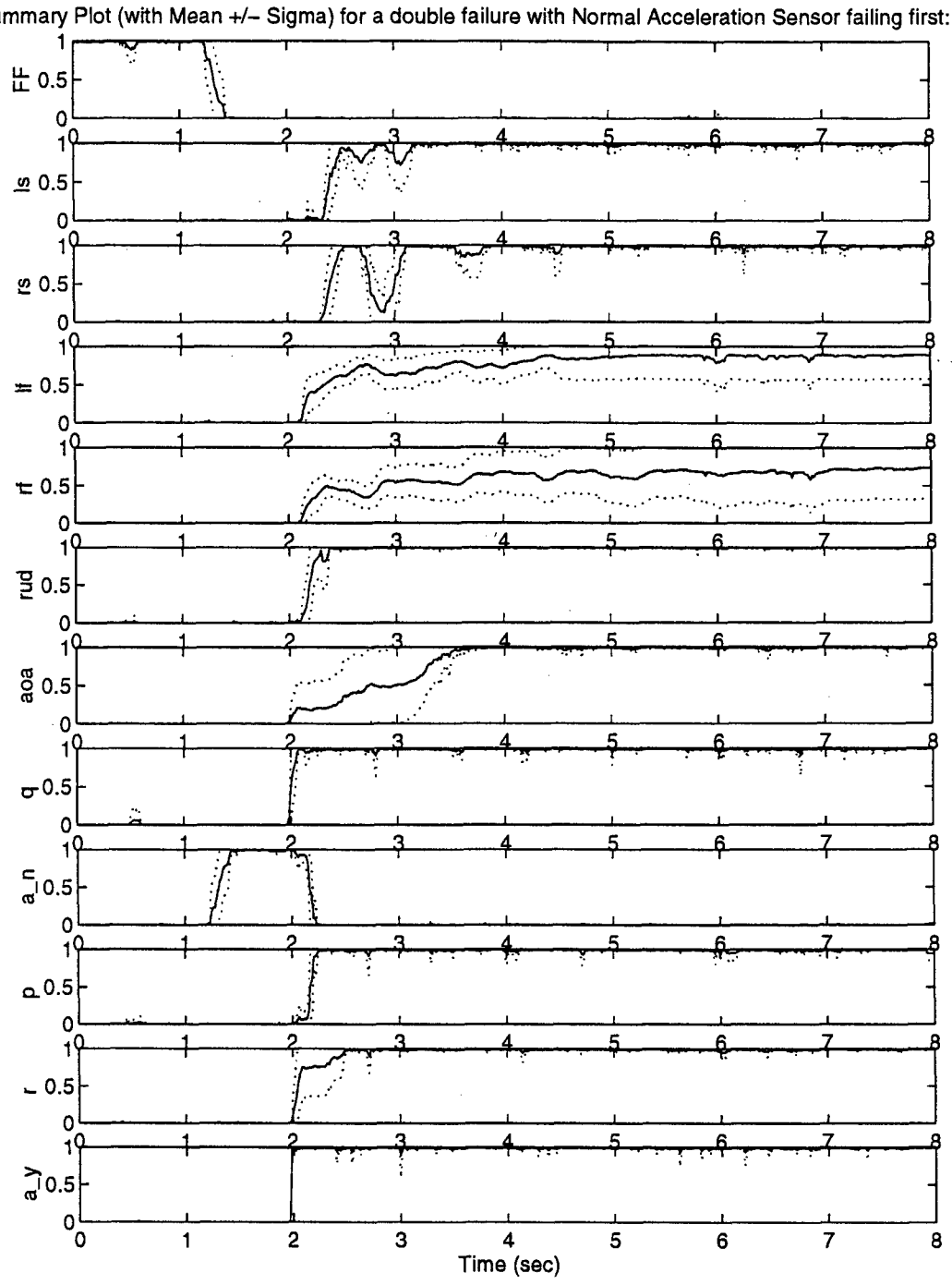

Figure B-8 Double Failure Summary Plot - No Control Redistribution

Normal Acceleration Sensor Failing First

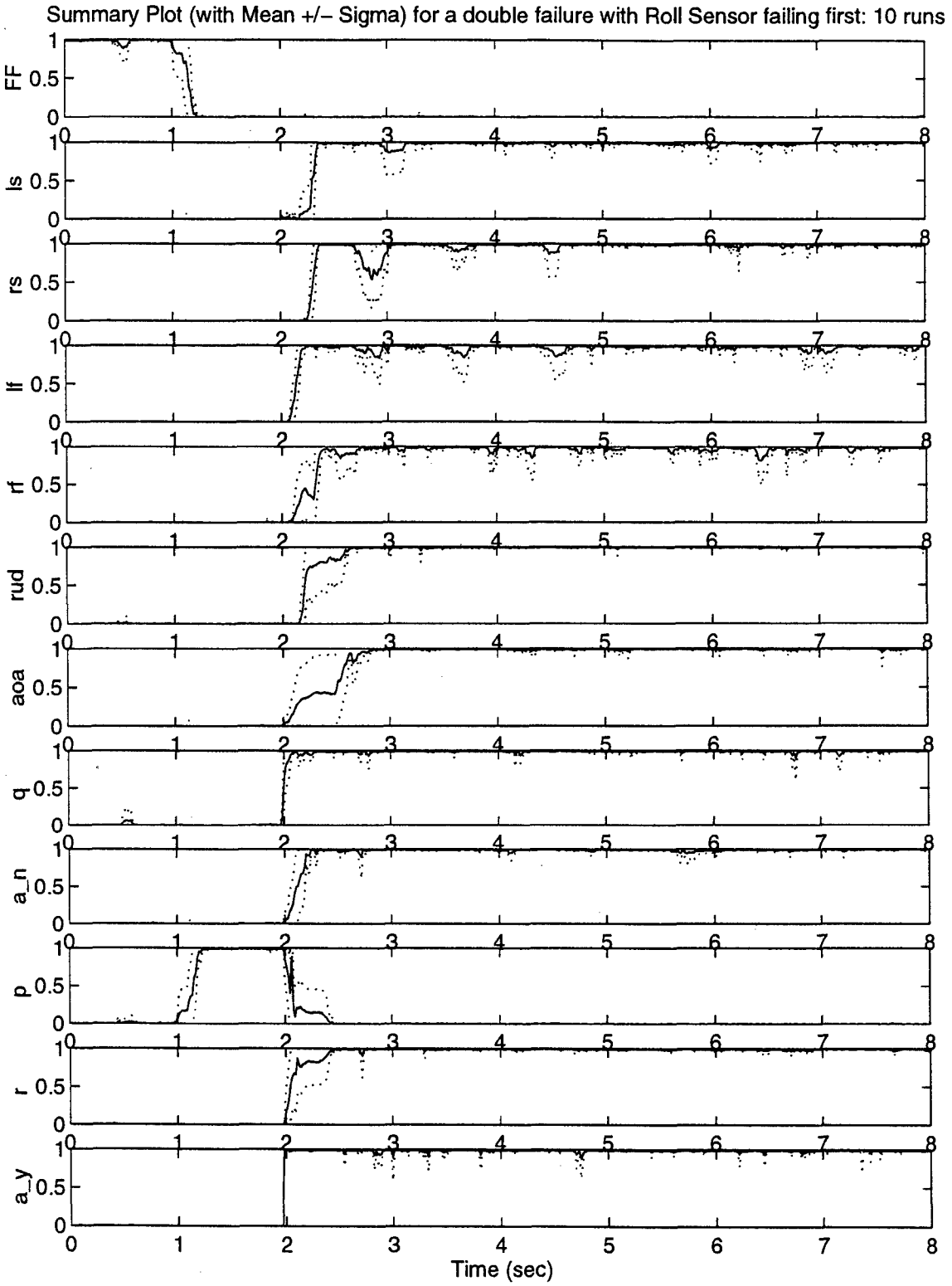

Figure B-9 Double Failure Summary Plot - No Control Redistribution

Roll Rate Sensor Failing First

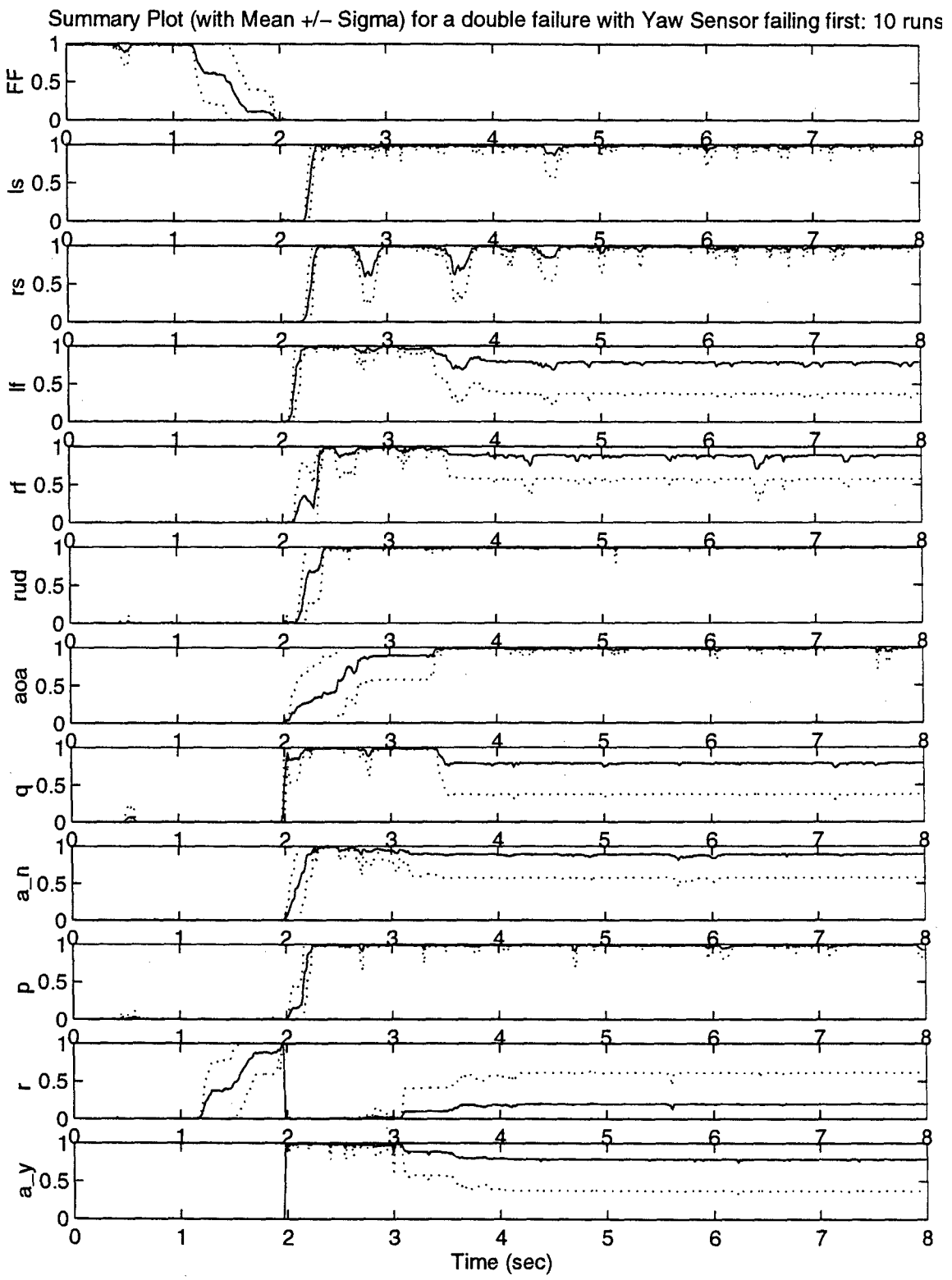

Figure B-10 Double Failure Summary Plot - No Control Redistribution

Yaw Rate Sensor Failing First

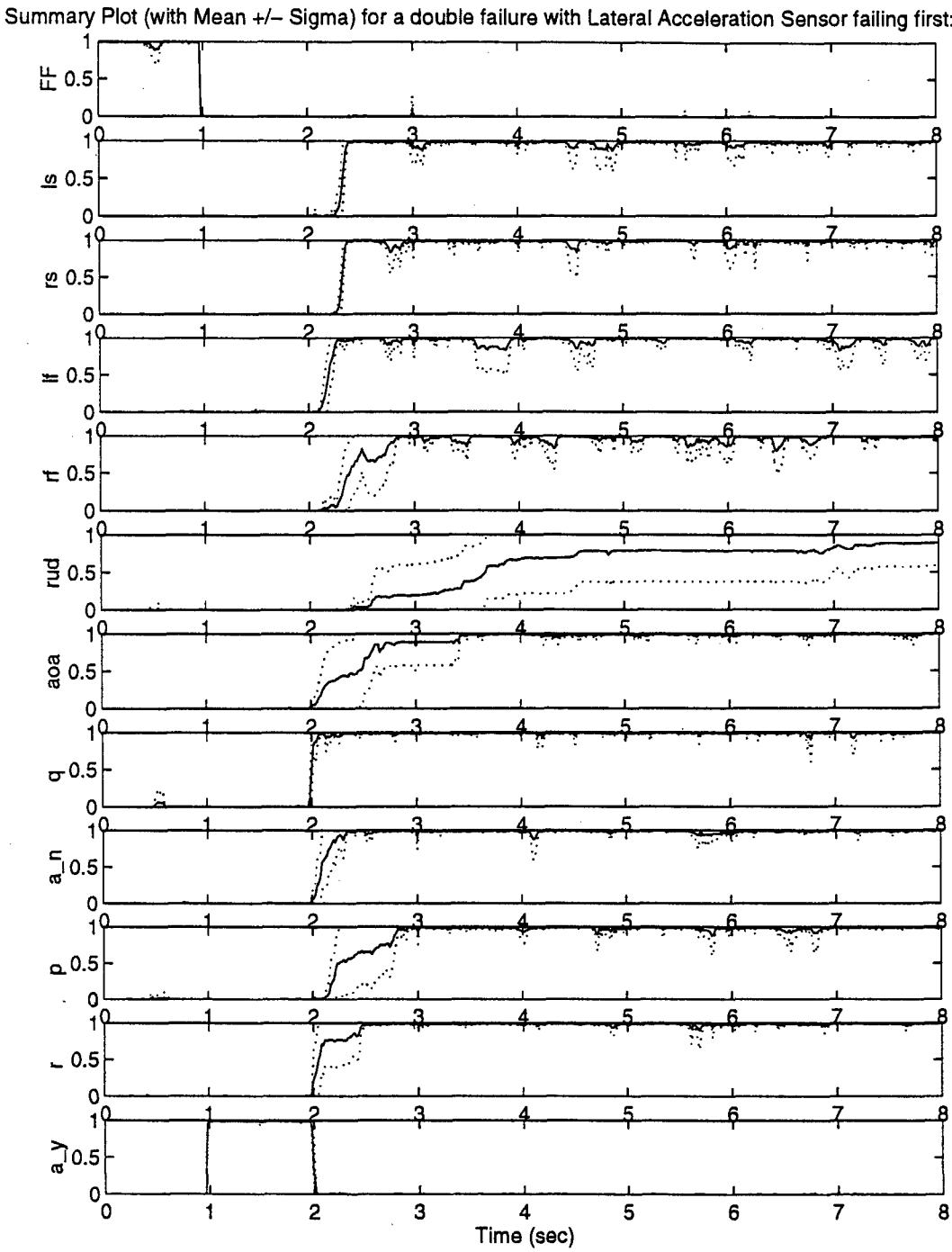

Figure B-11 Double Failure Summary Plot - No Control Redistribution

Lateral Acceleration Sensor Failing First

Appendix C

Double Failure Probability Plots for MMAE-based Block 40 with Scaled Control Redistribution

This appendix contains the probability plots of a simulation which was run with Control Redistribution. Because of actuator rate saturation problems, the amount of magnitude of the redistributed commands had to be scaled by varying amounts. The details concerning this scaling are presented in Section 4.3.3. These plots can be compared to the corresponding ones resulting from the *no* Control Redistribution simulations found in Appendix B. As noted in Section 4.3.3.1, the various scale factors for these simulations were chosen to provide an *acceptable* level of detection and detection lock, with a *minimum* amount of redistribution reduction. These constraints were implemented so that the Control Redistribution algorithm would provide the maximum amount of improvement in control authority in the face of multiple component failures. With these criteria in mind, the reader should not be concerned that some of the probability plots in this appendix do not show significant improvement over those found in Appendix B; in fact, some simulations were even slightly degraded.

As in Appendix B, each of the following probability summary plots is the result of 11 separate simulations. Each simulation is shown with a separate subplot, and each subplot shows the mean of 10 Monte Carlo runs (solid line) and the ± one sigma value (dotted line). The first subplot shown represents the probability of the Kalman filter which hypothesizes a fully functional (FF) aircraft, and therefore should drop from one to zero soon after the injection of the first failure at one second. The remaining 11 represent simulations in which the first failure (injected at one second) is combined with a particular secondary failure (injected at two seconds). The exception to this is the location which, by

this convention, would be showing the double failure of the same component (an impossible scenario). In this location, the probability plot for the Kalman filter which represents the first failure, with no secondary failure, is presented. The simulation for this subplot is usually chosen as the simulation shown in the next consecutive subplot, with the exception of the first failure being the rudder (rud = last actuator), which utilizes the simulation for the lateral acceleration sensor (a_y) as the second failure, and the first failure being the lateral acceleration sensor (a_y = last sensor), which utilizes the simulation for the pitch sensor (q) as the second failure.

Note that because of relatively long titles, some of the figures in this section are scaled to a slightly smaller size, the most notable being Figure C-8. The plots in these figures are no different from the others, they've only been proportionally scaled to fit the margins of the page.

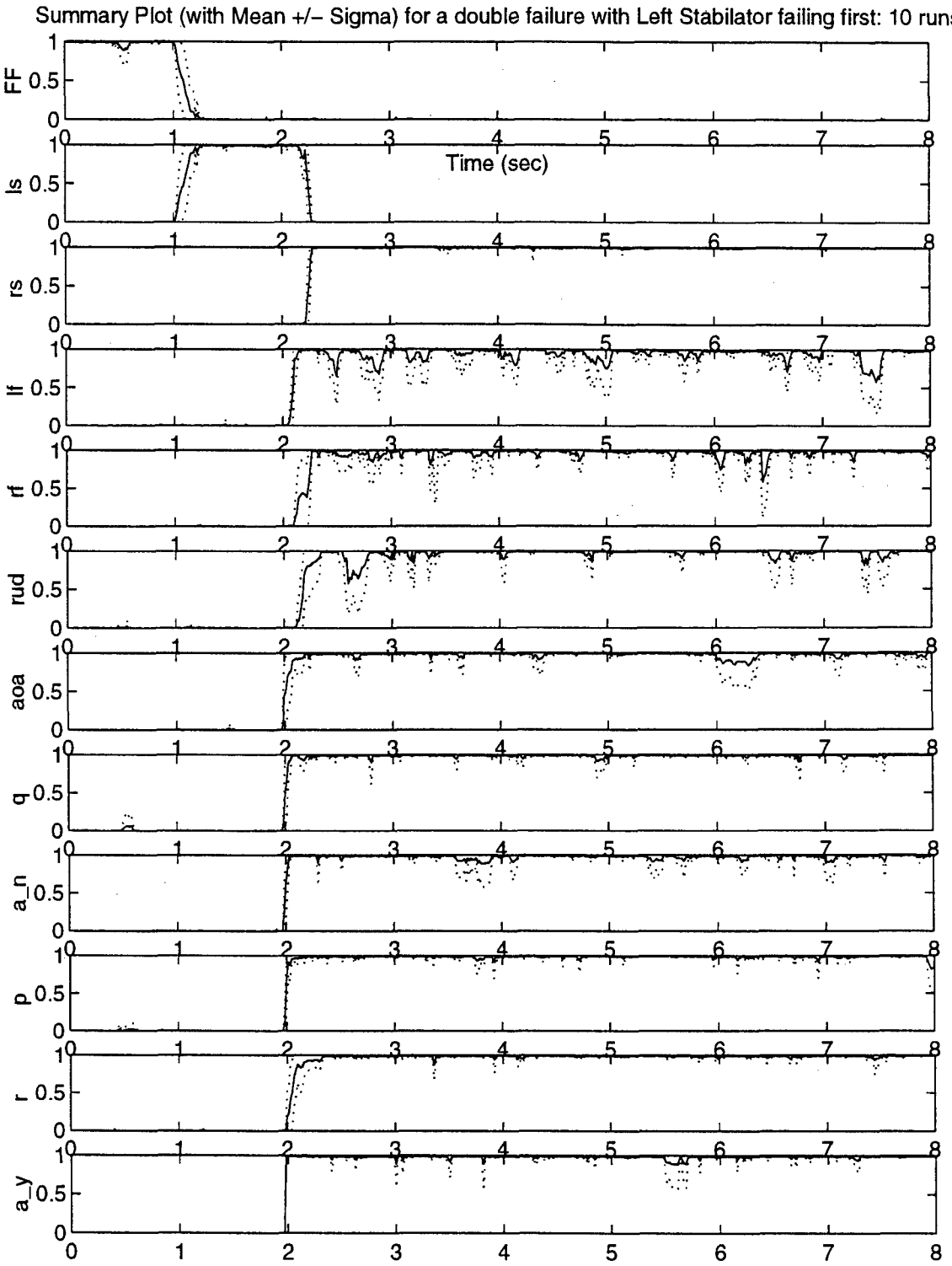

Figure C-1 Double Failure Summary Plot - Scaled Control Redistribution

Left Stabilator Failing First

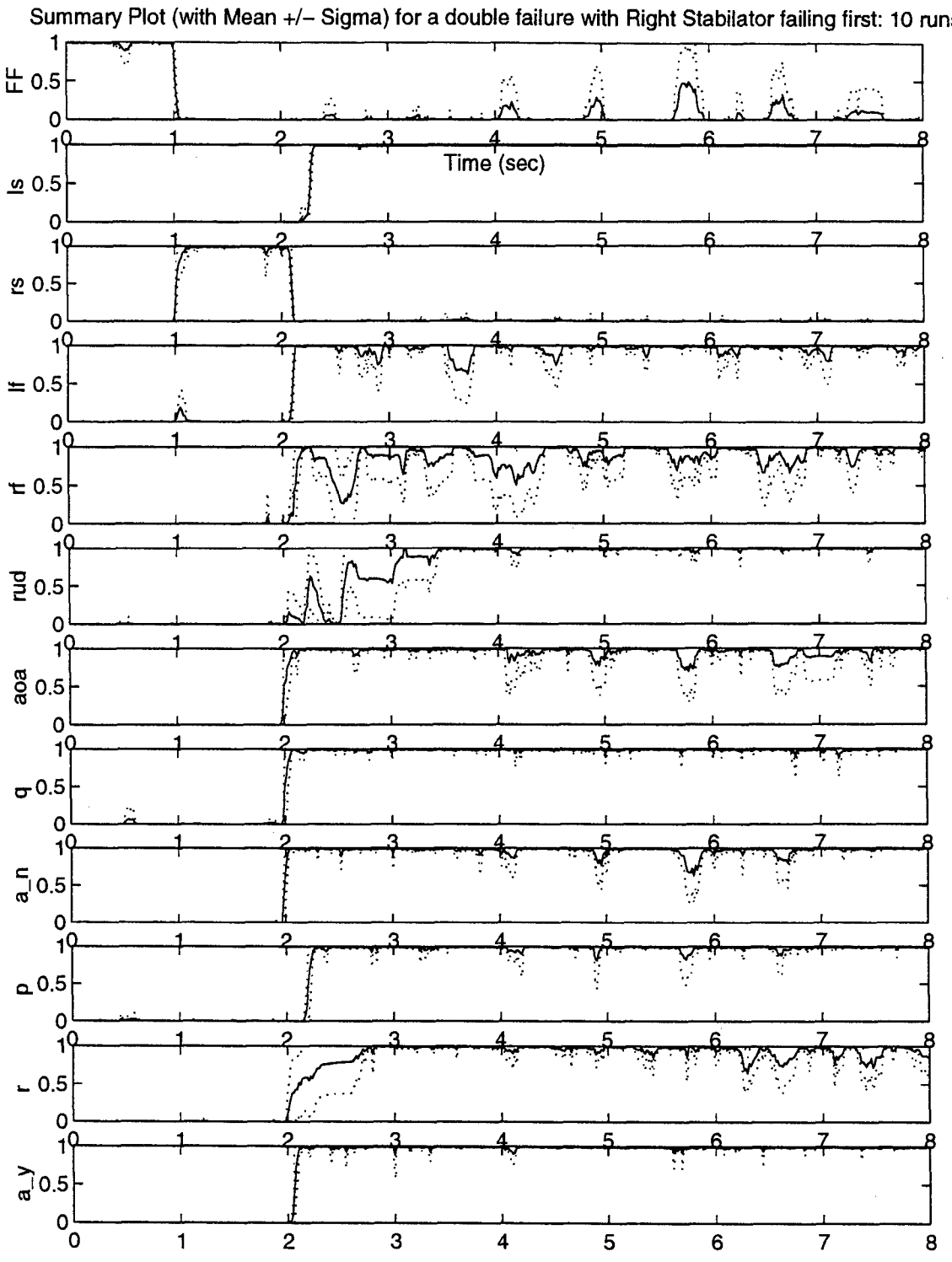

Figure C-2 Double Failure Summary Plot - Scaled Control Redistribution

Right Stabilator Failing First

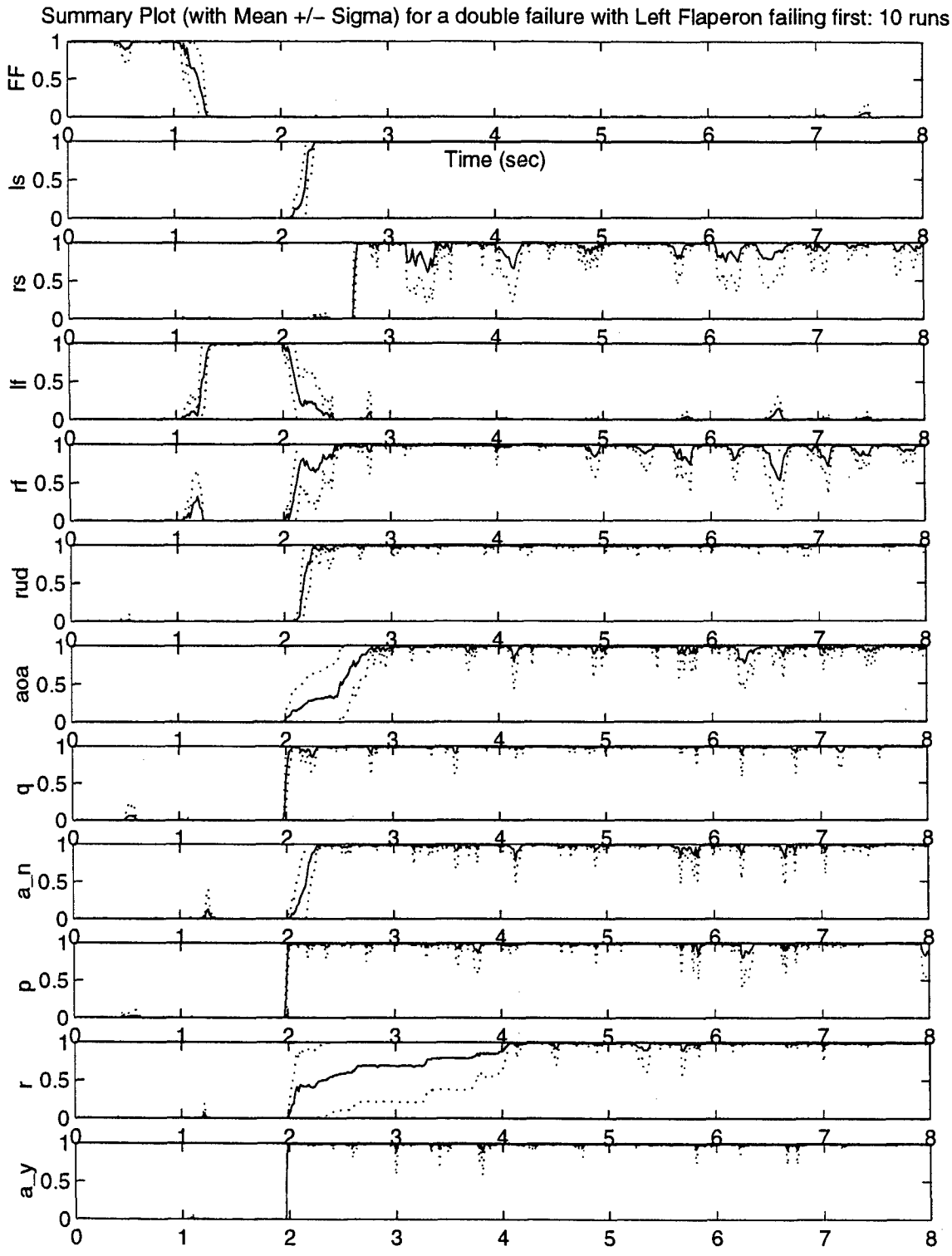

Figure C-3 Double Failure Summary Plot - Scaled Control Redistribution

Left Flaperon Failing First

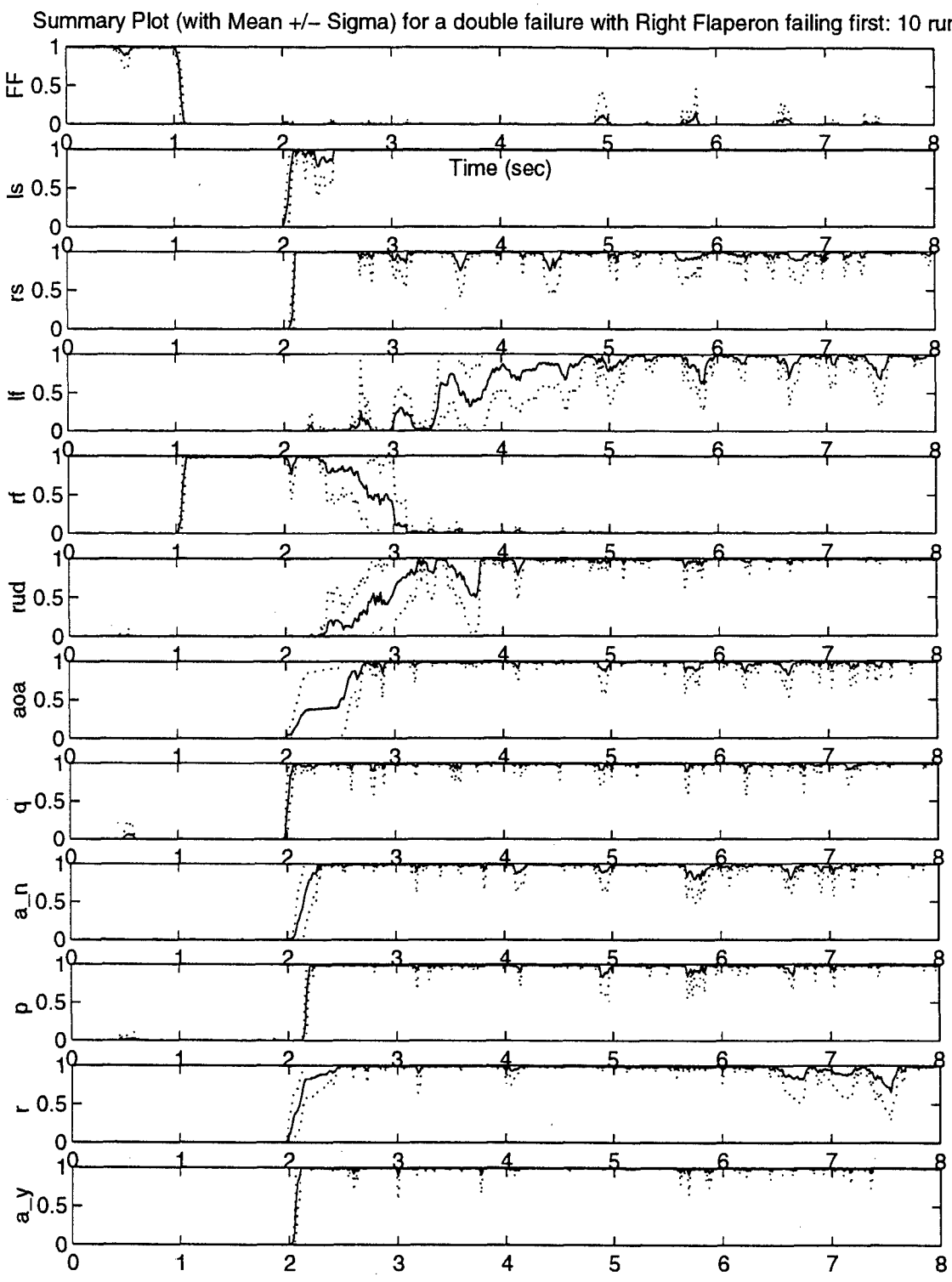

Figure C-4 Double Failure Summary Plot - Scaled Control Redistribution

Right Flaperon Failing First

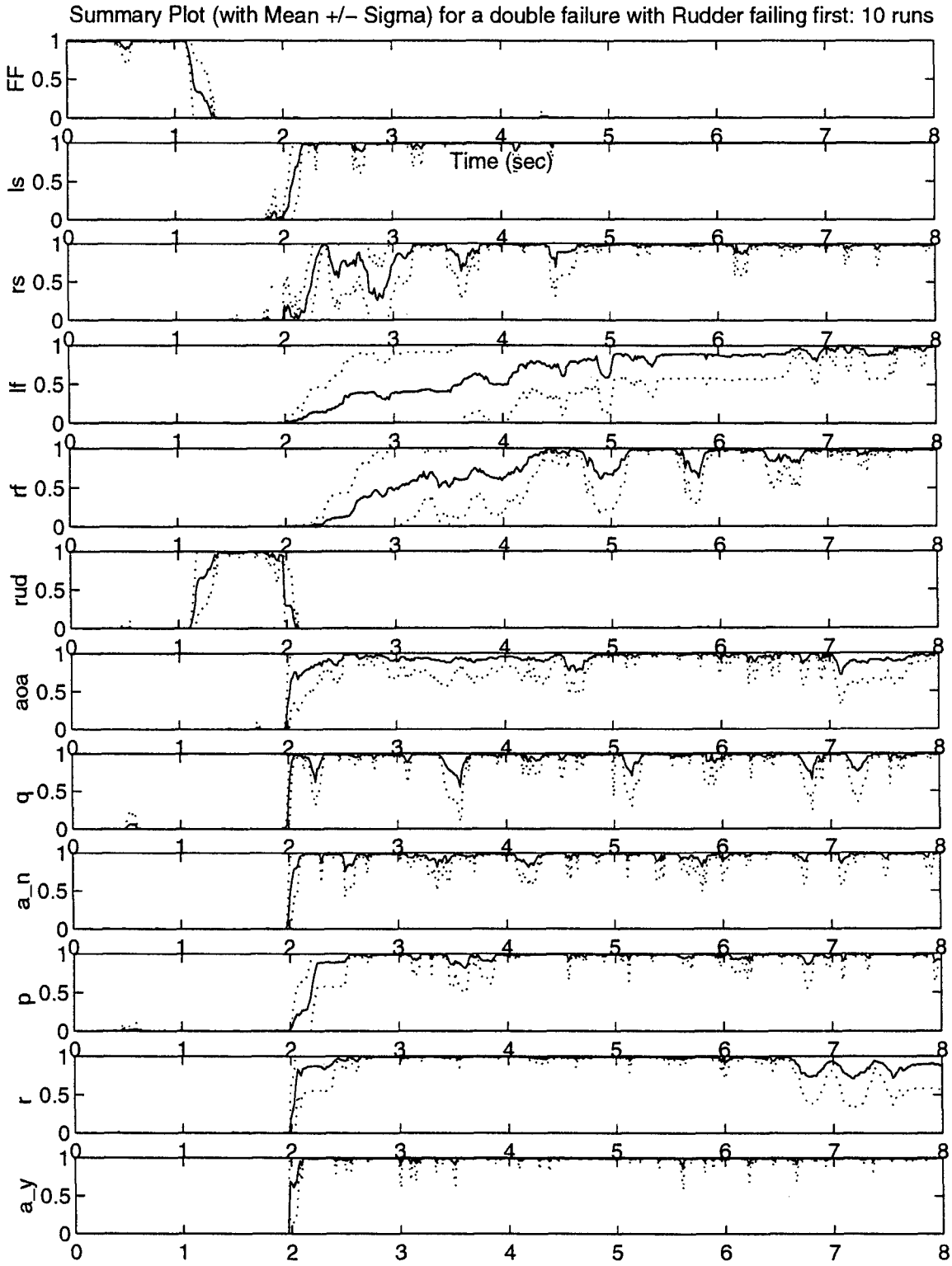

Figure C-5 Double Failure Summary Plot - Scaled Control Redistribution

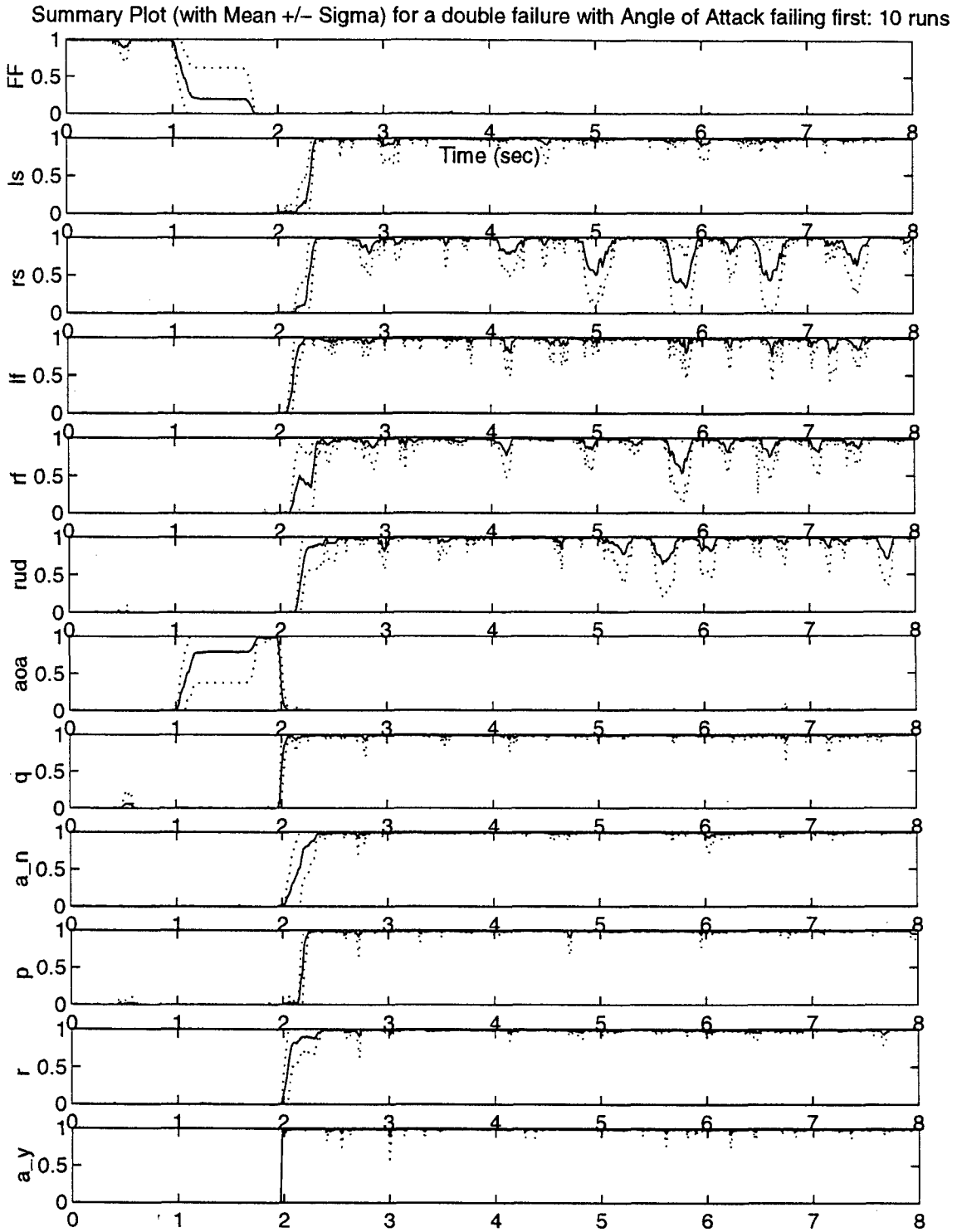

Figure C-6 Double Failure Summary Plot - Scaled Control Redistribution

Angle of Attack Sensor Failing First

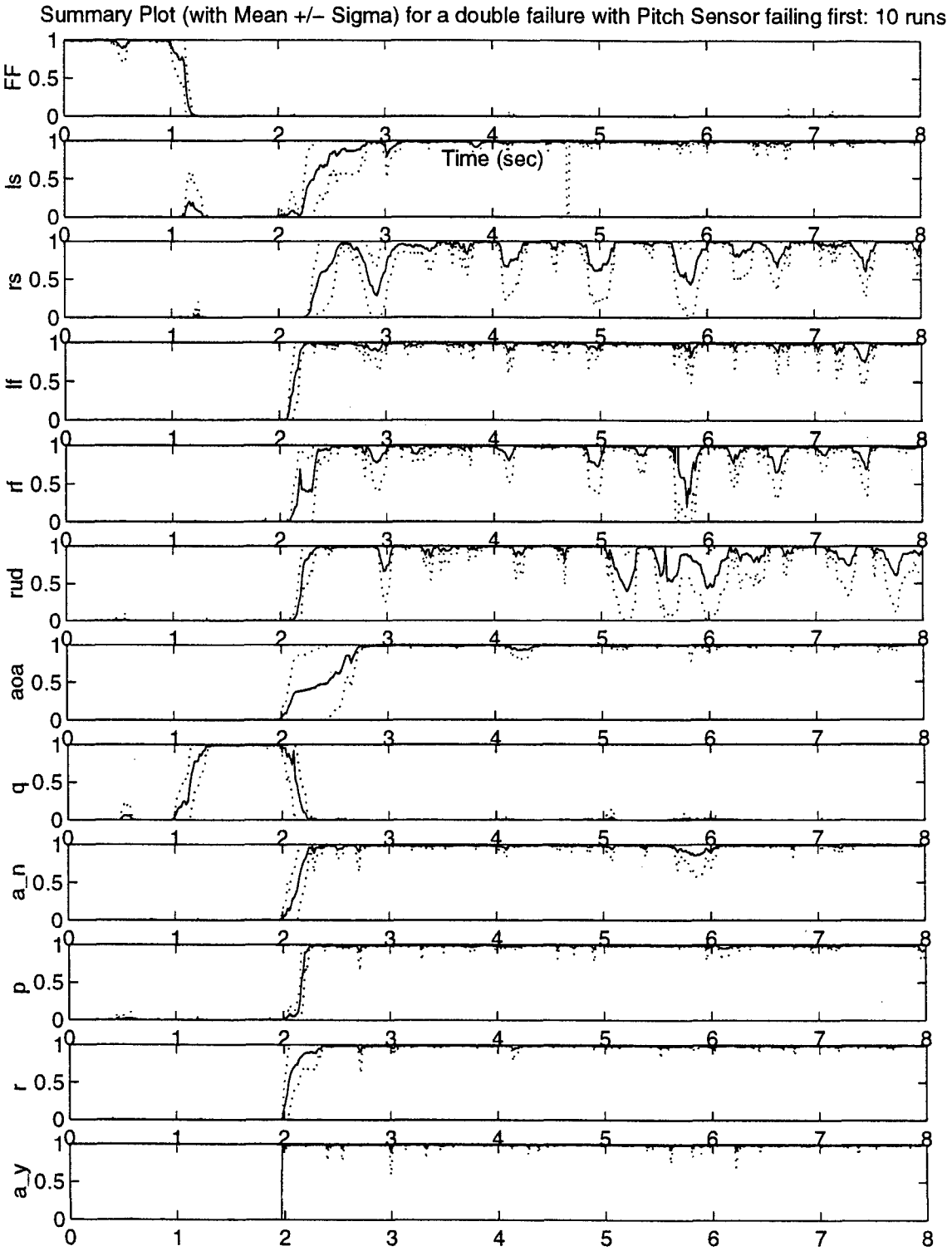

Figure C-7 Double Failure Summary Plot - Scaled Control Redistribution

Pitch Rate Sensor Failing First

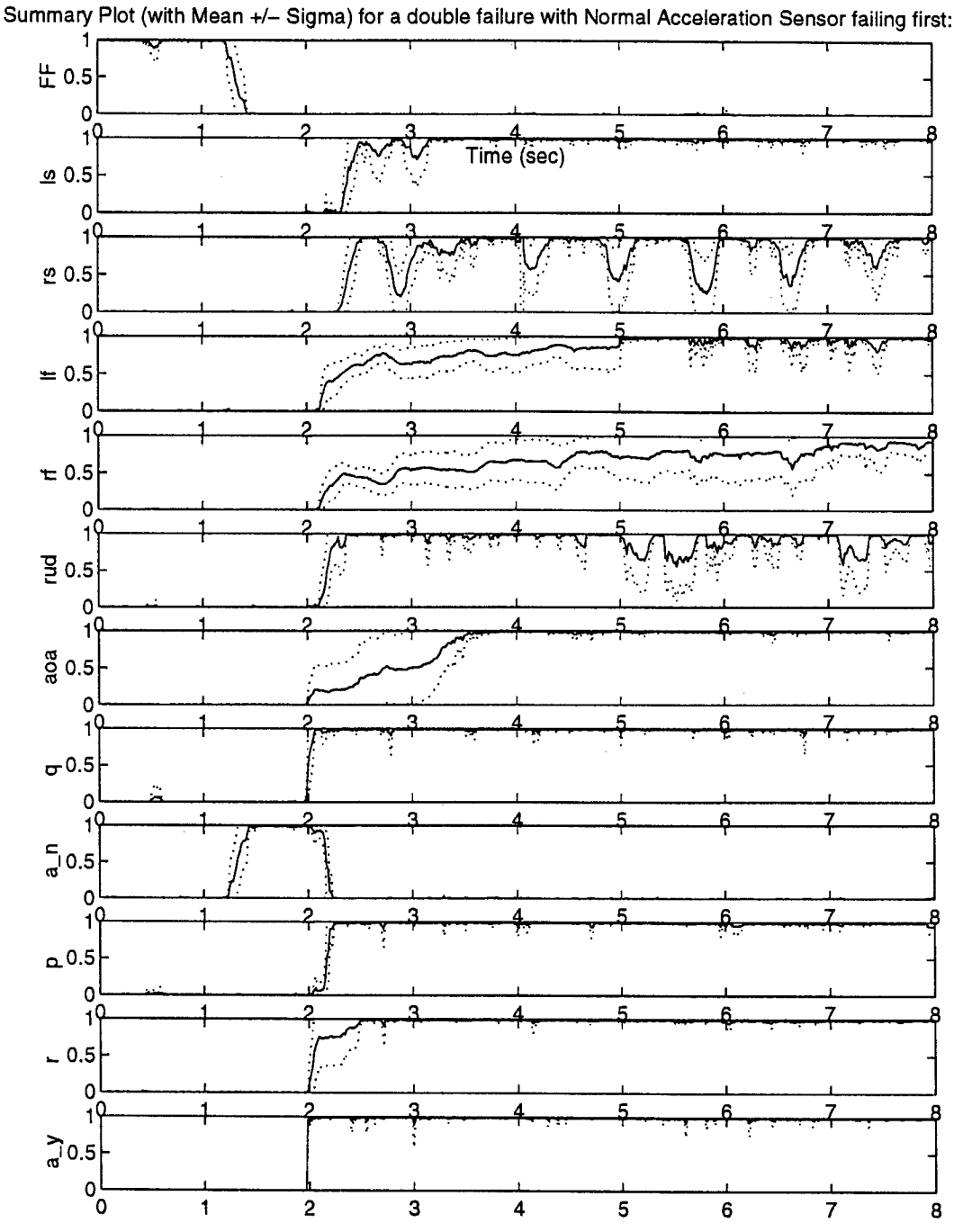

Figure C-8 Double Failure Summary Plot - Scaled Control Redistribution

Normal Acceleration Sensor Failing First

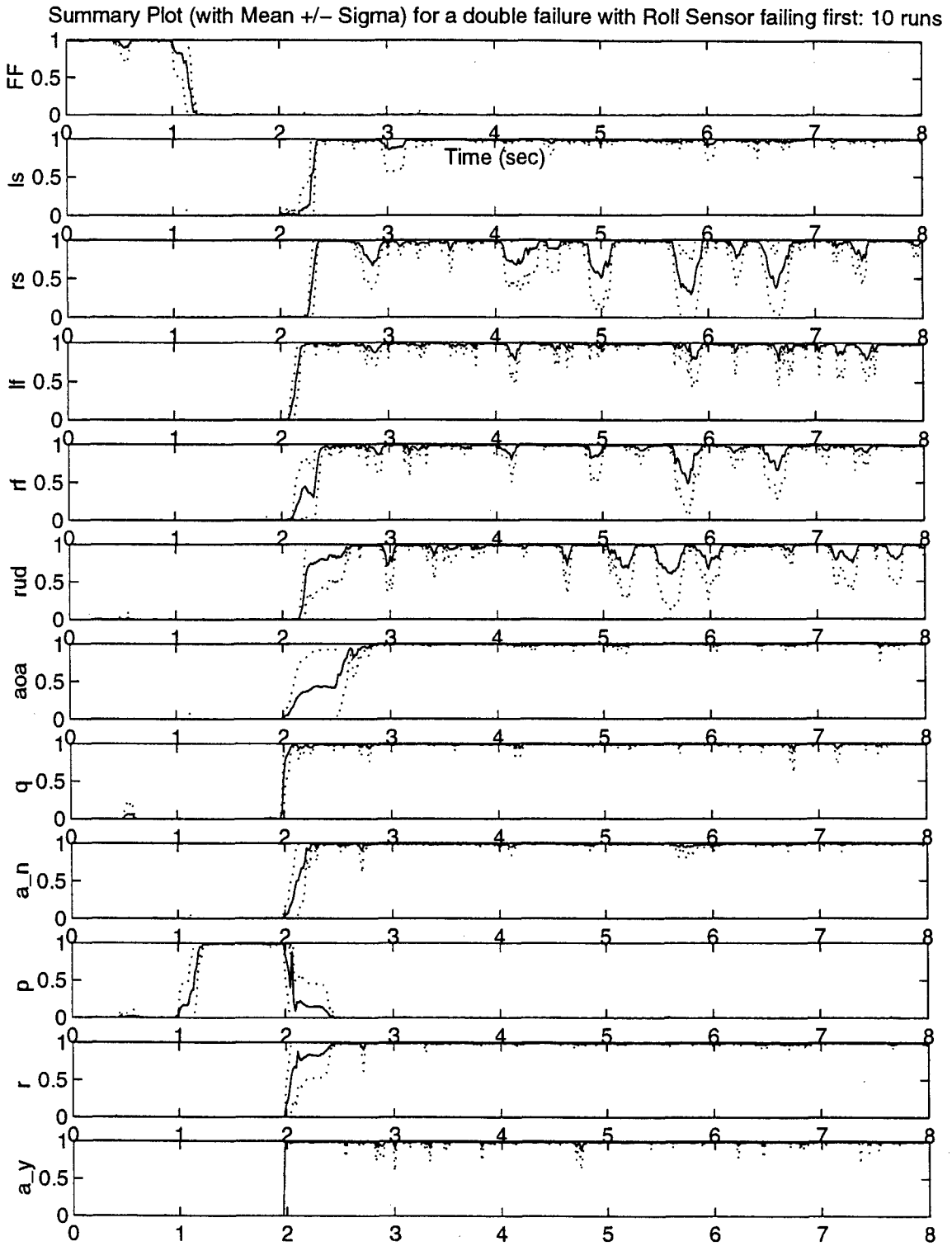

Figure C-9 Double Failure Summary Plot - Scaled Control Redistribution

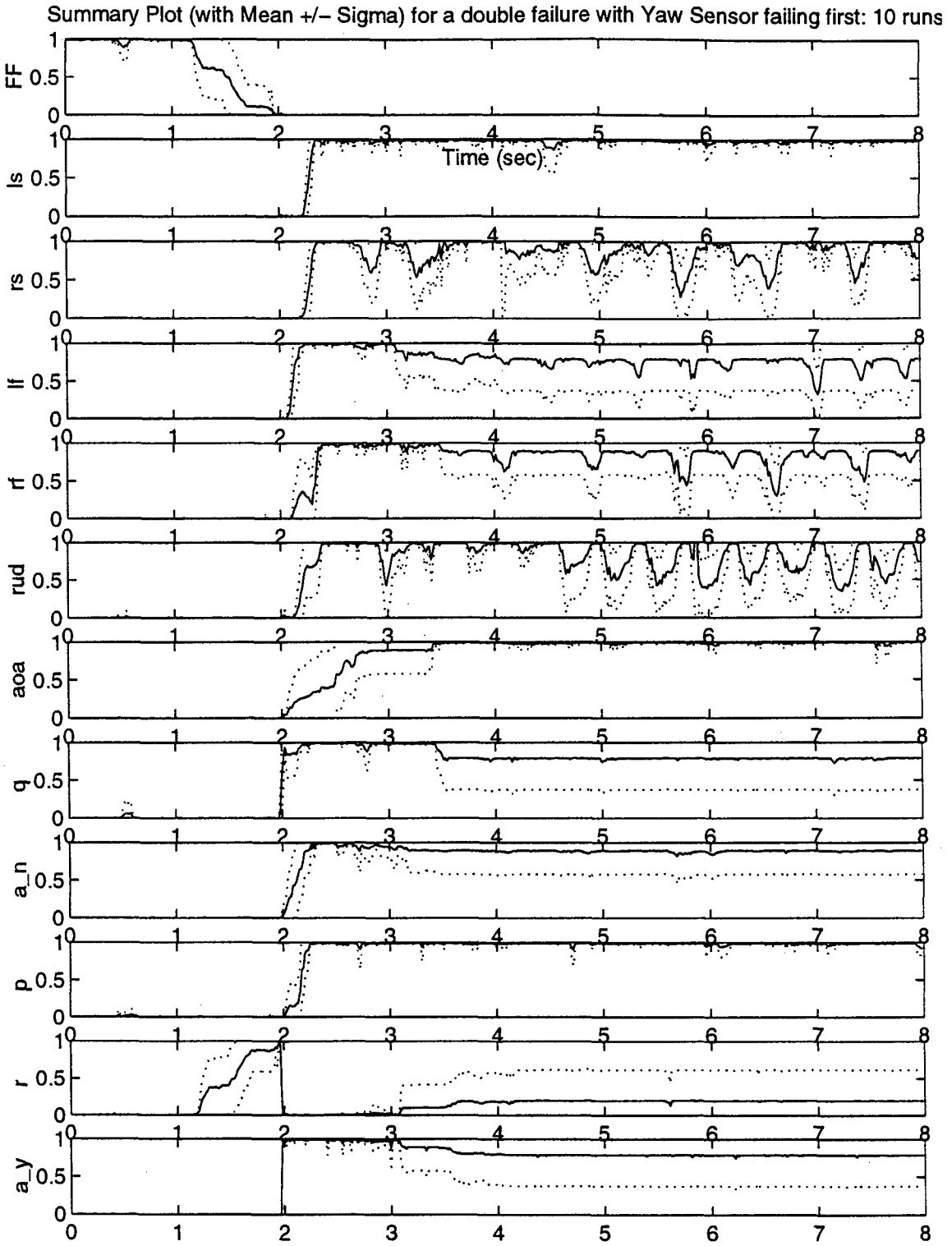

Figure C-10 Double Failure Summary Plot - Scaled Control Redistribution

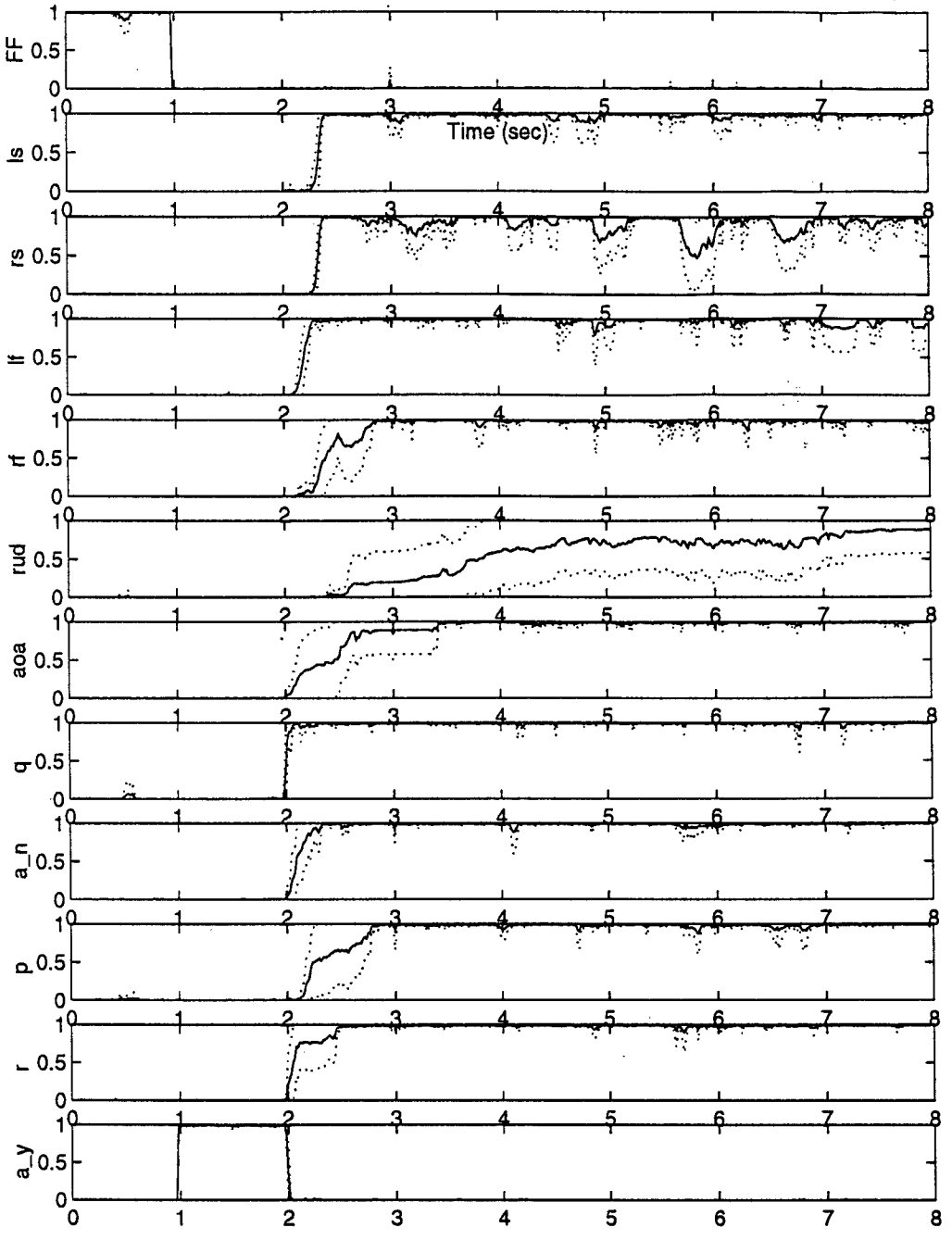

Figure C-11 Double Failure Summary Plot - Scaled Control Redistribution

Lateral Acceleration Sensor Failing First

Appendix D

Output Plots of Double Failures

This appendix contains output plots for the analysis of double failures of the VISTA F-16 using a standard Block 40 controller preceded by an MMAE algorithm, both with and without control redistribution. Plots represent simulations which were performed using separate pitch, roll and yaw doublet inputs. Specific analysis of these plots can be found in Section 4.3.

This appendix is structured into three sections, each representing the response of the system to a either a pitch, roll or yaw doublet. Each section is subdivided into a simulation which does not utilize Control Redistribution, and one that does. Each of these subdivisions contains ten plots, one for each actuator double failure combination, and each plot contains 15 subplots representing the various longitudinal (first column) and lateral (second column) output variables of the system, along with the output positions of the five actuators (third column).

The subplots are once again further subdivided into two separate traces. The first trace shows the mean of the responses from 10 Monte Carlo runs of the simulation against a fully functional aircraft. This trace is represented by a dashed line, and its ± one sigma value is shown with a dashed dotted line. Note that if the variance is small, these lines will coincide and appear as solid. The second trace represents the mean of the responses from 10 Monte Carlo runs during the particular double failure combination. This trace is represented by a solid line, with the ± one sigma value shown by a dotted line.

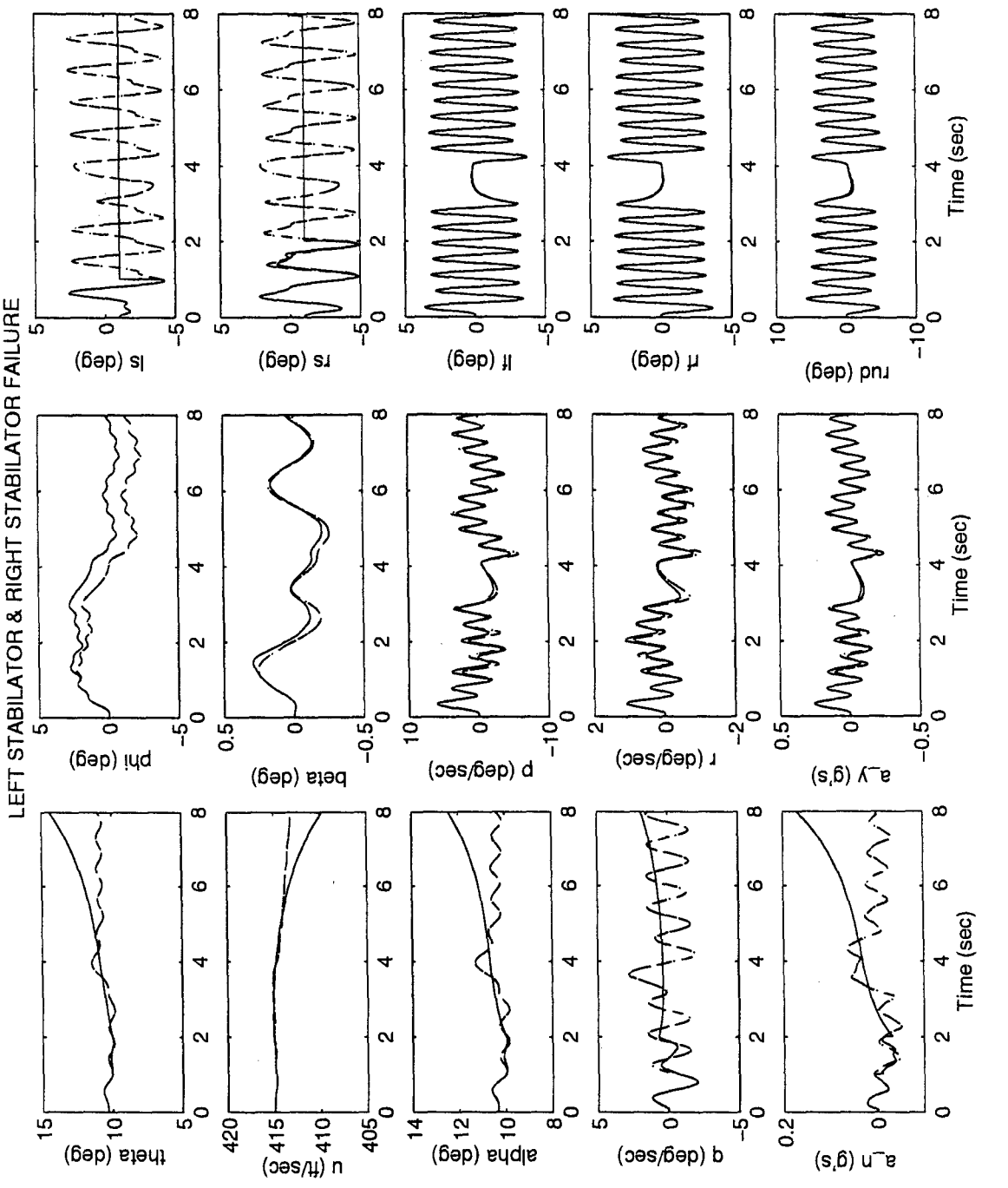

Figure D-1 MMAE-Based - Pitch Doublet - No Control Redistribution
Left Stabilator & Right Stabilator Failure

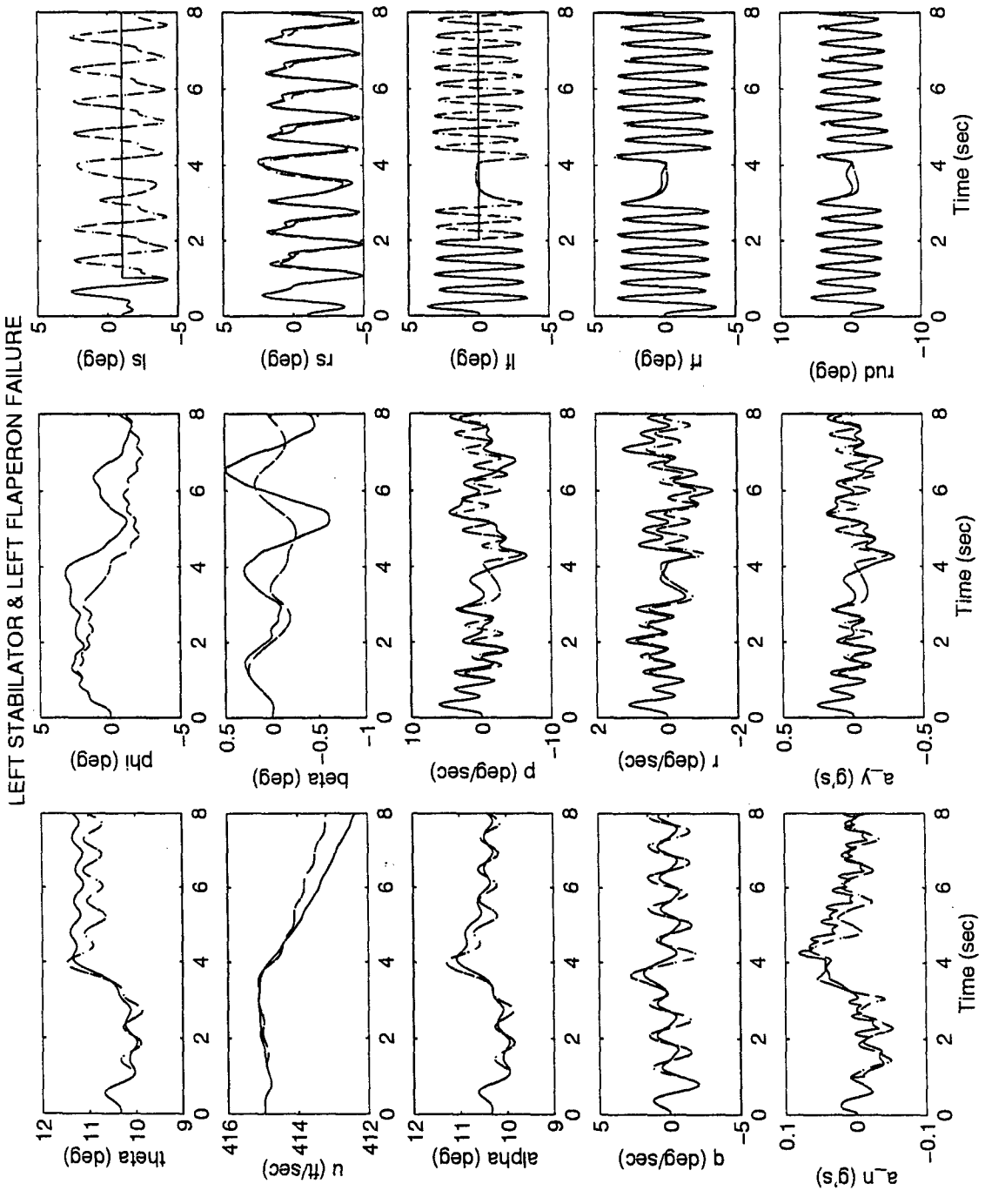

Figure D-2 MMAE-Based - Pitch Doublet - No Control Redistribution

Left Stabilator & Left Flaperon Failure

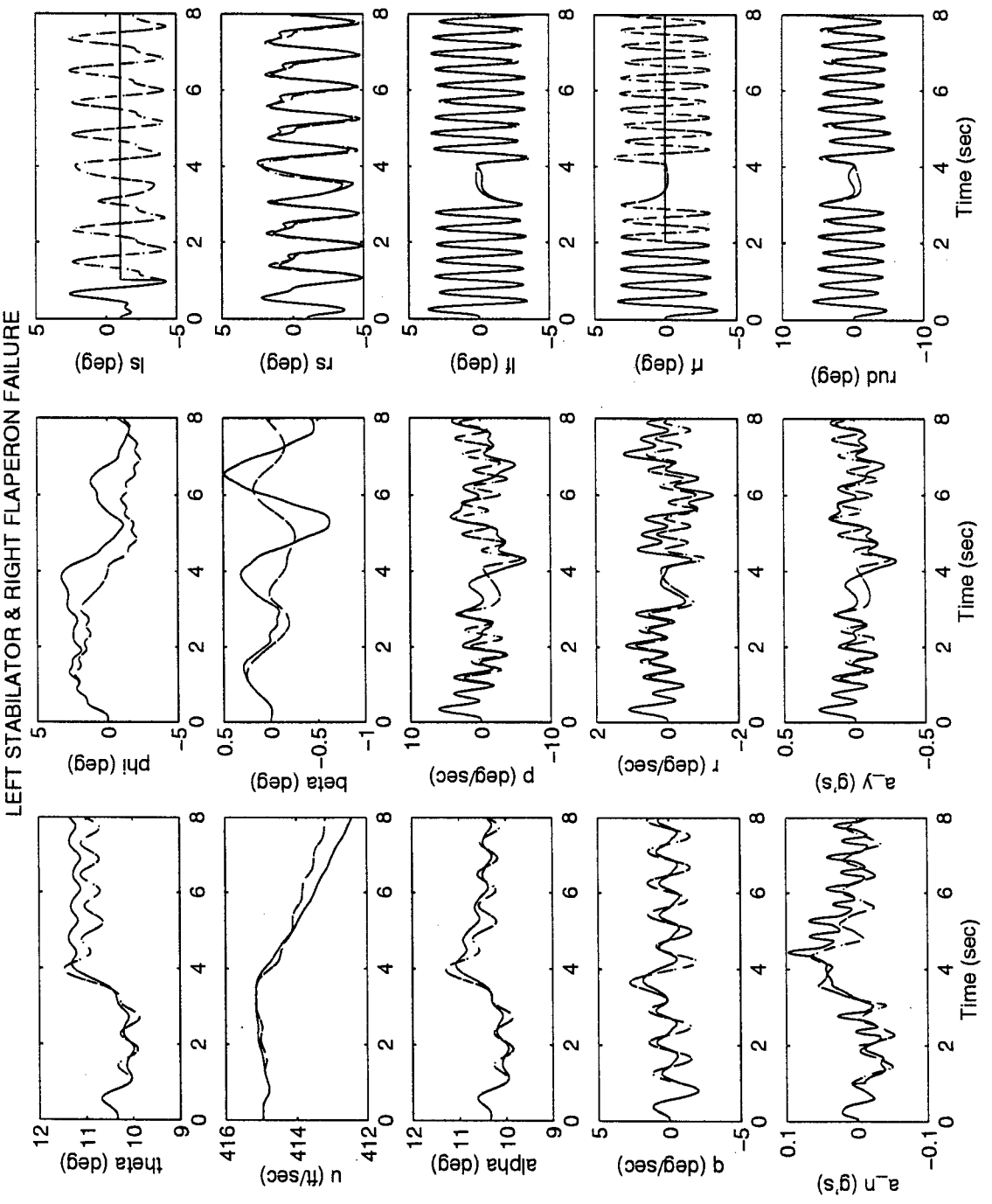

Figure D-3 MMAE-Based - Pitch Doublet - No Control Redistribution
Left Stabilator & Right Flaperon Failure

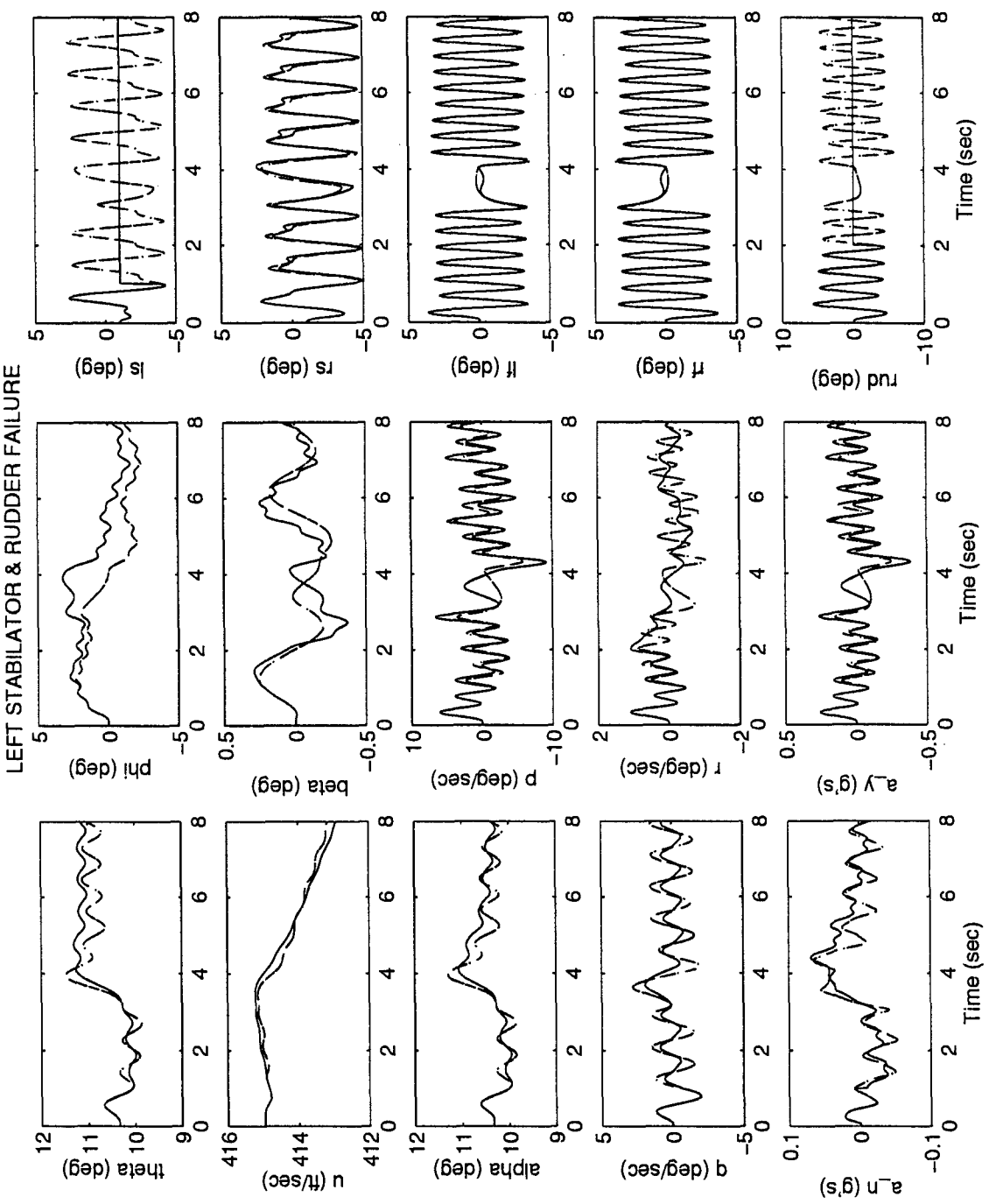

Figure D-4 MMAE-Based - Pitch Doublet - No Control Redistribution
Left Stabilator & Rudder Failure

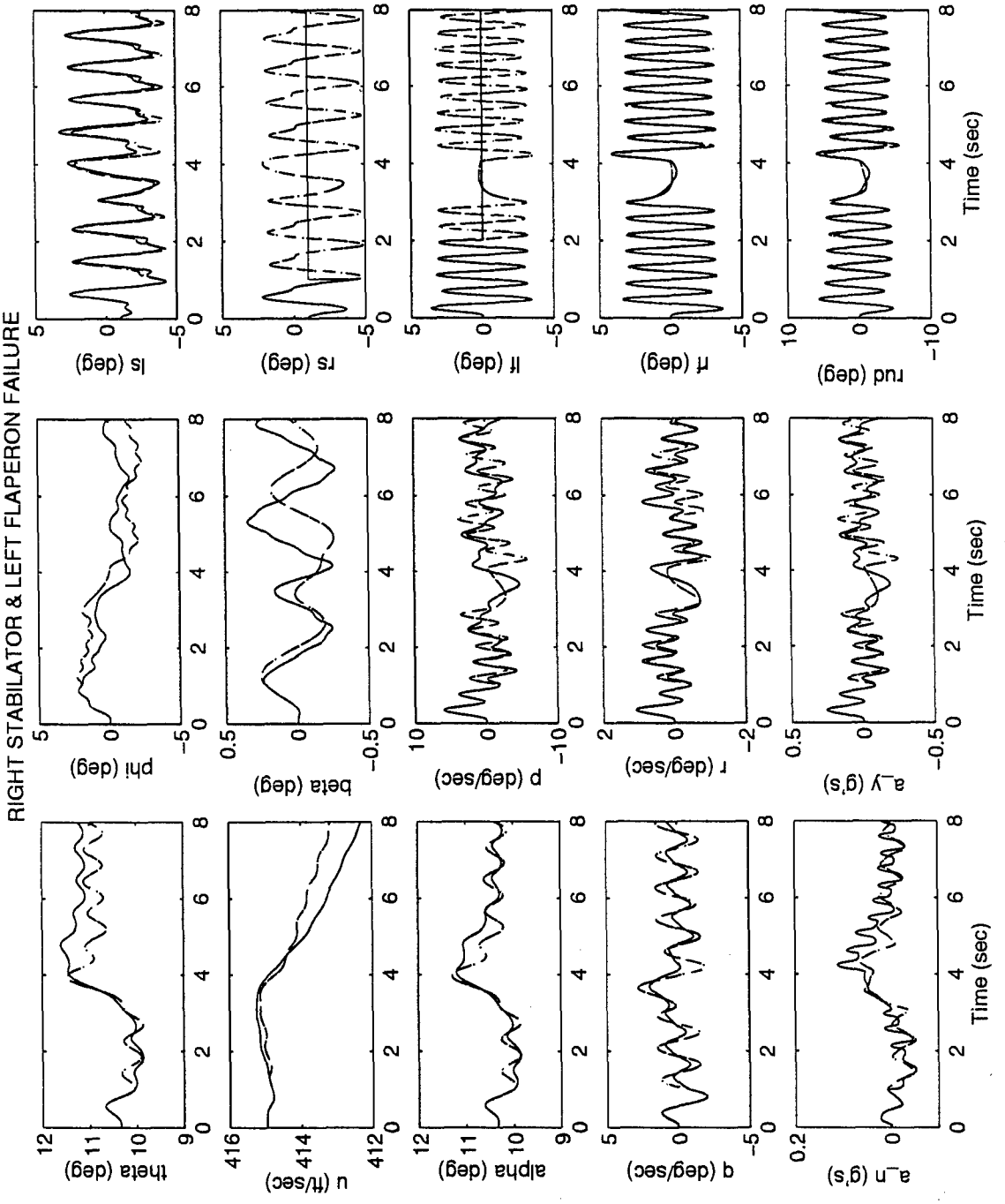

Figure D-5 MMAE-Based - Pitch Doublet - No Control Redistribution
Right Stabilator & Left Flaperon Failure

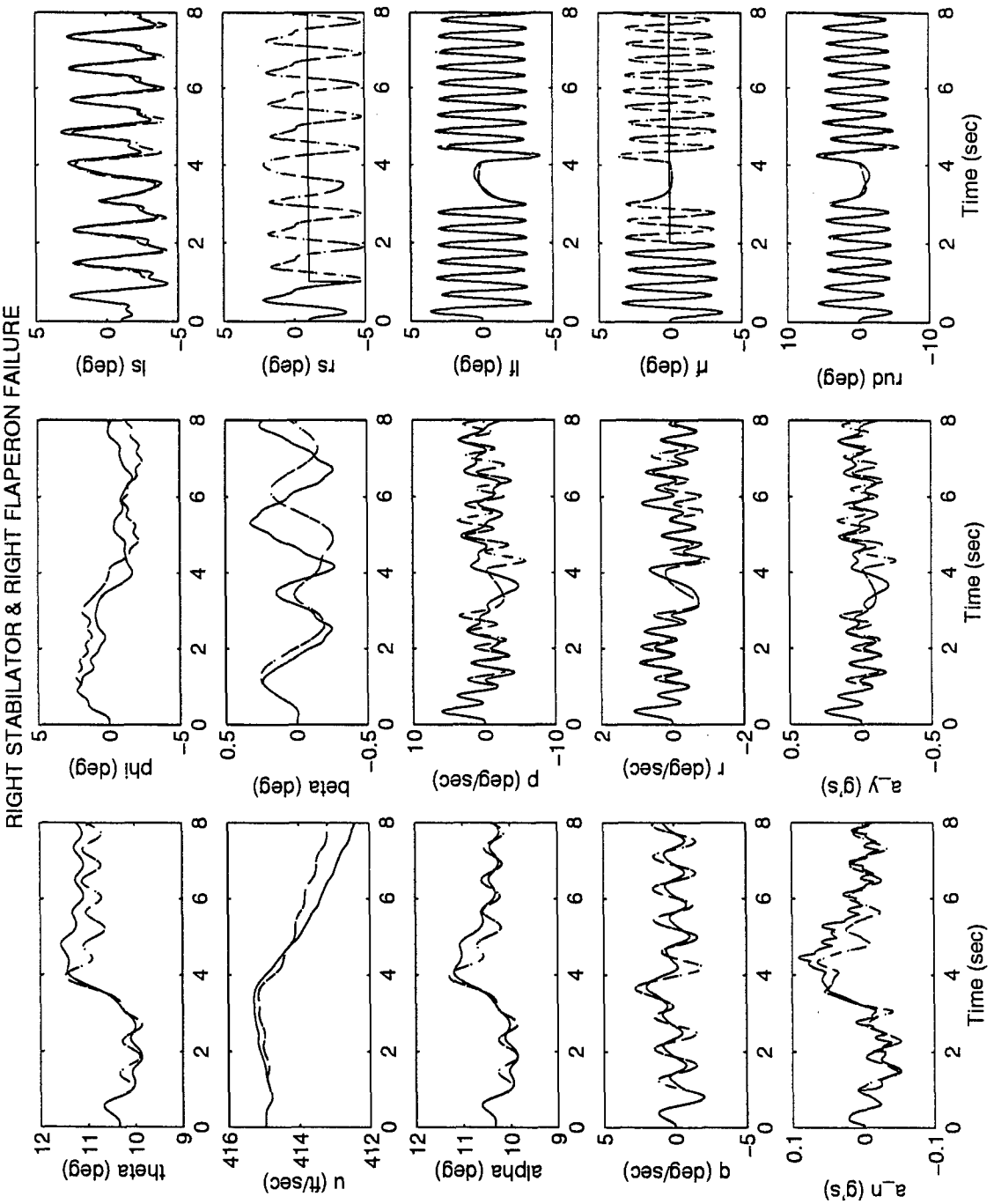

Figure D-6 MMAE-Based - Pitch Doublet - No Control Redistribution
Right Stabilator & Right Flaperon Failure

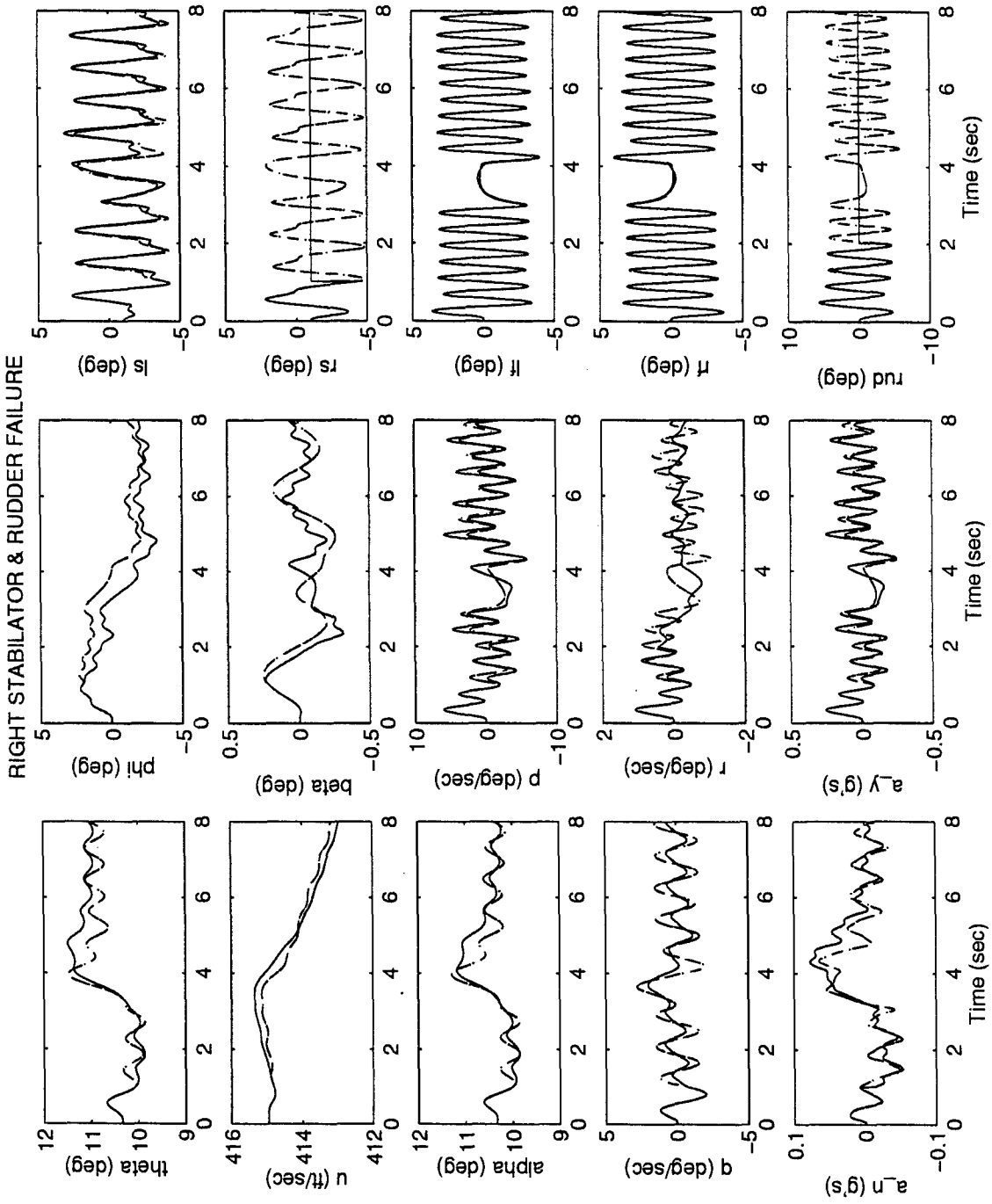

Figure D-7 MMAE-Based - Pitch Doublet - No Control Redistribution
Right Stabilator & Rudder Failure

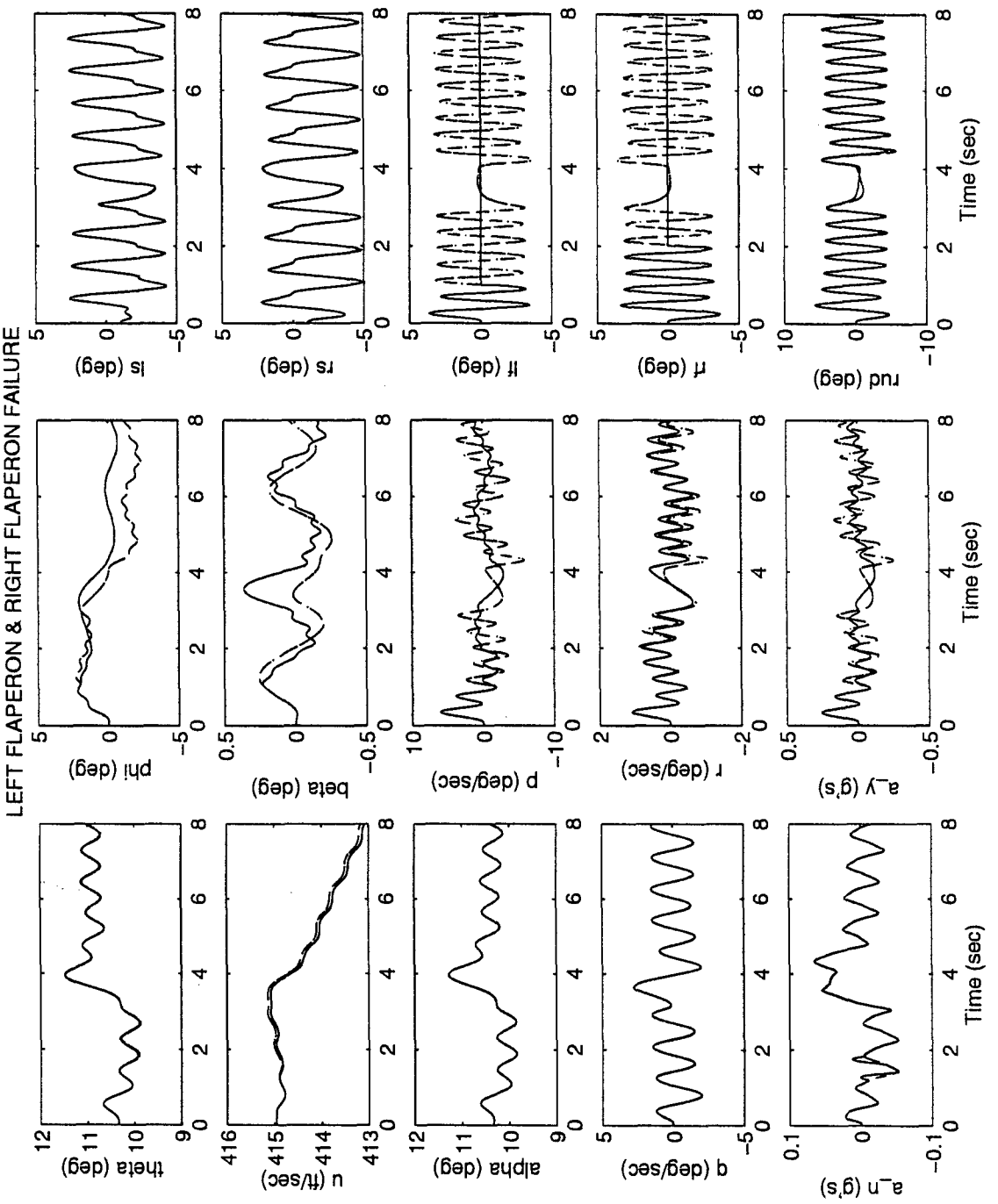

Figure D-8 MMAE-Based - Pitch Doublet - No Control Redistribution
Left Flaperon & Right Flaperon Failure

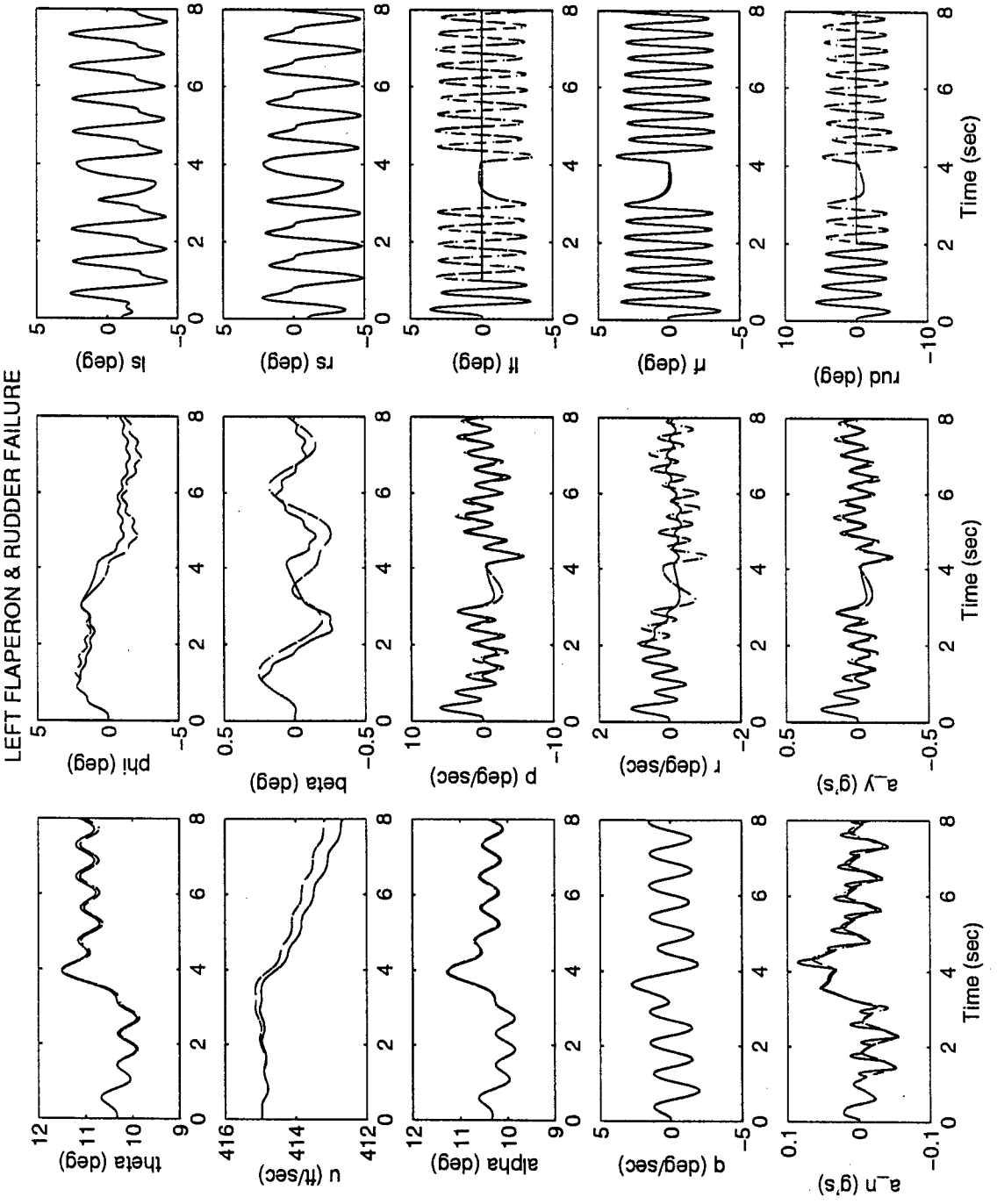

Figure D-9 MMAE-Based - Pitch Doublet - No Control Redistribution
Left Flaperon & Rudder Failure

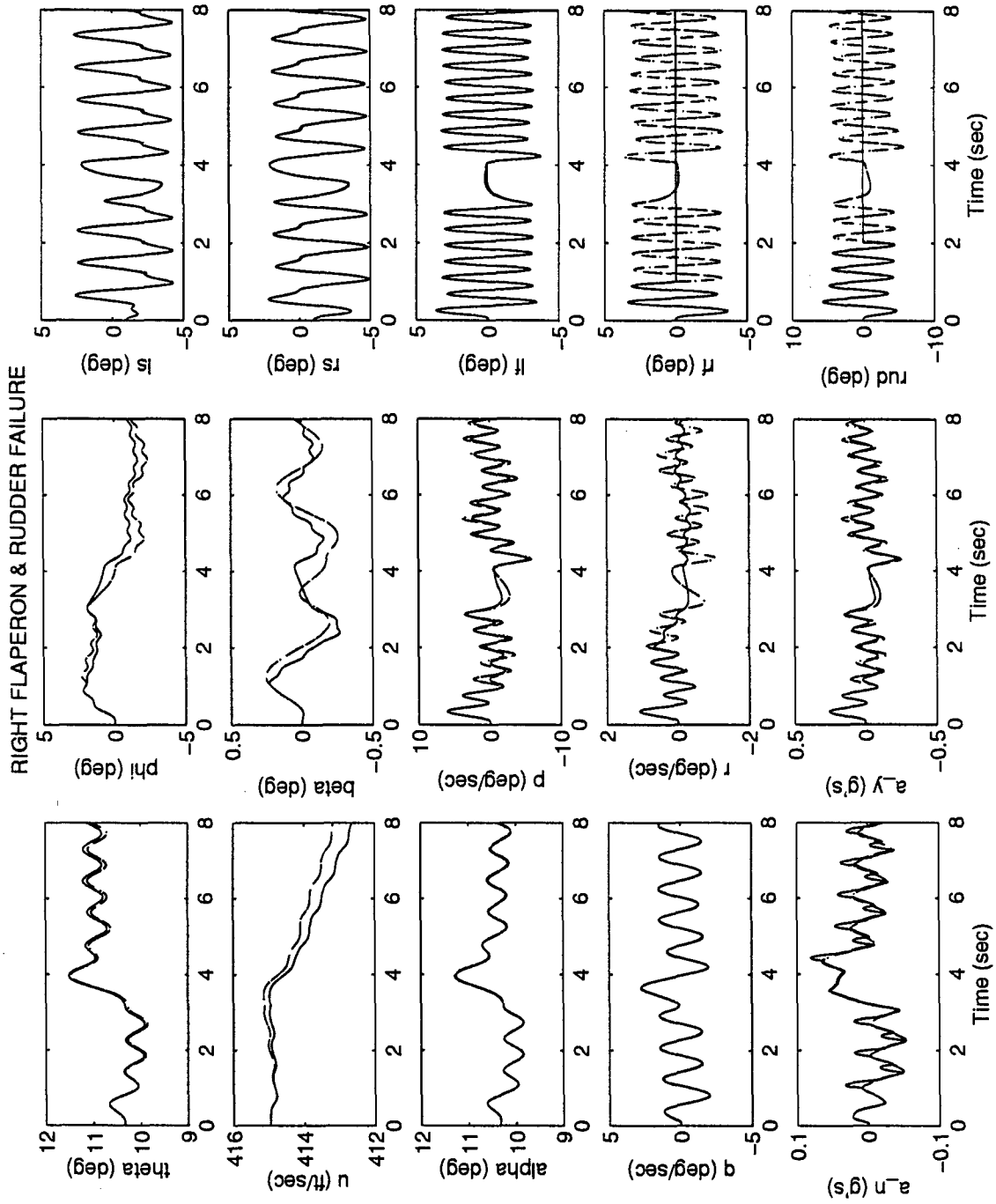

Figure D-10 MMAE-Based - Pitch Doublet - No Control Redistribution
Right Flaperon & Rudder Failure

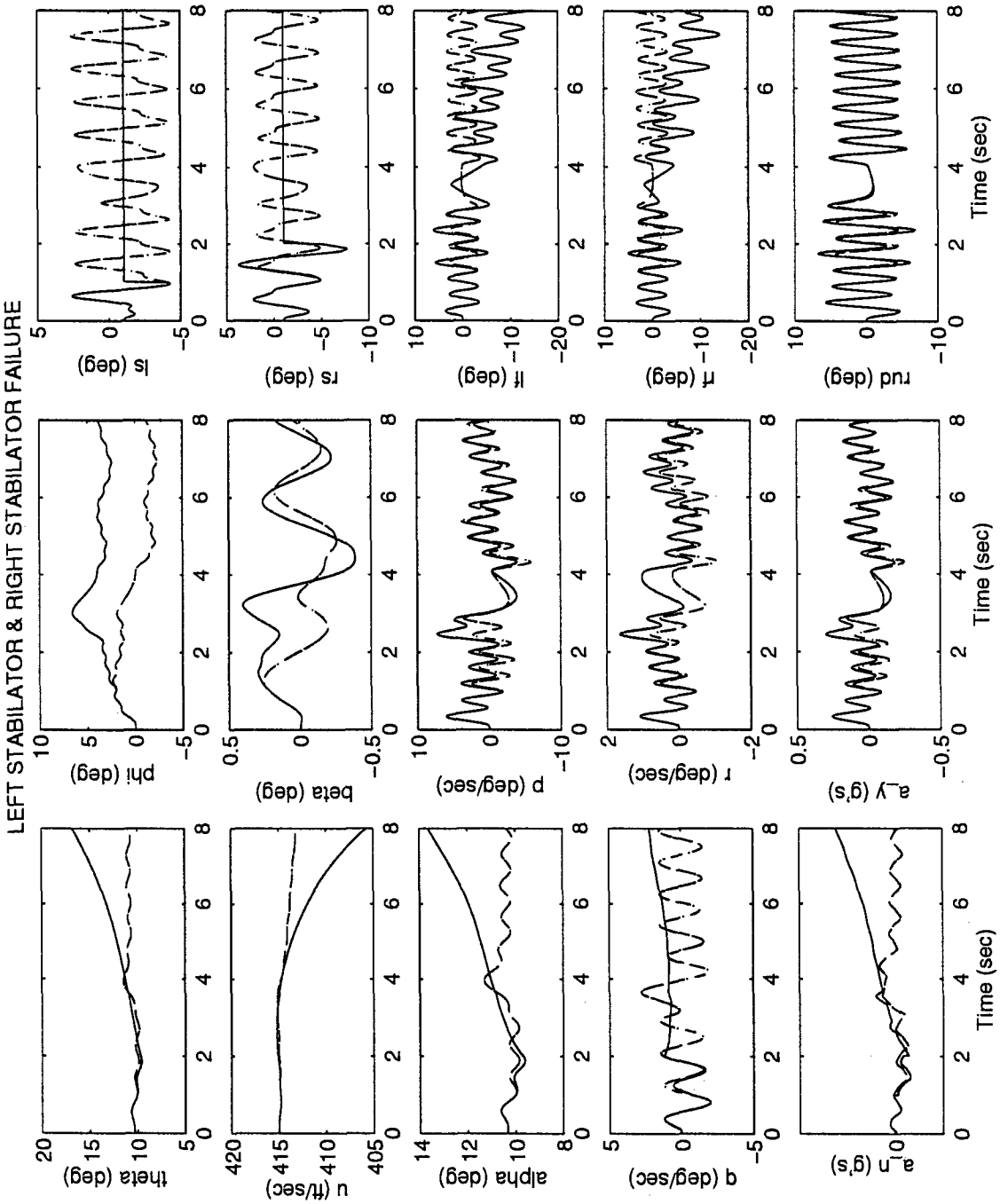

Figure D-11 MMAE-Based - Pitch Doublet - With Control Redistribution
Left Stabilator & Right Stabilator Failure

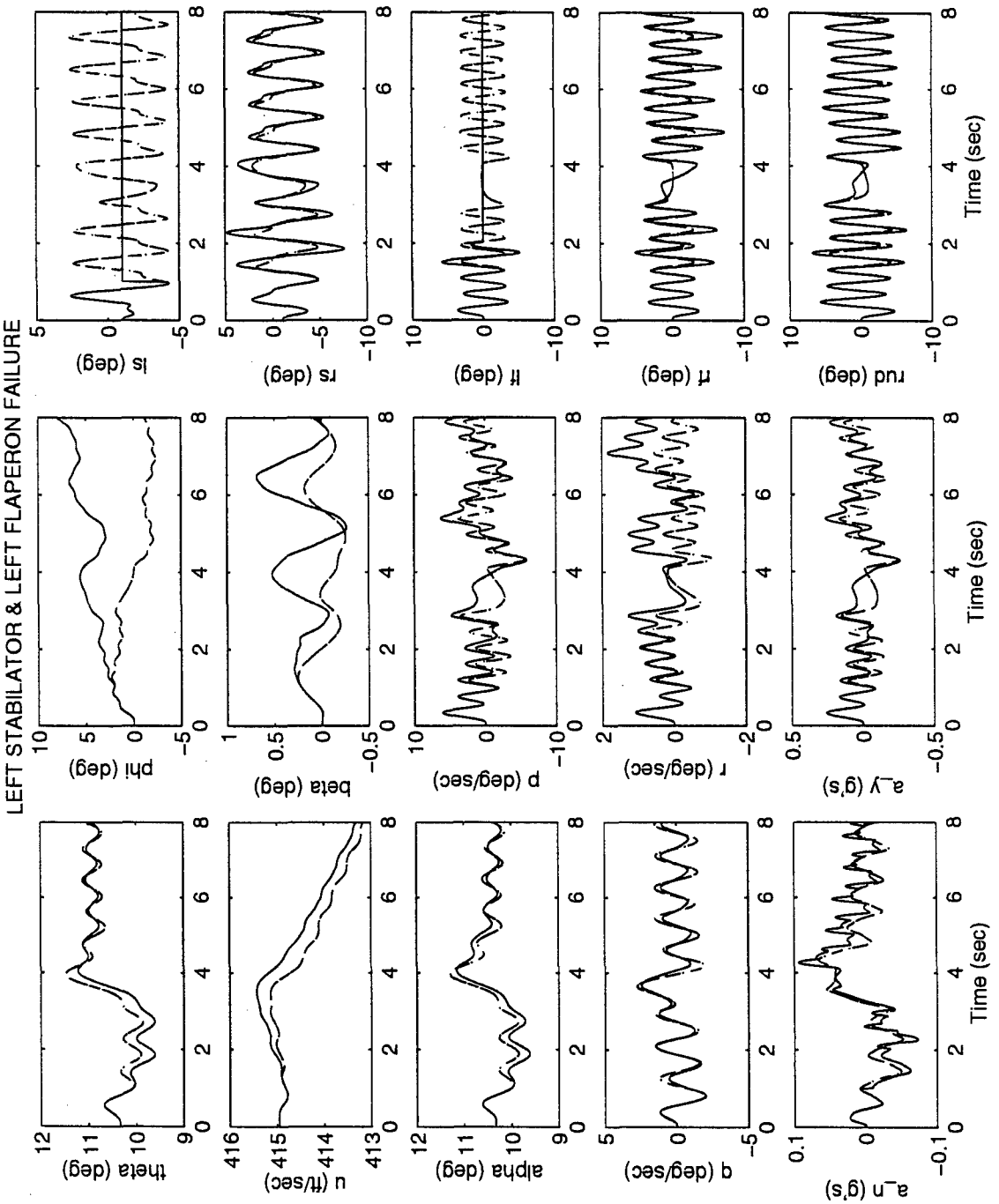

Figure D-12 MMAE-Based - Pitch Doublet - With Control Redistribution

Left Stabilator & Left Flaperon Failure

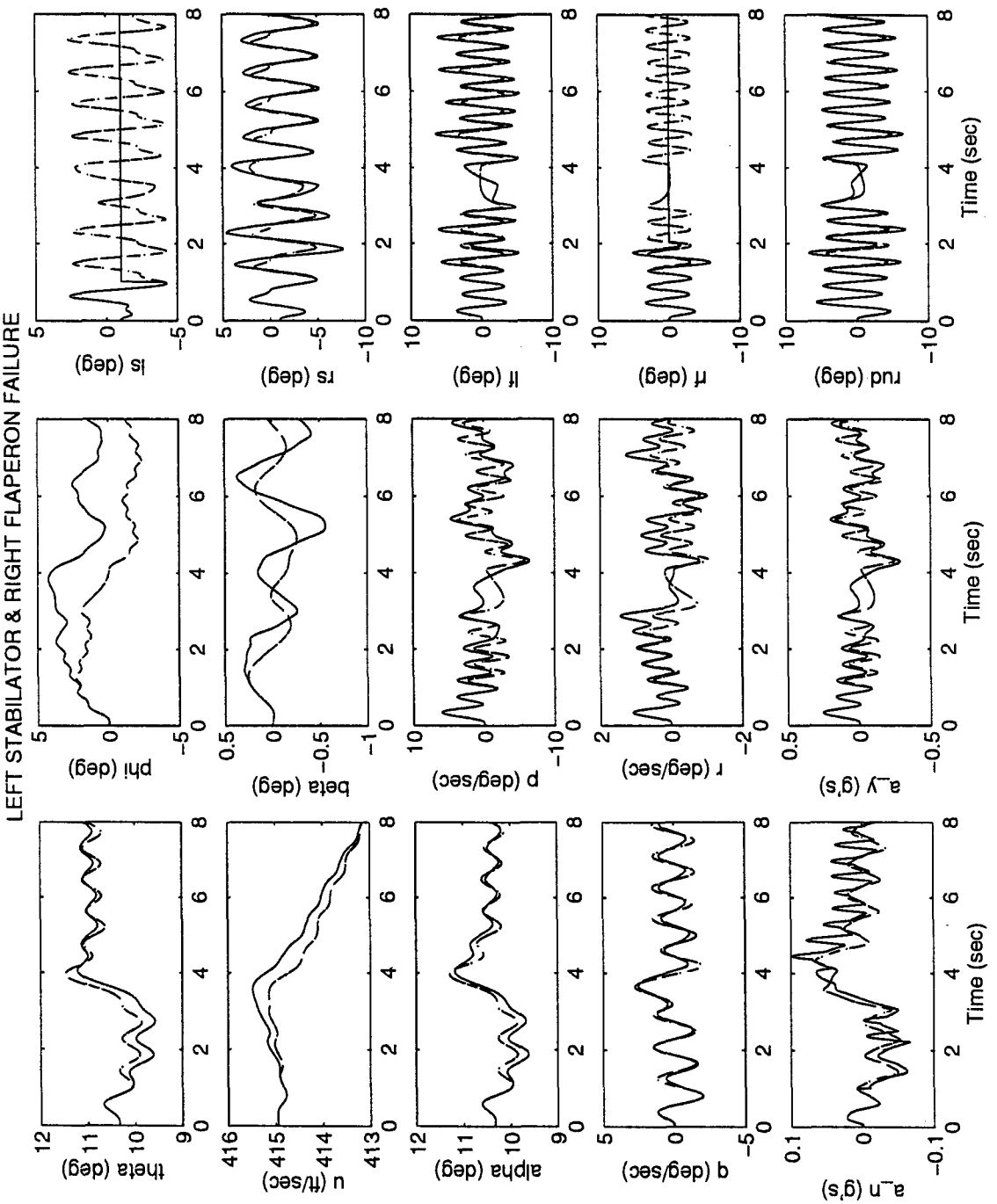

Figure D-13 MMAE-Based - Pitch Doublet - With Control Redistribution
Left Stabilator & Right Flaperon Failure

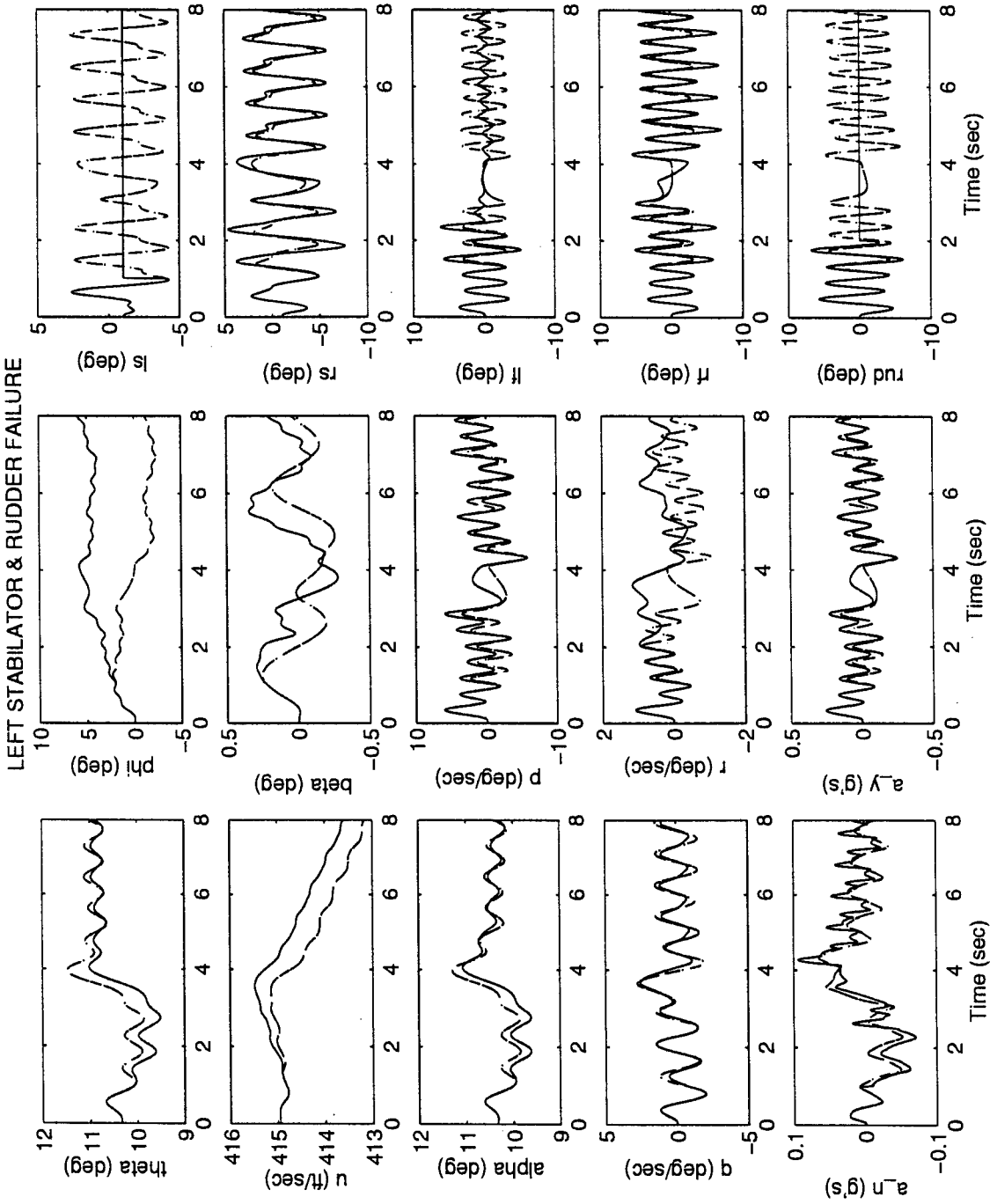

Figure D-14 MMAE-Based - Pitch Doublet - With Control Redistribution
Left Stabilator & Rudder Failure

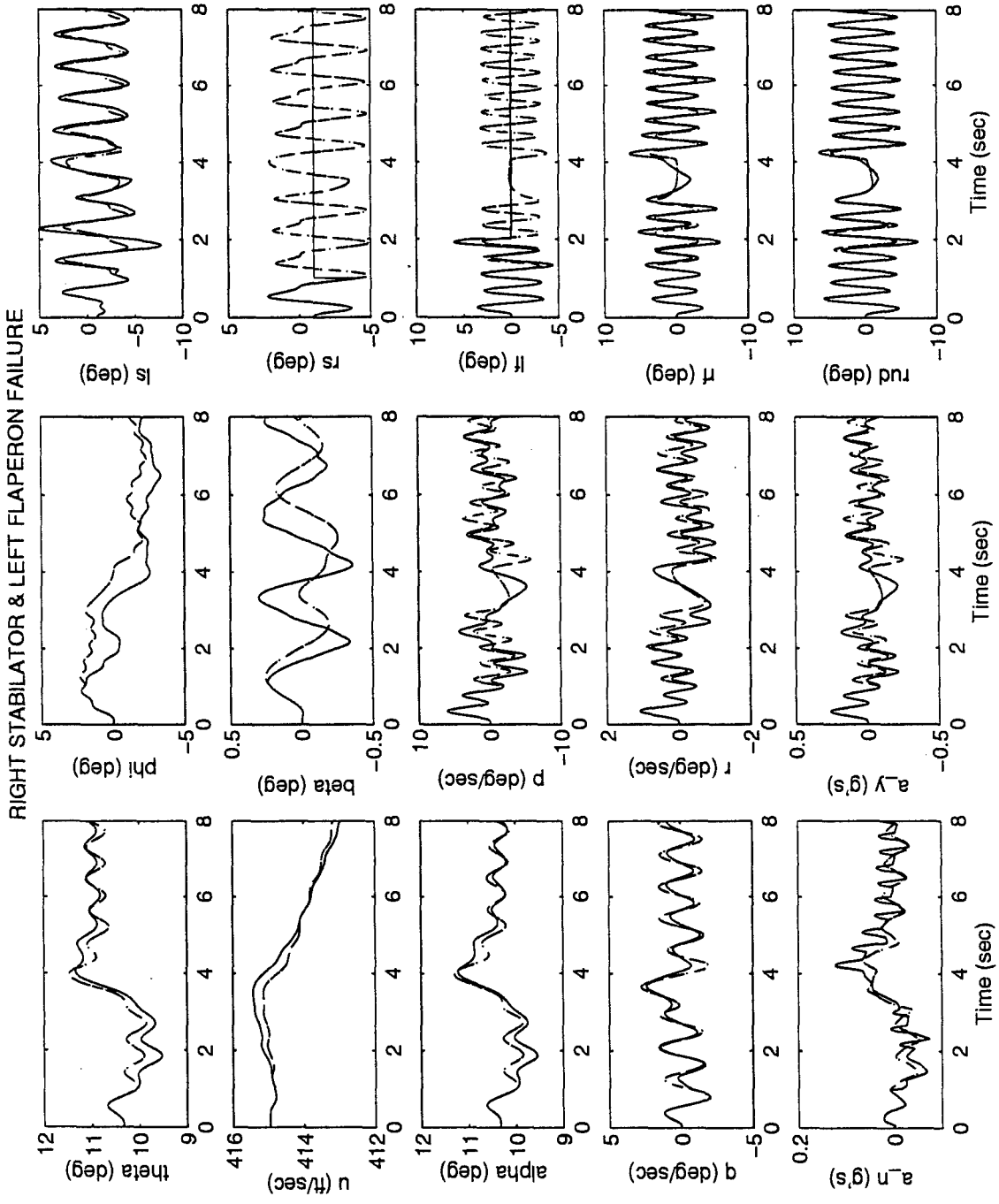

Figure D-15 MMAE-Based - Pitch Doublet - With Control Redistribution
Right Stabilator & Left Flaperon Failure

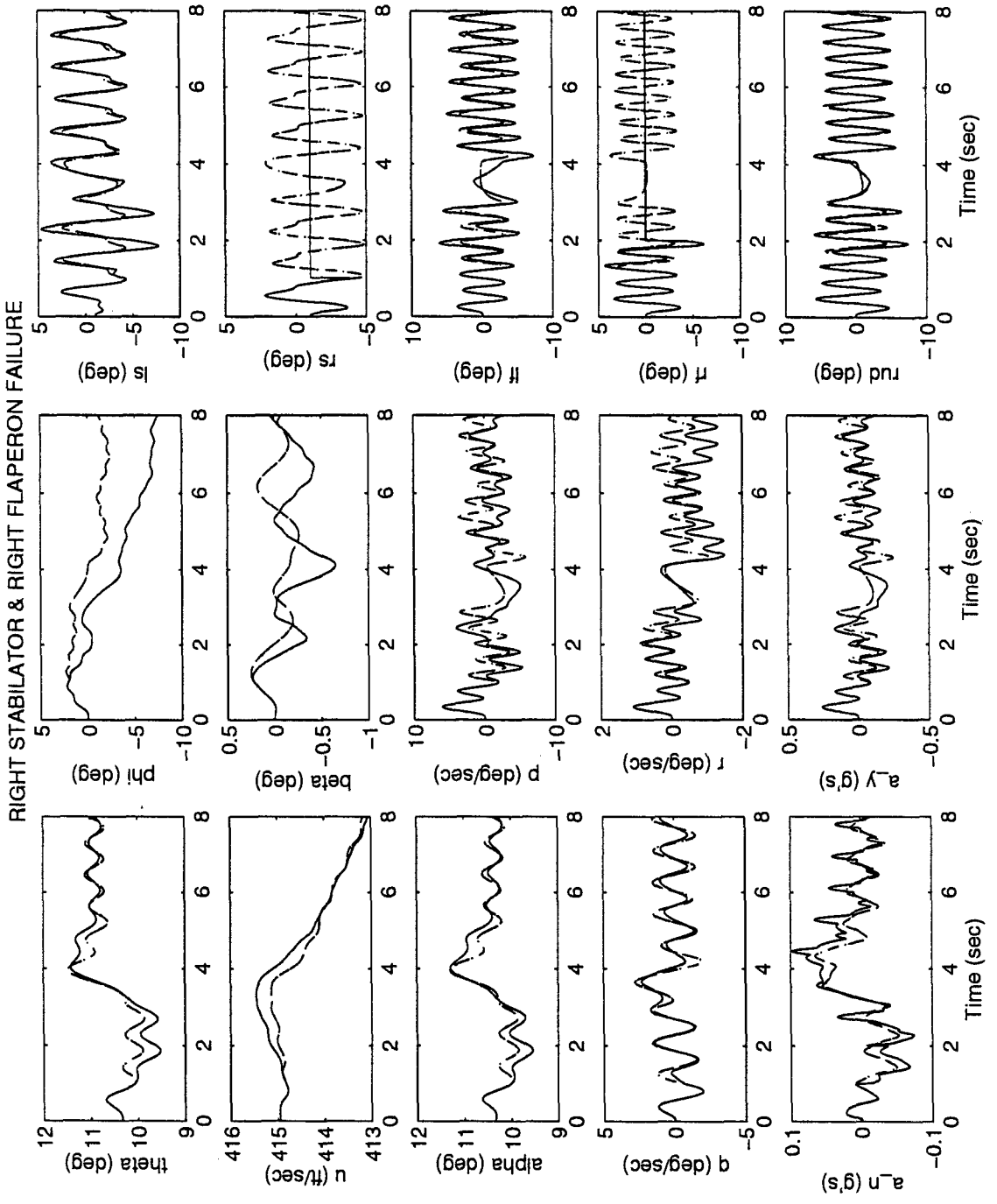

Figure D-16 MMAE-Based - Pitch Doublet - With Control Redistribution
Right Stabilator & Right Flaperon Failure

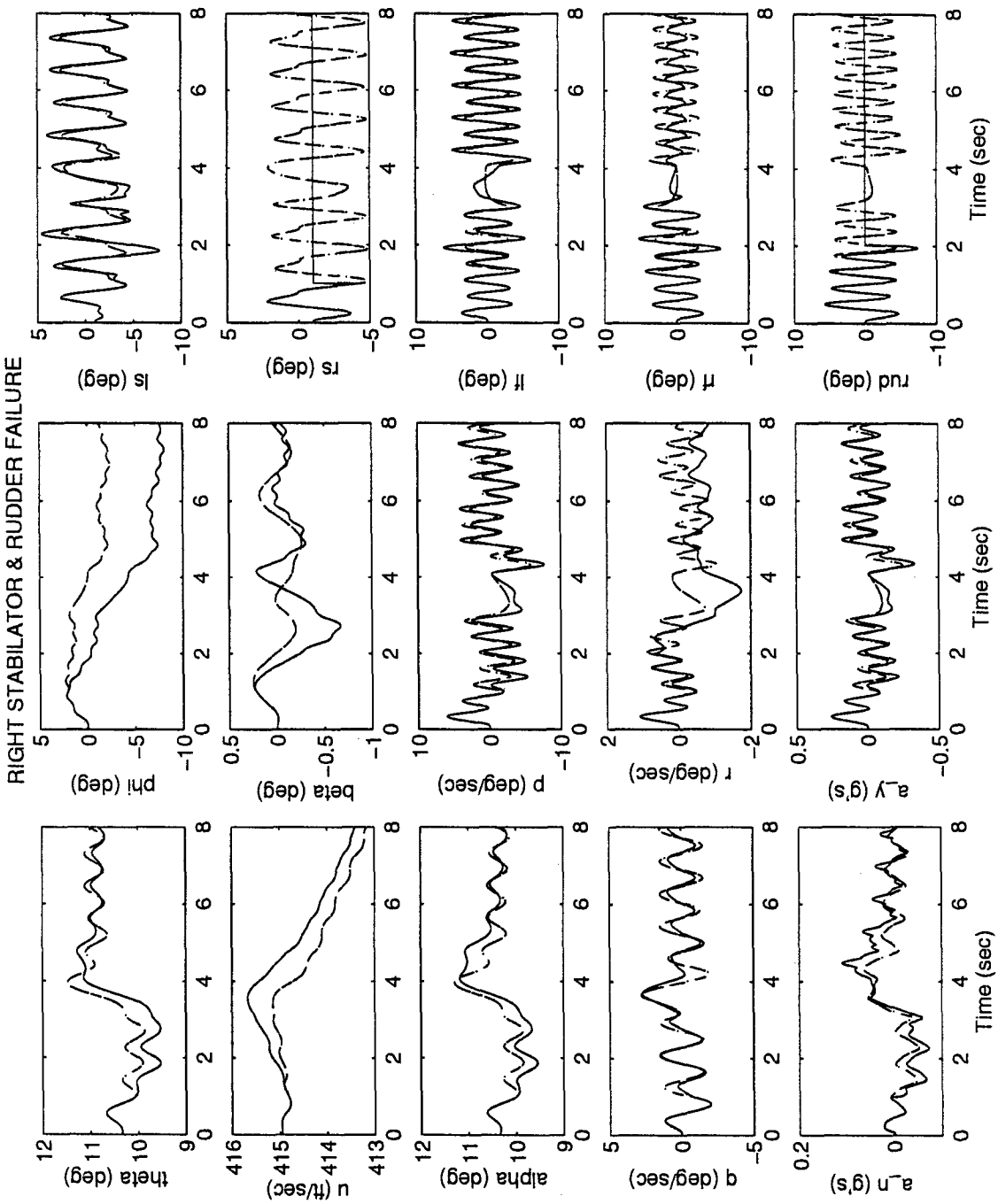

Figure D-17 MMAE-Based - Pitch Doublet - With Control Redistribution
Right Stabilator & Rudder Failure

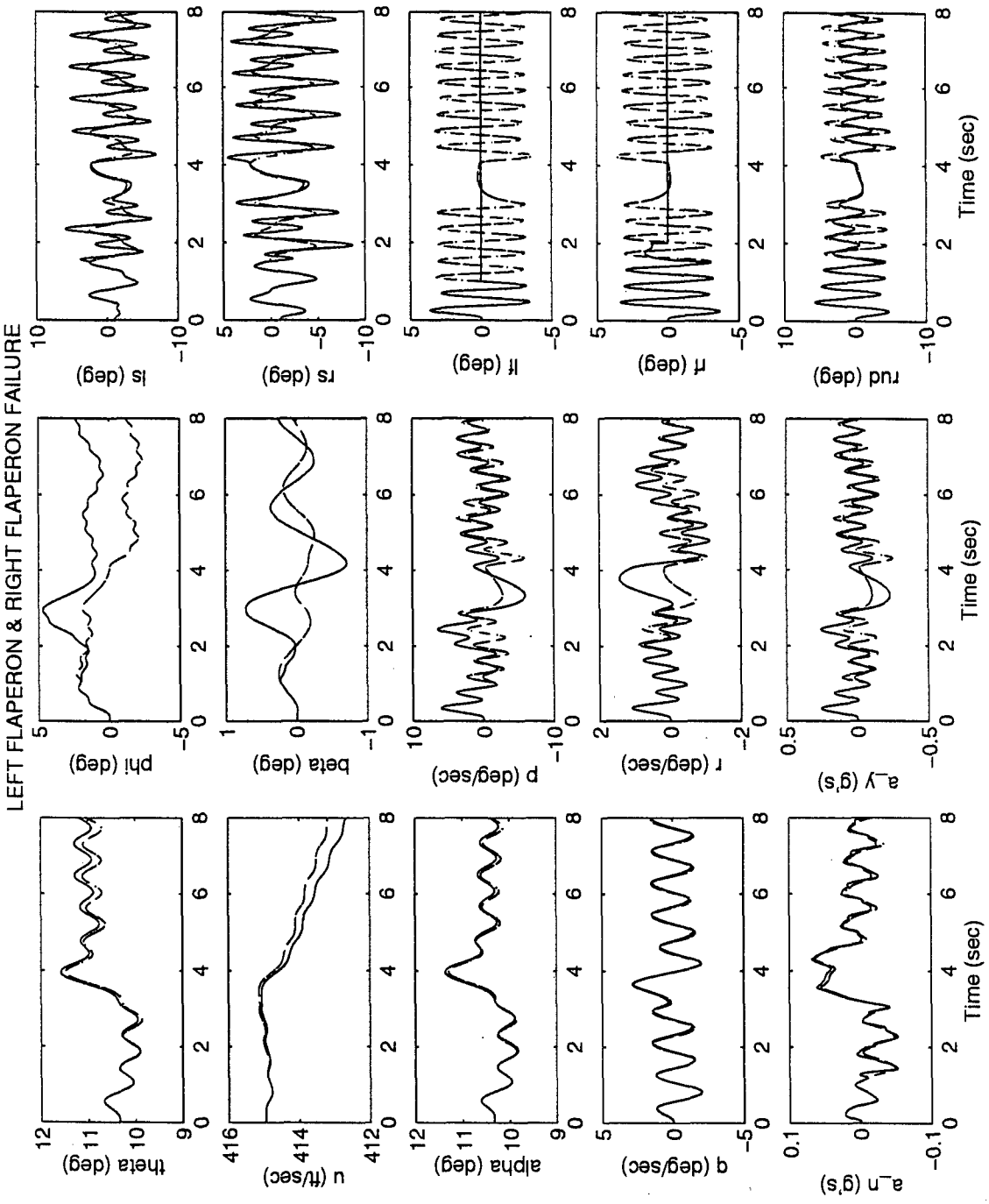

Figure D-18 MMAE-Based - Pitch Doublet - With Control Redistribution
Left Flaperon & Right Flaperon Failure

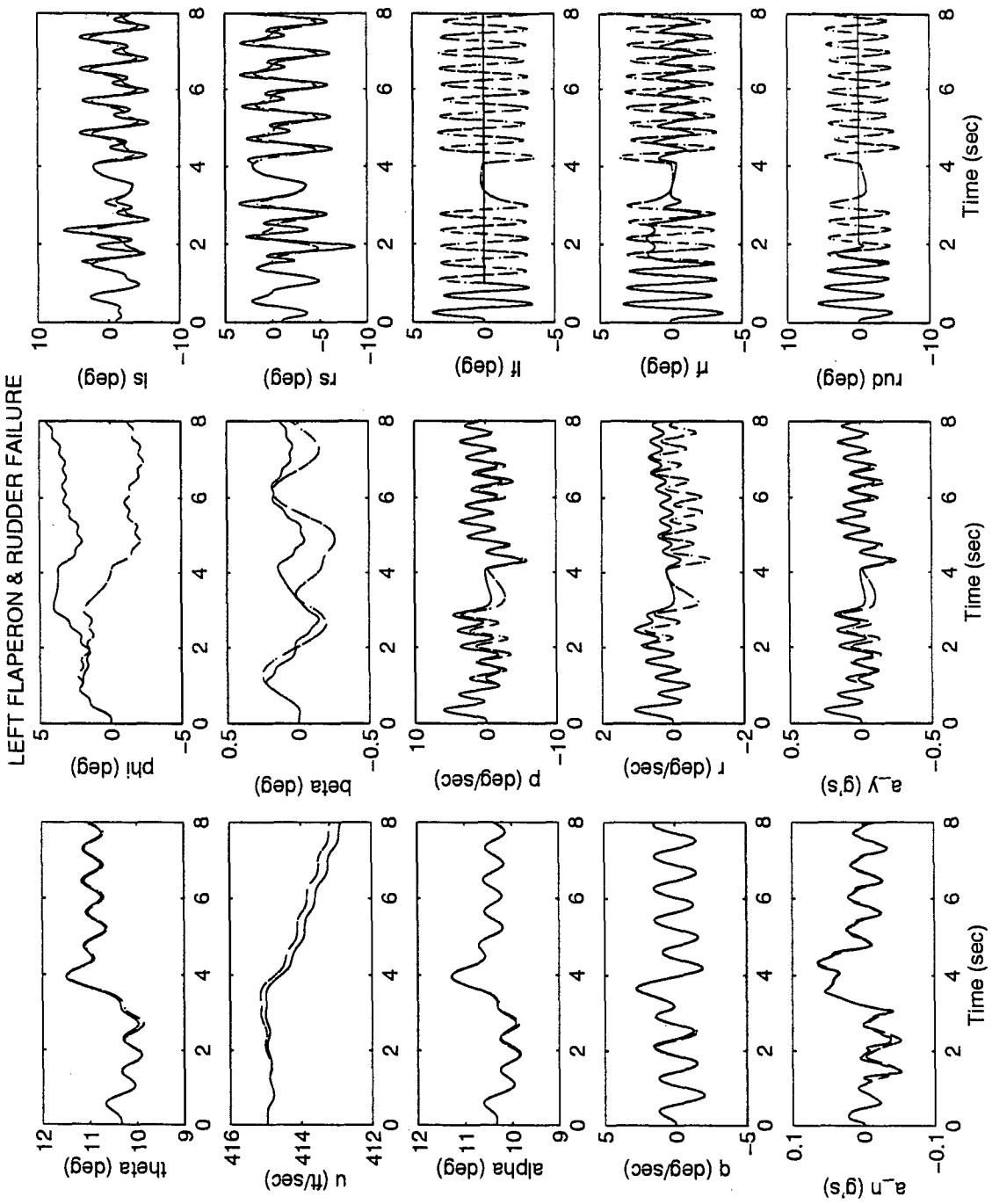

Figure D-19 MMAE-Based - Pitch Doublet - With Control Redistribution
Left Flaperon & Rudder Failure

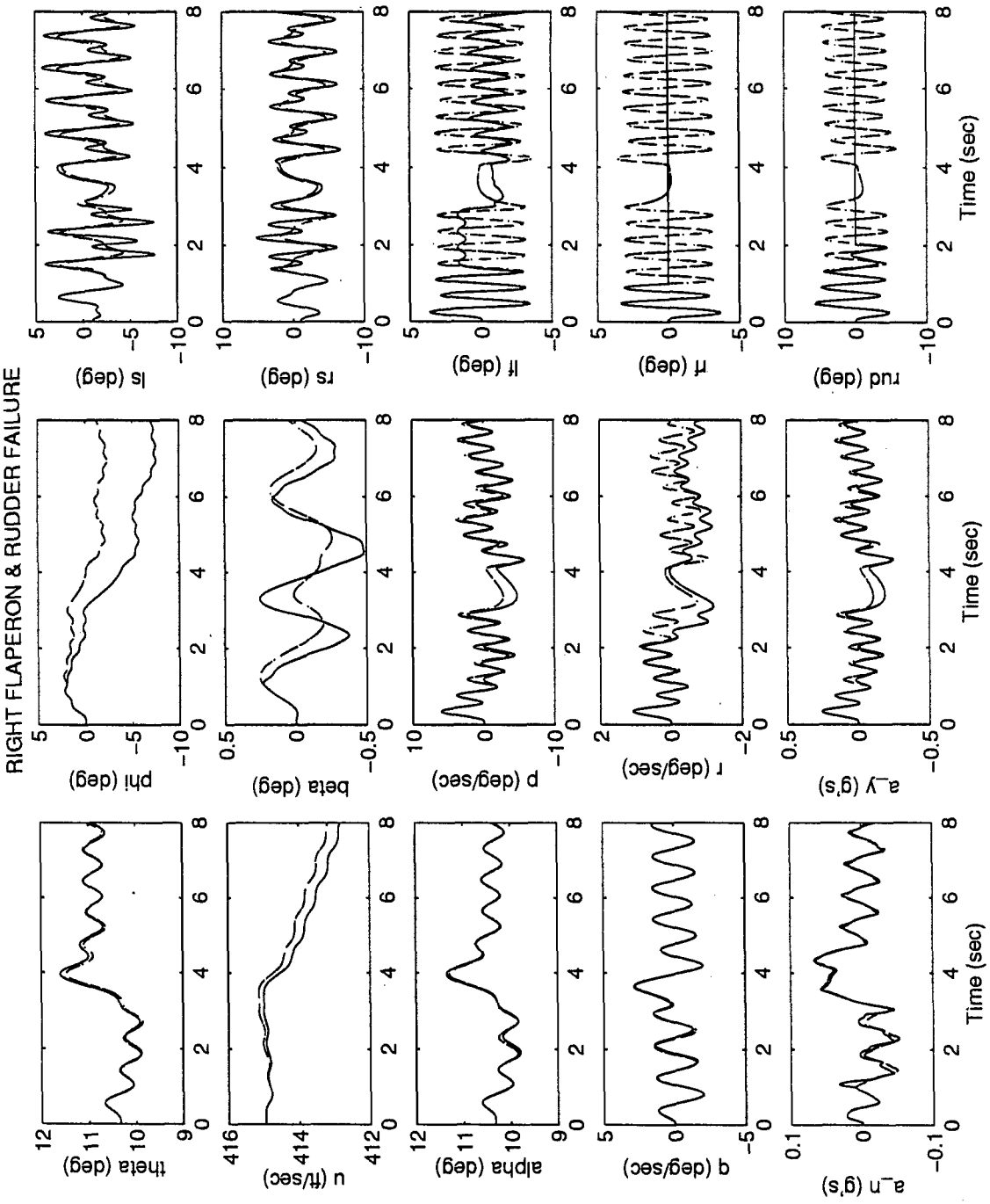

Figure D-20 MMAE-Based - Pitch Doublet - With Control Redistribution
Right Flaperon & Rudder Failure

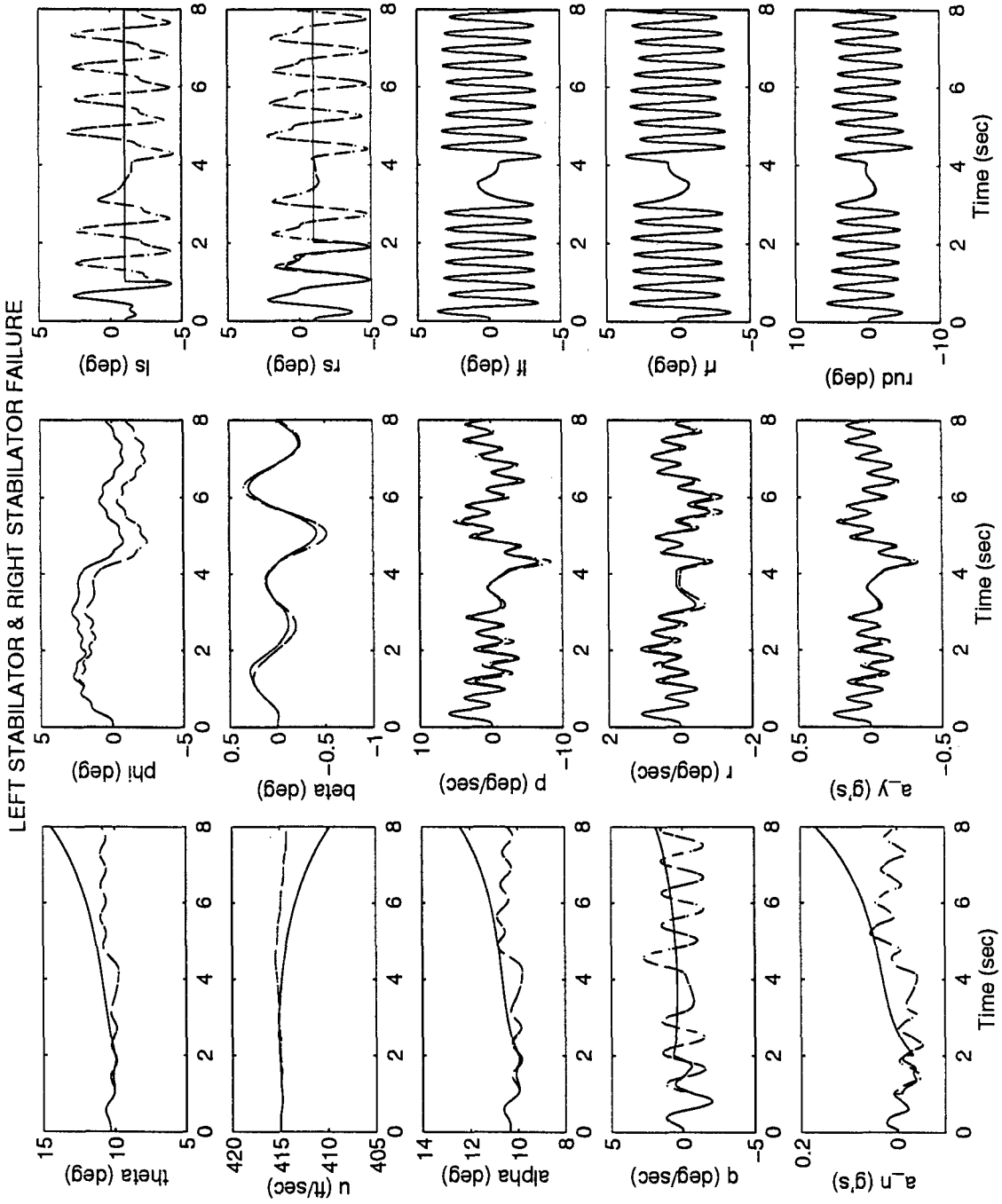

Figure D-21 MMAE-Based - Roll Doublet - No Control Redistribution
Left Stabilator & Right Stabilator Failure

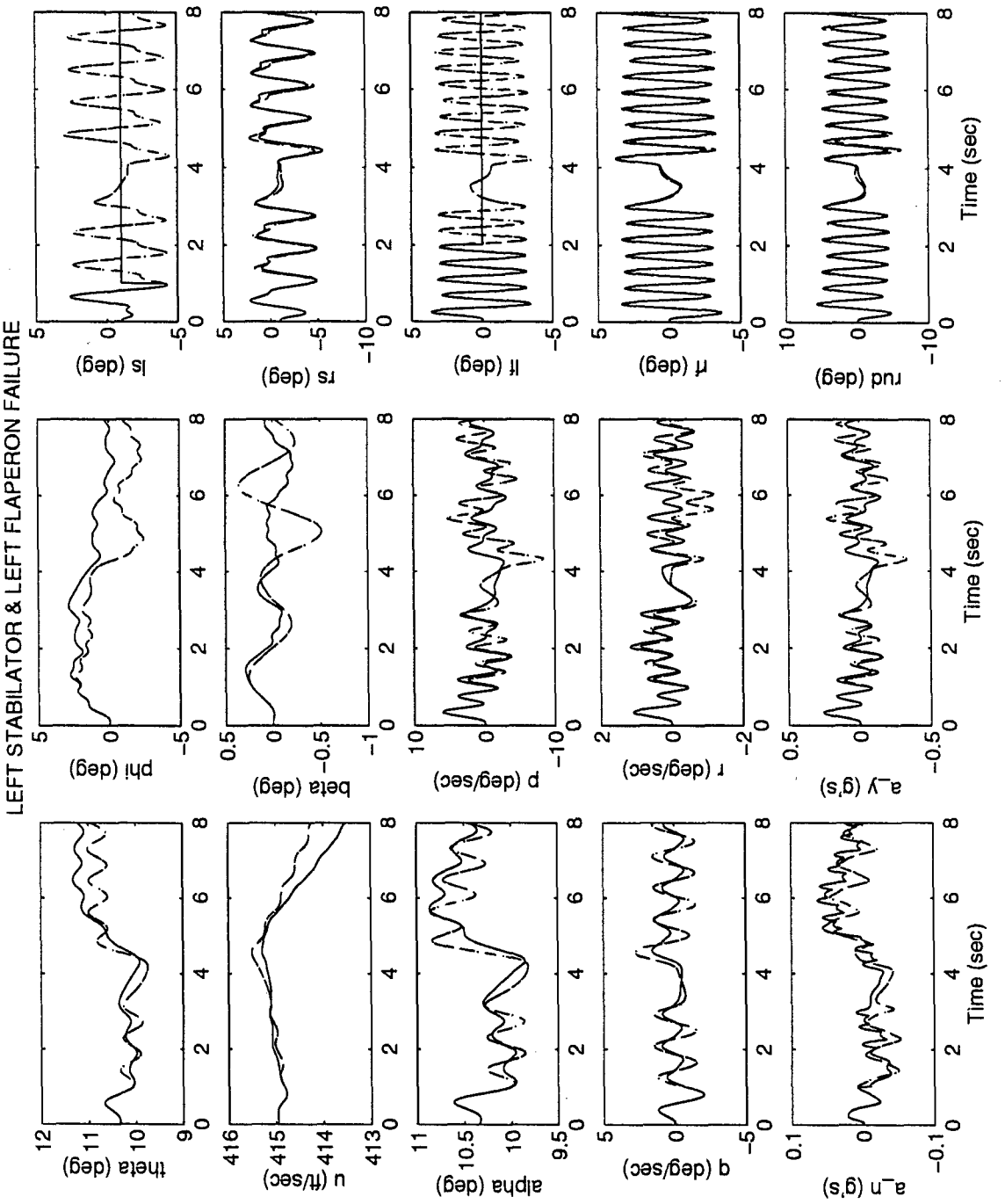

Figure D-22 MMAE-Based - Roll Doublet - No Control Redistribution

Left Stabilator & Left Flaperon Failure

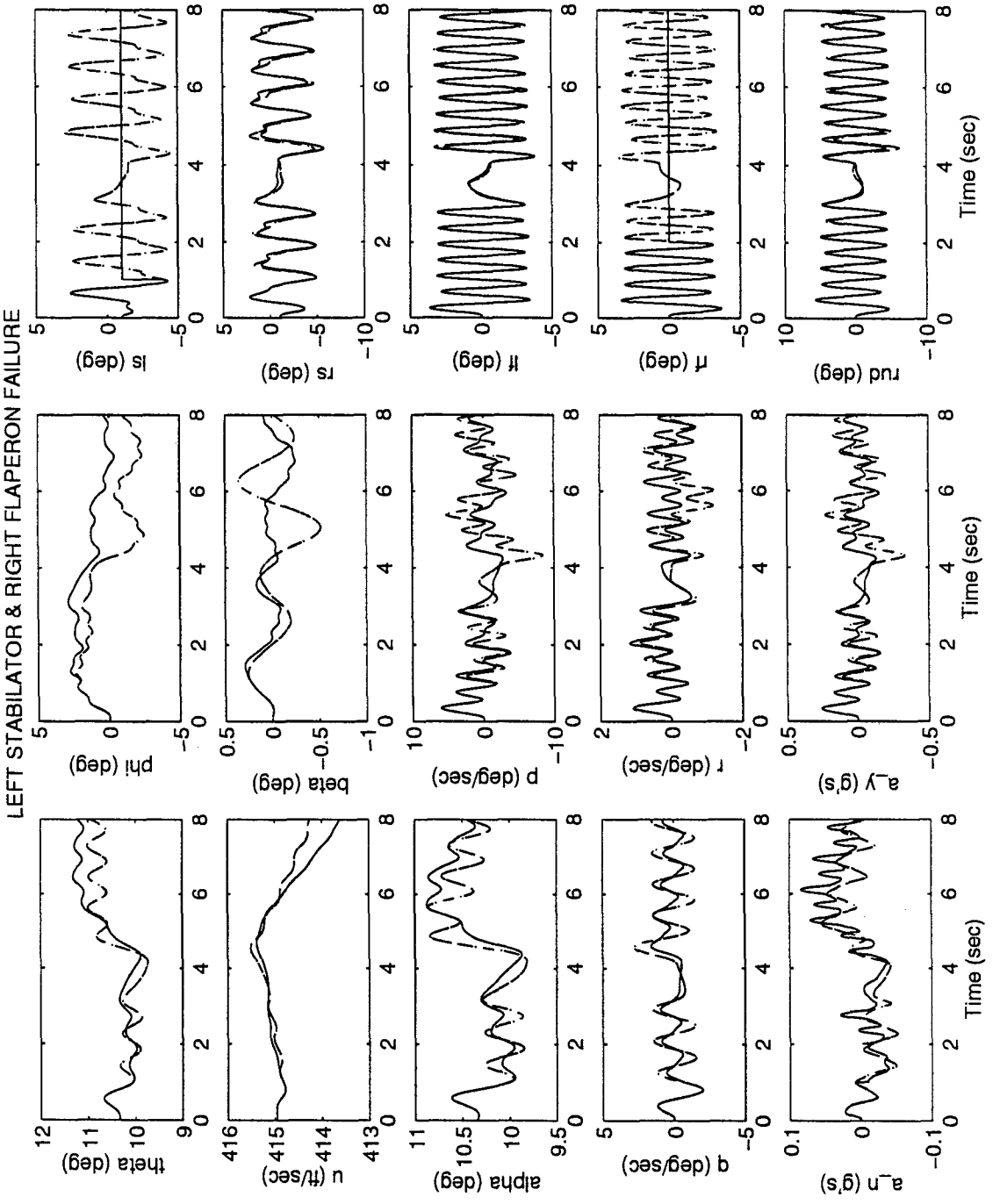

Figure D-23 MMAE-Based - Roll Doublet - No Control Redistribution
Left Stabilator & Right Flaperon Failure

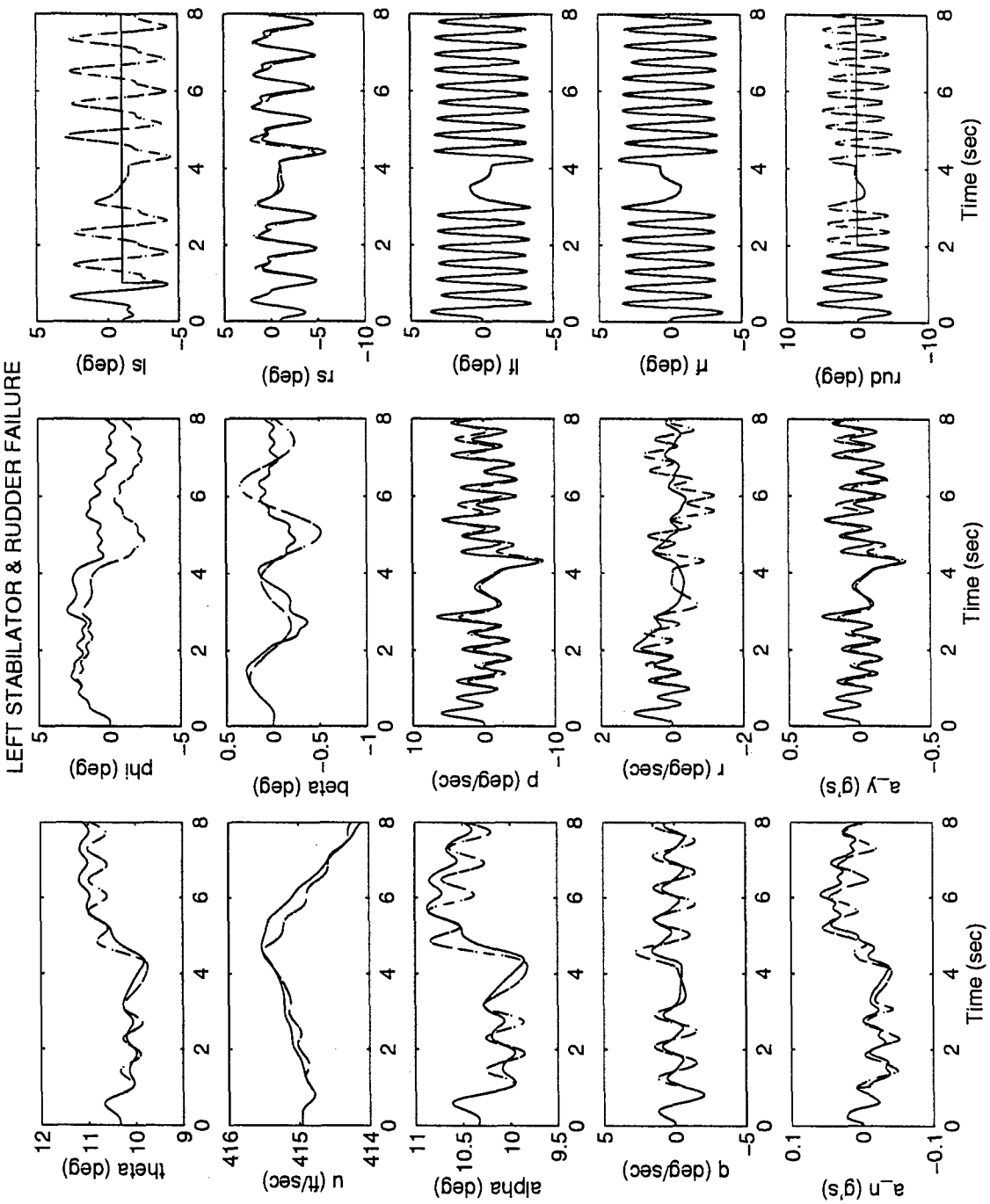

Figure D-24 MMAE-Based - Roll Doublet - No Control Redistribution
Left Stabilator & Rudder Failure

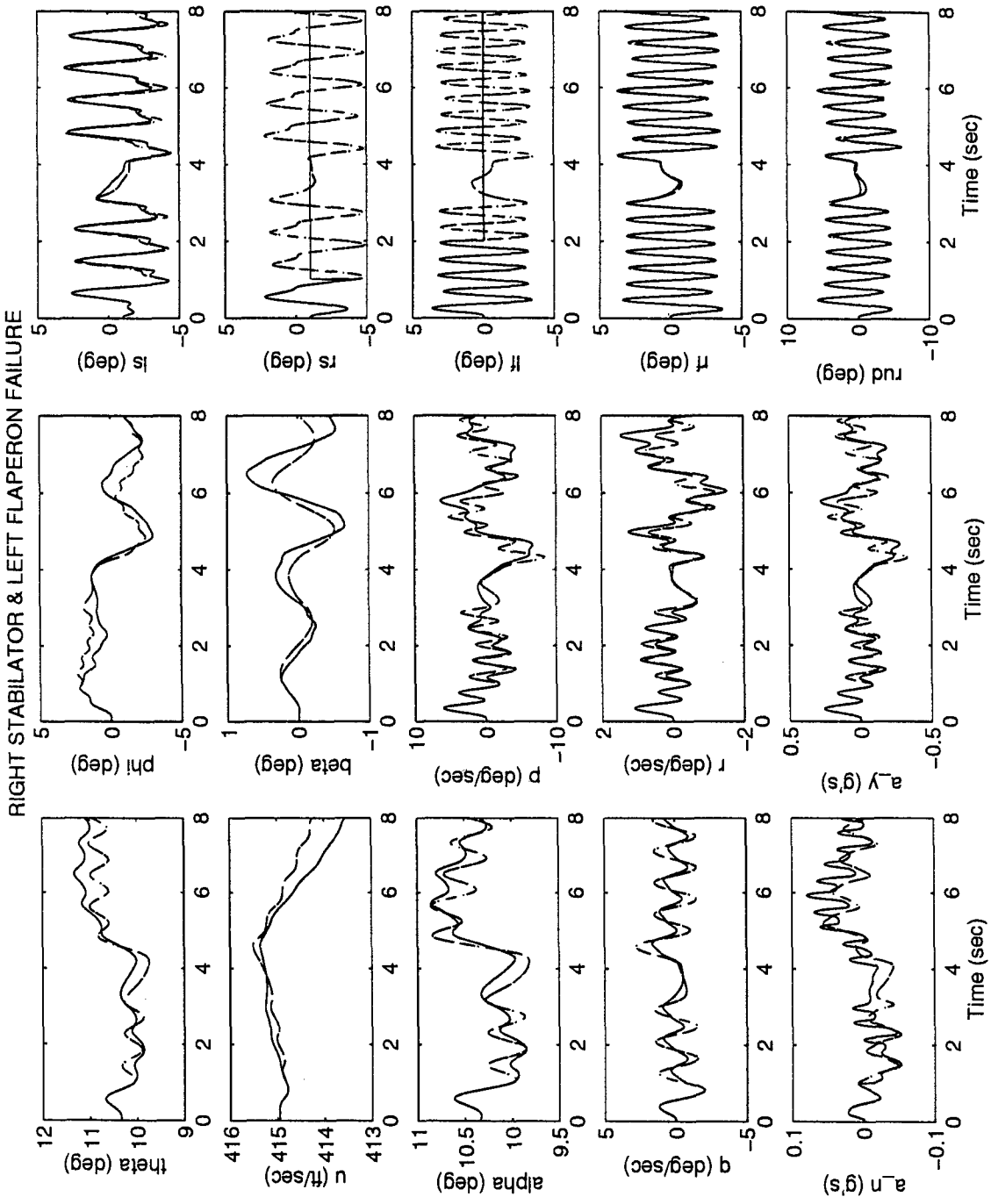

Figure D-25 MMAE-Based - Roll Doublet - No Control Redistribution
Right Stabilator & Left Flaperon Failure

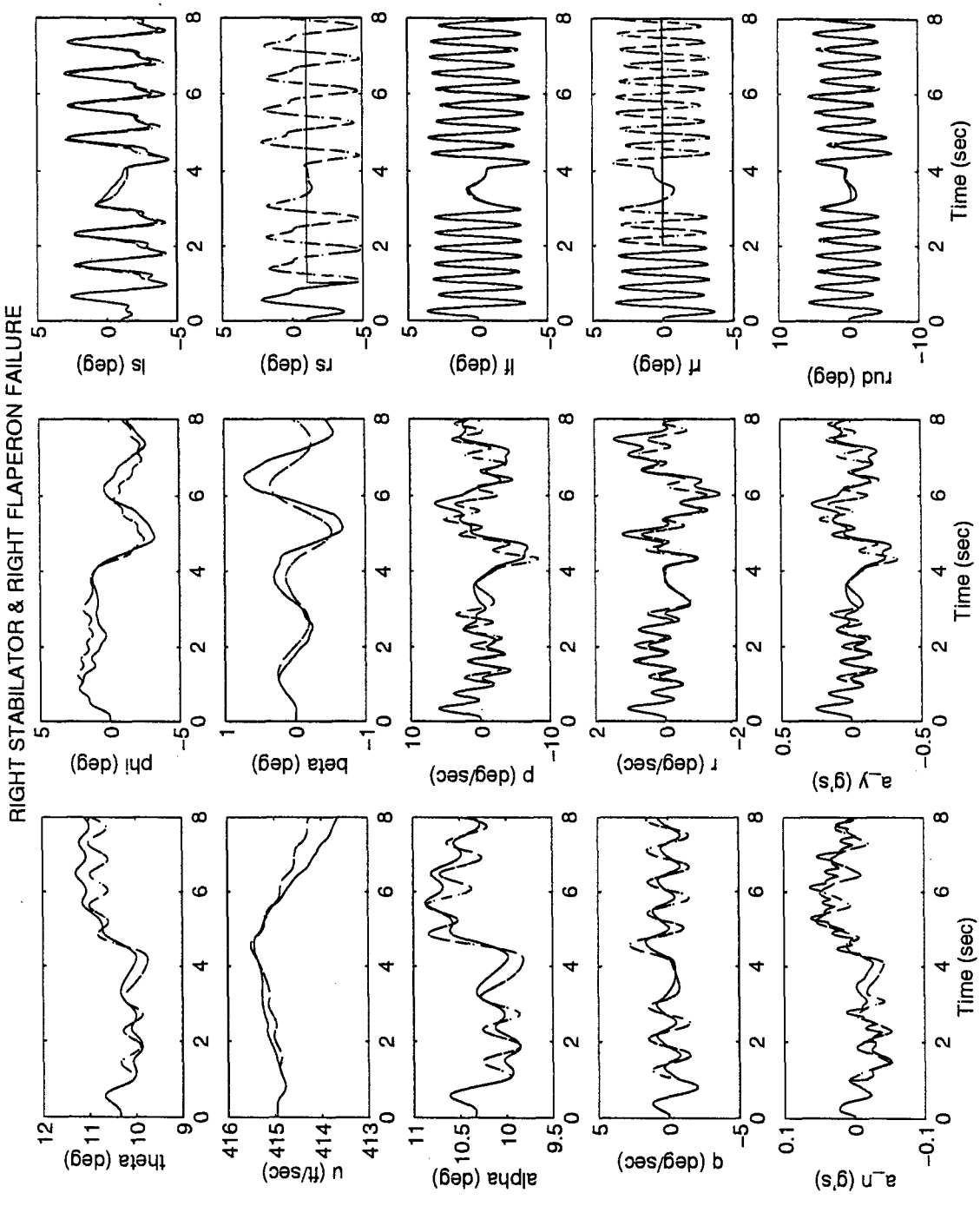

Figure D-26 MMAE-Based - Roll Doublet - No Control Redistribution
Right Stabilator & Right Flaperon Failure

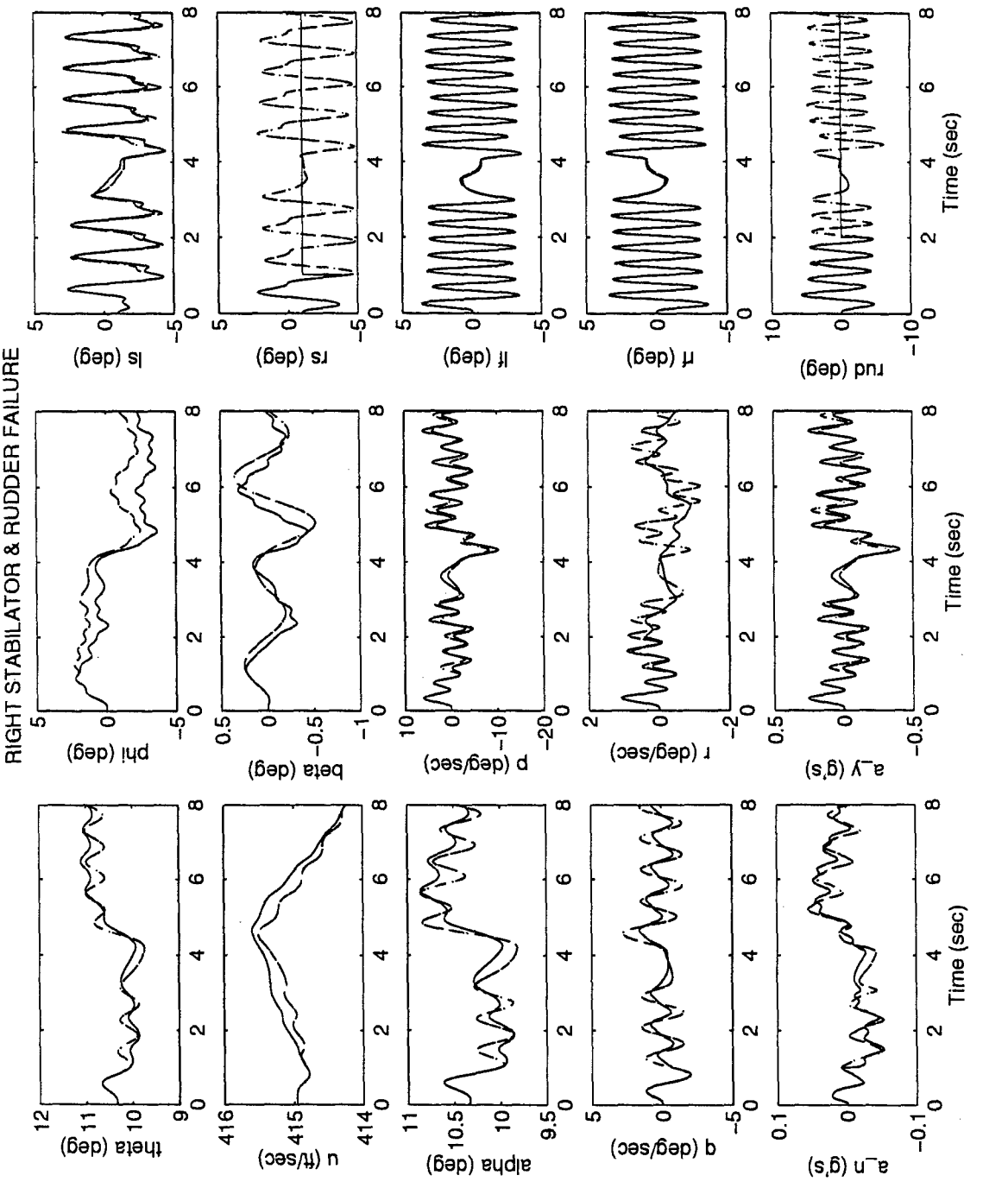

Figure D-27 MMAE-Based - Roll Doublet - No Control Redistribution
Right Stabilator & Rudder Failure

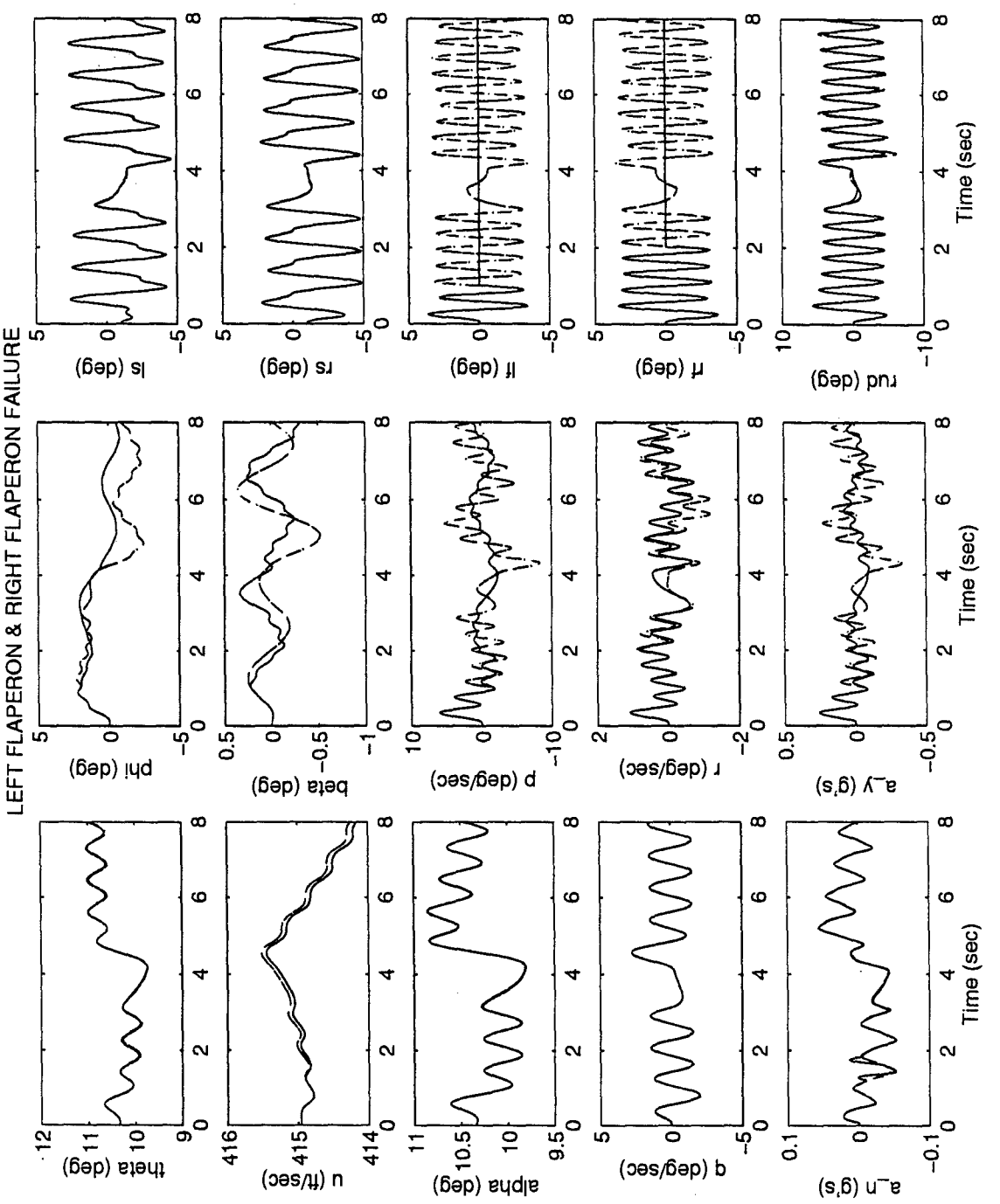

Figure D-28 MMAE-Based - Roll Doublet - No Control Redistribution
Left Flaperon & Right Flaperon Failure

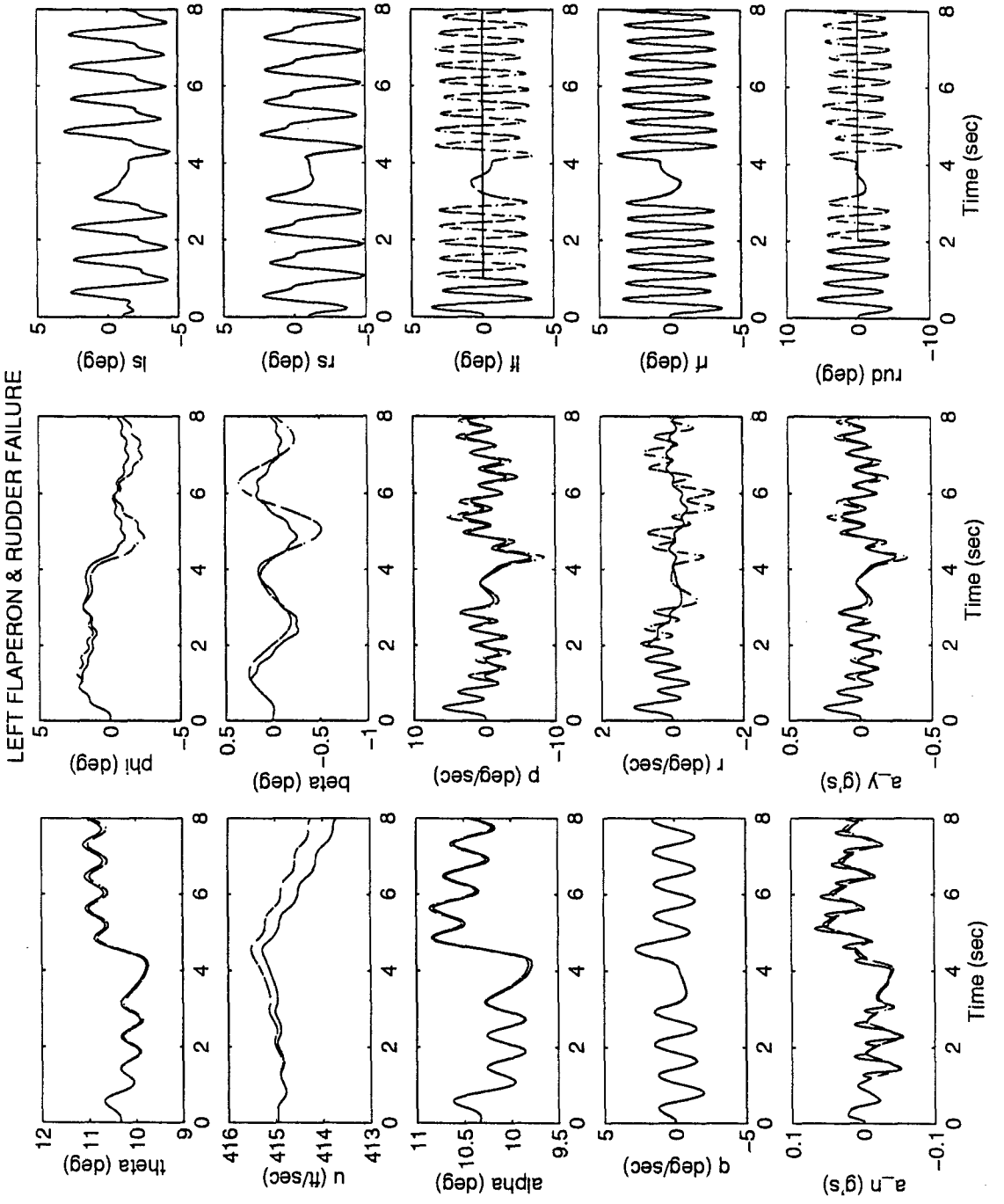

Figure D-29 MMAE-Based - Roll Doublet - No Control Redistribution
Left Flaperon & Rudder Failure

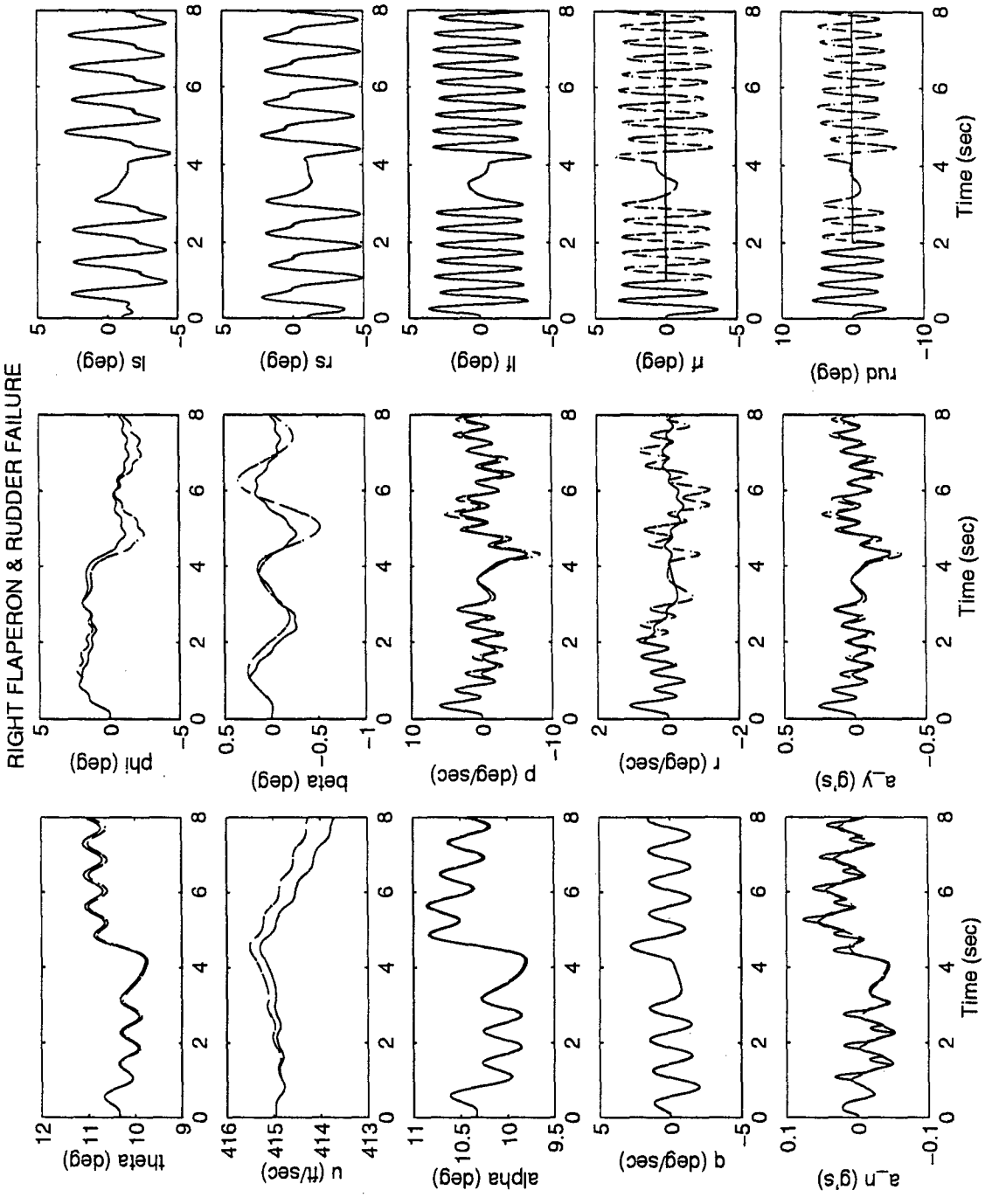

Figure D-30 MMAE-Based - Roll Doublet - No Control Redistribution
Right Flaperon & Rudder Failure

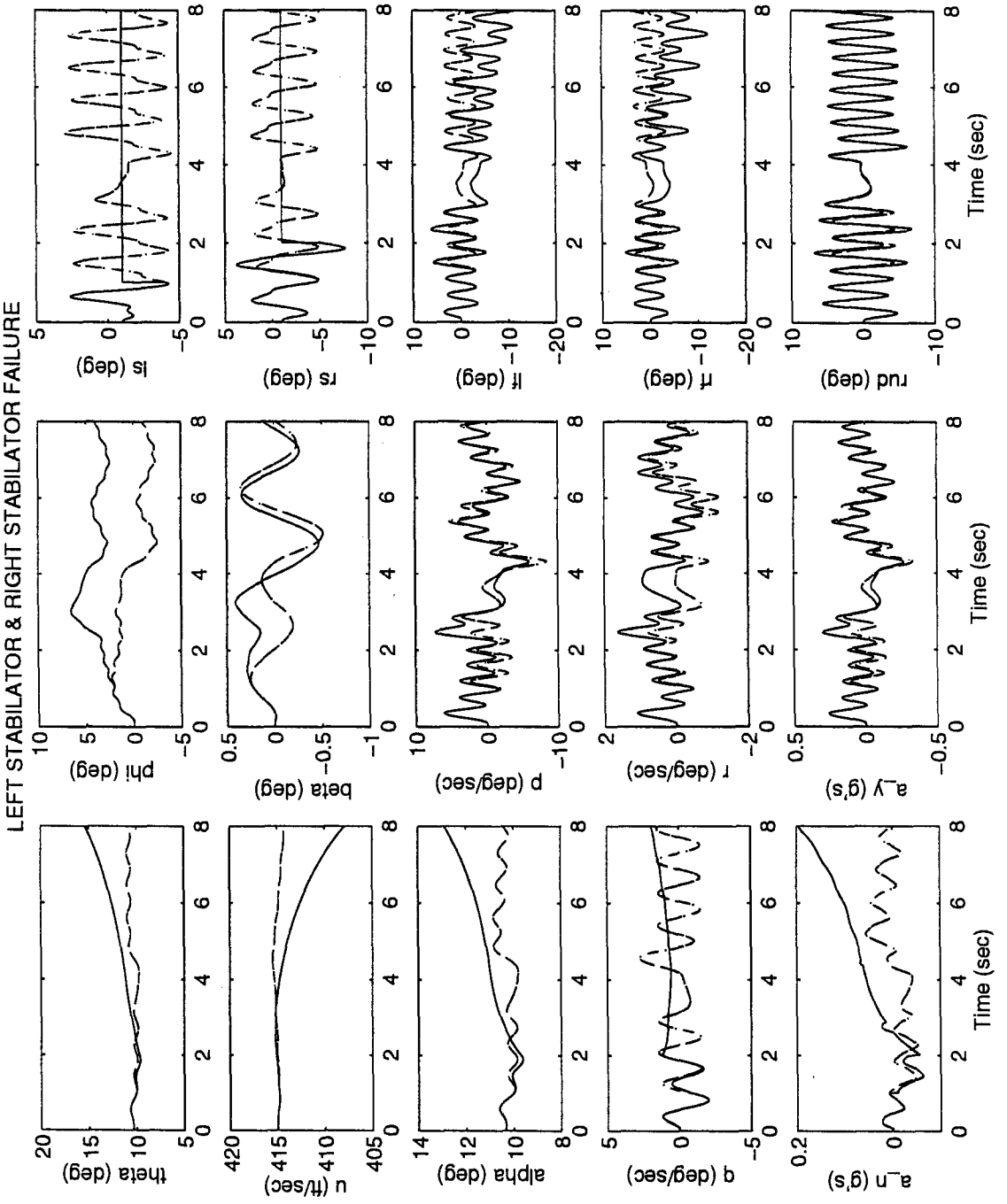

Figure D-31 MMAE-Based - Roll Doublet - With Control Redistribution
Left Stabilator & Right Stabilator Failure

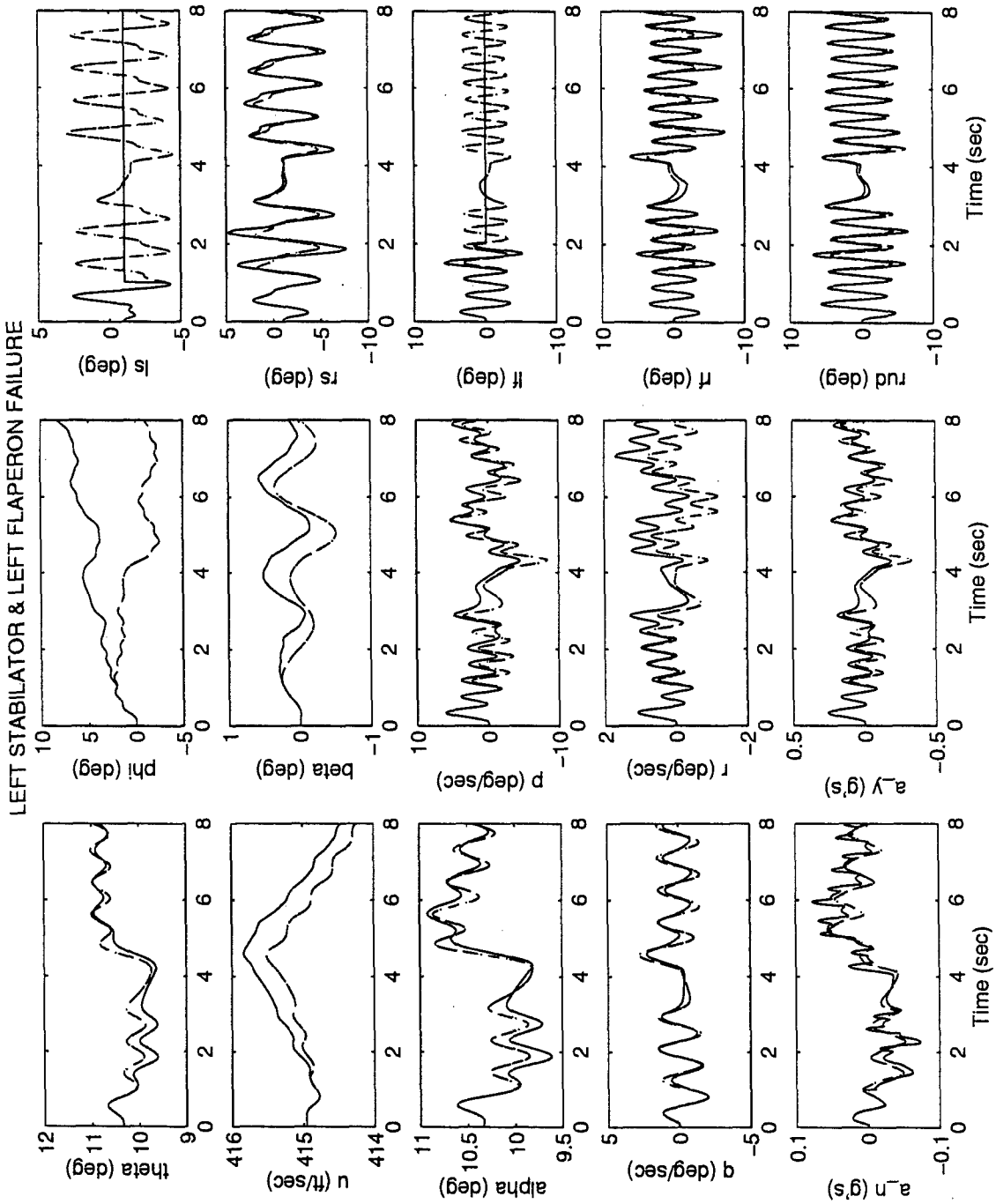

Figure D-32 MMAE-Based - Roll Doublet - With Control Redistribution

Left Stabilator & Left Flaperon Failure

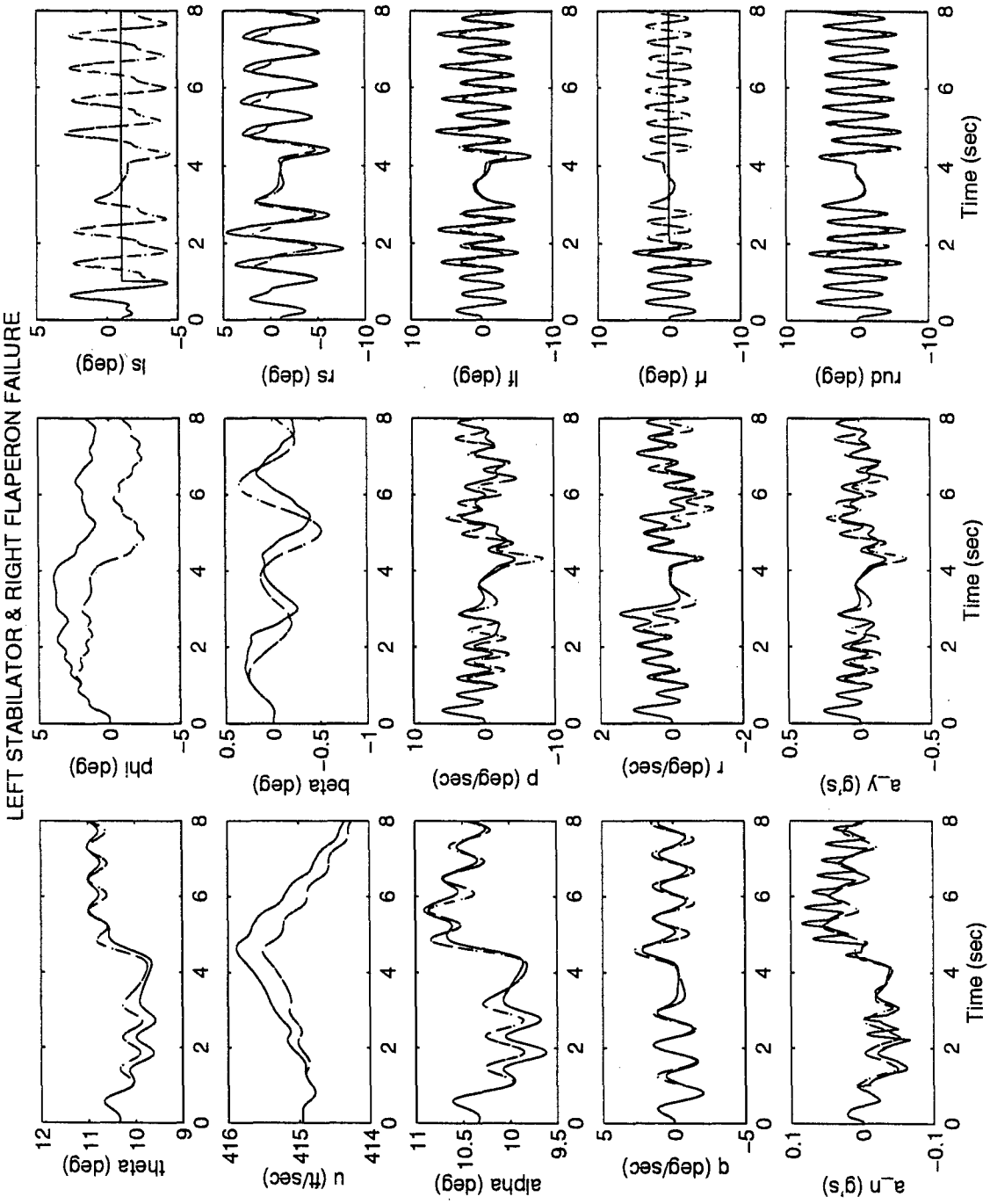

Figure D-33 MMAE-Based - Roll Doublet - With Control Redistribution
Left Stabilator & Right Flaperon Failure

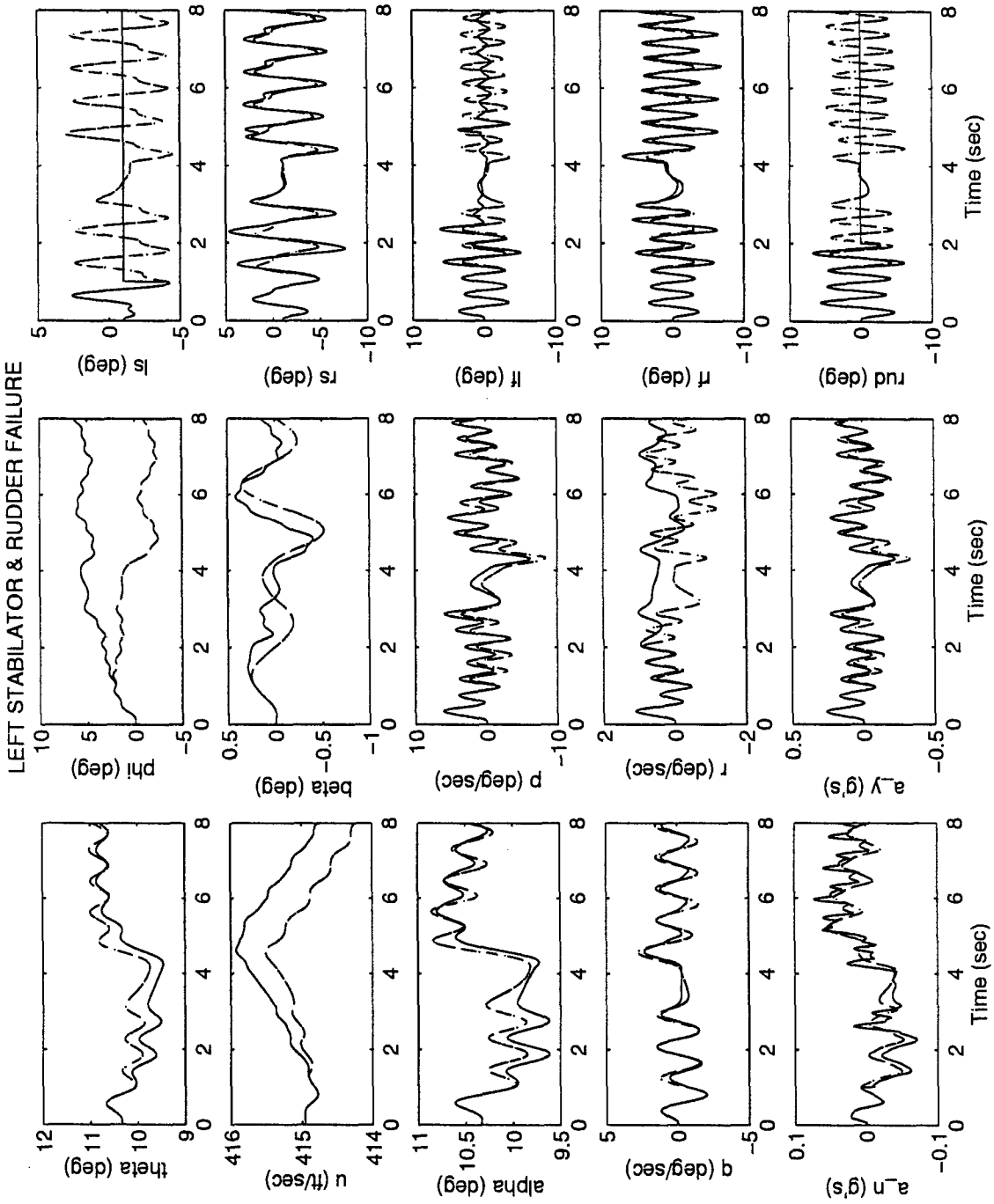

Figure D-34 MMAE-Based - Roll Doublet - With Control Redistribution
Left Stabilator & Rudder Failure

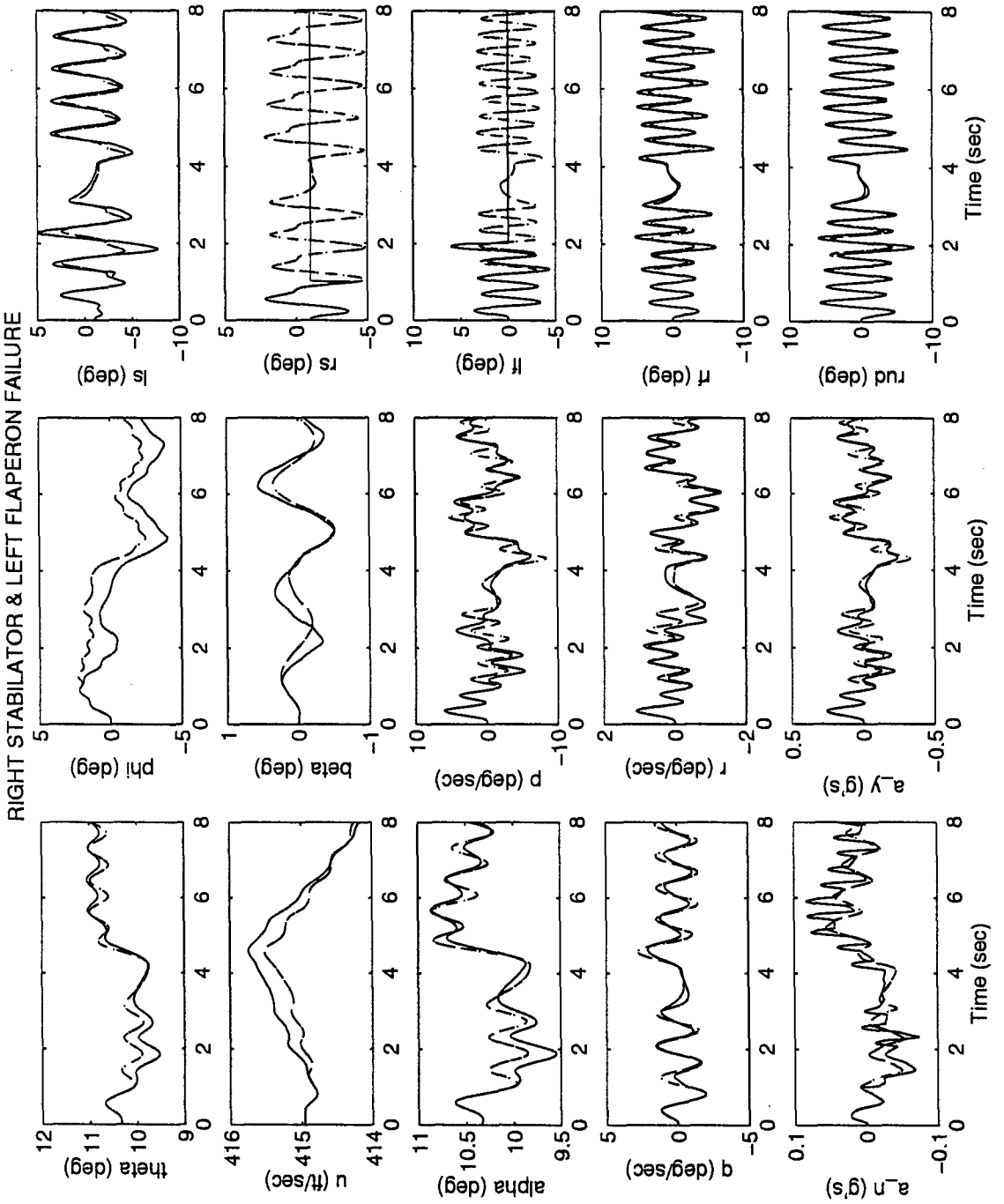

Figure D-35 MMAE-Based - Roll Doublet - With Control Redistribution
Right Stabilator & Left Flaperon Failure

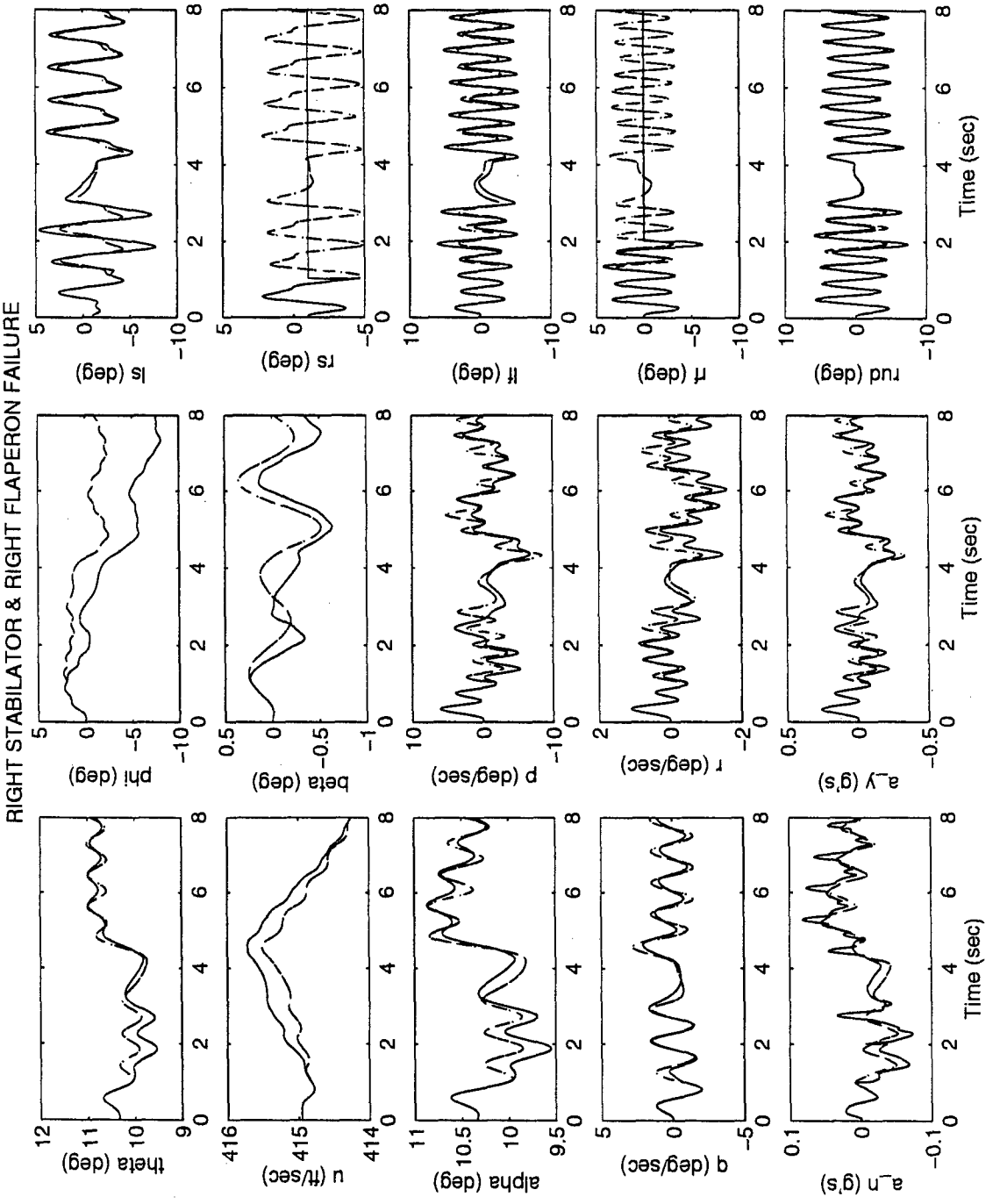

Figure D-36 MMAE-Based - Roll Doublet - With Control Redistribution
Right Stabilator & Right Flaperon Failure

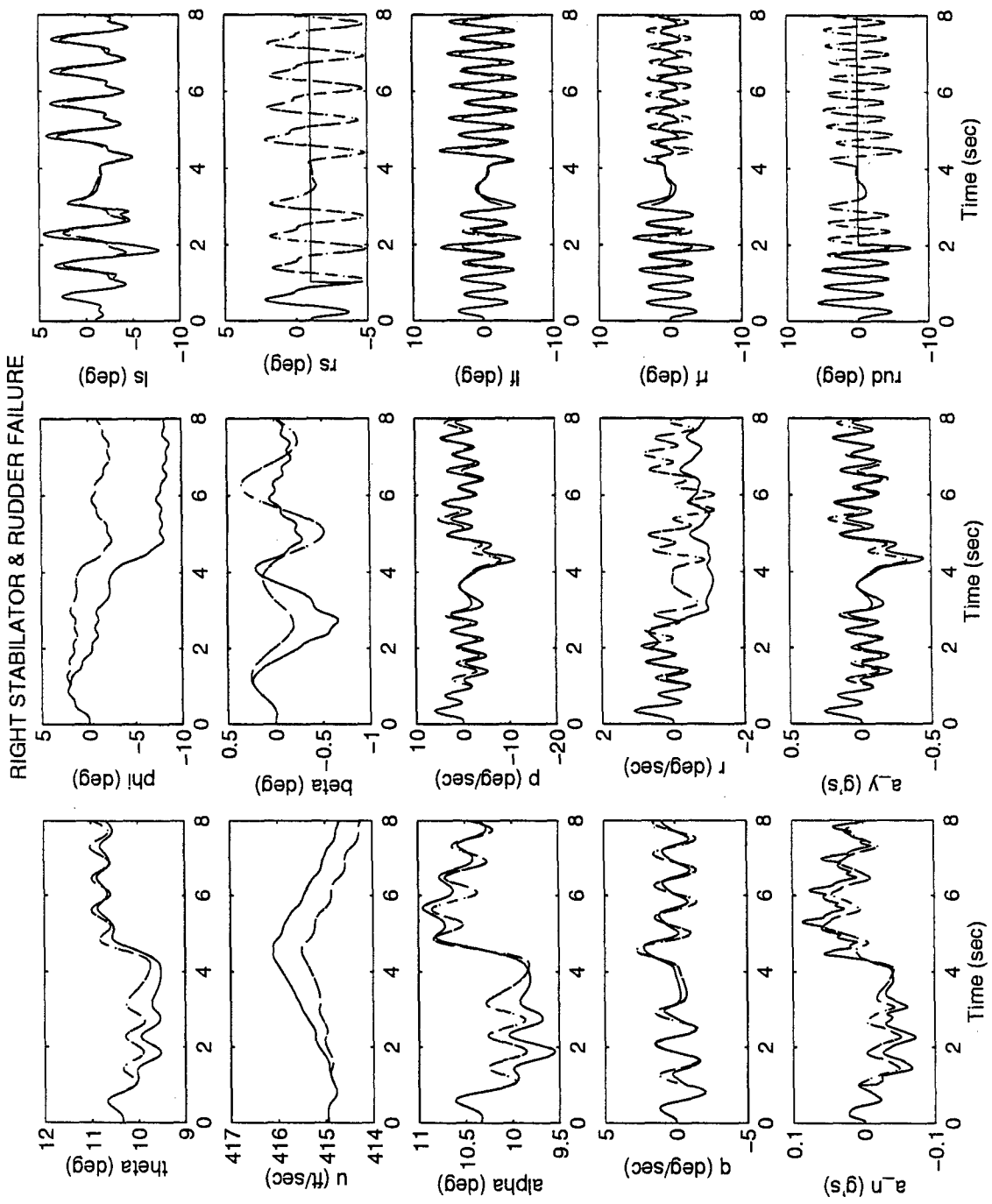

Figure D-37 MMAE-Based - Roll Doublet - With Control Redistribution
Right Stabilator & Rudder Failure

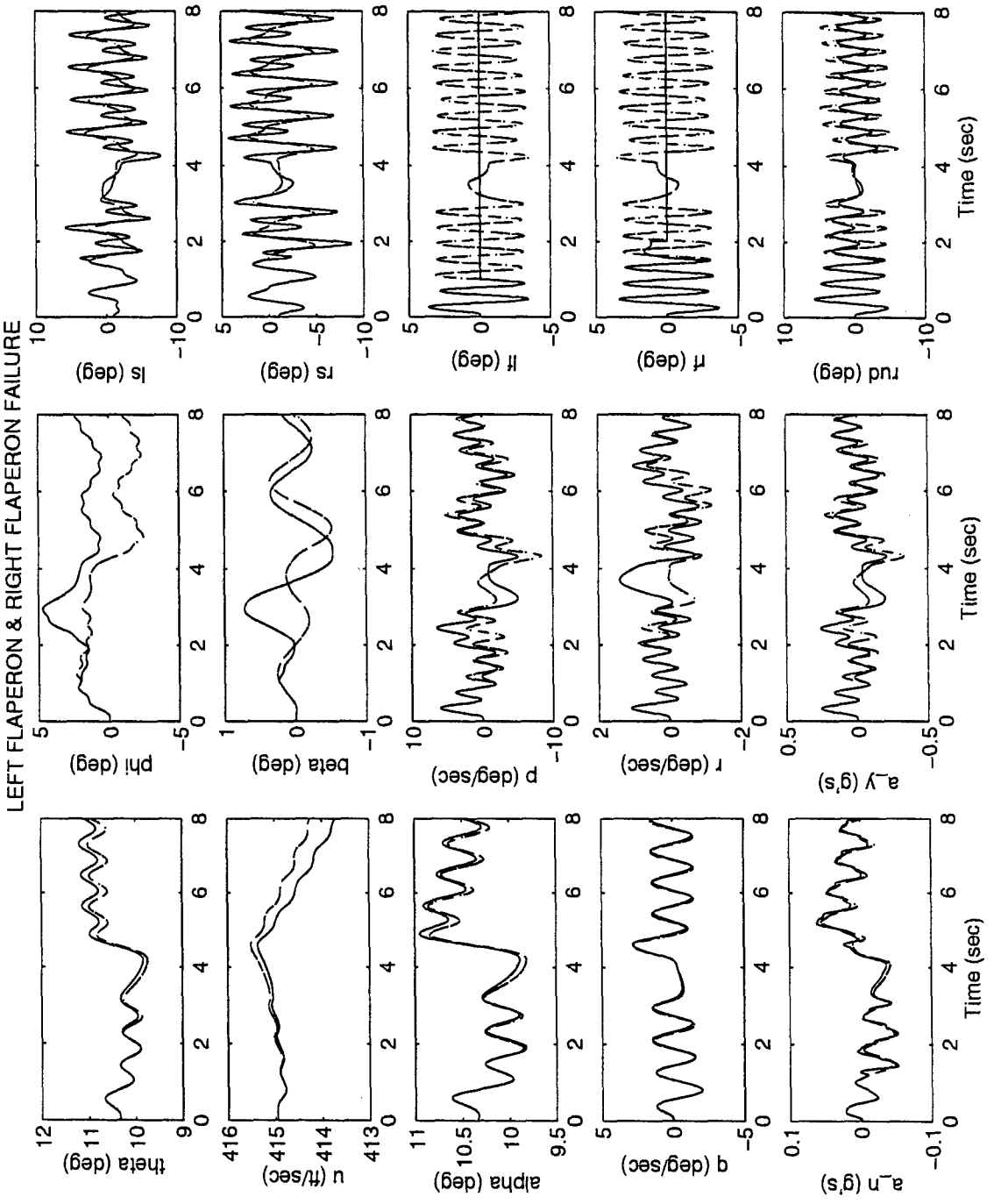

Figure D-38 MMAE-Based - Roll Doublet - With Control Redistribution
Left Flaperon & Right Flaperon Failure

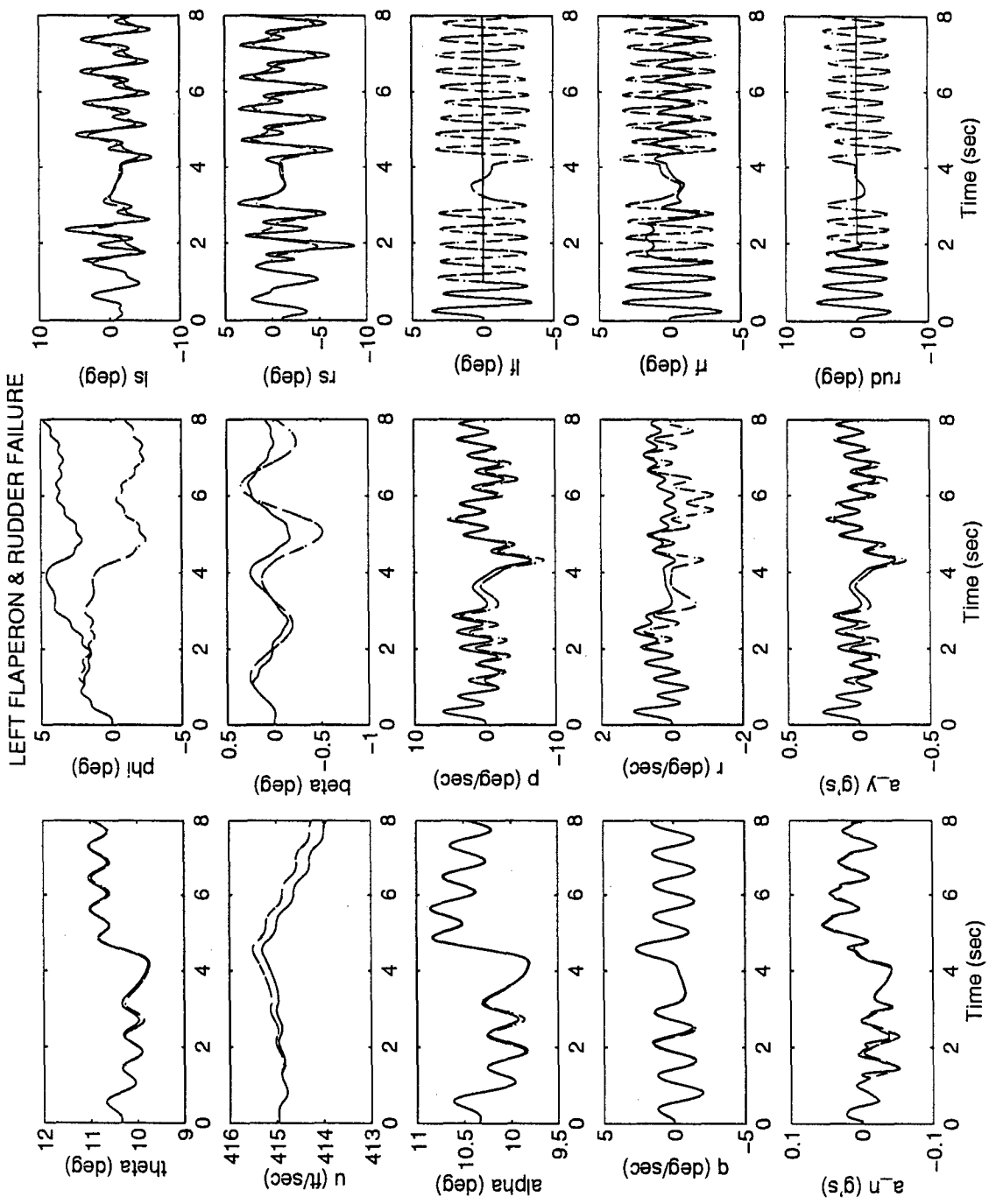

Figure D-39 MMAE-Based - Roll Doublet - With Control Redistribution
Left Flaperon & Rudder Failure

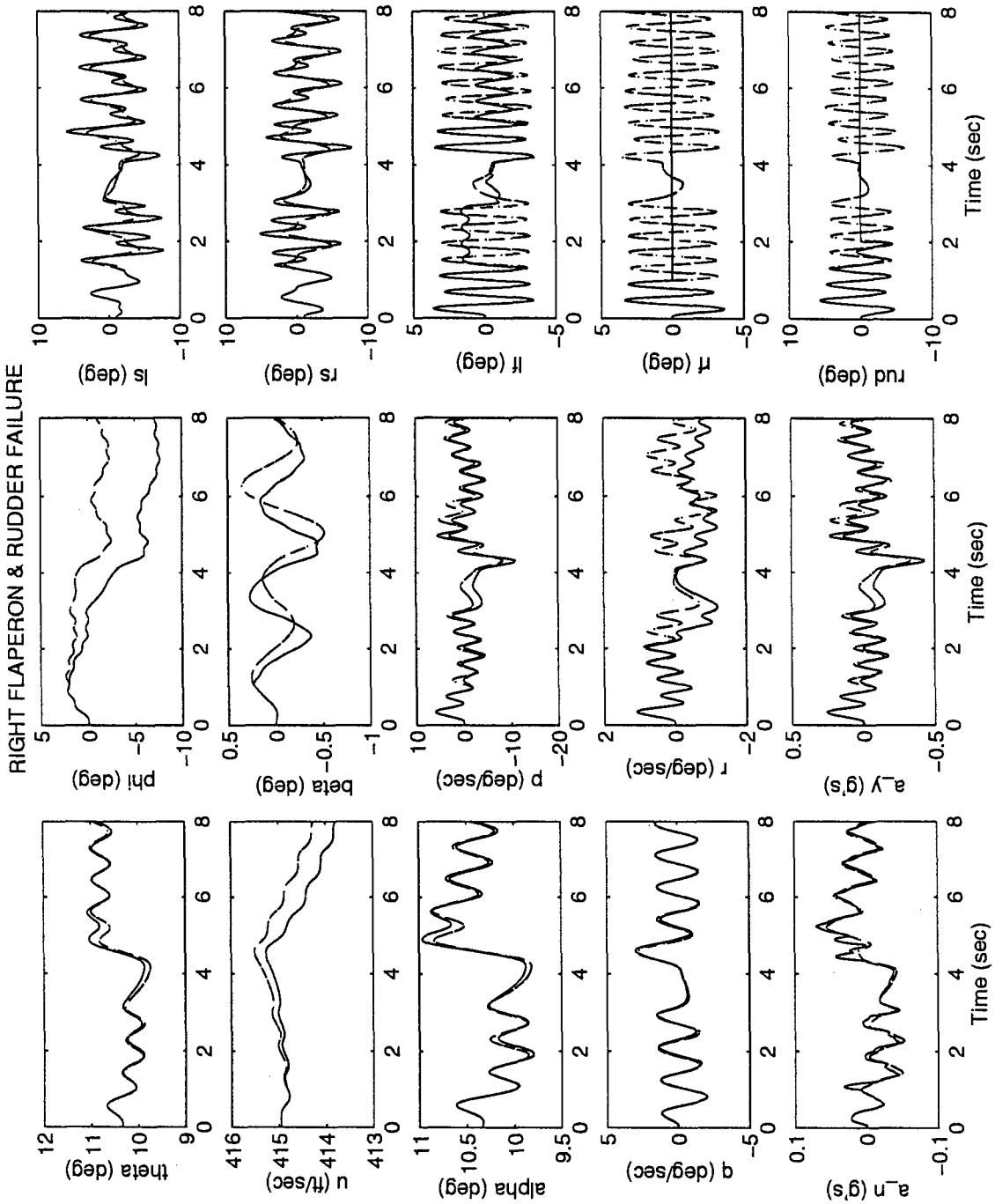

Figure D-40 MMAE-Based - Roll Doublet - With Control Redistribution
Right Flaperon & Rudder Failure

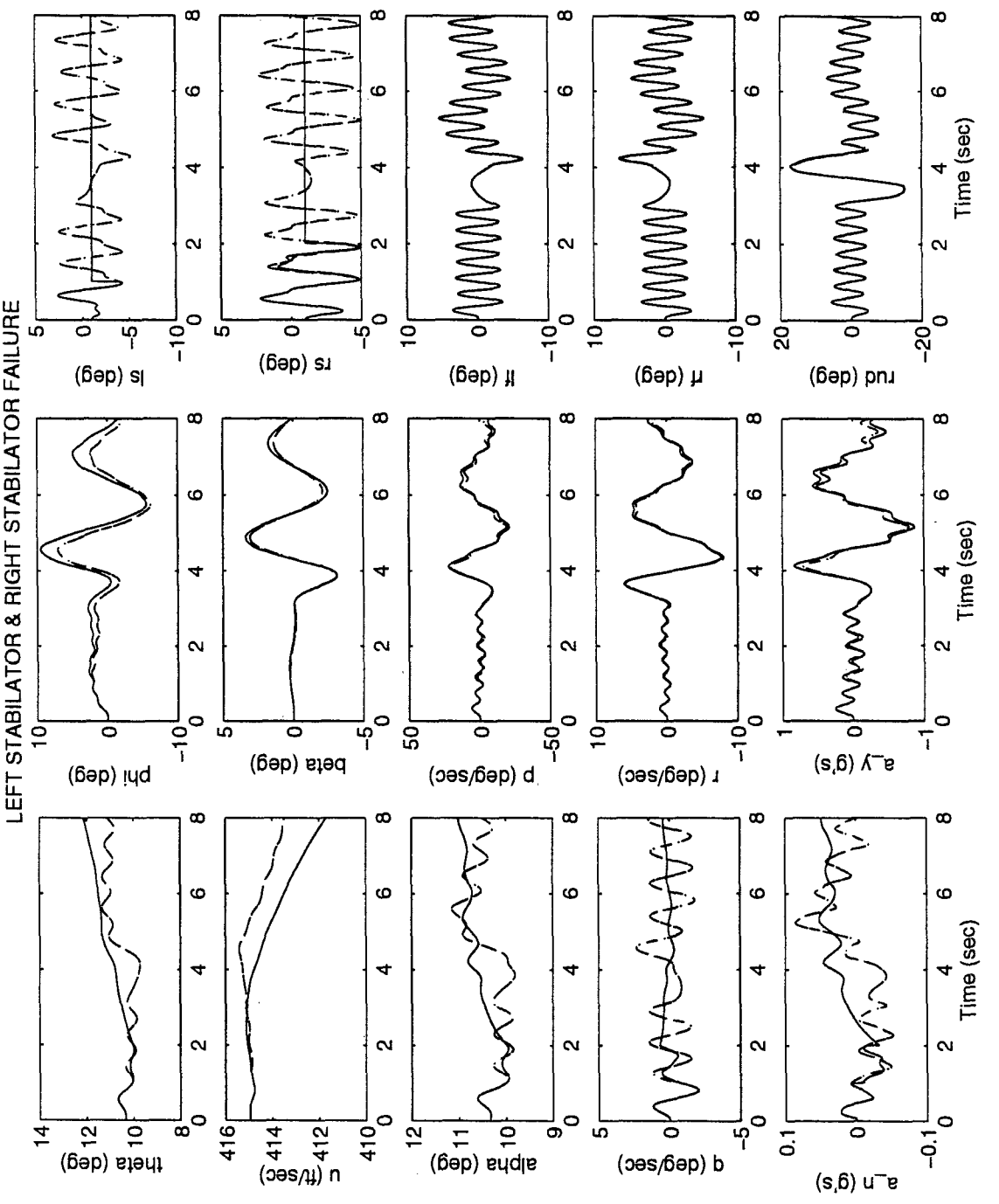

Figure D-41 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Stabilator & Right Stabilator Failure

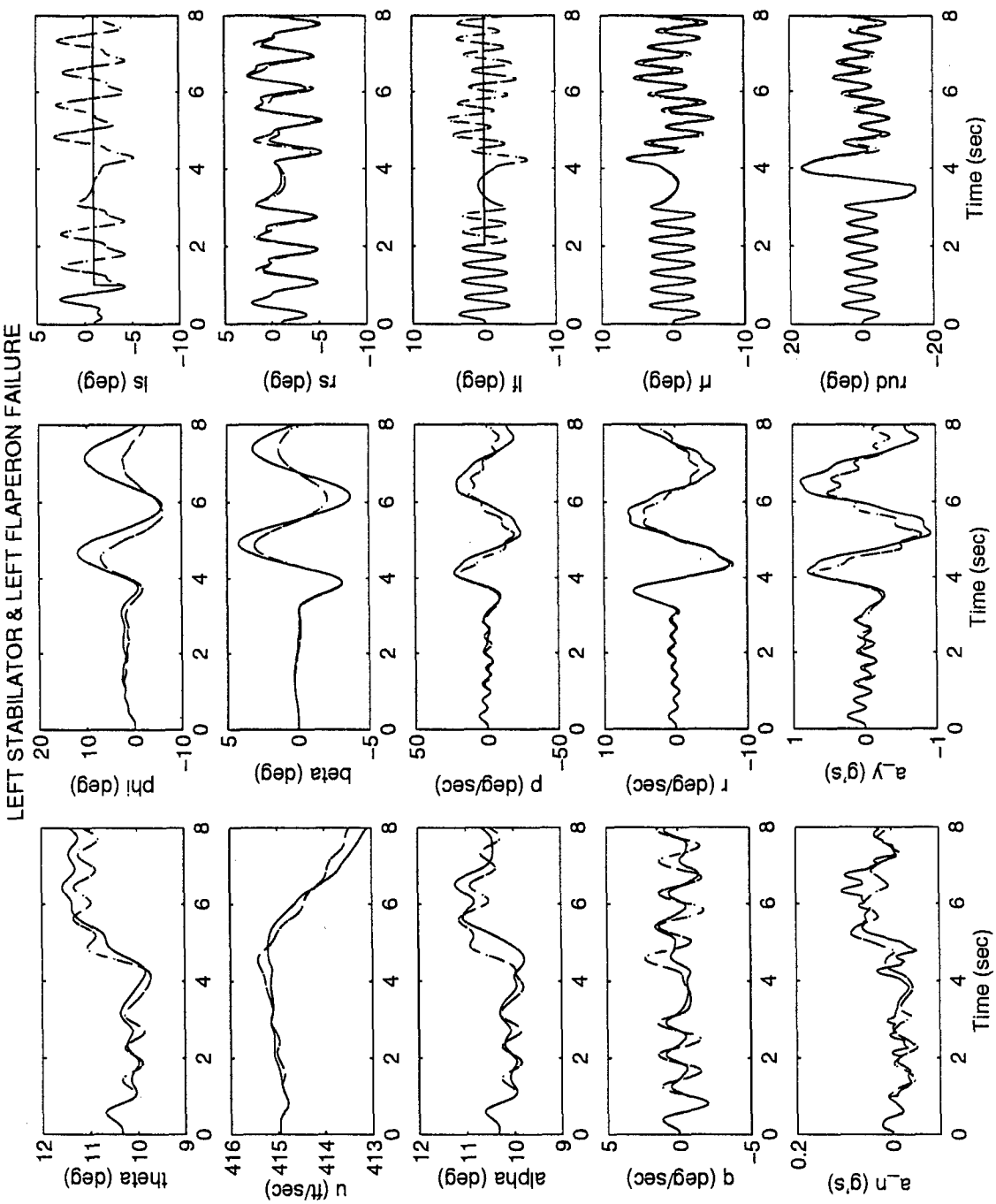

Figure D-42 MMAE-Based - Yaw Doublet - No Control Redistribution

Left Stabilator & Left Flaperon Failure

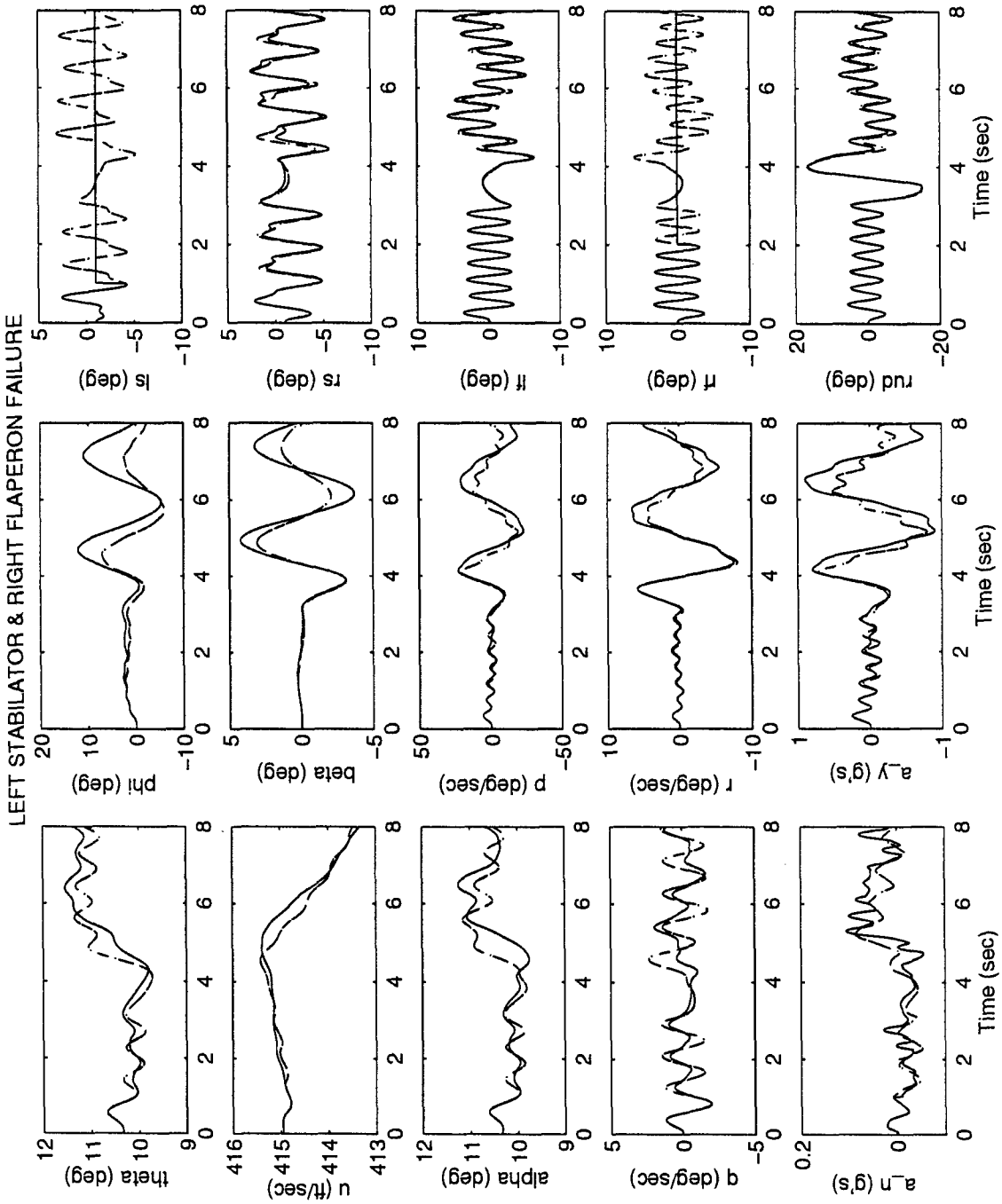

Figure D-43 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Stabilator & Right Flaperon Failure

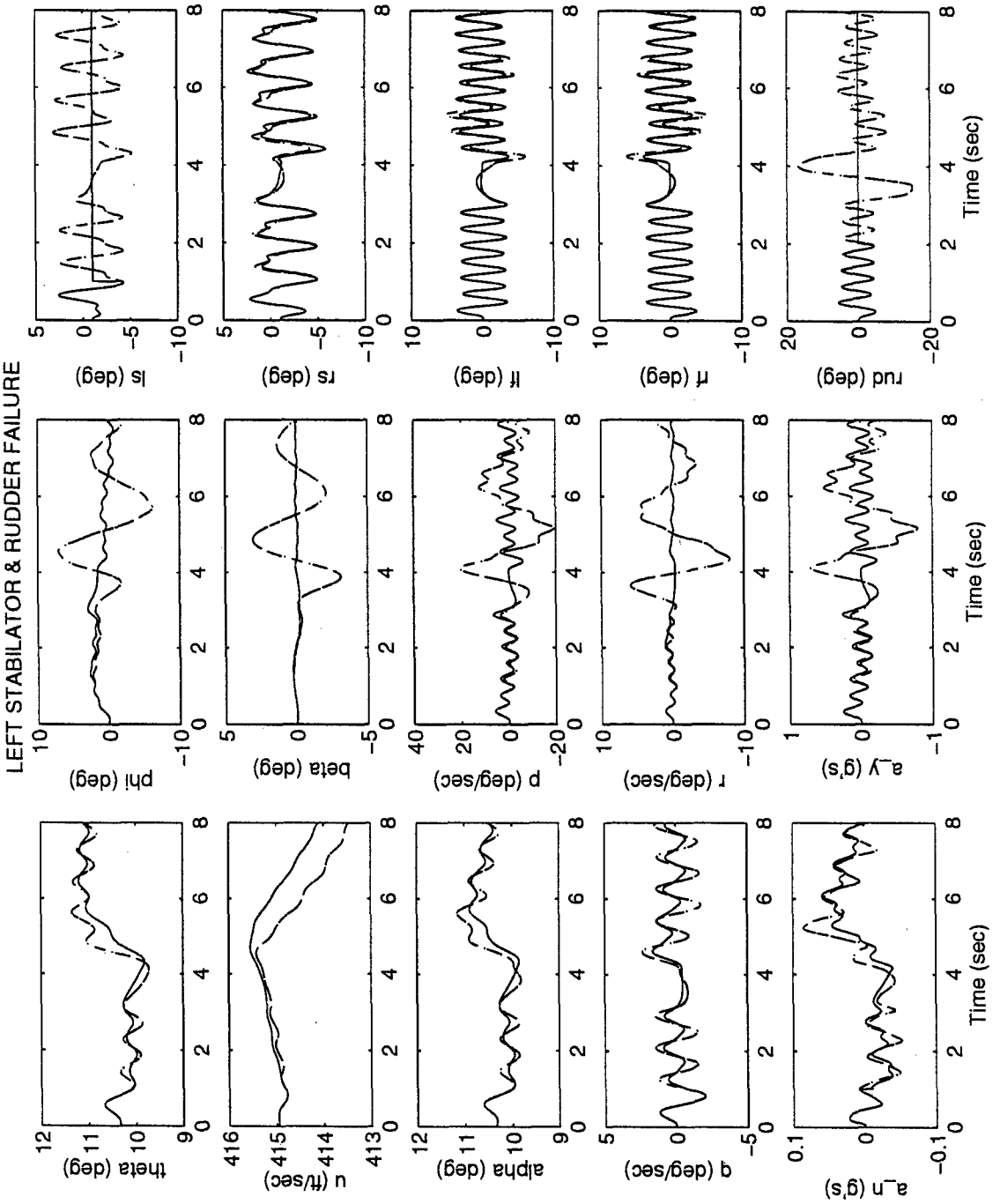

Figure D-44 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Stabilator & Rudder Failure

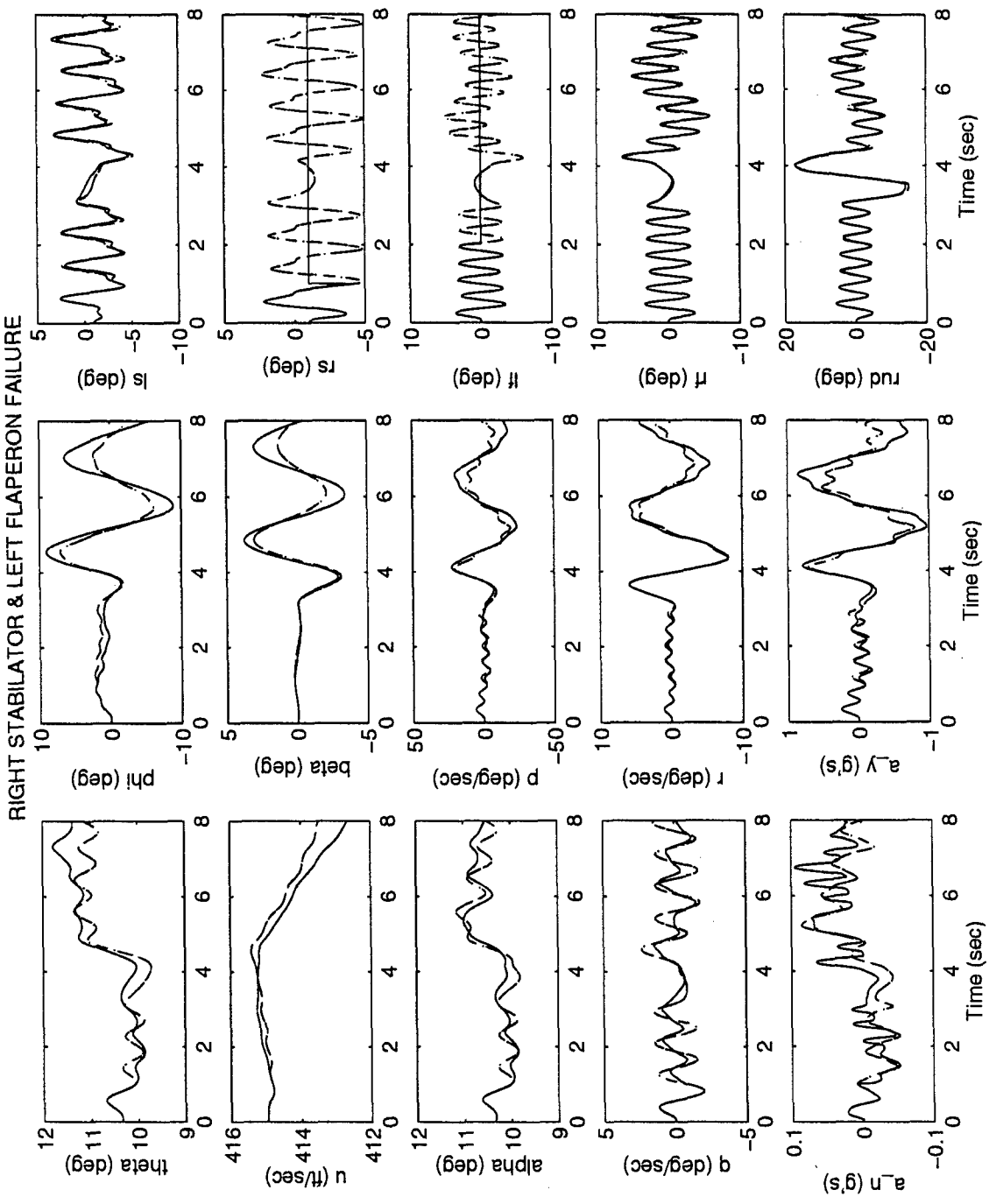

Figure D-45 MMAE-Based - Yaw Doublet - No Control Redistribution
Right Stabilator & Left Flaperon Failure

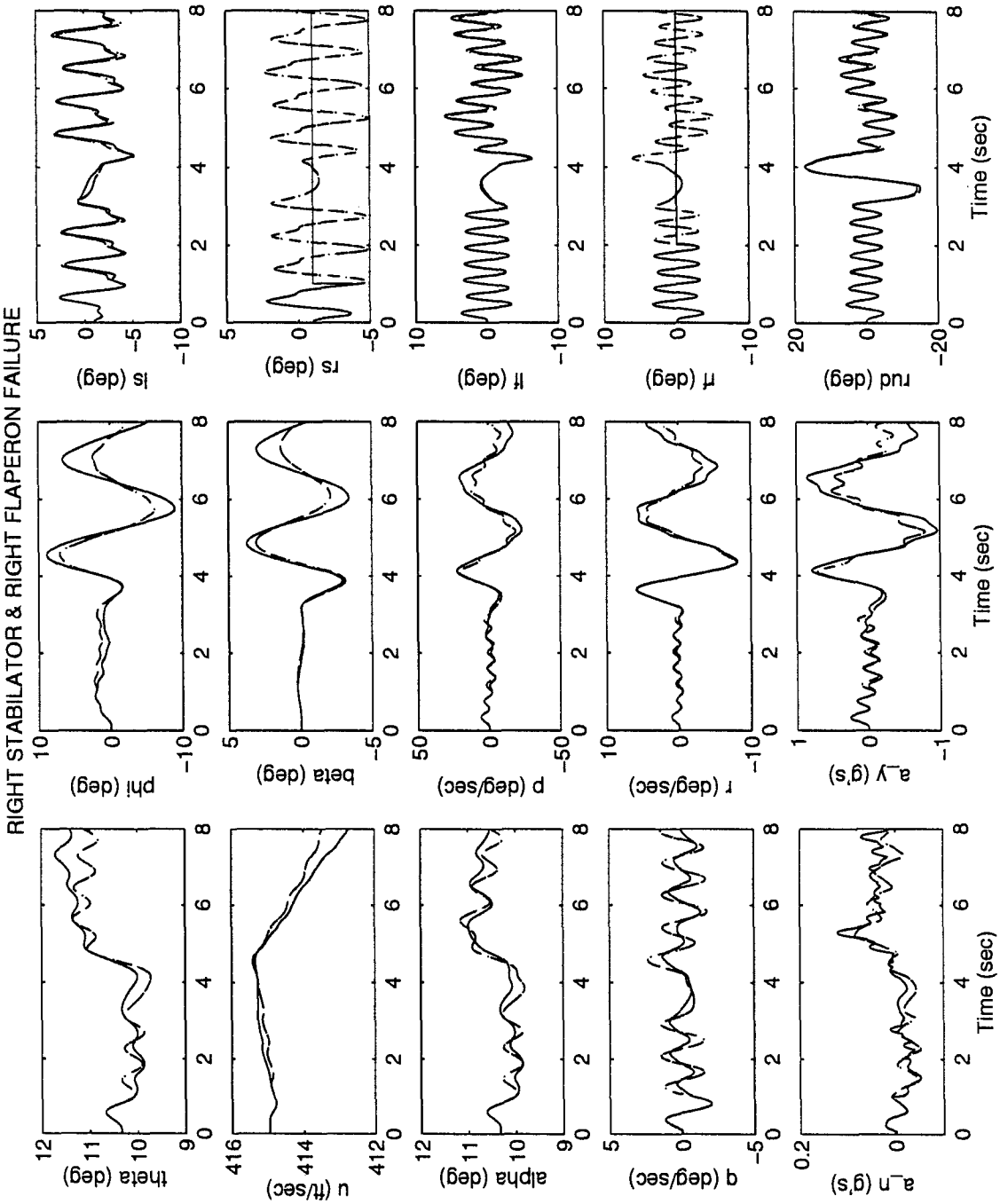

Figure D-46 MMAE-Based - Yaw Doublet - No Control Redistribution
Right Stabilator & Right Flaperon Failure

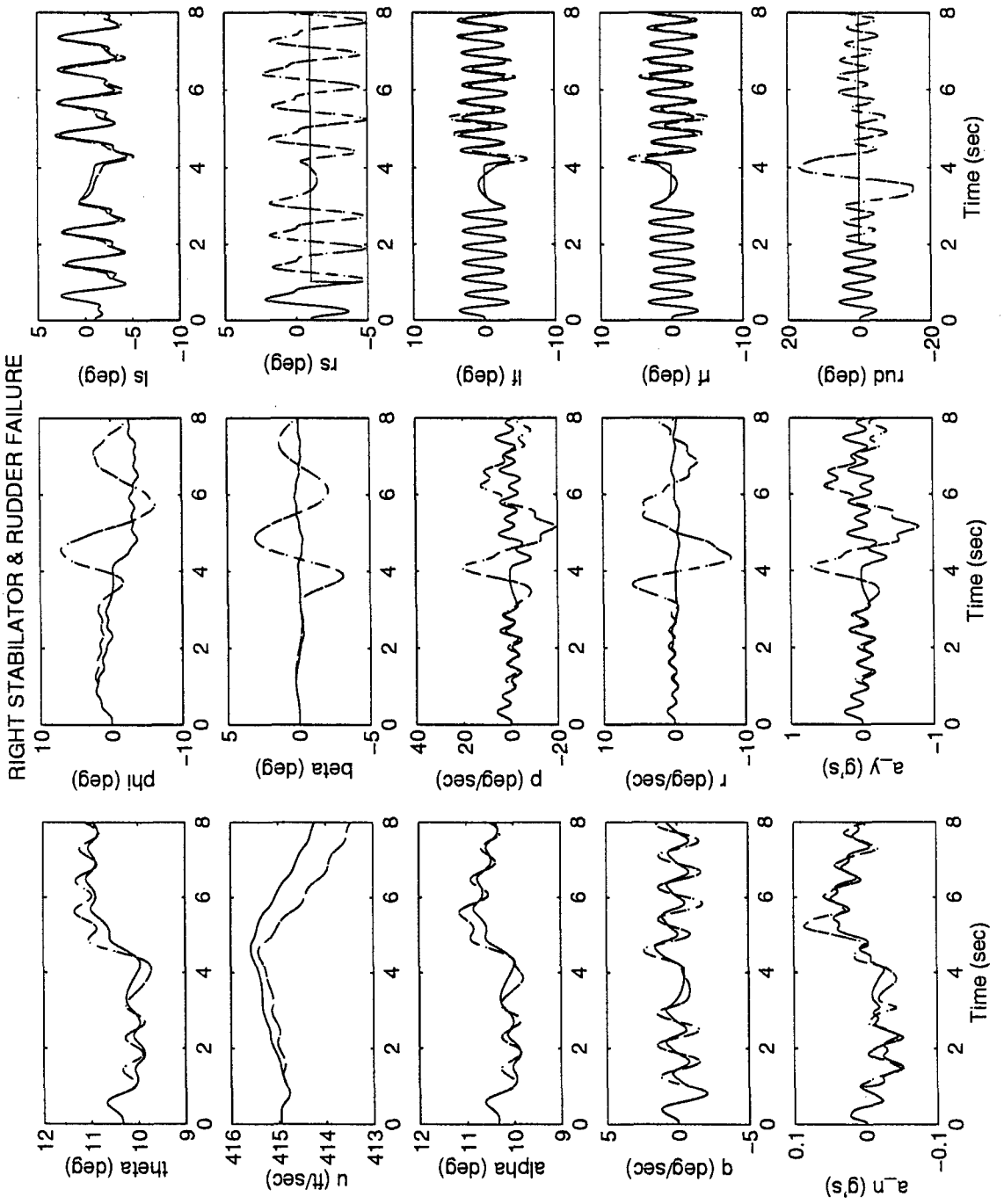

Figure D-47 MMAE-Based - Yaw Doublet - No Control Redistribution
Right Stabilator & Rudder Failure

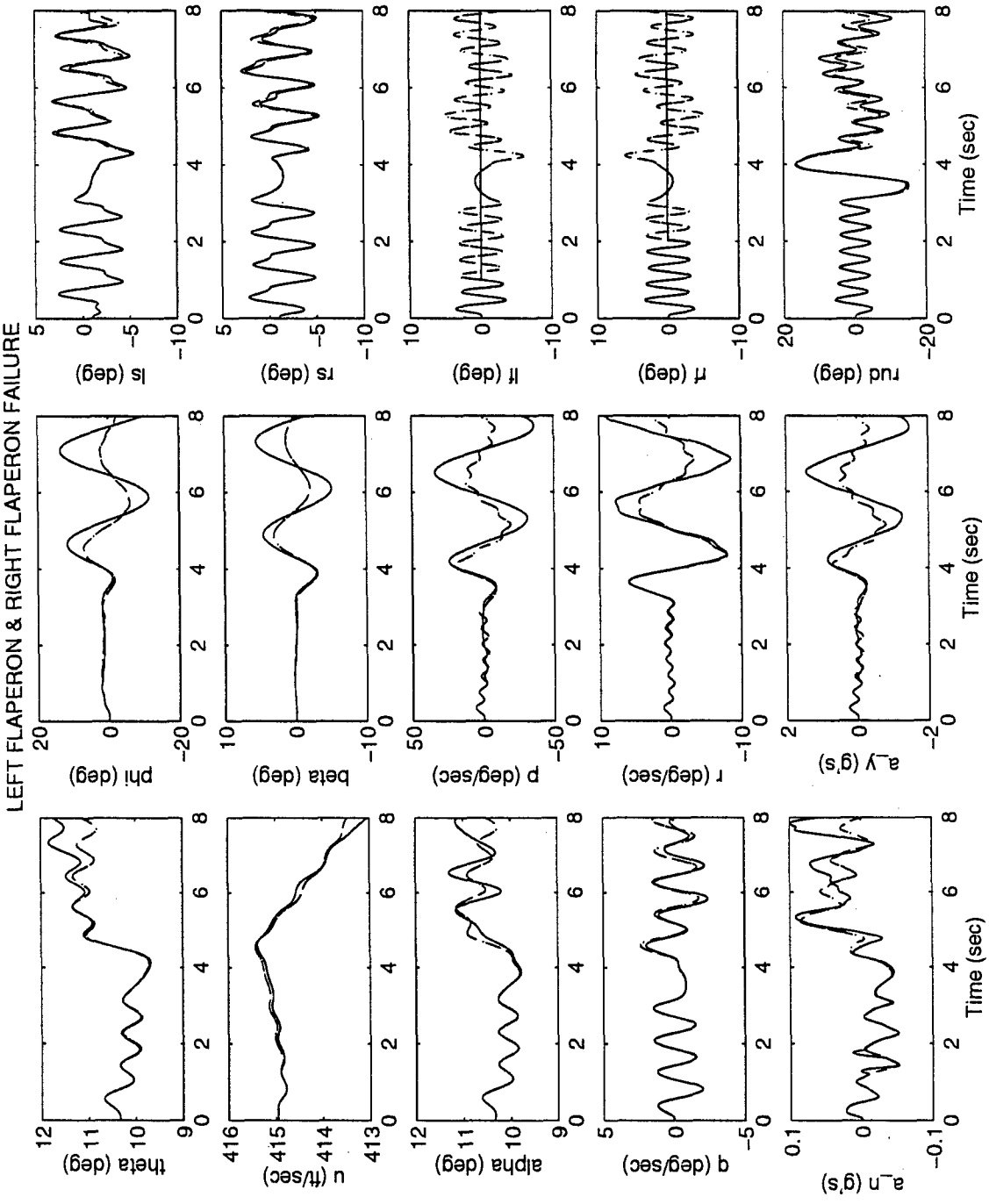

Figure D-48 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Flaperon & Right Flaperon Failure

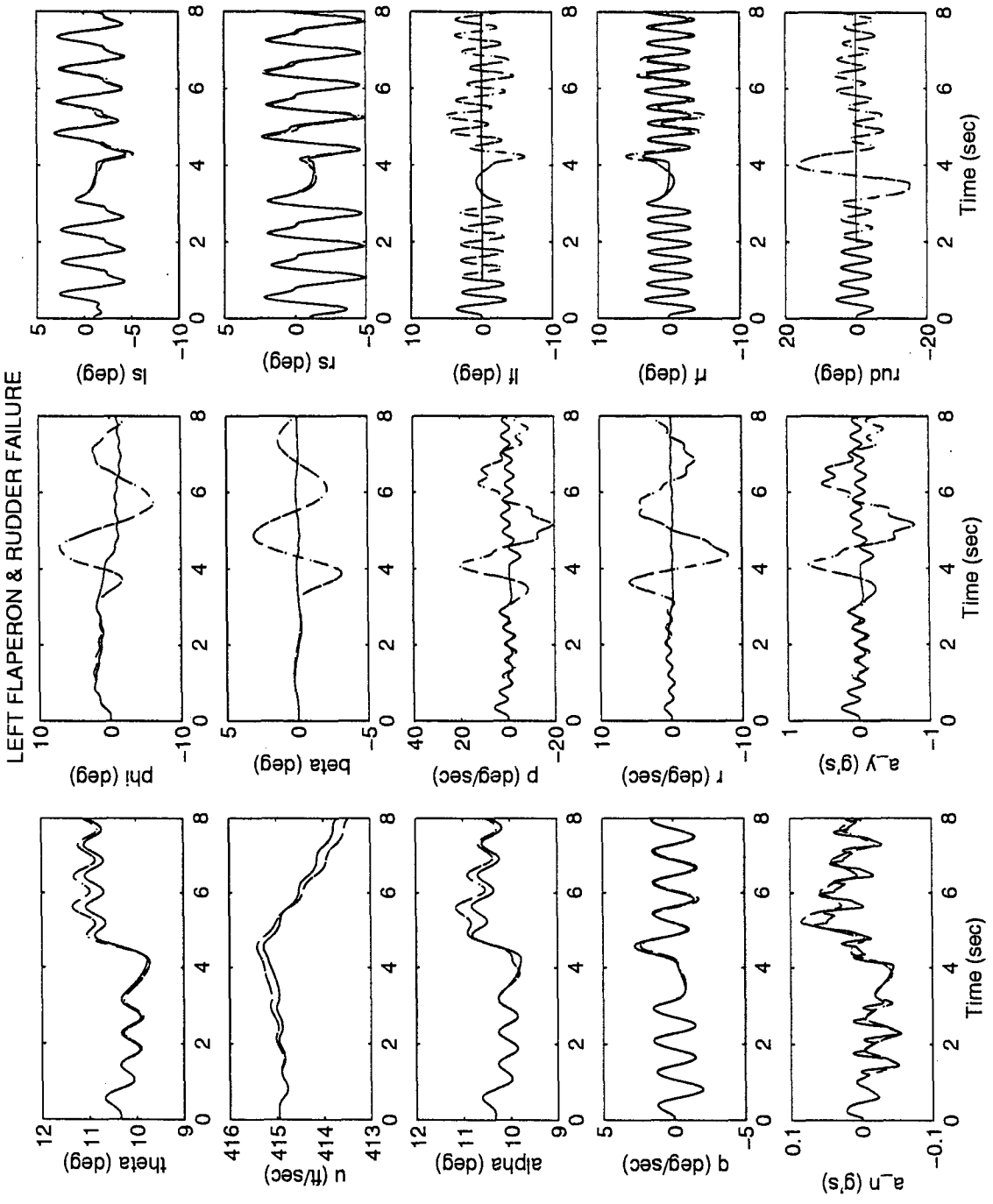

Figure D-49 MMAE-Based - Yaw Doublet - No Control Redistribution
Left Flaperon & Rudder Failure

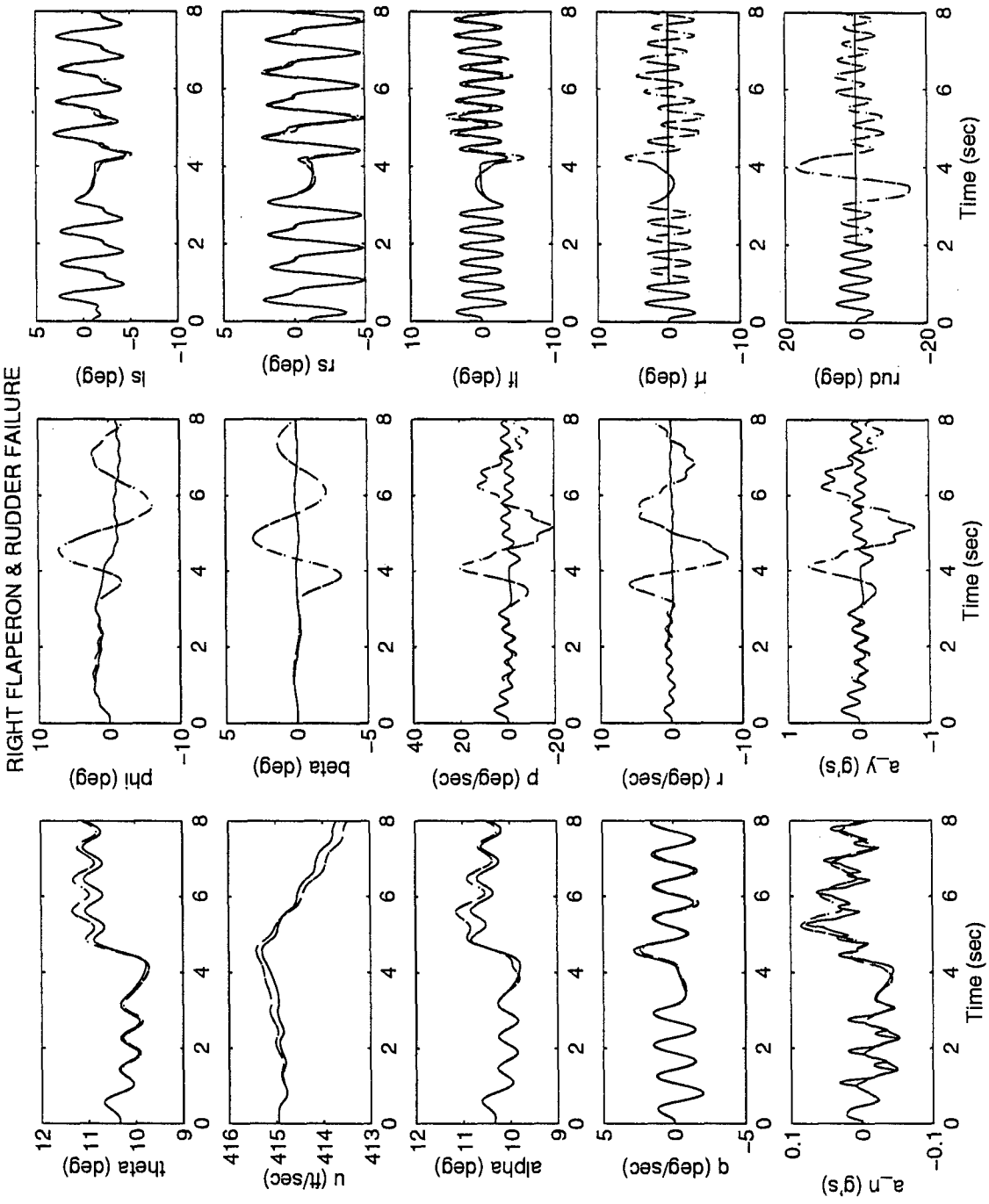

Figure D-50 MMAE-Based - Yaw Doublet - No Control Redistribution
Right Flaperon & Rudder Failure

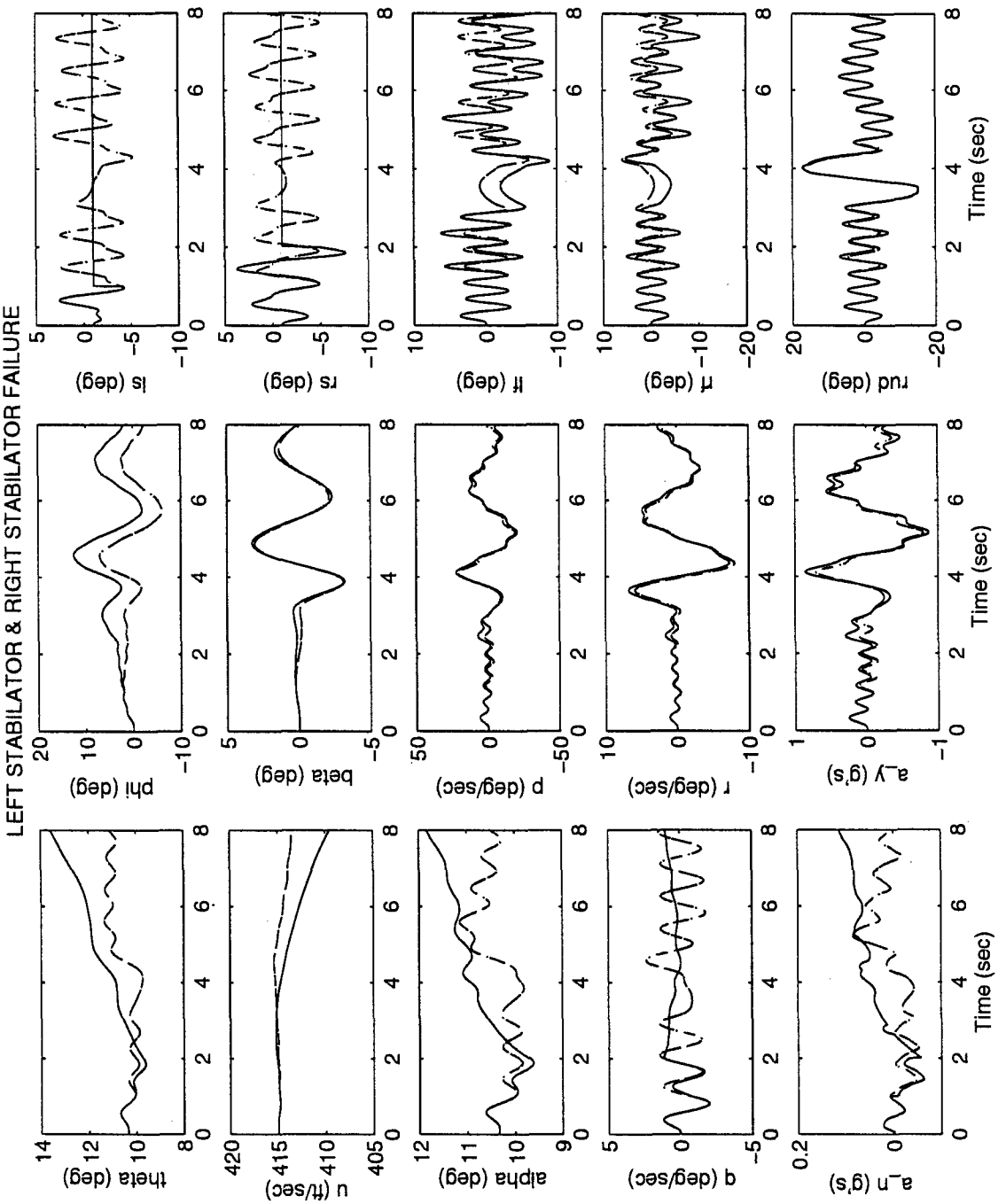

Figure D-51 MMAE-Based - Yaw Doublet - With Control Redistribution
Left Stabilator & Right Stabilator Failure

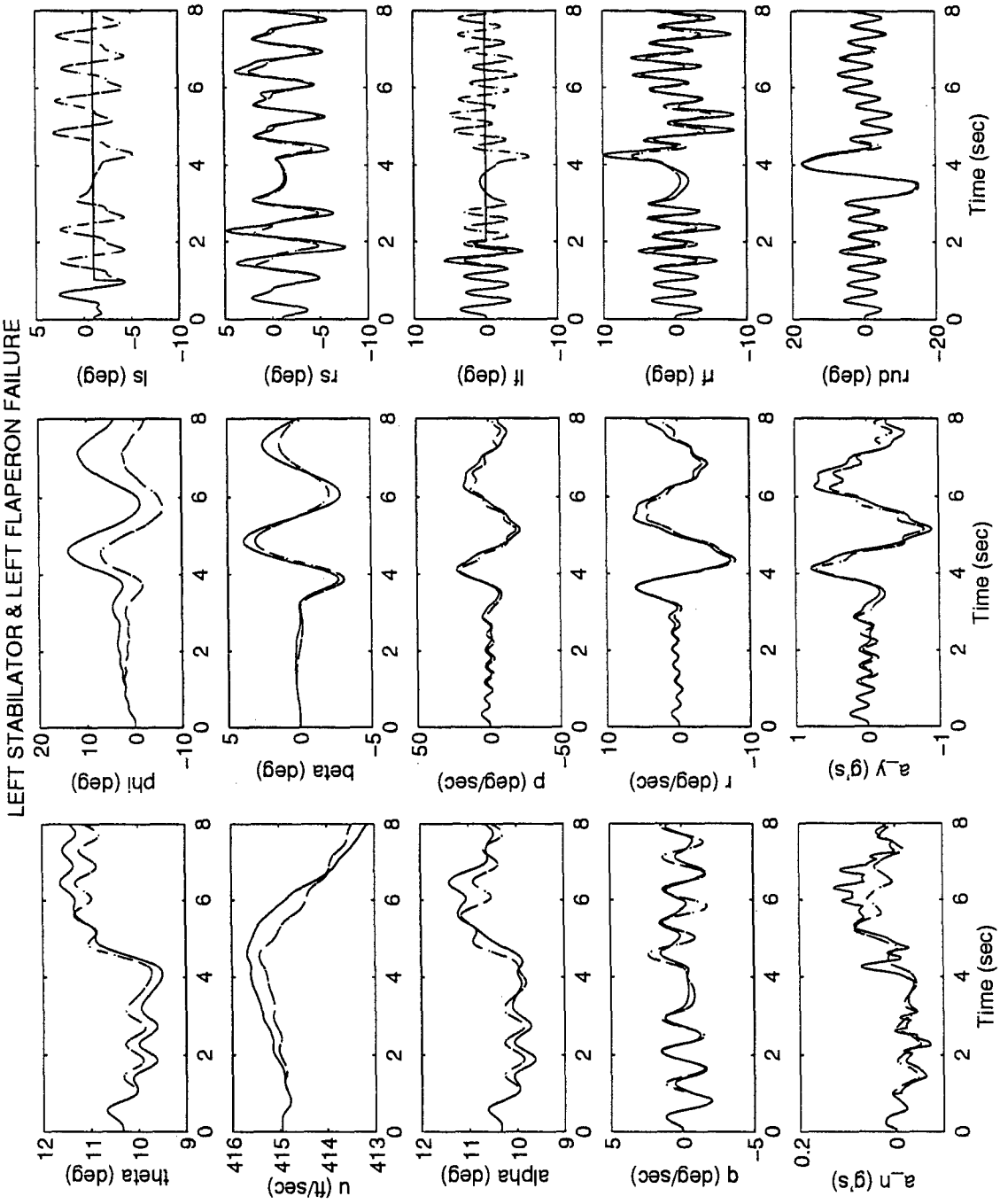

Figure D-52 MMAE-Based - Yaw Doublet - With Control Redistribution

Left Stabilator & Left Flaperon Failure

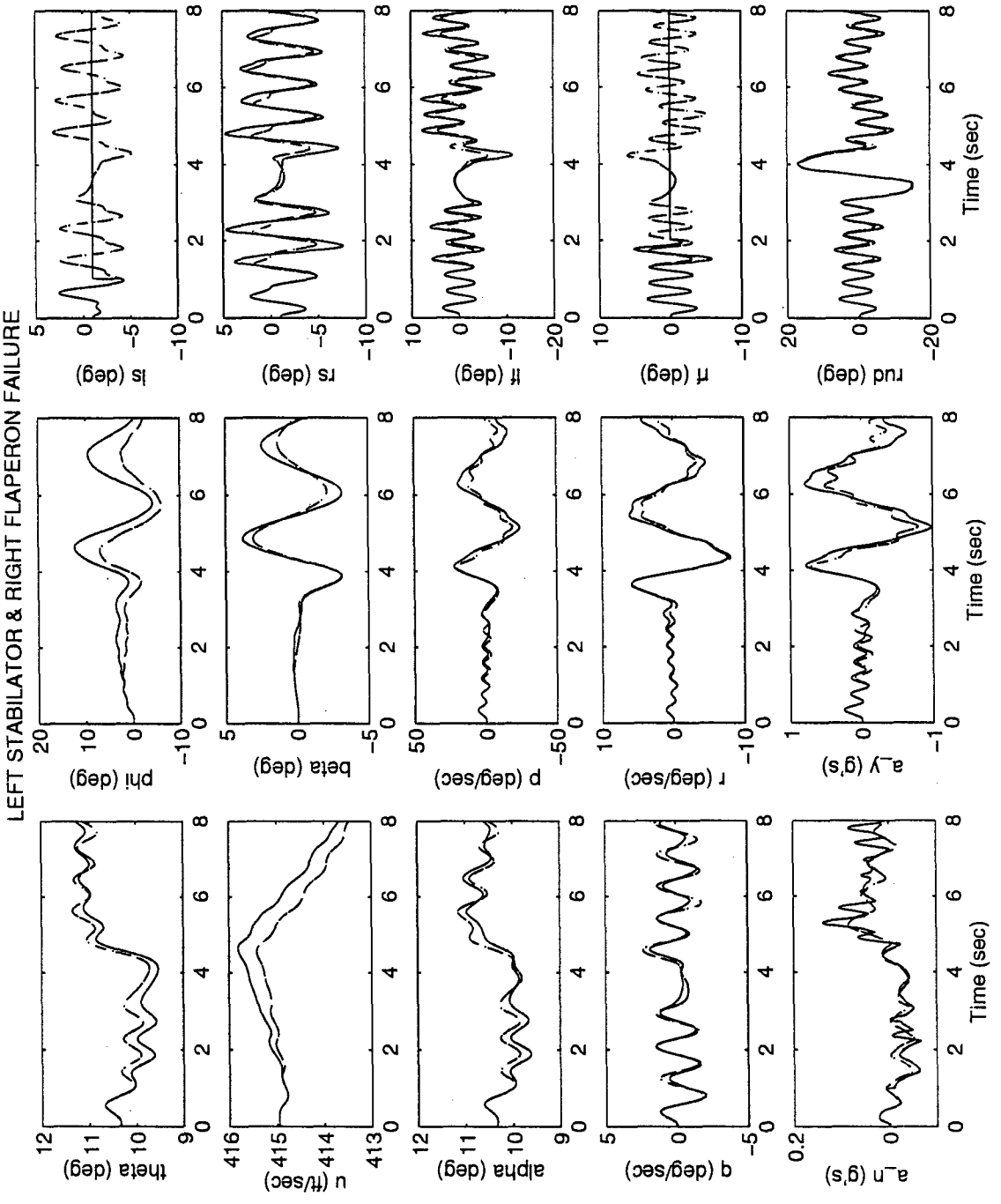

Figure D-53 MMAE-Based - Yaw Doublet - With Control Redistribution
Left Stabilator & Right Flaperon Failure

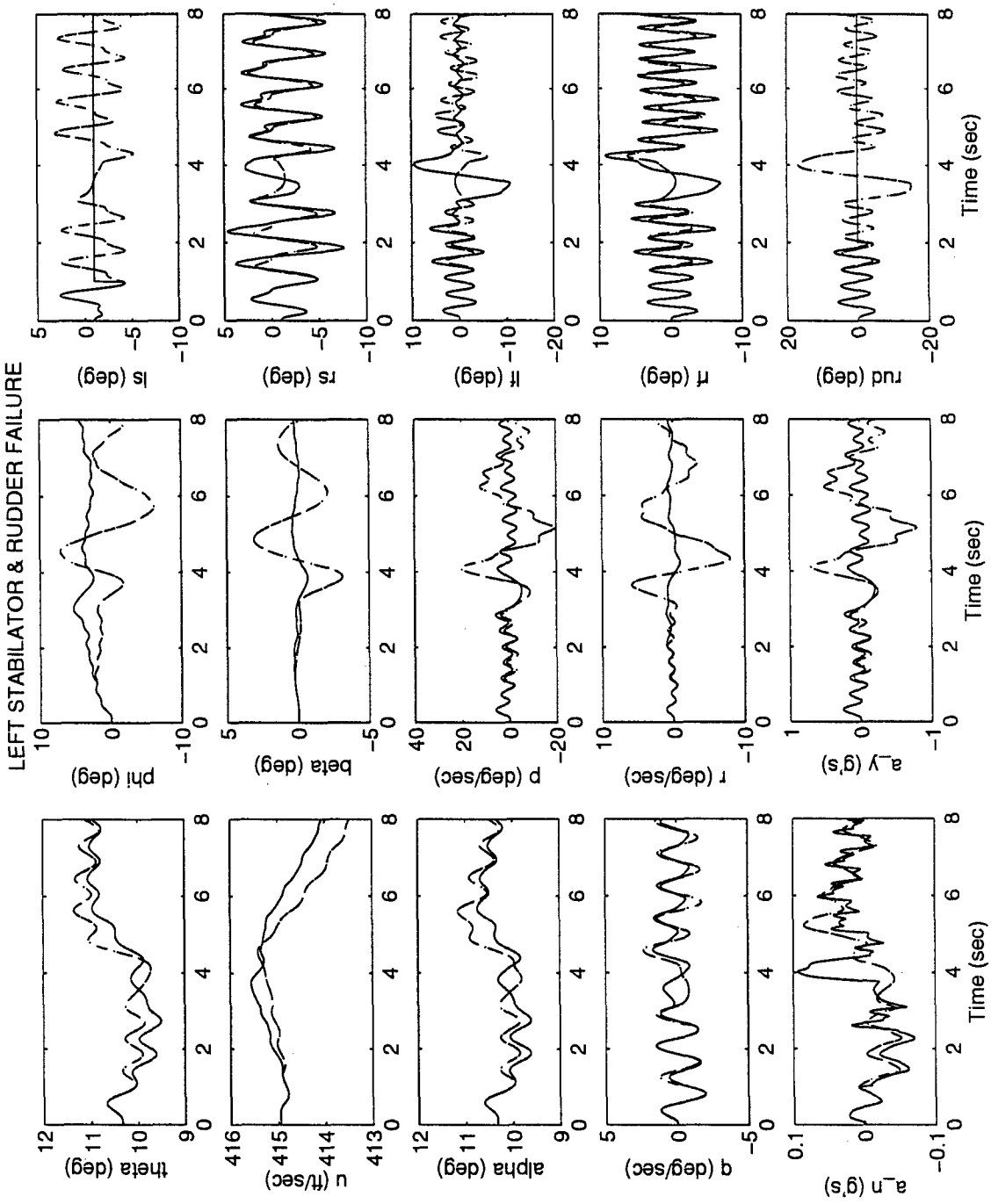

Figure D-54 MMAE-Based - Yaw Doublet - With Control Redistribution
Left Stabilator & Rudder Failure

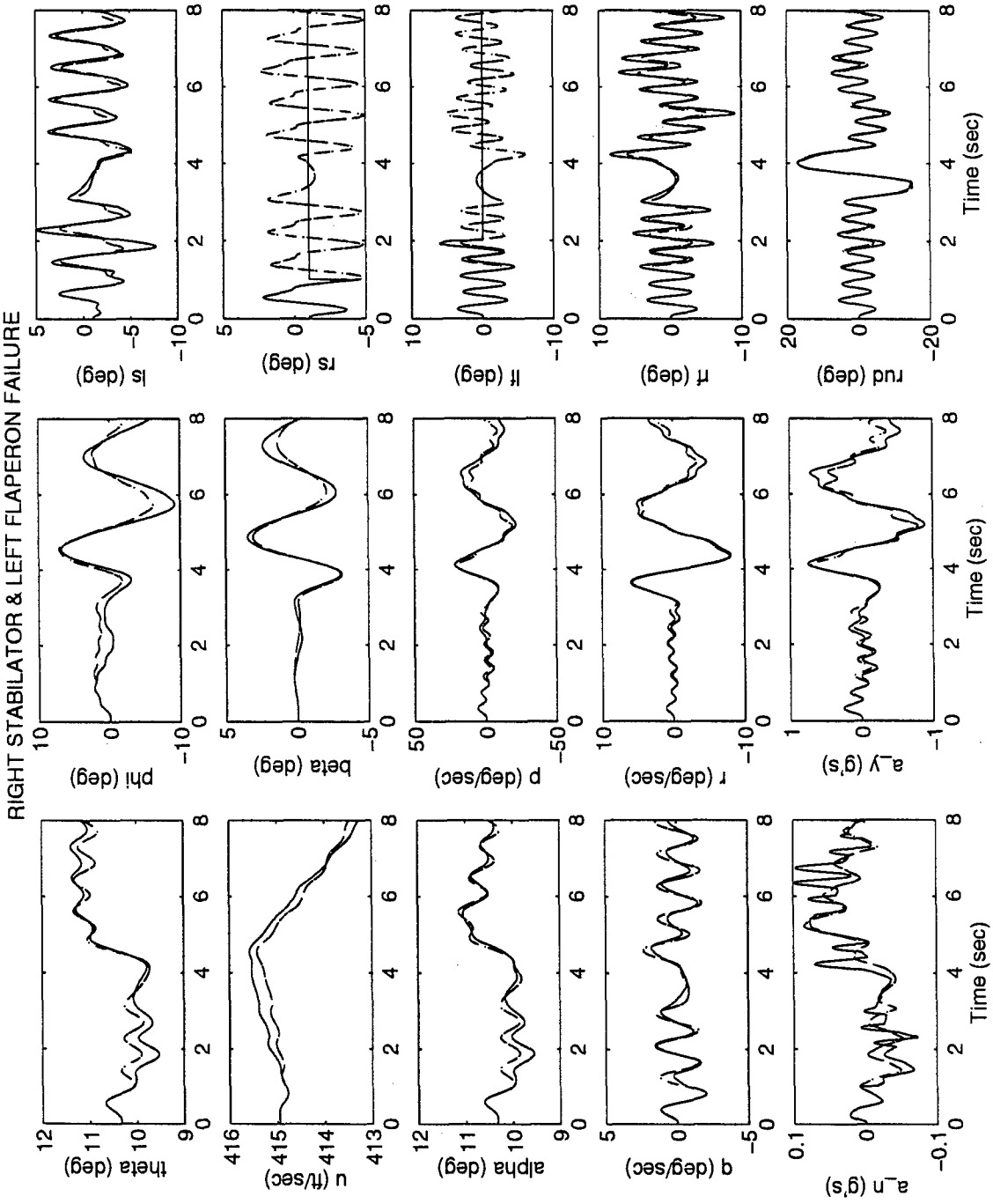

Figure D-55 MMAE-Based - Yaw Doublet - With Control Redistribution
Right Stabilator & Left Flaperon Failure

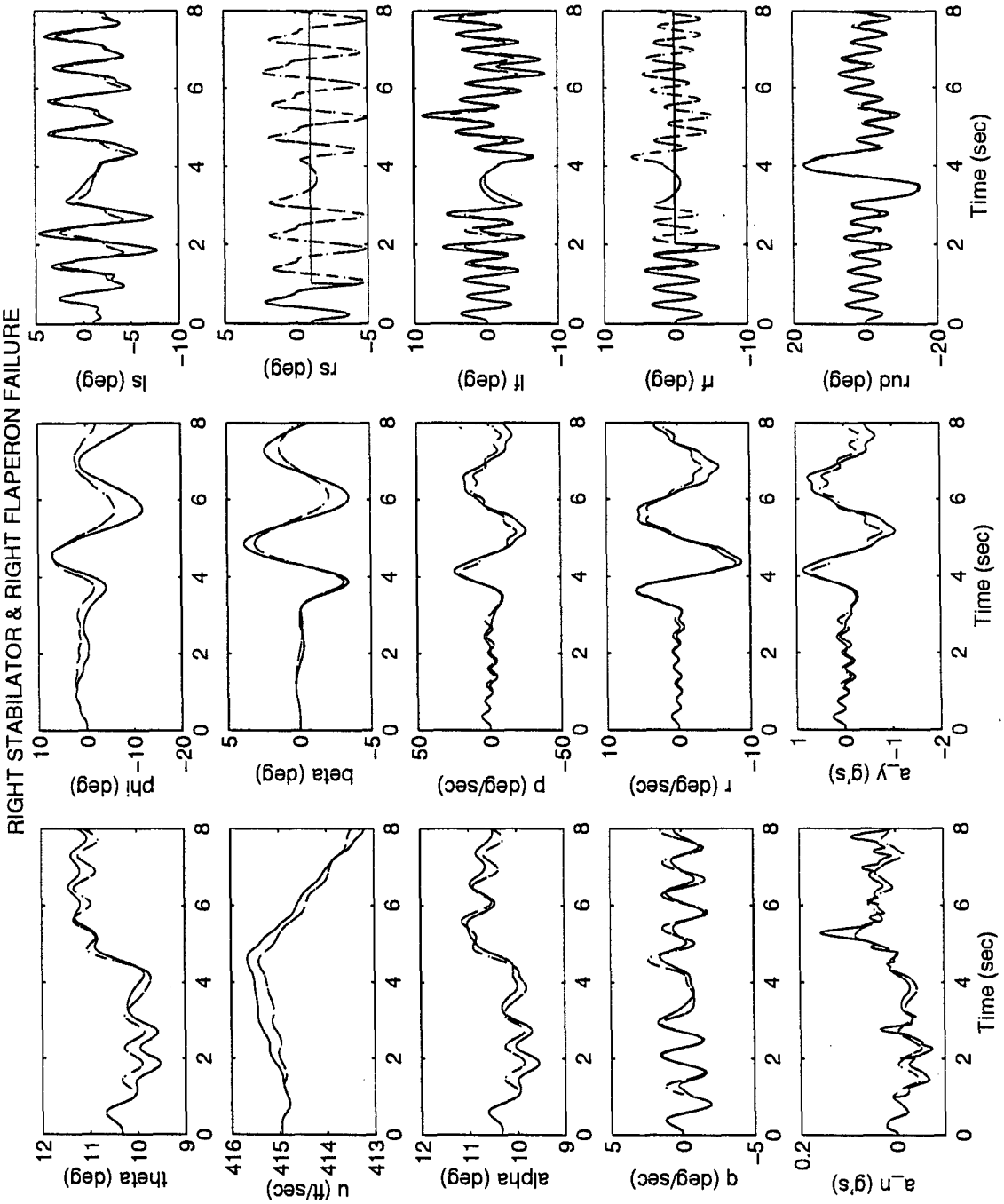

Figure D-56 MMAE-Based - Yaw Doublet - With Control Redistribution
Right Stabilator & Right Flaperon Failure

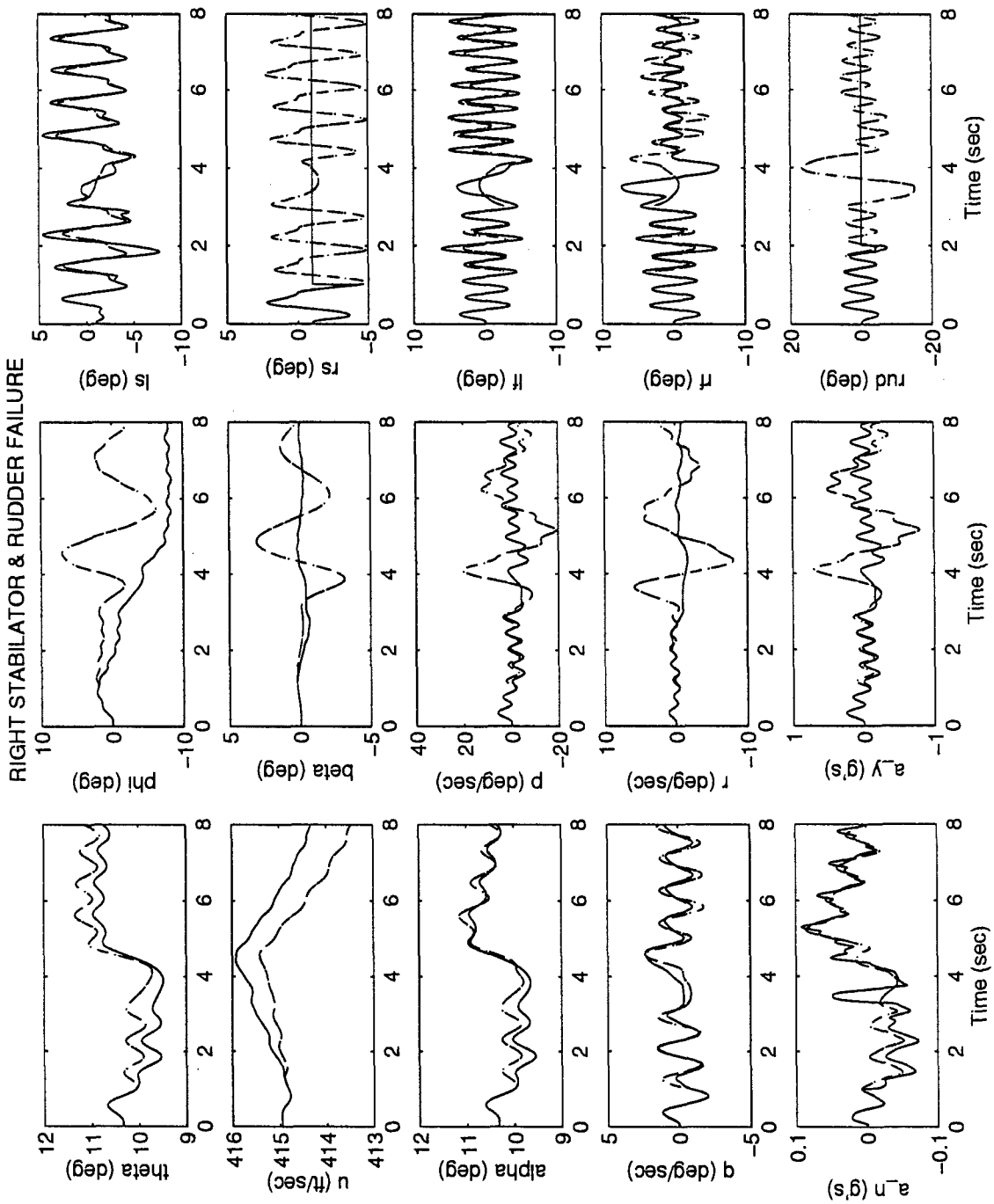

Figure D-57 MMAE-Based - Yaw Doublet - With Control Redistribution
Right Stabilator & Rudder Failure

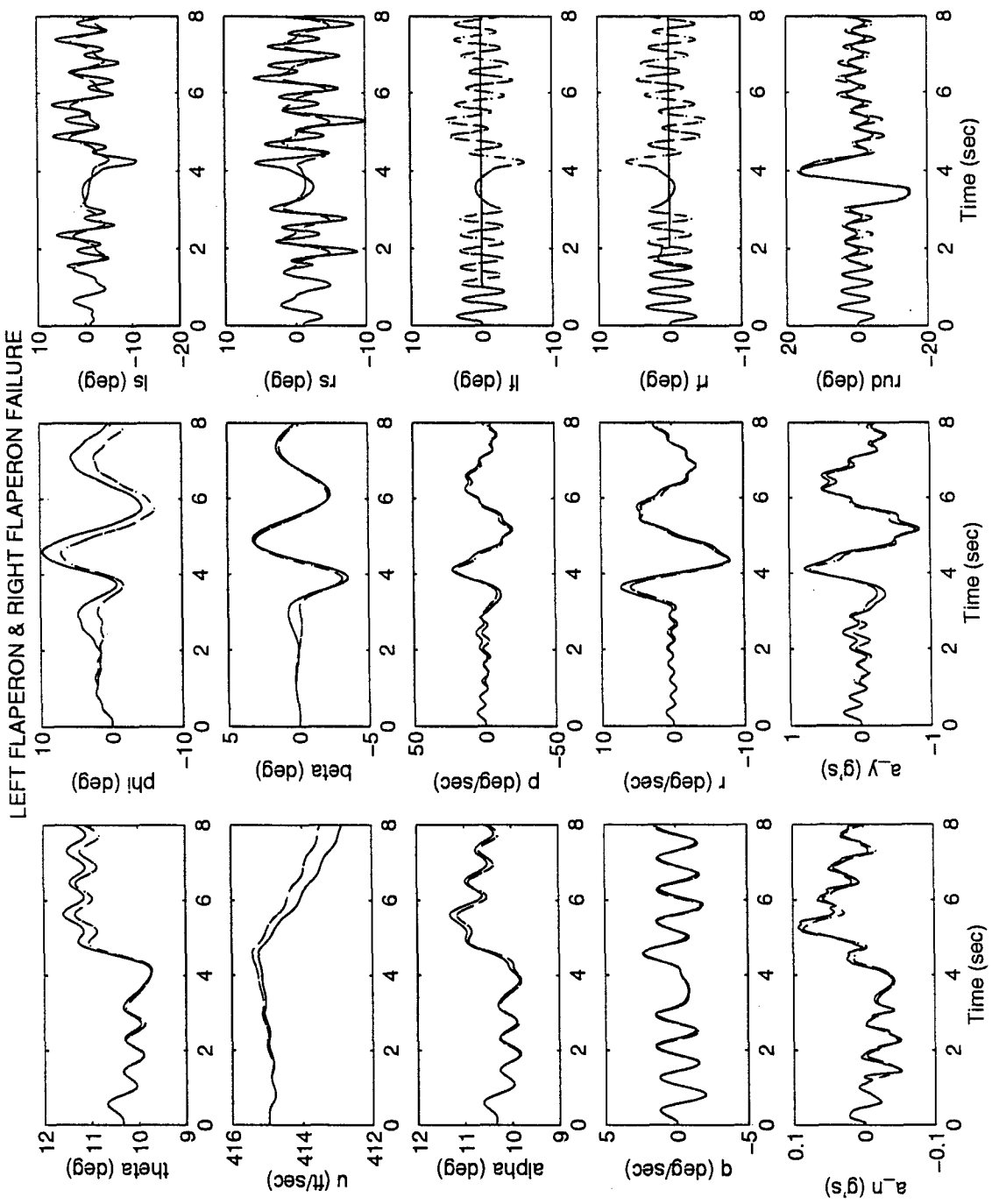

Figure D-58 MMAE-Based - Yaw Doublet - With Control Redistribution
Left Flaperon & Right Flaperon Failure

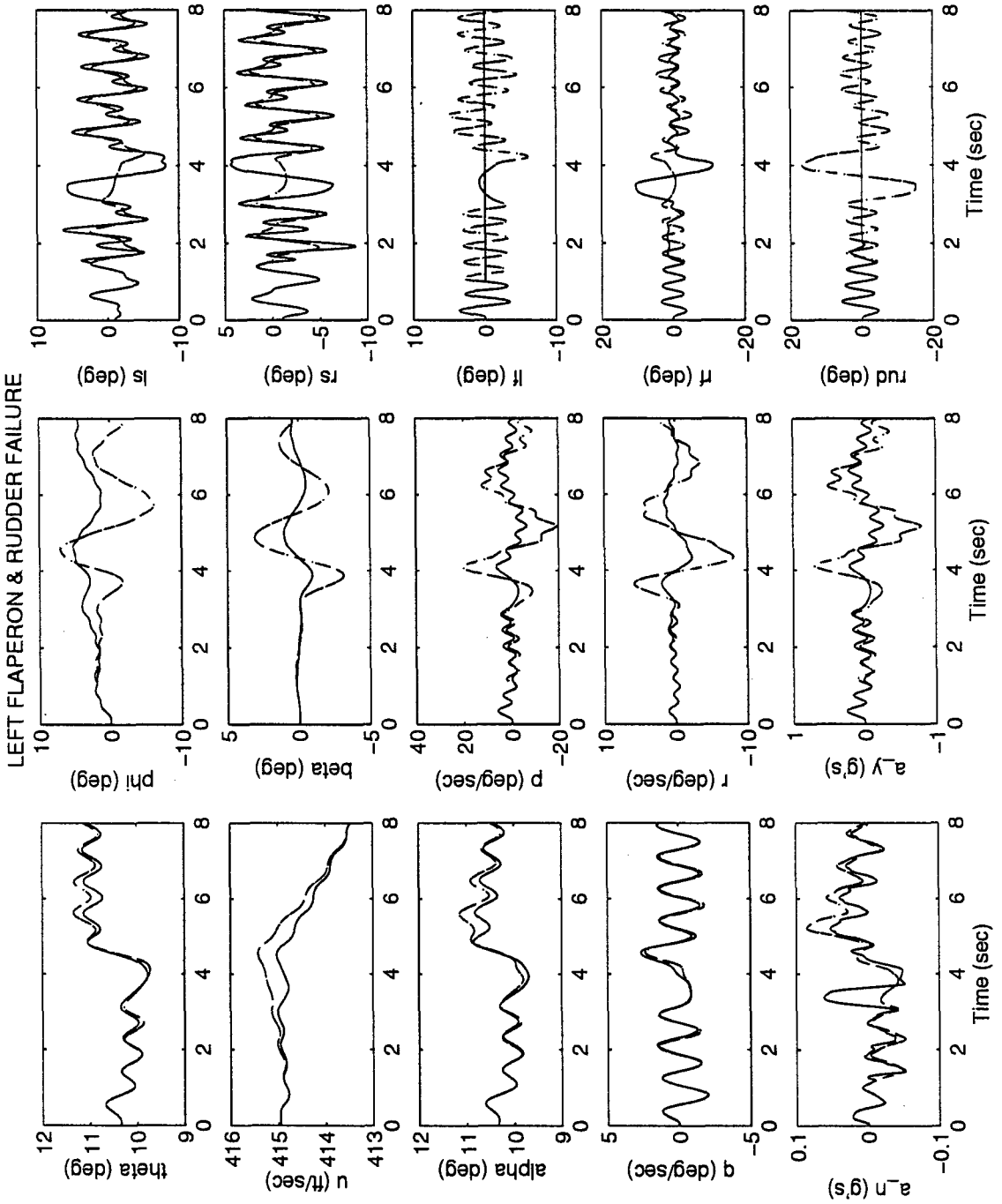

Figure D-59 MMAE-Based - Yaw Doublet - With Control Redistribution
Left Flaperon & Rudder Failure

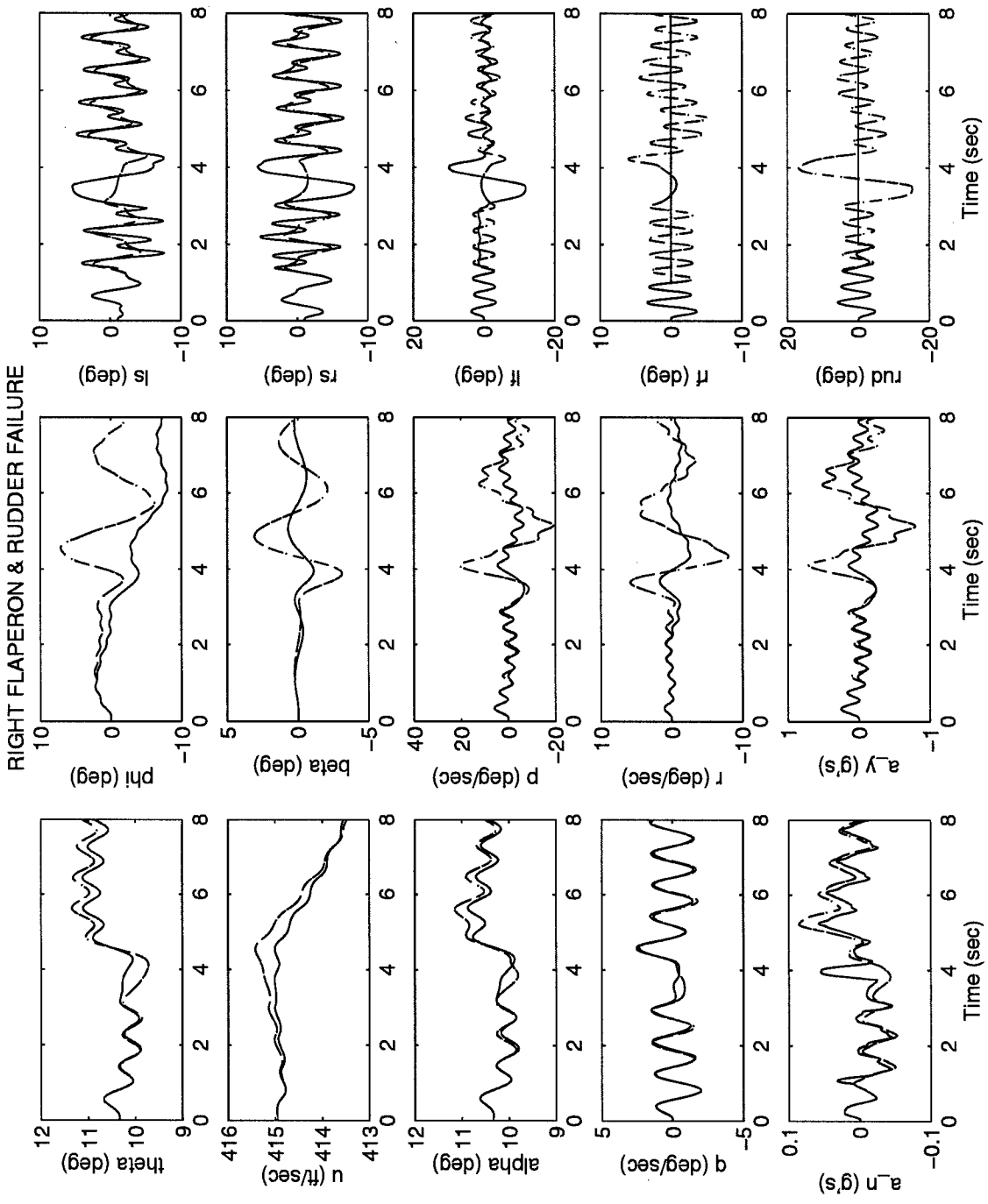

Figure D-60 MMAE-Based - Yaw Doublet - With Control Redistribution
Right Flaperon & Rudder Failure

References

1. Adams, Captain Terry A. *The Requirements, Design, and Development of a Rapidly Reconfigureable, Photo-Realistic Virtual Cockpit Prototype.* MS Thesis, AFIT/GA/GCS/ENG/96-02, School of Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1996.

2. Athans, M., *et al.* "The Stochastic Controll of the F8-C 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).

3. Baram, Y. Information, Consistent Estimation and Dynamic System Identification, Rep. ESL-R-718. Electronic Systems Lab., Dept. of Elec. Eng., MIT, Cambridge, Massachusetts (November 1976).

4. Baram, Y., and Sandell, N. R., Jr. "An Information Theoretic Approach to Dynamic System Modelling and Identification", *IEEE Transactions on Automated Control,* AC-23 (1), 61-66 (1978).

5. Baram, Y., and Sandell, N. R., Jr. "Consistent Estimation of Finite Parameter Sets With Application to Linear Systems Identification", *IEEE Transactions on Automated Control,* AC-17 (3), 308-319 (1972).

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

7. Cacciatore, 2Lt Vincent J. A. *A Quantitative Feedback Theory FCS Design for the Subsonic Envelope of the VISTA F-16 Including Configuration Variation and Aerodynamic Control Effector Failures.* MS Thesis, AFIT/GE/ENG/95D-04, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.

8. Chang, C. B. and M. Athans. "State Estimation for Discrete Systems with Switching Parameters," *IEEE Transactions on Aerospace and Electronic Systems, AES-14*(4):418-424 (May 1978).

9. Eide, Capt Peter Keith. *Implementation and Demonstration of a Multiple Model Adaptive Estimation Failure Detection System for the F-16.* MS Thesis, AFIT/GE/ENG/94D-06, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994.

10. Eide, Capt Peter K. and and Peter S. Maybeck. "Implementation and Demonstration of a Multiple Model Adaptive Estimation Failure Detection System fo the F-16," *Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA,* 1873-1878 (December 1995).

11. Eide, P. and P. Maybeck. "An MMAE Failure Detection System for the F-16," *IEEE Transactions on Aerospace and Electronic Systems, 32(3)*:1125-1136 (July 1996).

12. Fry, C. M. and A. P. Sage. "On Hierarchical Structure Adaptation and Systems Identification," *International Journal of Control,* 20(3):433-552 (September 1974).

13. General Dynamics, Fort Worth Division, "DWG 711ZC001 - Rev A," September 1990.

14. Gierke, Henning E., et al. *Stochasic Models, Estimation, and Control, II, Book 1,* 355-405. Washington, D.C.: Scientific and Technical Information Office, National Aeronautics and Space Administration, 1975.

15. Griffin G. C. Jr. and P. S. Maybeck. "MMAE/MMAC Techniques Applied to Large Space Structure Bending with Multiple Uncertain Parameters," *Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA,* 1153-1158 (December 1995).

16. Gustafson, J. A. and Peter S. Maybeck. "Flexible Spacestructure Control Via Moving-Bank Multiple Model Algorithms," *IEEE Transactions on Aerospace and Electronic Systems, 30*:750-757 (July 1994).

17. Hanlon, Captain Peter D. *Failure Identification Using Multiple Model Adaptive Estimation for the Lambda Flight Vehicle.* MS Thesis, AFIT/GE/ENG/92D-19, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1992.

18. Kyger, Lieutenant David W. *Reducing Lag in Virtual Displays Using Multiple Model Adaptive Estimation.* MS Thesis, AFIT/GE/ENG/95D-11, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.

19. Lane, Captain David W. *Multiple Model Adaptive Estimation Applied to the LAMBDA URV for Failure Detection and Identification.* MS Thesis, AFIT/GA/ENG/93D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1993.

20. Lane, Capt David W. and Peter S. Maybeck. "Multiple Model Adaptive Estimation Applied to the LAMBDA URV for Failure Detection and Identification," *Proceedings of the IEEE Conference on Decision and Control, Lake Buena Vista, FL,* 678-683 (December 1994).

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

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

23. Martin, Capt Richard Maurice. *LQG Synthesis of Elemental Controllers for AFTI/F-16 Adaptive Flight Control.* MS Thesis, AFIT/GE/ENG/90D-36, School of Engineering Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1990.

24. The Math Works, Inc. *MATLAB (registered trademark) (Version 4.2c Edition),* November 1994.

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

26. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, I.* New York: Academic Press, Inc., 1979.

27. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, II.* New York: Academic Press, Inc., 1982

28. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, III.* New York: Academic Press, Inc., 1982.

29. Maybeck, Peter S. and Capt Peter D. Hanlon. "Performance Enhancement of a Multiple Model Adaptive Estimator," *IEEE Transactions on Aerospace and Electronic Systems, 31(4)*:1240-1254 (October 1995).

30. Maybeck, Peter S. and Karl P. Hentz. "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *Journal of Guidance and Control, 10(1)*:90-96 (January-February 1987).

31. Maybeck, Peter S. and Capt Donald S. Pogoda. "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," *Advances in Robust Adaptive Control,* Edited by Kumpati S. Narena, Romeo Ortega, and Peter Dorato; IEEE Press, 1992.

32. Maybeck, Peter S. and Capt Donald S. Pogoda. "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," *Proceedings of the IEEE Conference on Decision and Control, Tampa, FL,* 1566-1572 (December 1989).

33. Maybeck, Peter S. and Steven K. Rogers. "Adaptive Tracking of Multiple Hot-Spot Target IR Images," *IEEE Transactions on Automatic Control,* 28(10) (October 1983).

34. Maybeck, Peter S. and Michael Schore. "Reduced Order Multiple Model Adaptive Controller for Flexible Spacestructure," *IEEE Transactions on Aerospace and Electronic Systems, 28*:756-767 (July 1992).

35. Maybeck, Peter S. and Capt Richard 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).

36. Maybeck, Peter S. and R. I. Suizu. "Adaptive Tracker Field of View Variation Via Multiple Model Filtering," *IEEE Transactions on Aerospace and Electronic Systems, 21(4)*:529-537 (July 1985).

37. Maybeck, Peter S., Professor, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Personal Interviews. January-October, 1996.

38. 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, Wright-Patterson AFB, OH, June 1992.

39. Menke, Timothy E. and Peter S. Maybeck. "Sensor/Actuator Failure Detection in the Vista F-16 by Multiple Model Adaptive Estimation," *IEEE Transactions on Aerospace and Electronic Systems, 31*(4):1218-1229. (October 1995).

40. Nelson, Robert C. *Flight Stability and Automatic Control.* New York: McGraw-Hill, Inc., 1989.

41. Nyhoff, Larry and Sanford Leestma. *Fortran 77 for Engineers and Scientists.* New York: Macmillan Publishing Co., 1988.

42. Pachter, Muir S., Professor, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Personal Interviews. January-October, 1996.

43. Phillips, Maj Scott N. *A Quantitative Feedback Theory FCS Design for the Subsonic Envelope of the VISTA F-16 Including Configuration Variation.* MS Thesis, AFIT/GE/ENG/94D-24, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994.

44. Pogoda, Capt Donald L. *Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures.* MS Thesis, AFIT/GE/ENG/88D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1988.

45. Schaefer, Karl E. *Bioastronautics.* New York: The Macmillan Company, 1964.

46. Schiller G. J. and Peter S.Maybeck. "Space Structure Control Using Multiple Model Adaptive Estimation and Control," *Proceedings of the 1994 IFAC Symposium on Automatic Control in Aerospace - Aerospace Control '94,* Palo Alto, California, pp. 216-221 (September 1994). (Accepted for publication in 1977 in *IEEE Transactions on Aerospace and Electronic Systems*)

47. Stepaniak, 2nd Lt. Michael J. *Multiple Model Adaptive Control of the Vista F-16.* MS Thesis, AFIT/GE/ENG/95D-04, School of Engieering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.

48. 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, Wright-Patterson AFB, OH, December 1989.

49. Strang, Gilbert. *Linear Algebra and Its Applications* (third Edition). San Diego: Harcourt Brace Jovanovich, Publishers, 1988.

50. Stratton, Capt Gregory L. *Actuator and Sensor Failure Identification using a Multiple Model Adaptive Technique for the VISTA/F16.* MS Thesis, AFIT/GE/ENG/91D-53, School of Engineering Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1991.

51. White, Lt Nathan. *MMAE Detection of Interference/Jamming and Spoofing in a DGPS-Aided Intertial System.* MS Thesis, AFIT/GE/ENG/96D, School of Engineering Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1996.

52. WL/FIGXF, WPAFB, OH, "VISTA/NF-16 Technical Data for Customer Usage," August 1991.

Vita

Robert Lewis [redacted]2. In 1980 he was chosen to spend a year in Chile as part of a high school foreign exchange program. While in Chile he met Elizabeth Sanchez, who would become his wife after his graduation from Carpinteria High school in 1981. In 1983 he enlisted in the Air Force where he was chosen to attend the Airman Education and Commissioning program in 1987. Though this program he obtained a Bachelor of Science degree in Electrical Engineering at New Mexico State University and received the Outstanding Graduate from the School of Engineering award in 1990. He was selected in 1995 to attend the Air Force Institute of Technology to pursue a Masters degree. Following Graduation Robert, his wife, and their three children will be assigned to Holloman AFB, NM, where he will be assigned to the 586th test group.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1996	Master's Thesis

4. TITLE AND SUBTITLE

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL RE-DISTRIBUTION FOR THE VISTA F-16

5. FUNDING NUMBERS

6. AUTHOR(S)

Robert W. Lewis
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/96D-29

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Lt Christina Osmon
WL/FIGS
2210 Eighth St STE 11
Wright-Patterson AFB, OH 45433-7521

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Performance characteristics of a Multiple Model Adaptive Estimation and Control Redistribution (MMAC/CR) algorithm are evaluated against single and double actuator and sensor failures. MMAE alone can compensate for sensor failures, whereas Control Redistribution compensates for actuator failures by redistributing commands, initially intended for failed actuators, to the unfailed actuators in such a way that the desired system response is achieved. Both failure detection and compensation capabilities are developed and analyzed through an extensive amount of simulation data, particularly addressing multiple failures. Results are presented which indicate the techniques incorporated provide an excellent means of both failure detection and compensation for the failures of both actuators and sensors. Approximately 98 percent of all secondary failures were successfully detected, and the majority of these detections are shown to occur in less than .5 seconds. The techniques of Multiple Model Adaptive Estimation and Control Redistribution are shown to complement each other well by providing improved failure detection in the face of actuator failures through the redistribution of the dither signal (used to enhance identifiability when there are no maneuvering commands to excite the system:), and improved control authority through enhanced state variable estimation in the face of sensor failures. Control Redistribution is also shown to be an effective tool for modifying the dither to be applied once some actuators have already failed.

14. SUBJECT TERMS

Multiple Model Adaptive Estimation, MMAE, Kalman Filter, F-16,
Control Redistribution, Flight Control, Failure Detection, Reconfigurable Control

15. NUMBER OF PAGES

245

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