Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations Student Graduate Works

3-2002

Flight Control Failure Detection and Control Redistribution Using Multiple Model Adaptive Estimation with Filter Spawning

Michael L. Torres Jr.

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Navigation, Guidance, Control and Dynamics Commons

Recommended Citation

Torres, Michael L. Jr., "Flight Control Failure Detection and Control Redistribution Using Multiple Model Adaptive Estimation with Filter Spawning" (2002). *Theses and Dissertations*. 4462.
https://scholar.afit.edu/etd/4462

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.

Flight Control Failure Detection and Control Redistribution Using Multiple Model Adaptive Estimation with Filter Spawning

THESIS

Michael L. Torres Jr., Captain, USAF

AFIT/GE/ENG/02M-25

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Report Documentation Page

Report Date 11 Mar 02	**Report Type** Final	**Dates Covered (from... to)** Aug 00 - Mar 02

Title and Subtitle Failure Detection and Flight Control redistribution using Multiple Model Adaptive Estimation with Filter Spawning	**Contract Number** **Grant Number** **Program Element Number**
Author(s) Capt Michael L. Torres, Jr., USAF	**Project Number** **Task Number** **Work Unit Number**
Performing Organization Name(s) and Address(es) Air Force Institute of Technology 2950 P Street, Bldg 640 WPAFB OH 45433-7765	**Performing Organization Report Number** FIT/GE/ENG/02M-25
Sponsoring/Monitoring Agency Name(s) and Address(es) ASC/GRA ATTN: Dr. Brian Kowal AMC-III Bldg 558 2590 Loop Road West WPAFB, OH 45433-7142	**Sponsor/Monitor's Acronym(s)** **Sponsor/Monitor's Report Number(s)**

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
The original document contains color images.

Abstract

In the current research, the MMAE with Filter Spawning and Control Redistribution (MMAE/FS/CR) are used together to identify failures and apply appropriate corrections. This effort explores the performance of the MMAE/FS/CR in different regions of the flight envelope using model and gain scheduling. The MMAE/FS/CR is able to detect complete and partial actuator/surface failures, as well as complete sensor failures. Once the actuator/surface failure is identified and the effectiveness is determined in the case of partial failures, proper control is applied in order to accomplish the desired pilot command. Improvements in the algorithm are required in order to enhance the MMAE/FS/CR ability to detect failures while undergoing maneuvering flight. This investigation shows the ability of the MMAE/FS to detect failures while transitioning through the flight envelope and while performing pilot commanded maneuvers. It also improves and demonstrates the CR ability to compensate for complete or partial actuator/surface failures.

Subject Terms

Multiple Model Adaptive Estimation, Kalman Filter, Reconfigurable Flight Controls, Adaptive Flight Controls, Sensor and Actuator Failure Detection

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 259	

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT/GE/ENG/02M-25

Flight Control Failure Detection and Control Redistribution Using Multiple Model Adaptive Estimation with Filter Spawning

THESIS

Presented to the Faculty of the Graduate School of Engineering of the Air Force Institute of

Technology, Air University, in partial fulfillment for the degree of

Masters of Science

specialization in: Electrical Engineering

Michael L. Torres Jr., B.S.E.E.
Captain, USAF

Air Force Institute of Technology

Wright-Patterson AFB, Ohio

March, 2002

Approved for public release; distribution unlimited

AFIT/GE/ENG/02M-25

Flight Control Failure Detection and Control Redistribution Using Multiple Model Adaptive Estimation with Filter Spawning

Michael L. Torres Jr., B.S.E.E.
Captain, USAF

Approved:

Peter S. Maybeck
Dr. Peter S. Maybeck
Committee Chairman

8 March 2002
Date

M. Pachter
Dr. Meir Pachter
Committee Member

March 8, 2002
Date

David Jacques
Lt. Col. David Jacques
Committee Member

8 MAR 02
Date

Acknowledgments

I would like to thank several people who have made this thesis effort possible. I would like to thank Dr. Peter Maybeck for his trust, guidance, and support that kept me on course through this thesis effort. I would also like to thank my committee members Dr. Meir Pachter and Lt. Col. David Jacques for their help as well. I would also like to thank Don Smith for keeping the computers working, particularly 'curly'. Most importantly, I would like to thank my wife for standing by me through this time and for giving me the gift of flight.

Michael L. Torres Jr.

Table of Contents

List of Figures

List of Tables

Abstract

Failure detection for aircraft sensor and actuator/actuating-surface systems and flight control redistribution in response to failed actuators/surfaces is an important research area. With the advent of faster flight control computers with greater memory available, parallel adaptive estimators can now be used to determine sensor and flight control actuator/surface failures. Since most of the sensors are somewhat functionally redundant, when a sensor is lost, the information it would have provided could be estimated from other operational sensors using a Kalman Filter. If a flight control actuator or surface fails, then a system having access to all the other flight control actuators/surfaces can properly adjust the remaining control actuators/surfaces (i.e., employ control redistribution) in order to achieve the commanded maneuver.

Research has been accomplished in this area using Multiple Model Adaptive Estimation (MMAE) to detect failures. Flight Control Redistribution (CR) is then used to compensate automatically for a failed flight control actuator/surface assuming that there is enough control authority in the aircraft. The MMAE is used to detect and identify flight control failures and also to provide estimated sensor outputs in the event of a failed sensor. Further research has been accomplished looking at single and multiple failures and MMAE with Filter Spawning (FS) to detect complete and partial failures. Flight control redistribution has also been researched.

In the current research, the MMAE with Filter Spawning and Control Redistribution (MMAE/FS/CR) are used together to identify failures and apply appropriate corrections. This research effort explores the performance of the MMAE/FS/CR in different regions of the flight envelope using model and gain scheduling.

The MMAE/FS/CR is able to detect complete and partial actuator/surface failures as well as complete sensor failures. Once the actuator/surface failure is identified and the effectiveness is determined, proper control is applied in order to accomplish the desired pilot command.

Improvements in the algorithm are required in order to enhance the MMAE/FS/CR ability to detect failures while undergoing maneuvering flight.

This research effort shows the ability of the MMAE/FS to detect failures while transitioning through the flight envelope and while performing pilot commanded maneuvers. This research also improved and demonstrated the CR ability to compensate for complete or partial actuator/surface failures.

Flight Control Failure Detection and Redistribution Using Multiple Model Adaptive Estimation With Filter Spawning

Chapter 1 – Introduction

1.1Chapter Overview

This thesis presents a flight control failure detection and control redistribution method using multiple model adaptive estimation [27,28] with filter spawning [11,12], combined with a flight control redistribution network [32,33] to respond appropriately to detected flight control actuator/actuating-surface failures. This thesis will show the capability of detecting complete and partial failures of flight control actuators/surfaces and aircraft sensors and the application of the proper control based on the commanded maneuver anywhere in the Variable In-flight Stability Test Aircraft (VISTA) F-16 flight envelope. This chapter presents the motivation behind this thesis and defines the problem. A summary of assumptions made for this thesis effort is also presented in this chapter. Finally, this chapter describes the general format of the remainder of the thesis document.

1.2Motivation

The United States Air Force recognizes the requirement of a flight control system that is tolerant of failures in both flight control actuators/surfaces and sensors. The flight control system should be able to detect partial and complete failures in both the flight control actuators/surfaces and sensors. Upon discovery of a failed surface or sensor, the flight control system should adapt to the available operational surfaces and adjust the flight control commands accordingly to accomplish the commanded aircraft maneuver through unfailed actuators/surfaces. This is especially important to Uninhabited Aerial Vehicles (UAV) in which there is no pilot aboard the aircraft to determine flight control problems and apply the appropriate corrections. This is also critical to pilot-operated vehicles since this failure detection and control system operates without the need for pilot interaction and, when required, can minimize the pilot workload by automatically compensating for the failed actuator or sensor.

Multiple Model Adaptive Estimation with Filter Spawning and Control Redistribution (MMAE/FS/CR) is chosen because of its capability to detect sensor and flight control failures rapidly and correctly, make estimations using all information available, and supply appropriate corrective control performance in the face of a partial or complete actuator/surface failure. The Filter Spawning (FS) algorithm introduces additional filters based on partial failure assumptions, once a specific actuator/surface has been identified as having suffered some level of failure, in order to identify the effectiveness of the partially failed actuator/surface. The FS portion of the MMAE/FS/CR has been shown to identify partial and complete actuator/surface failures, and single and dual failures correctly. By using the FS algorithm, the percentage of actuator/surface effectiveness is determined. This knowledge can assist in the decision making process whether to increase the command signal gain to a surface or to redistribute the command signal. The knowledge of the failure source is supplied by the MMAE and FS supplies the percentage of effectiveness of the failed system. Control commands from the flight control system are then increased in gain or redistributed to the remaining operational actuators/surfaces in order to accomplish the commanded maneuver in any position in the flight envelope. The flight envelope is a region defined by dynamic pressure. The aircraft's linearized equations of motion used in the MMAE/FS/CR are based on subsets of the flight envelope. As the aircraft changes airspeed or altitude, the dynamic pressure changes. A flight control failure detection and control system must operate properly over the entire flight envelope. This is typically done with flight control gains for specific regions of the flight envelope based on the aircraft's location within the flight envelope. This method is called gain scheduling. Model scheduling is similar to gain scheduling except the parameters for the MMAE are used based on the aircraft's location within the flight envelope. It has been shown that model scheduling is essential to demonstrate the MMAE/FS/CR performance completely [9,10]. If model scheduling is not used, after a few seconds of maneuvering flight, all the models within the MMAE's elemental filters become mismatched, incapacitating the MMAE's adaptation.

1.3Problem Statement

A great deal of research into the detection and estimation of single and dual, partial and complete, flight control actuator/surface and complete sensor failures has been accomplished on the VISTA F-16 simulation [7,9-12,17,27,28,32,33]. Adequate detection performance has been achieved in the determination of partial and complete, single and dual failures at one point in the flight envelope. This research intends to demonstrate the capability of the MMAE/FS/CR to detect single, partial and complete, flight control actuator/surface and complete sensor failures in various regions of the flight envelope and with full vehicle maneuvering.

1.4Assumptions

The VISTA F-16 is used to provide the "real world" data for this research. The VISTA F-16 Simulation Rapid-prototyping Facility (SRF) is a six-degree-of-freedom simulation tool developed by Calspan and General Dynamics [11]. The VISTA F-16 simulation provides the "truth model", and it is assumed that the outputs of the simulation are characteristic of the VISTA F-16 test aircraft.

The flight control failure modes used in this research are "failure to free stream", meaning a failed flight control surface does not contribute to the maneuvering of the aircraft. This failure mode is used in the "truth model" and the "design model" and is characteristic of flight control failures encountered in the "real world".

1.5Thesis Format

This thesis is divided into six chapters. This first chapter provides an introduction and problem definition. Chapter two presents the history, theory, and modifications of MMAE/CR with filter spawning for online implementation. Chapter three presents the “truth model” and the construction of the “design model”. Chapter four presents MMAE/FS as it is used for this research. Chapter four also presents the design for a MMAC controller and the CR algorithm. Chapter five presents the performance of the MMAE/FS/CR given single, partial and complete, flight control actuator/surface failures and complete sensor failures at different regions of the flight envelope and different degrees of maneuvering. Chapter six reviews the initial problem and interprets the results based on the research, yielding conclusions and recommendations for further research

Chapter 2 – Concept Development

2.1 Chapter Overview

In this chapter, the Multiple Model Adaptive Estimation using Filter Spawning and Control Redistribution (MMAE/FS/CR) concepts are covered in detail. This chapter is broken into two sections; the first deals with the MMAE with Filter Spawning (MMAE/FS) and the second deals with the CR. Each section provides an overview of the history, the theory, and practical implementation. This layout presents the material so that the reader may clearly understand the ideas and theory while also understanding the historical buildup of the theory and implementation.

2.2 Multiple Model Adaptive Estimation and Control History

2.2.1 Early Contributions

The MMAE was first introduced in 1965 by Magill [20]. Magill proposed the use of several elemental filters based on different hypotheses of 'real world' characteristics. The estimates of each elemental filter are weighted and summed together based on the conditional probability assigned to each elemental filter to form a blended estimate. The elemental conditional probability is the probability that a certain elemental filter is correct, conditioned on the observed measurements. Lainiotis first presented the concept of Multiple Model Adaptive Control (MMAC) in 1973 [15]. This algorithm is similar to the MMAE except elemental controllers are cascaded with the elemental filters. The elemental control commands are then weighted by the conditional probabilities and then summed together to form blended control commands. These ideas only existed in theory at this time because they required extensive computing power to carry out the parallel operations of the elemental filters and controllers. The MMAE and MMAC also suffered from algorithm problems that drove the conditional probability of certain elemental filters to zero, making the MMAE/MMAC insensitive to later environmental changes.

In 1978, Chang and Athans [6] researched the optimality of the MMAE. They found the MMAE is only optimal if the 'real world' parameter space exactly matched the parameter space of the elemental filters. Since this condition requires an infinite amount of elemental filters, the MMAE is suboptimal for estimation. Baram [2, 3, 4] showed that, for discretized spaces, the MMAE would converge (under specific assumptions) to the one discrete parameter value that is closest to the true parameter value in the "Baram distance measure" sense. Tugnait [35] also showed the suboptimal performance of the MMAE with Markov-1 processes. Fry and Sage [13] investigated the used of a hierarchical structure in order to reduce the computational loading of the MMAE.

2.2.2 Early Implementation

In 1977, Athans *et al.* [1] presented the first practical application of MMAC with aircraft. The MMAE/MMAC is used to determine the flight condition of a fly-by-wire F-8C aircraft. Two algorithm enhancements with the MMAC were identified in this research effort. The first is the application of a probability lower bound. In the MMAC, some of the elemental conditional probabilities went to zero, making the MMAC insensitive to later changes involving those particular elemental filters. A probability lower bound was applied that causes the conditional probability to be a non-zero number at all times, thereby precluding this "lockout" phenomenon. The second enhancement used in this research is using a weighted average, or Bayesian method, to form the control signal for the MMAC. This is used instead of selecting the single elemental controller with the highest conditional probability and ignoring the rest of the control signals (Maximum A Posteriori, or MAP, method). Two problems with the MMAE were found in this research. The first problem is Beta Dominance and the second is the requirement of a dither signal, both of which will be explained in detail later in this chapter. This research showed the implementation of MMAE and MMAC, but the aircraft selected for this research was inherently stable. This makes the advanced control not noteworthy and the flight control system did not allow the full implementation of the MMAE/MMAC.

2.2.3 AFIT Contributions

The Air Force Institute of Technology (AFIT) has made several advancements in the implementation of MMAE and MMAC for the detection of flight control failures and control of aircraft with failures. The use of the MMAE algorithm has been explored by AFIT in order to detect sensor and flight control actuator/surface failures on current fighter aircraft. The MMAE has been implemented on digital simulations of the Short-field Take-Off and Landing (STOL) F-15 test aircraft [24,26,31,34], unmanned aerial vehicles [14,16], and the VISTA F-16 [7,9-12, 17,27,28,32,33]. The research started with the detection of single full failures and currently includes exploration into detection of single and dual sensor failures and partial and complete flight control actuator/surface failures.

In 1985, Maybeck and Suizu first explored the Beta Dominance problem with the MMAE (for a different application). Beta Dominance can cause false declarations, as will be explained later in the thesis. Through their research, it was found that removing the Beta term increases the reliability of the MMAE by reducing the amount of false declarations on sensor failures without negatively affecting the MMAE estimates.

Maybeck and Hentz also explored the moving filter bank concept at AFIT in 1987 [25]. The use of a hierarchical structure was also explored [11,12].

AFIT made several contributions to flight control failure detection using the MMAE/MMAC. Pogoda conducted the first implementation of MMAC to detect sensor and flight control failures on the STOL F-15 in 1988 [24,31]. In 1989, Stevens [26,34] used the MMAC with and without the Beta term removed to detect complete sensor and flight control actuator failures on the STOL F-15. He combined the elemental filter estimates using the

Bayesian method. Stevens also investigated scalar residual monitoring and in 1993, Menke [27, 28] continued this research using the VISTA F-16 using MMAE-based control rather than the previously explored MMAC control. Menke also used the full flight control system of the VISTA F-16 as compared to previous research conducted with only the longitudinal channel of the STOL F-15. Menke used the Dryden wind model researched by Pogoda for the F-16 VISTA aircraft in a low dynamic pressure (0.4 Mach and 20,000 feet) environment. This environment was chosen because sensor and flight control failures are hardest to detect in this region of the flight envelope. Many types of sensor failures were tested by Menke, such as loss of signal, increased noise, and bias. From this research effort, it was found that a dither signal is required in order to detect some failures unambiguously in the absence of pilot commands.

Eide [9,10] first implemented the MMAE on the six-degree-of-freedom VISTA F-16 SRF simulation in 1994. The SRF is a simulation that includes the nonlinear dynamics of the longitudinal and lateral/directional channels. The simulation includes the complete F-16 Block 40 Flight Control System (FCS) which includes the aileron-rudder interconnect. The simulation tool also includes the capability to observe sensor data and flight control actuator commands and also allows the direct command of the flight control actuators. This is essential for detecting failures and applying appropriate control. Eide discovered that a MMAE equipped with models based on one location within the flight envelope are valid for only a small neighborhood about that point in the flight envelope, before all elemental filters look wrong, incapacitating the adaptation mechanism of the MMAE. This drives the requirement for model scheduling based on the dynamic pressure for any investigation of substantial aircraft maneuvering, as is the intent of this research.

Stepaniak [32,33] attempted to apply the first MMAC controller to the SRF VISTA F-16 in 1995. Stepaniak explored the use of Linear Quadratic Gaussian (LQG) synthesis in order to design the elemental controllers within the MMAC. The design intention was to replicate the Block 40 flight control system used by the VISTA F-16 via LQG CGI/PI (Command Generator

Tracking with Proportional plus Integral) synthesis. The resulting weighting matrices would have then been used to design the LQG elemental controllers to handle the conditions of a failed sensor or actuator/surface. He found that the Block 40 FCS could not be adequately reproduced using LQG synthesis methods. Stepaniak then designed a single controller that redistributes the Block 40 flight control system actuator control commands to the appropriate unfailed actuators based on the estimates and failure status provided by the MMAE. This method is called Control Redistribution (CR). He refined the required dither signal and also completely demonstrated the MMAE's capability to detect and compensate for complete failures of any sensor or flight control actuator/surface.

In 1996, Lewis [17] researched the use of a redistributed dither signal upon declaration of a failed flight control actuator/surface. He also demonstrated the MMAE's capability of detecting single and dual complete failures of any sensor or flight control actuator/surface combinations. Clark [7] researched the detection of partial failures in 1997. He blended the estimates of elemental filters based on the hypothesis of a fully functional aircraft and a completely failed flight control actuator in order to handle partially failed actuators. The inability of this approach to identify properly the percentage of effectiveness of partially failed filters motivated the idea of the FS algorithm.

Fisher [11, 12] developed the concept of filter spawning in 1999. This method uses an initial set of elemental filters to detect a sensor or flight control actuator/surface failure (assigning only "fully functional aircraft" and "fully failed sensor or actuator" hypotheses). Once an actuator/surface failure is detected, several more elemental filters are brought online or spawned. Each of the spawned filter models is based on different assumed values of flight control actuator effectiveness percentage. By using this approach, Fisher demonstrated the MMAE's capability to *detect* single failures and also determine the effectiveness of a partially failed flight control actuator/surface. The current research seeks to combine the performance of the MMAE with filter spawning as developed by Fisher, with the CR developed by Stepaniak. The current research then seeks to prove the effectiveness of this combination in various locations within the flight envelope of the VISTA F-16. Different levels of maneuvering are also examined in this research, using model and gain scheduling to handle the effects of maneuver commands causing the aircraft to transverse a path of points through the flight envelope.

2.3 MMAE Theory

The MMAE is a structure made of several Kalman filters operating in parallel used to provide an estimate given a substantially varying ‘real world’: a ‘real world’ described by parameter values or hypothesized conditions that can vary over a range of possible values. The structure of the MMAE is discussed in Section 2.3.1. The Kalman filter provides an optimal state estimate based on the ‘real world’ inputs and measurements, and the Kalman filter’s internal model, and this is discussed in detail in Section 2.3.2. The conditional probability generator computes the relative probability of the correctness of each elemental filter at the current time, conditioned on the measurements observed from the ‘real world’. The conditional probability generator is discussed in Section 2.3.3. Section 2.3.4 shows that the MMAE is guaranteed to converge onto one of the hypotheses under certain conditions, and Section 2.3.5 discusses the formulation of the state estimate from the MMAE. The hierarchical structure used for handling multiple failures without impractical computational burden is shown in Section 2.3.6, and filter spawning for addressing partial actuator/surface failures is explained in detail in Section 2.3.7.

2.3.1 Multiple Hypothesis Models

The MMAE is based upon multiple hypothesis models. MMAE is effective in a ‘real world’ that has changing parameters. If only one filter is used in this case, the Kalman filter state estimates would contain large errors since the linearized model is valid for one set of parameters. The MMAE uses several Kalman filters in parallel, each with a model corresponding to a different ‘real world’ hypothesis. When the conditions of the ‘real world’ change, such as a flight control actuator/surface failure, the fully functional aircraft hypothesis becomes invalid, while the appropriate hypothesis postulating the flight control actuator/surface failure becomes correct. The part of the MMAE that makes the determination of relative model adequacy is the conditional probability generator. By examining the residuals of each of the elemental filters and the expected covariance of the residual vector, the probability that an elemental filter is currently correct can be determined. The conditional probability is used as a multiplication factor for each elemental filter estimate and then the weighted estimates are summed together to form one MMAE estimate in a Bayesian (conditional mean estimation) approach. Alternatively, a Maximum A Priori (MAP) approach (computing a conditional mode instead) can be applied to the MMAE by using only the estimate from the elemental filter with the highest computed conditional probability. The individual components of the MMAE are described in detail in the following sections. The structure of the MMAE using the Bayesian approach is shown in Figure 1.

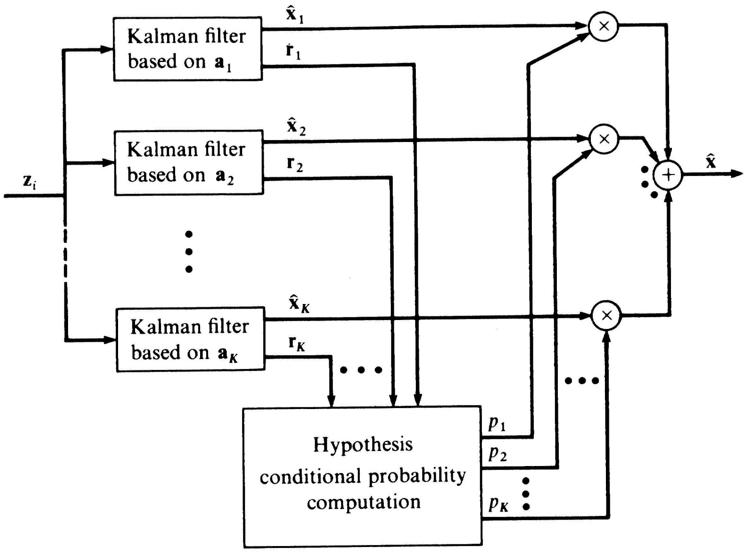


Figure 1: MMAE Structure [22]

2.3.2 Kalman Filter

The Kalman filter is an optimal recursive state estimator for a dynamic system adequately represented as a linear system driven by white Gaussian noises. The Kalman filter has two stages. The first stage is the propagate stage. This is where a mathematical model is used in the Kalman filter to compute a predicted state estimate. The predicted state error covariance is then calculated based on the covariance of the previous state estimate (right after the most recent measurement) and the assumed characteristics of the dynamics driving noise in the model. The second stage of the Kalman filter is the update stage. The Kalman gain is calculated using the predicted state estimate error covariance and the measurement noise covariance. The Kalman gain is used to weigh the incoming measurement and the predicted state estimate, to form an optimal updated state estimate. The Kalman gain is then used to update the error covariance of the filter state estimate as well.

The model used in the Kalman filter is linear. The discrete-time form of the filter design model is [22]

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}(t_{i+1},t_i)\, \mathbf{x}(t_i) + \mathbf{B}_d(t_i)\, \mathbf{u}(t_i) + \mathbf{G}_d(t_i)\, \mathbf{w}_d(t_i) \tag{2.1}$$

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\, \mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.2}$$

where $\mathbf{\Phi}(t_{i+1},t_i)$ is the discrete-time linear model state transition matrix used to propagate the state estimate, $\mathbf{x}(t_i)$, from one sample time to the next. $\mathbf{B}_d(t_i)$ is the control matrix that relates the input control vector, $\mathbf{u}(t_i)$, to the state vector. $\mathbf{G}_d(t_i)$ is the matrix that relates the dynamic driving noise, $\mathbf{w}_d(t_i)$, to the state vector. $\mathbf{H}(t_i)$ is the discrete-time matrix that relates the state vector to the measurement vector, $\mathbf{z}(t_i)$, and $\mathbf{v}(t_i)$ is the measurement corruption noise. The statistics of the white Gaussian noises are

$$\begin{array}{ll} E[\mathbf{w_d}(t_i)] = \mathbf{0} & E[\mathbf{w_d}(t_i)\mathbf{w_d}^T(t_j)] = \mathbf{Q_d}\boldsymbol{\delta}_{\mathbf{ij}} \\ E[\mathbf{v}\,(t_i)] = \mathbf{0} & E[\mathbf{v}(t_i)\mathbf{v}^T(t_j)] = \mathbf{R}\boldsymbol{\delta}_{\mathbf{ij}} \\ \multicolumn{2}{c}{E[\mathbf{w_d}(t_i)\mathbf{v}^T(t_j)] = \mathbf{0}} \end{array} \tag{2.3}$$

The algorithm used for the propagation of the Kalman filter forward in time is

$$\hat{\mathbf{x}}(t_{i+1}^-) = \mathbf{\Phi}(t_{i+1},t_i)\hat{\mathbf{x}}(t_i^+) + \mathbf{B}_d(t_i)\mathbf{u}(t_i) \tag{2.4}$$

$$\mathbf{P}(t_{i+1}^-) = \mathbf{\Phi}(t_{i+1},t_i)\mathbf{P}(t_i^+)\mathbf{\Phi}^T(t_{i+1},t_i) + \mathbf{G}_d(t_i)\mathbf{Q}_d(t_i)\mathbf{G}_d^T(t_i) \tag{2.5}$$

and the algorithm used for the measurement update of the Kalman filter is

$$\mathbf{A}(t_i) = \mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}(t_i)^T + \mathbf{R}(t_i) \tag{2.6}$$

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}(t_i)^T\mathbf{A}^{-1}(t_i) \tag{2.7}$$

$$\mathbf{r}(t_i) = \mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-) \tag{2.8}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\,\mathbf{r}(t_i) \tag{2.9}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{2.10}$$

The superscripts – and + indicate the value before and after the time the measurement is taken, respectively. The Kalman filter gain $\mathbf{K}(t_i)$ is calculated using the previous covariance matrix $\mathbf{P}(t_i^-)$ the measurement matrix $\mathbf{H}(t_i)$ and the measurement noise covariance matrix $\mathbf{R}(t_i)$. The $\mathbf{r}(t_i)$ is the residual vector and is the difference between the measurement at the current time and the predicted measurement using the measurement matrix and the predicted state vector $\hat{\mathbf{x}}(t_i^-)$. The Kalman filter gain is then used to update the state estimate and the state error covariance.

The residual vector is zero-mean and well described by the precomputed residual covariance $\mathbf{A}(t_i)$, as long as the linear model used in the Kalman filter is valid. If the 'real world' differs substantially from the linear model used in the Kalman filter, a bias or increased magnitude will appear in the residuals and the residual values will be larger than anticipated by the precomputed residual variance. Another way to describe this is, if the value of $[\mathbf{r}(t_i)^T\,\mathbf{A}(t_i)^{-1}\,\mathbf{r}(t_i)]$ is approximately equal to the number of measurements, then the hypothesis of the linear model is correct. If this value is much greater than the number of measurements, then the hypothesis model is incorrect.

The Kalman filter provides an optimal state estimate $\hat{\mathbf{x}}(t_i^+)$, and a filter-computed state covariance $\mathbf{P}(t_i^+)$ given a measurement history up through time t_i. The Kalman filter also provides a residual vector and a filter-computed residual covariance which are required to use the MMAE algorithm.

2.3.3 Conditional Probability Generator

The conditional probability generator determines the conditional probability of each elemental filter's estimates being correct. Each elemental Kalman filter provides a residual vector, $\mathbf{r}_k(t_i)$, and a residual covariance vector, $\mathbf{A}_k(t_i)$, where the k subscript corresponds to k^{th} elemental Kalman filter. Residual monitoring is inherent in the algorithm: real-time residuals are used to determine the correct hypothesis.

The conditional probability that the k^{th} elemental filter matches the 'real world', conditioned on the measurement history, $\mathbf{Z}_i$, is shown as

$$p_k(t_i) \;=\; \text{prob}\{\mathbf{a}=\mathbf{a}_k \mid \mathbf{Z}(t_i)=\mathbf{Z}_i\} \tag{2.11}$$

The conditional probability computation has been shown to be computable as [22]

$$p_k(t_i) \;=\; \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) \cdot p_k(t_{i-1})}{\sum\limits_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_j, \mathbf{Z}_{i-1}) \cdot p_j(t_{i-1})} \tag{2.12}$$

where the numerator is the density function of the measurement conditioned on the assumed parameter value or elemental filter being correct and on the entire observed measurement history, multiplied by the previously computed conditional probability for the elemental filter. The denominator is a scaling factor used to make the conditional probabilities for all of the elemental filters sum to one; it is simply the sum of all possible numerator terms. The density function above is defined as

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) \;=\; \frac{1}{(2\pi)^{m/2}\,|\mathbf{A}_k(t_i)|^{1/2}}\;\exp\{\cdot\} \tag{2.13}$$

$$\{\cdot\} \;=\; \{-\tfrac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\}$$

where the 'beta term' is

$$\beta_k(t_i) = \frac{1}{(2\pi)^{m/2}\,|\mathbf{A}_k(t_i)|^{1/2}} \tag{2.14}$$

and the likelihood quotient is

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

By using the equations shown above, the conditional density of the elemental filter k can be determined at time t_i by using the previous elemental conditional density and the $\mathbf{r}_k(t_i)$ and $\mathbf{A}_k(t_i)$ calculated by the elemental Kalman filter. If the 'real world' parameters closely resemble those of a certain elemental filter, then the residuals will be zero-mean and the covariance of the residuals will closely resemble the calculated residual covariance. In this case, the likelihood quotient will approach the dimension of the measurement vector, $\mathbf{z}(t_i)$. As the elemental filter becomes mismatched with the 'real world', the residuals will no longer be zero-mean and the covariance of the residuals will be substantially larger than the computed residual covariance. The likelihood quotient also shows this variation by becoming larger than the value of the dimension of the measurement vector.

2.3.4 MMAE Convergence

The convergence of the MMAE to the correct filter has been researched [2-4]. The MMAE has been shown always to converge to the correct filter assuming an elemental filter exactly matches the 'real world' [2-4]. In the case in which the 'real world' does not exactly match any of the elemental filter hypotheses, the hypothesis that most closely matches the 'real world' characteristics in a Baram distance sense will have the highest conditional probability [2-4]. These proofs can guarantee the convergence of the MMAE to a hypothesis but they cannot guarantee the speed at which the MMAE will converge to the hypothesis. It has been shown in previous research [11, 12] that, in the case of aircraft actuator/surface and sensor failure detection, the MMAE converges in a reasonable time (on the order of seconds or less).

2.3.5 MMAE State Estimate

There are two methods by which a state estimate can be made by the MMAE. The first method is the Bayesian method and the second is the Maximum A Posteriori (MAP) method. The Bayesian method uses the condition probability computed by the MMAE to weight the elemental state estimates from the elemental Kalman filters. The weighted estimates are then summed together, or blended, to form one estimate (a conditional mean estimate) that contains information from each of the elemental filters. The MAP method uses only the elemental state estimate of the hypothesis with the highest conditional probability (a conditional mode estimate). The other elemental state estimates are disregarded and the MMAE output only contains the information from the single elemental Kalman filter.

The MAP method can be useful when the MMAE is used in a large parameter space. The MAP method eliminates the estimates that contain large errors due to mismatched filters by only using the estimates from the filter that closely matches the ‘real world’. The Bayesian method is useful for a parameter space in which the elemental filter models are similar or contain similar information. One particularly useful application is to attempt to handle partial actuator/surface failures by blending the estimates of a filter based on the hypothesis of a fully functional aircraft with the estimates of a filter that assumes a fully failed actuator/surface. This allows a coarser discretization of the parameter space than does the MAP form of the MMAE. A mismatched filter can still contribute to the estimate and decrease the MMAE accuracy. In this research, the state estimates will be blended together using the Bayesian method.

2.3.6 Hierarchical Structure

The MMAE is based on a hypothesized set of system parameter vector values. In the case of detecting aircraft failures, the system parameter vector values are a representation of the failure status condition of the aircraft. If only single failures are considered, only 12 elemental filters would be required (one fully functional aircraft filter, five fully failed flight control actuator/surface filters and six total sensor failure filters):

$$A = \{a_1, a_2, \dots , a_K\} = \{a_1, a_2, \dots, a_{12}\}$$

If it is desired to estimate single and dual failures, then 67 filters would be required to run in parallel in order to detect all possible combinations of failures. Each filter would have to be propagated and updated and the conditional probability for each filter would have to be calculated in order to produce a state estimate from the MMAE. This can become computationally very burdensome.

In order to solve this computation problem, the failure sets can be broken up into a hierarchical set of banks. Initially, only 12 filters will be computed online for the detection of a single failure.

$$A_0 = \{a_0, a_1, \dots , a_K\}$$

Once a failure is detected, a second bank of filters is brought online, one of:

$$A_1 = \{a_0, a_1, a_{12}, \dots , a_{1K}\}$$

$$A_2 = \{a_0, a_2, a_{21}, \dots , a_{2K}\}$$

$$\vdots$$

$$A_K = \{a_0, a_K, a_{K1}, \dots , a_{K,K-1}\}$$

The second bank of filters is based on the first failure, once it is declared with assurance, and this second bank is used to detect a second failure. This reduces the computational burden by only requiring a maximum of 12 filters running in parallel. The 12 filters are the original 11 failure detection filters but each with the additional assumption of the first failure already having been declared, and the 12th filter allows the algorithm to reverse back to a fully functional aircraft, a_0, if a false failure declaration had actually occurred. The hierarchical structure is shown in Figure 2 on the next page. The hierarchical structure as presented here is used for the detection of single and dual failures without the requirement running a large number of filters in parallel based on every possible failure combination.

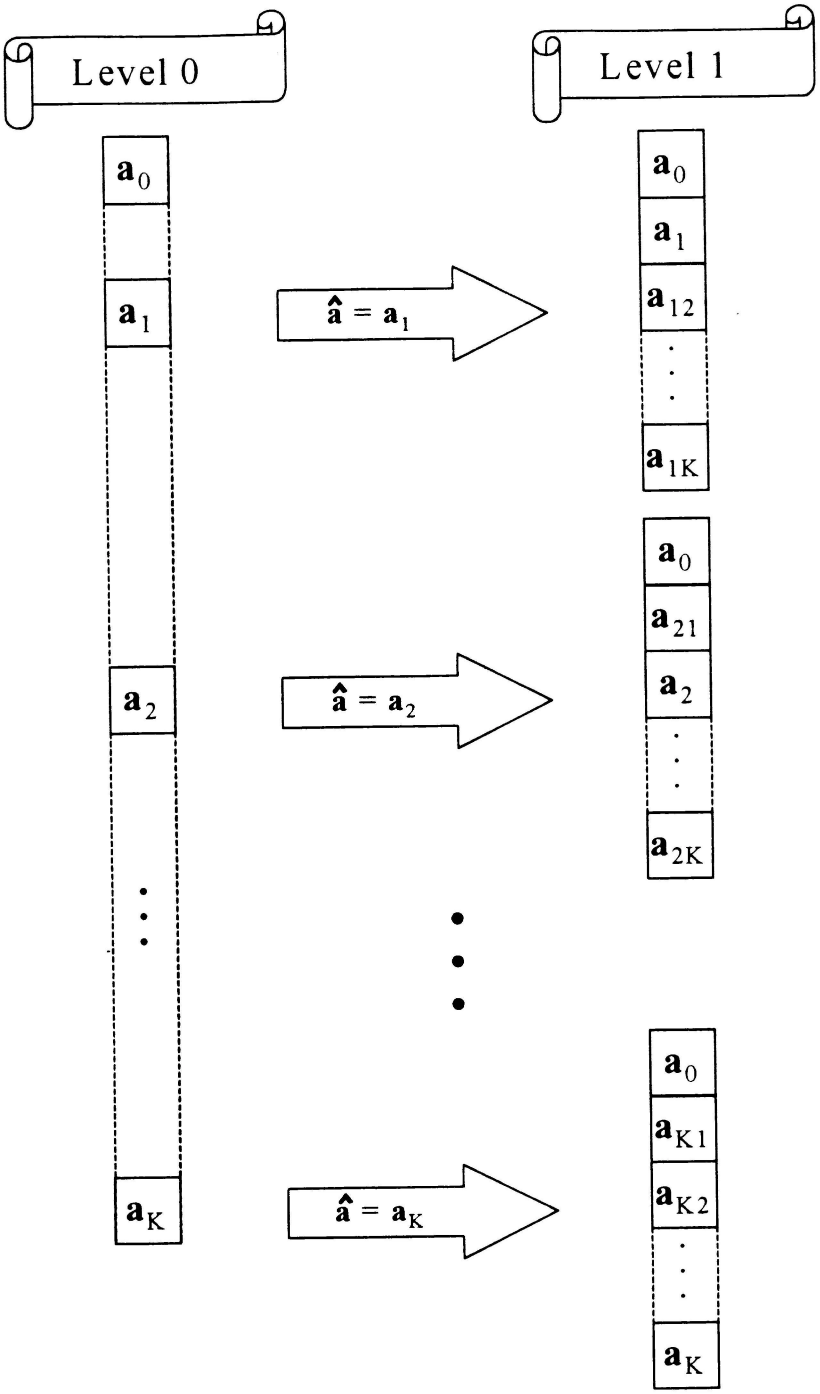


Figure 2: Hierarchical Structure [11]

2.3.7 Filter Spawning

Filter Spawning (FS) is a method used to determine the percent of effectiveness (i.e., 100% minus the percentage of failure) of a failed flight control actuator/surface. It has been shown [11,12] to provide adequate estimations of the flight controls actuator/surface effectiveness once a failure is declared by the MMAE. Once a specific flight control actuator/surface failure is declared, several online filters are spawned. Each filter is based on a specific effectiveness of that particular flight control actuator/surface. By observing the elemental probabilities in the spawned filters and the fully failed filter, a blended estimate of the effectiveness for a failed flight control actuator/surface is determined. This algorithm can be used in conjunction with the hierarchical structure since the hierarchical structure is used to detect failures in individual sensor and flight control actuators/surfaces and the filter spawning algorithm is used to evaluate the effectiveness of a single flight control actuator/surface once a failure declaration is made.

2.4 MMAE in Practice

2.4.1 Probability Lower Bounds

One problem encountered with the original design of the MMAE is the MMAE's convergence to one hypothesis and the condition of all the conditional probabilities of the other hypotheses going to zero over time. Since the previous elemental conditional probability is used to compute the current elemental condition probability (Equation (2.12)), once the elemental conditional probability equals zero, it will remain zero for all time thereafter. This locks the MMAE onto a single hypothesis even though the 'real world' may be changing. It has been found that setting a lower limit to the elemental conditional probabilities solves this problem [1, 22]. The probability lower bound ensures that an elemental conditional probability will always be greater than zero.

Setting the probability lower bound is also challenging. Setting the probability lower bound too low will cause the MMAE to react slowly to changes in the 'real world', while a high probability lower bound can cause the MMAE state estimate to contain a heavily weighted estimate from a mismatched elemental filter, thereby corrupting the MMAE output.

When the Bayesian blending method is used to create estimates, the incorrect estimate from a mismatched filter is still combined with the estimate from the matched filter. This can result in substantial estimate errors from the MMAE. An alternative approach has been used to solve this problem. This approach is the modified Bayesian method. In this method, the elemental conditional probabilities are examined, and when a conditional probability is equal to (or very close to) the probability lower bound, that estimate is removed from the estimation blending used to determine the state estimate and estimate of effectiveness. This means that only the filter estimates with a relatively high conditional probability will be allowed to blend to form the MMAE estimate, without the corrupt estimates from mismatched filters. This allows useful blending but precludes corruption of the state estimates solely because of imposing the artificial lower bound.

2.4.2 Beta Dominance

There has been much research in the area of Beta dominance [1,26-28,34]. Beta dominance is the condition in which the conditional probabilities are incorrectly calculated due to the Beta term in the previous equations (Equation (2.14)) when the likelihood quotients (Equation (2.15)) of all (or many) elemental filters are essentially the same. An example of this is that failed sensor hypotheses yield low computed residual covariances, which in turn yields to a tendency to declare false alarms on sensors because the associated likelihood quotient $[\mathbf{r}_k^{T}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)]$ can readily become large due to small $\mathbf{A}_k$ values. One solution to this problem is the complete removal of the Beta term from the conditional probability equations [11,12]. By taking this step, Equation (2.13) no longer yields a proper Gaussian density function since the area under the function no longer equals to unity. This is not a great concern since the denominator in Equation (2.12) scales the conditional probability so the summation of all conditional probabilities will always equal one.

2.4.3 Dither Signal

A dither signal is required in order to detect some failures unambiguously in the absence of pilot commands. A low magnitude (subliminal) control signal must be introduced into the flight control actuators in order to excite the aircraft sufficiently to detect failures properly while not exciting it so much as to be objectionable to the pilot. Without the dither signal, there does not exist enough observability to detect and disambiguate certain failure modes. For example, if the rudder fails on an aircraft and the pilot is flying straight and level, the failed rudder will only be noticed the first time the pilot requires the rudder. A dither signal would be used to excite the rudder and the rudder failure would be noticed well before the pilot applied a control input. This allows the MMAE to detect the failures and the controller to apply corrective control in lieu of pilot commands.

Various types of dither signals have been introduced [9,10] but only the sinusoidal dither signal provides the best performance with respect to failure detection. The frequency of the sinusoidal dither signal is known and is used to determine failures. Once a surface has failed, the sinusoidal dither signal contributed by that particular surface will not appear. This lack of signal is detected by the residuals of the various elemental filters and the residuals of the elemental filter that correctly hypothesizes the actual failure will look substantially different from the residuals of all the other elemental filters. The MMAE algorithm is then more capable of determining flight control actuator/surface failures. Different dither frequencies can also be used for each flight control surface and by detecting the frequencies through the aircraft sensors, any failed flight control actuator can be determined since its frequency will not be detected by the sensors.

The use of a dither signal must be well planned since the pilot will detect any additional movement in the flight control surfaces. The use of too large of a dither signal disturbs the pilot and a dither signal too small results in missed failure declarations. The dither is only required in the absence of pilot command inputs. Other methods can be used instead of a constant dither signal, such as using a (pilot-commanded) short-term, large magnitude maneuver during times of steady flight. The pilot-commanded maneuver does not have the subliminal requirement as does the dither signal. This means the pilot-commanded maneuver can be a large enough magnitude to allow the MMAE/FS to provide better detection performance compared to the subliminal dither signal. The dither signal used in this research is a sinusoidal dither with a frequency of 15 Hz and +/- 0.1 g's in the longitudinal channel and +/- 0.2 g's in the lateral channel. These dither parameters have been used in previous research efforts and are not truly subliminal to a pilot, but have been found tolerable to pilots in research conducted in the 1960's.

2.4.4 Model Scheduling

This research presents the application of *model scheduling* based on the dynamic pressure. Dynamic pressure is the distribution of force exerted on a control surface by the airflow, and this is dependent on the airspeed of the aircraft and the air density, which changes with altitude. Previous research efforts have explored the operation of the MMAE/FS and CR at only one dynamic pressure point (Mach 0.4 and 20,000 feet). The parameters of the linear models used in the elemental filters are dependent on the dynamic pressure. In order to allow the aircraft to change airspeed and altitude and still maintain detection and estimation performance by the MMAE/FS, stability derivatives and other parameters in the elemental linear filters used in the MMAE/FS are evaluated as a function of dynamic pressure (airspeed and altitude). This is called 'model scheduling' and is applied to the MMAE/FS algorithm in this research effort to allow the aircraft to transition the flight envelope (an area comprised of all dynamic pressures where the aircraft is able to fly) due to purposeful pilot commanded maneuvers.

2.5 Control Redistribution in Theory

Two forms of adaptive control are shown in this chapter. The first adaptive controller discussed is the Multiple Model Adaptive Controller (MMAC). This controller is discussed in Section 2.5.1. The MMAC is made up of parallel controllers, and the LQG synthesis of these controllers is shown in Section 2.5.2. Finally, an alternative approach called Control Redistribution (CR) is discussed in Section 2.5.3.

2.5.1 MMAC Description

The MMAC is built on an MMAE foundation. In a MMAC, there is an elemental controller for each hypothesis (assumed parameter value) of the MMAE. Each elemental controller uses the state estimates formed by the corresponding elemental filter to produce a control signal based on the assumed hypothesis of the 'real world'. The control signals are then blended together using the conditional probabilities as weighting factors. The MMAE filters are the same as described earlier in this chapter and the operation of the conditional probability generator is also the same as in the MMAE. The structure of the MMAC is shown in Figure 3.

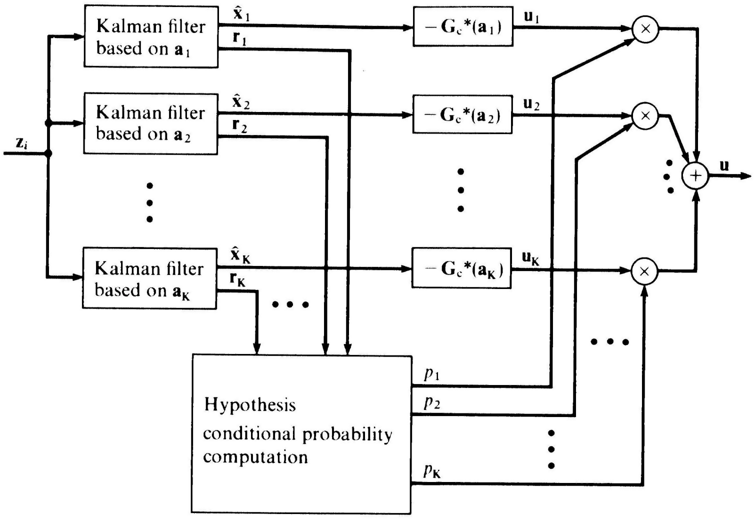


Figure 3: MMAC Structure [23]

2.5.2 LQG Synthesis

The MMAC controllers are designed using the Linear Quadratic Gaussian (LQG) synthesis. This method allows the designer to build a controller and adjust the performance of the controller by adjusting the quadratic cost function in the controller design. The general cost function is shown in Equation (2.16) [23]:

$$J = E\left\{ \sum_{i=0}^{N} \frac{1}{2}\left[\mathbf{x}^T(t_i)\mathbf{X}(t_i)\mathbf{x}(t_i) + \mathbf{u}^T(t_i)\mathbf{U}(t_i)\mathbf{u}(t_i)\right] + \frac{1}{2}\mathbf{x}^T(t_{N+1})\mathbf{X}_f\,\mathbf{x}(t_{N+1}) \right\} \tag{2.16}$$

The quadratic cost function penalizes the controller for applying too much control and for not following the reference input signal closely enough through the $\mathbf{U}(t_i)$ and $\mathbf{X}(t_i)$ weighting matrices. Typically, in flight control problems, the final term representing the cost for not getting to a particular final state, $\mathbf{X}_f$, is dropped, and $N \rightarrow \infty$ to allow for steady-state constant-gain controller design. For example, the cost function can be based on the amount of control applied and the ability of the controller to use the surfaces to perform a given command. By adjusting the cost function, the controller can be designed to perform a command perfectly at the cost of infinite flight control actuator commands at one extreme. The other extreme is zero use of flight control actuators and not performing the commanded task. The cost function is adjusted such that appropriate flight control actuator commands are given in order to perform a specific flight maneuver.

The original design intention for Stepaniak [32,33] was to use the LQG synthesis of a CGT/PI controller to replicate the Block 40 flight control system. The associated weighting matrices were then used to perform the synthesis of controllers for each of the failed actuator conditions. It has been found that a controller designed using LQG synthesis cannot adequately model the F-16 Flight Control System (FCS). This can possibly be attributed to the classical design approach taken to design the actual F-16 FCS. Alternative methods had to be explored in order to provide proper control of the F-16 (specifically MMAE-based control with CR) while using the MMAE to detect failures.

2.5.3 Control Redistribution

Control Redistribution (CR) is the process of applying commands to unfailed actuators/ surfaces in order to compensate for a failed actuator. In this case, the MMAE/FS algorithm provides the failure declaration and an estimation of actuator effectiveness in the case of partial actuator failures. The MMAE creates a state estimate and also decides which hypothesis represents the 'real world' failure status characteristics. The CR controller uses this knowledge to redistribute the control authority to the unfailed control actuators in order to accomplish the desired maneuver. The mathematical formulation of the CR algorithm is presented in detail in Section 4.5.3.

2.5.4 Gain Scheduling

Gain Scheduling is the dependence of the flight control surface commands from the flight control system on the dynamic pressure (combination of airspeed and altitude). As the dynamic pressure changes, the control system must compensate by increasing or decreasing the commands to the flight control surface. In this research effort, the aircraft will be transitioning the flight envelope due to purposeful pilot commanded maneuvers. Since the CR algorithm is used, the Block 40 Flight Control System (FCS) is used. The Block 40 FCS already incorporates gain scheduling to control the aircraft.

2.6 Chapter Summary

This chapter provided the basic background for MMAE/FS and for CR. The developmental history of the MMAE was covered first. The paths that have led to the current MMAE implementation have also been reviewed. The advantages of the MMAE to create state estimates as well as detect failures and the addition of FS allowing the MMAE to produce estimates of failure effectiveness has been presented in this chapter. Finally, the control aspect has also been briefly presented in this chapter.

Chapter 3: Truth and Design Model

3.1 Chapter Overview

This chapter presents the derivation of the truth and design models used in this research effort. Section 3.2 discusses the ‘real world’ and introduces the VISTA F-16 aircraft. The ‘truth model’ used in this research effort is the VISTA F-16 SRF simulation tool and is described in Section 3.3. The derivation of the ‘design model’ is shown in Section 3.4, as well as the modifications to the ‘design model’ in order to allow for realistic actuator responses. Finally, the failure modes are described in Section 3.5.

3.2 Real World

The Variable In-flight Stability Testbed Aircraft (VISTA) F-16 aircraft is a flight control test aircraft used to provide in-flight simulations of high performance aircraft [11]. The VISTA F-16 is a modified F-16D aircraft with a Block 40 flight control system. The modification allows the VISTA F-16 to simulate the responses of various high performance aircraft. This is accomplished by changing the aircraft’s flight control laws in order to achieve the characteristics of the aircraft under test. The advantage of the VISTA F-16 aircraft is the flexibility of the control system which provides a useful tool in the development and evaluation of flight control systems.

The VISTA F-16 aircraft was chosen for this research effort in order to continue the research efforts already accomplished at the Air Force Institute of Technology (AFIT) [7,9-12, 17,27,28,32,33]. The original reason behind the decision to use this aircraft was the availability of advanced computer simulation tools at the Air Vehicles Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH.

3.3 Truth Model

The ‘truth model’ used in this research effort is the VISTA F-16 Simulation Rapid-prototyping Facility (SRF) simulation tool. The SRF is a FORTRAN-based, nonlinear six-degree-of-freedom, computer simulation of the VISTA F-16 aircraft [11]. Calpsan and General Dynamics developed this simulation tool. The simulation program contains the Block 40 flight control system used by the VISTA F-16 aircraft and the flight control actuator dynamics. The Block 40 FCS has saturation (and anti-windup) compensation and the flight control actuator dynamics use 4th order actuator models. The simulation also includes the aileron-to-rudder interconnect used for coordinated turns, and allows for direct access of the aircraft sensors and the flight control actuators, which is a requirement for this research effort.

There have been many modifications made to the VISTA F-16 SRF simulation tool through the history of this research topic [7,9-12,17,27,28,32,33]. The zero-order Dryden wind model was added to the simulation in order to improve the accuracy of the simulation [9,10, 24,31]. Sensor noise was also incorporated into the simulation as well as a more realistic model of the lateral acceleration sensor [32,33]. The Multiple Model Adaptive Estimator with Filter Spawning (MMAE/FS) was added to the simulation to provide the capability of detecting complete actuator/surface and sensor failures and evaluating the effectiveness of partial actuator/surface failures [11,12]. Control redistribution (CR) was added to the output of the flight control portion of the VISTA F-16 SRF in order to accomplish appropriate control once an actuator failure is identified [32,33].

3.4 Design Model

The ‘truth model’ is a high order model that accurately depicts the ‘real world’. The ‘design model’ is a lower order model that is used to model the ‘truth model’ to a reasonable level of fidelity, and it is used in the MMAE/FS/CR algorithm. The order of the ‘design model’ is dictated by the computational limit of the flight control computer. In this case, the flight control computer operates at 64 Hz. The ‘design model’ must be of low enough order such that the resulting flight control system with the MMAE/FS/CR algorithm must operate within the 64 Hz limitation. The ‘design model’ must also accurately represent the ‘real world’, which sets a lower bound on order and complexity of the ‘design model’. The ‘truth model’ does not have the computational limitation that the ‘design model’ has, since the ‘truth model’ does not have to be implemented on the aircraft’s 64 Hz flight control computer. The ‘truth model’ is used as a basis or starting point of the reduced order ‘design model’ and also as a verification tool to evaluate the operation of the adaptive control system based upon the ‘design model’.

3.4.1 Linearized VISTA F-16

The ‘design model’ is created by the linearization of the nonlinear, continuous-time model used by the ‘truth model’ about a nominal point in the flight envelope. The flight envelope is the region of aircraft operation defined by airspeed, measured in Mach number, and altitude, measured in feet above mean sea level (MSL). Previous research efforts have studied the response of MMAE/FS/CR at one point within the flight envelope (Mach 0.4 and 20,000 ft) [7, 9-12,15,24,25,29,30]. This condition was chosen because of the difficulty in detecting sensor and flight control failures due to the low dynamic pressure: the previous research focused on detection and estimation, and so it was prudent to test against conditions under which the detection and estimation would be the most difficult. The aircraft is in the ‘up-and-away’ flight configuration, meaning the leading edge flaps and landing gear are in the up position. In the current research, various points within the flight envelope are tested using model and gain scheduling. The simulation begins at one point within the flight envelope and through the application of pilot commands, the aircraft transitions the flight envelope. This maneuver is used to test the detection and estimation capability of the MMAE/FS using ‘model scheduling’ while transitioning the flight envelope as well as testing the Block 40 ‘gain scheduling’ and CR algorithms. It has been shown in previous research that as the aircraft moves away from the nominal point (even over a few seconds of maneuvering flight), all failure mode models appear bad if they are not all altered as a function of dynamic pressure, thereby incapacitating the MMAE's detection [9,10]. Thus, gain and model scheduling, as a function of dynamic pressure (or altitude and Mach number), is a vital component of the operation of the MMAE/FS/CR for maneuvering aircraft vehicles. As the dynamic pressure changes as a result of altitude or airspeed changes, the filter models in the MMAE/FS/CR are changed to adapt to the new dynamic pressure. This is known as model scheduling. As the dynamic pressure changes, the Block 40 flight control gains are also changed in order to adapt to the new dynamic pressure. This type control action is known as gain scheduling and is already incorporated in to the standard Block 40 FCS. This research completely demonstrates the operational performance of the MMAE/FS/CR in various regions of the flight envelope using gain and model scheduling.

3.4.2 State Space Design Parameters

Once the aircraft's nonlinear equations of motion have been linearized about a nominal point in the flight envelope, the linear, time-invariant (LTI) 'design model' is formatted as

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

where **x**(*t*) is the state vector, **A** is the plant matrix based on the LTI model, **B** is the control input matrix, **u**(*t*) is the control input, **G** is the noise injection matrix, and **w**(*t*) is the zero-mean white dynamics driving noise. Previous research has shown the development of these matrices and vectors [11,12]. The following chapters will briefly detail their development as they are used in this research effort.

3.4.2.1 State Vector

The state vector, $\mathbf{x}(t)$, used for this research is shown below

$$\mathbf{x}(t) = \begin{bmatrix} \theta(t) \\ \mathrm{u}(t) \\ \alpha(t) \\ \mathrm{q}(t) \\ \phi(t) \\ \beta(t) \\ \mathrm{p}(t) \\ \mathrm{r}(t) \end{bmatrix} \tag{3.2}$$

where the state variables are defined as

$\theta(t)$	Pitch Angle (rad)
$\mathrm{u}(t)$	Forward Velocity (ft/sec)
$\alpha(t)$	Angle of Attack (rad)
$\mathrm{q}(t)$	Pitch Rate (rad/sec)
$\phi(t)$	Bank Angle (rad)
$\beta(t)$	Sideslip Angle (rad)
$\mathrm{p}(t)$	Roll Rate (rad/sec)
$\mathrm{r}(t)$	Yaw Rate (rad/sec)

The first four state variables (θ, u, α, q) describe the longitudinal motion of the aircraft while the last four state variables (ϕ, β, p, r) describe the lateral motion of the aircraft.

3.4.2.2 Plant Matrix

The plant matrix, $\mathbf{A}$, used for this research effort is

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X_\theta{}' & X_u{}' & X_\alpha{}' & X_q{}' & 0 & 0 & 0 & 0 \\ Z_\theta{}' & Z_u{}' & Z_\alpha{}' & Z_q{}' & 0 & 0 & 0 & 0 \\ M_\theta{}' & M_u{}' & M_\alpha{}' & M_q{}' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \phi_r{}' \\ 0 & 0 & 0 & 0 & Y_\phi{}' & Y_\beta{}' & Y_p{}' & Y_r{}' \\ 0 & 0 & 0 & 0 & 0 & L_\beta{}' & L_p{}' & L_r{}' \\ 0 & 0 & 0 & 0 & 0 & N_\beta{}' & N_p{}' & N_r{}' \end{bmatrix} \tag{3.3}$$

The plant matrix contains elements for both the longitudinal and lateral dynamics of the aircraft. The derivatives shown in the upper left of the plant matrix are the longitudinal derivatives, and the lateral derivatives are shown in the lower right portion of the plant matrix. It is seen from the plant matrix that the lateral and longitudinal channel state dynamics are independent. The X', Y' and Z' variables represent the aerodynamic forces in the x, y, and z directions, whereas the L', M', and N' variables represent the aerodynamic moments in the x, y, and z directions. The coordinate frame of the aircraft is set up with the positive x-axis direction coming out of the nose, the y-axis coming out the right wing, and the z-axis coming out the bottom of the aircraft. The origin of the coordinate frame is established in the early design of the aircraft and does not represent the center of gravity of the aircraft. The center of gravity changes with the payload of the aircraft. The variables are in terms of the angle-of-attack (α) and sideslip (β) angles, hence the primed notation ($'$) according to standard convention [5]. The variables use the shorthand notation

$$X_\theta' = \partial X'/\partial\theta \tag{3.4}$$

The variable X_θ' is read, "the partial derivative of the aerodynamic force in the x-axis direction with respect to the pitch angle." This variable is also read as, "the change in the aerodynamic force in the x-axis direction due to a change in the pitch angle."

3.4.2.3 Input Vector

The input vector, $\mathbf{u}(t)$, used in this research effort is

$$\mathbf{u}(t) = \begin{bmatrix} \delta_e(t) \\ \delta_{dt}(t) \\ \delta_f(t) \\ \delta_a(t) \\ \delta_r(t) \end{bmatrix} \tag{3.5}$$

where the input variables are defined as

$\delta_e(t)$	Elevator Position (rad)
$\delta_{dt}(t)$	Differential Tail Position (rad)
$\delta_f(t)$	Flap Position (rad)
$\delta_a(t)$	Aileron Position (rad)
$\delta_r(t)$	Rudder Position (rad)

3.4.2.4 Input Matrix

The input matrix, **B**, used in this research is shown in Equation (3.6) on the next page.

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ X'_{\delta_e} & 0 & X'_{\delta_f} & 0 & 0 \\ Z'_{\delta_e} & 0 & Z'_{\delta_f} & 0 & 0 \\ M'_{\delta_e} & 0 & M'_{\delta_f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & Y'_{\delta_{dt}} & 0 & Y'_{\delta_a} & Y'_{\delta_r} \\ 0 & L'_{\delta_{dt}} & 0 & L'_{\delta_a} & L'_{\delta_r} \\ 0 & N'_{\delta_{dt}} & 0 & N'_{\delta_a} & N'_{\delta_r} \end{bmatrix} \tag{3.6}$$

where, as before, the aerodynamic forces in the *x*, *y*, and *z* axes are represented by *X′*, *Y′* and *Z′*, respectively, and the moments about the x, *y*, and *z* axes are represented by *L′*, *M′*, and *N′*, respectively. The purpose of the primed notation (′) is the same as mentioned in the plant matrix Section 3.4.2.2.

3.4.1.5 Noise Injection Matrix

The noise injection matrix, **G**, used in this research is

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ X'_{u} & X'_{\alpha} & X'_{q} & 0 & 0 & 0 \\ Z'_{u} & Z'_{\alpha} & Z'_{q} & 0 & 0 & 0 \\ M'_{u} & M'_{\alpha} & M'_{q} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y'_{p} & Y'_{\beta} & Y'_{r} \\ 0 & 0 & 0 & L'_{p} & L'_{\beta} & L'_{r} \\ 0 & 0 & 0 & N'_{p} & N'_{\beta} & N'_{r} \end{bmatrix} \tag{3.7}$$

3.4.1.6 Dynamics Driving Noise

The dynamics driving noise is added to the system through the **Gw**(t) term from the state space equation, Equation (3.1). The dynamic driving noise is zero-mean, white Gaussian noise. The strength of the noise is given by the **Q** matrix. The dynamic driving noise strength used in this research is

$$\mathbf{Q}=\begin{bmatrix} 4.5\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.0\times10^{-6} & 1.1\times10^{-8} & 0 & 0 & 0 \\ 0 & 1.1\times10^{-8} & 1.5\times10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.0\times10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.0\times10^{-6} & 6.3\times10^{-9} \\ 0 & 0 & 0 & 0 & 6.3\times10^{-9} & 2.4\times10^{-6} \end{bmatrix} \tag{3.8}$$

The diagonal noise strengths are associated with the u, α, q, p, β, and r states, respectively. It is seen from the **Q** matrix that the α and q states are correlated with each other, and the β and r states are also correlated with each other. All other state noises are uncorrelated with each other. The units of the **Q** matrix are shown in Table (1) [11,12]:

Q	States	Units
Q(1,1)	u	ft^2/sec
Q(2,2)	α	$rad^2 \cdot sec$
Q(2,3)	α vs. q	rad^2
Q(3,3)	q	rad^2/sec
Q(4,4)	p	rad^2/sec
Q(5,5)	β	$rad^2 \cdot sec$
Q(5,6)	β vs. r	rad^2
Q(6,6)	r	rad^2/sec

Table 1: Units of the Dynamic Driving Noise

3.4.3 Redefinition of the Input Vector and Matrix

The initial input vector, ***u***(t), and input matrix, **B**, defined earlier represent the combined surfaces acting as one input into the system. In other terms, the elevator command is described as one variable rather than two, one for the right elevator and one for the left. In order to apply redistributed control commands to the surfaces, the state space input vector and matrix is altered to represent the physical surfaces of the aircraft. Previous research has shown the development of the redefined input vector and matrix [11,12].

3.4.3.1 Input Vector

The redefined input vector used in this research is

$$\mathbf{u}_{\text{mod}}(t) = \boldsymbol{\delta}_{\mathbf{act}}(t) = \begin{bmatrix} \delta_{ls}(t) \\ \delta_{rs}(t) \\ \delta_{lf}(t) \\ \delta_{rf}(t) \\ \delta_{r}(t) \end{bmatrix} \tag{3.9}$$

where the modified input vector variables are defined as

$\delta_{ls}(t)$	Left Stabilator Position (rad)
$\delta_{rs}(t)$	Right Stabilator Position (rad)
$\delta_{lf}(t)$	Left Flaperon Position (rad)
$\delta_{rf}(t)$	Right Flaperon Position (rad)
$\delta_{r}(t)$	Rudder Position (rad)

3.4.3.2 Input Matrix

The input matrix used in this research is shown in Equation (3.10):

$$\mathbf{B}_{\text{mod}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}X'_{\delta_e} & \frac{1}{2}X'_{\delta_e} & \frac{1}{2}X'_{\delta_f} & \frac{1}{2}X'_{\delta_f} & 0 \\ \frac{1}{2}Z'_{\delta_e} & \frac{1}{2}Z'_{\delta_e} & \frac{1}{2}Z'_{\delta_f} & \frac{1}{2}Z'_{\delta_f} & 0 \\ \frac{1}{2}M'_{\delta_e} & \frac{1}{2}M'_{\delta_e} & \frac{1}{2}M'_{\delta_f} & \frac{1}{2}M'_{\delta_f} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}Y'_{\delta_{dt}} & \frac{1}{2}Y'_{\delta_{dt}} & -\frac{1}{2}Y'_{\delta_a} & \frac{1}{2}Y'_{\delta_a} & \frac{1}{2}Y'_{\delta_r} \\ -\frac{1}{2}L'_{\delta_{dt}} & \frac{1}{2}L'_{\delta_{dt}} & -\frac{1}{2}L'_{\delta_a} & \frac{1}{2}L'_{\delta_a} & \frac{1}{2}L'_{\delta_r} \\ -\frac{1}{2}N'_{\delta_{dt}} & \frac{1}{2}N'_{\delta_{dt}} & -\frac{1}{2}N'_{\delta_a} & \frac{1}{2}N'_{\delta_a} & \frac{1}{2}N'_{\delta_r} \end{bmatrix} \tag{3.10}$$

3.4.4 Augmented System

The state space system with the modified input vector and matrix, is augmented with a reduced order actuator design model. The actuator design model used in this research effort is a first order lag approximation.

3.4.4.1 Actuator Design Model

The truth model for each actuator used in the VISTA F-16 SRF simulation is a fourth order model. The actuator truth model is

$$\frac{\delta_{act}}{\delta_{cmd}} = \frac{(20.2)(144.8)(71.4)^2}{(s+20.2)(s+144.8)(s^2+2(0.736)(71.4)s+71.4^2)} \tag{3.11}$$

The actuator design model used in this research effort, which has been shown to provide adequate performance in past research efforts [11, 12], is shown in Equation (3.12):

$$\frac{\delta_{act}}{\delta_{cmd}} = \frac{14}{s+14} \tag{3.12}$$

The actuator design model has a breakpoint at ω = 14 rad/sec which has been shown to yield better matching performance than having the breakpoint at ω = 20.2 rad/sec, which might seem to be the obvious reduced order model for Equation (3.11) [11, 12]. The actuator design model is then placed into the state space representation

$$\dot{\boldsymbol{\delta}}_{act}(t) = \begin{bmatrix} -14 & 0 & 0 & 0 & 0 \\ 0 & -14 & 0 & 0 & 0 \\ 0 & 0 & -14 & 0 & 0 \\ 0 & 0 & 0 & -14 & 0 \\ 0 & 0 & 0 & 0 & -14 \end{bmatrix} \boldsymbol{\delta}_{act}(t) + \begin{bmatrix} 14 & 0 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 14 \end{bmatrix} \boldsymbol{\delta}_{cmd}(t) \tag{3.12}$$

where

$$\boldsymbol{\delta}_{act}(t) = \begin{bmatrix} \delta_{ls}(t) \\ \delta_{rs}(t) \\ \delta_{lf}(t) \\ \delta_{rf}(t) \\ \delta_{r}(t) \end{bmatrix} \quad \textit{and} \quad \boldsymbol{\delta}_{cmd}(t) = \begin{bmatrix} \delta_{ls_{cmd}}(t) \\ \delta_{rs_{cmd}}(t) \\ \delta_{lf_{cmd}}(t) \\ \delta_{rf_{cmd}}(t) \\ \delta_{r_{cmd}}(t) \end{bmatrix} \tag{3.13}$$

3.4.4.2 System Design Augmentation

Once the state space representation of the actuator design model is defined, that state space model is augmented with the actuator design model. Previous research has shown the derivation of the augmented design model [11,12]. The continuous-time augmented model used in this research is

$$\dot{\mathbf{x}}_{aug}(t) = \mathbf{A}_{aug}\mathbf{x}_{aug}(t) + \mathbf{B}_{aug}\mathbf{u}_{aug}(t) + \mathbf{G}_{aug}\mathbf{w}(t) \tag{3.14}$$

where

$$\mathbf{x}_{aug}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\delta}_{act}(t) \end{bmatrix} \qquad \mathbf{u}_{aug}(t) = \left[\boldsymbol{\delta}_{cmd}(t)\right] \tag{3.15}$$

and

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_{\text{mod}} \\ \mathbf{0} & -14 \cdot \mathbf{I}_{5x5} \end{bmatrix} \qquad \mathbf{B}_{aug} = \begin{bmatrix} \mathbf{0} \\ 14 \cdot \mathbf{I}_{5x5} \end{bmatrix} \qquad \mathbf{G}_{aug} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \tag{3.16}$$

3.4.5 Discrete-Time Model Equivalence

The continuous-time model derived in the previous section is converted into an equivalent discrete-time model [11,12] for implementation on the VISTA F-16 aircraft. The VISTA F-16 digital flight computer receives discrete measurements at a rate of 64 Hz. These sampled measurements are noise-corrupted. Section 3.4.5.2 shows the measurement vector and Section 3.4.5.3 shows the associated measurement matrix.

3.4.5.1 Discrete-Time Equivalent Model

Since this research effort is implemented on a digital flight computer, the continuous-time model derived earlier cannot be used directly. The continuous-time model must be discretized into a discrete-time model that is equivalent to the continuous-time model in the sense that, when observed at the sample times, the outputs of the two models are indistinguishable. The discrete-time model has been derived earlier in previous research efforts [11,12]. The discrete-time model used in this research is:

$$\mathbf{x}_{aug}(t_{i+1}) = \mathbf{\Phi}_{aug}(t_{i+1}, t_i)\mathbf{x}_{aug}(t_i) + \mathbf{B}_{d-aug}\mathbf{u}_{aug}(t_i) + \mathbf{w}_{d-aug}(t_i) \tag{3.17}$$

where

$$\begin{aligned} \mathbf{\Phi}_{aug}(t_{i+1}, t_i) &= e^{\mathbf{A}_{aug}\Delta T} \\ \mathbf{B}_{d-aug}(t_i) &= \left(\int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau} d\tau\right)\mathbf{B}_{aug} \end{aligned} \tag{3.18}$$

The variable ΔT is the sample period. The statistics of the zero-mean, white noise, $\mathbf{w}_{\text{d-aug}}(t_i)$, are

$$\begin{aligned} &E\{\mathbf{w}_{d\_aug}(t_i)\} = \mathbf{0} \\ &E\{\mathbf{w}_{d\_aug}(t_i)\,\mathbf{w}_{d\_aug}{}^T(t_i)\} = \mathbf{Q}_{d-aug} \\ &E\{\mathbf{w}_{d\_aug}(t_i)\,\mathbf{w}_{d\_aug}{}^T(t_j)\} = \mathbf{0} \qquad t_i \neq t_j \end{aligned} \tag{3.19}$$

where

$$\mathbf{Q}_{d-aug} = \int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau}\mathbf{G}_{aug}\mathbf{Q}\mathbf{G}_{aug}^T e^{\mathbf{A}_{aug}^T\tau} d\tau \tag{3.20}$$

3.4.5.2 Discrete-Time Measurement Vector

The measurement equation used in this research, which has been derived in previous research efforts, is

$$\mathbf{z}(t_i) = \mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \tag{3.21}$$

where $\mathbf{H}_{aug}$ is the measurement matrix, and $\mathbf{v}(t_i)$ is the measurement noise. The measurement vector, $\mathbf{z}(t_i)$, is defined as

$$\mathbf{z}(t_i) = \begin{bmatrix} \alpha(t_i) \\ \mathrm{q}(t_i) \\ \mathrm{a}_n(t_i) \\ \mathrm{p}(t_i) \\ \mathrm{r}(t_i) \\ \mathrm{a}_y(t_i) \end{bmatrix} \tag{3.22}$$

The measurement variables are defined as

$\alpha(t_i)$	Angle of Attack (rad)
$q(t_i)$	Pitch Rate (rad/sec)
$a_n(t_i)$	Normal Acceleration (g's)
$p(t_i)$	Roll Rate (rad/sec)
$r(t_i)$	Yaw Rate (rad/sec)
$a_y(t_i)$	Lateral Acceleration (g's)

The normal and lateral accelerations are measured at the pilot's station. Previous research has shown the lateral acceleration model of the VISTA simulation did not provide satisfactory performance so it was replaced by a linear model [11,12]. This research effort uses the linear model for the lateral acceleration in both the design model and the truth model.

3.4.5.3 Discrete-Time Measurement Matrix

The measurement matrix, $\mathbf{H}_{aug}$, has been derived in previous research [11,12] and is defined as

$$\mathbf{H}_{aug} = [\mathbf{H} \quad \mathbf{D}_z] \tag{3.23}$$

where

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{\bar{U}}{322} Z_\theta' & -\frac{\bar{U}}{322} Z_u' & -\frac{\bar{U}}{322} Z_\alpha' & -\frac{\bar{U}}{322}(Z_q' - 1) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{\bar{U}}{322} Y_\phi' - 1 & \frac{\bar{U}}{322} Y_\beta' & \frac{\bar{U}}{322} Y_p' & \frac{\bar{U}}{322}(Y_r' - 1) \end{bmatrix} \tag{3.24}$$

$$\mathbf{D}_z = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{\bar{U}}{32.2}(\frac{1}{2} Z_{\delta_e}') & -\frac{\bar{U}}{32.2}(\frac{1}{2} Z_{\delta_e}') & -\frac{\bar{U}}{32.2}(\frac{1}{2} Z_{\delta_f}') & -\frac{\bar{U}}{32.2}(\frac{1}{2} Z_{\delta_f}') & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{\bar{U}}{32.2}(-\frac{1}{2} Y_{\delta_{dt}}') & \frac{\bar{U}}{32.2}(\frac{1}{2} Y_{\delta_{dt}}') & \frac{\bar{U}}{32.2}(-\frac{1}{2} Y_{\delta_a}') & \frac{\bar{U}}{32.2}(\frac{1}{2} Y_{\delta_a}') & \frac{\bar{U}}{32.2}(\frac{1}{2} Y_{\delta_r}') \end{bmatrix} \tag{3.25}$$

and where the $\overline{U}$ is the nominal forward velocity used for trim.

3.4.5.4 Measurement Noise Vector

The measurement noise covariance of the measurement noise vector, $\mathbf{v}(t_i)$, is defined as shown in Equation (3.26):

$$\mathbf{R} = \begin{bmatrix} 0.004^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.006^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.02^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.006^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.005^2 \end{bmatrix} \qquad (3.26)$$

The units of **R** are consistent with the units in Section 3.4.5.2 for the components of $\mathbf{z}(t_i)$. It is assumed that the measurement vector is zero-mean and has the statistical characteristic of discrete white Gaussian noise. It is seen from the measurement noise covariance matrix, **R**, that all the measurement noises are uncorrelated.

3.5 Failure Modes

Failure modes must be incorporated into the truth model and design model in order to test the MMAE/FS/CR's response to failures. Failures in this research will be described in terms of an effectiveness factor, ε, with a range between zero and one. An effectiveness of 0%, or $\varepsilon = 0$, indicates a complete failure whereas an effectiveness of 100%, or $\varepsilon = 1$, indicates completely operational. Intermediate effectiveness values indicate partial failures.

3.5.1 Truth Model Failure Modes

The truth model has been modified in previous research efforts to incorporate partial and complete failures. There has not been a complete failure mode study accomplished for the VISTA F-16 aircraft so failure mode assumptions have been made [11,12]. These assumptions are based on previous research in this field [7,9-12,17,27,28,32,33].

Sensor failures can be modeled in several different ways. In the 'real world', the failed sensor can output no measurements except for noise, a bias in the measurement, an increase in the noise level, or in the modeling sense, the **H** matrix can become modified. The method used in previous research and in this research effort is the removal of the measurement while leaving the zero-mean white Gaussian noise [11,12].

Flight control actuator or surface failures can also be modeled in several different ways. In the 'real world', the failed flight control actuator/surface can 'fail to free stream', meaning the flight control surface does not contribute to the control of the aircraft and that airflow over the surface is undisturbed and stability derivatives are unchanged. The failed actuator/surface can also fail in a locked position, contributing a constant control moment to the aircraft. Portions of the flight control surface can be battle damaged, resulting in reduced control authority and deterioration in the time response of the actuator/surface. Reduced hydraulic pressure can also yield such a partial failure. In this research effort, the actuator/surface total failure mode is assumed to be 'fail to free stream'. This decision is based on previous research efforts in this field [7,9-12,17,27,28,32,33]. It is also assumed that the failed surface can still contribute to the control of the aircraft if it is only partially failed. In this case, the actuator would have an effectiveness value, ε, between zero and one associated with it.

3.5.2 Design Model Failure Modes

The use of failure matrices in the linear design model has been introduced in previous research efforts [32,33]. The $\mathbf{F}_{ai}$ and $\mathbf{F}_{sj}$ matrices are used to alter the design model in order to duplicate an actuator/surface or sensor failure. $\mathbf{F}_{ai}$ is the failure matrix for the actuator/surface (the subscript *a* denotes 'actuator') where the subscript *i* indicates the index of the failed actuator/surface. The $\mathbf{F}_{ai}$ failure matrix is an identity matrix except for the i^{th} diagonal term, which is the effectiveness of the completely or partially failed actuator/surface. $\mathbf{F}_{sj}$ is the failure matrix for a sensor failure (the subscript *s* denotes sensor) where the subscript *j* is indicates the index of the failed sensor. The failure matrices are implemented in the continuous-time design models as

$$\dot{\mathbf{x}}_{aug}(t) = \mathbf{A}_{aug}\mathbf{x}_{aug}(t) + \mathbf{B}_{aug}\mathbf{F}_{ai}\mathbf{u}_{aug}(t) + \mathbf{G}_{aug}\mathbf{w}(t) \tag{3.27}$$

and the discrete-time measurement model is

$$\mathbf{z}(t_i) = \mathbf{F}_{sj}\mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \tag{3.28}$$

The only difference in the equivalent discrete-time design model with the addition of the failure matrix is

$$\mathbf{B}_{d-aug}(t_i) = \left(\int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau} d\tau \right) \mathbf{B}_{aug}\mathbf{F}_{ai} \tag{3.29}$$

On a fully-functional aircraft, the failure matrices are both identity matrices. A failure of an actuator/surface is shown by the value of $\mathbf{F}_{ai}(i,i)$, where i is the index of the actuator/surface failure, becomes the effectiveness value, ε. This reduces the actuator/surface contribution to the control of the aircraft, simulating a failure to free stream. By adjusting the effectiveness value, a partial actuator/surface failure is duplicated. A total failure of a sensor at the index, j, is duplicated by the value of $\mathbf{F}_{sj}(j,j)$ becoming zero. This eliminates the information for the sensor at that index while not eliminating the noise simulating a sensor failure. In this research, partial and complete flight control actuator/surface failures to free stream, and complete sensor failures are analyzed.

3.6 Chapter Summary

This chapter presented the various models used in this research effort. The ‘real world’ VISTA F-16 aircraft was presented first, and then the ‘truth model’ VISTA F-16 SRF computer simulation was presented. The state space design of the reduced-order ‘design model’ for the MMAE was discussed, including the addition of actuator dynamics. Finally, the various failure modes were presented. Now that the design models have been derived, they are used to form the elemental filter banks and the control redistribution algorithm of the MMAE/FS/CR of the next chapter.

Chapter 4: Multiple Model Adaptive Estimation with Filter Spawning and Control Redistribution (MMAE/FS/CR) Overview

4.1 Chapter Overview

This chapter presents the Multiple Model Adaptive Estimation with Filter Spawning and Control Redistribution (MMAE/FS/CR) algorithm in detail. The first section describes the derivation and implementation of the MMAE. The next section presents the filter spawning concept. This chapter concludes with a description of control redistribution as it applies to this research.

4.2 Filter Models

The models used in this research effort are described in chapter three. The ‘design model’ is used in each of the Kalman filters within the MMAE, each hypothesizing a different failure. The MMAE algorithm compares each model prediction of measurements, before they arrive and based on that model’s hypothesis of the current failure status, to the incoming measurements and assigns a probability to each hypothesis based on its relative closeness to the ‘real world’. The following sections describe the general use of the ‘design model’ in the MMAE and model scheduling.

4.2.1 MMAE Filter Model Description (Kalman Filter)

The MMAE consists of several elemental Kalman filters that run in parallel. Measurements are brought in and sent to each of the Kalman filters. All elemental filters conduct a measurement update and, based on their residuals, a probability is assigned to each hypothesis by the conditional probability generator. The hypotheses used for the basis of elemental filters within the MMAE in this research consist of several failure hypotheses and one fully functional hypothesis. By analyzing the residuals of each of the elemental filters, the failure status of the aircraft is determined.

The models used in the Kalman filters are linearized dynamic equations of the VISTA F-16 aircraft. These models have been linearized about a trimmed flight condition. Small perturbations in the trim conditions caused by the application of flight controls are the inputs to the state space model. It is seen in chapter three that a failure matrix can be mixed with the input matrix of the state space model in order to duplicate a flight control actuator/surface failure to free stream. By using a separate failure matrix for each elemental filter, the elemental hypothesis failure models are created. The fully functional model has the failure matrix equal to an identity matrix of the same size. The sensor failures are also identified in the same fashion except using a sensor failure matrix, as is also shown in chapter three. In the case of a sensor, a zero in the sensor failure matrix identifies the total failure. This creates the failure hypothesis models for the failed sensors.

In an initial implementation, having one filter hypothesize a fully functional aircraft and each other filter hypothesize a single failure, previous research has demonstrated the success of MMAE in detecting a single complete sensor or actuator/surface failure [7,9-12,17,27,28]. Once a single failure is detected, different hypotheses can be brought online that look for a second failure. As explained in Chapter 2, Section 2.3.6, a hierarchical method is used in order to look for the second failure while maintaining the same number of elemental filters as was used to detect the first failure. The second bank of filters still contains a fully functional aircraft hypothesis, in order to back up the decision 'tree' of what bank of filters to use in the case of additional measurements over time indicating that a failure had not occurred after all. In previous research, this method has shown satisfactory performance in the detection of two complete failures [7]. Section 4.3 discusses the filter spawning concept that allows the MMAE to handle partial failures.

Through these basic detection principles, the MMAE depends heavily upon the 'design model'. Previous research on this topic has been conducted under one flight condition (Mach 0.4

and 20,000 ft) because of the difficulty in the detection of failures. In order to test the MMAE completely, the algorithm must be tested under multiple flight conditions.

4.2.2 Model Scheduling

It has been shown in previous research that if the MMAE is based on a single flight condition, and the aircraft moves away from that flight condition, all linear perturbation models are invalidated to the extent that an MMAE for failure detection becomes incapacitated: no models correctly represents the ‘real world’ situation [9,10]. This drives the requirement for model scheduling. Model scheduling is the process of storing multiple models based on multiple regions of the flight envelope. The flight envelope is the region bounded by airspeed and altitude where the aircraft flies. As the aircraft transitions to a new location within the flight envelope, a new ‘design model’ based on a trim condition valid for that region within the flight envelope is used, either by simple lookup tables or by interpolation among table entries.

In the case of the MMAE, the trim condition changes for each of the hypotheses as the aircraft moves to a new trim condition. Model scheduling is required in order to estimate the aircraft status correctly throughout the flight envelope. Having a ‘design model’ based on all the trim conditions within the flight envelope completes this. As the aircraft nears a prestored trim condition, the ‘design model’ valid for that region is used. In this research effort, each prestored trim condition is 50 psf (pounds per square foot) from the next prestored trim condition. This is shown later to provide good detection performance. The ‘design model’ is then incorporated as before with the failure matrices in order to search continuously for all failures. The same is true with the measurement matrix. As the aircraft goes to a new trim condition, the measurement matrix that best duplicates the measurements for that trim condition is used. This process allows for the continuous operation of the MMAE for flight control failure detection, even though the aircraft is changing trim conditions.

4.3 Filter Spawning

Filter spawning is a concept developed in previous research [11,12]. This concept gives the MMAE the ability not only to detect failures but also determine the effectiveness of the failed flight control actuator/surface for the case of partial failures. This is accomplished by using the MMAE to look for an initial failure. Once an actuator/surface failure is identified, the filter spawning algorithm launches several filters based on that one actuator/surface having failed, but each modeled with a different effectiveness. By blending the MMAE/FS elemental probabilities for the fully functional aircraft, the fully failed actuator/surface and the partially failed actuator/surface as assumed by the spawned filters, an estimate of the actuator effectiveness is made. This method provides a finer level of discretization and produces a more accurate estimate of effectiveness than using a filter based only on a fully functional aircraft hypothesis and a filter based on a fully failed actuator/surface hypothesis. The effectiveness of the partially failed actuator/surface is important because the actuator/surface can still be used at the lower effectiveness value. This is important in the case of the F-16 aircraft for which the rudder is the surface that produces the primary yawing moment on the aircraft. If the rudder is operating at 50% effectiveness, then it is still useful for control of the aircraft whereas in the older configuration, the 50% failure of the rudder would be declared completely failed, loosing any effectiveness remaining and an important control surface. The following sections describe the filter spawning concept in detail.

4.3.1 Filter Spawning Design

Previous research has explored filter spawning in great detail [11,12]. Presented here is a brief description of the filter spawning design and operation. As mentioned above, the MMAE detects the initial actuator failure. The first step of the filter spawning algorithm is to determine the probability of the failure declaration from the MMAE in order to reduce false alarms. The initial detection of the failure is done in a MAP method. If the probability associated with the failure is large (98% probability of failure), then the failure status is checked as positive. If the failure is a flight control actuator/surface, then the actuator/surface failure is declared and the effectiveness of the failed actuator/surface is subsequently determined with the aid of spawning.

The effectiveness, ε, of the failed actuator/surface is determined by incorporating additional filters based on various levels of effectiveness online in the MMAE bank. This provides a finer discretization of the parameter space. Each spawned filter has a different effectiveness value between zero and one. Zero relates to the completely failed actuator/surface, whereas an effectiveness value of one relates to a fully functional actuator/surface.

4.3.2 Creating Estimates With Filter Spawning

The MMAE is used to determine the probabilities associated with each of the spawned filters and the Bayesian blending approach is taken in the estimation of the actuator effectiveness. The equation shown below illustrates this approach:

$$\hat{\varepsilon}(t_i) = \frac{\sum_{k \in K} [\varepsilon_k \cdot p_k(t_i)]}{\sum_{k \in K} p_k(t_i)} \tag{4.1}$$

where the ε is the estimate of effectiveness, and the ε_k value is the effectiveness hypothesis in any one of the spawned filters or the associated fully failed actuator/surface elemental filters or the fully functional aircraft elemental filter. Each of the filters is given the probability of $p_k(t_i)$ by the MMAE conditional probability generator. K is the set of the spawned filters, the fully failed actuator/surface elemental filter and the fully functional aircraft elemental filter. The denominator in Equation (4.1) is used as a scaling factor such that all the elemental probabilities will sum to equal one.

4.4 Filter Spawning Implementation

The implementation of MMAE/FS algorithm is broken up into several steps. The first step is the initialization of the filter banks and the application of initial conditions. The next step is the conventional MMAE computation. Once a failure declaration is made on a specific actuator/surface, the effectiveness is determined. This section also introduces bank swapping, which is the next and final step of this algorithm. Figure 4 shows the filter spawning algorithm.

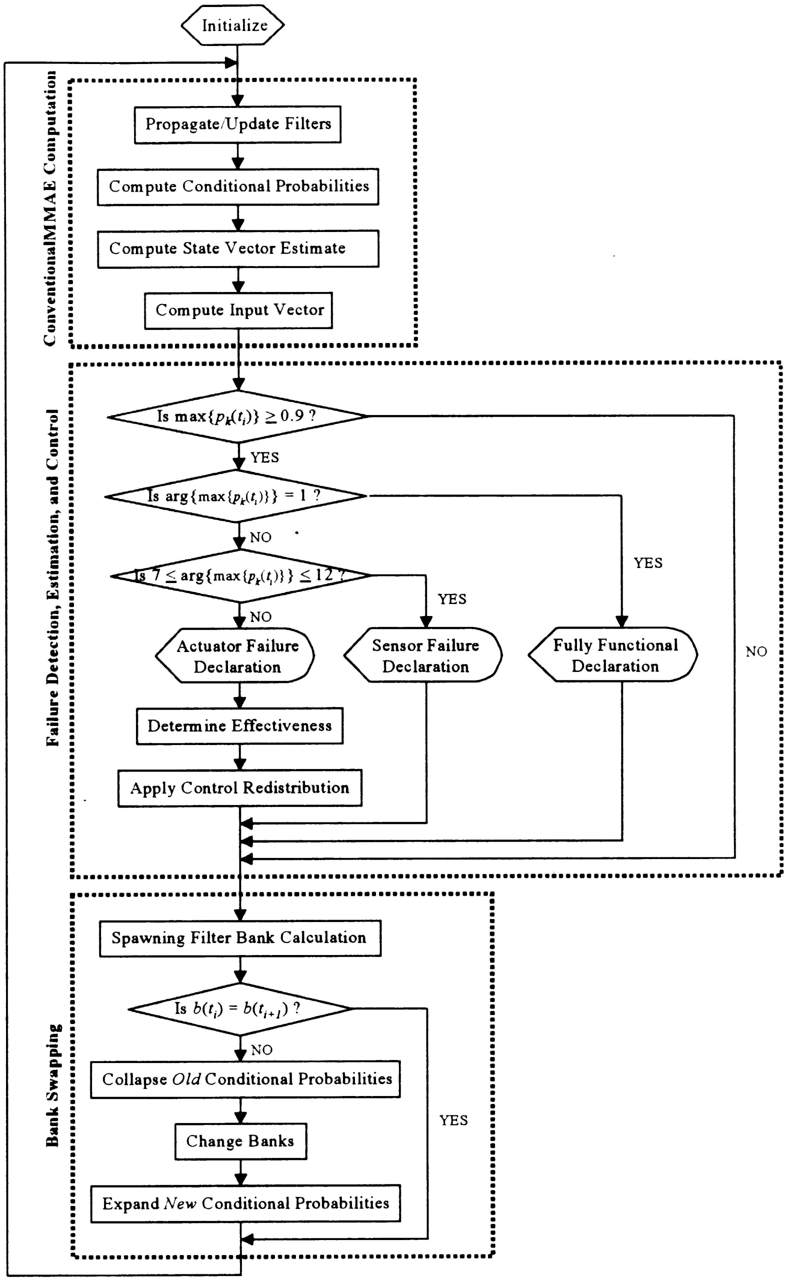


Figure 4: Filter Spawning Algorithm [11,12]

4.4.1 Initialization

The first step in the implementation of the MMAE/FS is the initialization of the filter banks. These banks contain each failure hypothesis for all single actuator/surface and sensor failures and a fully functional aircraft hypothesis. Each bank (except one) also includes spawned filters for a particular actuator/surface failure. Table 2 shows the bank descriptions of the MMAE with Filter Spawning [11,12].

		Bank					
		1	2	3	4	5	6
FF	1	FF	FF	FF	FF	FF	FF
Complete Actuator Failure	2	LS	LS	LS	LS	LS	LS
	3	RS	RS	RS	RS	RS	RS
	4	LF	LF	LF	LF	LF	LF
	5	RF	RF	RF	RF	RF	RF
	6	R	R	R	R	R	R
Complete Sensor Failure	7	AA	AA	AA	AA	AA	AA
	8	PR	PR	PR	PR	PR	PR
	9	NA	NA	NA	NA	NA	NA
	10	RR	RR	RR	RR	RR	RR
	11	YR	YR	YR	YR	YR	YR
	12	LA	LA	LA	LA	LA	LA
Spawned	13		$\mathrm{LS}_{\varepsilon 21}$	$\mathrm{RS}_{\varepsilon 31}$	$\mathrm{LF}_{\varepsilon 41}$	$\mathrm{RF}_{\varepsilon 51}$	$\mathrm{R}_{\varepsilon 61}$
	14		$\mathrm{LS}_{\varepsilon 22}$	$\mathrm{RS}_{\varepsilon 32}$	$\mathrm{LF}_{\varepsilon 42}$	$\mathrm{RF}_{\varepsilon 52}$	$\mathrm{R}_{\varepsilon 62}$
	15		$\mathrm{LS}_{\varepsilon 23}$	$\mathrm{RS}_{\varepsilon 33}$	$\mathrm{LF}_{\varepsilon 43}$	$\mathrm{RF}_{\varepsilon 53}$	$\mathrm{R}_{\varepsilon 63}$

Fully Functional (FF) Left Stabilator (LS) Right Stabiltator (RS)
Left Flaperon (LF) Right Flaperon (RF) Rudder (R)
Angle of Attack (AA) Pitch Rate (PR) Normal Acceleration (NA)
Roll Rate (RR) Yaw Rate (YR) Lateral Acceleration (LA)

Table 2: MMAE with Filter Spawning Bank Descriptions

Previous research began with three spawned failure hypotheses [11,12], but more spawned filters can be used to provide a better effectiveness estimate. The use of more spawned filters also increases the computational load, since more filters are required to run in parallel.

The next step in the initialization of the MMAE/FS is to apply the initial trim conditions to all the models within the MMAE/FS. As the flight conditions change, the new trim conditions are applied to all the elemental filters for use in the MMAE/FS in the model scheduling routine. The MMAE/FS uses only one bank at a time and the initial starting bank is declared at this time.

The aircraft is assumed to be fully functional at the start of the flight. The lower bound of the MMAE is also set. Previous research has found that a lower bound of 0.001 performs acceptably. Initially, the lower bound value is used for all failure probabilities. All the probability minus the sum of the lower bound elemental probabilities for each failure state is initially started in the fully functional aircraft hypothesis elemental filter. At this point, the MMAE/FS is initialized and ready to begin operation.

The implementation of the FS algorithm for this research only uses the banks 2 through 6 from Table 2. This was done because considerable software changes would have been required to include bank 1 (no active spawned filters). When the MMAE/FS is initialized, bank 2 is arbitrarily used. The use of bank 1 in the actual aircraft implementation would be a better decision.

4.4.2 Conventional MMAE Computation

The next step in the implementation of the MMAE/FS algorithm is the conventional MMAE computation. The equations shown in Chapter 2 (Equations (2.4) through (2.10)) describe the state propagation from the previous time to the current time, and the measurement update. From the measurement update algorithm, the residuals, $\mathbf{r}_k(t_i)$, and the covariance of the residuals, $\mathbf{A}_k(t_i)$, are produced. These values are used to determine the conditional probability, $p_k(t_i)$, of each filter. The hypothesis that most closely matches the incoming measurements is given the highest conditional probability.

4.4.3 Failure Detection, Estimation, and Control

Once the conventional MMAE has been propagated and the measurement has been incorporated, the conditional probabilities assigned to each failure hypothesis are analyzed. If the highest probability is above a certain threshold (0.98 for this research effort), the failure hypothesis with the highest conditional probability is established. If this failure hypothesis is the fully functional hypothesis, then no failure is declared and the MMAE/FS moves onto the next step. If the highest conditional probability rests on a sensor failure, then a sensor failure is declared and the MMAE/FS moves onto the next step. If the highest conditional probability lies on an actuator/surface failure, then a flight control actuator/surface failure is declared, and the appropriate filters are spawned. At this point, the MMAE/FS uses the spawned filters in the current filter bank to assess the degree of failure (or its complement, the degree of effectiveness). The spawned filters are used to determine the effectiveness of the actuator using the Bayesian blending technique mentioned earlier. Once the effectiveness of the actuator/surface is determined, the flight control commands are redistributed in order to compensate for the actuator/surface (partial or full) failure. Control redistribution is explained later in this chapter.

4.4.4 Bank Swapping

Once the detection, estimation, and control step is accomplished, bank swapping occurs. This is required since the MMAE/FS is based on banks of filters. In order to change the particular actuator/surface for which the spawned filters in the current bank are designed, bank swapping among the banks defined in Table 2 must occur. This step consists primarily of two parts, the first of which is bank calculation and the second of which is swapping conditional probabilities.

The bank calculation is based on the flight control failure hypothesis with the highest conditional probability. The actuator/surface failure with the highest conditional probability determines the bank of elemental filters used to detect and estimate failures. This presumes that there are no spawned filters yet; if there *are* spawned filters, then their probabilities are summed with the probability of the full failure of that actuator/surface. If a sensor failure hypothesis or fully functional hypothesis contains the highest conditional probability, then the filter bank without any spawned filters is selected.

When an actuator/surface failure is identified by the MMAE/FS, the spawned filters for that failure are brought online. The conditional probability within the fully failed hypothesis is evenly distributed into the newly spawned filters and the filter based on the fully failed hypothesis. If the spawned filters are collapsed, (i.e., retracted) the conditional probabilities are collapsed into the corresponding fully failed elemental filter [11,12].

4.5 Control Redistribution

The Control Redistribution (CR) algorithm used in this research effort has been previously explored [32,33]. This section briefly describes the previous research efforts to apply control based on the estimations of the MMAE/FS. This section then discusses the implementation of CR as it is used in this research effort.

4.5.1 Control Application

There have been two primary types of control investigated for use with the MMAE/FS methodology [32,33]. The first type of control is the Multiple Model Adaptive Controller (MMAC). In this controller, elemental controllers are created for each hypothesis using modern LQG synthesis [23]. One form of the MMAC applies blended control where, like the estimates from the MMAE, the elemental controller commands are weighted with the conditional probabilities and summed together. This is shown in Chapter 2, Figure 2. In some cases, this approach can cause unacceptable aircraft responses since some of the contributing control commands are based on failures that have not actually occurred, due to the lower bound on the conditional probability. Another form of MMAC uses MAP methodology in which the elemental LQG controller with the highest conditional probability is used to control the aircraft, eliminating the insertion of improper control commands. However, this precludes purposeful blending of outputs of more than one elemental LQG controller (which might be useful for best performance; recall Section 2.4.1).

A second type of controller tested in previous research [32,33] is the CR algorithm used with an MMAE-based controller shown in Chapter 2, Figure 1. In this case, the MMAE/FS is used to detect failures and estimate the actuator effectiveness, and that information is used to alter the commands from the Block 40 flight control system in order to perform the commanded maneuver. Bayesian blending versus MAP is an issue for both MMAE-based control as well as for MMAC-based control.

4.5.2 MMAC Controller

LQG synthesis [23] is used to form the elemental controllers for the MMAC. The LQG synthesis requires three assumptions that are readily met for this application. The first assumption is that the 'design model' used is a linear, discrete-time model. The linear model assumption is met because linear perturbation models about trim are used to describe the behavior of the aircraft. The second assumption is that a quadratic cost function is well suited to flight control problems. This is true for flight control problems since the cost function attempts to minimize the mean squared deviations in the states while not expending excessive amounts of control energy or power to do so. The final assumption is the use of white Gaussian noise in the models. It has been found in previous research that the noises such as sensor noise and wind gusts have been successfully modeled according to this assumption. The justification for the Gaussian noise assumption is through the use of the Central Limit Theorem [21].

The first part of the LQG synthesis is the LQ regulator. The purpose of the regulator is to determine the control function which minimizes a specific cost function. Recall the discussion in Section 2.5.2 and Equation (2.16) for the general cost function. By balancing the quadratic cost function penalties, proper flight control actuator commands are produced in order to achieve the commanded maneuver.

LQG synthesis for this case entails the application of PI compensation rather than a simple proportional regulator. PI compensation is required in order to achieve Type I performance characteristics [23], or in other words, to track a constant input signal with a zero steady-state error even though constant unknown disturbances are added to the system. Once the

PI compensator is incorporated into the controller, the LQG PI controller gains are determined and applied [23].

Model following techniques can be used to improve the performance and the robustness of the LQG controller. The first technique that can be used is implicit model following. In implicit model following, the controller is penalized for deviations from the desired system response characteristics as specified by the output of the 'implicit' system model. This method does not necessarily improve the performance of the system at design conditions but robustness can be increased by its use [23]. The second technique that can be used is explicit model following. A Command Generator Tracker (CGT) incorporates this method [23]. The CGT forces the system to follow a desired model output, where the model is called the command generator. In this method, the performance of the system as compared to a desired trajectory is enhanced, but the feed-forward portion of the compensator does not affect the robustness of the system.

Previous research has shown the attempted design of the CGT/PI controller in order to reproduce the commands from the Block 40 flight control system [32,33]. The attempt to use LQG synthesis of a CGT/PI controller to replicate the Block 40 flight control system failed because the associated weighting matrices that would have been used to perform the synthesis for each elemental controller could not be determined. Various quadratic cost function penalties were tried but no cost function was found that gave the LQG based controller the capability to follow the desired commands of the Block 40 flight control system. Therefore, MMAC was abandoned in favor of MMAE-based control with Control Redistribution (CR).

4.5.3 Control Redistribution

The alternative method of control, as previously researched [32,33], is Control Redistribution (CR). The advantage of CR is the ability to apply corrective control to the aircraft without the need to modify the Block 40 flight control system. This is advantageous to the user since the original flight control system is used and does not need to be changed or altered and then recertified for flight-worthiness. The CR algorithm uses the MMAE/FS to detect and estimate the effectiveness of flight control actuator/surface failures. The CR algorithm then redistributes the command signals from the flight control system to other operational actuators/surfaces in order to accomplish the commanded maneuver. It is assumed that enough flight control redundancy is available such that a command intended for a failed actuator/surface can be redistributed to the remaining actuators/surfaces to accomplish the desired maneuver. This may not always be the case, since the loss of an actuator decreases the total control authority for the aircraft.

The mathematical basis for CR is presented here [32,33]. The product of the control matrix **B** and the input vector $\mathbf{u}(t_i)$ for a fully functional aircraft must equal the product of the control matrix $\mathbf{B}_{fail}$ and redistributed input vector $\mathbf{u}_r(t_i)$ of the aircraft with a failed actuator/surface, as shown in Equation (4.1):

$$\mathbf{Bu}(t_i) \equiv \mathbf{B}_{fail}\mathbf{u}_r(t_i) \tag{4.1}$$

The matrix that describes the failure $\mathbf{F}_{ai}$ is described in Chapter 3, Section 3.5.2. The actuator/surface failure matrix is an identity matrix with the effectiveness value ε as the i^{th} diagonal term corresponding to the fully failed (ε = 0) actuator/surface. This development is only for the case of ε = 0. This actuator failure matrix is then used to determine $\mathbf{B}_{fail}$:

$$\mathbf{B}_{fail} = \mathbf{B}\,\mathbf{F}_{ai} \tag{4.2}$$

The redistribution input $\mathbf{u}_r$ is described mathematically by employing a linear transformation matrix $\mathbf{D}_{ai}$. The $\mathbf{D}_{ai}$ matrix represents the control redistribution corresponding to the case of the i^{th} actuator/surface having failed. The relationship is:

$$\mathbf{u}_r(t_i) = \mathbf{D}_{ai}\,\mathbf{u}(t_i) \tag{4.3}$$

By substituting Equations (4.2) and (4.3) into (4.1), and realizing the result must be true for all values of the control vector $\mathbf{u}(t_i)$, the resulting equation is generated:

$$\mathbf{B}\mathbf{F}_{ai}\mathbf{D}_{ai} = \mathbf{B} \tag{4.4}$$

Since $\mathbf{F}_{ai}$ is rank deficient during complete actuator/surface failures ($\varepsilon = 0$), an inverse operation on $[\mathbf{BF}_{\mathrm{ai}}]$ cannot be used to determine the control redistribution matrix, $\mathbf{D}_{ai}$. A pseudoinverse must be taken in order to solve Equation (4.4) for $\mathbf{D}_{ai}$. The pseudoinverse is denoted in the resulting Equation (4.5) by the + superscript [32,33].

$$\mathbf{D}_{ai} = (\mathbf{B}\mathbf{F}_{ai})^{+}\mathbf{B} \tag{4.5}$$

In the case of a partial failure where $\mathbf{F}_{ai}$ is not rank deficient, the actual inverse can be taken to determine the values of $\mathbf{D}_{ai}$. Here we consider only the case of total failures. Using this result to treat partial failures will be discussed later. The pseudoinverse used in this research only gives an approximate value that is the best solution in the least squares sense. A second approach to the solution is through the use of a weighted pseudoinverse [23]. This method allows the designer to adjust the redistribution matrix to allow more or less redistributed control to go to particular actuators in order to compensate for the failed actuator/surface, based upon his or her design preferences. The control redistribution matrix used in this research is shown in Equation (4.6):

$$\mathbf{D}_a = \begin{bmatrix} 0.0000 & 1.0000 & 1.1037 & -1.1037 & -1.2719 \\ 1.0000 & 0.0000 & -1.1037 & 1.1037 & 1.2719 \\ 0.9060 & -0.9060 & 0.0000 & 1.0000 & 1.1524 \\ -0.9060 & 0.9060 & 1.0000 & 0.0000 & -1.1524 \\ -0.7862 & 0.7862 & 0.8678 & -0.8678 & 0.0000 \end{bmatrix} \tag{4.6}$$

This equation shows the complete control redistribution matrix for all complete failures. Its columns are the i^{th} column of each $\mathbf{D}_{ai}$ of Equation (4.5). It can also be seen from equation (4.6) that the amount of control required in order to compensate for a complete rudder failure is large with respect to the other failures. During the actual simulation, if for example a right stabilator failure is identified ($i = 2$), the control redistribution matrix would be a 5-by-5 identity matrix with its i^{th} (2^{nd}) column replaced by the i^{th} (2^{nd}) column from Equation (4.6):

$$\mathbf{D}_{ai} = \begin{bmatrix} 1.0000 & 1.0000 & 0 & 0 & 0 \\ 0 & 0.0000 & 0 & 0 & 0 \\ 0 & -0.9060 & 1.0000 & 0 & 0 \\ 0 & 0.9060 & 0 & 1.0000 & 0 \\ 0 & 0.7862 & 0 & 0 & 1.0000 \end{bmatrix} \tag{4.7}$$

If, continuing with the example of the complete right stabilator failure, the actual control used based on the knowledge of the failure is

$$\mathbf{u}_r(t_i) = \mathbf{D}_{ai}\,\mathbf{u}(t_i) \tag{4.8}$$

4.5.4 Control Techniques with Partial Actuator/Surface Failures

There are two approaches analyzed in this research effort to apply control to an aircraft given a partial failure of the actuator/surface. The first approach is to adjust the redistribution matrix to compensate for the partial actuator/surface failure, and the second approach is to increase the control gain on the partially failed channel without applying CR. The first approach involves using the redistribution matrix $\mathbf{D}_{ai}$ matrix and multiplying the i^{th} column by the complement of the actuator/surface effectiveness $(1 - \varepsilon)$ and adding 1 to the i^{th} diagonal term. This results in a full command to the partially failed actuator/surface in anticipation that ε of the desired response is thereby accomplished; $(1 - \varepsilon)$ of the desired response is then achieved through the control redistributed to the other actuators/surfaces. The second approach is to multiply the gain of the partially failed actuator by the inverse of the actuator/surface effectiveness. This is actually equivalent to using the development of Equations (4.1)-(4.4) for *nonzero* ε: with $\mathbf{F}_{ai}$ of full rank, Equation (4.4) yields $\mathbf{D}_{ai} = \mathbf{F}_{ai}^{-1}$. The first method is preferred in this research effort since the cause of the partial failure is unknown. The second method increases the gain on that actuator/surface, and without knowing the cause of the partial failure, this could cause greater damage to the aircraft (such as a hydraulic leak in the actuator: by increasing the gain to that actuator, the hydraulic system can be drained, leading to other flight control problems).

4.5.5 ε_{hat} (a_{hat}) vs. ε_{map} (a_{out})

In previous research efforts, two methods have been used to compute the estimate of actuator/surface effectiveness ε [11,12]. The first method uses Equation (4.1) to determine an estimate of effectiveness $\hat{\varepsilon}$ = "ε_{hat}" of a partially failed actuator/surface. In previous research [11,12] it was found that, while the aircraft was not maneuvering and while applying purposeful dither, the ε_{hat} created a biased estimate of the effectiveness of the partially failed actuator/surface. To solve this problem, a mapping from ε_{hat} to ε_{map} was established experimentally to provide good estimates of effectiveness, ε_{map}, given a non-maneuvering aircraft using purposeful dither. This research effort shows the performance difference between the two estimates of effectiveness for a maneuvering aircraft without dither.

4.6 Chapter Summary

This chapter presented the operation of MMAE with filter spawning in order to detect sensor failures and detect and estimate the extent of flight control actuator failures for use on the VISTA F-16 SRF simulation tool. This chapter also provided an explanation of the different possible control applications once a failure has been identified. The MMAC controller contains a separate elemental controller design for each failure case. This method relies on LQG synthesis to develop the controller. Unfortunately, LQG synthesis could not produce a controller that adequately represented the Block 40 flight control system, so a second approach had to be taken to apply corrective control. This approach uses Control Redistribution (CR) to control the aircraft once the MMAE/FS algorithm detects a flight control actuator/surface failure. CR redistributes the control originally intended for a failed actuator, to other redundant flight control actuators in order to accomplish the intended maneuver. The next chapter will present the results of the MMAE/FS/CR while flying in various flight conditions.

Chapter 5 - Results

5.1 Chapter Introduction

This chapter provides the results and analysis of this research effort. Each section will introduce the concept, then present the results, and finally present an analysis of the results. The test flight maneuver and flight conditions used to test the performance of the MMAE/FS/CR are explained in Section 5.2. Section 5.3 presents the performance of the MMAE/FS detection capabilities. Section 5.4 shows the capability of the CR to compensate for failed actuators. Finally, Section 5.5 shows the performance of the MMAE/FS/CR.

5.2 Test Flight Maneuvers and Flight Conditions

This section describes the maneuver, flight conditions, and the type of analysis used to form the results. The maneuver is described first, followed by the initial flight conditions. The explanation for the dither removal is then presented. Finally, the Monte Carlo analysis is presented in more detail.

5.2.1 Flight Maneuvers

The original flight condition used in previous research efforts [7, 9-12, 15, 24, 25, 29, 30] has been straight and level flight at the airspeed of Mach 0.4 and an altitude of 20,000ft. Previous research [29, 30] into the application of control used a maneuver called a doublet. A doublet is the application of control in one direction, then the application of control in the opposite direction using the same magnitude and command duration. This type of maneuver is useful in the detection of failures without significantly changing the dynamic pressure or attitude of the aircraft.

In this research effort, the primary maneuver used to test the MMAE/FS/CR is a set of two doublets followed by a long duration commanded maneuver. This maneuver is created in order to detect failures and apply proper control while changing dynamic pressure and aircraft attitude. The doublets provide sufficient aircraft excitation to make failures apparent to the MMAE/FS. The long duration maneuver provides the demonstration of the CR capability to apply proper commands in order to achieve the desired maneuver while the aircraft changes dynamic pressure and attitude.

The longitudinal maneuver used in this research effort consists of two pitch doublets and a long duration pitch maneuver. The failure occurs at one second into the simulation. The first pitch doublet occurs at five seconds and lasts for two seconds. The second pitch doublet occurs at nine seconds and also lasts for two seconds. The doublet maneuvers occur five seconds into the simulation to demonstrate the inability of the MMAE/FS to detect failures without the excitation of the aircraft. In this research effort, the purposeful dither is not used, whereas a final implementation of this algorithm might use dither in the absence of pilot commanded maneuvers. The duration of the doublet maneuvers was chosen to be two seconds because of the aircraft response characteristics. The long duration pitch maneuver occurs at fifteen seconds and lasts five seconds. The long duration maneuver occurs approximately five seconds after the doublets in order to separate the pilot commands and let the aircraft dynamics caused by the doublets settle.

The lateral maneuver consists of two roll doublets conducted with the same time characteristics as the longitudinal maneuver. The lateral maneuver also contains a long duration roll maneuver starting at fifteen seconds into the simulation. The long duration roll maneuver lasts for four seconds to roll the aircraft into a desired 40-degree banked turn. The failure is also added at one second in the lateral case as well.

The complete longitudinal and lateral simulation times are twenty-four seconds. In all maneuvers, the pilot-commanded stick force is five pounds of force. The longitudinal maneuver is used to demonstrate the detection and control performance with stabilator failures and longitudinal sensor failures (angle of attack, pitch rate, and normal acceleration sensors). The lateral maneuver is used to demonstrate the detection and control performance with flaperon and rudder failures and lateral sensor failures (roll rate, yaw rate, and lateral acceleration sensors).

5.2.2. Flight Conditions

The flight conditions at the start of the maneuver are straight and level at the airspeed of Mach 0.47 and an altitude of 20,000ft. This flight condition was chosen because it has a dynamic pressure of approximately150 psf (pounds of force per square foot) whereas previous research efforts conducted at the Mach 0.40 airspeed and an altitude of 20,000ft had a dynamic pressure of approximately 100 psf. This will illustrate the performance of the MMAE/FS/CR at other than the proven original flight conditions.

5.2.3. Dither Removal

The dither signal used in previous research efforts has been removed for this research effort. It is found that the dither signal in its current state produces undesirable aircraft control performance when combined with purposeful, large maneuvers, resulting in false failure detections created by the MMAE/FS. The dither signal also uses a large percentage of the available actuator strength as shown in Table 3. The removal of the dither signal is discussed later in this chapter.

Control Channel	Maximum Magnitude	Dither Magnitude	Actuator Percentage Used
Pitch	±30.25 lbs	±12.00 lbs	39.7%
Roll	±17.00 lbs	±11.00 lbs	64.7%
Yaw	±160.0 lbs	±30.00 lbs	18.8%

Table 3: Dither Magnitudes Compared to Maximum Magnitudes Per Channel

5.2.4. Monte Carlo Analysis Definition

The performance of the MMAE/FS/CR is evaluated using a Monte Carlo analysis. Each Monte Carlo analysis is made up of ten separate simulations. Each simulation uses a different random noise sample but all simulations use the same noise strength for each noise within the overall simulation.

5.3 Failure Detection Performance

5.3.1 Failure Detection

The first major portion of this research effort is demonstrating the capability to detect complete and partial actuator failures and complete sensor failures while undergoing maneuvers and changing dynamic pressure. The longitudinal characteristics of the MMAE/FS are analyzed first, followed by the lateral characteristics. The long duration longitudinal maneuver specifically shows the performance of the MMAE/FS to detect failures during a change in dynamic pressure, due to the large changes in velocity and, to a lesser degree, altitude. The initial dynamic pressure before the long duration maneuver is approximately 150 psf. After the long duration maneuver, the dynamic pressure changes to 130 psf. In the case of actuator failures, CR based on knowing ε_{true} is used in order to analyze the MMAE/FS performance thoroughly. Later, in Section 5.4, the *full* MMAE/FS/CR algorithm with failures will be compared both to a fully functional aircraft commanded to maneuver (a baseline of what is actually desired from the aircraft) and to this case of the CR artificially knowing the ε_{true} (a second baseline indicating the best one could hope to accomplish via any CR algorithm in replicating the desired aircraft maneuvering). The following plots show the elemental probabilities for each complete sensor and actuator failure and each partial actuator failure. The plot presentation order is presented on the next page.

Surface Failures

1. Fully Functional Aircraft
2. Fully Failed Left Stabilator
3. Partially Failed Left Stabilator
4. Fully Failed Right Stabilator
5. Partially Failed Right Stabilator
6. Fully Failed Left Flaperon
7. Partially Failed Left Flaperon
8. Fully Failed Right Flaperon
9. Partially Failed Right Flaperon
10. Fully Failed Rudder
11. Partially Failed Rudder

Longitudinal Sensor Failures

12. Failed Angle of Attack Sensor
13. Failed Pitch Rate Sensor
14. Failed Normal Acceleration Sensor

Lateral Sensor Failures

15. Failed Roll Rate Sensor
16. Failed Yaw Rate Sensor
17. Failed Lateral Acceleration Sensor

These plots show the mean value of the Monte Carlo Simulations as the solid line and the mean ± one-sigma (standard deviation) bands are each shown with a dotted line. The simulation time is 24 seconds with the failure initially occurring at one second into the simulation. The x-axis of each plot represents the time in seconds.

On the first set of elemental probability plots (denoted as “Figure _a”), the probabilities corresponding to the elemental filters hypothesizing a fully functional aircraft and full actuator/surface failures (left stabilator failure, right stabilator failure, left flaperon failure, right flaperon failure, and rudder failure) are shown on the top of the page. On the lower half of the page, the probabilities associated with elemental filters based on assumed sensor failures are presented (angle of attack, pitch rate, normal acceleration, roll rate, yaw rate, and lateral acceleration sensor failures). On the second page of elemental probabilities (denoted as

"Figure _b"), the elemental probabilities corresponding to the spawned filters are presented as well as the "combined" probability (formed as the sum of the probabilities for the three spawned filters plus the probability of the associated fully-failed-actuator filter) and the estimates of effectiveness ε_{hat} and ε_{map}. The "combined" probability is presented to establish a method to detect partial actuator/surface failures faster by analyzing the elemental probability in a failed channel rather than just in each elemental filter separately. The ε_{hat} and ε_{map} plots are presented to determine which estimate of effectiveness produces a more accurate estimate during pilot-commanded maneuvering. The solid line on the effectiveness plots represents the mean value of effectiveness and the dotted line represents the true value. The mean ± one standard deviation is not shown on the plots for clarity, since the standard deviation is very small. The effectiveness in the plots shows the initial effectiveness value to be 100% fully functional. In actual implementation, the MMAE/FS would not calculate an estimate of effectiveness until after a failure is declared. In this research, the spawned filters are continuously running from the start of the simulation to allow easier software implementation (i.e., there are always 15 elemental filters and never 12 elemental filters, despite the discussion surrounding Table 2 in Chapter 3). The partial failure hypothesis for the spawned filters is initially started (arbitrarily) as a partially failed left stabilator. Since the MMAE/FS has not declared a failure, the bank initial condition does not affect the performance of the MMAE/FS. As a result of the initial condition, the estimate of effectiveness shows 100%, meaning that the Filter Spawning is using the left stabilator partial failure hypothesis in the three spawned filters, and the effectiveness of that actuator (left stabilator in this case) is declared to be 100%, or fully functional. The "desired" plot in each of the figures shows the pilot-commanded maneuver in pounds of stick force, so the elemental probabilities (and later the state values) can be compared directly to the purposeful pilot commands.

Figures 5 – 34 on failure detection performance are presented as a group on the following pages. Then, starting on page 119, they are discussed in detail.

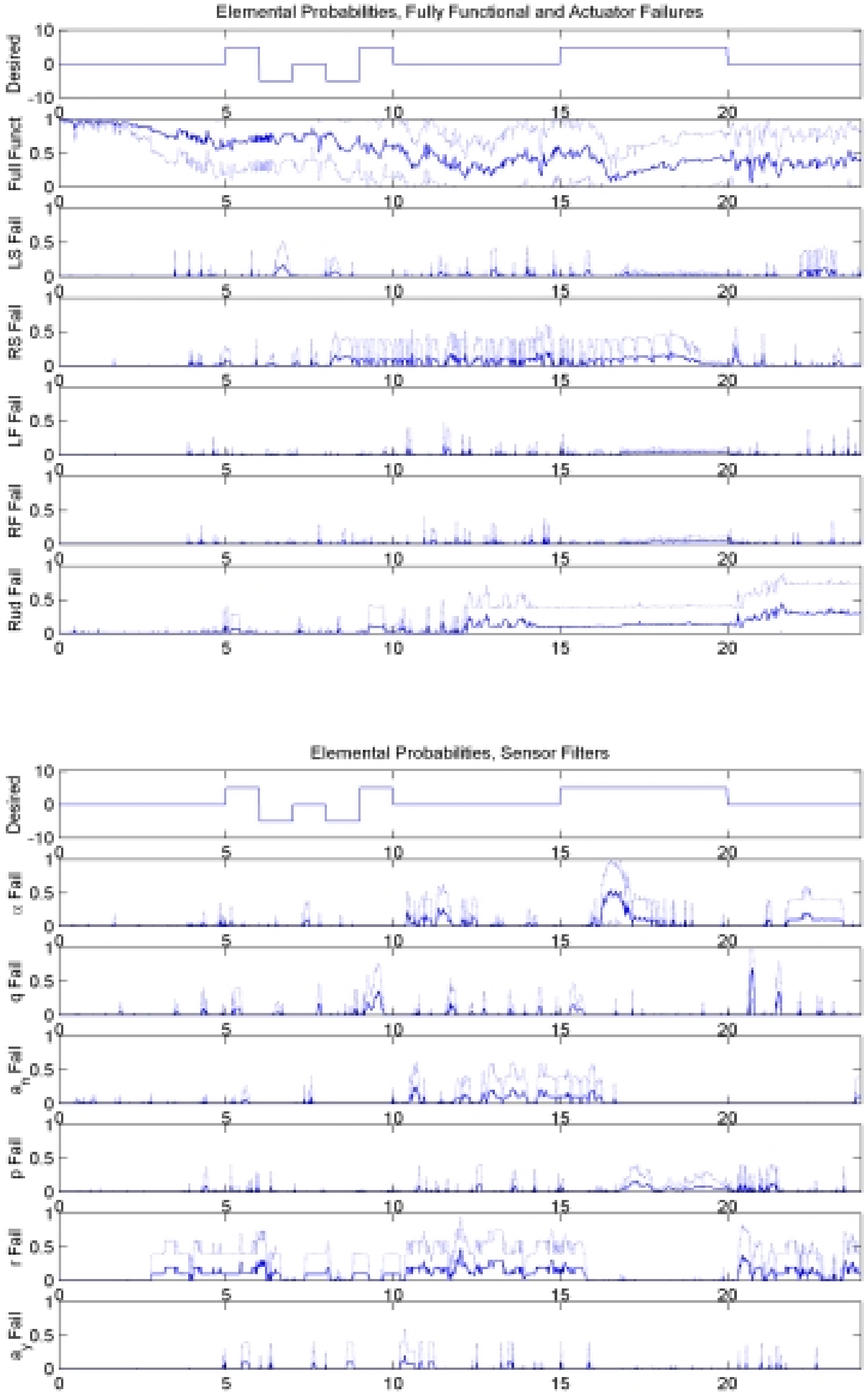


Figure 5a: Elemental Probabilities, Fully Functional Aircraft, Pitch Maneuver With Dither

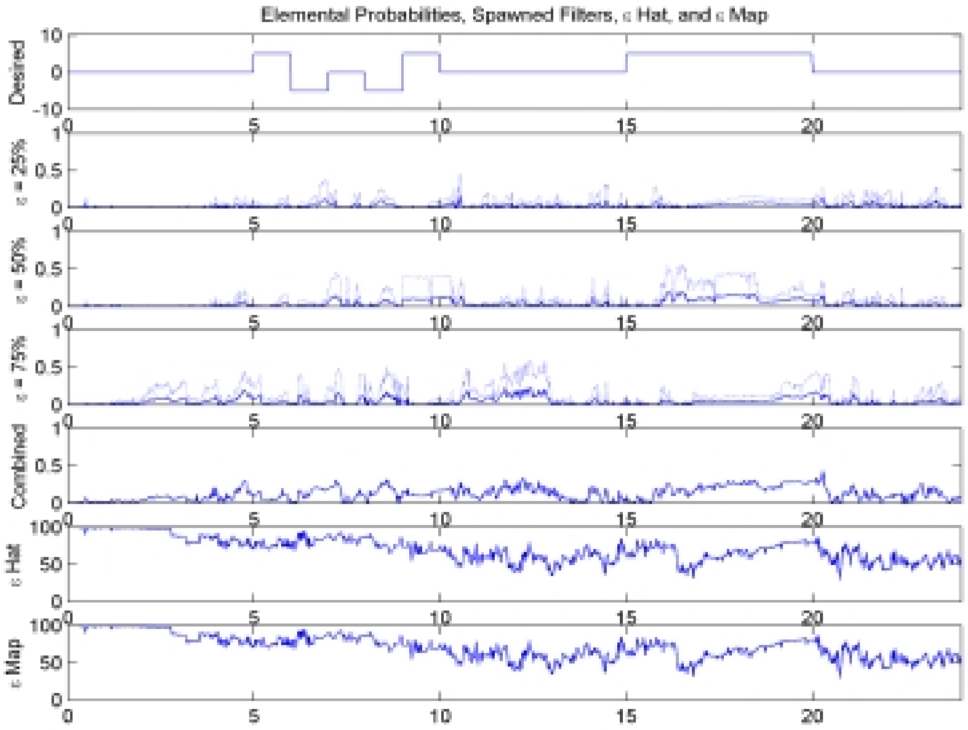


Figure 5b: Elemental Probabilities, Fully Functional Aircraft, Pitch Maneuver With Dither

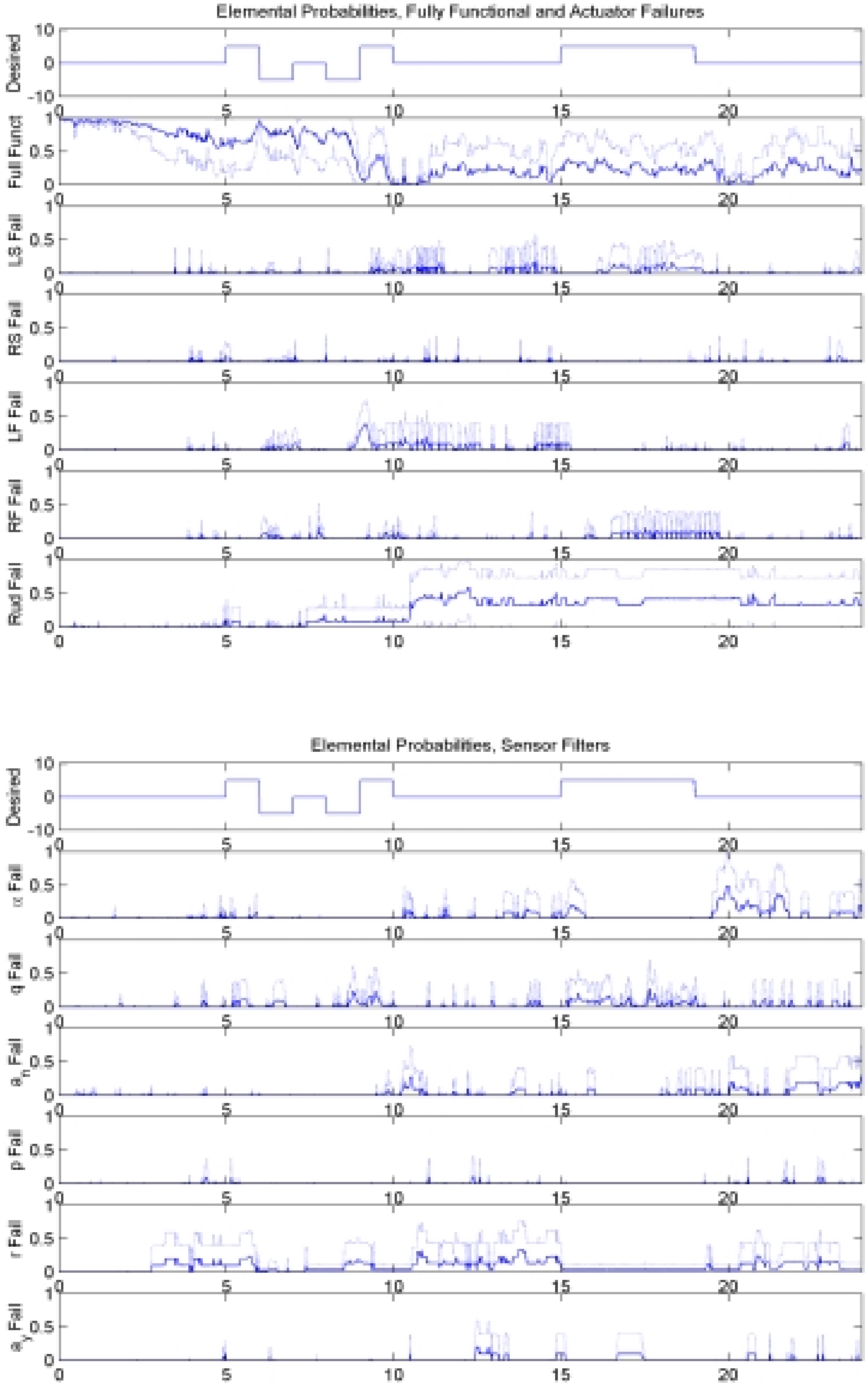


Figure 6a: Elemental Probabilities, Fully Functional Aircraft, Roll Maneuver With Dither

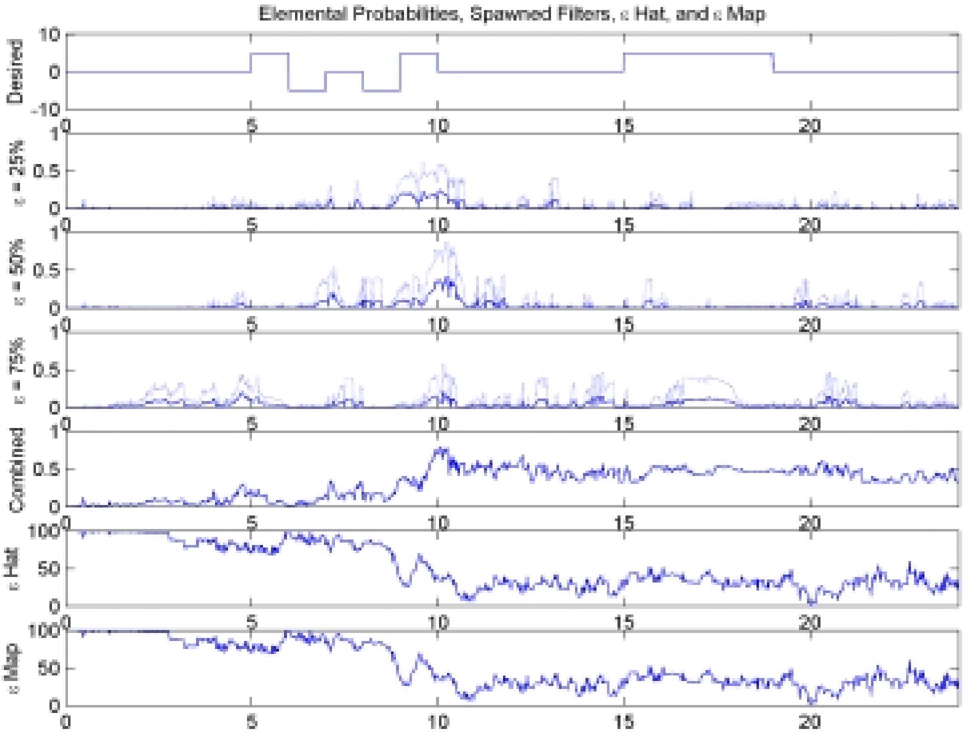


Figure 6b: Elemental Probabilities, Fully Functional Aircraft, Roll Maneuver With Dither

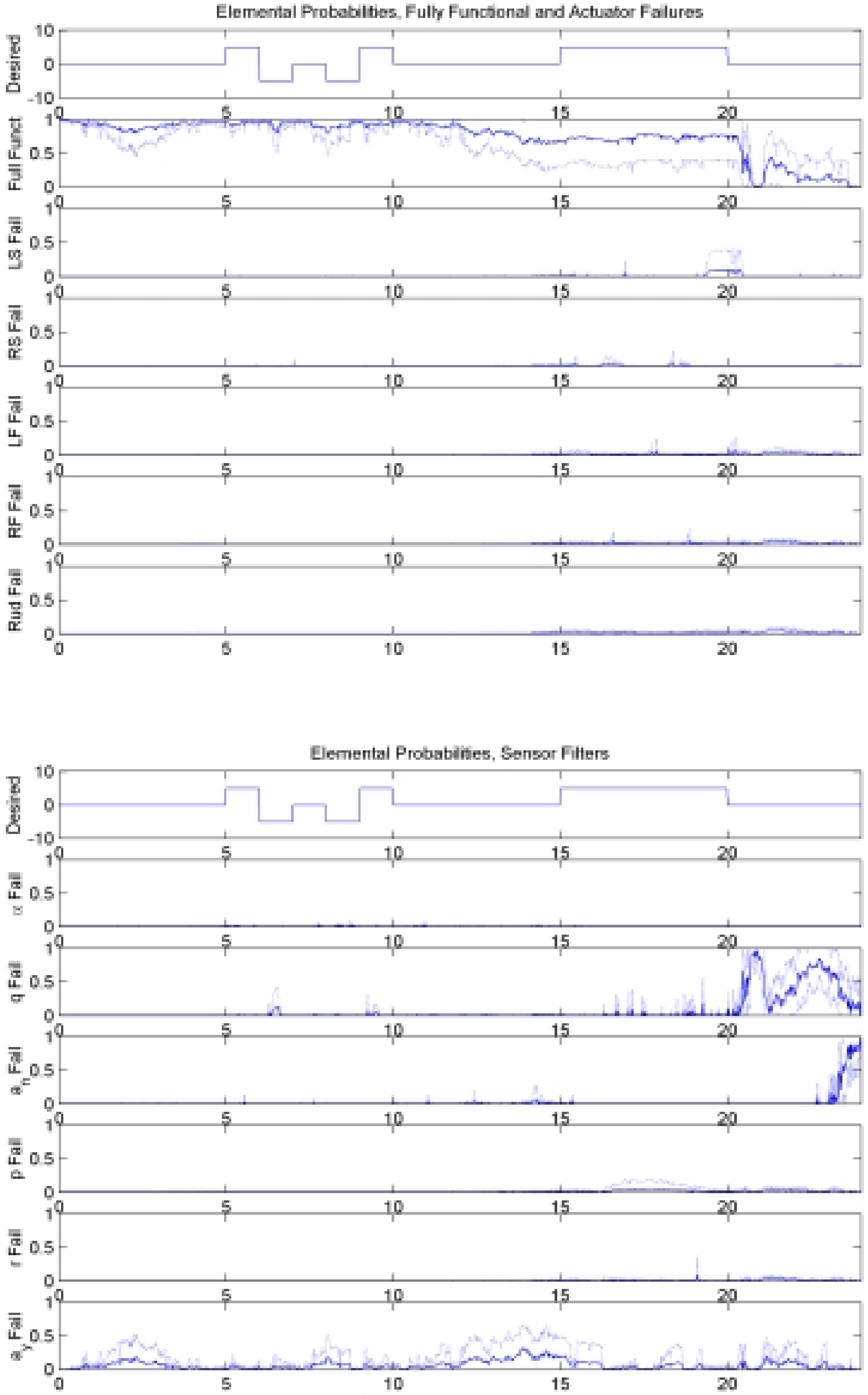


Figure 7a: Elemental Probabilities, Fully Functional Aircraft, Pitch Maneuver

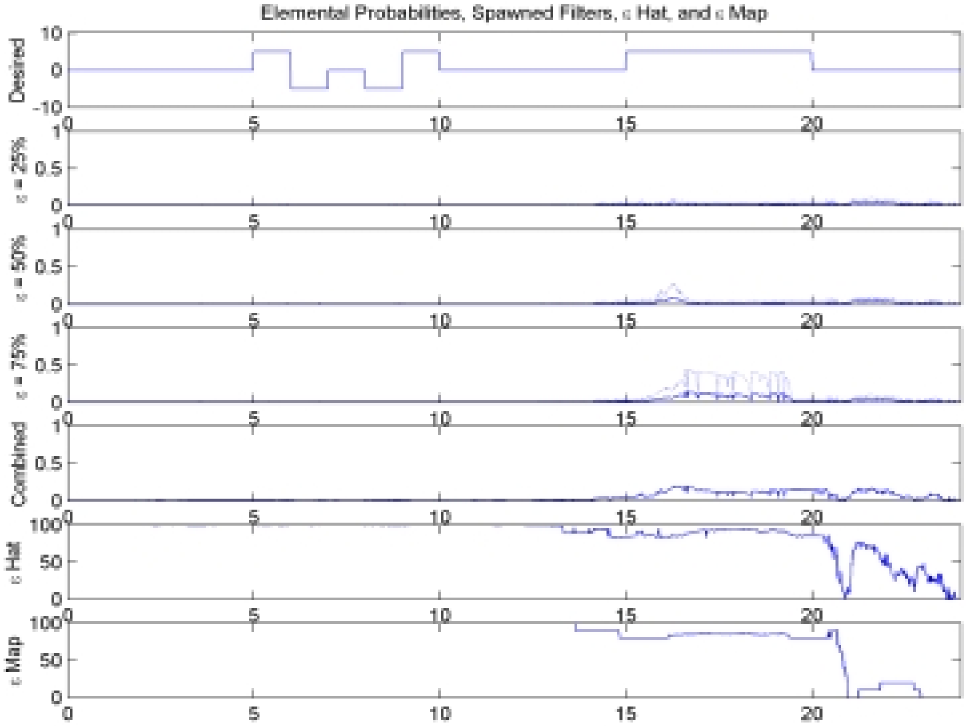


Figure 7b: Elemental Probabilities, Fully Functional Aircraft, Pitch Maneuver

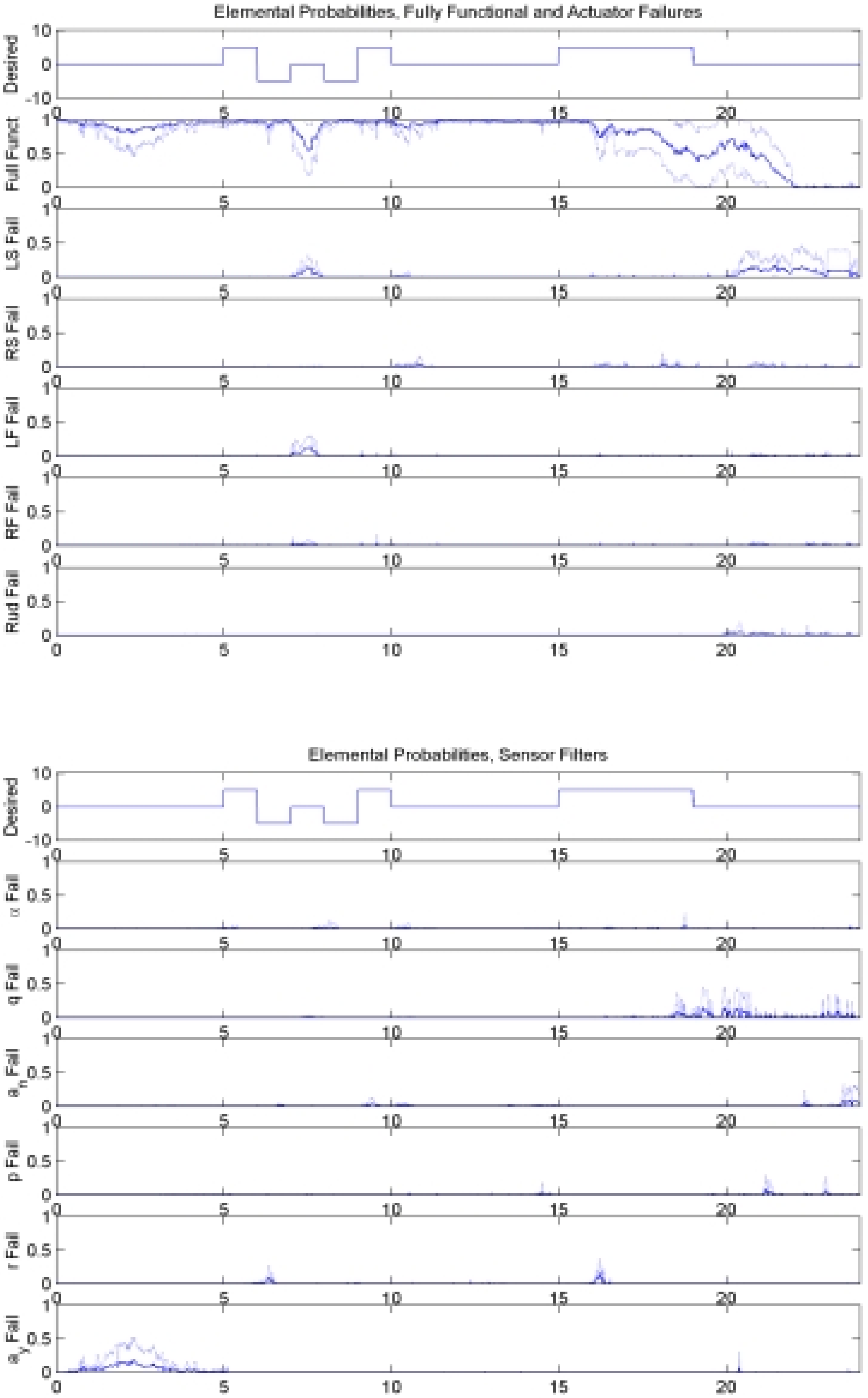


Figure 8a: Elemental Probabilities, Fully Functional Aircraft, Roll Maneuver

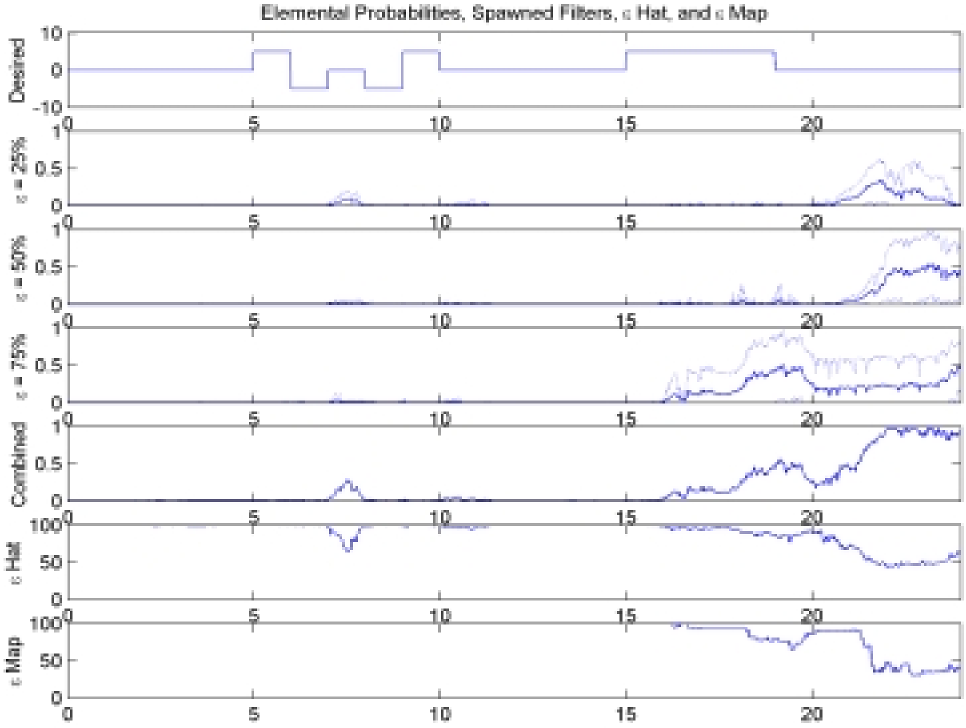
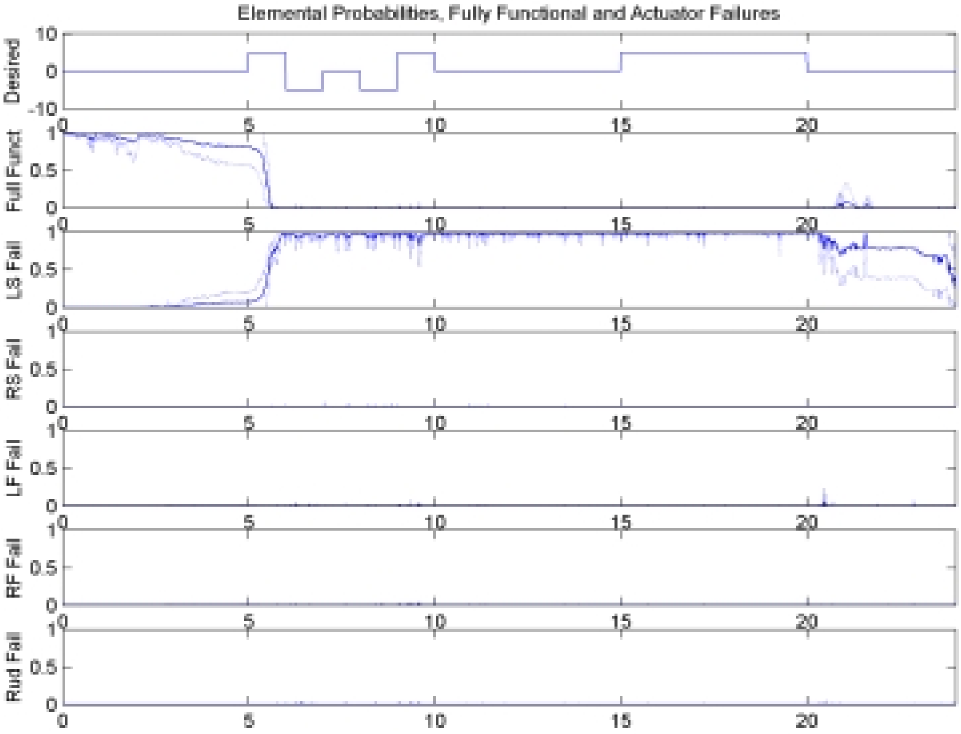


Figure 8b: Elemental Probabilities, Fully Functional Aircraft, Roll Maneuver



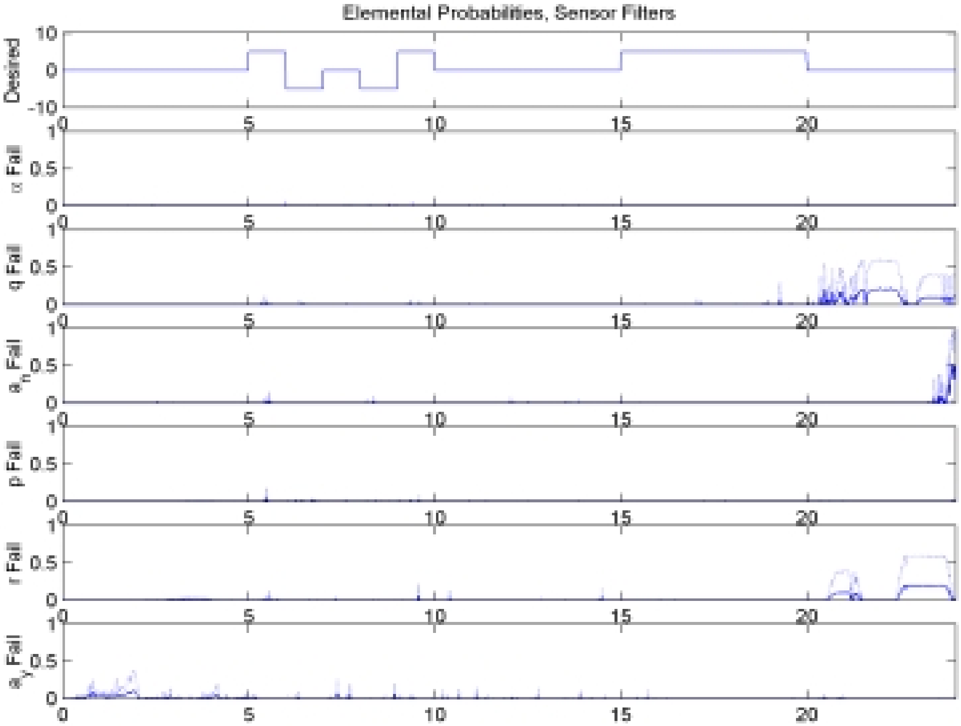


Figure 9a: Elemental Probabilities, Left Stabilator Failure, ε = 0%

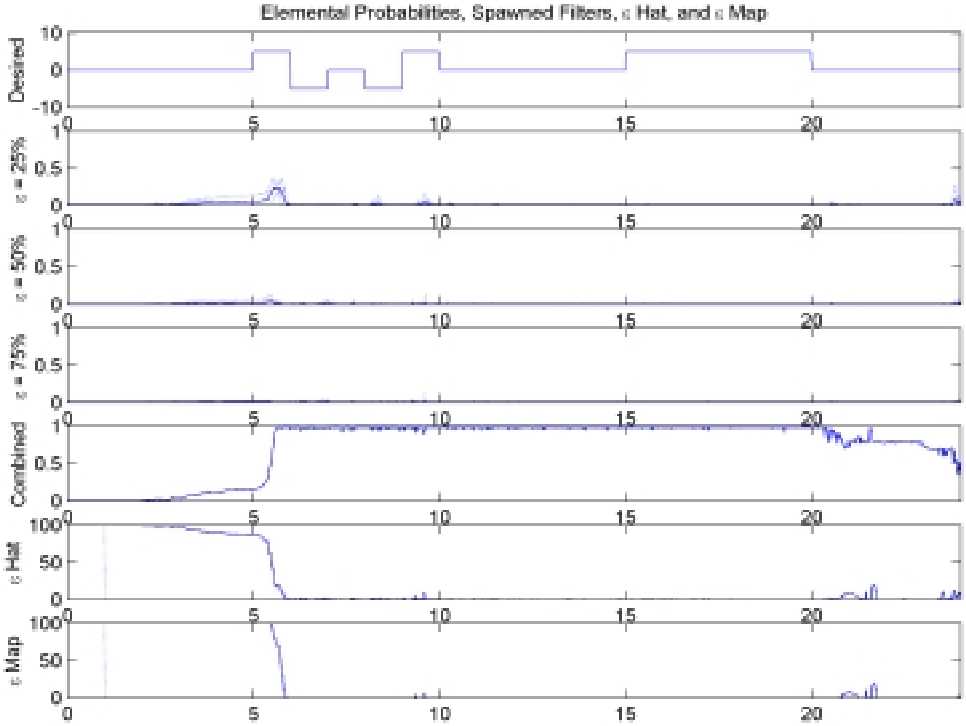


Figure 9b: Elemental Probabilities, Left Stabilator Failure, ε = 0%

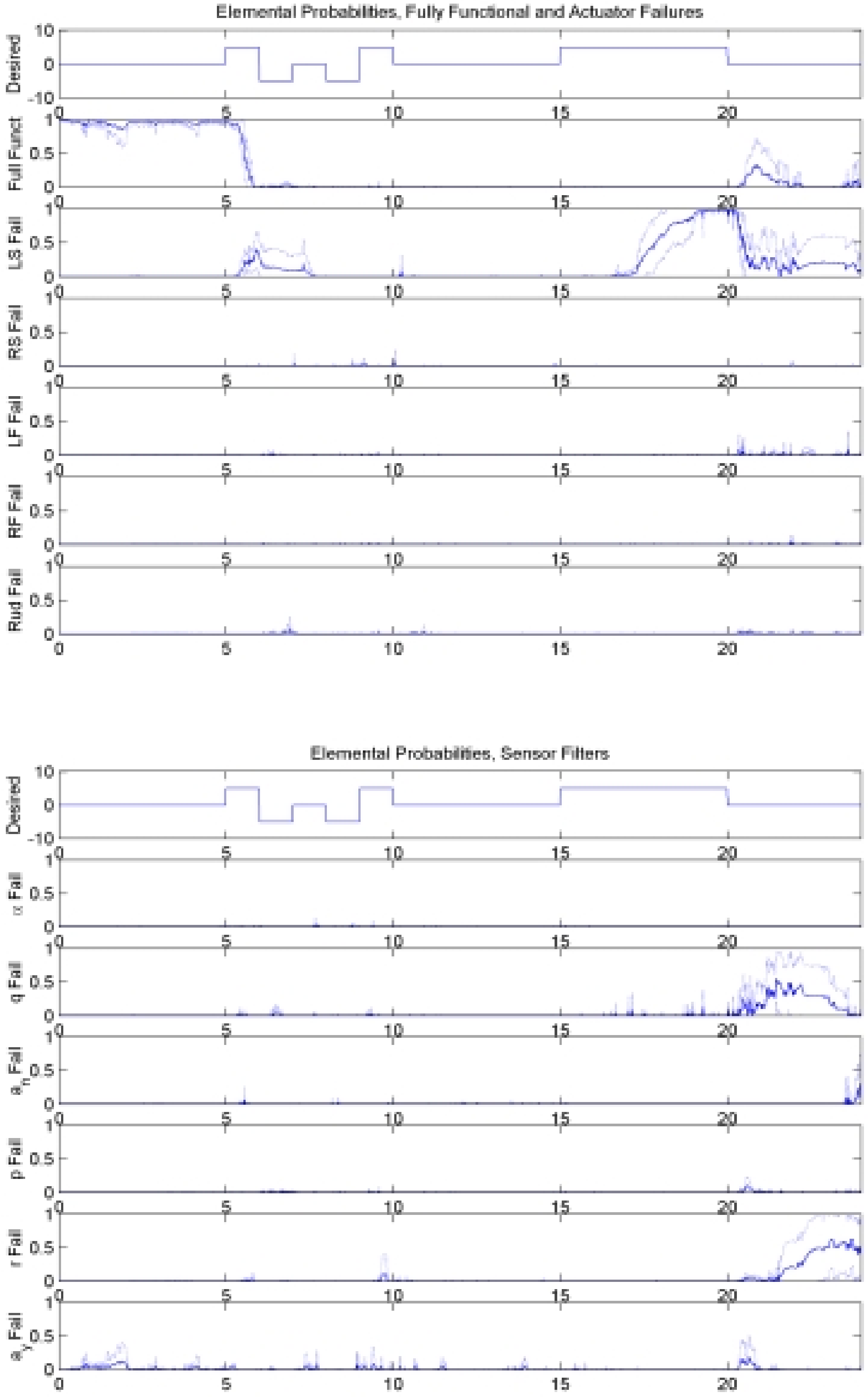


Figure 10a: Elemental Probabilities, Left Stabilator Failure, ε = 25%

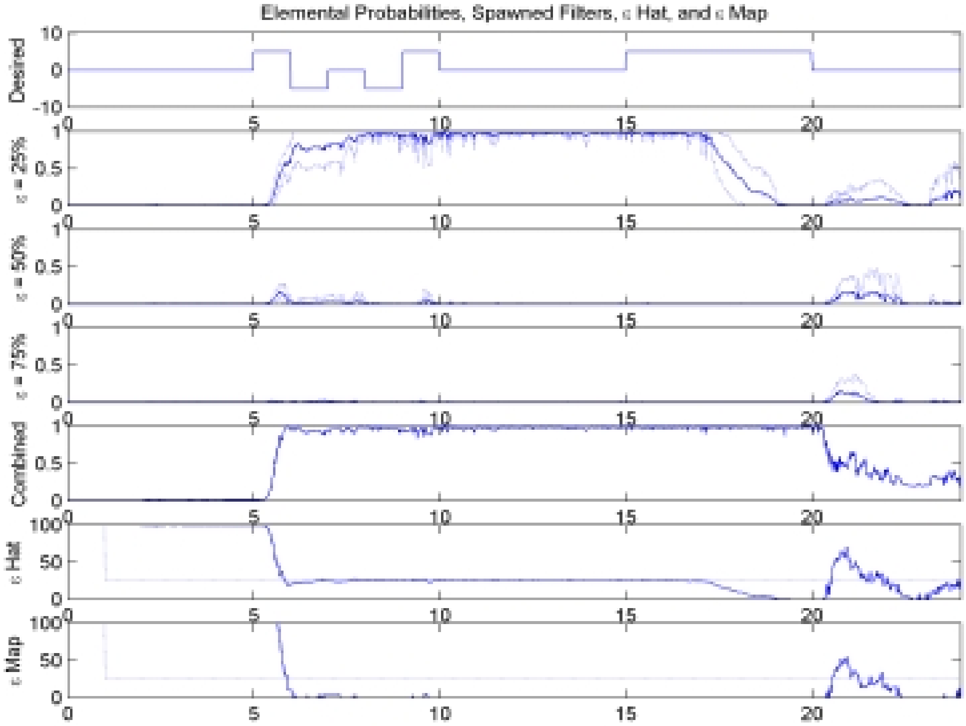
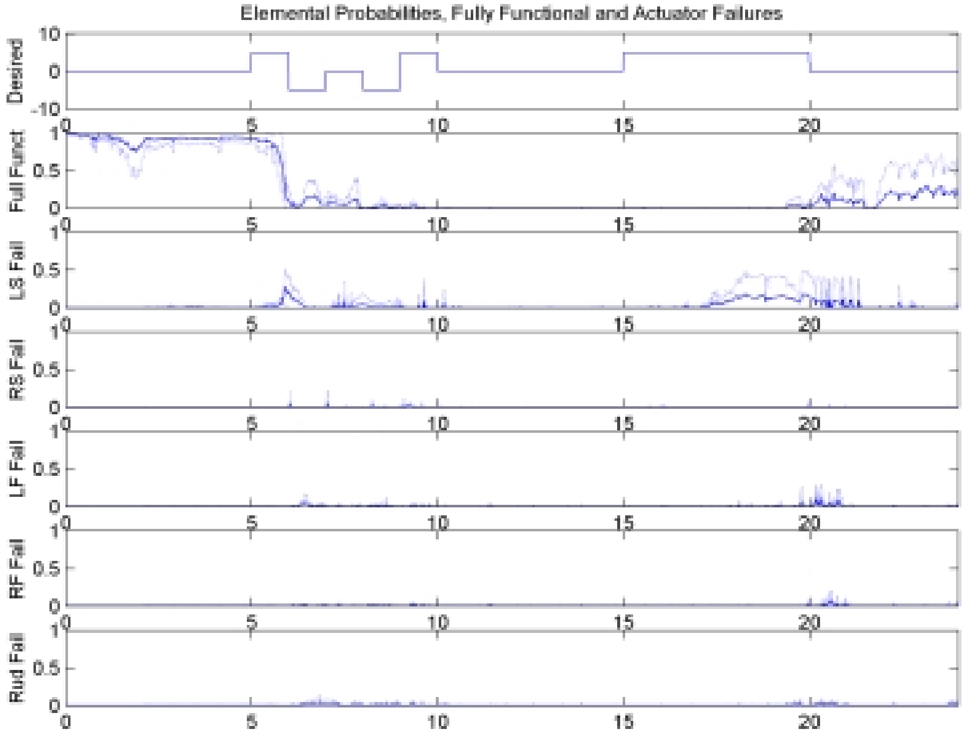


Figure 10b: Elemental Probabilities, Left Stabilator Failure, ε = 25%



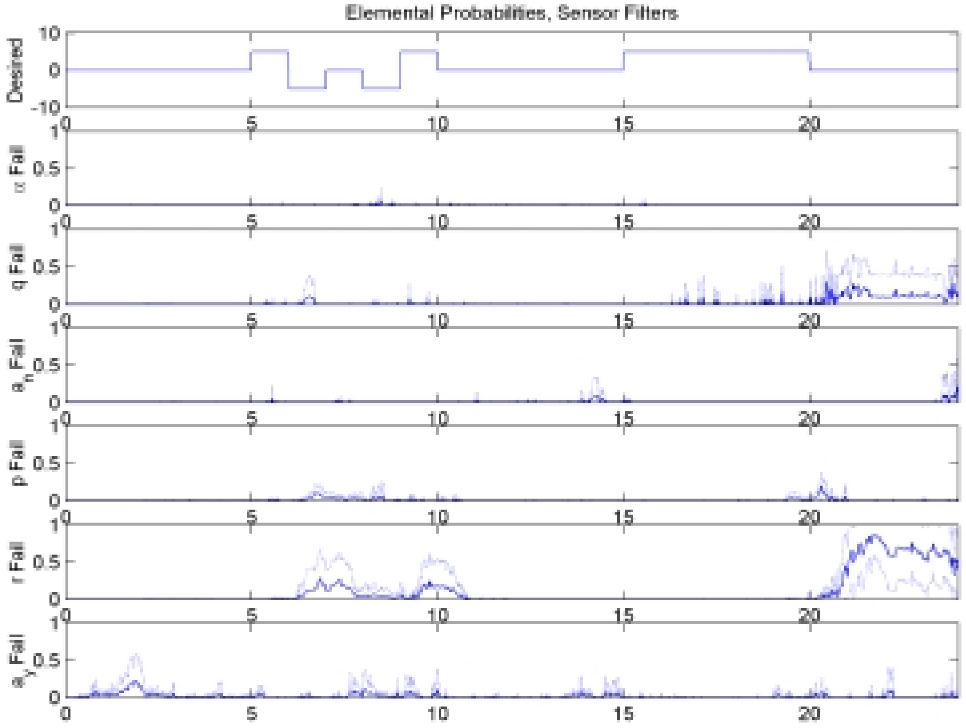


Figure 11a: Elemental Probabilities, Left Stabilator Failure, ε = 50%

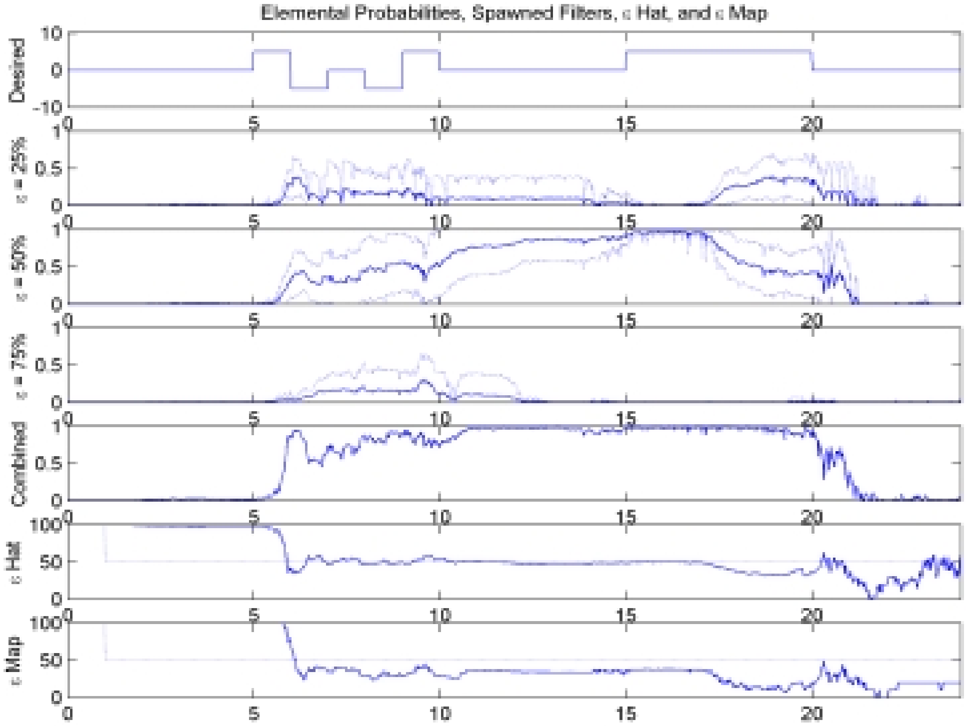


Figure 11b: Elemental Probabilities, Left Stabilator Failure, ε = 50%

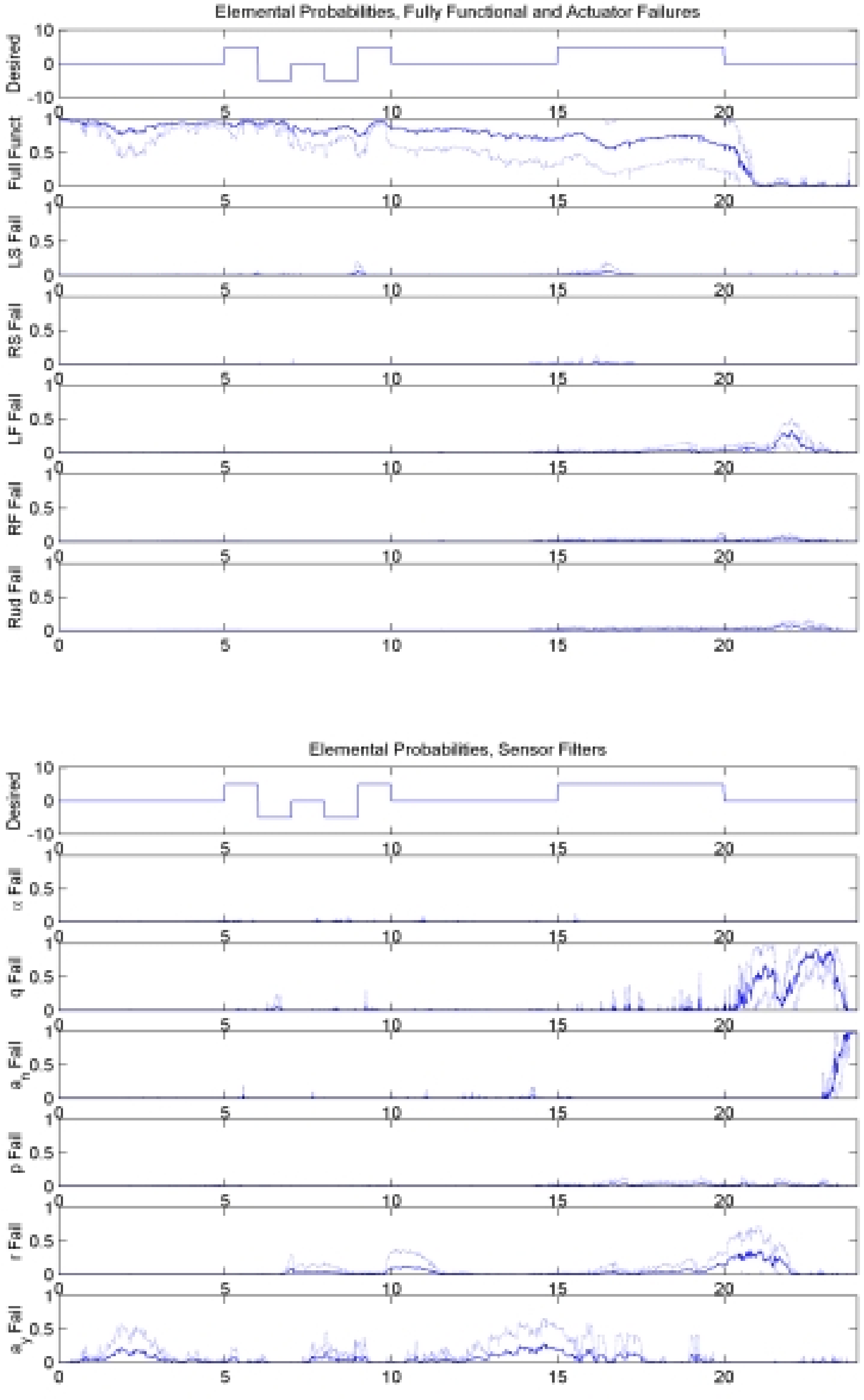


Figure 12a: Elemental Probabilities, Left Stabilator Failure, ε = 75%

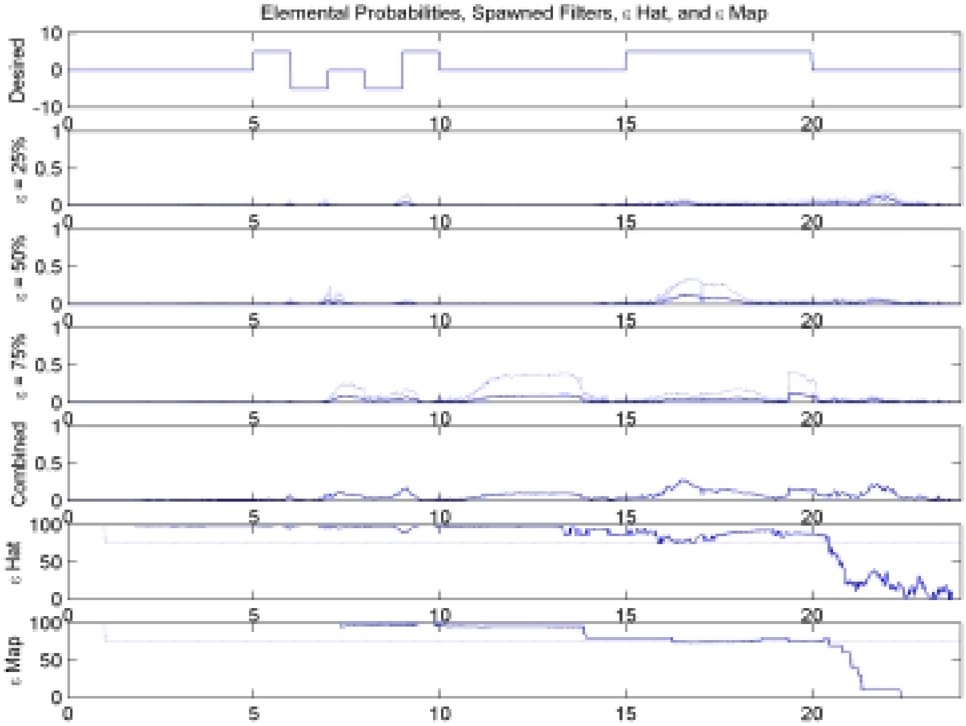


Figure 12b: Elemental Probabilities, Left Stabilator Failure, ε = 75%

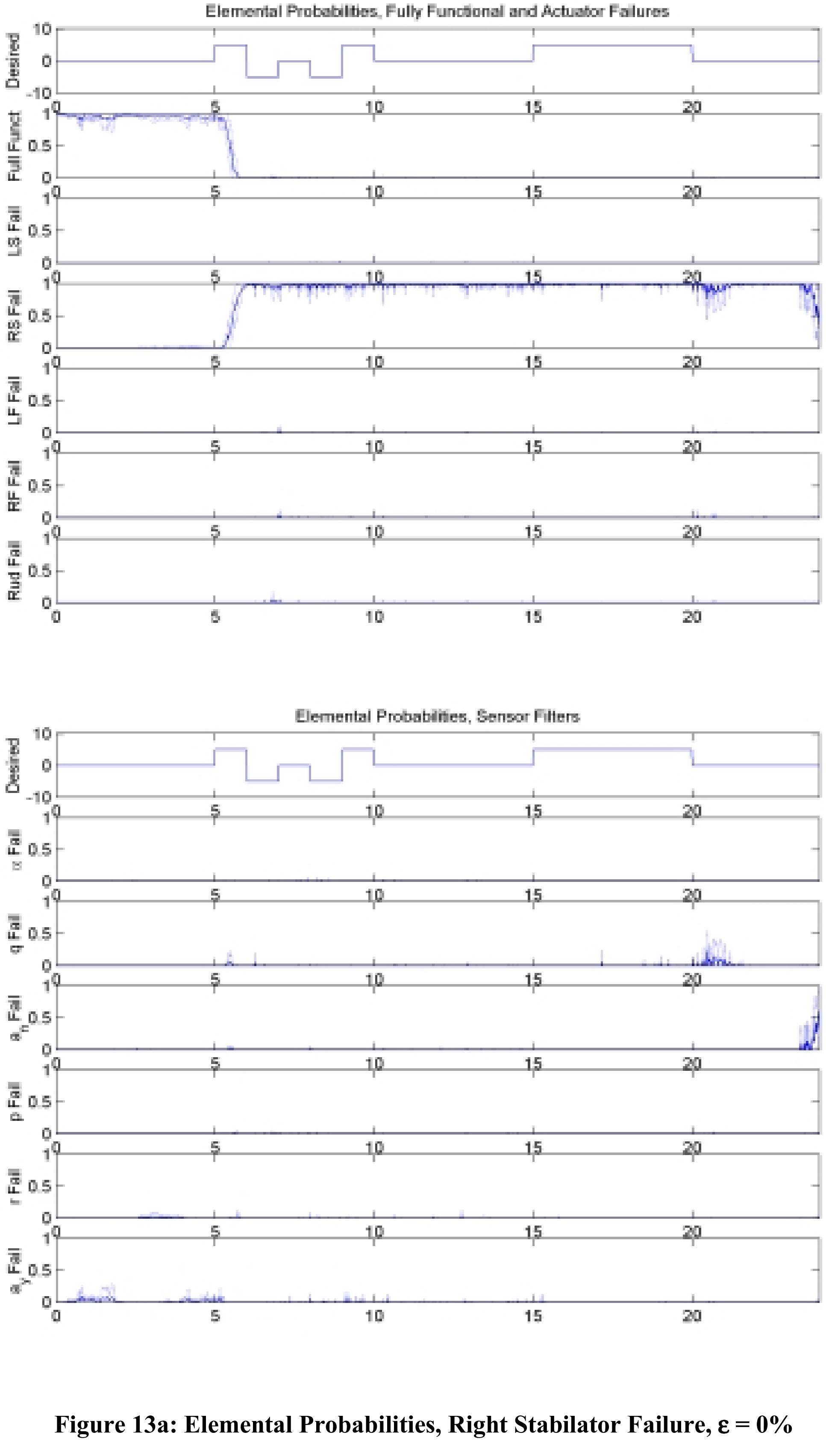


Figure 13a: Elemental Probabilities, Right Stabilator Failure, ε = 0%

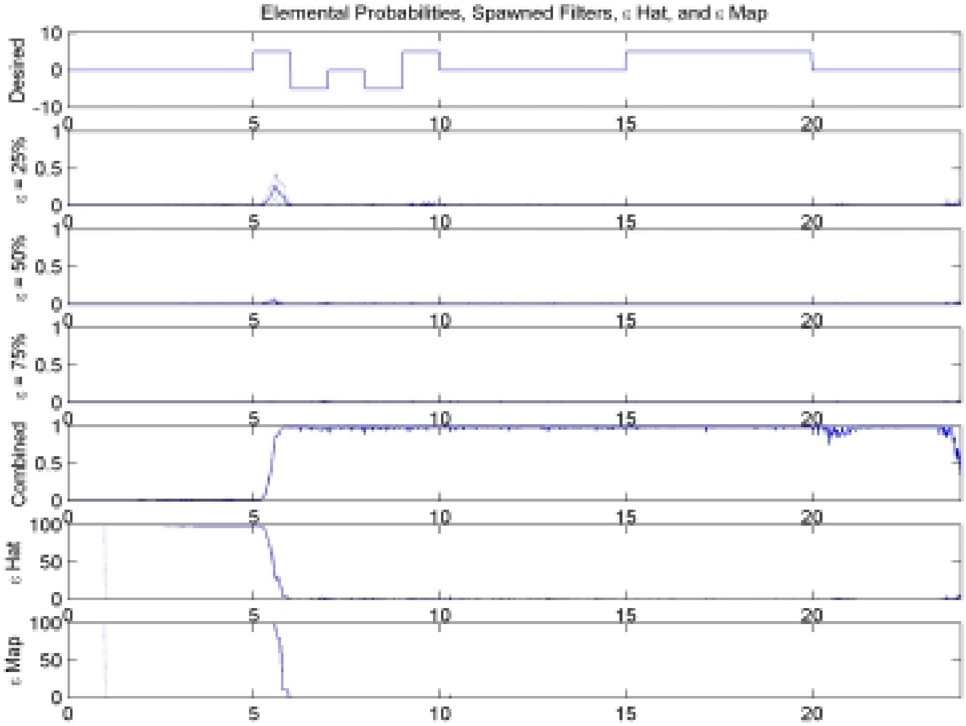
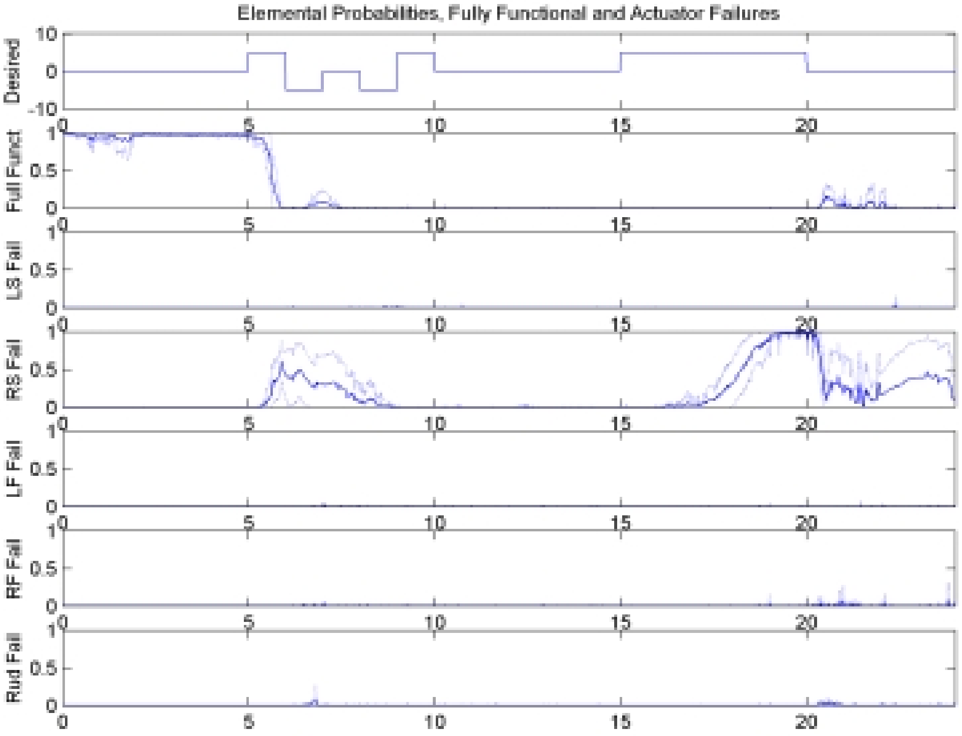


Figure 13b: Elemental Probabilities, Right Stabilator Failure, ε = 0%



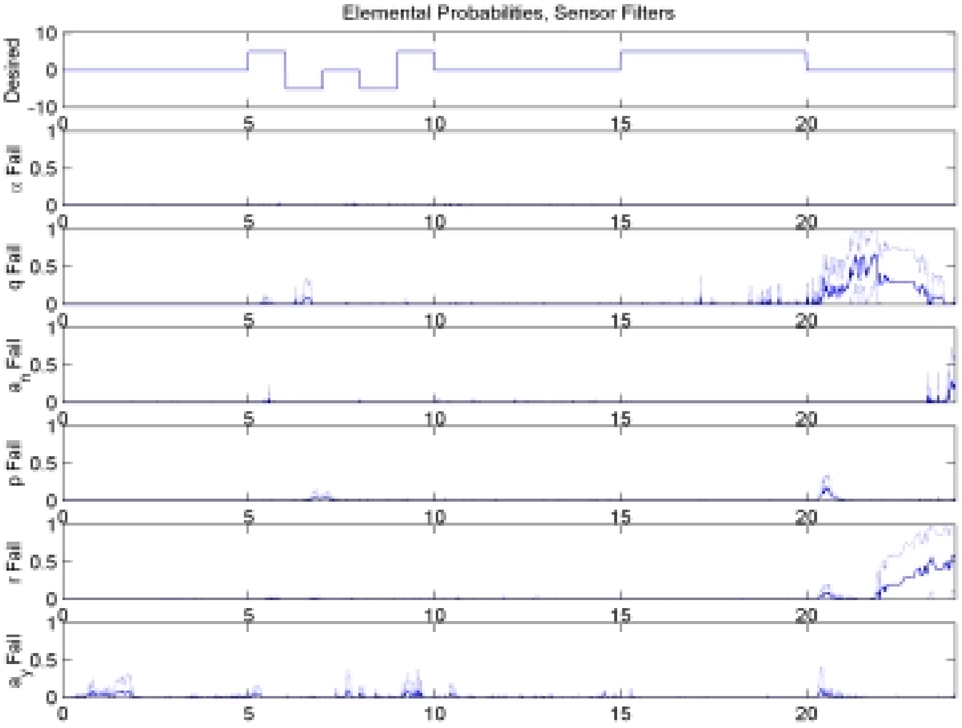


Figure 14a: Elemental Probabilities, Right Stabilator Failure, ε = 25%

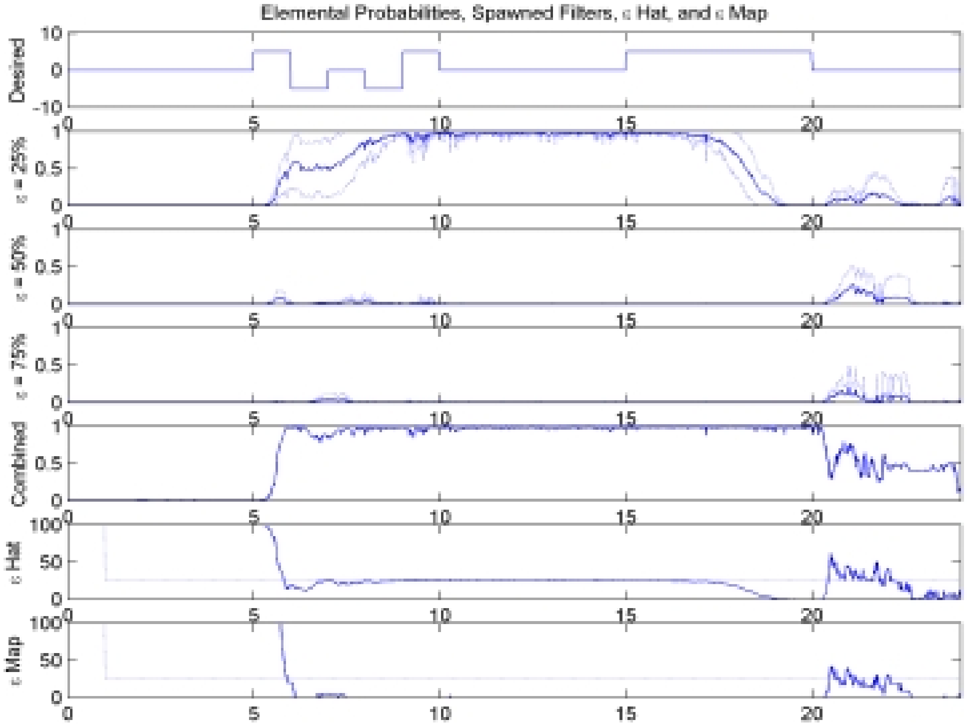


Figure 14b: Elemental Probabilities, Right Stabilator Failure, ε = 25%

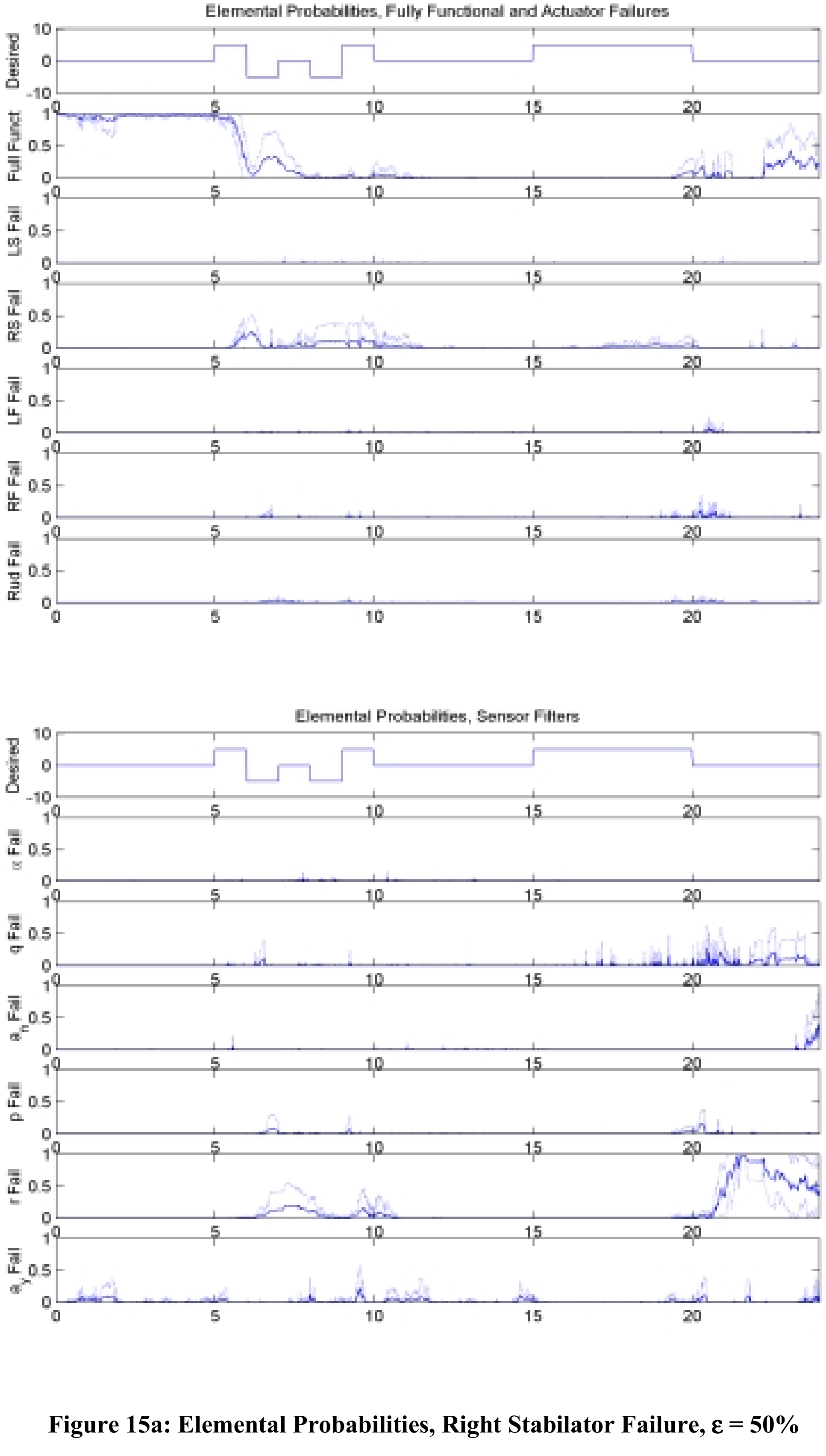


Figure 15a: Elemental Probabilities, Right Stabilator Failure, ε = 50%

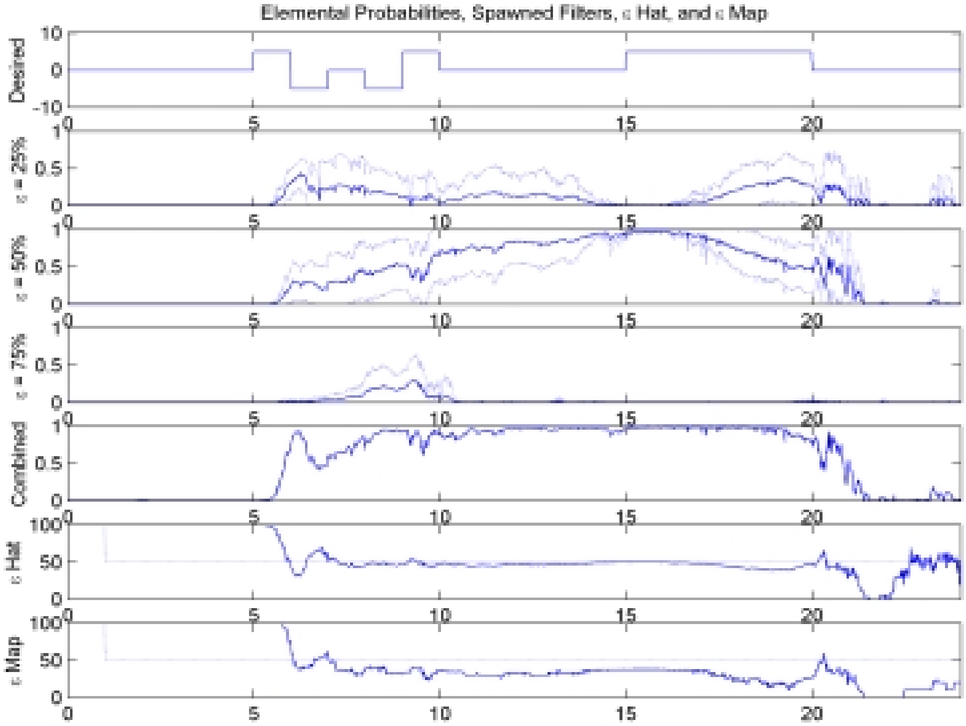


Figure 15b: Elemental Probabilities, Right Stabilator Failure, ε = 50%

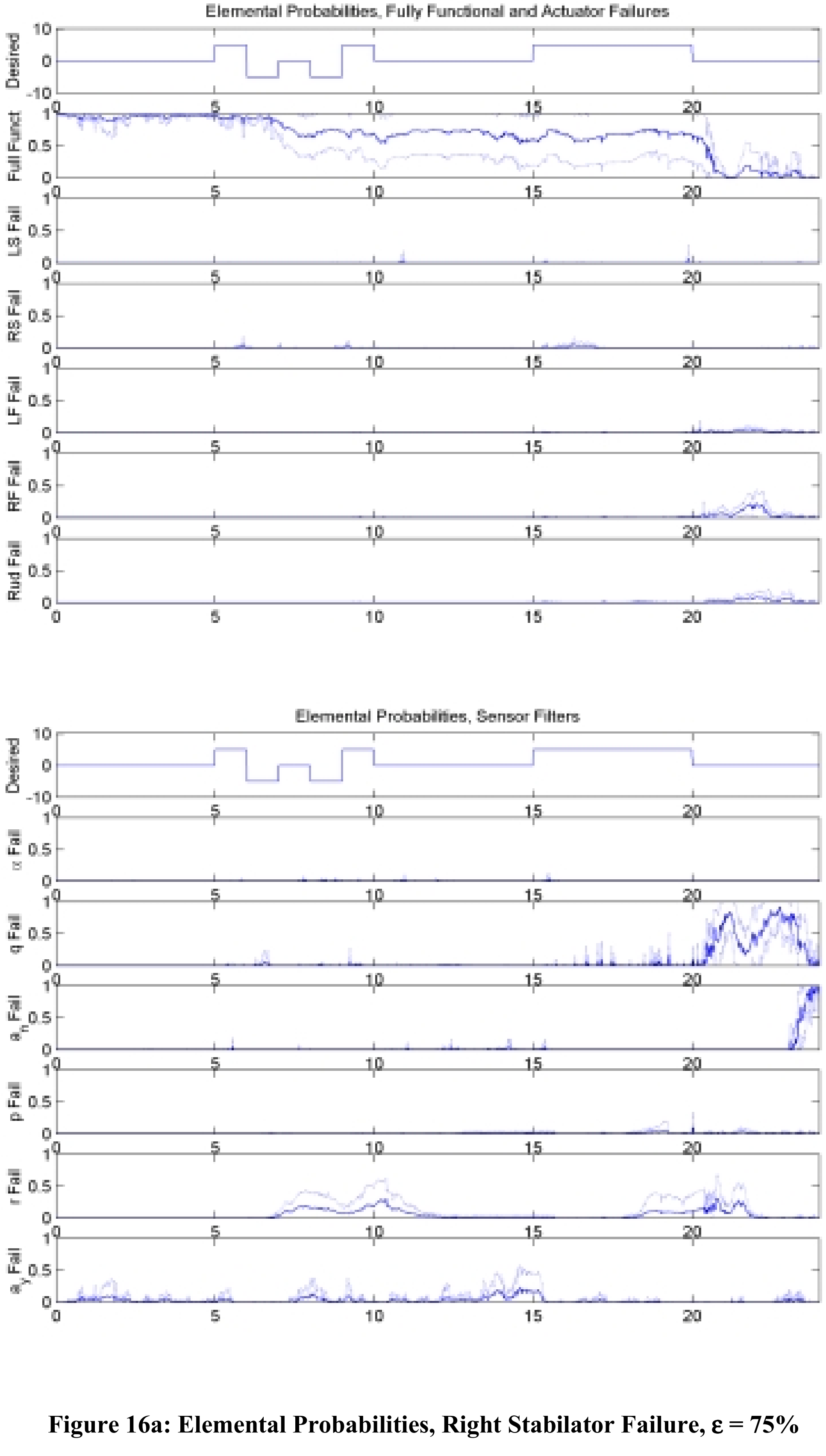


Figure 16a: Elemental Probabilities, Right Stabilator Failure, ε = 75%

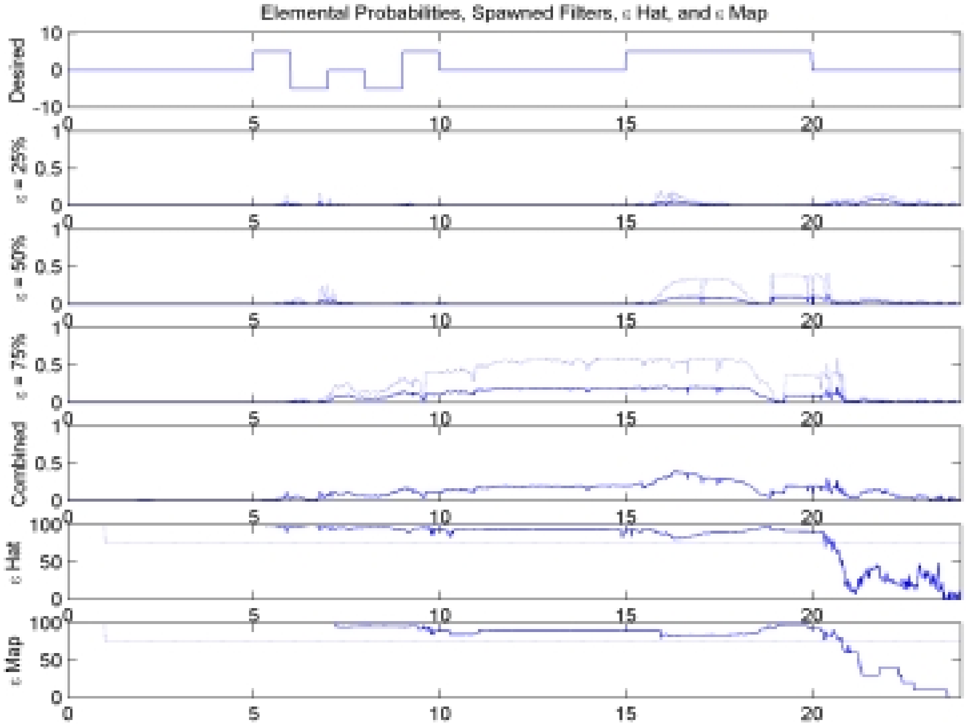


Figure 16b: Elemental Probabilities, Right Stabilator Failure, ε = 75%

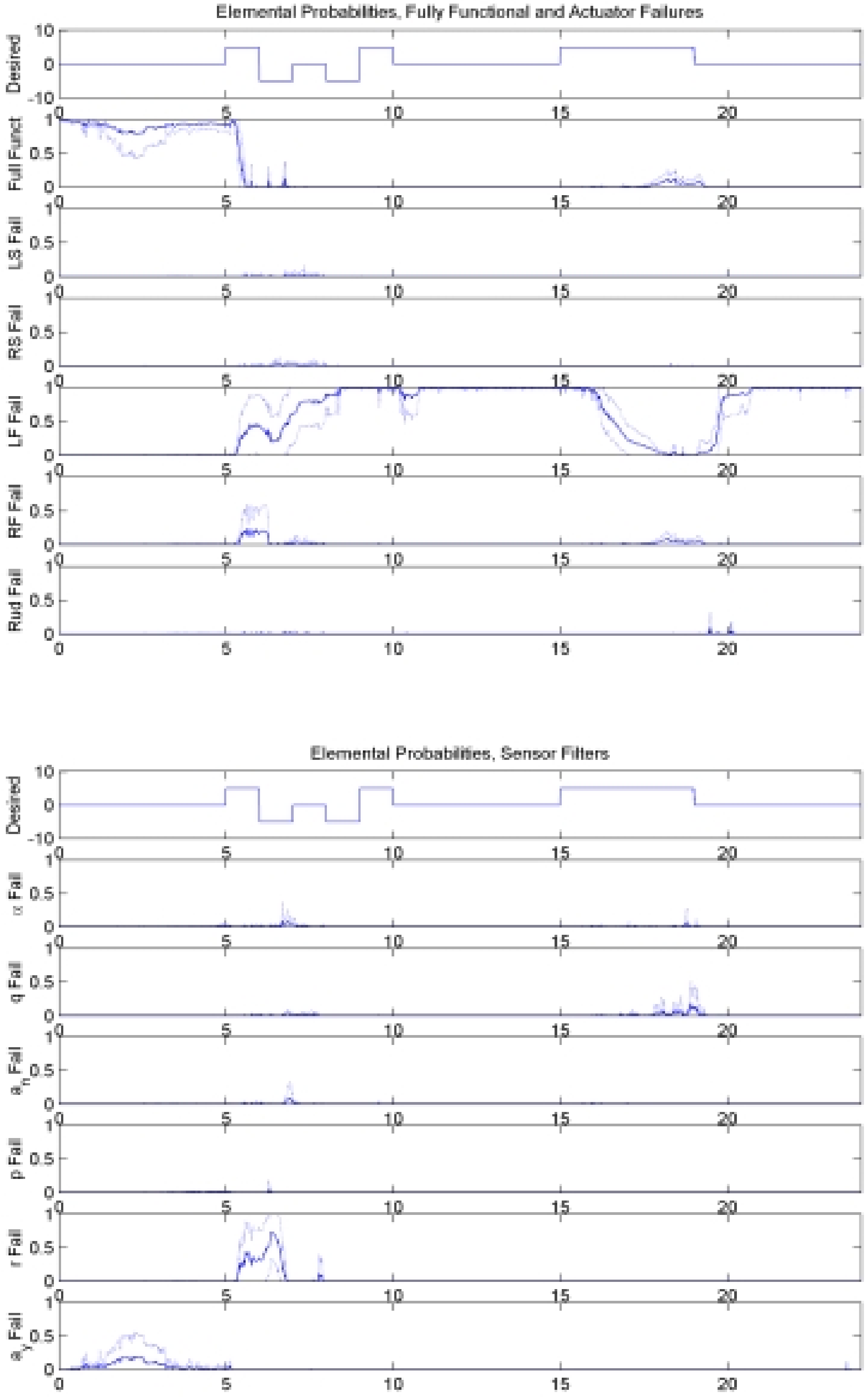


Figure 17a: Elemental Probabilities, Left Flaperon Failure, ε = 0%

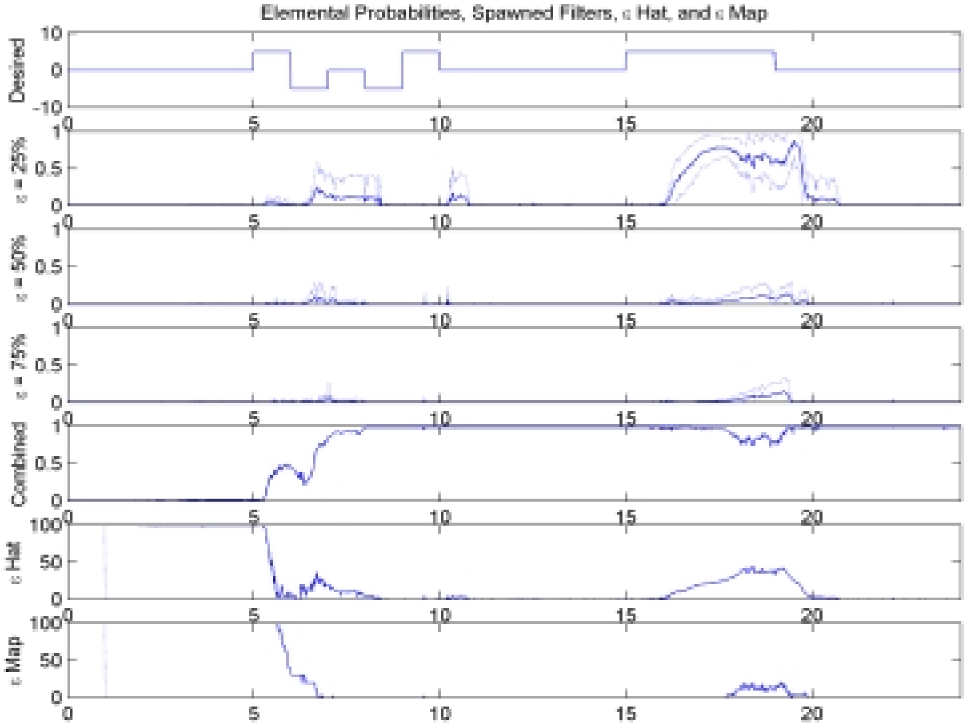


Figure 17b: Elemental Probabilities, Left Flaperon Failure, ε = 0%

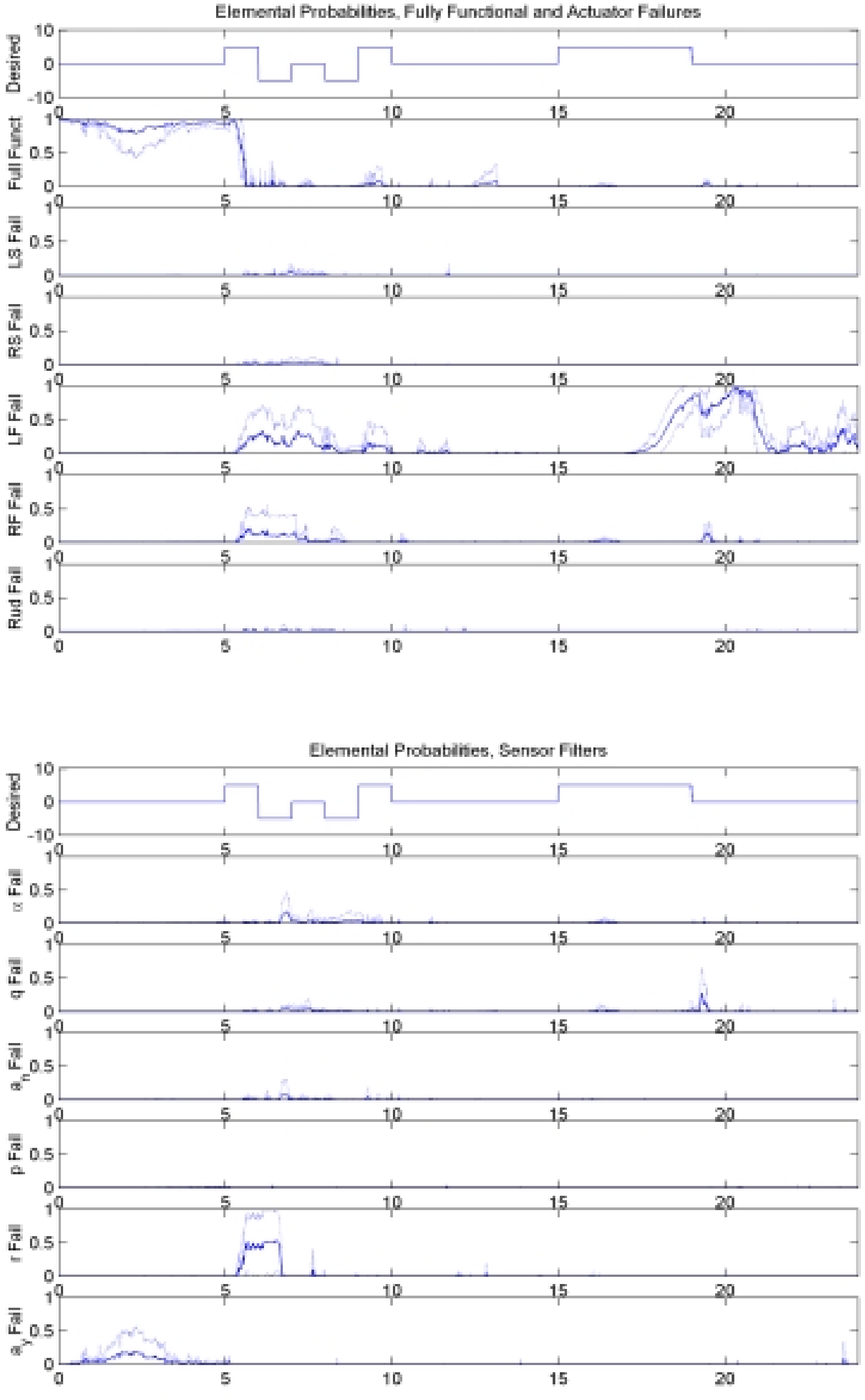


Figure 18a: Elemental Probabilities, Left Flaperon Failure, ε = 25%

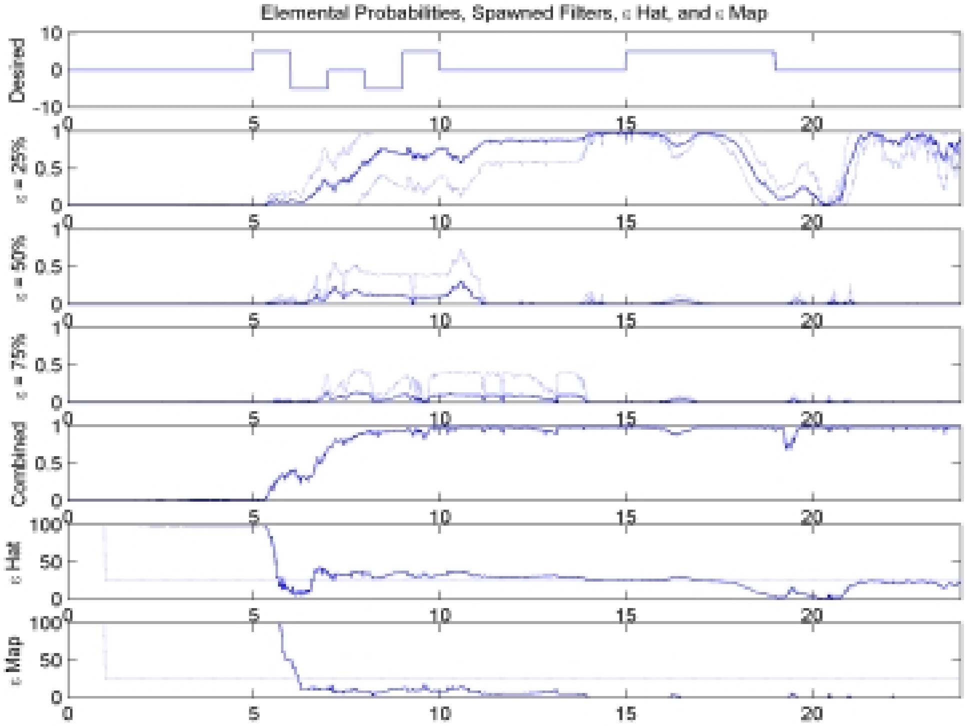


Figure 18b: Elemental Probabilities, Left Flaperon Failure, ε = 25%

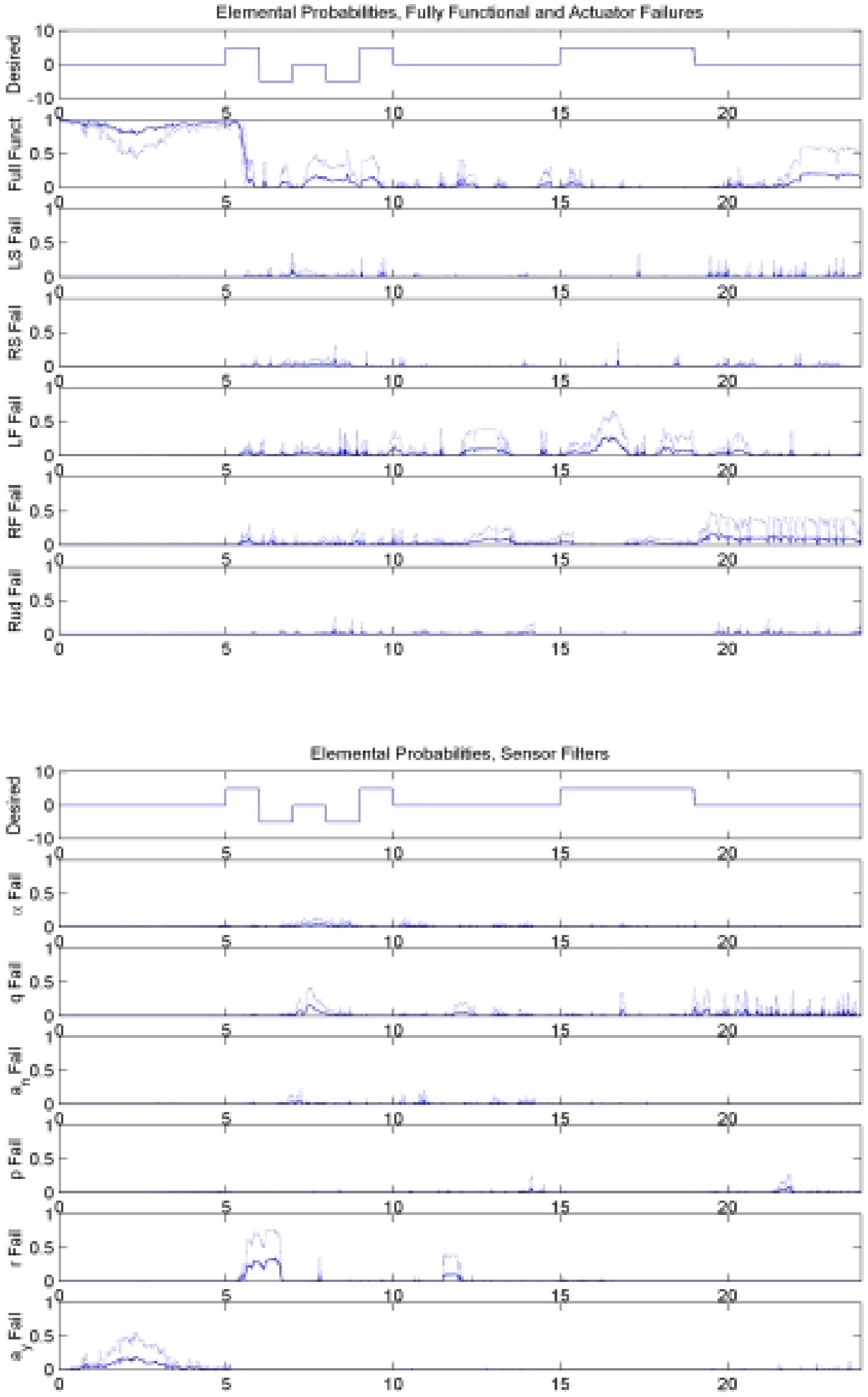


Figure 19a: Elemental Probabilities, Left Flaperon Failure, ε = 50%

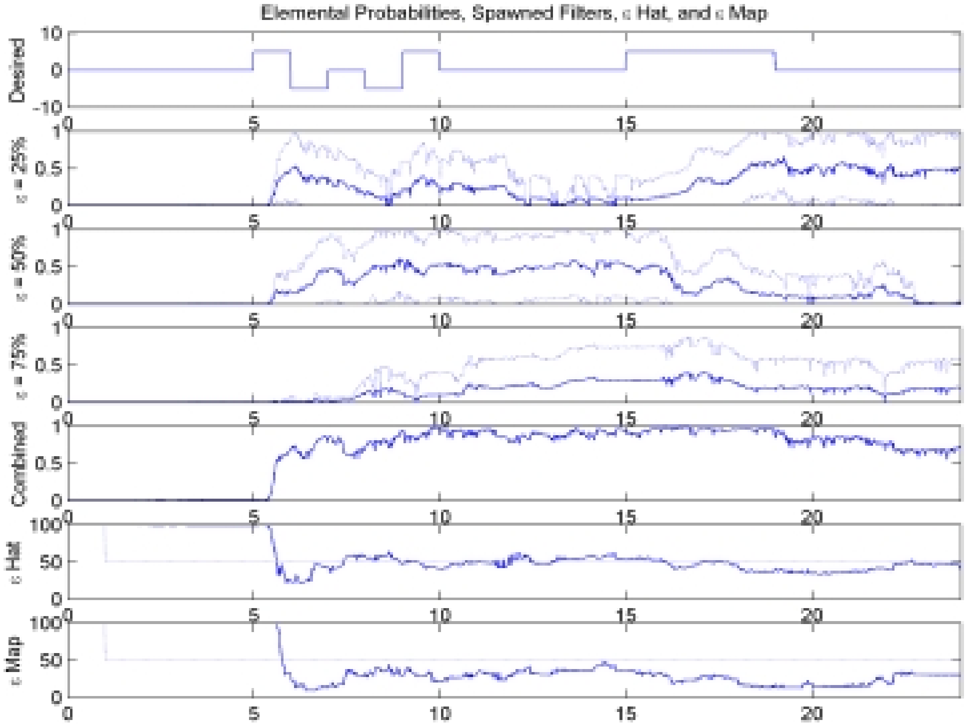


Figure 19b: Elemental Probabilities, Left Flaperon Failure, ε = 50%

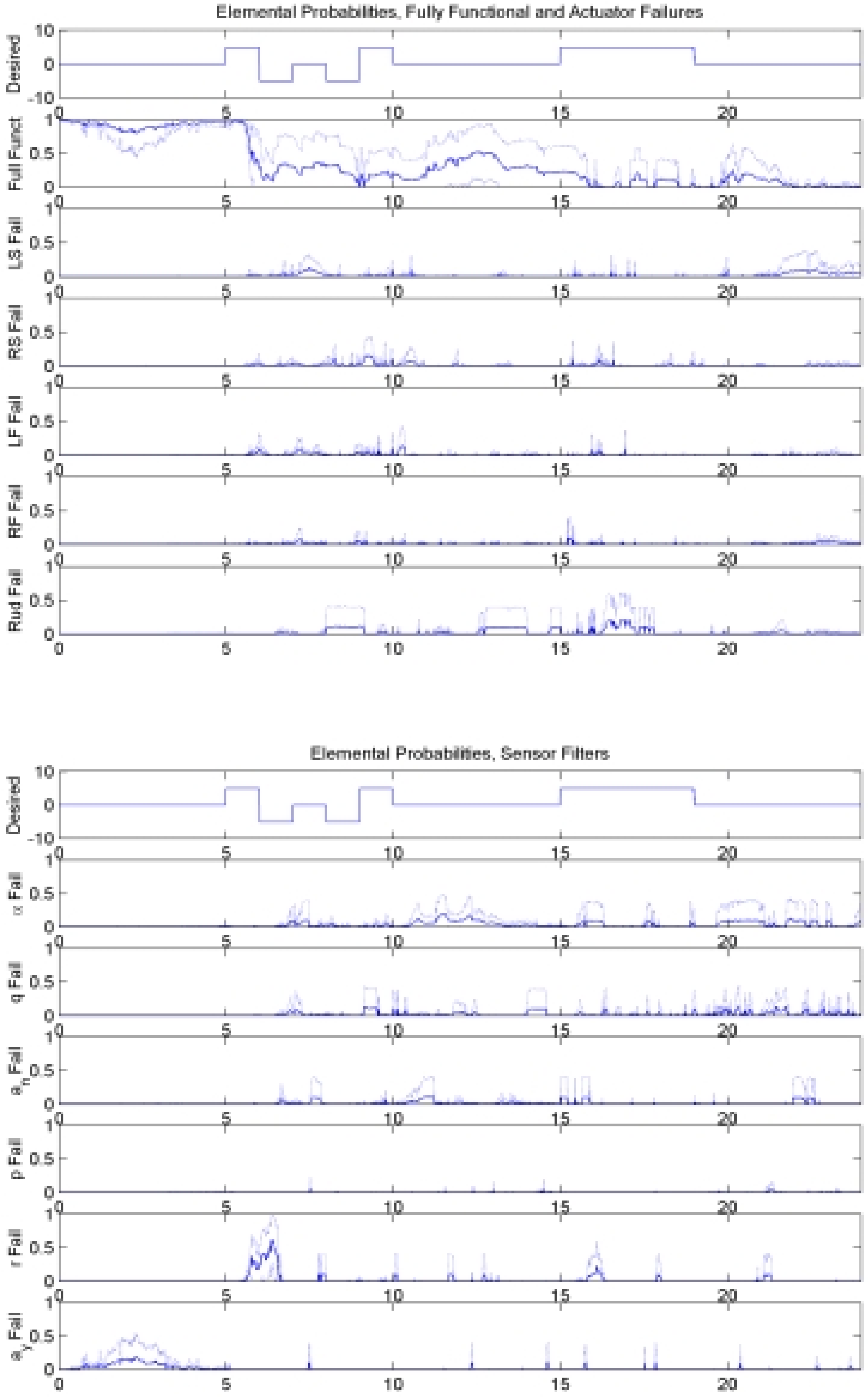


Figure 20a: Elemental Probabilities, Left Flaperon Failure, ε = 75%

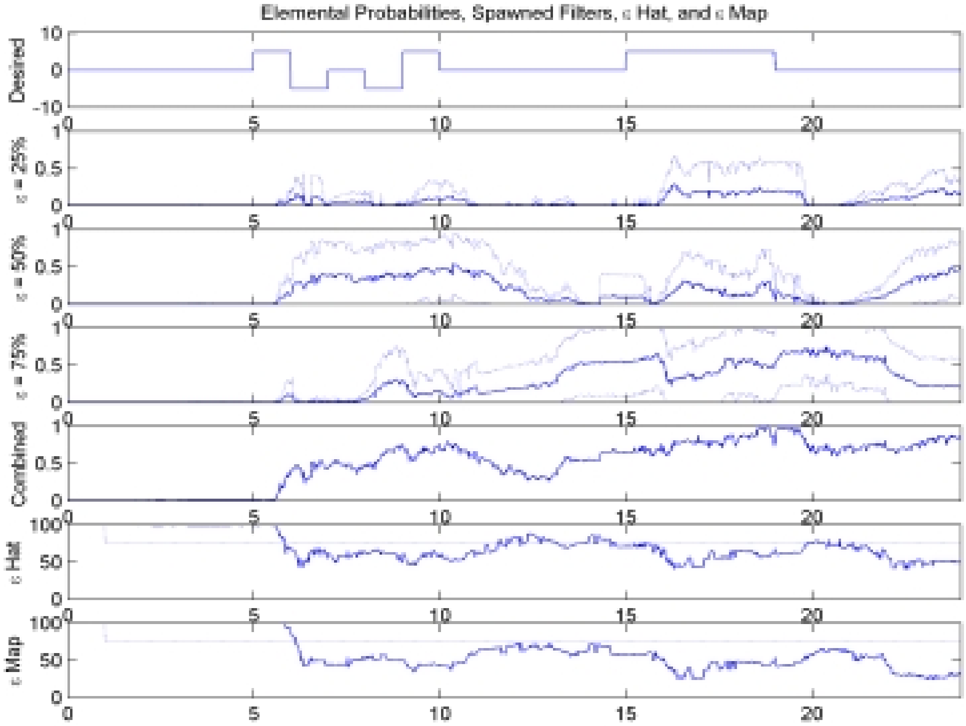


Figure 20b: Elemental Probabilities, Left Flaperon Failure, ε = 75%

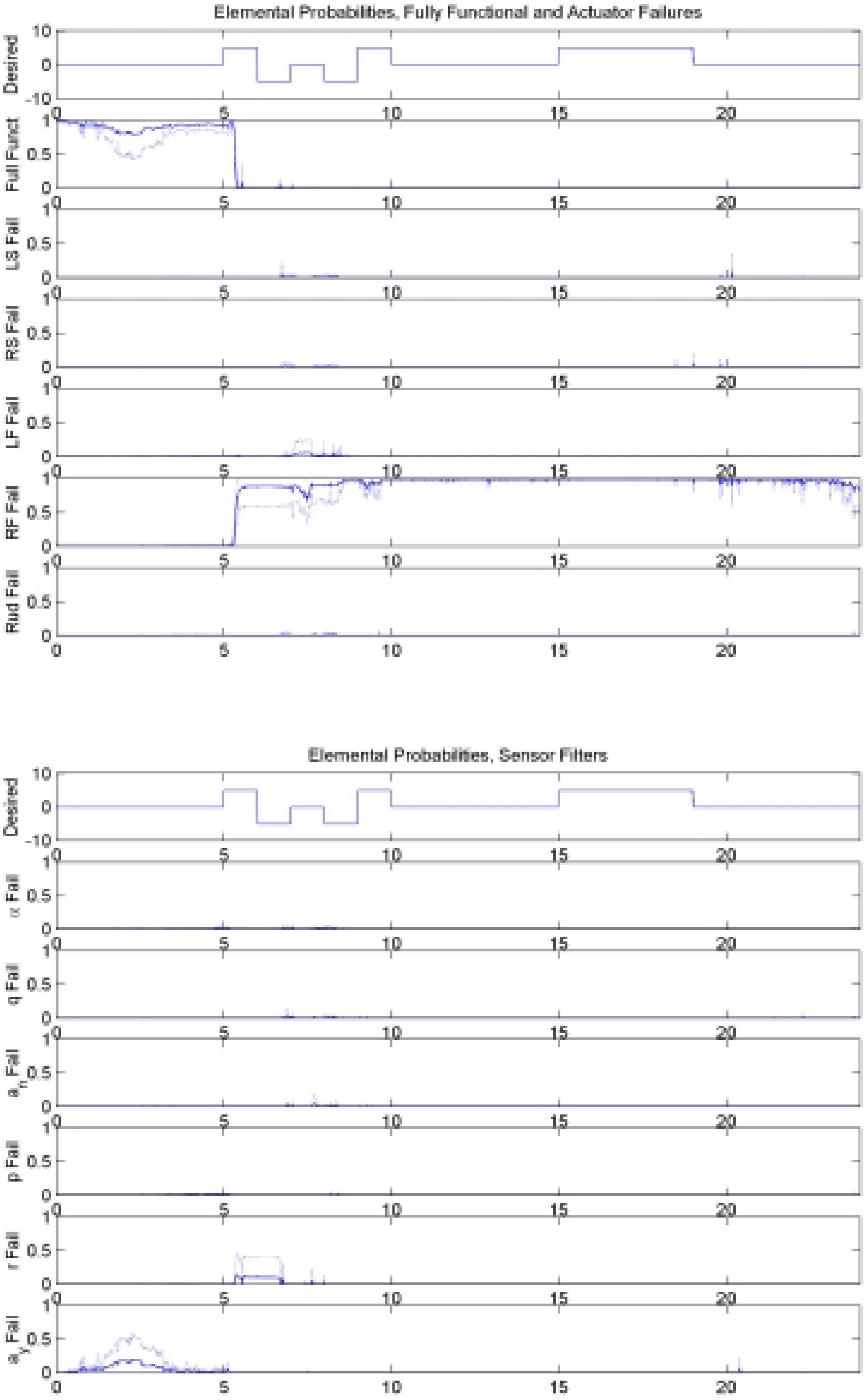


Figure 21a: Elemental Probabilities, Right Flaperon Failure, ε = 0%

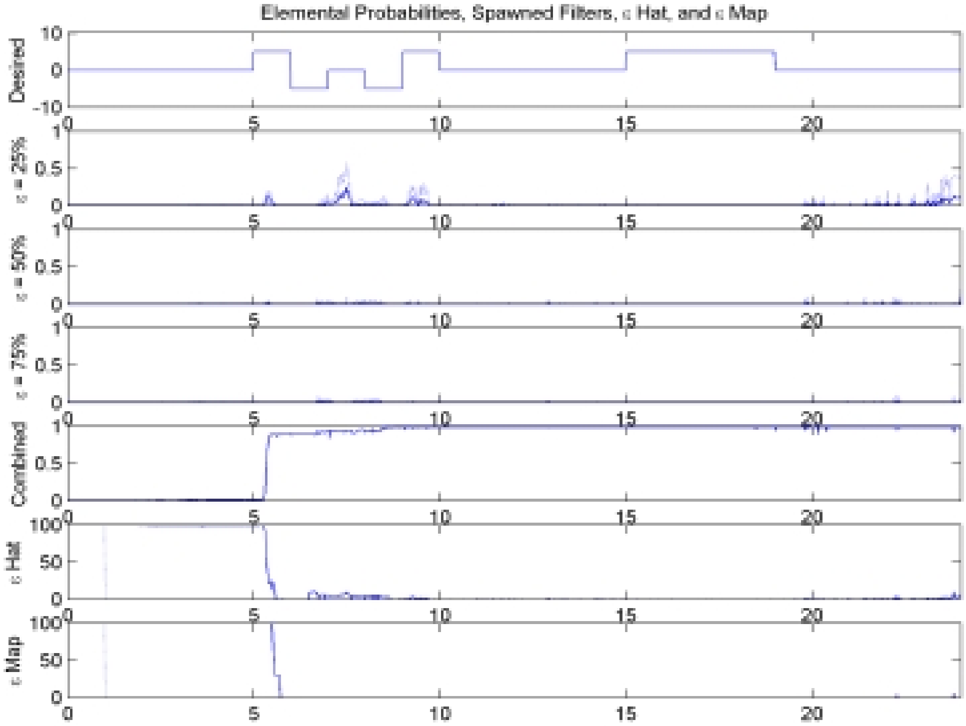


Figure 21b: Elemental Probabilities, Right Flaperon Failure, ε = 0%

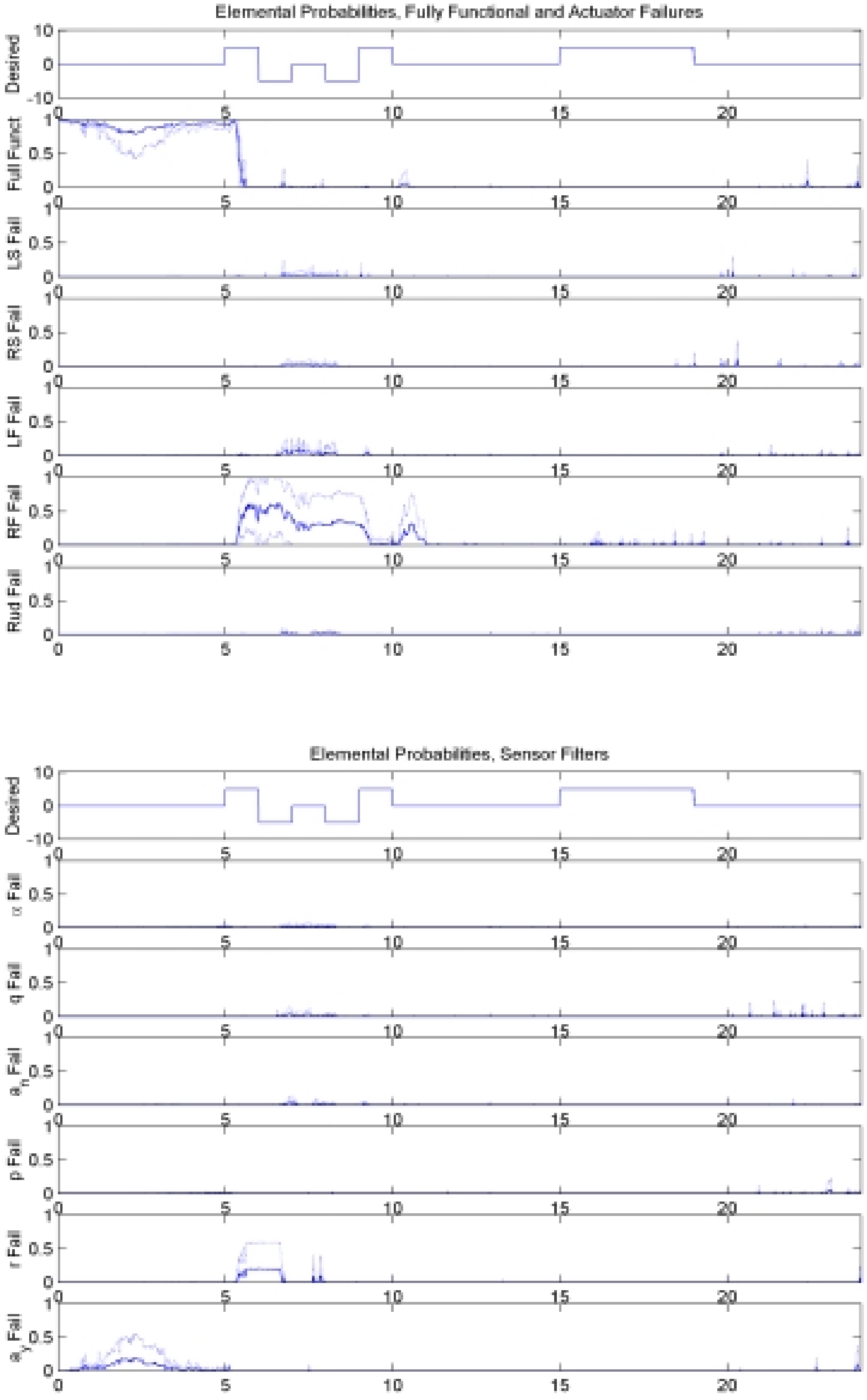


Figure 22a: Elemental Probabilities, Right Flaperon Failure, ε = 25%

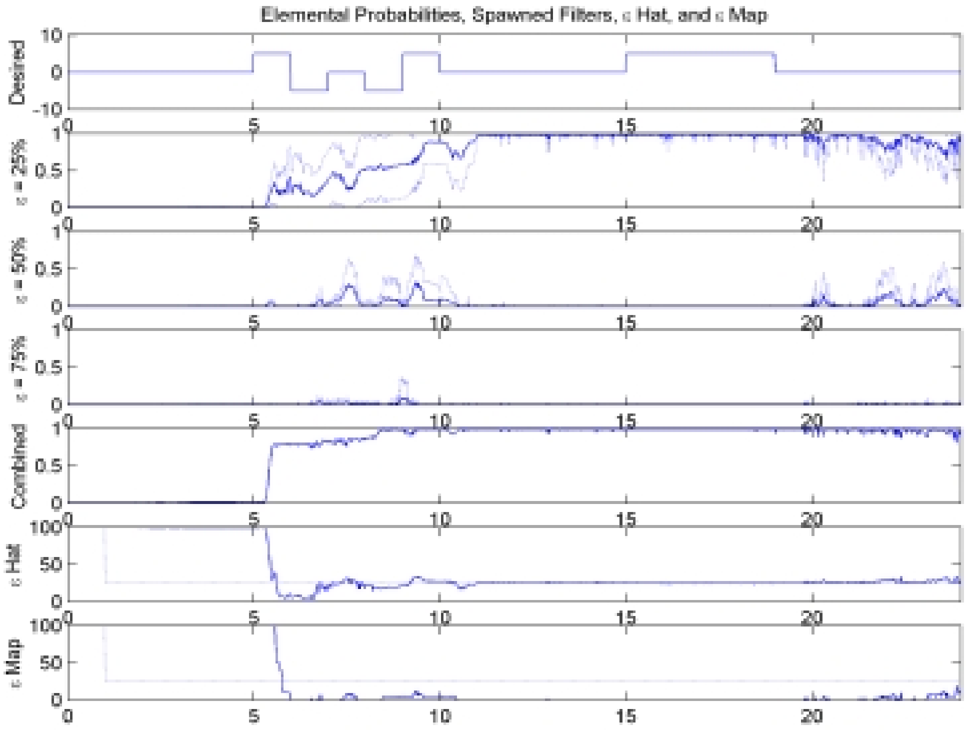


Figure 22b: Elemental Probabilities, Right Flaperon Failure, ε = 25%

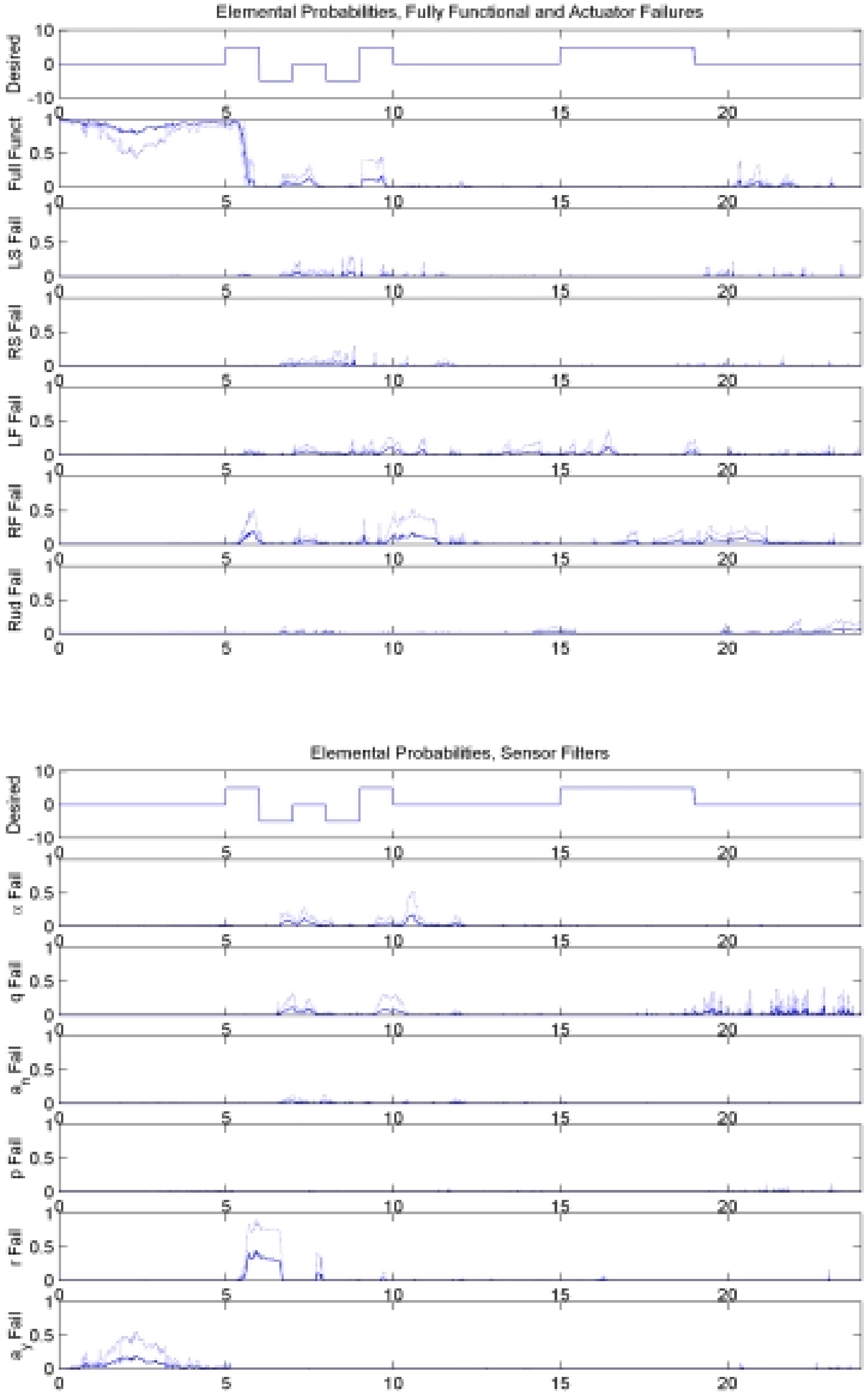


Figure 23a: Elemental Probabilities, Right Flaperon Failure, ε = 50%

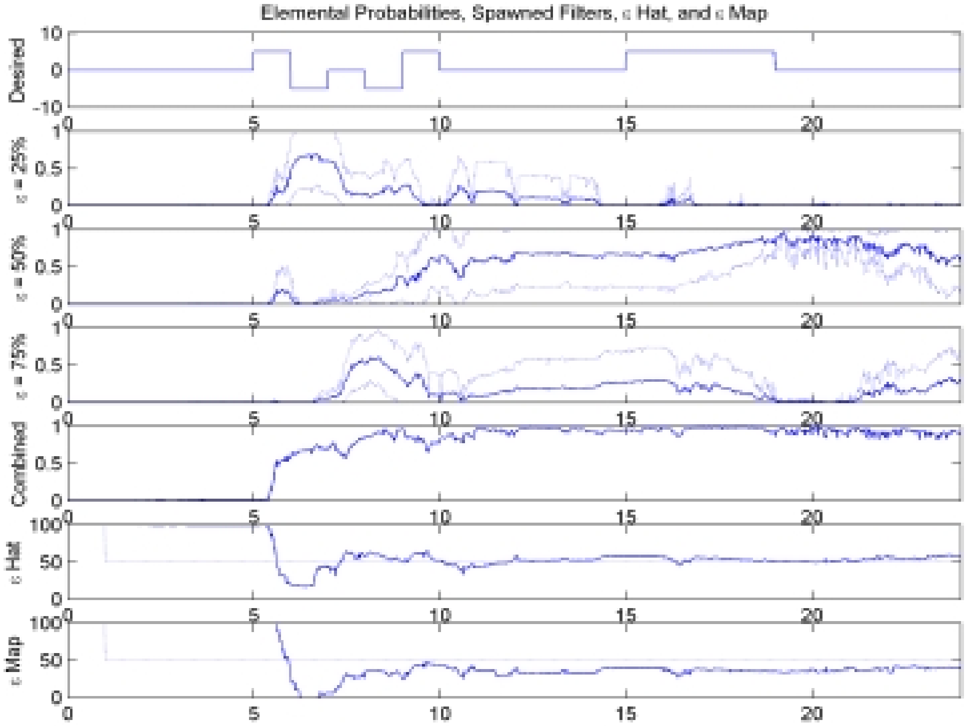


Figure 23b: Elemental Probabilities, Right Flaperon Failure, ε = 50%

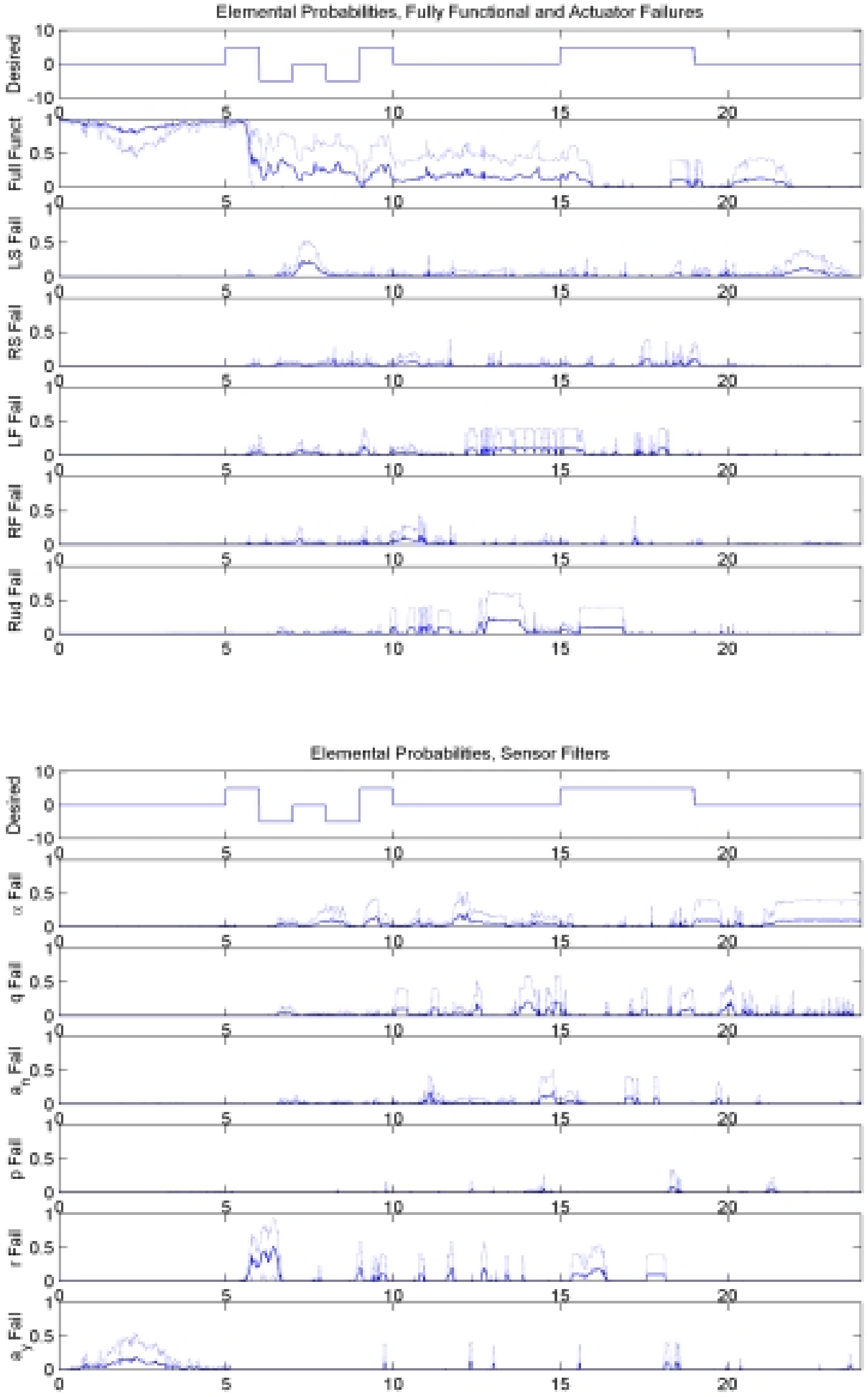


Figure 24a: Elemental Probabilities, Right Flaperon Failure, ε = 75%

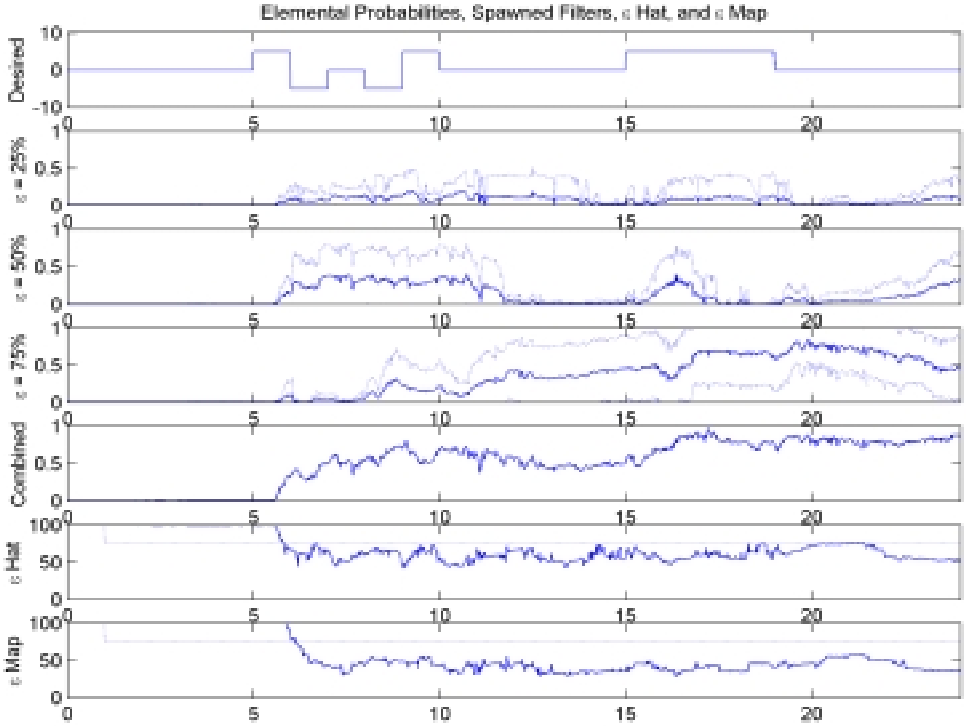


Figure 24b: Elemental Probabilities, Right Flaperon Failure, ε = 75%

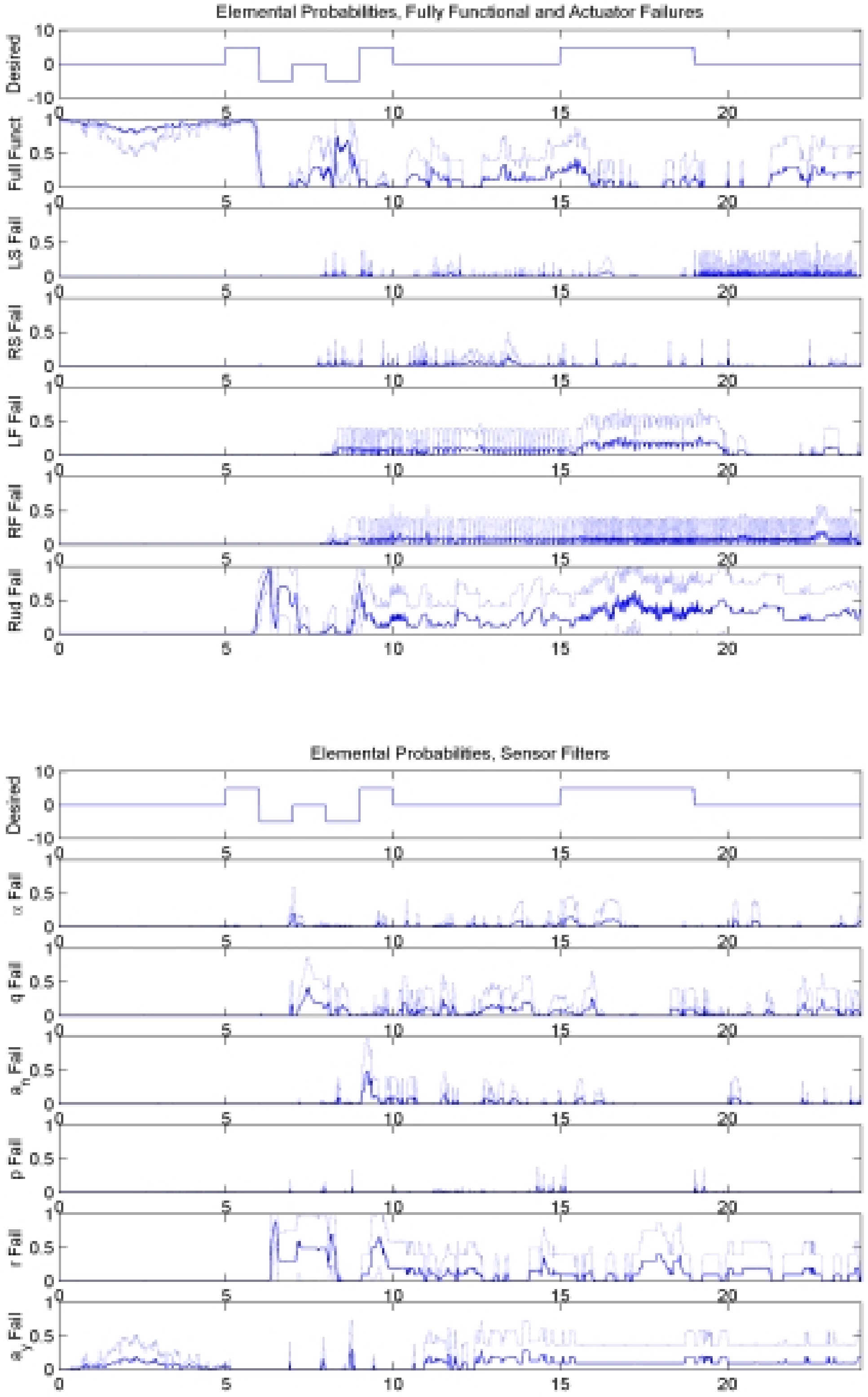


Figure 25a: Elemental Probabilities, Rudder Failure, ε = 0%

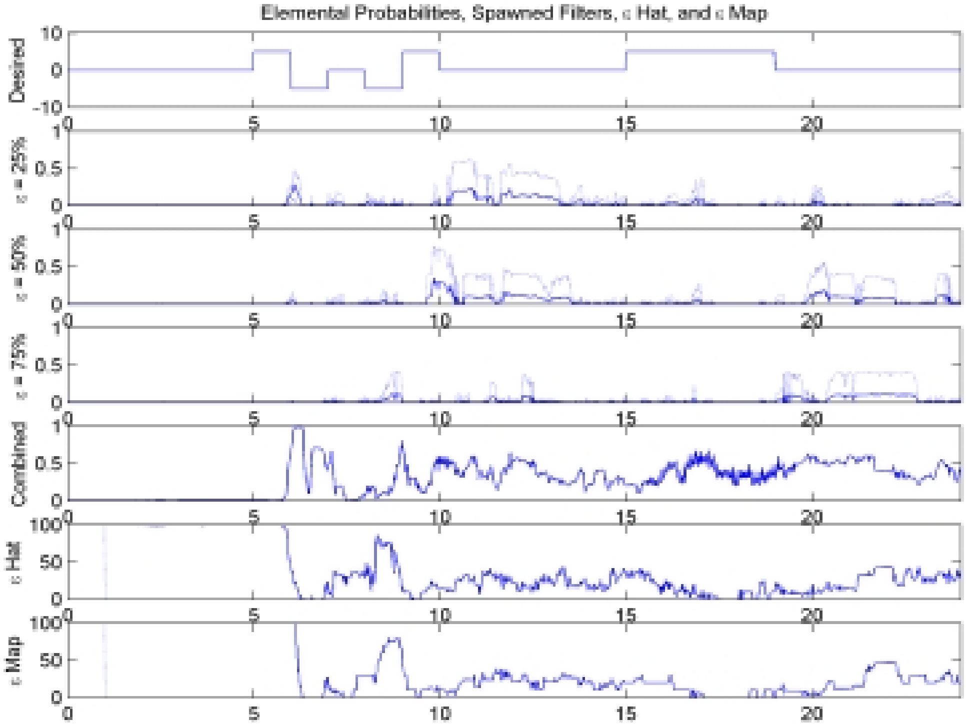


Figure 25b: Elemental Probabilities, Rudder Failure, ε = 0%

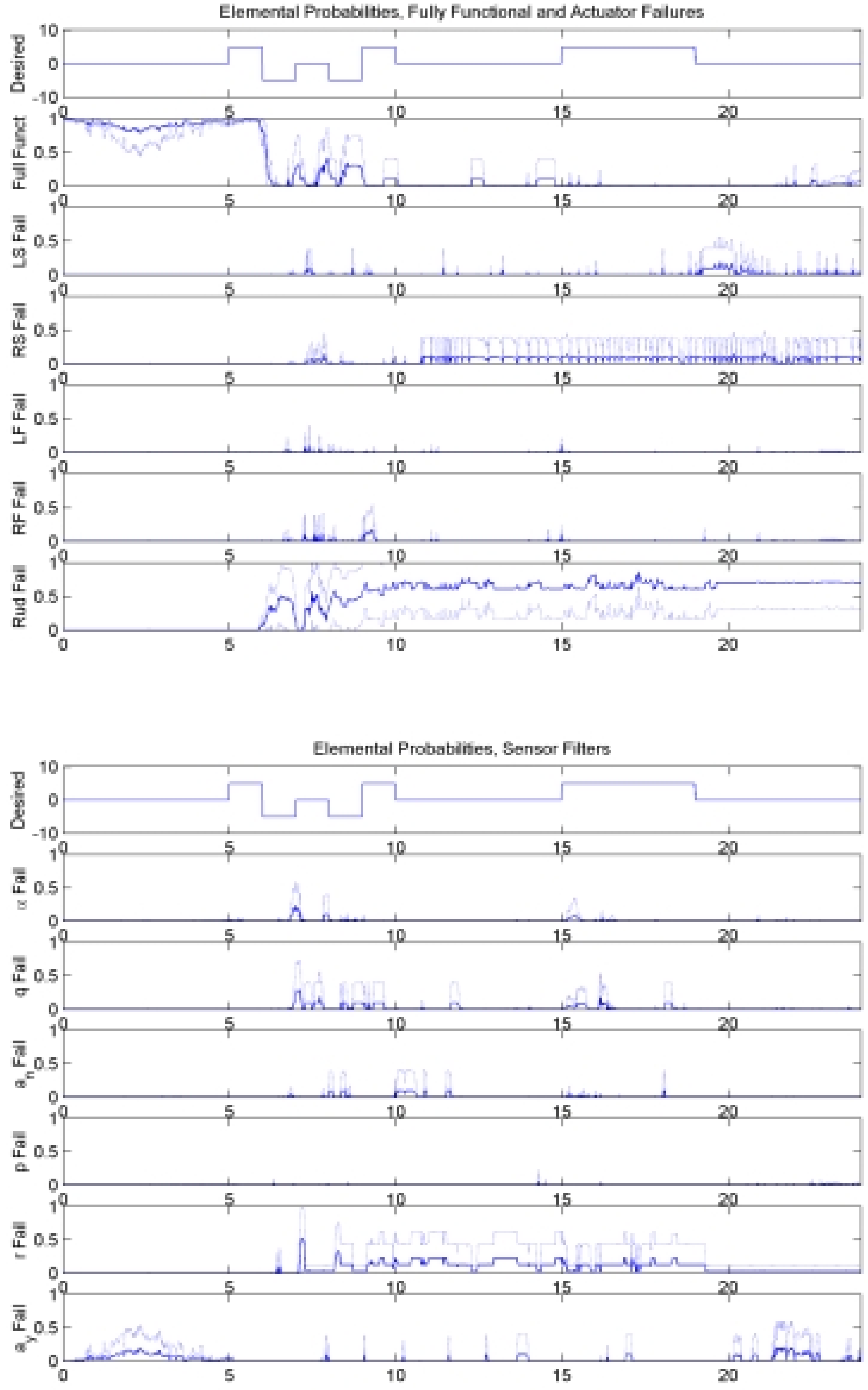


Figure 26a: Elemental Probabilities, Rudder Failure, ε = 25%

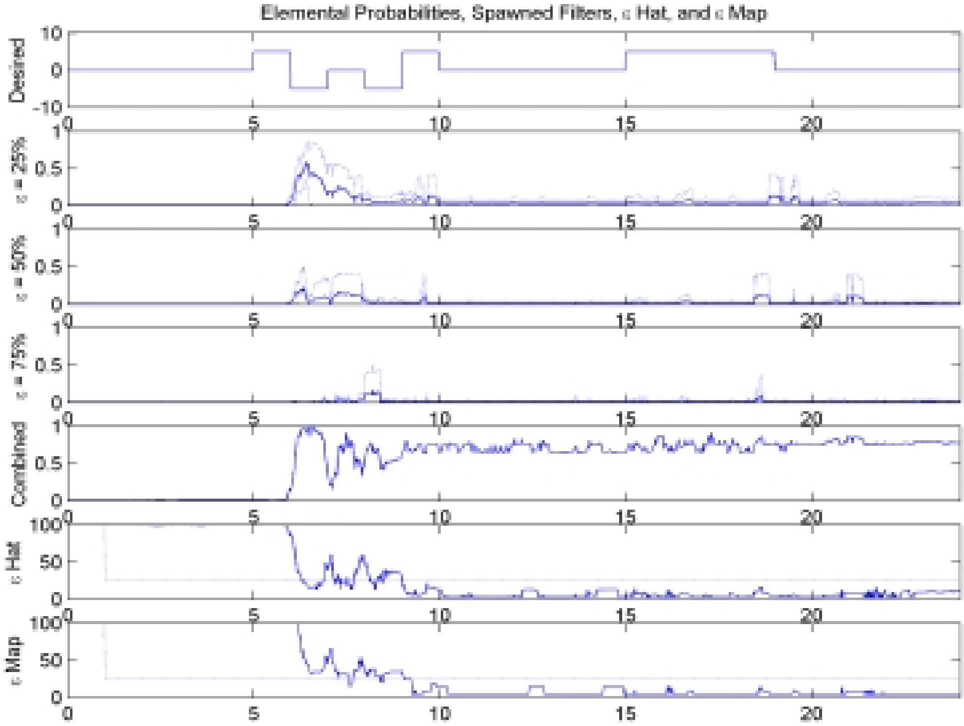
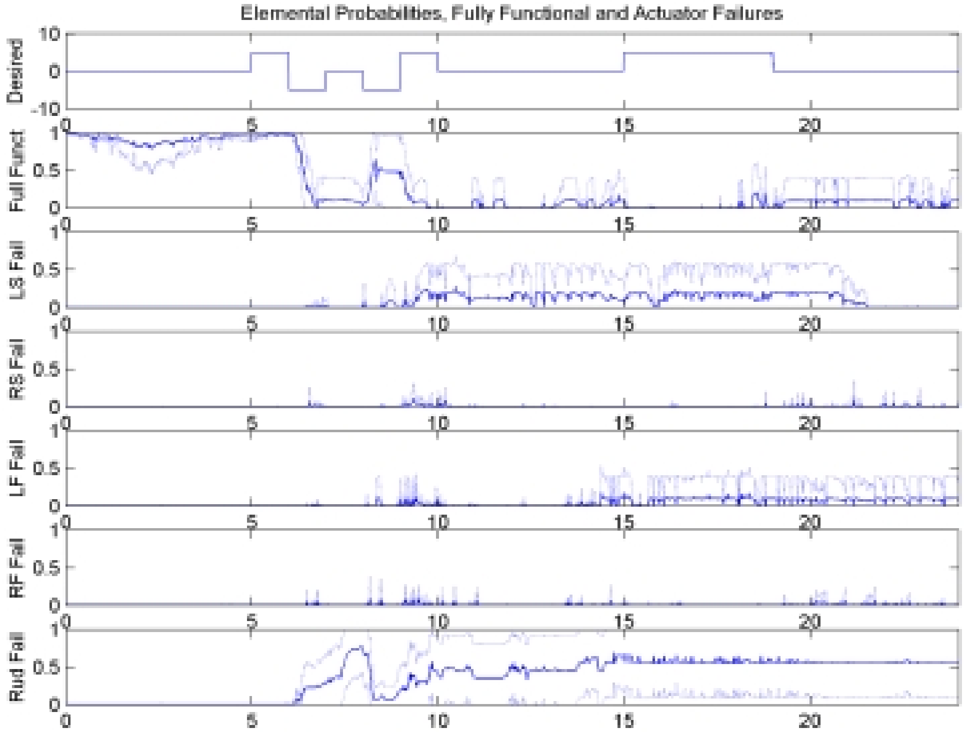


Figure 26b: Elemental Probabilities, Rudder Failure, ε = 25%


Elemental Probabilities, Fully Functional and Actuator Failures

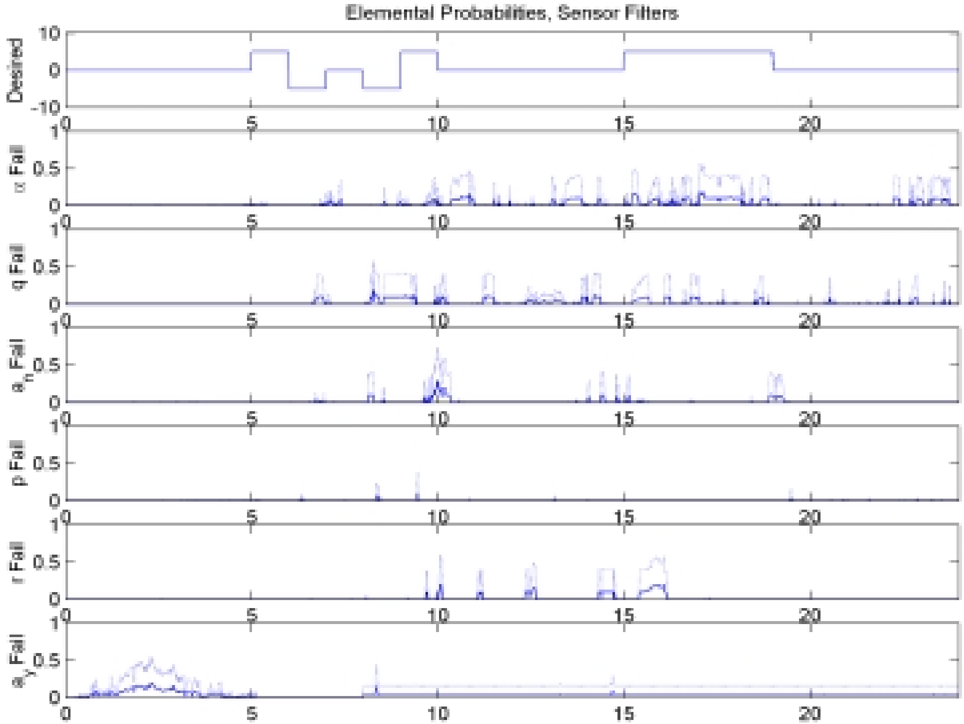

Elemental Probabilities, Sensor Filters

Figure 27a: Elemental Probabilities, Rudder Failure, ε = 50%

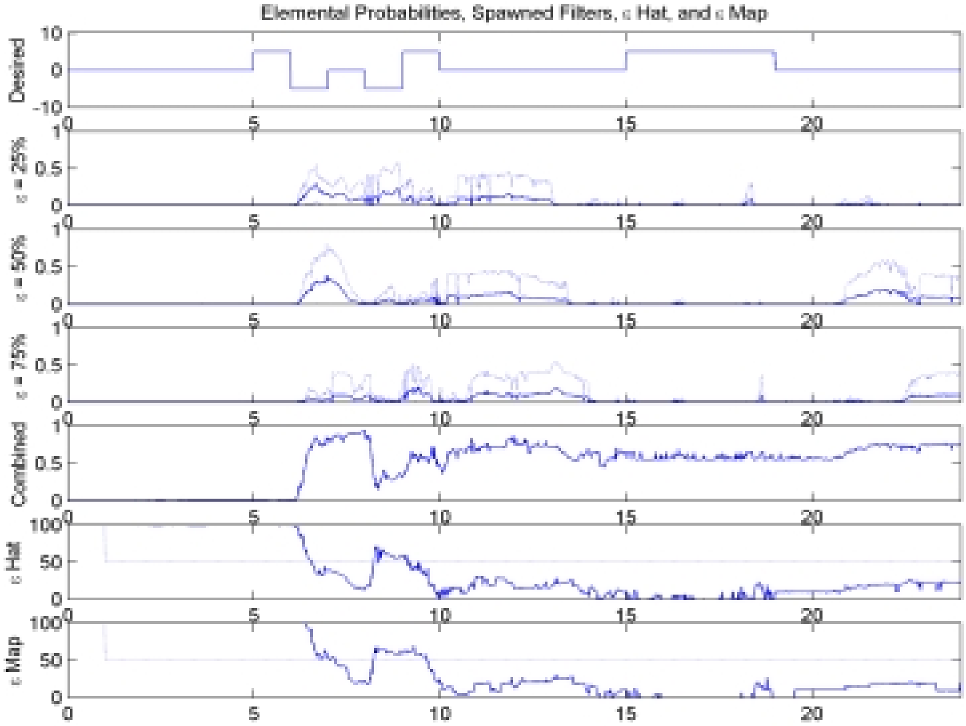
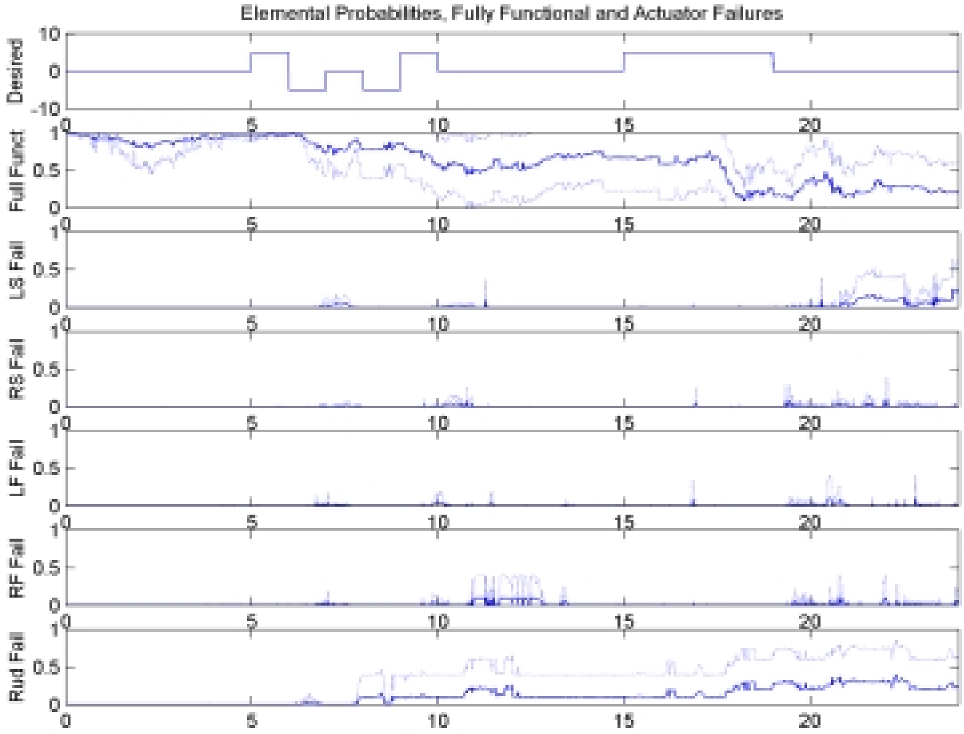


Figure 27b: Elemental Probabilities, Rudder Failure, ε = 50%



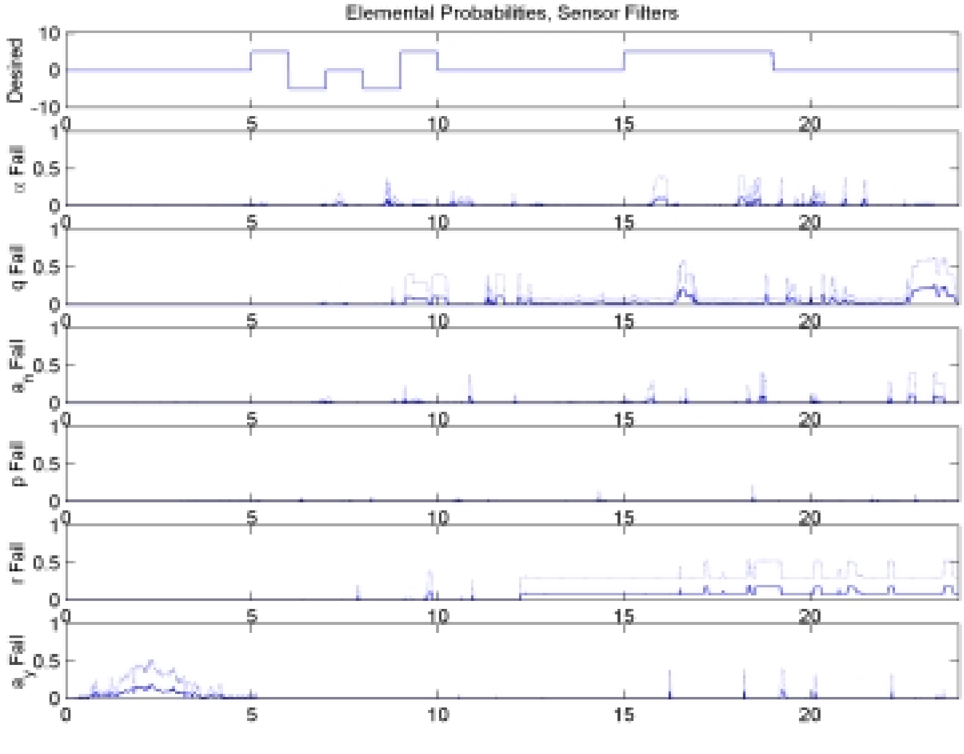


Figure 28a: Elemental Probabilities, Rudder Failure, ε = 75%

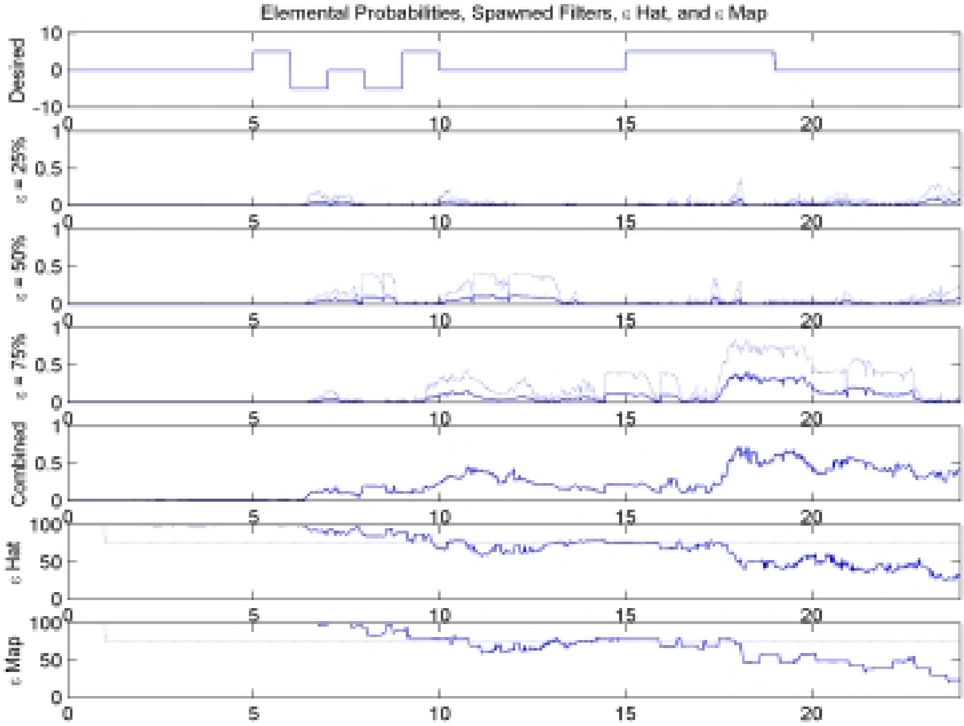
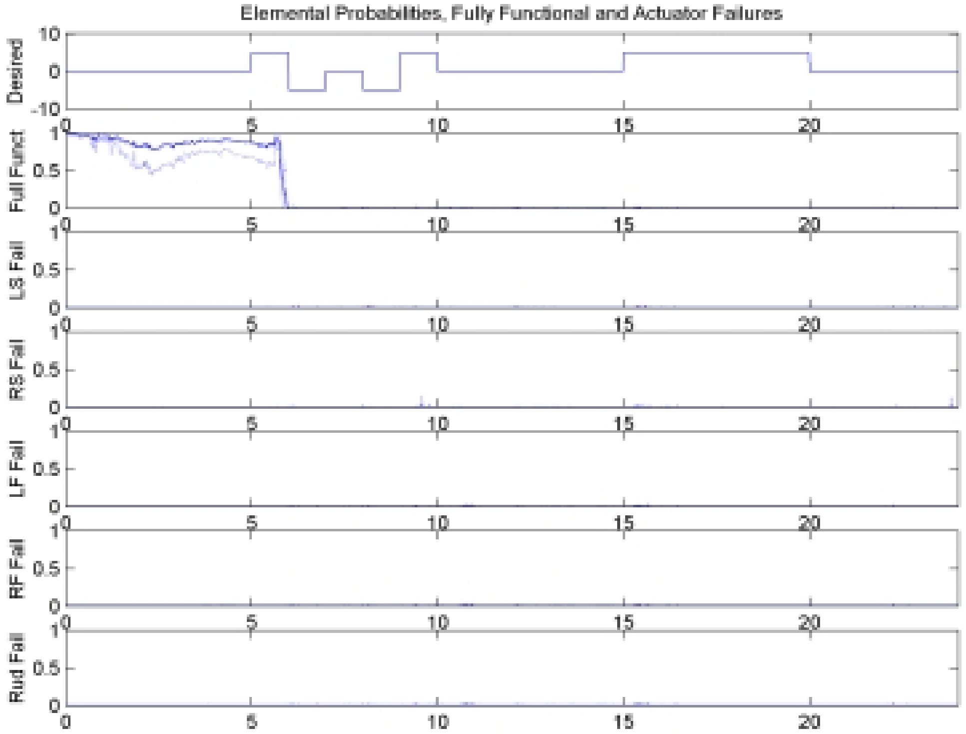


Figure 28b: Elemental Probabilities, Rudder Failure, ε = 75%



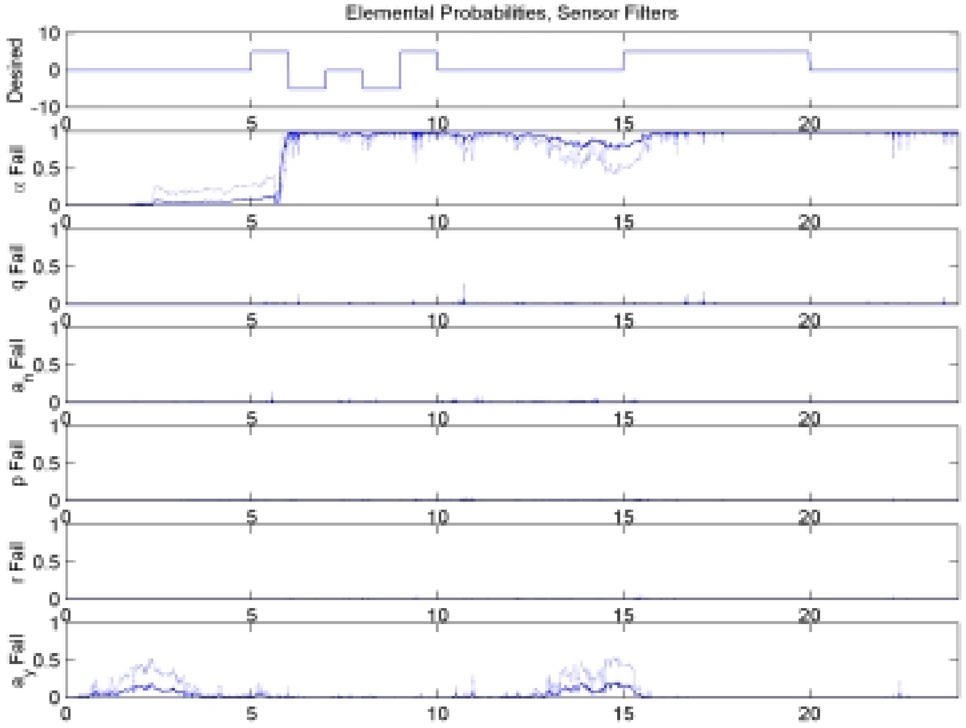


Figure 29: Elemental Probabilities, Angle of Attack Sensor Failure

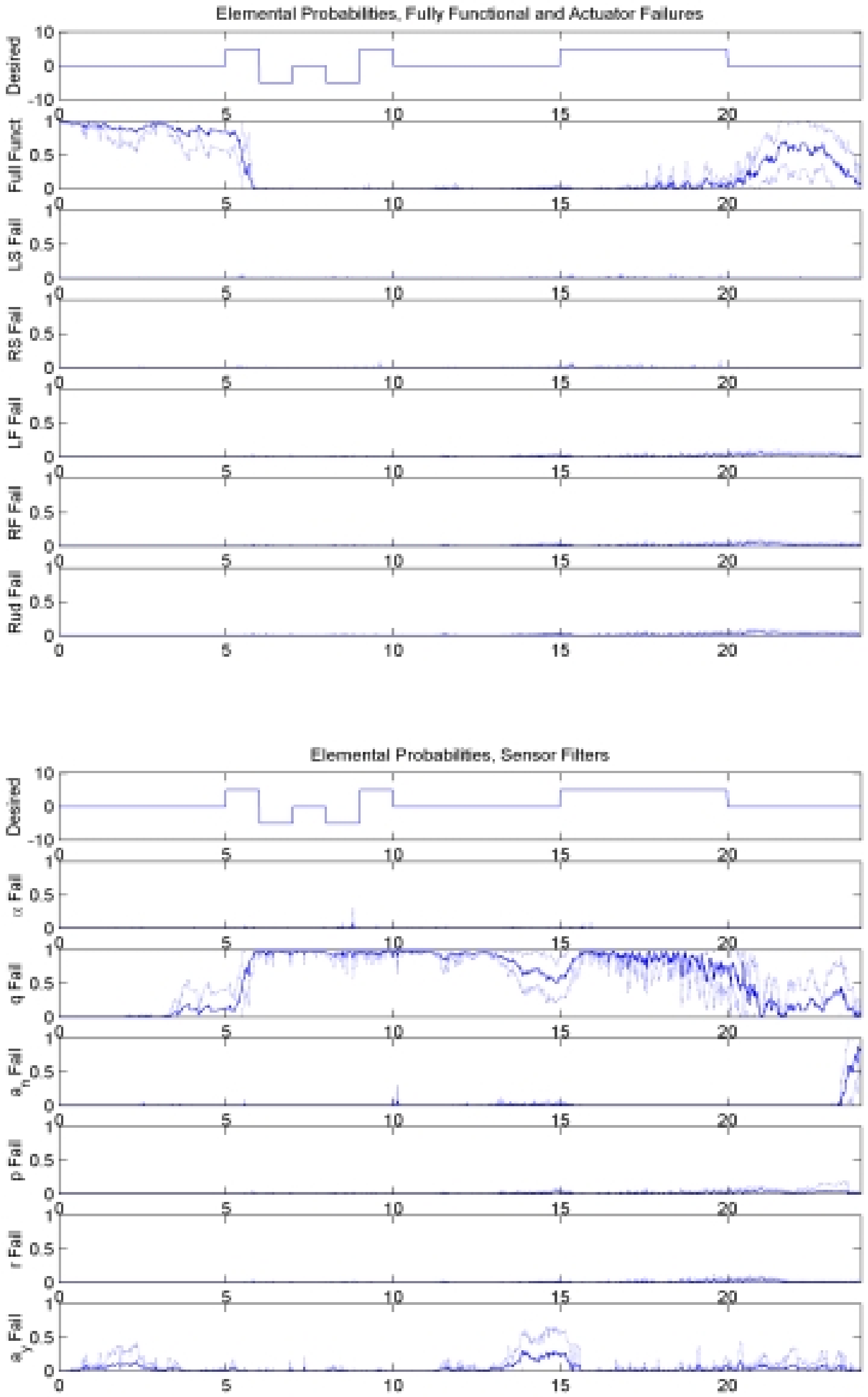
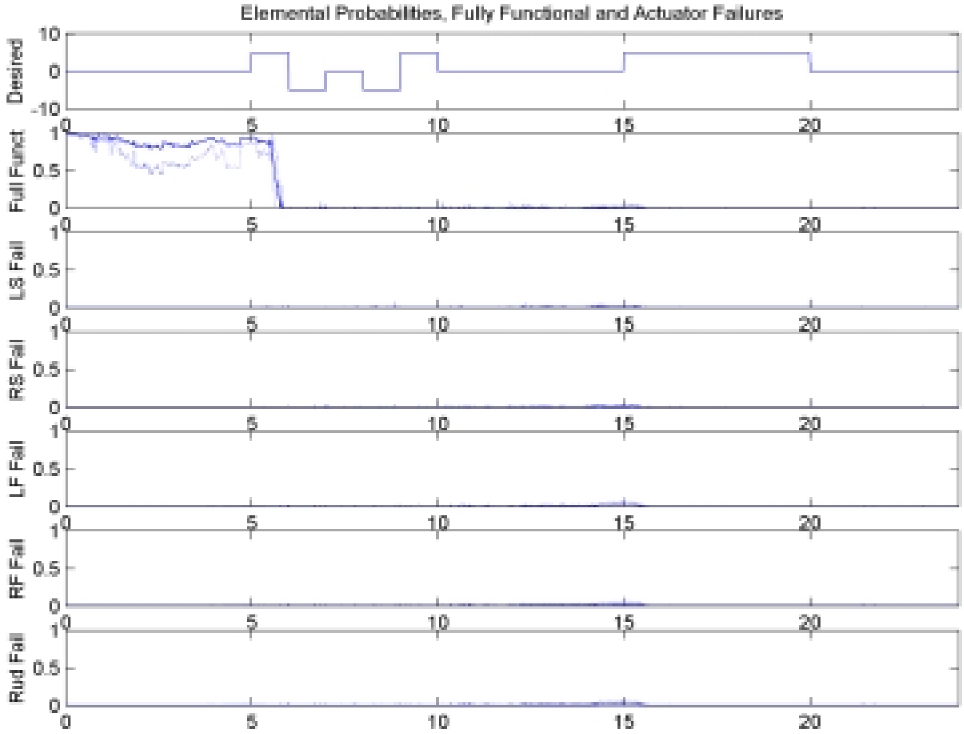


Figure 30: Elemental Probabilities, Pitch Rate Sensor Failure



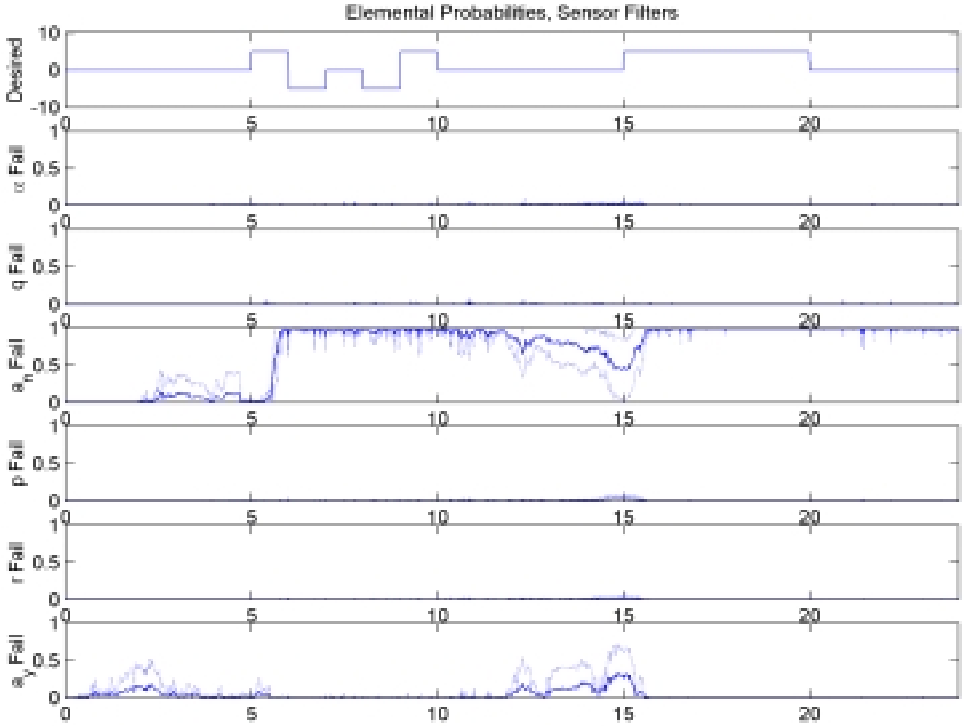


Figure 31: Elemental Probabilities, Normal Acceleration Sensor Failure

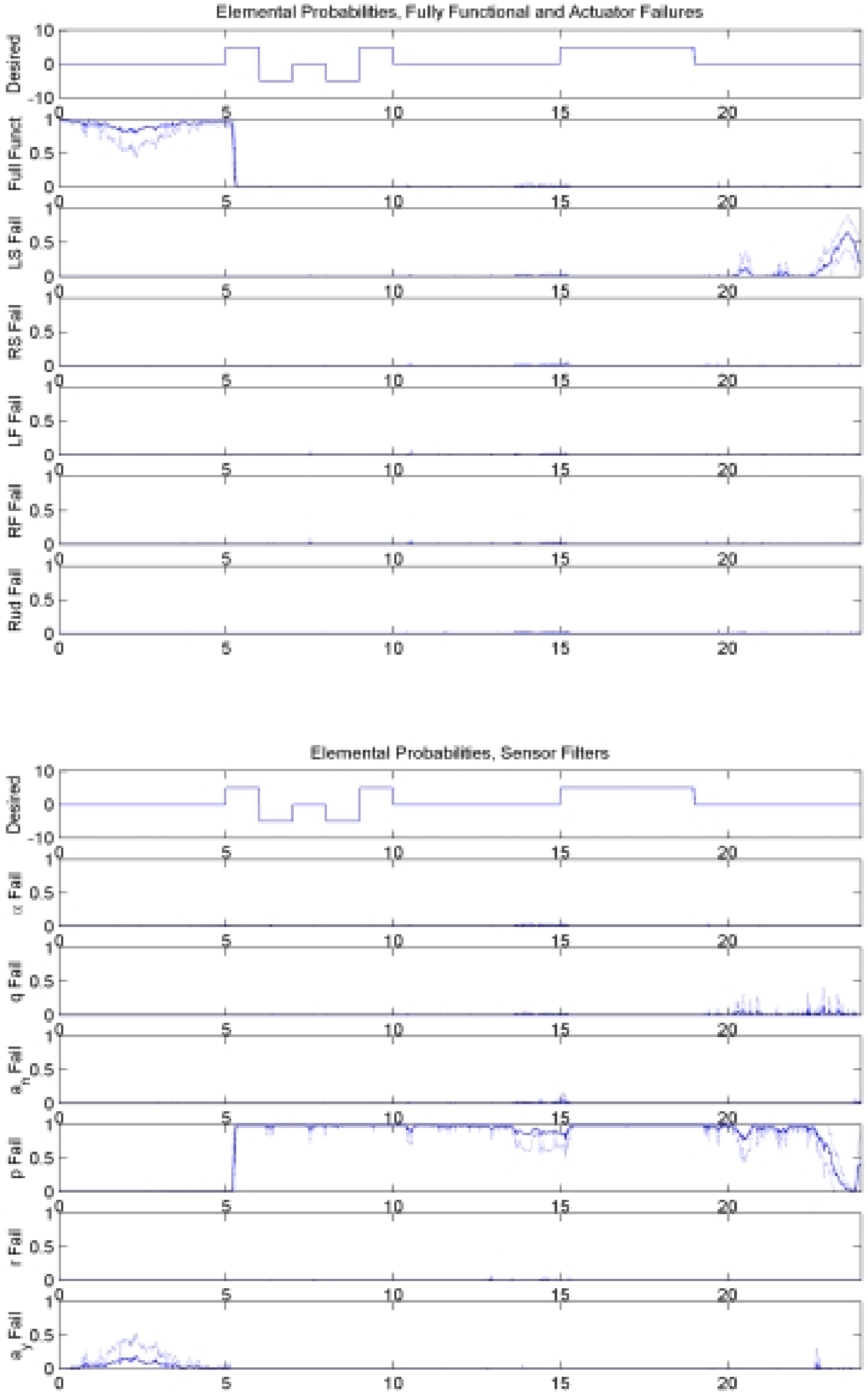


Figure 32: Elemental Probabilities, Roll Rate Sensor Failure

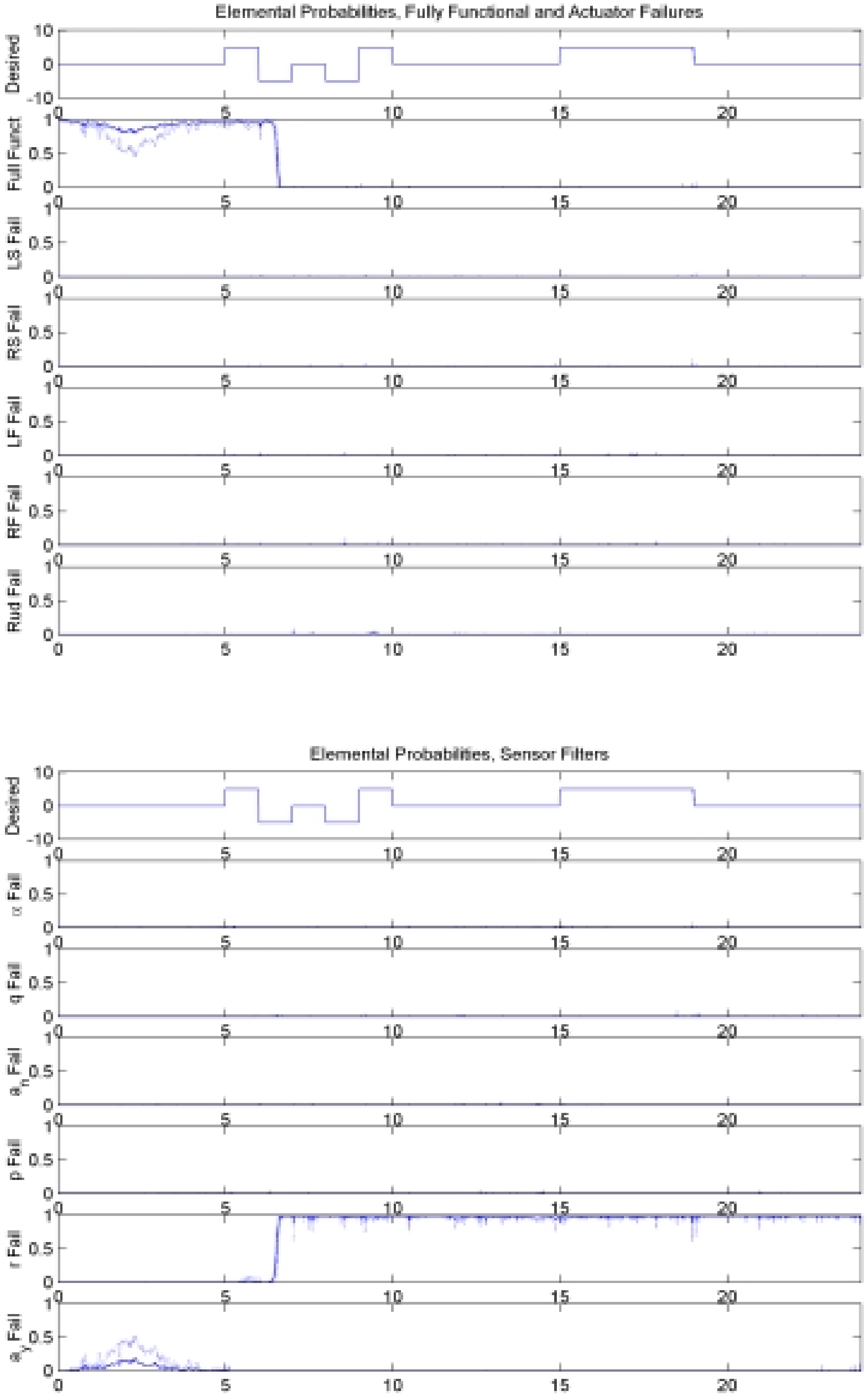


Figure 33: Elemental Probabilities, Yaw Rate Sensor Failure

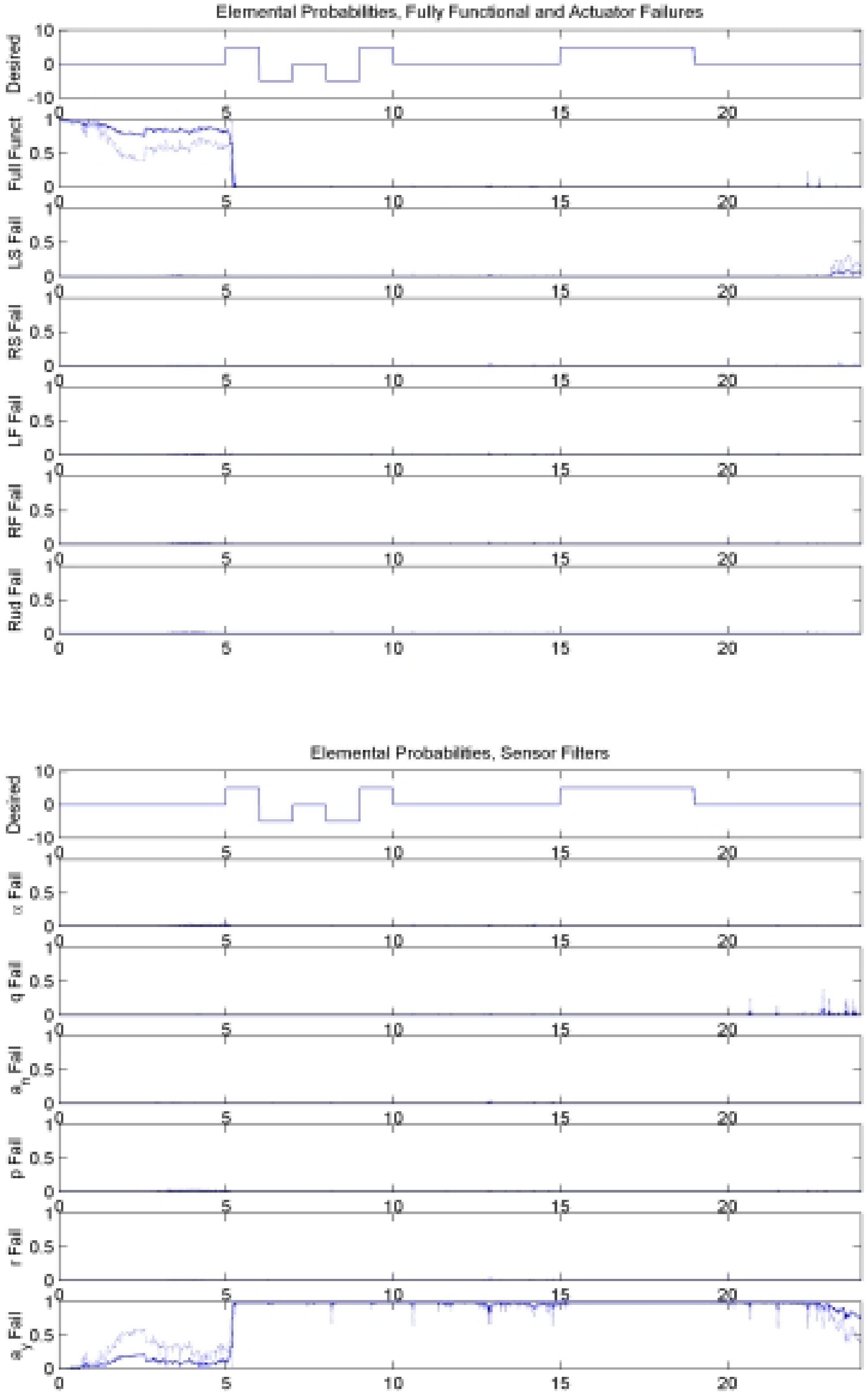


Figure 34: Elemental Probabilities, Lateral Acceleration Sensor Failure

5.3.2 Fully Functional Performance with Dither

Figures 5 and 6 present the MMAE/FS detection performance on a fully functional aircraft, using the purposeful dither signal discussed in Section 2.4.3. In this research, the purposeful dither signal is inserted as a pilot command through the cockpit control stick and rudder pedals. The entire dither signal (pitch, roll, and yaw) is turned off when the pilot inputs a purposely commanded maneuver. Thus, as seen in the 'Desired' plot of each figure, there is full dither over 0-5 seconds, 10-15 seconds, and 20-24 seconds, but there is no dither over 5-10 seconds and 15-20 seconds. The 'Desired' pilot commanded maneuvers (the doublets and the long duration maneuver) are accomplished in the plane under test (pitch maneuvers for the pitch test and roll maneuvers for the roll test). It can be seen from both plots that the elemental probabilities initially indicate there are no failures in the aircraft, although the fully functional aircraft elemental filter probabilities show a downward trend from the beginning of the simulation as compared to the fully-functional aircraft hypotheses without dither in Figures 7 and 8. The aircraft modes (including unmodeled aircraft dynamics) are being excited by the dither signal, causing small amounts of probability to disperse to the failed sensor and actuator/surface failure hypotheses. From these figures, the elemental probability mostly moves into the sensor failure hypothesis filters in the pitch case and into the rudder failure channel in the roll case. Once the maneuver is initiated, the dither signal is removed and from these figures it can be seen that the elemental probability within the fully-functional hypothesis does not substantially change during the maneuver. After the pilot commanded maneuvers (pitch doublets), the elemental probability of the fully-functional aircraft hypothesis is reduced and the elemental probability within the failed sensor hypotheses increases. This is a result of the aircraft dynamics following the maneuver (such as long and short period aircraft dynamics) and the dither signal occurring at the same time, causing additional higher order aircraft dynamics not modeled in the failure hypotheses of the linear filters, resulting in false sensor failure detections. Figure 6 shows similar results but the elemental probability moves to the rudder failure hypothesis. This causes further problems because false rudder failure declarations brought on by the application of the dither signal bring the CR algorithm online. At this point, the CR is causing the aircraft to maneuver even more during the period between pilot commanded maneuvers, worsening the detection performance of the MMAE/FS. These results show the need for time between the end of the pilot-commanded maneuver and the time to employ the purposeful dither signal. The purposeful dither does improve the detection of actual failures in a nonmaneuvering flight. The purposeful dither signal is not used in the rest of the simulations for this research effort.

5.3.3 Fully Functional Performance without Dither

Figures 7 and 8 show the performance of the MMAE/FS in a fully functional aircraft undergoing the pitch and roll maneuvers, respectively. This is the same as portrayed in Figures 5 and 6, but without dither at any time. These elemental probability plots also demonstrate detection problems at the end of the long duration maneuver when the aircraft is placed in a different attitude than straight-and-level. When the aircraft is in a different attitude than straight-and-level, the current state (as detected by the sensors) does not match any of the linear models and appears to the MMAE/FS as a sensor failure, as can be seen at the end of the simulation in Figure 7. The sensors that appear to be the most sensitive to the change in attitude are the failure hypotheses based on pitch rate and the yaw rate sensors, as can be seen in Figure 10 for example. The elemental linear filters are based on the parameters for straight-and-level flight. This shows the need for model scheduling based not only on the dynamic pressure but also on aircraft attitude and nominal values. This same pattern can be seen in several of the other plots, particularly in the partial actuator failure plots. The spawned filters used in the fully functional aircraft are from the partial left stabilator hypothesis (Bank 1 as mentioned earlier in Section 4.4.1). The bank number does not change through the fully functional aircraft simulations.

By comparing Figures 5 and 6 to Figures 7 and 8, the detection problem at the end of the simulation can be seen to be considerably worse without the purposeful dither signal. The dither signal excites the aircraft modes and makes them more visible to the MMAE/FS. After the long duration maneuver, the dither is applied to the case presented in Figures 5 and 6, whereas Figures 7 and 8 do not have the dither signal applied.

In comparing Figure 7 with Figure 8, it can be seen that false alarms on sensors is more severe with the pitch maneuvers than with the roll maneuvers. False partial actuator/surface failure alarms are worse with the roll maneuvers than with the pitch maneuvers.

5.3.4 Detection Performance of Fully Failed Actuators/Surfaces other than the Rudder

Figures 9, 13, 17, and 21 show the performance of the MMAE/FS used to detect full actuator failures of the stabilators and flaperons. In all cases, the MMAE/FS detects the failure during the first doublet maneuver and maintains the detection through the rest of the simulation. Figure 17, which portrays the left flaperon actuator failure simulation results, shows the elemental probability moving to the spawned filters during the long duration maneuver, and then the elemental probability returns to the correct failure hypothesis. This is caused by the direction of roll. In one direction, as in this case, the failure identifiability is decreased possibly because the other flaperon is providing a greater rolling force. The one flaperon can produce a greater rolling force because when the roll is begun, the sideslip angle increases, and since the rudder is also being used to achieve a coordinated turn, the aircraft does not roll solely on the longitudinal axis (axis from tail to the nose of the aircraft) but also rotates on the other axes (lateral and vertical axes). This geometry causes a greater rolling force on one flaperon as compared to the other flaperon. The other flaperon case is presented in Figure 21. This figure illustrates the same maneuver, including the same roll direction, but with the failure of the right flaperon actuator. In this case, the elemental probability flows to the correct failure hypothesis and remains there through the entire simulation, including the long duration maneuver. The flow of elemental probability to the spawned filter hypotheses in Figure 17 does not detract substantially from the MMAE/FS performance since the correct bank is used (the bank corresponding to a partial left stabilator actuator failure) and the “error” is primarily declaring ε to be 25% versus 0%. The elemental probability movement to the spawned filter in Figure 17 between 16 and 20 seconds (particularly to the $\varepsilon = 25\%$ filter) caused by the roll maneuver does cause the estimates of effectiveness (both ε_{hat} and ε_{map}) to increase from 0% effectiveness for a few seconds. Once the elemental probability flows back into the fully failed left flaperon hypothesis, both estimates of effectiveness return to the proper values. The same pattern can be seen in Figure 18 and Figure 19. The 75 percent effectiveness case shown in Figure 20 does not show the same pattern, and that can be caused by the lack of failure identifiability with the 75% effectiveness case, as will be mentioned in Section 5.3.5. In all the fully failed cases, the banks chosen by the MMAE/FS were the appropriate banks to use, based on the actual failure.

At the end of the simulation, after the long duration maneuver, the problems caused by the aircraft being at a different attitude than straight-and-level can be seen. Figure 9a shows performance at the end of the simulation similar to that of Figure 7a with respect to the elemental probability moving from the fully failed left stabilator hypothesis (fully functional hypothesis in the case of Figure 7) to the pitch rate and normal acceleration sensor failure hypotheses. This also shows the need to have model scheduling based on aircraft attitude, as well as dynamic pressure.

5.3.5 Detection Performance of Partially Failed Actuators/Surfaces

The other figures among Figures 9-24 show the performance of the MMAE/FS in determining the correct partial actuator/surface failure condition and providing an estimate of actuator/surface effectiveness given partial actuator/surface failures. In the cases in which the partial actuator failure has true effectiveness of $\varepsilon = 25\%$ and 50%, the ε_{hat} provides a better estimate of actuator effectiveness than ε_{map} (recall the discussion of ε_{hat} versus ε_{map} in Section 4.5.5), as can be seen in Figures 10, 11, 14, 15, 18, 19, 22, and 23. The ε_{map} was designed to improve performance of the MMAE/FS with acceptably "subliminal" dithers but no purposeful, large maneuvers. Although it yielded better performance under those conditions, apparently it does *not* for maneuvering flight.

As mentioned before, after the long duration maneuver, the aircraft is in a different attitude than straight-and-level. This accounts for the difficulty of the MMAE/FS to maintain a good estimate of effectiveness after the long duration maneuver, as can also be seen from the aforementioned figures. The MMAE/FS has difficulty detecting and estimating the 75% effective actuator/surface cases, as can be seen in Figures 12, 16, 20, and 24. Since the aircraft is maneuvering, the failure identifiability is apparently reduced, as compared to the cases where enough failure identifiability exists to determine the 25% and 50% effectiveness cases as well as the fully failed case (0% effectiveness). This is not a major issue since a 75% effectiveness case (i.e., 25% failure) still leaves enough control authority to complete the desired maneuvers without CR, as will be shown later in this chapter.

Figures 12 and 16 show the 75% effectiveness partial stabilator failure cases. From these plots, the change in attitude after the long duration pitch maneuver causes false pitch rate and normal acceleration sensor alarms. As mentioned before, the aircraft being in a different attitude than straight-and-level causes this. The ε_{hat} estimate shows approximately 100% effectiveness as mentioned earlier. Figures 20 and 24 show the 75% effectiveness partial flaperon failure cases. In these cases, it can be seen that, after the long duration roll maneuver, there are no false sensor failure alarms and the ε_{hat} estimate is approximately correct throughout the simulation. This shows that the elemental filters within the MMAE/FS are more sensitive to changes in pitch angle than in roll angle.

Figures 9-12 and Figures 14-16 show the poor detection performance of the MMAE/FS after the long duration pitch maneuver when the aircraft is in a different attitude than straight-and-level flight. This is similar to the MMAE/FS performance with no failures shown in Figure 7. This shows the requirement of model scheduling as a function of both aircraft attitude and dynamic pressure. It is also interesting to note that Figures 18-20 and Figures 22-24 do not show the sensor failure false alarms. Based on this, the pitch angle appears to have a more dramatic impact on the linear models within the MMAE than roll angle.

This research effort also includes the “combined” elemental probability plot, which is the combination of the probability assigned to the fully failed actuator/surface hypothesis with the elemental probabilities of the spawned filters. This is computed to improve the detection performance of the MMAE/FS during partial actuator/surface failures. A good example of the benefits of the “combined” channel is shown in Figure 22, the right flaperon partial failure ($\varepsilon = 25\%$). In this case, if the “combined” channel probability is not used to declare the partial actuator/surface failure, then the failure declaration would take place at approximately 11 seconds into the simulation. With the “combined” channel used, the partial failure declaration is made at approximately 6 seconds. Since the CR is using the declaration from the MMAE/FS to apply the proper redistributive control to the aircraft, the CR must be activated as early as possible in the maneuver. This allows the MMAE/FS to react faster to partial actuator/surface failures than looking at each failure hypothesis individually; this was the intention all along, any time there are spawned filters (which, in this simulation, is always).

5.3.6 Detection Performance of the Complete and Partial Rudder Failures

Figures 25, 26, 27, and 28 show the poor performance of the MMAE/FS in detecting complete and partial rudder actuator/surface failures. This is in part caused by the lack of a yaw maneuver, resulting in reduced failure identifiability for the rudder. The CR is also used in these cases in order to keep the aircraft with an actuator failure close to the original trimmed flight conditions. Since there is not enough control authority in the other flight control actuators to compensate for a failed rudder, the CR cannot maintain good control over the aircraft with a rudder failure. This then causes even worse detection failure detection performance by the MMAE/FS. This will be further explained later in this chapter. The elemental probability flows to the correct hypothesis (rudder failure) but also some elemental probability flows to the right stabilator fully failed hypothesis. This also causes the bank to be swapped, bringing online other spawned filters based on partial stabilator failures before swapping back to the correct bank representing partial rudder failures.

5.3.7 Detection Performance of Complete Sensor Failures

Finally, Figures 29 – 34 show the capability of the MMAE/FS to detect sensor failures during maneuvering flight. The elemental probability plots of the spawned filters and the estimates of effectiveness are left out since they show no information about the MMAE/FS detection capability in detecting sensor failures. In actual implementation, the spawned filters would not be brought online for a sensor failure and the MMAE/FS does not provide an estimate of sensor effectiveness. The first figures represent the longitudinal sensors (Figures 29-31) and the pitch maneuver is used to excite the aircraft in order for the MMAE/FS to detect these sensor failures. The last figures represent the lateral sensors (Figures 32-34) and the roll maneuver is used to excite the aircraft in order for the MMAE/FS to detect these sensor failures. In all cases, the sensor failures were detected by the MMAE/FS during the first doublet maneuver and the detection was maintained through the entire simulation. The two problems found in the sensor failure detection capability of the MMAE/FS during pilot commanded maneuvers are the normal acceleration sensor failure and the pitch rate and yaw rate sensor failure cases which are described below.

Figure 31 shows the elemental probability plots given the case of a normal acceleration sensor failure. The sensor failure is properly detected by the MMAE/FS but then, after the pitch doublet maneuver, the elemental probability flows to the hypothesis based on the lateral acceleration sensor failure. This elemental probability flow is then reversed at the onset of the long duration pitch maneuver. This elemental probability flow is caused by the lack of failure mode excitement. As the aircraft does not maneuver after the completion of the pitch doublet, the MMAE/FS has a difficult time maintaining the detection capability of the normal acceleration sensor. Once the long duration pitch maneuver is begun, enough failure mode identifiability exists for the MMAE/FS to identify the correct sensor failure clearly. This type of elemental probability flow is one of the factors for the dither timing (how long after a maneuver is completed to apply the purposeful dither signal in order to excite the failure modes, to maintain good failure detection performance of the MMAE/FS).

As mentioned earlier in this chapter, the effect of being at a different attitude than straight-and-level flight can be seen in the MMAE/FS ability to maintain detection of the pitch rate and yaw rate sensor failures, as shown in Figures 30 and 32 at the end of the simulation. By implementing model scheduling based on aircraft attitude, as well as dynamic pressure, the ability of the MMAE/FS to maintain detection capabilities through large pilot commanded maneuvers could be increased.

5.4 Control Redistribution Performance

5.4.1 Control Redistribution

Once an actuator failure is detected and an estimate of the effectiveness is made by the MMAE/FS, the proper control must be applied in order to achieve the desired performance. The following state value plots demonstrate the capability of the MMAE/FS/CR to apply the proper control once a full actuator failure is detected. The longitudinal plots are of the state values of pitch angle (θ in degrees), pitch rate (q in degrees per second), angle of attack (α in degrees), normal acceleration (a_n in ‘g’s’), velocity (feet per second) and altitude (feet). The lateral plots are of the state values of roll angle (ϕ in degrees), roll rate (p in degrees per second), yaw rate (r in degrees per second), slipstream angle (β in degrees), and normal acceleration (a_y in ‘g’s’). The angle of attack, pitch rate, normal acceleration, roll rate, yaw rate, and lateral acceleration are states in the MMAE/FS models. The state value plots are presented for the cases in the following order:

1. Fully Functional Aircraft
2. Left Stabilator Failure
3. Partially Failed Left Stabilator
4. Right Stabilator Failure
5. Partially Failed Right Stabilator
6. Left Flaperon Failure
7. Partially Failed Left Flaperon
8. Right Flaperon Failure
9. Partially Failed Right Flaperon
10. Rudder Failure
11. Partially Failed Rudder

The dotted line on each state value plot represents the fully functional performance of the aircraft. The solid line on each of the state value plots represent the mean performance of the aircraft given the failures and controlled by the CR. The mean ± 1 standard deviation is not shown because the standard deviation of the states is too small to show. In these plots, the true value of effectiveness, ε_{true}, is used rather than the MMAE-estimated value, to evaluate the best control performance achievable through a CR algorithm, assuming perfect identification of failure status. Subsequently in Section 5.5, the performance of an actually implementable MMAE/FS/CR algorithm will be compared to this artificially informed algorithm's performance. As was the case before, the "desired" plots show the desired pilot-commanded maneuver in pounds of force on the cockpit control stick.

Figures 35-56 on CR performance are presented in a group for convenience. Then, starting on page 150, they will be discussed in detail.

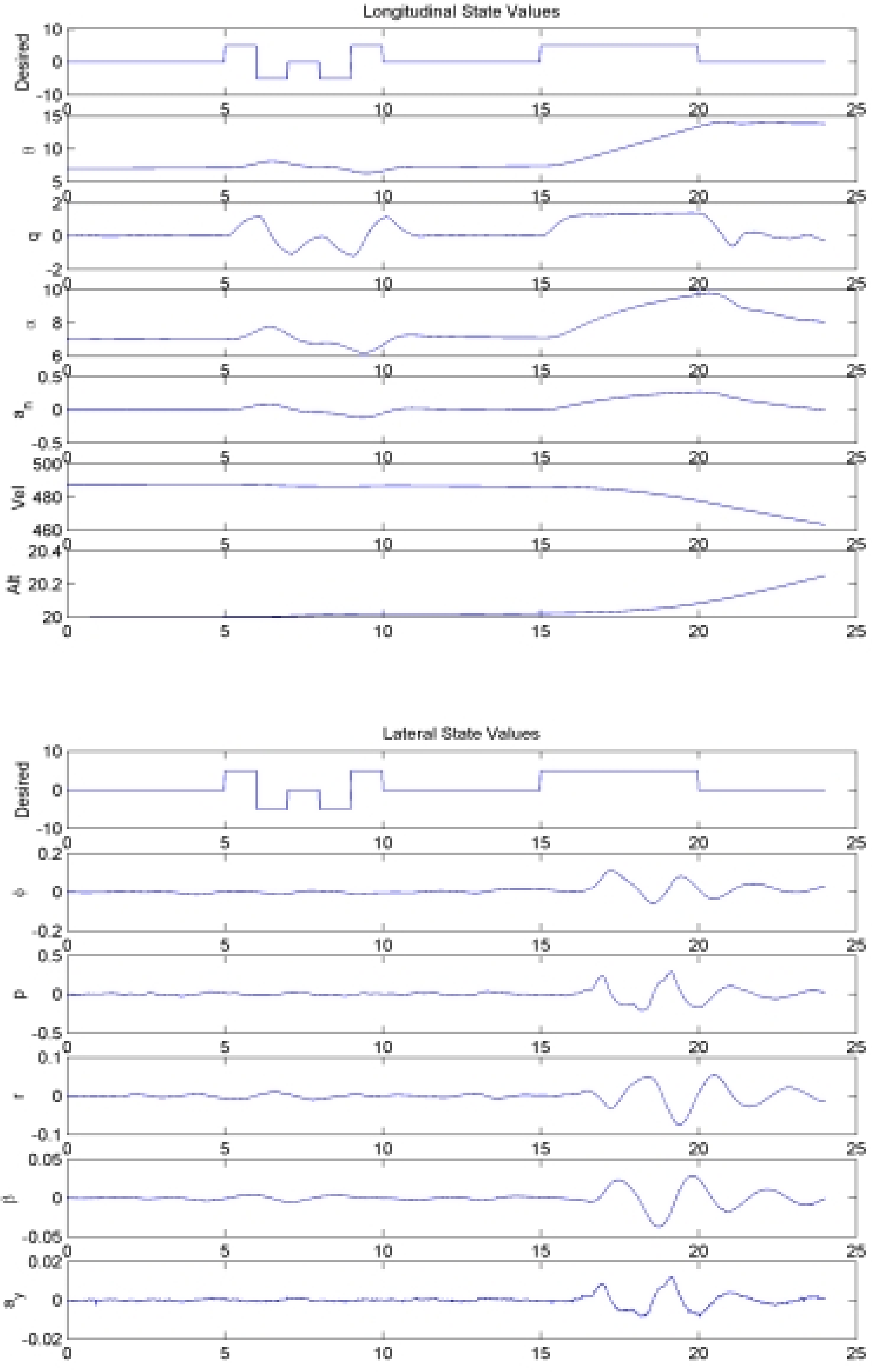


Figure 35: State Values, Fully Functional Aircraft, Pitch Maneuver

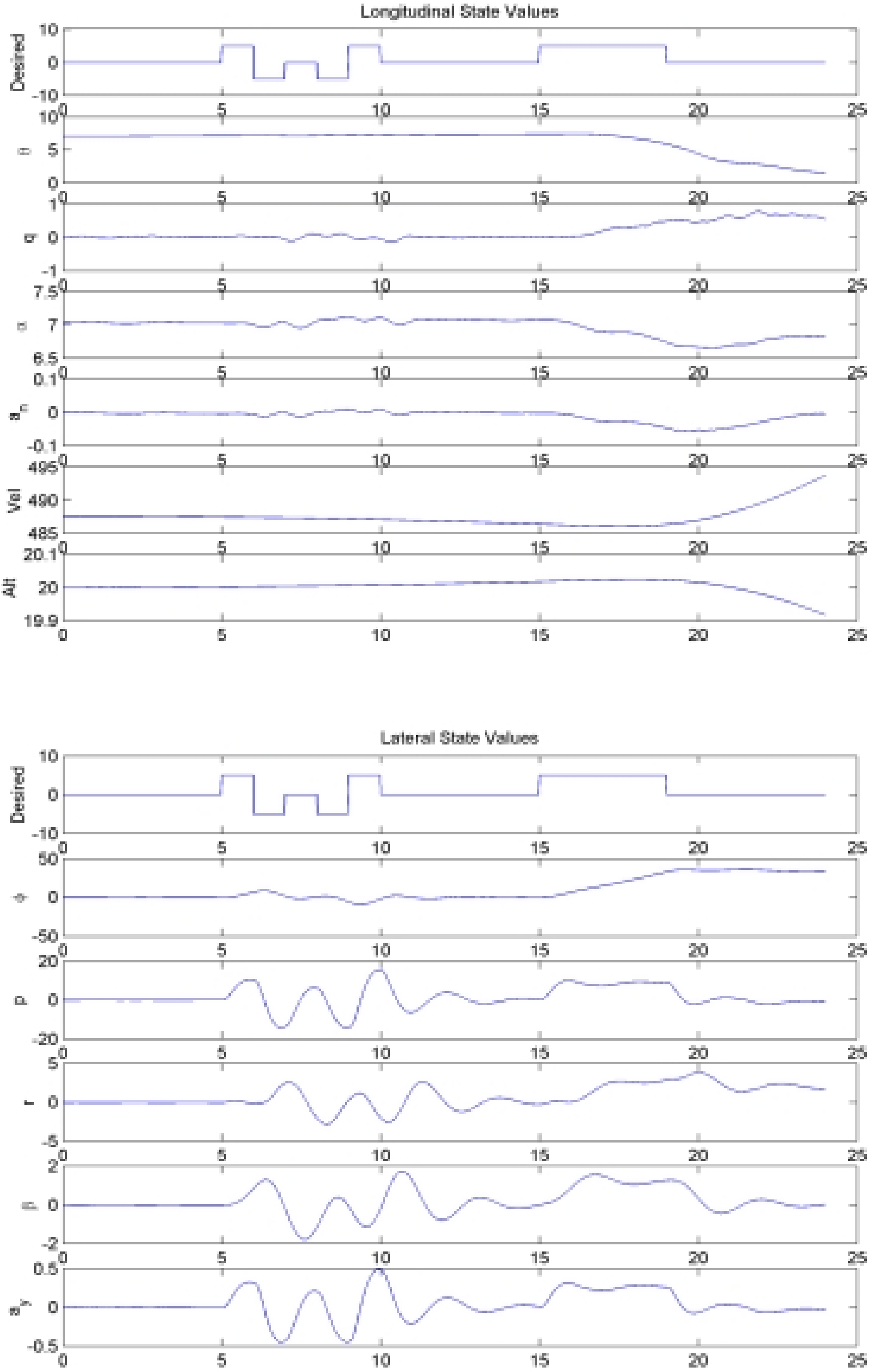


Figure 36: State Values, Fully Functional Aircraft, Roll Maneuver

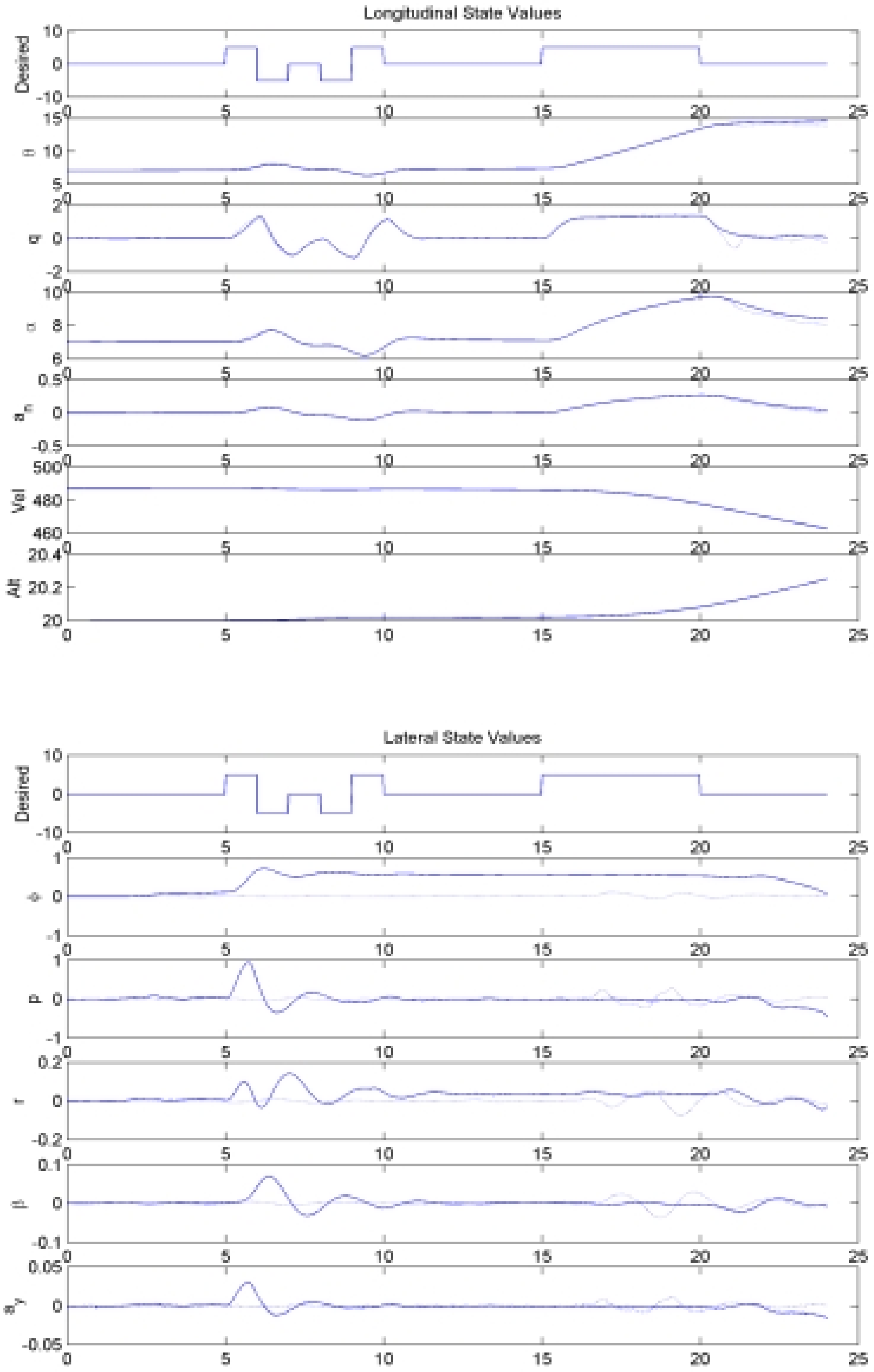


Figure 37: State Values, Left Stabilator Failure, ε = 0%

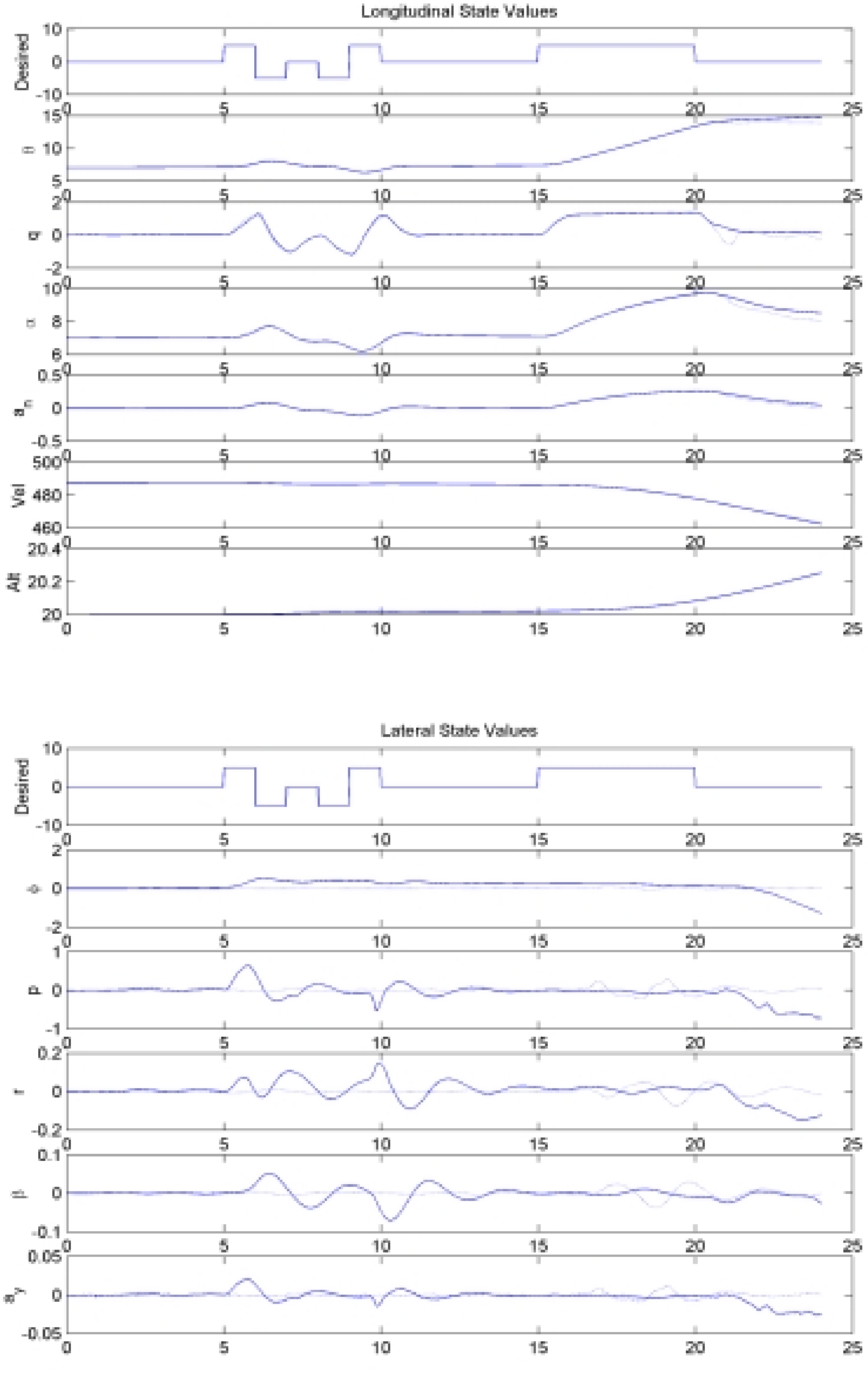


Figure 38: State Values, Left Stabilator Failure, ε = 25%

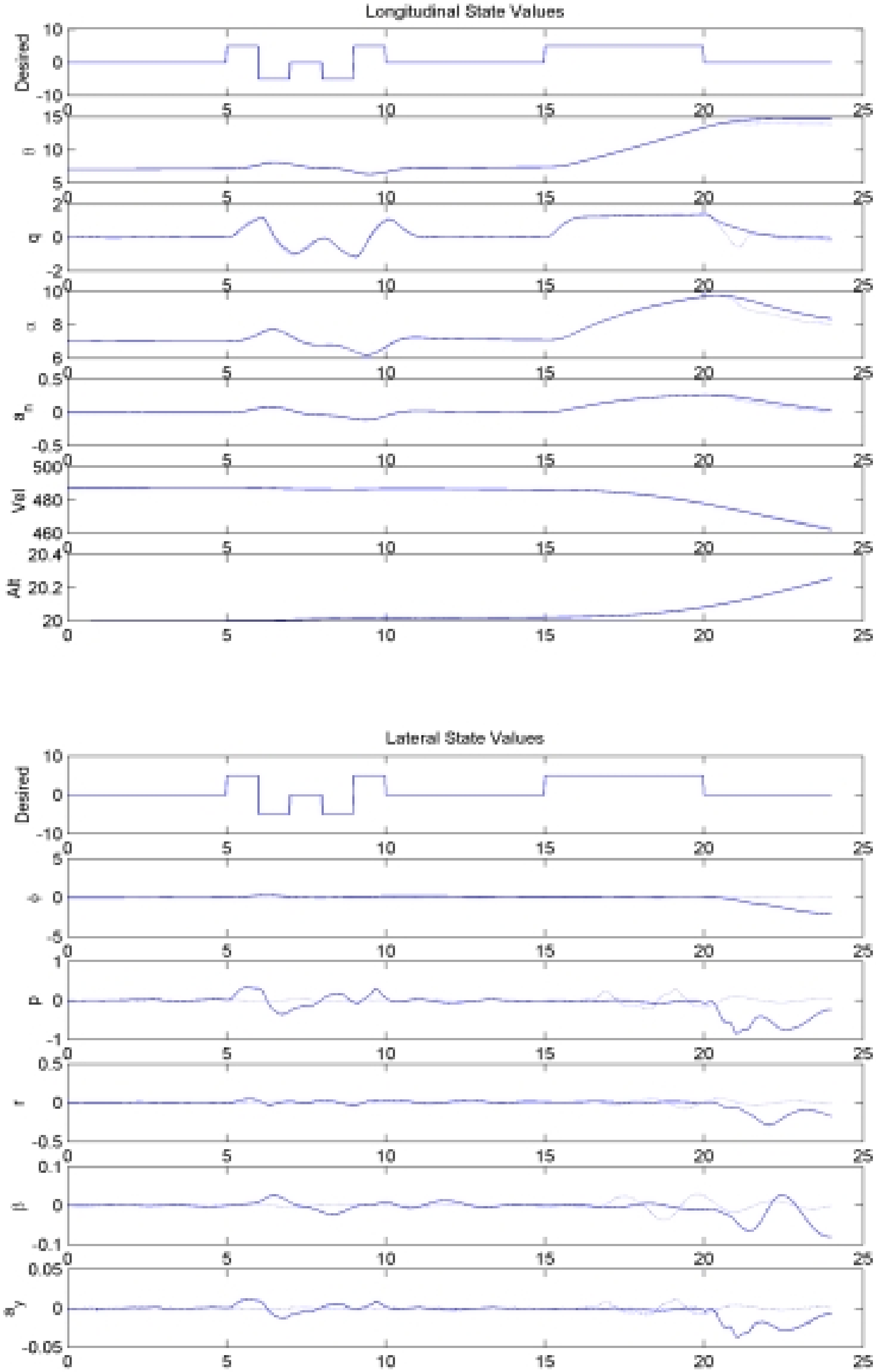
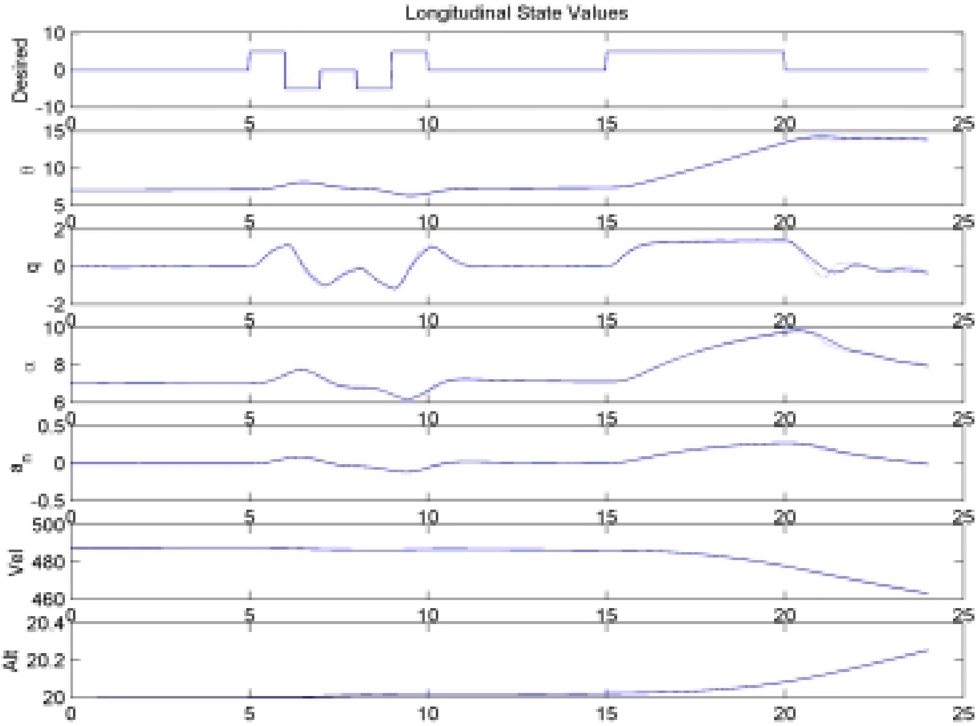


Figure 39: State Values, Left Stabilator Failure, ε = 50%



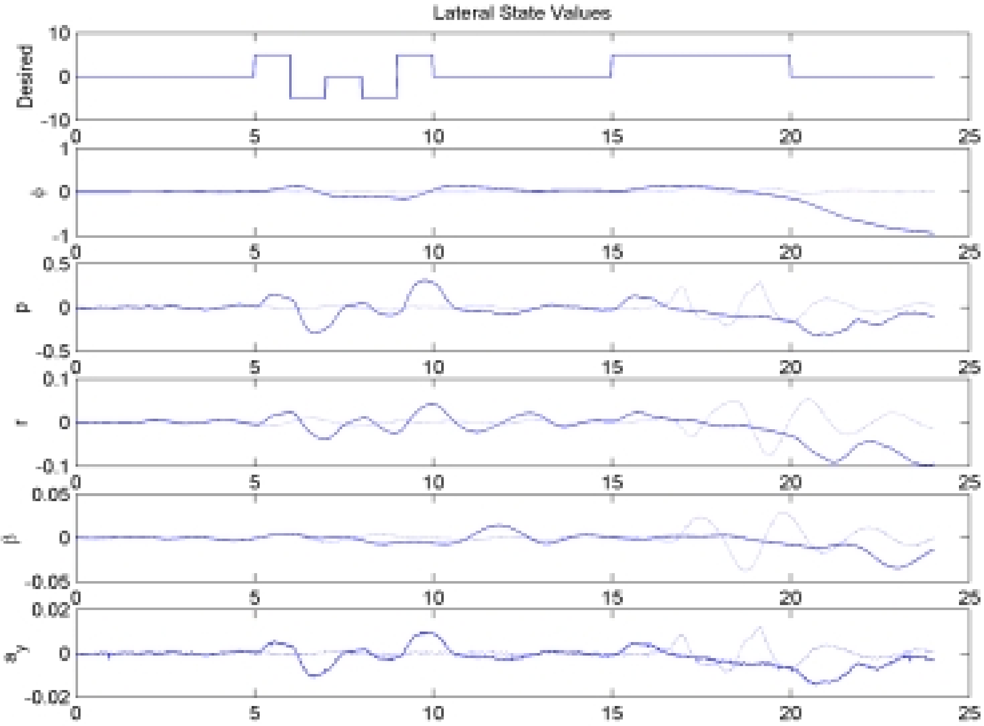


Figure 40: State Values, Left Stabilator Failure, ε = 75%

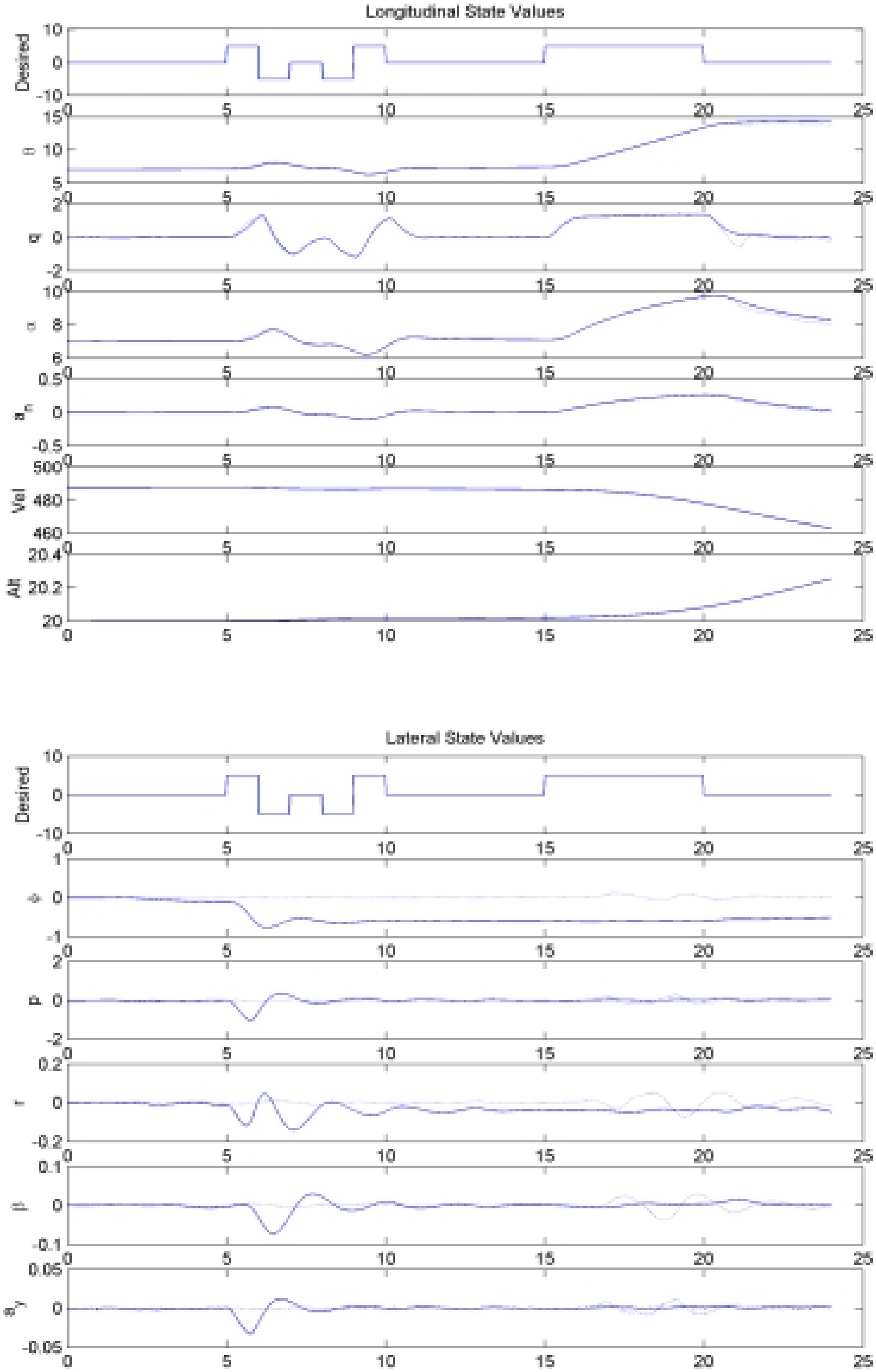
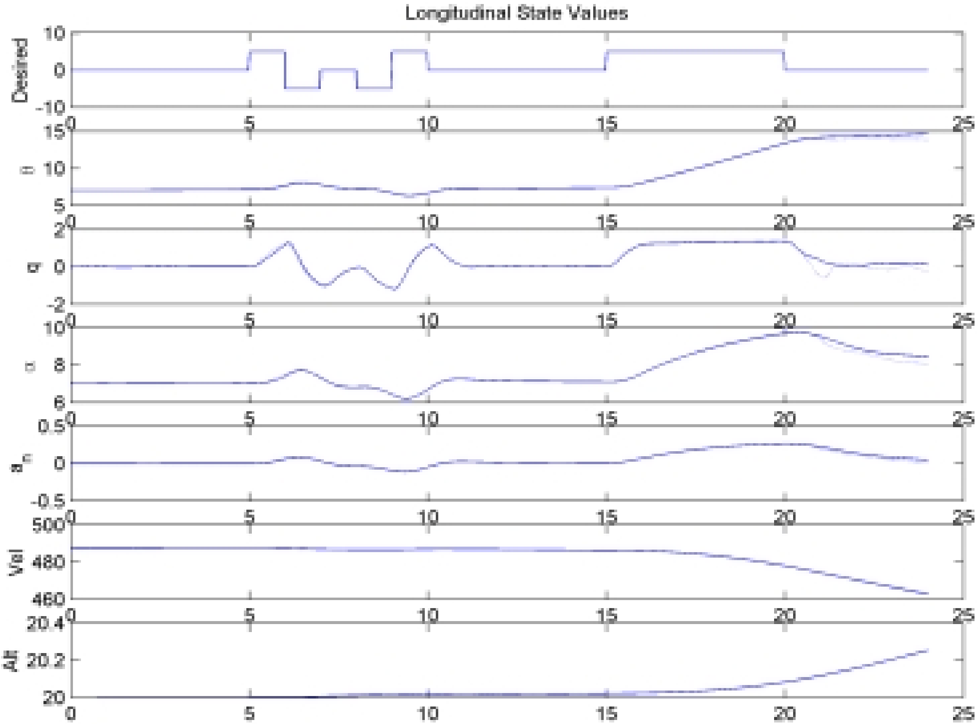


Figure 41: State Values, Right Stabilator Failure, ε = 0%



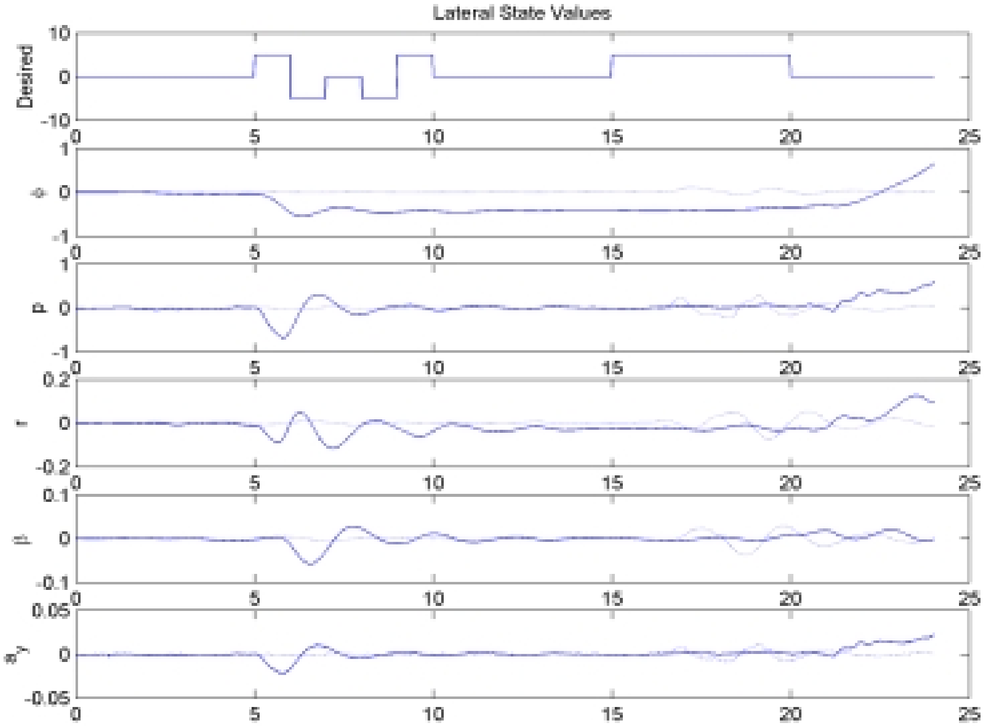


Figure 42: State Values, Right Stabilator Failure, ε = 25%

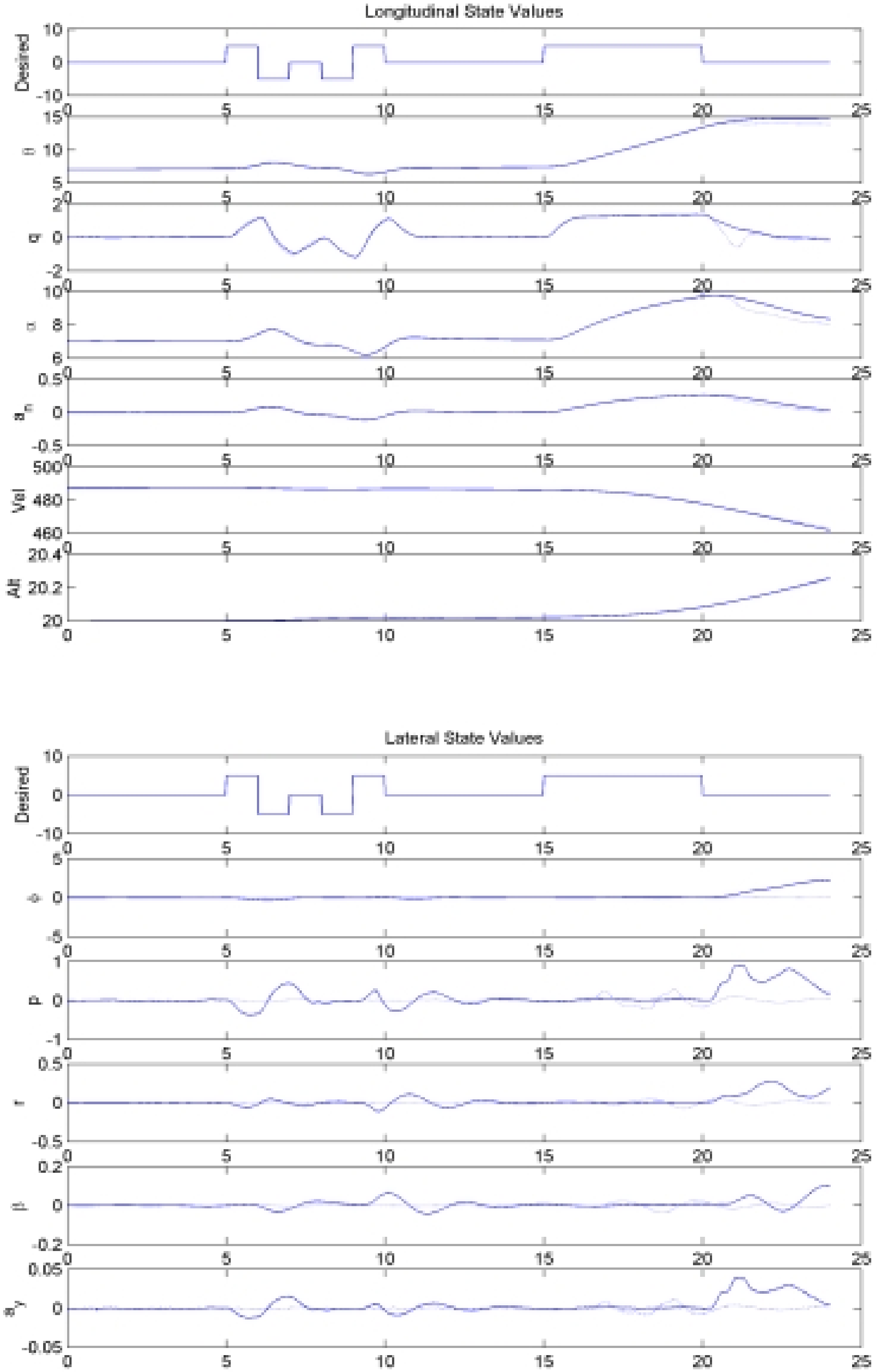


Figure 43: State Values, Right Stabilator Failure, ε = 50%

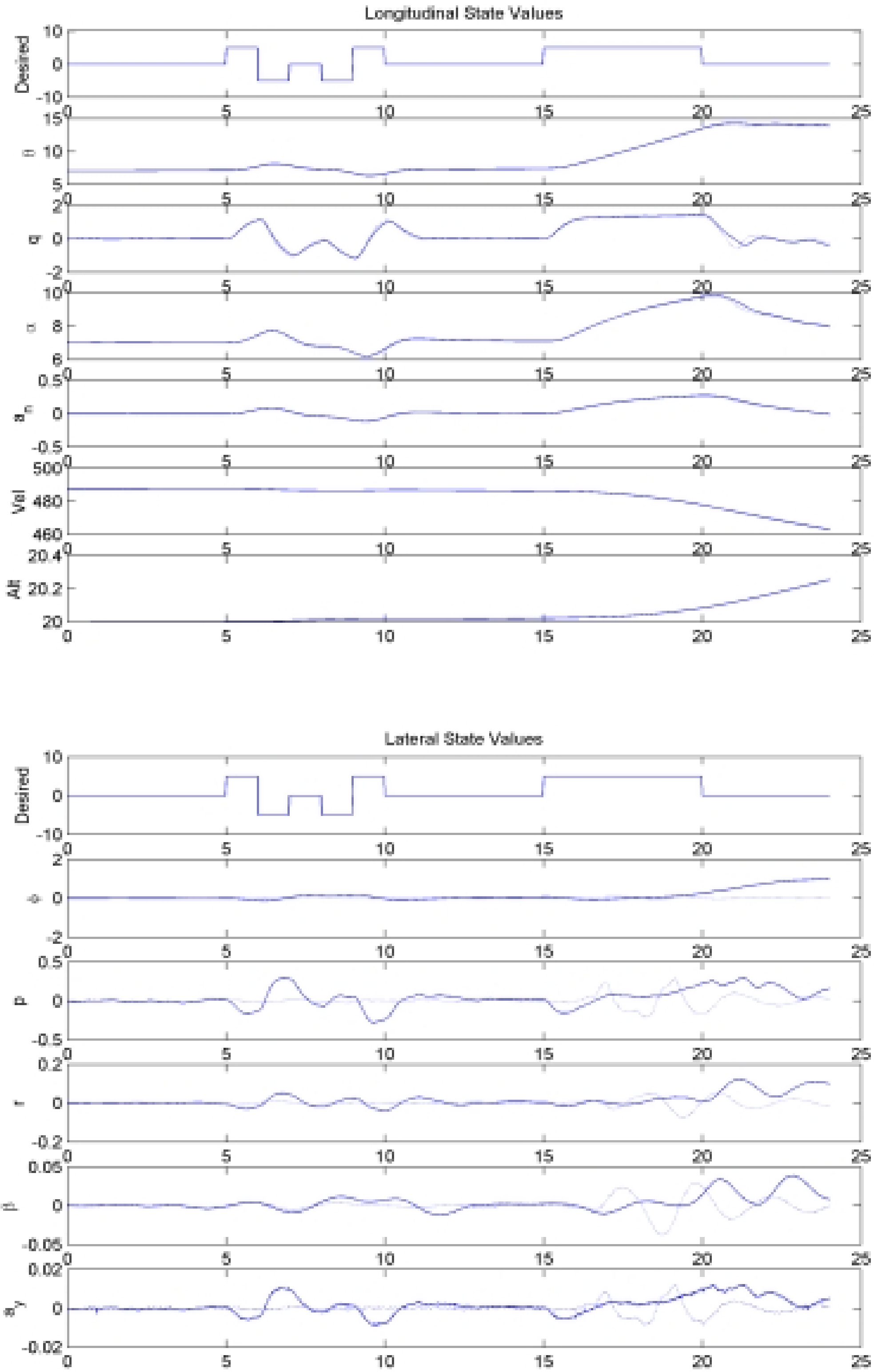


Figure 44: State Values, Right Stabilator Failure, ε = 75%

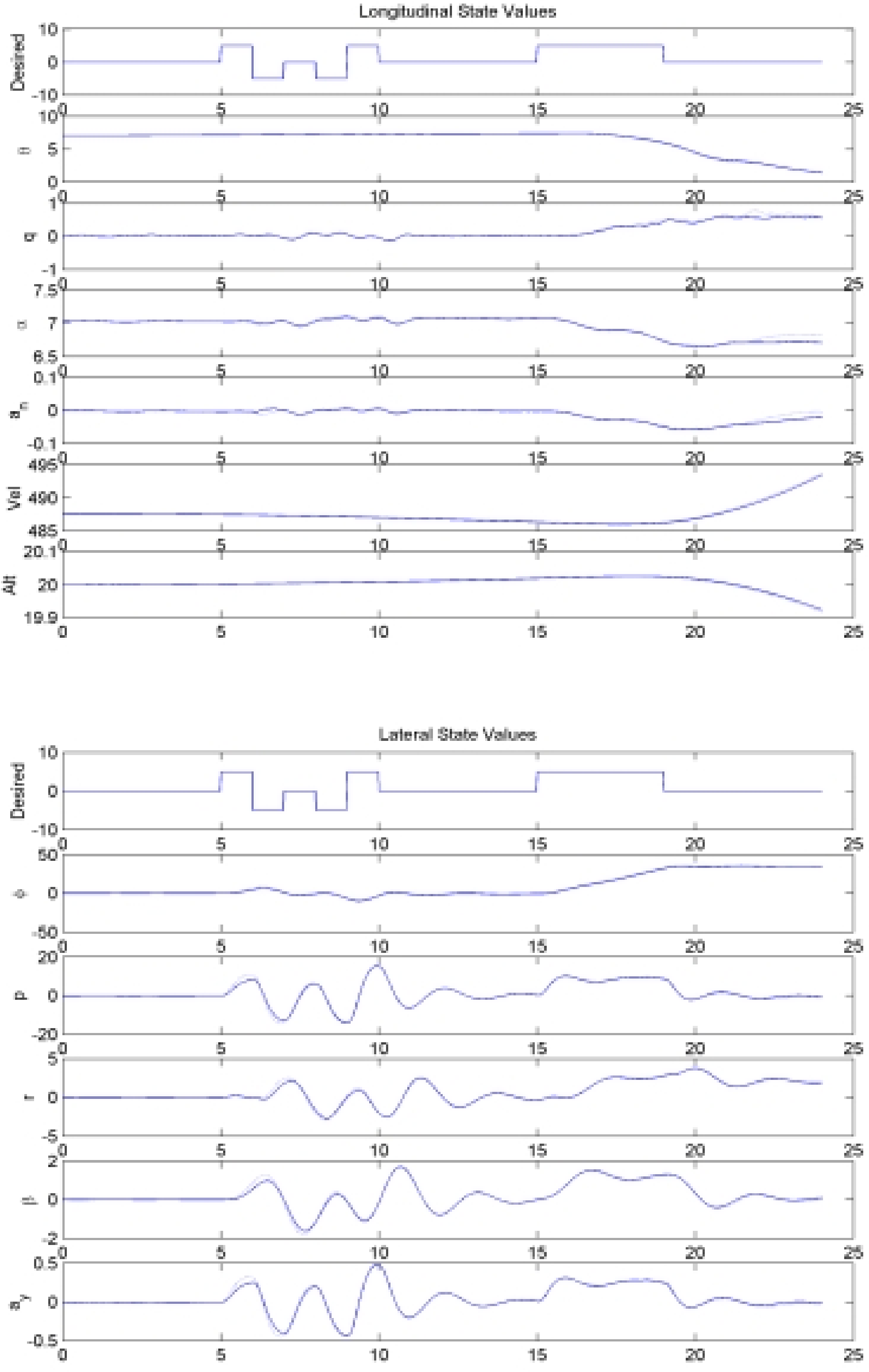
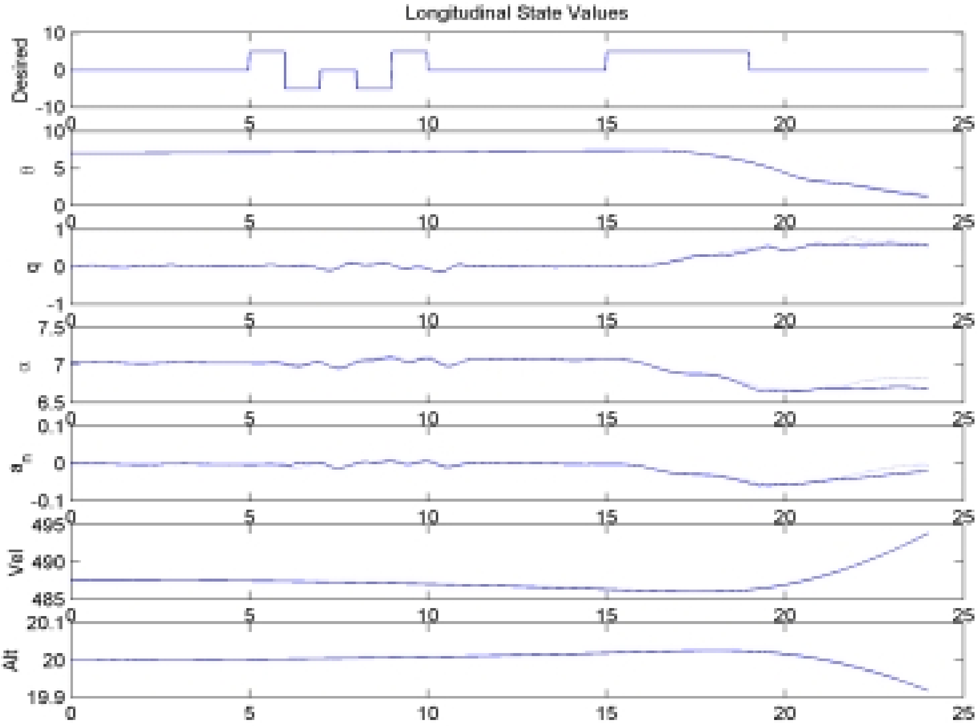


Figure 45: State Values, Left Flaperon Failure, ε = 0%



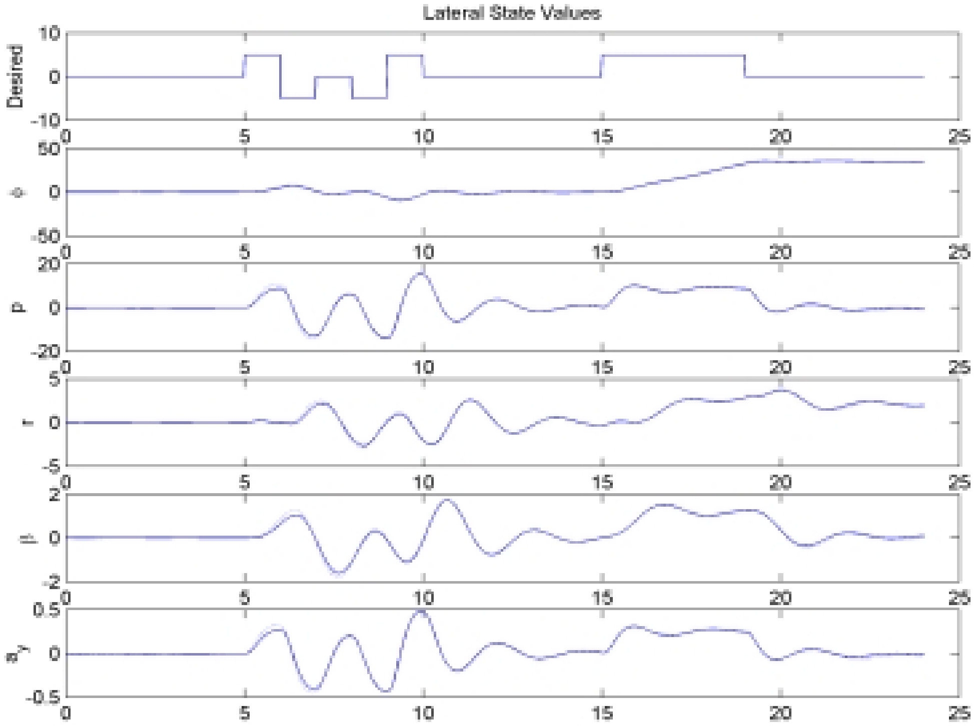


Figure 46: State Values, Left Flaperon Failure, ε = 25%

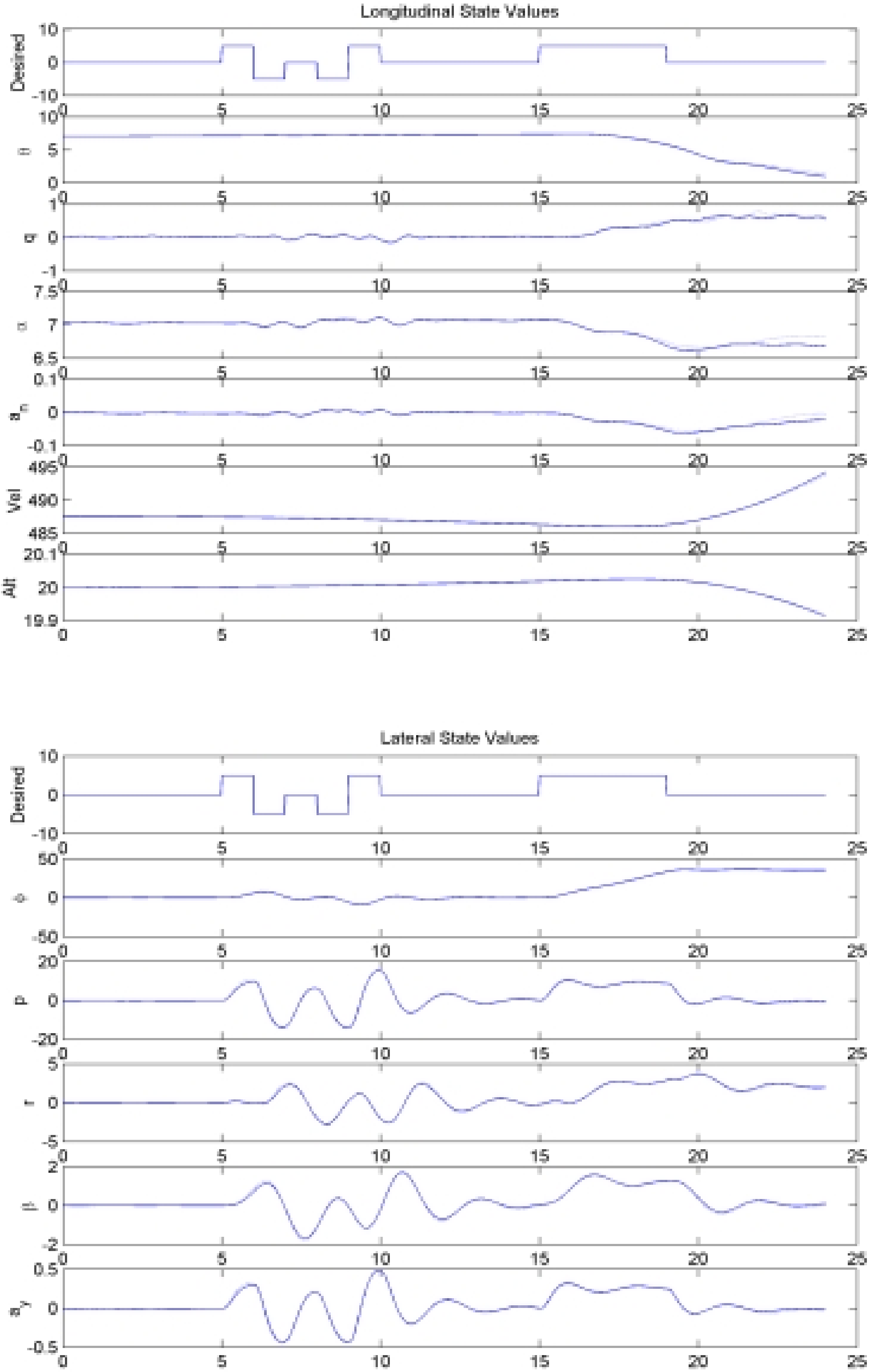


Figure 47: State Values, Left Flaperon Failure, ε = 50%

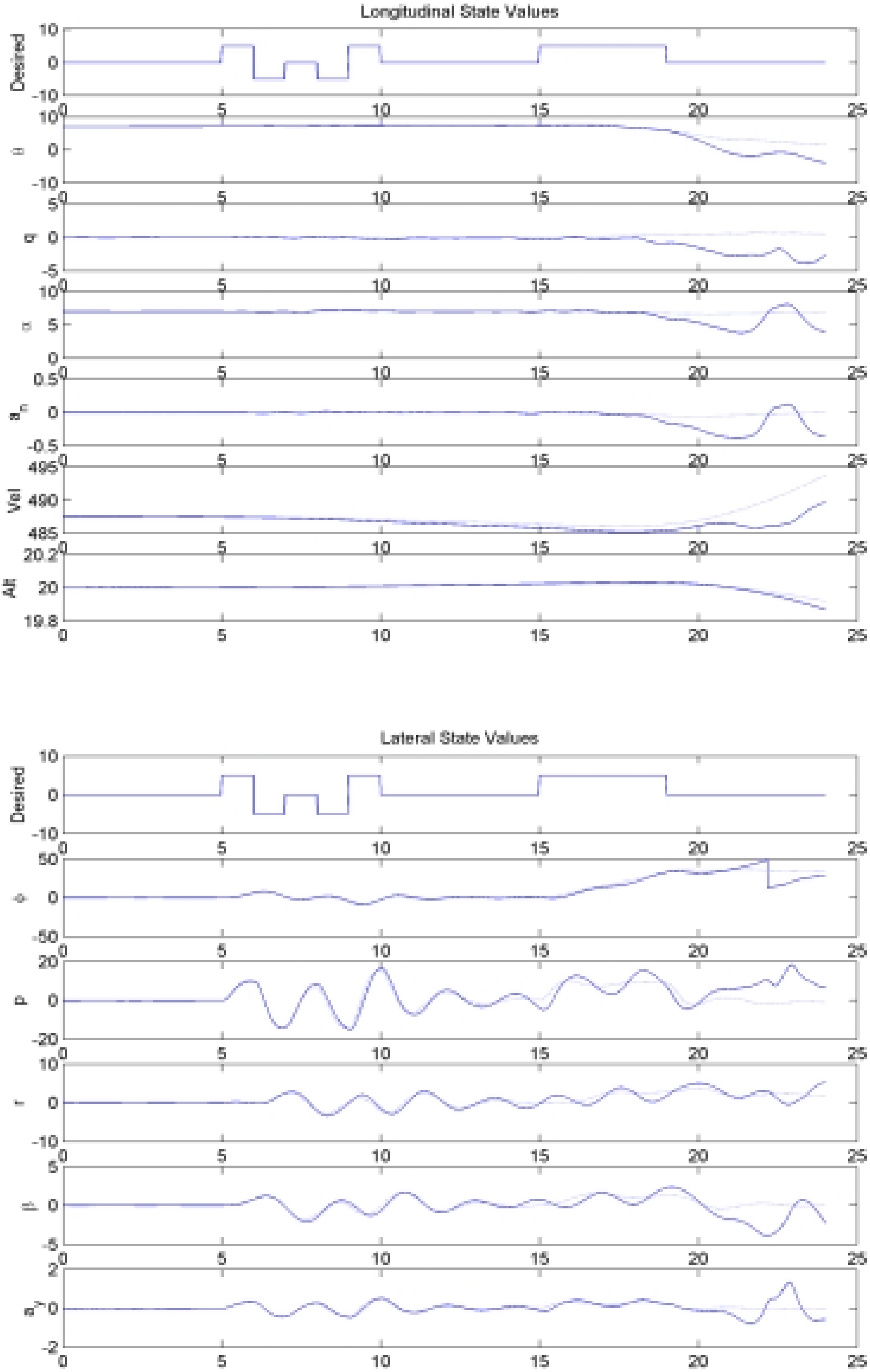


Figure 48: State Values, Left Flaperon Failure, ε = 75%

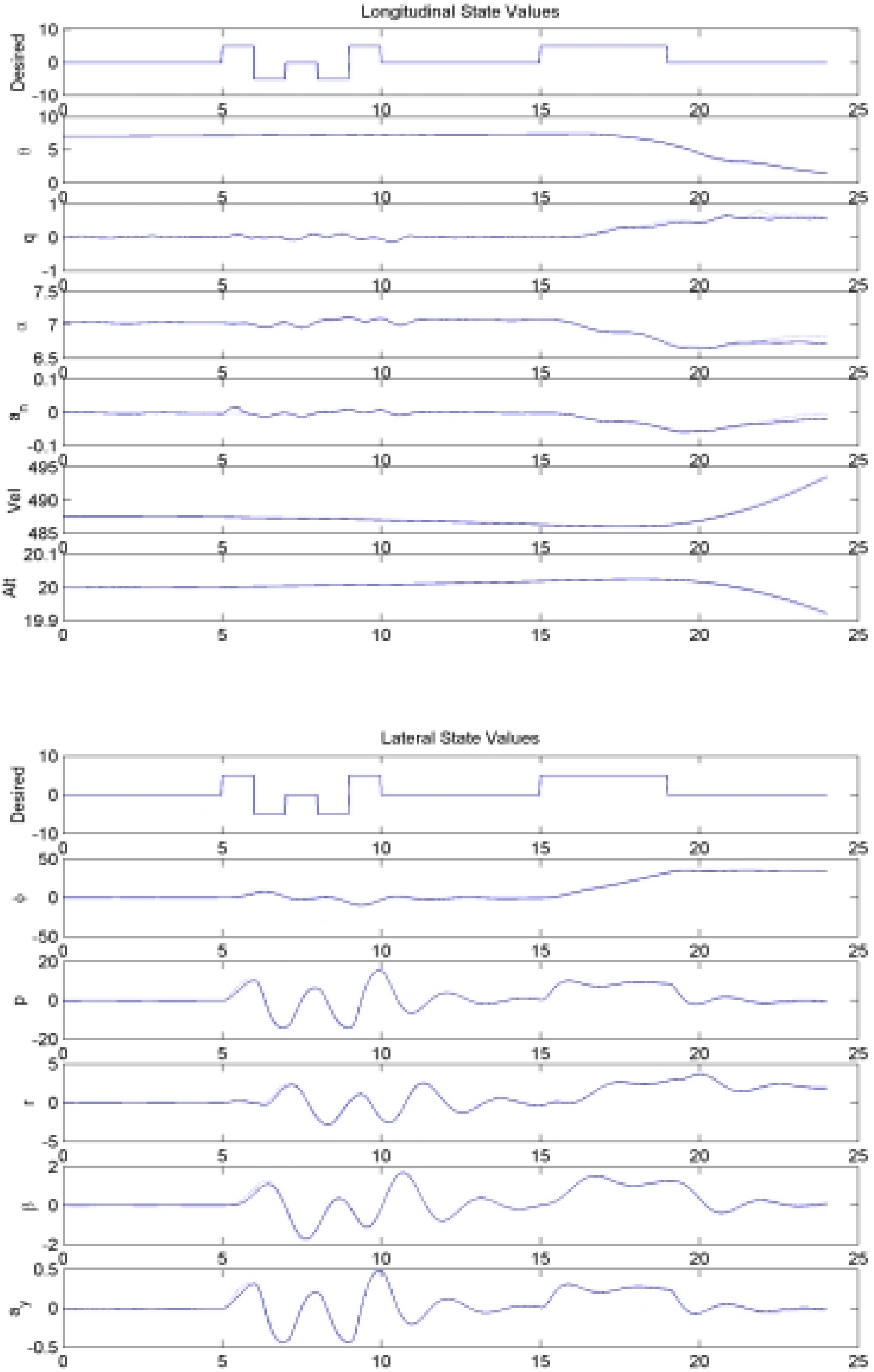


Figure 49: State Values, Right Flaperon Failure, ε = 0%

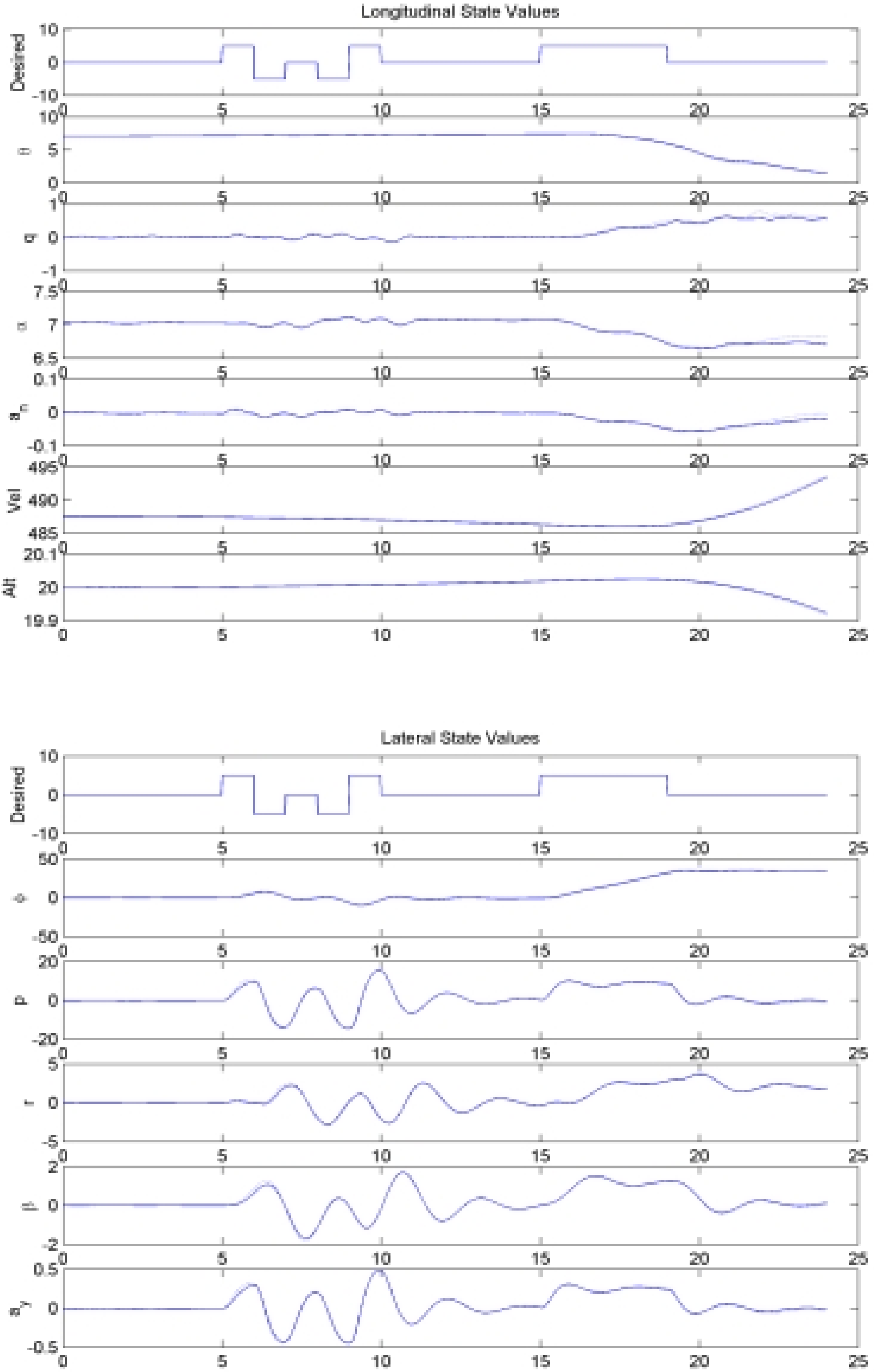
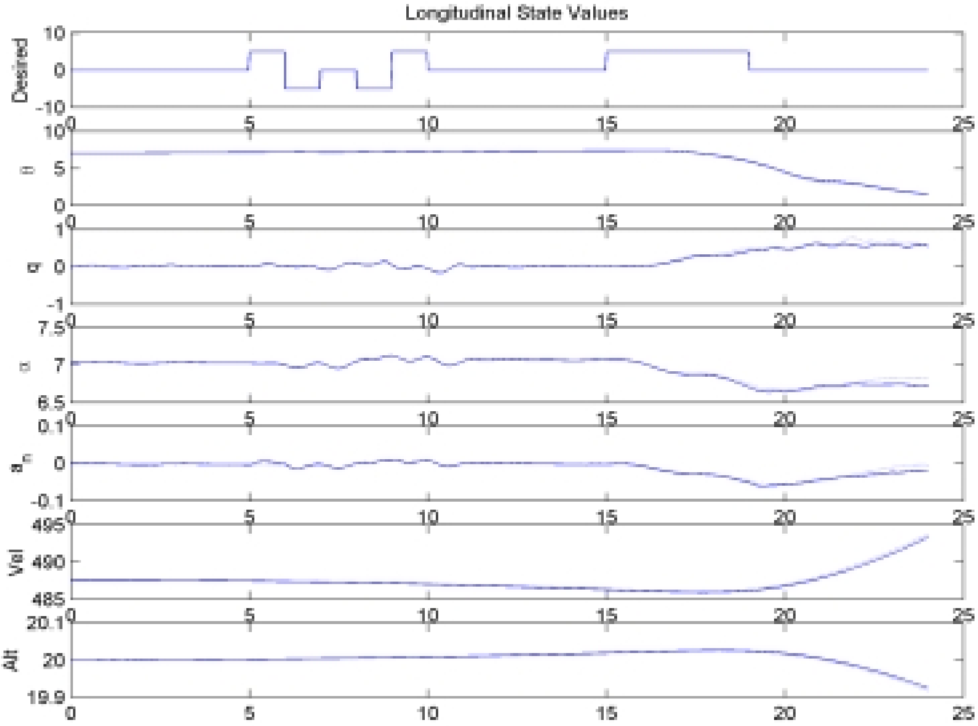


Figure 50: State Values, Right Flaperon Failure, ε = 25%



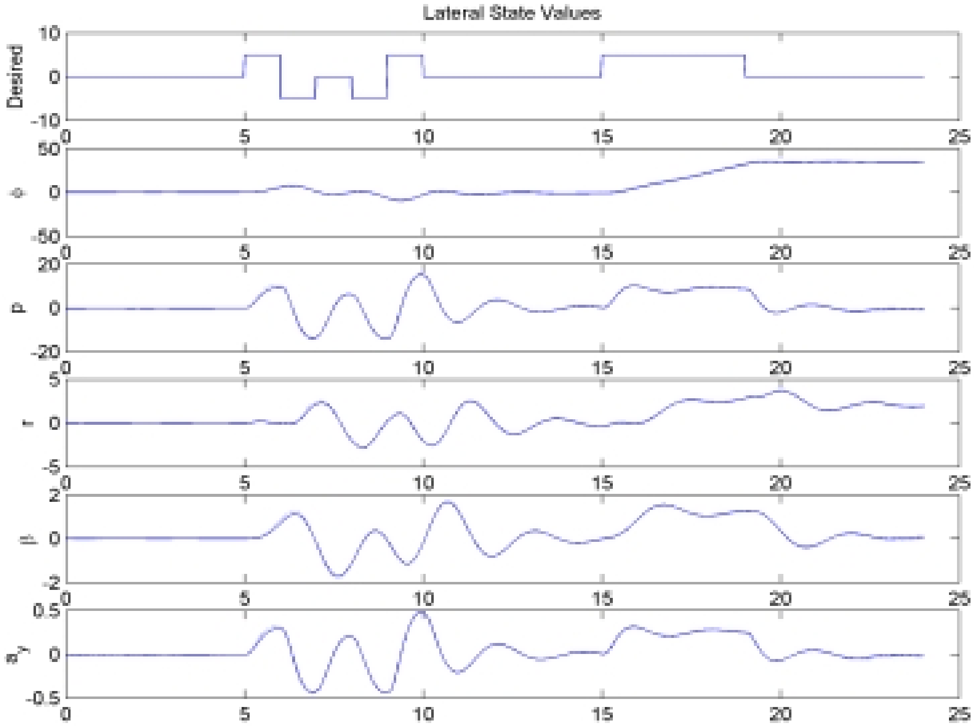


Figure 51: State Values, Right Flaperon Failure, ε = 50%

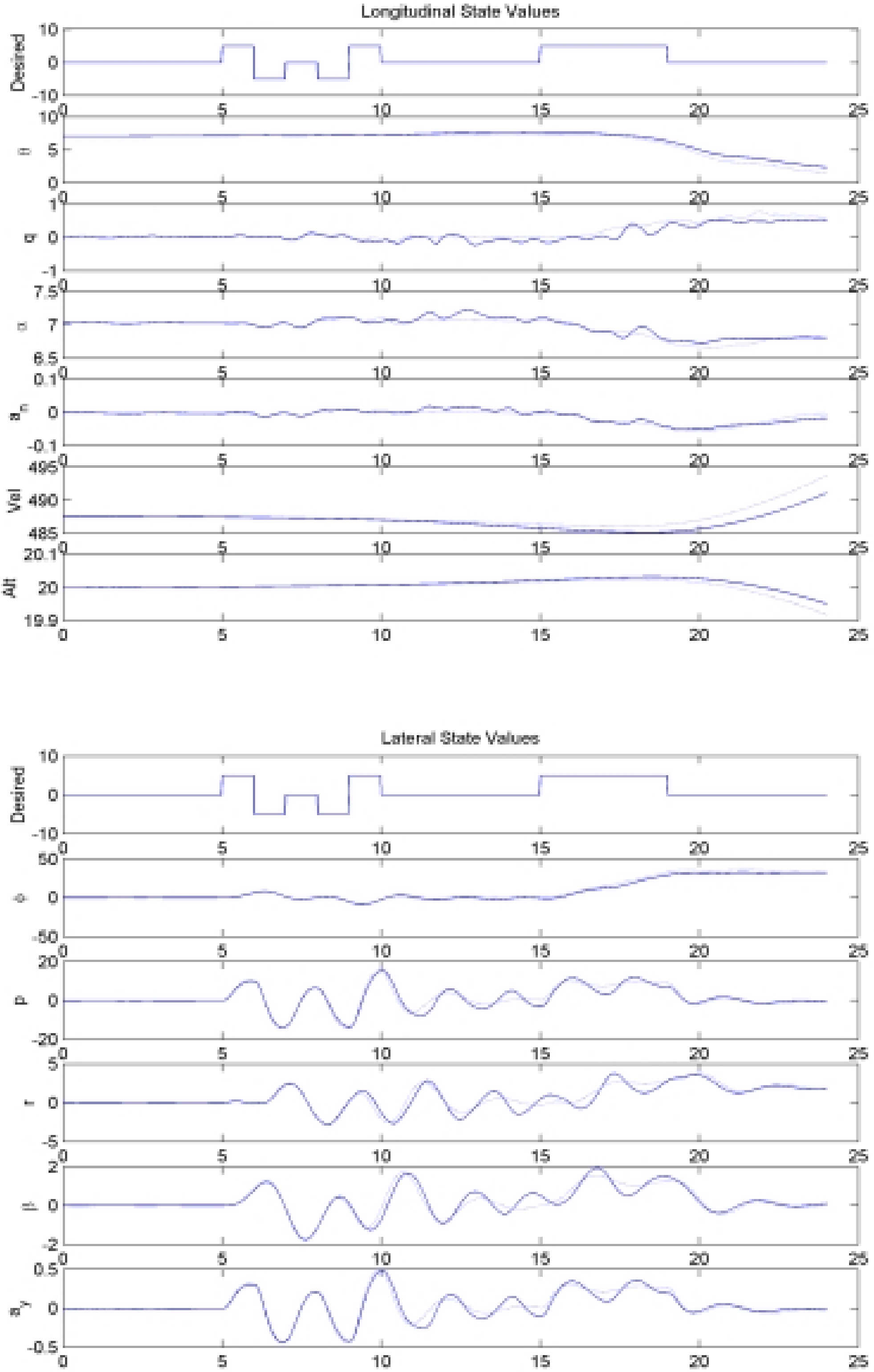


Figure 52: State Values, Right Flaperon Failure, ε = 75%

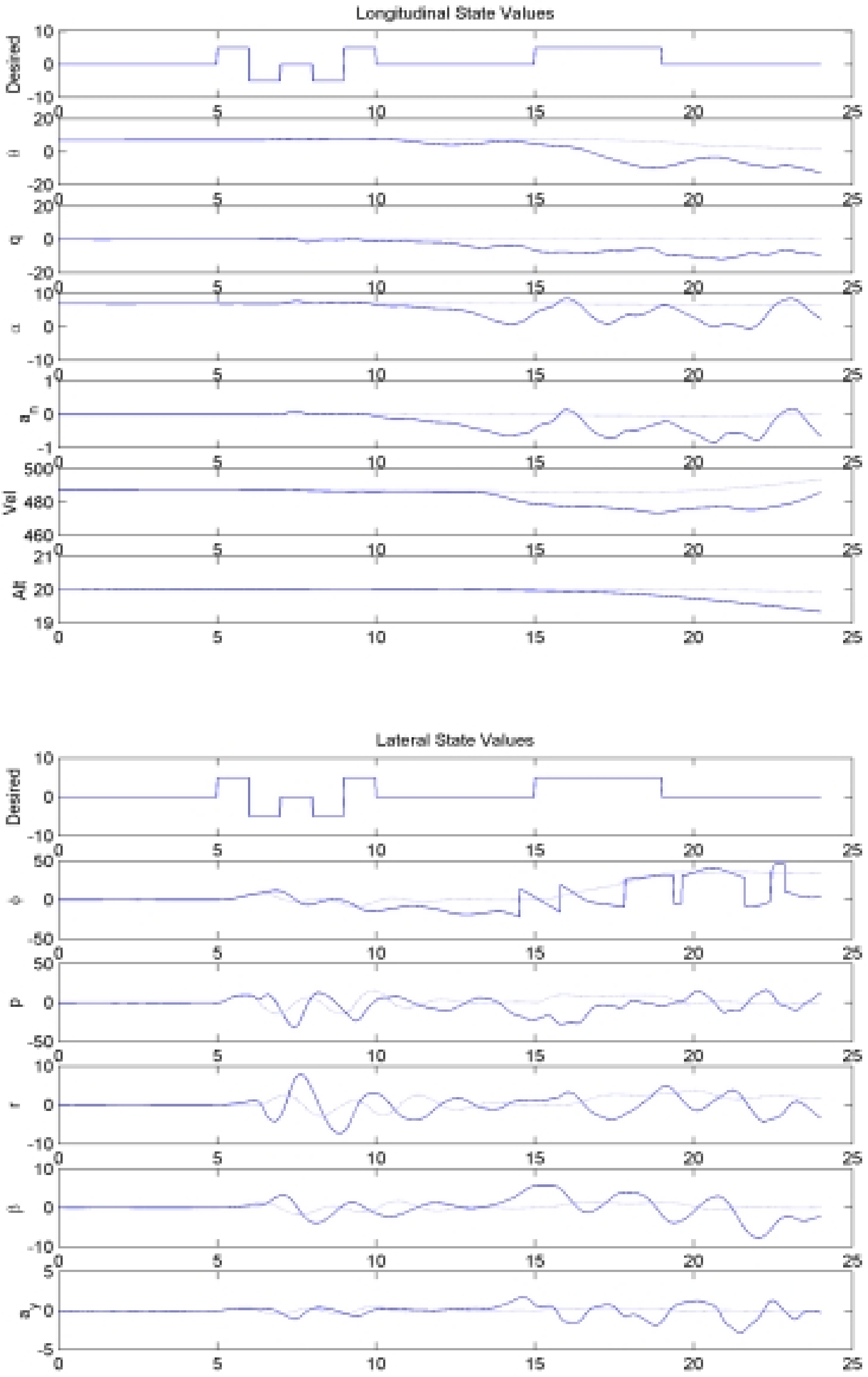


Figure 53: State Values, Rudder Failure, ε = 0%

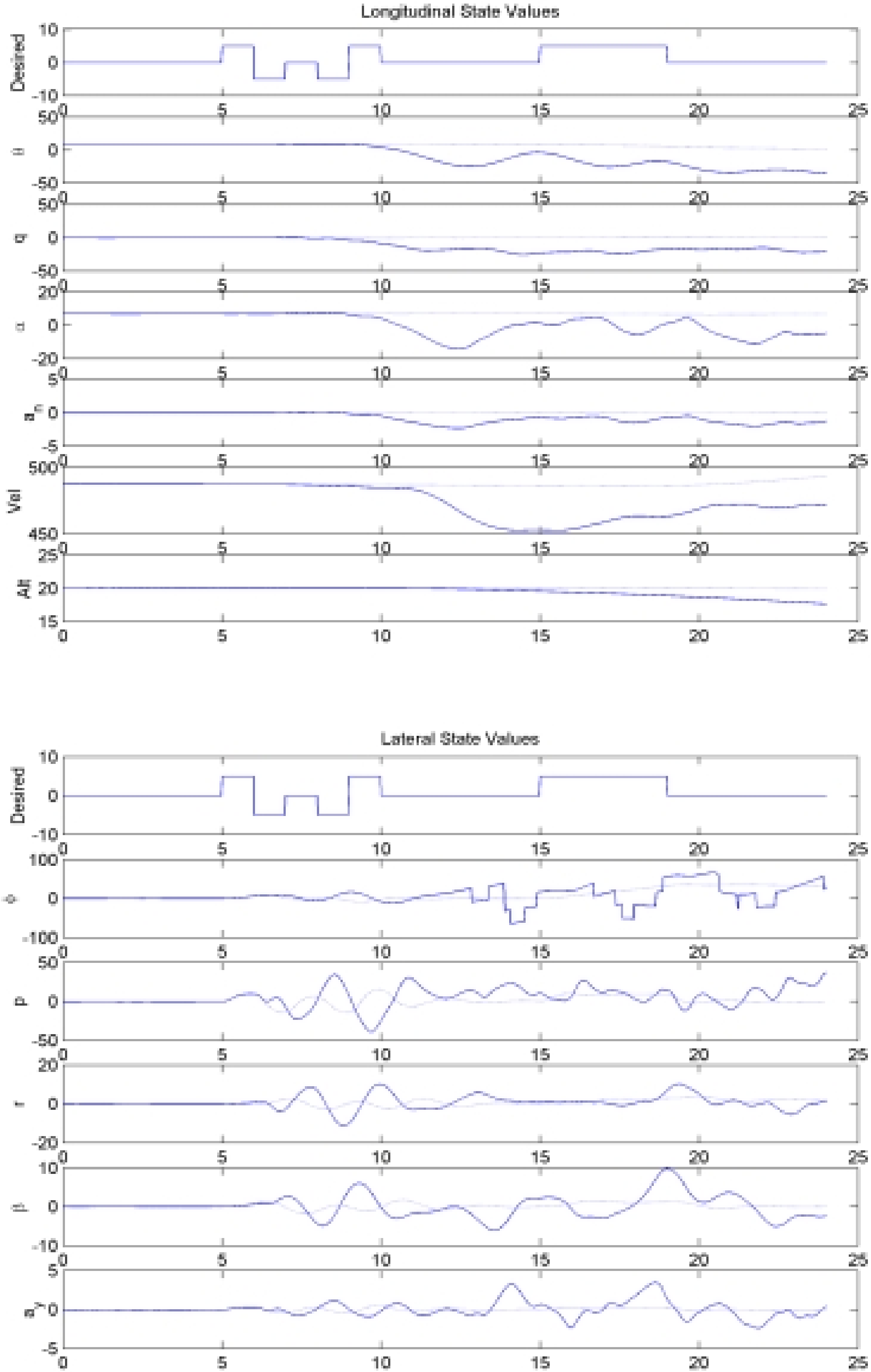


Figure 54: State Values, Rudder Failure, ε = 25%

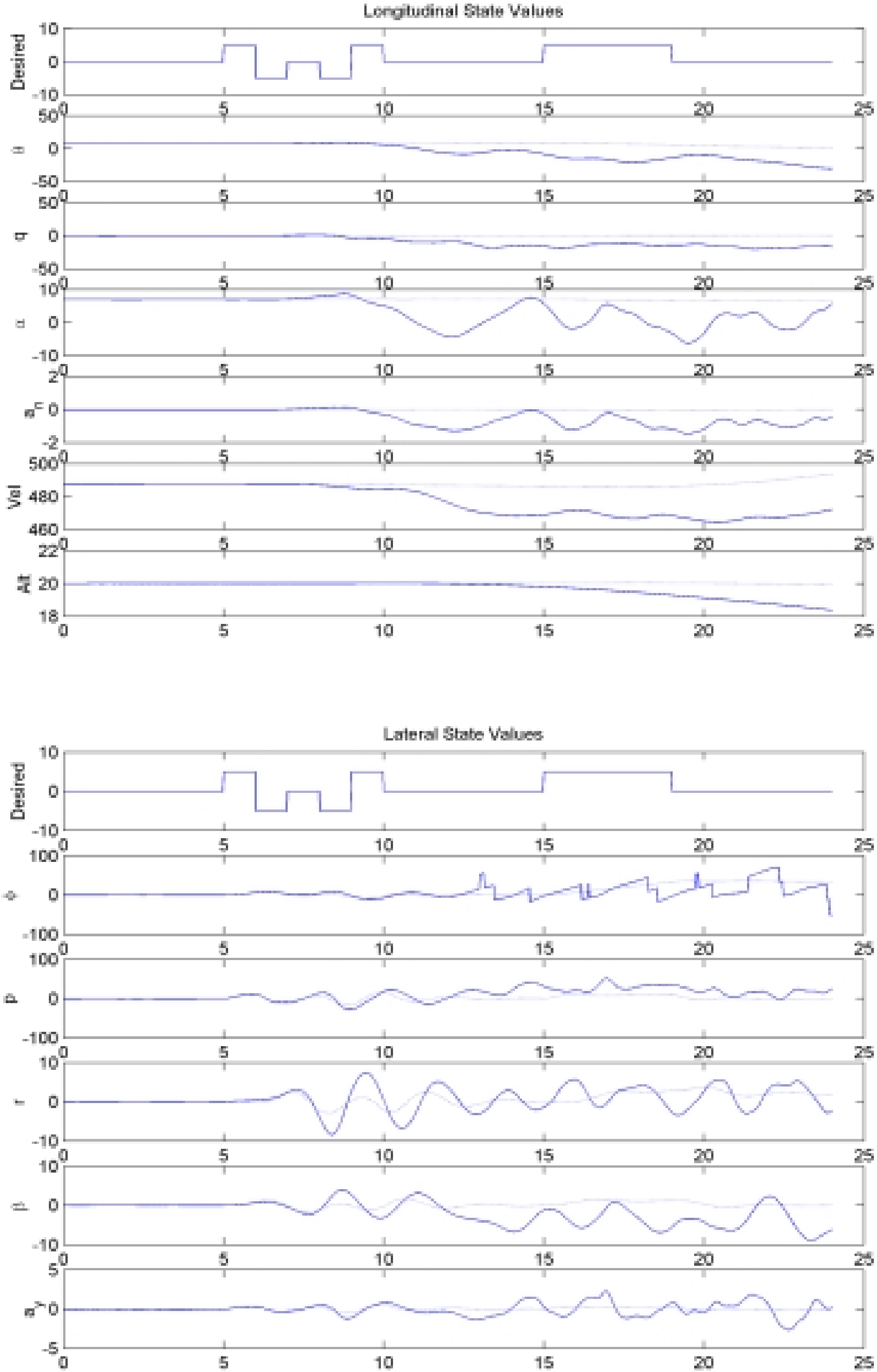


Figure 55: State Values, Rudder Failure, ε = 50%

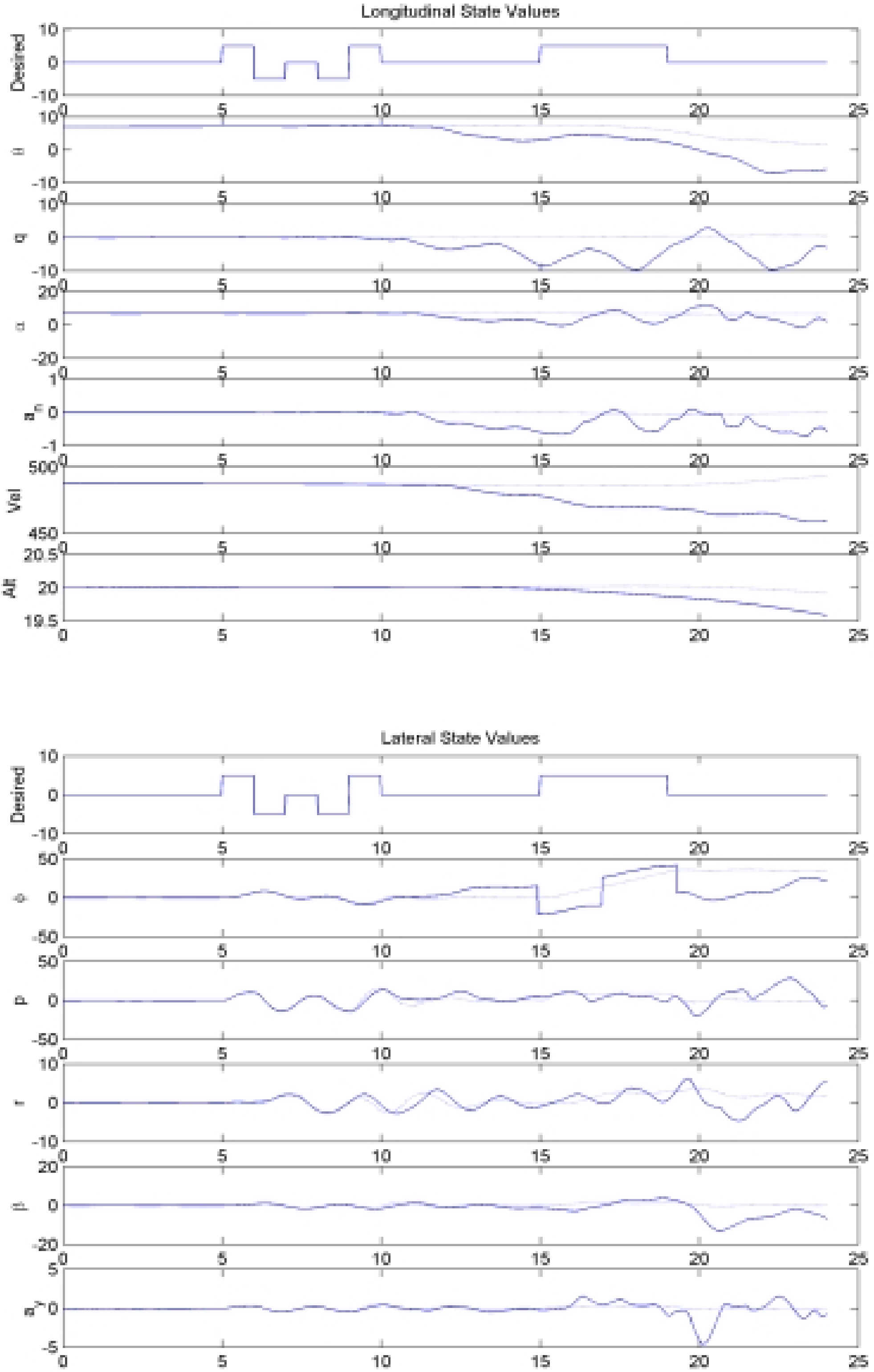


Figure 56: State Values, Rudder Failure, $\varepsilon = 75\%$

5.4.2 Control Redistribution Performance with Fully Functional Aircraft

Figures 35 and 36 present the fully functional aircraft cases for the pitch and roll maneuvers, respectively. These maneuvers are used as a baseline for the other figures representing the failure cases. The pitch maneuver is used for the longitudinal actuator failure cases (stabilators) and the roll maneuver is used for the lateral actuator failure cases (flaperons and rudder). It can be seen from Figures 35 and 36 that the fully functional aircraft state values are not exactly zero (or their nominal cases such as in angle of attack). The lateral state oscillations during the pitch maneuver in Figure 35, and the longitudinal state oscillations in Figure 36 are caused by the Block 40 Flight Control System (FCS) and are small in magnitude, as can be seen by the axis magnitudes. The large oscillations in the lateral states in Figure 36 are purposeful and are large in magnitude as compared to the lateral oscillations from Figure 35. It is important in review these figures (and all figures) to take note of the y- axis scale. In Figure 36, the aircraft looses altitude and gains airspeed at the end of the simulation. This is caused by the lack of autopilot control over the aircraft. As the aircraft rolls to approximately 40 degrees of bank, the nose of the aircraft drops since there is not backpressure on the pilot control stick applied and the autopilot system is not used.

5.4.3 Control Redistribution Performance with Full Actuator/Surface Failures other than the Rudder

The performance of the CR algorithm given complete actuator failures is shown in Figures 37, 41, 45, and 49 for the stabilator and flaperon failures. Once the failure is declared by the MMAE/FS, the CR algorithm is activated based on the failure declaration and these figures show the control performance of the CR algorithm. The effectiveness ε_{true} is used in the CR algorithm in all cases when that algorithm becomes activated, in order to demonstrate the best possible control given the exact value of failure effectiveness. In all the failure cases, the largest variation from the desired maneuver pitch and roll angles is approximately one to two degrees. The delays are caused mostly by the time the MMAE/FS requires to make the initial declaration.

Figure 37 clearly shows this. This is the completely failed left stabilator case. The aircraft begins the doublet maneuver and the aircraft begins to roll, as can be seen in the lateral states. The MMAE/FS then makes the failure declaration and the CR algorithm is brought online. At that point, the roll stops at less than one degree and does not increase through the rest of the simulation. It can be seen, for instance from Figure 37, that the dotted line representing the desired pitch maneuver performance matches Figure 35, once one notes the scale change on the plots. The same is true for the roll maneuvers and Figure 36.

5.4.4 Control Redistribution Performance with Partial Actuator/Surface Failures

The partial actuator failure performance is presented in Figures 38 - 40, 42 - 44, 46 - 48, and 50 - 52 for the stabilator and flaperon partial failures. From these figures, it can be seen that the CR algorithm functions properly, given the failure declaration from the MMAE/FS and knowing the effectiveness of the partially failed actuator/surface exactly. The aircraft tracks the desired maneuver with little deviation. The only times the aircraft does not track the desired maneuvers properly is after the long duration maneuver. This is caused by the poor detection performance of the MMAE/FS while at a different aircraft attitude than straight-and-level. Consistent with the results seen in Section 5.3, this leads to a strong recommendation later in Chapter 6 that model scheduling be based on both dynamic pressure and aircraft attitude, and not just dynamic pressure as in these simulations. Since the MMAE/FS does not make the correct failure declaration while at the different aircraft attitude, the CR does not apply the proper control to the vehicle. This can be clearly seen in Figure 48, the 75% effective left flaperon failure case. The control is appropriate to track the desired maneuver until the completion of the long duration maneuver, when the aircraft is in a different attitude. At this point in this case, the MMAE/FS does not make the proper failure determination, causing the application of incorrect control by the CR. The cause of the jump in roll angle at the end of the simulation in Figure 48 is caused by the lack of proper control over the aircraft. The aircraft does not roll as fast with the partially failed left stabilator and the CR is not applying correct control at that time because the MMAE/FS is not declaring the left flaperon failure. Since the aircraft is not rolling as fast but the flight control system is being ‘told’ that all its actuator/surface commands are being followed, the Block 40 Flight Control System (FCS) is alerted to the lack of proper control through the MMAE estimates of the sensor measurements. Detecting that the aircraft is not rolling as fast as commanded, the Block 40 FCS increases the gain to the flaperons, causing the aircraft roll angle to jump momentarily.

5.4.5 Control Redistribution Performance with Rudder Failures

Figures 53 through 56 represent the CR attempt to control the aircraft given various rudder actuator failures. The aircraft does not have enough control authority in the fully functional flight control actuators to compensate for the failed rudder. The CR applies maximum control to the other flight control surfaces, resulting in large deviations from the desired roll maneuver. This maneuvering also affects the MMAE/FS because the rapid maneuvering caused by the saturated stabilator and flaperon channels adds unmodeled dynamics to the aircraft, violating the linear model approximations of the failure hypotheses elemental filters.

5.5 MMAE/FS/CR Combined Performance

5.5.1 Combined MMAE/FS/CR

The performance of the complete MMAE/FS/CR is presented here. In the previous sections, the effectiveness value, ε, is treated as a known value in order to present the performance of the MMAE/FS and the CR separately. In this section, the ε value estimated by the MMAE/FS is directly used in the CR. Based on the analysis between ε_{map} and ε_{hat} presented in Section 5.3.5, the more accurate estimate of ε_{hat} is used. Although $\varepsilon_{\mathrm{map}}$ was found to be preferable for the case of no maneuvering with dither inputs, $\varepsilon_{\mathrm{hat}}$ appears to be more suitable for the case of a maneuvering aircraft without purposeful, additional dithering inputs. The elemental probability and state value plots are presented in the same order as above for actuator failures. Sensor failures are not presented for this case since they are not affected by the estimate of effectiveness. The solid lines show the mean values in the elemental probability plots and show the mean performance of the CR in the state value plots. The dashed lines represent the mean ± one standard deviation in the elemental probability plots and shows the desired values of the fully functional aircraft in the state value plots.

Figures 57-76 on MMAE/FS/CR combined performance are now presented as a group for convienence. Then, starting on page 214, they will be discussed in detail.

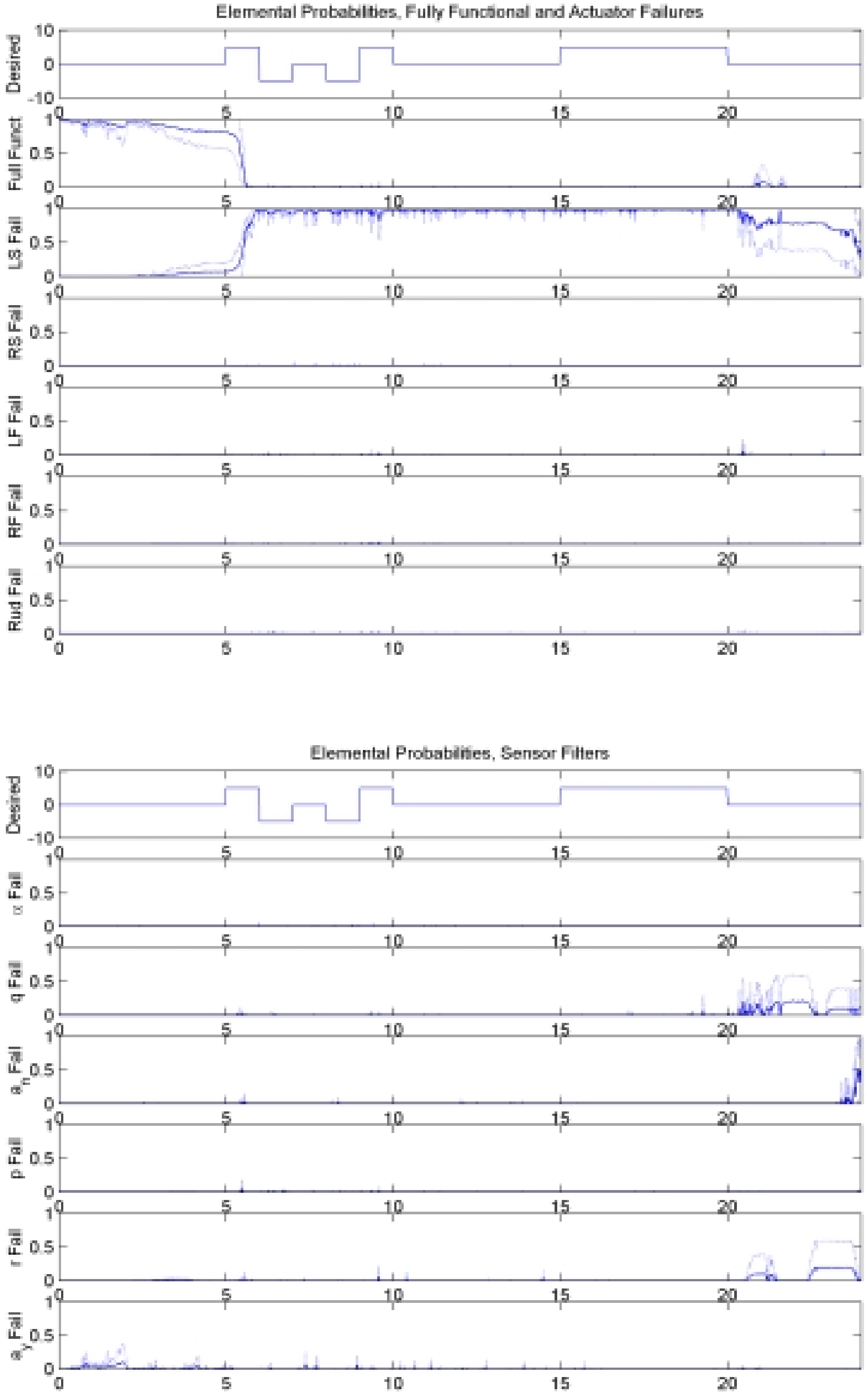


Figure 57a: Elemental Probabilities, Left Stabilator Failure, ε = 0% Using ε_{HAT}

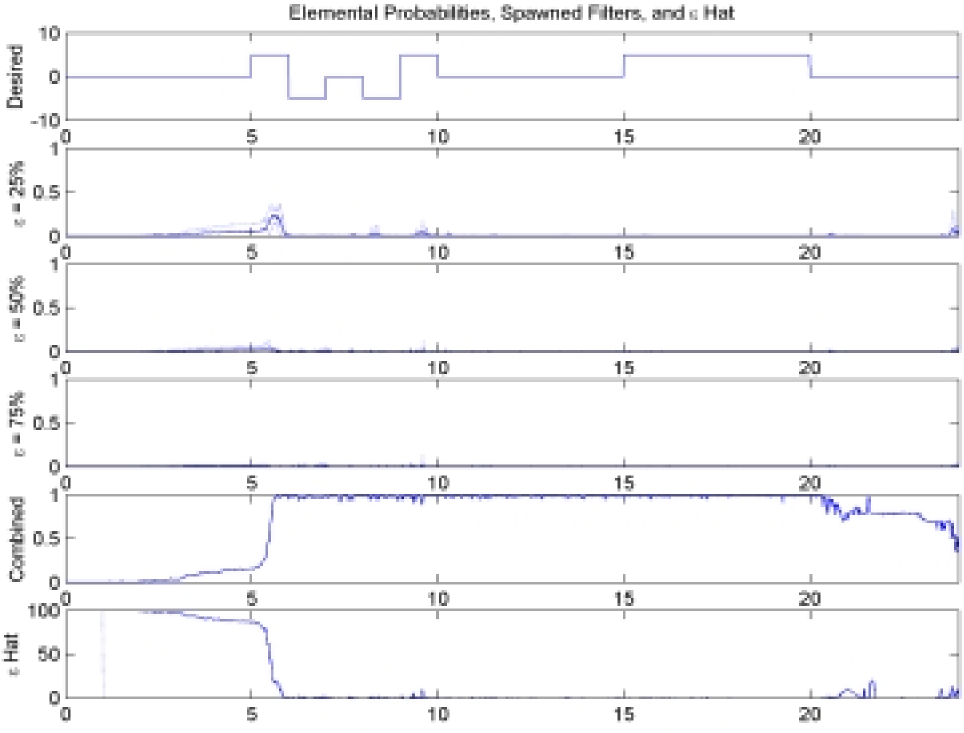


Figure 57b: Elemental Probabilities, Left Stabilator Failure, ε = 0% Using ε $_{\mathrm{HAT}}$

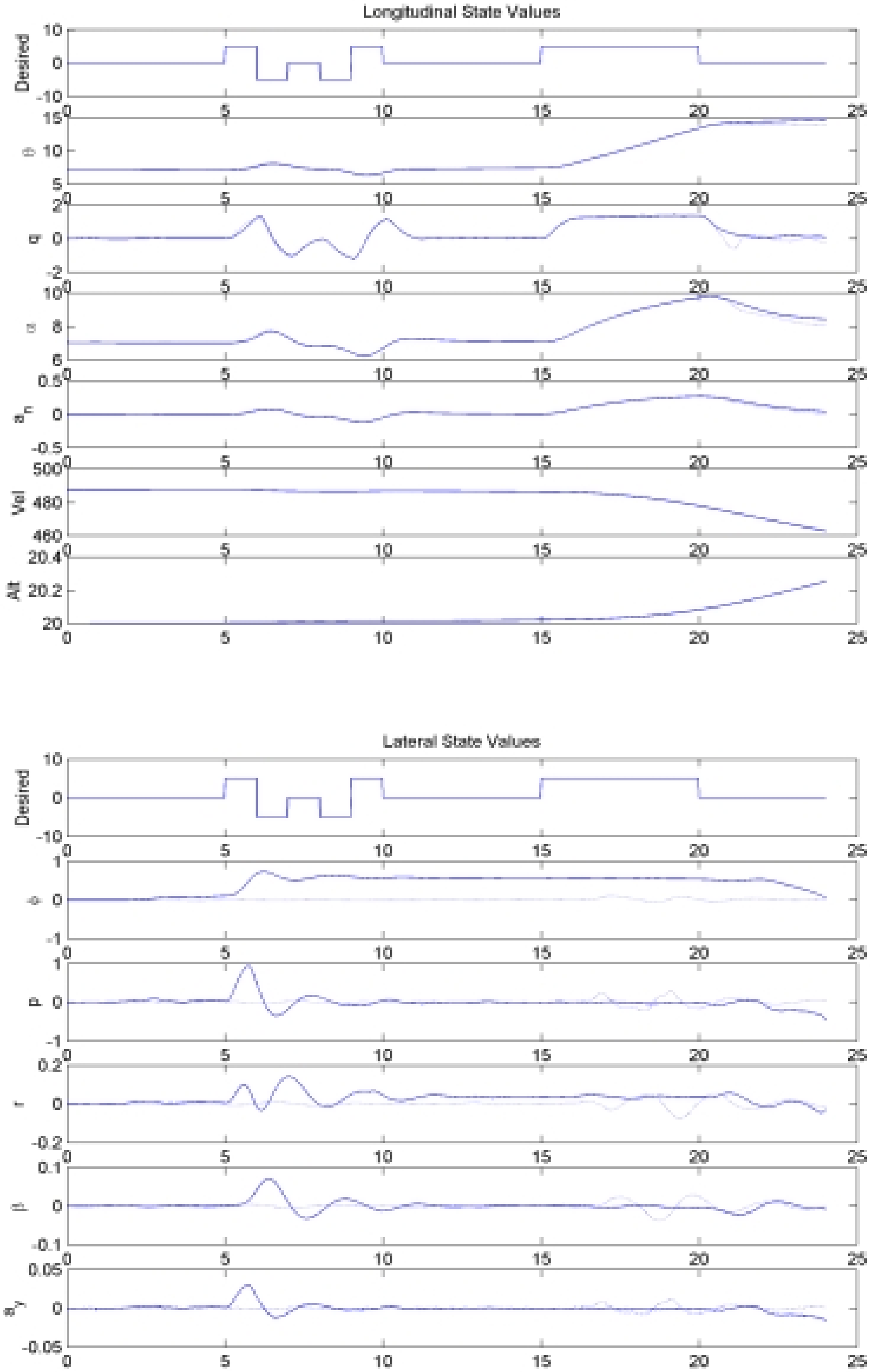


Figure 57c: State Values, Left Stabilator Failure, $\varepsilon = 0\%$ Using ε_{HAT}

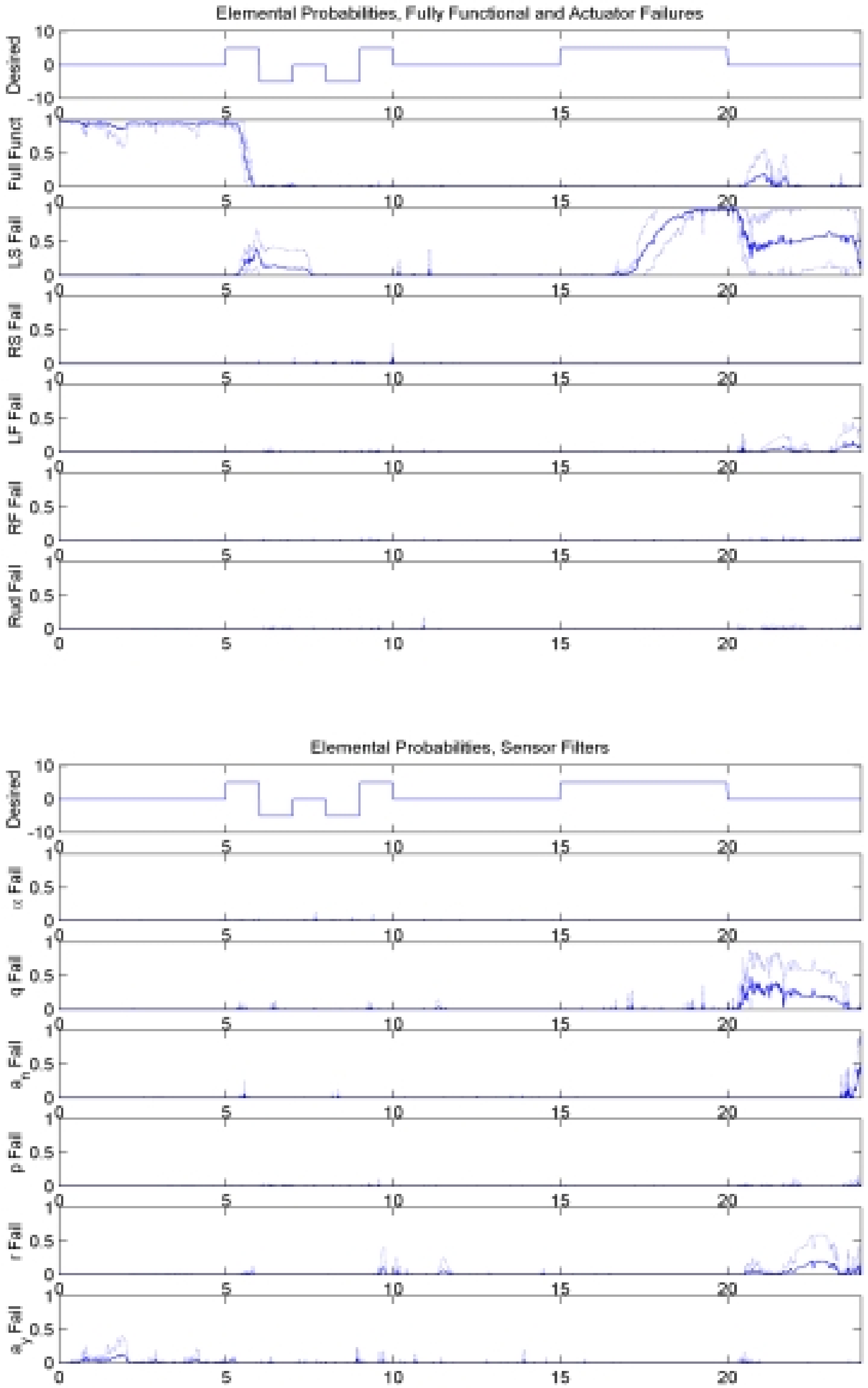


Figure 58a: Elemental Probabilities, Left Stabilator Failure, ε = 25% Using ε_{HAT}

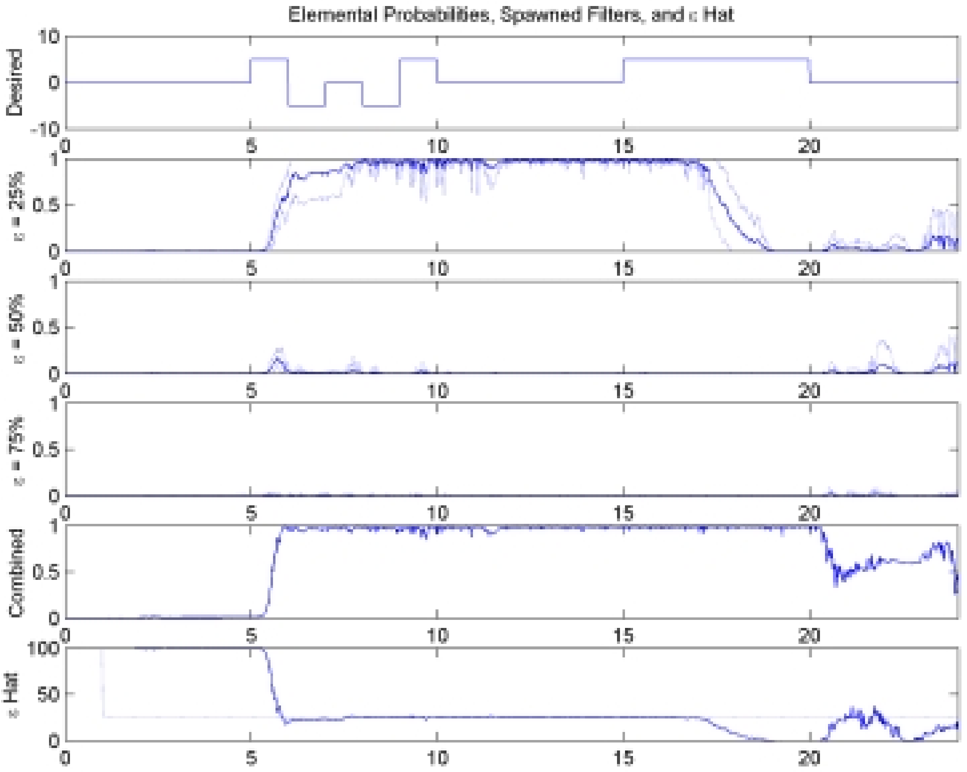


Figure 58b: Elemental Probabilities, Left Stabilator Failure, ε = 25% Using ε_{HAT}

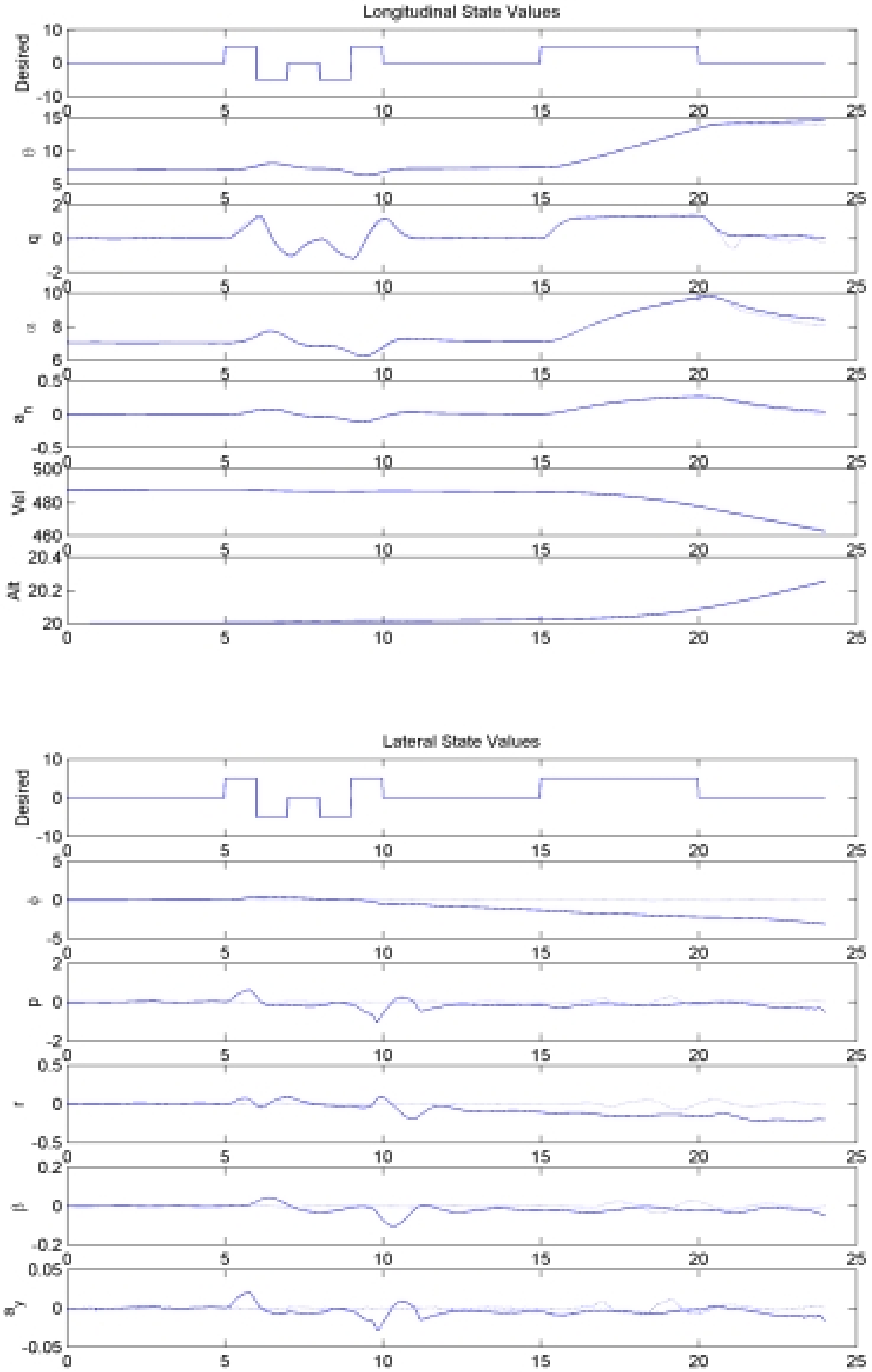


Figure 58c: State Values, Left Stabilator Failure, ε = 25% Using ε $_{\text{HAT}}$

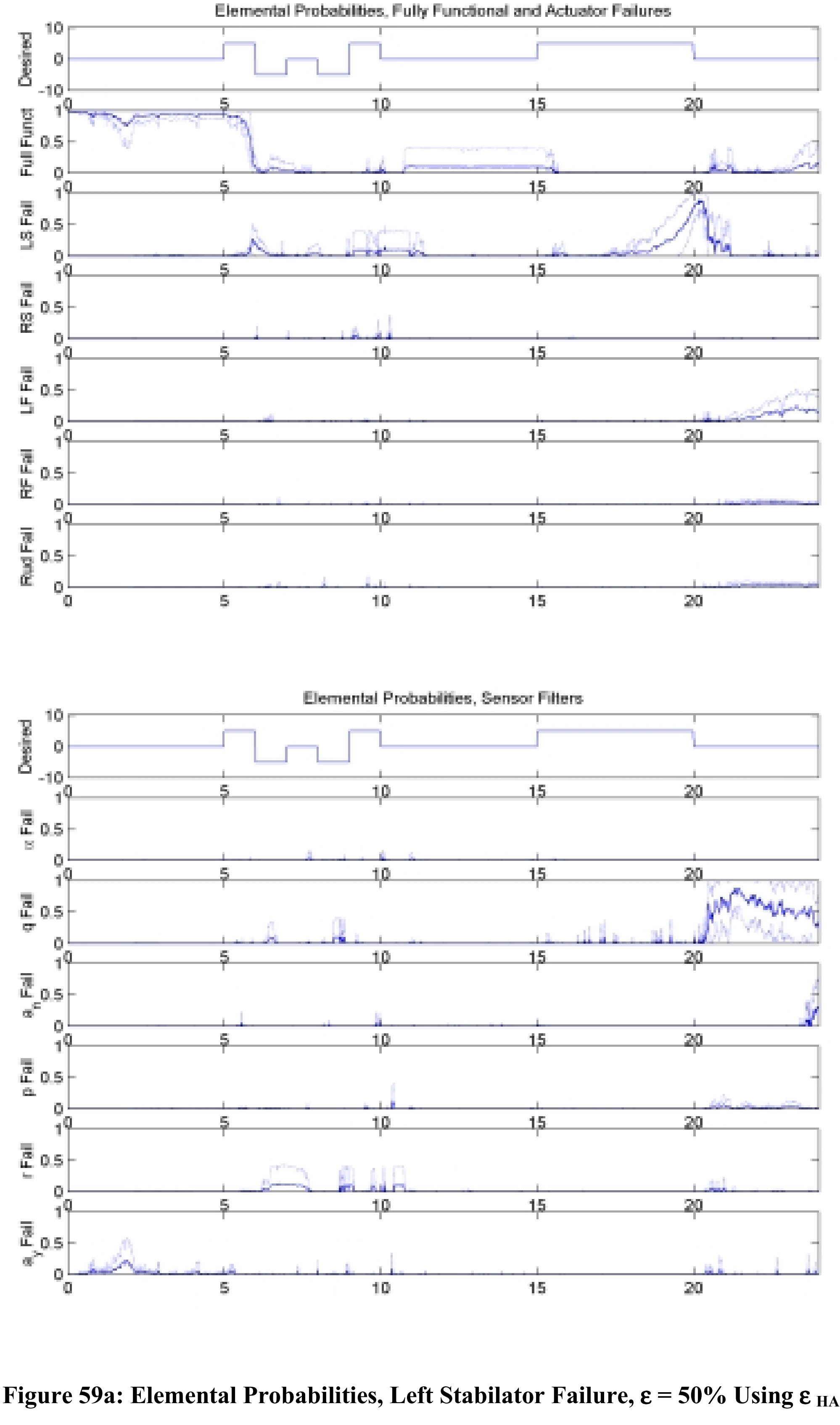


Figure 59a: Elemental Probabilities, Left Stabilator Failure, $\varepsilon = 50\%$ Using ε_{HAT}

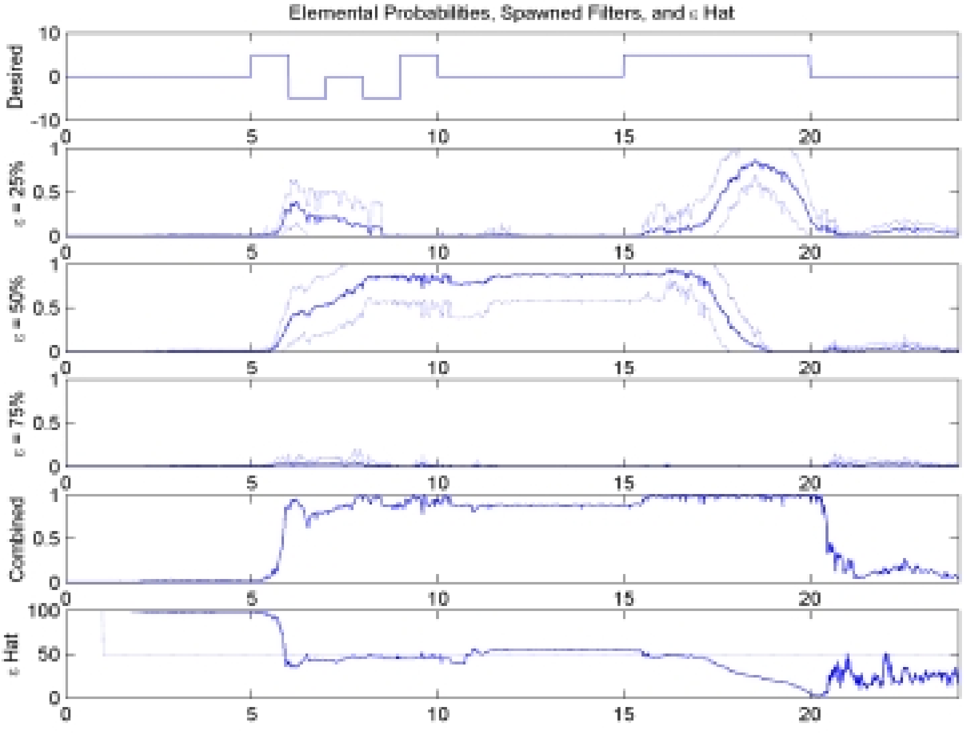
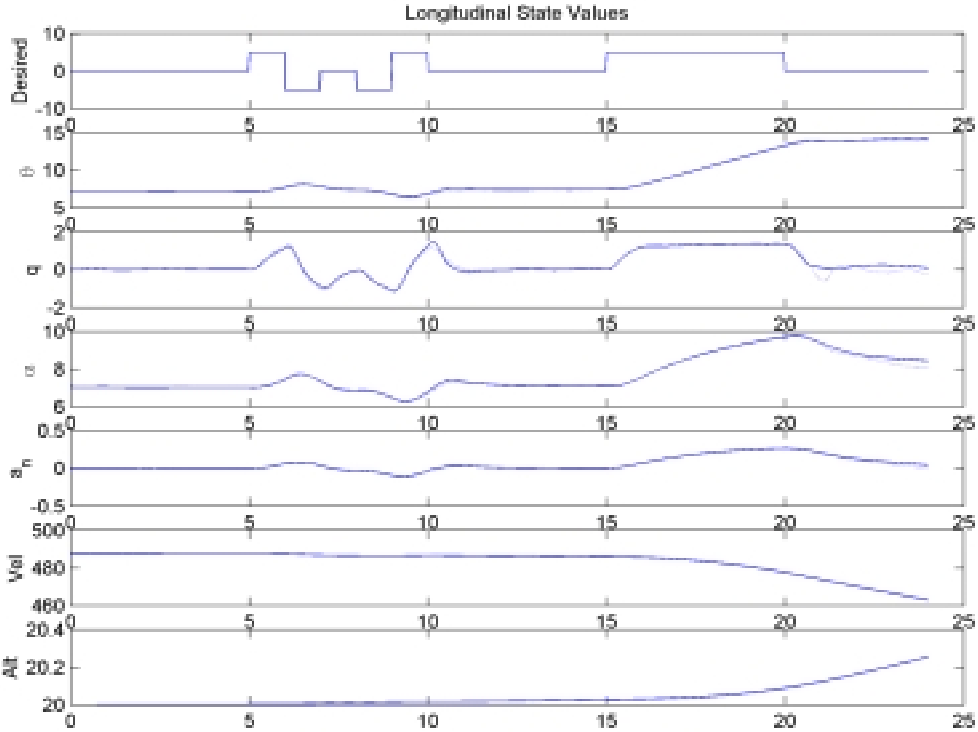


Figure 59b: Elemental Probabilities, Left Stabilator Failure, ε = 50% Using ε_{HAT}



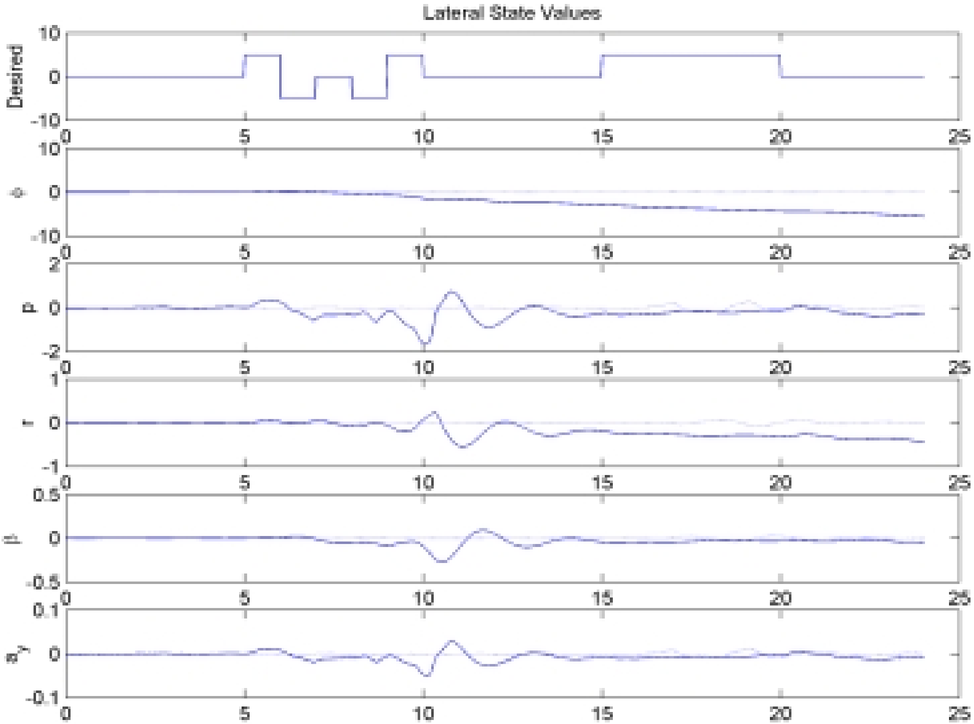


Figure 59c: State Values, Left Stabilator Failure, ε = 50% Using ε $_{HAT}$

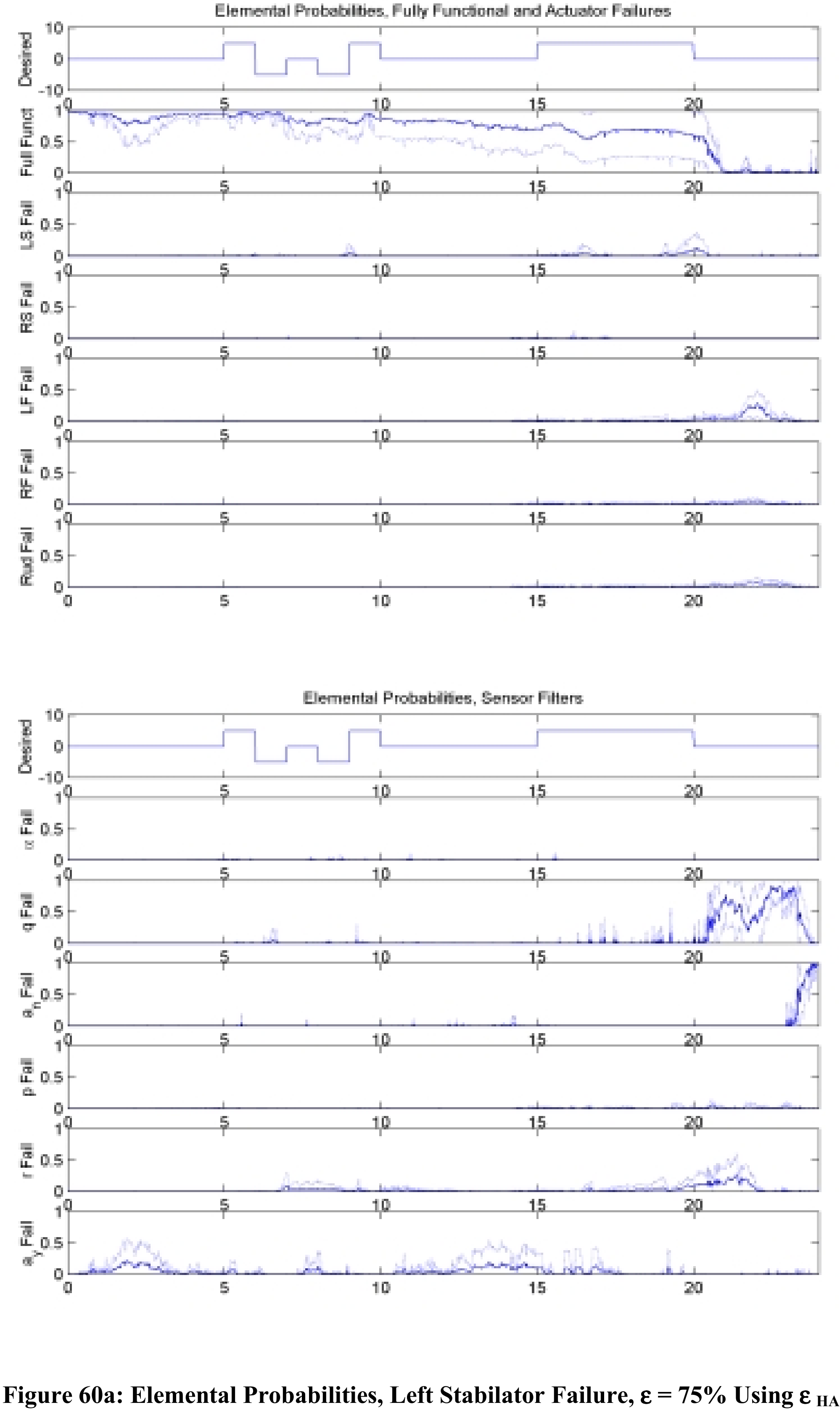


Figure 60a: Elemental Probabilities, Left Stabilator Failure, ε = 75% Using ε_{HAT}

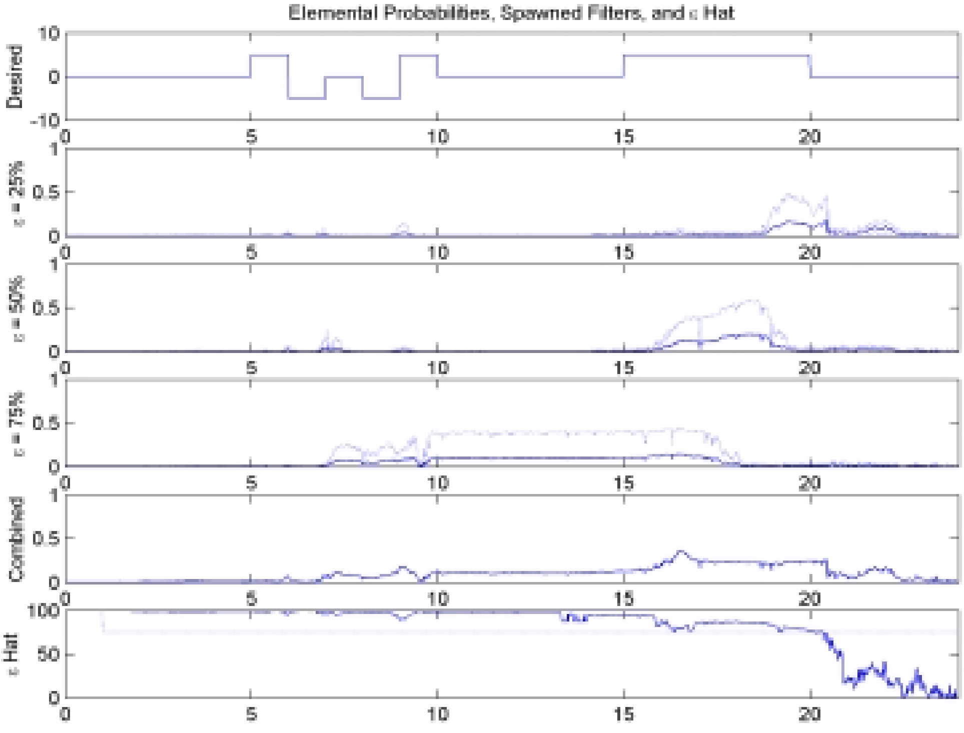


Figure 60b: Elemental Probabilities, Left Stabilator Failure, ε = 75% Using ε $_{\text{HAT}}$

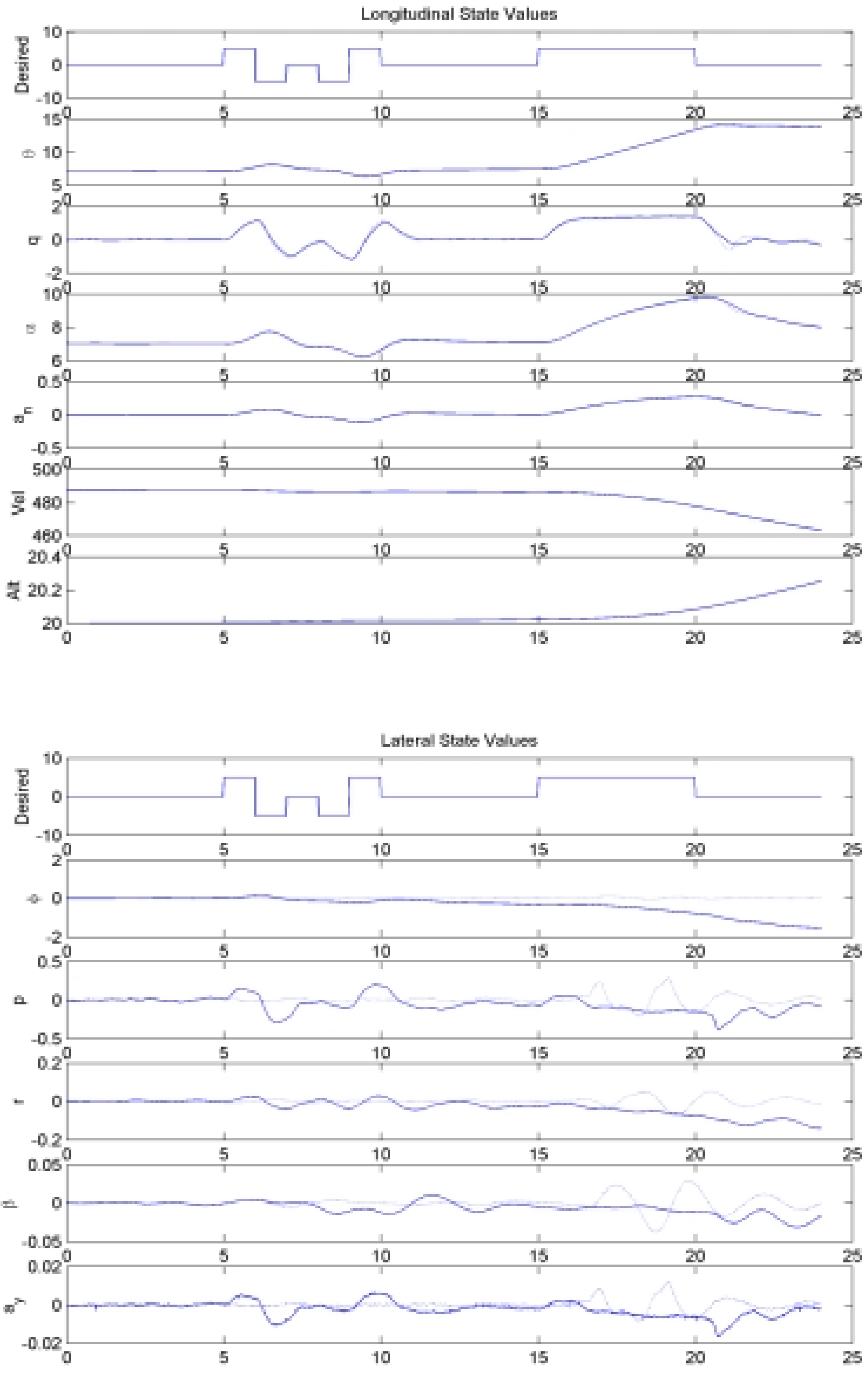


Figure 60c: State Values, Left Stabilator Failure, ε = 75% Using ε $_{HAT}$

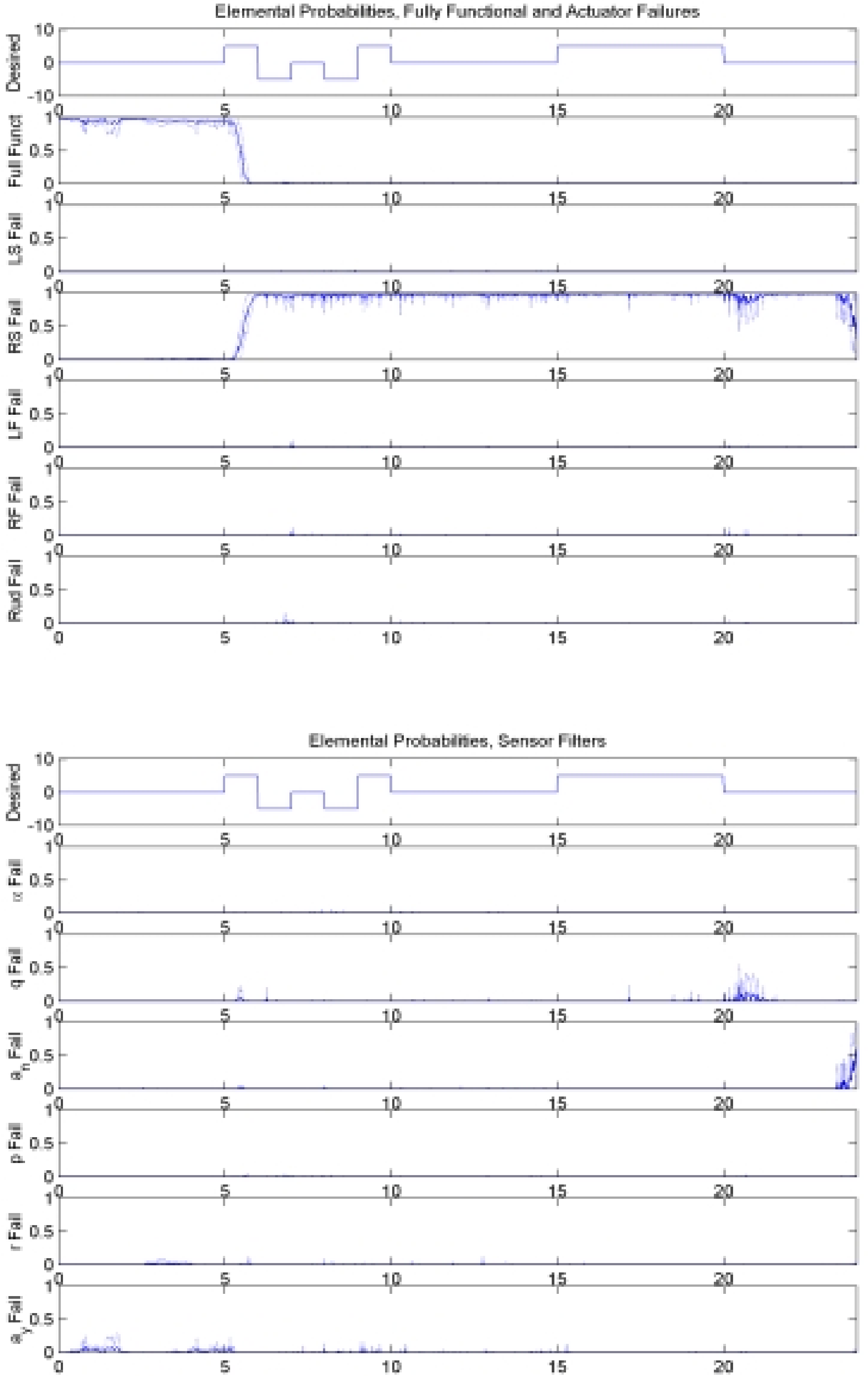


Figure 61a: Elemental Probabilities, Right Stabilator Failure, $\varepsilon = 0\%$ Using ε_{HAT}

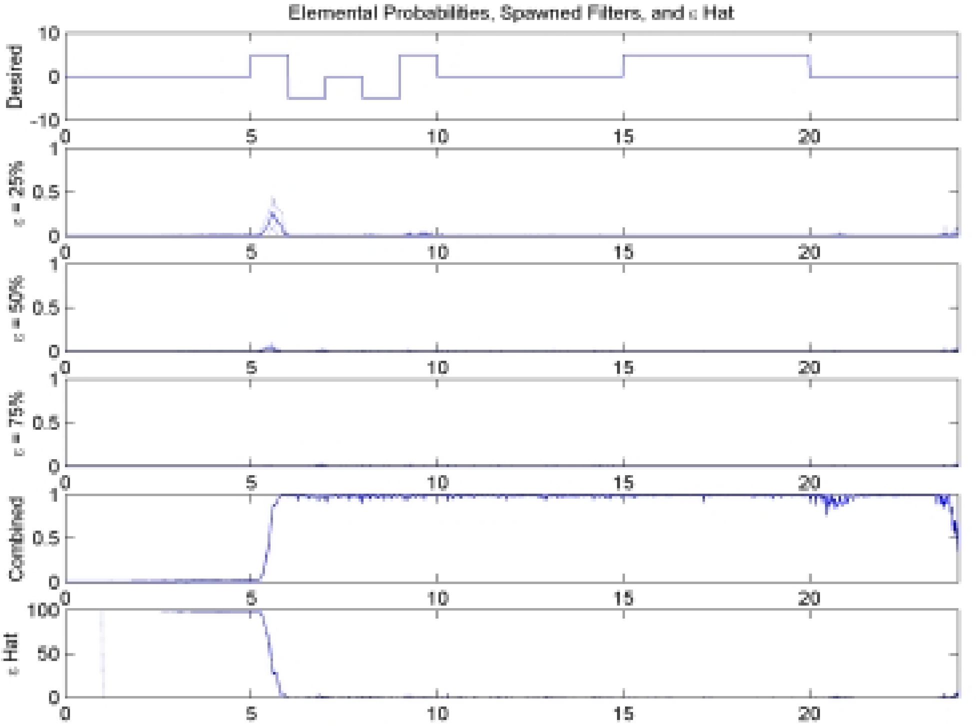


Figure 61b: Elemental Probabilities, Right Stabilator Failure, ε = 0% Using ε_{HAT}

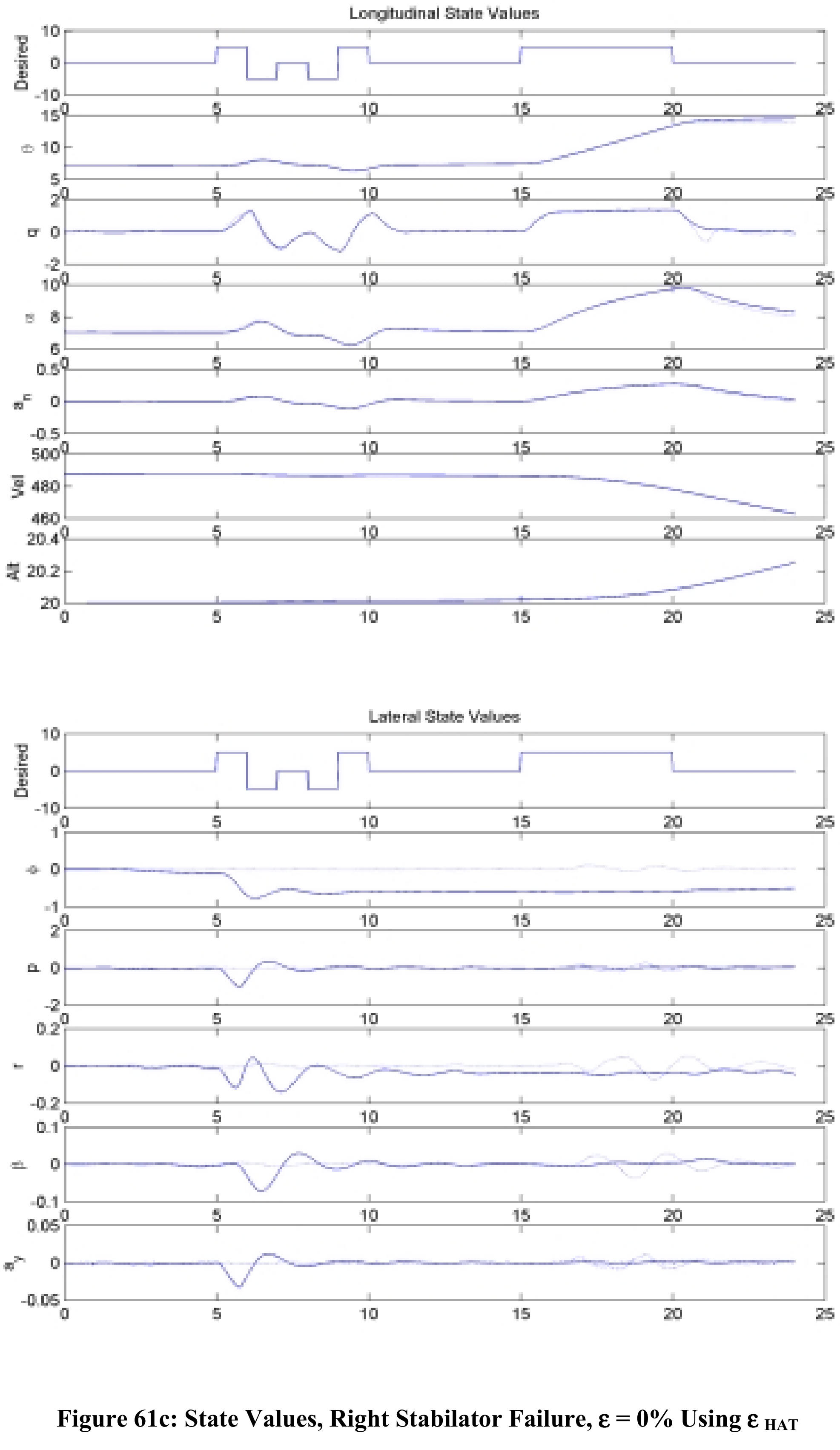


Figure 61c: State Values, Right Stabilator Failure, ε = 0% Using ε $_{\mathrm{HAT}}$

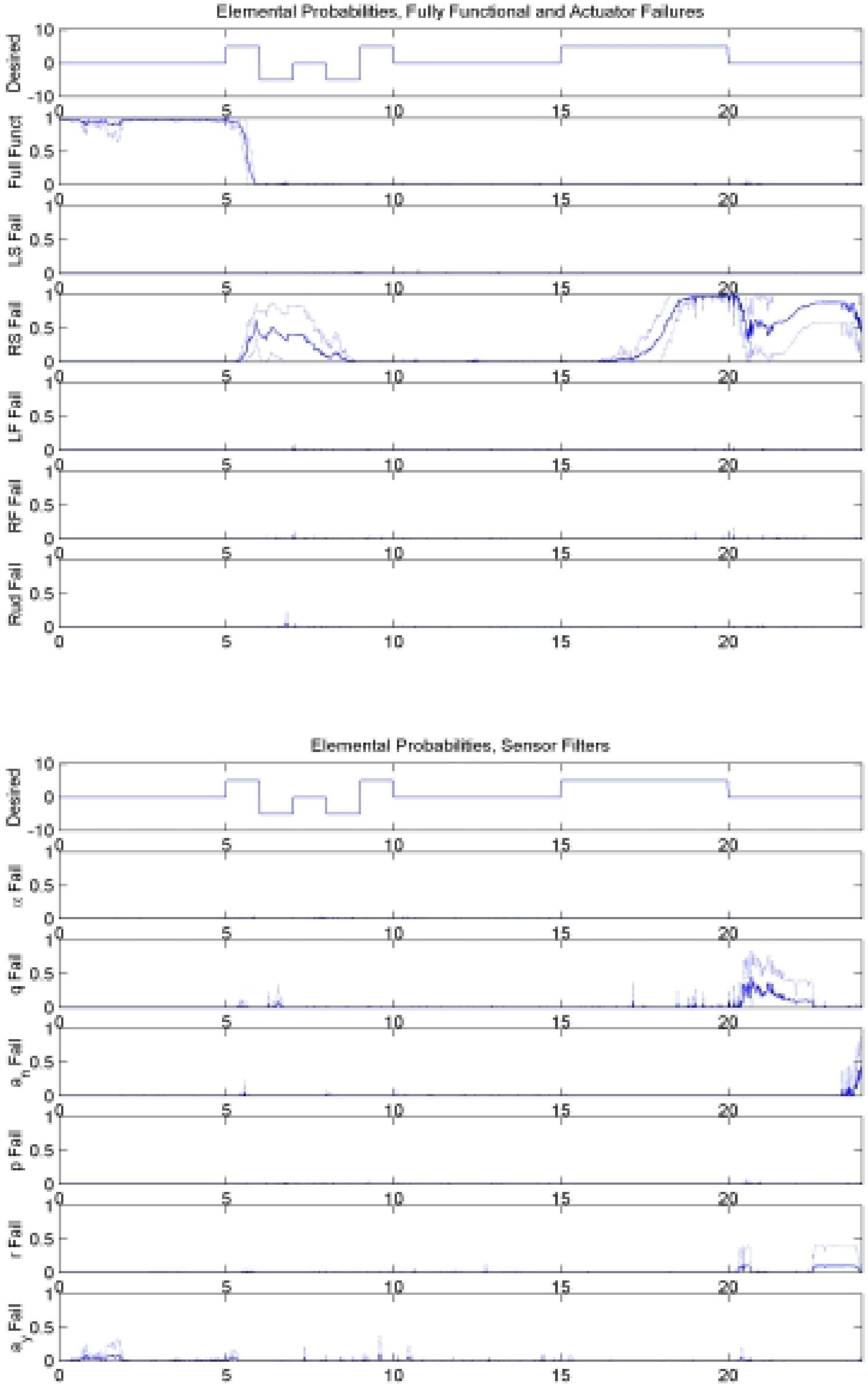


Figure 62a: Elemental Probabilities, Right Stabilator Failure, ε = 25% Using ε_{HAT}

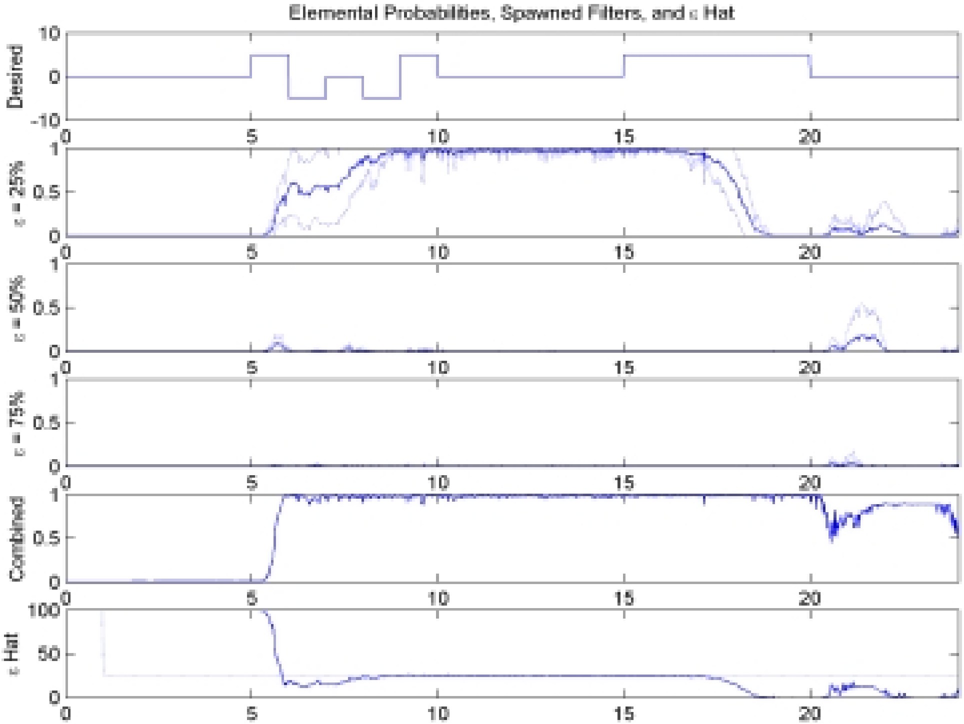


Figure 62b: Elemental Probabilities, Right Stabilator Failure, ε = 25% Using ε $_{\text{HAT}}$

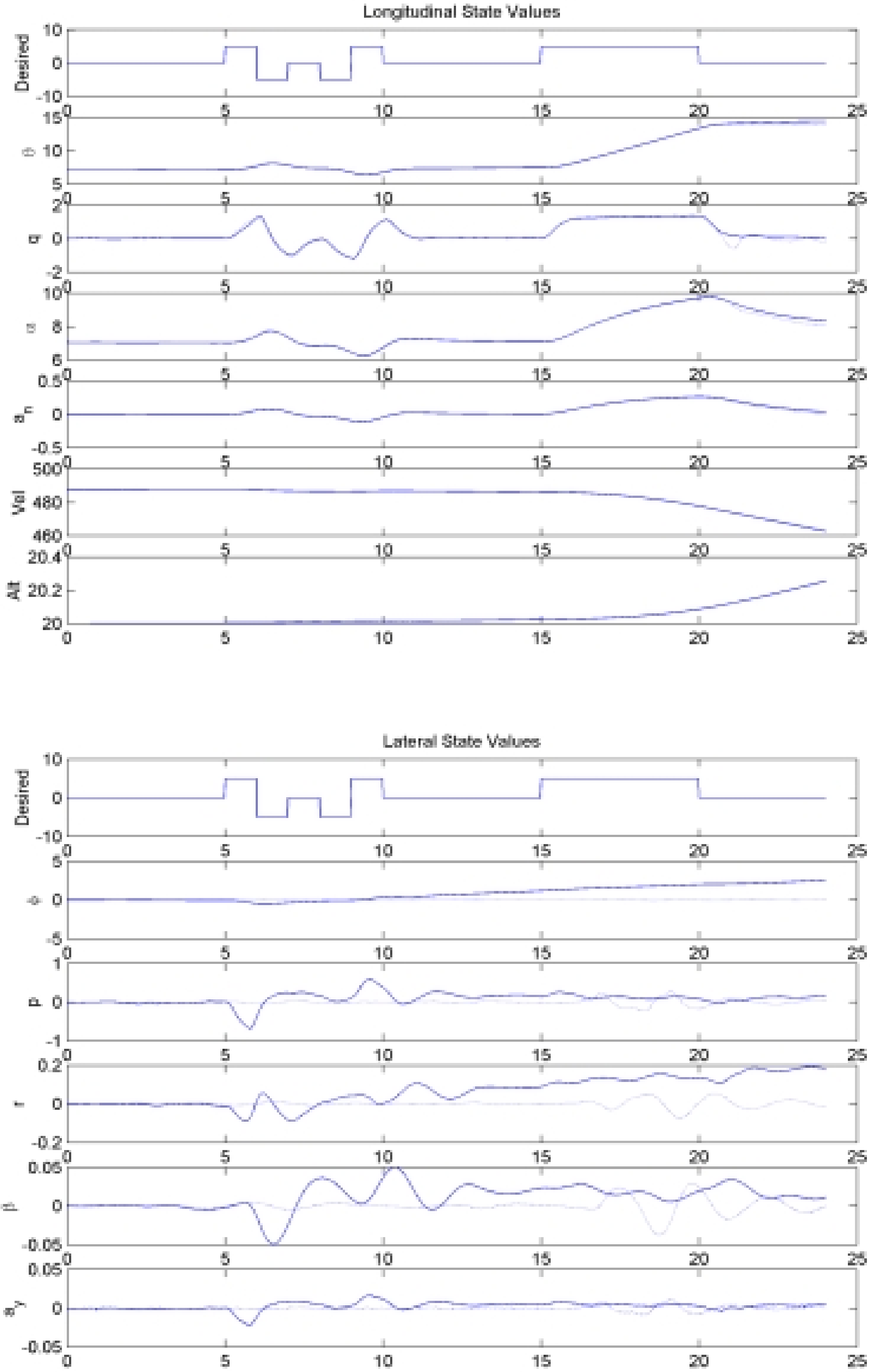
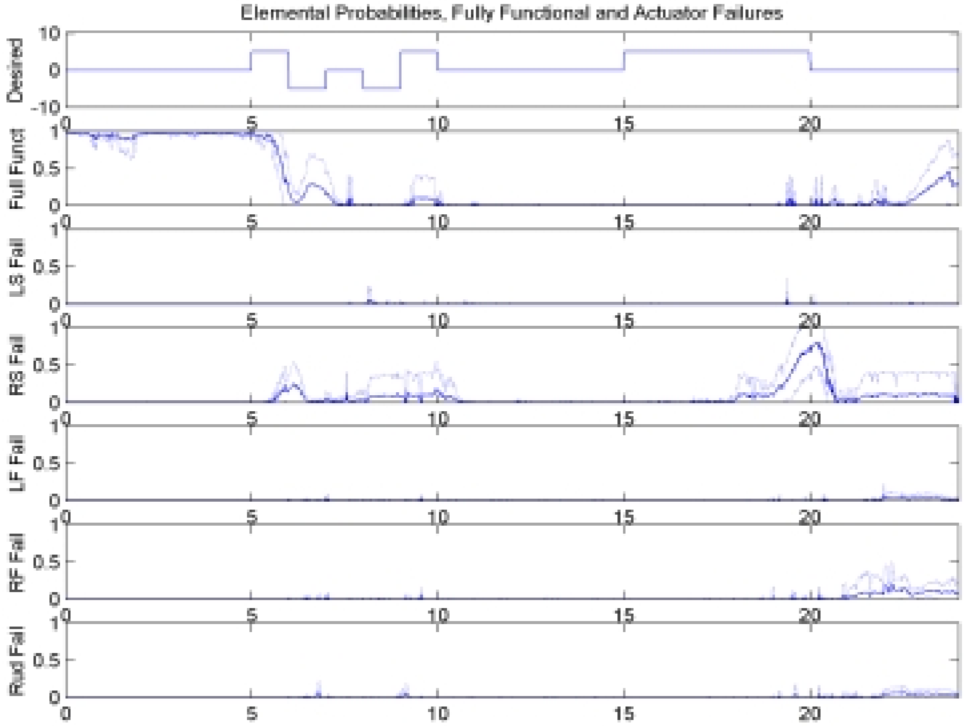


Figure 62c: State Values, Right Stabilator Failure, $\varepsilon = 25\%$ Using ε_{HAT}



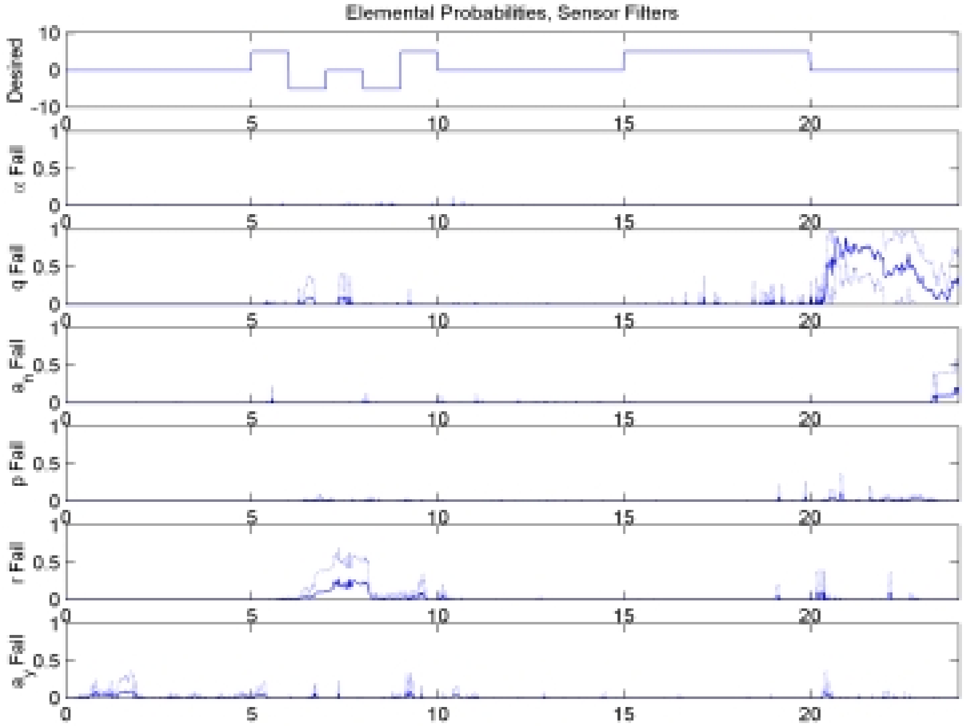


Figure 63a: Elemental Probabilities, Right Stabilator Failure, ε = 50% Using ε_{HAT}

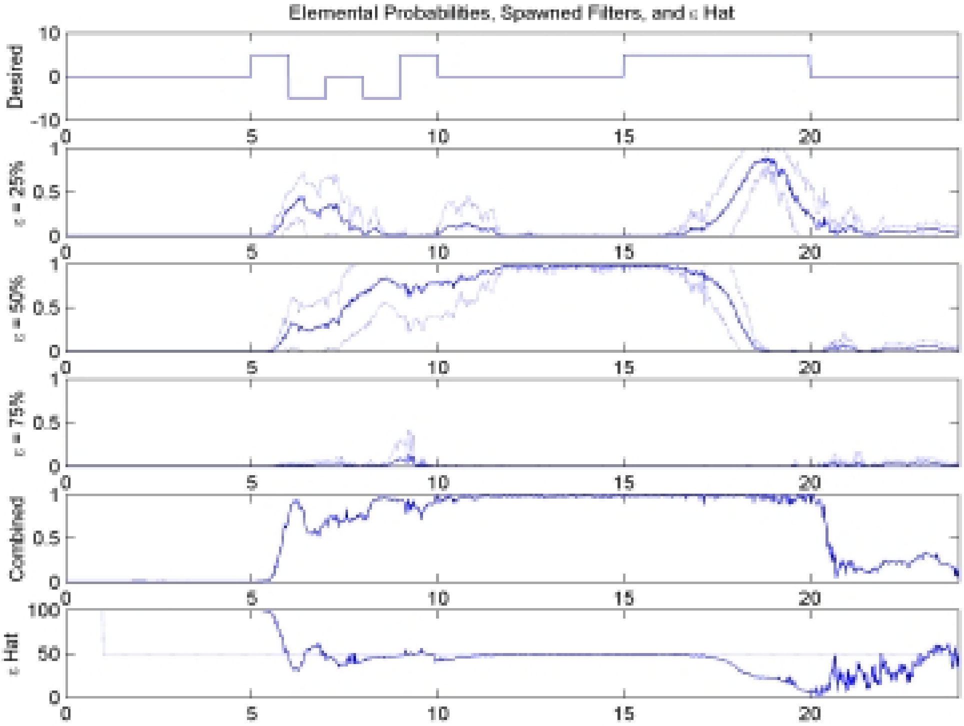


Figure 63b: Elemental Probabilities, Right Stabilator Failure, ε = 50% Using ε $_{\text{HAT}}$

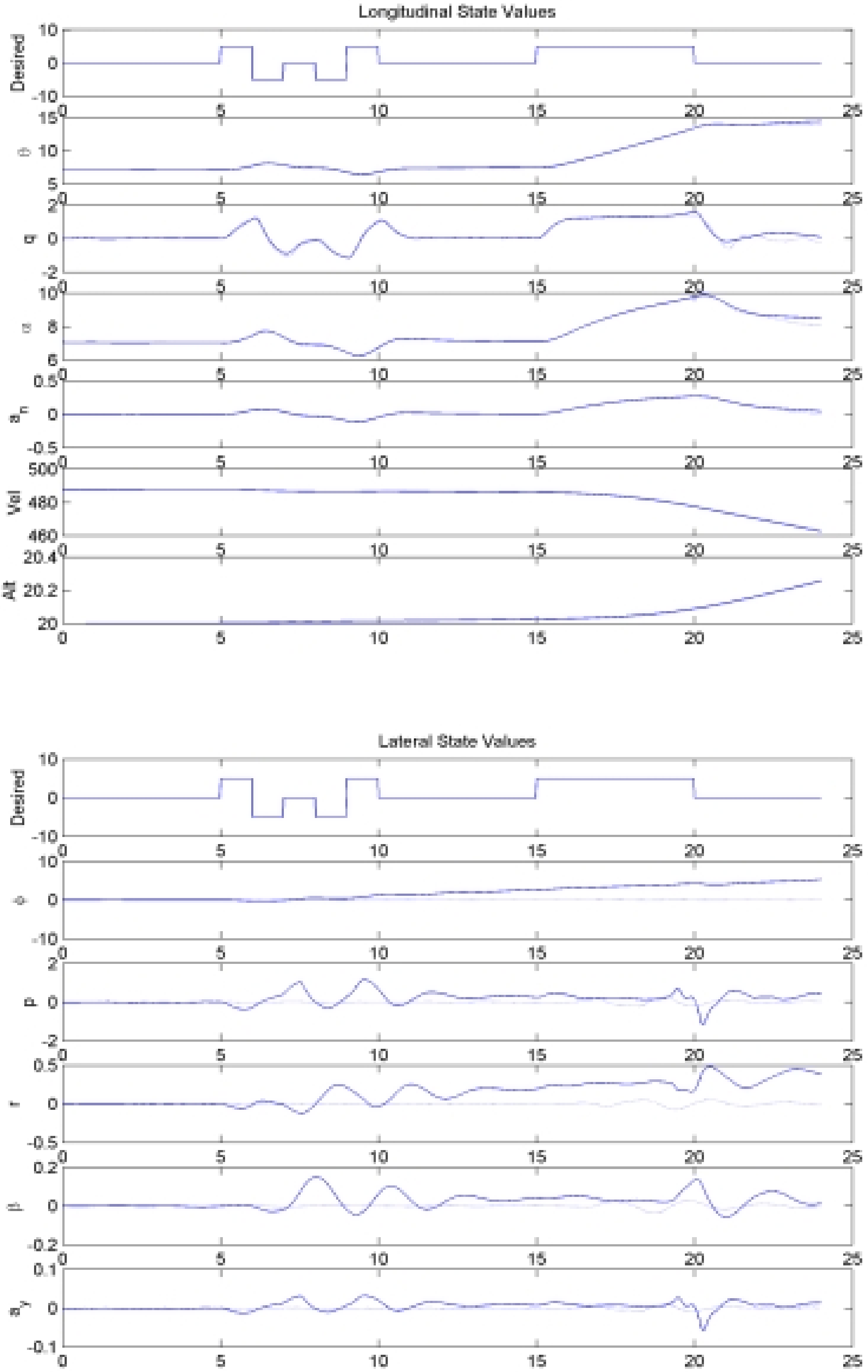


Figure 63c: State Values, Right Stabilator Failure, ε = 50% Using ε $_{HAT}$

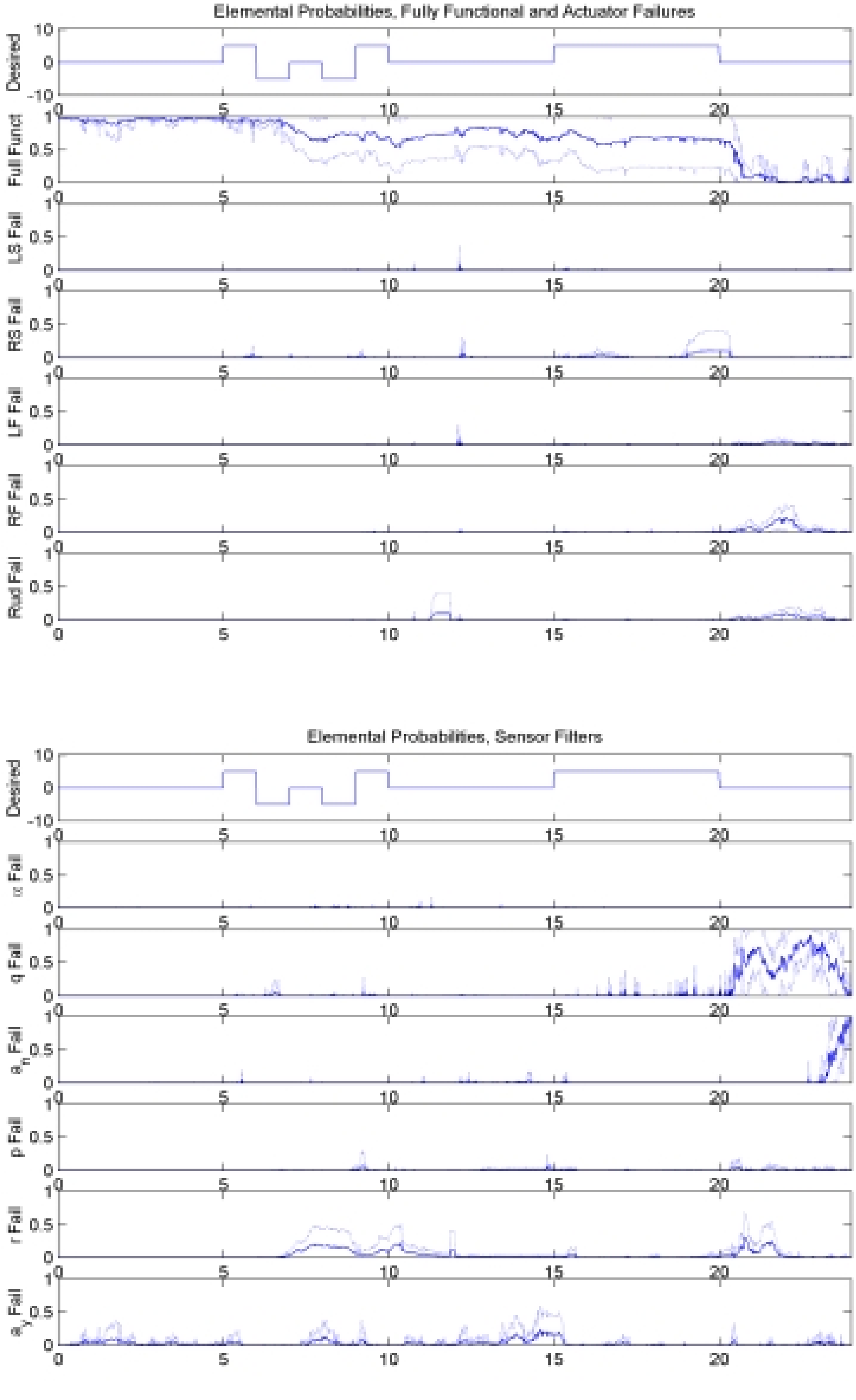


Figure 64a: Elemental Probabilities, Right Stabilator Failure, ε = 75% Using ε_{HAT}

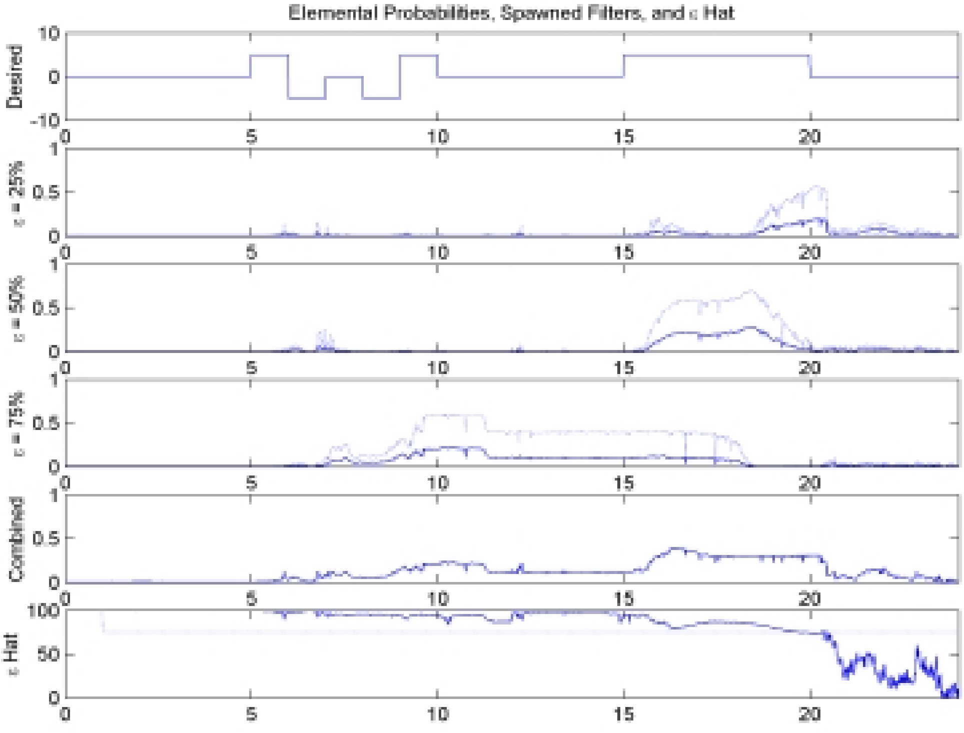


Figure 64b: Elemental Probabilities, Right Stabilator Failure, ε = 75% Using ε $_{\text{HAT}}$

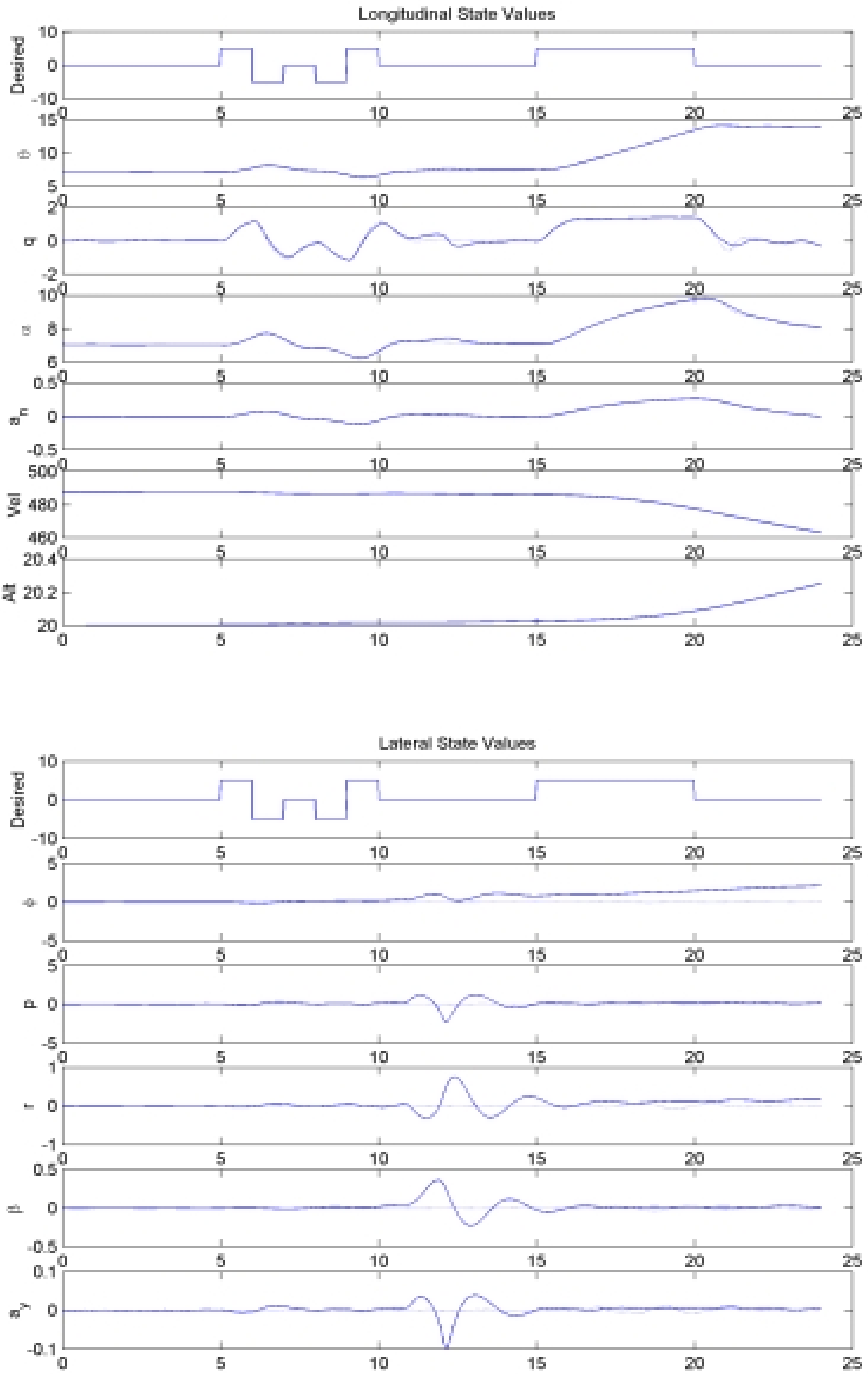
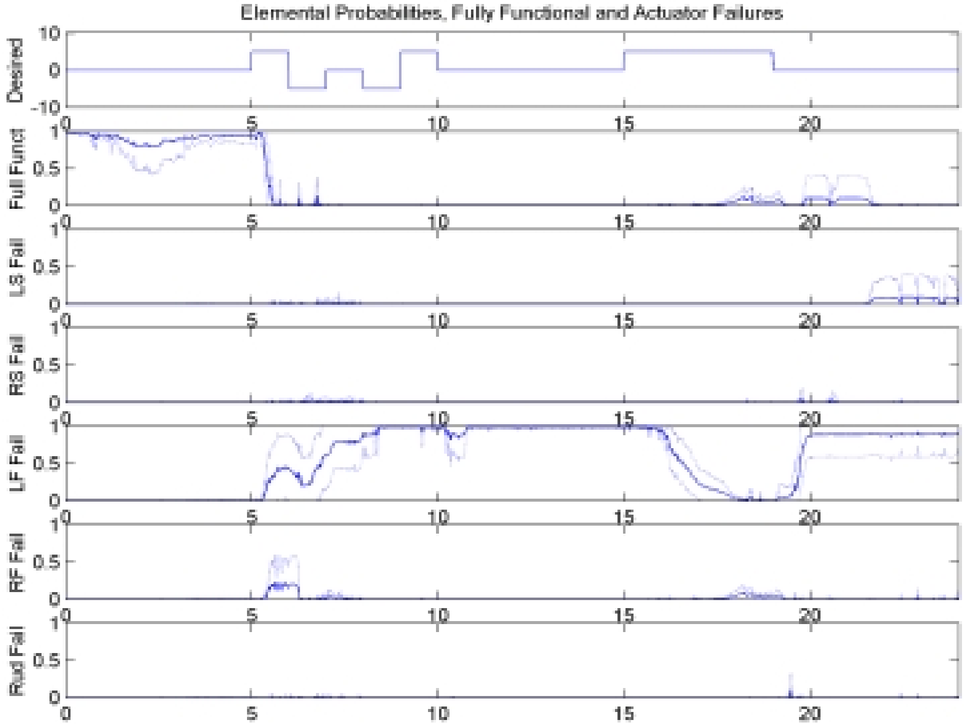


Figure 64c: State Values, Right Stabilator Failure, ε = 75% Using ε $_{HAT}$


Elemental Probabilities, Fully Functional and Actuator Failures

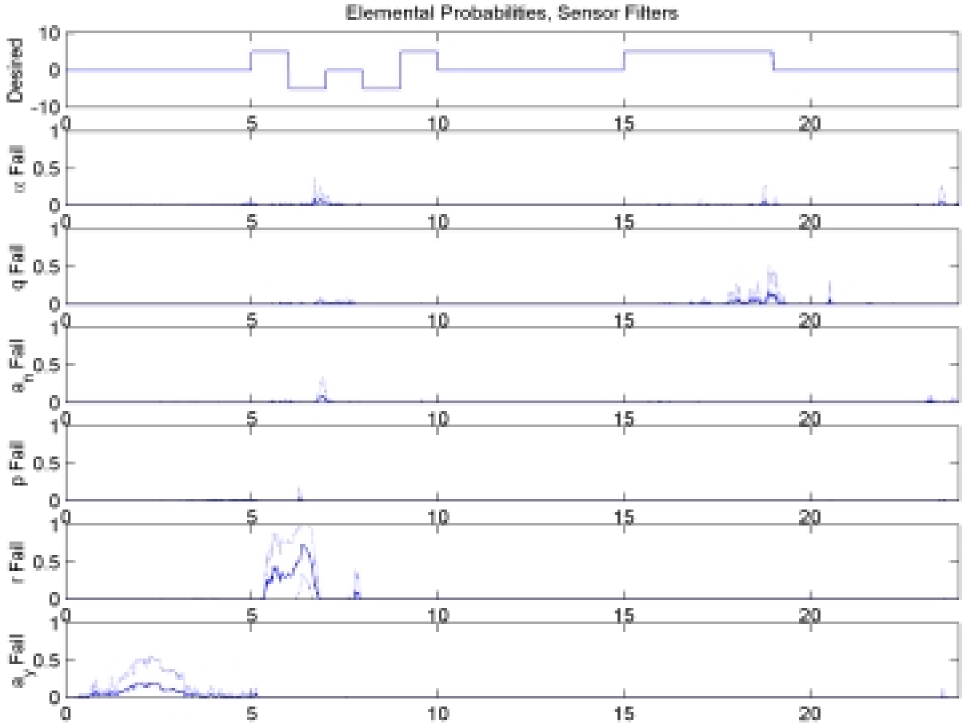

Elemental Probabilities, Sensor Filters

Figure 65a: Elemental Probabilities, Left Flaperon Failure, ε = 0% Using ε $_{\text{HAT}}$

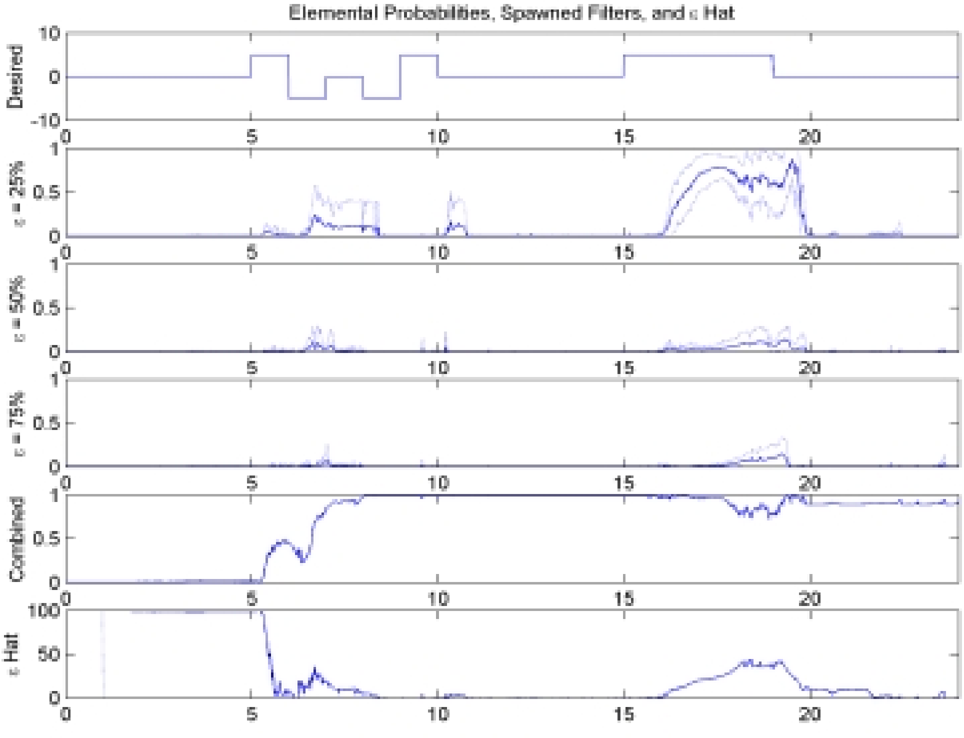


Figure 65b: Elemental Probabilities, Left Flaperon Failure, ε = 0% Using ε_{HAT}

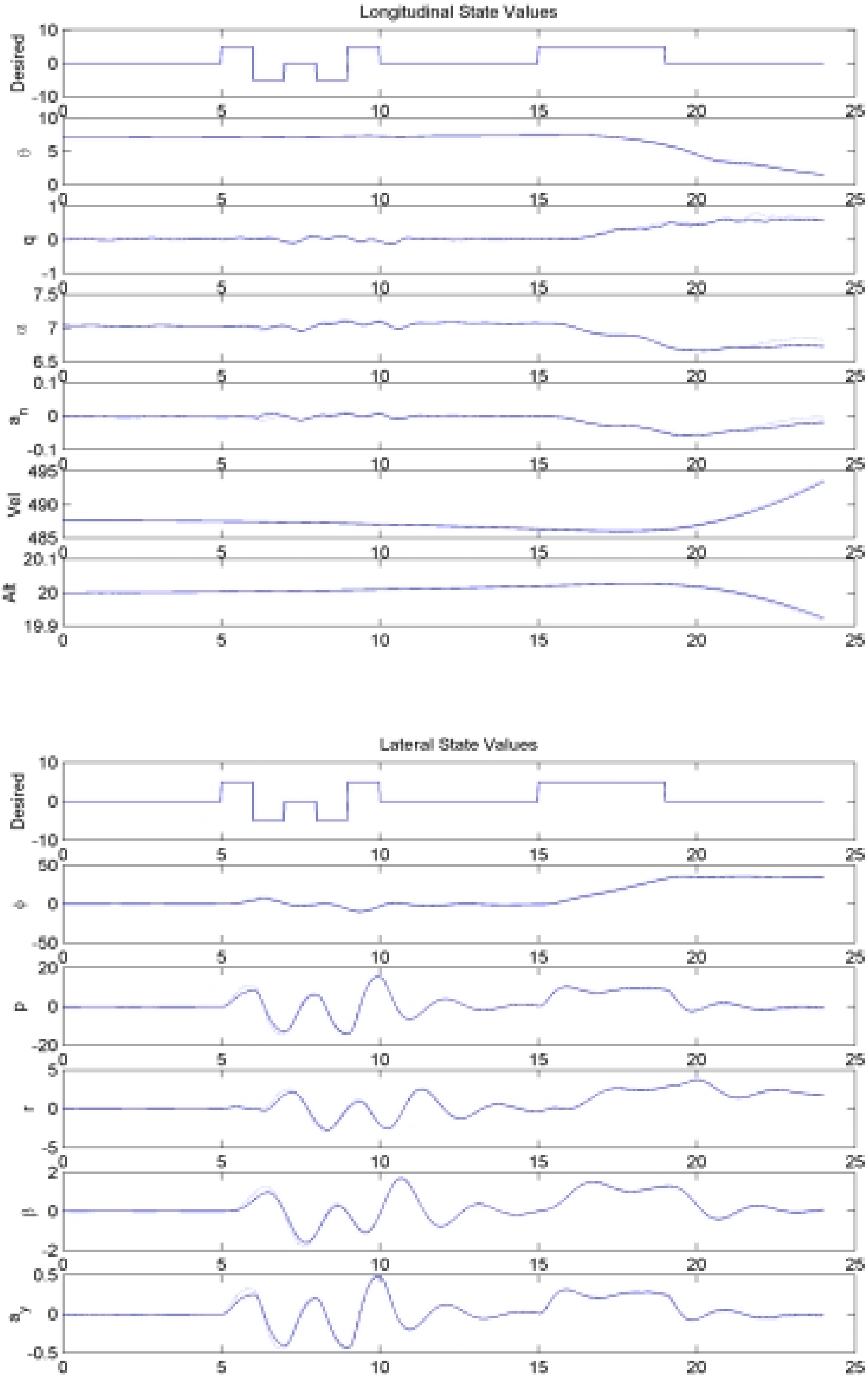
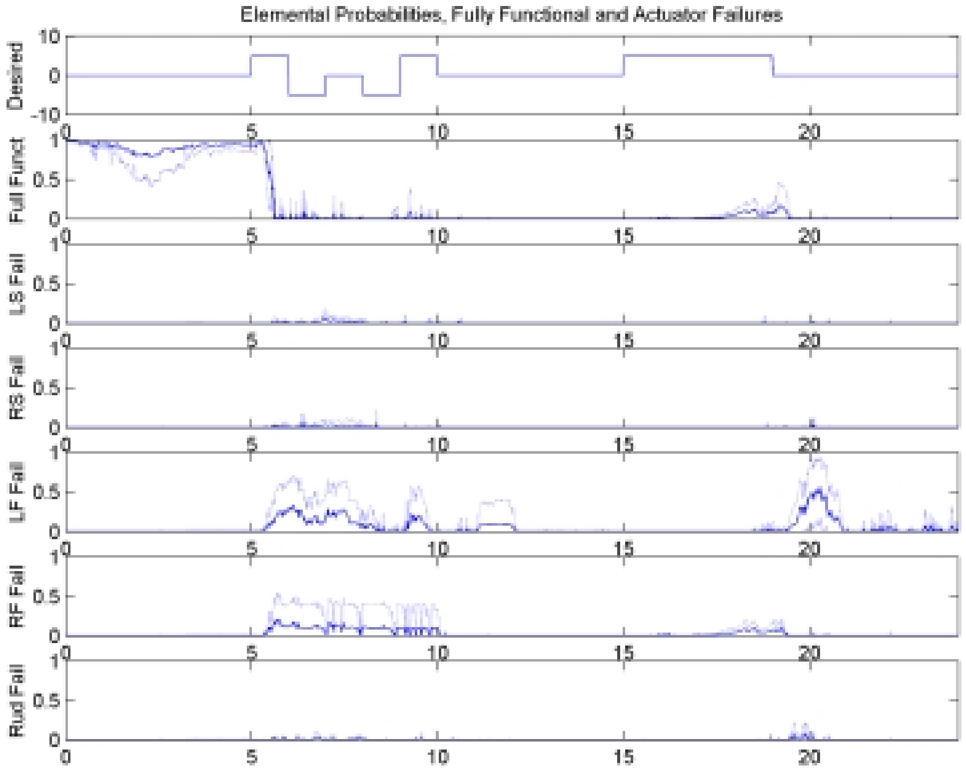


Figure 65c: State Values, Left Flaperon Failure, ε = 0% Using ε_{HAT}



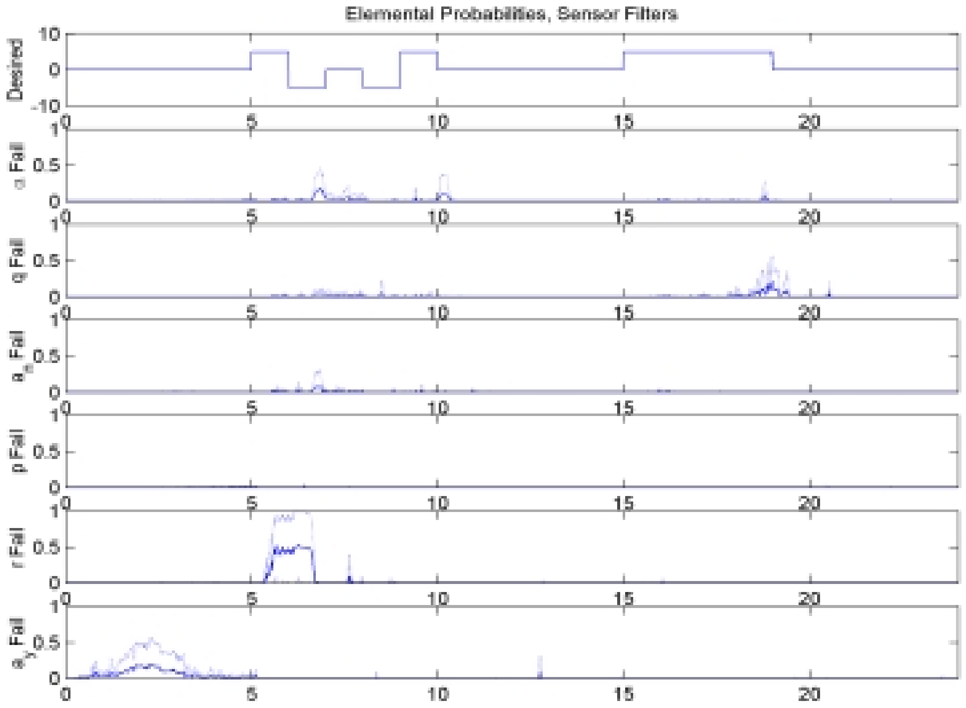


Figure 66a: Elemental Probabilities, Left Flaperon Failure, ε = 25% Using ε $_{\text{HAT}}$

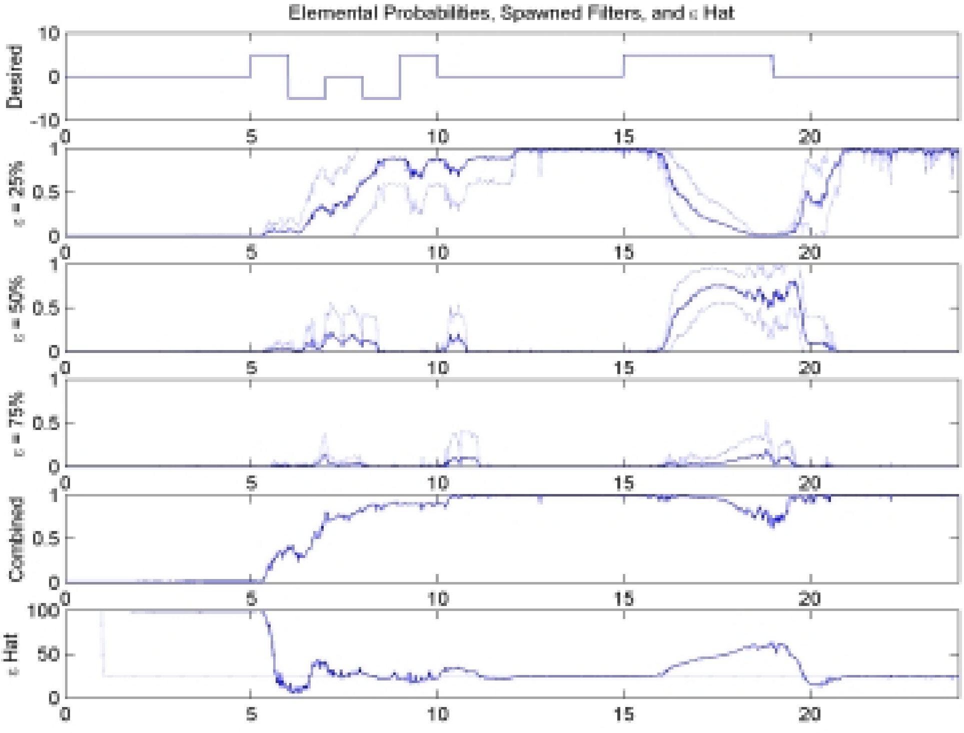


Figure 66b: Elemental Probabilities, Left Flaperon Failure, ε = 25% Using ε $_{HAT}$

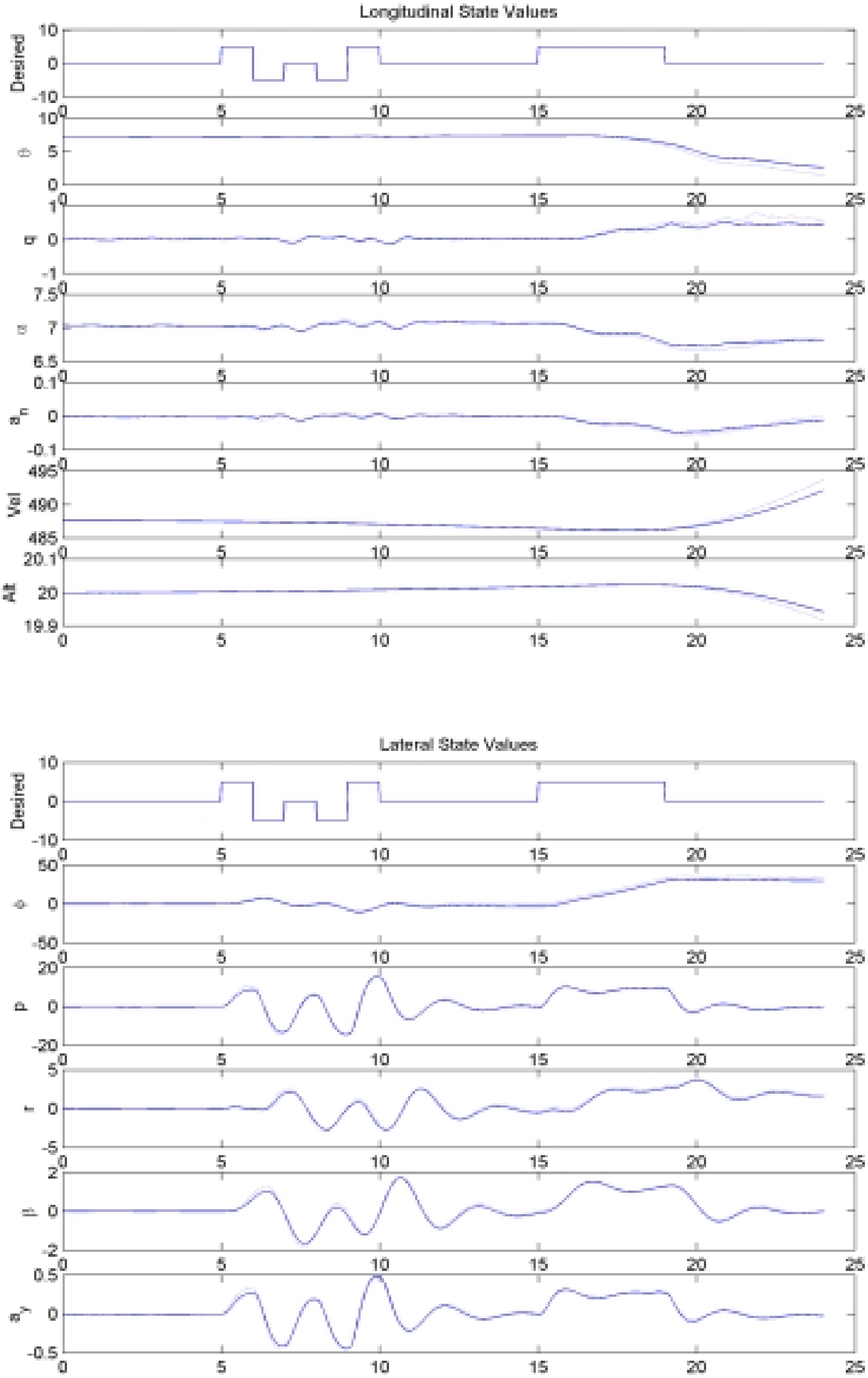


Figure 66c: State Values, Left Flaperon Failure, ε = 25% Using ε $_{HAT}$

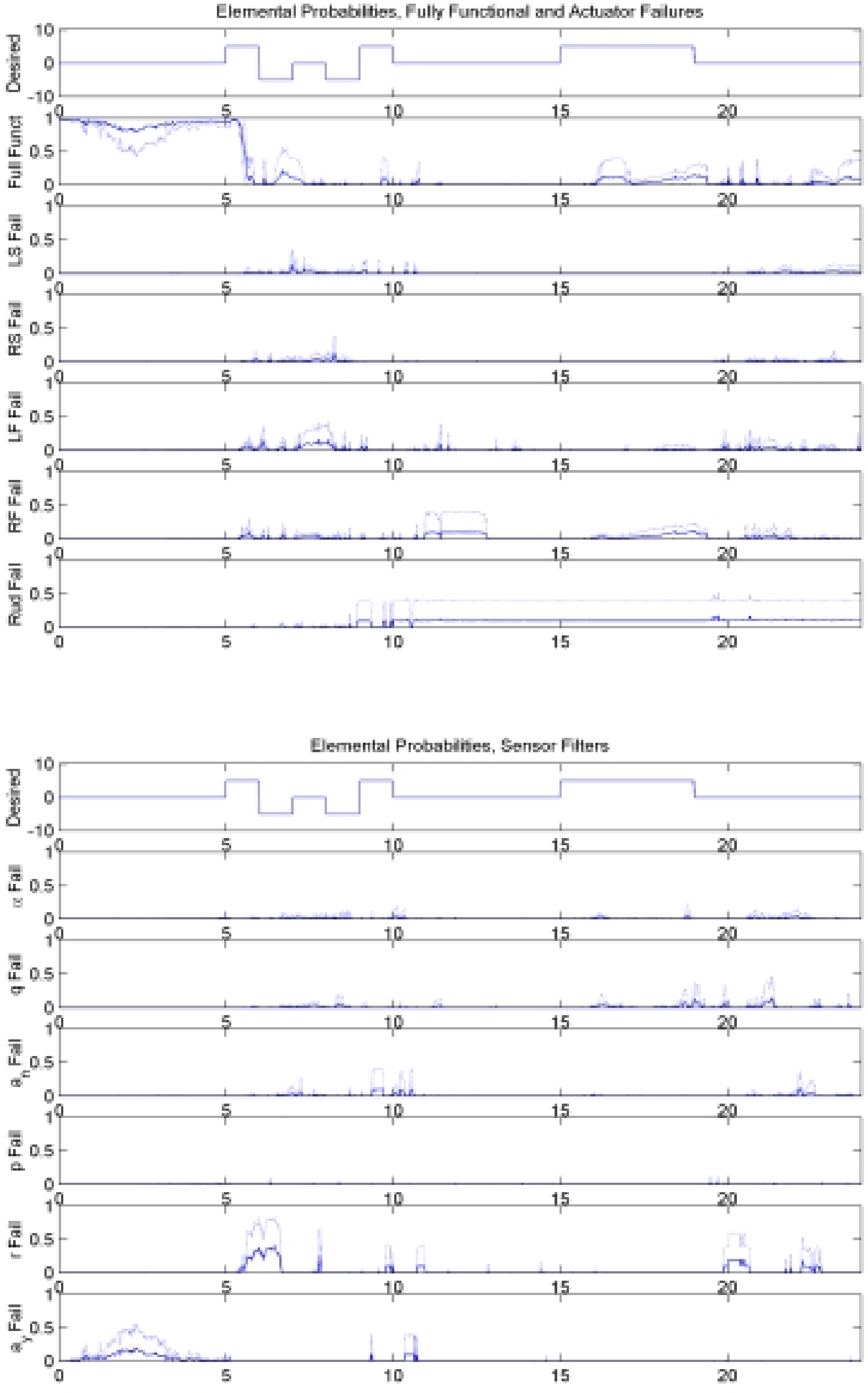


Figure 67a: Elemental Probabilities, Left Flaperon Failure, ε = 50% Using ε $_{\text{HAT}}$

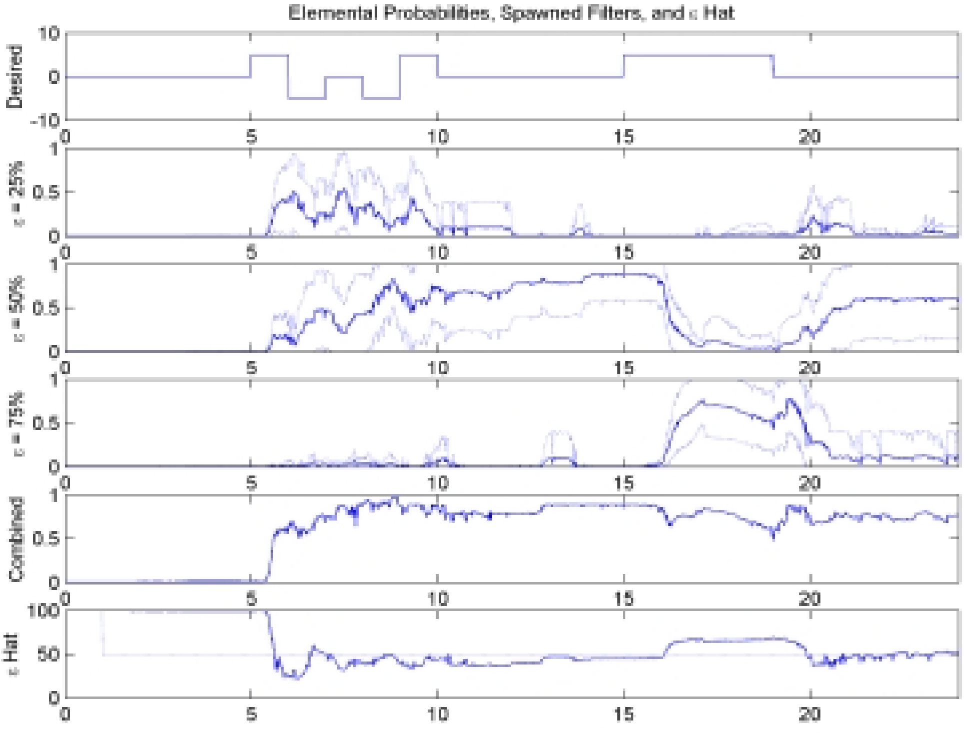
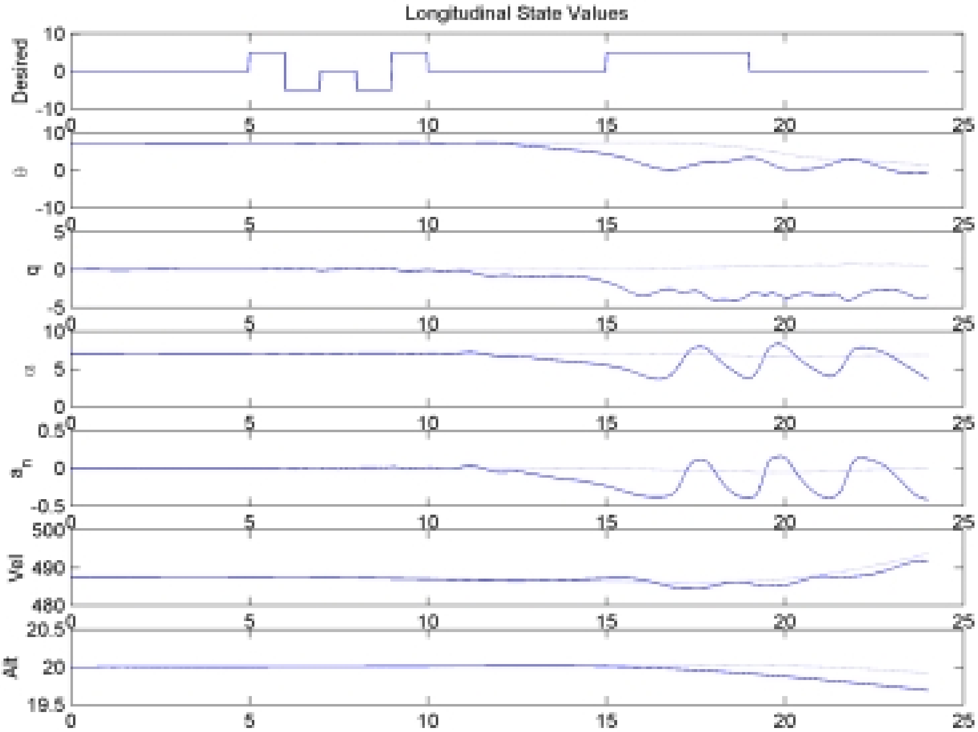


Figure 67b: Elemental Probabilities, Left Flaperon Failure, ε = 50% Using ε $_{HAT}$



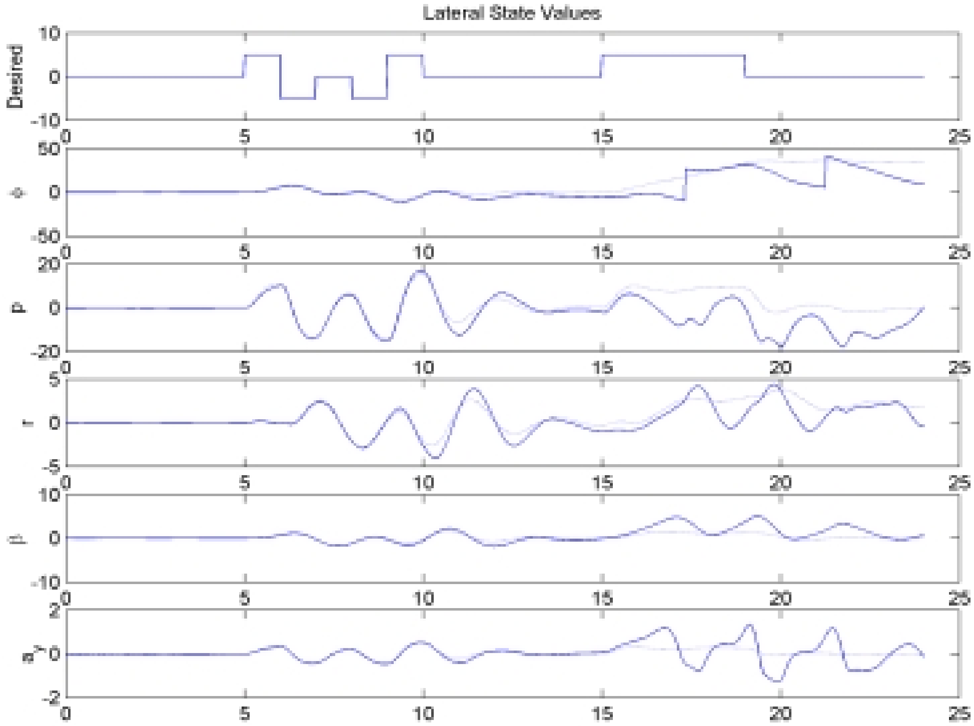
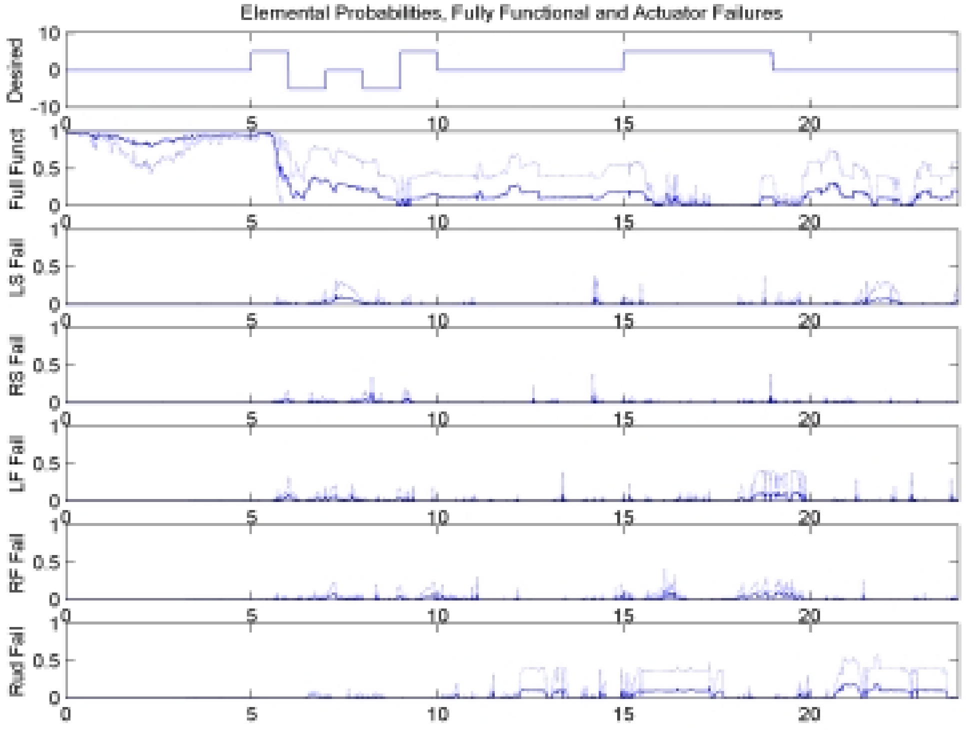


Figure 67c: State Values, Left Flaperon Failure, $\varepsilon = 50\%$ Using ε_{HAT}



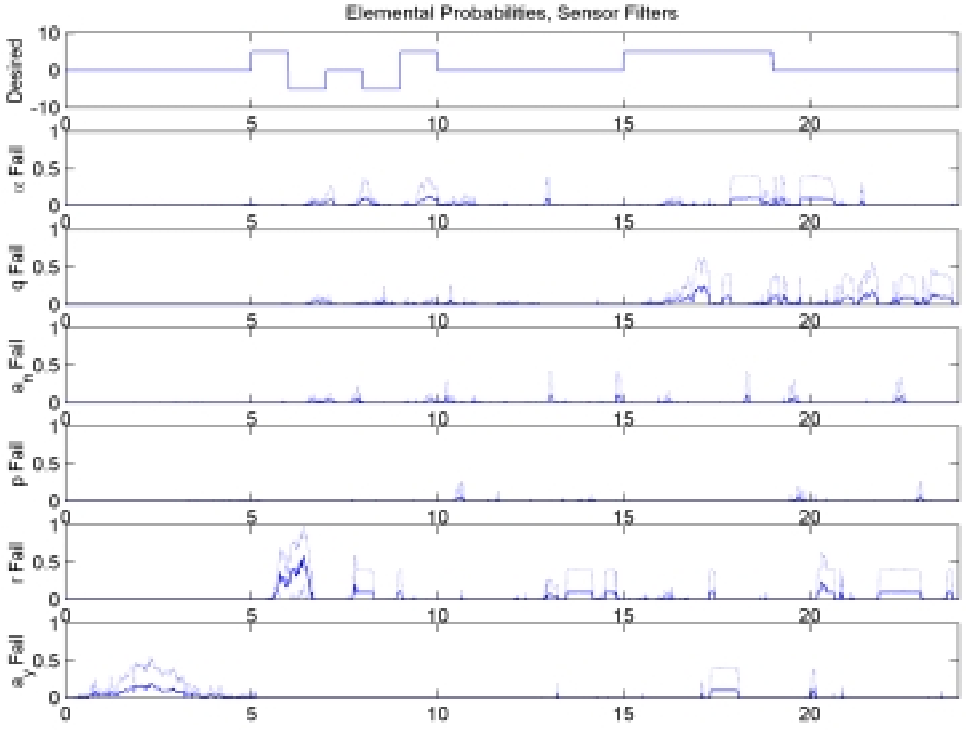


Figure 68a: Elemental Probabilities, Left Flaperon Failure, ε = 75% Using ε HAT

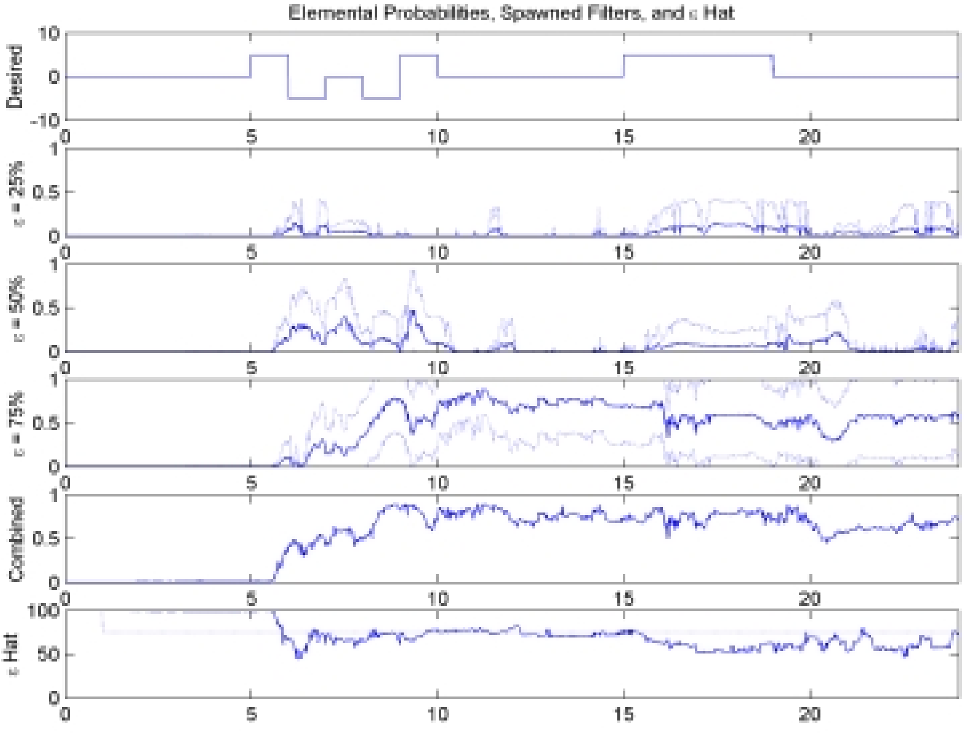


Figure 68b: Elemental Probabilities, Left Flaperon Failure, ε = 75% Using ε_{HAT}

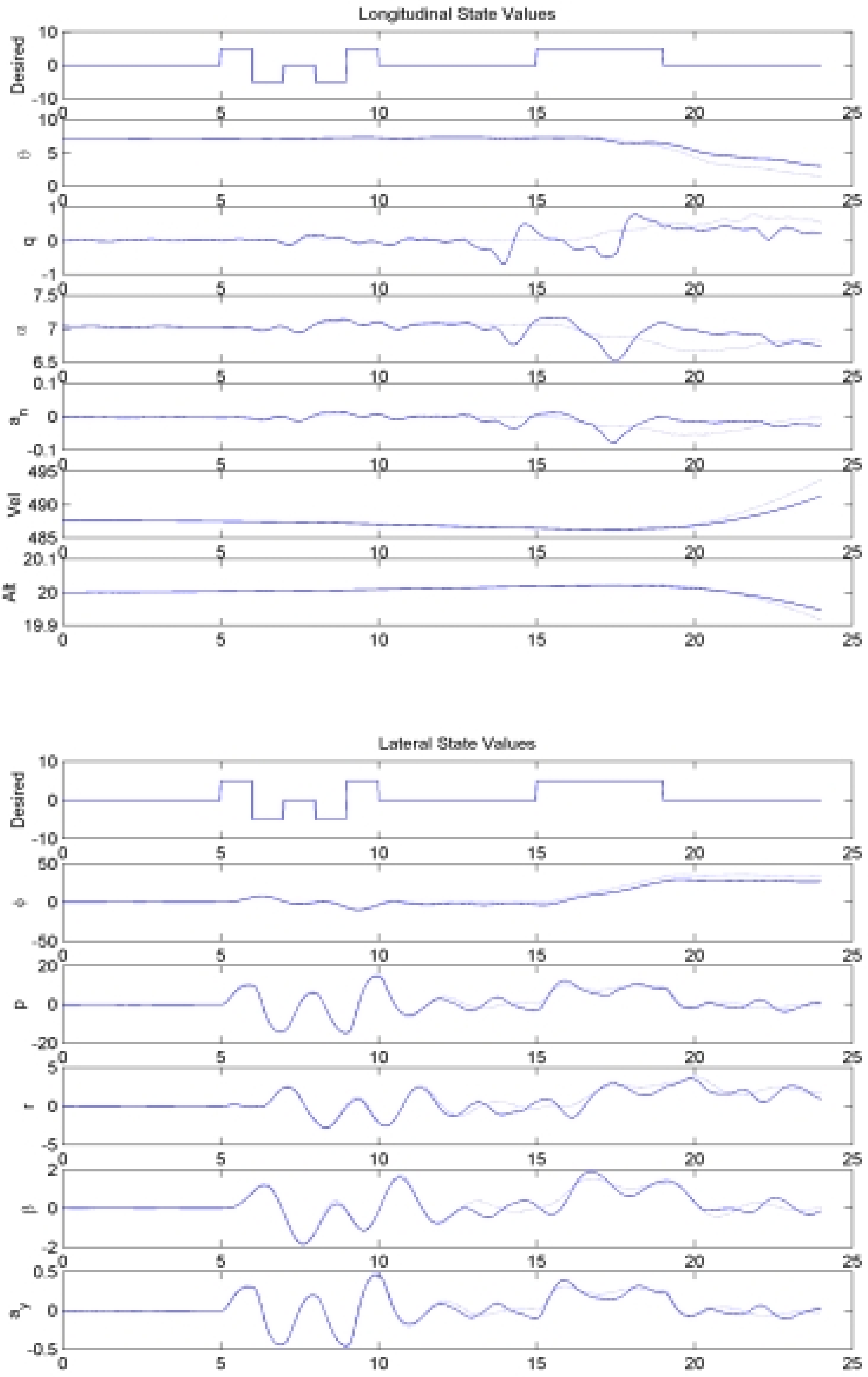


Figure 68c: State Values, Left Flaperon Failure, ε = 75% Using ε $_{HAT}$

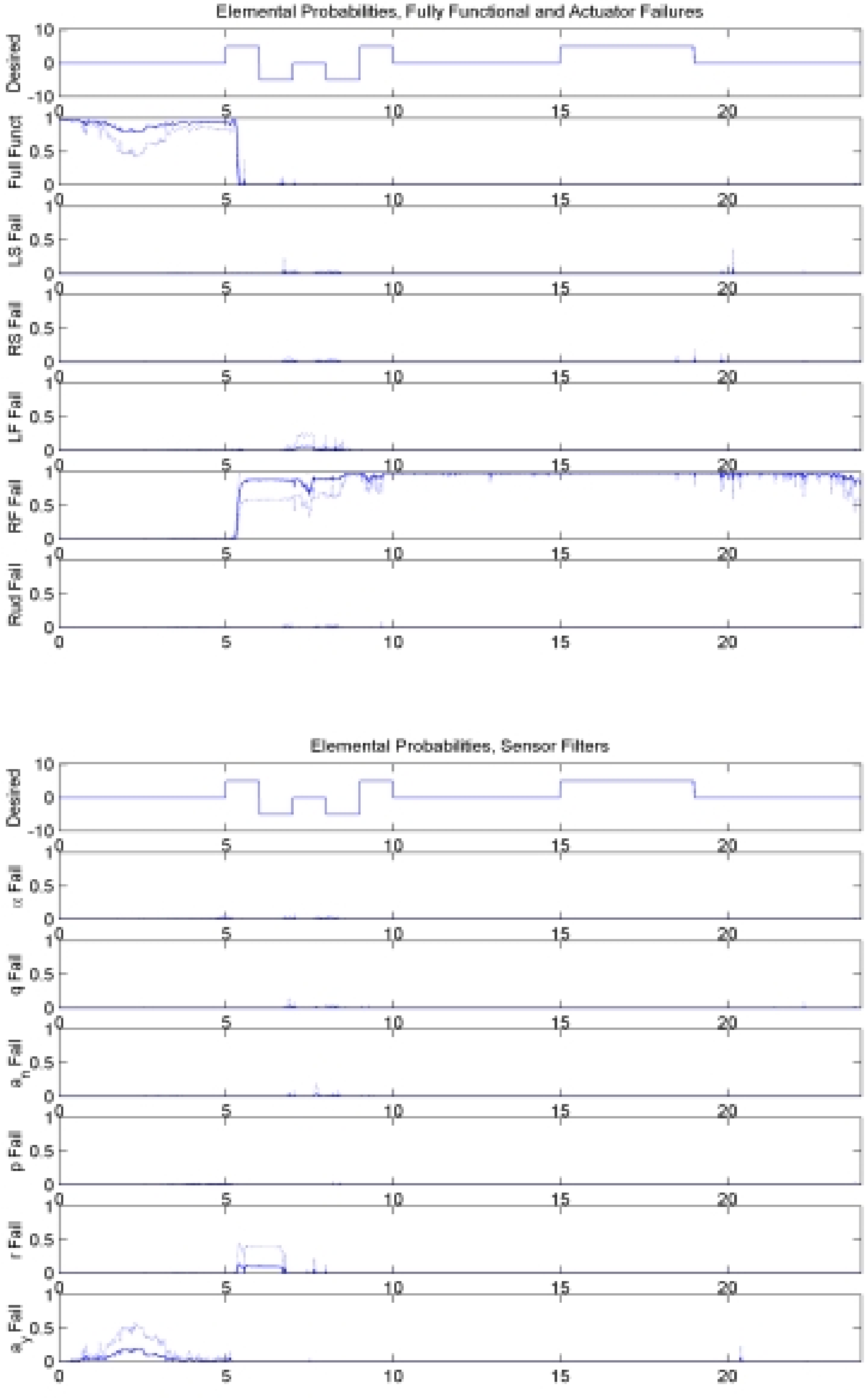


Figure 69a: Elemental Probabilities, Right Flaperon Failure, ε = 0% Using ε_{HAT}

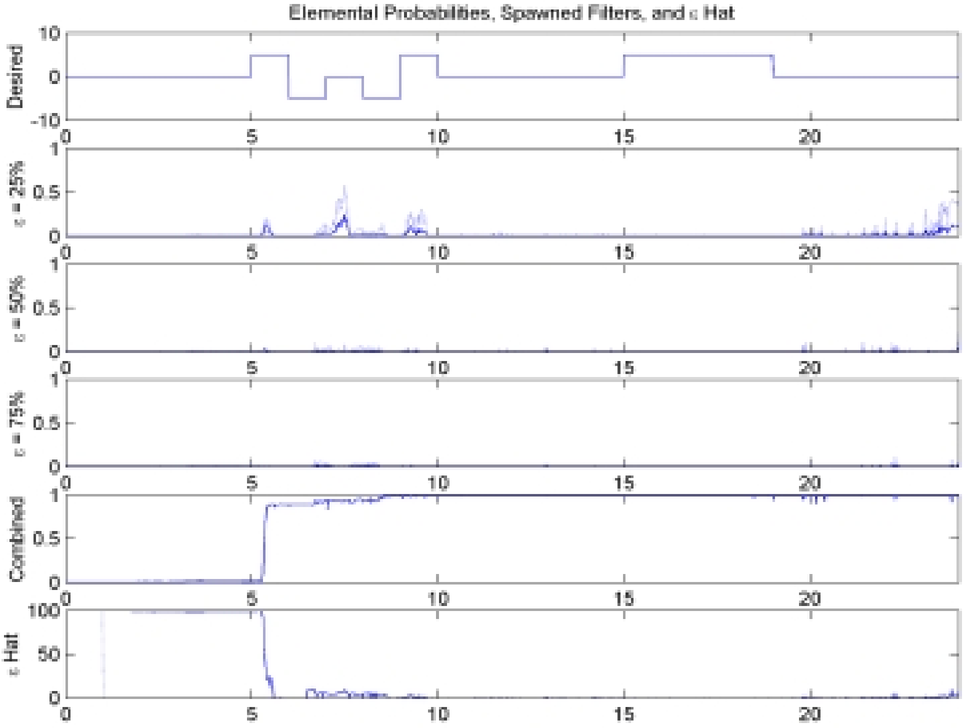


Figure 69b: Elemental Probabilities, Right Flaperon Failure, ε = 0% Using ε_{HAT}

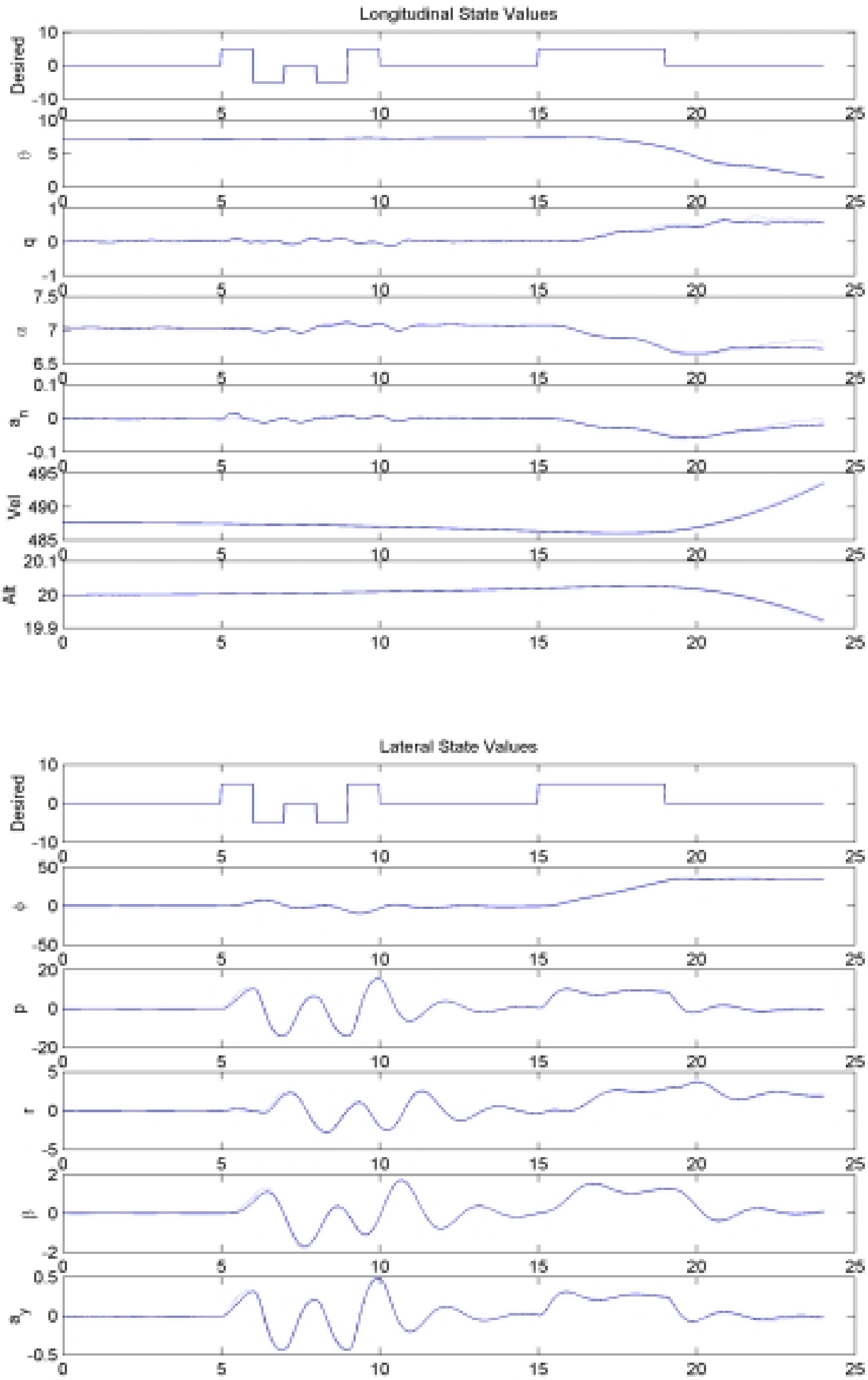


Figure 69c: State Values, Right Flaperon Failure, $\varepsilon = 0\%$ Using ε_{HAT}

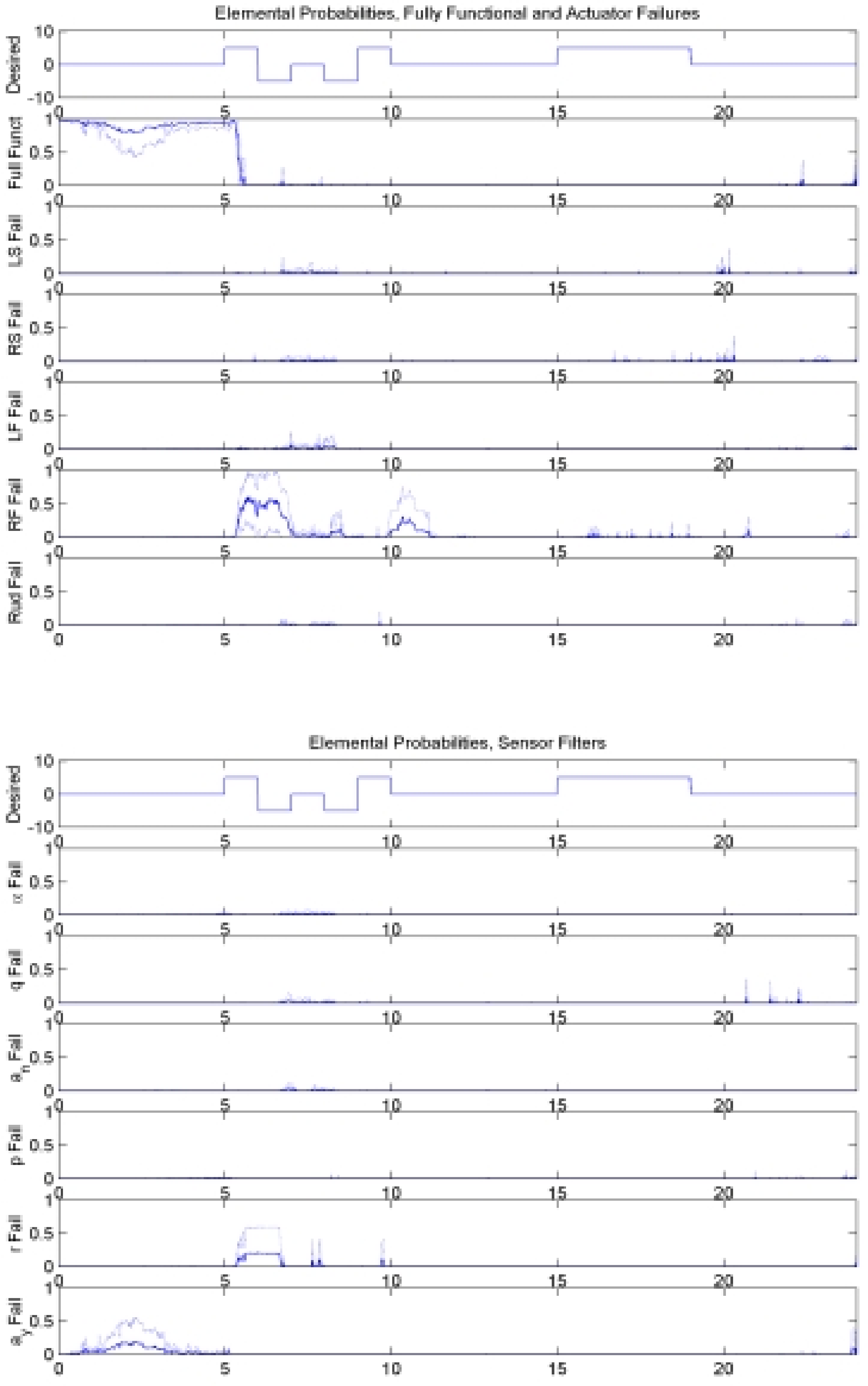


Figure 70a: Elemental Probabilities, Right Flaperon Failure, ε = 25% Using ε $_{HAT}$

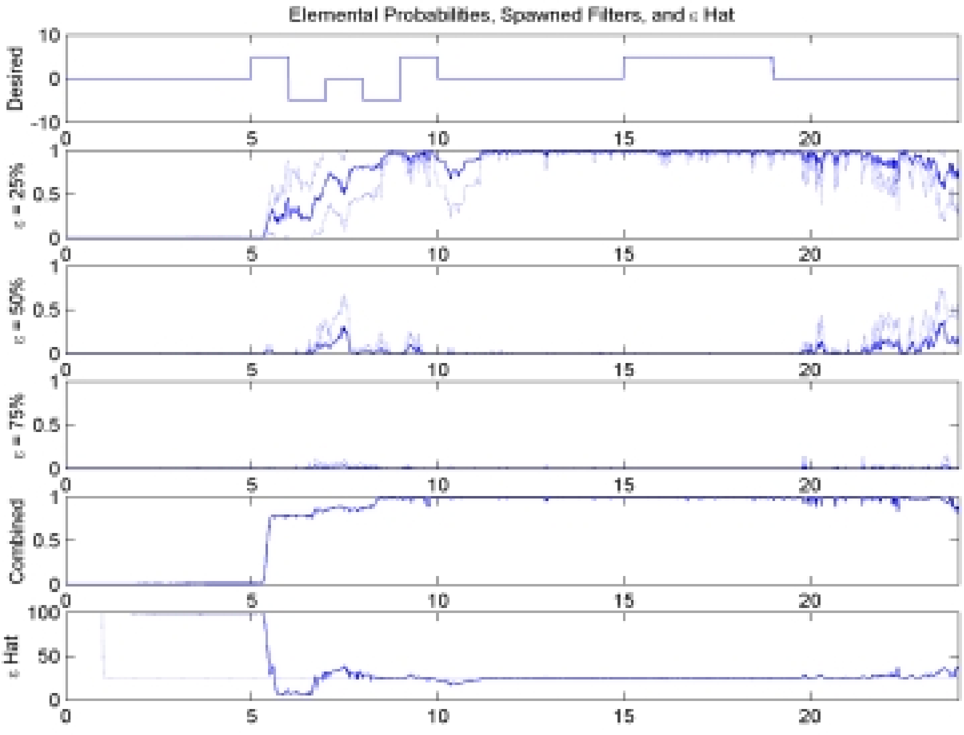
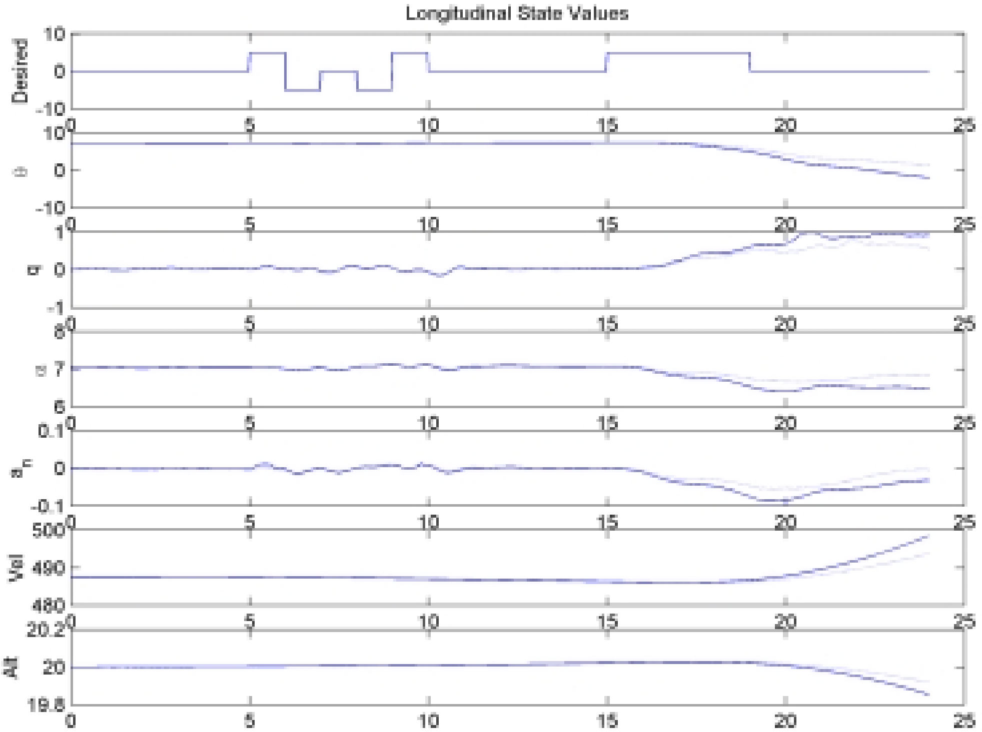


Figure 70b: Elemental Probabilities, Right Flaperon Failure, ε = 25% Using ε_{HAT}



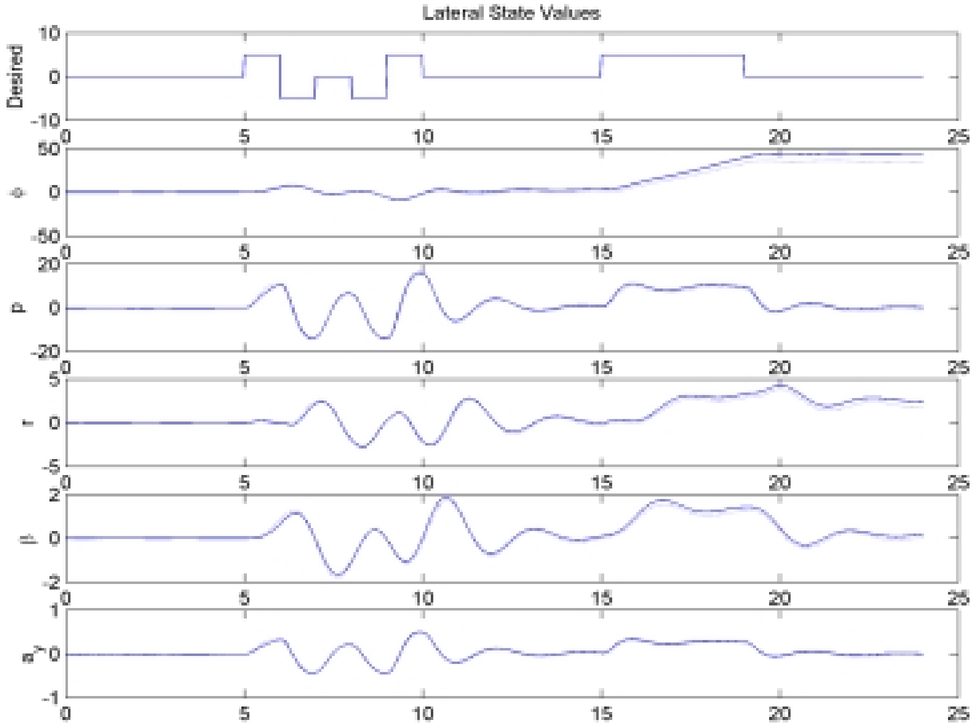


Figure 70c: State Values, Right Flaperon Failure, ε = 25% Using ε $_{HAT}$

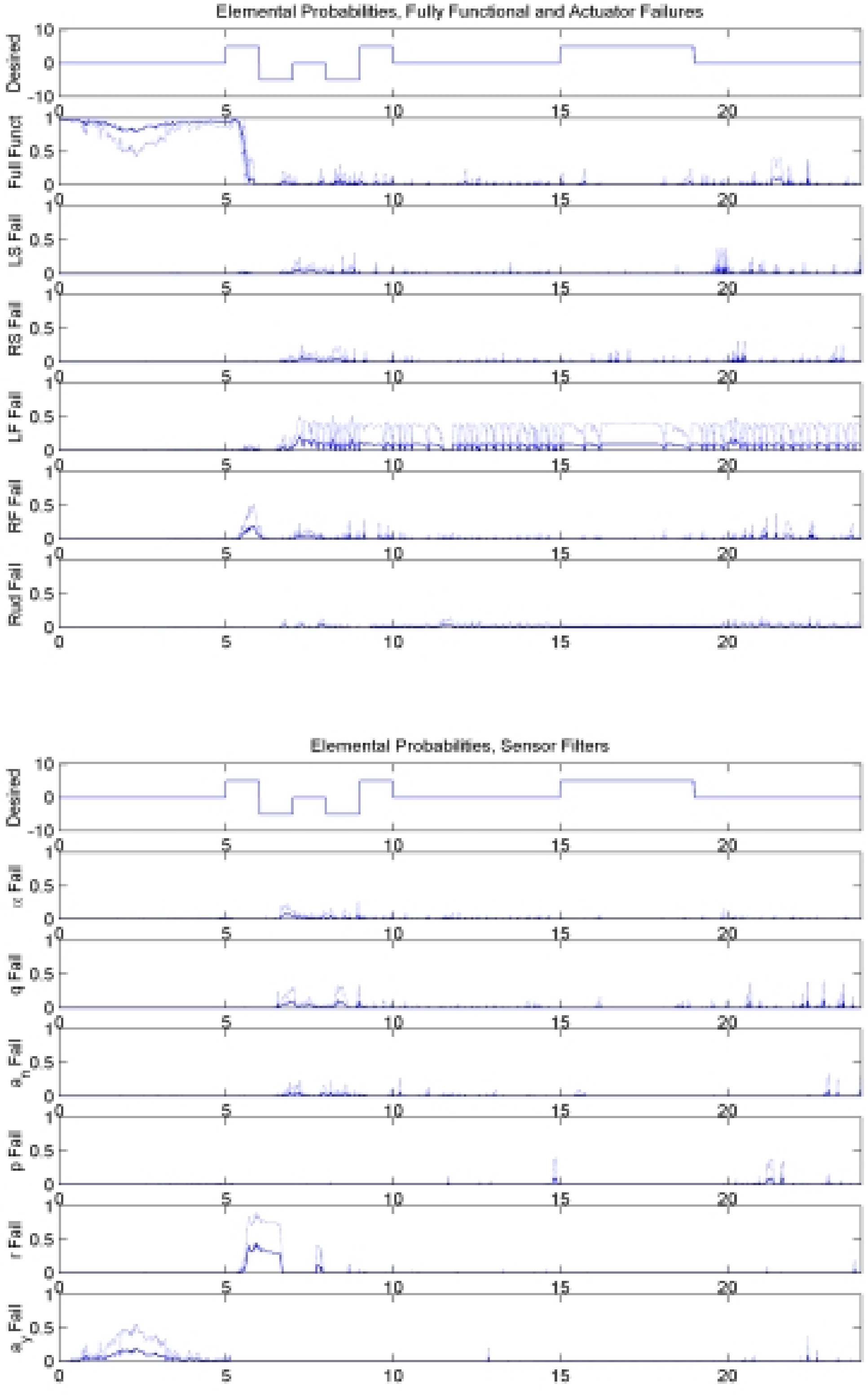


Figure 71a: Elemental Probabilities, Right Flaperon Failure, $\varepsilon = 50\%$ Using ε_{HAT}

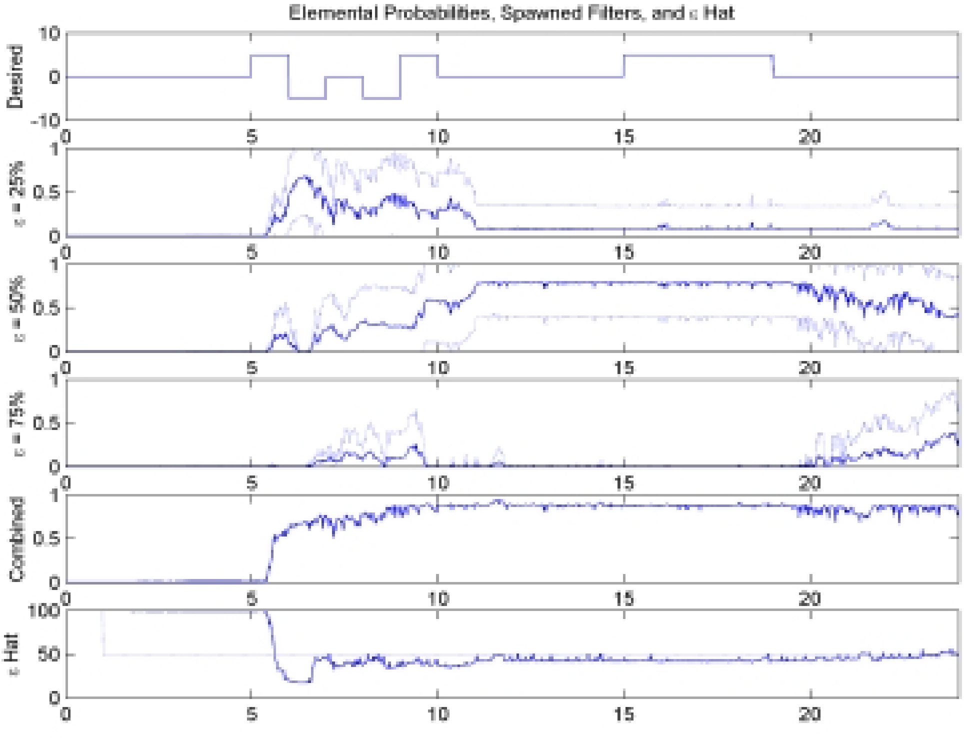


Figure 71b: Elemental Probabilities, Right Flaperon Failure, ε = 50% Using ε_{HAT}

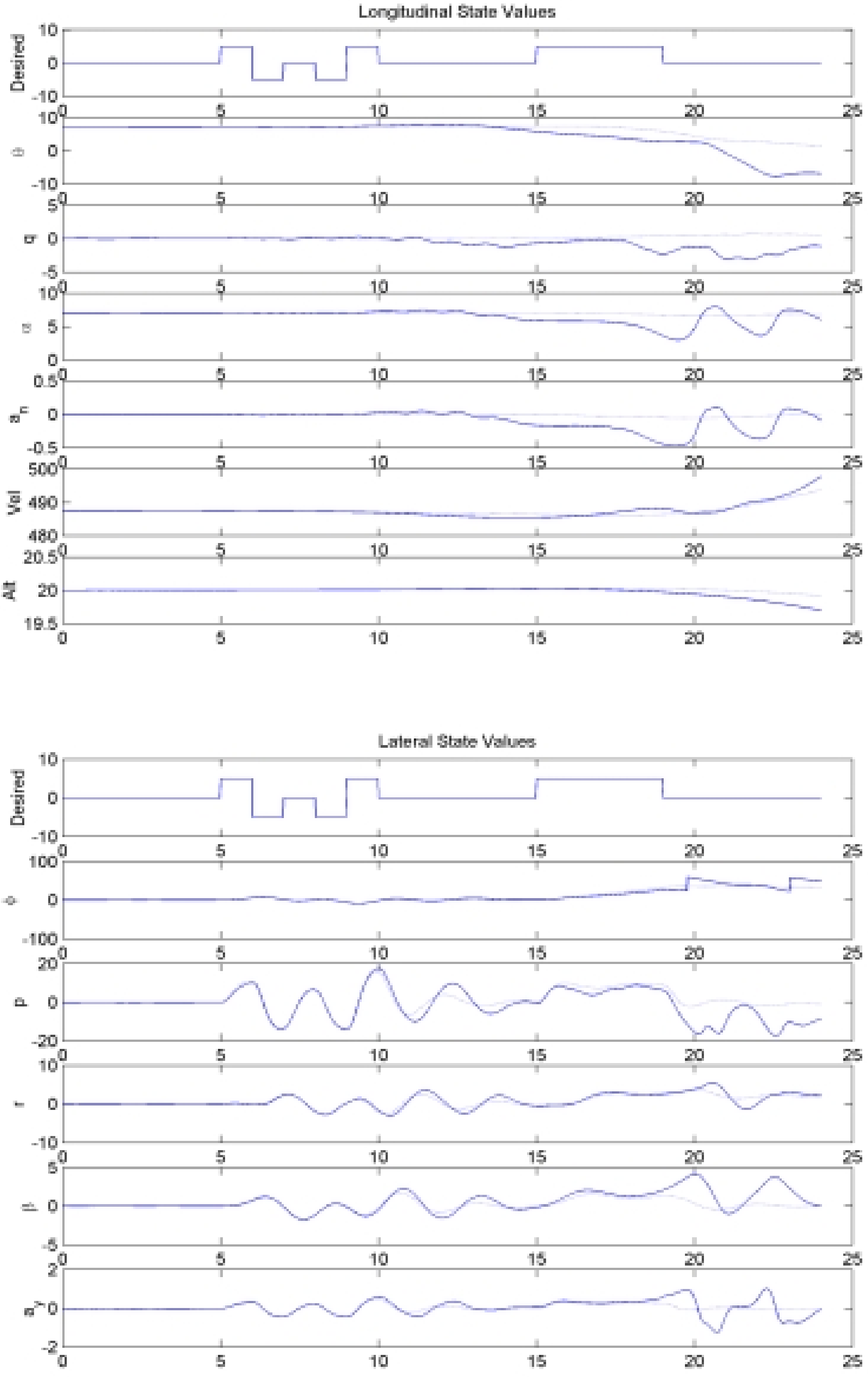


Figure 71c: State Values, Right Flaperon Failure, ε = 50% Using ε $_{HAT}$

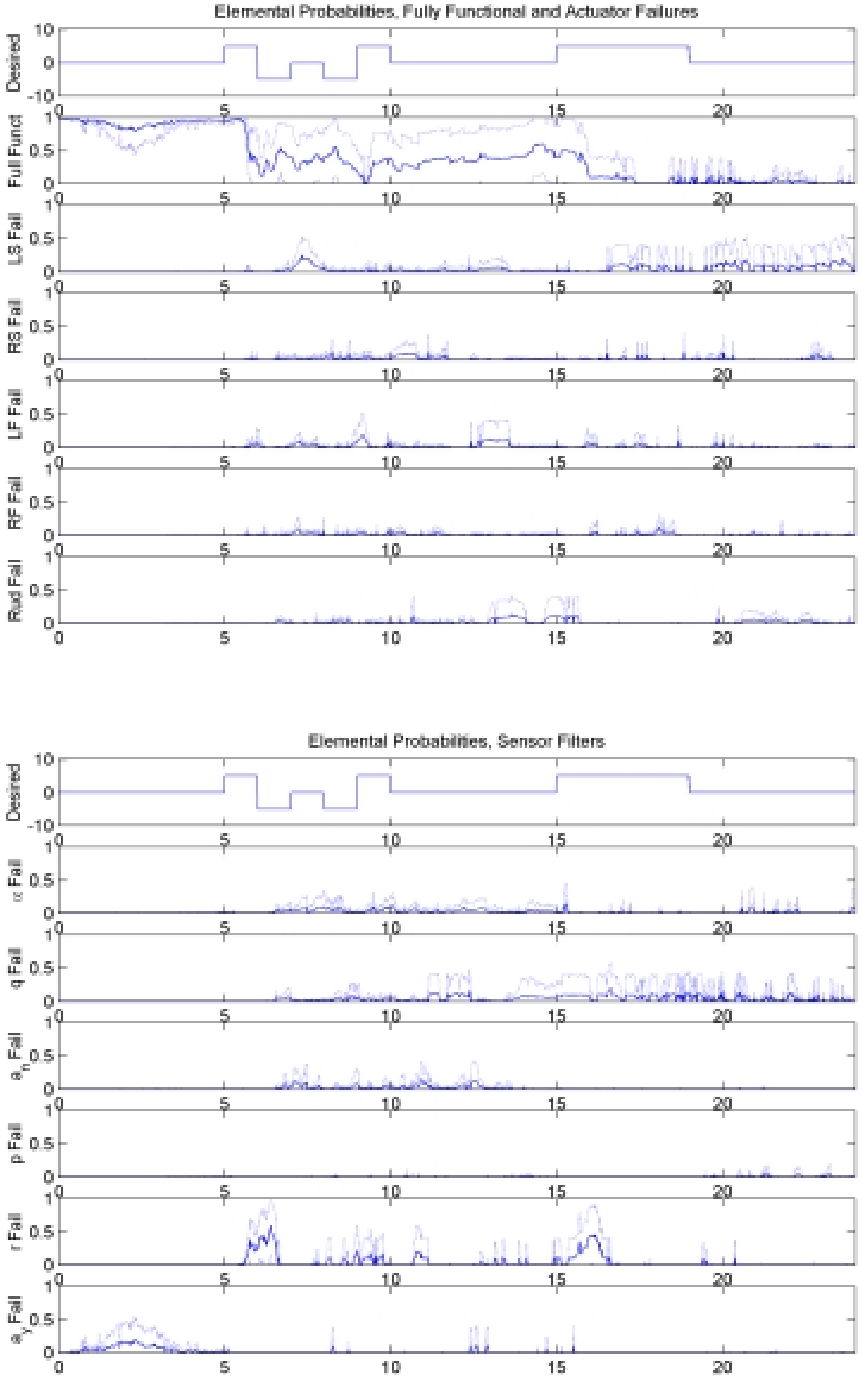


Figure 72a: Elemental Probabilities, Right Flaperon Failure, ε = 75% Using ε $_{HAT}$

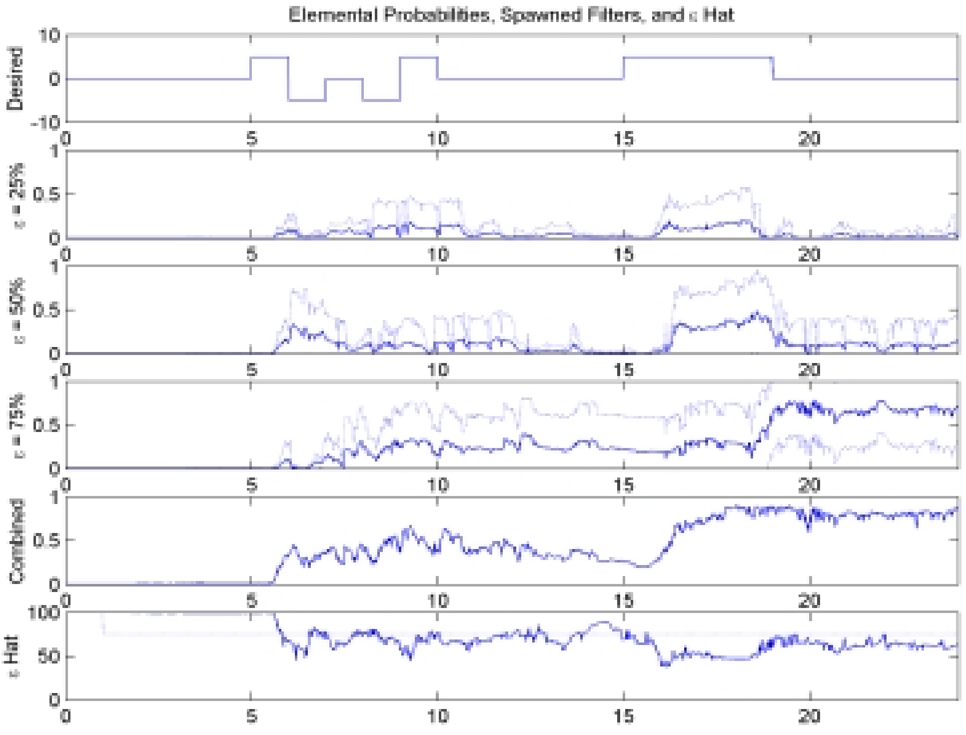
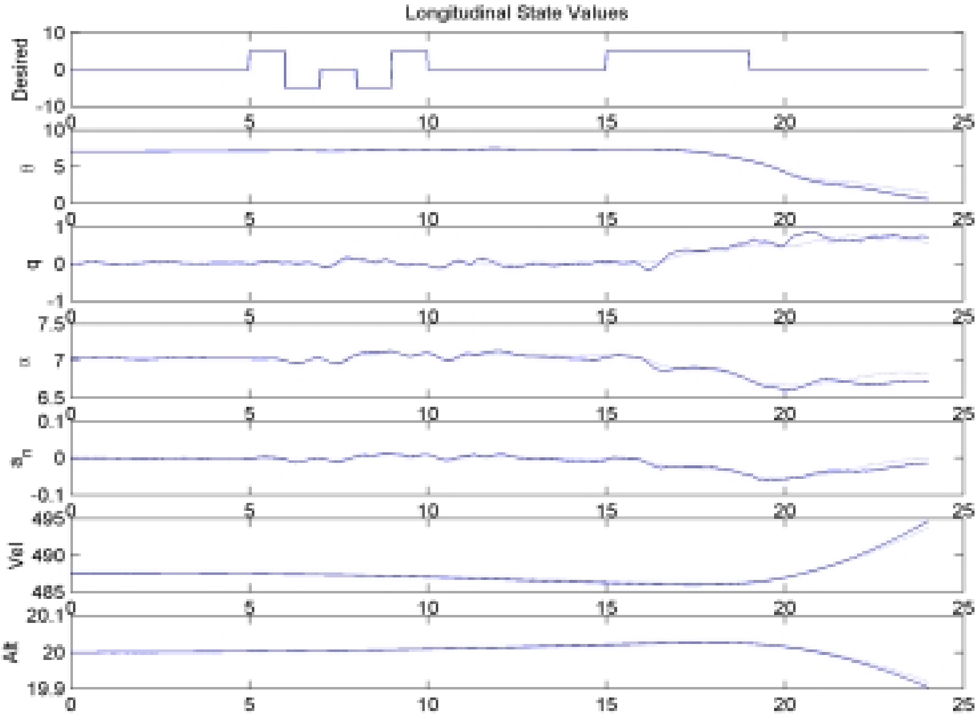


Figure 72b: Elemental Probabilities, Right Flaperon Failure, ε = 75% Using ε $_{\text{HAT}}$



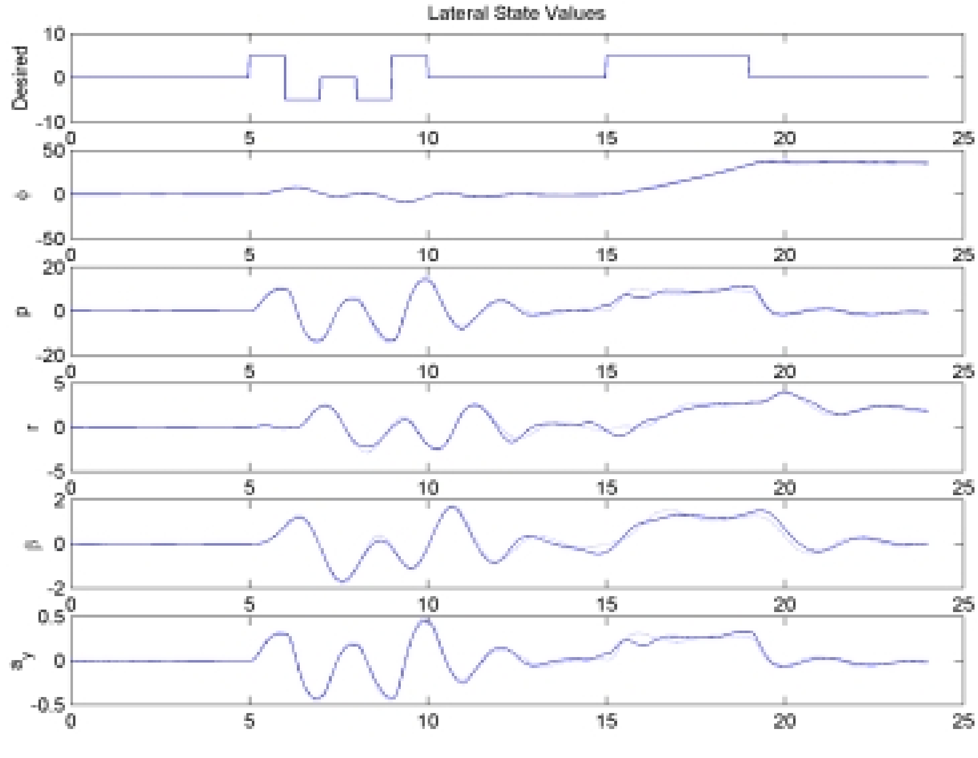


Figure 72c: State Values, Right Flaperon Failure, ε = 75% Using ε_{HAT}

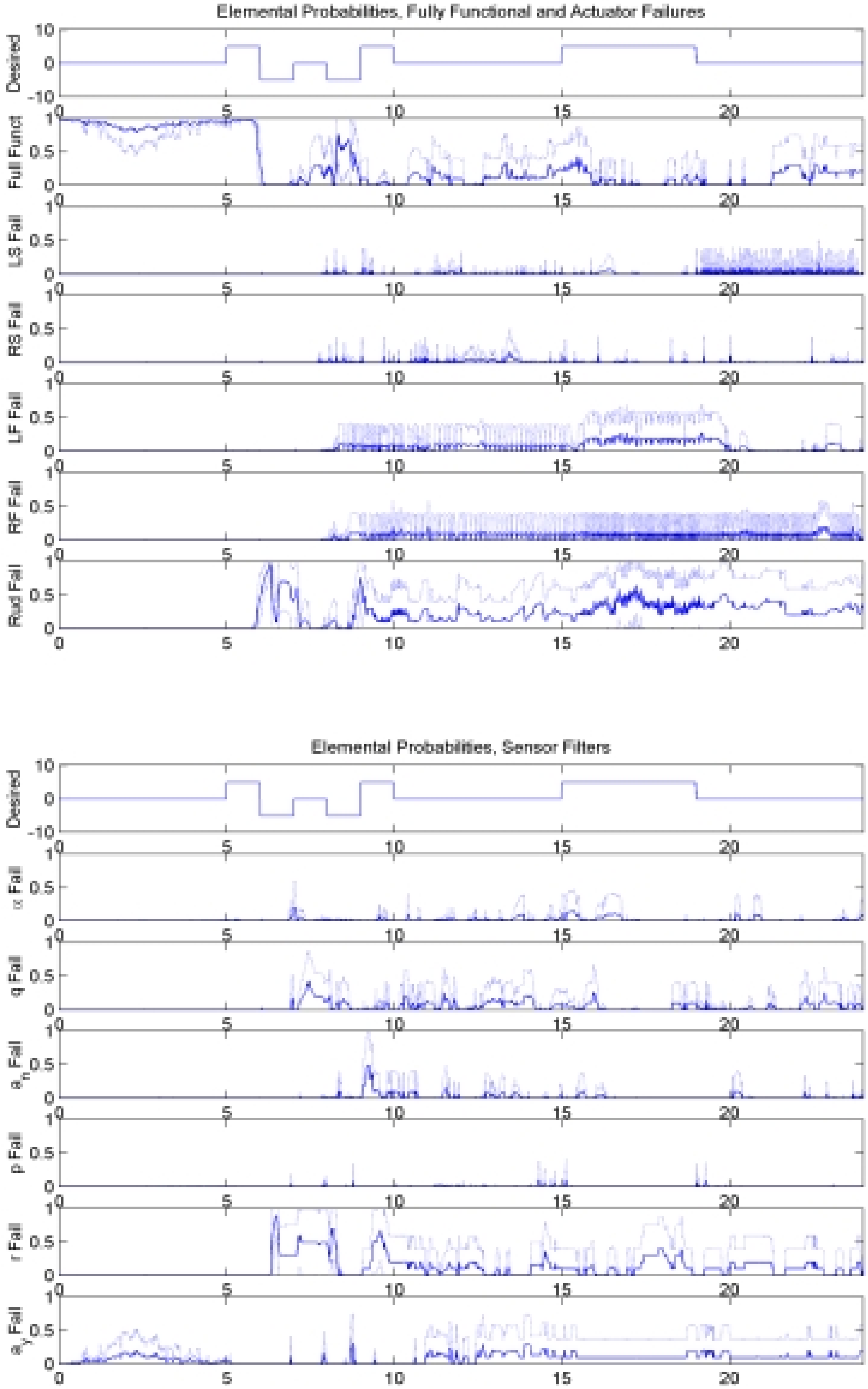


Figure 73a: Elemental Probabilities, Rudder Failure, ε = 0% Using ε_{HAT}

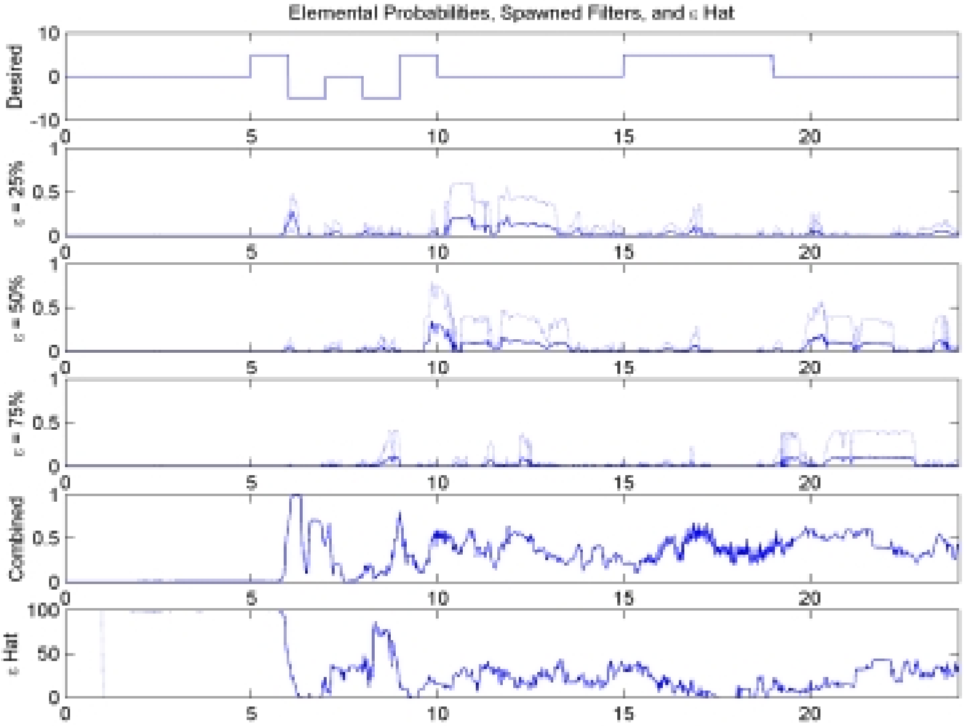


Figure 73b: Elemental Probabilities, Rudder Failure, ε = 0% Using ε $_{HAT}$

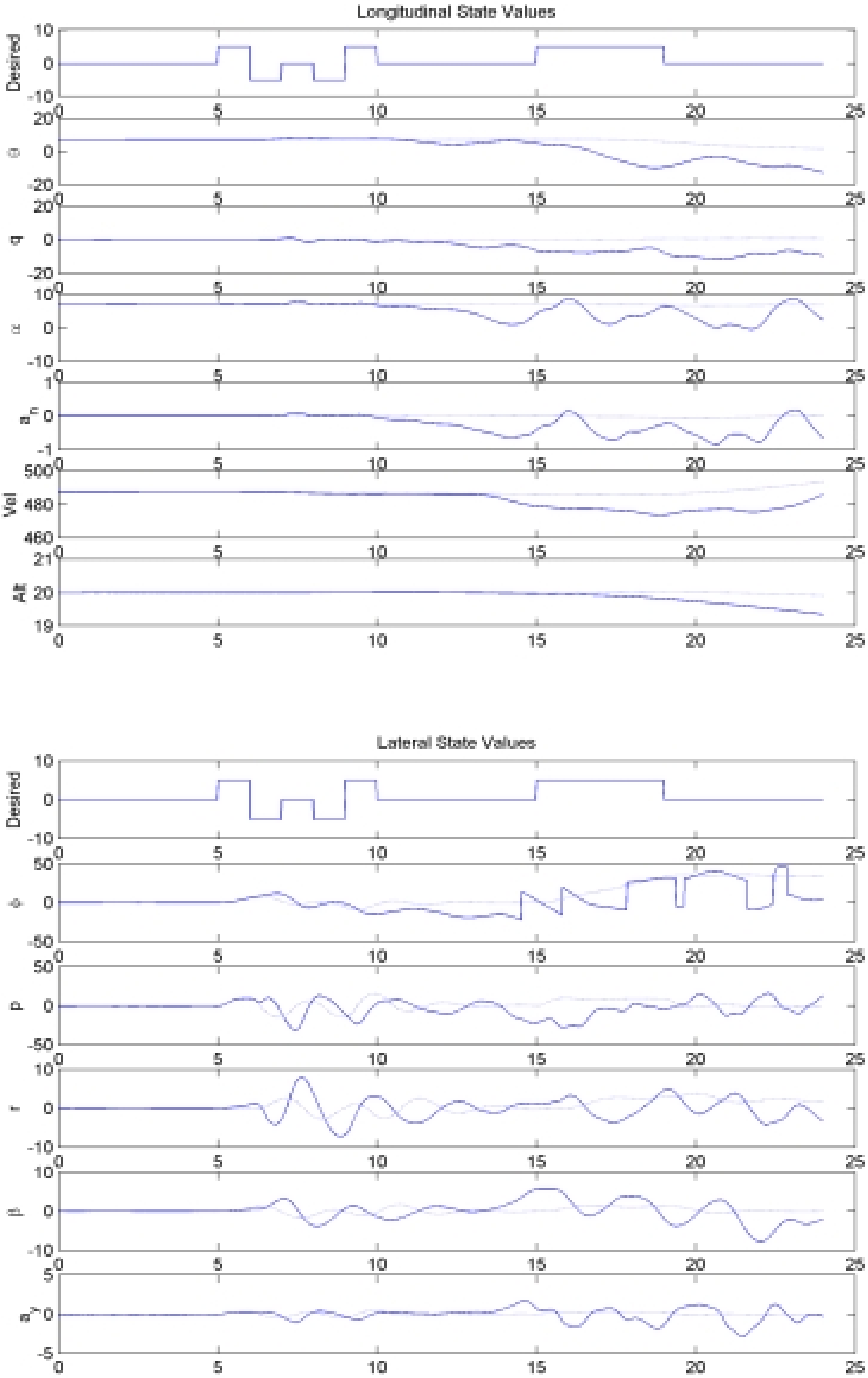


Figure 73c: State Values, Rudder Failure, ε = 0% Using ε_{HAT}

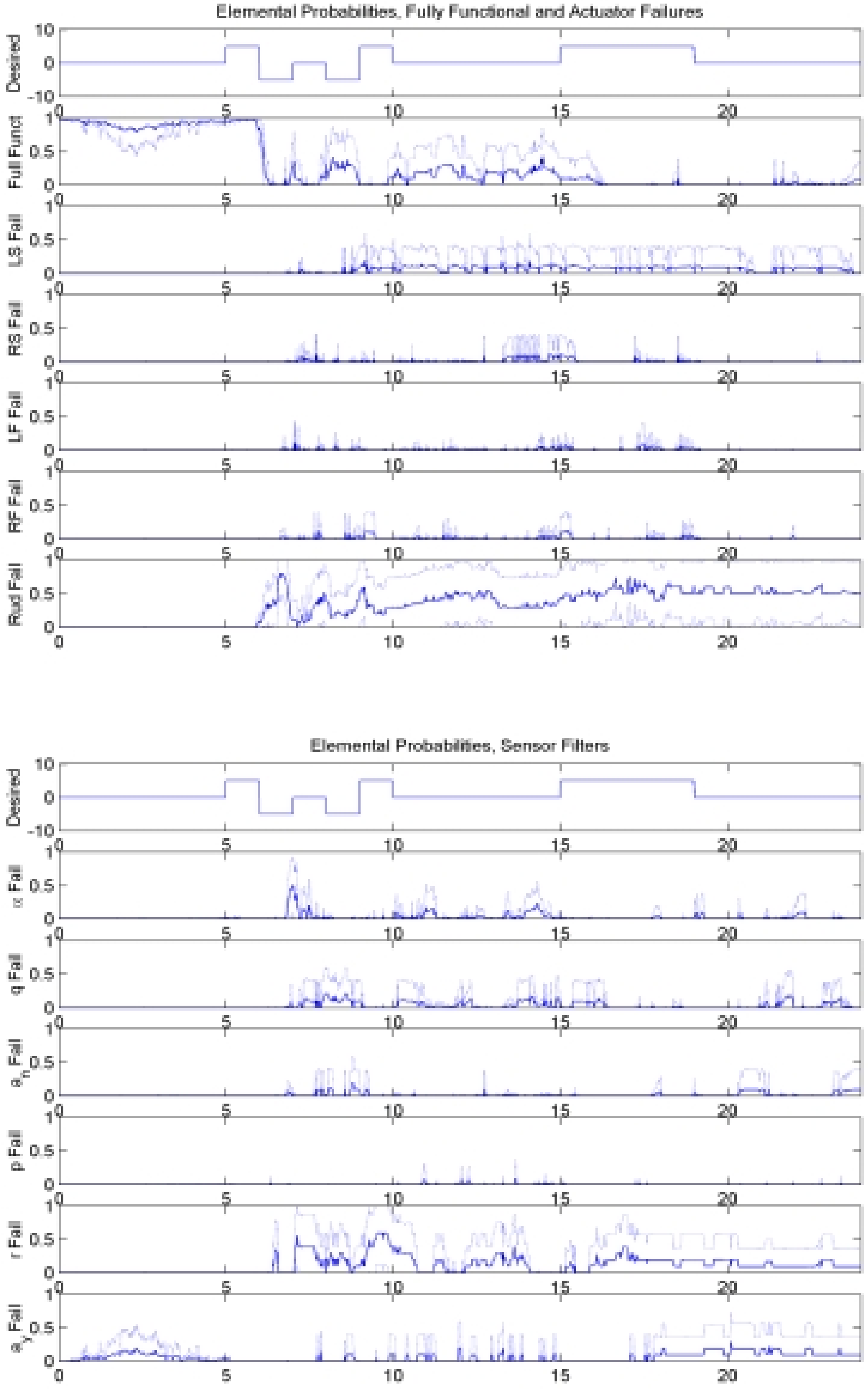


Figure 74a: Elemental Probabilities, Rudder Failure, ε = 25% Using ε HAT

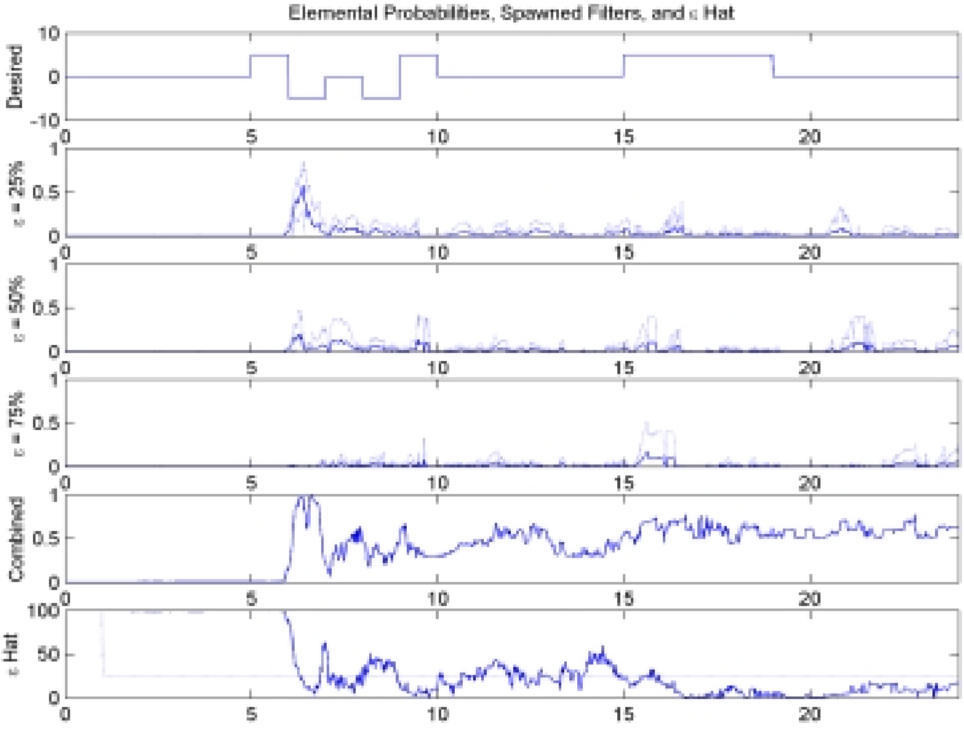


Figure 74b: Elemental Probabilities, Rudder Failure, ε = 25% Using ε $_{HAT}$

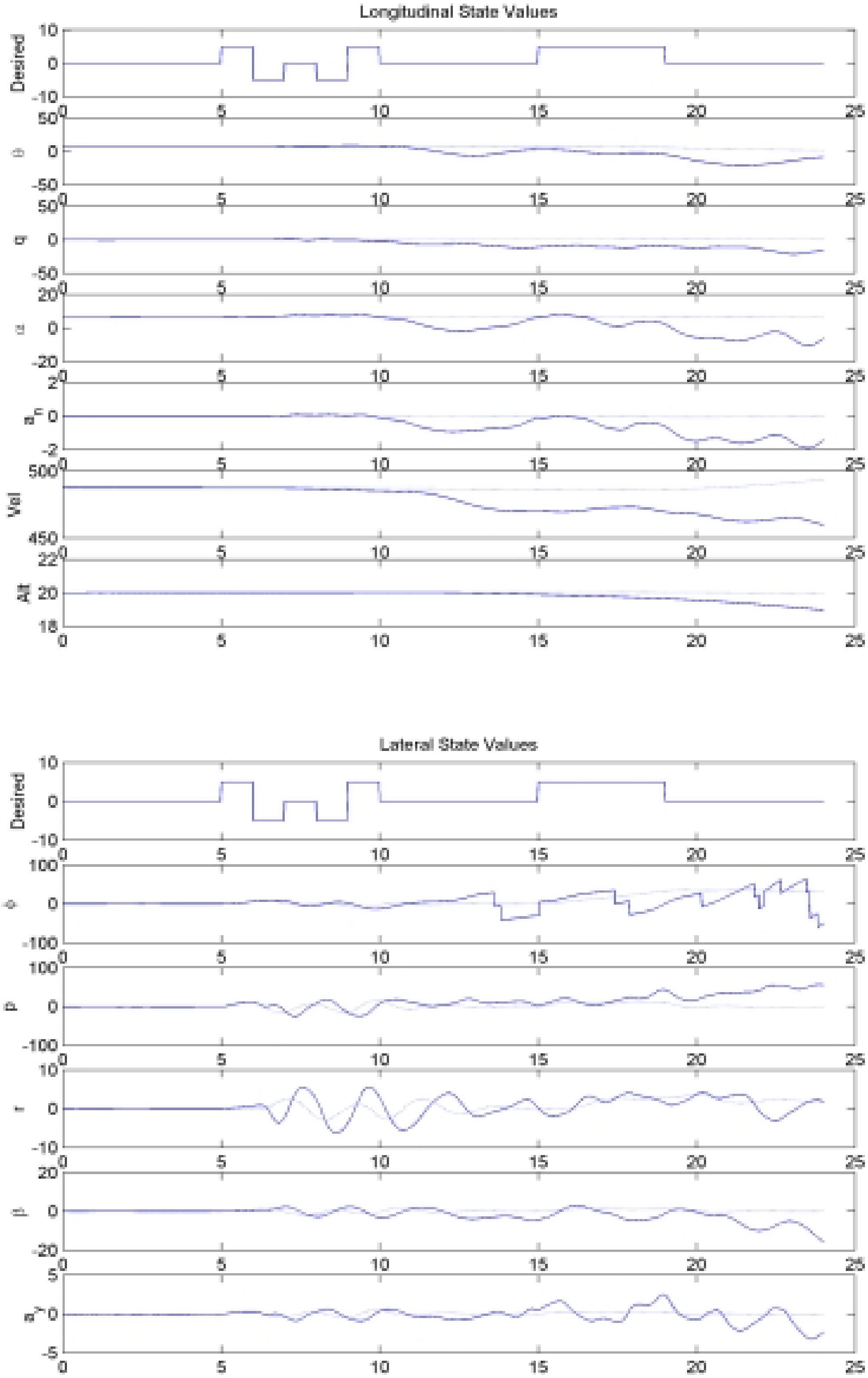
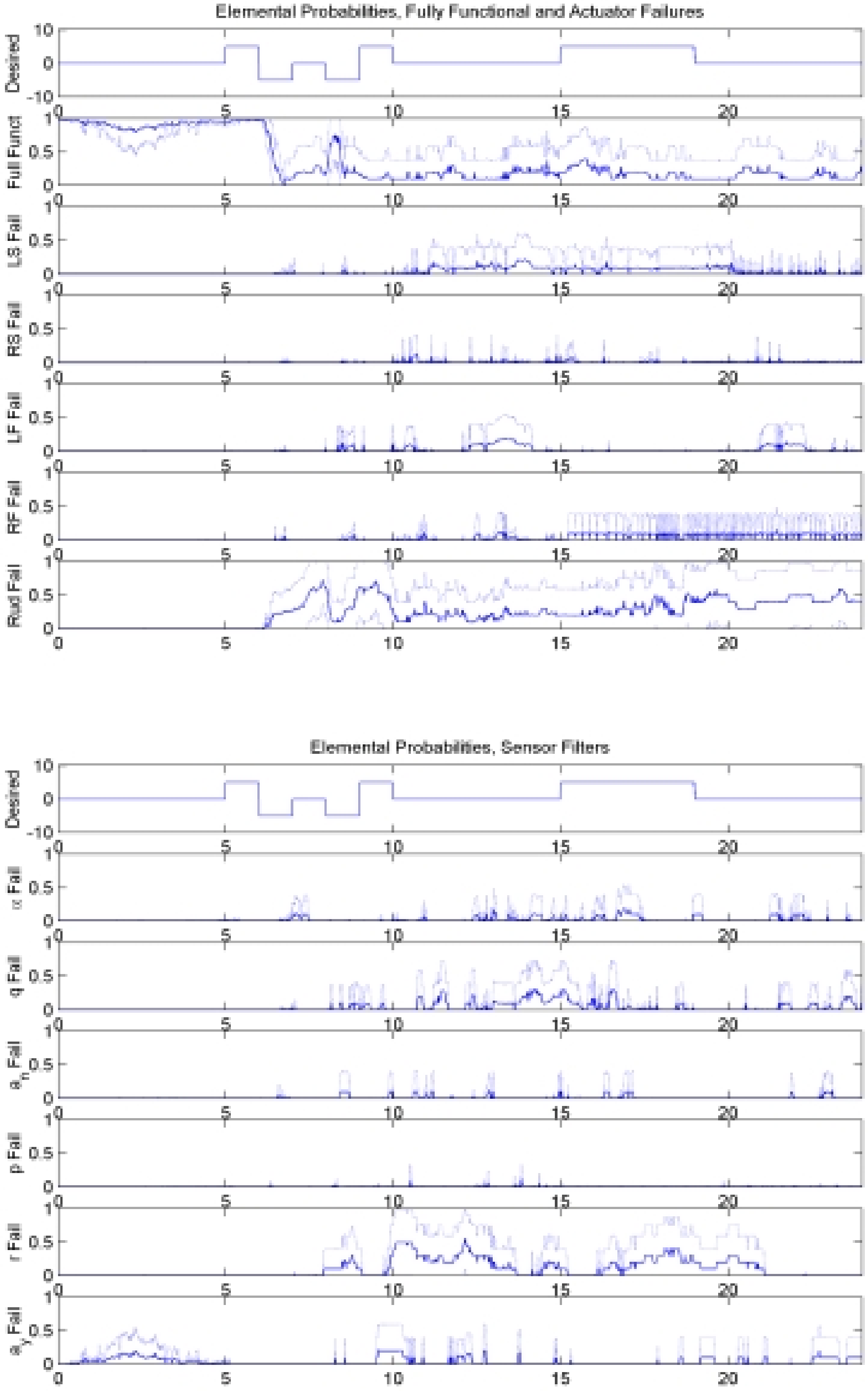


Figure 74c: State Values, Rudder Failure, ε = 25% Using ε $_{HAT}$


Elemental Probabilities, Fully Functional and Actuator Failures

Elemental Probabilities, Sensor Filters

Figure 75a: Elemental Probabilities, Rudder Failure, ε = 50% Using ε $_{\text{HAT}}$

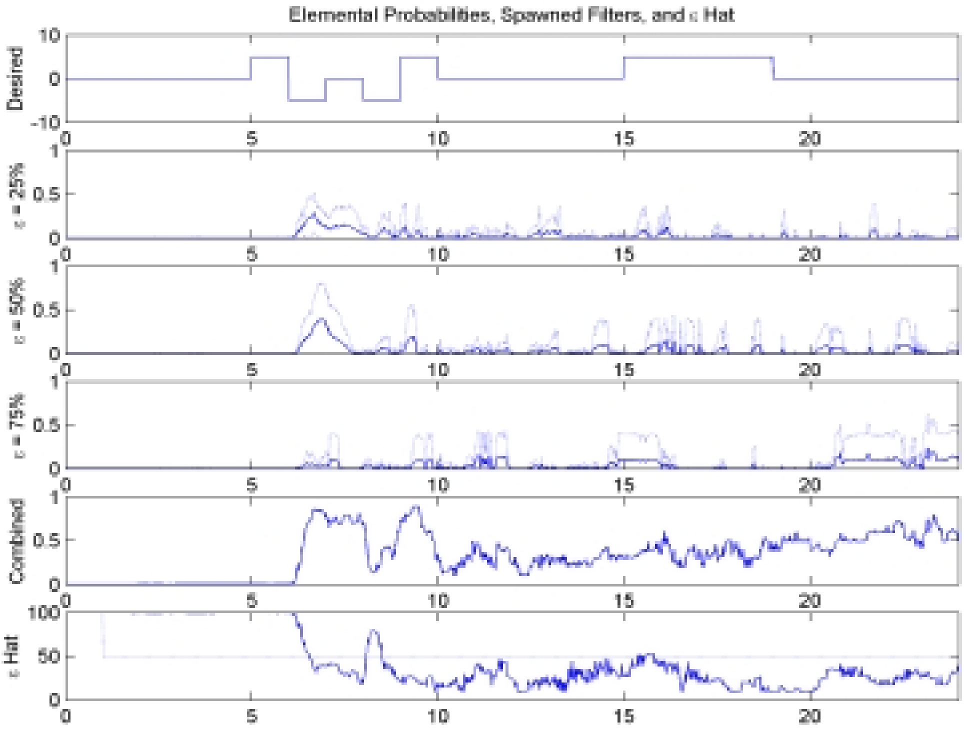


Figure 75b: Elemental Probabilities, Rudder Failure, ε = 50% Using ε $_{\text{HAT}}$

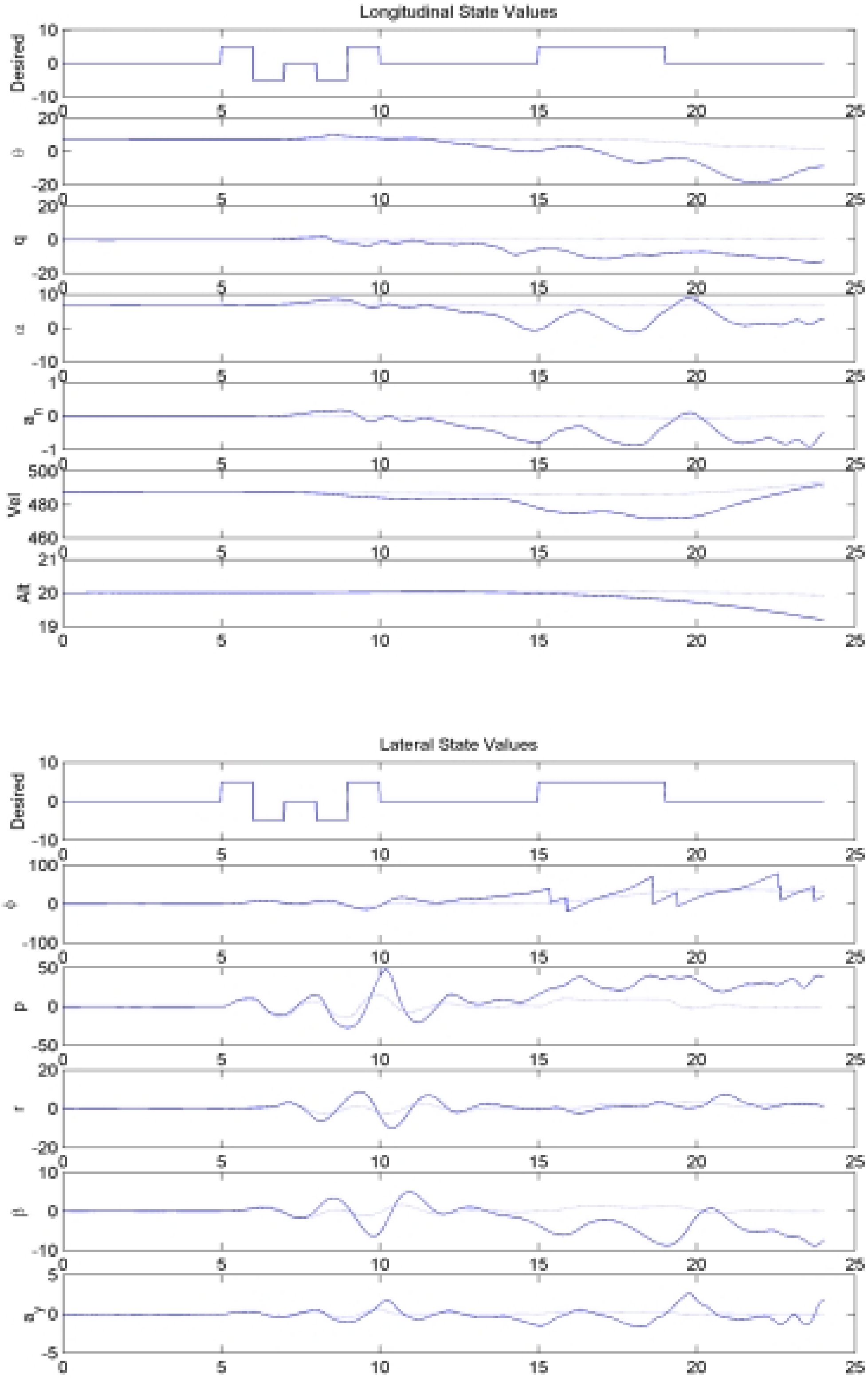


Figure 75c: State Values, Rudder Failure, ε = 50% Using ε_{HAT}

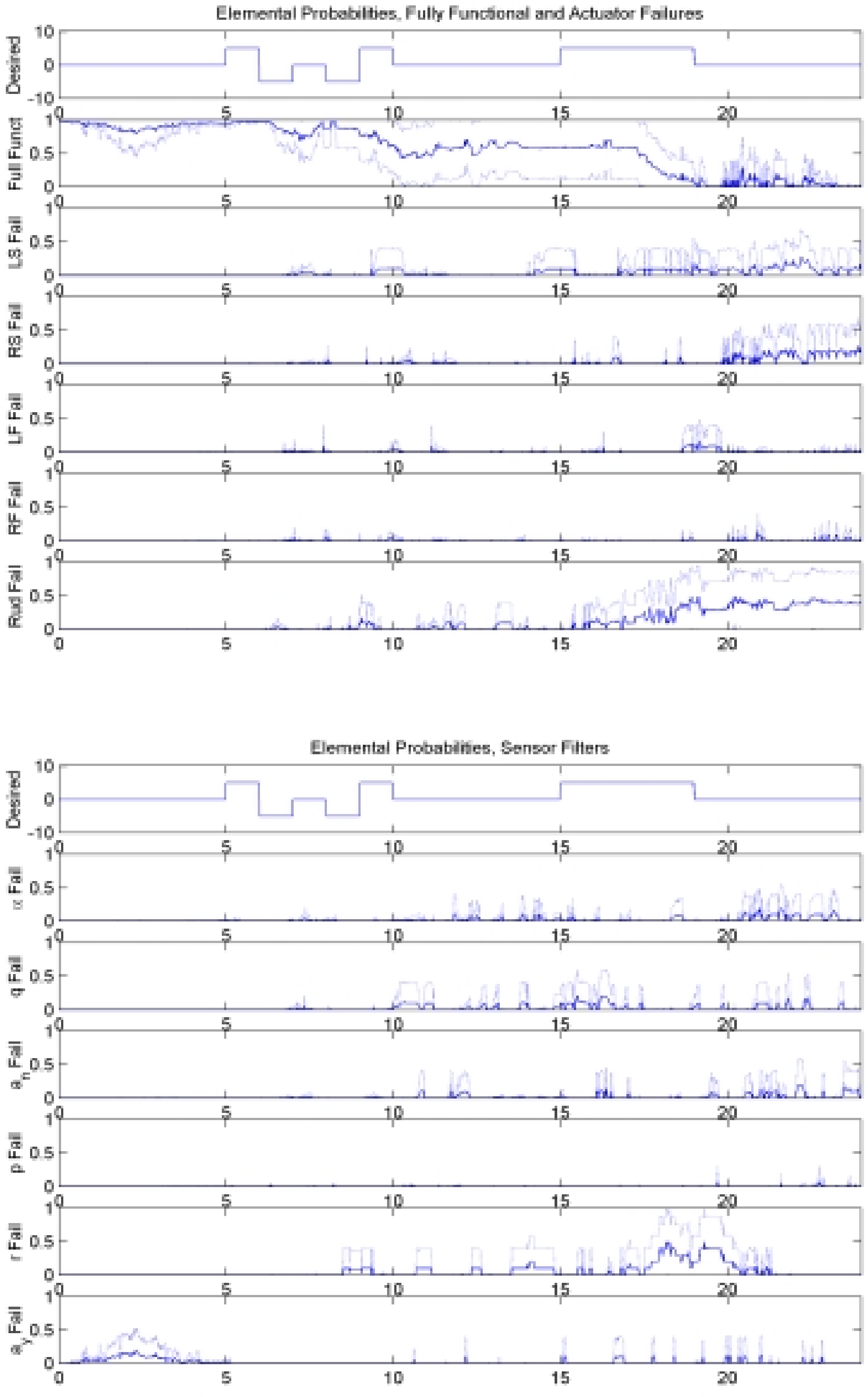


Figure 76a: Elemental Probabilities, Rudder Failure, ε = 75% Using $\varepsilon_{\mathrm{HAT}}$

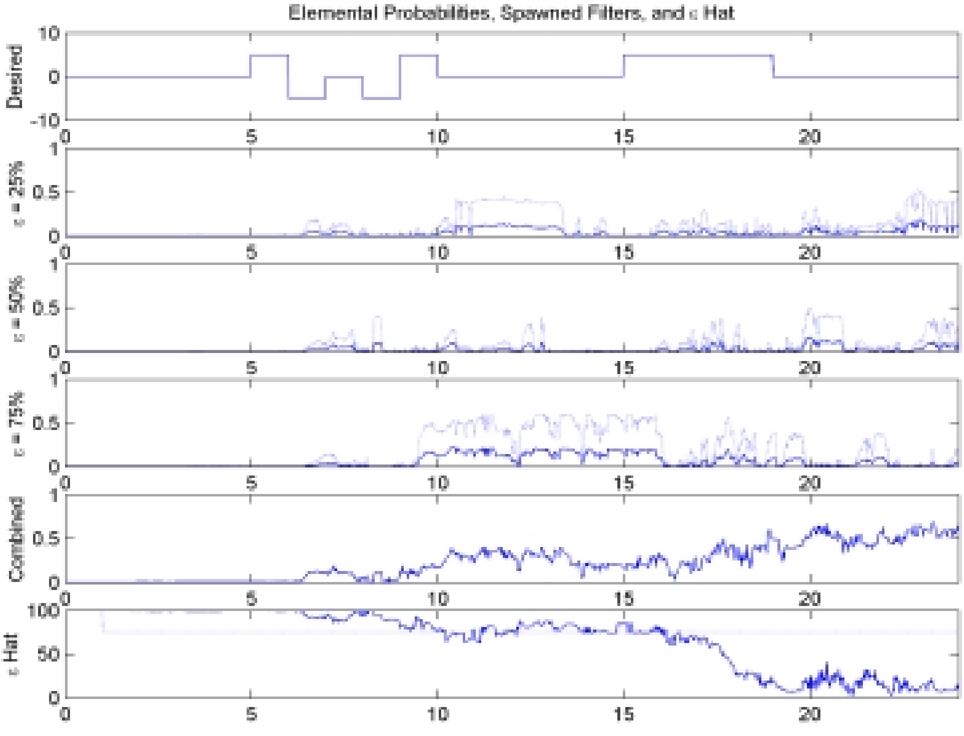


Figure 76b: Elemental Probabilities, Rudder Failure, ε = 75% Using ε $_{HAT}$

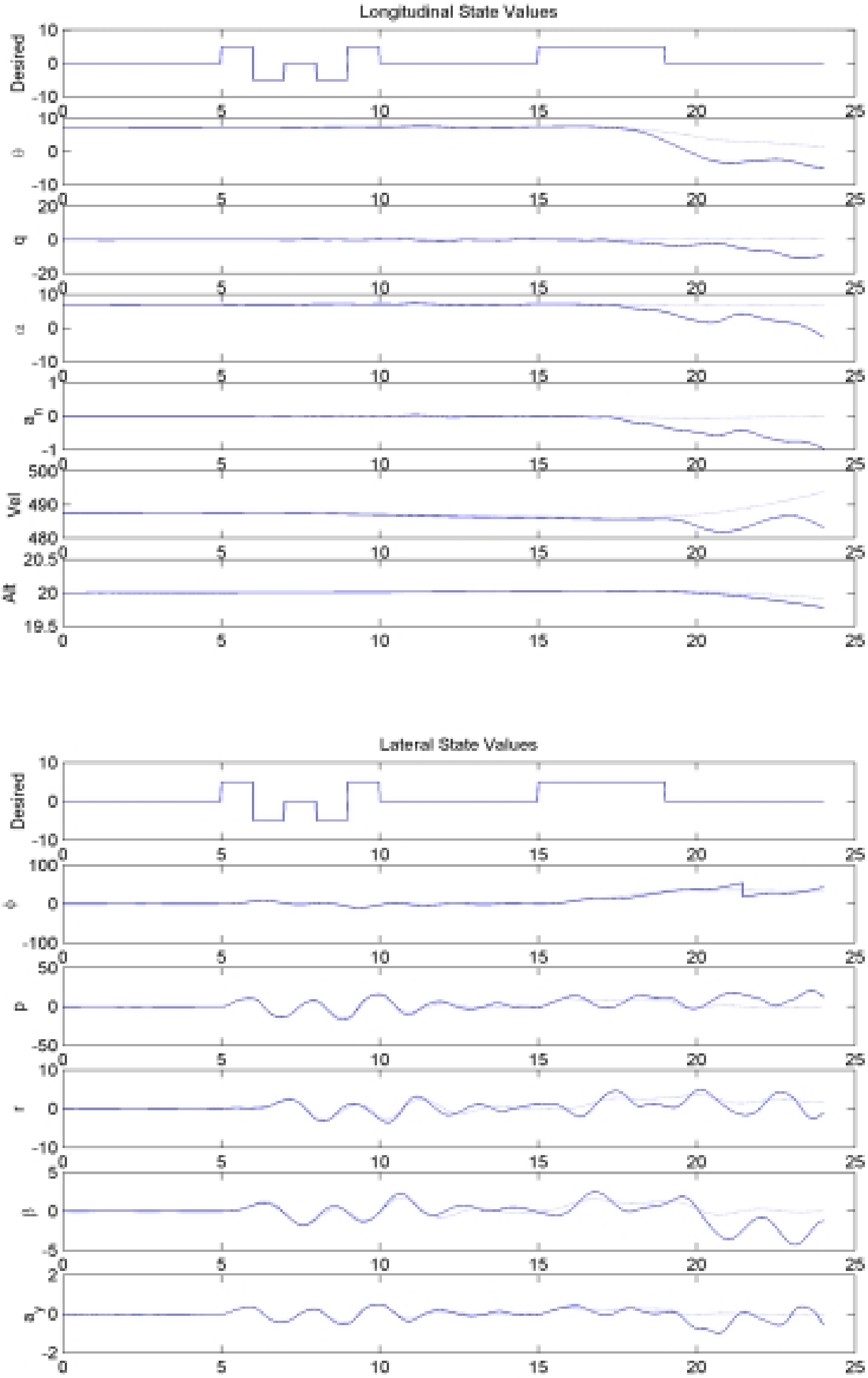


Figure 76c: State Values, Rudder Failure, ε = 75% Using ε $_{HAT}$

5.5.1 MMAE/FS/CR Performance with Full Actuator/Surface Failures

These figures show the performance of the MMAE/FS/CR used to detect and estimate flight control actuator failures and apply the appropriate control to achieve the desired control performance. In Section 5.4, the best achievable CR performance was analyzed showing the control performance based on the MMAE/FS used to detect the failure and ε_{true} used by the artificially informed CR algorithm. In this case, the ε_{true} is replaced by ε_{hat} as estimated by the MMAE/FS. It is seen from these figures that the MMAE/FS/CR is able to track the desired maneuver properly in most cases.

The MMAE/FS/CR is affected as before by the lack of linear models representing the change in aircraft attitude. This is even more important in this case, since the models are used also to determine the estimate of effectiveness driving the CR algorithm. In the cases of complete actuator failures shown in Figures 57, 61, 65, and 69, the ε_{hat} value is correct through most of the simulation, leading to the proper redistribution of control by the CR algorithm. This can be seen in the state value plots of these figures. At the end of the simulation, these figures also show the degradation of control from that seen in Section 5.4 caused by the poor ε_{hat} estimate due to the change in attitude.

5.5.2 MMAE/FS/CR Performance with Partial Actuator/Surface Failures other than Rudder

The partial actuator failures shown in Figures 58 - 60, 62 - 64, 66 - 68, and 70 - 72 show the change in attitude effects clearly. The elemental probabilities show the poor performance at the end of the simulation with the aircraft in a different attitude than used for the elemental filter's design (i.e., straight-and-level), and the state value plots show the degradation of CR performance due to the MMAE/FS estimates. This clearly shows the need for model scheduling based on the attitude of the aircraft, as well as on dynamic pressure.

The partial actuator failures also show improved ε_{hat} estimation performance in the 75% failure cases shown in Figures 60, 64, 68, and 72 in comparison to the ε_{hat} estimation performance from Section 5.3. This is most probably due to the additional maneuvering based on ε_{hat} changing in this case, as opposed to the constant ε_{true} value used in Section 5.3. Since ε_{hat} changed in this case, the CR based on the MMAE/FS-produced ε_{hat} value causes the aircraft to maneuver slightly more, making the 75% effectiveness case identifiable to the MMAE/FS.

The partial stabilator actuator failures also show a constant roll rate once the CR algorithm is initiated. This roll rate can be seen in Figures 58 - 60 and 62 - 64. This is caused by the mechanical biases in the simulation. The stabilators and flaperon positions are not equal to zero during straight-and-level flight due to the design of the aircraft. The difference between the zero position and the flight surface position during straight-and-level flight, for this research, is the mechanical bias. The CR uses the ε_{hat} to account for these biases, but if the ε_{hat} is not exact, then some of the bias will remain and cause the aircraft to roll as shown in these figures. In Section 5.4, this adverse roll was not present in the state value plots given partial stabilator failures because the ε_{true} was used instead of the MMAE/FS estimate ε_{hat}.

In Figures 67 and 71, it can be seen from the roll angle in the state value plots that the roll angle jumps across the desired control performance (fully-functional aircraft performance). This is caused by the inaccurate estimation of the partial actuator/surface failure effectiveness by the MMAE/FS. Since the estimate of effectiveness is not correct, the control applied to the aircraft is not the appropriate control for the failure conditions. As a result, the Block 40 Flight Control System (FCS), through the MMAE reconstructions of the sensor measurements (best estimates of sensed variables as if there were no sensor failures), detects the lack of good control performance. The Block 40 FCS then increases the gain in the flaperon channels, resulting in the jump in the roll angle state value. Even though the state value jumps, the overall performance is similar to the fully-functional aircraft performance with respect to the state values.

5.5.3 MMAE/FS/CR Performance with Rudder Failures

The rudder failure cases also showed performance similar to the results obtained in Section 5.4. Since the control authority required could not be achieved with the remaining control surfaces on the aircraft, the remaining control surfaces were driven to saturation, resulting in the poor control performance of the CR algorithm. This can be seen in Figures 72 - 74. It is interesting to note that in Figure 75, the 75% effective partial rudder failure case, enough control authority remained to track the desired maneuvers properly. The jumps in the roll angle state values are caused by the Block 40 FCS detecting the lack of appropriate control (in this case, caused by the lack of control authority), as mentioned in the partial actuator/surface failure case. Unfortunately, this is not a true representation of the performance of the CR, since the reason the CR does a decent job of following the desired maneuver is because the ε_{hat} is incorrect from the MMAE/FS. The errors in ε_{hat} are brought on by the attitude changing, altering the redistribution gains, which then do not drive the actuators into saturation. Section 5.4 shows the performance of the 75% rudder partial actuator failure with ε_{true} and there is not enough control authority to achieve the desired maneuvers using CR.

5.6 Chapter Summary

This chapter demonstrated the performance of the MMAE/FS to detect full and partial actuator failures and complete sensor failures. This chapter also demonstrates the shortcomings of the current design of the MMAE/FS and suggests some possible solutions to enhance the detection capability, such as model scheduling based on aircraft attitude, as well as on dynamic pressure. This section also demonstrated the control performance of the CR to apply proper control to an aircraft with a fully or partially failed actuator in order to accomplish a desired maneuver, in the case in which the effectiveness of the partially failed actuator/surface is artificially assumed to be known perfectly, in order to demonstrate the best performance one could hope to achieve from the CR algorithm. Finally, this chapter presented the complete MMAE/FS/CR algorithm and its ability to detect complete and partial failures and estimate the effectiveness of the partial failures, and using this information to apply CR to correct for the failed actuators in order to accomplish a desired maneuver.

Chapter 6 - Conclusions and Recommendations

6.1 Chapter Introduction

This chapter presents the conclusions and recommendations for this research effort based on the achieved results. Section 6.2 presents the conclusions based on the failure detection performance of the MMAE/FS, while Section 6.3 presents the conclusions based on the control redistribution performance of the CR. Section 6.4 presents future applications of this research in the fields of Uninhabited Aerial Vehicles (UAVs). Finally, Section 6.5 presents the recommendations for further research in this field.

6.2 Failure Detection Conclusions

The results of this research effort have shown the need for model scheduling based on dynamic pressure and aircraft attitude. The performance of the MMAE/FS can also be improved by making the elemental filters' dynamic noise strength and measurement noise covariance (**Q** and **R**) matrices adaptive to the maneuvers encountered. These additions will greatly increase the capability of the MMAE/FS to detect failures while undergoing pilot commanded maneuvers. This enhanced detection is important, especially in the case of partial failures. It can be seen from the results that, due to the less accurate estimations of effectiveness, the proper control was not always applied. If the linear models are enhanced, then the MMAE/FS will provide better estimates during partial actuator failures. The results of this research effort have also shown the detection performance using a purposeful set of pilot induced commands.

The application of the MMAE/FS to detect failures while maneuvering can benefit from adaptive **Q** and **R** matrices. In this research, the original **Q** and **R** matrix values [11,12] are used. These values were determined based on non-maneuvering flight. During maneuvering flight, unmodelled dynamics can appear in the residuals. These unmodeled dynamics can appear as sensor failures, causing false declarations. One method of solving this problem is to change the

Q and **R** matrix values to compensate for the maneuver [18,19,29]. The **Q** matrix is changed in order to compensate for the changing aircraft attitude, since the models are based on straight-and-level flight conditions, and thus the problem is inadequacies of the assumed linear perturbation *dynamics* model. The **R** matrix is changed in order to compensate for the sensors detecting the unmodelled dynamics. Once the maneuver is completed, the **Q** and **R** matrices may be returned to their original values in order to improve the detection of failures. By using this method, failure detection and estimation of the MMAE/FS can be improved, which also improves the estimate of effectiveness in the case of partial actuator failures, allowing for better CR response.

The current research has also shown the need for model scheduling based on the dynamic pressure and attitude of the aircraft. The model scheduling based on dynamic pressure has been added in this research effort and has been shown to enable the MMAE/FS to detect failures and maintain good estimation capabilities while transitioning the flight envelope (changing of both airspeed and altitude). Previous research efforts [9,10] showed that even small deviations in dynamic pressure incapacitate the failure detection capability of the MMAE. It can also be seen from this research that, when the aircraft is in a different attitude than straight-and-level, the elemental filters based on the straight-and-level flight condition no longer produce good failure detection and estimation capabilities. This demonstrates the need for model scheduling based not only on dynamic pressure, but also on the attitude of the aircraft.

An alternative method for improving the performance of the MMAE/FS to detect failures besides a purposeful dither signal is the application of pilot-induced maneuvers. In previous research [11,12], dither is used to excite the aircraft to allow the MMAE/FS to detect failures in the absence of pilot-commanded maneuvers. The design problem with the dither signal is having a large enough command magnitude to detect failures while remaining subliminal to the pilot. An alternative to using the dither signal is using a set of automated commands initiated by the pilot to check the failure status of the aircraft. Since the pilot initiates the commands, there is no need for the commanded maneuvers to remain subliminal. This allows larger command magnitudes to be used to increase the failure status identifiability of the MMAE/FS. Both methods can also be combined in implementation. The dither signal can be used to detect failures automatically, while giving the pilot the capability to initiate the automated sequence of commands to detect and identify failures not detected using the subliminal dither signal.

6.3 Control Redistribution Conclusions

This research showed the ability of the CR to apply the correct alteration to the Block 40 flight control system commands in order to achieve a desired maneuver in the face of actuator/surface failures. The CR tracked the desired maneuver in most cases with the MMAE/FS used to declare actuator/surface failures and when the effectiveness of a partially failed surface is exactly known. The case where the effectiveness of a partially failed surface is exactly known is used to demonstrate the best possible control capability of the CR. The CR has shown its capability to maintain desired control while using long duration maneuvers and while transitioning the flight envelope. The CR is extremely dependent upon the effectiveness in the case of partial failures, as can be seen in the results when the MMAE/FS provide both the declaration of an actuator/surface failure and the estimate of actuator/surface effectiveness. In order to achieve the desired performance from the CR, the MMAE/FS or other algorithm used for the detection and estimation of actuator failures must supply a reliable and accurate declaration of actuator failure and estimate of actuator effectiveness.

One point where the CR cannot apply the appropriate control is in the case of rudder failures as shown in Section 5.3.6. Since the F-16 aircraft only has one rudder, the other flight control surfaces must work extremely hard in order to achieve the desired performance given a rudder failure. In this case, the other control surfaces do not provide enough control authority to compensate for the rudder failure. Since the Block 40 flight control system is being told that it is working with a fully functional aircraft, detecting the lack of yaw caused by the failed rudder, increases the gain to the rudder. This gain is redistributed to the other surfaces but even with full detection, the desired yaw moment is not achieved. One method of handling this case in actual implementation is not to apply CR to the rudder failure case. The pilot should be alerted to the rudder failure. This procedure is acceptable in the view of this author because the rudder control is not essential for this type of aircraft. In other aircraft that have two vertical stabilators with two rudders, CR should be employed in the case of a single rudder failure since another redundant flight control surface is available.

The CR performance is heavily dependent on the algorithm used for failure detection and partial actuator/surface estimate of effectiveness (in this research, the MMAE/FS). If the declaration from the detection algorithm is not correct (false declaration or no declaration), or if the estimate of actuator/surface effectiveness is not accurate, the CR will not apply the proper control to accomplish the desired pilot-commanded maneuver. Unfortunately, as shown in this research, if the proper control over the aircraft is not accomplished, the Block 40 Flight Control System (FCS) will detect it through the estimates of the sensor measurements and improperly increase the command magnitudes to the actuators/surfaces in an effort to achieve the desired performance. The Block 40 FCS is unaware of the actuator/surface failure and the increase in command magnitude is redistributed, causing a jump in the state values. In this research, the jumps in state values particularly occurred in the roll angle state values and were typically less than a ten-degree jump in roll angle. The pilot will notice this but the jump in roll angle will not damage the aircraft. A method that might solve this problem is to redefine the redistribution matrix using the weighted pseudoinverse [23] rather than the unweighted pseudoinverse as used in previous research [32,33]. With the weighted pseudoinverse, penalties in the form of a cost function can be placed on the use of operating control surfaces used to compensate for the failed control surface. In the case of a flaperon failure, it may be beneficial to use the stabilators (differentially) rather than the rudder. This can be done by increasing the cost of using the rudder with respect to the cost of using the stabilators in the weighted pseudoinverse cost function.

Using the weighted pseudoinverse gives the designer more control over which flight control surfaces are used to compensate for a failed flight control surface.

Two methodologies of partial actuator failure control are examined here. The first method is simply to increase the gain to the partially failed actuator and the second method is to apply CR based on the effectiveness of the partially failed actuator. These methods are discussed in Section 4.5.4 and the later control logic (CR) is used based on that discussion. The first method results in the aircraft perfectly following a desired maneuver assuming there is enough control authority left in the partially failed actuator/surface. For example, if an actuator has an $\varepsilon = 25\%$, increasing the gain by a factor of four to achieve the desired maneuver might not be possible. The issue with simply increasing the gain to the partially failed actuator/surface is that the exact reason for the failure may not be known and increasing the gain could cause more problems with the aircraft, including disabling other flight control actuators/surfaces. An example of this case is a hydraulic leak on an actuator. If the gain is increased to the actuator/surface, this could cause the hydraulic leak to drain the hydraulic fluid from the aircraft which then disables other flight control actuators. In the opinion of the author, the best method to handle partial failures based on current aircraft technology is through the use of CR as shown in the results. The CR still uses the partially failed actuator but does not increase the gain to that actuator so as not to cause potentially further damage to the aircraft. Increasing the gain to compensate for a partial actuator failure may be appropriate for some aircraft, such as fly-by-power aircraft in which electronic servomotors are used to drive the flight control surface.

There are some of limitations of CR. These include the drag induced by the control redistribution and the control authority. CR uses other functional actuators to compensate for a failed actuator. By using other actuators, the amount of induced drag created by the flight control surfaces is increased. This is a limitation since now more thrust is required in order to compensate for the increased application of control surfaces. Another limitation of the CR is the control authority required to compensate for a failed actuator. In this research, simple pitch and roll commands are used to show the performance of the CR. In actual use, combined large pitch and roll commands are employed. This can quickly overwhelm the CR with a complete actuator failure. There may not be enough control authority in these cases to provide proper control of the aircraft given a complete actuator failure. This is especially important in the case of multiple actuator failures.

6.4 Applications for UAV

Through this and previous research efforts, the MMAE/FS/CR has been shown to detect complete and partial flight control actuator failures and complete sensor failures without direct pilot interaction. The MMAE/FS has been shown to detect failures in both the flight control actuators/surfaces and sensors. In the case of partial flight control actuator/surface failures, the MMAE/FS creates an estimate of effectiveness, as well as the failure declaration. The declaration and estimate of effectiveness are then passed to the CR algorithm to redistribute the control authority properly to other operational flight control actuators/surfaces to compensate for a completely or partially failed flight control actuator/surface. The MMAE/FS is implemented without changing the air vehicle's flight control system. The sensor information is used to update the MMAE/FS, and then the MMAE/FS produces a state estimate, which is fed into the flight control system. This information, with the pilot commands, is also used to determine sensor and flight control actuator failures. The CR algorithm is implemented at the output of the onboard flight control system and adjusts the surface commands produced by the flight control system in order to redistribute the commands to operational flight control actuators in the case of partial or complete flight control actuator failures. The MMAE/FS/CR algorithm has been proven through various flight conditions and failure cases, and the actual implementation of the MMAE/FS/CR does not consist of major software changes to the onboard flight control system itself.

The purposeful dither is a sinusoidal command to the flight control system used to excite the aircraft in the absence of pilot-commanded maneuvers. On piloted aircraft such as the VISTA

F-16 aircraft, the magnitude of the purposeful dither signal is kept to a minimum so as not to bother the pilot. Unfortunately, a low magnitude dither signal may not be able to yield complete detection and identification of all failure modes. For UAV applications, the purposeful dither signal magnitude can be larger in order to excite the aircraft in the absence of controller commands and thus enable detection of most if not all failure modes, since there is no pilot onboard. This condition improves the detection performance of the MMAE/FS while the aircraft is not maneuvering.

The major advantage of using the MMAE/FS/CR on a UAV aircraft is that, if the aircraft is damaged during a mission, the MMAE/FS can detect and evaluate the damage and inform the controller, and the CR can compensate for the damage (assuming enough control authority is remaining in the other flight control surfaces), allowing the controller to return the UAV to base for repairs. This capability can save assets and the costs associated with them, such as the cost to replace a damaged UAV that had to be destroyed. Moreover, it can increase the chances of accomplishing missions since the controller will have better information on the status of the UAV and can make informed decisions whether to continue a mission with a slightly damaged UAV or return the UAV to base to effect repairs.

Future advances are available in the area of MMAE/FS/CR application on UAV aircraft. Filter design models can be altered and new algorithms made for tailless aircraft technology. Providing the MMAE/FS with more sensor information, such as inertial navigation system information, will improve the detection and estimation capability of the MMAE/FS. The use of a UAV software simulation platform would greatly improve the capability to test the MMAE/FS/CR on UAV platforms. These are some of the improvements that can be made to the MMAE/FS/CR with respect to the application to UAV aircraft.

This section presented the advantages of implementing a MMAE/FS/CR algorithm on a UAV platform. This section briefly presented the implementation of the MMAE/FS/CR algorithm on a UAV, and the advantage of being able to increase the magnitude of the purposeful dither signal to improve the detection and estimation capability of the MMAE/FS. The ability of the MMAE/FS to provide more information concerning the failure status of the UAV to the controller and for the CR to compensate for the failures (assuming a flight control actuator/surface failure) to allow the UAV to continue with the mission or to return to base to effect repairs is also addressed in this section. Finally, the opportunities for further advancement in the area of applying MMAE/FS/CR to UAV aircraft are presented in this section.

6.5 Recommendations

In order to improve the performance of the MMAE/FS/CR, there are several recommendations for further research that can be exploited. These recommendations are the application of dynamically changing **R** and **Q** matrix values, model scheduling based on aircraft attitude, detection and control of multiple failures, increasing the number of sensor sources, and updating the simulation.

As mentioned earlier, using dynamically changing **R** and **Q** matrices as a function of a commanded maneuver will allow for better detection during maneuvering flight. Research into an adaptive **Q** matrix for use within an MMAE algorithm, in order to enhance the distinguishability of assumed parameter values (hypothesized failure status and ε values) has been completed previously in both the continuous-time case and then in the discrete-time case [18,19, 29]. An adaptive **Q** and **R** matrix has not yet been applied to this case but may result in enhanced failure detection capabilities and fewer false alarms during maneuvering flight.

A second area that may be expanded is the model scheduling based on aircraft attitude. As the aircraft changes attitude, the linear elemental filters based on the trimmed, straight-and-level flight condition become inaccurate, degrading the detection capability. Filters based on the actual aircraft attitude using a lookup table for filters based on high bank and pitch angles, and linear interpolation between these and the straight-and-level flight condition, would enhance the detection and control characteristics of the MMAE/FS/CR.

A natural next area of investigation is to allow the MMAE/FS/CR to detect multiple failures and attempt to provide the proper control. In the current state of the software, the detection and control redistribution can only handle one sensor or actuator failure. The “Level 2” banks of filters that are used to handle dual failures have not been completely defined in the software as yet but the existing architecture can conceptually handle this simulation [7]. This will also bring up the limitation of the CR to handle dual actuator failures. The amount of control authority may not be able to handle a dual actuator failure.

The fourth area is the addition of more sensors to enhance the failure detection and estimation. This research uses the bare minimum sensors to provide the observability to detect failures and to provide adequate sensor information to the flight control system given a sensor failure. Other sensors can be added, such as inertial navigation sensors (accurate pitch, roll, and yaw angles) and actuator position sensors. This will greatly enhance the capability of the MMAE/FS/CR to detect partial and complete actuator failures and complete sensor failures. This will also enhance the MMAE/FS/CR capability to create an estimated sensor output to the flight control system if one of the sensors has actually failed. This will also improve the MMAE/FS capability to provide a reliable and accurate estimate of effectiveness to the CR, which will then result in better CR performance during partial actuator failures.

The final area of improvement is the expansion of the software to a new aircraft such as UAVs, tailless aircraft, or other aircraft in general. The current simulation is limited to an old SUN computer system. New simulation tools based on current computer technology representing modern aircraft (with different geometries, additional actuation surfaces, more and better sensors) and their associated flight control systems should be used in order to move the MMAE/FS/CR closer to actual aircraft implementation. The use of modern computers and technology will also make the simulation more convenient although the current VISTA F-16 simulation ‘does the job’.

6.6 Chapter Summary

This chapter provides the reader with the conclusions for the failure detection and control redistribution performance in this research effort. This chapter also provided discussions on possible improvements to the MMAE/FS/CR. A discussion of the application of MMAE/FS/CR onto UAVs is also presented. Finally, this chapter presented the possible areas of exploitation in order to move this research closer to actual implementation.

Appendix A: VISTA F-16 SRF General Information

A.1 Variable-stability In-flight Simulation Test Aircraft (VISTA)

The Variable-stability In-flight Simulation Test Aircraft (VISTA) is a modified F-16D aircraft used to simulate various flight control designs. The aircraft is currently used as a test aircraft at Edwards Air Force Base, California. The VISTA F-16 flight control system is composed of the original Block 40 Flight Control System (FCS) and the Variable Stability system (VSS) that allows the VISTA F-16 to act like other aircraft with respect to their control performance and characteristics. This can be accomplished with the F-16 aircraft since the FCS is based on the Fly-By-Wire design, meaning there is no physical contact between the pilot's flight controls and the flight control actuators.

Another advantage to the VISTA F-16 aircraft is the capability to test various flight control systems and be able to switch back to the proven Block 40 FCS if a problem is discovered with the FCS under test. This ensures the safety of the pilot and crew while accomplishing tests on new flight control designs. The VSS of the VISTA F-16 can also duplicate flight control failures in order to test the FCS capability of controlling a damaged aircraft.

These advantages make the VISTA F-16 a uniquely useful test aircraft for the initial implementation of the Multiple Model Adaptive Estimator with Filter Spawning and with Control Redistribution (MMAE/FS/CR). The disadvantage of using the VISTA F-16 is the single rudder surface. During a rudder failure, there is not enough control authority in the other flight controls to command a yaw to make up for the failed rudder. The MMAE/FS/CR works best when there is enough redundancy in all flight control surfaces to allow the failure in one flight control surface and still maintain control over the aircraft.

A.2 Flight Control System

This section presents a discussion of the Flight Control System (FCS) of the VISTA F-16 aircraft with and without the proposed implementation of the MMAE/FS/CR. The actual aircraft FCS contains various sensors and hardware linked together using the MIL-STD-1553B avionics databus. The Block 40 FCS then uses the information collected from the sensors and pilot commands entered through the flight control stick and rudder pedals to determine the proper flight control actuator commands required to accomplish the desired maneuver. These commands are then electronically sent to hydraulic actuators that move the flight control surfaces. The VSS is an addition to the original F-16D aircraft that can adjust the electronic signals going to the actuators in order to duplicate various types of control systems. MMAE/FS/CR does not require the use of the VSS so it will not be discussed in this section.

The illustration below represents the original Block 40 FCS on an F-16D aircraft. This illustration does not depict all the databus and sensor connections for simplicity.

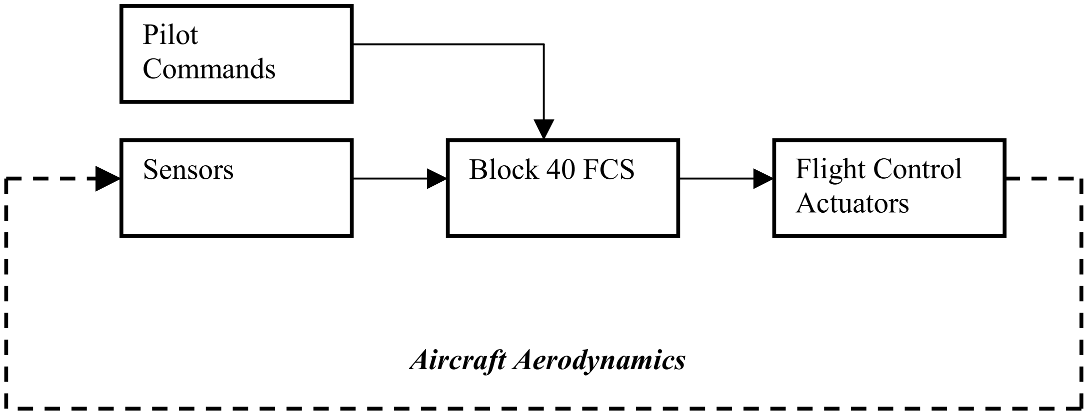


Figure 77: Original F-16D Flight Control System

In order to apply the MMAE/FS/CR, several changes had to be made to the original FCS. The first change is the insertion of MMAE/FS algorithm between the sensors and the Block 40

FCS. This allows the MMAE/FS access to the sensor information to update the elemental filters within the MMAE/FS. The MMAE/FS then produces an estimate of the sensor information (i.e., a best reconstruction of sensor measurements as if there were no sensor failures), which is passed on to the Block 40 FCS. The next change to the original FCS is the insertion of the CR algorithm after the Block 40 FCS and before the flight control actuators. This allows the CR algorithm to redistribute the original Block 40 FCS commands based on the declared failure. The MMAE/FS is connected to the CR to provide the actuator failure declaration, as well as the estimate of actuator effectiveness (ε). A loop from the original Block 40 FCS outputs back to the input of the FCS had to be added to allow the Block 40 FCS to believe that its commands are being implemented even though a flight control failure may occur and the CR is redistributing the command signal. The new FCS with the MMAE/FS/CR implementation is shown below.

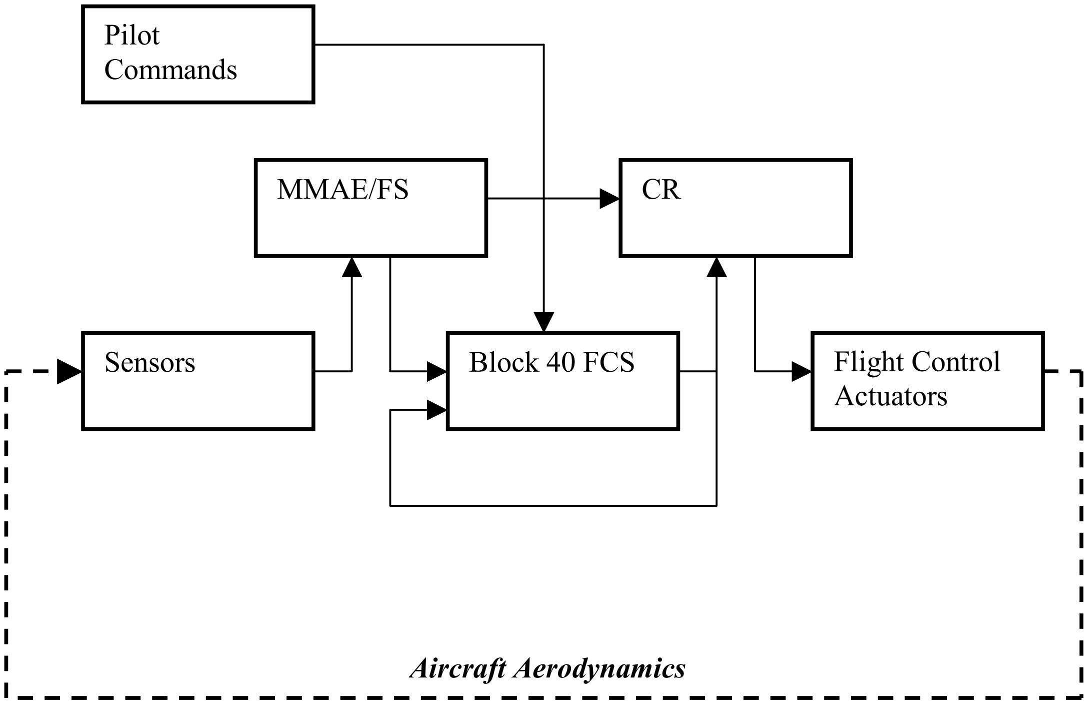


Figure 78: F-16D Flight Control System with MMAE/FS/CR Implementation

The advantage of MMAE/FS/CR is that it uses all of the original systems and does not require major changes in the current avionics system. The sensor data is intercepted by the MMAE/FS and then the estimates are sent to the original Block 40 FCS. The Block 40 FCS then sends the flight control actuator commands that the CR intercepts. The CR then sends the redistributed flight control commands to the flight control actuators.

A.3 VISTA F-16 Simulation Rapid-prototyping Facility

This section describes the VISTA F-16 Simulation Rapid-prototyping Facility (SRF) simulation tool and its operation with the MMAE/FS/CR implemented. The SRF is a nonlinear, six-degree-of-freedom simulation of the VISTA F-16 aircraft. The simulation contains added Dryden Wind Tunnel noise and employs 4th order actuator models. Calspan and General Dynamics developed the simulation [11]. The Air Vehicles Laboratory at Wright-Patterson Air Force Base originally used the SRF simulation. The simulation tool is written in the Fortran 77 language and is limited to operation on an older Sun Operating System. Efforts are underway to transport the simulation software to a more modern operating system but the original source code is required for some of the called libraries. Unfortunately, Calspan and General Dynamics are not maintaining the simulation code anymore, so the source code may be completely lost, limiting the simulation to run only on the Sun Operating System.

The simulation is set up using a series of subroutines, each representing the operation of a particular section of the VISTA F-16 FCS. The algorithm for the simulation is shown in Figure79. The boxes show the functions with the actual subroutine name that is related to that function. The labels outside the boxes show the variable latch names that carry the data from one subroutine to the other.

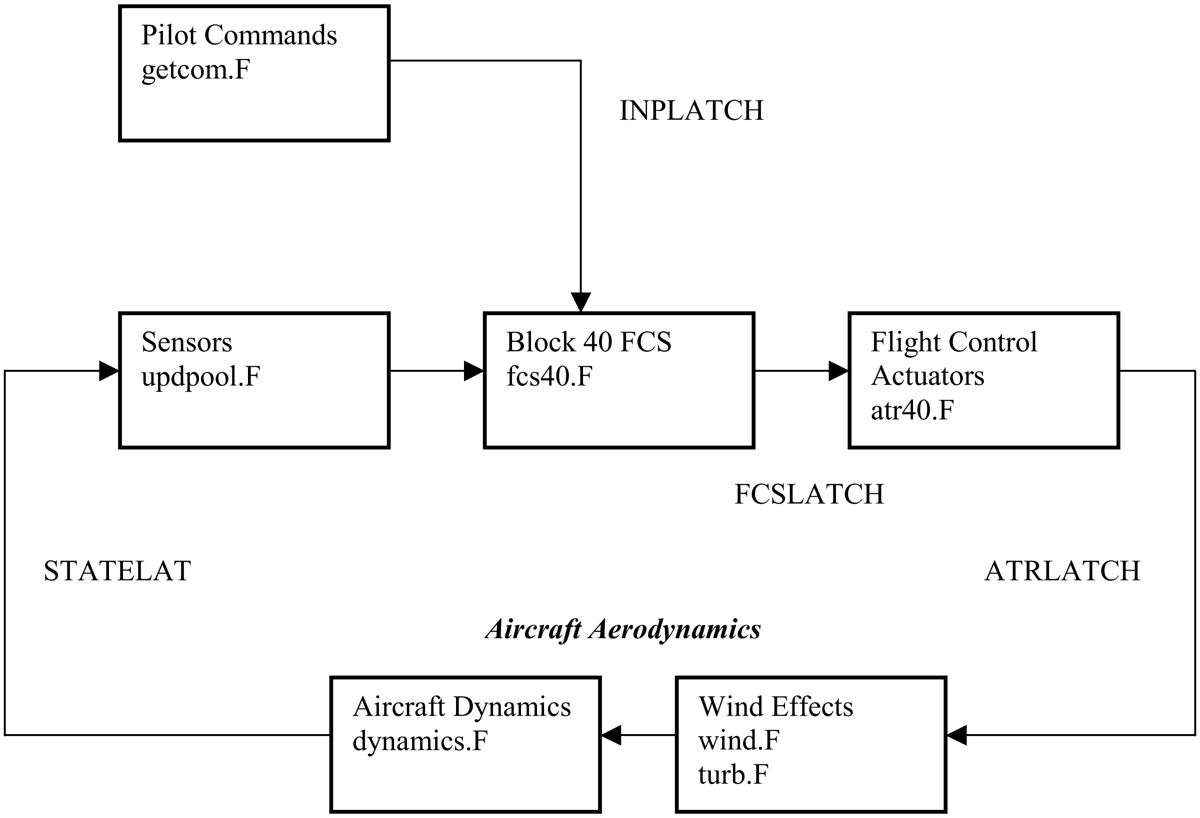


Figure 79: Original VISTA F-16 SRF Algorithm

These are the primary subroutines and their functions. The FCSLATCH variables are used to transfer the commands from the Block 40 FCS to the actuators. The actuator models are applied and the actuator commands are saved to the ATRLATCH variables. The aircraft dynamics subroutines use the ATRLATCH variables to propagate the aircraft states forward in time. The states are then stored in the STATELAT variables. The pilot commands are stored in the INPLATCH variables to go to the Block 40 FCS. The entire simulation is saved in the file named vista.F and the simulation loop above is the subroutine simulate.F. The implementation of MMAE/FS/CR is shown in the algorithm In Figure 80. The same label formats are used.

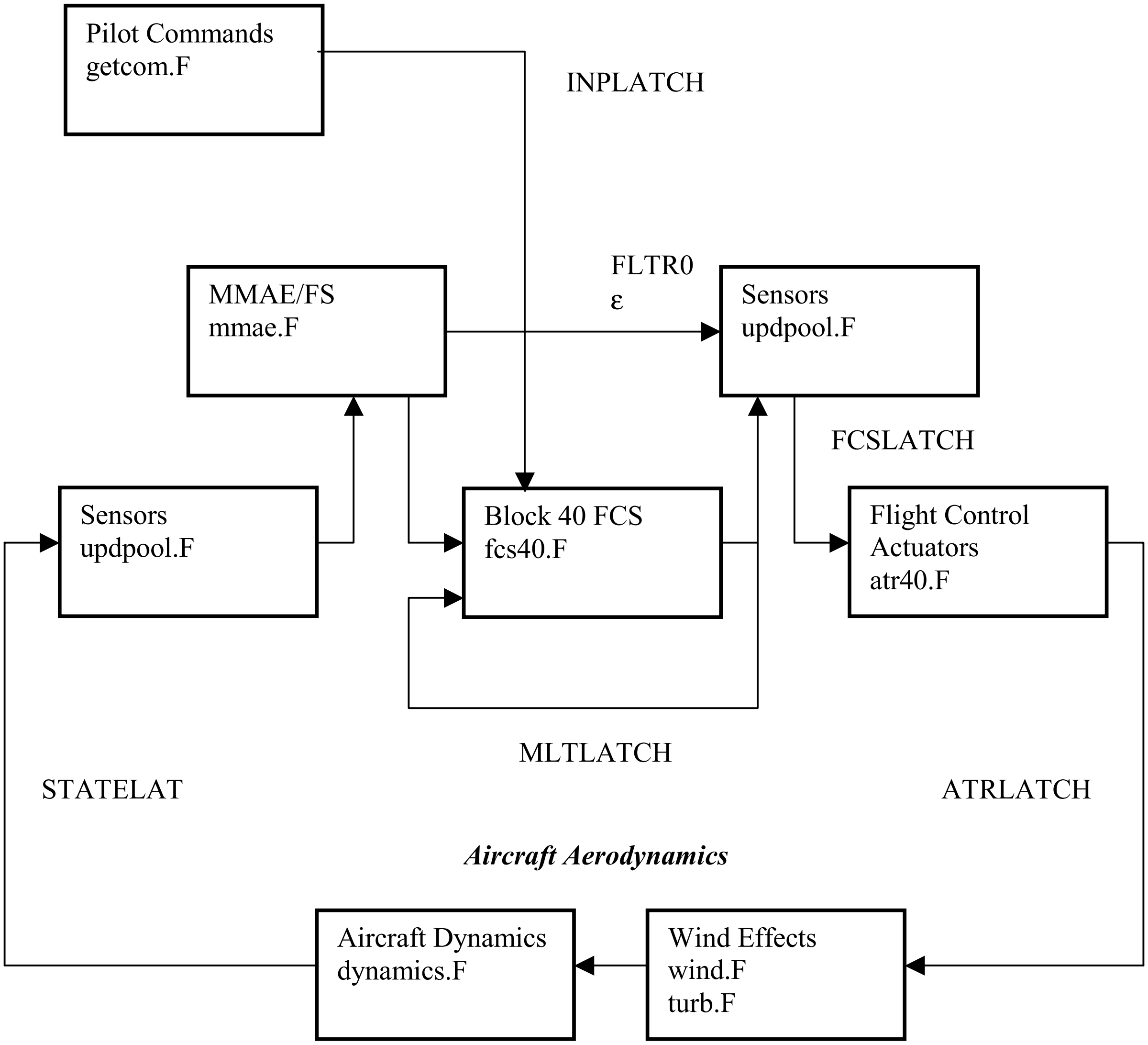


Figure 80: VISTA F-16 SRF with MMAE/FS/CR Implementation

The FLTR0 and ε variables are the actuator failure declaration and the estimate of effectiveness from the MMAE/FS. The MLTLATCH are the control variables that complete the FCS loop.

The simulation requires a parameter file in order to run the simulation. This parameter file contains all the settings (such as initial airspeed, altitude, flaps settings, and gear settings to name a few) and the pilot commands for the simulation. The Transportable Applications Executive (TAE) software creates the parameter file. The following table shows the standard settings for variables used in this research effort. All the other parameter values within the TAE setup menus can remain at the nominal values. At anytime, the user can type HELP and the name of the variable to get information on the variable as well as options where applicable.

TAE Menu	**Variable**	**Description**	**Nominal Value**
1 – Standard Parameters	stoptime	Simulation Duration (seconds)	5.0
	altitude	Initial Altitude (ft)	10000
	mach	Initial Airspeed (Mach)	0.7
	thrcntrl	Throttle Control Type (Constant, Manual, Auto)	"auto"
	commands(n)	Pilot Commands (pstep, rstep, pdoublet, rdoublet)	(null value)
	cmdmagn(n)	Magnitude of Pilot Commands (pounds)	(null value)
	cmdtime(n)	Start Time for Each Pilot Command (seconds)	(null value)
	cmddur(n)	Duration of Each Command (seconds)	(null value)
	mxfile	Name of MATRIXx File for Linear Model Generation	(null value)
3 – Wind Parameters	seed	Seed for Random Number Generation	1987

Table 4: TAE Menu Options

The altitude and Mach variables are used to set the initial simulation altitude and airspeed. Since linear models are used for this research effort, the thrcntrl variable should be set to "constant", meaning the throttle control is not changed through the simulation. A throttle control setting of "auto" allows the simulation to adjust the throttle to maintain the trim airspeed and a setting of "manual" allows the throttle to be manually adjusted through the commands(n) variable. The commands(n) variable is used to insert up to five pilot commanded maneuvers into the simulation. The commands are pitch and roll step (pstep and rstep) and pitch and roll doublets (pdoublet and rdoublet). Other pilot commands are available and can be accessed through the HELP menu. The cmdmagn(n) variable sets the magnitude of the commanded maneuver in pounds of force. The stick force limits are shown in the HELP menu. The cmdtime(n) variable is the commanded maneuver start time and the cmddur(n) is the duration of the commanded maneuver. Both of these variables are in seconds.

The mxfile variable is used to generate the MATRIXx file for use in generating the linear filters based on the failure hypotheses. The MATRIXx file should be saved as srfdat.m. This file contains the linear model parameter values, including all the time conditions, based on the initial airspeed and altitude settings above. The VISTA simulation must be run from the TAE software in order to generate the srfdat.m file. The final setting in the TAE software relevant to this research is the seed variable. The seed variable is used in the zero-order Dryden noise applied to the simulation. The turbulence routines used by the original VISTA software are not used. The updated Dryden noise can be removed through a setting in the FLAGS.DAT file. The seed variable defines which random sequence to use in the simulation. This also allows random sequences to be repeated in the case of various Monte Carlo analyses.

Bibliography

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

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

[3] Baram, Y., and N. R. Sandell Jr., "Consistent Estimation of Finite Parameter Sets with Application to Linear System Identification," *IEEE Transactions on Automatic Control*, AC-23(3), 451-454 (1978).

[4] Baram, Yoram. *Information, Consistent Estimation and Dynamic System Identification*. Ph.D Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, November 1976.

[5] Blackelock, John H. *Automatic Control of Aircraft and Missiles.* New York: John Wiley & Sons, Inc., 1991.

[6] 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).

[7] Clark, Curtis S. *Multiple Model Adaptive Estimation and Control Redistribution Performance on the VISTA F-16 During Partial Actuator Impairments*. MS Thesis, AFIT/GE/ENG/97D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1997.

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

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

[11] Fisher, Kenneth A. *Multiple Model Adaptive Estimation Using Filter Spawning*. MS Thesis, AFIT/EN/ENG/99M-09, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1999.

[12] Fisher, Kenneth A., and Peter S. Maybeck. "Multiple Model Adaptive Estimation with Filter Spawning," *Proceedings of the American Control Conference*, Chicago, IL, 2326-2331 (June 2000).

[13] Fry, C. M., and A. P. Sage "On Hierarchical Structure Adaptation and System Identification," *International Journal of Control*, 20(3):433-452 (September 1974).

[14] Hanlon, P. 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.

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

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

[17] Lewis, Robert W. *Multiple Model Adaptive Estimation and Control Redistribution for the VISTA F-16*. MS Thesis, AFIT/GE/ENG/96D-29, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1996.

[18] Lund, Eivind J. *On-Line Discrimination and Estimation in Multiple Regime Systems*. Ph.D Dissertation, Division of Engineering Cybernetics, The Norwegian Institute of Technology, University of Trondheim, June 1992.

[19] Lund, Eivind J., Jens G. Balchen and Bjarne A. Foss, "Multiple Model Estimation with Inter-Residual Distance Feedback," *Modeling, Identification, and Control*, Vol. 13, No. 3, pp. 127-140, 1992.

[20] Magill, D. T. "Optimal Adaptive Estimation of Sample Stochastic Processes," *IEEE Transactions on Automatic Control*, AC-10(5):434-439 (October 1965).

[21] Maybeck, Peter S. *Stochastic Models, Estimation, and Control, I*. New York: Academic Press, Inc., 1979. Republished – Arlington, VA: Navtech, 1994.

[22] Maybeck, Peter S. *Stochastic Models, Estimation, and Control, II*. New York: Academic Press, Inc., 1982. Republished – Arlington, VA: Navtech, 1994.

[23] Maybeck, Peter S. *Stochastic Models, Estimation, and Control, III*. New York: Academic Press, Inc., 1982. Republished – Arlington, VA: Navtech, 2001.

[24] Maybeck, Peter S., and Donald L. Pogoda. "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," In *Proceedings of the 28th Conference on Decision and Control*, 1566-1572 (December 1989).

[25] 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).

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

[27] 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.

[28] 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).

[29] Miller, Mikel M. *Modified Multiple Model Adaptive Estimation (M^3AE) for Simultaneous Parameter and State Estimation*. Ph.D Dissertation, Air Force Institute of Technology, AFIT/DS/END/98-02, Wright-Patterson AFB, OH, March 1998.

[30] *MIL-STD-1797A Flying Qualities of Piloted Aircraft*, January 1990.

[31] Pogoda, 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.

[32] Stepaniak, Michael J. *Multiple Model Adaptive Control of the VISTA F-16*. MS Thesis, AFIT/GE/ENG/95D-04, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.

[33] Stepaniak, Michael J., and Peter S. Maybeck. "MMAE-Based Control Redistribution Applied to the VISTA F-16," *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1249-1260 (October 1998).

[34] Stevens, 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.

[35] Tugnait, J.K. "Comments on 'State Estimation for Discrete Systems with Switching Parameters'," *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(3):464 (May 1979).

Vita

Captain Michael L. Torres Jr. graduated from Lake Weir High School in Ocala, Florida in 1993. He received his Bachelor of Science degree in Electrical Engineering from Embry-Riddle Aeronautical University at Prescott, Arizona in 1997. At Embry-Riddle Aeronautical University, he was part of the AFROTC Detachment 28.

His first assignment was as an Avionics Systems Engineer working with the 84th Test and Evaluation Squadron located at Tyndall AFB, Florida. There, he worked avionics and electrical integration issues of systems under test and the QF-4E/G, F-15D/E, F-16C, and other modern day fighter/bomber aircraft. In 2000, he was accepted to the Graduate School of Engineering and Management, Air Force Institute of Technology where he is working on a Masters of Science degree in Electrical Engineering specializing in Stochastic Controls, Flight Controls, and Navigation Subsystems. After he graduates, he will be assigned to the National Air Intelligence Center at Wright-Patterson AFB.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 11-03-2002	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* Aug 00 – Mar 02

4. TITLE AND SUBTITLE

Failure Detection and Flight Control Redistribution
using Multiple Model Adaptive Estimation
with Filter Spawning

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Michael L. Torres Jr.
Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/02M-25

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

ASC/GRA
Attn: Dr. Brian Kowal
AMC-III
Bldg 558
2590 Loop Road West
WPAFB OH 45433-7142

DSN: 986-9436
e-mail: Brian.Kowal2@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In the current research, the MMAE with Filter Spawning and Control Redistribution (MMAE/FS/CR) are used together to identify failures and apply appropriate corrections. This effort explores the performance of the MMAE/FS/CR in different regions of the flight envelope using model and gain scheduling.

The MMAE/FS/CR is able to detect complete and partial actuator/surface failures, as well as complete sensor failures. Once the actuator/surface failure is identified and the effectiveness is determined in the case of partial failures, proper control is applied in order to accomplish the desired pilot command. Improvements in the algorithm are required in order to enhance the MMAE/FS/CR ability to detect failures while undergoing maneuvering flight.

This investigation shows the ability of the MMAE/FS to detect failures while transitioning through the flight envelope and while performing pilot commanded maneuvers. It also improves and demonstrates the CR ability to compensate for complete or partial actuator/surface failures.

15. SUBJECT TERMS

Multiple Model Adaptive Estimation, Kalman Filter, Reconfigurable Flight Controls, Adaptive Flight Controls, Sensor and Actuator Failure Detection

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Peter S. Maybeck, Civ (ENG)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4639; e-mail: Peter.Maybeck@afit.edu
U	U	U	UL	259	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18