MMAE DETECTION OF INTERFERENCE/JAMMING AND SPOOFING IN A DGPS-AIDED INERTIAL SYSTEM

THESIS

Nathan Alan White
Second Lieutenant, USAF

AFIT/GE/ENG/96D-21

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

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/96D-21

MMAE DETECTION OF INTERFERENCE/JAMMING AND SPOOFING IN A DGPS-AIDED INERTIAL SYSTEM

THESIS

Nathan Alan White
Second Lieutenant, USAF

AFIT/GE/ENG/96D-21

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

AFIT/GE/ENG/96D-21

MMAE DETECTION OF INTERFERENCE/JAMMING AND SPOOFING IN A DGPS-AIDED INERTIAL SYSTEM

THESIS

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

Nathan Alan White, B.S.E.E.
Second Lieutenant, USAF

December, 1996

Acknowledgements

First and foremost I must thank my Father in Heaven for patiently nurturing my soul and caring for my family while I pursued my studies at AFIT. These 18 months could not have been so fulfilling, both at home and at work, without His constant help. I must also thank my wife, Miriam. Her companionship enriches my life and makes me a better man than I could otherwise be. Her love and understanding made the long hours tolerable; with her, even the intensity of AFIT was sweet.

I was fortunate to work with many capable people at AFIT. I owe a big thanks to my advisors, Dr Peter Maybeck and Captain Stew DeVilbiss. Dr Maybeck, in particular, kept me excited about my research and helped me to mold a quality report when it was through. I am very lucky to have worked with him. Mr Stan Musick is credited with the operability of the FORTRAN code. This research very literally could not have been completed without my many discussions with him. The rest of the G&C class was a pleasure to work with. Captain Robert Lewis and Lieutenant Jason McKay were good study partners and great company. I thank you all.

Nathan Alan White

Table of Contents

List of Figures

List of Tables

Abstract

Previous research at AFIT has resulted in the development of a DGPS-aided INS-based precision landing system (PLS) capable of meeting the FAA precision requirements for instrument landings. The susceptibility of DGPS transmissions to interference/jamming and spoofing must be addressed before DGPS may be safely used as a major component of such a safety-of-flight critical navigational device. This thesis applies multiple model adaptive estimation (MMAE) techniques to the problem of detecting and identifying interference/jamming and spoofing failures in the DGPS signal. Such an MMAE is composed of a bank of parallel filters, each hypothesizing a different failure status, along with an evaluation of the current probability of each hypothesis being correct, to form a probability-weighted average output. Performance for a representative selection of navigation component cases is examined.

For interference/jamming failures represented as increased measurement noise variance, results show that, because of the good FDI performance using MMAE, the blended navigation performance is essentially that of a single extended Kalman filter artificially informed of the actual interference noise variance. Standard MMAE is completely unable to detect spoofing failures (modelled as a bias or ramp offset signal directly added to the measurement). This thesis shows the development of a moving-bank pseudo-residual MMAE (PRMMAE) to detect and identify spoofing failures. Using the PRMMAE algorithm, spoofing failures are very effectively detected and isolated; the resulting navigation performance is equivalent to that of an extended Kalman filter operating in a no-fail environment.

MMAE DETECTION OF INTERFERENCE/JAMMING AND SPOOFING IN A DGPS-AIDED INERTIAL SYSTEM

1. Introduction

1.1 Background

Aircraft are often required by necessity or bad luck to fly under adverse weather conditions. The purposes of military air operations are often best accomplished under such low-visibility conditions. Precise, safe landings are required even when lack of visibility would make landing by a human pilot impossible. The availability and integrity of a precision landing system is critical for safety of flight during low visibility conditions.

The Instrument Landing System (ILS) currently in use works on a relatively simple localizer/glide slope method. Guidance to the runway is provided by highly directional radio signals. The localizer provides the information needed for lateral guidance on the approach path. The glide slope signal provides the needed vertical guidance. Figure 1.1 shows a representation of the localizer and glide slope radiation patterns. The carrier frequencies used in the ILS are about 110MHz and 330MHz for the localizer and glide slope signals, respectively. The frequencies indicated on the diagram describe the modulation frequencies used. The effect of the ILS signals just described is to create a virtual funnel which will guide an approaching airplane to the runway, as is depicted in Figure 1.2.

The FAA has established clear precision requirements for instrument landings. These requirements constitute the performance constraints for the current ILS as well as for the replacement GPS-based precision landing system (PLS) examined in this thesis. The FAA requirements are shown in Category III required navigational

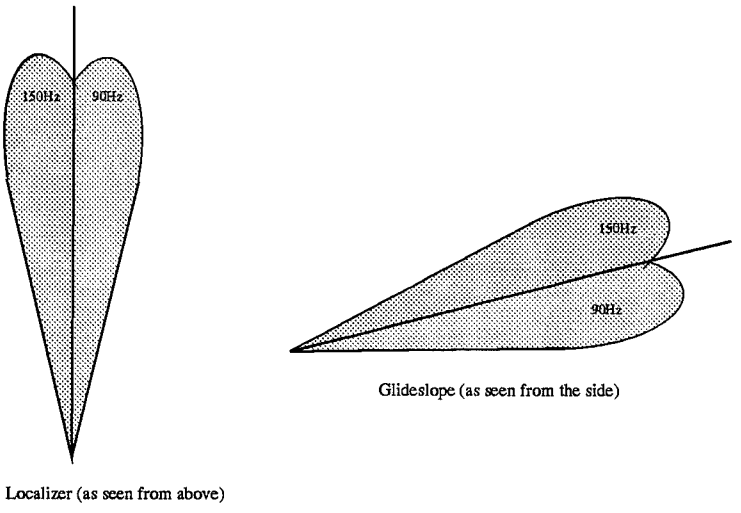

Figure 1.1 Localizer/Glideslope Method

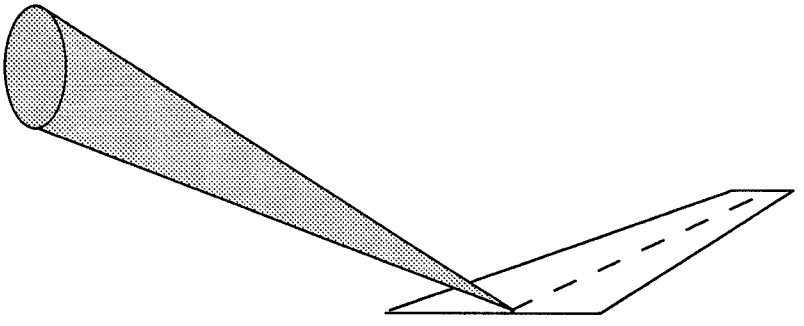

Figure 1.2 Funnel Approach

precision (at the 200 foot "decision height") is roughly depicted by the ellipse (two-sigma values) shown in Table 1.1. Figure 1.3. The accuracies listed under Category III represent the required precision for a zero visibility landing, wherein the pilot does not make visual contact with the runway until after touchdown. Conceivably, this precision would also make completely autonomous landings possible for unmanned vehicles or if the pilot became incapacitated. The localizer/glide slope ILS currently in use is capable of providing Category III accuracy. See Table 1.1.

The current ILS has two major drawbacks which have motivated the desire for an operationally certified GPS-based precision landing system (PLS). First, the

	Vertical Accuracy	Horizontal Accuracy
Category I	$\pm 4.1m$	$\pm 17.1m$
Category II	$\pm 1.7m$	$\pm 5.2m$
Category III	$\pm 0.6m$	$\pm 4.1m$

Table 1.1 Precision Landing System Two-Sigma Accuracies [3]

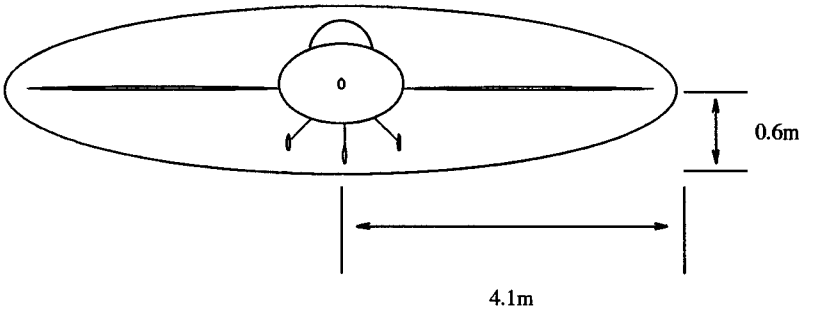

Figure 1.3 Category III Precision (Two Standard Deviations)

current ILS has a very limited area of usefulness (see Figure 1.2). The area of guidance is restricted to a straight path extended for several miles from the end of the runway. It would greatly contribute to the safety and efficiency of air operations if controllers were able to know the precise locations and courses of all airplanes in the controlled area. The current ILS has no capacity to contribute, in this manner, to the safety, guidance, or regulation of typical traffic patterns of aircraft in various stages of landing or taking off around an airfield. The second major drawback of the current ILS is the maintenance cost of the aging system. The Microwave Landing System (MLS) was originally thought to be the replacement for the current ILS. The MLS however, works on the same directional radiation principles and so has the same limited coverage drawback of the current ILS. Recently the MLS replacement plan has been permanently discarded due to the high cost of the upgrade and the good potential of a GPS-based system for precision landings [33, 37, 46].

The global positioning system (GPS) has been demonstrated to provide very precise position measurements, especially when combined with other sensors [18, 36]. Figure 1.4 [34] shows the two-sigma accuracy capability of various GPS implemen-

tations which use the standard positioning service (SPS) L1 GPS transmission. The GPS accuracy ellipse shown does not reflect the scheduled removal of selective availability (SA). SA is a random dither imposed on the GPS signals in order to degrade the SPS precision available to non-authorized (non-U.S. military) users. Differential GPS (DGPS) provides increased navigation precision using two closely located GPS receivers. One of the receivers must be at a known location so that the GPS solution errors may be isolated. The error corrections are then transmitted to the second receiver, which corrects its own navigation solution. It can be seen that, although neither basic GPS nor DGPS gives the degree of accuracy required for precision landings, the accuracy of carrier phase GPS (CPGPS) is more than sufficient for aircraft landings. At the time of this writing, receiver technology has not yet made CPGPS available during all phases of flight, especially during highly dynamic flight. However, for the benign flight patterns typical in the area of airfields, the accuracy of CPGPS could be fully utilized with current technology.

The GPS has several inherent strengths and weaknesses which directly influence its application to the PLS problem. The continual, global availability (intermittent at very high latitudes) of the GPS signal, along with the accuracy available from CPGPS, make a GPS-based ILS appear to be an improvement (in positioning) from the current ILS. A GPS-based PLS would overcome all of the major limitations of the current ILS, while exceeding its current performance specifications. It can be seen in Figure 1.4 that GPS alone (the outer ellipse) and even differential GPS do not meet the accuracy requirements for a PLS (the central ellipse), especially in the vertical direction. This vertical weakness of GPS leads to the desire to include a radar altimeter in the integrated PLS system. A ground-based pseudolite will also help with the vertical precision problem. A pseudolite improves the navigation solution by providing GPS satellite-like signals from a known position (surveyed, eliminating satellite position uncertainty) below the user's horizon.

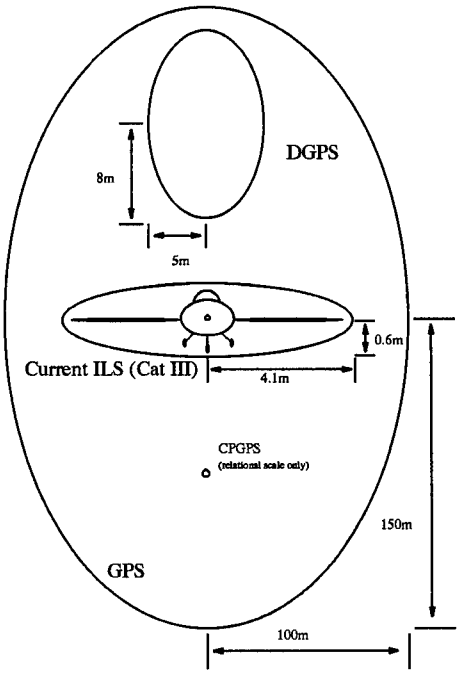

Figure 1.4 GPS Accuracies (Two Standard Deviations)

A significant weakness of GPS, with respect to the PLS application, is the susceptibility of the very low power GPS signal to interference. For military applications this interference could originate from enemy interference/jamming or spoofing. Benign interference such as that from unregulated electronic devices would probably have the greatest effect on civil aviation, although interference from low-wattage jammers placed by terrorists is a viable concern. The susceptibility of the GPS signal to external interference strongly motivates the use of a GPS-aided INS-based PLS, rather than a standalone GPS-based PLS. The navigation solution of a low-quality INS drifts at a rate of about four nautical miles per hour or 400 feet per minute. While this rate of drift makes the INS unsuitable for a PLS, it does give an accurate enough solution for an aircraft to navigate safely away from the landing area when a GPS failure is detected. This work will consider integration of low-, medium-, and

high-quality INSs as discussed in Section 1.4. The low-quality INS is used here to illustrate the drift problem.

The accuracy potential of GPS as a primary sensor in a precision landing system is evident. However, possible interference of the GPS signal, as well as possible system failures from other sources, remain a major concern. Whatever the source of the interference, the navigation solution provided by any GPS receiver under interference conditions would be unreliable. The dependability of the GPS signal as a safety-of-flight-critical sensor in a precision landing system remains to be adequately addressed.

1.2 Problem Definition

The purpose of this thesis is to develop an effective method of receiver autonomous integrity monitoring (RAIM) for GPS-aided INS-based PLS. The RAIM scheme to be developed in this work will accurately indicate the integrity of the GPS (and overall system) navigation solution during interference/jamming or spoofing conditions.

1.3 Summary of Current Knowledge

Recent research in the area of GPS-based PLS conducted by the FAA and other researchers has focused on developing and verifying a GPS-based navigation architecture [3, 10, 33, 37, 46] which can provide the necessary navigational accuracy for precision landings. The work of Gray [10] (using standard GPS) and Britton [3] (using DGPS) at AFIT has shown that an integrated GPS-aided INS-based PLS meets FAA requirements for Category I and II precision approaches. Paielli, Russell, and others [33] have demonstrated the increased accuracy potential which may be obtained by using carrier phase GPS measurements.

Much research effort has focused on developing a working RAIM method for GPS. Comparatively little work has focused on integrity monitoring for a GPS-aided

INS-based navigation system. Vasquez [41, 42] has applied generalized likelihood ratio and chi-square testing schemes to the problem of interference/jamming and spoofing detection for the GPS signal in a GPS-aided INS-based system.

1.4 Assumptions

The integrated system to be used will include a ground-based pseudolite which will provide GPS-like transmissions from a surveyed location near the runway [10]. The main advantage of a ground-based pseudolite is the improvement it makes in the geometry of the GPS problem. GPS satellites are located strictly above the user's horizon. The pseudolite provides a range measurement from beneath (below the horizon) the user. In this research, it will be assumed that differential GPS, pseudolite, radar altimeter, baro altimeter, and INS navigation signals will all be available. The benefits (to failure identification) of the pseudolite and radar altimeter will be analyzed by conducting some performance simulations without one or both of these measurements. Three different accuracies of INSs will be used, a high-quality INS as might be used by the military, a medium-quality INS like those used in commercial transportation, and a low-quality INS which may be inexpensive enough to be used in small private aircraft. Radar altimeter measurements are assumed to be reflected from the WGS-84 reference ellipsoid (no terrain effects will be modeled) when the aircraft is at less than 3000 feet AGL (above ground level). A radar altimeter is considered to be a reasonable part of such a PLS because of its common availability on military aircraft and commercial airliners, as well as the non-prohibitive cost of incorporating it into small civil aviation vehicles.

1.5 Literature Review

This section reviews the current literature relative to the GPS-aided INS-based Precision Landing System (PLS). It also reviews the literature pertaining to several of the most viable techniques of failure detection and identification. The algorithm

discussions presented in this section are general and quite terse; a detailed technical description of each algorithm is taken up in Chapter 2.

1.5.1 Basis of a GPS-aided INS-based PLS. Much recent research has focused on developing and flight testing a GPS-based PLS to replace the current ILS (and recently cancelled MLS) system [3, 10, 12, 33]. To ensure safety of flight, aircraft require an internal, independent navigation reference system that may be used when the GPS signal is unhealthy or jammed. Where the cost is justified, inertial navigation systems (INSs) are used to provide this independent, internal navigation reference. Military and large commercial aircraft use INSs. The cost of lower-quality INSs (and of GPS systems) has become realistic for civil aviation applications. Additional navigation devices may be required to aid the INS during lengthy GPS outages.

INSs provide very accurate (relative) navigation over short periods of time but drift substantially (accumulate position bias) over time. The GPS provides accurate navigation information over long periods (no drifting trend) but can easily be lost for short times during periods of highly dynamic aircraft maneuvering or GPS signal interference. Integration of INS and GPS navigation data with other navigation aids provides increased navigation accuracy during benign flight and allows reliable navigation in the face of aircraft dynamics or intermittent GPS interference/jamming [10, 12, 36]. The standard navigation system on military aircraft is based on an INS aided by GPS and other navigation aids. A GPS user can only receive signals from satellites located above the horizon (ideally, transmitters would be available from below the horizon as well, as is the case when pseudolites are present in the local area); there is no sensitivity to the cardinal direction (azimuth) to the satellite. This geometry makes the GPS navigation solution least precise in the vertical direction. To help overcome this weakness, altitude sensors are standard navigation aids for systems using GPS. An integrated system consisting of (at least) GPS, an INS, and barometric altimeter is common on military aircraft. A radar

altimeter is also generally present on military and commercial aircraft although it is not typically a part of the integrated navigation system. For this work a radar altimeter, when present, will be modelled as a part of the integrated navigation system. The integration of the radar altimeter was first done by Gray [10] and Britton [3] to improve the vertical-channel navigational precision during approach and landing maneuvers.

Where an integrated navigation system is in place, especially in military applications, it is desired to develop a PLS based on the entire available set of navigation components. Past AFIT theses have focused on the development of a GPS-aided INS-based landing system. This research has shown that a differential GPS-aided INS-based PLS provides Category I and II landing precision [3, 10]. The reliability of this landing configuration, especially in possibly jammed or spoofed areas of operation, remains to be fully investigated.

1.5.2 Integrity Monitoring. This work focuses on the problem of detecting radio frequency (**rf**) interference which would corrupt local GPS signals and the resulting navigation solution. Civilian GPS users will most often face benign interference from **rf** sources such as microwave or television transmitters. Military GPS users must anticipate malignant interference/jamming or spoofing from enemy sources. The signal fault identification process is referred to as Fault Detection and Identification (FDI). This section reviews the literature pertaining to four techniques of FDI, which have been or could be applied to the current integrity monitoring problem. Kalman filtering will be used in all of the cases examined to provide integration of the INS with other systems. The measurement residual signals generated by the Kalman filters will be used to perform the FDI functions [13–15, 20, 25]. See Maybeck [21–23] for a thorough presentation of Kalman filter theory and applications.

1.5.2.1 Integrated Navigation and FDI Concepts. Real-world navigation sensors cannot produce an exact navigation solution; they give a measurement,

corrupted by noise, of the unknown navigation parameters. Modern integrated navigation is implemented by using Kalman filters to combine measurements optimally from multiple information sources, based upon the known (or determined) relative precision of each source. Conceptually, Kalman filters estimate the system state, propagate that estimate forward to the next measurement sample time using an assumed system model, then update that state estimate using actual measurement information. The difference, at each update time, between the filter's state prediction (based on propagating its system model) and the measurement actually taken, is called the measurement residual. The measurement residual is a reliable indicator of how well the filter's assumed system model matches the actual system. When the filter model disagrees with the real world, the characteristics of the residuals can provide information about how real system differs from the filter's model.

The fundamental objective of FDI is to examine the available information in such a way that system failures can be detected and identified. Navigation sensor failures are identifiable as discrepancies in the solutions provided by different information sources [44], either as directly viewed from the sensor, or as observed in the character of the measurement residual of the integration Kalman filter. The PLS under examination here, consisting of (at least) GPS, an INS, barometric altimeter, and radar altimeter, is well-suited to FDI techniques based on multiple, redundant information sources. The solution to the current problem will involve checking the received GPS navigation data against the internal INS navigation solution, which is not subject to external interference. It is desired to incorporate the received GPS signals when they are currently valid, because the GPS solution accuracy greatly contributes to the precision required for ILS-like aiding.

Each FDI algorithm examined differs in function and complexity. Of the algorithms discussed, chi-square testing only detects the existence of failures in the system, while the other three FDI methods perform some degree of fault identification and recovery [22, 26, 40, 44].

1.5.2.2 Redundancy (The Voting Method). Perhaps the simplest and most reliable failure detection technique is the use of duplicate hardware for voting [41,42,44]. Simple voting requires three redundant information sources. The outputs of each identical element are compared, allowing them to vote on what information is valid. If two elements agree on the aircraft position and the third provides a different solution, the deviant element is declared to have failed.

The voting method for triply redundant devices has two restrictive weaknesses. Once any single element has failed and is removed from the redundant triad, the system can detect, but is unable to isolate, a second failure. If a second failure occurs, the system cannot determine which element is in error and both remaining elements should be taken off line in order to maintain the integrity of the system's navigation solution. A second weakness of the voting method is the additional expense, space, weight, and computational ability required for redundant hardware [26,40,44]. The requirement for these additional resources typically makes the voting method unacceptable for avionics applications.

1.5.2.3 Chi-Square Testing. The chi-square testing algorithm [40, 44] assumes the use of a Kalman filter to combine information from the available navigation sensors. This method is based on monitoring the measurement residuals of the Kalman filter. The filter's residuals and their filter-computed residual covariance are monitored. Measurement residuals for a filter having the correct model of the real world should display four well-defined characteristics. Residuals should be white, Gaussian, zero mean, and have covariance $HP^{-}H^{T} + R$ (or closely approximate those properties) [6, 13, 21, 22, 40]. Gaussian-ness, zero-mean-ness, and covariance $HP^{-}H^{T} + R$ are exploited in the test conducted in this fault detection method. The chi-square test declares a failure when the tested properties of the filter residuals are inconsistent with those expected from a filter having the correct model. For instance, one can run a hypothesis test (to some confidence level) that 95% of a residual's scalar component lies between $0\pm$ twice the filter-computed standard deviation for

that residual component. The chi-square test is easy to implement and runs quickly but provides only a binary fail/no-fail indication of system operation. It is not useful for failure identification [41,42,44].

1.5.2.4 Generalized Likelihood Ratio Testing. Like chi-square testing, Generalized Likelihood Ratio testing (GLR) uses the measurement residuals of the navigation system's Kalman filter [40–42,44,45]. The GLR algorithm attempts to detect and isolate failures by knowing the effect that each failure has on the character of the filter residuals. Maximum Likelihood Estimation (MLE) is used to determine which condition is most probable, given the current character of the filter residuals. In general, to implement the GLR method, the designer is required to provide a signature failure matrix which provides the algorithm with the description of how different failures modify the filter residuals [44]. Uncertain parameters such as failure magnitudes may also be estimated by modelling their effects in the signature failure matrix.

Van Trees [40] shows the development of multiple GLR testing. In multiple GLR testing, the system Kalman filter is designed based on the no-fail condition. Matching filters are designed based on the expected failure conditions (including no-fail). Each matching filter generates a GLR using MLE. Based on the GLRs, central testing logic determines which matching filter best matches the real world. Once a failure is detected, a corrective signal can be fed back to the Kalman filter for adaptation to the failure. Further detail on GLR's is presented by Willsky and Jones [44,45]. See Vasquez [41,42] for an example of multiple GLR testing applied to a navigation problem.

1.5.2.5 MMAE. The MMAE technique uses multiple Kalman filters running in parallel to model the dynamics of the system (in this case the PLS) under different conditions of failed or no-fail operation [1,20,22]. A separate Kalman filter is designed for each failure condition and, during operation, the residuals are used

to determine how well each filter models the current system state. The MMAE technique is similar to multiple GLR testing in many ways, but differs in its structure and decision making methods. Multiple GLR testing uses a variety of matching filters to model failure characteristics of the residuals of a single Kalman filter. In the MMAE algorithm, a probability of model accuracy ranging from zero to one is computed for each elemental filter within the MMAE structure. Each separate filter's state estimates are scaled by the computed probabilities that each filter has the correct model of the real world. These scaled state estimates are then added together to produce a probability-weighted blending of the state estimates. This blending technique has particular advantages when the real world system is not exactly described by any one filter model, but exists in the parameter space between discrete filter models. In such a situation the filter residuals of more than one of the modeled conditions look reasonably valid. The weighted state estimates provided by the MMAE algorithm are used to construct the navigation solution and to provide system feedback. A major strength of the MMAE technique lies in its capacity for rapid and valid adaptation. Multiple filters are running in parallel, making multiple sets of residuals available at all times with which to decide which filter model looks best (according to the "good" residual qualities [21,22] described in Section 1.5.2.3).

For robust operation, the bank of Kalman filters must be specified in such a way that it spans the entire failure space of the application of interest. Rigorous general convergence proofs for MMAE do not exist [7, 9, 43], but some limited proofs have been established [4, 5, 22, 24] and experience has shown that, as long as the failure condition of the system remains within the span of the filter bank, the MMAE technique is robust to that condition and will respond to it very quickly and accurately.

1.5.3 Literature Review Conclusion. If it is to be used in actual operation, the proposed GPS-aided INS-based PLS will require rapid and accurate fault detection and identification to establish the integrity of the PLS aid at all times.

The error detection techniques presented in Section 1.5.2 are representative of the methods available to accomplish the required fault detection.

The redundancy technique is not feasible for this application. Multiple GPS receivers on a single aircraft would probably not be restrictive on the basis of size, weight, or computational loading, but each receiver would be subject to the same failure environment, e.g. interference/jamming or spoofing, and so nothing would be gained by this redundancy.

Because of its speed and simplicity, chi-square testing has been widely used to detect failures. However, chi-square testing alone has no ability to isolate failures and so does not have much applicability to the integrity monitoring problem under examination, if used alone. This testing method has been effectively used when combined with another test to perform the identification function.

Multiple GLR testing and MMAE are both well suited to the FDI problem. Both of these techniques are flexible enough to be applied to the complex GPS-aided INS-based problem. This work uses the MMAE technique to perform the failure detection and identification. Vasquez [41, 42] has used a combination of chi-square and GLR testing to detect and isolate failures in a GPS-aided INS-based navigation system. Performance comparisons will be made, as appropriate, between the MMAE FDI and GLR/chi-square FDI algorithms.

1.6 Scope

This work will be limited to the development of a multiple model adaptive estimation (MMAE) architecture to provide the integrity monitoring previously described. Comparison will also be made to the generalized likelihood ratio and chi-square testing algorithms developed by Vasquez. The failure modes to be addressed will be limited to the onset of interference/jamming and spoofing. Filters based on ramp failures will not be included in the working MMAE filter bank. When a ramp occurs, the elemental filter with the hypothesized constant parameter value

that most closely matches the current ramp value will receive the highest probability weighting. The ramp should be detectable as a growing trend in the MMAE's blended estimate of the affected parameter (and a gradual shift in the computed probabilities of each hypothesized parameter being correct).

1.7 Methodology Overview

The research consisted first of studying the stochastic and dynamics modeling of the GPS-aided INS-based navigation system, primarily as represented in the theses of Gray [10], Britton [3], and Negast [30]. The model development work accomplished in these previous AFIT theses was duplicated and then confirmed. Once the model had been successfully duplicated, the research effort focused on failure detection within the augmented system. The failure detection analysis of the multiple model architecture used was accomplished using Multiple Model Simulation for Optimal Filter Evaluation (MMSOFE [32]). MMSOFE allows for the simultaneous testing of multiple extended (or linear) Kalman filter (EKF) models in an MMAE architecture. MMSOFE is based on the MSOFE [28] program designed for the testing of single Kalman filters. MSOFE was written in FORTRAN 77 [39] and its use involves significant modification of up to 14 user-definable modules.

Comparison to the work of Vasquez is performed. Vasquez used generalized likelihood ratio (GLR) and chi-square testing schemes to detect and estimate interference/jamming- and spoofing-induced failures in a GPS-aided INS-based system. GLR techniques are based on residual monitoring and can be very effective. The *failure detection and identification* results of Vasquez' work are compared to those generated using the MMAE-based FDI methods.

1.8 Overview of Thesis

Chapter 2 presents the theory used in this research. Kalman filter theory is introduced, along with the basics of several FDI algorithms, including MMAE, GLR, and chi-square testing.

Chapter 3 describes the navigation system's parameters and operation through an overall system description. Models for each of the PLS components, including the INS (with barometric altimeter), GPS, DGPS, pseudolite, and radar altimeter, are defined in detail. The failure models used to represent interference/jamming and spoofing failures are developed, along with a description of the MMAE-based methods that are used to detect those failures.

Results of the work done are shown in Chapter 4, including an analysis of the FDI (and navigation guidance) performance observed. Chapter 5 summarizes the research through conclusions and recommendations.

2. Theory

2.1 Overview

This chapter presents the fundamental theory of the Kalman filter, and the sampled-data Kalman filter equations used in this work are developed. The extended Kalman filter (EKF), a well-established ad hoc method based on the Kalman filter, is used in estimation problems involving nonlinear dynamics and/or nonlinear measurement models. The equations defining (EKF) are presented in this chapter, along with a more detailed discussion of the MMAE, GLR, and chi-square failure detection methods introduced in Chapter 1.

2.2 The Extended Kalman Filter

The EKF allows for nonlinear, time-varying system dynamics and/or measurement vectors, as are found in this GPS/INS navigation problem. In simple linearized Kalman filtering (LKF), the dynamics and/or measurement equations are linearized through first-order perturbation techniques about a fixed nominal trajectory. The LKF is the conceptual basis for the EKF. During operation the EKF is continually relinearized about the most recent state estimate trajectory rather than about a fixed nominal trajectory.

For the sampled data Kalman filter, let the system model be expressed as a state equation of the form

$$\dot{\mathbf{x}}(t) \;=\; \mathbf{f}[\mathbf{x}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.1}$$

where the state dynamics vector $\mathbf{f}[\mathbf{x}(t), t]$ is a (possibly) nonlinear function of the state vector $\mathbf{x}(t)$ and time t. Let $\mathbf{w}(t)$ be a white Gaussian noise with mean

$$E[\mathbf{w}(t)] \;=\; \mathbf{0} \tag{2.2}$$

and noise strength $\mathbf{Q}(t)$ defined by

$$E[\mathbf{w}(t)\mathbf{w}^T(t+\tau)] \;=\; \mathbf{Q}(t)\delta(\tau) \tag{2.3}$$

The discrete-time measurements, $\mathbf{z}(t_i)$, are modeled as a (possibly) nonlinear vector of functions of the state vector and time, $\mathbf{h}[\mathbf{x}(t_i), t_i]$, plus additive white Gaussian noise:

$$\mathbf{z}(t_i) \;=\; \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \tag{2.4}$$

where $\mathbf{h}[\mathbf{x}(t_i), t_i]$ is the nonlinear observation vector and $\mathbf{v}(t_i)$ is a zero-mean discrete-time white Gaussian noise, independent of the dynamics driving noise $\mathbf{w}(t)$ and having covariance $\mathbf{R}(t_i)$ defined by

$$E[\mathbf{v}(t_i)\mathbf{v}^T(t_j)] \;=\; \begin{cases} \mathbf{R}(t_i) & for \;\; t_i \;=\; t_j \\ \mathbf{0} & for \;\; t_i \neq t_j \end{cases} \tag{2.5}$$

The LKF is based on perturbation states about a nominal state trajectory $\mathbf{x}_n(t)$ satisfying $\mathbf{x}_n(t_0) \;=\; \mathbf{x}_{n_0}$ and

$$\dot{\mathbf{x}}_n(t) \;=\; \mathbf{f}[\mathbf{x}_n(t), t] \tag{2.6}$$

using the same $\mathbf{f}[\cdot, \cdot]$ as in Equation (2.1). The nominal, noise-free measurements are also based on the nominal states and are defined as

$$\mathbf{z}_n(t_i) \;=\; \mathbf{h}[\mathbf{x}_n(t_i), t_i] \tag{2.7}$$

The perturbation states are found by subtracting the nominal states in Equation (2.6) from the original states in Equation (2.1):

$$[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)] \;=\; \mathbf{f}[\mathbf{x}(t), t] - \mathbf{f}[\mathbf{x}_n(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.8}$$

Equation (2.8) is approximated to first order through a truncated Taylor series expansion (letting $\dot{\delta}\mathbf{x}(t)$ be a first-order approximation to $[\mathbf{x}(t) - \mathbf{x}_n(t)]$):

$$\dot{\delta}\mathbf{x}(t) \;=\; \mathbf{F}[t;\mathbf{x}(t)]\delta\mathbf{x}_n(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.9}$$

where $\delta\mathbf{x}(t)$ are the perturbation states. The definitions for $\mathbf{G}(t)$ and $\mathbf{w}(t)$ are unchanged, and the new linearized dynamics matrix $\mathbf{F}[t;\mathbf{x}_n(t)]$ is found by taking partial derivatives of $\mathbf{f}[\mathbf{x}(t),t]$ with respect to $\mathbf{x}(t)$ and evaluated at the nominal values for the trajectory $\mathbf{x}_n(t)$:

$$\mathbf{F}[t;\mathbf{x}_n(t)] \;=\; \left.\frac{\partial\mathbf{f}[\mathbf{x}(t),t]}{\partial\mathbf{x}}\right|_{\mathbf{x}\,=\,\mathbf{x}_n(t)} \tag{2.10}$$

The discrete-time perturbation measurements are similarly approximated to first order from the measurement difference equation

$$\delta\mathbf{z}(t_i) \;\equiv\; \mathbf{z}(t_i) - \mathbf{z}_n(t_i) \;=\; \mathbf{h}[\mathbf{x}(t_i),t_i] - \mathbf{h}[\mathbf{x}_n(t_i),t_i] + \mathbf{v}(t_i) \tag{2.11}$$

yielding the perturbation form (letting $\delta\mathbf{z}(t_i)$ be a first-order approximation to $[\mathbf{z}(t_i) - \mathbf{z}_n(t_i)]$:

$$\delta\mathbf{z}(t_i) \;=\; \mathbf{H}[t_i;\mathbf{x}(t_i)]\delta\mathbf{x}_n(t_i) + \mathbf{v}(t_i) \tag{2.12}$$

The same partial derivative methods used to derive the linearized state dynamics matrix $\mathbf{F}[t;\mathbf{x}_n(t)]$ are used again to derive the linearized observation matrix:

$$\mathbf{H}[t_i;\mathbf{x}_n(t_i)] \;=\; \left.\frac{\partial\mathbf{h}[\mathbf{x}(t_i),t_i]}{\partial\mathbf{x}}\right|_{\mathbf{x}\,=\,\mathbf{x}_n(t_i)} \tag{2.13}$$

Because the LKF generates error state estimates, $\widehat{\delta\mathbf{x}}(t)$, they must be added to the nominal states to provide whole state estimates $\hat{\mathbf{x}}(t)$ in the form

$$\hat{\mathbf{x}}(t) \;=\; \mathbf{x}_n(t) + \widehat{\delta\mathbf{x}}(t) \tag{2.14}$$

If the nominal and the "true" trajectories differ too greatly, a linearized Kalman filter will yield erroneous state estimates because the condition for neglecting higher order terms of the Taylor series expansion is violated. The EKF will be formed by linearizing about the most recent state estimate $\hat{\mathbf{x}}$ rather than about the nominal trajectory $\mathbf{x}_n$, as is done in the LKF. The following sampled data EKF equations use the notation $t|t_i$ to represent the value of a given variable at time t, conditioned on the measurements taken through time t_i. Also, t_i^- represents the value after propagation but prior to the measurement update at sample time t_i, and t_i^+ corresponds to the value after the measurement update. The state estimates $\hat{\mathbf{x}}(t|t_i)$ and covariance values $\mathbf{P}(t|t_i)$ are propagated from t_i to t_{i+1} by solving the following differential equations:

$$\dot{\hat{\mathbf{x}}}(t|t_i) = \mathbf{f}[\hat{\mathbf{x}}(t|t_i), t] \tag{2.15}$$

$$\dot{\mathbf{P}}(t|t_i) = \mathbf{F}[t;\hat{\mathbf{x}}(t|t_i)]\mathbf{P}(t|t_i) + \mathbf{P}(t|t_i)\mathbf{F}^T[t;\hat{\mathbf{x}}(t|t_i)] + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \tag{2.16}$$

where

$$\mathbf{F}[t;\hat{\mathbf{x}}(t|t_i)] = \left.\frac{\partial \mathbf{f}[\mathbf{x}(t),t]}{\partial \mathbf{x}}\right|_{\mathbf{x}\,=\,\hat{\mathbf{x}}(t|t_i)} \tag{2.17}$$

and initial conditions are given by:

$$\hat{\mathbf{x}}(t_i|t_i) = \hat{\mathbf{x}}(t_i^+) \tag{2.18}$$

$$\mathbf{P}(t_i|t_i) = \mathbf{P}(t_i^+) \tag{2.19}$$

The discrete-time measurements are processed in the EKF through the update equations:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[t_i;\hat{\mathbf{x}}(t_i^-)]\Big\{\mathbf{H}[t_i;\hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-)\mathbf{H}^T[t_i;\hat{\mathbf{x}}(t_i^-)] + \mathbf{R}(t_i)\Big\}^{-1} \tag{2.20}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\Big\{\mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i]\Big\} \tag{2.21}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}[t_i;\hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-) \tag{2.22}$$

where

$$\mathbf{P}(t_i^-) \;=\; \mathbf{P}(t_i|t_{i-1}) \;\text{ and }\; \hat{\mathbf{x}}(t_i^-) \;=\; \hat{\mathbf{x}}(t_i|t_{i-1}) \tag{2.23}$$

as produced by the most recent propagation cycle. The variable $\mathbf{z}_i$ represents the actual realization of the measurement $\mathbf{z}(t_i)$, $\mathbf{H}[t_i; \hat{x}(t_i^-)]$ is given by:

$$\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i{}^-)] \;=\; \left.\frac{\partial \mathbf{h}[\mathbf{x}(t_i), t_i]}{\partial \mathbf{x}}\right|_{\mathbf{x}\,=\,\hat{\mathbf{x}}(t_i{}^-)} \tag{2.24}$$

and $\mathbf{K}(t_i)$ is the discrete-time Kalman filter gain. Recall from the previous page that, for the EKF, the measurement and dynamics vectors are calculated about the last state estimate $\hat{\mathbf{x}}(t_i{}^-)$ and the state trajectory $\mathbf{x}(t|t_i)$, rather than about the nominal.

Before proceeding to a discussion of failure detection and identification (FDI) techniques, the reader should make note of Equation (2.21). The term $\{\mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i{}^-), t_i]\}$ on the right hand side of Equation (2.21) is called the measurement residual. At each sample time t_i, the residual is the difference between the actual measurement of the real world $\mathbf{z}(t_i)$ and the filter's best prediction of what the measurement should have been before it arrived, $\mathbf{h}[\hat{\mathbf{x}}(t_i{}^-), t_i]$. The filter's prediction is based on its model of the system, so the characteristics of the measurement residual are an indication of how well the filter's model currently matches the real world. The residuals from a filter (the k^{th} filter of a number of possible filters) having the correct model will be (for a linear KF, or to first order for an EKF) white, Gaussian, zero-mean, and have covariance $[\mathbf{H}_k(t_i)\mathbf{P}_k(t_i{}^-)\mathbf{H}_k{}^T(t_i) + \mathbf{R}_k(t_i)]$. This information contained in the residual is the basis of the FDI techniques discussed in the next section.

2.3 Failure Detection

The MMAE and the GLR/chi-square methods of FDI [22, 40] are residual-based techniques built upon the Kalman filter just developed in Section 2.2. The basic ideas upon which these FDI methods are based were discussed in Section 1.5.2. This section presents a theoretical discussion of the MMAE, GLR, and chi-square

algorithms. See VanTrees [40] for a more complete development of the GLR and chi-square methods. Maybeck [22] presents a clear and rigorous development of MMAE theory.

2.3.1 Multiple Model Adaptive Estimation. Multiple Model Adaptive Estimation (MMAE) can provide simultaneous state and parameter estimation. MMAE is composed of multiple "elemental" Kalman filters running in parallel, each using an identical deterministic input and measurement environment. Each of the individual elemental filters may model a different set of system parameter values based on known possible operating conditions, or may model some possible failure condition such as greatly increased measurement noise covariance $\mathbf{R}(t_i)$, which might represent the effects of GPS interference/jamming. The magnitude of a failure can also be estimated by using multiple elemental filters with different assumed failure magnitudes. The residuals from each filter are used to calculate the conditional probabilities that each of the filters has the most correct model. Uncertain parameters are estimated in this way because specific parameter values are associated with each filter. The conditional probabilities, also referred to as hypothesis conditional probabilities, will be used as weights for blending the state estimates from the elemental filters to produce the final blended MMAE state estimates. See Figure 2.1 for a graphical representation of the MMAE algorithm.

The conditional probability $p_k(t_i)$ for the k^{th} elemental filter, $k = 1, 2, \ldots, K$ is determined by:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_k, \mathbf{Z}_{i-1}) p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_j, \mathbf{Z}_{i-1}) p_j(t_{i-1})} \tag{2.25}$$

where, for this development, the $\mathbf{a}_k$ parameters may represent measurement bias magnitudes and/or measurement noise covariance values. The numerator of Equation (2.25) is the product of two terms. The first, $p_k(t_{i-1})$, is merely the most recent value of the conditional probability for the k^{th} elemental filter, making this equation

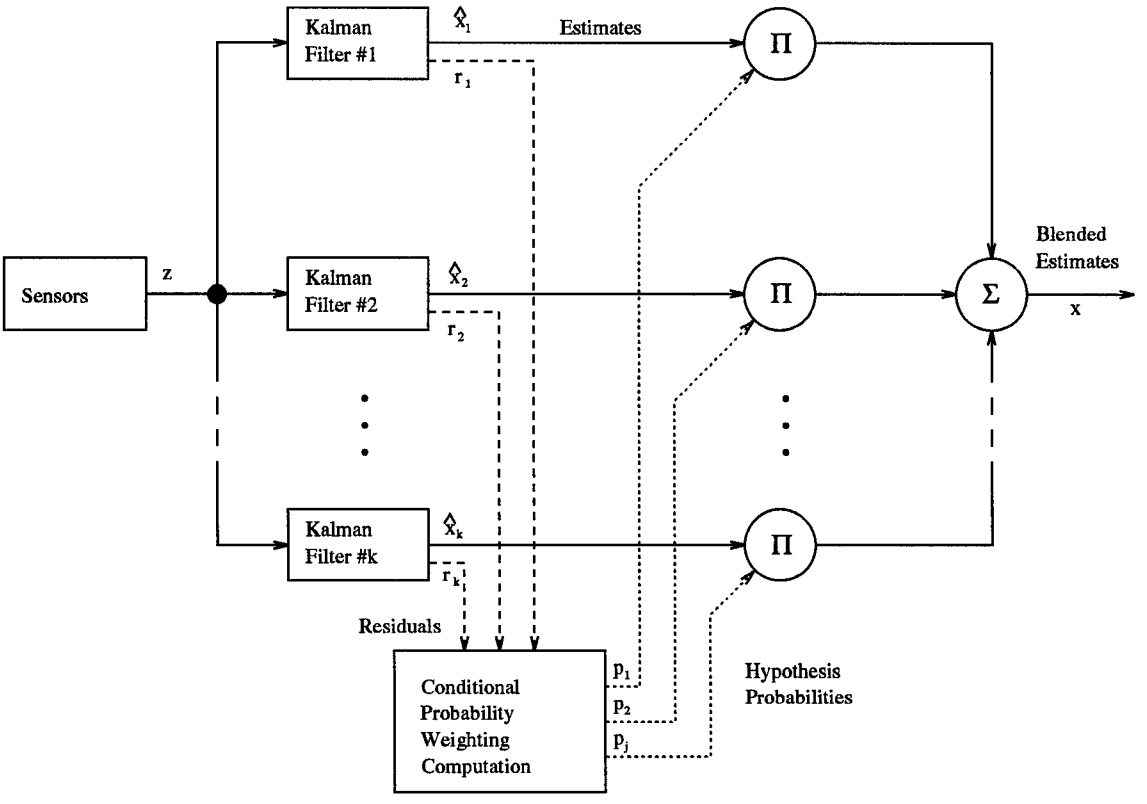

Figure 2.1 Multiple Model Adaptive Estimation

iterative. All K numerator terms for the K elemental filters at t_{i-1} must be available before the denominator and consequently the k^{th} conditional probability at the current time, t_i, can be calculated. The first numerator term is the probability density function of the current measurements, conditioned on both the assumed parameter values and the observed past measurements. This probability density function is computed by:

$$\begin{aligned} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})(\mathbf{z}_i|\mathbf{a}_k,\mathbf{Z}_{i-1})} &= 1/(2\pi)^{\frac{m}{2}} \left|\mathbf{\Lambda}_k(t_i)\right|^{\frac{1}{2}} \exp\{\cdot\} \\ \{\cdot\} &= \left\{-\tfrac{1}{2}\mathbf{r}_k^T(t_i)\mathbf{\Lambda}_k^{-1}(t_i)\mathbf{r}_k(t_i)\right\} \end{aligned} \tag{2.26}$$

where m is the measurement vector dimension, and the filter residual is given (for linear filters)by:

$$\mathbf{r}_k(t_i) = [\mathbf{z}(t_i) - \mathbf{H}_k(t_i)\hat{\mathbf{x}}_k(t_i^-)] \tag{2.27}$$

and

$$\mathbf{r}_k(t_i) \;=\; \left\{[\mathbf{z}(t_i) - \mathbf{h}_k[\hat{\mathbf{x}}_k(t_i^-), (t_i)]\right\} \tag{2.28}$$

for the more general extended Kalman filters, and where the residual covariance is computed by the k^{th} elemental filter as:

$$\mathbf{\Lambda}_k(t_i) \;=\; [\mathbf{H}_k(t_i)\mathbf{P}_k(t_i{}^-)\mathbf{H}_k{}^T(t_i) + \mathbf{R}_k(t_i)] \tag{2.29}$$

The k^{th} filter residual, $\mathbf{r}_k(t_i)$, is dependent upon the current measurement, $\mathbf{z}(t_i)$, the measurement matrix, $\mathbf{H}_k(t_i)$ (or the measurement function $\mathbf{h}_k(\cdot, t_i)$ for an EKF), and the state estimate prior to the i^{th} measurement, $\hat{\mathbf{x}}(t_i{}^-)$. The k^{th} elemental filter's state estimation error covariance matrix before the i^{th} measurement, $\mathbf{P}_k(t_i{}^-)$, the measurement matrix, and the observation noise covariance matrix, $\mathbf{R}_k(t_i)$, are used to construct the residual covariance for each filter. The Kalman filter equations and notation were discussed in Section 2.2.

If the residuals in the k^{th} filter display a mean of zero and the corresponding filter-computed residual covariance $\mathbf{\Lambda}_k(t_i)$, the exponential term $\{\cdot\}$ in Equation (2.26) is approximately equal to $-\frac{m}{2}$, where m is the dimension of $\mathbf{z}(t_i)$ and $\mathbf{r}(t_i)$. However, if the residuals are much bigger than anticipated due to the wrong parameter hypothesis, the exponential term $\{\cdot\}$ in Equation 2.26 is a much larger negative number ($\mathbf{\Lambda}_k^{-1}$ is positive definite), so p_k decreases exponentially. The denominator of Equation (2.25) represents the sum of the numerator terms from each elemental filter computed at time t_i. This scales the numerator to ensure that the sum of the p_k's is always one. However, a difficulty arises if the conditional probability of a state estimate were to become zero. In this case the conditional probability, p_k, becomes identically equal to zero for all time thereafter. Once equal to zero, the iterative form of Equation (2.25) locks that p_k at a zero value for all subsequent calculations, even if that filter begins to produce good state estimates. To avoid this

problem, a lower bound (threshold) should be set on each p_k to prevent such a zero lock-in condition [22].

The state estimates from each filter, $\hat{\mathbf{x}}_k$, are then scaled by the corresponding weighting factor p_k. These weighted state estimates are summed for all K filters, resulting in the Bayesian blended state vector estimate:

$$\hat{\mathbf{x}}(t_i^+) = \sum_{k=1}^{K} \hat{\mathbf{x}}_k(t_i^+) \cdot p_k(t_i) \tag{2.30}$$

The covariance, $\mathbf{P}(t_i^+)$, of the blended solution is given by [22]:

$$\mathbf{P}(t_i^+) = \sum_{k=1}^{K} p_k(t_i) \left\{ \mathbf{P}_k(t_i^+) + [\hat{\mathbf{x}}_k(t_i^+) - \hat{\mathbf{x}}(t_i^+)][\hat{\mathbf{x}}_k(t_i^+) - \hat{\mathbf{x}}(t_i^+)]^T \right\} \tag{2.31}$$

The parameter estimates, $\hat{\mathbf{a}}_k$, are calculated by scaling the assumed parameter values from each elemental filter by the corresponding weighting factor, p_k, in the same manner as for the state estimates. The weighted estimates of all K filters are summed according to the relationship:

$$\hat{\mathbf{a}}(t_i) = \sum_{k=1}^{K} \mathbf{a}_k \cdot p_k(t_i) \tag{2.32}$$

This Bayesian form of adaptive estimation is depicted in Figure 2.1.

As may be inferred from the preceding presentation, failure detection and identification using MMAE becomes virtually automatic once the estimator has been developed. FDI is accomplished by observing the conditional probabilities, p_k, of the elemental filters. The filter with the highest probability p_k is based on the model which most closely matches the real world (truth model in the case of simulations). Most likely the true parameter value will not exactly match that modeled in any one of the elemental filters, so the unknown parameters will be estimated via blending in a manner similar to the MMAE state estimation.

This thesis research is directly concerned with detecting and isolating failures in the form of GPS interference/jamming or spoofing. Several Kalman filters are used which differ only by the assumed measurement noise variance or measurement signal bias magnitude. Measurement signal biases model measurement jumps and/or ramps (through the MMAE blending just discussed) as might result from intentional spoofing. Increased measurement noise variance would model interference/jamming-type real world measurement noise. Nominally, for the case of measurement signal biases, two filters with different positive bias magnitudes, a matched set with negative magnitudes, and one filter with a zero bias magnitude can be assumed (i.e., $K = 5$, where K is the number of elemental Kalman filters). Using variances of measurement noise instead of biases in another set of elemental filters could similarly model interference/jamming noise. As was stated in Section 1.6, there is no need to include elemental filters based on a ramp failure of an assumed slope and/or time of onset. When a ramp occurs, the elemental filter with the bias value that most closely matches the current ramp value will receive the highest probability weighting. The ramp should be observed as a growing trend in the MMAE's blended estimate of the measurement bias.

2.3.2 Moving-Bank MMAE. The MMAE algorithm propagates multiple filters forward in time, continually selecting the filter, or weighted combination of filters, that has the best model of the real world. Often, the possible parameter space is so large that completely discretizing it requires many more filter hypotheses than can realistically be run in parallel. In this case, a small group of filters, whose parameter assumptions are in the close neighborhood of the current parameter estimate, are chosen to be active at one given time. If the estimated parameter moves significantly, the bank of active filters is moved to be centered around the new parameter estimate. This algorithm is called a moving-bank MMAE.

2.3.3 Generalized Likelihood Ratio Testing. The primary goal of GLR is to define a likelihood function $l(t_i, \theta)$ that, when compared to a threshold, will identify the onset of a failure such as interference/jamming or spoofing. The GLR algorithm is depicted in Figure 2.2. Multiple hypotheses are established with a Kalman filter

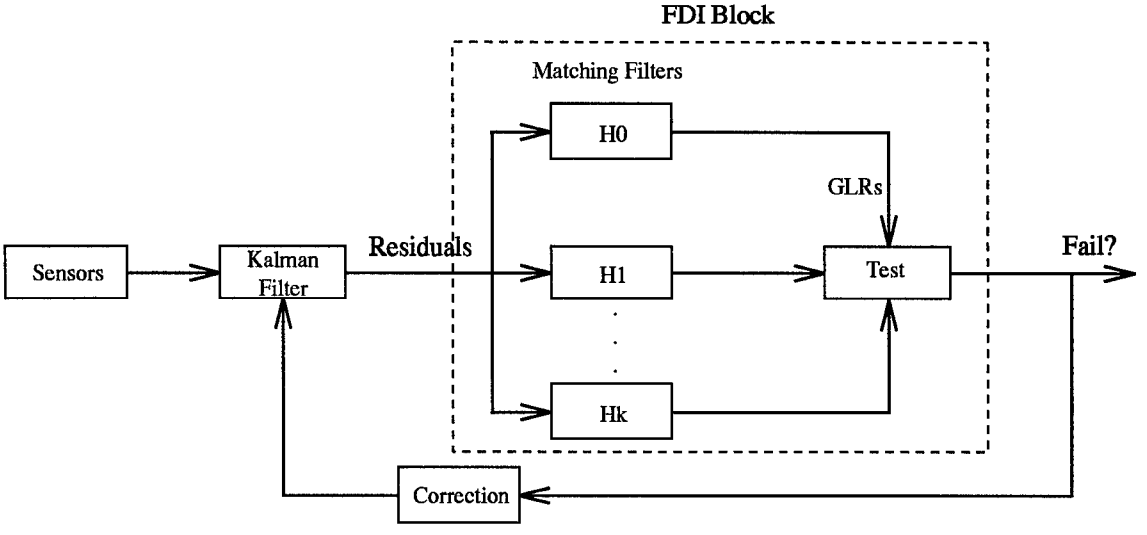

Figure 2.2 Multiple GLR Testing

based on hypothesis $\mathcal{H}_0$ (no failure) and matching filters based on hypothesis $\mathcal{H}_k$ (the k^{th} failure has occurred). The matching filters do not provide state estimation but are designed for failure detection by assuming failures in the system to be modelled as some variation in the actual measurement beyond those variations caused by the dynamics of the system. Each matching filter is designed to inject the kind of residual modification that would actually be experienced in the system Kalman filter if that failure (modelled by $\mathcal{H}_k$) had actually occurred. The failure vector $\mathbf{d}(t_i)$ is $m-$by-1 where m is the number of measurements. Non-failed elements are indicated in $\mathbf{d}(t_i)$ as zeros, while 1's in the failure vector correspond to a measurement device assumed to be induced with a failure. The arguments of the likelihood function introduced above are t_i and θ. As may be seen in Figure 2.3, the variable t_i represents the time of failure onset, while t_0 represents the current time, and t_{0-N} the time of the trailing edge, termed the "beginning", of the GLR search window. As the search window

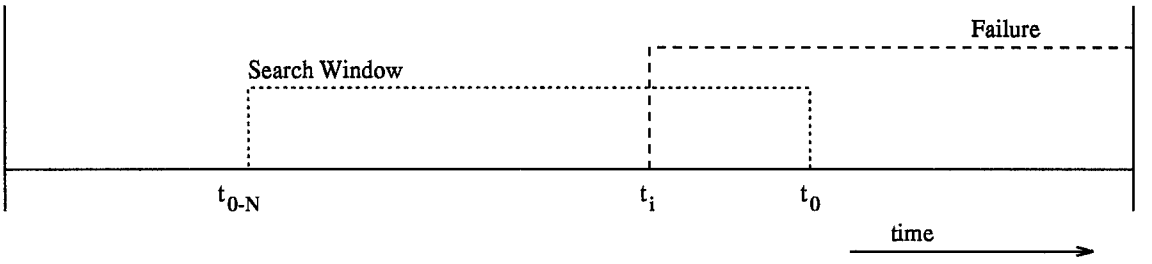

Figure 2.3 GLR Search Window

proceeds forward in time, the failure function $n(t_i, \theta)$ is either 1 or 0 depending on whether the detected failure is currently before or after the θ index value in the GLR algorithm's search window. In general, θ takes on each index value within the search window at each time step, so that the failure may be detected at the earliest possible time after its onset. In order to simplify the GLR algorithm, θ is permanently set to N, so the onset of a failure will not be detected until it reaches the beginning of the search window. Riggins [35] showed that this simplification gives up comparatively little identification performance while significantly reducing the computational load required for implementation.

Under nominal conditions, the Kalman filter residuals $\mathbf{r}^0(t_i)$ are:

$$\mathbf{r}^0(t_i) \;=\; \mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i] \tag{2.33}$$

and the residuals can be expressed for each hypothesis (matching filter) as

$$\mathcal{H}_0 \;:\; \mathbf{r}(t_i) \;=\; \mathbf{r}^0(t_i) \quad \mathcal{H}_k \;:\; \mathbf{r}(t_i) \;=\; \mathbf{r}^0(t_i) + \mathbf{m}(t_i, \theta)\nu \tag{2.34}$$

For a Kalman filter successfully tracking the true states, the nominal no-failure residual $\mathbf{r}^0(t_i)$ will appear as zero-mean white Gaussian noise of covariance $[\mathbf{H}_k(t_i)\mathbf{P}_k(t_i^-)\mathbf{H}_k{}^T(t_i) + \mathbf{R}_k(t_i)]$. When a failure is induced on the measurements, a signal $\mathbf{m}(t_i, \theta)\nu$ will also be present in the residuals, where ν is the unknown magnitude and $\mathbf{m}(t_i, \theta)$ will be presented momentarily. The goal of the GLR algorithm is to identify this signal by recognizing variations in the residuals from their nor-

mal unfailed values. The GLR tests are particularly good at detecting jumps in the residuals, with the key being how closely the matching filters model actual failures. The GLR algorithm is a function of overall system dynamics and behavior ($\mathbf{\Phi}$, $\mathbf{H}$, and Kalman filter gain $\mathbf{K}$); this is shown mathematically below. The failure residual offset $\mathbf{m}(t_i, \theta)$ is found through

$$\mathbf{m}(t_i, \theta) = \mathbf{H}(t_i)\mathbf{y}(t_i, \theta) + \mathbf{d}(t_i)n(t_i, \theta) \tag{2.35}$$

where the recursive failure quantity $\mathbf{y}(t_{i+1}, \theta)$ is given by

$$\mathbf{y}(t_{i+1}, \theta) = \mathbf{\Phi}(t_{i+1}, t_i)[\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]\mathbf{y}(t_i, \theta) - \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{K}(t_i)\mathbf{d}(t_i)n(t_i, \theta) \tag{2.36}$$

and $\mathbf{\Phi}(t_{i+1}, t_i)$ is the state transition matrix for the dynamic system model's propagation from sample time t_i to time t_{i+1}. If the failure is assumed to occur at the beginning of the GLR search window (i.e. $n(t_i, \theta) = 1$ for all t_i), the above equations can be simplified to

$$\mathbf{m}(t_i) = \mathbf{H}(t_i)\mathbf{y}(t_i) + \mathbf{d}(t_i) \tag{2.37}$$

$$\mathbf{y}(t_{i+1}, \theta) = \mathbf{\Phi}(t_{i+1}, t_i)[\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]\mathbf{y}(t_i, \theta) - \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{K}(t_i)\mathbf{d}(t_i) \tag{2.38}$$

The consequence of this simplification is a delay in detecting a failure because it is not realized until it reaches the beginning of the GLR search window. The combination of the Kalman filter outputs and the matching filter model will determine the magnitude of the likelihood function defined as:

$$l(t_i, \theta) = S^T(t_i, \theta)C^{-1}(t_i, \theta)S(t_i, \theta) \tag{2.39}$$

where

$$S(t_i, \theta) = \sum_{j=1}^{i} \mathbf{m}^T(t_j, \theta)\mathbf{\Lambda}^{-1}(t_j)\mathbf{r}(t_j) \tag{2.40}$$

$$C(t_i,\theta) \;=\; \sum_{j=1}^{i} \mathbf{m}^T(t_j,\theta)\mathbf{\Lambda}^{-1}(t_j)\mathbf{m}(t_j,\theta) \tag{2.41}$$

given

$$\mathbf{\Lambda}(t_j) \;=\; \mathbf{H}(t_j)\mathbf{P}(t_j{}^-)\mathbf{H}(t_j) + \mathbf{R}(t_j) \tag{2.42}$$

and the maximum likelihood estimate of the unknown magnitude of the failure, ν, is found by

$$\widehat{\nu}(t_i,\theta) \;=\; \frac{S(t_i,\theta)}{C(t_i,\theta)} \tag{2.43}$$

The residual covariance $\mathbf{\Lambda}(t_j)$ and the residuals are combined with Equations (2.35) and (2.36) or Equations (2.37) and (2.38) to give a linear combination of the residuals $S(t_i,\theta)$ and a deterministic value $C(t_i,\theta)$ defined in Equations (2.40) and (2.41) above. Finally, a decision rule based on a threshold, ϵ, would be

$$\begin{aligned} l(t_i,\theta) \;&>\; \epsilon \;\Rightarrow\; \textit{Declare } FAILURE \\ l(t_i,\theta) \;&\leq\; \epsilon \;\Rightarrow\; \textit{Declare } NO\ FAILURE \end{aligned} \tag{2.44}$$

2.3.4 Chi-Square Testing. A chi-square test is based on the Kalman filter residuals $\mathbf{r}(t_j)$ which are zero-mean and white with known residual covariance $\mathbf{\Lambda}(t_j)$ (under nominal, hypothesized conditions). The chi-square random variable $\mathcal{X}(t_k)$ is given by

$$\mathcal{X}(t_k) \;=\; \sum_{j=k-N+1}^{k} \mathbf{r}^T(t_j)\mathbf{\Lambda}^{-1}(t_j)\mathbf{r}(t_j) \tag{2.45}$$

with N being the size of a sliding detection window. Notice that the system dynamics are not included in Equation (2.45) and that only one failure hypothesis is available. This agrees with the discussion in Section 1.5.2.3 which stated that the chi-square test is very simple and can only be used to detect, not identify, failures. A detection rule based on an established threshold ϵ would be

$$\begin{aligned} \mathcal{X}(t_k) \;&>\; \epsilon \;\Rightarrow\; \textit{Declare } FAILURE \\ \mathcal{X}(t_k) \;&\leq\; \epsilon \;\Rightarrow\; \textit{Declare } NO\ FAILURE \end{aligned} \tag{2.46}$$

2.4 Summary

This chapter provides the theoretical basis for the remaining chapters. MMAE will be used for FDI in this work. The GLR and chi-square algorithms are discussed here because they will be used as a benchmark in comparing the MMAE FDI results to those achieved by Vasquez [41, 42] using more traditional GLR/chi-square FDI. The theoretical developments of this chapter closely follow those presented by Vasquez [41, 42] and Nielsen [31].

3. Methodology

3.1 Overview

This chapter introduces the computer software and simulation techniques used in pursuing this research effort. An overall description of the integrated system is given, followed by detailed state and measurement models for each of the navigation subsystems used. The specific multiple model filter structure used to detect failures is shown. Finally, the simulation approach and research goals are reviewed prior to the presentation of the results in Chapter 4.

3.2 Overall System Description

The main elements of the PLS being tested for failures are the INS and the GPS or DGPS. The barometric altimeter, radar altimeter, and ground-based pseudolite also provide measurements to the Kalman filter. The following measurements are available: four satellite vehicle (SV) pseudoranges, altitude from the barometric altimeter, one surveyed-point (pseudolite) range measurement (optionally), and height above ground level from the radar altimeter (optionally). The truth model used to represent the real world consists of 62 states, while the Kalman filter model is made up of 13 states.

A block diagram representing the system PLS configuration is shown in Figure 3.1. The *true* aircraft position is generated by the trajectory profile generator PROFGEN [27] and provided to each navigation system. The GPS satellite vehicle (SV) positions are given by actual satellite data recorded on 4 May 1991 and are combined with the true aircraft position to obtain true ranges, which are modified with noise to provide *pseudoranges* measurements for use by the GPS. Each navigation system generates perturbations from the true range and the final *difference* measurements are then formed by subtracting the GPS measured ranges from their corresponding INS-calculated ranges. The EKF equations propagate the PLS error

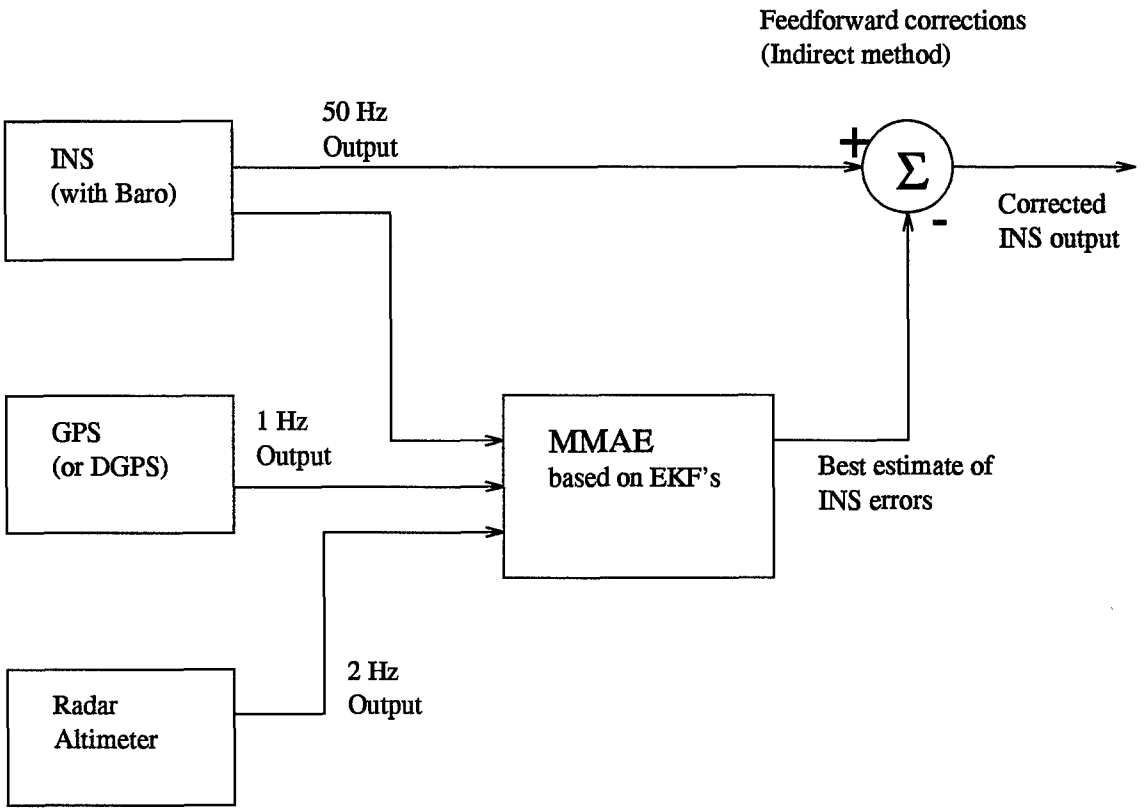

Figure 3.1 Overall PLS Block Diagram

states and use the measurements to calibrate its state estimates. Finally, these state estimates are used to correct the INS-indicated position at each sample time.

3.3 PLS Component Model Descriptions

The truth model consists of 39 INS states, 30 GPS or 22 DGPS states, and a single pseudolite state (optionally). The filter model consists of 11 INS states and two GPS or DGPS states. The following sections will provide the details of these models and the bases for their selection.

3.3.1 INS Models.

3.3.1.1 The INS Truth Model. This section presents the truth model used for the INS. The INS is a strapped-down wander azimuth system based on

the Litton LN-93. The manufacturer, Litton developed a 93-state error model [16] describing the error characteristics of the LN-93. The error states $\delta\mathbf{x}$ used in the full model may be separated into 6 categories.

$$\delta\mathbf{x} = \left[\delta\mathbf{x}_1^T \delta\mathbf{x}_2^T \delta\mathbf{x}_3^T \delta\mathbf{x}_4^T \delta\mathbf{x}_5^T \delta\mathbf{x}_6^T\right]^T \tag{3.1}$$

where $\delta\mathbf{x}$ is a 93 x 1 column vector and

$\delta\mathbf{x}_1$ represents the "general" error vector containing 13 position, velocity, attitude, and vertical channel errors; the first nine states are those of the standard Pinson model of INS error characteristics.

$\delta\mathbf{x}_2$ consists of 16 gyro, accelerometer, and baro-altimeter exponentially time-correlated errors, and "trend" states. These states are modeled as first order Gauss-Markov processes in the truth model.

$\delta\mathbf{x}_3$ represents gyro bias errors. These 18 states are modeled as random constants in the truth model.

$\delta\mathbf{x}_4$ is composed of accelerometer bias error states. These 22 states are modeled in the same manner as the gyro bias states.

$\delta\mathbf{x}_5$ depicts accelerometer and initial thermal transients. The 6 thermal transient states are first-order Gauss-Markov processes in the truth model.

$\delta\mathbf{x}_6$ models gyro compliance errors. These 18 error states are modeled as biases in the truth model.

The truth model state space differential equation is given by

$$\begin{Bmatrix} \delta\dot{\mathbf{x}}_1 \\ \delta\dot{\mathbf{x}}_2 \\ \delta\dot{\mathbf{x}}_3 \\ \delta\dot{\mathbf{x}}_4 \\ \delta\dot{\mathbf{x}}_5 \\ \delta\dot{\mathbf{x}}_6 \end{Bmatrix} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} & \mathbf{F}_{14} & \mathbf{F}_{15} & \mathbf{F}_{16} \\ \mathbf{0} & \mathbf{F}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{55} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\mathbf{x}_3 \\ \delta\mathbf{x}_4 \\ \delta\mathbf{x}_5 \\ \delta\mathbf{x}_6 \end{Bmatrix} + \begin{Bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.2}$$

A full description of the submatrices for this error model differential equation is given in Appendix B. This 93-state error model is a highly accurate LN-93 representation, but the high dimensionality of the state equation makes the model prohibitively CPU-intensive (computationally, and in terms of storage) for projects examining a large number of problem variations. The work of Negast at AFIT addressed the reduction of the INS error-state model [30] while preserving enough fidelity to be considered a viable truth model.

The reduced-order model to be used as the truth model in this research is defined in Equation (3.3):

$$\begin{Bmatrix} \delta\dot{\mathbf{x}}_1 \\ \delta\dot{\mathbf{x}}_2 \\ \delta\dot{\mathbf{x}}_3 \\ \delta\dot{\mathbf{x}}_4 \end{Bmatrix} = \begin{bmatrix} \mathbf{F}_{(red)11} & \mathbf{F}_{(red)12} & \mathbf{F}_{(red)13} & \mathbf{F}_{(red)14} \\ \mathbf{0} & \mathbf{F}_{(red)22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\mathbf{x}_3 \\ \delta\mathbf{x}_4 \end{Bmatrix} + \begin{Bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.3}$$

This model was shown by Negast [30] to be sufficient to represent the 93-state model accurately. Note that the submatrix indices used in representing the 39-state model are not identical to those used in outlining the 93-state INS error model. The relationship between the two models is given in Tables A.5 and A.6 of Appendix A.

3.3.1.2 The INS Filter Design Model. The INS filter design model is the model which would be used by the EKF operating on an aircraft using the PLS described here. The limited computational power available and the requirement for real-time processing motivates making the dimension of the filter model as small as possible. The INS filter model is comprised of 11 states (the first nine being the standard Pinson error model states): 6 misalignment errors, 3 velocity biases, and 2 states for barometric stabilization. Table A.9 shows the 11 INS states used in the filter model.

3.3.1.3 The INS Measurement Model. The only measurement model associated with INS is that for barometric altimeter aiding. The altimeter aiding is used to compensate for the instability inherent in the vertical channel of the INS. The altimeter output Alt_{Baro} is modeled as the sum of the true altitude h_t, the total error in the barometric altimeter δh_B, and a random measurement noise v. Similarly, the INS calculated altitude Alt_{INS} is the sum of the true altitude and the INS error in vehicle altitude above the reference ellipsoid, δh. A *difference* measurement is used to eliminate the unknown true altitude, h_t, resulting in Equation (3.4):

$$\begin{aligned} \delta z &= Alt_{INS} - Alt_{Baro} \\ &= [h_t + \delta h] - [h_t + \delta h_B - v] \\ &= \delta h - \delta h_B + v \end{aligned} \tag{3.4}$$

INS error in vehicle altitude above the reference ellipsoid, δh, and total barometric altimeter correlated error, δh_B, are states 10 and 11 in the 11- and 39-state INS models.

3.3.2 The Radar Altimeter Model. The measurement equation of the radar altimeter is based on the difference between the INS-predicted altitude Alt_{INS} and

the radar altimeter predicted altitude Alt_{Ralt}:

$$\begin{aligned} \delta z &= Alt_{INS} - Alt_{Ralt} \\ &= [h_t + \delta h] - [h_t - v] \\ &= \delta h + v \end{aligned} \tag{3.5}$$

The errors in the radar altimeter are modeled as white noise with no time-correlated component. This may be a rather crude model, but should be sufficient to demonstrate performance trends. Note that no additional states are required with the addition of this radar altimeter model.

The radar altimeter measurement noise covariance R_{Ralt} is a function of aircraft altitude above ground level (AGL) and will be the same in the truth and filter models. The radar altimeter noise covariance from [11] is altitude-dependent and is given by

$$R_{Ralt} = \left\{[0.01]^2 * [\mathrm{AGL}_{true}]^2\right\} + 0.25\ ft^2 \tag{3.6}$$

3.3.3 GPS Models. The GPS (also DGPS) generates user position based on "known" ranges to satellites at "known" positions. The satellites themselves transmit their position in space (in the form of ephemeris data) as accurately as it is known and the exact time (also a best estimate) at which the transmission is sent. The actual range information is calculated based on knowledge of the satellite position and the finite propagation speed of the electromagnetic radiation emitted from the satellite.

3.3.3.1 The GPS Truth Model. The GPS model used in this work was developed by past researchers at AFIT [10, 30, 41, 42]. The dynamics and measurement equations for the full 30-state truth model are presented in this section and a tabular listing of the states is shown in Table A.7 of Appendix A.

Five types of error sources are modeled in the GPS truth state equations. Two of the five error sources are insignificant when differential corrections are applied (DGPS). The first error type, user clock error, is common to all SV's. The remaining four error types are unique to each SV. The first two states represent user clock errors and are modeled as:

$$\begin{Bmatrix} \dot{x}_{Uclk_b} \\ \dot{x}_{Uclk_{dr}} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x_{Uclk_b} \\ x_{Uclk_{dr}} \end{Bmatrix} \tag{3.7}$$

where

x_{Uclk_b} = range equivalent of user clock bias

$x_{Uclk_{dr}}$ = velocity equivalent of user clock drift

The initial state estimates and covariances for these states were chosen to be consistent with previous AFIT research [3, 10, 30, 41, 42] and are:

$$\begin{Bmatrix} \widehat{x}_{Uclk_b}(t_0) \\ \widehat{x}_{Uclk_{dr}}(t_0) \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.8}$$

and

$$\mathbf{P}_{Uclk_b,Uclk_{dr}}(t_0) = \begin{bmatrix} 9.0 \text{ x } 10^{14}\ ft^2 & 0 \\ 0 & 9.0 \text{ x } 10^{10}\ ft^2/sec^2 \end{bmatrix} \tag{3.9}$$

Because these error sources are a function of the user equipment, they are common to all the SV's. Recall that each of the remaining error types is specific to each SV, denoted by a subscript j.

The second error type is the code loop error δR_{cl_j}. The code loop is part of the user equipment shared by all the SV's, but its error magnitude is relative to each SV. The work of Negast [30] shows that this error source may be disregarded in the DGPS model. The third GPS error type is the result of atmospheric interference with the EM signals broadcast by each SV, specifically, ionospheric and tropospheric delay, δR_{ion_j}, and δR_{trop_j}. The code loop error, tropospheric delay, and ionospheric

delay are all modeled as first-order Markov processes with time constants shown in Equation (3.10). All three are driven by zero-mean white Gaussian noise with strength shown in Equation (3.13). The fourth error source is due to inaccuracies of the clocks on board the individual SV's, δR_{Sclk_j}. Like code loop error, this error source is also removed when differential corrections are applied. The final GPS error source is based on line-of-sight errors between the SV's and the receiver, δx_{s_j}, δy_{s_j}, and δz_{s_j}. The DGPS models for these states are shown later in Equations (3.24) - (3.27). Note that, if DGPS were used exclusively in this research, the states for code loop error and satellite clock error could be removed completely. The modifications required for DGPS are summarized in Section 3.3.3.4.

$$\begin{Bmatrix} \dot{\delta R}_{cl_j} \\ \dot{\delta R}_{trop_j} \\ \dot{\delta R}_{ion_j} \\ \dot{\delta R}_{Sclk_j} \\ \dot{\delta x}_{s_j} \\ \dot{\delta y}_{s_j} \\ \dot{\delta z}_{s_j} \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{500} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{1500} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{cl_j} \\ \delta R_{trop_j} \\ \delta R_{ion_j} \\ \delta R_{Sclk_j} \\ \delta x_{s_j} \\ \delta y_{s_j} \\ \delta z_{s_j} \end{Bmatrix} + \begin{Bmatrix} w_{cl} \\ w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.10}$$

with initial covariances given by

$$\mathbf{P}_{GPS} = \begin{bmatrix} 0.25\ ft^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0\ ft^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0\ ft^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25\ ft^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25\ ft^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25\ ft^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25\ ft^2 \end{bmatrix} \tag{3.11}$$

and noise means and covariances given by

$$E[\mathbf{w}_{GPS}(t)] = \mathbf{0} \tag{3.12}$$

$$E[\mathbf{w}_{GPS}(t)\mathbf{w}_{GPS}^T(t+\tau)] = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.004 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ft^2/sec \cdot \delta(\tau) \tag{3.13}$$

The full 30-state GPS dynamics matrix is not shown explicitly but may be easily constructed by augmenting Equation (3.7) and four copies (one for each SV) of Equation (3.10).

3.3.3.2 The GPS Filter Design Model. Research has shown [26, 30] that the two user clock error states provide a sufficient filter model for GPS. The primary argument is that the errors modeled for the 28 other GPS (20 other DGPS) states (assuming four SV's) are small when compared to the user clock errors which are common to all SV's. By increasing the dynamics driving noise and re-tuning the filter, the overall performance of the integrated navigation system can be maintained. The GPS filter model is given by Equation (3.7) plus noise:

$$\begin{Bmatrix} \dot{x}_{Uclk_b} \\ \dot{x}_{Uclk_{dr}} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x_{Uclk_b} \\ x_{Uclk_{dr}} \end{Bmatrix} + \begin{Bmatrix} w_{clk_b} \\ w_{clk_{dr}} \end{Bmatrix} \tag{3.14}$$

Because simulations were only performed using DGPS, and not GPS, no experimentally-determined Q tuning values for GPS are shown.

3.3.3.3 The GPS Measurement Model. The pseudorange measurements available to the GPS receiver are the sum of the true range, several error sources, and a random noise:

$$R_{GPS_j} = R_{t_j} + \delta R_{cloop_j} + \delta R_{trop_j} + \delta R_{ion_j} + \delta R_{Sclk_j} + \delta R_{Uclk} - v_j \tag{3.15}$$

or, after differential corrections are applied:

$$R_{GPS_j} = R_{t_j} + \delta R_{trop_j} + \delta R_{ion_j} + \delta R_{Uclk} - v_j \tag{3.16}$$

where

R_{GPS_j} = GPS pseudorange measurement, from SV_j to user
R_{t_j} = true range, from SV_j to user
δR_{cloop_j} = range error due to code loop error
δR_{trop_j} = range error due to tropospheric delay
δR_{ion_j} = range error due to ionospheric delay
δR_{Sclk_j} = range error due to SV_j clock error
δR_{Uclk} = range error due to user clock error
v_j = zero − mean white Gaussian measurement noise

Because R_t is not available to the filter, a substitution will be make to eliminate this term. First, the satellite position vector $\mathbf{X}_S$ and the user position vector $\mathbf{X}_U$ are defined as:

$$\mathbf{X}_U = \begin{Bmatrix} x_u \\ y_u \\ z_u \end{Bmatrix}^e , \quad \mathbf{X}_S = \begin{Bmatrix} x_s \\ y_s \\ z_s \end{Bmatrix}^e \tag{3.17}$$

where the superscript e denotes coordinates in the earth-centered earth-fixed (ECEF) frame. The pseudorange from the user to the satellites calculated by the INS, R_{INS}, is the difference between the PLS-calculated user position, $\mathbf{X}_U$, and the satellite

position given by the ephemeris data, $\mathbf{X}_S$:

$$R_{INS} = |\mathbf{X}_U - \mathbf{X}_S| = \left| \left\{ \begin{array}{c} x_U \\ y_U \\ z_U \end{array} \right\}^e - \left\{ \begin{array}{c} x_S \\ y_S \\ z_S \end{array} \right\}^e \right| \tag{3.18}$$

An equivalent form of Equation (3.18) is:

$$R_{INS} = \sqrt{(x_U - x_S)^2 + (y_U - y_S)^2 + (z_U - z_S)^2} \tag{3.19}$$

With perturbations representing errors in $\mathbf{X}_U$ and $\mathbf{X}_S$, Equation (3.19) can be written in terms of the true range and a truncated first-order Taylor series:

$$\begin{aligned} R_{INS} = \; & R_t + \frac{\partial R_{INS}(\mathbf{X}_S, \mathbf{X}_U)}{\partial \mathbf{X}_S}\Big|_{(\mathbf{X}_S, \mathbf{X}_U)_{nom}} \cdot \delta \mathbf{X}_S \\ & + \frac{\partial R_{INS}(\mathbf{X}_S, \mathbf{X}_U)}{\partial \mathbf{X}_U}\Big|_{(\mathbf{X}_S, \mathbf{X}_U)_{nom}} \cdot \delta \mathbf{X}_U \end{aligned} \tag{3.20}$$

The solution for R_{INS} is found by substituting Equation (3.19) into Equation (3.20) and evaluating the partial derivatives to get:

$$\begin{aligned} R_{INS} = \; & R_t - \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_U \\ & + \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_S + \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_S + \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_S \end{aligned} \tag{3.21}$$

Finally, the truth model GPS pseudorange *difference* measurement is given as:

$$\begin{aligned} \delta z = \; & R_{INS} - R_{GPS} \\ = \; & - \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_U \\ & + \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_S + \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_S + \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_S \\ & - \delta R_{cloop} - \delta R_{trop} - \delta R_{ion} - \delta R_{Sclk} - \delta R_{Uclk} + v \end{aligned} \tag{3.22}$$

The user position errors in Equation (3.22) can be derived from the first three (position error) states of the filter or truth model using an orthogonal transformation [2].

The filter design measurement model for the GPS measurement does not contain terms for the errors due to code loop variations, atmospheric delays, satellite clock deviations, or errors in ephemeris-given satellite position. The filter GPS measurement model can be written as:

$$\delta z = -\left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_U - \delta R_{Uclk} + v \quad (3.23)$$

3.3.3.4 The DGPS Truth Model. Differential GPS (DGPS) is modelled very similarly to GPS. The justifications for the differences were given above in Sections 3.3.3.1 and 3.3.3.3 at the points during the GPS development where the differences were relevant. This section will present the mathematical specifics of DGPS as they differ from those of GPS. A tabular listing of the 22 DGPS states is shown in Table A.8 of Appendix A.

The error sources for DGPS are identical to those for GPS, with the noted exceptions that the code loop and satellite clock errors may be disregarded when using differential corrections. The differential equation for the DGPS error states becomes:

$$\begin{Bmatrix} \dot{\delta R}_{trop_j} \\ \dot{\delta R}_{ion_j} \\ \dot{\delta x}_{s_j} \\ \dot{\delta y}_{s_j} \\ \dot{\delta z}_{s_j} \end{Bmatrix} = \begin{bmatrix} -\frac{1}{500} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{1500} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{trop_j} \\ \delta R_{ion_j} \\ \delta x_{s_j} \\ \delta y_{s_j} \\ \delta z_{s_j} \end{Bmatrix} + \begin{Bmatrix} w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (3.24)$$

with initial covariances (note the differences from Equation (3.11)) given by

$$\mathbf{P}_{DGPS} = \begin{bmatrix} 1.0\ ft^2 & 0 & 0 & 0 & 0 \\ 0 & 1.0\ ft^2 & 0 & 0 & 0 \\ 0 & 0 & .35\ ft^2 & 0 & 0 \\ 0 & 0 & 0 & .35\ ft^2 & 0 \\ 0 & 0 & 0 & 0 & .35\ ft^2 \end{bmatrix} \tag{3.25}$$

and noise means and strengths (note the differences between Equation (3.13)) given by

$$E[\mathbf{w}_{DGPS}(t)] = \mathbf{0} \tag{3.26}$$

$$E[\mathbf{w}_{DGPS}(t)\mathbf{w}_{DGPS}^T(t+\tau)] = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0.0004 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} ft^2/sec \cdot \delta(\tau) \tag{3.27}$$

3.3.3.5 The DGPS Filter Design Model. The filter design model for DGPS is identical to that for GPS. The only difference lies in the gains used to tune the filter when using DGPS. The filter Q values used to tune the filters in this research may be found in Appendix C.

3.3.3.6 The DGPS Measurement Model. After differential corrections are applied, the measurement equation is

$$R_{DGPS_j} = R_{t_j} + \delta R_{trop_j} + \delta R_{ion_j} + \delta R_{Uclk} - v_j \tag{3.28}$$

as noted in Section 3.3.3.3. After noting the removal of the code loop and satellite clock error sources in Equation (3.15), leaving Equation (3.28), the remainder of the DGPS measurement model development is identical to that for GPS.

3.3.4 The Pseudolite Model. Pseudolite measurements are treated as measurements from a 5^{th} GPS satellite, with the following three exceptions in the truth model.

- Transmissions from a pseudolite do not pass through the ionosphere, so there is no ionospheric delay error term for a pseudolite measurement. The tropospheric delay error term is still included.
- The pseudolite is assumed to be located at a surveyed position, so there is no uncertainty in the "SV" position for a pseudolite measurement.
- There is assumed to be no bias or drift errors in the "satellite" clock for a pseudolite measurement.

As may be inferred from the list above, as far as the filter is concerned, there is no difference between a satellite measurement and the pseudolite measurement.

3.4 Failure Models

This section discusses the methods used to model failures in the MMSOFE simulations. For each failure type to be simulated, the corresponding MMAE-based FDI methods used to detect that failures are also discussed.

3.4.1 Simulation Failure Models. Seven variations of three different failure types are modeled in simulation. The seven failure cases are presented with actual values in Table 3.1. (The contents of Table 3.1 will be discussed in detail later in this section.) A short description of each failure type follows:

1. Interference/Jamming – Interference is modeled as a sudden increase in the measurement noise associated with all four SV's, resulting in lower carrier-to-noise ratios, C/N_0, in the GPS receiver. This failure is induced in all SV measurements because interference/jamming is assumed to occur at the receiver, which will affect all four (five, if the pseudolite is included) channels

simultaneously. The interference noise, R_{int}, is added to the truth model measurements only. Specific magnitudes of interference will not be considered as in previous work at AFIT [41, 42]; rather, real-world interference will be allowed to take on selected values within the interference parameter space spanned by the MMAE filter bank. Emphasis will be placed on determining (demonstrating) the capability of MMAE to detect and identify interference failures of unspecified magnitude quickly.

GPS *jamming* is used to refer to the total loss of useful GPS transmissions due to very large rf interference. A GPS jamming failure is well-modelled (and much more easily modelled) via *very large* measurement noise. When the MMAE algorithm detects very large real-world measurement noise $\mathbf{R}$ then the corresponding measurements will be *very* lightly weighted by the elemental Kalman filters; the effect is essentially the same as if those measurements were never received, hence the use of the term "interference/jamming" throughout this report.

2. Spoofing – Spoofing is modeled as a bias added to the measurements associated with all GPS SV's. The addition of the bias (if it is undetected) has the effect of placing a bias on the position solution of the GPS system. Specific magnitudes of spoofing failures will not be calculated to produce some effective aircraft position error, according to a single strategy of spoofing, as the emphasis of this work is an assessment of the FDI capabilities of MMAE. Instead, a *range* of possible spoof magnitudes will be investigated to exercise the MMAE algorithm fully. Two models of spoofing bias addition are used and will be discussed shortly.

   In this research, spoofing is modelled as a *uniform* bias addition to each of GPS (and pseudolite) pseudorange. Spoofing failures modelled in this way result in (mainly) vertical changes in the user's GPS navigation solution (see Figure 3.2). A better model (more representative of probable malignant spoof-

ing techniques) of spoofing would be the addition of calculated biases, *specific to each SV*, designed to yield a desired net change in the user's navigation solution; such a real-world spoof would probably be designed to produce a *horizontal* shift in the navigation solution so that unaffected sensors (baro and radar altimeters) could not be used to eliminate the error over several measurement updates. Although not an entirely accurate model of probable real-world spoofing changes, the uniform bias model is much more easily implemented in software and does correctly yields cohesive changes in the user's GPS navigation solution which effectively test the spoofing FDI performance of the MMAE algorithm.

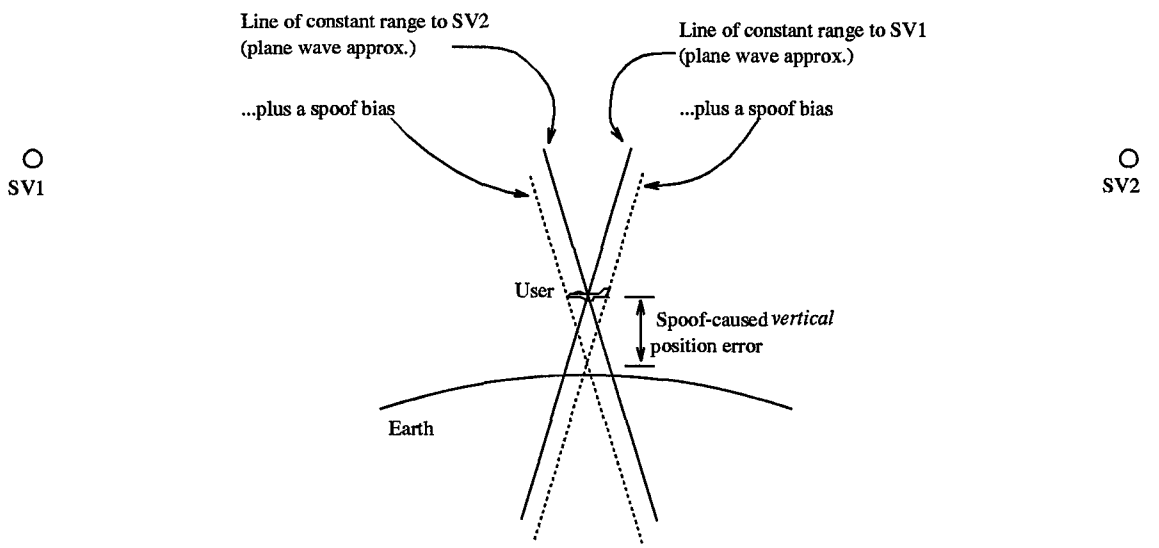

Figure 3.2 Uniform-bias spoofing model: vertical effect

(a) Spoofing Model A – A bias is added suddenly in time to the measurements associated with each SV, with the net effect being a jump in the GPS-derived aircraft position. More sophisticated spoofers may have a fairly accurate estimation of the GPS measurements being received by the aircraft and can *discretely* add a small, "undetectable" step bias to these measurements. A less subtle spoofer would have to use a larger bias

to ensure effectiveness in corrupting the measurements while running the risk of being detected.

(b) Spoofing Model B – A more intelligent (and considerably more difficult to detect) spoofing failure is modeled as a steadily increasing (ramp) bias value added to each SV pseudo-measurement. Smaller ramp rates are more difficult to detect than large rates because a slow change may be hidden in the noise expected by the elemental filters. A range of ramp rates will be used with the purpose of determining the detection and isolation capabilities of MMAE. Ramp spoofing will be termed "intelligent" for the remainder of this report.

Table 3.1 summarizes the different failures that are simulated in this research. These failures are induced on each of the PLS hardware configurations under test.

Table 3.1 Failure Types and Models [41,42]

	Fail Type	Description	Fail Method
0	No Failure	N/A	$R = R_0$
1	Interference	Jump in Measurement Noise	Increase R from R_0 to $R_0 \times \text{Magn}_{int}$
2	Step Spoofing	Step Bias on each Pseudorange	Add bias=Magn_{spf} to each GPS measurement
3	Ramp Spoofing	Ramp Bias on each Pseudorange	Add bias=$\text{Magn}_{spf} \times (t - t_0)$ to each GPS measurement

3.4.2 MMAE Failure Models. A bank of three elemental filters in an MMAE structure is used to detect and isolate interference failures. One filter is tuned for operation when no interference failure has occurred (i.e. measurement noise $\mathbf{R} = \mathbf{R_0}$). The two remaining elemental filters are each tuned for best operation with increased levels of measurement noise $\mathbf{R}$. An increase in measurement noise by a factor of 2000 (i.e. measurement noise $\mathbf{R} = \mathbf{R_0}$x2000) is assumed to be the highest level of real-world interference that will be encountered. The third elemental

filter models this highest assumed level of interference; it is tuned for best operation when $\mathbf{R} = \mathbf{R}_0$x2000. It is hypothesized that the interference elemental filters will operate most effectively when the levels of their respective interference assumptions are separated by approximately an order of magnitude. This hypothesis will be verified experimentally. Based on this hypothesis, the second elemental filter is tuned for $\mathbf{R} = \mathbf{R}_0$x200.

MMAE-based FDI is accomplished via the information contained in the residuals of the elemental filters. The residuals produced by a filter with an accurate model of the real world are zero-mean, white, Gaussian, and have covariance $\mathbf{HP}^{-}\mathbf{H}^{T} + \mathbf{R}$. For this discussion, we will term these residual characteristics "good". If the residuals of any elemental filter have good characteristics, then that filter's hypothesis of the real world is assumed to be correct. For example, if the $\mathbf{R} = \mathbf{R}_0$x200 interference filter, filter two, displays good residuals, then the probability of model correctness flows to filter two and the MMAE algorithm detects a real world interference level 200 times $\mathbf{R}_0$. If the level of jamming seen by the filter lies between, for example, the 200x$\mathbf{R}_0$ and 2000x$\mathbf{R}_0$ levels of interference modeled by elemental filters two and three, then the residuals of those two filters will both display some good characteristics. In this case the probability of having the correct model will be shared by models two and three, and the level of real world interference is isolated by blending the measurement noise hypotheses of these two filters based on the computed probability of each being correct.

Five elemental filters are used to detect and isolate spoofing failures of both step and ramp types. As with interference failures, the first elemental filter is tuned for best operation with no failure induced. The second elemental filter is tuned for best operation given a small measurement bias added to the SV pseudoranges. A third elemental filter is tuned based on a much larger measurement bias added to the SV pseudoranges. The final two elemental filters model correspondingly small and large measurement biases of a negative magnitude. The choice of this spoofing

filter bank configuration is made based on the same order-of-magnitude separation arguments used to hypothesize the configuration of the interference MMAE filter bank. In each case, the effectiveness of the proposed filter bank configurations will be evaluated experimentally and changed as required for best FDI performance.

For step-type spoofing failures, MMAE-based FDI is conceptually done in exactly the same way jamming failures are detected and isolated (although the next chapter will show the need for, and implementation of, a *moving-bank "pseudo-residual"* MMAE to handle spoofs, rather than a standard MMAE). Probability flows to the failure models (or model, in the case of an exact hypothesis match) that most closely match the real world spoofing condition. The level of real world spoofing is estimated by blending the spoofing hypotheses of the elemental filters based on the computed probability associated with each filter. Because of the blending capability of MMAE, there is no need to hypothesize ramp bias failures. As the value of the ramp increases, the hypothesis probability should gradually flow from one model to the next. The resulting blended estimate of the magnitude of the real world spoofing failure will follow the growth of the actual ramp failure.

The final steps of this research will involve identification of jamming or spoofing failures using all of the eight elemental filters described above (one nominal, three jamming strengths, and four spoofing biases). These simulations will establish the performance of MMAE FDI under the operational-like assumption that either jamming or spoofing is possible. The greatest difficulty for the FDI algorithm under this assumption will be disambiguating between the failure types. Performance analyses will argue for a two-tiered algorithm, one seeking spoofs and the second seeking interference/jamming, rather than a single MMAE attempting to detect all failure modes simultaneously.

3.5 Simulation Software

Multimode Simulation for Optimal Filter Evaluation (MSOFE) is a general-purpose, multimode simulation program for designing integrated systems that employ optimal (Kalman) filtering techniques and for evaluating their performance [29]. The general-purpose construction of MSOFE allows its application to a wide variety of user-specific problems with a minimal amount of new software development. The United States Air Force uses MSOFE for the validation of all systems that use optimal filtering techniques. MSOFE provides Monte Carlo and covariance simulation modes.

Multiple Model Simulation for Optimal Filter Evaluation (MMSOFE) was developed by Nielsen [31, 32] at the Air Force Institute of Technology to support the analysis of systems using a multiple model adaptive filter structure. MMSOFE is written as an extension to MSOFE and is based on the same core code. The MMSOFE program propagates multiple filters forward in parallel while performing the hypothesis probability and blending calculations required for MMAE and other multiple model algorithms. The Monte Carlo simulation mode of MMSOFE is used in all phases of the work.

3.6 Summary

This chapter has presented the models used for simulating the PLS, including the navigation component models and the failure implementation and detection models. The failure detection and simulation methods used in this research have also been discussed. Results and analysis of these simulations are presented in Chapter 4.

4. Results and Analysis

4.1 Introduction

This section reviews the failure models and navigation component configurations used in this research, along with a brief discussion of why each is of interest. A standard MMAE algorithm will be used subsequently to address detection of jamming/interference (increased measurement noise), whereas a moving-bank MMAE will be shown to be more appropriate to detect and compensate for the onset of spoofing (measurement bias).

4.1.1 Failure Models. Jamming is the total loss of GPS satellite transmissions due to heavy rf interference. With the loss of GPS signals, navigation is totally dependent on the remaining sensors, and for this study, almost entirely based on the onboard INS. No INS currently in production, or even realistically foreseen, has error characteristics small enough to be the sole primary sensor in a PLS. While it would be of interest to observe the inflight (not landing) navigation performance of an MMAE filter bank specifically modelling this failure, that extension is beyond the scope of this work. The problem of jamming was briefly considered to confirm the above presumptions. These results are presented in Section 4.2.2.

RF interference not strong enough to cause total loss of the GPS transmissions is modelled as increased measurement noise. Interference models low-power jamming from hostile sources, as well as navigation in areas of abnormally high rf activity, as in the immediate area of television or radio microwave relay stations. Additionally, as antenna and signal processing technology improves, devices which now result in a total loss of GPS transmissions may, in the future, only have the effect of increasing the measurement noise associated with the GPS receiver. It is anticipated that the MMAE algorithm will do a good job of detection and isolation against this failure type.

Spoofing of a GPS receiver occurs when a hostile source presents GPS-like transmissions to the receiver which are slightly stronger than the real GPS signals and which will give an incorrect navigation solution. Intelligent spoofers would be able to present GPS-like signals giving a navigation solution with very little initial offset from the real GPS navigation solution. Once the receiver accepts these false signals they are "walked off" the real solution. This is modelled as a ramp offset from the true GPS solution. Less intelligent spoofers would present signals with a very large, easily identifiable step offset. The onset of unintelligent spoofing will be seen as a large spike in the measurement residuals within the elemental filters of the MMAE (or within a single non-adaptive Kalman filter if one were used rather than an MMAE algorithm). Identification is easily done based on that spike. Intelligent spoofing is potentially much more difficult to detect. For ramp offsets which grow slowly enough to be hidden in the noise, there are no means available for detection.

Realistically speaking, intelligent spoofing would be tremendously difficult to implement. Prior to, and over the duration of, the spoof, the spoofer would have to know the position of the aircraft with very nearly the same precision as the aircraft's onboard filters. This is required so that the spoofer could maintain a slowly growing offset in the false GPS solution. It is expected that the MMAE algorithm will effectively detect and isolate spoofing with a step onset or a ramp onset with a slope large enough to be distinguished from the noise.

4.1.2 Navigation Component Configurations. The four navigation component configurations used are shown in Table 4.1. Case one represents the best configuration available. Cases two and three show, respectively, the effect on FDI performance of eliminating the pseudolite and radar altimeter sensors. Case four shows the effect of a much lower quality INS, which might be representative of INS's available to civil aviation groups.

Table 4.1 Navigation Component Cases

Case	INS Type	GPS Type	Altimeter Aiding
1	0.4nm/hr	DGPS and Pseudolite	Baro and Radar Alt
2	0.4nm/hr	DGPS	Baro and Radar Alt
3	0.4nm/hr	DGPS	Baro
4	4.0nm/hr	DGPS	Baro

4.2 FDI and Navigation Results

4.2.1 Summary Plots. Two types of plot summaries will be used to display the simulation results of this study. Each simulation was performed over the interval 3810 to 3910 seconds, which represents the final 100 seconds before touchdown in the flight profile used in this study. A lower bound probability value of 0.01 was used in all cases shown. The value of 0.01 was experimentally determined; smaller lower bounds slowed the response of the p_k's to onsets of their corresponding failures (see Section 2.3.1). Larger lower bound values did not provide any improvement in the p_k response time and reduced the total probability available for assignment to the filter with the currently correct failure hypothesis.

Figure 4.1 is representative of the first type of summary plot. The first subplot shows the magnitude time history of the induced failure, indicated by dark x's. The second trace on the first subplot shows the MMAE filter bank's blended estimate of the real-world failure magnitude. The y-axis of this first subplot is the magnitude of the total variance multiplier (times the no-failure noise variance) present/detected at any time during the simulation. Recall that interference is modelled as a large increase in the measurement noise variance. The remaining subplots show the 15-run mean of the probability (p_k) time histories for each of the elemental filters. The 15-run standard deviation of each of the p_k's is also shown. The y-axis labels of

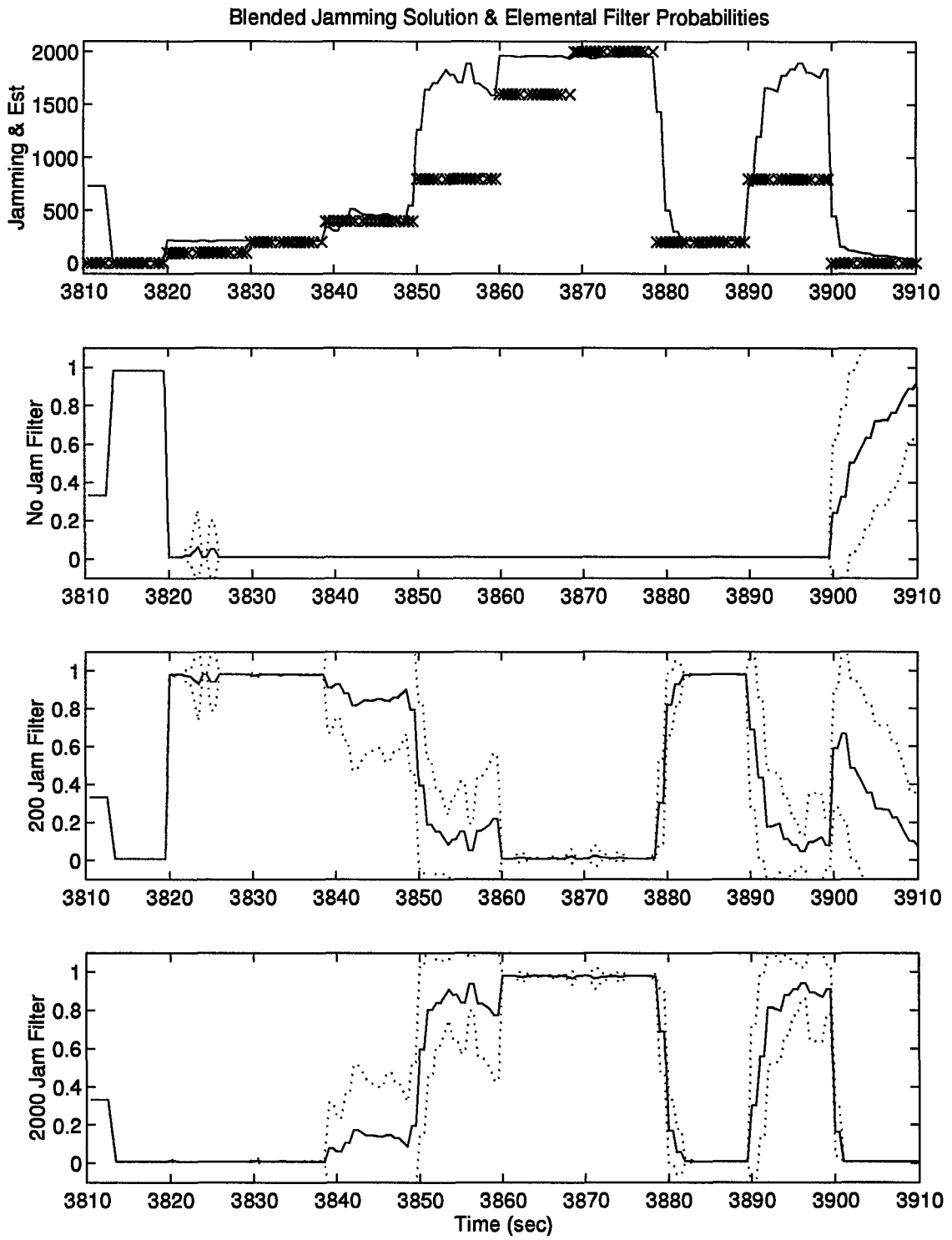

Figure 4.1 FDI Performance and Elemental Filter Probabilities, Interference Failures, Three Elemental Filters - Nav Case 1

these probability plots describe the failure assumptions of that elemental filter. The reader should note that, as is expected, the blended estimate of the failure magnitude visually corresponds to the sum of the plotted p_k values times their respective failure assumptions as indicated in the y-axis labels.

Figure 4.2 is representative of the second type of summary plot. Three subplots will always be included, showing the latitude, longitude, and altitude errors (in feet) of the MMAE blended navigation solution. The solid trace represents the blended filter-predicted sigma bound for that variable. The dotted line shows the actual 15-run mean of the same variable and the dashed line shows the 15-run mean $\pm$ one standard deviation.

4.2.2 Jamming. Total GPS failure is modelled in this work by zeroing the **H** (measurement) matrix entries corresponding to the GPS measurements. Better performance might be realized if the MMAE bank were to include a filter only modelling INS and altimeter states, as opposed to the current model including the GPS states and using the zero **H** matrix entries. This extension was not a primary focus of this work because DGPS jamming is well modelled (and more easily simulated) as *very* strong interference. When a filter assumes a *very* large noise variance **R** associated with some measurements, then those measurements are essentially discarded by the filter when it calculates its state updates. Telling the filter (via large **R** values) not to use some measurements has the same effect as not including those measurements at all.

4.2.3 Interference. MMAE FDI was, as expected, effective against interference failures. However, as can be seen in Figure 4.1, the filter bank has some difficulty in blending its two noise hypotheses. With the aid of hindsight, it could have been expected that the MMAE algorithm would be prone to bouncing from hypothesis to hypothesis without much blending, and to have a tendency to choose the model with the larger assumed noise level. As the real world noise level becomes even slightly

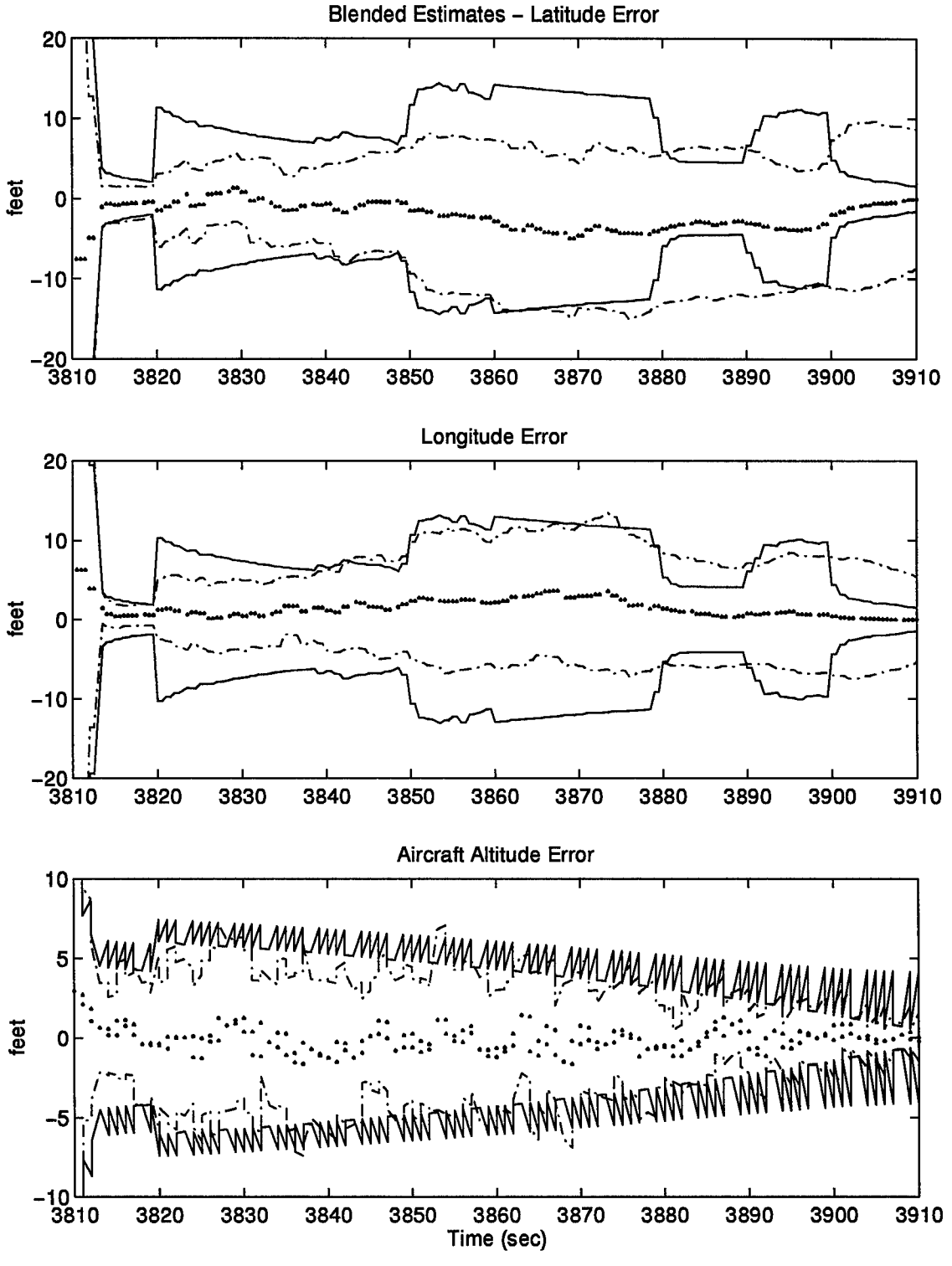

Figure 4.2 Navigation Performance, Interference Failures - Nav Case 1

greater than that assumed by one filter, that filter's residuals very quickly look bad (i.e., its $[\mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)]$ becomes a large value) because the measurements are consistently violating its assumptions. Although the measurement noise variance is much less than it expects, the filter assuming a larger noise value sees measurements that do consistently fall within its expected variance (its $[\mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)]$ is considerably smaller). Probability quickly flows to the filter with the larger noise variance assumption. The state estimation provided by the three-filter (assuming 1x, 200x, and 2000x the original tuned $\mathbf{R}$ values) interference bank is shown in Figure 4.2. The 2000x variance level was chosen as the upper-bound of possible real-world interference. It was hypothesized that the elemental filters should be separated by approximately an order of magnitude to prevent ambiguity. The 200x variance filter was selected on this basis. Filters with still smaller noise variances hypotheses (20x for example) were not included because noise variances smaller than about 100x have virtually no effect on the navigation system's performance. A larger upper bound on the assumed real-world noise variance might just have easily been used, resulting in a different set of elemental filters. It is supposed that the FDI performance of the MMAE algorithm, as shown in the simulation results, would not differ dramatically with such a change. As can be seen in Figures 4.1 and 4.2, the estimation performance of this filter bank is quite good, and very effectively increases the filter bank's predicted bounds on the state estimation error standard deviations and so prevents the unacceptable performance or filter divergence that might otherwise result.

Figures 4.3 and 4.4 show the performance of a single Kalman filter subjected to interference and spoofing failures, respectively. The unacceptable performance seen in these figures (note the scale changes from Figure 4.2) provides both a *benchmark* for performance and a *strong motivation* for adaptive filtering via MMAE techniques.

Figure 4.5 shows an attempt to achieve better estimation blending by including two more elemental filters with interference assumptions in the same range spanned by the original three filters, with the elemental filters now assuming 1x, 100x, 300x,

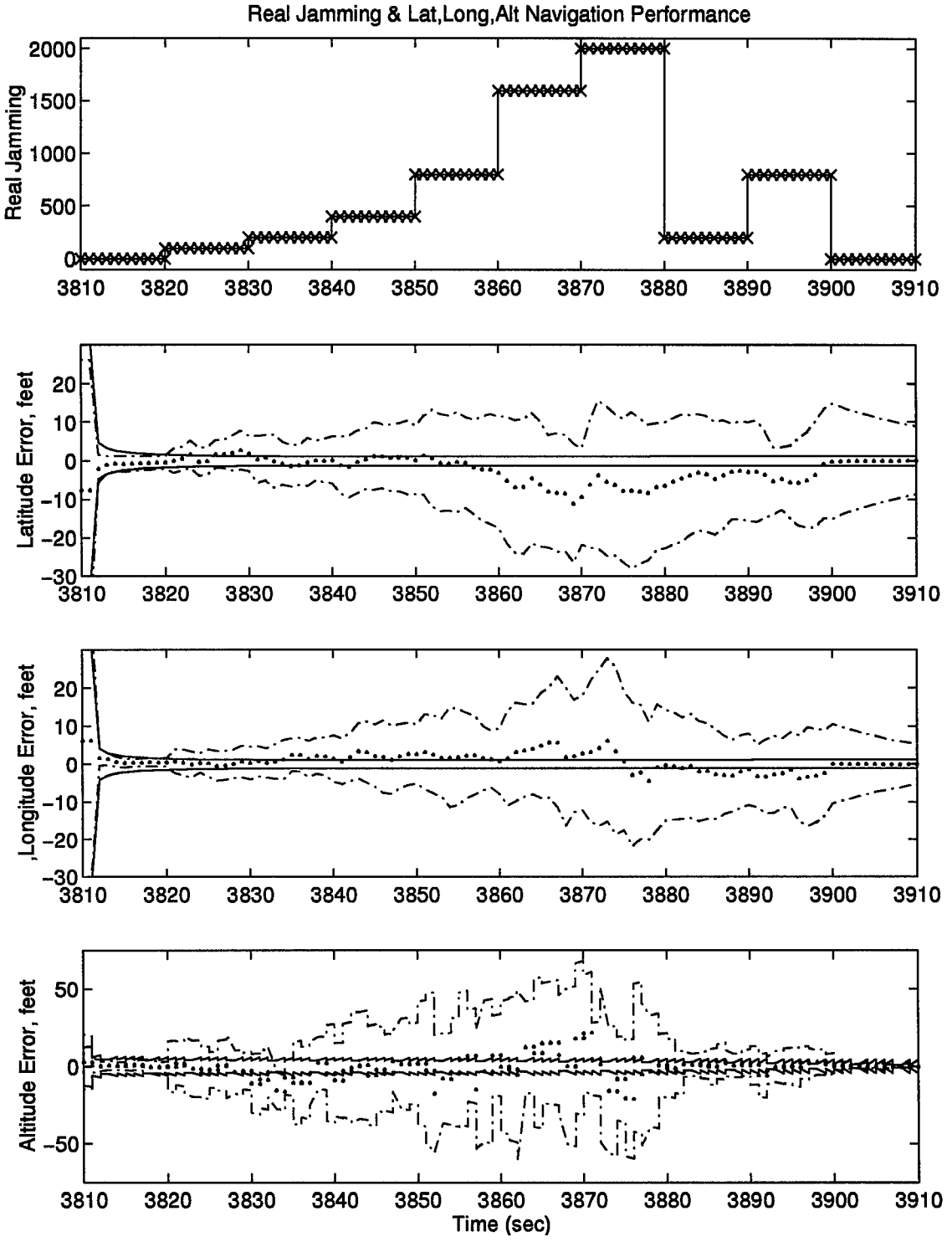

Figure 4.3 Navigation Performance, Interference Failures, *single* Kalman filter - Nav Case 1

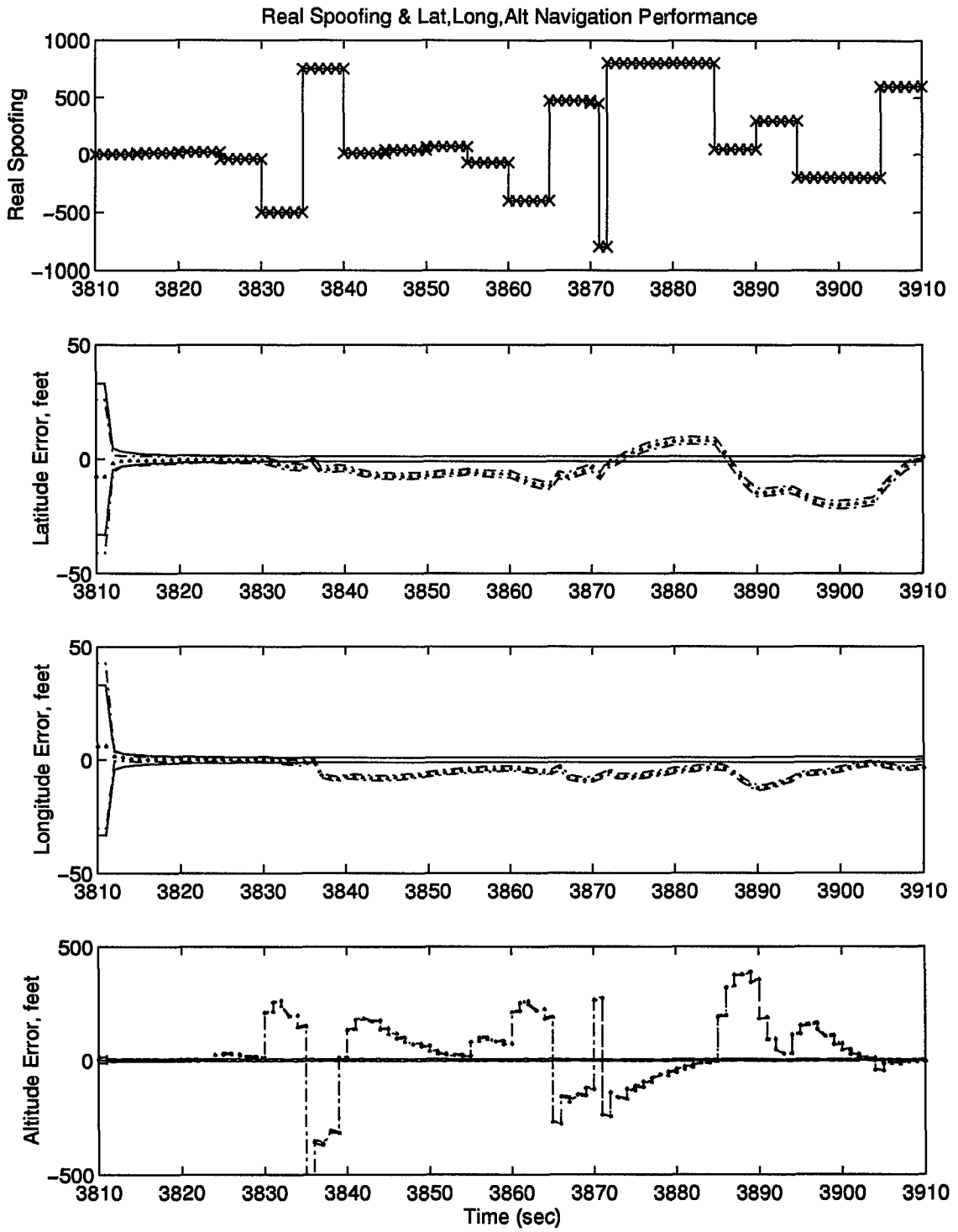

Figure 4.4 Navigation Performance, Spoofing Step Failures, *single* Kalman Filter - Nav Case 1

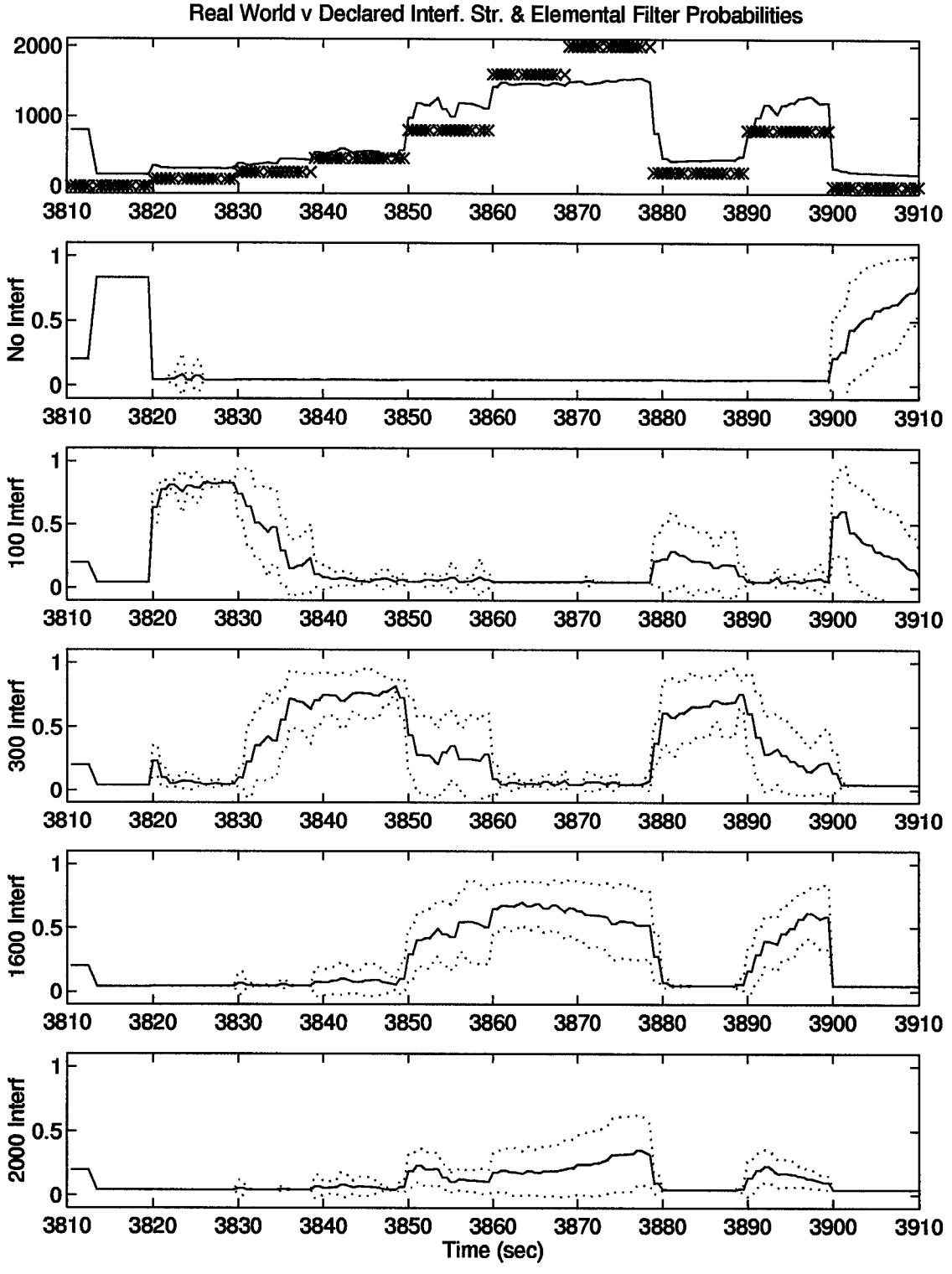

Figure 4.5 Interference FDI Performance and Elemental Filter Probabilities, Five Filters - Nav Case 1

1600x, and 2000x the original tuned **R** values. The values of these noise variance assumptions were chosen somewhat arbitrarily, with the purpose of testing the earlier hypothesis that an order of magnitude should separate the variance assumptions of the elemental filters. As might be expected (again with the aid of hindsight) the filters close together at the upper end of the interference range tend to confuse each other as their assumed noise variances differ by only twenty percent. Additional blending can be seen at low interference values, where the elemental filter's **R** values are still separated by substantial percentages, however, as is seen in the figure, that this blending takes place does not guarantee that better estimation performance is realized in the low true **R** range. The probability plot summary of this bank is included only to justify the selection of the three-filter interference bank for final implementation and so the state estimation of the five-filter interference bank is not shown. These results suggest the use of an MMAE filter bank spanning the interference range of interest and composed of elemental filters separated by approximately an order of magnitude in their parameter assumptions.

4.2.4 Spoofing. A five-filter (zero measurement bias, ±15 foot bias, and ±240 foot bias) MMAE bank was used for measurement bias detection. Assumed bias values of ±15 feet were included to allow the bank to fine tune its bias estimation. Filters separated by 15 feet are close enough in parameter space to provide clear blending and are far enough away to avoid ambiguity between the filter models. The bias assumption of the ±240 foot bias filter pair was determined as follows. It was found experimentally that an elemental filter was able to detect actual biases within 150 feet of its own assumed bias, while biases further removed than 150 feet resulted in zero (lower bound probability) p_k values for those elemental filters. It was decided that, for good blending, one half of this detection range should overlap the detection range of the next closest filter, e.g., the 225 foot bias assumption difference between the two positive filters at 15 feet and 240 feet is obtained as 150 feet(range) + 150 feet(range) - 75 feet(overlap). These filter assignments are,

of course, problem-specific. The reader will note that each of the elemental filters hypothesizes a constant bias offset. Ramp spoof offsets are estimated over time by blending the constant bias hypotheses of the same elemental filters.

The following paragraph discusses the summary plots of two spoofing simulations in order to motivate (1) the development of a modified MMAE algorithm to detect measurement offset failures, and (2) the use of moving-bank MMAE in conjunction with this modification. The questions raised in the next paragraph will be answered in Section 4.2.4.1, which develops the theory of the MMAE modification mentioned above. Throughout the remainder of this report, this modified MMAE algorithm will be referred to as "pseudo-residual" MMAE (PRMMAE) for reasons discussed in Section 4.2.4.1.

Figure 4.6 shows the spoofing detection performance of the *presumedly* standard MMAE algorithm (it was later realized that the MMAE algorithm actually implemented in the MMSOFE software was a non-moving-bank *pseudo-residual* MMAE, to be described in the next section). In order to observe the detection behavior of the algorithm clearly, each real-world spoofing level change was chosen to match exactly the spoofing level modelled by one of the elemental filters. The no-spoof filter correctly acquires the GPS satellites and initializes the state estimation. At the onset of the first spoofing block, the probability of the elemental filter assuming the *negative* of the real world spoof value spikes for one sample period (see Section 4.2.4.1). After this one sample period of useful information from the MMAE filter bank, the probabilities of all of the elemental filters become equal (see Section 4.2.4.1), suggesting that some kind of re-centering of the filter bank at the initial time when information is available may prove useful. Good state estimation is not recovered even when the real world bias is removed. While searching for a means of improving the meager detection performance shown in Figure 4.6, it was discovered that the the MMAE algorithm had been incorrectly implemented with respect to the addition of measurement biases ("pseudo-residuals" versus true residuals were used

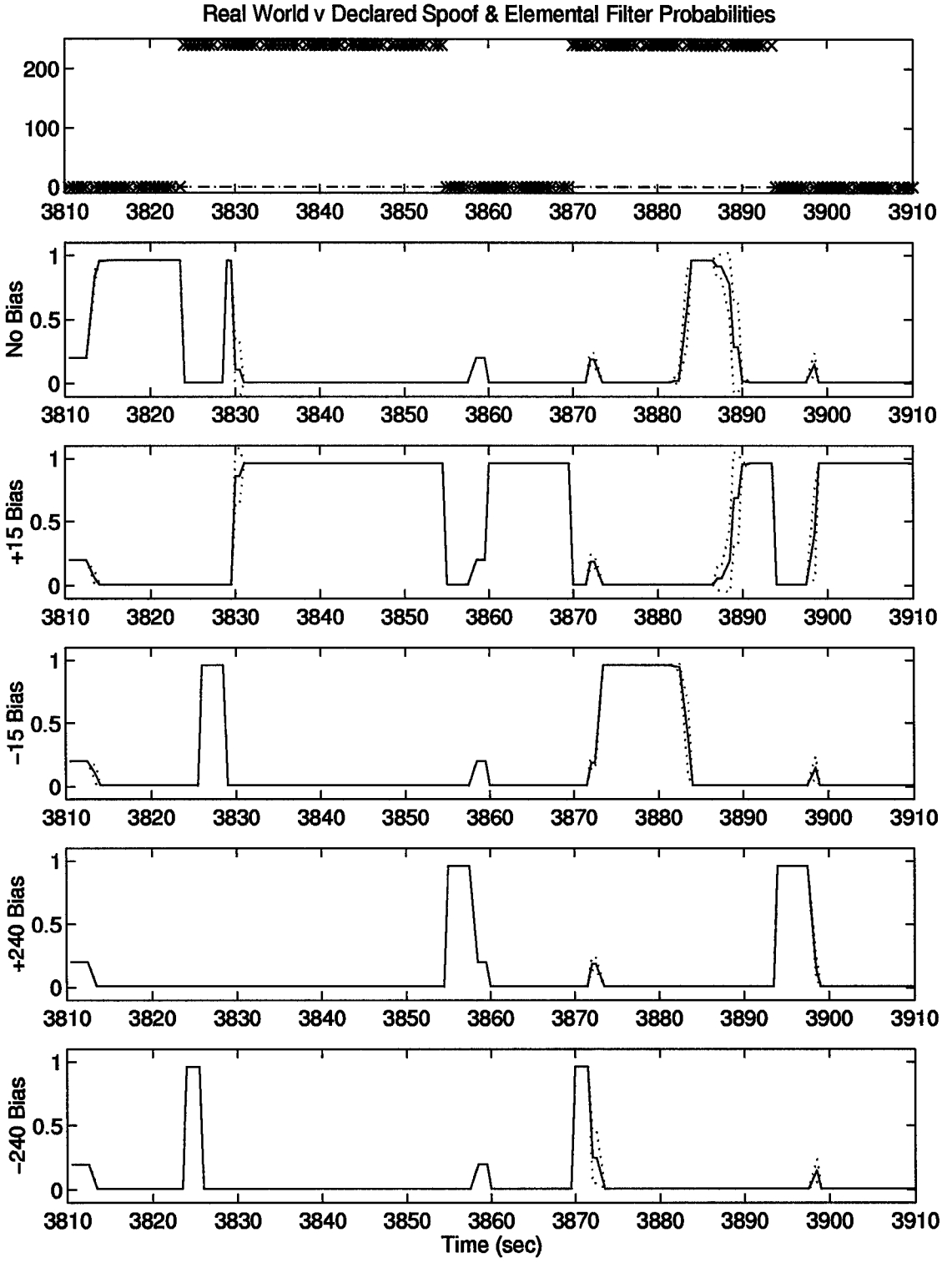

Figure 4.6 Spoofing FDI Performance and Elemental Filter Probabilities, Non-Moving Bank PRMMAE - Nav Case 1

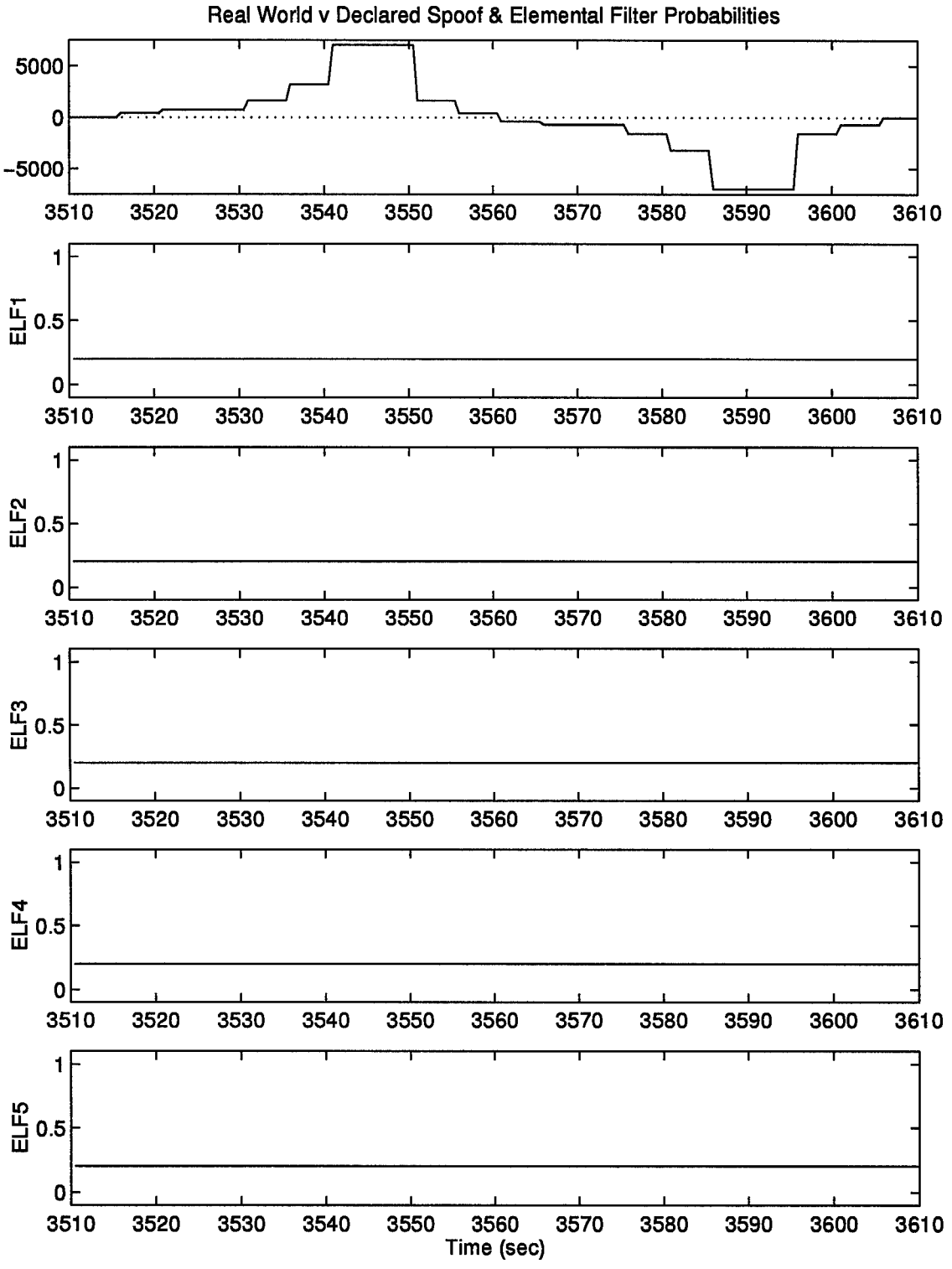

Figure 4.7 Spoofing FDI Performance and Elemental Filter Probabilities, Standard MMAE - Nav Case 1

to form $[\mathbf{r}_k^T \mathbf{A}_k^{-1} \mathbf{r}_k]$ for the MMAE's probability calculations; see Section 4.2.4.1). Figure 4.7 shows the detection performance of the corrected (see Section 4.2.4.1) MMAE algorithm. This performance clearly shows that correct MMAE is unable to detect measurement bias offsets. The total probability is equally divided among the five elemental filters because all of the filters have indistinguishable residuals. The "ELF1"-"ELF5" labels seen in Figure 4.7 were used early in this research to denote **EL**emental **F**ilters one through five, and correspond to bias offset assumptions of 0, +350, -350, +700, and -700 feet, respectively. State estimation performance plots are not included for Figures 4.6 and 4.7 because these cases are included only to motivate the development of the pseudo-residual MMAE (PRMMAE) algorithm.

4.2.4.1 Pseudo-Residual MMAE Theory. The incorrect MMAE algorithm produces good, potentially exploitable information for one sample period while the correct MMAE algorithm does not. The mathematical reasons for this unexpected result follow.

At all times

$$\mathbf{z}_{true} = \mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{v}_{true} + \mathbf{b}_{true} \tag{4.1}$$

Let us assume two filter models, (1) assuming no measurement bias, i.e., that $\mathbf{b}_{true} = \mathbf{0}$, and (2) assuming a positive measurement bias, i.e., that $\mathbf{b}_{true} = \mathbf{b}_1$, where $\mathbf{b}_1$ will, at least initially, be the bias actually simulated in the real world. These two filters will have, as their best prediction $\hat{\mathbf{z}}$ of the measurement before it arrives

$$\begin{aligned} (1) \quad \hat{\mathbf{z}}_1 &= \mathbf{H}\hat{\mathbf{x}}_1^- \\ (2) \quad \hat{\mathbf{z}}_2 &= \mathbf{H}\hat{\mathbf{x}}_2^- + \mathbf{b}_1 \end{aligned} \tag{4.2}$$

which will produce the following update equations used to generate $\hat{\mathbf{x}}^+$ at each sample period:

$$\begin{aligned} (1) \quad \hat{\mathbf{x}}_1^+ &= \hat{\mathbf{x}}_1^- + \mathbf{K}_1[\mathbf{z}_{true} - \hat{\mathbf{z}}_1] \\ (2) \quad \hat{\mathbf{x}}_2^+ &= \hat{\mathbf{x}}_2^- + \mathbf{K}_2[\mathbf{z}_{true} - \hat{\mathbf{z}}_2] \end{aligned} \tag{4.3}$$

or

$$\begin{aligned} (1) \quad \hat{\mathbf{x}}_1^+ &= \hat{\mathbf{x}}_1^- + \mathbf{K}_1[\mathbf{z}_{true} - \mathbf{H}\hat{\mathbf{x}}_1^-] \\ (2) \quad \hat{\mathbf{x}}_2^+ &= \hat{\mathbf{x}}_2^- + \mathbf{K}_2[\mathbf{z}_{true} - (\mathbf{H}\hat{\mathbf{x}}_2^- + \mathbf{b}_1)] \end{aligned} \tag{4.4}$$

After allowing both filters to run to steady state, each filter will modify its state estimates so that the residual is zero-mean, i.e. $\mathrm{E}[\mathbf{z}_{true} - \hat{\mathbf{z}}] = \mathbf{0}$. In fact, it was observed that the filter's primarily altered their *user clock bias state* to yield such zero-mean residuals. This has the following effect:

$$\begin{aligned} (1) \quad \mathrm{E}[\mathbf{H}\hat{\mathbf{x}}_1^-] &= \mathbf{z}_{true}(\mathbf{b}_{true} = \mathbf{0}) &&= \mathbf{H}_{true}\mathbf{x}_{true} \\ (2) \quad \mathrm{E}[\mathbf{H}\hat{\mathbf{x}}_2^-] &= \mathbf{z}_{true}(\mathbf{b}_{true} = \mathbf{0}) - \mathbf{b}_1 &&= \mathbf{H}_{true}\mathbf{x}_{true} - \mathbf{b}_1 \end{aligned} \tag{4.5}$$

At the onset of a spoof, $\mathbf{b}_{true}$ becomes non-zero, $\mathbf{b}_{true} = \mathbf{b}_1$ for example. At the next measurement update, the *true* residuals are (discounting noise)

$$\begin{aligned} (1) \quad [\mathbf{z}_{true} - \hat{\mathbf{z}}_1] &= [(\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1) - \mathbf{H}\hat{\mathbf{x}}_1^-] \\ (2) \quad [\mathbf{z}_{true} - \hat{\mathbf{z}}_2] &= [(\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1) - (\mathbf{H}\hat{\mathbf{x}}_2^- + \mathbf{b}_1)] \end{aligned} \tag{4.6}$$

and the *pseudo*-residuals, namely $[\mathbf{z}_{true} - \mathbf{H}\hat{\mathbf{x}}_k^-]$ for all k, are (discounting noise)

$$\begin{aligned} (1) \quad [\mathbf{z}_{true} - \mathbf{H}\hat{\mathbf{x}}_1^-] &= [(\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1) - \mathbf{H}\hat{\mathbf{x}}_1^-] \\ (2) \quad [\mathbf{z}_{true} - \mathbf{H}\hat{\mathbf{x}}_2^-] &= [(\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1) - \mathbf{H}\hat{\mathbf{x}}_2^-] \end{aligned} \tag{4.7}$$

The *true* residuals, using $\mathrm{E}[\mathbf{H}\hat{\mathbf{x}}_k^-]$ from Equation (4.5), become

$$\begin{aligned} (1) \quad [\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1 - \mathbf{H}_{true}\mathbf{x}_{true}] &= \mathbf{b}_1 \\ (2) \quad [\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1 - ((\mathbf{H}_{true}\mathbf{x}_{true} - \mathbf{b}_1) + \mathbf{b}_1)] &= \mathbf{b}_1 \end{aligned} \tag{4.8}$$

and the *pseudo*-residuals, using $\mathrm{E}[\mathbf{H}\hat{\mathbf{x}}_k^-]$ from Equation (4.5), become

$$\begin{aligned} (1) \quad [\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1 - \mathbf{H}_{true}\mathbf{x}_{true}] &= \mathbf{b}_1 \\ (2) \quad [\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1 - (\mathbf{H}_{true}\mathbf{x}_{true} - \mathbf{b}_1)] &= 2\mathbf{b}_1 \end{aligned} \tag{4.9}$$

In writing Equations (4.1) to (4.9), it is assumed that *true* residuals are used to update the elemental filters; what is subject to consideration here is whether the true residuals or the pseudo-residuals are more useful in forming $[\mathbf{r}_k^T \mathbf{A}_k^{-1} \mathbf{r}_k]$ for the MMAE's probability computations. As may be seen from Equation (4.8), the true residuals from each elemental filter do not reveal the real world measurement bias and are indistinguishable from one another (see Figure 4.6). It may be deduced from Equation (4.9) that the filter assuming the negative of the actual bias will show nearly zero-mean pseudo-residuals at the measurement update immediately following the spoof onset (recall Figure 4.6). Hence, good identification may be achieved by using the pseudo-residuals to form $[\mathbf{r}_k^T \mathbf{A}_k^{-1} \mathbf{r}_k]$. In the above discussion, had a filter assumed a bias of $-\mathbf{b}_1$, then in steady state its $\mathbf{H}\hat{\mathbf{x}}^-$ would have become $\mathbf{H}_{true}\mathbf{x}_{true} + \mathbf{b}_1$. The pseudo-residuals of such a filter would be roughly zero-mean at the measurement update following the spoof onset. The information provided by this zero-mean measurement pseudo-residual is used to isolate the actual bias. *The true residuals must be used (as in a single Kalman filter or regular MMAE) to update the elemental filters.*

4.2.4.2 Pseudo-Residual MMAE Performance. Examination of Figure 4.6 in the light of the previous section suggests that good, reliable information does exist with which to identify the onset of a spoof (isolation is also possible if an individual filter happens to assume exact value of bias offset that is applied), but it is only visible for a single measurement update. The moving-bank MMAE algorithm introduced here is useful in exploiting this information. The reader should note that, up until this point, we have limited our discussion to a fixed-bank MMAE.

To get around the quirk of only having the decision information available for one update cycle, one only has to accept the single sample period of information as an identification of the spoof, subsequently estimate the spoof, then go back and reprocess the last measurement assuming that the estimated spoof is present in the real world (*the entire filter bank is moved to the neighborhood of the estimated*

spoof). Filter bank movement is accomplished by subtracting $\mathbf{b}_{estimated}$ from the true measurements before they enter the MMAE algorithm, rather than adding $\mathbf{b}_{estimated}$ to each filter's $\hat{\mathbf{z}}_k$, since it is computationally more efficient. This process is repeated for that single update time until the new spoof bias value is completely identified and the bank is recentered. After this correction, each of the elemental filters in the bank steps forward into the next propagation cycle without knowing it ever experienced a spoof. This process can be observed in the p_k plots of simulations involving spoofing elemental filters. In the first, third, and fifth subplots of Figure 4.10 (to be seen and described in detail later in this section, associated with ramp rather than step spoofs, but called out here because it clearly shows the desired phenomenon), for example, it can be seen that when the real-world spoof bias changes by +15 feet/sec (the first spoofing ramp), the p_k associated with the -15 foot bias elemental filter spikes for one second. The bank is moved based on this spike, and in the next second the probability returns to the no-bias elemental filter. Throughout the simulation depicted in Figure 4.10 (and each spoofing simulation), it may be seen that the p_k value corresponding to the no-bias elemental filter displays such downward spikes each time a change in the real-world spoof bias value is detected. These downward spikes depict the information given by the *first* measurement update which caused the MMAE filter bank to be moved; in reality, due to the measurement reprocessing described above, the no-bias elemental filter's p_k value never moves from its near-unity value.

Figure 4.8 shows the detection and isolation performance of the PRMMAE algorithm in the moving-bank configuration. The reader should note several aspects of Figure 4.8 which are consistent with the description of the moving-bank PRMMAE algorithm's operation as described so far. It can be seen that the probability rests consistently (excepting the downward spikes mentioned above) on the no-bias elemental filter (fourth sub-plot), while the real-world and declared spoof values range over 2000 feet (first sub-plot). This is a clear indication that the no-fail filter is con-

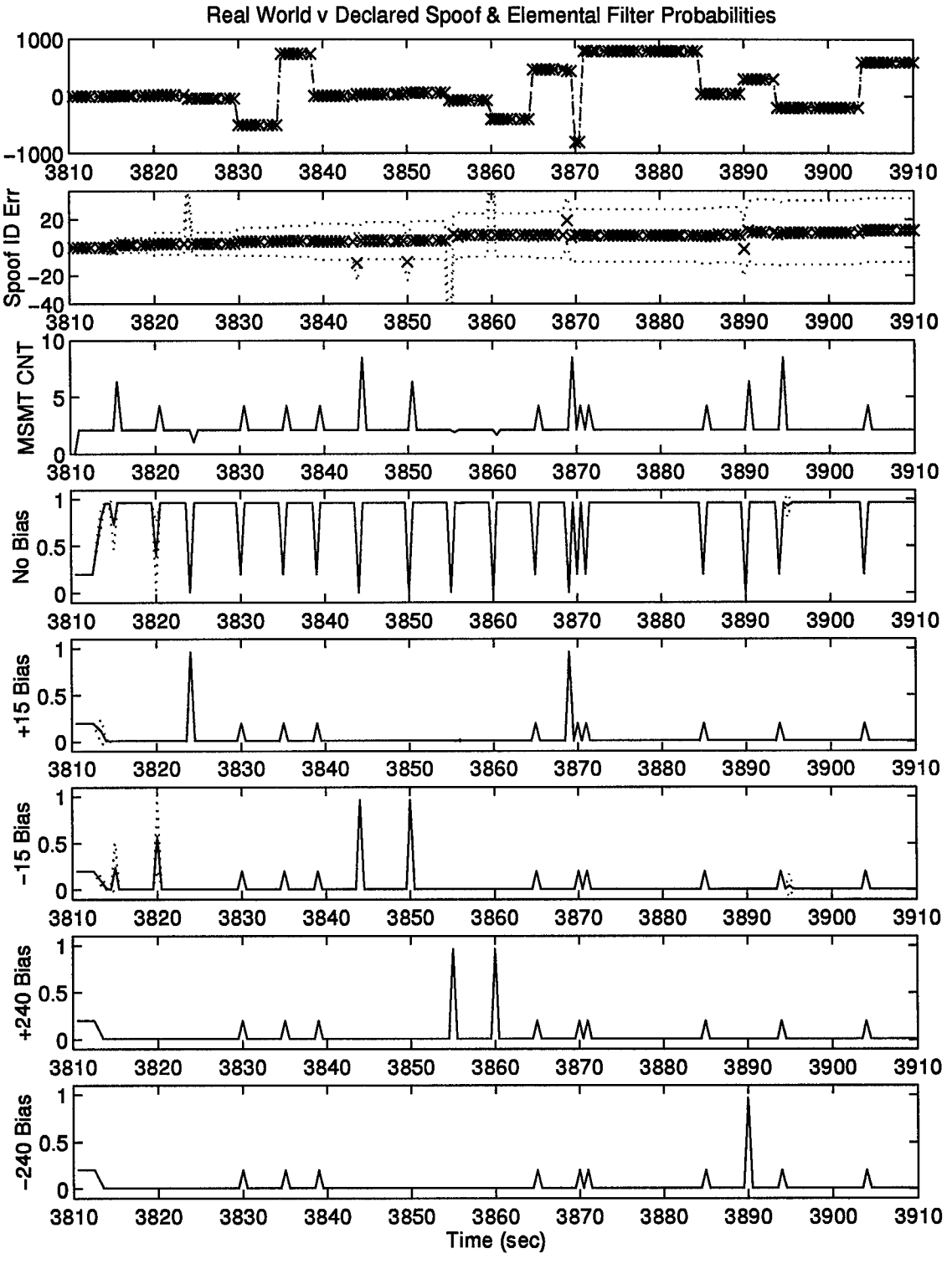

Figure 4.8 Spoofing (Steps) FDI Performance and Elemental Filter Probabilities, Moving Bank PRMMAE - Nav Case 1

sistently recentered on the real-world spoof value. Each time the real-world spoof value jumps by a value modelled in one of the elemental filters, the probability spikes on the filter assuming the *negative* of the spoof jump value (sub-plots five-eight). The third sub-plot (measurement count) shows the number of measurement updates that were taken to identify the new spoof offset value fully at each sample period before the following propagation cycle was entered.

Figure 4.8 and the corresponding state estimation error plots of Figure 4.9 indicate that spoofing jumps are identified accurately regardless of whether their value is exactly modelled in the bank of elemental filters. The PRMMAE algorithm itself is unable to identify spoofing values greatly different from those modelled in its bank. The difficulty here is that the MMAE residual information term, $\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r}$, becomes very large when the spoof offset is numerically greater than 150 (feet) displaced from the bias assumption of a modelled filter. Because $-1/2$ times this large term appears in an exponential, the MMAE conditional hypothesis calculation value goes to zero and the MMAE algorithm cannot make a decision about which direction to move the bank in the possible failure space. The first proposed solution to this problem was to attempt to move the existing bank throughout the possible "spoof offset" parameter space, doing measurement updates at each assumed bias offset, until the new spoof was encountered. As it turns out, this is not necessary. Even though the MMAE calculations using the $\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r}$ term become useless, the individual measurement residuals associated with the DGPS measurements in each elemental filter allow simple isolation of the spoof magnitude. These residuals are zero-mean up until the addition of the spoof. When the spoof occurs, it shows up directly on the residuals. Estimation is a matter of simply reading the number. It can be seen in Figure 4.8 (sub-plots 1, 5-8) that large changes in the real-world spoof value correspond to probability values of $0.2 = 1/(\#filters)$ for each filter. When all filters in the bank have equally bad residuals, the probability is equally divided

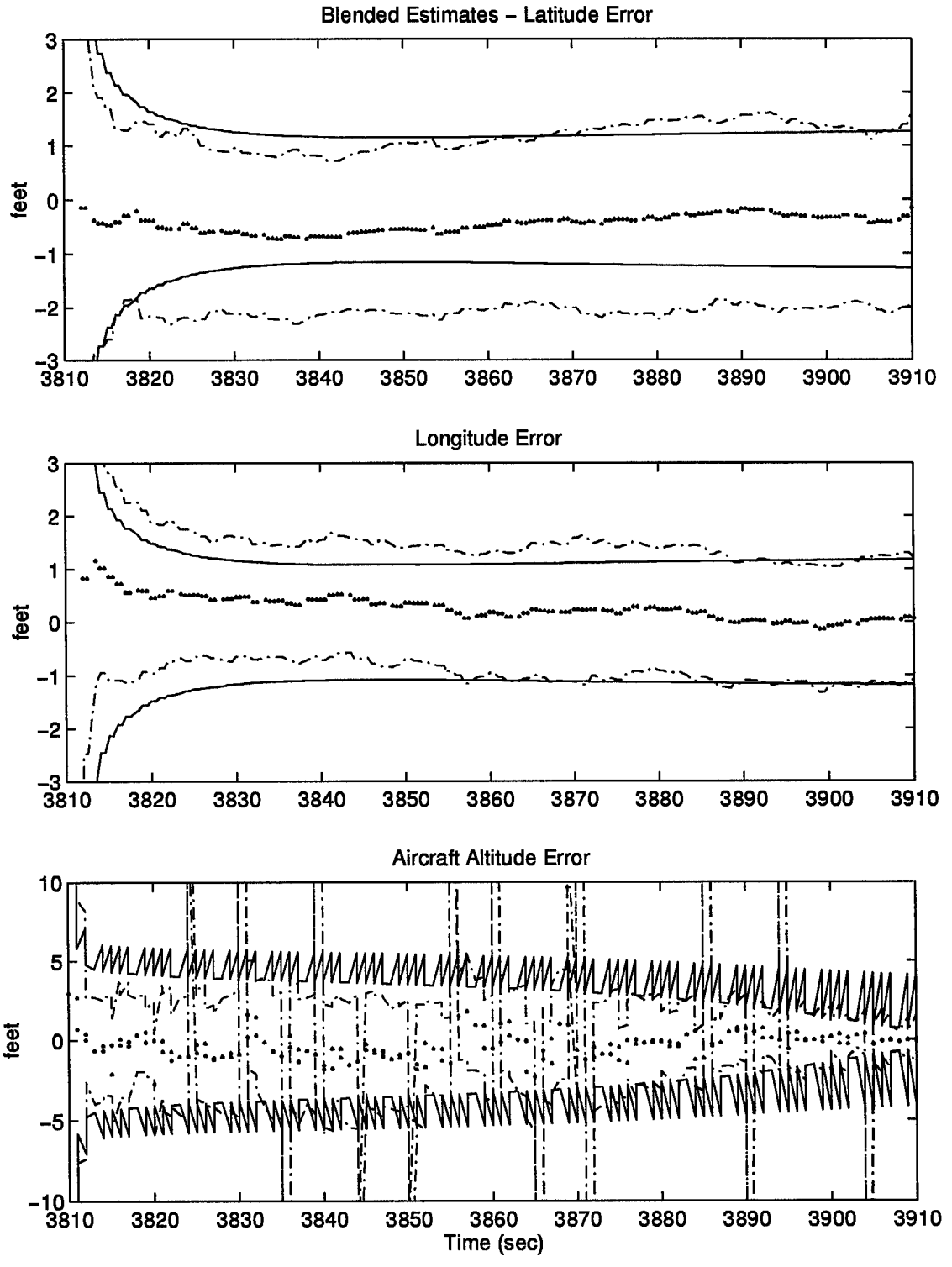

Figure 4.9 Spoofing (Steps) Navigation Performance, Moving-Bank PRMMAE - Nav Case 1

between the filters and the individual residual terms are examined for the spoof offset as described above.

Identification of spoofing as described in the previous paragraph is accomplished equally well with only two symmetrical bias-assuming filters displaced about a filter assuming no bias offset, and not the four symmetrical filters that were used earlier in this research. The three-filter spoofing bank is used to identify the spoof onset and to fine-tune the spoof estimation once the first guess is made. Based on the experience gained during this research, and given the methods finally used to detect measurement bias failures, it may be *possible* to identify the spoof with only one filter. In fact, Vasquez' [41, 42] GLR methods of spoof identification, on first inspection, appear to be mathematically quite similar to the methods used in this research but using a single Kalman filter. Bank movement and fine-tuning the spoof isolation are only *easily* performed via MMAE blending using the no-bias and two symmetrical-bias filters.

Figure 4.10 shows the FDI performance of the moving-bank PRMMAE algorithm in the face of ramped (intelligent) spoofing offsets. The first ten real-world spoofing ramp segments (sub-plot one) have slopes of 30, 24, 20, 16, 12, 10, 8, 6, 4, and 2 feet per second, respectively. It can be seen that no significant identification error occurs until the 8 ft/sec spoofing segment (note the dotted mean$\pm\sigma$ trajectories on the first subplot). The smaller spoofing ramps do cause considerable confusion to the algorithm, although the state estimation still does not go too far awry, as seen in Figure 4.11. It can be concluded that, given the navigation component configuration of case one, spoofing steps and spoofing ramps as small as 10 ft/sec can be identified with no significant error. The FDI performance of cases 2-4 is discussed in Section 4.2.6. The blended navigation solution provided by the PRMMAE algorithm is quite good and is shown in Figures 4.9 and 4.11. Comparison of these results with those of a single non-adaptive Kalman filter (refer to Figure 4.4) shows the dramatic

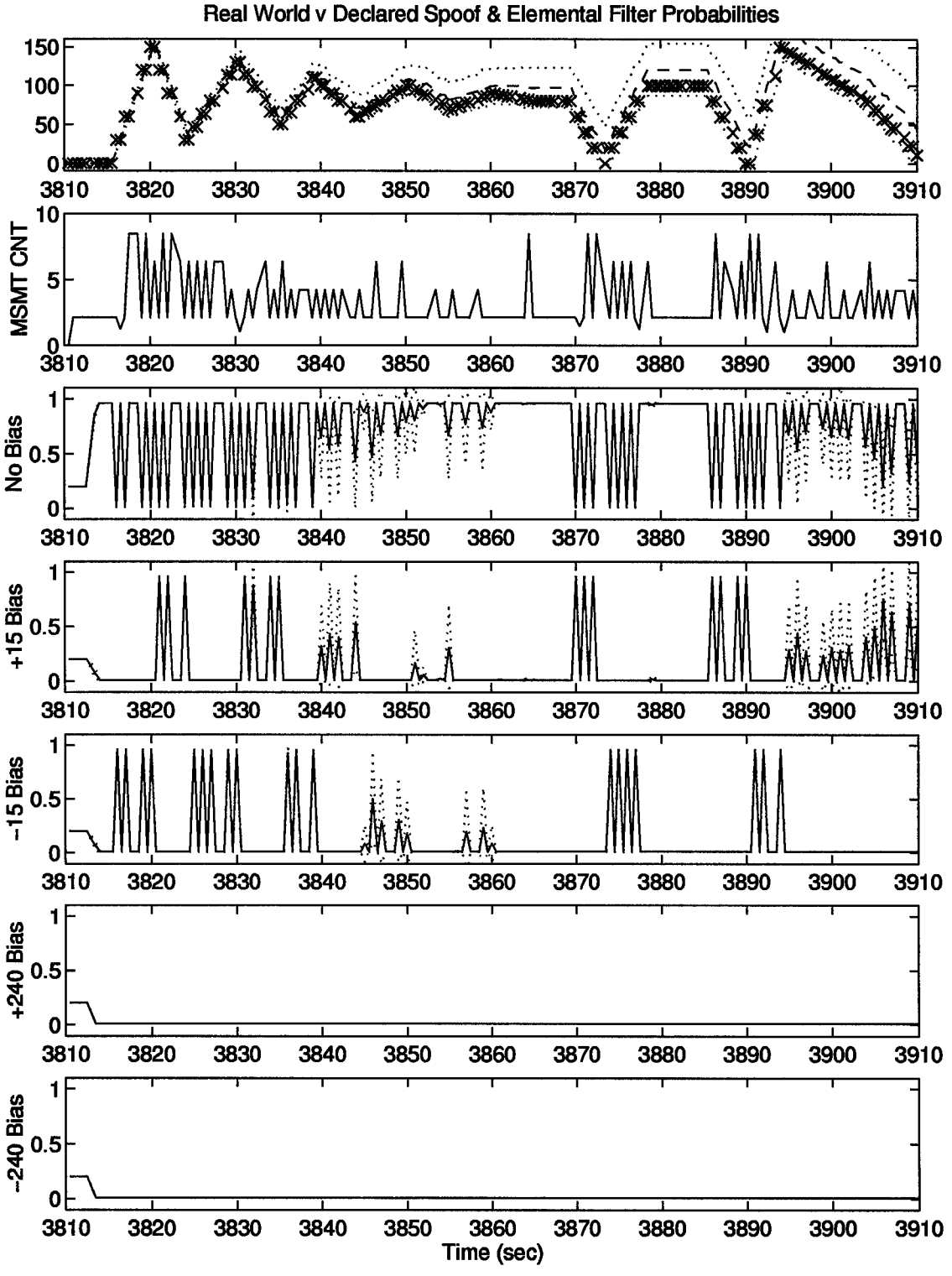

Figure 4.10 Spoofing (Ramps) FDI Performance and Elemental Filter Probabilities, Moving-Bank PRMMAE - Nav Case 1

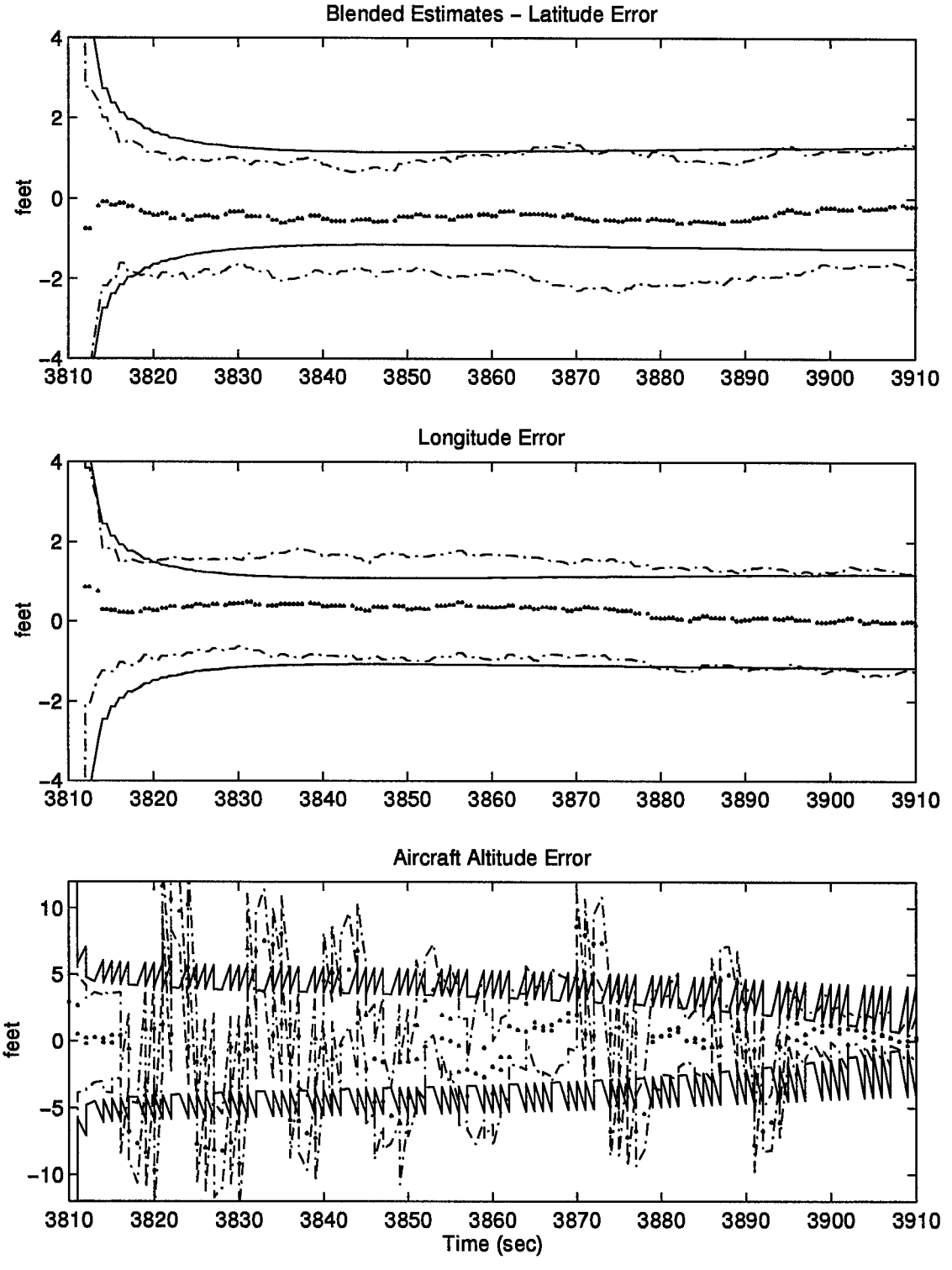

Figure 4.11 Spoofing (Ramps) Navigation Performance, Moving-Bank PRMMAE - Nav Case 1

improvement in performance given by the moving-bank PRMMAE algorithm used in this research.

An important consideration is inherent throughout the above discussion. The elemental filters of the MMAE must be allowed to initialize with no measurement bias present in the real world. What is actually detected and tracked is the offset of the measurement bias from what was present during initialization. This assumption is reasonable because GPS initialization will presumably be done before take-off from a friendly air base. Any attempted spoofing there would be detected by surveyed receivers at the station, and shortly removed.

4.2.5 Detection of Both Spoofing and Interference in a Single MMAE. Figure 4.12 shows the FDI performance of a filter bank containing some elemental filters assuming interference and some assuming spoofing offsets. Figure 4.13 shows the corresponding state estimation error. The five modelled filters assume (1) zero bias, no interference (1x original measurement noise variance), (2) zero bias, interference represented by 200x original $\mathbf{R}$, (3) zero bias, interference modelled by 2000x original $\mathbf{R}$, (4) +15 feet bias, no interference, and (5) -15 feet bias, no interference. The inclusion of the spoofing filters is not expected to, and in fact does not, detract from the MMAE's performance against interference because increased random measurement noise looks nothing like the uniform measurement bias shown by a spoof. Detection of interference is accomplished with no degradation of performance versus the case when no spoofing filters were present. However, using only a *single* sample period test, there is no possible way for the algorithm to distinguish between the onset of a constant measurement bias and a large measurement noise value superimposed on the unbiased measurement. In an MMAE filter bank containing both interference and spoof elemental filters, the spoof elemental filters do not accept probability, at each sample time, for instances of random interference or small spoof steps because the interference elemental filters are pre-disposed to declare failures. The increased measurement noise $\mathbf{R}$ assumption of the interference elemental filters causes their

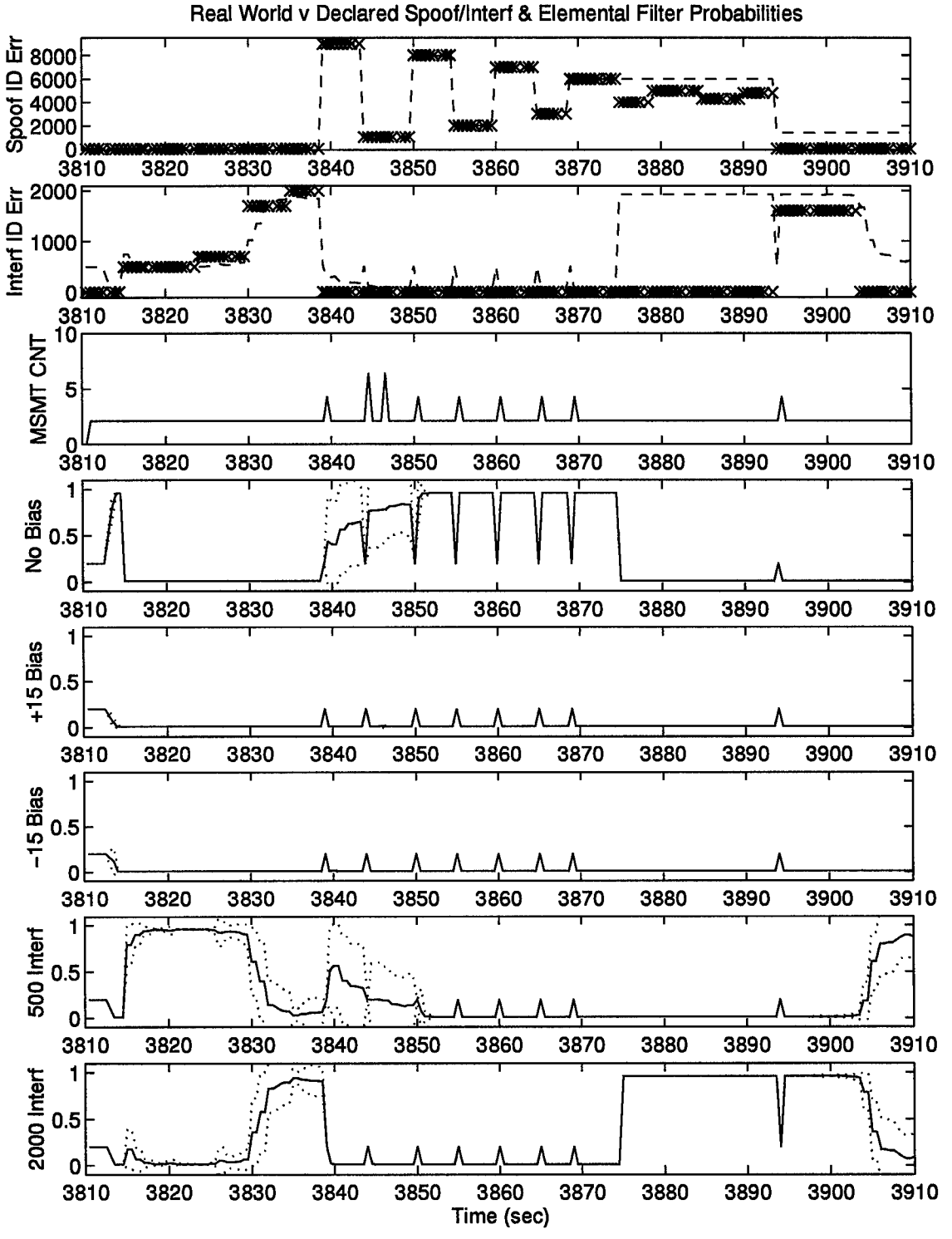

Figure 4.12 Spoofing & Jamming FDI Performance and Elemental Filter Probabilities, Moving-Bank PRMMAE - Nav Case 1

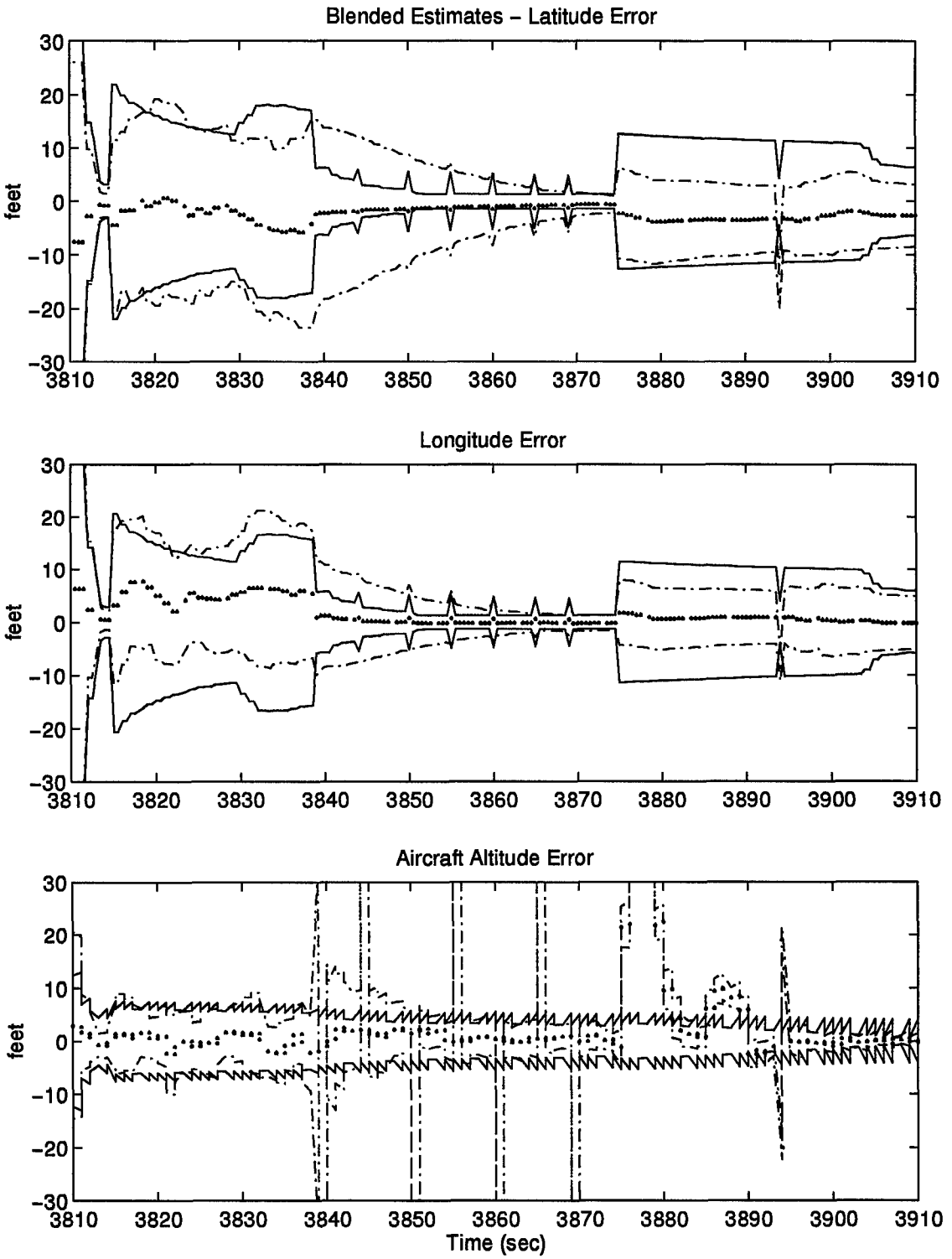

Figure 4.13 Spoofing & Jamming Navigation Performance, Moving-Bank PRMMAE - Nav Case 1

$\mathbf{A}$ matrices to be much larger than the $\mathbf{A}$ matrices associated with the spoofing elemental filters ($\mathbf{A} = \mathbf{HPH}^T + \mathbf{R}$). As a result, the elemental filter probability calculation exponential $\{-\frac{1}{2}\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r}\}$ is a negative scalar number of smaller magnitude for interference filters, because of their larger $\mathbf{A}$ matrices, than for spoofing filters assuming correct bias offsets, even when the spoofing filters have somewhat smaller residuals $\mathbf{r}$. The interference elemental filters have a tendency to declare failures in this case because they are (incorrectly) more heavily weighted than the spoofing elemental filters.

As might be supposed, identification in this case (spoof and interference elemental filters) is possible if the measurement bias added is of greater magnitude than is consistent with a modelled noise variance increase. The magnitudes of the first nine spoofing steps (sub-plot one) are 9000, 8000, 7000, 6000, 5000, 4000, 3000, 2000, and 1000 feet, respectively (offsets are bigger for this test because smaller spoof offsets, as in the earlier simulations, were misidentified as noise). Spoofing identification is done correctly until the 2000 foot spoofing change. Generally speaking, interference and spoofing cannot be distinguished by combining, into a single MMAE algorithm, the methods used in this research to identify interference and spoofing separately.

Future researchers may find great success identifying both interference and spoofing using only filters with spoofing assumptions and extending the detection test period over two or more sample periods. If the residual deviations and estimated spoof value display a constant bias, then spoofing may be identified. If residual deviations and estimated spoof value instead display substantial sample-to-sample changes, then increased measurement noise variance may be assumed *and* estimated (over the course of several sample periods used to build an estimate of the noise strength). This conjecture of spoofing bank behavior when subjected to interference failures was not verified in this research due to a shortcoming of the simulation tool MMSOFE. MMSOFE assigns a new random noise realization each time a single

measurement is reprocessed, making it impossible for the spoof filters to recenter on the original noise-corrupted measurement. Note that this noise realization concern does imply that MMSOFE was also *incorrectly* assigning a new measurement noise realization each time the PRMMAE algorithm recentered the filter bank against a real-world spoof; recentering in this case was done correctly however because the magnitude of the spoof bias **b** dominated relatively small magnitudes of the inconsistent measurement noise realizations.

One can envision a hierarchical MMAE structure that initially looks only for spoofs (using a moving-bank PRMMAE algorithm structure), but if the estimated spoof exhibits large sample-to-sample changes, the elemental filters in the MMAE could be redefined to look for interference instead (via a non-moving-bank standard MMAE algorithm structure). After the noise variance is estimated, the MMAE elemental filters can be returned to look for spoofs only (and the algorithm returned to a moving-bank PRMMAE form), but now with each elemental filter being tuned for the correctly estimated measurement noise variance. The greatest difficulty in this method will be the (possibly) required storage of the state estimation during the several sample periods required to isolate the nature of the failure. Once the failure type and magnitude is identified, the MMAE state estimates might be reprocessed over that period of time. This storage and reprocessing may or may not be necessary. That will need to be determined via trial and error to see whether good estimation is maintained with the simpler method of performing sample-to-sample estimation and not reprocessing. For example, it may be acceptable to use the state estimates produced by an MMAE looking only for spoof offsets (with its moving-bank structure) even if the decision is made, over a few sample periods, that interference more accurately models the current real-world failure. Once the failure is isolated, it may be required to reprocess all of the propagation and update cycles that were used to isolate the failure, using the newly declared bank definition. Alternatively, a *parallel* set of MMAE's, one searching for spoofs and the other for interference, could be

employed; again the filter seeking only spoofs would be used as the primary algorithm, but if the decision is made to declare an interference rather than a spoofing failure, the alternate MMAE might be usable *without* delay for any reprocessing of the propagation and update cycles.

The use of a jamming/spoofing sequence (described shortly) presents a very difficult challenge for any detection algorithm. A hostile enemy employing this method would heavily jam the area, causing the loss of the GPS satellite signals. The enemy would next introduce spoofed GPS-like signals, then remove the jamming so that the GPS receiver re-acquires the spoofing signals rather than the actual GPS signals. The MMAE FDI algorithm (or any other algorithm) does not maintain good state estimation during the jamming portion of this failure. When the jamming is removed, the filter's state estimation is so far off that it has no choice but to acquire whatever GPS or GPS-like signal is present in the real world. MMAE fails at detecting the spoofing signal following the jamming phase. The best that can be hoped for against such a failure is that, following the removal of jamming, spoofing values greater than the possible drift of the onboard INS will be detected. Although the MMAE FDI would have difficulty detecting the *initial* offset due to spoof once the jamming were removed, it *could* compensate for step or ramp spoof offsets thereafter, relative to that initial offset. This failure sequence was not tested during this research. The inclusion of an elemental filter only modelling INS and altimeter states, i.e., not including DGPS states rather than assuming zero **H** matrix entries corresponding to the DGPS measurements, may give useful information in the face of this failure sequence. Using this filter, at re-acquisition, any spoofing signals with an offset larger than the possible INS drift, would be rejected and the spoof identified.

4.2.6 Performance Comparisons: Navigation Cases 2-4. This section shows the failure detection and isolation performance available using a navigation component suite of lower quality (and less cost) than that used earlier in this chap-

ter to demonstrate the performance of the MMAE and PRMMAE algorithms (see Table 4.1). Unlike the plots presented previously in this chapter, which are grouped to show the FDI and corresponding navigation performance or a given failure case, the plots of this section will be organized to emphasize the changes, first, of FDI performance, and second, of navigation performance, with the range of navigation suites given in Table 4.1. Case 2 depicts the effect of removing pseudolites, case 3 the further impact of removing the radar altimeter, and case 4 the compounded impact of using a lower-precision INS. The FDI performance of each navigation case against interference and spoofing step failure types are examined.

Figures 4.14 through 4.17 show the interference FDI performance of navigation component cases one through four. Note that the speed and accuracy of detection and isolation of interference are nearly identical for each of the four cases.

Figures 4.18 through 4.21 show the spoofing FDI performance of navigation component cases one through four. It can be seen in these four plots that spoofing *detection* occurs in one second independent of the navigation suite used. The degree of misidentification of the spoofing magnitude shown in the second subplot of Figures 4.18 through 4.21 suggests that estimation of the magnitude of spoofing failures is also independent of the navigation components used.

Figures 4.22 through 4.25 show the navigation performance of navigation cases one through four in the presence of interference failures. Examination of these figures reveals an expected degradation of the navigation performance from navigation case one to two, two to three, and three to four. Comparison of Figure 4.23 (navigation case two) with Figure 4.22 (navigation case one) shows the effect on navigation performance of removing the ground-based pseudolite from the navigation suite. Comparison of Figure 4.24 (navigation case three) with Figure 4.23 (navigation case two) shows the effect on navigation performance of removing the radar altimeter. It can be seen that this change has a fairly substantial impact on the precision of the aircraft altitude estimates, as anticipated. Comparison of Figure 4.25 (navigation

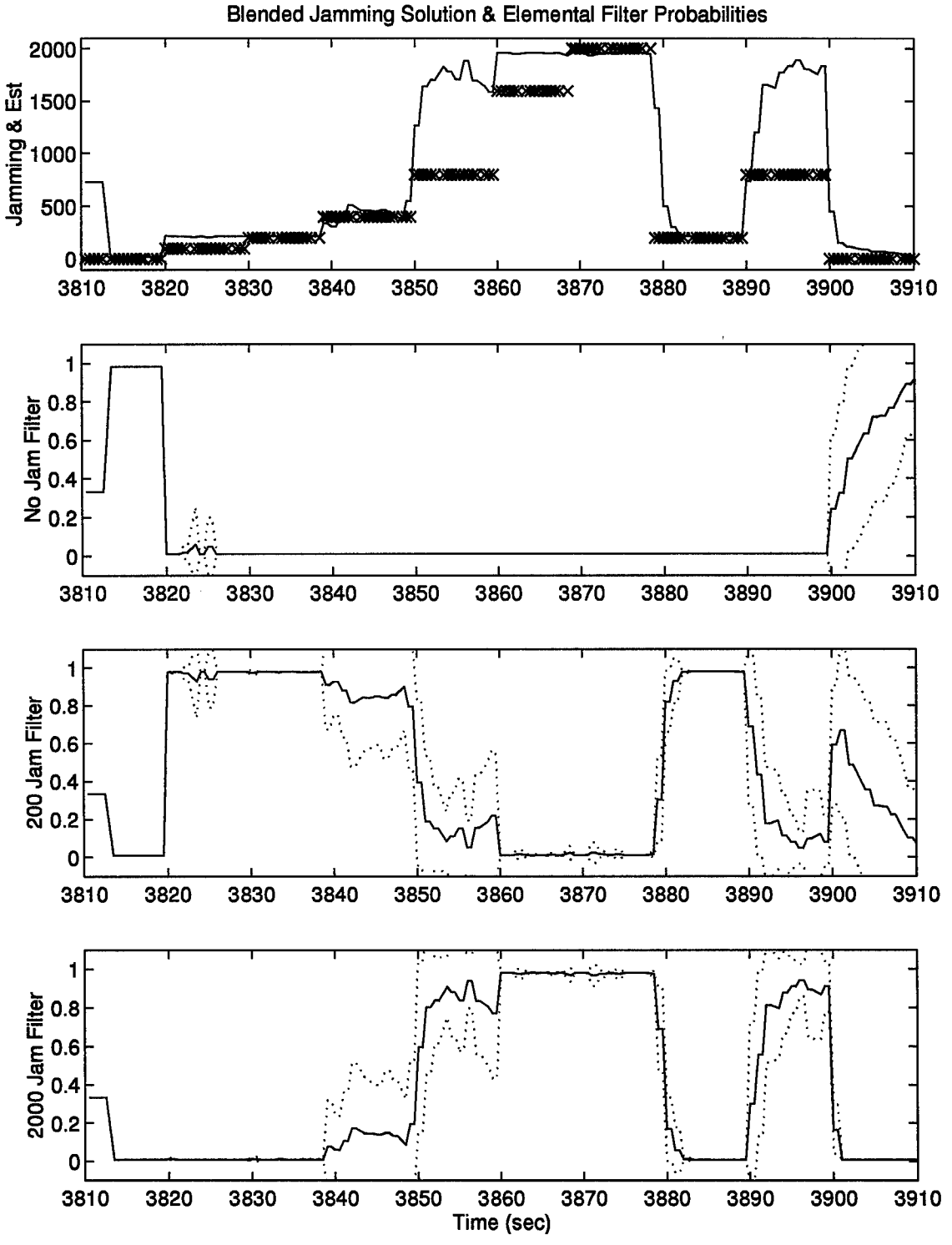

Figure 4.14 FDI Performance, Interference Failures - Nav Case 1

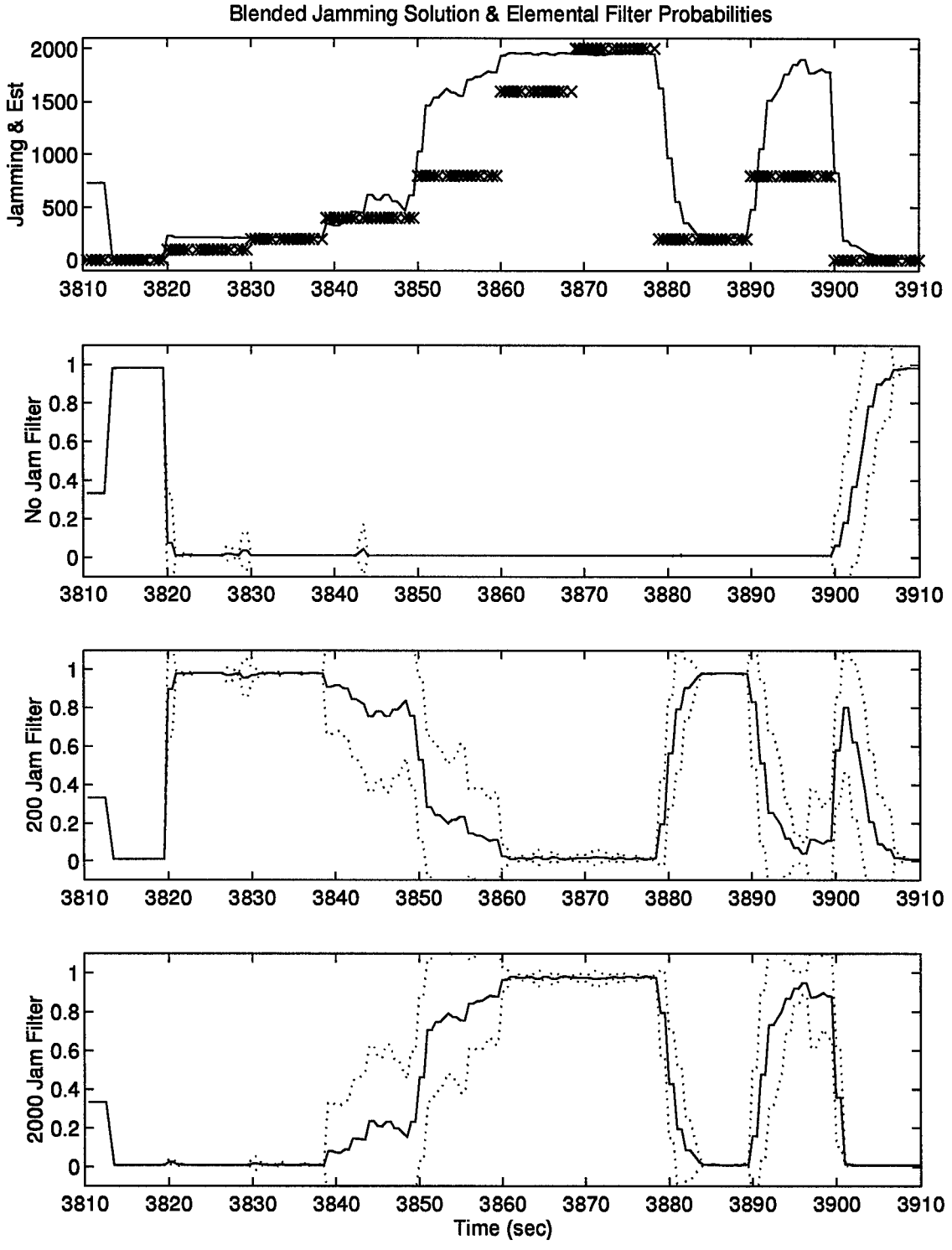

Figure 4.15 FDI Performance, Interference Failures - Nav Case 2

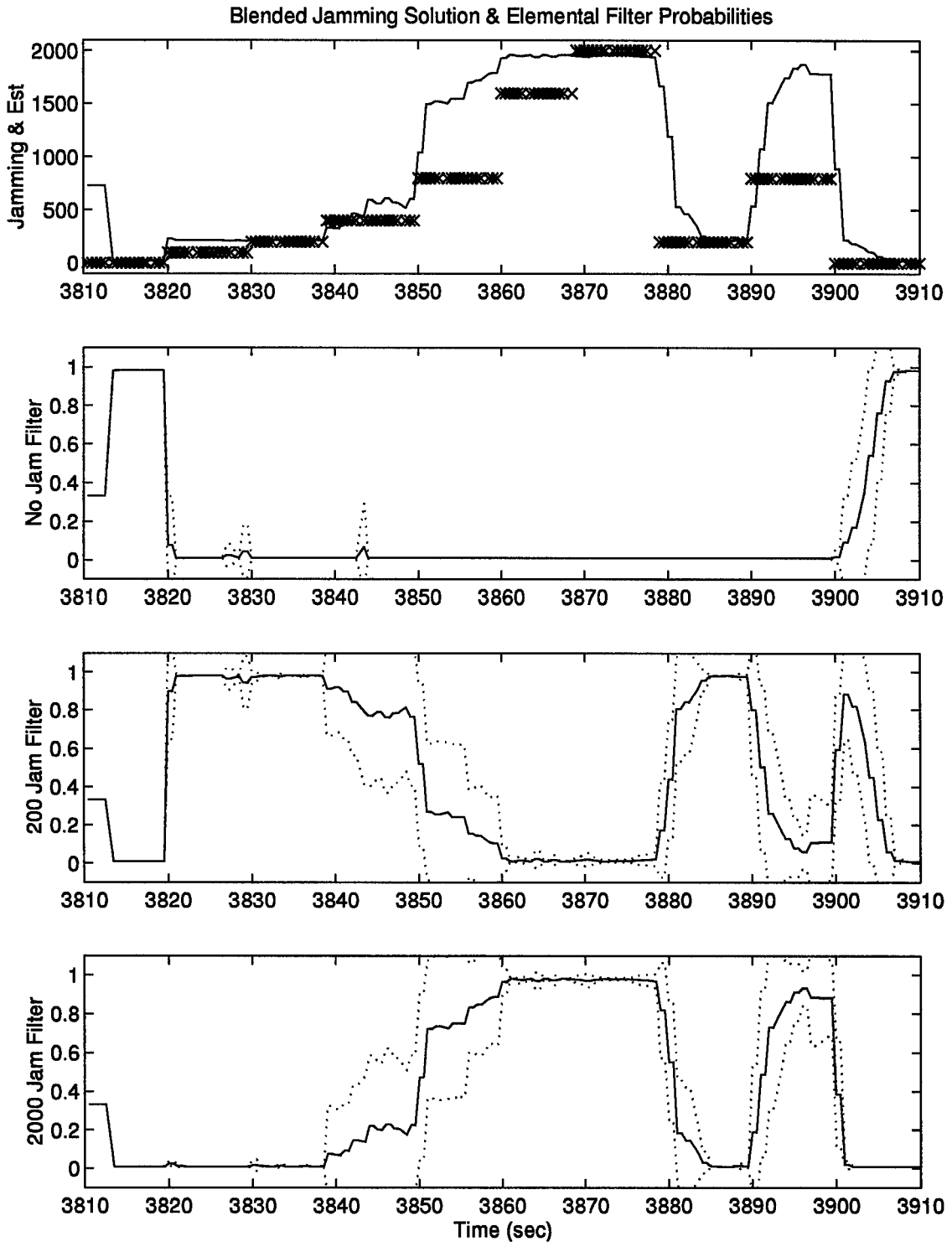

Figure 4.16 FDI Performance, Interference Failures - Nav Case 3

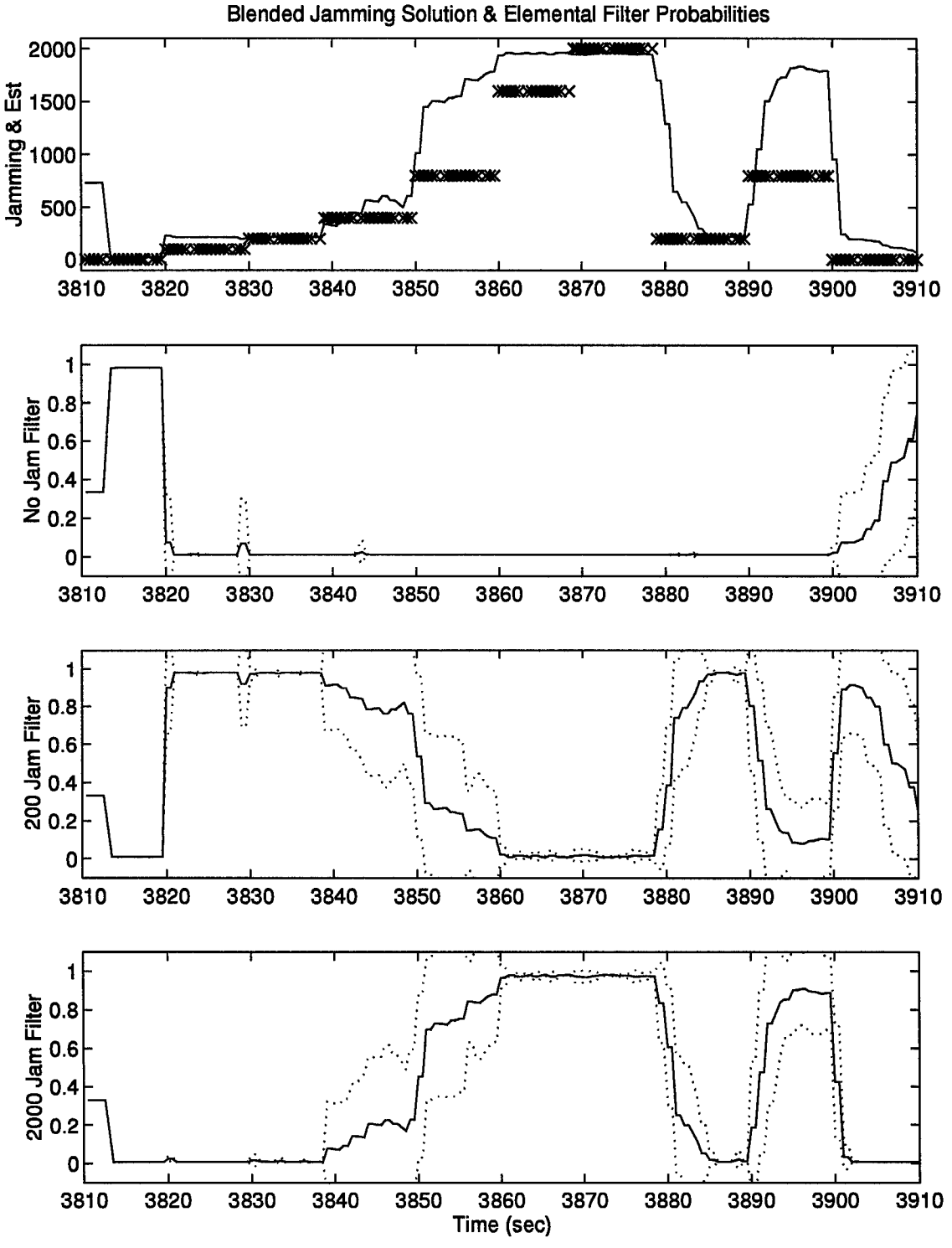

Figure 4.17 FDI Performance, Interference Failures - Nav Case 4

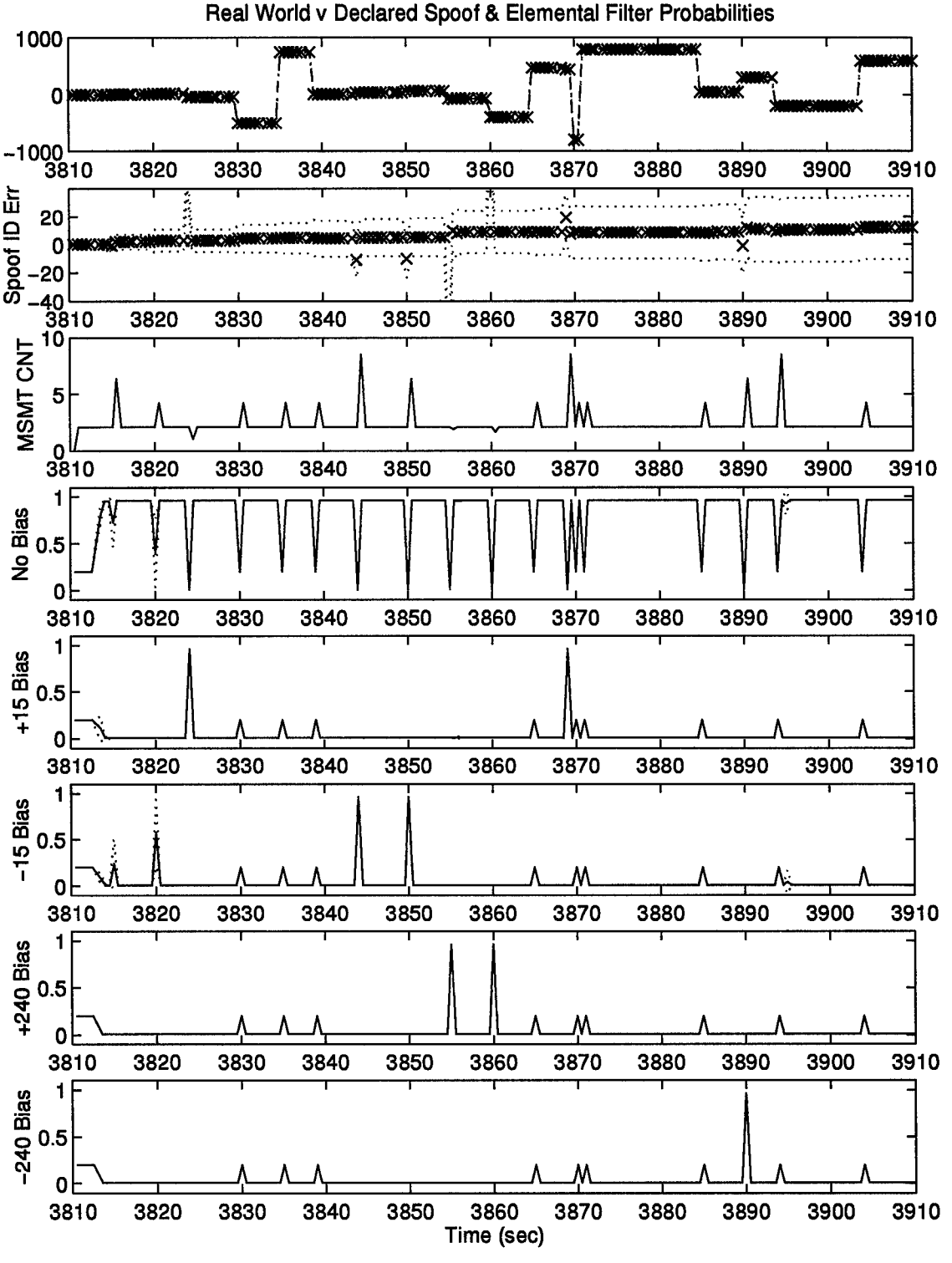

Figure 4.18 FDI Performance, Spoofing Step Failures - Nav Case 1

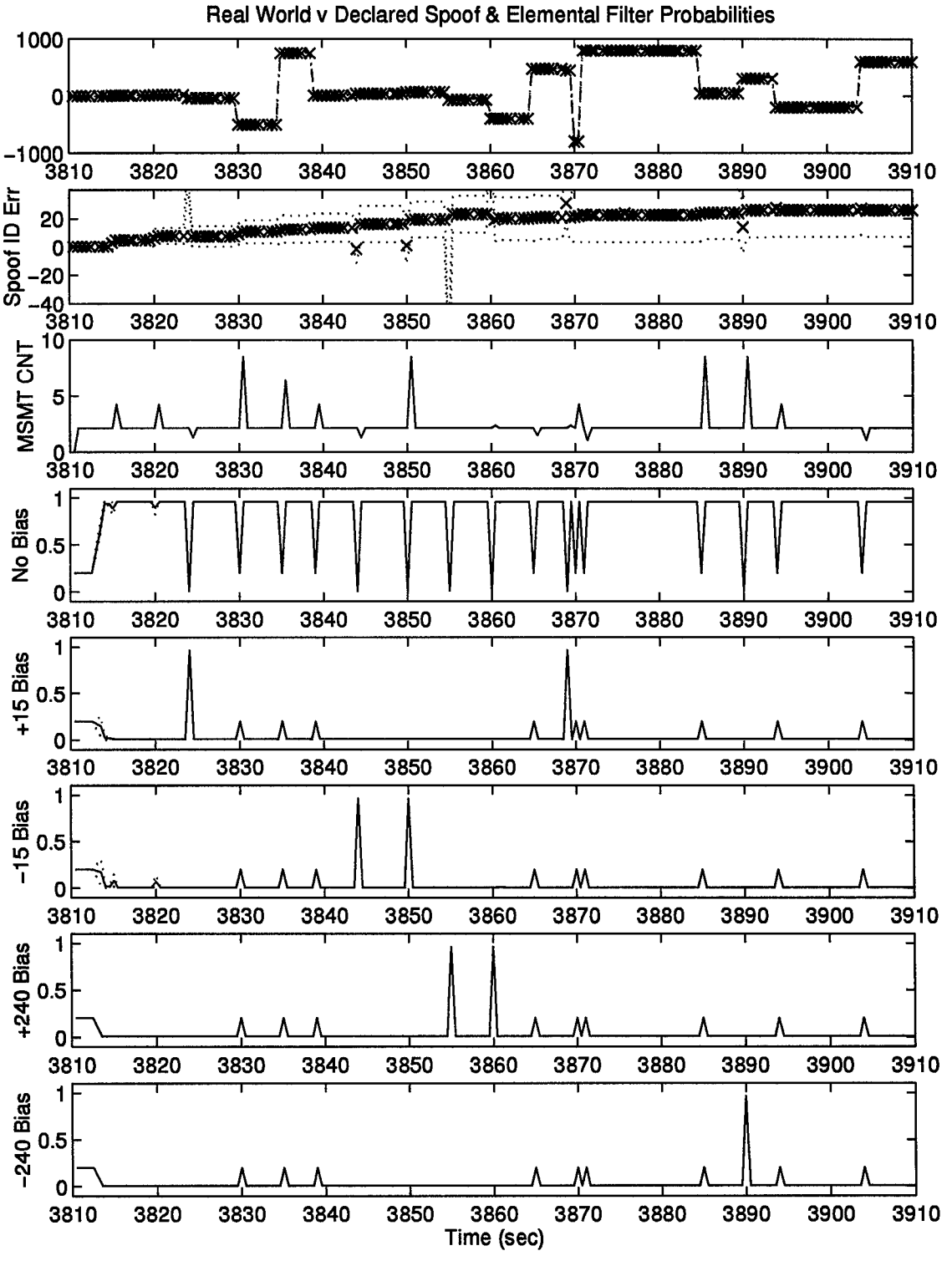

Figure 4.19 FDI Performance, Spoofing Step Failures - Nav Case 2

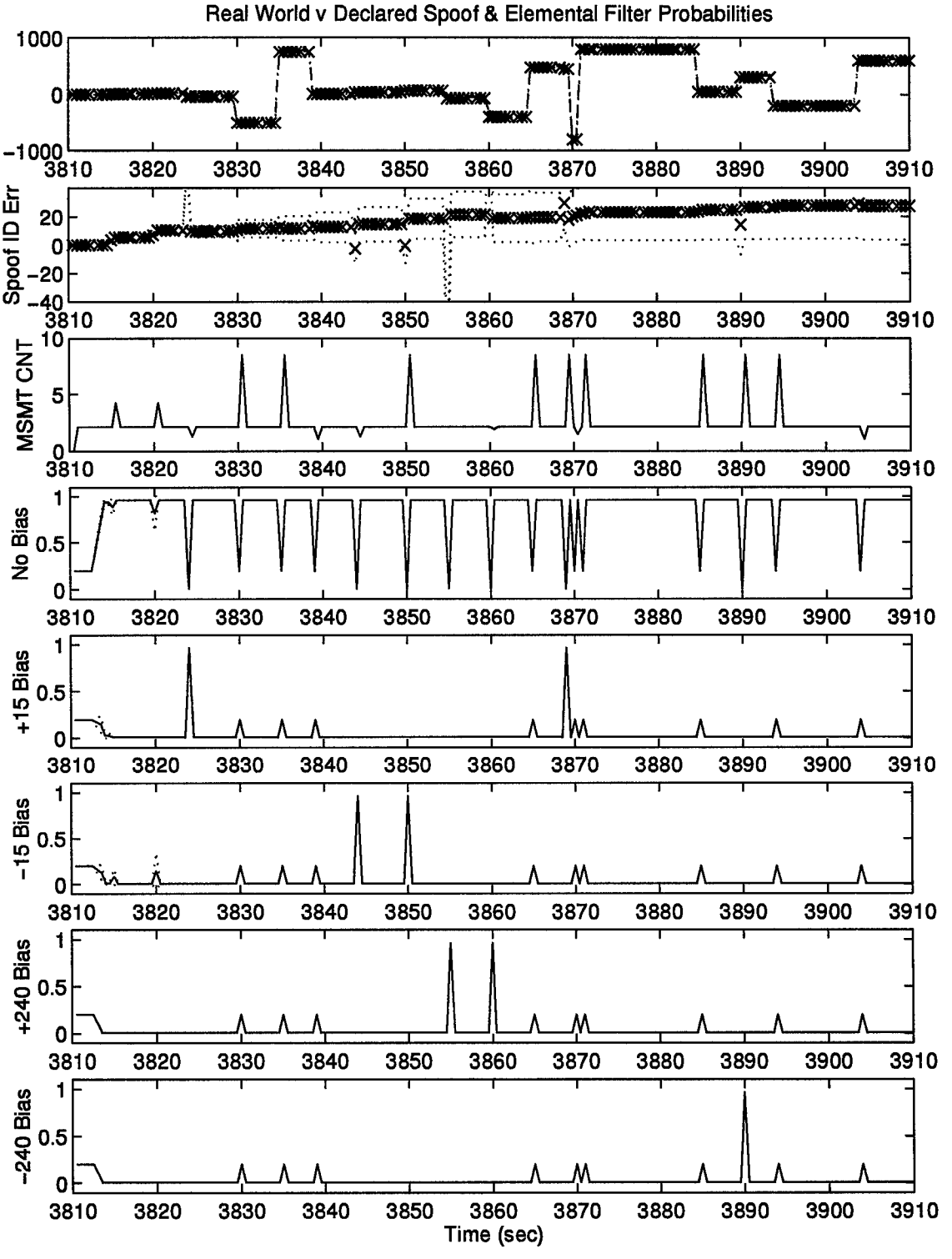

Figure 4.20 FDI Performance, Spoofing Step Failures - Nav Case 3

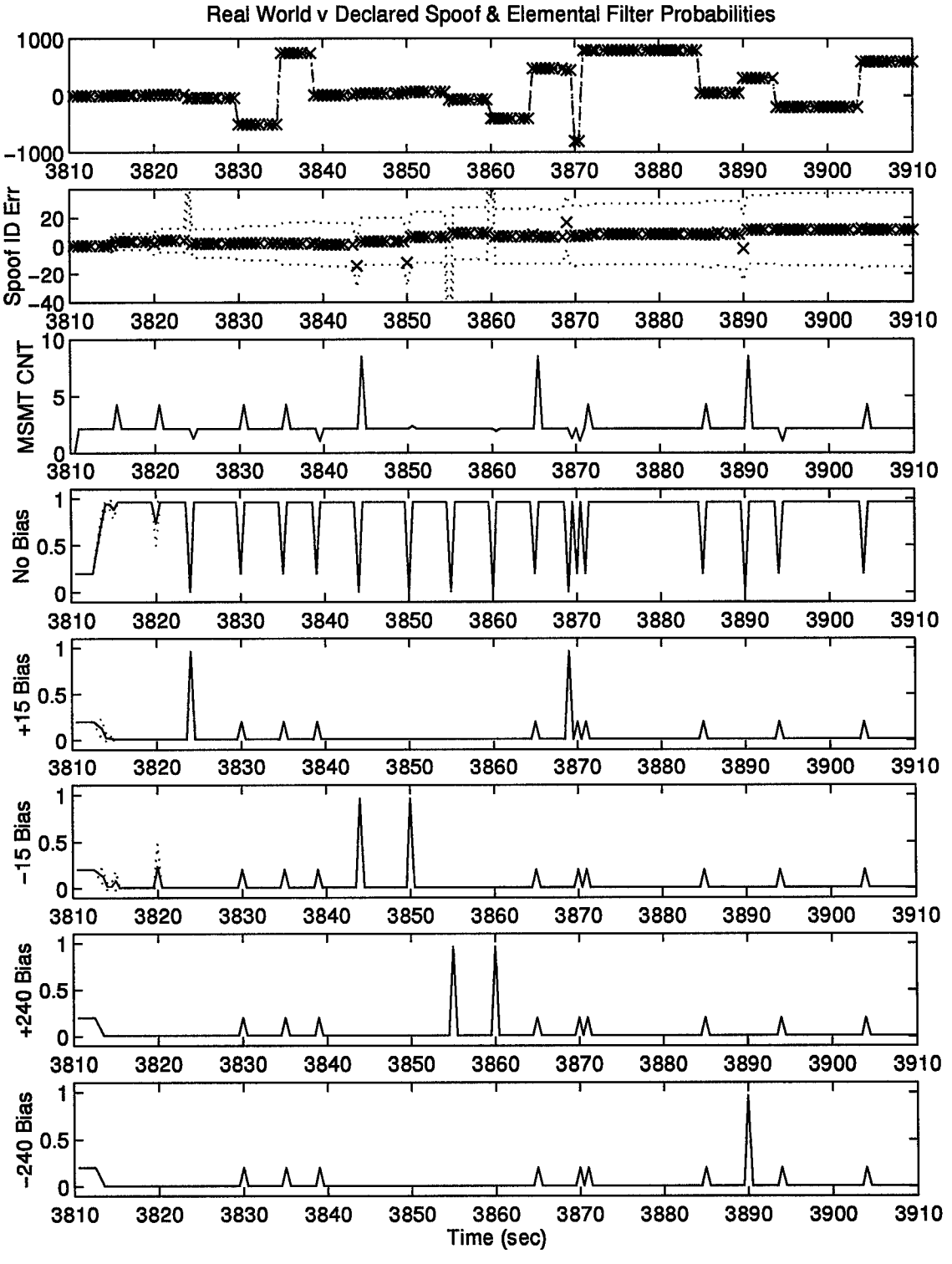

Figure 4.21 FDI Performance, Spoofing Step Failures - Nav Case 4

case four) with Figure 4.24 (navigation case three) shows the effect on navigation performance of using a navigation suite with a much less precise INS. Because of the consistently good FDI performances for these four cases, the navigation performance for each case is essentially that of a single extended Kalman filter correctly designed and tuned for the navigation system components *and artificially informed* of the actual interference noise variance.

Figures 4.26 through 4.29 show the navigation performance of navigation cases one through four when subjected to spoofing step failures. Similar observations may be made about the relative navigation precision offered by the different navigation suites.

4.2.7 Comparison to GLR/Chi-Square FDI. The FDI performance of the MMAE techniques used in this work compare very favorably with the GLR/chi-square techniques applied to a similar problem. Due to time constraints, direct comparisons of MMAE and GLR/chi-square using the detection problems of this work were not made; however, several conclusions may be drawn based on the similar FDI study performed by Vasquez [41, 42]. Vasquez' study was similar (dis-similar) in the following ways: Vasquez used

1. essentially the same INS as that used for navigation component case 1 of this study
2. GPS rather than DGPS
3. no radar altimeter and no pseudolites
4. a six-transponder range range-rate velocity aiding system

The GLR/chi-square scheme used by Vasquez [41, 42] on a GPS-aided inertial navigation system is effective at detecting interference failures, but this identification comes after a (sometimes large) delay. In Vasquez' [41, 42] work, identification of the onset of interference required a delay of 2 seconds. The MMAE algorithm used

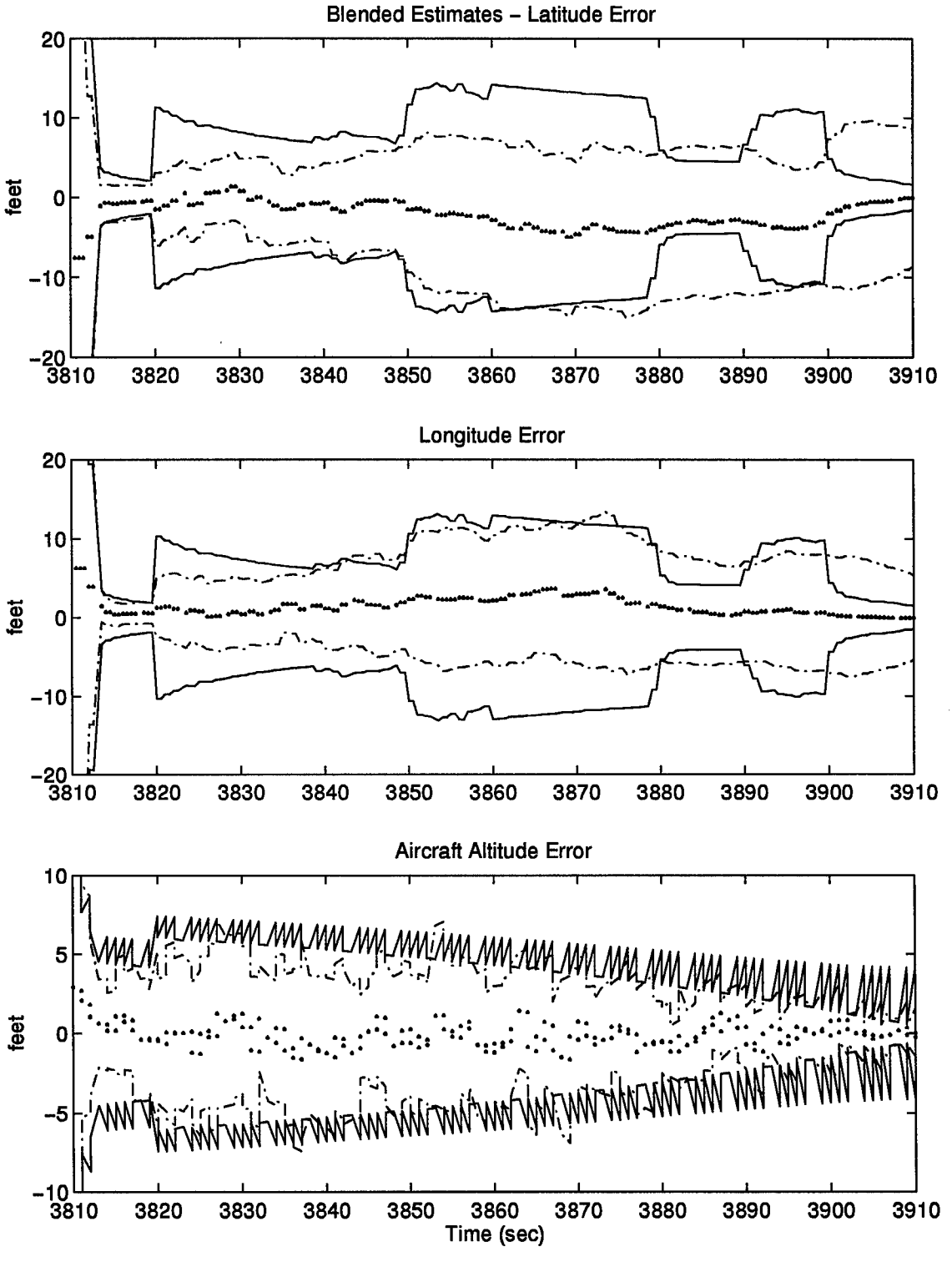

Figure 4.22 Navigation Performance, Interference Failures - Nav Case 1

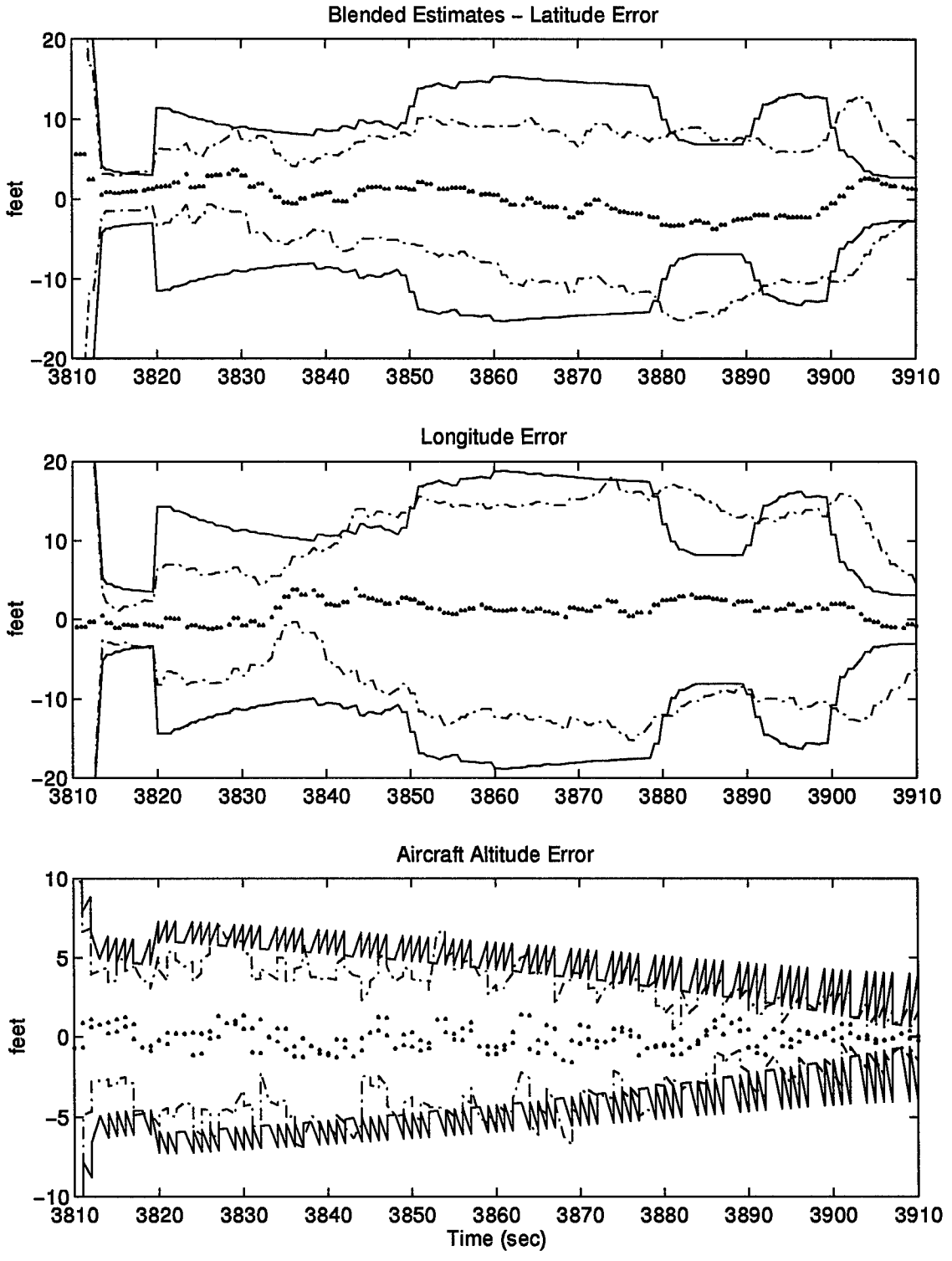

Figure 4.23 Navigation Performance, Interference Failures - Nav Case 2

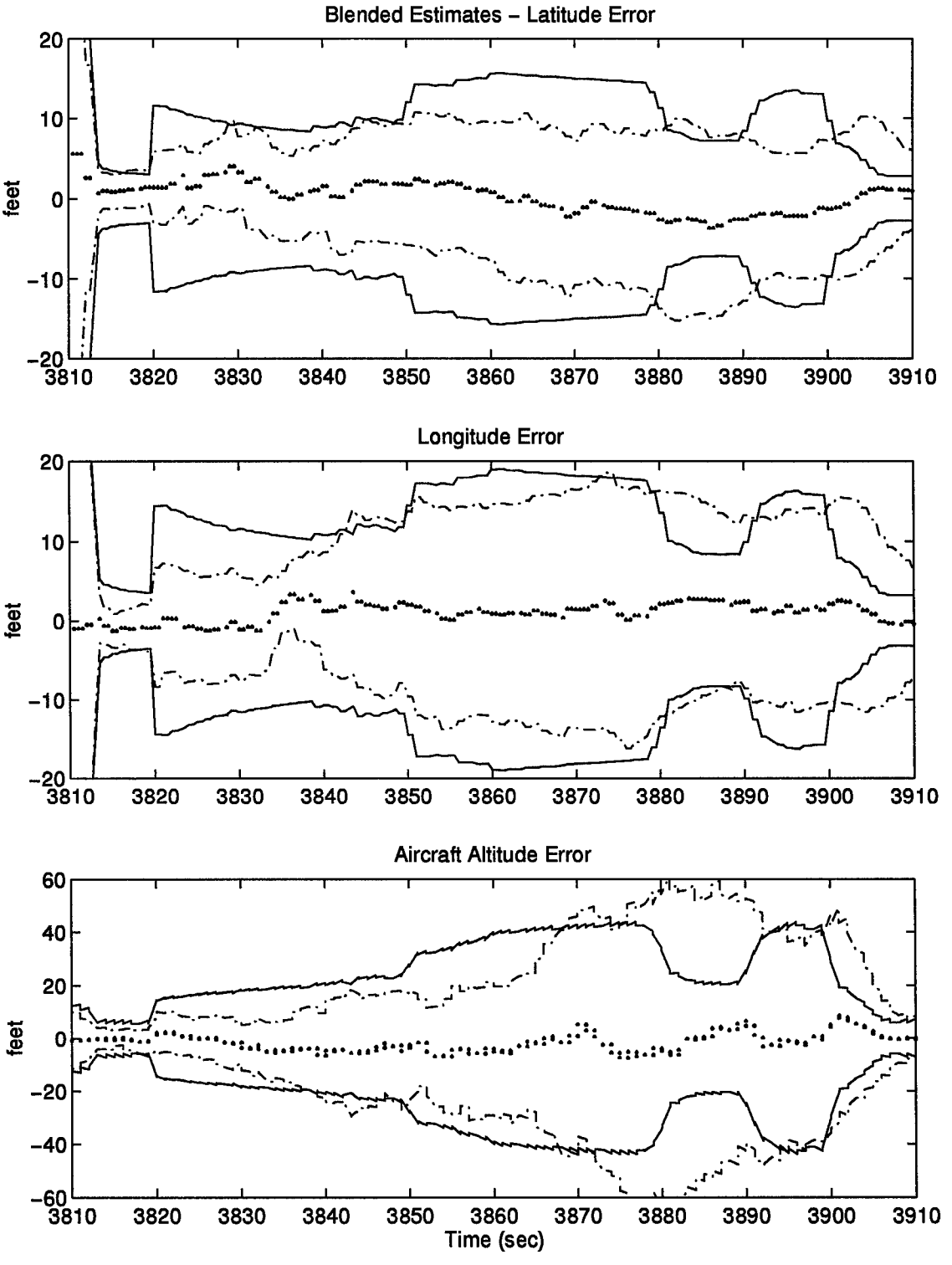

Figure 4.24 Navigation Performance, Interference Failures - Nav Case 3

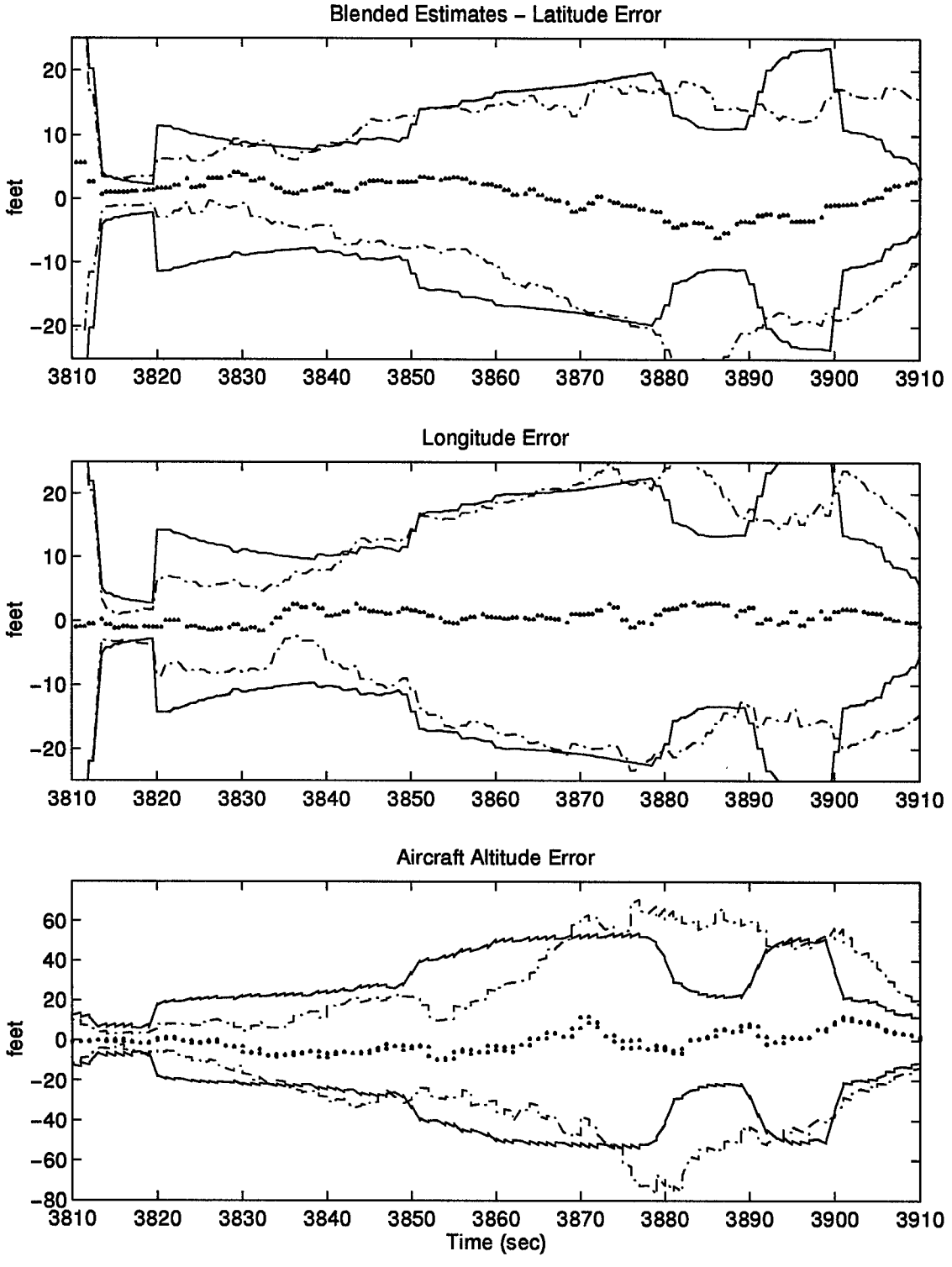

Figure 4.25 Navigation Performance, Interference Failures - Nav Case 4

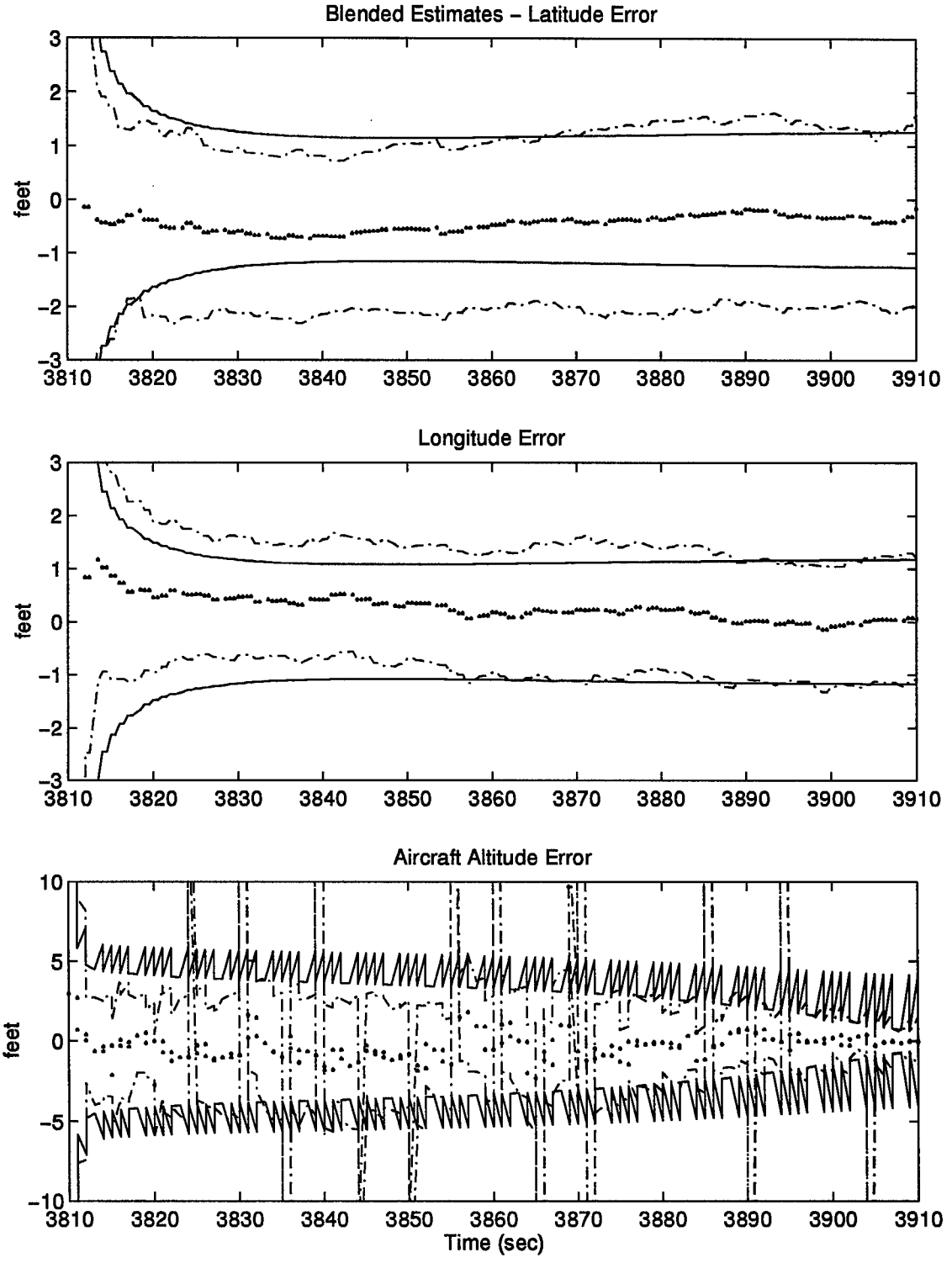

Figure 4.26 Navigation Performance, Spoofing Step Failures - Nav Case 1

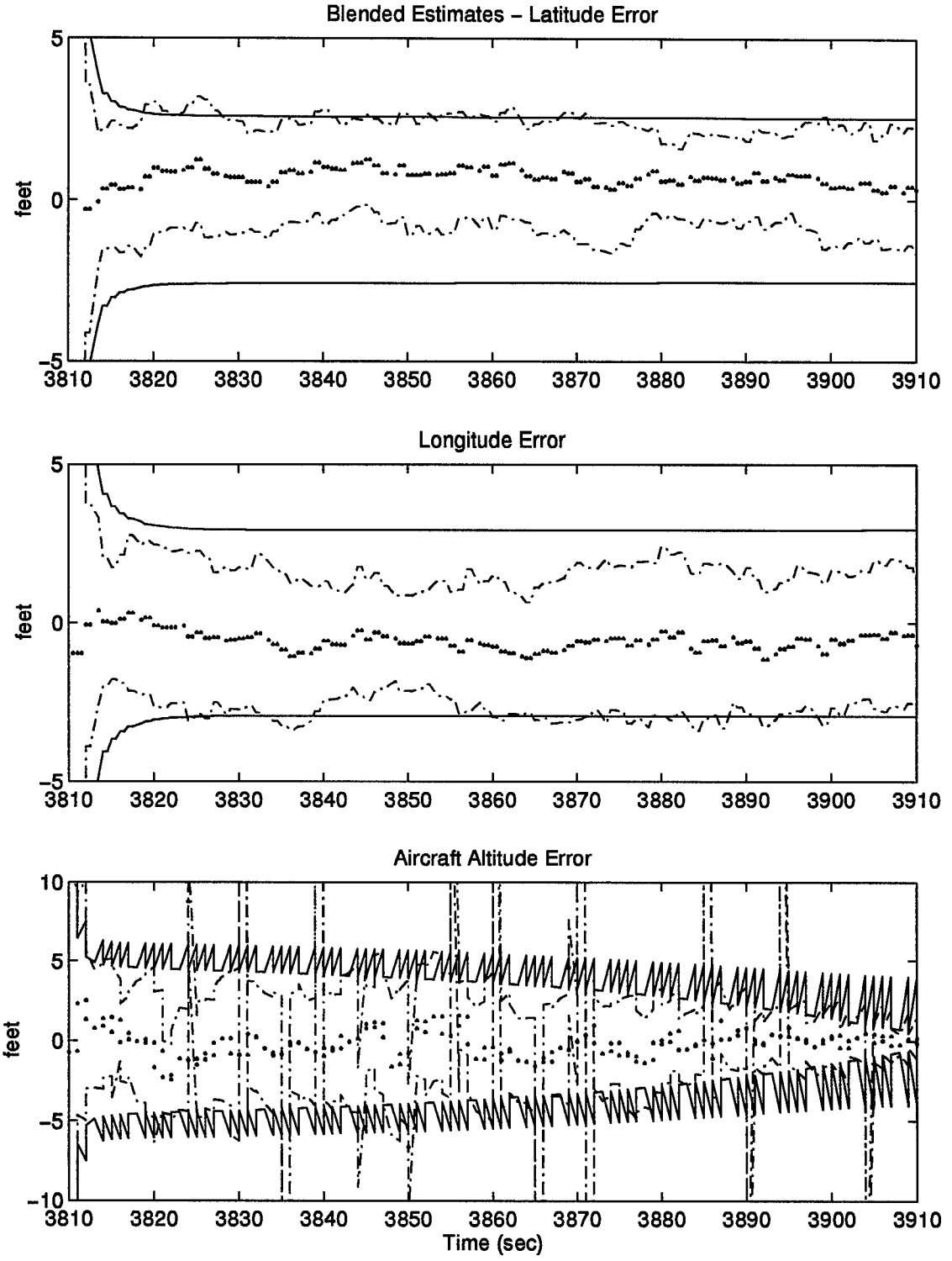

Figure 4.27 Navigation Performance, Spoofing Step Failures - Nav Case 2

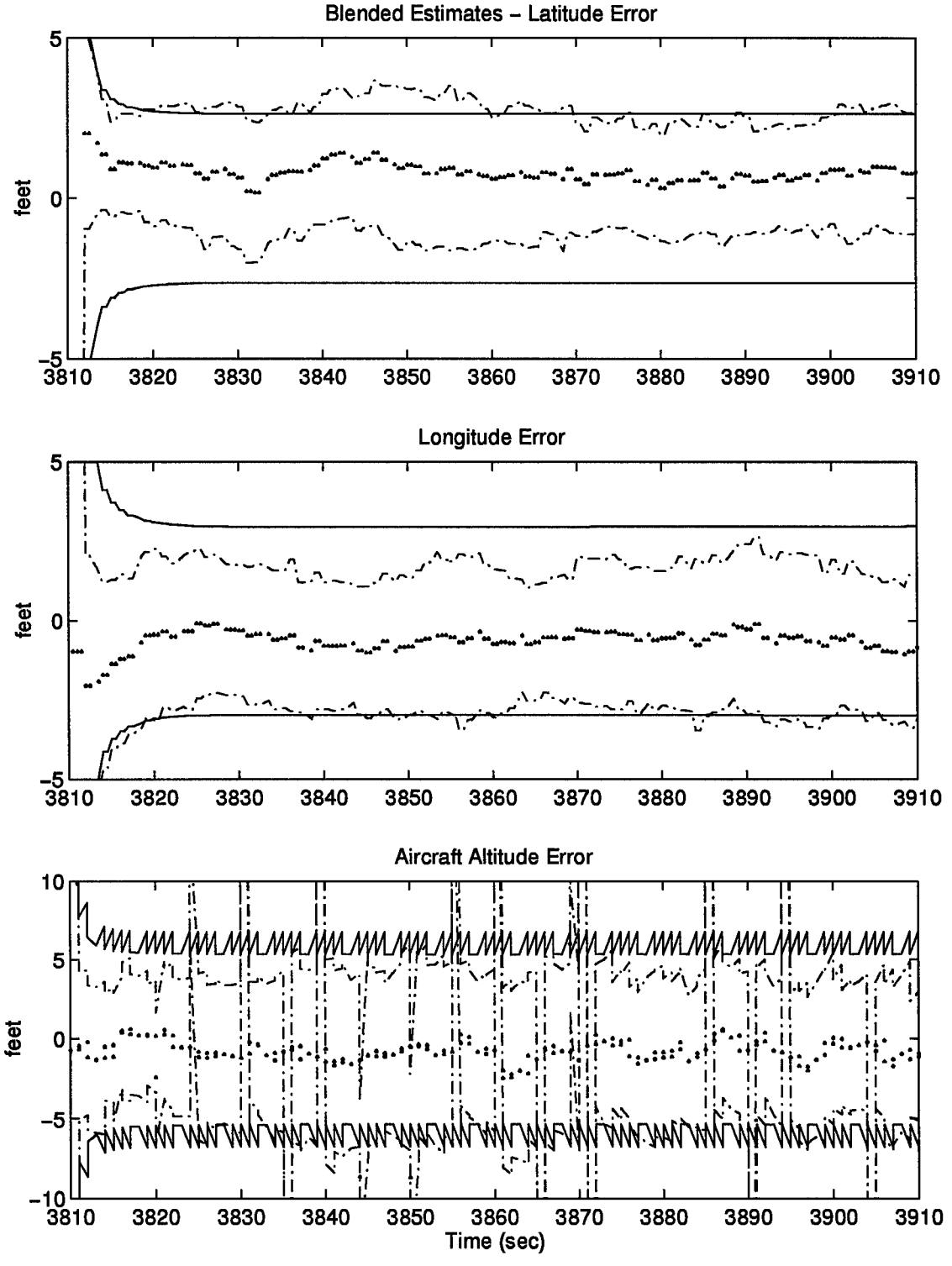

Figure 4.28 Navigation Performance, Spoofing Step Failures - Nav Case 3

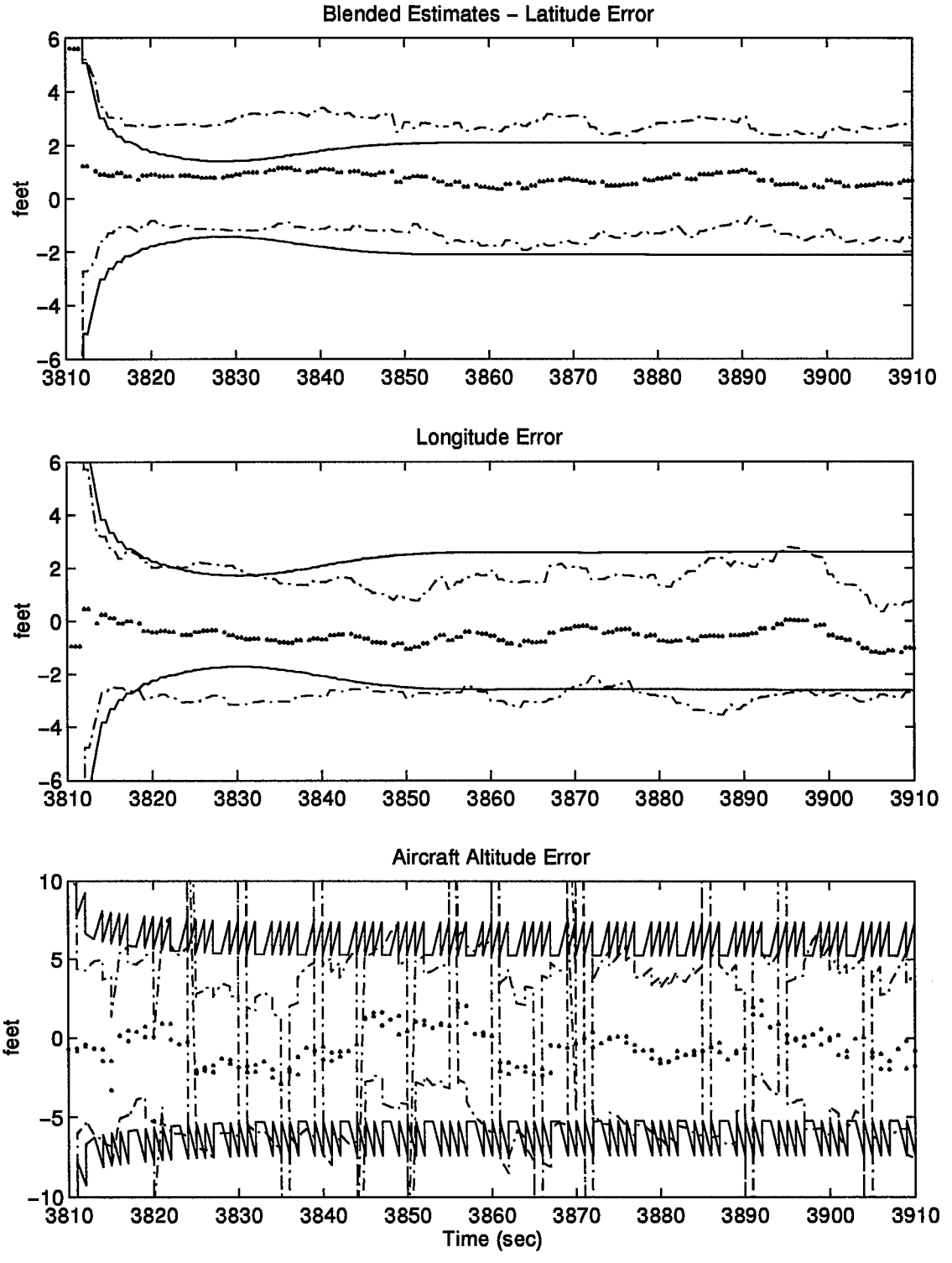

Figure 4.29 Navigation Performance, Spoofing Step Failures - Nav Case 4

in this work identifies *and* estimates the magnitudes of interference failures with a one sample period delay (one second). Identification of the interference magnitude is not attempted using the GLR/chi-square FDI methods. A much longer delay, 46 seconds in one case, is required for the GLR/chi-square algorithm to return to a nominal no-fail declaration from a large interference. MMAE's also suffer from a longer delay in returning to the nominal condition from a large failure, compared to the time required to move either from a nominal condition to a failure condition or from one failure condition to another. The first sub-plot of Figure 4.1 shows, for example, that ten seconds are required for the MMAE algorithm implemented in this research to return from a failed to a no-fail declaration. Nevertheless, this is far less than the 46 second delay of the GLR/chi-square alternative.

Spoofing type failures (step and ramp) were detected/identified less effectively by the GLR/chi-square detection scheme than were interference failures. Vasquez' [41, 42] simulations showed that spoofing step failures with magnitudes as low as 50 feet were detected and identified (accurate state tracking regained) in about 20 seconds. This report shows that the PRMMAE algorithm detects and identifies spoofing steps as small as *15 feet* with a *one second* delay. The largest (ramp) spoofing failure simulated by Vasquez was 2 feet/sec and required a minimum of 250 seconds (chi-square test) to detect this failure. The smallest spoofing ramp that was effectively detected *and* identified by PRMMAE techniques (in one second) was about 8 feet/sec; however, it can seen by close examination of Figure 4.10 that a spoofing ramp of 2 feet/sec is detected within about *5 seconds* even though reliable isolation at this low magnitude is prevented by system noise.

It may be concluded that GLR/chi-square failure testing has a significant associated delay when compared to MMAE failure testing, especially in the face of spoofing (bias-like) failures. The selection of one or the other of these FDI schemes for application to a particular problem should be made while fully taking into consideration this performance difference.

4.3 Conclusions

This chapter shows the development of moving-bank pseudo-residual MMAE identified as PRMMAE, a new technique for the identification of measurement offset (bias or ramp "spoof") failures. PRMMAE (for spoofs) and standard MMAE (for interference) are used to detect and compensate for interference and spoofing failures in the GPS portion of a navigation configuration. State estimation before, during, and after these failures is maintained. These results show that a precision landing system (PLS) based on these navigation components can reliably detect degradation of the navigation solution due to external rf sources, and can preserve the quality of navigation so that flight and some categories of instrument landings may safely be continued.

5. *Conclusions and Recommendations*

Chapter 4 presented and analyzed the results obtained using MMAE and the newly developed moving-bank pseudo-residual MMAE (PRMMAE) to detect and isolate interference and spoofing failures in a DGPS-aided inertial system. This chapter briefly summarizes and extends the conclusions suggested by the simulation results and analysis discussed at length in Chapter 4.

5.1 Introduction

Much recent research and FAA interest has been directed toward the development of a system to aid landing navigation to replace the aging instrument landing system (ILS) now in use. It is widely assumed that the replacement for the ILS will be based on the global positioning system (GPS). Previous research at AFIT has resulted in the development of a DGPS-aided (and radar altimeter-aided) INS-based precision landing system (PLS) capable of meeting the FAA precision requirements for instrument landings. The susceptibility of DGPS transmissions to interference/jamming and spoofing must be addressed before DGPS may be safely used as a major component of such a safety-of-flight critical navigational device.

This thesis applies multiple model adaptive estimation (MMAE) techniques to the problem of detecting and identifying interference/jamming or spoofing failures in the DGPS signal. Interference/jamming failures are modelled as significantly increased measurement noise associated with the DGPS pseudorange measurements. Spoofing failures are modelled as constant- (or ramped-) offset values added directly to the DGPS pseudorange measurements.

5.2 FDI Conclusions

The poor navigation performance of a non-adaptive Kalman filter subjected to interference/jamming or spoofing motivated the investigation of MMAE as a means

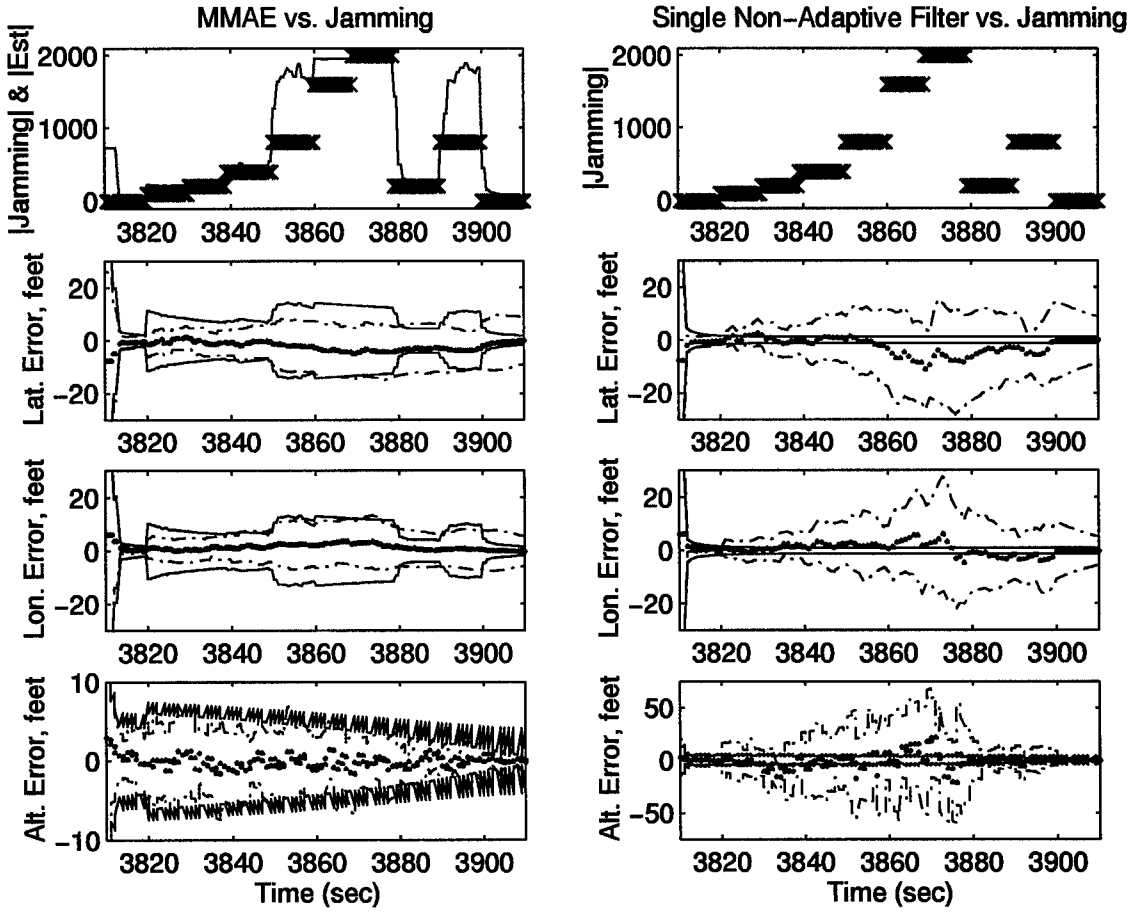

Figure 5.1 MMAE and Non-Adaptive Kalman Filter Navigation Performance Comparison, Interference/Jamming Failures

of failure detection and identification. MMAE is very effective at detecting and isolating interference/jamming failures. The navigation performance achieved using MMAE is greatly improved from the non-adaptive case. The left-hand side of Figure 5.1 shows the detection, identification, and resulting *blended* navigation performance of the MMAE filter bank subjected to a jamming signal. The right-hand side of Figure 5.1 shows the performance of a single non-adaptive Kalman filter with the same jamming signal applied. (Note the scale differences of the two altitude error subplots). Because of the good FDI performance of the MMAE algorithm (see Figure 5.1, first subplot, left-hand side), the navigation performance is essentially that of a single extended Kalman filter correctly designed and tuned for the navigation system components *and artificially informed* of the actual interference noise

variance. Figure 5.1 pertains to the case of a high-precision INS aided by both DGPS (with pseudolite) and a radar altimeter, but the conclusion remains valid even as pseudolite and radar altimeter measurements are removed, and the high-precision INS is replaced with an inexpensive INS with accuracy degraded by an order of magnitude.

When subjected to spoofing (measurement bias) failures, both the non-adaptive Kalman filter *and* a standard MMAE algorithm (designed to seek spoofing) yield extremely poor navigation solutions. This thesis shows the development of a moving-bank pseudo-residual MMAE (PRMMAE) algorithm to detect and identify spoofing failures in the DGPS measurements. The PRMMAE algorithm uses the pseudo-residuals, rather than the conventional residuals, to form $[\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r}]$ for use in the MMAE hypothesis probability calculations. Moving-bank PRMMAE is very effective at detecting and identifying spoof bias and ramp failures; the resulting navigation performance is equivalent to that of a single extended Kalman filter operating in a no-fail environment. Figure 5.2 shows the navigation performance of the PRMMAE algorithm compared to that of a non-adaptive extended Kalman filter when subjected to a sequence of spoofing bias failures (topmost subplots).

Detection and isolation of bias-like (or, even worse, ramp-like) failures on the measurement signal is a difficult task for any FDI algorithm, including MMAE. Perhaps the most significant result of this research is the development and verification of the moving-bank PRMMAE algorithm (see Section 4.2.4.1) to detect and isolate such bias-like or ramp-like failures. It is hypothesized that moving-bank PRMMAE will prove to be a useful and generally applicable tool for the detection and identification of signal bias-like or ramp-like failures.

Sections 4.2.3 and 4.2.4 show the FDI effectiveness of standard MMAE and moving-bank PRMMAE against DGPS interference and spoofing failures, respectively. It may be generally observed, based on the results of these sections, that detection of interference or spoofing failures (by two separate algorithms or, presum-

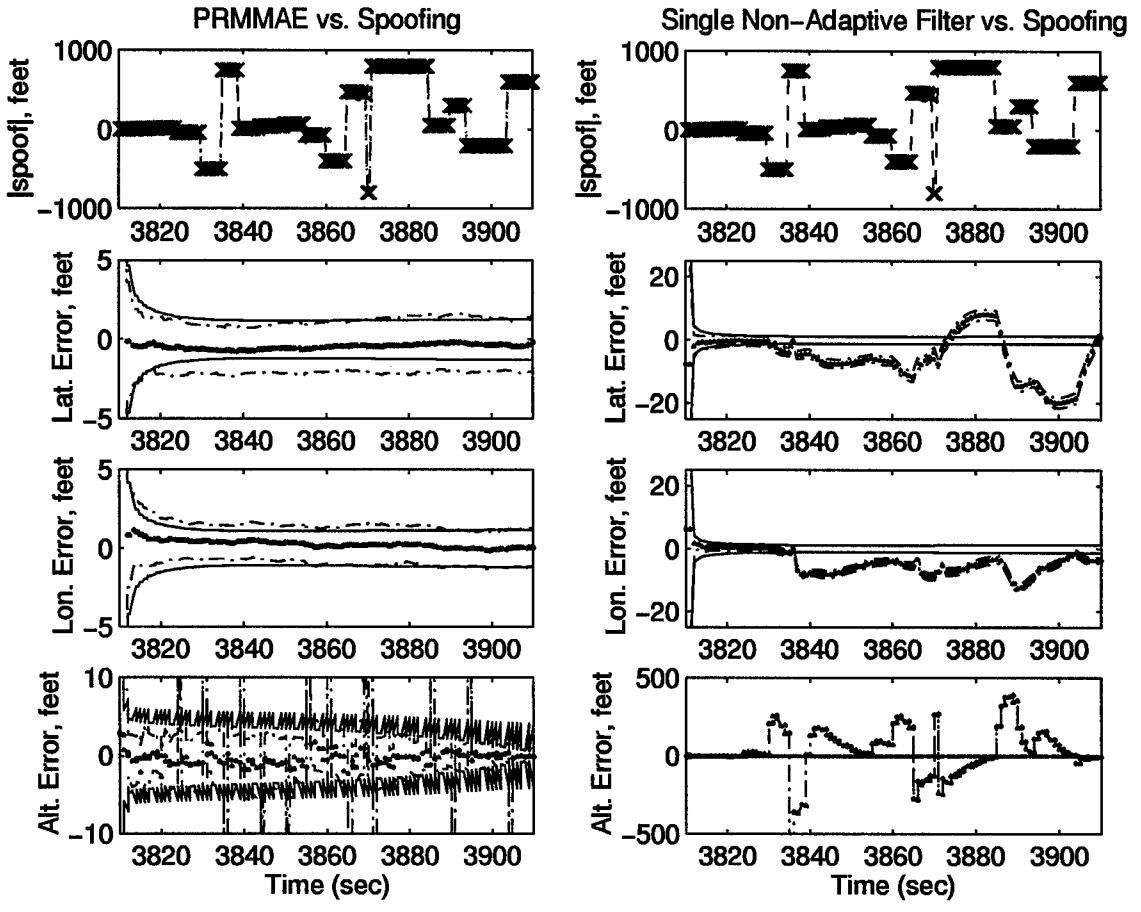

Figure 5.2 MMAE and Non-Adaptive Kalman Filter Navigation Performance Comparison, Spoofing Step Failures

ably, by two algorithms invoked in a hierarchical form, but not by a single filter bank: see Section 4.2.5) is accomplished in a single sample period (one second). Identification of correct parameter values associated with these failures, in general, requires between one and three seconds. This performance is exceptional when compared to a GLR/chi-square algorithm, (arguably) the next best FDI alternative in terms of effectiveness (see Section 4.2.7).

5.3 Navigation Conclusions

Detection and identification of interference/jamming and spoofing failures is accomplished very effectively using MMAE and moving-bank PRMMAE, as discussed in the previous section. The resulting navigation performance of the integrated system with these failures induced is only slightly degraded from that attained with no failures, rather than unacceptably degraded as is the case for a single non-adaptive extended Kalman filter as conventionally used for an integration filter for operational aided inertial navigation systems (compare Figure 4.2 to 4.3 and Figure 4.9 to 4.4). Table 5.1 shows a listing of the navigation component cases tested; these

Case	INS Type	GPS Type	Altimeter Aiding
1	0.4nm/hr	DGPS and Pseudolite	Baro and Radar Alt
2	0.4nm/hr	DGPS	Baro and Radar Alt
3	0.4nm/hr	DGPS	Baro
4	4.0nm/hr	DGPS	Baro

Table 5.1 Navigation Component Cases

cases were selected to demonstrate the FDI and navigation performance impact of eliminating (or degrading) the individual navigation sensors, with the hope of extending the results of this research to potential medium- and low-cost commercial or civilian aviation applications. As was hoped, the failure-induced degradation of navigation performance in the integrated system, based on the MMAE and moving-bank PRMMAE, for all navigation component cases and failures tested, is not substantial enough to prevent sufficiently accurate navigation to achieve mission requirements in other (not landing) phases of flight.

It is assumed that commercial and civilian aircraft will be much more likely to face interference/jamming GPS failures (due to the low cost of implementing an

interference device) than spoofing failures. Of particular note, then, is the FDI performance of each navigation component case against interference/jamming failures. Section 4.2.3 shows that the interference/jamming FDI performance of navigation component case four of Table 5.1 is essentially identical to that of navigation component case one. Spoofing identification is also accomplished very effectively by all four navigation component cases (see Section 4.2.4). As was desired, the FDI and navigation performance of these cases is good enough, and the cost of these lower-quality navigation components is low enough, to warrant investigating the application of the PLS to commercial and civil aviation as well as to military aircraft.

5.4 Recommendations for Future Research

Future researchers may find good results by pursuing:

- Hierarchical structure to accomplish FDI of jamming/interference and spoofing simultaneously.

  The results of Chapter 4 show that a composite MMAE filter bank (some elemental filters looking for spoof- and some for interference-type failures) cannot disambiguate between interference/jamming and spoofing failures. One can envision a hierarchical MMAE structure that initially looks only for spoofs (using a moving-bank PRMMAE algorithm structure), but if the estimated spoof exhibits large sample-to-sample changes, the elemental filters in the MMAE could be redefined to look for interference instead (via a non-moving-bank standard MMAE algorithm structure). After the noise variance is estimated, the MMAE elemental filters can be returned to look for spoofs only (and the algorithm returned to a moving-bank PRMMAE form), but now with each elemental filter being tuned for the correctly estimated measurement noise variance.

- Use of dither signals to enhance navigation performance of the integrated system.

Much research has demonstrated the usefulness of applying a known dither signal to improve the identifiability of a system [8, 19, 38]. The application of such a dither signal has the effect of "shaking" up the modes of the system that might otherwise be difficult to observe and use for identification purposes. While the FDI and navigation performance results obtained in this research are quite good, no attempt was made to apply a dither signal (or even to correlate the dynamics of the flight profile used) to achieve improved performance. Adding purposeful input dithers to enhance failure identifiability may warrant further investigation, particularly in phases of flight prior to final approach, flare, and landing.

Appendix A. Error Model State Definitions

This appendix contains a tabular listing of the 93 Litton INS, 39 reduced-order INS truth model, 30 GPS, 22 DGPS, and 13 simulation filter error states. Note that the INS model is shown with 41 states. This is because the two barometric aiding states are included as states 10 and 11 of this model. The 71-state (72-state when the pseudolite is included) PLS truth model states are defined in the right-most column of Tables A.5 through A.8.

A.1 Litton LN-93 Error-States

Tables A.1 through A.4 list the LN-93 error model (93 states) as defined in the Litton CDRL [16]. Note that the Litton document contains several errors which are corrected in these tables [30].

A.2 Reduced Order INS Truth Model States

Tables A.5 and A.6 list the 39 INS states (plus two barometric aiding states) used in all the full-order models in this thesis. This reduced order model was developed and verified by Negast [30]. States 12 and 13 of the PLS are defined later in Tables A.7 and A.8.

A.3 GPS Error States

Table A.7 lists the GPS error states. A total of 30 states are included to model the error characteristics of 4 space vehicles (7 states each) plus two states for user equipment error sources [30].

A.4 DGPS Error States

Table A.8 lists the DGPS error states developed by Negast [30]. A total of 22 states are included to model the error characteristics of 4 space vehicles (5 states each) plus user equipment (two states) error sources.

A.5 Simulation Filter States

Table A.9 lists the 13 states used in the PLS filter model.

Table A.1 Litton 93-state INS Model: INS States 1 $\rightarrow$ 29

State Number	State Symbol	Definition
1	$\delta\theta_x$	X-component of vector angle from true to computer frame
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame
4	ϕ_x	X-component of vector angle from true to platform frame
5	ϕ_y	Y-component of vector angle from true to platform frame
6	ϕ_z	Z-component of vector angle from true to platform frame
7	δV_x	X-component of error in computed velocity
8	δV_y	Y-component of error in computed velocity
9	δV_z	Z-component of error in computed velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_L	Error in lagged inertial altitude
12	δS_3	Error in vertical channel aiding state
13	δS_4	Error in vertical channel aiding state
14	b_{x_c}	X-component of gyro correlated drift rate
15	b_{y_c}	Y-component of gyro correlated drift rate
16	b_{z_c}	Z-component of gyro correlated drift rate
17	∇_{x_c}	X-component of accelerometer and velocity quantizer correlated noise
18	∇_{y_c}	Y-component of accelerometer and velocity quantizer correlated noise
19	∇_{z_c}	Z-component of accelerometer and velocity quantizer correlated noise
20	δg_x	X-component of gravity vector errors
21	δg_y	Y-component of gravity vector errors
22	δg_z	Z-component of gravity vector errors
23	δh_B	Total baro-altimeter correlated error
24	b_{x_t}	X-component of gyro trend
25	b_{y_t}	Y-component of gyro trend
26	b_{z_t}	Z-component of gyro trend
27	∇_{x_t}	X-component of accelerometer trend
28	∇_{y_t}	Y-component of accelerometer trend
29	∇_{z_t}	Z-component of accelerometer trend

Table A.2 Litton 93-state INS Model: INS States 30 → 47

State Number	State Symbol	Definition
30	b_x	X-component of gyro drift rate repeatability
31	b_y	Y-component of gyro drift rate repeatability
32	b_z	Z-component of gyro drift rate repeatability
33	S_{g_x}	X-component of gyro scale factor error
34	S_{g_y}	Y-component of gyro scale factor error
35	S_{g_z}	Z-component of gyro scale factor error
36	χ_1	X gyro misalignment about Y-axis
37	χ_2	Y gyro misalignment about X-axis
38	χ_3	Z gyro misalignment about X-axis
39	ν_1	X gyro misalignment about Z-axis
40	ν_2	Y gyro misalignment about Z-axis
41	ν_3	Z gyro misalignment about Y-axis
42	D_{xxx}	X gyro scale factor nonlinearity
43	D_{yyy}	Y gyro scale factor nonlinearity
44	D_{zzz}	Z gyro scale factor nonlinearity
45	S_{Qb_x}	X gyro scale factor asymmetry error
46	S_{Qb_y}	Y gyro scale factor asymmetry error
47	S_{Qb_z}	Z gyro scale factor asymmetry error

Table A.3 Litton 93-state INS Model: INS States 48 → 69

State Number	State Symbol	Definition
48	∇_{b_x}	X-component of accelerometer bias repeatability
49	∇_{b_y}	Y-component of accelerometer bias repeatability
50	∇_{b_z}	Z-component of accelerometer bias repeatability
51	S_{A_x}	X-component of accelerometer and velocity quantizer scale factor error
52	S_{A_y}	Y-component of accelerometer and velocity quantizer scale factor error
53	S_{A_z}	Z-component of accelerometer and velocity quantizer scale factor error
54	S_{QA_x}	X-component of accelerometer and velocity quantizer scale factor asymmetry
55	S_{QA_y}	Y-component of accelerometer and velocity quantizer scale factor asymmetry
56	S_{QA_z}	Z-component of accelerometer and velocity quantizer scale factor asymmetry
57	f_{xx}	Coefficient of error proportional to square of measured acceleration
58	f_{yy}	Coefficient of error proportional to square of measured acceleration
59	f_{zz}	Coefficient of error proportional to square of measured acceleration
60	f_{xy}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
61	f_{xz}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
62	f_{yx}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
63	f_{yz}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
64	f_{zx}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
65	f_{zy}	Coefficient of error proportional to products of acceleration along and orthogonal to accelerometer sensitive axis
66	μ_1	X accelerometer misalignment about Z-axis
67	μ_2	Y accelerometer misalignment about Z-axis
68	μ_3	Z accelerometer misalignment about Y-axis
69	σ_3	Z-accelerometer misalignment about X-axis

Table A.4 Litton 93-state INS Model: INS States 70 → 93

State Number	State Symbol	Definition
70	∇_{x_q}	X-component of accelerometer bias thermal transient
71	∇_{y_q}	Y-component of accelerometer bias thermal transient
72	∇_{z_q}	Z-component of accelerometer bias thermal transient
73	b_{x_q}	X-component of initial gyro drift rate bias thermal transient
74	b_{y_q}	Y-component of initial gyro drift rate bias thermal transient
75	b_{z_q}	Z-component of initial gyro drift rate bias thermal transient
76	F_{xyz}	X gyro compliance term
77	F_{xyy}	X gyro compliance term
78	F_{xyx}	X gyro compliance term
79	F_{xzy}	X gyro compliance term
80	F_{xzz}	X gyro compliance term
81	F_{xzx}	X gyro compliance term
82	F_{yzx}	Y gyro compliance term
83	F_{yzz}	Y gyro compliance term
84	F_{yzy}	Y gyro compliance term
85	F_{yxz}	Y gyro compliance term
86	F_{yxx}	Y gyro compliance term
87	F_{yxy}	Y gyro compliance term
88	F_{zxy}	Z gyro compliance term
89	F_{zxx}	Z gyro compliance term
90	F_{zxz}	Z gyro compliance term
91	F_{zyx}	Z gyro compliance term
92	F_{zyy}	Z gyro compliance term
93	F_{zyz}	Z gyro compliance term

Table A.5 Reduced-Order INS System Model: INS States 1 $\rightarrow$ 20

State Number	State Symbol	Definition	LN-93 State	PLS State
1	$\delta\theta_x$	X-component of vector angle from true to computer frame	1	1
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame	2	2
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame	3	3
4	ϕ_x	X-component of vector angle from true to platform frame	4	4
5	ϕ_y	Y-component of vector angle from true to platform frame	5	5
6	ϕ_z	Z-component of vector angle from true to platform frame	6	6
7	δV_x	X-component of error in computed velocity	7	7
8	δV_y	Y-component of error in computed velocity	8	8
9	δV_z	Z-component of error in computed velocity	9	9
10	δh	Error in vehicle altitude above reference ellipsoid	10	10
11	δh_B	Total baro-altimeter correlated error	23	11
12	δh_L	Error in lagged inertial altitude	11	14
13	δS_3	Error in vertical channel aiding state	12	15
14	δS_4	Error in vertical channel aiding state	13	16
15	∇_{x_c}	X-component of accelerometer and velocity quantizer correlated noise	17	17
16	∇_{y_c}	Y-component of accelerometer and velocity quantizer correlated noise	18	18
17	∇_{z_c}	Z-component of accelerometer and velocity quantizer correlated noise	19	19
18	δg_x	X-component of gravity vector errors	20	20
19	δg_y	Y-component of gravity vector errors	21	21
20	δg_z	Z-component of gravity vector errors	22	22

Table A.6 Reduced-Order INS System Model: INS States 21 → 41

State Number	State Symbol	Definition	LN-93 State	PLS State
21	b_x	X-component of gyro drift rate repeatability	30	23
22	b_y	Y-component of gyro drift rate repeatability	31	24
23	b_z	Z-component of gyro drift rate repeatability	32	25
24	S_{g_x}	X-component of gyro scale factor error	33	26
25	S_{g_y}	Y-component of gyro scale factor error	34	27
26	S_{g_z}	Z-component of gyro scale factor error	35	28
27	∇_{b_x}	X-component of accelerometer bias repeatability	48	29
28	∇_{b_y}	Y-component of accelerometer bias repeatability	49	30
29	∇_{b_z}	Z-component of accelerometer bias repeatability	50	31
30	S_{A_x}	X-component of accelerometer and velocity quantizer scale factor error	51	32
31	S_{A_y}	Y-component of accelerometer and velocity quantizer scale factor error	52	33
32	S_{A_z}	Z-component of accelerometer and velocity quantizer scale factor error	53	34
33	S_{QA_x}	X-component of accelerometer and velocity quantizer scale factor asymmetry	54	35
34	S_{QA_y}	Y-component of accelerometer and velocity quantizer scale factor asymmetry	55	36
35	S_{QA_z}	Z-component of accelerometer and velocity quantizer scale factor asymmetry	56	37
36	μ_1	X accelerometer misalignment about Z-axis	66	38
37	μ_2	Y accelerometer misalignment about Z-axis	67	39
38	μ_3	Z accelerometer misalignment about Y-axis	68	40
39	σ_1	X-accelerometer misalignment about Y-axis		41
40	σ_2	Y-accelerometer misalignment about X-axis		42
41	σ_3	Z-accelerometer misalignment about X-axis	69	43

Table A.7 GPS Error States

State Number	State Symbol	Definition	PLS State
1	δR_{clk_u}	User clock bias	12
2	δD_{clk_u}	User clock drift	13
3	δR_{cloop_1}	SV 1 code loop error	44
4	δR_{trop_1}	SV 1 tropospheric error	45
5	δR_{ion_1}	SV 1 ionospheric error	46
6	$\delta R_{clk_{sv_1}}$	SV 1 clock error	47
7	δx_{sv_1}	SV 1 x-component of position error	48
8	δy_{sv_1}	SV 1 y-component of position error	49
9	δz_{sv_1}	SV 1 z-component of position error	50
10	δR_{cloop_2}	SV 2 code loop error	51
11	δR_{trop_2}	SV 2 tropospheric error	52
12	δR_{ion_2}	SV 2 ionospheric error	53
13	$\delta R_{clk_{sv_2}}$	SV 2 clock error	54
14	δx_{sv_2}	SV 2 x-component of position error	55
15	δy_{sv_2}	SV 2 y-component of position error	56
16	δz_{sv_2}	SV 2 z-component of position error	57
17	δR_{cloop_3}	SV 3 code loop error	58
18	δR_{trop_3}	SV 3 tropospheric error	59
19	δR_{ion_3}	SV 3 ionospheric error	60
20	$\delta R_{clk_{sv_3}}$	SV 3 clock error	61
21	δx_{sv_3}	SV 3 x-component of position error	62
22	δy_{sv_3}	SV 3 y-component of position error	63
23	δz_{sv_3}	SV 3 z-component of position error	64
24	δR_{cloop_4}	SV 4 code loop error	65
25	δR_{trop_4}	SV 4 tropospheric error	66
26	δR_{ion_4}	SV 4 ionospheric error	67
27	$\delta R_{clk_{sv_4}}$	SV 4 clock error	68
28	δx_{sv_4}	SV 4 x-component of position error	69
29	δy_{sv_4}	SV 4 y-component of position error	70
30	δz_{sv_4}	SV 4 z-component of position error	71

Table A.8 DGPS Error States

State Number	State Symbol	Definition	PLS State
1	δR_{Uclk_u}	User clock bias	12
2	δD_{Uclk_u}	User clock drift	13
3	δR_{trop_1}	SV 1 tropospheric error	45
4	δR_{ion_1}	SV 1 ionospheric error	46
5	δx_{sv_1}	SV 1 x-component of position error	48
6	δy_{sv_1}	SV 1 y-component of position error	49
7	δz_{sv_1}	SV 1 z-component of position error	50
8	δR_{trop_2}	SV 2 tropospheric error	52
9	δR_{ion_2}	SV 2 ionospheric error	53
10	δx_{sv_2}	SV 2 x-component of position error	55
11	δy_{sv_2}	SV 2 y-component of position error	56
12	δz_{sv_2}	SV 2 z-component of position error	57
13	δR_{trop_3}	SV 3 tropospheric error	59
14	δR_{ion_3}	SV 3 ionospheric error	60
15	δx_{sv_3}	SV 3 x-component of position error	62
16	δy_{sv_3}	SV 3 y-component of position error	63
17	δz_{sv_3}	SV 3 z-component of position error	64
18	δR_{trop_4}	SV 4 tropospheric error	66
19	δR_{ion_4}	SV 4 ionospheric error	67
20	δx_{sv_4}	SV 4 x-component of position error	69
21	δy_{sv_4}	SV 4 y-component of position error	70
22	δz_{sv_4}	SV 4 z-component of position error	71

Table A.9 Simulation Filter States:

State Number	State Symbol	Definition
1	$\delta\theta_x$	X-component of vector angle from true to computer frame
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame
4	ϕ_x	X-component of vector angle from true to platform frame
5	ϕ_y	Y-component of vector angle from true to platform frame
6	ϕ_z	Z-component of vector angle from true to platform frame
7	δV_x	X-component of error in computed velocity
8	δV_y	Y-component of error in computed velocity
9	δV_z	Z-component of error in computed velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_B	Total baro-altimeter correlated error
12	δR_{Uclk_u}	GPS User clock bias
13	δD_{Uclk_u}	GPS User clock drift

Appendix B. Dynamics and Noise Matrices

B.1 Model Dynamics Matrices [16, 41, 42]

The LN-93 error-state dynamics matrix **F** as provided by Litton is a 93-by-93 array that contains a large number of elements that are identically zero. Litton partitions the **F** matrix into thirty-six subarrays [16] reflecting the logical divisions of error sources discussed in Chapter 3.

The *non-zero* elements of the Litton model are included in Tables B.1 through B.8. The dynamics submatrices for the 41-state INS model may be constructed as needed from the LN-93 dynamics submatrices and the full- to reduced-order state relations given in Tables A.1 through A.4. The reader should note that Negast's [30] revised baro-altimeter model states are *not* included in this set of *original* **F** matrix elements extracted from the Litton document [16].

A notational convention is to label elements of the $\mathbf{C}_s^t$, sensor-to-true, matrix as C_{ij} where i is the row and j is the column in the transformation matrix.

The dynamics matrices for the GPS and DGPS truth models are included in the body of Chapter 3.

B.2 Process Noise Matrices

The Litton document [16] includes a 93-by-93 process noise matrix **Q** for the LN-93 error model. Like the **F** matrix, the **Q** matrix is partitioned into subarrays which correspond to the error-state subvectors discussed in Chapter 3. The vast majority of the elements in the process noise matrix are identically zero. The non-zero elements of **Q** are shown in Tables B.9 and B.10.

The process noise matrices for the GPS and DGPS models are included in the body of Chapter 3, while the Q noise strengths used to tune the 13-state DGPS filter model for each of the four navigation component cases is given in Appendix C.

Table B.1 Elements of the Dynamics Submatrix $\mathbf{F}_{11}$

Element	Term	Element	Term
(1,3)	$-\rho_y$	(1,8)	$-C_{RY}$
(2,3)	ρ_x	(2,7)	C_{RX}
(3,1)	ρ_y	(3,2)	$-\rho_x$
(4,2)	$-\Omega_z$	(4,3)	Ω_y
(4,5)	ω_{in_z}	(4,6)	$-\omega_{in_y}$
(4,8)	$-C_{RY}$	(5,1)	Ω_z
(5,3)	$-\Omega_x$	(5,4)	$-\omega_{in_z}$
(5,6)	ω_{in_x}	(5,7)	C_{RX}
(6,1)	$-\Omega_y$	(6,2)	Ω_x
(6,4)	ω_{in_y}	(6,5)	$-\omega_{in_x}$
(7,1)	$-2V_y\Omega_y - 2V_z\Omega_z$	(7,2)	$2V_y\Omega_x$
(7,3)	$2V_z\Omega_y$	(7,5)	$-A_z$
(7,6)	A_y	(7,7)	$-V_zC_{RX}$
(7,8)	$2\Omega_z$	(7,9)	$-\rho_y - 2\Omega_y$
(8,1)	$2V_x\Omega_y$	(8,2)	$-2V_x\Omega_x - 2V_z\Omega_z$
(8,3)	$2v_z\Omega_y$	(8,4)	A_z
(8,6)	$-A_x$	(8,7)	$-2\Omega_z$
(8,8)	$-V_zC_{RY}$	(8,9)	$\rho_x + 2\Omega_x$
(9,1)	$2V_x\Omega_z$	(9,2)	$2V_y\Omega_z$
(9,3)	$-2V_y\Omega_y - 2V_x\Omega_x$	(9,4)	$-A_y$
(9,5)	A_x	(9,7)	$\rho_y + 2\Omega_y + V_xC_{RX}$
(9,8)	$-\rho_x - 2\Omega_x + V_yC_{RY}$	(9,10)	$2g_o/a$
(9,11)	$-k_2$	(9,12)	-1
(9,13)	k_2	(10,9)	1
(10,11)	$-k_1$	(10,13)	$k_1 - 1$
(11,10)	1	(11,11)	-1
(12,11)	k_3	(12,13)	$-k_3$
(13,10)	k_4	(13,11)	$-k_4$
(13,13)	$k_4 - 1$		

$\rho_{x,y}$ = Components of angular rate, nav reference frame to earth-fixed frame
$\Omega_{x,y,z}$ = Components of angular rate, earth-fixed frame to inertial frame
$\omega_{in_{x,y,z}}$ = Components of angular rate, nav reference frame to inertial frame
$V_{x,y,z}$ = Components of vehicle velocity vector in earth-fixed coordinates
$A_{x,y,z}$ = Components of specific force in the sensor reference frame
$k_{1,2,3,4}$ = Vertical channel gains. See LN-93 documentation [17] for equations
$C_{RX,RY}$ = Components of earth spheroid inverse radii of curvature
g_0 = Equatorial gravity magnitude ($32.08744 ft/sec^2$)
a = Equatorial radius of the earth ($6378388m$)

Table B.2 Elements of the Dynamics Submatrix $\mathbf{F}_{12}$

Element	Term	Element	Term	Element	Term
(4,14)	C_{11}	(4,15)	C_{12}	(4,16)	C_{13}
(4,24)	$C_{11}t$	(4,25)	$C_{12}t$	(4,26)	$C_{13}t$
(5,14)	C_{21}	(5,15)	C_{22}	(5,16)	C_{23}
(5,24)	$C_{21}t$	(5,25)	$C_{22}t$	(5,26)	$C_{23}t$
(6,14)	C_{31}	(6,15)	C_{32}	(6,16)	C_{33}
(6,24)	$C_{31}t$	(6,25)	$C_{32}t$	(6,26)	$C_{33}t$
(7,17)	C_{11}	(7,18)	C_{12}	(7,19)	C_{13}
(7,20)	1	(7,27)	$C_{11}t$	(7,28)	$C_{12}t$
(7,29)	$C_{13}t$	(8,17)	C_{21}	(8,18)	C_{22}
(8,19)	C_{23}	(8,21)	1	(8,27)	$C_{21}t$
(8,28)	$C_{22}t$	(8,29)	$C_{23}t$	(9,17)	C_{31}
(9,18)	C_{32}	(9,19)	C_{33}	(9,22)	1
(9,23)	k_2	(9,27)	$C_{31}t$	(9,28)	$C_{32}t$
(9,29)	$C_{33}t$	(10,23)	k_1	(12,23)	$-k_3$
(13,23)	$k_4/600$				

Note: For the above element definitions $t_0 = 0$

$\mathbf{C}$ = Coordinate transformation matrix, body frame to nav reference frame

Table B.3 Elements of the Dynamics Submatrix $\mathbf{F}_{13}$

Element	Term	Element	Term	Element	Term
(4,30)	C_{11}	(4,31)	C_{12}	(4,32)	C_{13}
(4,33)	$C_{11}\omega_{in_x}$	(4,34)	$C_{12}\omega_{in_y}$	(4,35)	$C_{13}\omega_{in_z}$
(4,36)	$C_{11}\omega_{in_z}$	(4,37)	$-C_{12}\omega_{in_z}$	(4,38)	$C_{13}\omega_{in_y}$
(4,39)	$-C_{11}\omega_{in_y}$	(4,40)	$C_{12}\omega_{in_x}$	(4,41)	$-C13\omega_{in_x}$
(4,42)	$C_{11}\omega_{in_x}^2$	(4,43)	$C_{12}\omega_{in_y}^2$	(4,44)	$C_{13}\omega_{in_z}^2$
(4,45)	$0.5C_{11}\lvert\omega_{in_x}\rvert$	(4,46)	$0.5C_{12}\lvert\omega_{in_y}\rvert$	(4,47)	$0.5C_{13}\lvert\omega_{in_z}\rvert$
(5,30)	C_{21}	(5,31)	C_{22}	(5,32)	C_{23}
(5,33)	$C_{21}\omega_{in_x}$	(5,34)	$C_{22}\omega_{in_y}$	(5,35)	$C_{23}\omega_{in_z}$
(5,36)	$C_{21}\omega_{in_z}$	(5,37)	$-C_{22}\omega_{in_z}$	(5,38)	$C_{23}\omega_{in_y}$
(5,39)	$-C_{21}\omega_{in_y}$	(5,40)	$C_{22}\omega_{in_x}$	(5,41)	$-C_{23}\omega_{in_x}$
(5,42)	$C_{21}\omega_{in_x}^2$	(5,43)	$C_{22}\omega_{in_y}^2$	(5,44)	$C_{23}\omega_{in_z}^2$
(5,45)	$0.5C_{21}\lvert\omega_{in_x}\rvert$	(5,46)	$0.5C_{22}\lvert\omega_{in_y}\rvert$	(5,47)	$0.5C_{23}\lvert\omega_{in_z}\rvert$
(6,30)	C_{31}	(6,31)	C_{32}	(6,32)	C_{33}
(6,33)	$C_{31}\omega_{in_x}$	(6,34)	$C_{32}\omega_{in_y}$	(6,35)	$C_{33}\omega_{in_z}$
(6,36)	$C_{31}\omega_{in_z}$	(6,37)	$-C_{32}\omega_{in_z}$	(6,38)	$C_{33}\omega_{in_y}$
(6,39)	$-C_{31}\omega_{in_y}$	(6,40)	$C_{32}\omega_{in_x}$	(6,41)	$-C_{33}\omega_{in_x}$
(6,42)	$C_{31}\omega_{in_x}^2$	(6,43)	$C_{32}\omega_{in_y}^2$	(6,44)	$C_{33}\omega_{in_z}^2$
(6,45)	$0.5C_{31}\lvert\omega_{in_x}\rvert$	(6,46)	$0.5C_{32}\lvert\omega_{in_y}\rvert$	(6,47)	$0.5C_{33}\lvert\omega_{in_z}\rvert$

Table B.4 Elements of the Dynamics Submatrix $\mathbf{F}_{14}$

Element	Term	Element	Term	Element	Term
(7,48)	C_{11}	(7,49)	C_{12}	(7,50)	C_{13}
(7,51)	$C_{11}A_x^B$	(7,52)	$C_{12}A_y^B$	(7,53)	$C_{13}A_z^{B'}$
(7,54)	$C_{11}\|A_x^B\|$	(7,55)	$C_{12}\|A_y^B\|$	(7,56)	$C_{13}\|A_z^{B'}\|$
(7,57)	$C_{11}A_x^{B2}$	(7,58)	$C_{12}A_y^{B2}$	(7,59)	$C_{13}A_z^{B'2}$
(7,60)	$C_{11}A_x^BA_y^B$	(7,61)	$C_{11}A_x^BA_z^B$	(7,62)	$C_{12}A_y^BA_x^B$
(7,63)	$C_{12}A_y^BA_z^B$	(7,64)	$C_{13}A_x^BA_z^B$	(7,65)	$C_{13}A_y^BA_z^B$
(7,66)	$C_{11}A_y^B$	(7,67)	$-C_{12}A_x^B$	(7,68)	$C_{13}A_y^B$
(7,69)	$C_{13}A_x^B$	(8,48)	C_{21}	(8,49)	C_{22}
(8,50)	C_{23}	(8,51)	$C_{21}A_x^B$	(8,52)	$C_{22}A_y^B$
(8,53)	$C_{23}A_z^{B'}$	(8,54)	$C_{21}\|A_x^B\|$	(8,55)	$C_{22}\|A_y^B\|$
(8,56)	$C_{23}\|A_z^{B'}\|$	(8,57)	$C_{21}A_x^{B2}$	(8,58)	$C_{22}A_y^{B2}$
(8,59)	$C_{23}A_z^{B'2}$	(8,60)	$C_{21}A_x^BA_y^B$	(8,61)	$C_{21}A_x^BA_z^B$
(8,62)	$C_{22}A_y^BA_x^B$	(8,63)	$C_{22}A_y^BA_z^B$	(8,64)	$C_{23}A_x^BA_z^B$
(8,65)	$C_{23}A_y^BA_z^B$	(8,66)	$C_{21}A_y^B$	(8,67)	$-C_{22}A_x^B$
(8,68)	$C_{23}A_y^B$	(8,69)	$C_{23}A_x^B$	(9,48)	C_{31}
(9,49)	C_{32}	(9,50)	C_{33}	(9,51)	$C_{31}A_x^B$
(9,52)	$C_{32}A_y^B$	(9,53)	$C_{33}A_z^{B'}$	(9,54)	$C_{31}\|A_x^B\|$
(9,55)	$C_{32}\|A_y^B\|$	(9,56)	$C_{33}\|A_z^{B'}\|$	(9,57)	$C_{31}A_x^{B2}$
(9,58)	$C_{32}A_y^{B2}$	(9,59)	$C_{33}A_z^{B'2}$	(9,60)	$C_{31}A_x^BA_y^B$
(9,61)	$C_{31}A_x^BA_z^B$	(9,62)	$C_{32}A_y^BA_x^B$	(9,63)	$C_{32}A_y^BA_z^B$
(9,64)	$C_{33}A_x^BA_z^B$	(9,65)	$C_{33}A_y^BA_z^B$	(9,66)	$C_{31}A_y^B$
(9,67)	$-C_{32}A_x^B$	(9,68)	$C_{33}A_y^B$	(9,69)	$C_{33}A_x^B$

$A_{x,y,z}^B$ = Components of acceleration in the body frame
$A_z^{B'}$ = Specific force component (includes gravity)

Table B.5 Elements of the Dynamics Submatrix $\mathbf{F}_{15}$

Element	Term	Element	Term	Element	Term
(4,73)	C_{11}	(4,74)	C_{12}	(4,75)	C_{13}
(5,73)	C_{21}	(5,74)	C_{22}	(5,75)	C_{23}
(6,73)	C_{31}	(6,74)	C_{32}	(6,75)	C_{33}
(7,70)	C_{11}	(7,71)	C_{12}	(7,72)	C_{13}
(8,70)	C_{21}	(8,71)	C_{22}	(8,72)	C_{23}
(9,70)	C_{31}	(9,71)	C_{32}	(9,72)	C_{33}

Table B.6 Elements of the Dynamics Submatrix $\mathbf{F}_{16}$

Element	Term	Element	Term	Element	Term
(4,76)	$C_{11}A_y^B\omega_{in_z}$	(4,77)	$C_{11}A_y^B\omega_{in_y}$	(4,78)	$C_{11}A_y^B\omega_{in_x}$
(4,79)	$C_{11}A_z^B\omega_{in_y}$	(4,80)	$C_{11}A_z^B\omega_{in_z}$	(4,81)	$C_{11}A_z^B\omega_{in_x}$
(4,82)	$C_{12}A_z^B\omega_{in_x}$	(4,83)	$C_{12}A_z^B\omega_{in_z}$	(4,84)	$C_{12}A_z^B\omega_{in_y}$
(4,85)	$C_{12}A_x^B\omega_{in_z}$	(4,86)	$C_{12}A_x^B\omega_{in_x}$	(4,87)	$C_{12}A_x^B\omega_{in_y}$
(4,88)	$C_{13}A_x^B\omega_{in_y}$	(4,89)	$C_{13}A_x^B\omega_{in_x}$	(4,90)	$C_{13}A_x^B\omega_{in_z}$
(4,91)	$C_{13}A_y^B\omega_{in_x}$	(4,92)	$C_{13}A_y^B\omega_{in_y}$	(4,93)	$C_{13}A_y^B\omega_{in_z}$
(5,76)	$C_{21}A_y^B\omega_{in_z}$	(5,77)	$C_{21}A_y^B\omega_{in_y}$	(5,78)	$C_{21}A_y^B\omega_{in_x}$
(5,79)	$C_{21}A_z^B\omega_{in_y}$	(5,80)	$C_{21}A_z^B\omega_{in_z}$	(5,81)	$C_{21}A_z^B\omega_{in_x}$
(5,82)	$C_{22}A_z^B\omega_{in_x}$	(5,83)	$C_{22}A_z^B\omega_{in_z}$	(5,84)	$C_{22}A_z^B\omega_{in_y}$
(5,85)	$C_{22}A_x^B\omega_{in_z}$	(5,86)	$C_{22}A_x^B\omega_{in_x}$	(5,87)	$C_{22}A_x^B\omega_{in_y}$
(5,88)	$C_{23}A_x^B\omega_{in_y}$	(5,89)	$C_{23}A_x^B\omega_{in_x}$	(5,90)	$C_{23}A_x^B\omega_{in_z}$
(5,91)	$C_{23}A_y^B\omega_{in_x}$	(5,92)	$C_{23}A_y^B\omega_{in_y}$	(5,93)	$C_{23}A_y^B\omega_{in_z}$
(6,76)	$C_{31}A_y^B\omega_{in_z}$	(6,77)	$C_{31}A_y^B\omega_{in_y}$	(6,78)	$C_{31}A_y^B\omega_{in_x}$
(6,79)	$C_{31}A_z^B\omega_{in_y}$	(6,80)	$C_{31}A_z^B\omega_{in_z}$	(6,81)	$C_{31}A_z^B\omega_{in_x}$
(6,82)	$C_{32}A_z^B\omega_{in_x}$	(6,83)	$C_{32}A_z^B\omega_{in_z}$	(6,84)	$C_{32}A_z^B\omega_{in_y}$
(6,85)	$C_{32}A_x^B\omega_{in_z}$	(6,86)	$C_{32}A_x^B\omega_{in_x}$	(6,87)	$C_{32}A_x^B\omega_{in_y}$
(6,88)	$C_{33}A_x^B\omega_{in_y}$	(6,89)	$C_{33}A_x^B\omega_{in_x}$	(6,90)	$C_{33}A_x^B\omega_{in_z}$
(6,91)	$C_{33}A_y^B\omega_{in_x}$	(6,92)	$C_{33}A_y^B\omega_{in_y}$	(6,93)	$C_{33}A_y^B\omega_{in_z}$

Table B.7 Elements of the Dynamics Submatrix $\mathbf{F}_{22}$

Element	Term	Element	Term	Element	Term
(14,14)	$-\beta_{b_{x_c}}$	(15,15)	$-\beta_{b_{y_c}}$	(16,16)	$-\beta_{b_{z_c}}$
(17,17)	$-\beta_{\nabla_{x_c}}$	(18,18)	$-\beta_{\nabla_{y_c}}$	(19,19)	$-\beta_{\nabla_{z_c}}$
(20,20)	$-\beta_{\delta g_x}$	(21,21)	$-\beta_{\delta g_y}$	(22,22)	$-\beta_{\delta g_z}$
(23,23)	$-\beta_{\delta h_c}$				

$\beta_{b_{x_c,y_c,z_c}}$ = Gyro inverse correlation time constants (5 min)
$\beta_{\nabla_{x_c,y_c,z_c}}$ = Accelerometer inverse correlation time constants (5 min)
$\beta_{\delta_{g_{x,y,z}}}$ = Gravity vector error inverse correlation time constants (V/20NM)
$\beta_{\delta h_c}$ = Barometer inverse correlation time (10 min)

Table B.8 Elements of the Dynamics Submatrix $\mathbf{F}_{55}$

Element	Term	Element	Term	Element	Term
(70,70)	$-\beta_{\nabla_{x_q}}$	(71,71)	$-\beta_{\nabla_{y_q}}$	(72,72)	$-\beta_{\nabla_{z_q}}$
(73,73)	$-\beta_{b_{x_q}}$	(74,74)	$-\beta_{b_{y_q}}$	(75,75)	$-\beta_{b_{z_q}}$

$\beta_{\nabla_{x_q,y_q,z_q}}$ = Accelerometer bias thermal transient inverse time constants (1 min)
$\beta_{b_{x_q,y_q,z_q}}$ = Gyro drift rate bias thermal transient inverse time constants (1 min)

Table B.9 Non-zero Elements of Process Noise Submatrix $\mathbf{Q}_{11}$

Element	Term	Element	Term
(4,4)	$\sigma^2_{\eta_{b_x}}$	(5,5)	$\sigma^2_{\eta_{b_y}}$
(6,6)	$\sigma^2_{\eta_{b_z}}$	(7,7)	$\sigma^2_{\eta_{A_x}}$
(8,8)	$\sigma^2_{\eta_{A_y}}$	(9,9)	$\sigma^2_{\eta_{A_z}}$

Table B.10 Non-zero Elements of Process Noise Submatrix $\mathbf{Q}_{22}$

Element	Term	Element	Term
(14,14)	$2\beta_{b_{x_c}}\sigma^2_{b_{x_c}}$	(15,15)	$2\beta_{b_{y_c}}\sigma^2_{b_{y_c}}$
(16,16)	$2\beta_{b_{z_c}}\sigma^2_{b_{z_c}}$	(17,17)	$2\beta_{\nabla_{x_c}}\sigma^2_{\nabla_{x_c}}$
(18,18)	$2\beta_{\nabla_{y_c}}\sigma^2_{\nabla_{y_c}}$	(18,18)	$2\beta_{\nabla_{z_c}}\sigma^2_{\nabla_{z_c}}$
(20,20)	$2\beta_{\delta g_x}\sigma^2_{\delta g_x}$	(21,21)	$2\beta_{\delta g_y}\sigma^2_{\delta g_y}$
(22,22)	$2\beta_{\delta g_z}\sigma^2_{\delta g_z}$	(23,23)	$2\beta_{\delta h_c}\sigma^2_{\delta h_c}$

$\sigma^2_{\eta_{b_{x,y,z}}}$ = PSD value of gyro drift rate white noise (6.25e-10$\frac{deg^2}{sec^3}$)
$\sigma^2_{\eta_{A_{x,y,z}}}$ = PSD value of accelerometer white noise (1.037e-7$\frac{ft^2}{sec^3}$)
$\sigma^2_{b_{x_c,y_c,z_c}}$ = Variances of gyro drift correlated noise (3.086e-13$\frac{deg^2}{sec^2}$)
$\sigma^2_{\nabla_{x_c,y_c,z_c}}$ = Variances of accelerometer correlated noise (4.147e-9$\frac{ft^2}{sec^4}$)
$\sigma^2_{\delta_{g_{x,y,z}}}$ = Variances of gravity vector error component correlated noise (1.93e-6deg^2)
$\sigma^2_{\delta h_c}$ = Variance of barometer correlated noise (10000ft^2)

Appendix C. Filter Tuning Values

This appendix contains a listing of the dynamics driving noise strength values used to tune the Kalman filters for each navigation component case shown in Table C.1. "Tuning", or purposeful addition of dynamics driving noise, is done to account for the modelling inadequacies introduced by using a small number of states (13 in this study) to represent a high-order truth model (95 or 96 states in this study), which in turn uses (usually much smaller) values of dynamics driving noise strengths to account for its reduction from the infinite-state real-world model. The 95-state truth model developed by Litton for the LN-93 INS [17] is accompanied by the dynamics driving noise strength values shown in the variable definitions below Table B.10.

For the 13 states in the filter model used in this application (see Table A.9), tuning was accomplished by experimentally adjusting the Q values shown in Table C.2 for each of the four navigation component cases. The 13 Q rows of Table C.2 correspond directly to the 13 states of the filter model as defined in Table A.9.

Table C.1 Navigation Component Cases

Case	INS Type	GPS Type	Altimeter Aiding
1	0.4nm/hr	DGPS and Pseudolite	Baro and Radar Alt
2	0.4nm/hr	DGPS	Baro and Radar Alt
3	0.4nm/hr	DGPS	Baro
4	4.0nm/hr	DGPS	Baro

Table C.2 Tuning Values for Filter States

State No.	Nav Case 1 Values	Nav Case 2 Values	Nav Case 3 Values	Nav Case 4 Values
1	1.2e-16	4.0e-15	4.0e-16	1.2e-23
2	1.5e-16	3.0e-15	3.0e-16	1.5e-23
3	0.0	0.0	0.0	0.0
4	3.81e-12	1.43e-12	1.43e-12	5.71e-9
5	5.33e-12	3.81e-12	3.81e-12	5.71e-9
6	1.62e-11	2.38e-12	2.38e-12	3.43e-10
7	1.03e-8	1.29e-5	1.29e-5	5.15e-5
8	1.03e-14	1.29e-5	1.29e-5	1.03e-8
9	3.4e-3	2.78e-3	2.78e-3	2.98e-3
10	16.0	15.0	15.0	24.0
11	6.67e3	6.67e3	6.67e3	3.33e4
12	0.2	7.5	7.5	0.2
13	5.0e-15	5.0e-10	5.0e-10	5.0e-16

Bibliography

1. Athans, M. and C. B. Chang. *Adaptive Estimation and Parameter Identification Using a Multiple Model Estimation Algorithm.* Technical Note 1976-28, Lexington, MA: Lincoln Lab., MIT, June 1976. ESD-TR-76-184.
2. Britting, Kenneth R. *Inertial Navigation Systems Analysis.* New York: Wiley-Interscience, 1971.
3. Britton, Ryan, L. *A Differential GPS-aided INS for Aircraft Landings.* MS Thesis, AFIT/GE/ENG/95D-03, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.
4. Chang, C. B., et al. *The Application of Adaptive Filtering Mothods to Maneuvering Trajectory Estimation.* Technical Report, Lexington, MA: Lincoln Lab., MIT, November 1975. TN-1975-59.
5. Chang, C. B., et al. "On the State and Parameter Estimation for Maneuvering Re-entry Vehicles," *IEEE Transactions on Automatic Control*, *AC–22*(1):99–105 (1977).
6. Chow, E. Y. and A. S. Willsky. "Issues in the Development of a General Design Algorithm for Reliable Failure Detection," *Proceedings of the IEEE Conf. Decision and Control, Albuquerque, NM*, 1006–1012 (December 1980).
7. 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).
8. Eide, Capt Peter Keith. *Implementation and Demonstration of a Multiple Model Adaptive Estimation Failure Detection System for the F-16.* MS Thesis, AFIT/GE/ENG/94D-06, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994.
9. Fitzgerald, R. J. "Divergence of the Kalman Filter," *IEEE Transactions on Automatic Control*, *AC–16*(6):736–743 (1971).
10. Gray, Robert A. *An Integrated GPS/INS/BARO and Radar Altimeter System for Aircraft Precision Approach Landings.* MS Thesis, AFIT/GE/ENG/94D-13, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994.
11. Honeywell Military Avionics Division, Minneapolis, MN. *Honeywell AN/APN-194 Pulse Radar Altimeter System*, June 1989. Honeywell Technical Description.
12. Huangqi, Sun et al. "An Investigation of Airborne GPS/INS for High Accuracy Position and Velocity Determination," *Proceedings of the Institute of Navigation 1994 National Technical Meeting*, 801–809 (1994). San Diego, CA.

13. Kalaith, T. "The Innovations Approach to Detection and Estimation Theory," *Proceedings of the IEEE*, *58*:680–695 (1970).

14. Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Series D: Journal of Basic Engineering*, *82*:34–45 (1960).

15. Kalman, R. E. and R. S. Bucy. "New Results in Linear Filtering and Prediction Theory," *Transactions of the ASME, Series D: Journal of Basic Engineering*, *83*:95–108 (1961).

16. Knudsen, L. *Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation System Inertial Navigation Unit*. Technical Report, 5500 Canoga Avenue, Woodland Hills, California 91365: Litton Guidance and Control Systems, January 1985. DID No. Di-S-21433 B/T: CDRL No. 1002.

17. Knudsen, L. *Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation Unit*. Technical Report, January 1985. DID No. DI-S-21433 B/T:CDRL No. 1002.

18. Lewantowicz, Z. H. and D. W. Keen. "Graceful Degradation of GPS/INS Performance with Fewer Than Four Satellites," *The Institute of Navigation, National Technical Meeting*, 269–275 (January 1991).

19. Lewis, Robert W. *Multiple Model Adaptive Estimation and Control Redistribution for the Vista F-16*. MS Thesis, AFIT/GE/ENG/96D-11, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1996.

20. Magill, D. T. "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Transactions on Automatic Control*, *AC–10*(5):434–439 (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.

24. Mehra, R. K. "Identification in Control and Econometrics: Similarities and Differences," *Ann. Econ. Soc. Measurement*, *3*:21–48 (1974).

25. Mehra, R. K. and J. Peschon. "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems," *Automatica*, 7:637–640 (1971).

26. Mosle, William B. *Detection, Isolation, and Recovery of Failures in an Integrated Navigation System*. MS Thesis, AFIT/GE/ENG/93D-28, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1993.

27. Musick, Stanton H. *PROFGEN – A Computer Program for Generating Flight Profiles*. Technical Report, Wright-Patterson AFB, OH: Air Force Avionics Laboratory, November 1976. AFAL-TR-76-247.

28. Musick, Stanton H. *MSOFE – Multimode Simulation for Optimal Filter Evaluation*. Technical Report, Wright-Patterson AFB, OH: Air Force Avionics Laboratory, October 1980. AFWAL-TR-88-1138.

29. Musick, Stanton H. and Neil Carlson. *User's Manual for a Multimode Simulation for Optimal Filter Evaluation (MSOFE)*. Air Force Avionics Laboratory, Wright-Patterson AFB, OH, April 1990. AFWAL-TR-88-1136.

30. Negast, William J. *Incorporation of Differential Global Positioning System Measurements Using an Extended Kalman Filter for Improved Reference System Performance*. MS Thesis, AFIT/GE/ENG/91D-41, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1991.

31. Nielsen, Robert L. *Development of a Performance Evaluation Tool (MMSOFE) for Detection of Failures with Multiple Model Adaptive Estimation (MMAE)*. MS Thesis, AFIT/GE/ENG/93S-37, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1993.

32. Nielsen, Robert L. and Stanton H. Musick. *MMSOFE – Multiple Model Simulation for Optimal Filter Evaluation*. Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1993. User's Manual.

33. Paielli, Russell et al. "Carrier-Phase Differential GPS for Approach and Landing: Algorithms and Preliminary Results," *Proceedings of the ION GPS-93 Sixth International Technical Meeting of the Satellite Division*, 22–24 (September 1993).

34. Riggins, Robert N., Assistant Professor of Electrical Engineering. Air Force Institute of Technology, Wright-Patterson AFB, OH, 1996. Course Notes - EENG 644, Navigation Systems Laboratory.

35. Riggins, Robert N. Jr. *Detection and Isolation of Plant Failures in Dynamic Systems*. PhD dissertation, University of Michigan, 1991.

36. Ritland, John T. "Impact of Inertial System Quality on GPS-Inertial Performance in a Jamming Environment," *Proceedings of the AIAA*, 1459–1467 (1987).

37. Rowson, Stephen V. et al. "Performance of Cattegory IIIB Automatic Landings Using C/A Code Tracking Differential GPS," *Proceedings of the Institute of Navigation 1994 National Technical Meeting*, 759–767 (1994). San Diego, CA.

38. Stepaniak, Michael J. *Multiple Model Adaptive Control of the Vista F-16*. MS Thesis, AFIT/GE/ENG/95D-26, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995.

39. Sun Microsystems, Inc. *Sun FORTRAN* (Sun release 4.1 Edition), December 1990.

40. Van Trees, H. L. *Detection, Estimation and Modulation Theory.* New York: Wiley and Sons, 1968.

41. Vasquez, Juan and Riggins Bob. "Detection Spoofing, Jamming or Failure of GPS." *Proceedings of the Annual Meeting - Institute of Navigation 1993.* 447–451. 1993.

42. Vasquez, Juan R. *Detection of Spoofing, Jamming, or Failure of a Global Positioning System (GPS).* MS Thesis, AFIT/GE/ENG/92D-37, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1992.

43. Westerlund, T. and A. Tysso. "Remarks on "Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems"," *IEEE Transactions on Automatic Control*, *AC–25*(5):1011–1012 (1980).

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

45. Willsky, A. S. and H. Jones. *A Generalized Likelihood Ratio Approach to State Estimation in Linear Systems Subject to Abrupt Changes.* Technical Report, Reading, MA: The Analytic Sciences Corporation, 1976. Under USAF Contract No. F04701-74-C-0095.

46. Wullschleger, Victor et al. "FAA/Wilcox Flight Test Results of DGPS System for Precision Approach," *Proceedings of the Institute of Navigation 49th Annual Meeting*, 111–118 (1994). Cambridge, MA.

Vita

Nathan Alan White [redacted]. He was fortunate to live happily in the small town of Providence until graduating from Mountain Crest High School in 1989. After two years of church service in Tacoma Washington he returned to Utah State University to pursue a degree in Electrical Engineering. In December 1994 he was sealed to Miriam Jones in the Logan LDS Temple for time and all eternity. Then in June, 1995 he received a Bachelor of Science Degree in Electrical Engineering from Utah State University and was commissioned into the United States Air Force. Through the direct accession program, he reported immediately to the Air Force Institute of Technology to pursue a Master's Degree. The arrival of the White's first child, Abraham Douglas, [redacted] was the highlight of their stay in Ohio. In December 1996 he graduated from the Air Force Institute of Technology and received a Master of Science Degree in Electrical Engineering with emphasis in navigation and control systems. Following graduation, he will be assigned to the Electro-Optics Warfare Branch of Wright Laboratories, Wright-Patterson AFB, Ohio.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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

4. TITLE AND SUBTITLE

MMAE DETECTION OF INTERFERENCE/JAMMING AND SPOOFING IN A DGPS-AIDED INERTIAL SYSTEM

5. FUNDING NUMBERS

6. AUTHOR(S)

Nathan A. White
Second Lieutenant, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/96D-21

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

Captain Anthony Romano
WL/AACN Bldg 620
2241 Avionics Circle
Wright-Patterson AFB, OH 45433-7304

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Previous research at AFIT has resulted in the development of a DGPS-aided INS-based precision landing system (PLS) capable of meeting the FAA precision requirements for instrument landings. The susceptibility of DGPS transmissions to interference/jamming and spoofing must be addressed before DGPS may be used in such a safety-of-flight critical role. This thesis applies multiple model adaptive estimation (MMAE) techniques to the problem of detecting and identifying interference/jamming and spoofing failures in the DGPS signal. Such an MMAE is composed of a bank of parallel filters, each hypothesizing a different failure status, along with an evaluation of the current probability of each hypothesis being correct, to form a probability-weighted average output. Performance for a representative selection of navigation component cases is examined.

For interference/jamming failures represented as increased measurement noise variance, results show that, because of the good FDI performance using MMAE, the blended navigation performance is essentially that of a single extended Kalman filter artificially informed of the actual interference noise variance. Standard MMAE is completely unable to detect spoofing failures (modelled as a bias or ramp offset signal directly added to the measurement). This thesis shows the development of a moving-bank pseudo-residual MMAE (PRMMAE) to detect and identify spoofing failures. Using the PRMMAE algorithm, the resulting navigation performance is equivalent to that of an extended Kalman filter operating in a no-fail environment.

14. SUBJECT TERMS

Multiple Model Adaptive Estimation, MMAE, Kalman Filter, ILS, PLS, GPS, FDI, Precision Landing, Failure Detection

15. NUMBER OF PAGES

148

16. PRICE CODE

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

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.