A Fault Tolerant System for an Integrated Avionics Sensor Configuration

Alper K. Caglayan and Roy E. Lancraft

NASA

NASA Contractor Report 3834

A Fault Tolerant System for an Integrated Avionics Sensor Configuration

Alper K. Caglayan and Roy E. Lancraft

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

LIMITED DISTRIBUTION

This document will remain under
distribution limitation until
September 1985.

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1984

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A/C   Aircraft

ADC   Air Data Computer

ATOPS   Advanced Transport Operating Systems

B-frame   body frame

det   determinant

DME   Distance Measuring Equipment

E-frame   earth fixed rotating frame (Earth-frame)

EKF   Extended Kalman Filter

FDI   Failure Detection and Isolation

FINDS   Fault Inferring Nonlinear Detection System (computer program)

FTS   Fault Tolerant System

G-frame   geographic frame located at the runway

I-frame   earth centered nonrotating frame (Inertial-frame)

IAS   Indicated Airspeed

IMU   Inertial Measurement Unit

L-frame	vehicle carried (N,E,D) frame (Local Level frame)
ln	natural logarithm
ML	Maximum Likelihood
MLS	Microwave Landing System
MTBF	Mean Time between Failures
RA	Radar Altimeter
RSDIMU	Dual Fail-Operational Redundant Strapdown Inertial Measurement Unit
TSRV	Transport Systems Research Vehicle

LIST OF SYMBOLS

A	A/C discrete state transition matrix
A_c	A/C continuous state transition matrix
A_w	continuous state transition matrix for wind dynamics
A_x, A_y, A_z	body accelerometer sensors
Azm	MLS azimuth sensor
b	composite bias vector containing selected input and measurement sensor biases, $b = [b_u, b_y]$
b_a	accelerometer triad bias vector (m/sec^2)
b_{az}	MLS azimuth measurement bias (deg)
B_c	A/C continuous input matrix
B_G^c	continuous input correction matrix
B_r^c	A/C continuous input matrix for corrections due to the earth's rotation
b_{el}	MLS elevation measurement bias (deg)
b_i	bias level for the i'th input sensor
B_i	i'th column of the input sensor transition matrix B

b_{ra} RA measurement bias (m)

b_{rn} MLS range measurement bias (m)

b_{sp} IAS measurement bias (m/sec)

b_u vector composed of accelerometer and rate gyro bias vectors, $b_u^{\cdot} = [b_a^{\cdot}, b_\omega^{\cdot}]$

$B(x)$ A/C discrete input transition matrix

b_y measurement sensor bias vector, $b_y = [b_{az}, b_{el}, b_{rn}, b_{sp}, b_\varphi, b_\theta, b_\psi]^{\cdot}$

b_o, b_θ, b_ψ IMU roll, pitch, and yaw measurement biases (deg)

b_ω rate gyro triad bias vector (deg/sec)

C_i measurement matrix relating i'th sensor failure to the residuals of the no-fail EKF

D measurement sensor bias measurement matrix

D_i i'th row of the measurement sensor bias matrix D

e vector composed of A/C Euler angles, $e = [o, \theta, \psi]^{\cdot}$ (deg)

E_c continuous process noise input matrix

El MLS elevation sensor

F linearized A/C state transition matrix

f_b true specific force (m/sec^2)

F_i	linearized A/C state transition matrix under the i'th failure hypothesis
f_u	expected failure level for an arbitrary input sensor
f_z	expected failure level for an arbitrary measurement sensor
g	earth's gravitational constant (m/sec^2)
g_I	inertial gravitational acceleration at A/C (m/sec^2)
G_i	i'th detector's Kalman filter gain
g_L	gravity field vector in L-frame (m/sec^2)
H	linearized measurement matrix
H_i	i'th failure hypothesis
H_o	no failure hypothesis
$h(x)$	nonlinear transformation relating A/C states to measurements
IAS	indicated airspeed sensor
K	no-fail EKF gain
K_b	EKF gain partition corresponding to bias estimates
k_d	sample time to start of current decision residual window (samples)
K_o	bias-free no-fail EKF gain

K_x	EKF gain partition corresponding to A/C state estimates
l_d	decision residual window length (samples)
l_e	detector (estimator) residual window length (samples)
l_h	healing test residual window length (samples)
M	maximum number of possible sensor failures, (total number of sensors used by the no-fail filter)
m_i	i'th sensor bias failure magnitude
n	discrete process noise vector, $n' = [n'_a, n'_\omega, n'_w]$
n_a	accelerometer noise vector (m/sec^2)
n_u	vector composed of accelerometer and rate gyro measurement noise vectors, $n'_u = [n'_a, n'_\omega]$
n_w	EKF process noise for horizontal winds (m/sec^2)
n_ω	rate gyro triad noise vector (deg/sec)
$P_b(0)$	bias vector initial condition variance
$P_b(k+1/k)$	EKF bias prediction error covariance
$P_b(k/k)$	EKF bias estimation error covariance
P_{H_i}	a priori probability of i'th hypothesis being true
$P_{H_{ij}}$	a priori probability of the i and j sensors failing simultaneously

$P(H_i|Y(k))$	a posteriori probability of the i'th hypothesis conditioned on the residual window $Y(K)$
P_i	i'th detector's bias jump estimation error covariance
$P_o(k+1/k)$	bias-free EKF prediction error covariance
$P_o(k/k)$	bias-free EKF estimation error covariance
p,q,r	true (absolute) roll, pitch, yaw body rates (deg/sec)
P,Q,R	body rate gyro sensors
$P_x(0)$	A/C state initial condition variance
$P_x(k+1/k)$	EKF state prediction error covariance
$P_x(k/k)$	EKF state estimation error covariance
$P_{xb}(k+1/k)$	EKF state and bias prediction error cross covariance
$P_{xb}(k/k)$	EKF state and bias estimation error cross covariance
Q_a	accelerometer measurement noise variance matrix $((m/sec^2)^2/sec)$
Q_w	wind process noise variance matrix $((m/sec)^2/sec)$
Q_ω	rate gyro measurement noise variance matrix $((deg/sec)^2/sec)$
r	EKF measurement residual vector
$\tilde{r}$	innovations sequence if no sensor failures are present

R	constant EKF measurement noise variance matrix for single replication of measurement sensors
$\tilde{R}$	innovation's covariance of $\tilde{r}$
r_{az}	A/C range from the azimuth antenna (m)
r_e	expanded residual vector of the EKF
r_{el}	A/C range from the elevation antenna (m)
r_G	A/C G-frame position vector expressed in G-frame, $r_G = [X_{rw}, Y_{rw}, Z_{rw}]'$ (m)
r_i	residual vector of the i'th detector
$R(k+1)$	EKF innovations variance for the expanded residuals sequence $r_e(k+1)$
Rng	MLS range sensor
$R_o(k+1)$	EKF innovations variance for the measurement residuals sequence $r(k+1)$
(r_x, r_y, r_z)	A/C position vector relative to the G-frame expressed in the G-frame (m)
T_{BI}	transformation matrix from I into B-frame
T_{GL}	transformation matrix from L into G-frame
T_{LB}	transformation matrix from B into L-frame
T_{LG}	transformation matrix from G into L-frame

u	vector composed of accelerometer and rate gyro measurements, $u' = [u_a', u_\omega']$
$\bar{u}$	an active set of the replicated input sensor measurements used by the no-fail filter
u_a	accelerometer triad measurement vector (m/sec^2)
u_g	correction to no-fail inputs, due to earth's rotation and gravity
u_i	i'th element of the input sensor measurement vector
u_i^t	"true" value for the i'th input sensor signal
U_i	bias failure propagation matrix for the i'th hypothesis
u_r^c	correction to no-fail inputs, due to earth's rotation and gravity expressed in continuous time
u_ω	rate gyro triad measurement vector (deg/sec)
v	measurement noise vector, $v_y = [v_{az}, b_{el}, b_{rn}, b_{sp}, b_\Phi, b_\theta, b_\psi]'$
V	bias correction matrix
v_{az}	MLS azimuth measurement noise (deg)
v_{el}	MLS elevation measurement noise (deg)
v_i	measurement noise for the i'th input sensor
V_i	bias failure correction matrix for the i'th failure hypothesis
v_{ra}	RA measurement noise (m)

v_{rn} — MLS range measurement noise (m)

v_{sp} — IAS measurement noise (m/sec)

v_ϕ, v_θ, v_ψ — IMU roll, pitch, and yaw measurement noises (deg)

w — vector composed of horizontal wind components, $w = [w_x, w_y]'$ (m/sec)

w_x, w_y — horizontal wind components in G-frame (m/sec)

x — A/C state consisting of r_G, $\dot{r}_G$, e, w, vectors

(x_E, y_E, z_E) — elevation antenna location in the G-frame (m)

$\tilde{x}_i$ — no-fail filter prediction error assuming the i'th failure

(x_M, y_M, z_M) — azimuth antenna location in the G-frame (m)

$\tilde{x}_o$ — no-fail filter prediction error assuming no failures

(X_{rw}, Y_{rw}, Z_{rw}) — A/C position vector relative to the G-frame expressed in the G-frame (m)

y — measurement vector consisting of MLS, IAS, and IMU measurements, $y = [y_{az}, y_{el}, y_{rn}, y_{sp}, y_\phi, y_\theta, y_\psi]'$

$\bar{y}$ — average of the replicated measurement sensors used by the no-fail filter

y_{az} — MLS azimuth measurement (deg)

y_{el} — MLS elevation measurement (deg)

y_{ra} — RA measurement (m)

y_{rn}	MLS range measurement (m)
y_{sp}	IAS measurement (m/sec)
y_ϕ, y_θ, y_ψ	IMU roll, pitch, and yaw attitude measurement (deg)
$Y(K)$	set of measurements over the window K (i.e. from $k=k_d$ to $k=k_d+l_h$)
Z_i	an arbitrary measurement sensor's output

Greek Symbols

β_u	expected normal bias level for an arbitrary input sensor
β_z	expected normal bias level for an arbitrary measurement sensor
γ	confidence threshold for healing tests
Γ_ω	transformation matrix relating body to Euler angle rates
$\Delta@$	difference between two replications or estimates of "@" (where "@" is an arbitrary variable designation)
λ	geographic latitude (deg)
Λ_i	likelihood ratio for i'th hypothesis
Λ_{ij}	likelihood ratio for the dual failure hypothesis for i and j sensors
ρ_G	A/C I-frame position vector expressed in G-frame (m)
σ_u^2	variance of an arbitrary element of the input sensor noise vector, v_i

	sampling period (sec)
$\phi, \theta, \psi,$	A/C Euler angles (deg)
ω_B	vector composed of absolute body rates, $\omega_B = [p,q,r]'$ (deg/sec)
ω^E	earth's rotation rate (rad/sec)
ω_G	earth's inertial angular velocity vector expressed in G-frame (rad/sec)
Ω_G	skew symmetric form of ω_G (rad/sec)

Subscripts

b	bias
B	coordinization in the B-frame
E	coordinization in the E-frame
G	coordinization in the G-frame
i	i'th failure
I	coordinization in the I-frame
L	coordinization in the L-frame
o	no-fail
x	A/C state
xb	A/C state and normal operating bias

Superscripts

$\cdot$	time derivative
$\wedge$	least mean square estimate
$-$	average of replicated measurements
i	i'th replication
	transpose of a matrix

1. INTRODUCTION

This study considers the design and digital simulation of an integrated fault tolerant system (FTS) using analytic redundancy for avionics sensors on the NASA-Langley Research Center Advanced Transport Operating Systems (ATOPS) Transport Systems Research Vehicle (TSRV) in a Microwave Landing System (MLS) environment. The overall objective of the fault tolerant system is to provide reliable estimates for aircraft position, velocity, and attitude in the presence of possible failures in ground-based navigation aids, and on-board flight control and navigation sensors.[1] The estimates, provided by the fault tolerant system, are used by a fully automated guidance and control system to land the aircraft along a prescribed path. Sensor failures are identified by utilizing the analytic relationship between the various sensor outputs arising from the aircraft equations of motion [1].

An aircraft sensor fault tolerant system design methodology is developed by formulating the problem in the context of simultaneous state estimation and failure identification in discrete time nonlinear stochastic systems. The resulting sensor fault tolerant system consists of 1) a no-fail estimator, implemented as an extended Kalman filter (EKF) based on the assumption of no failures, which provides estimates for aircraft state variables and normal operating sensor biases; 2) a bank of detectors which are first order filters for estimating bias jump failures in sensor outputs; 3) likelihood ratio computers; and 4) a decision function which selects the most likely

---

[1]Integrated FTS refers to the capability of handling all of these three different sensor subsets simultaneously in contrast to earlier studies in which only one subset such as flight control or navigation sensors are considered.

failure mode based on the likelihood ratios.

The operation of the fault tolerant system is as follows: First, the EKF computes estimates for aircraft position, velocity, attitude, horizontal winds, and normal operating sensor biases on the assumption of no sensor failures. The residuals of this EKF drive a bank of detectors, where each detector has been designed to estimate a postulated bias jump failure for a given sensor. Then, a multiple hypothesis testing procedure is employed to decide whether the EKF is operating with healthy sensors or under one of the hypothesized failed sensor modes. The multiple hypothesis test selects the most likely failure mode based on the likelihood ratios which are computed using the bias jump failure estimates from the detectors. When a failure is declared by the decision logic, the filter—detector structure is reconfigured by eliminating the failed sensor, making the appropriate changes in the no—fail filter and detectors, and reinitializing the likelihood ratios and a priori probabilities.

The no—fail filter is implemented in a rectangular coordinate system with origin on the runway by using a new separated bias EKF algorithm which has been obtained by extending the known results for the linear case to nonlinear systems. Body mounted accelerometers and rate gyros form the inputs into the EKF, while MLS range, azimuth, elevation measurements, IAS (indicated airspeed), and IMU (inertial measurement unit) attitude outputs are utilized as measurements by the EKF. Alternatively, an RSDIMU (dual fail—operational two—degree—of—freedom strapdown inertial measurement unit [8]) can be used instead of a platform IMU and the body mounted accelerometers and rate gyros. The function of the no—fail filter is similar to that of a navigator coordinatized in a local runway frame of reference. Whereas

traditional navigation equations usually involve open loop integration of the body accelerations in the runway frame with occasional position and velocity fixes, the no-fail EKF in our study performs the position, velocity, and attitude updates continuously in a closed loop fashion.

The proposed filter-detector structure is computationally feasible. The integrated sensor FTS design requires a single high order EKF (no-fail filter). The state estimation and failure detection performance of the developed sensor fault tolerant system is analyzed by using a nonlinear six-degree-of-freedom simulation of the TSRV aircraft.

In this report, we will discuss the failure detection and isolation (FDI) performance of the system using a dual-redundant sensor configuration. In particular the system will be shown capable of detecting failures even if only one sensor of a given type remains.

The sensor fault tolerant algorithm developed here has been incorporated into a computer program called FINDS (Fault Inferring Nonlinear Detection System) which is described in detail in [2]. The simulation portion of the software is essentially an integration of the NASA-LRC supplied TSRV and RSDIMU computer simulation programs. Aircraft sensor models have been developed and appended into the simulation to provide more realistic normal operating errors. Furthermore, sensor failure models for increased bias, hardover, null, scale factor, ramp, and increased noise type sensor malfunctions have also been assimilated into the software.

The simulation results indicate that the no-fail EKF estimation errors compare favorably to those obtained with other types of navigation filters employed in the same MLS environment. Sensor failure detection performance of the fault tolerant system is excellent for the EKF measurement sensors such as MLS, IAS, and IMU, while the failure detection speed for the EKF input sensors such as accelerometers and rate gyros has been found to be slower than that of measurement sensors.

1.1 Relation to Previous Work

Here, we will discuss the differences between the major aspects of FINDS and earlier sensor failure detection studies such as the nonlinear multiple hypothesis testing approach, reported in [20]-[21], F-8 DFBW [25], DIGITAC A7 [24], and the RSDIMU [23] FDI studies.

o **sensor complement**

FINDS is an integrated FTS in the sense that failures in on-board flight control as well as inertial sensors and ground-based navigation-aid instruments are considered. For instance, FINDS can detect not only a fault in an on-board MLS receiver, but also a fault in the ground-based transmitting antenna for that receiver. FINDS can operate without any hardware redundancy in that it can detect failures even if there remains only one sensor of a given type.

In contrast, earlier studies are concerned with only a single subset of the sensor complement considered here. For instance, the F-8 and DIGITAC A-7 studies deal with flight control sensors only and the RSDIMU FDI considers failures only in inertial

sensors. F-8 FDI requires dual sensor redundancy so that, if only one sensor of a given type remains in the configuration, then the failure of that sensor cannot be detected.

o **FDI algorithm structure**

FINDS has a single large order estimator (no-fail filter) driven by all the sensor outputs. Failures are identified by analyzing the signature of sensor faults on the no-fail residuals by processing the residual sequence through a bank of first-order detectors. The estimator/detector structure in FINDS is an extension of the structure used in [27] to nonlinear dynamic systems.

In contrast, the nonlinear multiple hypothesis testing approach [21] requires the implementation of M+1 (where M is the total number of sensors) large order estimators (each of which has complexity equal to the no-fail filter in FINDS). In F-8 FDI, each sensor output is estimated by using a subset of the other available sensor types in order to have three voting sensors (2 hardware/1 analytically constructed). Hence, the number of filters (each with a different order depending on the analytic relationships used) are equal to the number of sensor types. In DIGITAC A7, several different filter assemblies of varying orders with comparators are used.

o **treatment of nonlinearities**

FINDS analyzes the residuals of a nonlinear no-fail filter by processing them through nonlinear detectors to find sensor faults. One advantage of using nonlinear filters is that the fault tolerant system is independent of the flight path so that it

5

does not need gain scheduling.

On the other hand, the nonlinear filter residuals in [21] are directly used in the likelihood ratio computations without any processing. Similarly, in F-8 FDI, some nonlinear filters are used but only for constructing sensor outputs.

o state estimation

FINDS supplies the vehicle state estimates used by the guidance and flight control algorithms. That is to say, the no-fail filter in our application would have been there even if there were no sensor fault monitoring system. Consequently, the no-fail filter should not be considered an additional requirement. Therefore, in comparing the complexity of FINDS with other FDI applications, only the bank of first-order detectors in FINDS should be .considered. For instance, the complexity of the bank of first-order detectors are roughly equivalent to the bank of filters in the F-8 FDI study.

o normal operating errors

In FINDS, important normal operating sensor biases are estimated in order to remove their false alarm effects. Sensor failures are modelled as bias jumps with infinite uncertainty whereas a sensor normal operating error is modeled by a constant random variable with a finite uncertainty (as defined by the sensor specifications). Hence, the no-fail filter attempts to distinguish between a normal operating sensor bias and a bias jump failure in that sensor. On the other hand, in F-8 FDI, normal operating sensor biases are considered only in the selection of decision thresholds.

6

In the RSDIMU work, the sensor operating errors (biases as well as scale factors and misalignment) are used in the selection of thresholds.

o information pattern

Finally, FINDS looks at all sensor outputs simultaneously in deciding a sensor fault, in contrast to other FDI studies in which only those sensors explicitly related to a specific sensor are used in deciding whether that sensor is at fault or not. For instance, in F-8 FDI study, a roll attitude sensor failure is decided by considering only the roll rate, pitch, and yaw attitude sensors while ignoring the analytic redundancy from the other sensors. On the other hand, FINDS looks at all of the rate attitude sensors as well as other dynamically coupled sensors in deciding a roll attitude sensor failure. Hence, FDI information contained in the dynamic redundancy of all sensor outputs are simultaneously used in FINDS. Of the other studies, only the detection and estimation algorithm of the multiple hypothesis testing approach would be more optimal (least mean square sense) than that employed in FINDS, but only at the expense of a severe computational burden.

The organization of the report is as follows. The developed fault tolerant system methodology is described in Chapter II. A tutorial description of each major block of the FTS is given in Section 2.1. Analytic description of the developed FTS is contained in Sections 2.2-2.6. This chapter ends with Section 2.7 where an illustrative example is given showing the exploited failure signature in the design. Chapter III examines the simulated performance of the developed FTS. Conclusions are presented in Chapter IV.

8

2. FAULT TOLERANT SYSTEM

In this chapter the analytical structure of the developed aircraft sensor fault tolerant navigation system will be discussed in detail.

The objective of the fault tolerant system is to provide reliable aircraft state estimates to an automated guidance and control system which accomplishes automatic landing in an MLS environment. The developed fault tolerant system can detect failures in navigation-aid instruments (e.g. on-board navaid receiver as well as ground-based navaid antenna failures), on-board inertial, and flight control sensors. Since the developed FTS uses the analytic redundancy between various sensor outputs arising from aircraft equations of motion, sensor failures can be detected even if there is only one sensor of a given type in the configuration. We envision the practical use of our developed FTS in a triple or dual redundant (or combinations thereof) sensor configuration. For instance, our FTS would improve the fail-op/fail-safe capability of a triple redundant voting system to at least a fail-op/fail-op/fail-safe capability.

Basically, the fault tolerant system consists of a navigation filter conditioned on the assumption of no failures, followed by a bank of low-order failure detectors and their companion decision and reconfiguration logic. The estimation, detection, decision and reconfiguration algorithms are derived by using nonlinear aircraft point mass equations of motion.

Although simpler linear filtering algorithms could have been used, the nonlinear

filtering algorithms used in our FTS have the advantage of being independent of landing path and selected trim conditions. In contrast, linear filtering algorithms[2] would necessitate the scheduling of gains.

The outline of Chapter II is as follows. An overall description of the fault-tolerant system is given in Section 2.1 by going over the operation of each major block. The aircraft point mass equations of motion and sensor dynamics, on which the filter-detector development is based, is then discussed in Section 2.2. Section 2.2 also outlines the operation of the no-fail filter. Failure detector implementation is discussed in Section 2.3, and in Section 2.4, the employed decision rule is explained. Tests for multiple simultaneous failures are discussed in Section 2.6. The next section, 2.5, describes the operation of healing tests. In Section 2.6, the reinitialization procedure is outlined. An example, designed to highlight the various failure signature information contained in the no-fail filter residuals, is given in Section 2.9.

2.1 Fault Tolerant System Overview

The design problem in our application can be broadly stated as follows: Given redundant discrete-time measurements of various navigation-aid and on-board flight control and inertial sensor measurements on an aircraft, generate estimates for the vehicle states required by the automatic guidance and control laws in the possible presence of failures in these sensors. The desired FTS qualities dictated by our

---

[2]Except, of course, if a constant gain linear navigation filter could be designed with satisfactory estimation performance.

application are the following:

o use inherent analytic redundancy arising from a knowledge of the aircraft dynamics so that hardware redundancy requirements are reduced for a given mission reliability

o fast detection of hard failures, and detection of mid— and soft—level failures before their effects on system performance become significant

o ability to handle different types of failures (i.e. hardover, null, increased inaccuracy, ramp, etc.

o acceptable false alarm/detection probability performance in the presence of colored measurement noises (since MLS sensor noises are time—correlated in our application and induce an unacceptably high false alarm rate if they are not compensated)

o distinguish between normal operating sensor errors, such as biases, and sensor failures (this issue is especially critical since most analytic FTS techniques model failures as bias jumps in sensor outputs)

o ability to recover from false alarms which occur during aircraft maneuvers due to misalignment and scale factor errors in body—mounted instruments

o feasible computational complexity enabling future on—board real—time implementation with appropriate modifications.

With these goals in mind, the aircraft sensor fault detection design problem was formulated in the context of simultaneous state estimation and failure detection in nonlinear discrete time stochastic systems. Figure 1 displays the major components of the resulting filter—detector structure. The major parts of this system are as follows:

o a nonlinear no—fail filter which estimates aircraft states and sensor biases assuming no sensor failures

o a bank of first—order detectors which estimate hypothesized sensor failure levels using the residuals of the no—fail filter as inputs

o likelihood ratio computers, driven by the detector outputs, which perform the necessary computations for the multiple failure hypotheses

o a decision rule which selects the most likely failure mode based on the likelihood ratios

11

REPLICATED AIRCRAFT SENSOR MEASUREMENTS

$y(k)$ $u(k)$

SUMMER LOGIC   SELECTION LOGIC

$\bar{y}(k)$ $\bar{u}(k)$

NO-FAIL FILTER $\hat{x}(k), \hat{b}(k)$ TO GUIDANCE CONTROL ALGORITHMS

$\hat{\bar{y}}(k)$

$y(k)$ → RESIDUALS COMPUTATION

$r_0(k), \tilde{R}^{-1}(k)$

DETECTOR 1   DETECTOR 2   ...   ...   DETECTOR M-1   DETECTOR M

P_{H_1} $r_1(k)$ $r_2(k)$ P_{H_2} P_{H_0} $P_{H_{M-1}}$ $r_{M-1}(k)$ $r_M(k)$ P_{H_M}

LIKELIHOOD RATIO FOR H_1 LIKELIHOOD RATIO FOR H_2 ...   LIKELIHOOD RATIO FOR H_0 ...   LIKELIHOOD RATIO FOR H_{M-1} LIKELIHOOD RATIO FOR H_M

$\Lambda_1(k)$ $\Lambda_2(k)$ $\Lambda_0(k)$ $\Lambda_{M-1}(k)$ $\Lambda_M(k)$

COSTS ⟹ DECISION LOGIC

H_i^*

FILTER-DETECTOR RE-CONFIGURATION LOGIC

FIG. 1.   FAULT TOLERANT SYSTEM STRUCTURE

12
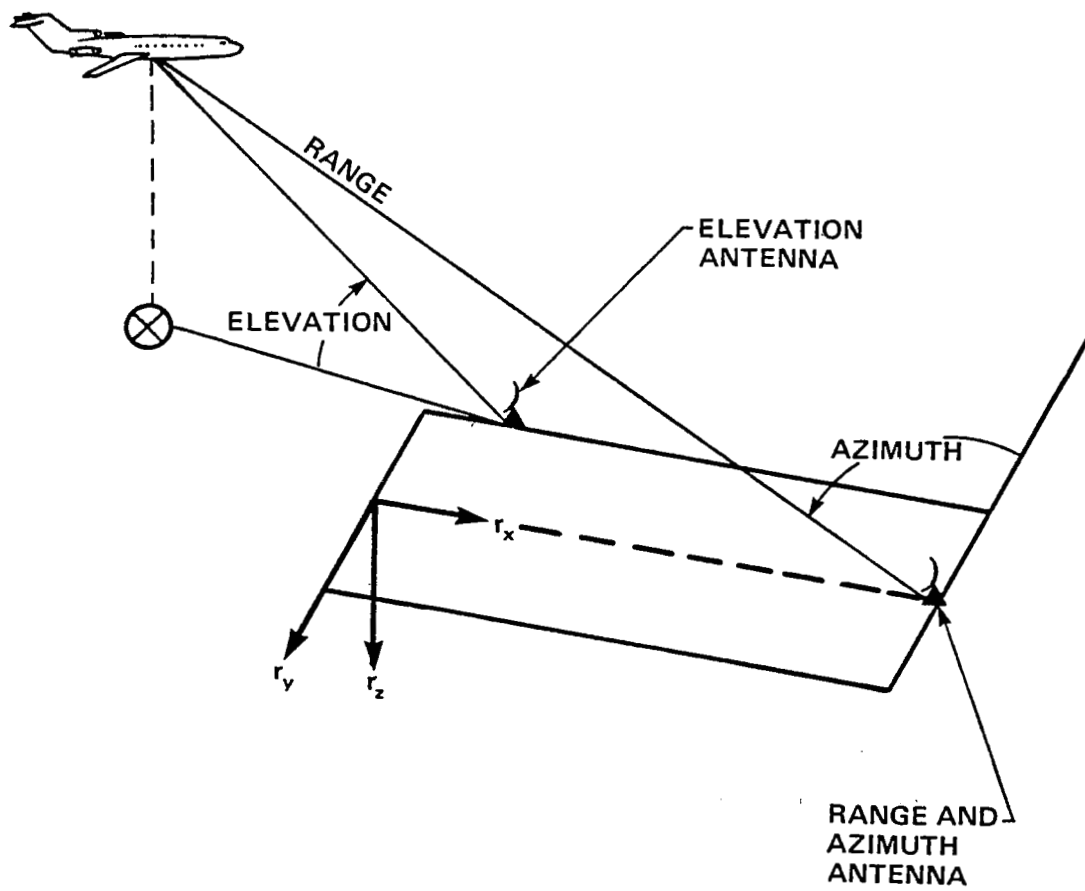
FIG. 2. MLS AND RUNWAY GEOMETRY

13

o   a   reconfiguration   module   which   performs   the   various   reinitialization
procedures after the detection of a failure

o   a   healing   test   module   which   monitors   the   failed   sensors   to   check   their
possible recovery.

The fault tolerant system is concerned with failures in the sensor configuration
consisting of:

o   body mounted accelerometers (A_x, A_y, A_z)

o   body mounted rate gyros (P,Q,R)

o   microwave landing system (MLS)

o   indicated airspeed (IAS)

o   IMU attitudes from a stabilized platform (ϕ, θ, ψ)

o   radar altimeter (RA)

The three body mounted accelerometers and rate gyros above are flight control
quality sensors, each of which is aligned along one of the body frame axes.   An
alternative sensor complement, containing a prototype dual—fail operational Redundant
Strapped—Down Inertial Measurement Unit (RSDIMU), is also considered.   In this sensor
configuration, body mounted accelerometers and angular rate gyros are replaced by
the navigation quality acceleration and rate measurements from the RSDIMU while the
RSDIMU attitude outputs replace the IMU Euler angle measurements.

The navigation aid is a ground—based Microwave Landing System (MLS) which
transmits position information to aircraft within its volumetric coverage at discrete
time intervals.   The MLS (see Figure 2) consists of a Distance Measuring Equipment
(DME) providing aircraft range information, an azimuth antenna co—located with the

DME provides the aircraft's angle relative to the runway, and an elevation antenna, located near the glide path intercept point provides the aircraft with its elevation angle relative to the local horizon.

The radar altimeter replaces the MLS elevation measurement when the aircraft is over the runway during which the elevation measurements are normally invalid. In the next six subsections, we will describe each major block of the fault tolerant system.

2.1.1 No-Fail Filter

The no-fail filter shown in Figure 1 is an extended Kalman Filter (EKF) [15] which is designed on the assumption of no failures. Although we have used an EKF in our study, any other nonlinear filter could have been used without significantly affecting the failure detection algorithms. We have chosen a nonlinear filtering formulation in order to have a flight path independent estimator. The EKF development is based on a discrete-time difference equation for the aircraft point mass equations of motion mechanized in a ground-based, flat earth Cartesian coordinate system with its origin located on the runway (Figure 3). This nonlinear, stochastic difference equation is obtained by transforming the specific force measured by the body mounted accelerometers into the runway frame, and integrating this expression along with the differential equations for the Euler angles over a fixed sampling interval.

The no-fail filter provides estimates for the aircraft states, $\hat{x}(k)$, which consist of aircraft position, velocity, attitude, and horizontal winds, and estimates for the "normal operating" biases, $\hat{b}(k)$, associated with a specified subset of the sensors.

15
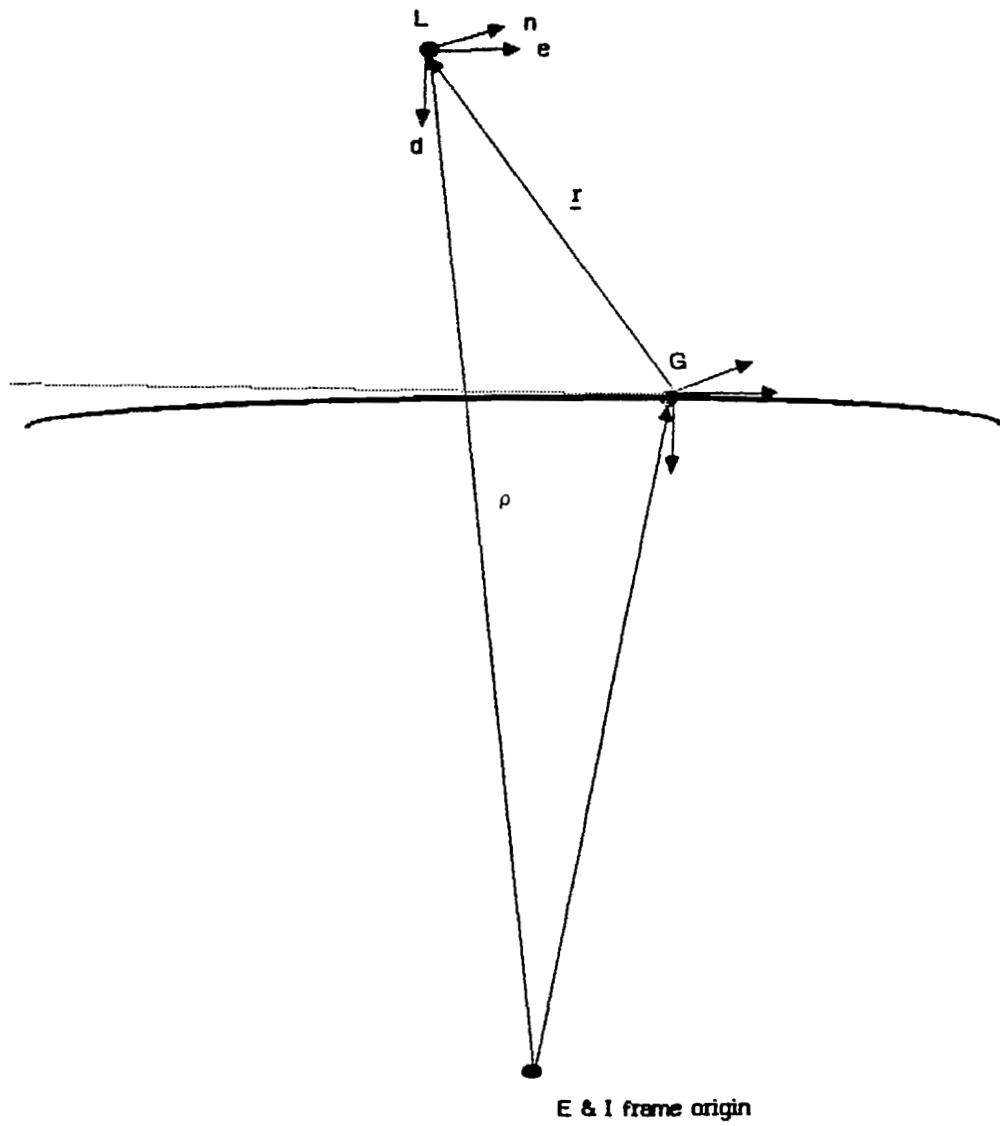
FIG. 3. REFERENCE FRAMES

16

State estimates provided by the no-fail filter are used by an automated guidance and control system to land the aircraft along a prescribed path [4]-[5],[16]-[17].

The no-fail filter functions essentially as a navigator in this system, estimating the state of the aircraft and the "normal operating" biases on selected sensors. However, unlike most navigators, this one continuously filters the navigation aid, IAS, and attitude measurements, so as to constantly correct the propagated state estimates. In addition, since the no-fail filter is based on the nonlinear aircraft equations of motion, it is independent of flight path and trim conditions and does not require any gain scheduling.

According to the manner processed by the no-fail filter, the replicated sensor set is divided into two groups: 1) no-fail filter input sensors, u(k), consisting of body acceleration and angular rate measurements; 2) no-fail filter measurement sensors, y(k), formed by the MLS, IAS, IMU, and RA outputs. The input sensor outputs are integrated in the no-fail filter, without any closed-loop filtering, after they are compensated by the "normal-operating" bias estimates. Only one set of the replicated input sensors, $\bar{u}(k)$, and the average of the replicated measurement sensors, $\bar{y}(k)$, are used by the no-fail filter after being processed in the "selection logic" and "summer logic" blocks. Replicated input measurements are kept as standby equipment. Thus, the filter size is kept to a minimum without a loss of generality.

We have employed a new separated EKF algorithm for the implementation of the no-fail filter [6]-[7]. The separated EKF algorithm provides a numerical decomposition procedure for obtaining the EKF filter gains. At each sampling instant, this algorithm

17

sequentially computes: 1) a bias−free gain; 2) a bias correction matrix; 3) a bias gain; and 4) a correction to the bias−free gain. The separated EKF also improves numerical accuracy since lower order matrices are used in the numerical decomposition, and finite variance for the plant state initial conditions, and infinite uncertainty in the a priori bias estimates are easily handled.

2.1.2 Detectors

Since the no−fail filter computes the residuals for the averaged measurement sensor outputs, $\bar{y}(k)$, the residual sequences for the individual measurement sensors, $y(k)$, need to be computed. This is accomplished in the "residuals computation" block by using the no−fail filter's estimate, $\hat{y}(k)$, for the measurement sensor outputs. The output of this block is the output measurement residual sequence, $r_o(k)$, which is the difference between the measurement sensor outputs and their corresponding predicted estimates provided by the no−fail filter. This residual sequence is the same one that would have been generated by an EKF formulated to use the unaveraged measurements, $y(k)$.

When the measurement noises are zero mean, white, and Gaussian, then the residual sequence, $r_o(k)$, of the no−fail filter −− in the absence of input or output sensor failures −− is approximately (exactly, in the linear case) a zero mean, white Gaussian sequence of random vectors. The no−fail filter was designed by making the above assumptions for the measurement noises. However, the MLS noise in our application is time correlated, rather than white. This necessitated the post−filtering of the MLS residuals to remove these correlations. This was accomplished by passing each MLS residual sequence through a first order filter also located in the "residuals

computation" block. The measurement residuals are then used by the bank of detectors.

The bank of detectors, which follow the residuals computation block, are a set of first order filters, each estimating the level of an hypothesized sensor failure. In the case of single sensor failures, the total number of detectors is equal to the sum of the number of input sensors and the number of measurement (replicated ones included) sensors. For instance, with dual sensor redundancy, there would be twenty of these first order detectors: three for the body mounted accelerometers, three for the body mounted rate gyros, six for the MLS range, azimuth and elevation measurements, two for the IAS outputs, and six for the IMU measurements.

Using the no-fail filter residuals as measurements, each detector estimates the failure level associated with that sensor. Failures are modelled as bias jumps in the measurement equations. Failure bias jumps are assumed to be zero mean random variables with infinite covariance (equivalently zero information). In the linear case, bias type sensor failures manifest themselves in an additive fashion onto the no-fail filter residuals. For the nonlinear problem considered, similar relations have been derived by making suitable approximations.

Each detector puts out a compensated residual sequence, $\{r_1(k), r_2(k), ..., r_M(k)\}$, such that the effects of the hypothesized sensor failure are removed from the no-fail filter residuals by processing the estimated sensor failure level. Detectors operate over a "window" of the residuals, with the initial failure level estimates and uncertainties reset at the beginning of each residual window. Each detector estimates

the level of a bias jump in the associated sensor output which is hypothesized to occur at the beginning of the corresponding window. The start of a new window determines the hypothesized time of failure, and the maximum length of the window determines the time to wait before initiating a new hypothesis.

Figure 4 shows the synchronization of these various residual windows for a typical run. In this run, the decision window length is 1 second. Estimation window lengths for input and measurement sensors are 3 and 1 seconds, respectively. The length of the healer window is 3 seconds. At 6.4 seconds, due to a sensor failure detection decision, all of the residual windows are restarted. Estimation and healing (discussed in 2.1.6) residual window lengths are constrained to be integer multiples of the decision residual window length.

The choice of residual window lengths is based on the sensor type, the expected failure level (hard, mid, soft), the specified probability of false alarm, and the desired detection speed. Since the detectors keep track of how each hypothesized sensor failure propagates through the no-fail filter dynamics to affect the no-fail filter residuals, the sensor type definitely plays an important role in the determination of residual window lengths. For instance, we have chosen the residual window length for input sensors to be three times the length for measurement sensors in our application. Finally, the residual windows should be large enough to produce a tolerable probability of false alarm rate and small enough to permit rapid detection of sensor failures.

In summary, the detector block consists of a bank of first order estimators

20

healing test windows

input sensor
estimation windows

measurement sensor
estimation windows

decision
windows

TIHE (seconds)
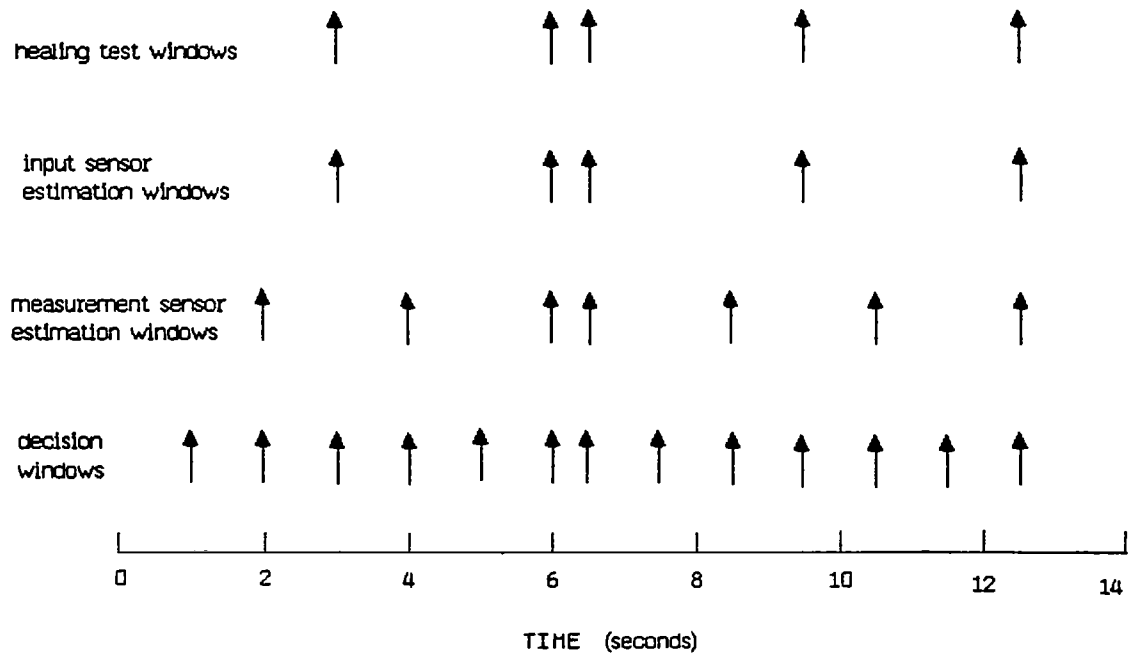
FIG. 4. SYNCHRONIZATION OF RESIDUAL WINDOWS

21

driven by the expanded innovations of the no-fail filter. Each detector corresponds to a different sensor failure hypothesis, and, corresponding to each detector, there is an associated residual data window length. The bias jump magnitude for a given sensor failure, hypothesized to happen at the start of the residual window, is estimated by the detector corresponding to that sensor. The residuals of the detectors along with the residual of the no-fail filter are used in the likelihood ratio (LR) computations which are discussed in the next section.

2.1.3 Likelihood Ratio Computations

As seen in Figure 1, each compensated residual sequence, $r_i(k)$, is used in the computation of the likelihood ratio, $\Lambda_i(k)$, for hypothesis H_i corresponding to the i'th sensor failure. Likelihood ratio computations are also based on a fixed window of the residuals. The length of this residual window for the LR computations is the same for every hypothesis. However, the length of this decision residual window is, in general, different from that of the detector estimation residual windows described in the previous section. The likelihood ratio, for a particular hypothesis, H_i, is proportional to the a posteriori probability (conditioned on the residuals in the decision window) that the compensated residuals model (used by the LR) corresponds to the "best" hypothesis.

Each likelihood ratio is initialized with the a priori probability P_{H_i}, of that hypothesis. A priori probabilities are determined from known sensor failure rates and modified according to the expected estimation degradation due to modelling errors. Each likelihood ratio is a function of a sum of residual quadratic forms weighted by the residuals' statistics. Likelihood ratios are used by the decision rule which is

22

discussed next.

2.1.4 Decision Rule

The decision rule selects the most likely sensor failure based on an M–ary hypothesis testing procedure. This test minimizes the Bayes risk which is a weighted average of making incorrect decisions. These weightings are shown as the input "costs" in Figure 1. If it is assumed that costs associated with making incorrect decisions (selecting hypothesis H_i when H_j is true) are all equal and those of making correct decisions (selecting hypothesis H_i when H_i is true) are all zero, then the M–ary decision rule is equivalent to choosing hypothesis H_i corresponding to the largest a posteriori probability. The decision logic provides the output of the M–ary hypothesis test indicating whether the no–fail filter is operating under no failures (hypothesis H_o), or under the i'th sensor bias jump failure (H_i).

2.1.5 Reconfiguration Logic

Once a failure decision is reached, the necessary filter/detector changes are made in the reconfiguration block. For input sensor failures, this process includes removing the faulty sensor from the no–fail filter inputs and replacing it with a redundant one of the same type. If there are no more healthy sensors of that type left in the stand–by queue, then the no–fail filter is restructured to reflect the loss of that sensor type input, provided that the filter is capable of operating without it. Similarly, if a measurement sensor fails, then the faulty sensor is removed from the corresponding average and the appropriate changes in the no–fail filter statistics are made. Again, when no sensor of a given type remains, then the no–fail filter structure is collapsed to accommodate the loss of that type sensor measurement.

The next function of the reconfiguration block is to reinitialize the no—fail filter, detectors, and the likelihood ratio computers following the identification of a failure. The reinitialization of the no—fail filter is necessary since undetected sensor failures propagate through the no—fail filter dynamics to corrupt the state and bias estimates. The reinitialization of the no—fail filter is performed by increasing the estimation error covariance by an amount reflecting the effect of uncertainty caused by the identified failure. This incremental covariance is a function of the sensor type, the sensor failure level estimate, and the elapsed time since the hypothesized failure onset time.

For instance, if a body mounted normal accelerometer failure is detected, then the incremental covariance would principally involve terms related to altitude and normal velocity. The estimates of the no—fail filter are not reinitialized directly in order to minimize transients. The state estimates gradually eliminate the effects of the sensor failure due to the increased estimation error covariance. The initialization of detectors and likelihood ratio computers after a sensor failure is identical to the procedure for starting a new detector and estimation residual window.

2.1.6 Healing Tests

In order to recover from false alarms associated with modelling errors (e.g. scale factor errors during significant maneuvers), tests for healing of a failed sensor are performed after the detection and isolation of a failure. Input sensors are tested for healing by comparing their outputs with a sensor of the same type currently used by the no—fail filter. This test is a binary hypothesis test conditioned on the decision rule outcome that the sensor used by the no—fail filter is healthy. The recovery of a

24

failed measurement sensor is tested by comparing its output with the estimate of that sensor output provided by the no—fail filter. Again, this test is a binary hypothesis test conditioned on the decision rule outcome that all sensors currently used by the no—fail filter are healthy. Both input and measurement sensor tests are performed only at the end of a healing test window, which is constrained to be an integer multiple of the decision residual window length as shown in Figure 4.

2.2 No—Fail Filter

In this section, we will present the no—fail filter algorithm along with the underlying aircraft dynamics which the filter design is based on. Our discussion begins in subsection 2.2.1 with a derivation of the aircraft equations of motion and the analytic relationships relating the no—fail filter sensor outputs to the aircraft dynamics. Subsection 2.2.2 contains the implemented filtering algorithm.

2.2.1 Aircraft Dynamics

The function of the no—fail filter is to provide estimates for the aircraft's position, velocity and attitude with respect to a ground frame located on the runway. As dictated by our application, the no—fail filter also provides normal operating bias estimates for a selected sensor subset and estimates for horizontal winds. Clearly, the degree of analytic redundancy which can be exploited by the FTS is dependent on the choice of underlying system dynamics for the no—fail filter design. In our study, we have chosen the aircraft <u>point</u> <u>mass</u> equations of motion for the system dynamics and a simple "signal plus bias—plus noise" model for the sensor measurements.

The following frames of reference (definitions can be found in [9]) will be used in

our discussion:

I frame: earth centered nonrotating (inertial) frame

E frame: earth fixed (rotating) frame

L frame: local level North, East, Down (N,E,D) frame located at A/C center of gravity

B frame: body frame

G frame: a geographic frame located at the start of the airport runway

Our goal is first to describe aircraft motion with respect to the G–frame while allowing for the earth's rotation and assuming a locally flat earth in the vicinity of the terminal area. Secondly, we will relate the sensor measurements to these equations of motion. The vector equation for the aircraft acceleration with respect to the G–frame which is itself rotating with respect to the inertial frame is given by [9]–[10]: (referring to Figure 3 for frame geometry)

$$\ddot{r}_G = T_{GL}[T_{LB} f_B + g_L] - 2 \, \Omega_G \dot{r}_G \tag{2.2.1}$$

where capital subscripts denote coordinatization (i.e. r_G in the r vector coordinatized in the G–frame). T_{GL} and T_{LB} are the transformation matrices from the L–frame into G–frame and from the B–frame into L–frame respectively. The vector f_B is the true specific force which would be measured by an ideal accelerometer in the body frame:

$$f_B = T_{BI}(\ddot{r}_I - g_I) \tag{2.2.2}$$

where g_I is the gravitational acceleration at the instrument location expressed in the inertial frame, g_L is the gravity field vector representing the acceleration from the

26

combined effects of earth's gravitational field and the centripetal acceleration defined by

$$g_L = g - T_{LG}\Omega_G\Omega_G\rho_G \qquad (2.2.3)$$

where ρ_G is the position vector from the center of earth to A/C center of gravity coordinatized in the G−frame. The rotation matrix Ω_G is the skew symmetric form of the angular rate vector ω_G defined by

$$\omega_G = T_{GI} \; [\omega^E,0,0]' \qquad (2.2.4)$$

where ω^E is the earth's rotation rate (7.27 x 10^{-5} rad/sec).

Modelling the accelerometer measurement inaccuracies by a "noise plus bias" type model, we have for the accelerometer measurement output u_a

$$u_a = T_{BI}(\ddot{r}_I - g_I) + b_a + n_a \qquad (2.2.5)$$

where b_a is the accelerometer bias vector in the body frame and n_a is the accelerometer noise vector. Substituting the expression for the accelerometer measurements (eq. (2.2.5)) into equation (2.2.1) for f_B, we get

$$\ddot{r}_G = T_{GL}T_{LB}u_a + T_{GL}g_L - 2\Omega_G\dot{r}_G - T_{GB}(b_a+n_a) \qquad (2.2.6)$$

Equation 2.2.6 above represents the equations of motion relating to the accelerometer measurements. The transformation matrices are given in Appendix A.

The equations relating the rate gyro measurements to the Euler angles are obtained from [10]:

$$\dot{e} = \Gamma_\omega[\omega_B - T_{BL}T_{LG}\omega_G] \qquad (2.2.7)$$

27

where e is the vehicle Euler angles defind by $e' = [\phi, \theta, \psi]$, ω_B is the true absolute vehicle rates vector in the body frame which would be measured by an ideal rate gyro triad, and Γ_ω is the nonorthogonal transformation matrix relating the body rates to the Euler angle rates. Assuming a "bias plus noise" model for the rate gyros defined by $u_\omega = \omega + b_\omega + n_\omega$ where b_ω is the rate gyro bias and n_ω is the associated noise vector, we get the following kinematics relationship:

$$e = \Gamma_\omega [u_\omega - T_{BG}\omega_G - b_\omega + n_\omega] \qquad (2.2.8)$$

We have used the following model for the horizontal winds

$$w = A_w w + n_w \qquad (2.2.9)$$

where $w' = [w_x, w_y]'$ with w_x and w_y are the horizontal wind components, n_w is a white Gaussian process noise with covariance Q_w. Defining the vehicle state $x' = [r_G' \dot{r}_G' e', w']$ and combining eqs. 2.2.1-9 we obtain the following state space description of the A/C point mass equations of motion.

$$\dot{x}(t) = A_c x(t) + B_c [u(t) - b_u] + B_r^c u_r^c + E_c n(t) \qquad (2.2.10)$$

where $u' = [u_a', u_\omega']$, $b'_u = [b_a', b_\omega']$, $n' = [n_a', n_\omega', n_w']$ and

$$A_c = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_w \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 \\ T_{GB} & 0 \\ 0 & \Gamma_\omega \\ 0 & 0 \end{bmatrix} \qquad (2.2.11)$$

28

$$B_r^c = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & \Gamma_\omega \\ 0 & 0 \end{bmatrix} \quad , \quad E_c = \begin{bmatrix} 0 & 0 & 0 \\ T_{GB} & 0 & 0 \\ 0 & \Gamma_\omega & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$U_r^c = \begin{bmatrix} T_{GL} \, g_L \, - \, 2\Omega_G \, \dot{r}_G \\ \omega_G \end{bmatrix}$$

Integrating this expression over a sampling interval of τ seconds [11], the following nonlinear discrete-time stochastic difference equation describing the aircraft dynamics is obtained:

$$x(k+1)=Ax(k) + B(x(k))[u(k)-b_u(k)] + u_g + n(k) \qquad (2.2.12)$$

where the six dimensional vectors u and b_u are composed of accelerometer and rate gyro measurements, and their associated biases, respectively. The vector u_g represents the incremental effect of the earth's constant gravitational force on the system state. The matrices A and B are defined by

$$A = \begin{bmatrix} I & \tau I & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad , \quad B(x(k)) = \begin{bmatrix} \tau^2/2 \, T_{GB} & 0 \\ \tau T_{GB} & 0 \\ 0 & \tau \Gamma_\omega \\ 0 & 0 \end{bmatrix} \qquad (2.2.13)$$

A_w is the 2x2 system matrix associated with the wind dynamics. The 3x3 matrix T_{GB} is the transformation from the body axes into the G frame [10], and Γ_ω is the 3x3 matrix relating the body rates to the Euler angles [10] defined in appendix A.

29

The variance, Q, of the white noise, n(k), is given by

$$Q = \begin{bmatrix} \tau^3/3 \, T_{GB} Q_a T'_{GB} & \tau^2/2 \, T_{GB} Q_a T'_{GB} & 0 & 0 \\ \tau^2/2 \, T_{GB} Q_a T'_{GB} & \tau T_{GB} Q_a T'_{GB} & 0 & 0 \\ 0 & 0 & \tau \Gamma_\omega Q_\omega \Gamma'_\omega & 0 \\ 0 & 0 & 0 & \int_0^\tau e^{A_w s} Q_w e^{A'_w s} \, ds \end{bmatrix} \qquad (2.2.14)$$

where Q_a and Q_ω are the measurement noise variances for the accelerometers and rate gyros, and Q_w is the process noise variance associated with the wind dynamics.

Note that the state transition matrix, A, is constant. However, both the process noise variance, Q(k), and the system input matrix, B, are state dependent due to the nonlinear state dependent transformation T_{GB} and Γ_ω. Now let us consider the measurement equations for the system described by eqs. (2.2.1)–(2.2.14). Let (x_M, y_M, z_M) and (x_E, y_E, z_E) be the azimuth and elevation antenna locations in the runway frame, and (r_x, r_y, r_z) be the A/C position relative to the runway expressed in the runway frame. Then, the MLS azimuth (y_{az}), elevation (y_{el}), and range (y_{rn}) measurements are defined by:

$$y_{az} = \sin^{-1}[(-r_y + y_M)/r_{az}] + b_{az} + v_{az} \qquad (2.2.15)$$

$$y_{el} = \sin^{-1}[(-r_z + z_E)/r_{el}] + b_{el} + v_{el} \qquad (2.2.16)$$

$$y_{rn} = r_{az} + b_{rn} + v_{rn} \qquad (2.2.17)$$

where (b_{az}, b_{el}, b_{rn}) and (v_{az}, v_{el}, v_{rn}) are biases and measurement noises associated with the MLS and r_{az}, r_{el} are the aircraft range from the azimuth and elevation antennas

given by:

$$r_{az} = \sqrt{(r_x - x_M)^2 + (r_y - y_M)^2 + (r_z - z_M)^2} \qquad (2.2.18)$$

$$r_{el} = \sqrt{(r_x - x_E)^2 + (r_y - y_E)^2 + (r_z - z_E)^2} \qquad (2.2.19)$$

Assuming a zero angle of attack, the airspeed indicator output, y_{sp}, is a noisy version of the aircraft velocity with respect to the atmosphere given by:

$$y_{sp} = \sqrt{(\dot{r}_x - w_x)^2 + (\dot{r}_y - w_y)^2 + \dot{r}_z^2} + b_{sp} + v_{sp} \qquad (2.2.20)$$

where (w_x, w_y) are the horizontal wind components and b_{sp} and v_{sp} are the IAS normal operating bias and white measurement noise. If the angle of attack measurement is available, then eq. (2.2.20) would be appropriately modified.

The IMU platform provides the Euler angle outputs. These roll (y_ϕ), pitch (y_θ), and yaw (y_ψ) angle measurements are modelled via

$$y_\phi = \phi + b_\phi + v_\phi \qquad (2.2.21)$$

$$y_\theta = \theta + b_\theta + v_\theta \qquad (2.2.22)$$

$$y_\psi = \psi + b_\psi + v_\psi \qquad (2.2.23)$$

where $(b_\phi, b_\theta, b_\psi)$ and $(v_\phi, v_\theta, v_\psi)$ are the biases and white measurement noises associated with platform outputs. Defining the measurement vector, $y' = [y_{az}, y_{el}, y_{rn}, y_{sp}, y_\phi, y_\theta, y_\psi]$, the system dynamics output becomes

$$y(k+1) = h(x(k+1)) + b_y + v(k+1) \qquad (2.2.24)$$

where b_y is the measurement sensor bias vector defined by $b_y' = [b_{rn}, b_{az}, b_{el}, b_{sp}, b_\phi, b_\theta, b_\psi]$ and v is the measurement noise vector defined by

31

$v'_y = [v_{rn}, v_{az}, v_{el}, v_{sp}, v_\phi, v_\theta, v_\psi]$. The nonlinear measurement function h(x) is defined by eqs. (2.2.15)−(2.2.24). In the next section, the no−fail filter which estimates the state variables and the normal operating biases of the stochastic nonlinear dynamic system described above will be discussed.

2.2.2 No−Fail Filter

In this section, we will describe the operation of the no−fail filter in detail. The no−fail filter is an extended Kalman filter estimating the aircraft runway position and velocity attitude and horizontal winds along with the normal operating biases of its inputs and measurements. The estimator uses either RSDIMU body outputs, or a set of body mounted accelerometer and rate gyro measurements as its inputs as discussed in the overview section. In the case of replicated inputs, redundant accelerometer and rate gyro sensors are kept as standby equipment.

MLS range, azimuth, and elevation sensors, and the IAS provide the measurements into the filter. If desired, IMU platform outputs, or RSDIMU computed attitudes, can also be included in the measurement set. For the case of hardware redundant measurements, the no−fail filter uses an average of the replicated sensor outputs as its measurement. In this way, filter size is kept to a minimum, without loss of generality. The no−fail filter also estimates the normal operating biases of any specified subset of the sensor complement.

In the process of obtaining the EKF used in our study, we have extended the separate bias estimation algorithms for linear systems to nonlinear systems via the extended Kalman filter framework. As discussed in the overview section, our extension

yields a numerical decomposition procedure for obtaining the extended Kalman filter gains. We will not discuss the the details of this procedure since they were adequately covered in the Interim Report [1] and associated papers [6]-[7]. Here, we will present the computational structure of the EKF algorithm for the system dynamics described by eqs. (2.2.12)-(2.2.24).

The following assumptions are made in obtaining the EKF algorithm. The system state and bias initial conditions are assumed to be zero mean Gaussian random variables with variances $P_x(0)$ and $P_b(0)$, respectively. In addition, it is assumed that the measurement noise $\{v(k), k=1,2,...\}$ is a zero mean, white Gaussian sequence with constant variance R. Furthermore, the plant state and bias initial conditions, measurement and process noise sequences are all assumed to be mutually uncorrelated.

In [1], [6]-[7], it is shown that the EKF equations for the nonlinear system dynamics described by eqs. (2.2.12)-(2.2.24) will be given by (dropping the functional dependence of variables and forming a composite bias vector b as $b=[b_u', b_y']'$

$$\hat{x}(k+1) = A\hat{x}(k) + B(\hat{x}(k))\hat{u}(k) + u_g + K_x(k+1)r(k+1) \tag{2.2.25}$$

$$\hat{b}(k+1) = \hat{b}(k) + K_b(k+1)r(k+1) \tag{2.2.26}$$

where the innovations sequence of the no-fail filter, r(k+1), is given by:

$$r(k+1) = \bar{y}(k+1) - h(\hat{x}(k+1/k)) - D\hat{b}(k) \tag{2.2.27}$$

and the bias compensated input vector, $\hat{u}(k)$, is given by:

$$\hat{u}(k) = u(k) - B_b\hat{b}(k) \tag{2.2.28}$$

Note that $D\hat{b}(k) = \hat{b}_y(k)$ and $B_b\hat{b}(k) = \hat{b}_u(k)$; therefore, these matrices are defined as $D = [0 \ I]$, $B_b = [I \ 0]$ if all input and output biases are estimated. The filter gain partition, K_x, is defined by:

$$K_x(k+1) = K_o(k+1) + V(k+1)K_b(k+1) \qquad (2.2.29)$$

where K_o is the "bias-free" filter given, V is the bias correction matrix and K_b is the bias filter gain. $K_o(k+1)$, $V(k+1)$ and $K_b(k+1)$ are computed sequentially using the linearized quantities:

$$F(\hat{x}(k),\hat{u}(k)) = A + \left. \frac{\partial}{\partial x}B(x(k))u(k) \right|_{\hat{x}(k),\hat{u}(k)} \qquad (2.2.30)$$

$$H(\hat{x}(k+1/k)) = \left. \frac{\partial}{\partial x}h(x(k+1)) \right|_{\hat{x}(k+1/k)} \qquad (2.2.31)$$

The expressions for the above partials are given in Appendix A of [1]. Recursive equations for the "bias-free" gain K_o, bias correction matrix V, and the bias gain K_b are given in Chapter 2 of [1].

The state estimation error covariance $P_x(k+1/k)$, bias estimation error covariance $P_b(k+1/k)$, and cross covariance of state and bias $P_{xb}(k+1/k)$ together define the prediction error covariance for the composite no-fail filter. They are defined by [7],[13]:

$$P_x(k+1/k) = P_o(k+1/k) + U(k)P_b(k)U'(k) \qquad (2.2.32)$$

$$P_{xb}(k+1/k) = U(k)P_b(k) \qquad (2.2.33)$$

$$P_b(k+1/k) = P_b(k) \qquad (2.2.34)$$

with

$$P(k+1/k) = \begin{bmatrix} P_x(k+1/k) & P_{xb}(k+1/k) \\ P'_{xb}(k+1/k) & P_b(k) \end{bmatrix} \qquad (2.2.35)$$

and where $P_o(k+1/k)$ is the prediction error covariance associated with the bias-free computations and V is the bias correction matrix. (See eq. 2.23 in [1]). The matrix U(k) is defined as:

$$U(k) = F(\hat{x}(k),\hat{u}(k))V(k) + B(\hat{x}(k)) \qquad (2.2.36)$$

Recursive equations for these matrices are given in [1]. The innovations variance, $R_o(k+1)$, can be expressed as:

$$R_o(k+1) = E\{r(k+1)r'(k+1)\}$$

$$= [H\hat{x}(k+1/k)D] \; P(k+1/k) \; [H(\hat{x}(k+1/k))D]' + R \qquad (2.2.37)$$

In the next section, the operation of the detectors, which are driven by the expanded innovations of the fail-free filter described above, will be discussed.

2.3 Detector Implementation

In this section, the blocks in Figure 1 labeled "residual computation" and "detectors" will be explained. In the residuals computation block, the residuals for the individual sensors are first computed, and then, the MLS measurement residuals are filtered to compensate for colored noise in these sensor outputs. The processed measurement residuals then drive a bank of detectors, where each detector delivers a failure corrected residual to the likelihood ratio computers. Each detector tracks the occurrence and level of a hypothesized sensor failure and compensates the no-fail

35

filter residuals such that the effects of the hypothesized sensor failure are removed from the residuals.

2.3.1 Expanded Residuals

As seen from Figure 1, the residual computation block receives as inputs, the replicated measurement sensor signals and the no-fail filter's estimates for the averaged measurements. It gives as its output an expanded residual and inverse of the innovations covariance for these expanded residuals. That is, this block generates the residual sequence (and its associated covariance) which would have been generated by the no-fail filter if it had used the unreplicated measurements.

In discussing these issues, it is convenient to define sensor <u>type</u> to be the generic type of the sensor measurement of interest, such as MLS azimuth, or body P gyro output, and sensor <u>replication</u> to be the particular replication of interest (i.e., second replication of MLS range). The replication will be noted by a superscript in the text (i.e., y_{az}^1 = first replication of MLS azimuth).

The residuals for each replicated measurement are formed as follows:

$$r^i(k+1) = y^i(k+1) - h(\hat{x}(k+1/k)) - D\hat{b}(k) \tag{2.3.1}$$

where

$$y^i = [y_{az}^i, y_{el}^i, y_{rn}^i, y_{IAS}^i, y_\phi^i, y_\theta^i, y_\psi^i]' \tag{2.3.2}$$

The expanded innovations for a dual redundant sensor set are then given by [note: in [1] r_e was referred to as r_o]

$$r_e(k+1) = \begin{bmatrix} r^1(k+1) \\ r^2(k+1) \end{bmatrix}$$

(2.3.3)

The innovations variance, $R(k+1)$ (called $\tilde{R}(k+1)$ in [1]), of the expanded residuals is found by straightforward substitutions to eq. (2.2.37) as:

$$R(k+1) = \begin{bmatrix} R_o(k+1) & R_o(k+1)-R \\ R_o(k+1)-R & R_o(k+1) \end{bmatrix}$$

(2.3.4)

where R is the measurement noise covariance for each set of replicated measurements, respectively. Equation (2.3.4) assumes that all measurement sensors are healthy. If, however, the jth sensor has been removed from the EKF, then $R(k+1)$ must be collapsed by eliminating the jth row and column.

2.3.2 Treatment of Colored Noise

As discussed in the Interim Report [1], the failure detection performance of the fault tolerant system with colored MLS measurement noises severely degraded due to false alarms. This is to be expected, since any time correlation in the no-fail filter residuals looks like a time-varying bias failure to the detectors. Therefore, it is essential to filter out the correlation in the residuals due to the colored MLS noise in order to have a robust failure detection system. We have investigated the following methods of treating colored measurement noise in our study:

  I. Estimate MLS noise states in the no-fail filter
  II. Use difference of MLS measurements
  III.Use suboptimal no-fail filter accounting for colored
       noise
  IV. Post process MLS residuals to remove colored noise

37

I) <u>Estimate MLS noise states in the no-fail filter</u>:

From Section 2.2 eq. (2.2.24) we have for the MLS measurements:

$$y_i(k+1) = h_i(x(k+1)) + b_i v_i(k+1) \quad i=1,2,3 \tag{2.3.5}$$

where $\{y_i, \; i=1,2,3\}$ are the MLS range, azimuth, and elevation measurements, respectively. In the derivation of the detector-estimator algorithms, the noises $v_i(k)$, were assumed to be white Gaussian sequences. However, these noises are, in fact, time correlated and are generated [3] via:

$$v_i(k+1) = \phi_i v_i(k) + n_i(k) \tag{2.3.6}$$

where $n_i(k)$ is a white Gaussian sequence. So the direct approach would be to augment the system states with the MLS noise states, v_i, and to estimate these variables along with other states. The obvious advantage of this method is that the resulting filter residuals would be white and the false alarms would be greatly reduced. The disadvantage of this technique is that the numerical complexity of the filtering algorithms would be increased due to the higher order covariance computations involved. Since the execution time of the filter algorithm was already high, we have decided not to implement this approach.

II) <u>Use difference of measurements</u>:

In this approach developed by Bryson and Henrikson [26], the filter equations are driven by the difference of measurements defined by:

$$z_i(k+1) = y_i(k+1) - \phi_i y_i(k) \tag{2.3.7}$$

The noises associated with the derived measurements $z_i(k+1)$ are white. Although the

38

filter dimension does not increase in this method, the filter algorithms become more complicated due to the correlation between process and measurement noises implemented. The extension of this differencing scheme to the EKF gets even more complex due to the linearizations involved. For instance, the measurement partials need to be computed both at the filtered state estimates and at the predicted state estimate. We have implemented this scheme by making some simplifying assumptions and found the detection capability of this technique to be unacceptable. This was largely due to the fact that a bias jump of magnitude m in the measurement y_i results in a jump sequence defined by:

$$\{m,(1-\phi_i)m,(1-\phi_i)m,...\} \tag{2.3.8}$$

in the derived measurements $z_i(k)$. Therefore, the detectors had difficulty in estimating the failure magnitude due to the initial spike of magnitude m.

III Use suboptimal filter accounting for colored noise:

In this approach suggested in [15], suboptimal filter gains are determined by minimizing the filtering error covariance accounting for the colored noise. The advantage of this approach is that the filter dimension does not increase. We have derived the algorithms for this filter along the lines presented in [15] for colored process noise. However, an analytic evaluation revealed that, although this suboptimal filter could improve no-fail filter estimation performance, it could not guarantee the whiteness of the resulting innovations sequence.

IV) Post process residuals to remove colored noise:

In this approach, the residuals $\{r_i(k), i=1,2,3\}$ corresponding to MLS range, azimuth, and elevation, measurements are further processed through first order filters given by

$$\hat{v}_i(k+1) = \phi_i v_i(k) + g_i[r_i(k+1) - \phi_i \hat{v}_i(k)] \qquad (2.3.9)$$

where $\hat{v}(k)$ is the estimate of the MLS noise state. The processed innovations $r_i(k+1)[r_i(k+1) - \phi_i \hat{v}_i(k)]$ are then used to drive the bank of detectors.

The implementation of this scheme resulted in a substantial reduction in the false alarms associated with colored MLS noises. The failure detection performance is essentially the same as the white noise case for measurement sensors. However, the addition of these first order filters degraded the detection capability for soft input sensor failures. The insertion of these first—order filters naturally necessitate changes in the variance of the innovations used by the detectors, and detector observation matrices.

2.3.3 Detectors

Every detector is driven by the expanded residuals sequence of the no—fail filter. For each sensor type and replication, there is a specific detector which keeps track of how a failure in that sensor occurring at the beginning of a decision window propagates through the no—fail filter dynamics to affect the expanded residuals. Based on this propagation effect, each detector estimates the level of the corresponding sensor failure and outputs a failure compensated residual sequence which is used by the likelihood ratio computers.

A typical (say, i'th) input sensor detector estimates a postulated bias jump in the i'th input at the beginning of a decision window (denoted by time k_o) so that the i'th input sensor detector design is based on the following modification of the system dynamics given by eq. (2.2.12):

$$x(k+1) = Ax(k) + B(x(k))[u(k) - b_u] + B_i(x(k))m_i(k) + u_g + n(k) \qquad (2.3.10)$$

$$m_i(k+1) = m_i(k) \quad \text{with } m_i(k_o) = m_i \text{ and } m_i(k) = 0 \text{ for } k < k_o \qquad (2.3.11)$$

where $B_i(x(k))$ is minus the i'th column of the input matrix $B(x(k))$ and m_i is the failed bias jump magnitude of the i'th sensor to be estimated. On the other hand, the detector for the i'th measurement sensor failure is based on the following modification of the measurement equation given by eq. (2.2.26):

$$y(k+1) = h(x(k+1)) + b_y + D_i m_i(k) + v(k+1) \qquad (2.3.12)$$

$$m_i(k+1) = m_i(k) \quad \text{with } m_i(k_o) = m_i \text{ and } m_i(k) = 0 \text{ for } k < k_o \qquad (2.3.13)$$

where m_i is the failed bias jump magnitude for the i'th output sensor and D_i is a column vector with unity entry at the i'th row and zeroes elsewhere. It is assumed that the failed bias jump magnitudes are unknown nonrandom variables.

As mentioned previously, the detectors utilize the residual of the no−fail filter as a measurement equation. In Appendix C of [1], it is shown that the residual of the no−fail filter, in the case of i'th failure hypothesis, can be expressed as:

$$r(k+1) = C_i(\hat{x}(k+1/k))m_i + \tilde{r}(k+1) \qquad (2.3.14)$$

where $\tilde{r}(k)$ is the innovations of the no−fail filter under the no−fail hypothesis. Therefore, $\tilde{r}(k)$ is approximately a zero mean white noise sequence with variance $R(k+1)$ defined by eq. (2.3.4). Referring back to eq. (2.3.14), $\tilde{r}(k+1)$ would then be the measurement noise in the i'th detector model and the measurement matrix

41

$C_i(\hat{x}(k+1/k))$ would be given by (see Appendix C in [1] for the derivation):

$$C_i(\hat{x}(k+1/k)) = [H(\hat{x}(k+1/k))\ D] \begin{bmatrix} F_i(\hat{x}(k),u(k)) & B_b \\ O & I \end{bmatrix} V_i(k)$$

$$+ [H(\hat{x}(k+1/k)\ D] \begin{bmatrix} B_i \\ 0 \end{bmatrix} + D_i \qquad (2.3.15)$$

In the case of linear systems the relations above, which show the additive effects of bias jump failures on the no-fail filter, are exact. In the nonlinear case, they are obtained by expanding the system nonlinearities about the no-fail filter estimate under the i'th hypothesis, deriving the linearized filtering error equations [7], and following the procedure outlined in [1].

Note that the left most matrix product in C_i above shows how the failure propagates through the dynamics to affect the residuals; the middle product depicts the direct effects of input failures, and the right most matrix illustrates the direct effect of output failures. Furthermore, $B_i(\hat{x}(k))$ is zero in the case of measurement sensor failures and, D_i is zero in the case of input sensor failures. The matrix $F_i(\hat{x}(k),\hat{u}(k))$ is defined by:

$$F_i(\hat{x}(k),\hat{u}(k)) = F(\hat{x}(k),\hat{u}(k))$$

$$+ \left. \frac{\partial B_i}{\partial x}(x(k))m_i \right|_{\hat{x}(k),\hat{m}_i(k)} \qquad (2.3.16)$$

Where $F(\hat{x}(k),\hat{u}(k))$ is given by eq. (2.2.30). (Note, for measurement sensor failures $F_i = F$ since the failures do not enter through the input weighting matrix B.)

The matrix $V_i(k)$ is analogous to the bias correction matrix in the separated EKF algorithm [7] and represents the propagation of a sensor failure, occurring at time k_o (recall k_o is the start of the estimation residual window), through the no-fail filter dynamics. It is computed using the following recursive relationship:

$$V_i(k+1) = \bar{A}_i V_i(k) + \bar{B}_i \qquad (2.3.17)$$

where $V_i(k_o) = 0$, and;

$$\bar{A}_i = \bar{E}_i \begin{bmatrix} F_i(\hat{x}(k), u(k)) & B_b \\ 0 & I \end{bmatrix}$$

$$\bar{E}_i = I - \begin{bmatrix} K_x \\ K_b \end{bmatrix} [H(\hat{x}(k+1/k)) \quad D]$$

$$\bar{B}_i = \begin{bmatrix} B_i \\ 0 \end{bmatrix} - \begin{bmatrix} K_x \\ K_b \end{bmatrix} \left\{ [H(\hat{x}(k+1/k)) \quad D] \begin{bmatrix} B_i \\ 0 \end{bmatrix} + D_i \right\}$$

The gains K_x and K_b are given by eqs.(2.2.29)–(2.2.31). Note that eq. (2.3.17) is similar to the recursive relation for the bias correction matrix recursive relation in the separated EKF algorithm. This is to be expected since $V_i(k+1)$ represents the effect of a sensor bias failure on the composite no-fail filter and $V(k+1)$ in the separated EKF represents the effect of a normal operating bias on the bias free portion of the fail free filter. The postulated sensor failure's effect on both state and normal operating bias estimates are thus computed.

Summarizing, the i'th detector design is based on the observation model

43

described by eq. (2.3.14) and constant failure dynamics. The development up to this point has assumed the value of m_i is known. In reality, m_i is nonrandom, but unknown. Therefore, one must continuously estimate its value.

The i'th detector estimate, $\hat{m}_i(k)$, of the i'th sensor failure jump, $m_i(k)$, can be computed by the following first order linear Kalman filter for the case of measurement sensor failures, and by a first order approximate nonlinear filter in the case of input sensor failures:

$$\hat{m}_i(k+1) = \hat{m}_i(k) + G_i(k+1,\hat{x}(k+1/k))[r(k+1)$$

$$- C_i(k+1,\hat{x}(k+1/k))\hat{m}_i(k)] \tag{2.3.18}$$

where the detector estimate $\hat{m}_i(k)$ is initialized at the start of a residual window with $\hat{m}_i(k_o)=0$. The detector gain is computed by:

$$G_i(k+1,\hat{x}(k+1/k))=P_i(k+1/k+1)C_i'(k+1,\hat{x}(k+1/k))R^{-1}(k+1) \tag{2.3.19}$$

where $P_i(k+1/k+1)$ is the error covariance of the i'th detector bias jump estimate. The information matrix, $P_i^{-1}(k/k)$, of the i'th detector is propagated recursively through:

$$P_i^{-1}(k+1/k+1) = P_i^{-1}(k/k)$$

$$+ C_i'(k+1,\hat{x}(k+1/k))R^{-1}(k+1)C_i(k+1,\hat{x}(k+1/k)) \tag{2.3.20}$$

with

$$P_i^{-1}(k_o/k_o)=0$$

That is, the failure bias jump at time k_o is assumed to be a zero mean random variable with infinite covariance (or equivalently, zero information). In the case of output

sensor failures, the detector implementation described by eqs. (2.3.18)-(2.3.20) above is an exact linear Kalman filter for the hypothesized failure model specified by eqs. (2.3.10)-(2.3.15). In the case of input sensor failures, the detector becomes an approximate nonlinear filter due to the dependence of F_i in eq. (2.3.16) on the failure bias magnitude, m_i where $\hat{m}_i(k)$ is used in the evaluation of $F_i(\hat{x}(k),\hat{u}(k))$.

In summary, the detector block consists of a bank of first order estimators driven by the expanded innovations of the no-fail filter. Each detector corresponds to a different sensor failure hypothesis, and, corresponding to each detector, there is an associated residual data window length. The bias jump magnitude for a given sensor failure, hypothesized to happen at the start of the residual window, is estimated by the detector corresponding to that sensor. The residuals of the detectors along with the residual of the no-fail filter are used in the decision block. In the next section, the decision rules used in FINDS will be discussed.

2.4 Decision Rule

As seen in Figure 1, the failure compensated residuals from each of the sensor failure detectors along with the expanded innovations sequence of the no-fail filter are used in deciding the most likely failure mode. To arrive at this decision, M-ary hypothesis testing, based on a decision residual window, is utilized.

Tests for isolated, singleton sensor failures will be examined in the next section. Tests for multiple failures (two failures occurring at the same instant of time) will be discussed in the last section. Currently, tests for multiple failures are only performed for MLS azimuth elevation, and range sensors in order to detect antenna failures.

45

2.4.1 Tests for Single Sensor Failures

Tests for single sensor failures are derived through a multiple hypothesis testing formulation. Given M sensor failure models, formulate the following M+1 hypotheses:

$$H_o: \quad r_o(k) = \tilde{r}(k) \qquad\qquad k=k_d, k_d+1,...,k_d+l_d \qquad (2.4.1)$$

$$H_i: \quad r_o(k) = \tilde{r}(k) + C_i(\hat{x}(k/k-1))m_i \qquad i=1,2,...,M$$

where $r_o(k)$ is the actual expanded innovation sequence of the no-fail filter, $\tilde{r}(k)$ is the innovations sequence of the no-fail filter under no-fail conditions, and l_d is the length of the decision residual data window on which the M-ary hypothesis test is based. Recall from the previous section, that $\tilde{r}(k)$ is a zero mean white noise sequence with variance $\tilde{R}(k)$ defined by eq. 2.3.4. The length of the decision residual window is, in general, different from the estimation residual data windows described in the previous section. An M-ary hypothesis test will be used to decide whether the no-fail filter is operating under no failures (hypothesis H_o), or under the i'th sensor bias jump failure (hypothesis H_i).

The M-ary hypothesis test, described in detail in [18], minimizes the Bayes risk which is a weighted cost of making incorrect decisions. In the special case, when costs associated with making wrong decisions are all equal and those of making correct decision are zero (i.e., $C_{ij}=1$ for $i \neq j$ and $C_{ii}=0$), then the optimal Bayesian decision would be to choose H_i corresponding to the smallest one of the M+1 likelihood ratios Λ_i given by:

$$\Lambda_i = \frac{1}{2} \sum_{k=k_d}^{k_d+l_d} r'_i(k) \, \tilde{R}^{-1}(k) r_i(k) - \ln P_{H_i} \quad ;i = 0,1,...m \qquad (2.4.2)$$

Stated differently, the decision rule is equivalent to choosing the hypothesis, H_i,

corresponding to the largest a posteriori probability conditioned on the residual window Y(K).

The a priori probabilities, P_{H_i}, represent our prior knowledge about how often particular sensors fail. In this study, the a priori probabilities will be based on typical manufacturers specifications of mean time between failures (MTBF). The following rule, which simply converts MTBF (in hours) to MTBF in 1/2 a decision window, will be applied.

$$P_{H_i} = 2*MTBF/(3600*l_d*\tau)$$

$$P_{H_0} = 1 - \sum_{i=1}^{M} P_{H_i} \qquad (2.4.3)$$

2.4.2 Test for Simultaneous Multiple Failures

While the single sensor failure model described in the previous subsection, may be able to handle multiple failures by viewing them as a sequence of single sensor failures, it is not clear that the fault tolerant system can decipher simultaneous multiple failures without any modifications. These types of failures are especially important for the MLS measurements since an antenna failure would produce a simultaneous multiple failure in the corresponding MLS measurements. In this subsection, a number of possible additions to the fault tolerant system structure, which have been analyzed for the testing of multiple simultaneous failures, will be discussed.

Recall that the previously discussed fault tolerant system structure requires a (first order) detector for each sensor utilized by the no-fail filter. Since the no-fail

47

filter uses only one set of the redundant input sensors (accelerometers and rate gyros), multiple failures for input sensors need not be considered. On the other hand, because the no—fail filter uses both of the dual redundant measurement sensors, (MLS, IAS, IMU measurements), multiple failures for output sensors must be taken into account. Of course, any combination of (input or measurement) sensors would constitute a valid simultaneous failure. Since the number of possible combinations are exceedingly high, and since most probable simultaneous failures are due to MLS antenna malfunctions, we have decided to incorporate multiple failure tests for MLS azimuth, elevation and range sensors only. There are basically three approaches one can take in dealing with simultaneous multiple failures within the context of the existing fault tolerant system structure:

    I. No Modifications

   II. Multiple Failure LR's

  III. Multiple Failure Detectors and LR's

I. <u>No Modifications</u>

The thought behind this approach is that the FTS, designed for single failures, might be robust enough to detect multiple failures sequentially. This would be a desirable property, since arbitrary multiple failures could be handled without additional computational burden. To investigate this avenue we have examined the failure detection and isolation performance of the existing FTS under multiple MLS, IAS and IMU failures. As one might expect, one problem with this approach is that the posteriori probabilities tend to converge to equal values (such as 1/2 or 1/3) . Generally speaking, the FTS worked quite well for cases when the a posteriori

probabilities converged to values greater than 1/3. Multiple failures were correctly detected in a sequential fashion (decisions were one sampling interval apart). However, since other cases produced more unsatisfactory results, it was clear that this approach was not adequate in general.

II. Multiple Failure LR's

The next level of complexity would be to include new LR tests for simultaneous multiple failures, without adding new detectors for these hypotheses. Recall from eq.(2.3.14) that the LR tests for single failures assume the following failure signature models for the no-fail filter residuals:

$$r_o(k+1) = C_i(k+1)m_i + \tilde{r}(k+1) \qquad i=0,1,..,M \text{ with } m_o=0 \qquad (2.4.4)$$

where r_o is the residual sequence of the no-fail filter, m_i is the failure level for the i'th sensor, C_i is the "observation" vector which relates the i'th failure level onto the no-fail filter residuals, and $\tilde{r}(k+1)$ is the white noise sequence which would have been obtained for the no-fail filter residuals if there were no failures present (Remember that this failure signature model was derived by linearizing the error dynamics). Since the error dynamics are linear, if there were two sensor failures (say i and j) simultaneously present, then the following failure signature model would result:

$$r_o(k+1) = C_i(k+1)m_i + C_j(k+1)m_j + \tilde{r}(k+1) \qquad (2.4.5)$$

where m_i and m_j are the failure levels for the i'th and j'th sensors, C_i and C_j are the measurement vectors for the two failure models (dependency has been dropped for convenience). Combining the failure levels into a single vector, $\bar{m}$ the failure signature model becomes:

49

$$r_o(k+1) = [C_i(k+1) \ C_j(k+1)]\bar{m} + \tilde{r}(k+1) \qquad (2.4.6)$$

$$i = 1,2,\ldots,M$$
$$j = 1,2,\ldots,M$$

This model is exact for measurement sensors and approximate for input sensors due to the dependence of C_i and C_j on m_i and m_j respectively. That is, when the i'th and j'th sensor failures are considered simultaneously, the linearizations for the input matrices can be slightly different than the case in which the failures are considered separately. For the evaluation of the LR's, we need their estimates $\hat{m}_i$ and $\hat{m}_j$. One approximation is to use $\hat{m}_i$ and $\hat{m}_j$ from the single sensor failure detectors. Clearly, by considering the i'th and j'th sensor simultaneously, estimates for m_i and m_j may be improved by using a second order detector. Using the estimates from the single sensor failure detectors, we obtain the following residual, corrected for the simultaneous i'th and j'th failure:

$$r_{ij}(k+1) = r_o(k+1) - C_i(k+1) \ \hat{m}_i(k) - C_j(k+1) \ \hat{m}_j(k) \qquad (2.4.7)$$

So that LR's for the dual sensor failures can be computed by:

$$\Lambda_{ij}(r(K)) = \frac{1}{2} \sum_{k=k_d}^{k_d+l_d} r'_{ij}(k)\tilde{R}^{-1}(k)r_{ij}(k) - \ln P_{H_{ij}} \qquad (2.4.8)$$

$$i = 1,\ldots,M$$
$$j = 1,\ldots,M$$

where $P_{H_{ij}}$ is the a priori probability of the i and j sensors failing simultaneously. Computational requirements for this procedure are relatively modest. Since new LR's introduce only recursive scalar quantities, the additional computations are not excessive. Caution should be exercised in selecting the a priori multiple failure probabilities because false alarms associated with these failure models could have

50

serious effects on the system performance. In this study the following rule (which assumes independence between failures) is used:

$$P_{H_{ij}} = P_{H_i} * P_{H_j} \qquad (2.4.9)$$

In addition, to reflect the type of multiple failure expected, we have chosen the following form for $\bar{m}$ defined in eq. (2.4.6):

$$\bar{m} = [(m_i + m_j)/2 \quad , \quad (m_i + m_j)/2] \qquad (2.4.10)$$

III. New Detectors and LR Tests

In this approach, the same procedure is followed as described in the previous section, except, new second order detectors are implemented using the failure signature models, eq. (2.4.6), for multiple failures.

These detectors will estimate m_i and m_j, simultaneously. Note that they will be second order for the dual failure case considered here. While this procedure is optimal for multiple failures, we believe that the computational benefit to be gained is far outweighed by the increased computational requirements.

In conclusion, after examining the three approaches outlined above, we've adopted approach II, modification of the LR's, as a reasonable method of handling multiple failure conditions.

51

2.5 Healing Tests

An important aspect of the developed fault tolerant system is its ability to monitor sensors which it has isolated as "failed", and to determine if they have recovered.

A healing test can be useful for a number of practical reasons. For example, if the sensor only fails intermittently, or if unmodeled (normal operating) characteristics of the sensor manifest themselves as moderate errors for short periods of time, it would be useful to heal the sensor after the sensor recovers or the transient has passed. Another practical utility of these tests occurs when the FTS incorrectly detects a sensor as failed (false alarm).

There are, of course, pros and cons to using healing tests at all. It can be argued that if, for example, a sensor fails with an increased scale factor error then it is only detectable during transient maneuvers and will appear "healthy" otherwise. In this case, the sensor may be correctly detected as "failed", only to be "healed" once the transient has passed. Admittedly, these sorts of problems do exist; however, since in this study, we have adopted simple models for the normal operational sensors, healing tests are a sensible alternative to increased redundancy or modelling complexity. Moreover, the FTS maintains a log of all FDI activity, such that a sensor with a history of chronic problems can be identified and dealt with.

In our FTS environment, sensors used by the no—fail filter as inputs are treated differently than those used as measurements, when testing for healing. The essential

difference is that failed input sensors (body mounted accelerometers and rate gyros) are compared to "healthy" sensors of the same type (with the constraint that these healthy sensors are also being used by the no—fail filter), whereas failed measurement sensors can be compared to estimates provided by the no—fail filter. The implications of this difference, along with the specifics of the employed healing tests, will be discussed in the ensuing subsections.

2.5.1 Test for Input Sensor Recovery

A likelihood ratio (LR) test will be used to determine whether or not an input sensor, which has failed and been taken out of the no—fail filter, has recovered. The basic idea is to compare the output of a failed sensor with another, like sensor, which we assume is healthy. The comparison is carried out as the difference between the two signals, over a fixed length healing test window of length l_h. Likelihood Ratios are computed based on this comparison, along with information about expected normal operating bias levels and expected failure levels. Discrete decisions are made at the end of a complete healing test window. In other words, if the FTS decides that a sensor has failed before the end of a healing window, no healing decision will be made. An important constraint imposed by our FTS is that the "healthy" sensor be currently used by the no—fail filter. This is done because only those sensors used by the filter are monitored for failures. As a result, standby sensors (not used by the filter), could have failed already.

Here we will consider the healing test for an arbitrary input sensor. Initially, all input sensors are assumed to be working. Therefore, there will not be any tests for healing until an input sensor actually fails. To describe the healing mechanism, we

begin by making the following assumptions about the characteristics of a working and a failed sensor:

o "healthy" sensor model:

$$u_i(k) = u_i^t(k) + b_i + v_i(k) \tag{2.5.1}$$

o "failed" sensor model

$$u_i(k) = u_i^t(k) + b_i + v_i(k) + m_i \tag{2.5.2}$$

where $u_i^t(k)$ is the "true" or ideal sensor signal, and b_i and $v_i(k)$ are the normal operating bias and white gaussian measurement noise associated with the i-th input sensor $u_i(k)$ respectively. The term m_i represents a bias failure of magnitude m_i. Suppose an input sensor u_1, fails and is replaced by a second (standby) input sensor of the same type, u_2. These two signals can be compared by computing the following difference signal over a healing test window that is synchronized to the start of a decision window (see Figure 4).

$$u(k) = [u_2(k)-u_1(k)] \qquad\qquad k=k_d+1,...,k_d+l_h \tag{2.5.3}$$

At the end of a healing test window initiated for that input sensor, we test for the healing of the failed input sensor, $u_1(k)$, provided no failures were announced by the fault tolerant system. Note that testing is not performed at every sample, but rather only at the end of each healer window. Defining the following two hypotheses, and incorporating the appropriate sensor/failure models into eq. (2.5.3), we obtain:

$$H_0(u_1 \text{ healthy}) \qquad\qquad\qquad H_1(u_1 \text{ not healthy})$$

$$u(k) = b_2 - b_1 + v_2(k)-v_1(k) \qquad\qquad u(k) = b_2-b_1-m_1+v_2(k)-v_1(k)$$

$$= \Delta b + \Delta v(k) \qquad\qquad\qquad = \Delta m + \Delta v(k)$$

where $\Delta b = b_2 - b_1$, and $\Delta v(k) = v_2(k) - v_1(k)$. If the variance of the two

54

measurements are equal (say σ_u^2), then $v(k)$ would be a zero mean white Gaussian sequence with variance $2\sigma_u^2$.

Based on the expected value of the normal operating biases, we can put the following constraints on Δb and Δm:

$$-\beta_u \leq \Delta b \leq \beta_u$$

$$\Delta m < -f_u \quad \text{or} \quad \Delta m \geq f_u$$

where β_u and f_u are expected levels for the normal operating bias and failure level for input sensor type u, respectively. We can now apply the likelihood ratio for this composite hypothesis testing problem to get the following decision rule [18]:

$$\sum_{k=1}^{k_d + l_h} [\Delta u(k) - \hat{b}_o]^2 - [\Delta u(k) - \hat{m}_1]^2 \underset{H_o}{\overset{H_1}{\gtrless}} 2(2\sigma_u^2)\ln\gamma$$

where[3] $\hat{b}_o$ and $\hat{m}_1$ are the maximum likelihood (ML) estimates under each hypothesis given by:

$$\hat{b}_o = \Delta\bar{u} \quad \text{if} \quad -\beta_u \leq \Delta\bar{u} \leq \beta_u$$

$$\hat{b}_o = \beta_u \quad \text{if} \quad \Delta\bar{u} > \beta_u$$

$$\hat{b}_o = \beta_u \quad \text{if} \quad \Delta\bar{u} < -\beta$$

---

[3]The mathematical symbol $\gtrless$ is defined as follows, given: $a \underset{H_o}{\overset{H_1}{\gtrless}} b$, hypothesis H_1 is considered true if $a > b$; and hypothesis H_o is true if $a < b$.

where,

$$\Delta \bar{u} = \frac{1}{I_h} \sum_{k=k_d+I}^{k_d+I_h} \Delta u(k)$$

$$\hat{m}_1 = \Delta \bar{u} \quad \text{if} \ \Delta \bar{u} \geq f_u \ \text{or} \ \Delta \bar{u} \leq - f_u$$

$$\hat{m}_1 = f_u \quad \text{if} \ 0 \leq \Delta \bar{u} < f_u$$

$$\hat{m}_1 = -f_u \quad \text{if} \ -f_u < \Delta \bar{u} < 0$$

For instance, the choice for the threshold $\gamma = \frac{1}{19}$ would imply that the a priori probability of a failed sensor not healing is 0.95 [18].

In summary, the input sensor healing test is done in batches. The test is only performed while the FTS considers a sensor "failed". At the end of each complete healer window, a likelihood ratio test is performed to ascertain whether the faulty instrument has recovered or not. If a decision indicating that the faulty sensor has recovered is made, the only action is to change the status flag of that input sensor to standby status.

2.5.2 Test for Measurement Sensor Recovery

The recovery of a failed measurement sensor can be detected, as in the case of input sensors, by comparing the output of the failed sensor with that of a sensor of the same type which is currently used by the no-fail filter. However, another possibility is to compare the failed sensor with an estimate of its output provided by the no-fail filter. (Note that this procedure is not possible for input sensors). The latter approach has been adopted in this study since it has the advantage of applicability even when a given sensor type is not utilized by the no-fail filter. That

56

is, all sensors of a given type have failed and been taken out.

Let's consider the healing test for an arbitrary measurement sensor. Suppose the first replication of a particular measurement type, call it Z_1, has failed and been removed from the no-fail filter. Assuming dual redundancy, only the second measurement, Z_2, remains in the no-fail measurements. To mimic the input healing test the following residual over the healing test window is computed:

$$\Delta Z(k) = Z_1(k) - \hat{Z}(k/k-1)$$

$$\Delta Z(k) = Z_1(k) - h(\hat{x}(k/k-1)) - \hat{b}(k),$$

$$k = k_d+1, k_d+2, \ldots, k_d+l_h \tag{2.5.4}$$

Here it is assumed that the normal operating biases for the two measurements are similar. Where $h(\hat{x}(k/k-1))$ above is the estimate of $Z(k)$ provided by the no-fail EKF, and $\hat{b}(k)$ is the no-fail filter bias estimate. We then have the two hypotheses (after substituting eq. (2.2.24)):

$H_o(Z_1$ healthy) $\qquad\qquad\qquad$ $H_1(Z_1$ not healthy)

$\Delta Z = h(x(k)) - h(\hat{x}(k/k-1))$ $\qquad$ $\Delta Z = m_1 + \tilde{v}(k)$

$\qquad + b - \hat{b}_2 = \tilde{v}(k)$ $\qquad\qquad\qquad$ where $m_1 \geq f_Z$ or $m_1 \leq - f_Z$

Under hypothesis H_o, $\tilde{v}(k)$ would be a white Gaussian sequence with zero mean and variance, σ_Z^2, given by the appropriate diagonal entry of the residual variance matrix of the no-fail filter.

The LR test can now be applied in a manner similar to the input healing test.

The ML estimate for m_1 would be given by:

$$\hat{m}_1 = \Delta\bar{Z} = \frac{1}{l_h} \sum_{k=k_d+1}^{k_d+l_h} Z(k), \qquad \text{if } \Delta\bar{Z} \le -f_Z \text{ or } \Delta\bar{Z} \ge f_Z$$

$$\hat{m}_1 = +f_Z \qquad \text{if } 0 \le \Delta\bar{Z} < f_Z$$

$$\hat{m}_1 = -f_Z \qquad \text{if } -f_Z < \Delta\bar{Z} < 0$$

So that the composite hypothesis test for this problem becomes:

$$\sum_{k=k_d+1}^{k_d+l_h} \{\Delta\bar{Z}^2 - [\Delta\bar{Z}-\hat{m}_1]^2\} \underset{H_o}{\overset{H_1}{\gtrless}} 2\sigma_Z^2 \ln\gamma$$

If a measurement sensor bias is <u>not</u> estimated, under H_o, there will be a normal operating bias mean in the residuals. In this case, under H_o, residuals can be computed as:

$$\Delta Z(k) = \tilde{b} + \tilde{v}(k), \quad \text{where } -\beta_Z \le \tilde{b} \le \beta_Z$$

where,

$$\hat{b}_o = \bar{Z}, \qquad\qquad \text{if } -\beta_Z \le \Delta\bar{Z} \le \beta_Z$$

$$\hat{b}_o = \beta_Z, \qquad\qquad \text{if } \Delta\bar{Z} > \beta_Z$$

$$\hat{b}_o = -\beta_Z, \qquad\qquad \text{if } \Delta\bar{Z} < -\beta_Z$$

This results in the following composite hypothesis test:

$$\sum_{K=K_d+1}^{K_d+l_h} \{[Z(k)-\hat{b}_o]^2 - [Z(k)-\hat{m}_1]^2\} \underset{H_o}{\overset{H_1}{\gtrless}} 2\sigma_Z^2 \ln\gamma$$

In summary, when an output sensor fails and is removed from the no-fail filter measurement set, its output is compared with an estimate provided by the no-fail

filter. The LR computation generates a maximum likelihood estimate for the normal operating bias and failure level. The composite hypothesis test is done at the end of every complete healing test window. If the test indicates that the sensor has recovered, the measurement is incorporated back into the no-fail filter measurement set.

2.6 Reinitialization Procedure

Fault tolerant systems in which analytic failure detection and isolation (FDI) techniques are used on-line to identify system failures usually require some level of compensation in order to remove the accumulated effects of the detected failure on the system. In the developed FTS, sensor failures (especially input failures, and soft measurement failures) have to propagate through the no-fail filter dynamics (until a significant residual signature is generated) in order to be detected. Therefore, the no-fail filter must be reinitialized in order to remove the accumulated effects of the detected failure on the filter. In addition, the no-fail filter must be restructured after the isolation of a failure to account for the loss of a sensor input or measurement.

There are a number of ways in which reinitialization can be accomplished within our framework (see [29]). For instance, the measurements can be reprocessed if the failure onset time can be estimated accurately. In our problem, however,the exact failure time is not estimated since a fixed length window of measurements are used.[4]

---

[4]The procedure for finding the exact failure times is described in [13] and involves using a set of moving windows corresponding to different failure detection times.

A second problem with this strategy is the additional requirement of saving the past measurements for the moving windows. Finally, re–running of the no–fail filter with a previous set of measurements, generally, cannot be done in real time.

Another method involves resetting the no–fail filter state estimate and covariance using the same procedure followed in setting the initial levels for these variables. The drawback of this procedure is its neglect of the information embedded in the failure decision logic and associated failure level estimates.

If the failure level estimates provided by the detectors can be trusted, then the no–fail filter state estimate can be corrected by adding an appropriate increment due to the detected failure as suggested in [27]. The no–fail filter covariance is also incremented in this procedure by using the covariance associated with the failure level estimate. Although the failure level estimates provided by the detectors usually provide a reliable failure direction, the magnitude of the failure level estimates are not usually very accurate due to the uncertainty associated with failure onset time and detector settling time. Moreover, in some applications, it is not desirable to have step changes in the plant state estimates due to the transients produced, in devices which use these estimates as inputs.

These drawbacks associated with the reinitialization procedures above can be minimized by resetting only the covariance of the no–fail filter. In this method, the appropriate increment of the no–fail filter covariance is found by computing the conditional covariance of the no–fail filter state estimate conditioned on the observation sequence under the detected failure mode.

Finally, the last possibility is the weighting of the no—fail filter state and detector failure level estimates by the a posteriori probabilities. In this technique, instead of the hard switching produced by a decision rule, the a posteriori probabilities provide a soft switching between the failure modes.

The reinitialization methods discussed above can be grouped into the following categories:

o Reprocess Measurements

o Reinitialize (Estimate and) Covariance

o Reset Estimate and Increment Covariance

o Conditional Covariance

o A Posteriori Probability Weighting

In our study, we have compared only the second, third and fourth approaches which are described next.

I) Reinitialize (Estimate and) Covariance: In this approach, the no—fail filter covariance parameters are set to the values used as initial conditions. The state estimate is be reinitialized by following the procedure employed in selecting the plant state estimate initial conditions. Naturally, this approach generates transients associated with the settling of the filter gains similar to that encountered in the initial stage of the problem.

II) Reset Estimate and Increment Covariance: In the case of an i'th failure mode decision, the no—fail filter state estimate would be reset to [24]:

61

$$\hat{x}_i(k) = E[x(k)/Y(k),H_i] = \hat{x}(k) + V_i(k)\hat{m}_i(k) \tag{2.6.1}$$

where $\hat{x}(k)$ is the state estimate from the no-fail filter (eq. (2.2.25)), $V_i(k)$ is the failure correction matrix for the i'th failure mode (eq. (2.3.17)), and $\hat{m}_i(k)$ is the i'th failure state estimate (eq. (2.3.18)) provided by the i'th detector. When the state estimate is reset according to the procedure above, then the corresponding prediction error covariance is given by:

$$E[\tilde{x}_i(k+1/k)\tilde{x}_i'(k+1/k)/H_i] = P(k+1/k) + U_i(k)P_i(k)U_i'(k) \tag{2.6.2}$$

where $\tilde{x}_i(k+1/k)$ is the single stage prediction error associated with the estimate, $\hat{x}_i(k)$, given by

$$\tilde{x}_i(k+1/k) = x(k+1) - \hat{x}_i(k+1/k) \tag{2.6.3}$$

or, rearranging eq. (2.6.3)

$$\hat{x}_i(k+1/k) = A[\hat{x}(k) + V_i(k)\hat{m}_i(k)] + B_i\hat{m}_i(k) \tag{2.6.4}$$

$$= A\hat{x}(k) + [AV_i(k) + B_i]\hat{m}_i(k)$$

$$= \hat{x}(k+1/k) + U_i(k)\hat{m}_i(k)$$

where $U_i(k)=AV_i(k)+B_i$, $P_o(k+1/k)$ is the prediction error covariance of the no-fail filter, and $P_i(k)$ is the prediction error covariance of the i'th detector. Therefore, the prediction error covariance of the no-fail filter can be incremented by the second term on the right hand side of eq. (2.6.2) above if the state estimate is reset according to eq. (2.6.4).

The state estimate above is the optimal (least mean square sense) estimate conditioned on the i'th failure mode provided that the failure onset time and the

62

failure state initial statistics correspond to those of the actual failure. However, in practical applications such as ours, the failure onset time cannot be accurately estimated due to the necessity of using a bounded set of detectors as opposed to a growing number required by a fully optimal decision rule. Moreover, sudden changes in the state estimates would not be desirable due to their effects on the automated landing and control laws.

III) <u>Conditional Covariance</u>: To reduce the transient effects produced by the previous method, the no-fail filter can be reinitialized by incrementing only the error covariance by an appropriate amount following the isolation of a failure. In this manner, the state estimation error in the no-fail filter can be compensated gradually. In this procedure, the appropriate covariance to be used is the <u>conditional covariance</u> of the no-fail filter <u>conditioned</u> <u>on</u> <u>the</u> <u>given</u> <u>observations</u> under the decided failure mode. In other words, we need to compute $E[\tilde{x}(k+1/k)\tilde{x}(k+1/k)/Y(k),H_i]$ where $\tilde{x}(k+1/k)$ is the prediction error of the no-fail filter defined by $\tilde{x}(k+1/k)=x(k+1)-\hat{x}(k+1/k)$. Since the single stage prediction of the no-fail filter can be expressed by (adding and subtracting the term $V_i(k)\hat{m}_i(k)$ to the R.H.S.)

$$\tilde{x}_o(k+1/k) = x(k+1)-[\hat{x}(k+1/k) + U_i(k)\hat{m}_i(k)] + U_i(k)\hat{m}_i(k)$$

$$= x(k+1) - \hat{x}_i(k+1/k) + U_i(k)\hat{m}_i(k) \qquad (2.6.5)$$

$$= \tilde{x}_i(k+1/k) + U_i(k)\hat{m}_i(k)$$

we have

63

$$\tilde{x}_o(k+1/k)\tilde{x}_o'(k+1/k) \; = \; \tilde{x}_i(k+1/k)\tilde{x}_i'(k+1/k)$$

$$+ \; [\tilde{x}_i(k+1/k)\hat{m}_i'(k)U_i'(k) \; + \; U_i(k)\hat{m}_i(k)\tilde{X}_i'(k+1)]$$

$$+ \; U_i(k)\hat{m}_i(k)\hat{m}_i'(k)U_i'(k) \tag{2.6.6}$$

Taking the conditional expectation, given the observation sequence Y(k) and the hypothesis H_i, of both sides above, we get

$$E[\tilde{x}_o(k+1/k)\tilde{x}_o'(k+1/k)/Y(k),H_i]$$

$$= \; E\{[\tilde{x}_i(k+1/k)\tilde{x}_i'(k+1/k)/Y(k),H_i] \; + \; U_i(k)\hat{m}_i(k)\hat{m}_i'(k)U_i'(k)\} \tag{2.6.7}$$

The middle term in eq. (2.6.6) vanishes because $E[\tilde{x}_i(k+1/k)/Y(k),H_i]= 0$ and the last term can be taken out of the conditional expectation sign. From the properties of the conditional expectation for Gaussian random variables (Proposition 3 on p. 246 in [28]), the Y(k) dependence of the first term on the right hand side of eq. (2.6.7) can be taken out so that using eq. (2.6.2) we get

$$E[\tilde{x}_o(k+1/k)\tilde{x}_o'(k+1/k)/Y(k),H_i] \; = \; P_o(k+1/k) \; + \; U_i(k)P_i(k)U_i'(k)$$

$$+ \; U_i(k)\hat{m}_i(k)\hat{m}_i'(k)U_i'(k) \tag{2.6.8}$$

Therefore, the no-fail filter covariance can be incremented by the last two terms in the equation above. The last term represents the uncertainty due to the accumulated error in the no-fail filter arising from the failure. The preceding term signifies the uncertainty associated with the estimation of the failure state. In this method, the accumulated effects of the failure on the no-fail filter state estimate are not taken out; however, the additional uncertainty added to the no-fail filter covariance reflects the error accumulation due to the detected failure mode.

Each of the three approaches outlined above can, in principle, provide an estimate for the reinitialization "direction" and magnitude after the detection and isolation of a failure. The failure directions (provided by the estimate and/or covariance increment) can be used to selectively reinitialize those parts of the no-fail filter affected by the failure. The magnitude of the reset can be scaled to reflect any inherent uncertainty about it. In this way, a slightly stiffer reset can be generated, as a conservative measure. Our fault tolerant system uses Method III, (conditional covariance) for resetting the no-fail filter.

2.7 Failure Signature —— An Example

In this section an example problem will be analyzed in an attempt to better understand the failure signature information seen by the detectors. In the context of this example, the failure signature of bias, ramp and null failures on no-fail filter residuals will be discussed. Moreover, the inherent difference in detectability between the input and measurement sensors and the distinguishability of various sensor failures will be apparent by analyzing this simple example.

The example chosen is a special case of our problem involving scalar position, x, velocity v, and acceleration variables without nonlinear coordinate transformations. Within this framework, consider the second-order system

$$x(k+1) = x(k) + \tau v(k)$$

$$v(k+1) = v(k) + \tau a_m(k) + n_a(k) \qquad (2.7.1)$$

where τ is the sampling interval and the sensor measurements x_m, v_m and a_m are defined by

$$x_m(k+1) = x(k+1) + n_x(k+1) \qquad (2.7.2)$$
$$v_m(k+1) = v(k+1) + n_v(k+1)$$
$$a_m(k+1) = a(k+1) + n_a(k+1)$$

The measurement noises n_a, n_v, and n_x are white Gaussian sequences with known statistics. For ease of presentation, assume that the no-fail filter is implemented with constant gains by:

$$\hat{x}(k+1) = \hat{x}(k) + \tau\hat{v}(k) + k_{xx}r_x(k+1) + k_{xv}r_v(k+1) \qquad (2.7.3)$$

$$\hat{v}(k+1) = \hat{v}(k) + \tau a_m(k) + k_{vx}r_x(k+1) + k_{vv}r_v(k+1)$$

where k_{xx}, k_{xv}, k_{vx} and k_{vv} are the no-fail filter gains and the measurement residual sequences r_x and r_v are defined by:

$$r_x(k+1) = x_m(k+1) - [\hat{x}(k) + \tau\hat{v}(k)] \qquad (2.7.4)$$

$$r_v(k+1) = v_m(k+1) - [\hat{v}(k) + \tau a_m(k)]$$

We will now analyze the failure signature induced by a bias failure jump in each of these three sensors. Using the expression for the failure correction matrix, eq. (2.3.17) we get the following recursive relation for V_x (i.e., the failure correction matrix for the position sensor failure)

$$V_x(km) = \begin{bmatrix} 1-k_{xx} & \tau(1-k_{xx}) - k_{xv} \\ \\ -k_{vx} & -\tau k_{vx} + 1-k_{vv} \end{bmatrix} \qquad (2.7.5)$$

Using the expression above and the one for the failure measurement matrix, eq. (2.3.15) we get the following failure signature mean for two samples in the position and

velocity measurement residuals for a bias jump m_x at time k_o:

$$E[r_x(k_o)] = m_x \quad ; \quad E[r_v(k_o)] = 0 \tag{2.7.6}$$

$$E[r_x(k_o)] = (1-k_{xx}- k_{vx})m_x \quad ; \quad E[r_v(k_o+1)] = k_{vx}m_x$$

As seen above, a position sensor bias failure induces a jump in position measurement (with a level equal to the failure magnitude) and one sample later induces a jump in the velocity sensor with a level scaled by the no-fail filter gain k_{vv}. Based on this observation, several simple tests for position sensor failures can be posed. For instance, we can evaluate, on-line, the statistic

$$\gamma(k_o+1) = (r_x(k_o) + r_v(k_o+1)/k_{vx})/2 \tag{2.7.7}$$

to estimate the position sensor failure level m_x

$$E[\gamma(k_o+1)] = m_x$$

These types of open-loop generated statistics would be susceptible to accumulated errors. The detectors described in the previous section use the failure signature information above in an optimal closed-loop manner (accounting for noise statistics and dynamics) to estimate the failure levels. Similarly, for a bias jump failure in the velocity sensor v_m with magnitude m_v, at time k_o, we have the following signature on the residuals:

$$E[r_x(k_o)] = 0 \quad ; \quad E[r_v(k_o)] = m_v \tag{2.7.8}$$

$$E[r_x(k_o+1)] = -(1+\tau)k_{xv}m_v \quad ; \quad E[r_v(k_o+1)] = (1-k_{vv})m_v$$

On the other hand, a bias jump failure in the accelerometer sensor a_m with magnitude m_a at time k_o induces the following signature mean time history on the measurement

67

residuals:

$$E[r_x(k_o)] = 0 \quad ; \quad E[r_v(k_o)] = 0 \tag{2.7.9}$$

$$E[r_x(k_o+1)] = 0 \quad ; \quad E[r_v(k_o+1)] = -m_a$$

$$E[r_x(k_o+2)] = (k_{xv} + (k_{vv}-1)^2)m_a$$

$$E[r_v(k_o+1)] = (k_{vv}-2\tau)m_a$$

Comparison of the input sensor (accelerometer) failure signature above with that measurement of sensors (position and velocity) shows the inherent delay in failure signature generation for input sensors. This is because an input sensor failure must propagate through the no-fail filter dynamics in order to generate failure signature on the measurement residuals. We also note the similarity of the signatures generated by accelerometer and velocity sensor failures. For instance, an accelerometer bias failure with level m_a looks like a velocity sensor failure with failure level $- m_a$. Moreover, if we had the unfortunate choice for no-fail filter gains such that the relations

$$-(1+\tau)k_{xv}(-\tau m_a) = (k_{xv}\tau + (k_{vv}-1)\tau^2)m_a$$

$$(1-k_{vv})(-\tau m_a) = (k_w - 2\tau)m_a \tag{2.7.10}$$

were approximately satisfied (choosing $k_{vv} = \tau/(1-\tau)$ and $k_{xv} = k_{vv} - 1$ would exactly satisfy them), then it would be impossible to distinguish between the velocity and accelerometer failures by looking at this failure signature time history. This simple example clearly shows how the choice of no-fail filter gains affect the distinguishability of various sensor failures. In this case, the next sample of the failure signatures involving terms including the gains k_{xx} and k_{vx} would provide the

necessary distinguishability information.

We also note that the residual signatures will eventually converge to finite steady-state values since the recursive relation for the failure correction matrix given by eq. (2.7.5) is governed by the stable closed-loop filter transition matrix.

Although we have modelled sensor failures as bias jumps, other types of failures also manifest themselves approximately as time varying biases. For instance, consider a ramp failure in position sensor, at time k_o with level s_x ($k \neq k_o$). In this case, the induced failure signature would be given by (using the expression for the time-varying failure levels in [15]):

$$E[r_x(k_o)] = 0 \quad ; \quad E[r_v(k_o)] = 0 \qquad (2.7.11)$$

$$E[r_x(k_o+1)] = s_x \tau \quad ; \quad E[r_v(k_o+1)] = 0$$

$$E[r_x(k_o+2)] = (2-k_{xx}-\tau k_{vx})s_x \quad ; \quad E[r_v(k_o+2)] = k_{vx}s_x \tau$$

A hardover failure in position sensor at time k_o with a level of h_x will approximately result in the following failure signature:

$$E[r_x(k_o)] = h_x - \bar{x}(k_o) \quad ; \quad E[r_v(k_o)] = 0 \qquad (2.7.12)$$

$$E[r_x(k_o+1)]=(1-k_{xx}-\tau k_{vx})h_x - \bar{x}(k_o) \quad ; \quad E[r_v(k_o)] = k_{vx}(h_x - \bar{x}(k_o))$$

where $\bar{x}(k_o) = E[x(k_o)]$ and assuming $\bar{x}(k_o) = \bar{x}(k_o+1)$. These relations follow since a position sensor failure with level h_x at time k_o is approximately equivalent to a bias failure with level $h_x - \bar{x}(k_o)$.

Similarly, a null failure in the velocity sensor at time k_o will approximately result

in the following failure signature mean time history:

$$E[r_x(k_o)] = 0 \quad ; \quad E[r_v(k_o)] = -\bar{v}(k_o) \tag{2.7.13}$$

$$E[r_x(k_o+1)] = (1+\tau)k_{xv}\bar{v}(k_o) \quad ; \quad E[r_v(k_o+1)] = (1-k_{vv})\bar{v}(k_o)$$

where it is assumed that $\bar{v}(k_o) = \bar{v}(k_o+1)$. These relations follow since a null failure in the velocity sensor at time k_o is approximately equivalent to a bias failure with level $-\bar{v}(k_o)$.

3. FTS PERFORMANCE EVALUATION

In this chapter estimation and failure detection performance of the developed integrated avionics sensor fault tolerant system will be discussed. The discussions will be centered around specific simulation runs, which point out characteristic traits of the FTS, rather than ensemble statistics (although the simulation was carried out in a Monte Carlo fashion). The computer program used for these simulations is called FINDS and is documented in [2]. Current performance of the FTS will be empirically examined in this chapter with regard to:

o Reliability of no—fail filter state estimates, i.e. "fault tolerance"

o Speed of detection and isolation.

o Failure distinguishability between dynamically related sensors.

o Robustness of the method for detection of non—bias failures.

o Use of navigation quality RSDIMU in lieu of flight quality body mounted
   accelerometers, rate gyros, and platform IMU.

The organization of the chapter is as follows: Section 3.1 outlines the format and goals of the simulation study. Its two subsections help the reader understand the later results by detailing the simulation parameters used, and the performance measures which will be employed, respectively. Typical FTS performance when failures are not simulated can be found in Section 3.2 along with a description of the nominal FTS parameters used in the study. The next section, 3.3, discusses performance under singleton and simultaneous multiple bias failures. Bias failure performance is reported in this section for both a "standard" (flight control quality) sensor configuration, and an RSDIMU configuration (navigation & flight control quality sensors). Section 3.4 shows how well the FTS works with non—bias type failures. Finally, the chapter closes

71

with Section 3.5 which summarizes the results of the chapter and gives an overall evaluation of the current system.

3.1 FTS Evaluation—Overview

A simulation study was performed to empirically determine the capabilities of the analytically derived fault tolerant system, developed in Chapter 2. The study had several goals — some of which were mentioned in the previous section. Here we would like to review the ground rules for this study, discuss the simulation and filter parameters used, and the performance measures to be employed.

The simulation environment used to test the developed FTS is provided by NASA's six degree of freedom nonlinear digital simulation of the TSRV research aircraft. The original program (supplied by NASA-LRC) was suitably modified to include realistic sensor and failure models (see [2]). Our simulation study uses this program in the terminal area — under MLS coverage only. Moreover, the fault tolerant estimates generated by the no-fail filter are used by a fully automatic landing system — in a closed loop fashion — to land the aircraft along a prescribed path. Therefore, our simulation study is concerned with detecting and isolating failures in sensors and providing fault tolerant aircraft state estimates to the automatic landing system in the terminal area. It is assumed that the reader is familiar with this problem (from reading [1], [2], [4], [5], or [6]).

The simulation study was performed by:

o implementing the system described in Chapter 2 — making little or no simplifying assumptions.

72

o using an update rate for all portions of the FTS of 20 Hz (i.e. a sample time of 0.05 seconds).

o simulating the normal operating characteristics of the sensors (i.e., bias, noise, scale factor, misalignment, etc.) — while using a simple bias plus noise model in designing the FTS.

o simulating the many failure modes for the sensors (i.e., bias null, hardover, ramp, increased noise, etc.) — while using a simple bias jump failure model in designing the FTS.

o estimating only selective normal operating biases — even though all sensors have some form of constant bias term in their outputs.

o simulating failures in the components of the RSDIMU (when used) are not detected by the FTS, nor are they considered in this study — instead the FDI techniques resident in the RSDIMU are used (see [23] for a description of these techniques).

These rules were adopted in order to evaluate the robustness of the FTS under simple internal models. Simplifying assumptions (even straightforward ones) were not made so that assumptions about one mode of operation (say with bias failures) would not impact performance under another (for example, with increased noise failures). In this way, the robustness of the original method would be examined rather than an unintended specialized variant of it. Moreover, by postponing these practical decisions they could then be made based on a broad experience base.

In addition to the basic rules given above, the particular runs chosen to be included in the study had the following general characteristics:

o Multiple isolated failures were simulated in each run to save on computer resources, and to see if detectability of the remaining sensors would be affected.

o A bias failure was simulated for every sensor at several points in the standard flight trajectory (for example, along a perpendicular path as well as a tangential path relative to the runway; or on a straight and level path, as well as a maneuvering one).

o Failures were randomized with respect to

    . time of failure

    . level of failure

    . sensor replication

o Each run was made in a Monte Carlo fashion — in that a different random seed was used for every run (therefore, filter initial conditions, bias levels, noise time histories, etc. were all randomized).

The Monte Carlo simulation approach was used to more faithfully test the FTS and explore the strengths and shortcomings of it. It also provides a good base of runs which can be incorporated with future ones to obtain average performance measures, and probabilities. By simulating failures at various points within the flight path, we can empirically examine the effects of flight path on failure identifiability and detectability.

The next subsection defines the nominal simulation used throughout this study. It also displays the nominal flight path along with other important flight profiles.

3.1.1 Simulation Description

The objective of this sub-section is to define the simulation parameters used in the study and to describe the nominal simulation environment — as it effects the FTS.

The simulation runs all start slightly before the point of transition to MLS coverage and end at touchdown on the runway. Sampling frequency for the simulation is 20 Hz. Figures 5-9 show the simulated aircraft state time histories for a typical simulation run. These figures are intended to be used for reference purposes. In the

74

rest of the simulation runs to be presented, these time—histories will not be re—shown. They may, in fact, be slightly different, since the no—fail filter state estimates are used by the automatic guidance and control laws to land the aircraft. In other words, under different noise, wind, failure, etc. conditions, the corresponding true variable time—histories may differ somewhat from the nominal.

Figure 5 shows the A/C ground track and altitude profile. This figure clearly shows the sequencing of the various flight segments as the A/C performs its automatic approach and landing. In addition, the direction of the simulated horizontal wind (165 degrees from north) is displayed. The magnitude of the wind was a constant 30 knots in all sample runs considered in this study. The simulation runs all start under VORTAC system coverage without the FTS. At approximately 35 seconds, MLS coverage begins and the FTS program (no—fail filter, bank of detectors, decision logic, etc.) is initiated. Figure 5 shows the initial ground track oriented roughly 45° relative to the runway. A banking maneuver is executed (from 55—100 seconds) which brings the flight track perpendicular to the runway. A second bank maneuver occurs 30—40 seconds thereafter which aligns the flight track with the runway. This flight segment runs to approximately 215 seconds. Glide slope capture occurs at approximately 180 seconds. Between 255 seconds and touchdown (approximately 275 seconds), decrab and flare maneuvers are executed. Also, MLS elevation measurements are replaced by altimeter measurements in the FTS at around 260 seconds. The flight path can be summarized by the following mapping (which will be convenient to refer to in later discussions):
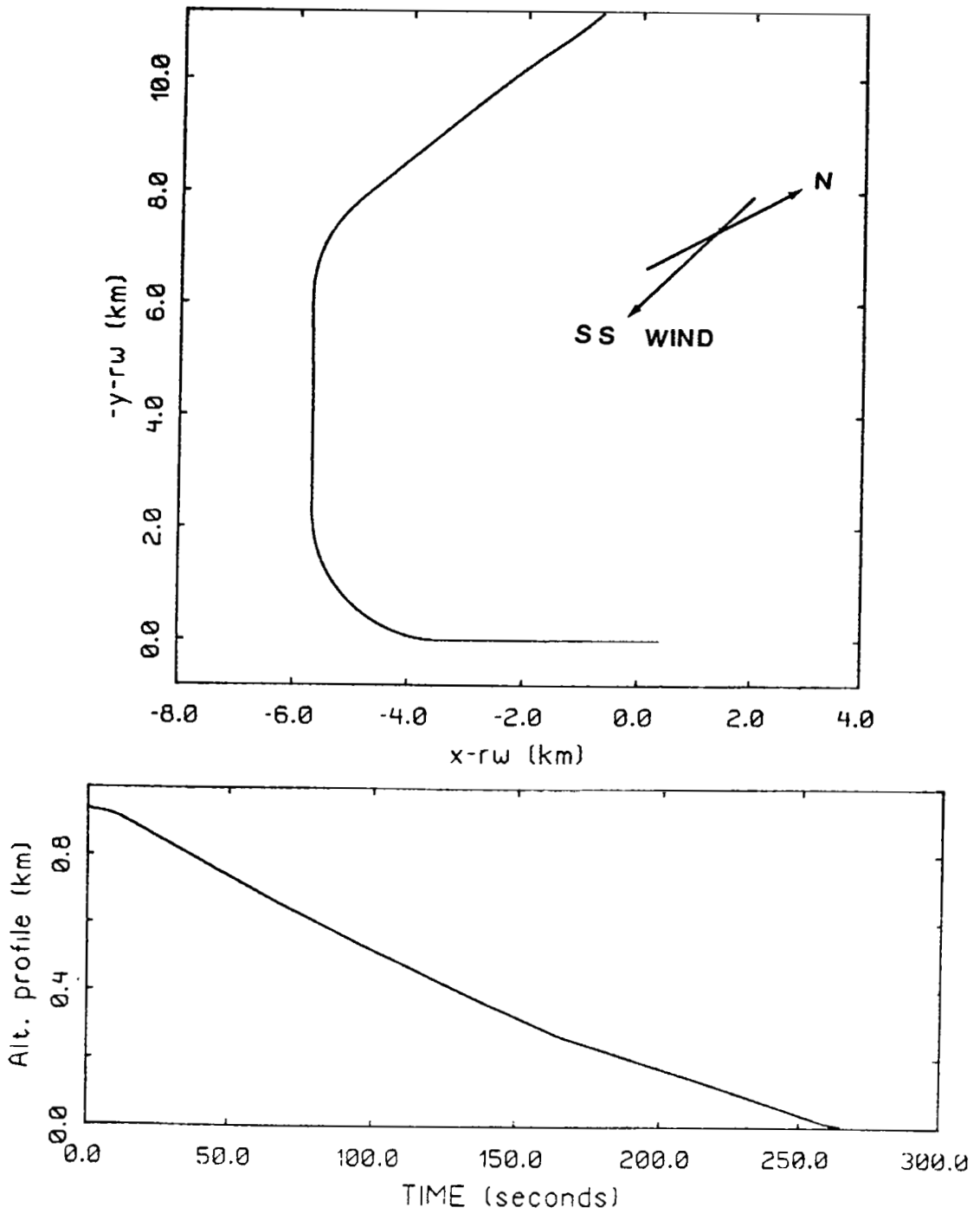
FIG. 5. TYPICAL A/C GROUND TRACK AND ALTITUDE PROFILE
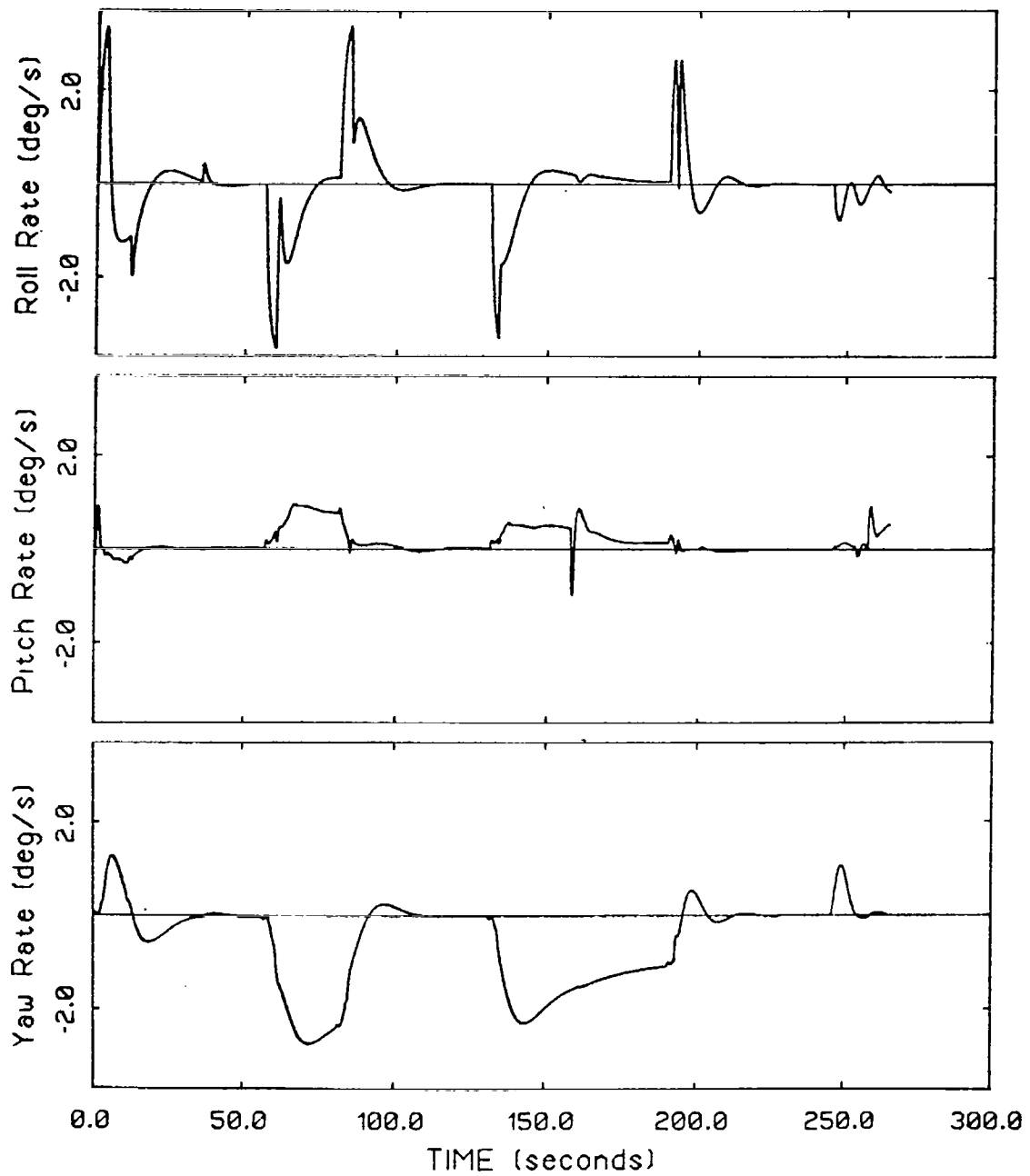
FIG. 6. TYPICAL ANGULAR BODY RATE PROFILES

77
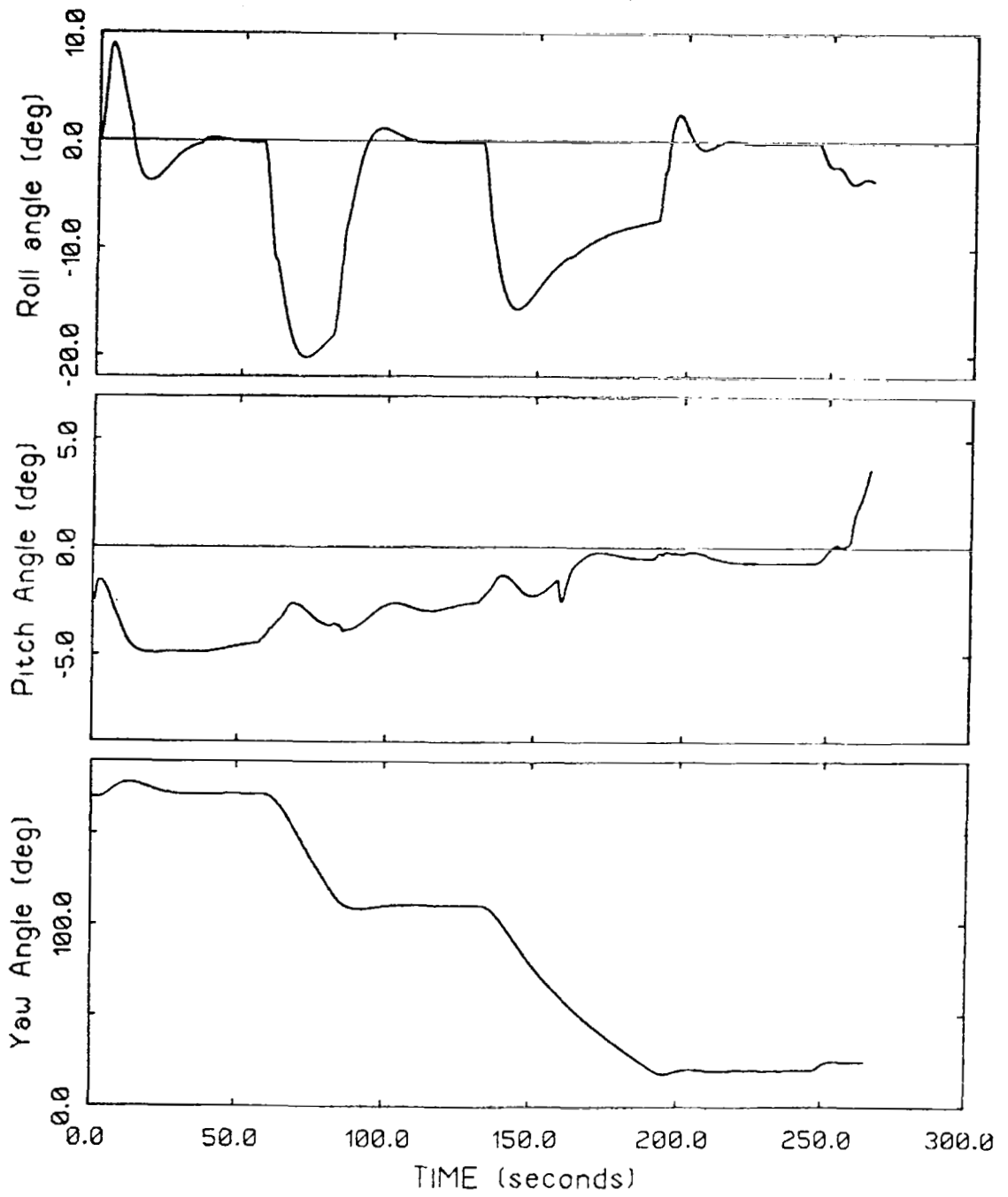
FIG. 7. TYPICAL EULER ANGLE PROFILES

78
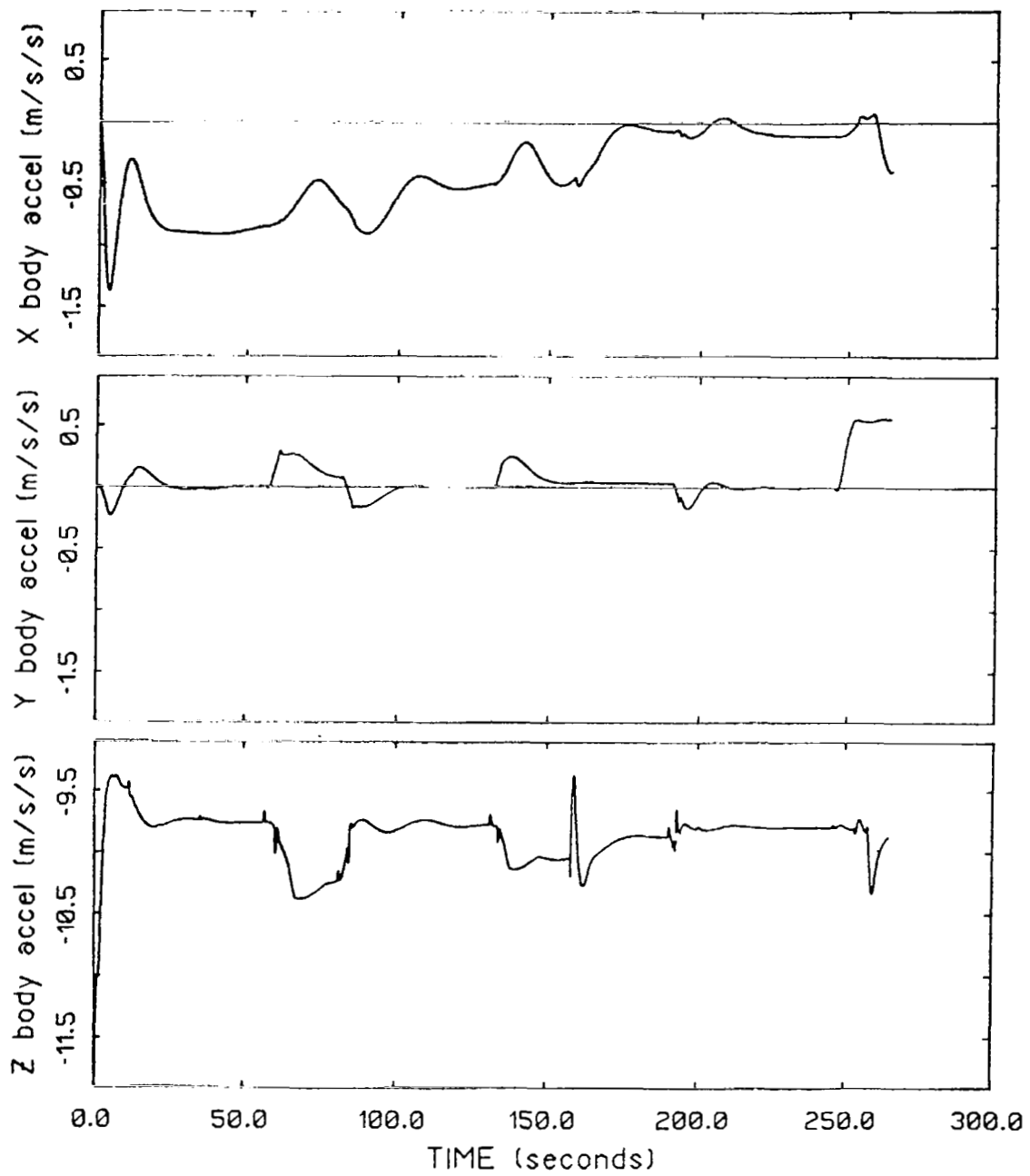
FIG. 8. TYPICAL BODY ACCELERATION PROFILES
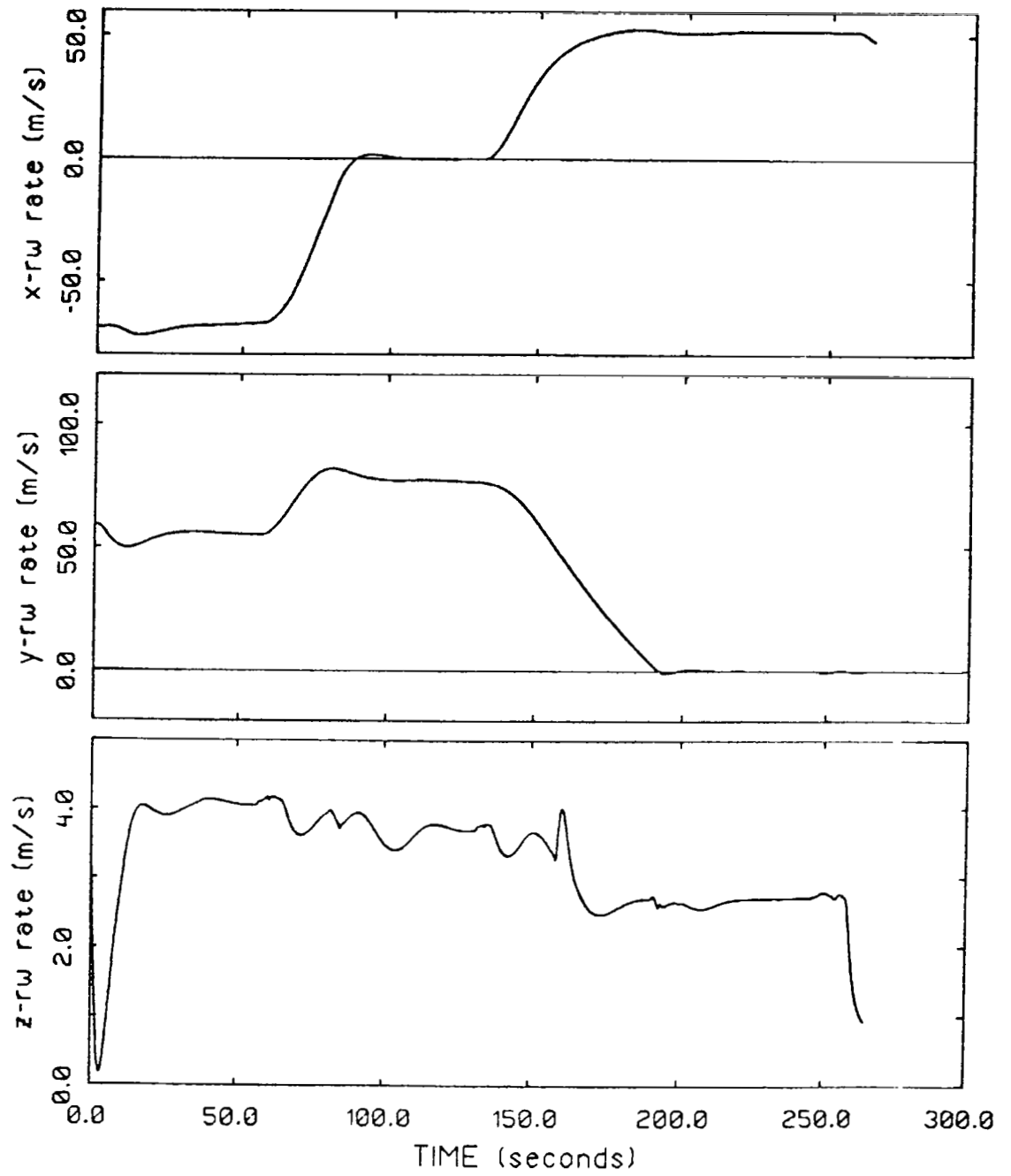
79

FIG. 9. TYPICAL A/C VELOCITY PROFILES

o time = 35—>55;   A/C path is neither perpendicular

                               nor parallel to runway

o time = 55—>100, & 130—>200; A/C is maneuvering

o time = 100—>130;   A/C path is perpendicular to runway

o time = 200—>touchdown;   A/C path is parallel to runway

The altitude profile curve shows the nearly constant sink rate of approximately 3 meters/s.

The maneuvers, shown in the ground track of figure 5 can be seen in more detail in the next four figures. The corresponding attitude and body attitude rate variations for this typical simulation run are displayed in figure 7 and 6, respectively. The following attributes are evident in these figures:

o Yaw angle has the largest amplitude range, and the lowest frequency content
   (i.e. changes occur more gradually and smoothly)

o Pitch angle has the smallest amplitude range — and is also fairly smooth,
   with some regulation evident during the roll angle transient.

o The roll angle profile shows the most variations. It also contains several
   periods where the roll angle is very nearly zero.

o All three body attitude rates are plotted using the same scale.

o Roll rate has the largest and sharpest transients, but the duration of each
   is typically confined to a 5–20 second period.

o Yaw rate, on the other hand, is smooth, but its duration at significant levels
   is much longer (30–70 seconds).

The implications of these variations are as follows:

o The effects of these attitude rate variations on the FTS will be felt as unmodeled errors due to misalignment, and scale factor errors — each of which is a function of the signal level.

o Both attitude and attitude rate profiles clearly show periods of large and small signal levels. The detectability of failures of varying types and levels during these periods is an important issue.

Typical body acceleration profiles experienced by the aircraft are found in figure 8. All three trajectories are plotted on the same relative scale — so that they can be compared. Notice, however, that the vertical acceleration is biased by the gravitational acceleration. Basically, these curves show the aircraft decelerating as it nears touchdown. The affects of cross—axis coupling on the lateral airframe dynamics can be seen in the lateral acceleration curve. Since body accelerometer measurements are inputs to the FTS, and since they include both scale factor and mounting misalignment errors, the absolute level of each of these curves should be kept in mind when interpreting the later results.

Since the FTS operates in a runway based coordinate frame, the mapping between body quantities and runway quantities is important since this transformation effects failure observability. Figure 9 shows the A/C velocity components in the runway frame. It can also be viewed as the transformation of the body accelerations (properly integrated) into the runway frame. Note in particular, the flight segments where the forward body velocity is either all in the x or y runway direction. During these periods, not only the signal, but also much of the modeling errors will be polarized in these directions.

Table 1 shows the sensor model parameter values used in the simulation. A dual, redundant, sensor configuration is used. For each sensor type, the standard deviation of the sensor measurement noise, normal operating bias and scale factor error is given, along with the associated stop limits. The units for the parameter values are given in the second column − except for scale factor which is expressed as a percent, and IAS measurement noise, which is multiplicative, and also expressed as a percent. As discussed earlier, MLS measurement noises are time−correlated with a steady−state rms value specified in Table 1. Furthermore, body−mounted instruments are misaligned with respect to the body axis through a random transformation. The standard deviation of the random misalignment is 0.4° for rate gyros and 0.36° for accelerometers. The normal sensor model parameters values for the RSDIMU, when it is used, is shown in Table 2.

3.1.2 Performance Measures

The performance of the FINDS program has been determined by making various runs under the following sensor failure conditions, listed in approximate order of increasing deviation from the underlying the FTS design assumptions:

o singleton bias failures

o multiple bias failures

o hardover failures

o null failures

o ramp failures

o increased scale factor failures

o increased noise failures

TABLE 1.   NOMINAL SENSOR MODEL PARAMETERS

SENSOR TYPE	UNITS	NOISE LEVEL	BIAS LEVEL	SCALE FACTOR	MISALIGNMENT ANGLE	SYMMETRIC OUTPUT LIMITS
A_x, A_y, A_z	m/s/s	.098	.098	.25%	$.40^0$	4.9,4.9,19.6
P,Q,R	deg/s	.02	2.8E-5	.01%	$.36^0$	100
Azm,El	deg	.03	.03	N/A*	N/A	N/A
Rng	m	3.0	4	N/A	N/A	N/A
IAS	m/s	2%(4.5)***	1.0	N/A	N/A	205/54**
ϕ, θ, ψ	deg	.23	.08	N/A	N/A	80,80,600
RA	m	.305	.305	N/A	N/A	N/A

* - not applied
** - two sided (asymmetric) stop limits
*** - percent (absolute)

84

TABLE 2.   NOMINAL RSDIMU SENSOR MODEL PARAMETERS

	RATE GYROS (deg/Hr)	LINEAR ACCELEROMETERS (g's)
Noise	0.125	1.25 E-5
Bias	0.015	1.0 E-4
Scale Factor (ppm)	75	75
Misalignment (deg)	3.333 E-3	3.333 E-3
G-sensitive drift (deg/Hr)	0.015	0.015

In all of the failure conditions above, except for the null, hardover, and ramp cases, it will be convenient to define a failure level with respect to the underlying normal operating error level. For instance, if the standard deviation of the normal operating bias for a given sensor is σ, then a 5σ bias failure for that sensor will signify that the failure level will come from a distribution with a standard deviation of 5σ. Similarly, the σ in a 10σ increased noise failure will be the standard deviation of the measurement noise associated with that sensor. A more detailed description of the sensor failure models can be found in [1] and [2].

Although repetitive runs for a given sensor failure type under varying random conditions have been made, the collected data was not sufficient to compute experimental false alarm and probability of detection figures. On the other hand, the repetitive runs, where available, have been used in obtaining an average "time—to—detect" and "time—to—heal" figure.

The primary way of evaluating performance will be to examine the actual time histories of the no—fail filter state estimation errors and covariances. The information contained in these plots, much of which would be lost in forming ensemble statistics, is particularly useful in providing insight into the operation of the FTS. In particular:

o The transient effects of the failures can be seen.

o The effects of using biased measurements are evident

o The interaction of the bias filter and choice of biases to be identified can be viewed

o The effects of the healer and re—configuration logic can be gauged.

o How the flight path effects the no—fail filter and the propagation of failures can also be seen.

Time histories will be presented as an estimation error profile, with the no—fail filter's a posteriori (e.g. after the measurement update) covariance envelope.

When viewing these error profiles note that although absolute performance of the no—fail filter is important, the filter serves a dual purpose. On one hand, it's function is to provide accurate estimates for the A/C's current state. While on the other hand, it must perform data fusion in such a way that all sensors play a role in the estimation of those same A/C states. This last function is critical if failures in individual sensors are to be detectable. Since, at times, these two functions are clearly in conflict, no—fail filter parameters were chosen to satisfy both goals.

3.2 Performance with No Failures

In this section, the performance of the FTS no—fail filter described in Section 2.2 will be discussed using the results obtained on the six—degree—of—freedom nonlinear simulation described in Section 3.1. The inherent modelling error in the no—fail filter, arising from the assumptions made in the design were discussed in Section 3.1.1. Before we begin, however, we will need to specify the specific parameters used in the FTS modules.

Initial errors for the state and normal operating bias estimates are randomized. Their one sigma levels at FTS start—up time are given in Table 3. This table also gives the standard deviation of the initial uncertainty for these variables. Furthermore,

accelerometer and rate gyro biases are normally estimated by the no-fail filter. Normal accelerometer bias levels were large enough that their compensation was necessary, however, normal rate gyro biases are very small. Rate gyro biases were estimated to help eliminate false alarms associated with scale factor and misalignment modeling errors.

Process and measurement noise statistics used by the no-fail filter are given in Table 4. Notice that the noise levels assumed by the filter are given in a per replication manner – to reflect the fact that the actual process and measurement noise statistics used depend on the replications of the corresponding sensors. Further note that the wind process noise doesn't relate to a physical sensor and therefore is replication independent.

FTS detector parameters are summarized in Tables 5 and 6. Table 5 gives the a priori probability of failure, detector estimation window length, and standard deviations of the estimation information used for detector resetting. Table 6 shows the bias and failure levels used in the healer module. The healer window length used was 3 seconds in all runs.

Figure 10 shows the no-fail filter position estimation error time histories for a typical run. Each plot is made up of two parts: an error time history – the solid line; and an a posteriori covariance envelope – the two symmetrical dashed lines. Also, keep in mind that the covariance envelope represents the a posteriori covariance (e.g. after processing the measurements), and therefore does not reflect the level used in forming the Kalman gains or the innovations covariance used by the detectors. The

TABLE 3.   NO—FAIL FILTER STATE INITIAL CONDITIONS

Variable	Est. Error (S.D.)	Uncertainty (S.D.)	Units
States:			
x-rw	1.5000E+01	4.0000E+01	m
y-rw	1.5000E+01	4.0000E+01	m
z-rw	5.0000E+00	3.0000E+01	m
x-dot-rw	1.5000E+00	4.0000E+00	m/s
y-dot-rw	1.5000E+00	4.0000E+00	m/s
z-dot-rw	1.5000E+00	1.2500E+00	m/s
phi	1.0000E-01	5.0000E-01	deg
theta	1.0000E-01	5.0000E-01	deg
psi	2.0000E-01	1.5000E+00	deg
x-wind-rw	5.0000E-01	7.5000E-01	m/s
y-wind-rw	5.0000E-01	7.5000E-01	m/s
Ave. Biases:			
x-accel	1.0000E-01	3.0480E-01	m/s/s
y-accel	1.0000E-01	3.0480E-01	m/s/s
z-accel	1.0000E-01	3.0480E-01	m/s/s
P-gyro	2.8000E-05	2.5600E-01	deg/s
Q-gyro	2.8000E-05	2.5600E-01	deg/s
R-gyro	2.8000E-05	2.5600E-01	deg/s

TABLE 4.   NO—FAIL FILTER PROCESS AND MEASUREMENT NOISE LEVELS

Variable	Noise S.D. Per Repl.	Replications Used	Units
Process Noises:			
x-accel	9.8066E-02	1	m/s/s
y-accel	9.8066E-02	1	m/s/s
z-accel	9.8066E-02	1	m/s/s
P-gyro	8.9400E-02	1	deg/s
Q-gyro	8.9400E-02	1	deg/s
R-gyro	8.9400E-02	1	deg/s
x-wind-rw	0.0000E+00	N/A	m/s
y-wind-rw	0.0000E+00	N/A	m/s
Measurement Noises:			
MLS azim	6.0000E-02	2	deg
MLS el	6.0000E-02	2	deg
MLS rng	6.0000E+00	2	m
IAS	1.4919E+00	2	m/s
IMU phi	2.5100E-01	2	deg
IMU theta	2.5100E-01	2	deg
IMU psi	2.5100E-01	2	deg
Radar alt	3.0480E-01	0	m

TABLE 5.   DETECTOR RESET PARAMETERS

Detector Window Length = 1.00000 Seconds

Sensor Type	Apriori Probability	Est. Window (sec)	Estimation Information (S.D.)	Units
Singular Failures:				
x-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
y-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
z-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
P-gyro	4.6000E-09	3.00000	0.0000E+00	deg/s
Q-gyro	4.6000E-09	3.00000	0.0000E+00	deg/s
R-gyro	4.6000E-10	3.00000	0.0000E+00	deg/s
MLS azim	3.5000E-09	1.00000	0.0000E+00	deg
MLS el	3.5000E-09	1.00000	0.0000E+00	deg
MLS rng	3.5000E-09	1.00000	0.0000E+00	m
IAS	3.1500E-08	1.00000	0.0000E+00	m/s
IMU phi	2.2500E-08	1.00000	0.0000E+00	deg
IMU theta	2.2500E-08	1.00000	0.0000E+00	deg
IMU psi	2.2500E-09	1.00000	0.0000E+00	deg
Radar alt	3.5000E-10	1.00000	0.0000E+00	m
Simultaneous Multiple Failures:				
MLS azim	1.2250E-14			
MLS el	1.2250E-14			
MLS rng	1.2250E-14			

TABLE 6.   DETECTOR HEALER PARAMETERS

Healer Window Length = 3.00000 Seconds

Sensor Type	Bias Est Thresold	Failure Est Threshold	Units
x-accel	2.0000E-01	5.0000E-01	m/s/s
y-accel	2.0000E-01	5.0000E-01	m/s/s
z-accel	2.0000E-01	5.0000E-01	m/s/s
P-gyro	1.2000E-03	1.2000E-02	deg/s
Q-gyro	1.2000E-03	1.2000E-02	deg/s
R-gyro	1.2000E-03	1.2000E-02	deg/s
MLS azim	6.0000E-02	1.5000E-01	deg
MLS el	6.0000E-02	1.5000E-01	deg
MLS rng	1.2000E+01	2.0000E+01	m
IAS	1.2000E+00	2.5000E+00	m/s
IMU phi	2.0000E-01	5.0000E-01	deg
IMU theta	2.0000E-01	5.0000E-01	deg
IMU psi	2.0000E-01	5.0000E-01	deg
Radar alt	6.1000E-01	1.5000E+00	m
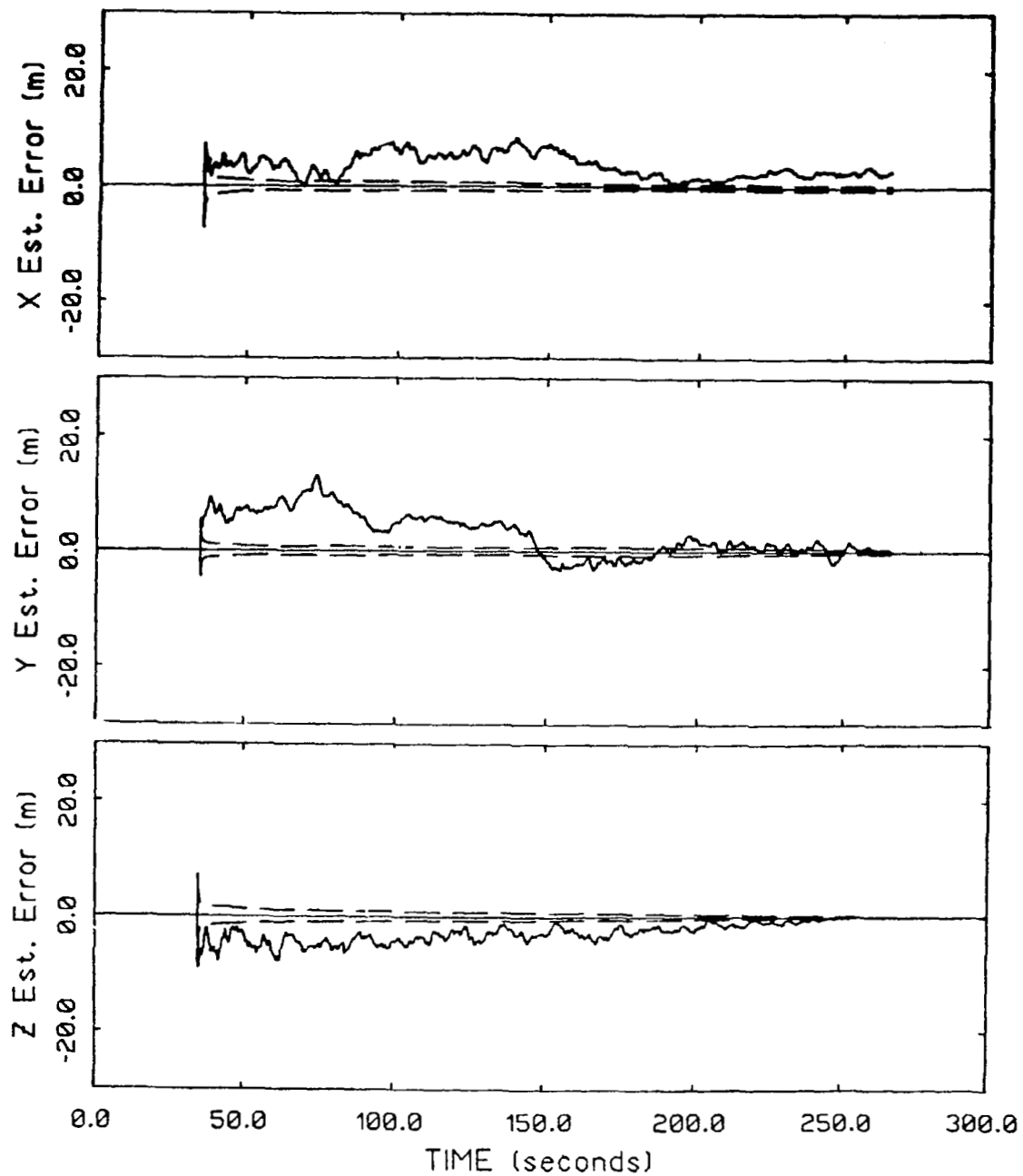
FIG. 10. POSITION ESTIMATION ERROR – NO–FAILURES

91
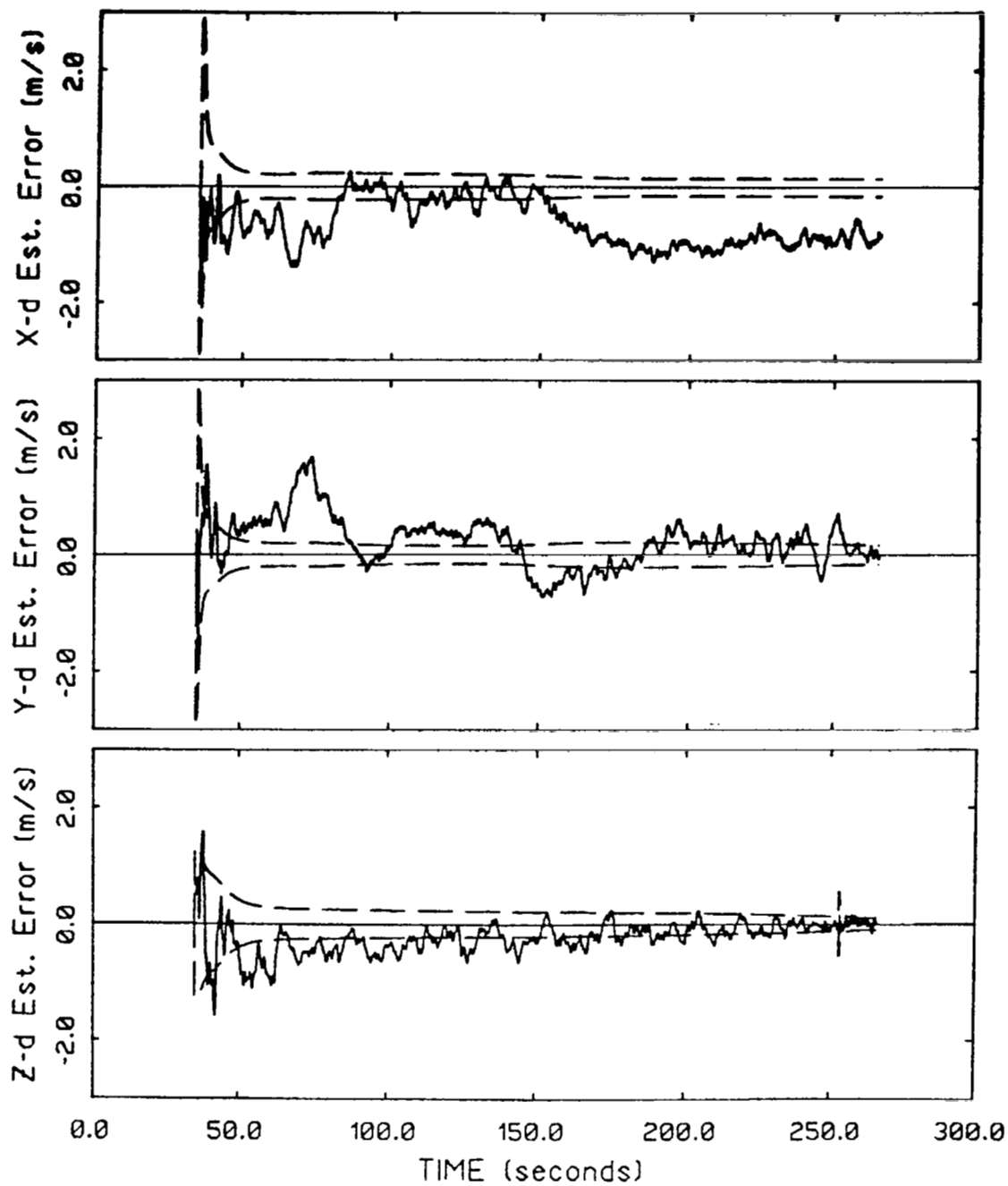
FIG. 11. VELOCITY ESTIMATION ERROR — NO—FAILURES

92
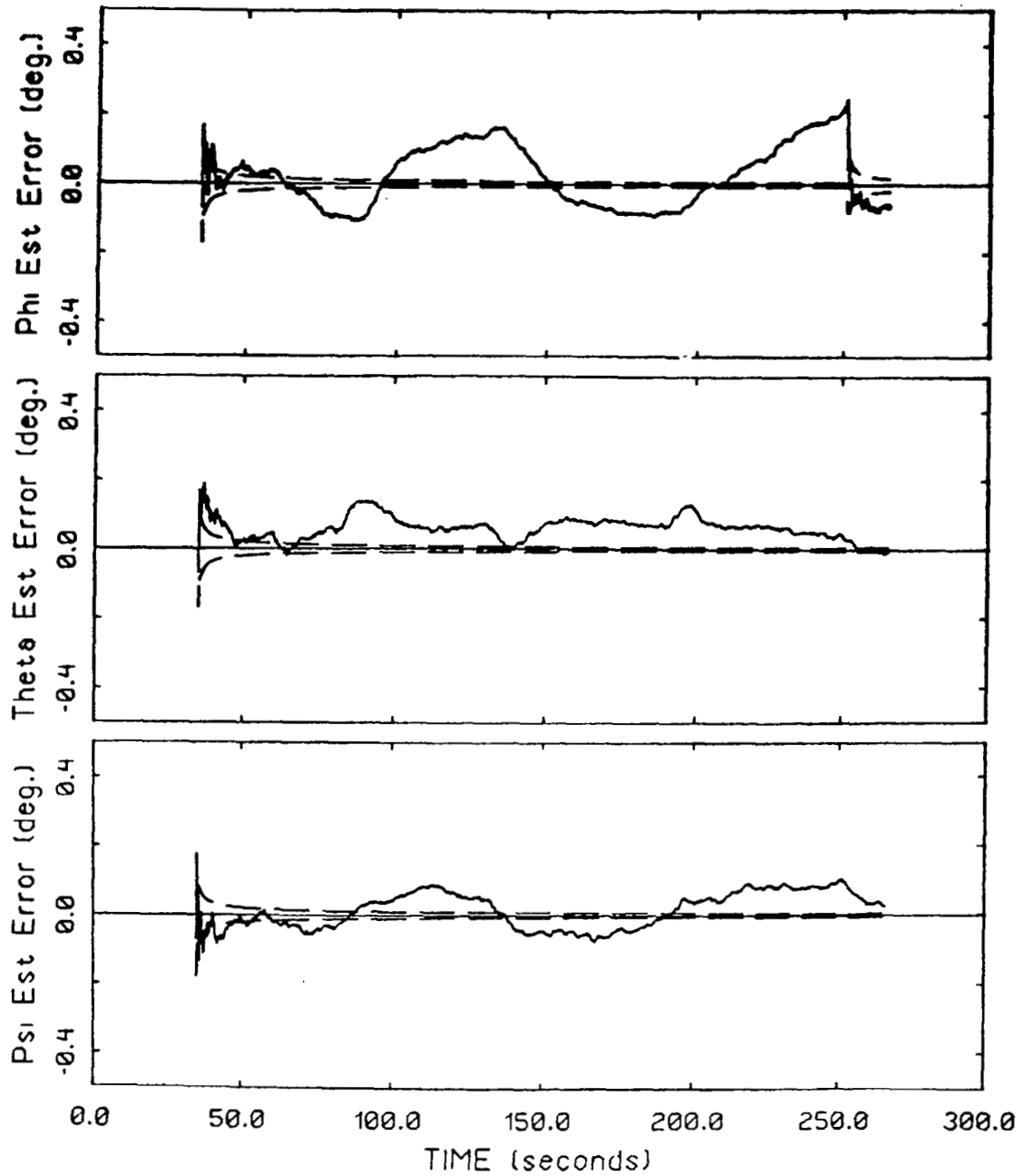
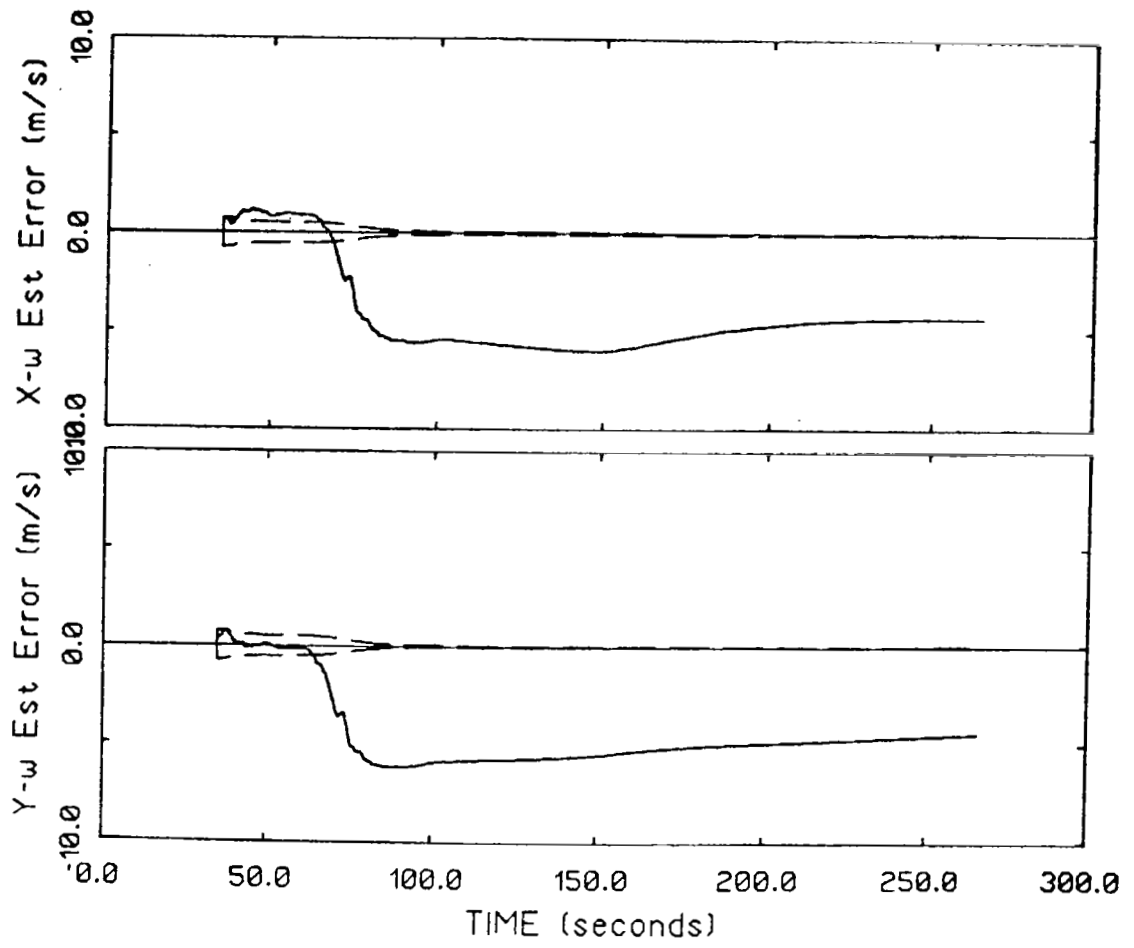FIG. 12. ATTITUDE ESTIMATION ERROR – NO–FAILURES

93

FIG. 13. HORIZONTAL WIND ESTIMATION ERROR - NO-FAILURES
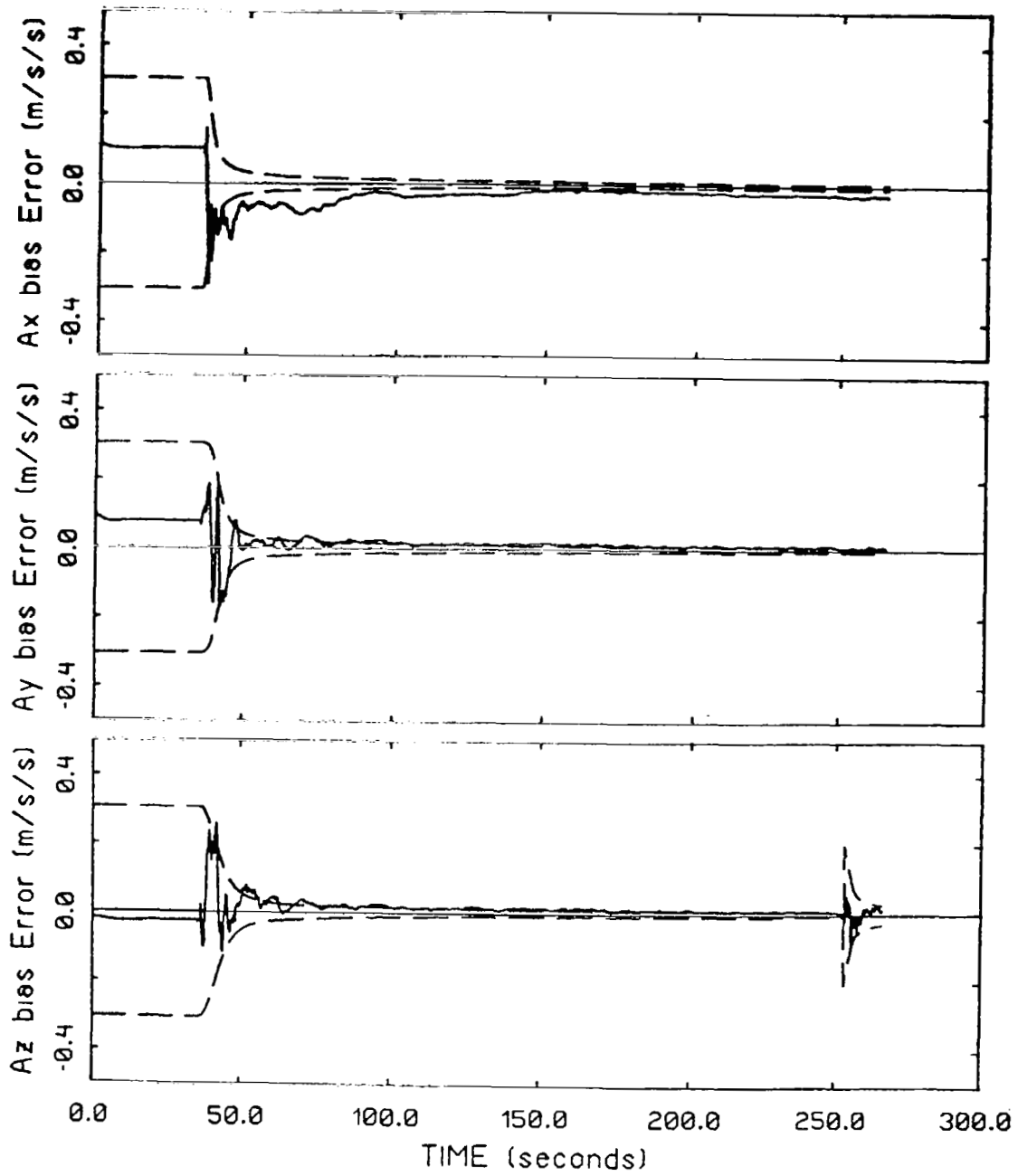
94

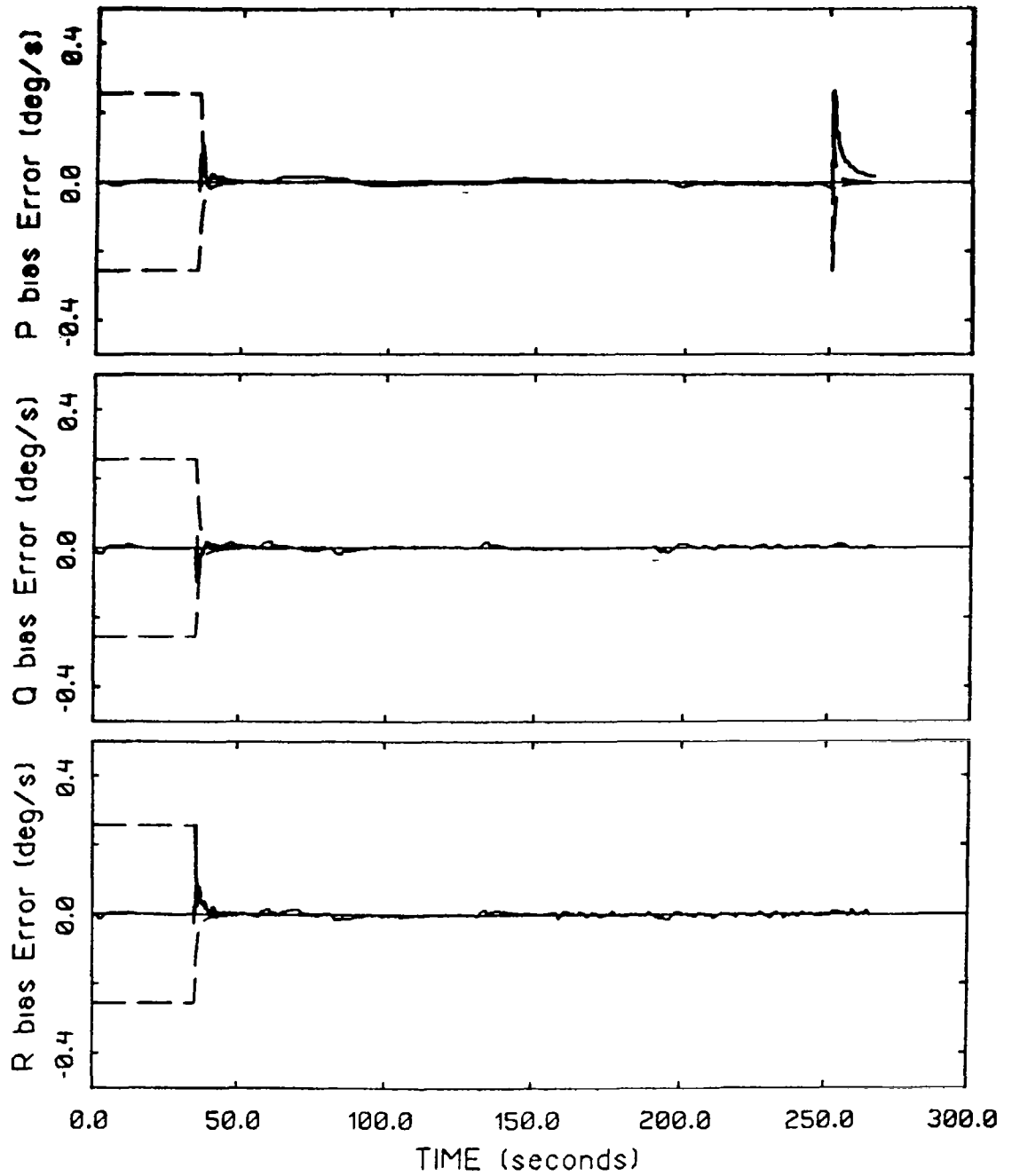FIG. 14. ACCELEROMETER BIAS ESTIMATION ERROR – NO–FAILURES
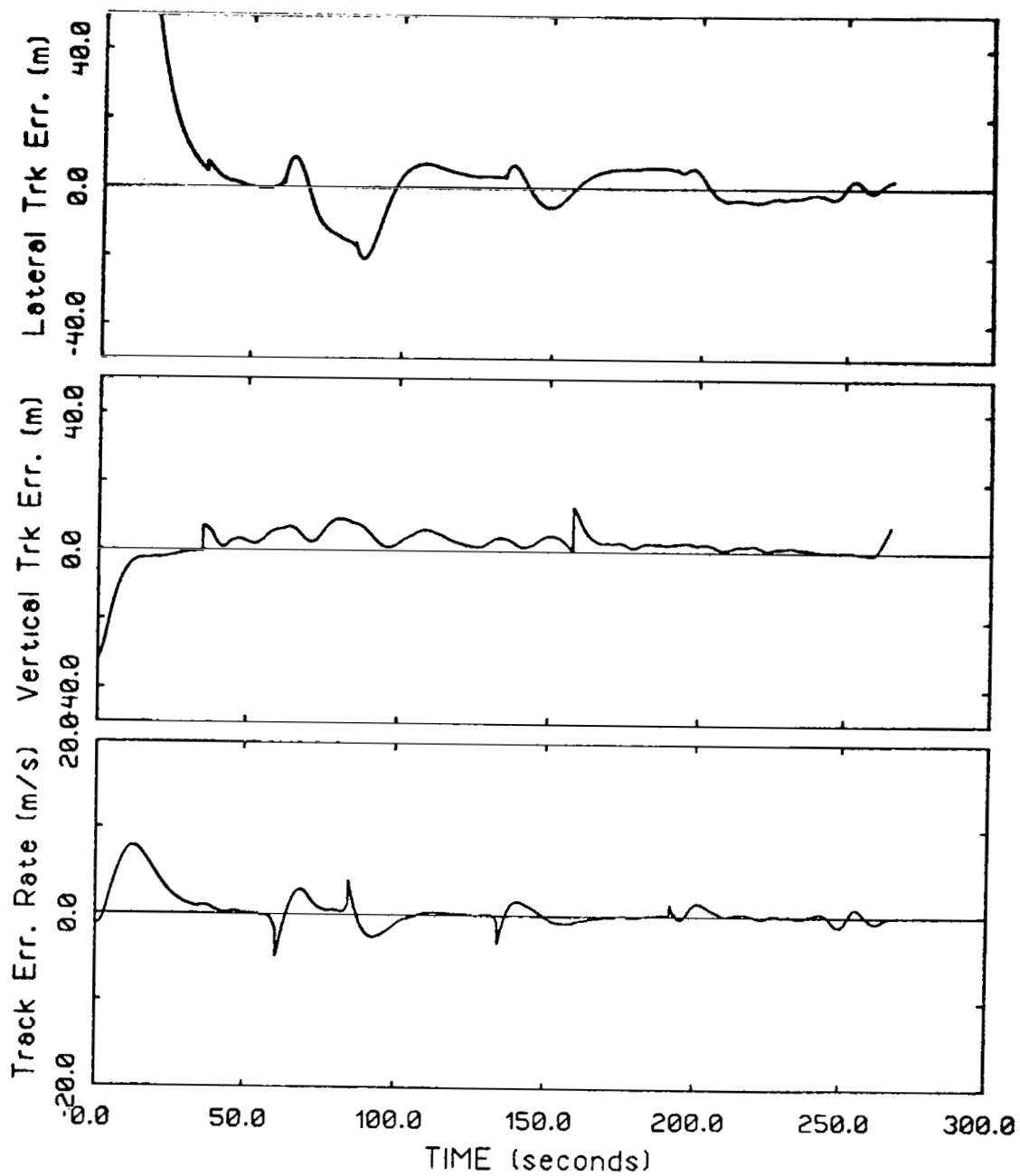
95

FIG. 15. RATE GYRO BIAS ESTIMATION ERROR - NO-FAILURES

96

FIG. 16. TRUE A/C TRACK ERRORS – NO–FAILURES

97

first attribute one notices is that position estimation error profiles are somewhat biased. This is attributable to the uncompensated normal operating biases of the measurements. Examination of the EKF gain (at several points in the run) for r_x and r_y show that MLS azimuth and range measurements are used heavily in forming these estimates. Whereas, MLS elevation and range are primarily used for estimating r_z. Given the level of uncompensated bias in MLS measurements (found in Table 1) and the range as a function of time, these biases can be almost totally accounted for. In general:

o The bias levels shift in response to changes in the flight path — since the EKF gain is redistributed for the new flight path.

o The quantity $SQRT[\ |\tilde{r}_x|^2 + |\tilde{r}_y|^2\ ]$ and $|\tilde{r}_z|$ diminish as touchdown is approached (where $\tilde{r}_m \overset{\Delta}{=} r_m - \hat{r}_m$ i.e. estimation error).

o The a posteriori covariance is small due to the low measurement noise assumed for MLS measurements.

o Vertical estimation error characteristically is not effected by changes in the flight path. Its error diminishes primarily due to the diminishing effect of the unidentified MLS elevation normal operating bias.

No—fail filter velocity estimation error profiles are shown in Figure 11 for the same sample run. Keep in mind that these variables are expressed in the runway — not the body frame of reference. The error levels for all the profiles are reasonable and both lateral and vertical velocity errors converge to within the one—sigma covariance envelopes. The nearly linear convergence of the vertical velocity error is again due to the diminishing impact of the unidentified MLS elevation sensor's normal operating bias. The convergence in the x and y directions is characteristically flight path dependent. The lowest errors tend to occur when the signal level of the variable is close to zero (see Figure 9). Therefore, periods when the aircraft is

perpendicular to the runway (the A/C's velocity is in the y runway direction), x estimation error will typically be low.

The approximately 1 m/s bias[5] visible in the x velocity error profile after 150 seconds is due principally to the poor wind estimation performance. Remember, a 30 knot horizontal wind is present in all our runs. Lower wind levels scaled this bias error downward. Note that a .5 m/s spike occurs around 255 seconds in the vertical velocity error curve. This is due to the reconfiguration logic. At this point in the trajectory, MLS elevation measurements are replaced by radar altimeter measurements. The important point is to notice the selectivity of the reset — it only effects vertical velocity and vertical accelerometer bias errors. This is an important feature of the developed FTS.

Attitude estimation errors are shown in Figure 12. An rms error value of $0.05°-0.1°$ in roll, pitch, and yaw was typical of all the runs. These error levels correspond roughly to the unidentified normal operating bias values in the IMU measurements. Increases in the attitude estimation errors during banking maneuvers is due to the approximations resulting from using Euler integration for the kinematic equations in the single stage prediction part of the no—fail filter, and the fact that the filter is running at only a 20 Hz sample rate (typical update rates in conventional navigators vary between 60 and 70 Hz).

---

[5]The level of this bias is actually larger than levels normally observed. A more typical level would be approximately 0.3 m/s.

In this run, there was a false alarm in the roll rate gyro at 251 seconds which was correctly declared "recovered" by the healing tests at 258.9 seconds. The favorable impact of the healing decision can be seen in the roll attitude estimation error time history.

Horizontal wind estimation error time histories are given in Figure 13. Although some degradation in the horizontal wind estimates are to be expected due to the assumptions of zero angle of attack and side slip in the indicated airspeed measurement model, the steady-state estimation errors in these variables are largely due to the wind model used in the design of the filter. Specifically, the unknown constant random variable model for the horizontal winds causes the filter to become oblivious to the indicated airspeed measurements after the first maneuver. We believe that gain limiting on the horizontal winds would eliminate this behavior.

Error time histories for the accelerometer and rate gyro bias estimates are shown in Figures 14 and 15. Note the initial bias errors visible in the error trace before the FTS is initiated (at approximately 35 seconds). The good convergence characteristics for the accelerometer bias estimates were typical of other runs with different bias levels. Identification of accelerometer bias levels made a significant contribution to the improvement of position and velocity estimates. On the other hand, rate gyro biases are identified in order to introduce uncertainty into the kinematic models, and to help compensate for transient modeling errors due to scale factor and misalignment errors – although the actual bias levels in the rate gyro measurements were too small to be of any significance.

Figure 16 shows various A/C track error profiles for this run. These error levels were typical for the no—fail case. Although these curves are on indication of overall system performance, the reader should remember that they include the idiosyncrasies of the automatic control law. That is to say, the control laws are parameterized in a particular way and are therefore more sensitive to particular parameter variations.

3.3 Performance with Bias Failures

This section describes the observed performance of the FTS under simulated bias failures. Since the FTS was designed with the implicit assumption that failures would appear as bias jumps in a sensor's outputs, we should expect the best performance here. As discussed in Section 3.1.2, the format of this section will be to discuss the results from the context of individual simulation runs, in order to give the reader a deeper insight into how the method operates. Observed FTS bias failure detection and isolation performance will be described using the standard sensor configuration, and an alternative configuration employing an RSDIMU in the first two subsections, respectively. The last subsection describes performance when multiple (MLS) failures occur simultaneously in time.

3.3.1 Singleton Bias Failures — Standard Sensor Configuration

In this section bias failure performance will be discussed for the standard sensor configuration. In particular, three sample runs will be utilized. These runs were chosen to be representative, and in addition, to highlight strong, as well as weak points in the current implementation. Table 7 describes the simulated failure conditions for each of these runs (where each run is identified mnemonically). The

names chosen are BF−1, BF−2, and BF−3, where BF stands for bias failure. Further notice in the table that the first run, BF−1, has a single sensor, MLS elevation, which fails at time 80.6 seconds, with a failure magnitude of 0.24 degrees (this represents an 8 sigma failure level). In the majority of the runs used in this chapter, three or more failures are simulated in a single sample run. The implications of this are that it will save CPU costs, however, observability and time to detect of the remaining sensors may be affected.

We will now discuss each of these runs in detail, beginning with run BF−1. Generally, only error profiles, which differ from those of the previous section, will be shown. In this sample run, no false alarms were recorded, and no false healings occurred. The bias failure was simulated to occur at 80.6 seconds, and it was detected at 81. seconds yielding a time to detect of .4 seconds. Turning our attention now to Figure 17 and 18, we see that the effects of the elevation failure are negligible on x and y position and velocity estimation errors. That is to say that the time histories are very similar to those one would have seen had no failure occurred. However, z position and velocity errors (the bottom curves), show pronounced transient effects at the time of failure onset. Although the failure is detected (and the system is reconfigured) quickly, it takes a period of approximately 20 seconds to completely remove the effects of this failure from the state estimates. On the other hand, the disruptive effects are localized and are at a level similar to that which occurred at initialization. The rate of convergence for these error curves is noticeably slower than observed at start−up. This is due to the higher effective measurement noise on MLS elevation measurements assumed by the no−fail filter after system reconfiguration.

TABLE 7.   DESCRIPTION OF BIAS FAILURE RUNS – STANDARD SENSOR CONFIGURATION

RUN I/D	SENSOR FAILED	FAILURE MAGNITUDE	FAILURE ONSET TIME (seconds)
BF-1	El-2	$.24^{0}(8\sigma)$	80.6
BF-2	P-1	.1 degrees/s	115.9
BF-2	R-1	.1 degrees/s	66.9
BF-2	θ-1	.8 degrees	223.65
BF-3	Azm-1	$.3^{0}(10\sigma)$	110.65
BF-3	El-1	$.3^{0}(10\sigma)$	221.9
BF-3	ψ-2	$.8^{0}(10\sigma)$	66.65

103
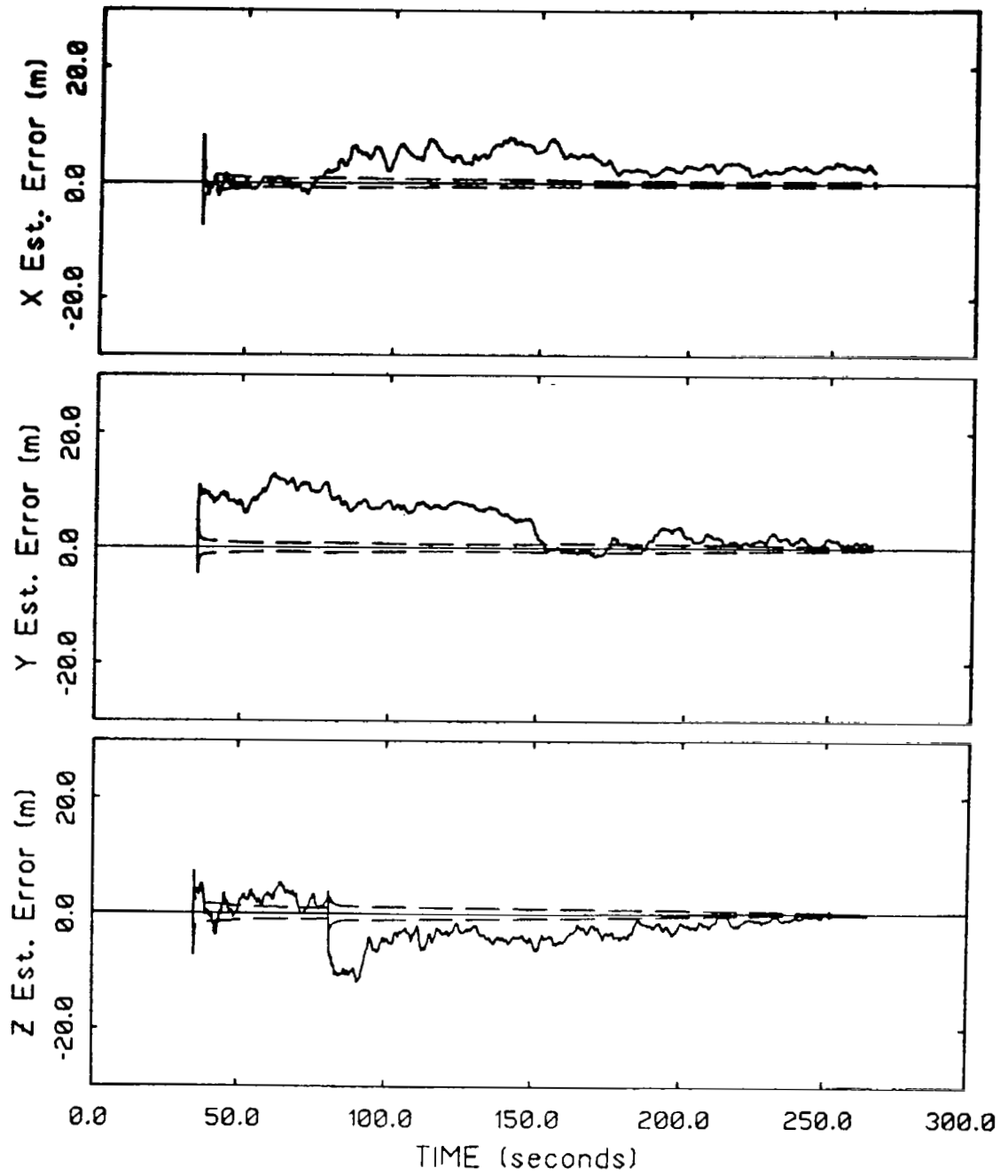
FIG. 17. POSITION ESTIMATION ERROR – BIAS FAILURE CASE BF-1
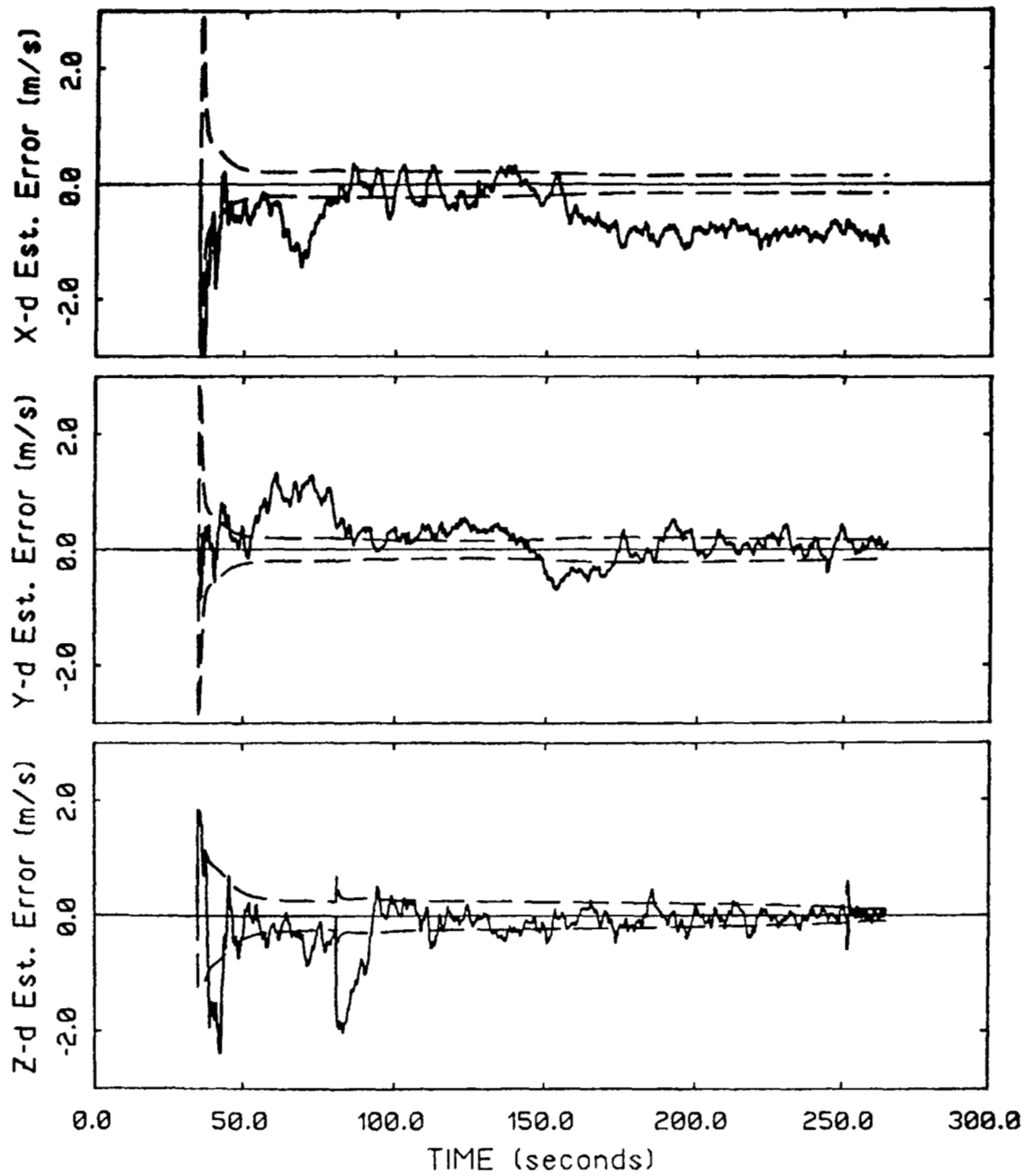
FIG. 18.   VELOCITY ESTIMATION ERROR — BIAS FAILURE CASE   BF-1

105
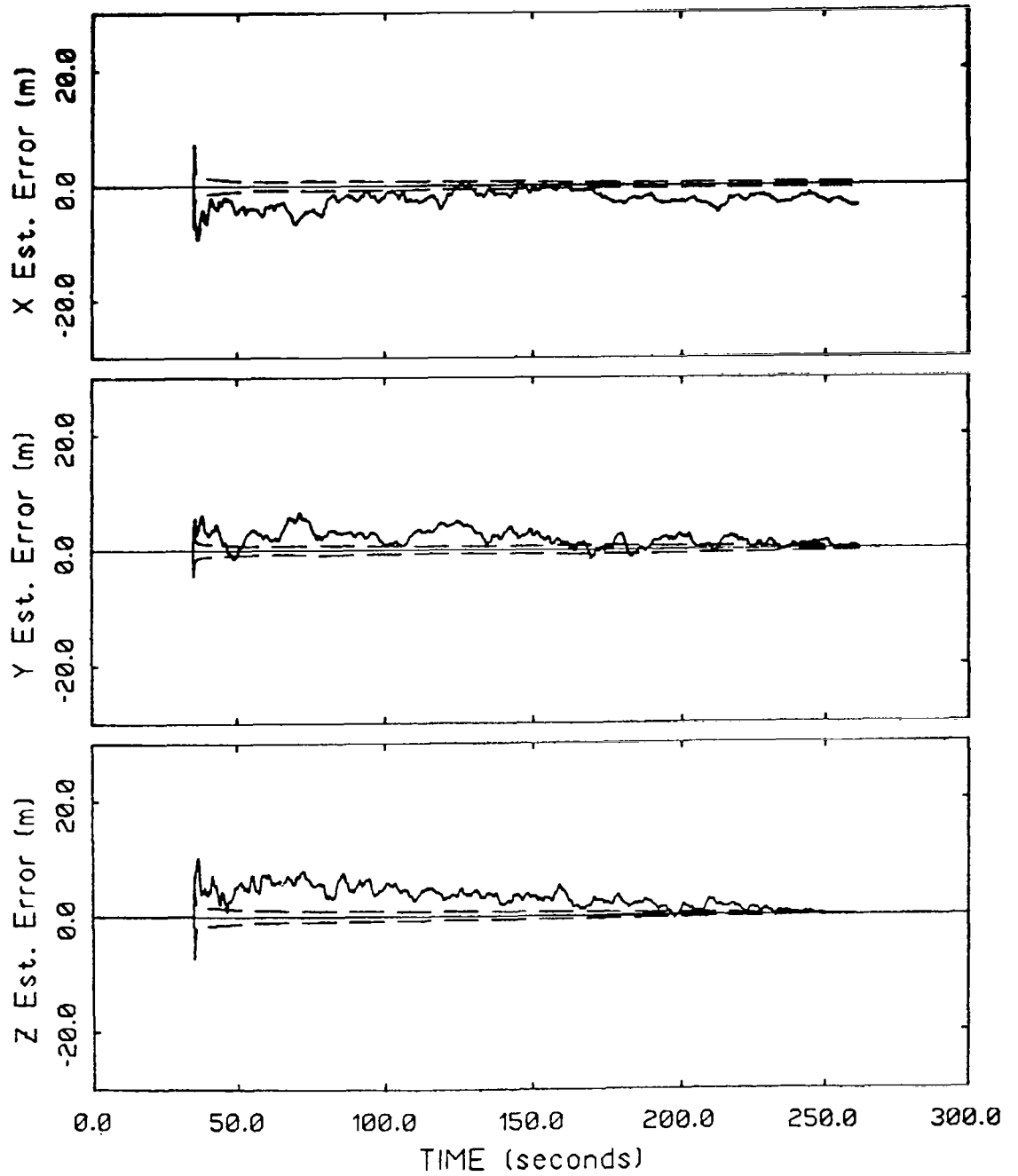
FIG. 19. POSITION ESTIMATION ERROR – BIAS FAILURE CASE BF–2
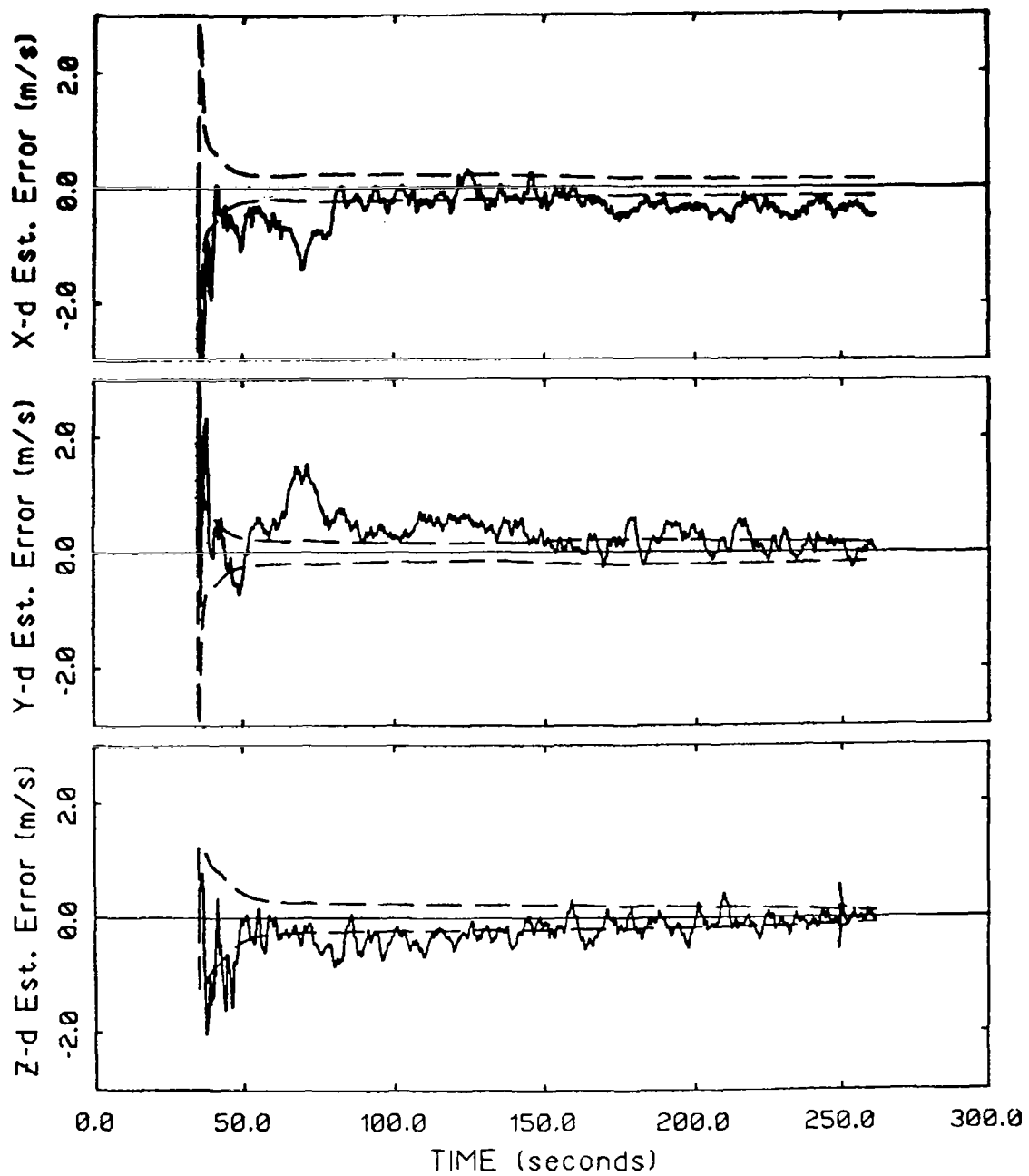
FIG. 20. VELOCITY ESTIMATION ERROR — BIAS FAILURE CASE BF—2

107
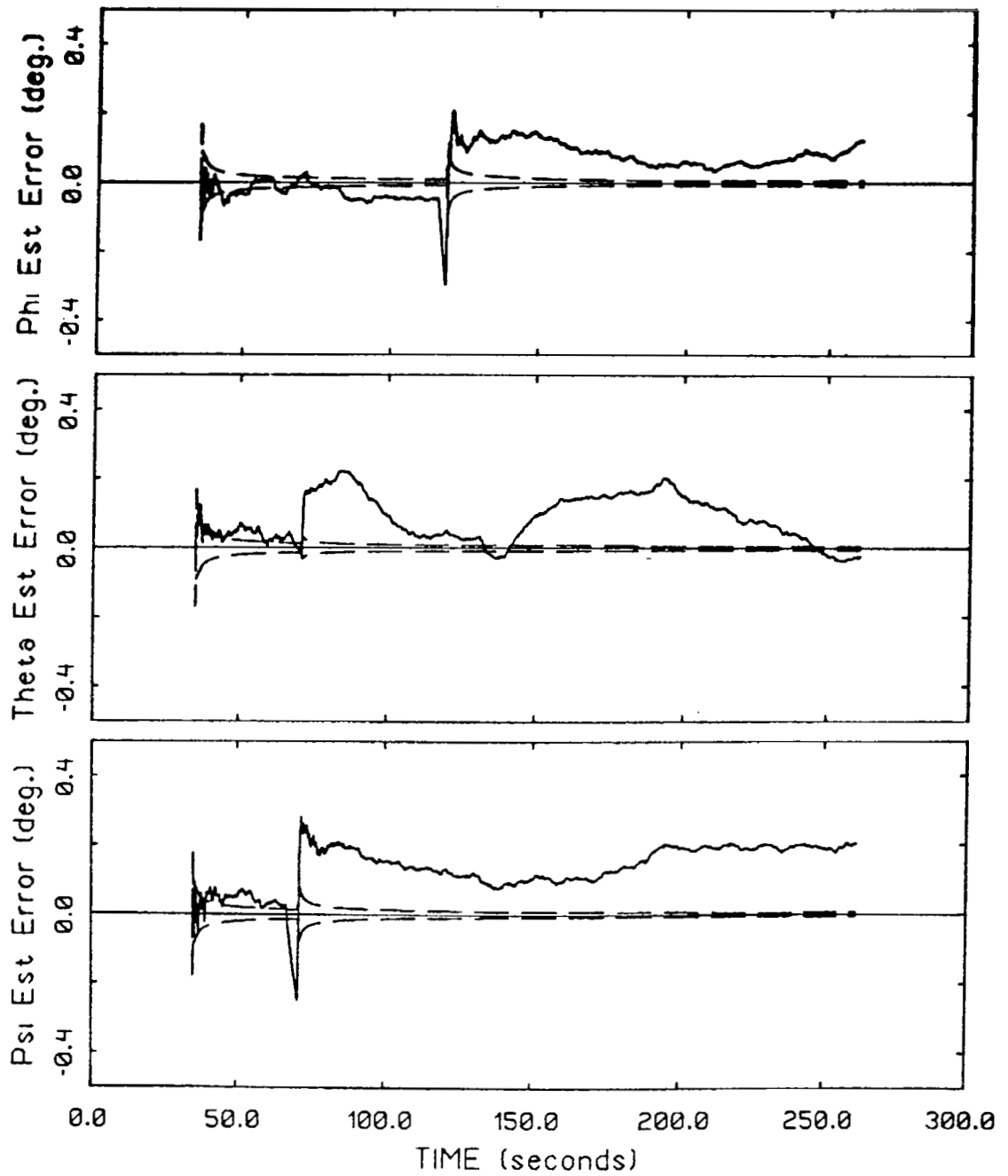
FIG. 21. ATTITUDE ESTIMATION ERROR – BIAS FAILURE CASE BF-2

108
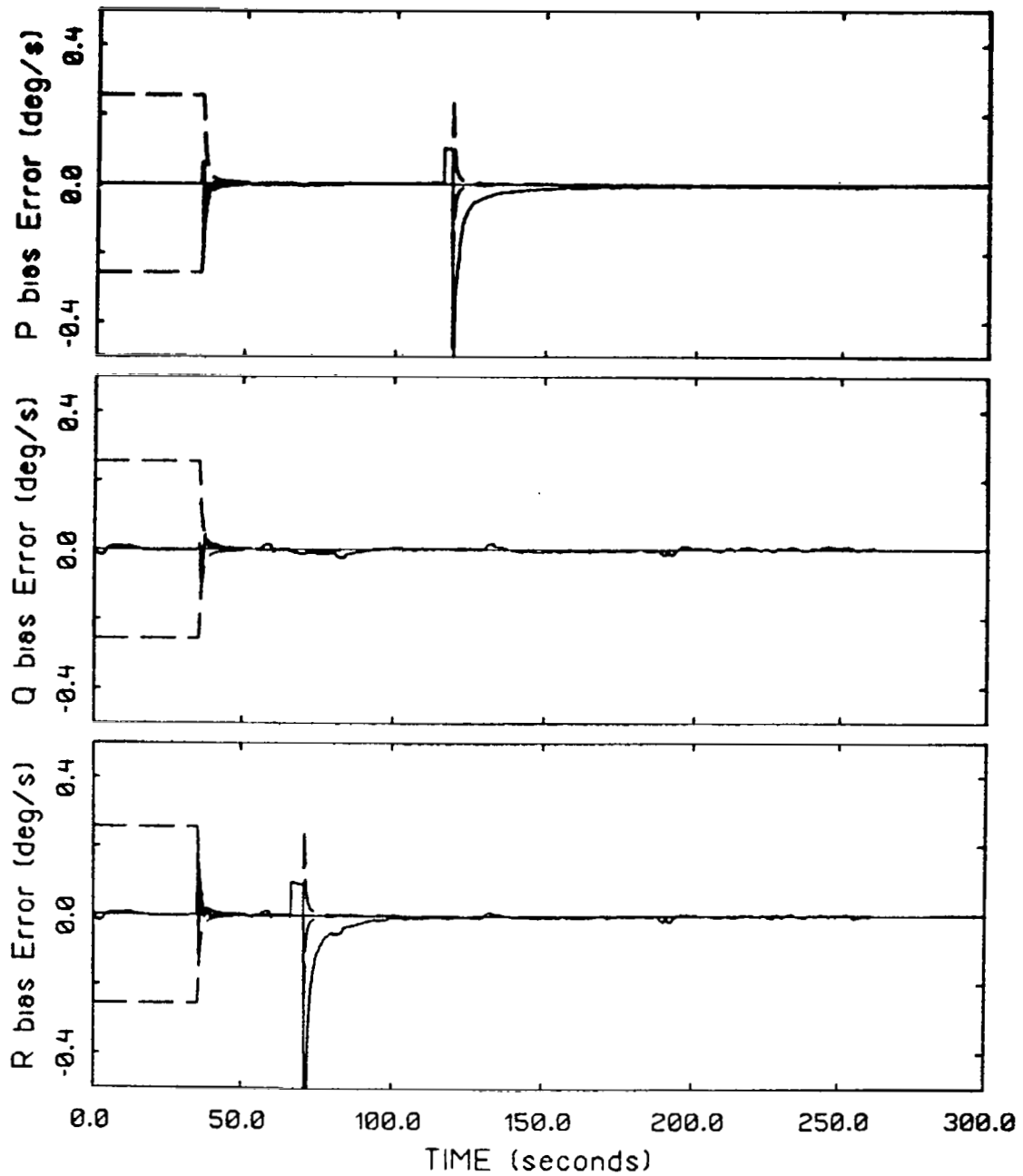
FIG. 22. RATE GYRO BIAS ESTIMATION ERROR – BIAS FAILURE CASE BF–2

The next sample run to consider is run BF-2. Table 7 shows that roll rate, yaw rate and IMU theta failures are all simulated in this run. These sensors all affect the attitude determination of the aircraft, and they are coupled through the kinematic equations of motion. Therefore, in this run, we will analyze how well the FTS system can discern input sensor from measurement sensor failures.

The failure detection performance in this case was quite good. All failures were correctly identified and there were no false alarms or false healings during this run. The time to detect for roll rate gyro was 2.75 seconds, and yaw rate gyro was 4.05 seconds. The detection time for theta was 0.3 seconds. This is a typical characteristic of input versus measurement sensor failures. In all of the runs measurement sensor failures were detected at least an order magnitude faster than input sensor failures. This is due primarily to the fact that the input failures, in order to be identified by the detectors, must propagate through the no-fail filter dynamics to put their signature on the residuals.

As mentioned previously, these sensors are used primarily to determine the A/C's attitude, and therefore only degradation to the no-fail filter's attitude estimates should be observed.[6] In fact, examination of the position and velocity estimation error curves shown in Figures 19 and 20 show that the FTS has this desirable property. Notice that the position and velocity estimation errors all lie within bounds which are typical of a no-failure case. Furthermore, disruptions due to resets after failures are

---

[6]Since the attitude estimates are used to resolve the accelerometer measurements, failures will eventually be observed in the other states — but to less an extent.

not felt by these state estimates. This is an important attribute of the developed FTS — without it resets due to correct failure detections (as well as false alarms) would drastically reduce the estimation error performance of the system as discussed in Section 2.6.

Viewing the attitude estimation error curve, Figure 21, the effects are seen to be pronounced here. Phi estimation error, the top diagram, shows a ramp occurring between the time of the roll rate gyro failure and its detection. At 118.65 seconds when the roll rate gyro bias failure is detected and the standby roll rate gyro replaces the failed sensor, a reset occurs. A similar effect is seen in the psi error when the yaw rate gyro fails. Notice the reset in phi has no effect on theta and psi errors. In this portion of the run, phi is less coupled to theta and psi — and the reset logic accounts for this fact. For example, the yaw rate gyro failure occurs during a banking maneuver, when couplings are stronger, and its reset effects theta estimation error. Moreover, notice that after both of the rate gyro resets, the estimation errors become biased. This seems to be a fairly typical phenomenon after an input sensor failure. We feel this is due to the fact that the input sensor bias estimates were reset in a rather hard fashion. These resets can be seen more clearly in Figure 22. Because the bias estimation error uncertainty on rate gyros is reset so hard, the no—fail filter primarily uses the raw IMU measurements to get an estimate for the attitudes. Since these measurements are biased, a step in the estimation error curves is observed. The effects of the IMU theta failure on the estimation error curves cannot be seen. This is due simply to the fact that the theta failure was detected and isolated quickly, in 0.3 seconds. Therefore, it simply wasn't present for a long enough period to affect the filter estimates.

TABLE 8. SUMMARY OF AVERAGE BIAS FAILURE PERFORMANCE — STANDARD
CONFIGURATION

SENSORS	AVERAGE TIME TO DETECT (Seconds)	NUMBER OF SIMULATED FAILURES	NUMBER OF MISSED DETECTIONS
Input Sensors			
A_x	4.65	3	1
A_y	14.65	3	2
A_z	N.D.*	3	3
P	13.32	4	1
Q	8.12	3	1
R	6.1	2	0
Measurement Sensors			
Azm	0.9	1	0
El	0.15	1	0
Rng	0.08	3	0
IAS	1.57	2	0
ϕ	0.57	4	0
θ	0.43	4	1
ψ	0.8	3	0
RA	0.3	3	1

* Not Detected

The third and final sample run, to be discussed, is called BF-3 in Table 7. It consists of three measurement failures. They include two MLS failures of azimuth and elevation, and also a psi attitude failure in the IMU. As we found out in the last example, the time to detect is much quicker for sensors which are treated as measurements into the no-fail filter. Detection times for this run were 0.9, 0.15, and 0.35 seconds for MLS azimuth, elevation and IMU psi, respectively. Because the detection times are so quick the estimation error profiles are not much different from those already seen — so they will not be presented. There were, however, two false alarms and two correct healings which occurred during this run. The affected sensors were the roll rate gyro, which was incorrectly detected as failed at time 248.9 seconds and then healed at 251.9 seconds[7], and the pitch rate gyro which failed at 253.65 seconds. The primary reason for these false alarms is that when the IMU psi failure is detected early in the run (at 67 seconds) an entire IMU is removed from the measurement set of the filter. This forces the no-fail filter to trust the rate gyros more, thereby increasing the effects of integration errors.

Up to this point details were primarily about individual sample time histories, and particular sequences of failures in order to understand how the fault tolerant system works. Here we present a summary of the average failure detection performance observed over all the runs. Table 8 describes the average performance by presenting the average time-to-detect for each sensor type. Also included in the table is the total number of simulated failures and number of missed detections by

---

[7]This is the fastest that healing is allowed to occur due to the assumptions made in the healer logic.

sensor type. The average time—to—heal is not shown in the table because it was approximately 3 seconds for all sensors. The runs which comprise the table have random failure levels of 3σ, 5σ, 8σ, 10σ, or 12σ. The reader should note that the total number of runs represented here is quite small. Furthermore, since the average time—to—detect figure shown is the average of from one to four samples (columns 2-3), one shouldn't place much weight on the actual numbers, but only look at them relative to one another.

Summarizing Table 8 we see that:

o Input sensors take considerably longer to detect than do sensors treated as measurements to the no—fail filter.

o Input sensors are harder to distinguish than measurement sensors — the number of missed detections is noticeably higher.

o Although not explicitly shown in the table, detection times for input sensors are much more sensitive to the absolute level of the failure simulated.

o Vertical accelerometer failures were not detected for 3σ,5σ,8σ,and 10σ failure levels.[8] Two factors contributed to this situation:

  . The signal level on the vertical accelerometer was 1g.

  . The normal operating bias filter was able to monotonically decrease the impact on the failure of the system, by slowly estimating it out.

o MLS azimuth takes considerably longer to detect relative to other MLS sensors.

o IAS is the slowest to detect of all measurement sensors.

o Of the IMU sensors, yaw failures are the most difficult to detect.

---

[8]However, null and hardover failures were detected.

114

Some general comments about particular problems that were encountered in running this matrix of cases seem appropriate at this point. Typically a missed detection of an accelerometer sensor failure induced an MLS false alarm. Also, for roll rate gyros, occasionally an IMU false alarm was observed, rather than a correct detection of the corresponding input sensor. In these cases — where the FTS was unable to distinguish dynamically related sensors — examination of the a—posteriori probability of failure obtained from the decision logic showed nearly equal probability of failure for the failed sensor and the dynamically related one. This indicates that this behavior can be improved by:

o Adding heuristics to the decision logic to evaluate this situation better.

o Add off diagonal costs in the decision logic — i.e. and a larger cost for making an incorrect decision.

In the case of radar altimeters, which turn on very late in the run, several times the wrong replication was chosen. This was due mainly to the fact that the filter's state estimates are biased due to the uncompensated normal operating biases contained in the measurements.

For the case of singleton bias failures the fault tolerant system works very capably. Detection speeds are very quick for measurement sensors into the filter, and also adequate for input sensors although they are typically detected an order of magnitude slower. The state estimates appear to have the quality of fault tolerance, in that they are able to recover in most cases from the effects of failure. Moreover, the fault tolerant system has the desirable property that a failure in one sensor affects only related quantities of the no—fail filter, therefore, transient effects are minimized. In addition, for the case of bias failures, the healers work quite adequately

115

as well, however, it's clear that the effects of false alarms can cause considerable problems.

3.3.2 Singleton Bias Failures — RSDIMU Sensor Configuration

This section discusses observed performance attained using a redundant strap down inertial measurement unit in place of the usual body mounted rate gyro and accelerometer measurements. In addition, since the RSDIMU also provides attitude outputs, these are used in lieu of measurements from the platform IMU.

The impact of these replacements on the FTS configuration and parameters are as follows:

o No estimation of input normal operating biases is performed.

o No detection and isolation is performed on RSDIMU based measurements — since it has its own on-board fault detection and isolation logic.

o The FTS reconfigures itself internally to operate the proper number of detector/LR computers (in our example this is 8, and in the standard configuration it is 20).

o Process and measurement noise levels chosen for the no-fail filter remain at the levels used in the standard sensor configuration.

The last item seems inappropriate at first — since the RSDIMU provides navigation quality rather than flight quality information. However, if process noise levels were set appropriate for the RSDIMU measurement noise level, the no-fail filter would ignore most of the other measurements, and therefore, detection of failures in these other sensors would not be possible.

For this configuration the transient effects due to failures, and the error time history plots are very similar to those observed for the standard configuration (they

116

are only slightly better), therefore, they will not be shown. The only exception to this is for ϕ,θ, and ψ, where — since we are not estimating normal operating biases on P,Q, and R — they appear more biased during maneuvers than the standard configuration. These effects can be minimized by "tuning" the measurement noise parameters assumed by the EKF on P,Q, and R, and on ϕ,θ, and ψ. Estimation of normal operational biases for the rate gyros will further reduce these effects. However, since we are only concerned with failures in MLS, IAS, and RA sensors, these errors don't impact performance, and therefore no further tuning was performed.[9]

Table 9 describes the average detection performance for the RSDIMU configuration. It should be noted that there were no false alarms or missed detections in any of these runs. However, since under this configuration, we are not considering failures in any RSDIMU sensors (i.e. only MLS, IAS, and RA failures are considered) this eliminates many sensors from consideration, thereby making the detection task easier.

Viewing Table 9 we see that:

o Detection times for all sensors are on the order of those found for the standard configuration (see Table 8).

o There were no missed detections.

o MLS azimuth is the most difficult MLS sensor to detect.

o Although not shown explicitly in the table, detection times were constant over all portions of the flight path.

---

[9]In fact, for this configuration, the filter could be collapsed to eliminate the attitude channel altogether — to further reduce the computational requirements.

In general, the RSDIMU configuration performed very well. Although we expected to see better performance and detection speeds for this configuration, we observed only slightly better performance. Additional Monte Carlo simulations of each configuration would be required to provide a better comparison.

3.3.3 Simultaneous Multiple Bias Failures – Standard Sensor Configuration

In this section, performance when two sensors fail at the same instant of time, will be examined. Only multiple failures in like MLS measurements were considered in this study, and will be reported herein.

The FTS parameters used are the same as described in Section 3.2. The error profiles for these failures look very much like the no-fail case, and therefore will not be presented here. In fact, after a multiple MLS failure is correctly detected a mission abort is issued, since the no-fail filter requires at least one of each MLS measurement to operate.

Table 10 describes the simultaneous multiple failure performance of the FTS. All multiple failure sample runs made in the study are represented here. Since MLS azimuth is harder to detect than MLS elevation or range, azimuth failures were simulated over all flight conditions, whereas MLS elevation and range were only failed once. Detection speed and selectivity is very good, with correct detections in all cases and average detection times of 6.37, 0.2, and 0.15 for azimuth, elevation, and range sensors, respectively. Over this set of runs the detection times were basically constant over different failure levels and flight path segments. Notice, further, that although false alarms were associated with some of the runs, they did not interfere

TABLE 9. SUMMARY OF AVERAGE BIAS FAILURE PERFORMANCE — RSDIMU CONFIGURATION

SENSOR	AVG. TIME TO DETECT (Seconds)	NO. OF RUNS SIMULATED	NO. OF MISSED DETECTIONS
Azm	0.42	5	0
El	0.18	5	0
Rng	0.09	5	0
IAS	0.4	1	0
RA	0.15	1	0

with the correct detection and isolation of this class of failure.

3.4 Performance with Non—Bias Failures

As mentioned at the start of this chapter, one of the goals of the study was to assess how well the developed FTS can operate when failures of a non—bias type are encountered. This section discusses the observed robustness of the system under the following failure modes: hardover, null, ramp, increased scale factor, and increased noise failures. A subsection is devoted to discussions pertinent to each of these conditions.

3.4.1 Hardover Failures

The primary attribute required of an FTS when hardover failures might be encountered is fast detection times. Ideally one would like to remove faulty measurements before they are used in the navigation filter. This is particularly important in our FTS since the no—fail filter is an extended Kalman filter (e.g. linearizations are about estimated states) and filter divergence can occur quickly. For this reason, if redundant sensors are available and hardover failures are a common problem, we advocate the use of voting techniques to achieve quick hardover failure detection and isolation. As we will see, the FTS discussed in this section provides an alternative approach if redundant sensors are not available.

Since hardover failures can be viewed as very large bias failures, we should expect good performance from the developed FTS. In fact, observed isolation and detection speed is quite good — it typically takes one filter cycle (i.e. two measurement samples) to detect a hardover failure.

TABLE 10. SUMMARY OF MULTIPLE BIAS FAILURE PERFORMANCE

MLS SENSOR TYPE	FAILURE LEVEL	FAILURE ONSET TIME (seconds)	TIME TO DETECT (seconds)	NUMBER OF FALSE ALARMS/ SENSOR TYPE
Azimuth	$.3^{\circ}(10\sigma)$	110.0	0.6	0
Elevation	$.3^{\circ}(10\sigma)$	70.0	0.2	$1/A_z$
Range	$40m(10\sigma)$	70.0	0.15	$1/A_y$
Azimuth	$.3^{\circ}(10\sigma)$	70.0	0.6	0
Azimuth	$.3^{\circ}(10\sigma)$	255	0.65	$1/A_y$

121

Table 11 shows the detection performance for a typical run. Hardover failures in indicated airspeed, IMU theta, and roll rate gyros are simulated. Remember that only one replication of the roll rate gyro is used by the FTS, and therefore, no direct redundancy is used in its detection.

The sample time for the system is 0.05 seconds (e.g. 20 Hz) and therefore all three failures are detected in one filter cycle. The fact that three very hard failures occurred fairly close to one another, and did not significantly affect the performance of the FTS is encouraging. This implies that the filter was able to recover quickly from the effects of failures.

Figures 23, 24, and 25 show the estimation error profiles for attitude, rate gyro biases, and accelerometer biases, respectively. These figures show how the reset logic can directly effect filter performance. From Table 10 we saw that a false alarm in IMU phi occurred right after the failed roll rate gyro was removed. This was directly due to the large reset used for the phi state and p bias estimates and their associated covariances. The filter diverges as a result of the reset, and cannot recover since both IMU's were removed (one due to a failure and the other due to the false alarm). The large reset was due to the reset logic applying a reset proportional to the failure level — it was designed assuming moderate bias failure levels. Since the failure estimates from the detectors may contain significant errors when hardover failures are encountered, an obvious solution would be to threshold the reset to remain within a reasonable range.

TABLE 11. DETECTION PERFORMANCE FOR TYPICAL HARDOVER

SENSOR	FAILURE MAG.	ONSET TIME (seconds)	TIME TO DETECT (seconds)
IAS	205.78 m/s	67.08	0.05
θ	80°	114.05	0.05
P	100 $^{\circ}$/s	153.50	0.05

False Alarms

Sensor	Time of False Alarm (seconds)
φ-2	153.65
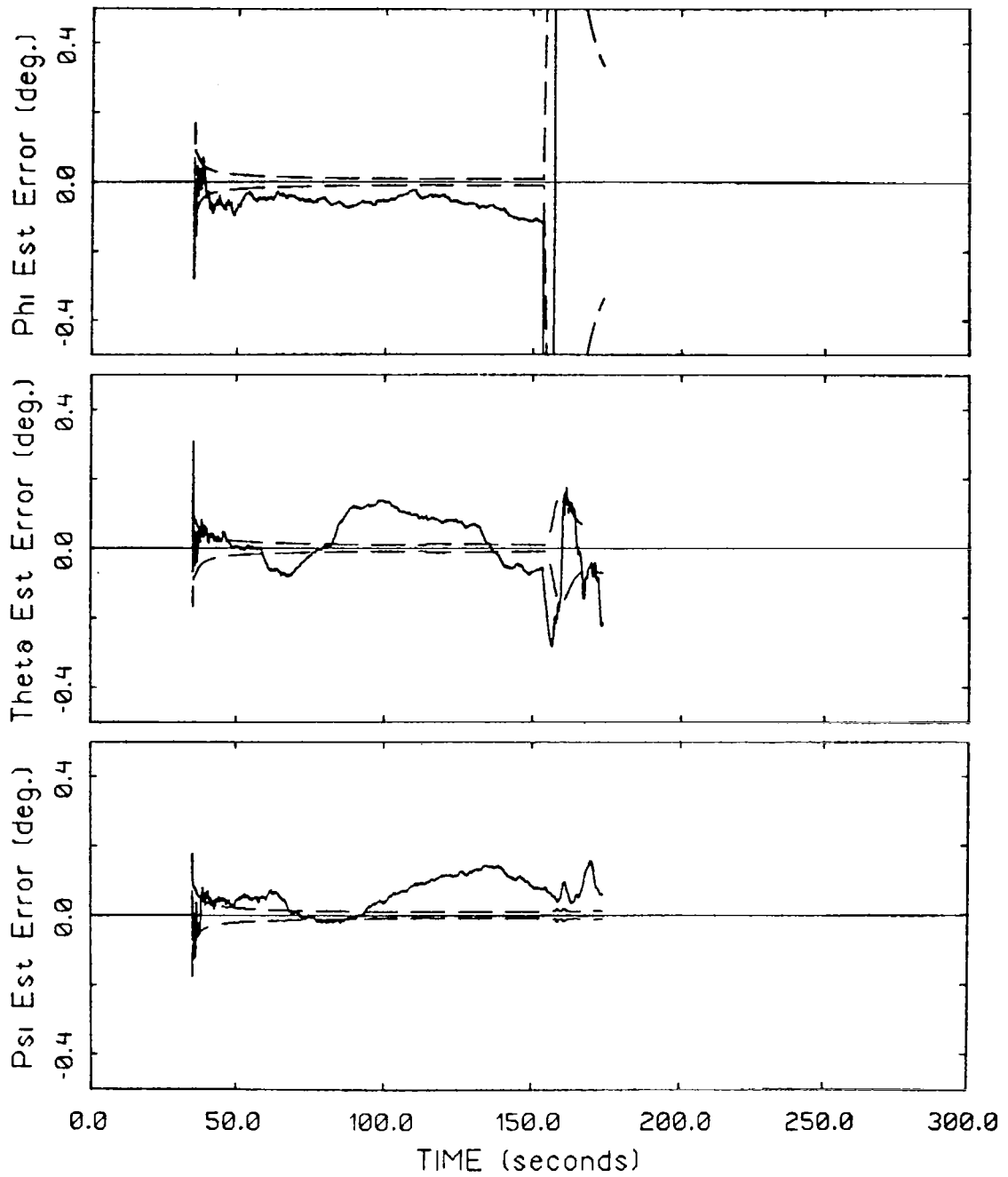
FIG. 23. ATTITUDE ESTIMATION ERROR - HARDOVER FAILURE

124
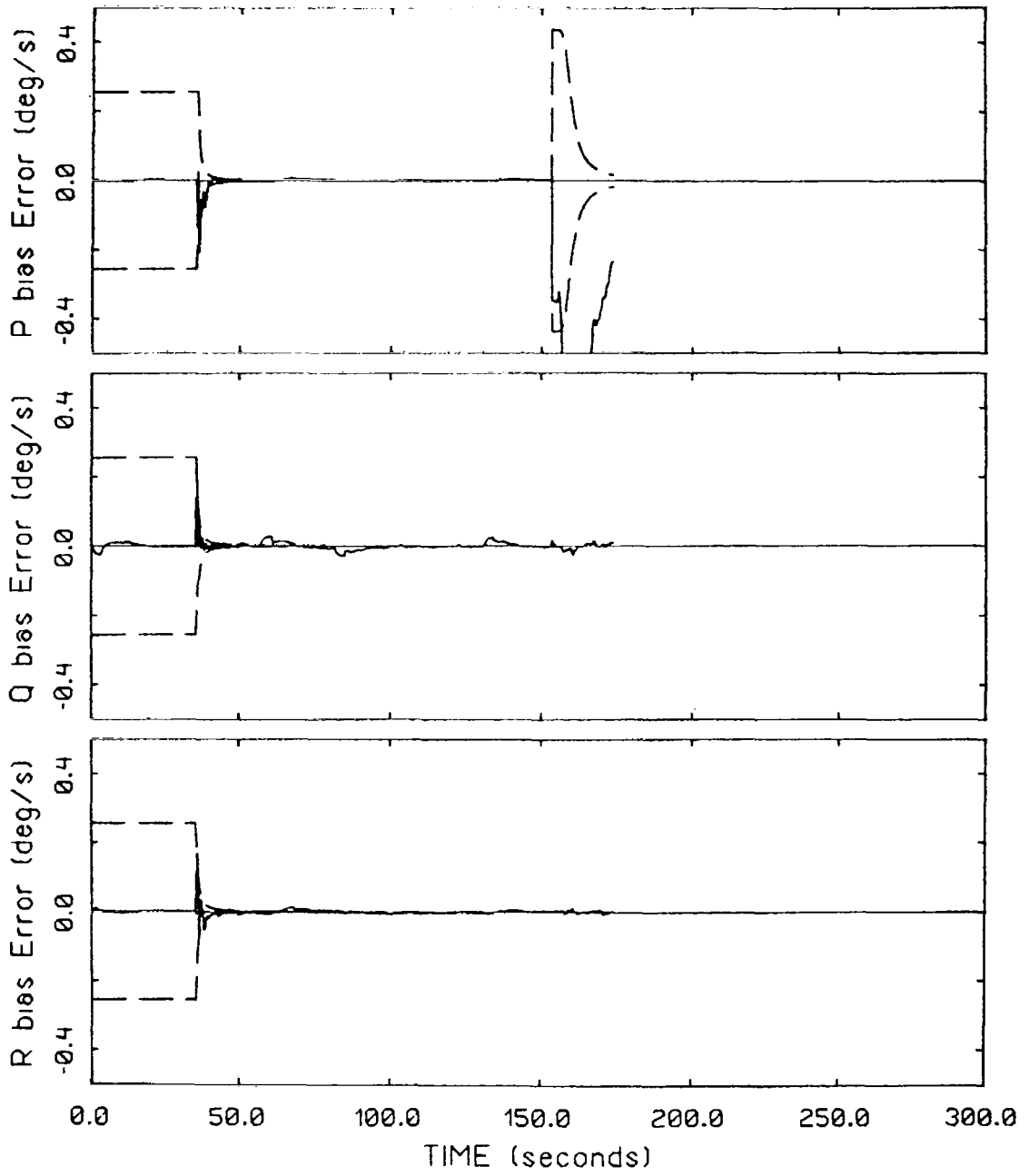
FIG. 24. RATE GYRO BIAS ESTIMATION ERROR – HARDOVER FAILURE
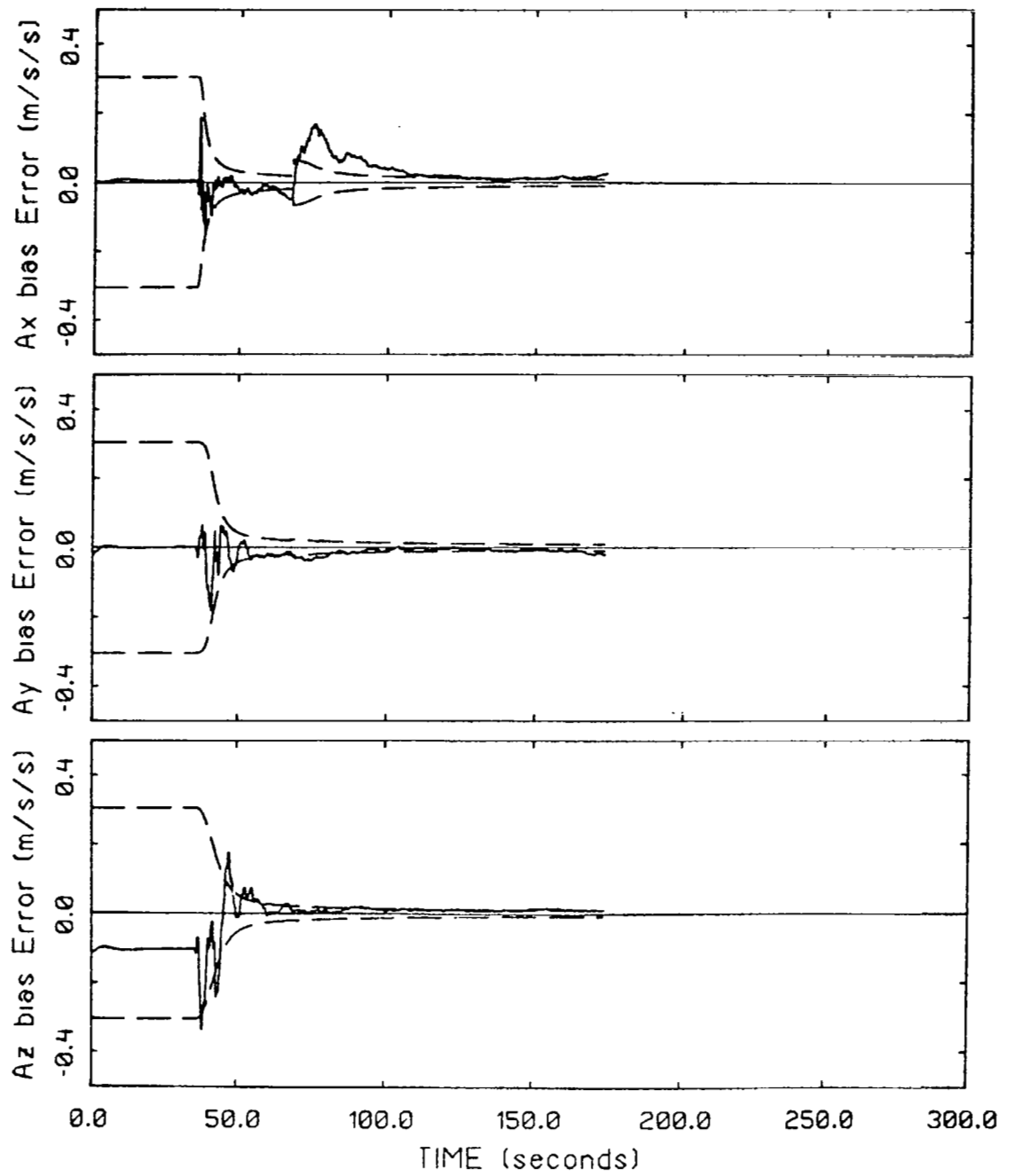
FIG. 25. ACCELEROMETER BIAS ESTIMATION ERROR – HARDOVER FAILURE

3.4.2 Null Failures

Null failures pose an interesting problem for the FTS. Although they may appear as a bias jump in the measurement, this only happens if the actual signal level is not close to zero. Essentially, to be detectable, the signal level must be greater than the smallest bias level detectable by the FTS. Moreover, the effective failure level fluctuates as a function of the signal level, such that at times it may appear as a soft, mid or hard failure. In fact, even if the failure is correctly detected, the healer logic can easily be tricked into believing the sensor has recovered during the next period of low signal level. A single sample run, which contains null failures, will be discussed in this section. Certainly many more simulation runs would be required to generalize the results presented here.

Table 12 shows the detection performance for a run with null failures present. The first null failure effects the IMU phi measurement during a banking maneuver (e.g. when the signal level is large), whereas the second failure is in the pitch rate gyro when its signal level is very close to zero. The first failure is detected in 0.15 seconds (e.g. three filter cycles), however, it is not removed quickly enough to prevent a roll rate gyro false alarm. The second failure is not detected, as expected, since its signal level is very small. It does, however, instigate a pitch sensor false alarm.

Another perspective is shown in the estimation error profiles for this run, Figures 26–30. These curves clearly show the coupling effects during the bank maneuver. Notice the null failure in ϕ effects ϕ, θ, P, and Q, but it also effects x_{rw},

127

y_{rw}, $\dot{x}_{rw}$, $\dot{y}_{rw}$, A_x, and A_y. The encouraging thing to note here is that the FTS manages to recover from the first failure.

The second failure requires some explanation to understand what is taking place. First of all, one might assume that since the signal level on Q is small, (see fig. 6), a null failure shouldn't impact the estimates very much (look at Q bias errors in figure 30, for example). However, in this case the null failure in Q instigates a θ false alarm. Note in figure 7 that θ is not zero, rather it's changing slowly. This removes all attitude measurements from the filter. Therefore, with Q=0, θ=constant. From this point on the estimation error in θ fluctuates as a function of the difference between the true signal and this constant, plus any other contributions from the filter update using the rest of the measurements (see fig. 28).

Although very few runs were made with null failures, we would anticipate the following problems:

o Occasionally dynamically coupled sensors will be chosen (e.g. φ when P fails, El when A_z fails, etc.)

o Filter divergence when detections are too slow, and the effective failure level is large.

o Declaring a faulty sensor "recovered" when a low signal level is monitored for a period of time.

On the other hand these results are encouraging since:

o They show that null failures can be detected without any modifications to the original method.

o The bias jump failure models are "robust" enough to provide coverage for

these failures — however, either a better model or more heuristics will be required for reliable detection, and healer operation.

3.4.3 Ramp Failure

Ramp failures appear in the measurements as slowly increasing biases — in other words, at every successive step, a new, incremental bias jump is applied to the measurement. If the slope of the ramp failure is large, the incremental bias jump is large, and it will appear much like a bias failure would. In this case we would expect the FTS to detect it without modification. However, if the slope of the failure is small, the failure may go undetected for a long period of time, or it may create a drift in the no—fail filter estimates and never be correctly identified. In this section, FTS performance under ramp failures will be discussed by means of a typical example.

Performance for a typical sample run is shown in Table 13. Ramp failures are simulated for Azm, θ, P, and A_z sensors. The failure levels chosen in the table reflect the slope required to attain a 3σ normal operating bias in one second for each sensor. Figures 31—36 show the estimation error and A/C track error profiles for the same case.

From Table 13 we see that all simulated failures are correctly detected and isolated by the FTS. Sensors which are treated as measurements are detected slightly faster than those treated as inputs to the no—fail filter. This was found to be a common trait of the developed system. Another characteristic for ramp, as well as other non—bias failures, is the induced false alarms caused by the reset after the failure is correctly detected. This is observed in the ϕ and RA failures in this case. The false alarm in MLS elevation, however, is not due to the reset logic, but rather to

TABLE 12.   DETECTION PERFORMANCE FOR NULL FAILURES

SENSOR TYPE	FAILURE ONSET TIME (seconds)	TIME TO DETECT (seconds)
ϕ-1	65.75	0.15
Q-1	111.65	X

False Alarms

Sensor	Time of False Alarm (seconds)
P	65.8
θ-2	121.5
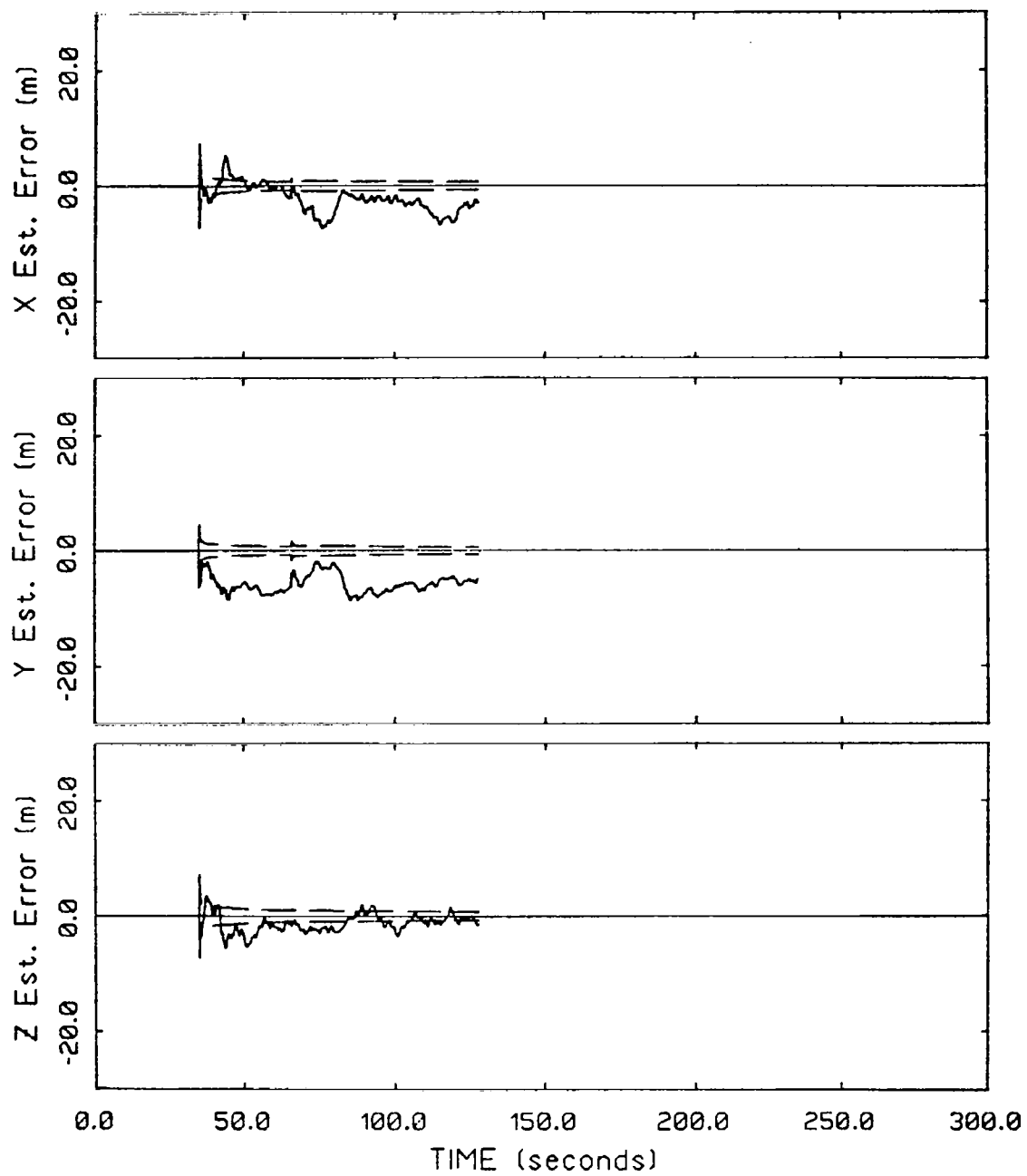
FIG. 26. POSITION ESTIMATION ERROR – NULL FAILURES
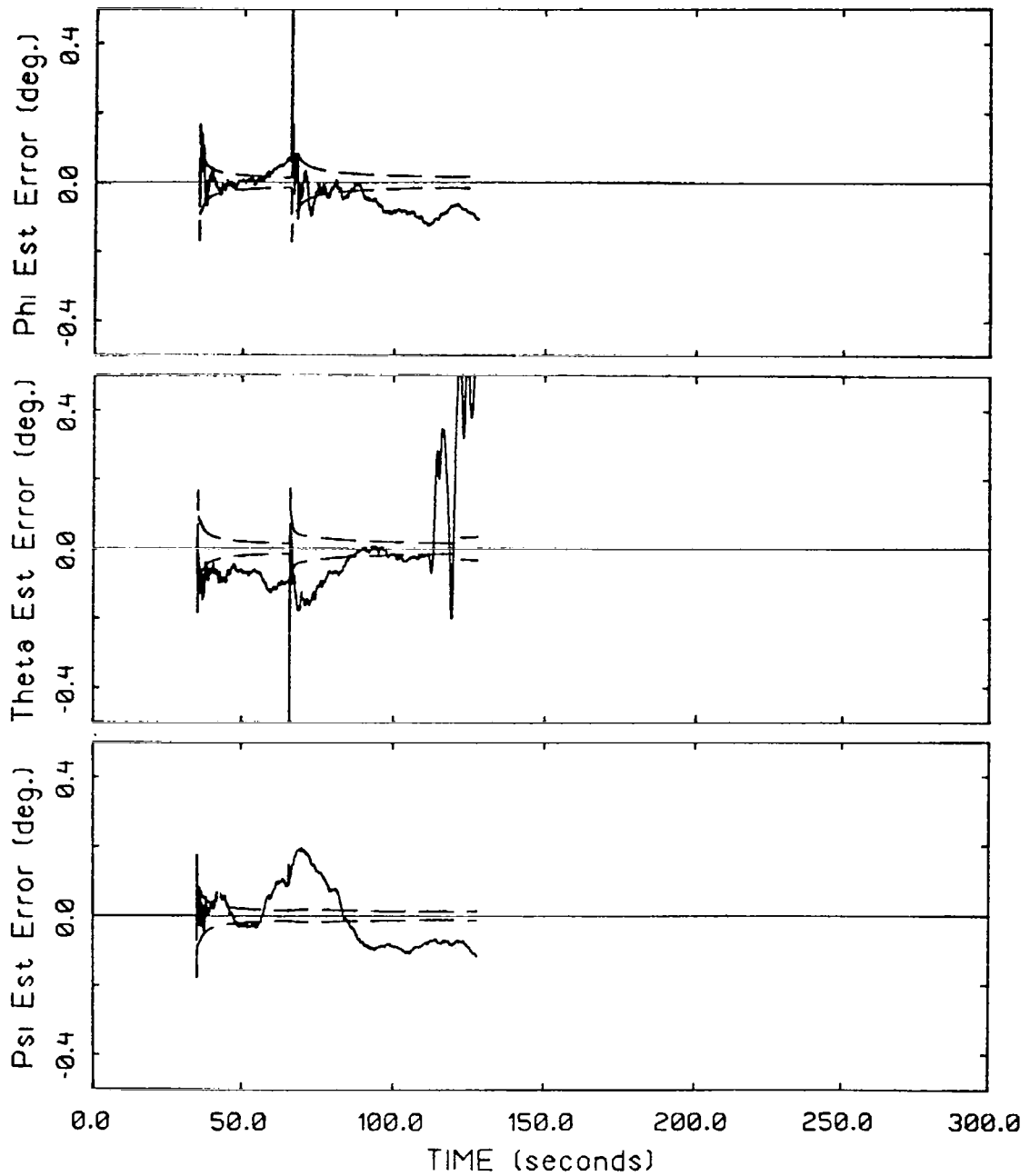
131
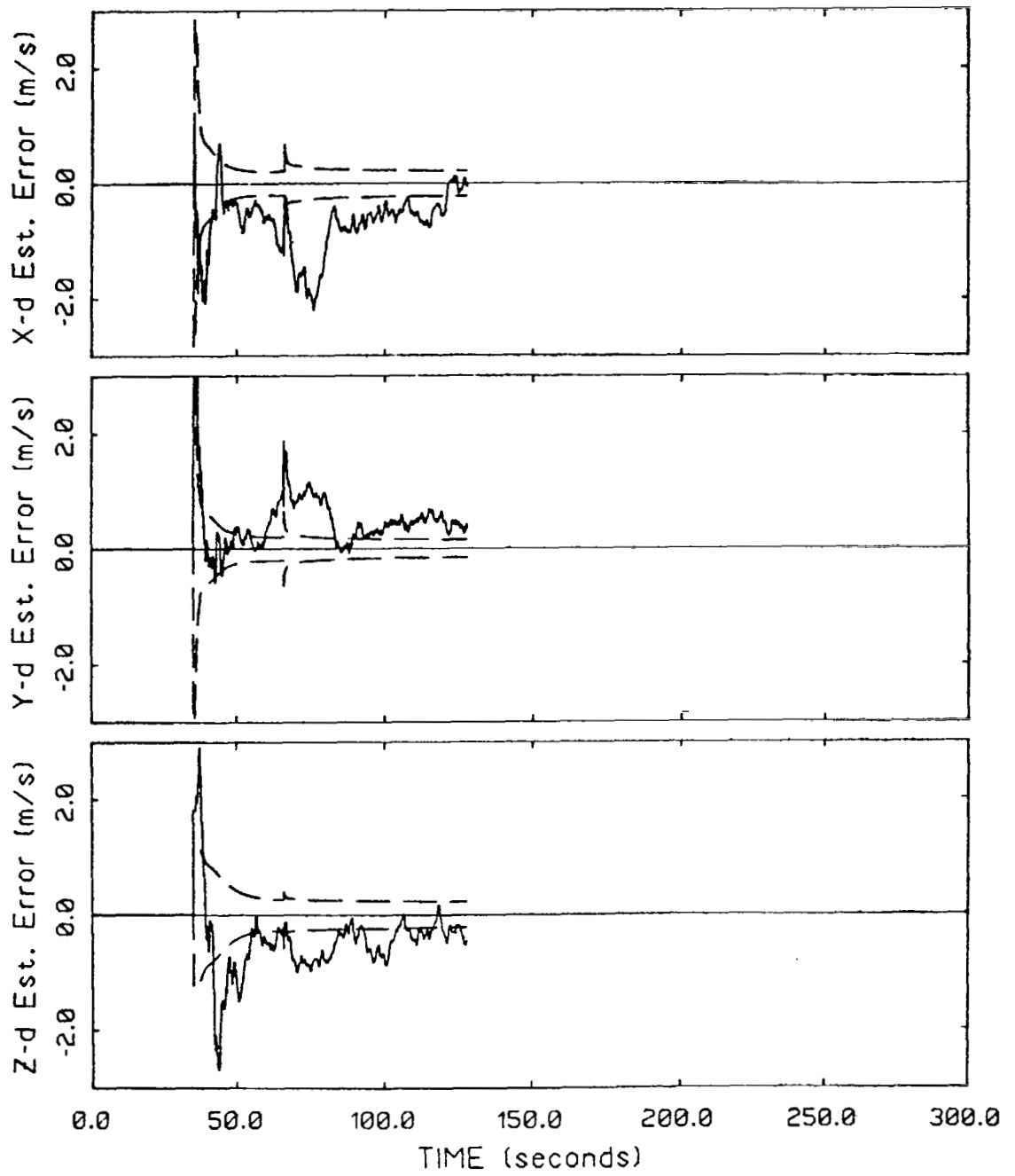
FIG. 27. VELOCITY ESTIMATION ERROR – NULL FAILURES
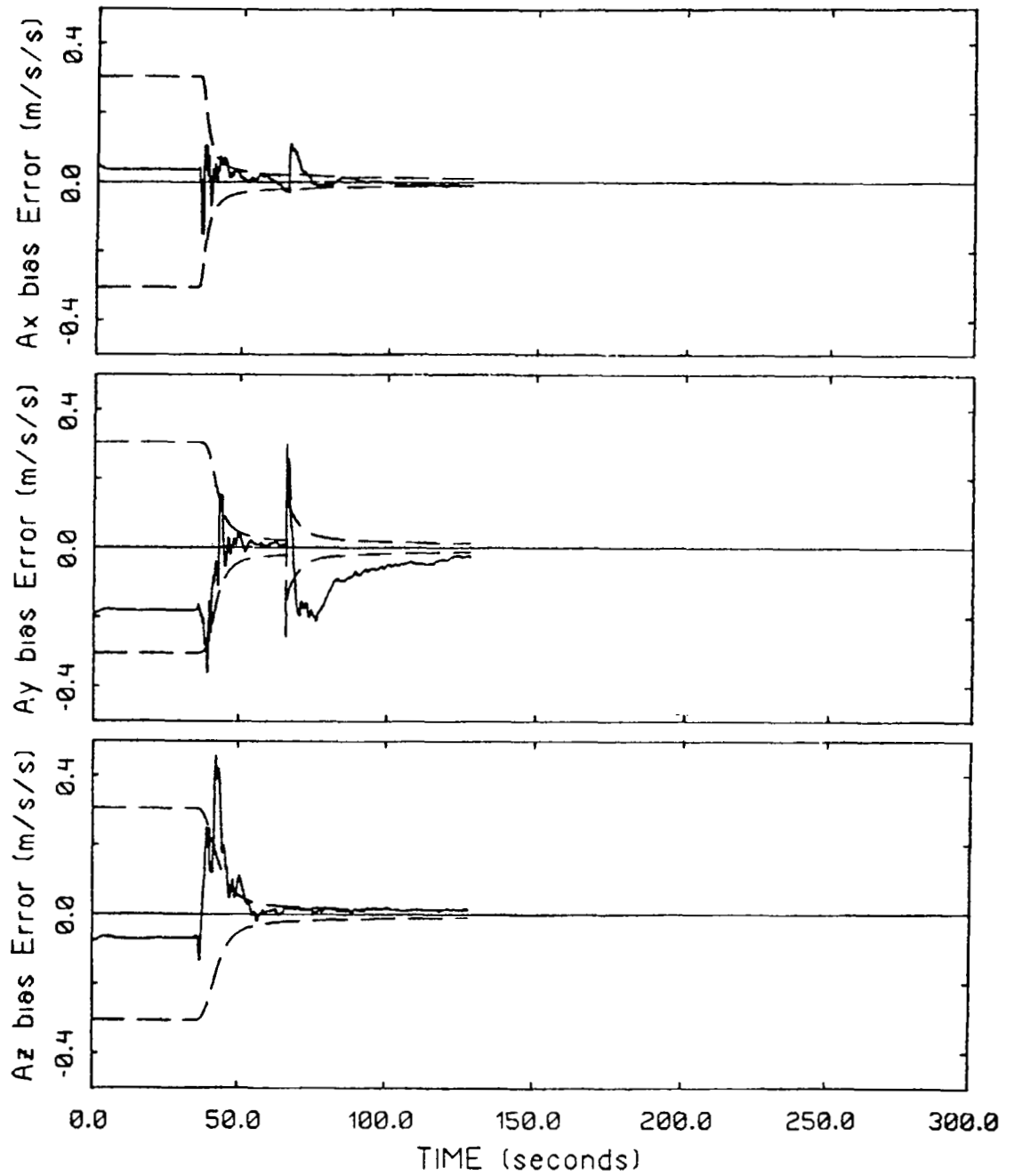
132

FIG. 28. ATTITUDE ESTIMATION ERROR – NULL FAILURES

133

FIG. 29.   ACCELEROMETER BIAS ESTIMATION − NULL FAILURES

134
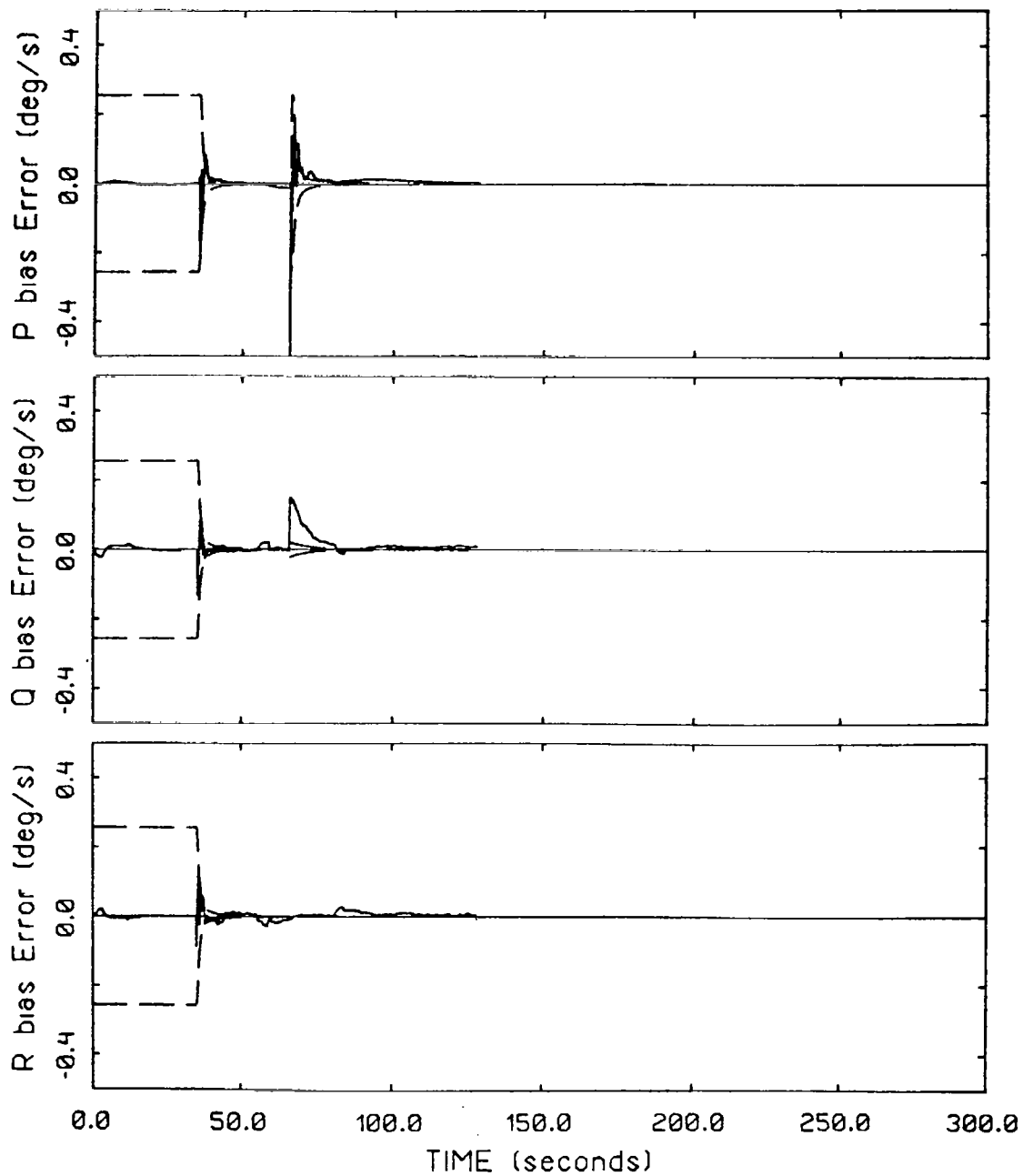
FIG. 30. RATE GYRO ESTIMATION ERROR – NULL FAILURES

135

the FTS's inability to correctly distinguish between the vertical accelerometer ramp failure and MLS elevation. However, as seen previously in other examples, this type of false alarm (due to indistinguishability) doesn't generate a missed alarm, rather it simply takes longer to make a correct detection, and causes more of a disruption to the state estimates.

The estimation error profiles again show the effects of a hard reset. It is especially evident in the ϕ estimation error. Otherwise, the FTS behaves quite reasonably, considering the number of failures simulated in this short run.

3.4.4 Increased Scale Factor Failures

Increased scale factor errors are quite different from bias failures, and therefore one should expect to see degraded performance for this class of failures. Consider the following: a scale factor error may look like a constant bias failure – if the signal level is constant; or it may look like a time varying failure – if the signal level varies. In between these two extremes, the failure will look like a combination of the two. In this section, performance of the FTS when increased scale factor errors are introduced will be discussed.

Several runs were made, initially, with the healer logic running, however, since the failures are a function of the signal and noise levels, failed sensors would heal after a short period of time and we found they could not be used reliably in their present form (a bias failure mode is implicitly assumed in the formulation of the healers). The run presented in this section, therefore, has the healers turned off.

136

TABLE 13.   DETECTION PERFORMANCE FOR RAMP FAILURES

SENSOR TYPE	FAILURE MAG.	FAILURE ONSET TIME (seconds)	TIME TO DETECT (seconds)
Azm	.009°/s	66.85	2.30
θ-1	.3°/s	112.25	1.9
P-1	.005°/s/s	154.1	4.5
A_z-1	.5m/s^2/s	223.9	6.6

False Alarm & Healings	Time of False Alarm (seconds)	Time of Healings (seconds)
φ-2	118.05	121.05
φ-2	159.6	171.6
E1-1	230.9	233.9
RA-2	255.9	258.9
RA-1	261.85	X
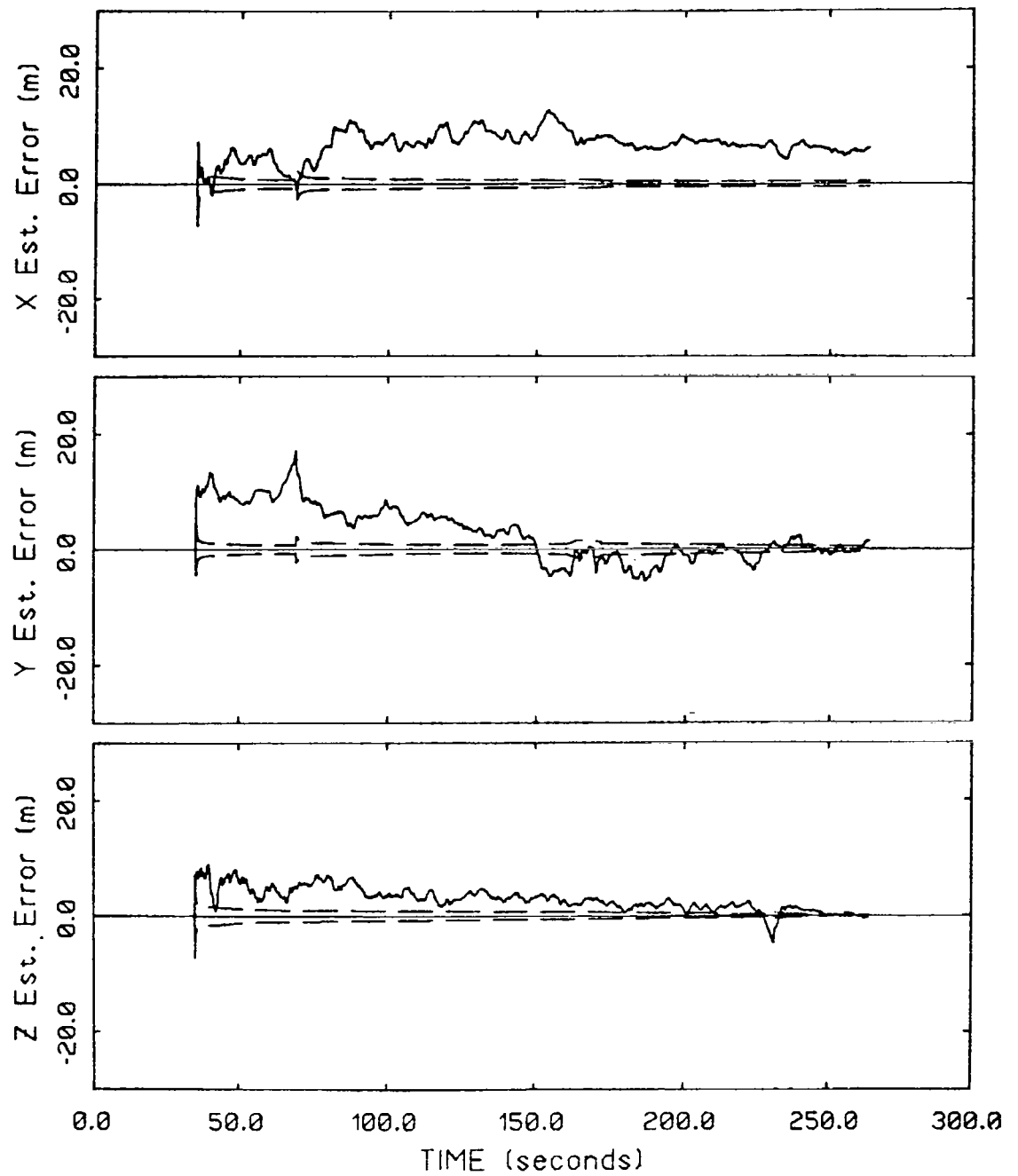
FIG. 31. POSITION ESTIMATION ERROR – RAMP FAILURES
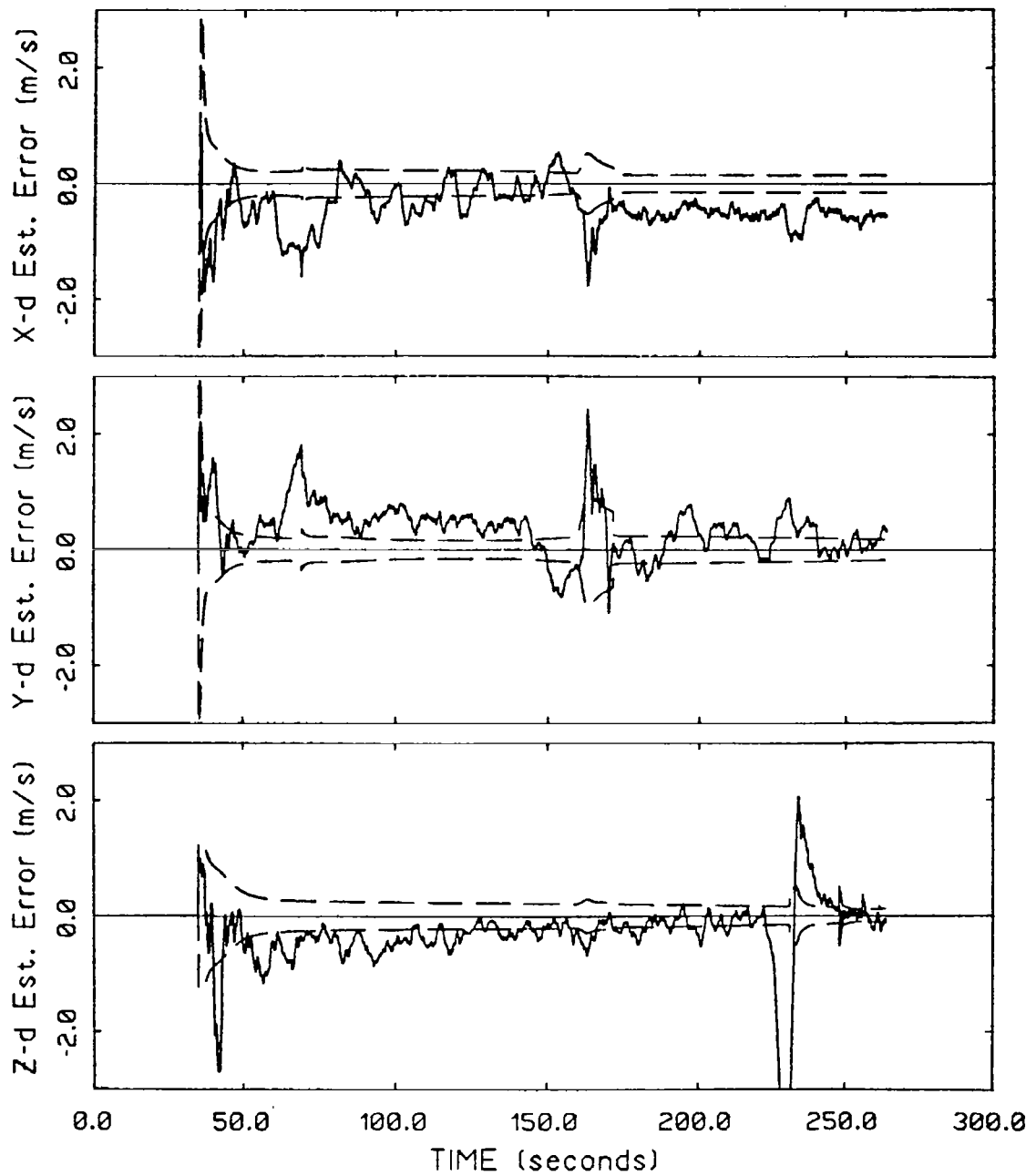
FIG. 32. VELOCITY ESTIMATION ERROR – RAMP FAILURES

139
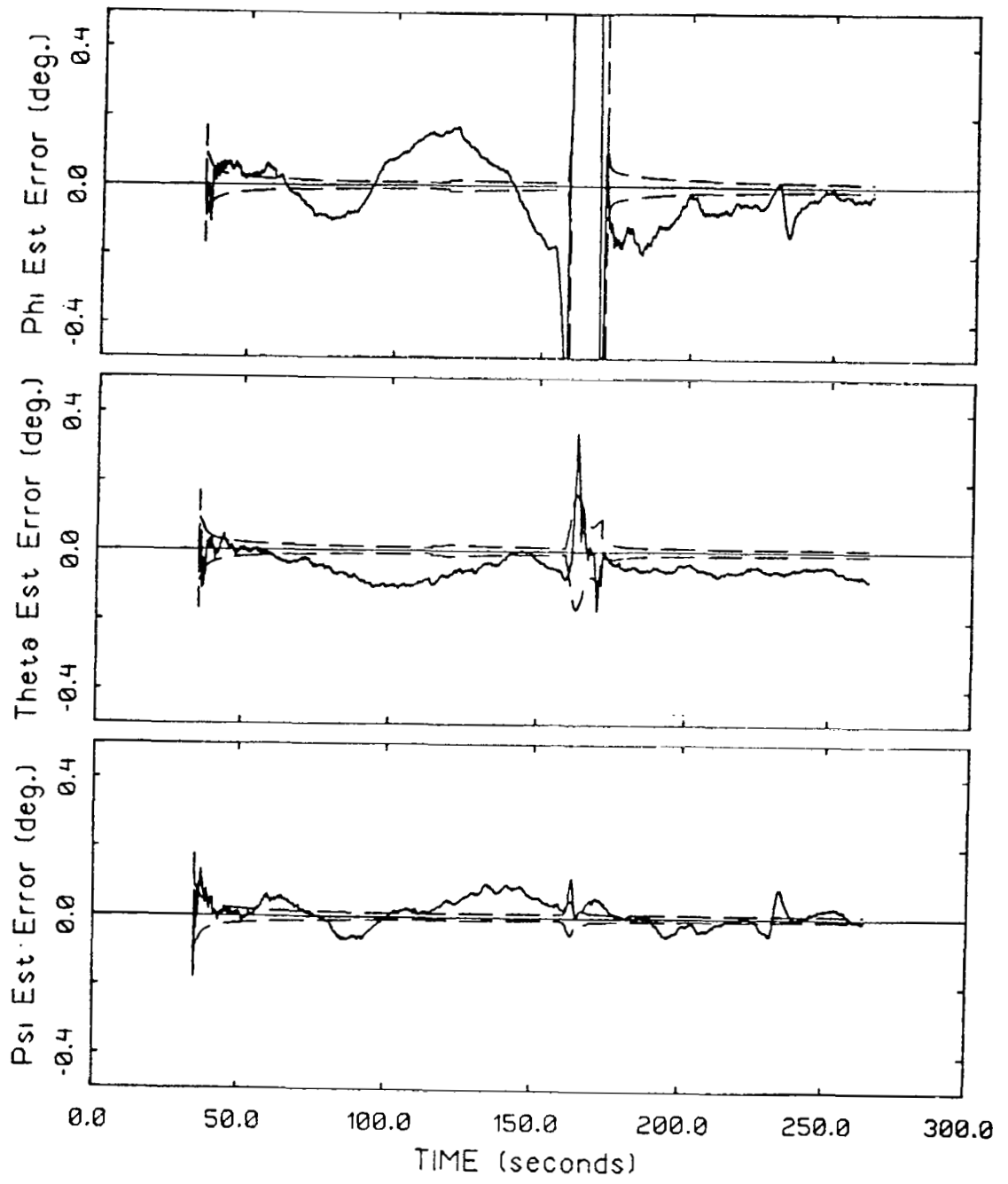
FIG. 33. ATTITUDE ESTIMATION ERROR – RAMP FAILURES

140
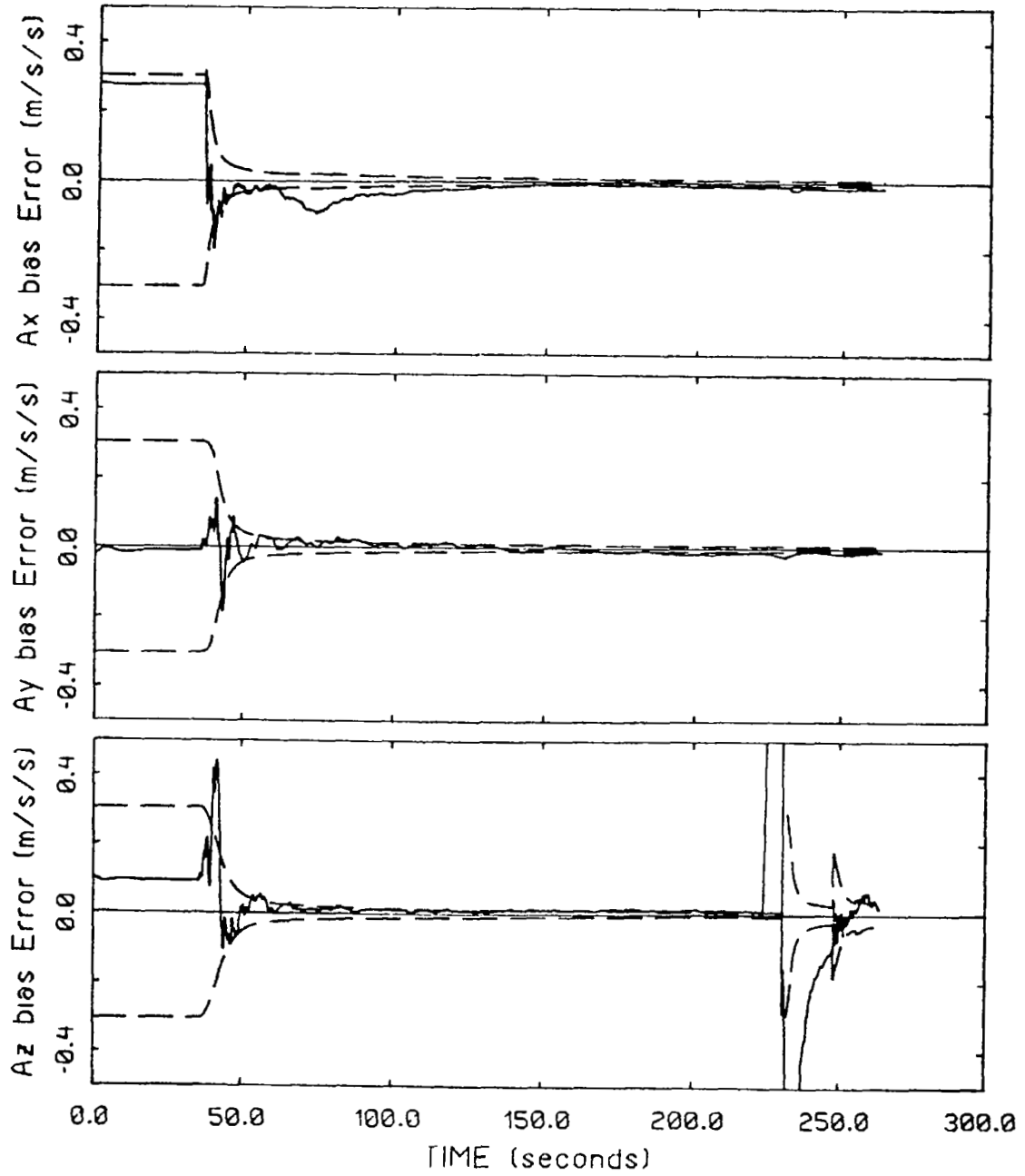
FIG. 34. ACCELEROMETER BIAS ESTIMATES ERROR – RAMP FAILURES

141
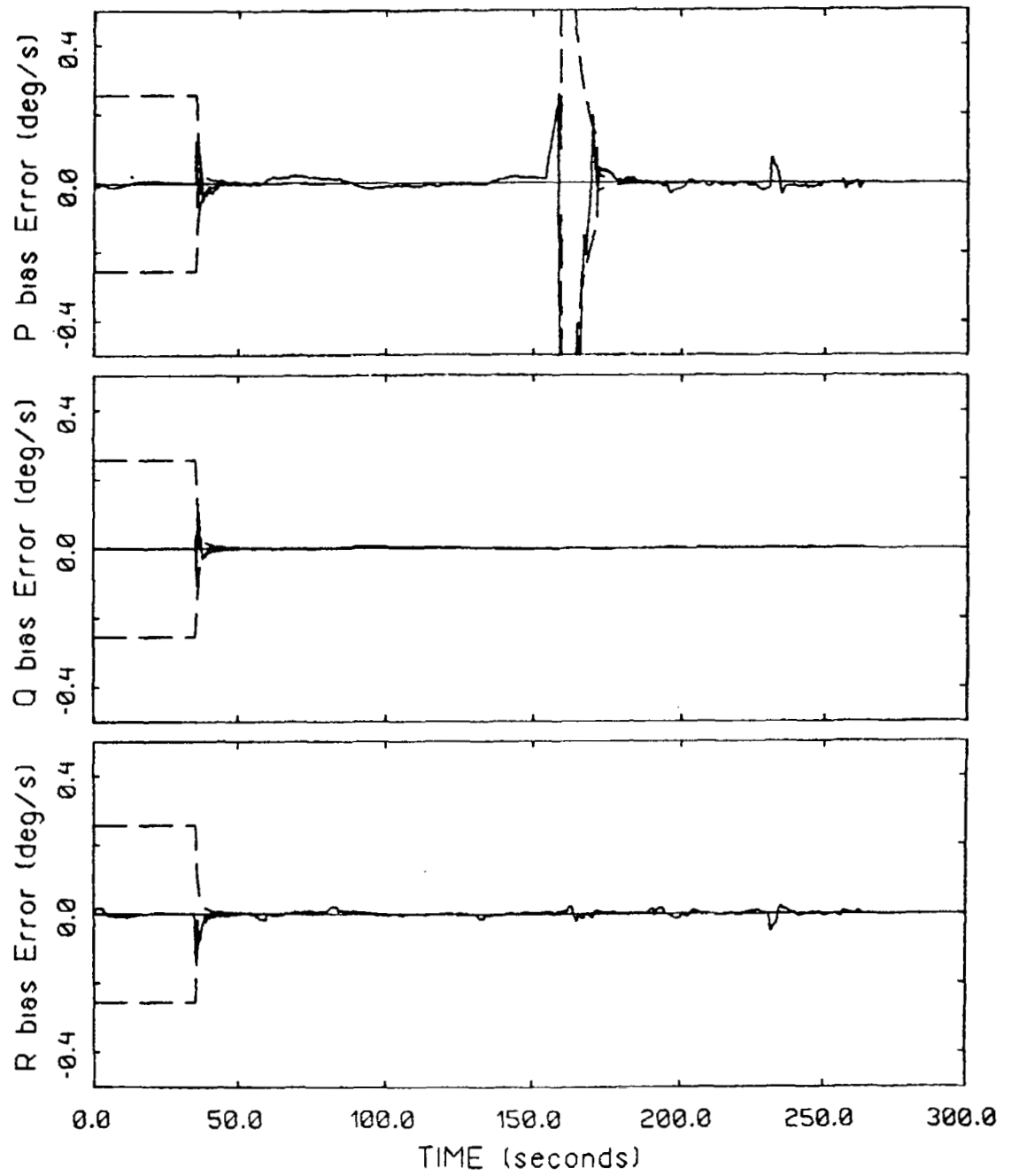
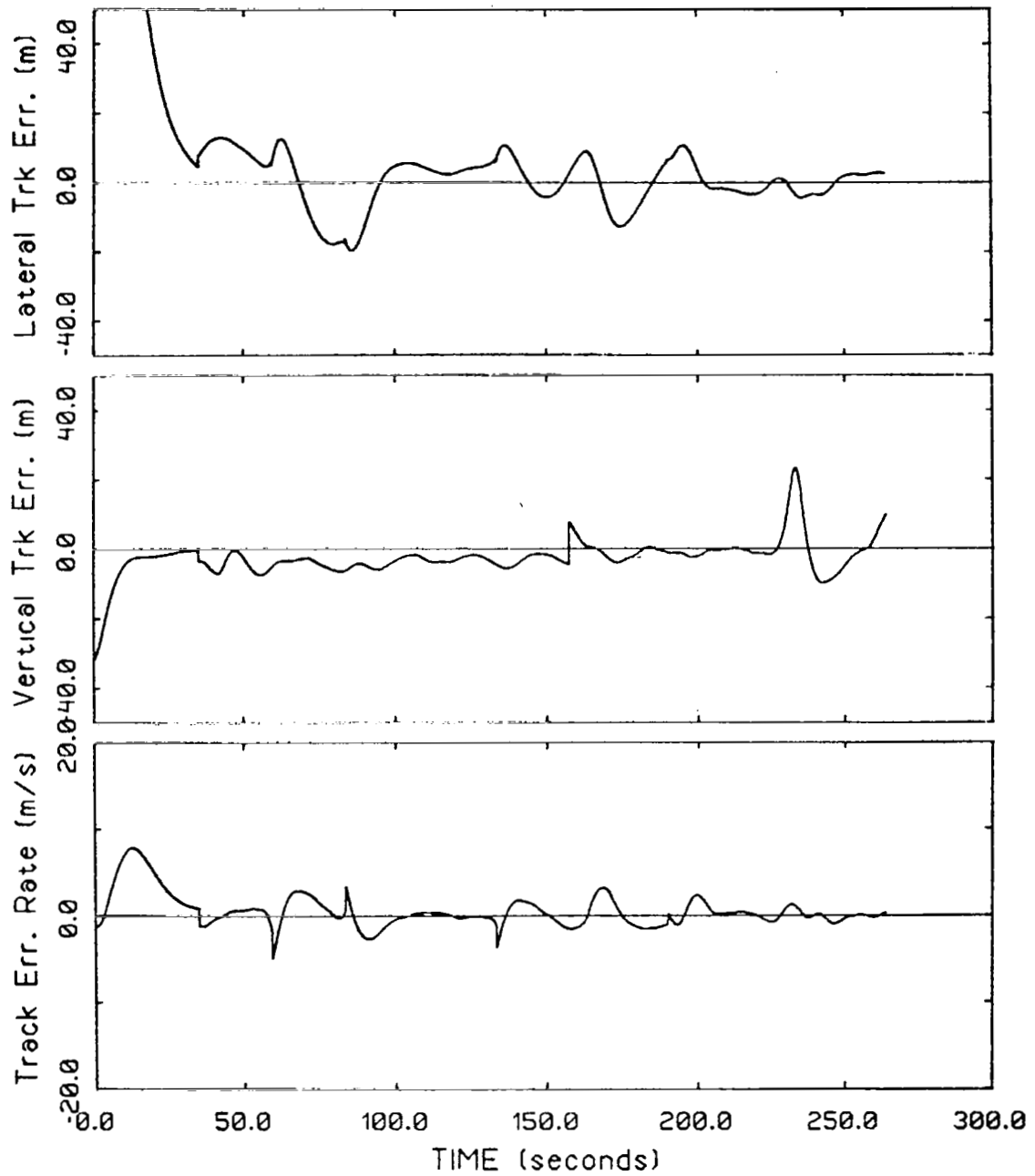FIG. 35. RATE GYRO BIAS ESTIMATION ERROR — RAMP FAILURES

142

FIG. 36.   TRUE A/C TRACK ERRORS – RAMP FAILURES

143

A typical run containing scale factor errors is shown in Table 14. Only input failures are considered, since they are the only sensors with normal operational scale factor errors. In this run, failures are simulated in A_y, A_z, P, and Q sensors. Surprisingly, all but the lateral accelerometer failures were correctly detected. The failure times for this run were chosen so that

- o P & Q failures occur when their respective signal levels are low
- o A_z failure occurs when its signal level is large (as it always is, since it measures the gravitational force)
- o A_y failure occurs when its signal is at a moderate and variable level.

From Table 14 we see that both gyro failures and the vertical accelerometer failure were all correctly detected, with detection times ranging from 0.9 to 3.5 seconds. The lateral accelerometer failure was not detected; this could be caused by the lateral accelerometer bias estimte tracking the failure before the input sensor failure estimates have time to converge.

The false alarms in Table 14 can be grouped into those that are a result of transients after the reset (ϕ,El,θ), and those that are due to the missed detection of the lateral accelerometer (Rng,A_x).

The estimation and A/C track error profiles are shown in Figures 37—42. In general, the filter estimates are much less fault tolerant for this type of failure. Some of the transients in the velocity, attitude and accelerometer profiles are unacceptably large. In addition, the A/C track errors were observed to be the largest for this failure type.

144

The estimation error curve for the lateral accelerometer bias looks very noisy. The signal plotted is actually the underlined effective bias due to the scale factor failure and all other unmodeled effects, however, due to a recording error the noise signal itself was not subtracted. Since this scale factor error curve consists of a time varying bias component and an increased noise component, the correct curve would look exactly like the one shown, but with the noise portion scaled down by the scale factor. One plausible reason for the FTS's inability to detect this failure is that the rms value of this curve is close to zero – violating the basic premise that a failure really is a bias shift in the signal. Even if there is a misdetection, as this example shows, the FTS algorithm can still identify other sensor failures.

3.4.5 Increased Noise Failures

Increased noise failures are by nature most different from the underlying bias failure assumptions used in the FTS. Since the rms value of the effective failure level is essentially zero, detections only occur if the noise level is large for a short period of time. This section discusses the performance of the fault tolerant system when increased noise failures are simulated.

Although, from the last section, we know that the healers are not adequate for these types of failure modes, in this section the healers were left operational in the FTS so that the reader would be exposed to this aspect of the problem.

Table 15 details the failure performance of a typical sample run containing increased noise failures. The reader will notice that the detection time is for the first detection – since many false healings/re-detections occur in this run with the

TABLE 14.   DETECTION PERFORMANCE FOR INCREASED SCALE FACTOR FAILURES

SENSOR	FAILURE MAG.	FAILURE ONSET TIME (seconds)	TIME TO DETECT (seconds)
A_y-1	2.5 (10σ)	66.45	X
A_z-1	2.5 (10σ)	151.8	1.85
P-1	.1 (10σ)	110.3	0.9
Q-1	.1 (10σ)-	225.95	3.5

False Alarms   (Healers are turned off)

Sensor	Time of False Alarm (seconds)
Rng-2	106.65
φ-2	114.75
E1-2	154.15
A_x	181.1
θ	234.4
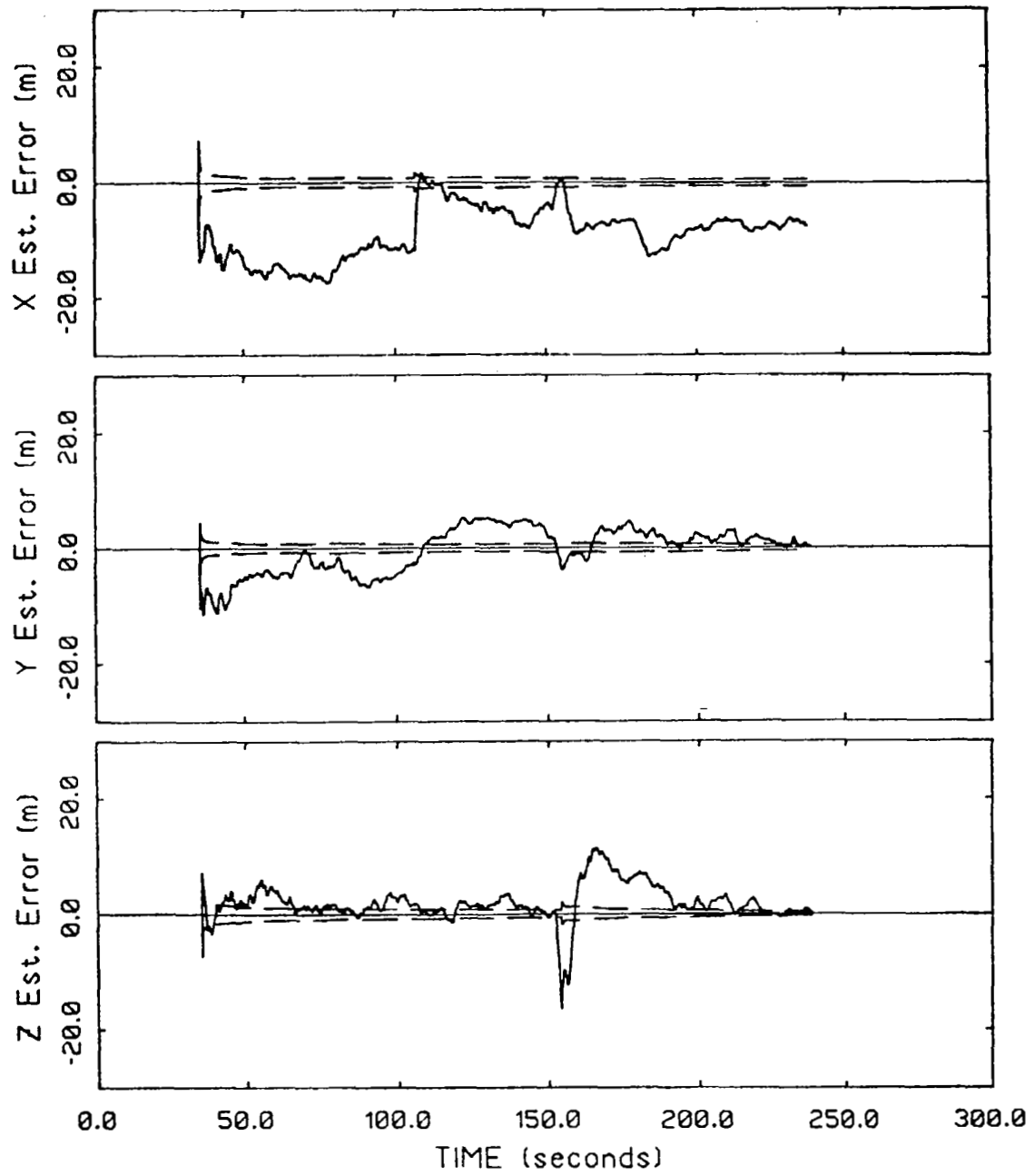
FIG. 37. POSITION ESTIMATION ERROR – INCREASED SCALE FACTOR FAILURES

147
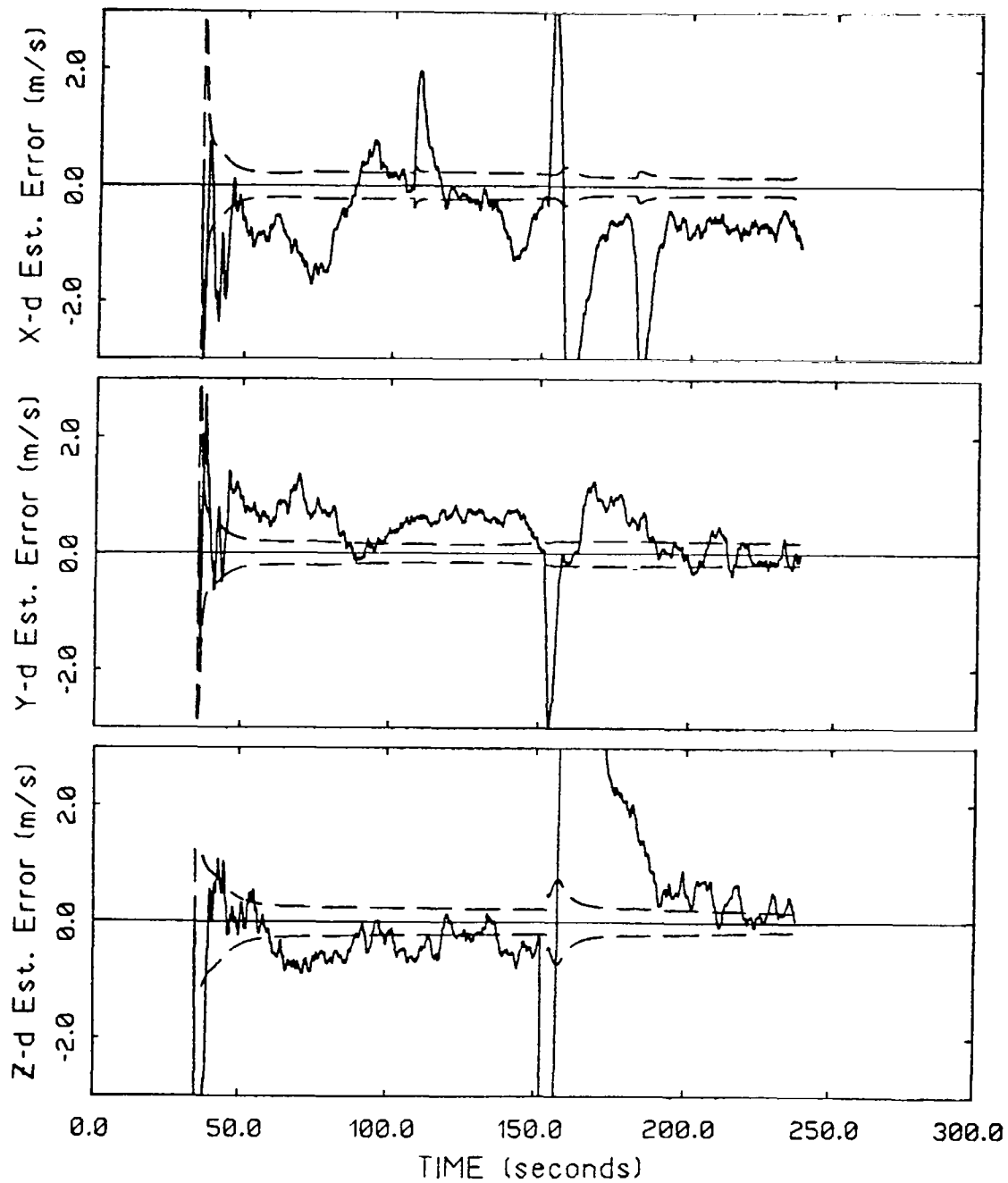
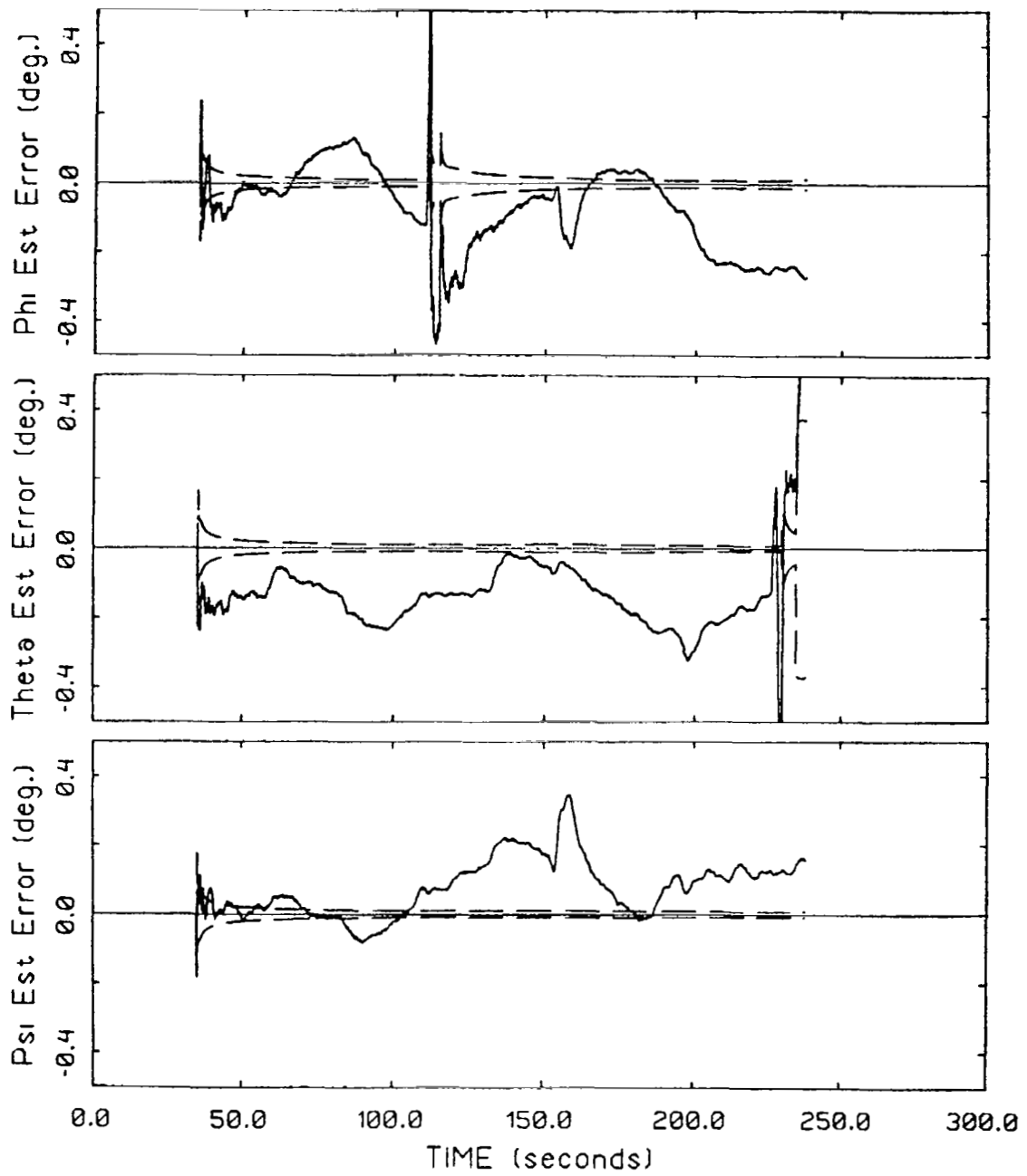FIG. 38. VELOCITY ESTIMATION ERROR – INCREASED SCALE FACTOR FAILURES

148

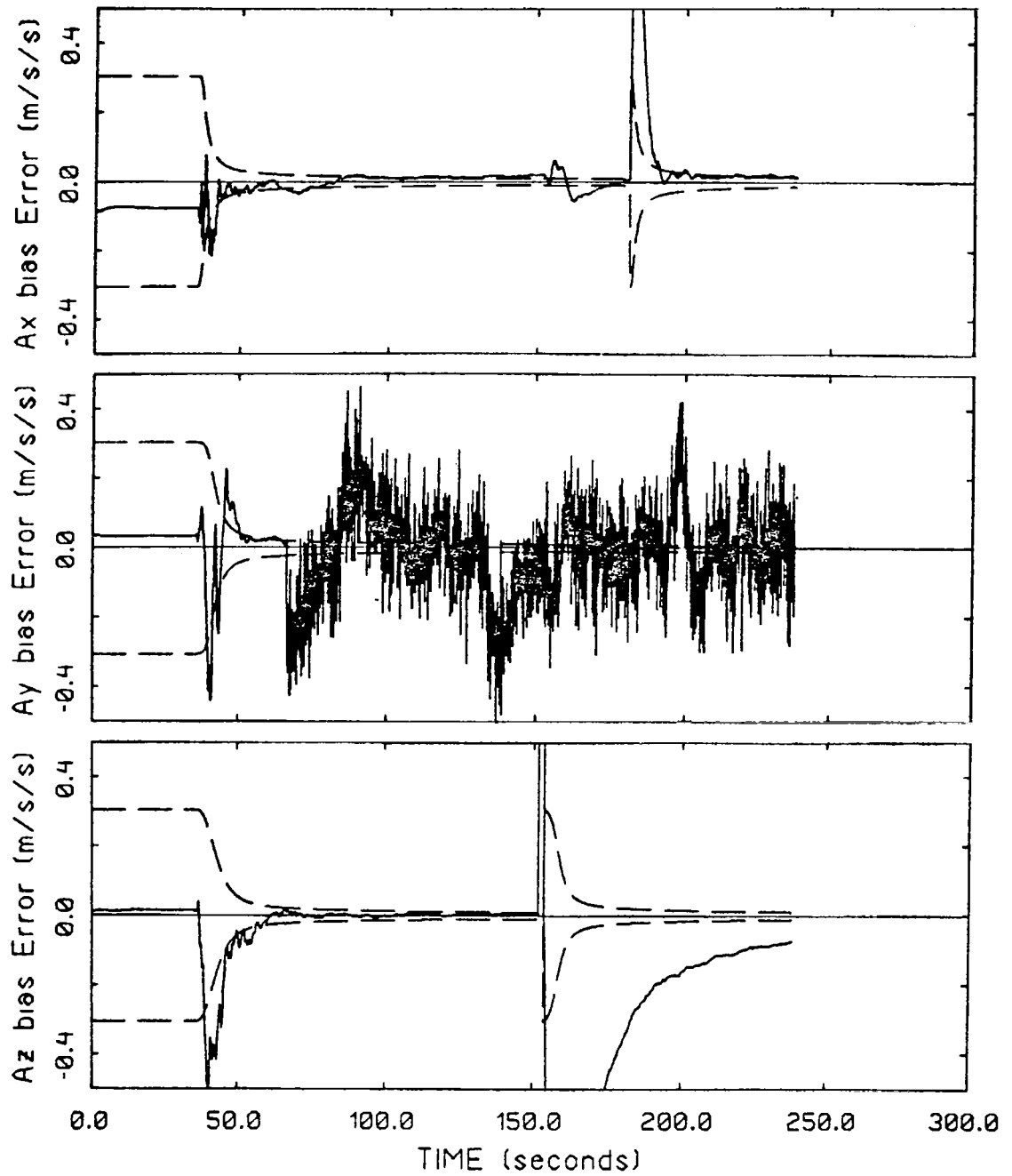FIG. 39. ATTITUDE ESTIMATION ERROR — INCREASED SCALE FACTOR FAILURES

149

FIG. 40. ACCELEROMETER BIAS ESTIMATION ERROR — INCREASED SCALE FACTOR FAILURES
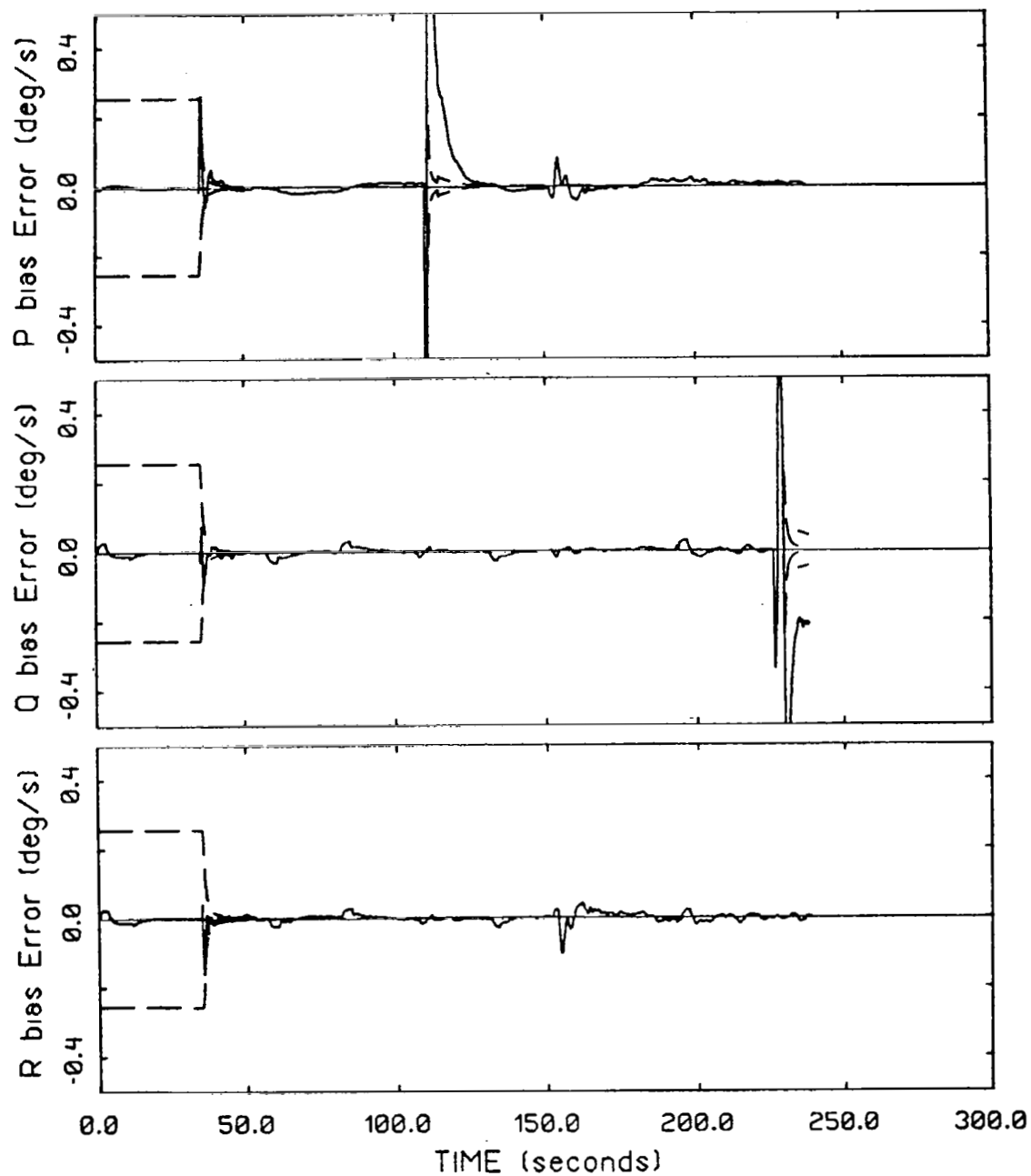
150

FIG. 41. RATE GYRO BIAS ESTIMATION ERROR — INCREASED SCALE FACTOR FAILURES
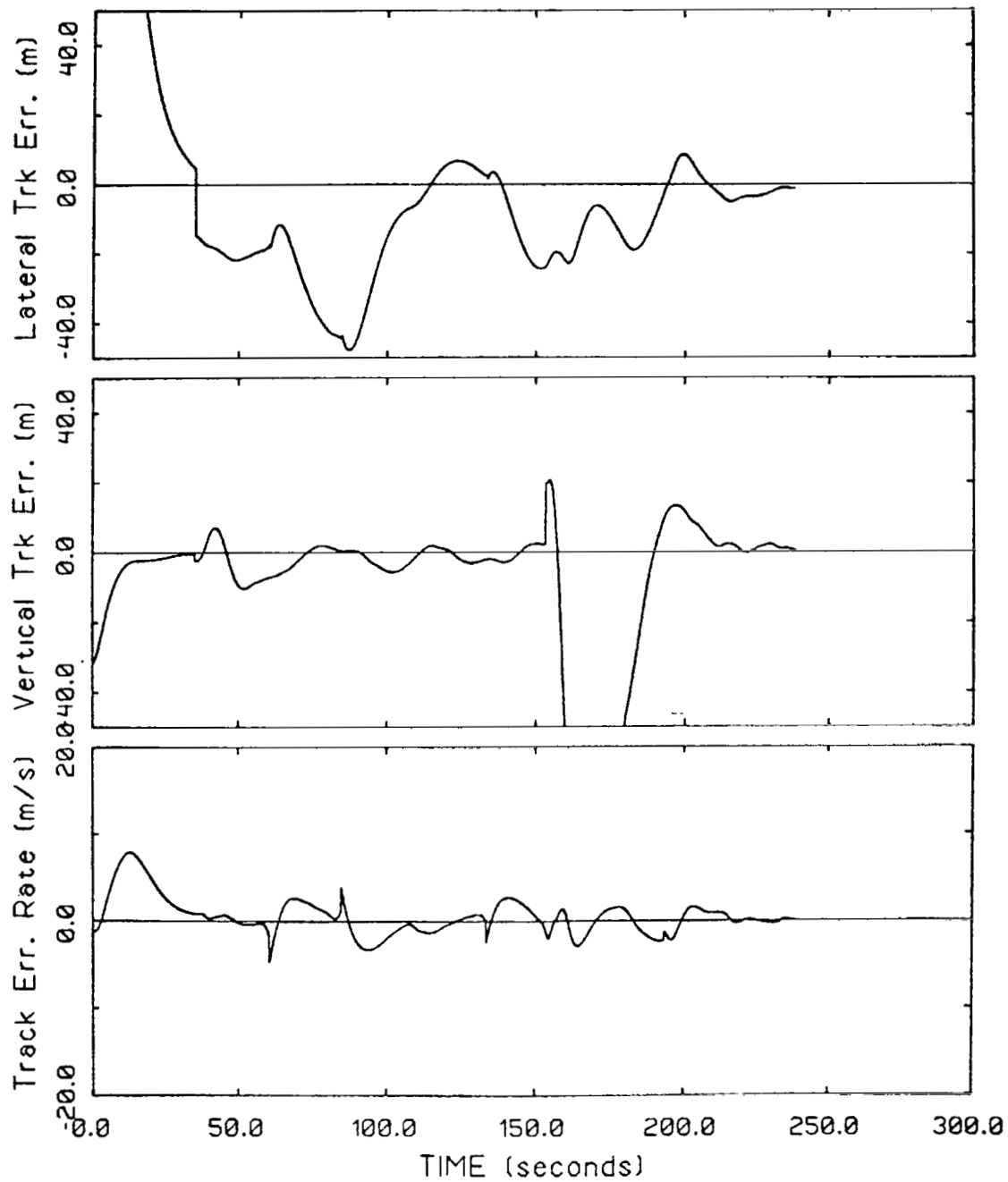
151

FIG. 42. TRUE A/C TRACK ERRORS — INCREASED SCALE FACTOR FAILURES

healers running. The detection times appear to be quite random for this class of failure, except that detection times are uniformly longer for input sensors (since they are smoothed by the no—fail filter) than for measurement sensors.

The important finding is that all the failures were correctly detected and that there were no false alarms in the run. Moreover, the constant healing and detection with its associated filter reconfiguration and reset did not interfere with correctly detecting the failures. Certainly the detection times for inputs are relatively long, but the fact that the simple bias jump model works at all is encouraging.

Figures 43—47 show the error estimation profiles for this run. The impact of the successive resets can be clearly seen in these figures. Notice that the errors are bounded and the filter doesn't diverge due to these repetitive disturbances. In fact, if a more robust healer — or no healer at all — were used the performance would be somewhat better.

3.5 Performance Summary and Overall Evaluation

The previous four sections have described in detail the performance of the developed fault tolerant system under a variety of simulated failure conditions. In these sections, characteristics, which appeared in our study to be common to a particular class of failure, were highlighted for discussion. This section will attempt to generalize the major findings.

Table 16 shows an overall performance summary under the different simulated failure conditions. As can be seen from the first column, there was complete coverage

TABLE 15.   DETECTION PERFORMANCE FOR INCREASED NOISE FAILURES

SENSOR TYPE	FAILURE MAG.	FAILURE ONSET TIME (seconds)	TIME TO (FIRST) DETECT (seconds)
A_z-1	.98m/s/s (10σ)	113.05	39.65
P-1	.2°/s (10σ)	68.35	55.75
E1-1	.3°/s (10σ)	152.6	0.2
θ-1	2.3° (10σ)	223.35	2.0

Incorrect Healings/Detections

Sensor	Number of Occurrences
A_z-1	4
P-1	None
E1-1	16
θ-1	8

TABLE 16. OVERALL PERFORMANCE SUMMARY

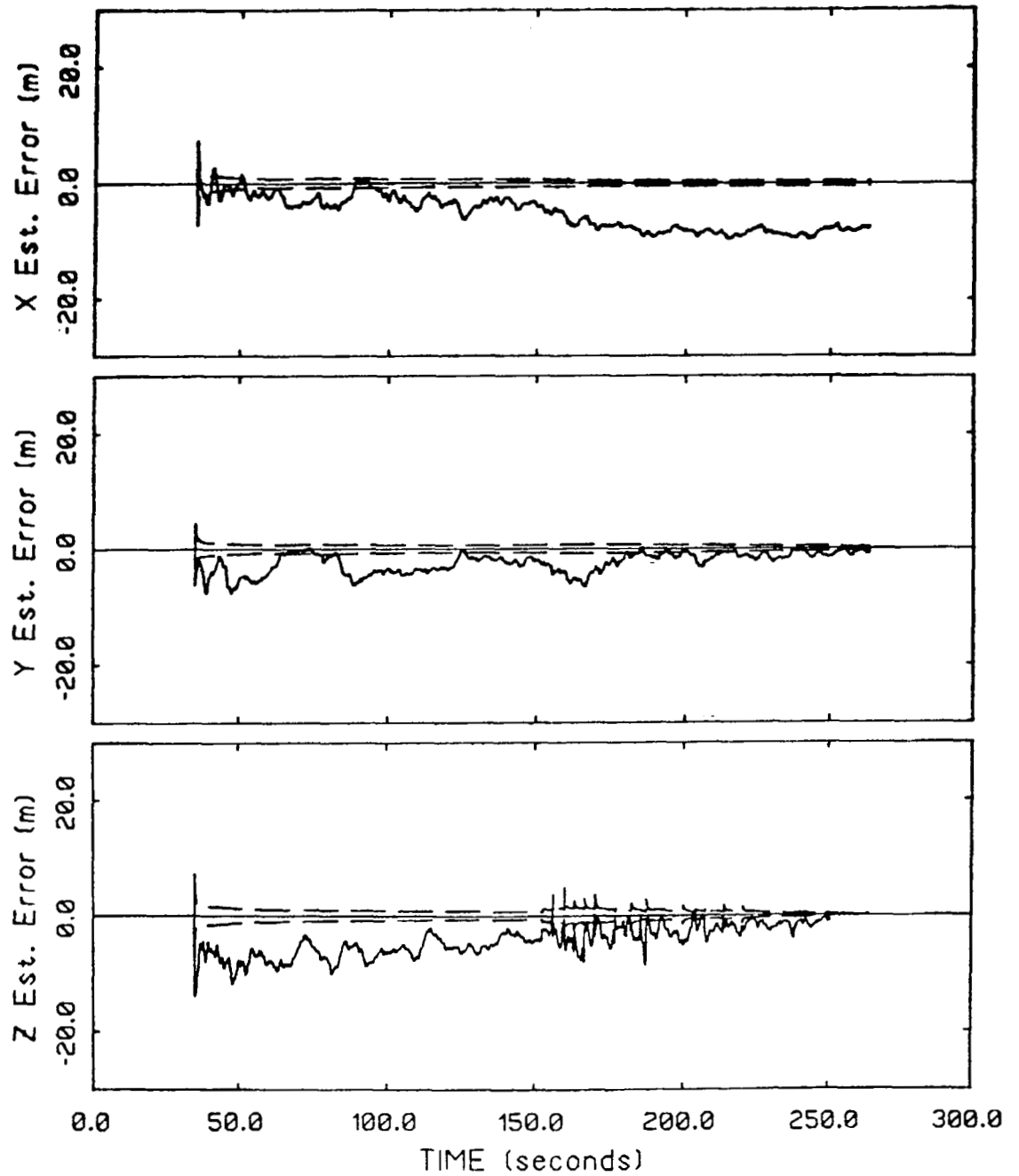| SENSOR / FAILURE TYPE	SINGLETON BIAS (SEC)	INCREASED NOISE (SEC)	INCREASED SCALE FACTOR (SEC)	NULL FAILURE (SEC)	HARDOVER FAILURE (SEC)	MULTIPLE FAILURE (SEC)	RAMP FAILURE (SEC)	RSF IMU SINGLETON BIAS (SEC)
A_x	4.65							
A_y	14.75							
A_z	N.D.	42.5	1.9		1.5		6.6	
P	13.32	55.75	0.79		0.05		4.5	
Q	8.12		0.5	0.7				
R	6.1							
Azm	0.9					0.637	2.3	0.42
E1	0.15	1.5				0.2		0.18
Rng	0.08					0.15		0.09
IAS	1.57				0.05			0.4
φ	0.57			0.15				
θ	0.43	3.3			0.08		1.9	
ψ	0.8							
RA	0.3							0.15

155

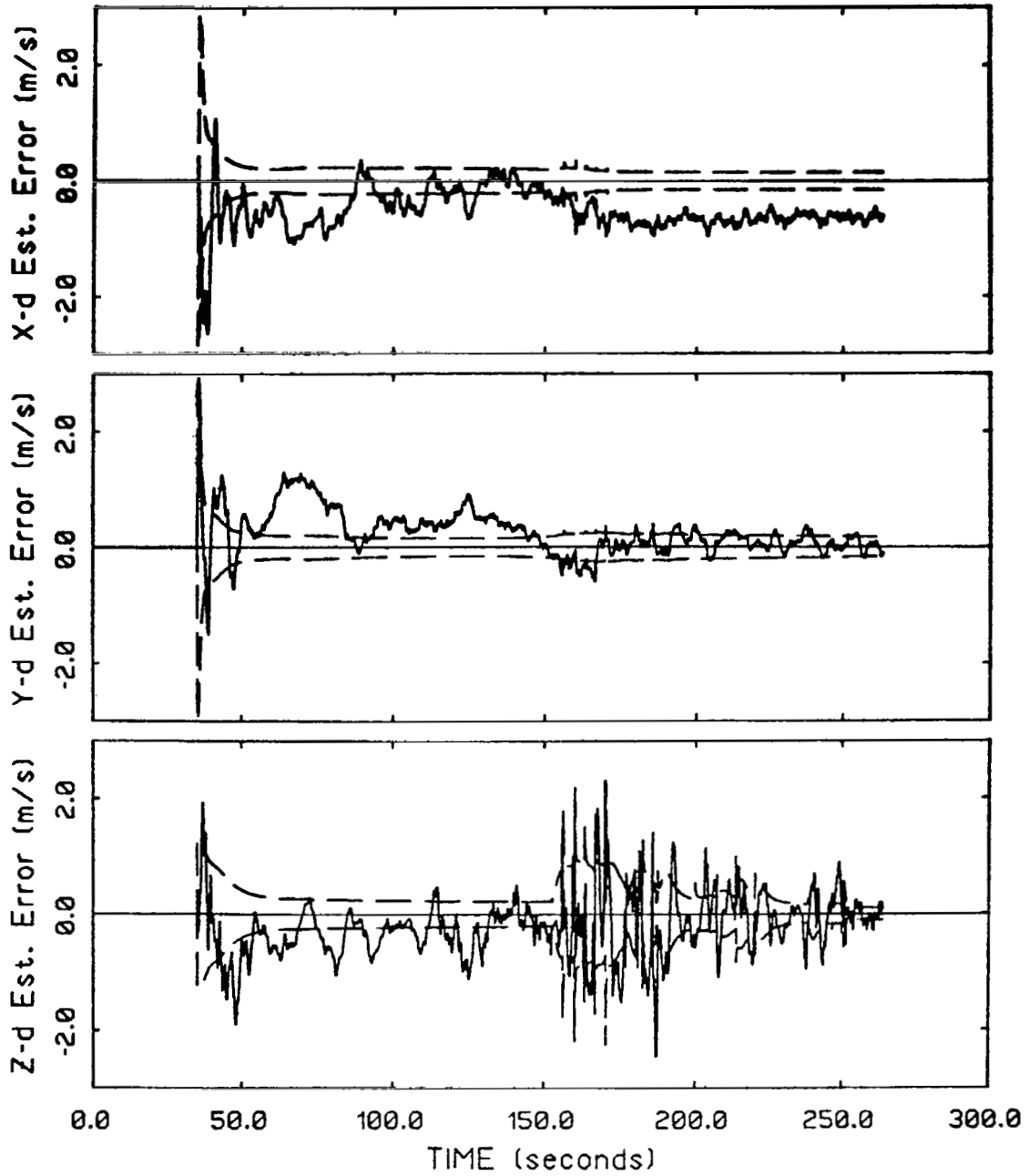FIG. 43. POSITION ESTIMATION ERROR – INCREASED NOISE FAILURES

156

FIG. 44. VELOCITY ESTIMATION ERROR – INCREASED NOISE FAILURES
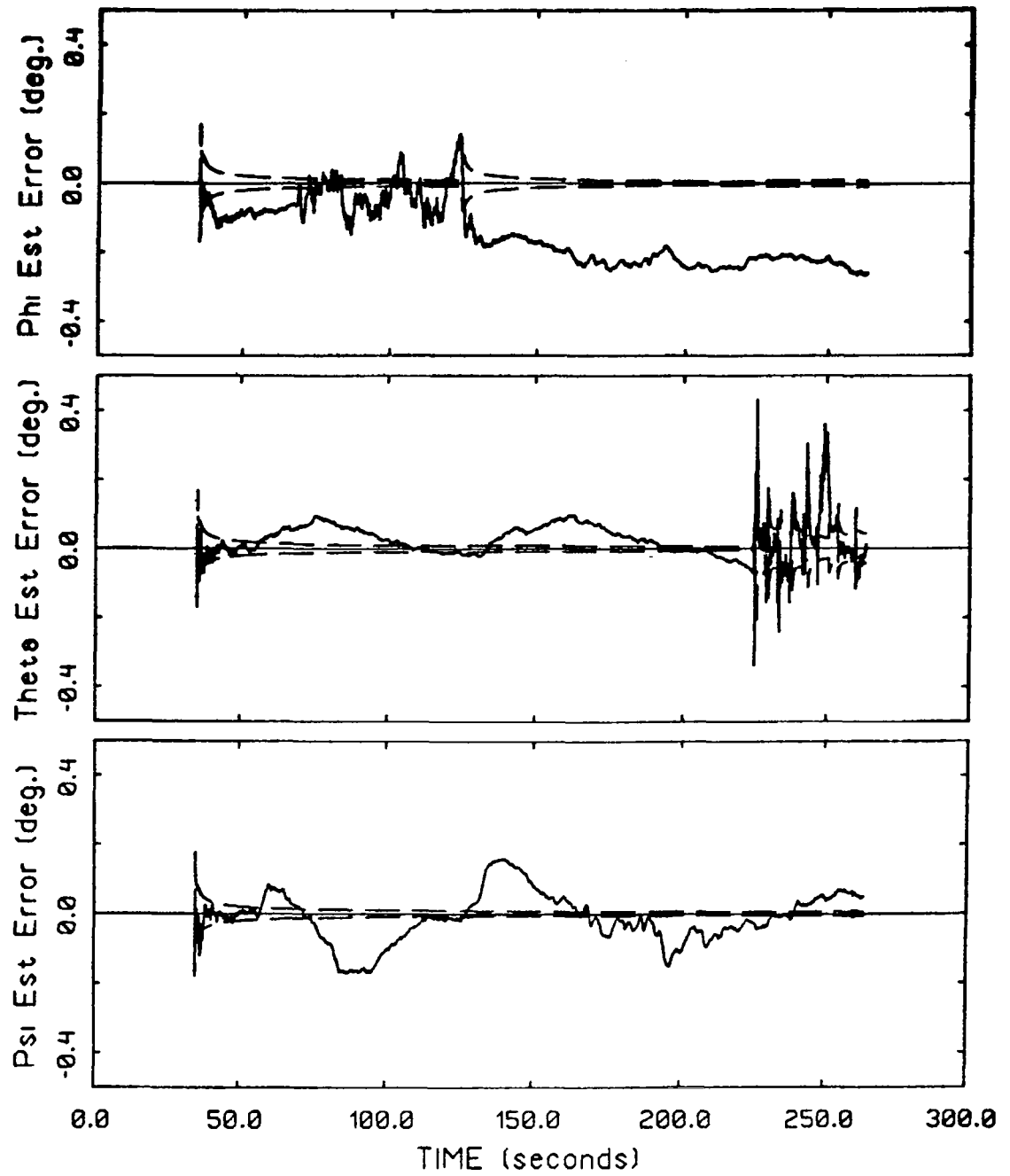
157

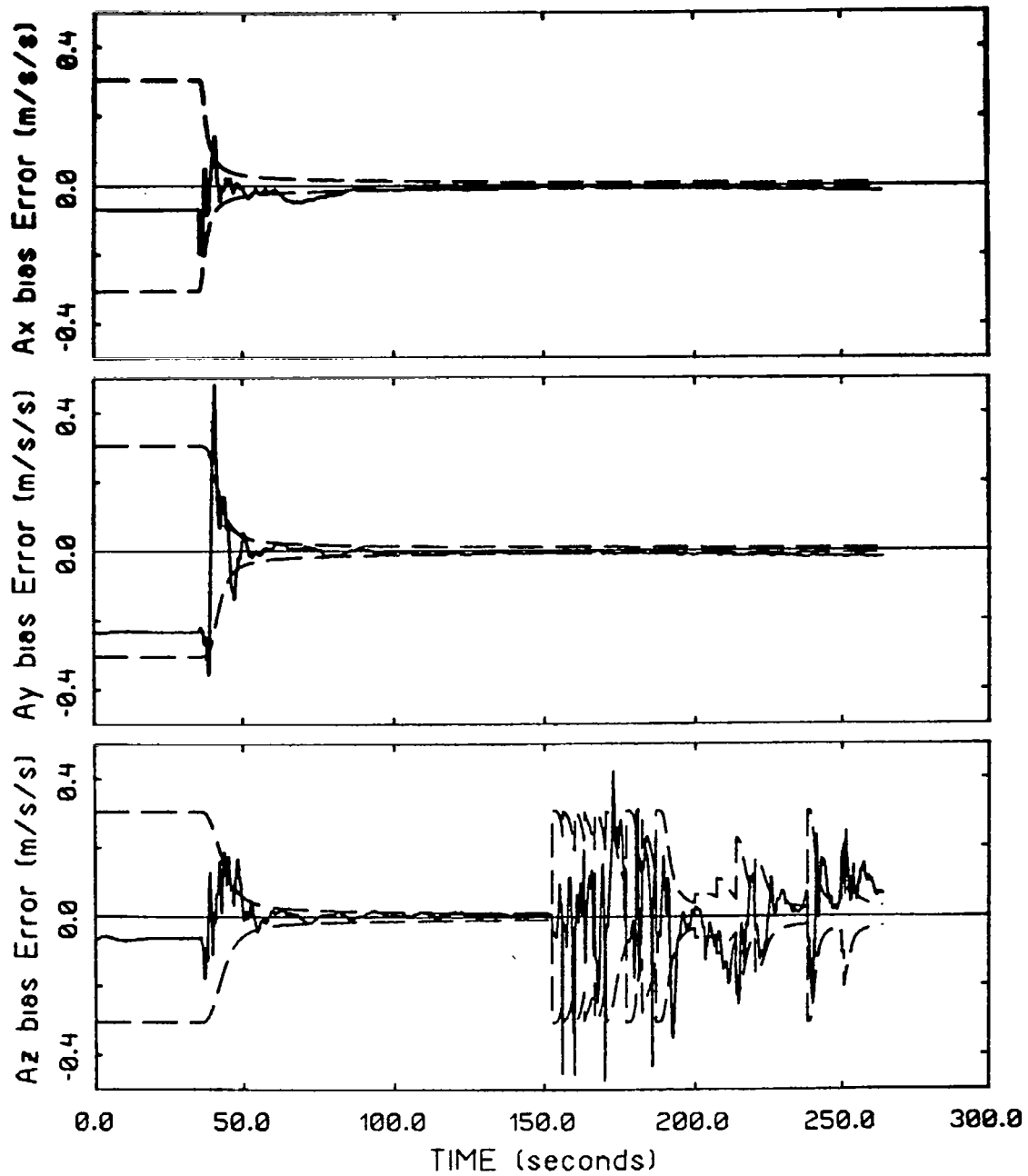FIG. 45. ATTITUDE ESTIMATION ERROR – INCREASED NOISE FAILURES

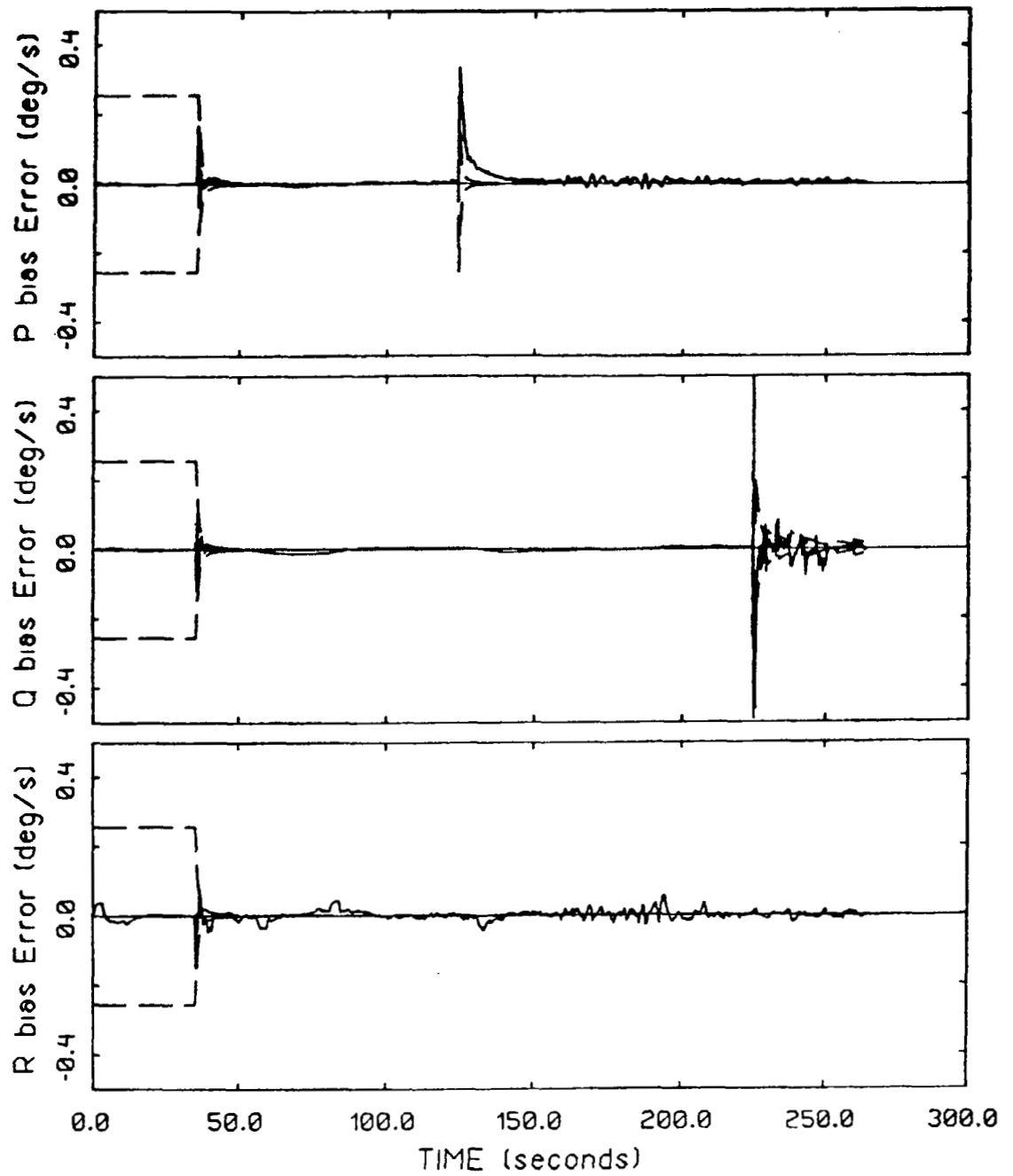FIG. 46.   ACCELEROMETER BIAS ESTIMATION ERROR·—   INCREASED NOISE FAILURES

159

FIG. 47. RATE GYRO BIAS ESTIMATION ERROR – INCREASED NOISE FAILURES

160

for bias type failures with the exception of A_z. The numbers in Table 16 depict the average time to detect over the available runs. Singleton bias failures were averaged over $3\sigma, 5\sigma, 8\sigma, 10\sigma$, and 12σ bias levels. Referring to Table 16, input sensor (accelerometer and rate gyro) bias failures were detected and isolated much slower than the other sensors. That is, for the same level bias type failure, the detection speed for input sensors were about 10 times slower than that of the measurement (MLS,IMU,IAS,RA) sensors. This is because the input sensor failures have to propagate through the no-fail filter dynamics in order to get detected. In our study, the redundancy for the input sensors were utilized only for backup. Hence, the comparison of like input sensors may improve the detection speed for these instruments.

The second column in Table 16 shows the average time-to-detect for 10σ increased noise type failures. For these failures, the detection speed for input sensor failures was about 20 times slower than that of measurement sensor failures. On the other hand, the average time-to-detect for the 10σ level increased noise input sensor failures is about 10 times slower than that for $3-5\sigma$ bias type failures. Similarly, for measurement sensors, the average time-to-detect in the increased noise failure case is about twice as slow as that in the bias type failure case. This is to be expected since the bias type failure model used by the detectors is a poor model for increased noise type failures. As before, if dual or more redundancy is available for the input sensors then the comparison monitoring of like sensors may improve the detection performance.

The third column in Table 16 shows the average time-to-detect for increased

scale factor failures with a 10σ level. Increased scale factor failures were easier to detect than increased noise type failures since the bias–type failure model is adequate in emulating the sensor output behavior under a scale factor failure. Columns four and five show the average time–to–detect for null and hardover failures. Since hardover and null (if there is a sizable signal level in the measurement) failures look like a large increased bias jump to detectors, they were identified within approximately to 2 sampling intervals. The relatively longer time–to–detect for null failure in the rate gyros indicates the need for a vehicle maneuver to detect this type of failure.

Multiple sensor failure detection performance levels are given in the next column. As discussed in Section 2.4.2 simultaneous failures were considered only for the MLS sensors since these failures represent MLS antenna failures. The detection speed for multiple failures is comparable to that for the singleton bias failures. It is worth noting that a simultaneous MLS sensor failure never introduced a false alarm arising from the selection of a singleton failure in the associated instrument. The detection performance for ramp failures is given in the last column. The failures correspond to a ramp failure level equal to 3 times the normal operating bias in one second. The detection speed for ramp failures was slightly slower than that for the corresponding bias–type failures.

The final column summarizes bias failure performance using the RSDIMU sensor configuration. In this case note that detection times are comparable to those obtained with the standard configuration. In addition, no false alarms or missed detections were observed with the RSDIMU configuration.

Based on our analysis of all the simulation runs made for this study, the following summary of the operation and performance of the current fault tolerant system is outlined.

o The no-fail filter's state estimates were "fault tolerant" — Filter divergence, caused by failures, occurred infrequently and the absolute level of filter errors was within tolerable bounds for the automatic landing system to function properly.

o The reconfiguration logic worked very well for bias failures — providing selective, moderate resets. For non-bias failures, resets were generally too hard. Suggestions have been offered for resolving these problems.

o Healer mechanisims worked well for bias, hardover, and ramp failures, but were inadequate — in its present form for other types of failure modes.

o As documented in this chapter, detection speeds were quite good for bias failures (with failure levels between 3 and 10σ), with detection times between 4.65 and 14.65 seconds for input sensors, and 0.08 to 1.57 seconds for measurement sensors.

o Detection speeds for non-bias failures were in general adequate. For increased noise failures the time-to-detect was random.

o The FTS was generally able to distinguish between failures in dynamically coupled sensors, (for example, ϕ and P). However, the uncompensated normal operating errors in the sensors at times exacerbated this problem.

o False alarms were usually due to either:

  . uncompensated normal operating errors

  . filter resets which were too "hard"

  . occasional indistinguishability of dynamically coupled sensors

  . secondary effect of a missed detection.

o Missed detections were usually due to either:

  . the normal operating bias filter estimating out its effects

  . a simulated failure level that was too low relative to the signal, noise, and bias levels for that sensor

  . in the case of hardover and null failures, if the failure was not

163

detected very quickly the filter would diverge.

o The method was observed to be surprisingly robust with regards to non−bias type failures. However, explicit tests for non−bias failure, (either model based or heuristic in nature) would be required for reliable detection.

o Use of the RSDIMU configuration provided:

  . slightly better estimation performance

  . no false alarms or missed detections were observed

  . detection times were comparable to the standard configuration.

164

4. CONCLUSIONS AND RECOMMENDATIONS

In this study, we have developed a fault tolerant system design methodology for general nonlinear stochastic dynamic systems. In particular, we have applied the developed methodology to the design of a sensor fault tolerant system in aircraft navigation, guidance, and control systems in a Microwave Landing System environment. The fault tolerant system provides aircraft position, velocity, attitude, and sensor normal operating bias estimates tolerant of faults in the ground−based navigation aids, and on−board flight control and navigation sensors. We have analyzed the estimation and failure detection performance of the software implementation of our design (called FINDS) on the six−degree−of−freedom nonlinear digital simulation of the ATOPS aircraft.

The state and normal operating sensor bias estimation performance of the separated EKF algorithm in FINDS compared favorably with that of other navigation filters employed within the same environment. Although sensor failures are modelled as bias jumps in FINDS, we have investigated the failure detection performance on other types of sensor failures as well. The failure detection and isolation performance of FINDS was excellent for bias failures with the detection speed considerably better for measurement sensors such as MLS than for input sensors such as accelerometers. The failure detection performance for non−bias type failures was surprisingly good, although healing tests designed for bias type failures caused problems, especially for increased noise type failures. The detection speed for catastrophic failures such as hardover was extremely fast.

The following is a list of our recommendations for future study:

o Although the simulation employed was fairly realistic, it did not account for structural modes and lever arm effects. We believe that the sensitivity of the FTS to these modelling errors can be uncovered by testing FINDS using flight recorded data.

o In this study, computational efficiency was not a major concern. Hence, the developed algorithm does not currently run in real-time. Analysis of the simulation data suggests that there are quite a number of simplifying assumptions which can be introduced into the filter/detector structure for significantly improving computational efficiency.

o There are a number of places in FINDS which can be modified for improved failure detection performance. For instance, a better internal model for wind dynamics, the use of standby input sensors for failure detection, and a better integration routine for the kinematic equations are such examples.

o Although the developed sensor failure detection algorithm, which is designed for bias type failures, proved to be capable of identifying other types of non-bias sensor failures, it is desirable to be able to classify these non-bias failures as well. Such a classification would be useful in employing different types of healing tests for different types of sensor failures.

o Although FINDS was tested within an MLS environment, the developed methodology is quite general and applicable to other types of sensor environments as well. For instance, FINDS can be applied to navigation under VORTAC, and satellite position fixing systems with appropriate modifications.

5. REFERENCES

[1] Caglayan, A.K. and Lancraft, R.E., "An Aircraft Sensor Fault Tolerant System", NASA CR-165876, April 1982.

[2] Lancraft, R.E. and Caglayan, A.K., "FINDS: A Fault Inferring Nonlinear Detection System: User's Guide", NASA CR-172199, September 1983.

[3] Duff, W.G. and Guarino, C.R., "Refinement and Validation of Two Digital Microwave Landing System (MLS) Theoretical Models", NASA CR-132713, August, 1975.

[4] Pines, S., "Terminal Area Automatic Navigation, Guidance, and Control Research Using the Microwave Landing System (MLS), Part 2 – RNAV/MLS Transition Problems for Aircraft", NASA CR-3511, January 1982.

[5] Pines, S., "Terminal Area Automatic Navigation, Guidance, and Control Research Using the Microwave Landing System (MLS), Part 3 – a comparison of Waypoint guidance algorithms for RNAV/MLS Transition", NASA CR-3512, January 1982.

[6] Caglayan, A.K. and Lancraft, R.E., "A Bias Identification and State Estimation Methodology for Nonlinear Systems", Proc. of the 6th IFAC Symposium on Identification and System Parameter Estimation, Washington, D.C., June 7-11, 1982.

[7] Caglayan, A.K. and Lancraft R.E., "A Separated Bias Identification and State Estimation Algorithm for Nonlinear Systems", Automatica, Vol. 19, No. 5, pp. 561-570, September 1983.

[8] Morrell, F.R. and Russell, J.G., "Design of a Developmental Dual Fail Operational Redundant Strapped Down Inertial Measurement Unit", NAECON 1980.

[9] Britting, K.R., Inertial Navigation Systems Analysis, John Wiley & Sons, Inc., New York, 1971.

[10] Etkin, B., Dynamics of Atmospheric Flight, John Wiley & Sons, Inc., New York, 1972.

[11] Halyo, N. and Caglayan, A.K., "A Separation Theorem for the Stochastic Sampled-Data LQG Problem", International J. of Control, Vol. 23, No. 2, pp. 237-244, February 1976.

[12] Caglayan, A.K., "Necessary and Sufficient Conditions for Detectability of Jumps in Linear Systems", IEEE Trans. on Automatic Control, Vol. AC-25, No. 4, pp. 833-836, August 1980.

[13] Caglayan, A.K., "Simultaneous Failure Detection and Estimation in Linear Systems", Proc. of 1980 Conf. on Decision and Control, December 10-12, 1980, Albuquerque, NM.

[14] Jazwinski, A.H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

[15] Anderson, B.D.O. and Moore, J.B., Optimal Filtering, Prentice-Hall, Inc., New Jersey, 1979.

[16] Pines, S., Schmidt, S.F., and Mann, F., "Automated Landing, Rollout and Turnoff Using MLS and Magnetic Cable Sensors", NASA CR-2907, Oct. 1977.

[17] Halyo, N., "Development of a Digital Guidance and Control Law for Steep Approach Automatic Landings Using Modern Control Techniques", NASA CR-3074, February 1979.

[18] Van Trees, H.L., Detection Estimation and Modulation Theory, Academic Press, New York 1970.

[19] G.J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, New York 1977.

[20] Montgomery, R.C. and Caglayan, A.K., "Failure Accommodation in Digital Flight Control Systems by Bayesian Theory", J. of Aircraft, Vol. 13, No. 2, pp. 69-75, February 1976.

[21] Montgomery, R.C. and Price, D.B., "Failure Accommodation in Digital Flight Control Systems Accounting for Nonlinear Aircraft Dynamics", J. of Aircraft, Vol. 13, No. 2, pp. 76-82, February 1976.

[22] Willsky, A.S., "A Survey of Design Methods for Failure Detection in Dynamic Systems", Automatica, Vol. 12, No. 6, pp. 601-611, June1976.

[23] Motyka, P., Landey, M. and McKern, R., "Failure Detection and Isolation Analysis of a Redundant Strapdown Inertial Measurement Unit", NASA CR-165658, February 1981.

[24] Cunningham, T. et.al., "Fault Tolerant Digital Flight Control with Analytic Redundancy", AFFDL-TR-77-25, May 1977.

[25] Deckert, J.C., Desai, M.N., Deyst, J.J., Jr., and Willsky, A.S., "Reliable Dual-Redundent Sensor Failure Detection and Identification for the NASA, F-8 DFBW Aircraft, NASA CR-2944, February 1978.

[26] Bryson, A.E. and Henrikson, L.J., "Estimation Using Sampled-Data Containing Sequentially Correlated Noise", J. of Spacecraft and Rockets, Vol. 5, pp. 662-665, June 1968.

[27] Willsky, A.S. and Jones, H.L., "A Generalized Likelihood Ratio Approach to the Detection and Estimation Jumps in Linear Systems", IEEE Trans. on Auto. Contr., Vol. AC-21, No. 1, pp. 108-112, February 1976.

[28] Kushner, H., Introduction to Stochastic Control, Holt, Rinehart and Winston, Inc., 1971.

[29] Caglayan, A.C., and Lancraft, R.E., "Reinitialization Issues in Fault Tolerant Systems", Proceedings of the ACC, June 1983

APPENDIX A
DEFINITIONS OF USEFUL QUANTITIES

This appendix basically defines the quantities used in the no-fail filter and detector derivations. Specifically, the following quantities are defined herein:

o ω_G

o T_{GL}, T_{LB}, T_{GB}

o T_{IG}, T_{BI}

o Γ_ω

A set of abreviations, used to condense the descriptions, are given below:

o $s\theta$ = $sine(\theta)$

o $c\theta$ = $cosine(\theta)$

o $t\theta$ = $tangent(\theta)$

o $sec\theta$ = $secant(\theta)$

o λ_G =   Latitude to origin of G frame

Descriptions:

o ω_G = $[c\lambda_G, \; 0, \; -s\lambda_G \;]\omega^E$

o T_{GL} = I , assuming a locally flat earth.

o T_{LB} = T_{GB}

$$= \begin{bmatrix} c\theta c\sigma & s\phi s\theta c\sigma - c\phi s\sigma & c\phi s\theta c\sigma + s\phi s\sigma \\ c\theta s\sigma & s\phi s\theta s\sigma + c\phi c\sigma & c\phi s\theta s\sigma - s\phi c\sigma \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

171

where $\sigma = \psi - \psi_R$

$$\circ \; T_{IG} = \begin{bmatrix} c\lambda_G & 0 & -s\lambda_G \\ s(\omega t)s\lambda_G & -c(\omega t) & s(\omega t)c\lambda_G \\ -c(\omega t)s\lambda_G & -s(\omega t) & -c(\omega t)c\lambda_G \end{bmatrix}$$

$$\circ \; T_{BI} = T_{BG}T_{GI} = T_{GB}^{'}T_{IG}$$

$$\circ \; \Gamma_\omega = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec\theta & c\phi \sec\theta \end{bmatrix}$$

4. Title and Subtitle		5. Report Date
A Fault Tolerant System for an Integrated Avionics Sensor Configuration		September 1984
		6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Alper K. Caglayan and Roy E. Lancraft	Report No. 5352
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238	NAS1-16579
	13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code
	505-34-13-12

15. Supplementary Notes

Langley technical monitor: Frederick R. Morrell
Final Report

16. Abstract

This report describes the development of an aircraft sensor fault tolerant system methodology for the Transport Systems Research Vehicle in a Microwave Landing System (MLS) environment. The fault tolerant system provides reliable estimates in the presence of possible failures both in ground-based navigation aids, and in on-board flight control and intertial sensors. Sensor failures are identified by utilizing the analytic relationships between the various sensors arising from the aircraft point mass equations of motion.

The estimation and failure detection performance of the software implementation (called FINDS) of the developed system has been analyzed on a nonlinear digital simulation of the research aircraft. Simulation results showing the detection performance of FINDS, using a dual redundant sensor compliment, are presented for bias, hardover, null, ramp, increased noise and scale factor failures. In general, the results show that FINDS can distinguish between normal operating sensor errors and failures while providing an excellent detection speed for bias failures in the MLS, indicated airspeed, attitude and radar altimeter sensors.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
sensor failure detection, nonlinear estimation, fault tolerant navigation, analytic redundancy, MLS, no-fail filter, EKF, reconfiguration	LIMITED DISTRIBUTION Until September 1985 Subject Category 04

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	193	