NASA CONTRACTOR REPORT 178043

A PRELIMINARY DESIGN FOR FLIGHT TESTING THE FINDS ALGORITHM

A. K. CAGLAYAN AND P.M GODIWALA

CHARLES RIVER ANALYTICS INC.
CAMBRIDGE, MA 02138

CONTRACT NAS1-17719
MARCH 1986

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

This supplemental Final Report presents the progress made in NASA Contract NAS1-17719, entitled "Evaluation of Fault Tolerant Concepts", over the results reported in [1]-[2]. The main objective of the effort in this phase has been to reduce the program size of the FINDS (Fault Inferring Nonlinear Detection System) algorithm [6], and to increase its execution speed without compromising on the aircraft state and sensor bias estimation performance or failure detection/isolation performance presented in [1]-[2]. The modified algorithm has been tested using about five minutes of sensor flight data for the NASA ATOPS B-737 aircraft in a Microwave Landing System (MLS) environment.

This report summarizes the modifications made to the FINDS algorithm in order to achieve the objectives of the current study. In summary, these modifications have resulted in a considerably smaller program size accommodating government flight computer constraints, and a faster execution speed allowing near real-time operation. In addition, the changes resulted in a significant improvement in the estimation and failure detection performance results reported in [1]-[2].

The target flight computer selected for this study has a dual configuration with each side having 128 Kb of memory. The compiler selected for the target flight computer conforms to ANSI FORTRAN 66 standards and also has a significant number of extensions allowing real-time use. This target flight computer has approximately a 255,000 Whetstones [4] floating point performance in 32 bit single precision. At the start of this study, the size of the FINDS program, implemented in FORTRAN 77 using double precision, was 340 Kb. On the host development computer, having approximately 300,000 Whetstones floating point performance in 32 bit single precision [5], the FINDS algorithm ran 30 times slower than real-time with detectors off, and 120 times slower than real-time with detectors on.

Hence, the FINDS program had to be scaled down in size to be ported onto the government flight computer. In addition, the algorithm execution speed had to be significantly increased to allow real-time operation. In this study, the following was accomplished:

Reduction In Program Size:

Starting from a program size of 340 Kb in double precision FORTRAN 77, an equivalent single precision FORTRAN 66 program of size 116 Kb was developed, and ported onto the target flight computer. This was accomplished by:

(a) eliminating the interactive input/output routines in the code,

(b) converting the program from a double precision implementation to single precision,

(c) reducing program variable dimensions to handle dual sensor redundancy instead of triple redundancy,

(d) incorporating all sensor failure simulation routines into an external preprocessor program, and

(e) deleting the radar altimeter from the sensor suite.

Increase In Execution Speed:

Starting from a FINDS program with an execution speed of 30/120 times slower than real-time (with detectors off/on) on the host development computer, we have obtained an equivalent program with an execution speed of 1.4 times slower than real-time on the same machine. This was accomplished by:

(a) converting the code into single precision,

(b) replacing general purpose matrix computations with specialized routines,

(c) using a constant state transition matrix in the system model instead of a time-varying one,

(d) implementing piece-wise constant gains in the no-fail filter, and

(e) replacing the FINDS multiple hypothesis test allowing simultaneous detection and isolation with a sequential detection and isolation test.

Improved Estimation Performance:

Through the course of this study, we have changed the sensor noise design parameters used in the no-fail filter to reflect a more cohesive use of the entire sensor complement in the estimation process. In addition, we have modified the steady state wind model used in the no-fail filter design. These changes resulted in a better bias and aircraft state estimation error performance, thus resulting in a better behaved (less time correlated and closer to zero-mean) no-fail filter residuals sequence.

Improved Detection Performance:

In this study, we have also significantly improved the FINDS algorithm detection performance reported in [1]. The FDI algorithm can now detect sensor failures, injected into the flight data, considerably faster and without any false alarms. False alarm performance improvement is due to the better estimation performance. Based on a sensor failure detectability analysis, detection speed was improved by replacing the multiple hypothesis test (over a fixed window of expanded no-fail filter residuals) implemented in FINDS with a set of mean detection tests over various moving windows of the averaged no-fail filter residuals. Low level MLS, IMU and IAS sensor failures are detected instantaneously with the new detection strategy, while accelerometer and rate gyro failures are detected within the minimum time allowed by the information generated in the sensor residuals based on the aircraft point mass equations of motion.

The discussion of the above modifications is organized in this report as follows:

Chapter 2 presents the state and bias estimation performance results of the no-fail filter along with an overview of the flight data driven emulation. The detectability analysis, FINDS false alarm and failure detection/isolation performance, and initial results of the new detection strategy are presented in Chapter 3. Chapter 4 contains a series of proposed experiments for the modified FINDS algorithm on government flight computers. Concluding remarks and further recommendations end this chapter.

2. ESTIMATION PERFORMANCE

In this chapter, we discuss the various changes made to the no-fail filter in the FINDS algorithm in order to improve estimation error performance. In the first section, we present a brief overview of the flight data driven emulation, and the modifications made to the flight data interface with respect to the sensor failure injection modules. This is followed by a discussion of the sensor noise statistics utilized by the no-fail filter. In the next section, estimation error performance is analyzed by examining the aircraft state estimates and body mounted sensor bias estimates. This section ends with a discussion of the statistics of the no-fail filter residual sequence which forms the inputs to the detection and isolation algorithm analyzed in the next chapter. In the last section, we discuss the effects of using piecewise constant no-fail filter gains on the estimation error and execution speed.

2.1 Emulation Review

As discussed in [1], the flight recorded sensor data contained a usable third channel only for the rate gyros. Hence, a dual redundant sensor complement has been used throughout this study. A second channel has been simulated for the MLS measurements since only a single channel of azimuth, elevation and range measurements was available from the flight data. As mentioned in the introduction chapter, the radar altimeter measurements have also been deleted from the sensor complement. Thus the current version of the emulation goes to 266-267 seconds when the aircraft reaches radar altimeter height, at which point the FINDS program execution is halted.

A sensor difference signal error analysis, similar to the one performed in [1], now gives us a new set of empirical statistics for the sensor flight data up to 266 seconds. These empirical statistics which determine the sensor noise parameters are given in Table 2.1. The mean values of these difference signals clearly indicate the presence of sensor biases, especially for the longitudinal accelerometer and rate gyros. It should also be noted that the whiteness test performed on these sensor flight data channel differences does not show any major deviations from the results shown in Table 2.2 in [1].

Table 2.1: Empirical statistics for sensor flight data channel differences

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
Acc.-Long.	-1.99E-01	+4.37E-02	-3.81E-02	-3.85E-01	m/s/s
Acc.-Lat.	+7.92E-02	+5.01E-02	+2.73E-01	-1.15E-01	m/s/s
Acc.-Vert.	+6.51E-02	+1.17E-01	+5.12E-01	-3.57E-01	m/s/s
Gyro-Roll	-2.66E-01	+4.47E-02	-1.03E-01	-4.44E-01	deg/s
Gyro-Pitch	-2.12E-01	+4.66E-02	-3.44E-02	-3.90E-01	deg/s
Gyro-Yaw	+1.07E-01	+3.11E-02	+1.82E-01	-6.09E-03	deg/s
IAS	+8.96E-01	+2.28E-01	+2.05E+00	-3.03E-01	m/s
IMU-Roll	+6.88E-02	+1.06E-01	+3.37E-01	-2.37E-01	deg
IMU-Pitch	-1.32E-01	+2.27E-01	+3.01E-01	-5.98E-01	deg
IMU-Yaw	-4.54E-03	+8.54E-02	+6.57E-01	-7.11E-01	deg

Finally, the sensor failure injection modules in the emulation have now been incorporated into an external flight data preprocessor routine to simulate failed sensor data. In addition to reducing the program size by about 30 Kb, this change also preserves the input functionality of the failure injection routines. Thus, the FINDS algorithm can still be tested with the same simulated failures as in [1].

2.2 Improved Estimation Performance

In this study, the no-fail filter performance has been evaluated by the behavior of the no-fail residual sequence; an approximately zero mean and white sequence under no failures implies satisfactory state and bias estimates. However, as explained in section 3.2 of [1], the 'best' estimation performance does not necessarily result in the 'best' failure detection performance since the filter makes less use of noisier sensors, thus reducing the associated failure signatures on the residuals. Hence, the design values for the no-fail filter noise parameters have been chosen so as to not only reflect the empirical statistics presented in Table 2.1, but also to ensure that the filter makes adequate use of all sensors in generating the aircraft state estimates. Table 2.2 shows these design values for the no-fail filter process noise (associated with input sensors and wind dynamics) and measurement sensor noise variances. All of the results presented in this report have been obtained by using these parameters.

Table 2.2: Design Values for no-fail filter noise parameters

VARIABLE	NOISE S.D.	REPLICATIONS	UNITS
PROCESS NOISES:			
Acc.-Long.	5.00E-02	1	m/s/s
Acc.-Lat.	5.00E-02	1	m/s/s
Acc.-Vert.	5.00E-02	1	m/s/s
Gyro-Roll	5.00E-02	1	deg/s
Gyro-Pitch	5.00E-02	1	deg/s
Gyro-Yaw	5.00E-02	1	deg/s
X-Wind-rw	1.00E-01	N/A	m/s
Y-Wind-rw	1.00E-01	N/A	m/s
MEASUREMENT NOISES:			
MLS-Azim.	6.00E-02	1	deg
MLS-Elev.	6.00E-02	1	deg
MLS-Range	6.00E+00	1	m
IAS	3.00E+00	2	m/s
IMU-Roll	2.50E-01	2	deg
IMU-Pitch	5.00E-01	2	deg
IMU-Yaw	3.00E-01	2	deg

Another change made in the filter design has been the introduction of a new steady-state wind model. In [1], the horizontal wind model assumed zero process noise, and a time constant of 1000 seconds. These wind estimates showed a marked dependence on aircraft maneuvers. In this study, we have introduced a process noise of 0.1 m/s on both the 'x' and 'y' direction winds in the runway frame, and the time constant has been reduced to 100 seconds. These values have been chosen so as to model a slowly time-varying wind component (with an RMS value of approximately 1.0 m/s) in addition to the steady-state winds.

Another modification is that the no-fail filter now uses a constant state transition matrix as opposed to a time-varying, state dependent one [6],[8] which had to be updated by the partials of the input transition matrix at every iteration. This change has reduced the execution time by almost 20% in addition to bringing the program size down by 5 Kb, without making any major impact on either the state or bias estimation performance. Moreover, this change has resulted in the decoupling of the no-fail filter's translational dynamics part (aircraft position, velocity and horizontal winds) from the rotational kinematics part (aircraft attitudes).

The no-fail filter now uses only one replication of the MLS sensor measurements. Since the second channel for these measurements had to be simulated in order to have a complete dual redundant sensor suite, this second channel is now kept in stand-by status to be used only in the event of a MLS failure (similar to the set-up for input sensors). Based on the earlier sensor error analysis of [1], the MLS sensor noise is assumed to be white.

We now present the aircraft state estimate time-histories for the nominal emulation run with the no-fail filter design based on the parameters given in Table 2.2. Beginning at time zero, this run ends at 266.2 seconds when radar altimeter height is attained. Figure 2.1 shows the aircraft ground track, and altitude profile as the aircraft goes through various flight segments from runway approach to altitude hold, and runway alignment to final descent. Figure 2.2 depicts the horizontal velocity estimates, with the y-runway direction velocity highlighting the runway alignment maneuver. The aircraft vertical velocity profile and roll attitude estimate time history

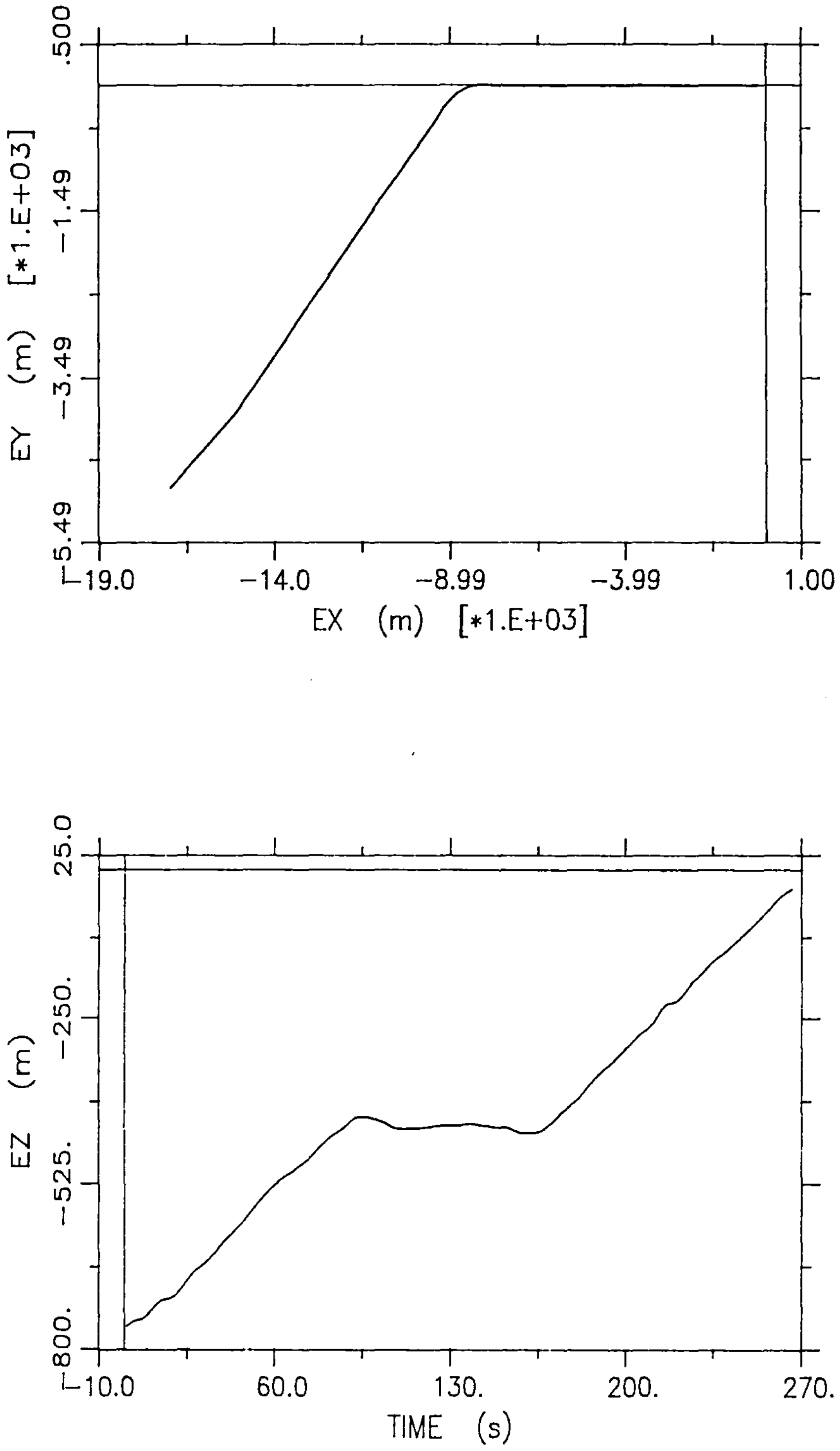


Figure 2.1: Estimated aircraft ground track and altitude profile

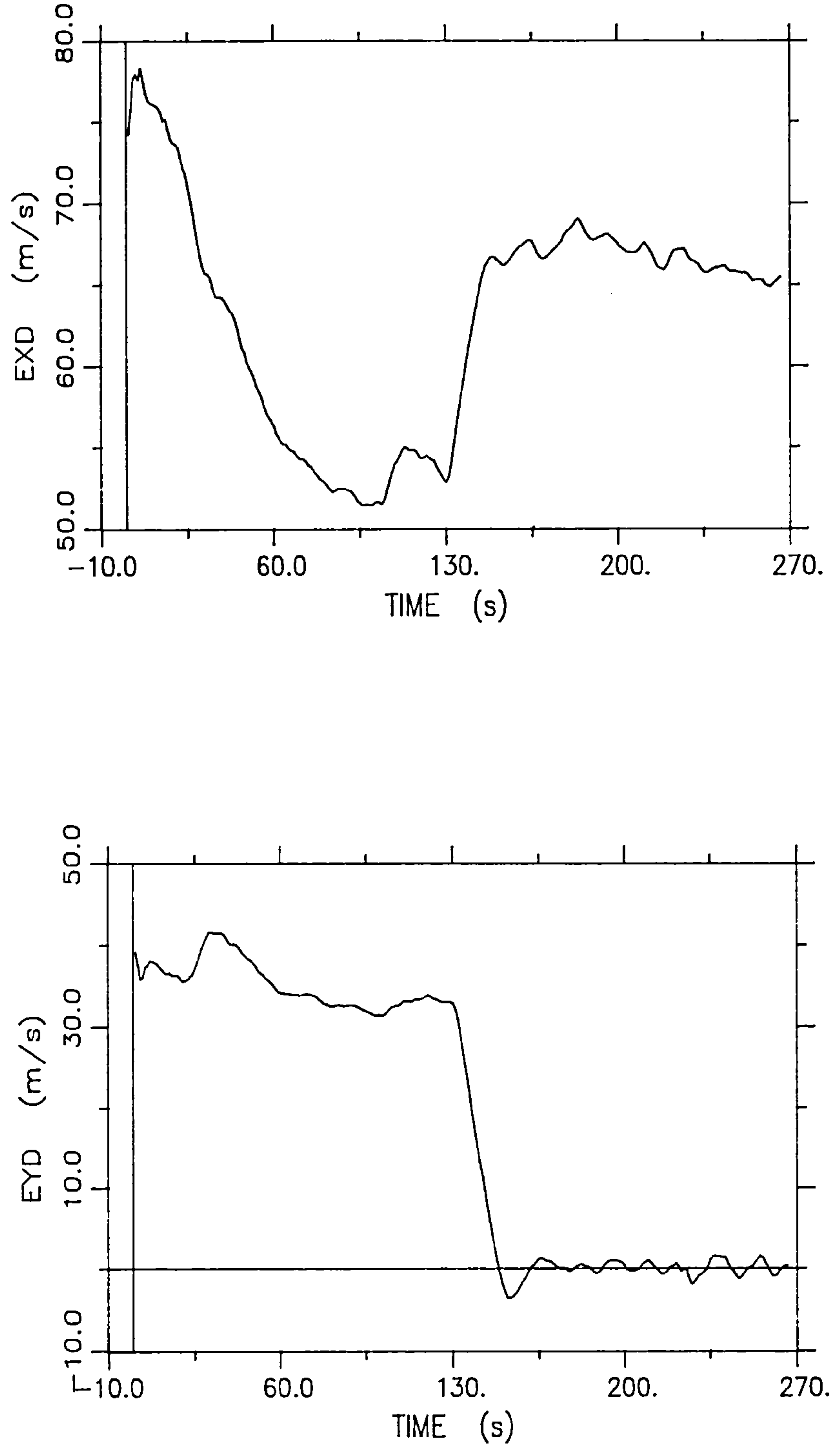
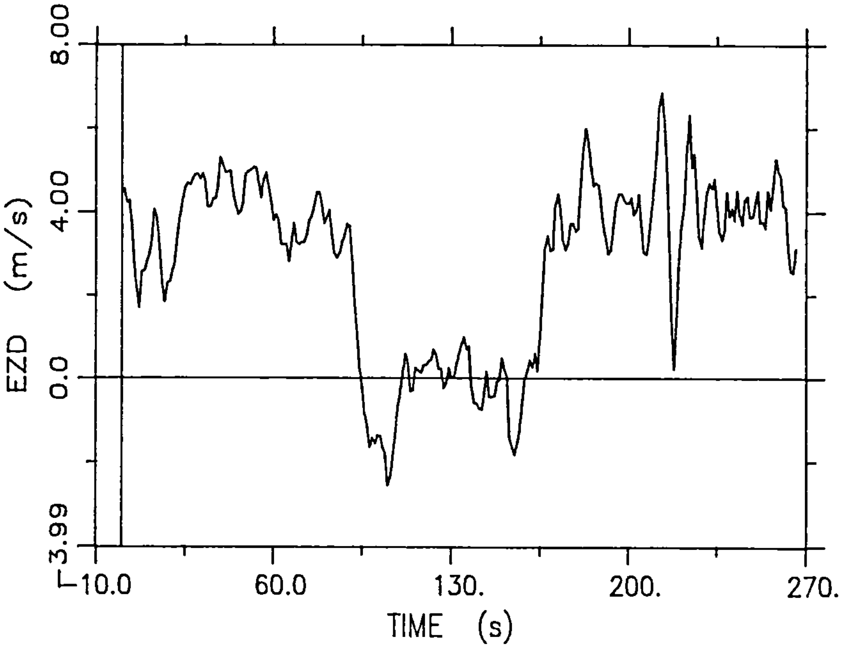


Figure 2.2: Horizontal velocity estimate time histories



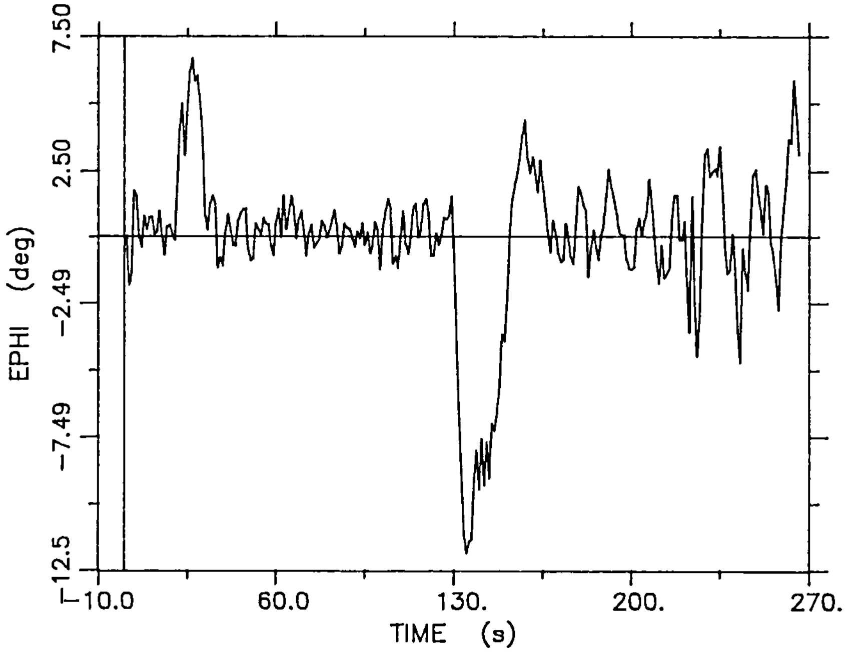
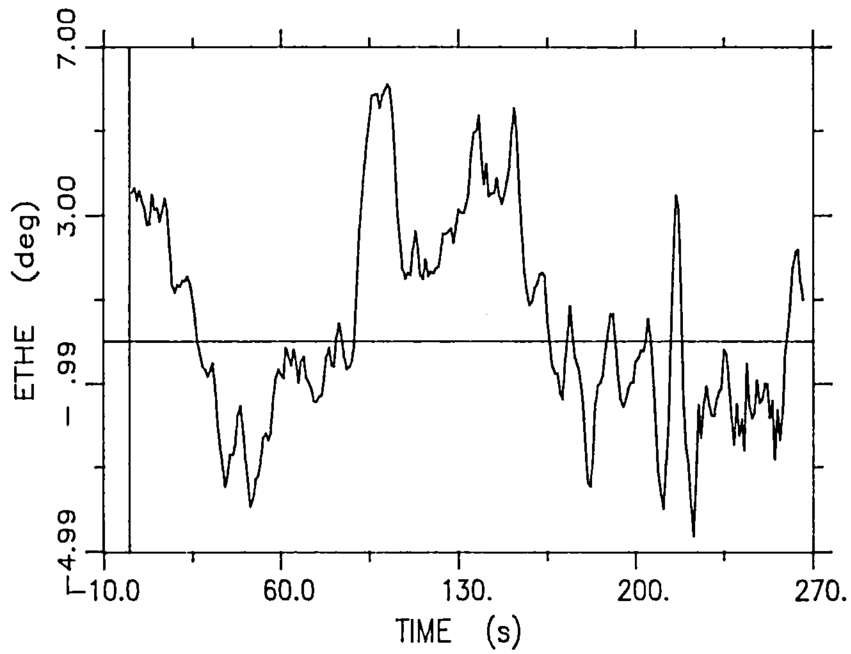


Figure 2.3: Vertical velocity profile and roll attitude estimate



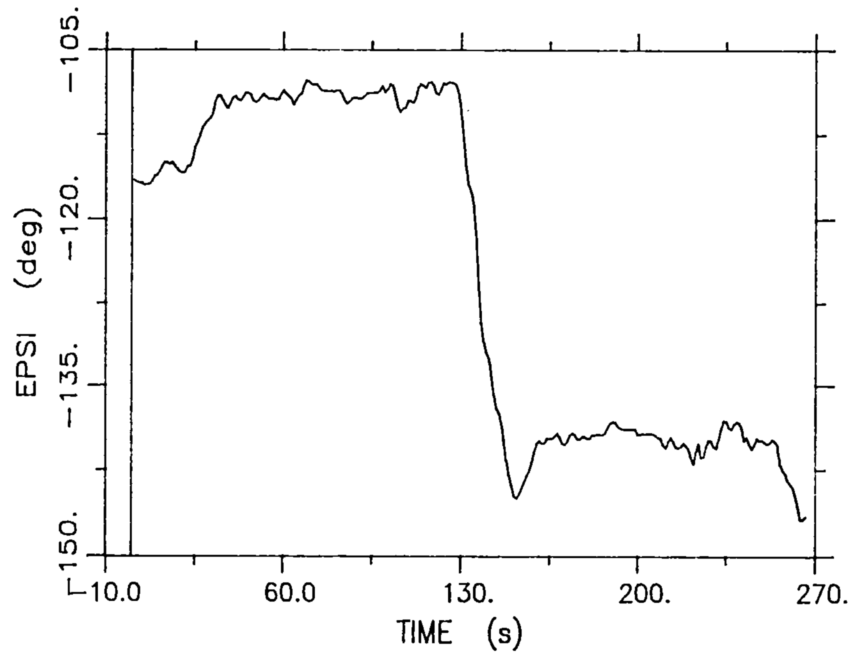
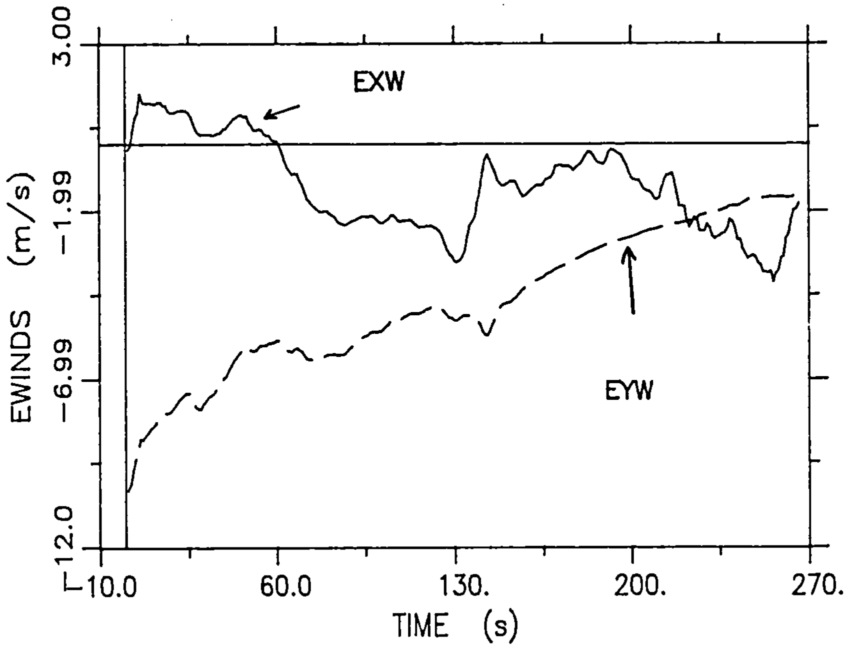


Figure 2.4: Pitch and yaw attitude estimates

are shown in Figure 2.3. These figures show that the altitude is held constant while performing the bank maneuver for runway alignment. Figure 2.4 shows a pitch down attitude of magnitude 2-4 degrees through most of the emulation run except for the alignment maneuver at which time the aircraft pitches up at approximately the same angle. Figure 2.4 also shows the yaw attitude estimate time history. Finally, the new horizontal wind estimates and aircraft latitude versus longitude track are shown in Figure 2.5. The wind estimates portray a gradually diminishing crosswind starting at 10 m/s and reducing to 2 m/s at the end of the run. The results also imply the presence of more wind gusts at lower altitudes as evidenced by the higher noise residuals in the IAS sensor during the latter segments of the flight.

The major impact of the new wind model is the quicker convergence of the normal operating biases for the body mounted accelerometers as depicted in Figures 2.6 and 2.7. As opposed to the approximately 60-70 seconds needed by the no-fail filter to reach the steady-state bias estimate, these bias estimates now converge to steady-state values in about 40 seconds. Comparisons to earlier results (Figures 3.8-3.9 in [1]) also show a more stable steady-state behavior implying smaller covariance statistics for the bias filter. Figure 2.7 also shows the bias estimate for the roll rate gyro and Figure 2.8 gives the bias estimate time histories for the pitch and yaw gyros. These bias estimates do not show any major change from earlier results.



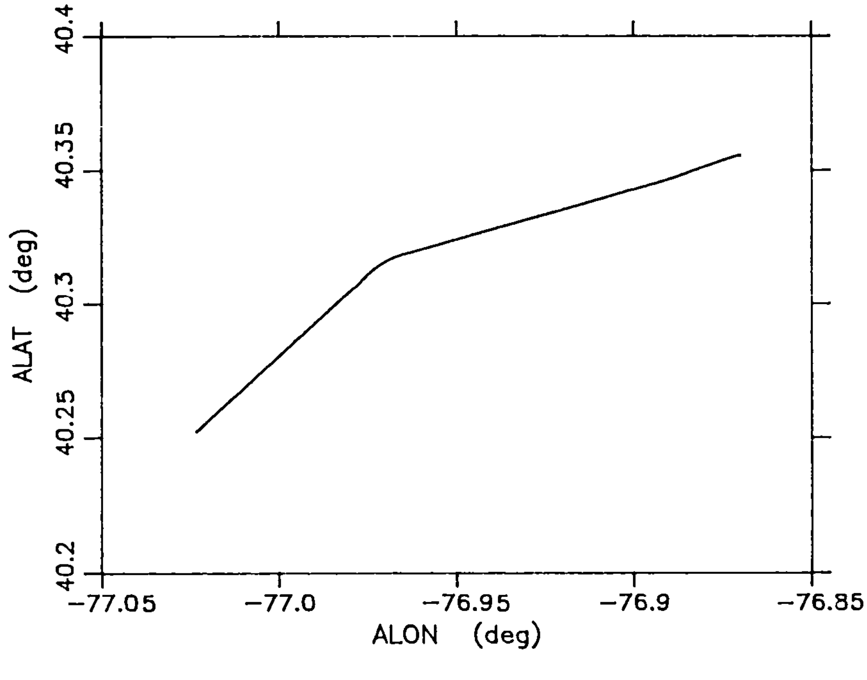
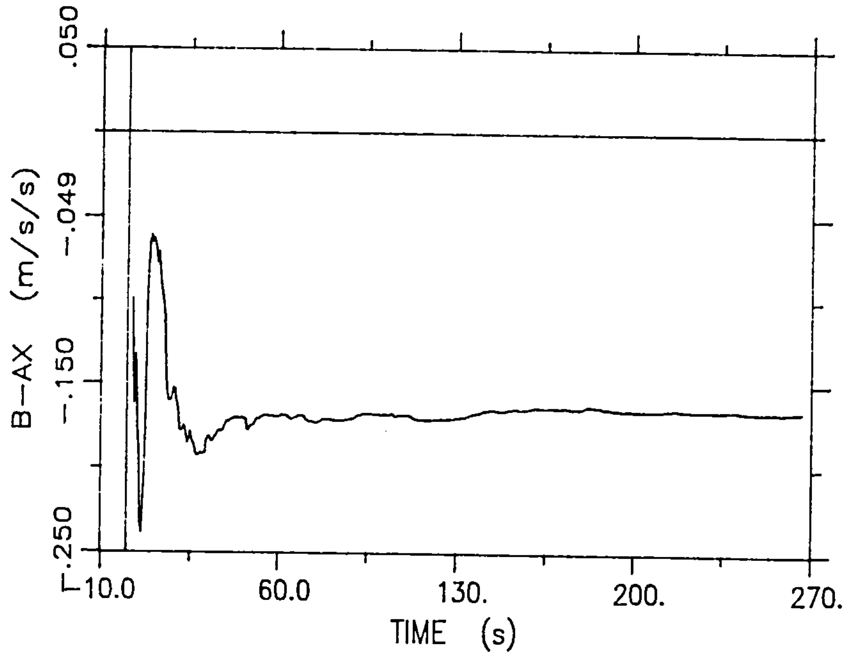


Figure 2.5: Horizontal wind estimates and aircraft latitude vs. longitude track



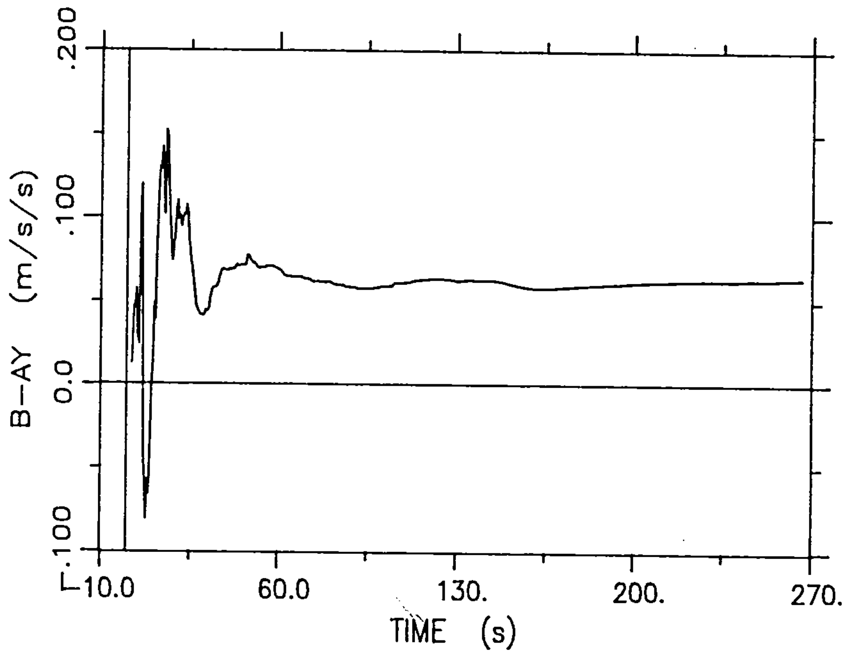


Figure 2.6: Longitudinal and lateral accelerometer bias estimates

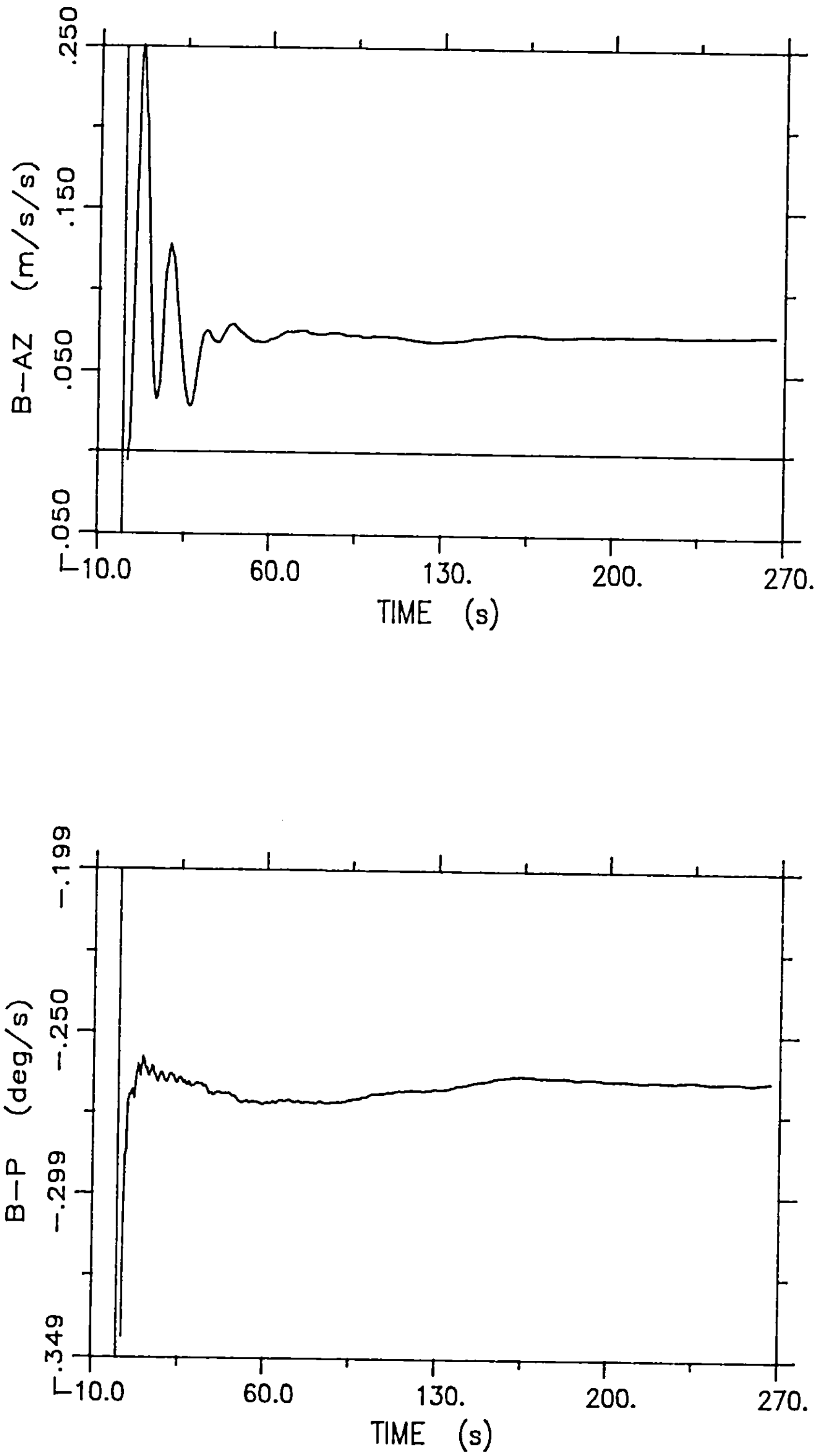


Figure 2.7: Vertical accelerometer and roll rate gyro bias estimates

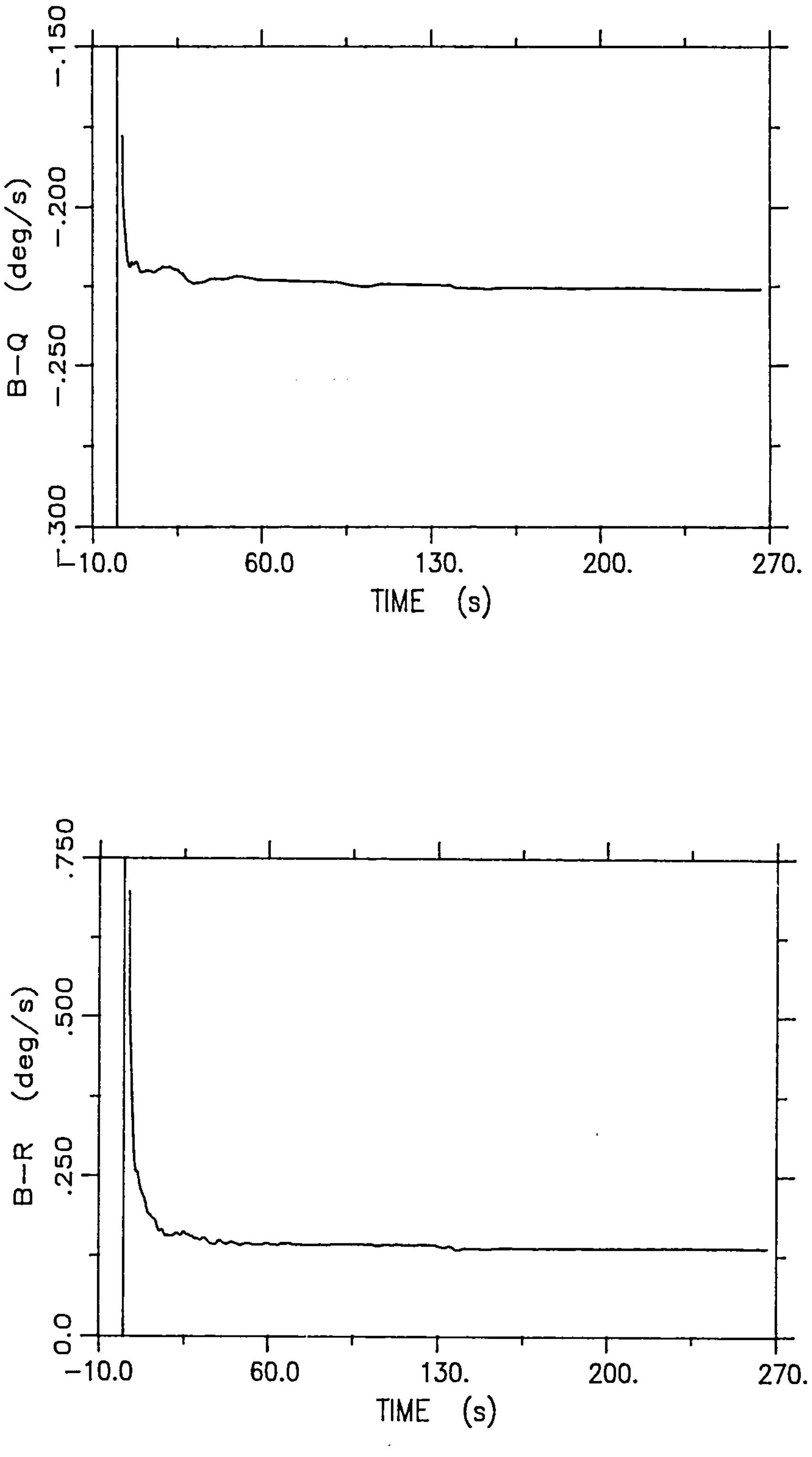


Figure 2.8: Pitch and yaw rate gyro bias estimates

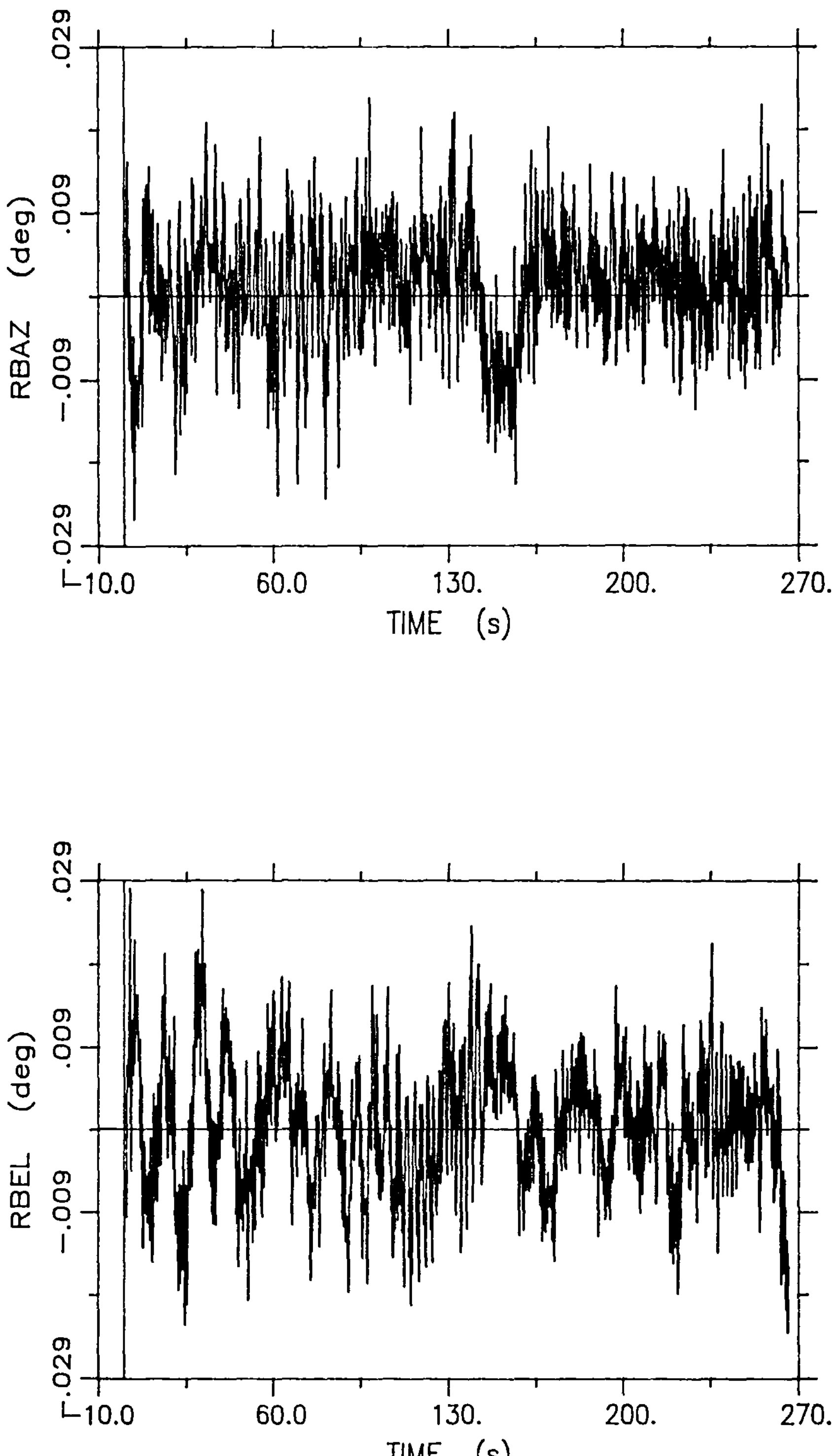


Figure 2.9: No-fail filter residuals for MLS azimuth and elevation

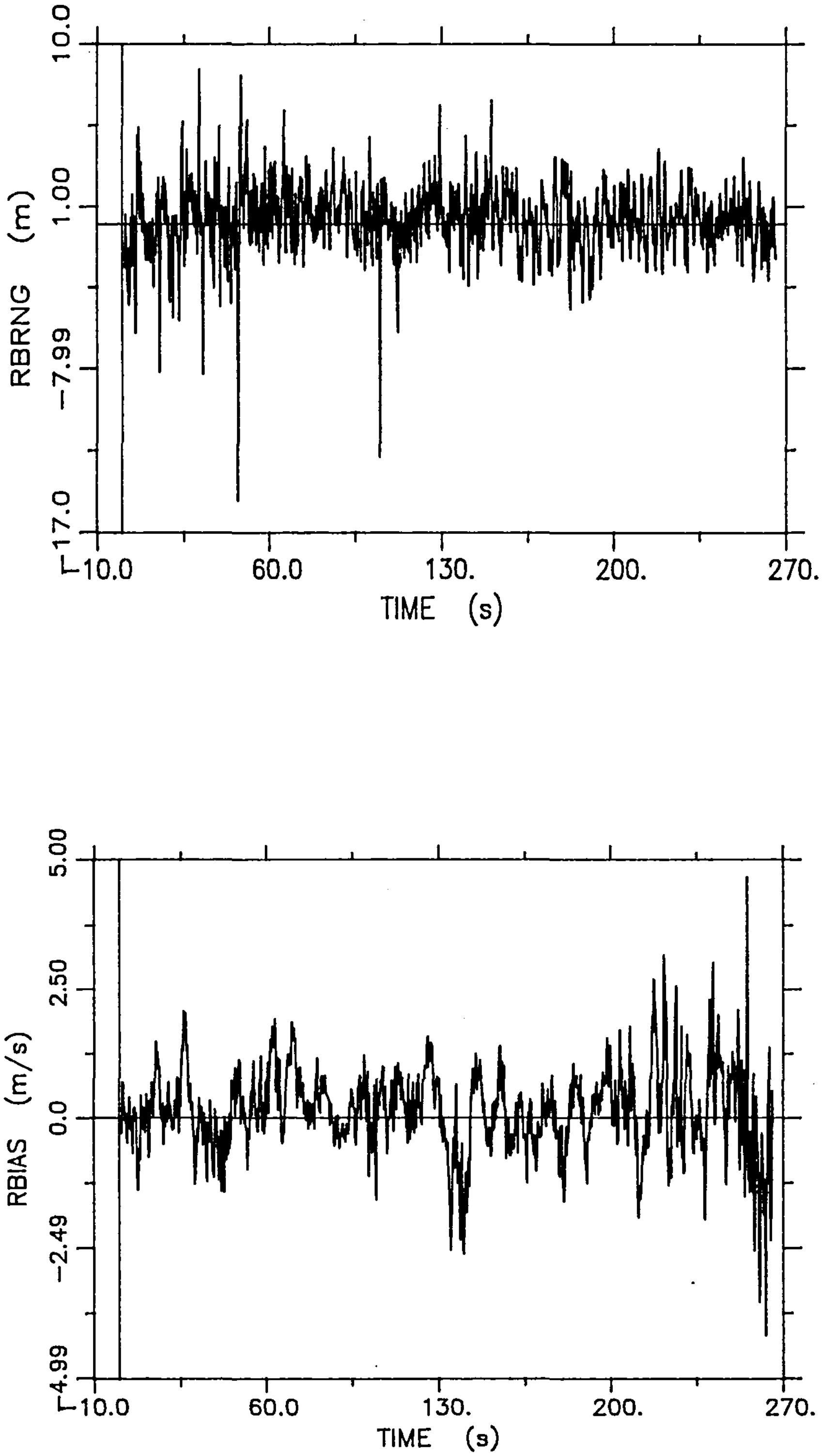
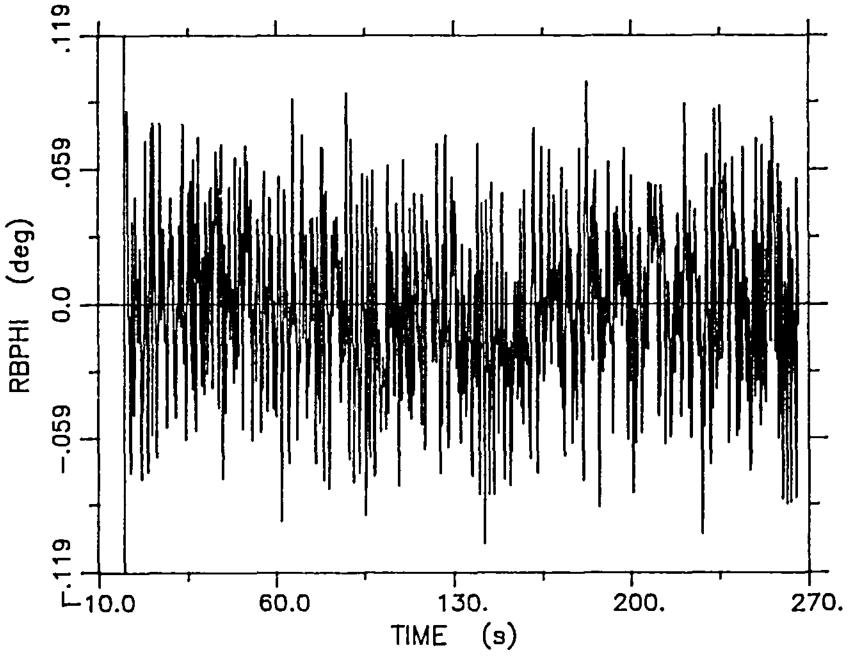


Figure 2.10: No-fail filter residuals for MLS range and IAS



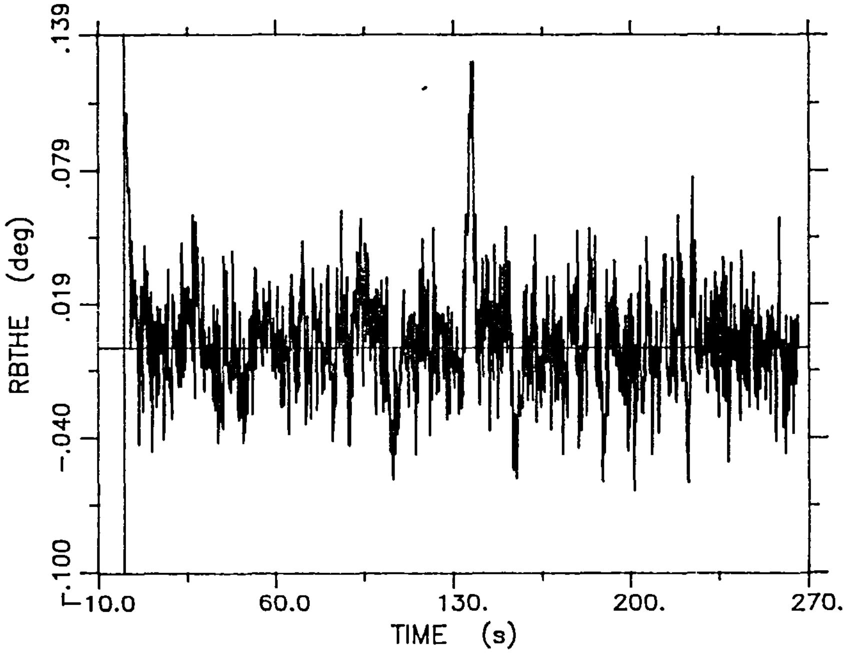


Figure 2.11: No-fail filter residuals for IMU roll and pitch

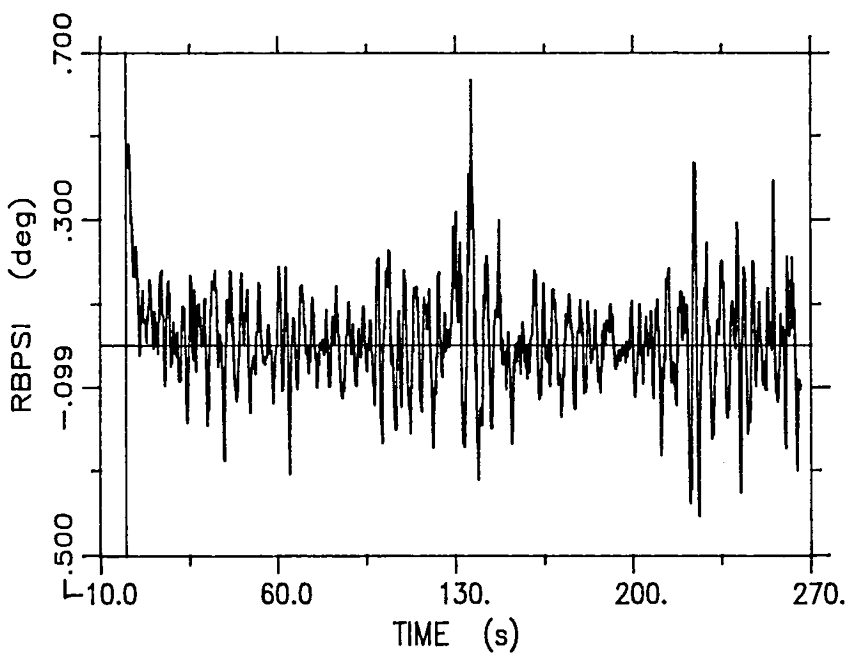


Figure 2.12: No-fail filter residuals for IMU yaw

Note that the steady state biases of approximately -0.16 m/s/s, 0.075 m/s/s, 0.07 m/s/s, -0.27 deg/s, -0.23 deg/s, and 0.14 deg/s for the longitudinal, lateral, vertical accelerometers, and roll, pitch, yaw rate gyros respectively compare very favorably with the empirical statistics of Table 2.1 for the same sensors.

The overall improved performance of the no-fail filter can truly be gauged by the residual time histories which are shown in Figures 2.9-2.12. With reference to earlier residual plots (Figures 3.11-3.14 in [1]), each of these residual sequences shows a markedly smaller mean and uncorrelated behavior. The residuals for MLS azimuth, elevation and IMU pitch continue to show some correlation between aircraft maneuvers and no-fail filter estimation errors. We also note that the few unusually large residuals in the case of the MLS range sensor (at approximately 47 seconds and 105 seconds) are caused by unusually high range measurement errors at these time instants.

Table 2.3 presents the computed empirical statistics for these no-fail filter residual sequences, calculated over the entire flight data run. These low sample means and standard deviations verify the extremely good estimation performance of the no-fail filter, particularly when we compare these statistics with the actual sensor noise parameters of Table 2.1. For example, in the case of the IMU pitch attitude sensor, the rms value of the difference signal is 0.263 deg/s whereas the rms value of of the measurement residual is only 0.025 deg/s. Similarly, the rms error decreased from 0.126 deg/s to 0.037 deg/s for the roll attitude sensor and from 0.93 m/s to 0.85 m/s for the IAS sensor.

Table 2.3: No-fail filter residuals statistics : nominal update frequency of 20 Hz

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
MLS-Azim.	+1.37E-03	+7.35E-03	+2.87E-02	-3.06E-02	deg
MLS-Elev.	+4.22E-04	+8.26E-03	+3.28E-02	-2.45E-02	deg
MLS-Range	+1.67E-01	+2.03E+00	+1.08E+01	-1.98E+01	m
IAS	+1.44E-01	+8.36E-01	+4.66E+00	-4.19E+00	m/s
IMU-Roll	-1.73E-03	+3.70E-02	+1.29E-01	-1.32E-01	deg
IMU-Pitch	+3.19E-03	+2.50E-02	+1.40E-01	-7.29E-02	deg
IMU-Yaw	+1.05E-02	+1.16E-01	+6.36E-01	-4.52E-01	deg

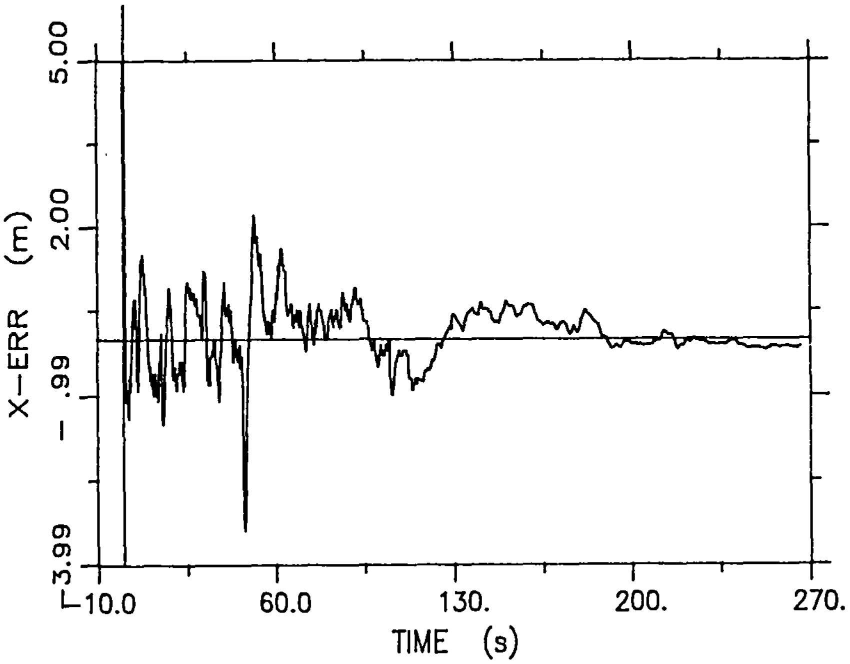
2.3 Estimation Performance With Piecewise Constant Gains

As discussed in the introductory chapter, the major thrust of the current study is to increase the execution speed of the FINDS algorithm for an actual flight test exercise. To this end, one of the first changes made to the algorithm was to convert it from a double precision implementation to a single precision one; this conversion not only increased the execution speed by a factor of two (on the host development computer) but also decreased the program size by about 100 Kb (from a starting size of 340 Kb). Another modification was the use of a time invariant state transition matrix, as discussed earlier. Also, specialized matrix routines were substituted for general purpose matrix computations to take advantage of some of the inherent system matrix properties. For example, a special positive definite symmetric matrix inverse routine was used in place of a generalized inverse one. These modifications reduced the execution time from 30 times slower than real-time (for the double precision, no detectors case) to about 10 times slower than real-time. In order to increase the execution speed further, we have investigated the suitability of using piecewise constant gains in the no-fail filter. We present the results of this analysis in this section.

A study of the no-fail filter gain time histories reveals that these gains along with their associated covariance matrices have a slowly time-varying behavior, except in the initial phase of the emulation run. Hence, we have investigated the estimation performance by updating these gains only at certain multiples of the sampling period instead of updating them at every sampling instant. This change has a favorable impact on execution speed because it eliminates two seventh order matrix inversions, system observation matrix updates and covariance matrix updates in the intermediate sampling instants.

As a first step, the entire emulation run with detectors off was repeated (similar to the previous section run) -- the only difference being that the no-fail filter gain and covariance calculations were performed at every fifth sampling instant, and held constant in between. As expected, this modification cut down the execution time considerably to about 3 times slower than real-time. An analysis of the aircraft state estimate time histories shows an increased initial transient period, but very little difference is observed in the same state estimates in the latter part of the flight. As a representative example, Figure 2.13 shows the difference between the aircraft x-runway position estimates for the nominal run of the previous section and this new run with no-fail filter gain update frequency of 4 Hz. It is clearly seen that the maximum estimation error is during initial filter transients, and gradually decays to almost zero. This figure also shows the same estimation error time history for the IMU pitch attitude. Here, we see that the estimation difference rapidly diminishes as compared to the position estimate error.

This particular behavior can be attributed to the bias estimation performance with constant gains and the relatively longer time it takes for the accelerometer biases to reach steady state than for the rate gyro biases. Figures 2.14-2.16 show the



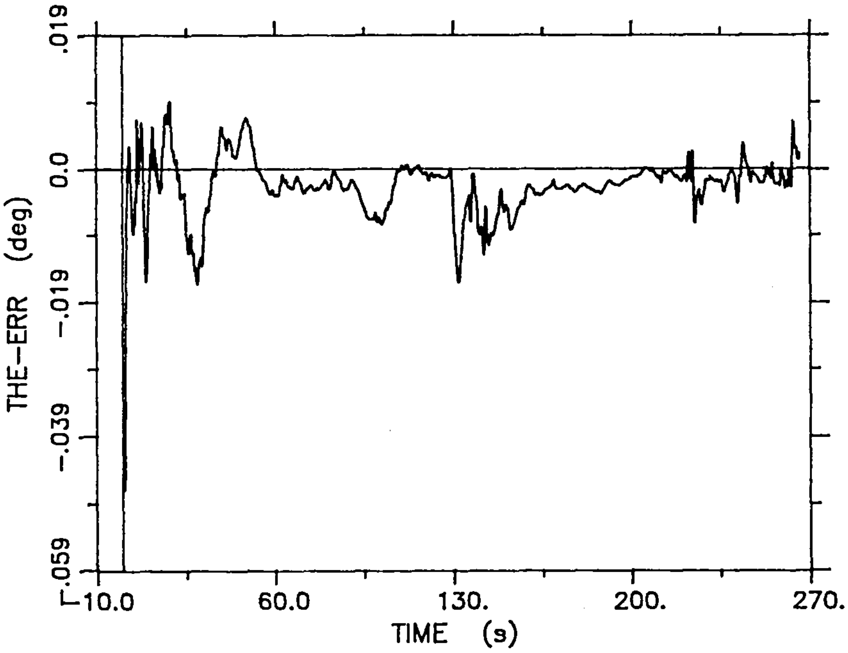
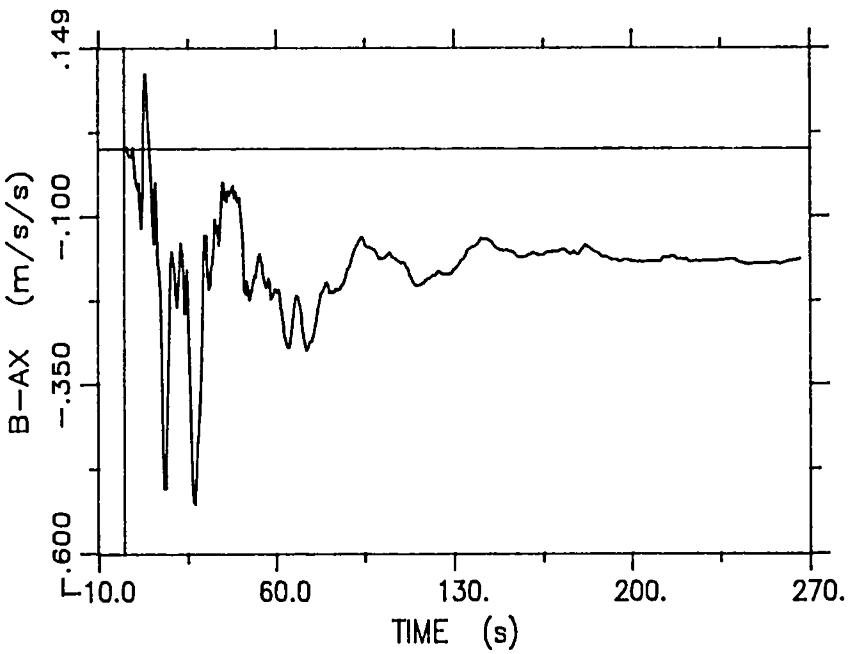


Figure 2.13: Longitudinal position and pitch attitude estimation error with 4 Hz gain update frequency



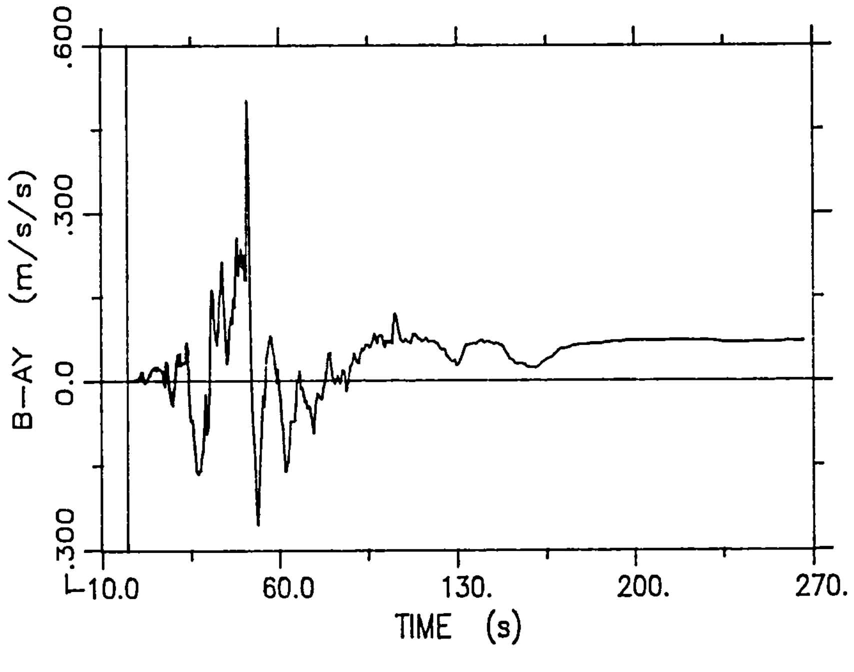


Figure 2.14: Longitudinal and lateral accelerometer bias estimates with 4 Hz gain update frequency

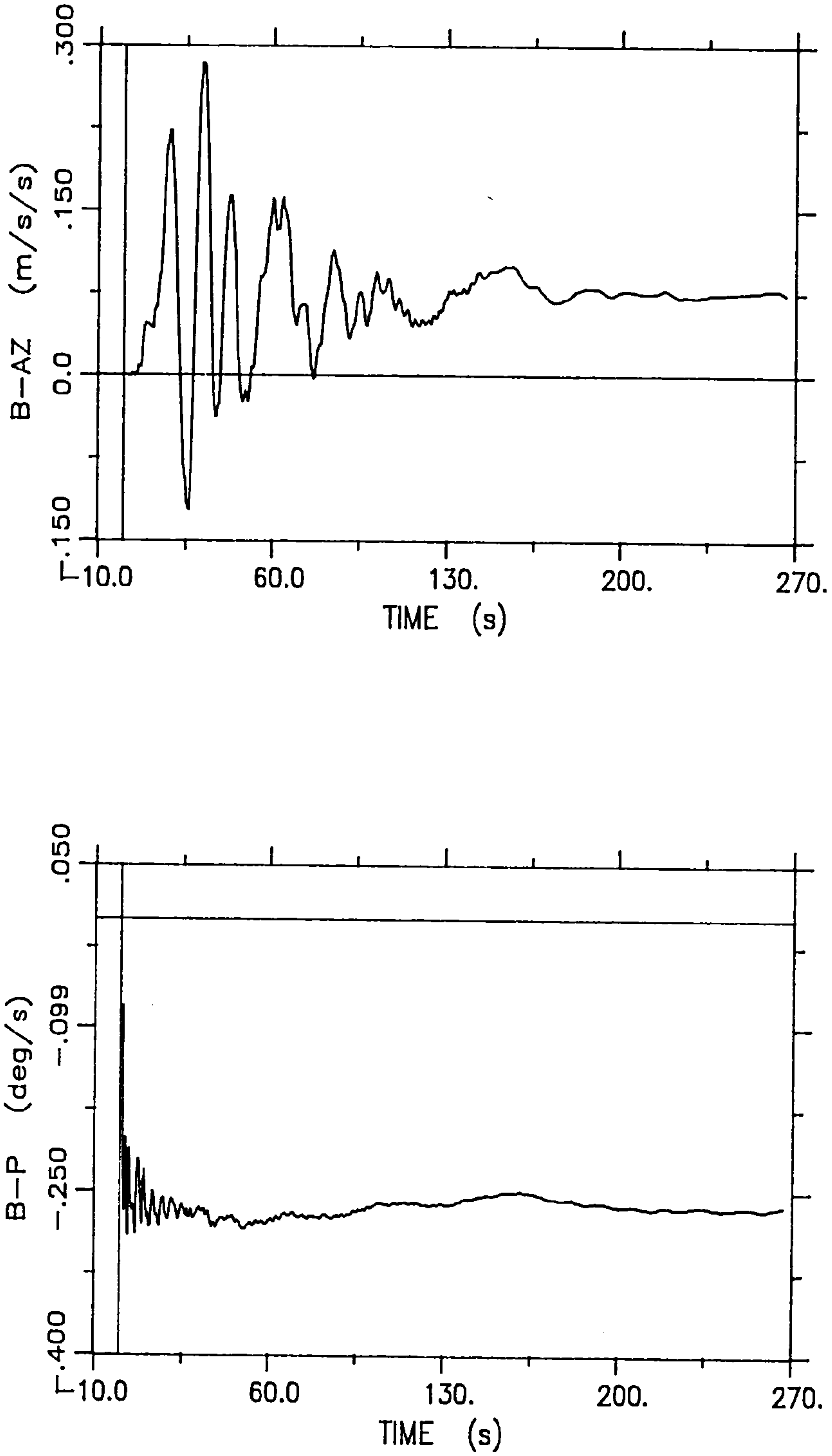
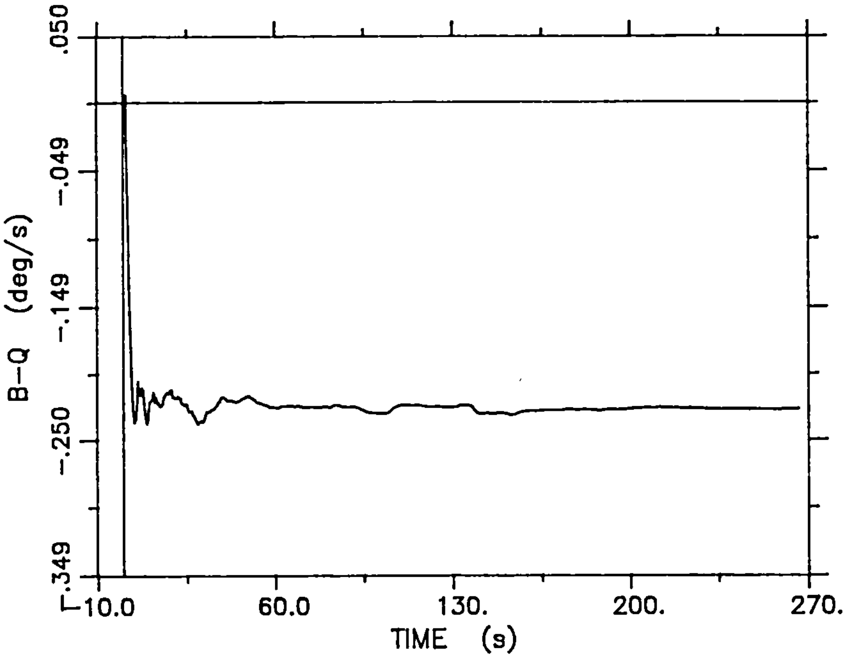


Figure 2.15: Vertical accelerometer and roll rate gyro bias estimates with 4 Hz gain update frequency



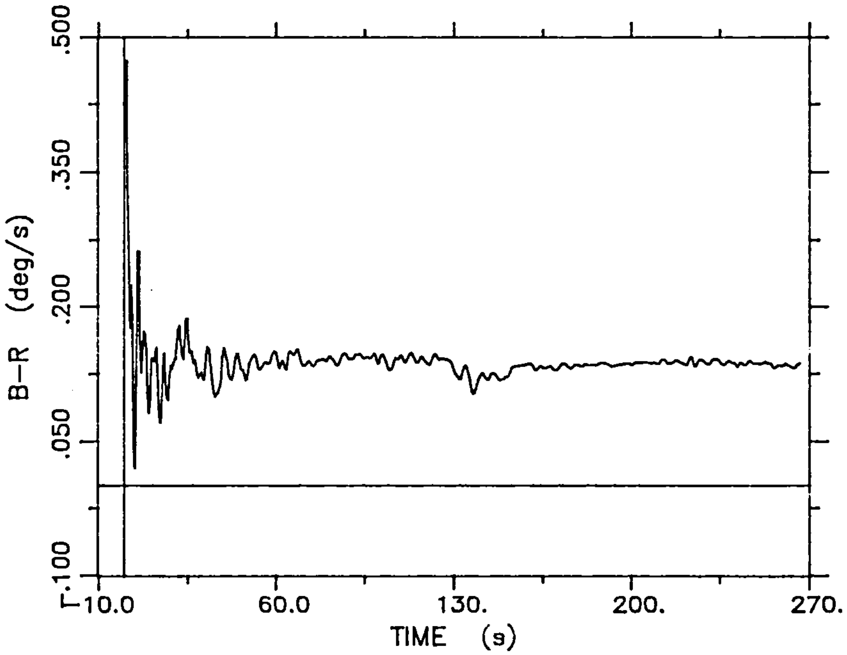
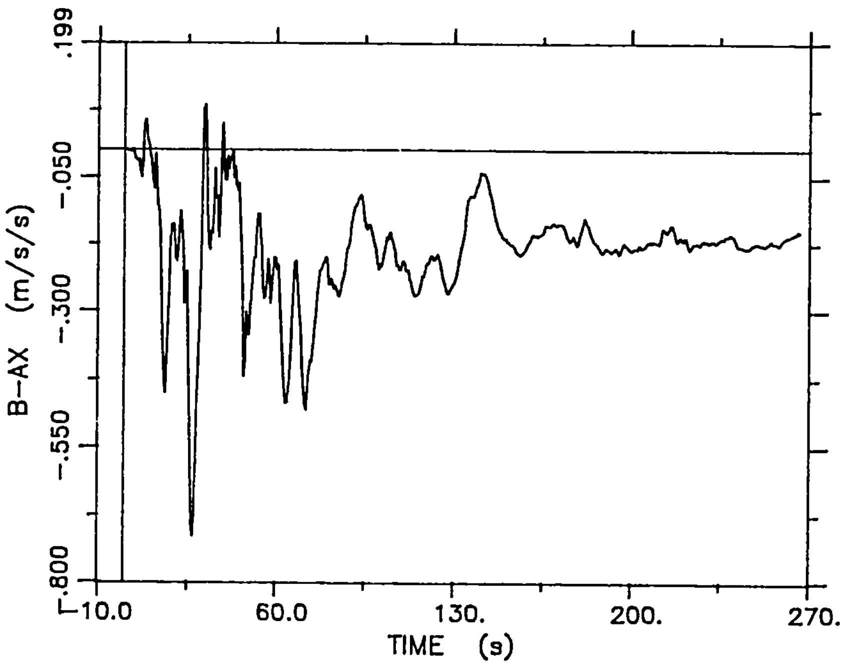


Figure 2.16: Pitch and yaw rate gyro bias estimates with 4 Hz gain update frequency

bias estimate time histories for the longitudinal, lateral, vertical accelerometers, and roll, pitch, yaw rate gyros for this particular run corresponding to the gain update frequency of 4 Hz. Note that in this case, the accelerometer biases take almost 100 seconds to converge as opposed to 40 seconds for the nominal run. Similarly, the rate gyro biases reach their steady-state values in about 20-25 seconds instead of 10 seconds. Note also, that the steady-state bias estimates at 4 Hz are the same as those at 20 Hz.

Plots of the no-fail filter residual sequence errors with respect to the nominal run also show the same trends as the state estimate errors. Table 2.4 presents the computed statistics for this new set of measurement residuals. The no-fail filter estimation performance at 4 Hz compares favorably with the performance at the nominal 20 Hz as demonstrated by the relative closeness of the residual statistics in Tables 2.3 and 2.4.

Similar runs were made by keeping the gains and covariances constant over 10, 15 and 20 time iterations. With respect to execution speed, these runs were 2, 1.5 and 1.3 times slower than real-time on the host development computer. The same estimation error/difference analysis performed in each of these cases shows similar trends as the 4 Hz. gain update frequency case, with the only difference being a longer convergence time for the gains and hence for the state estimates. These convergence rates can again be traced back to the bias estimation performance. Figures 2.17-2.19 show the bias estimate time histories for the case when the gain update frequency is 2 Hz.



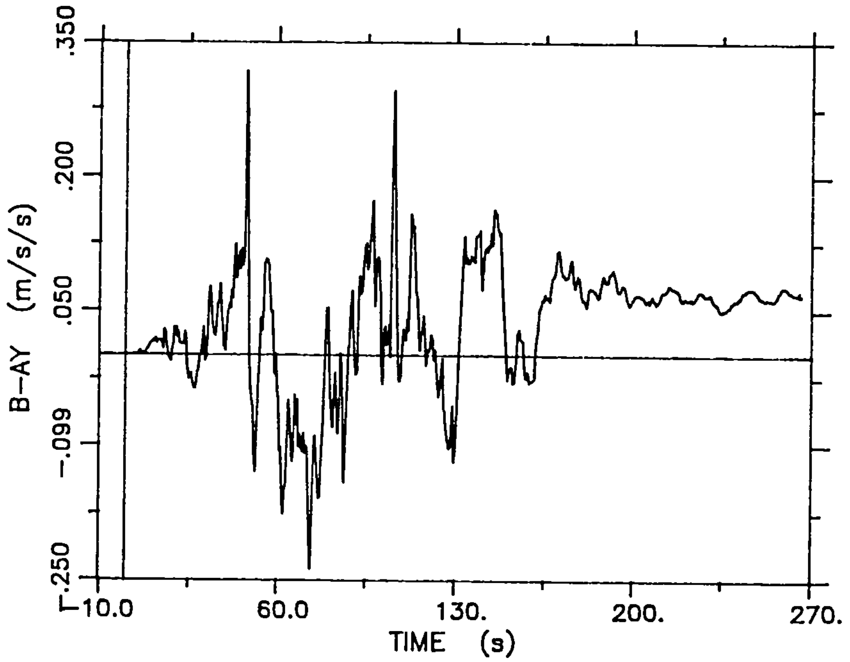
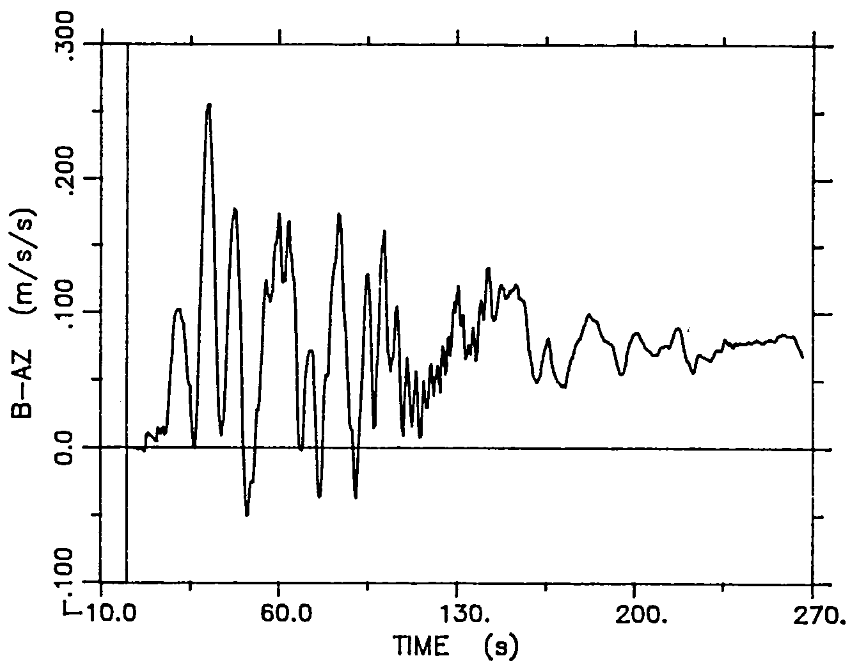


Figure 2.17: Longitudinal and lateral accelerometer bias estimates with 2 Hz gain update frequency



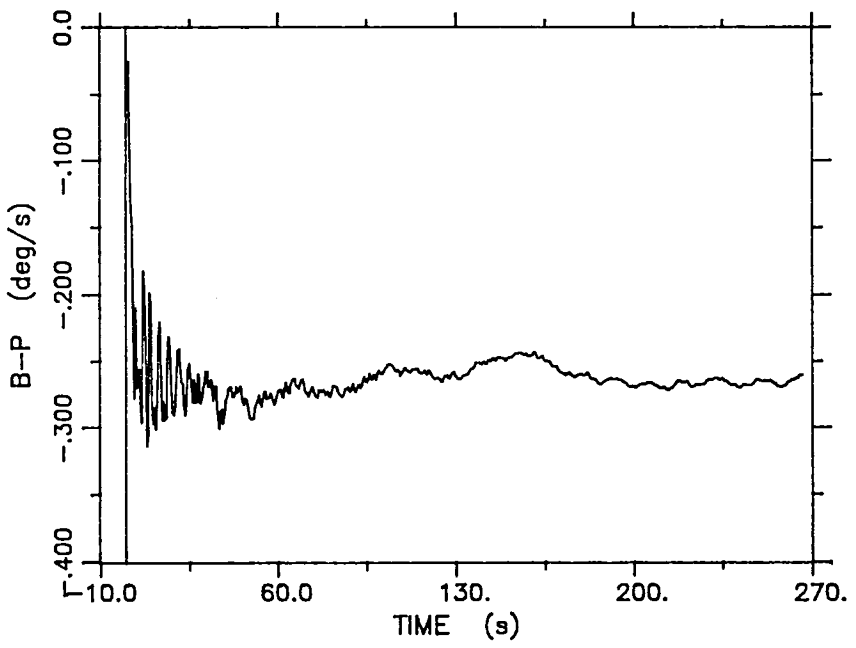


Figure 2.18: Vertical accelerometer and roll rate gyro bias estimates with 2 Hz gain update frequency

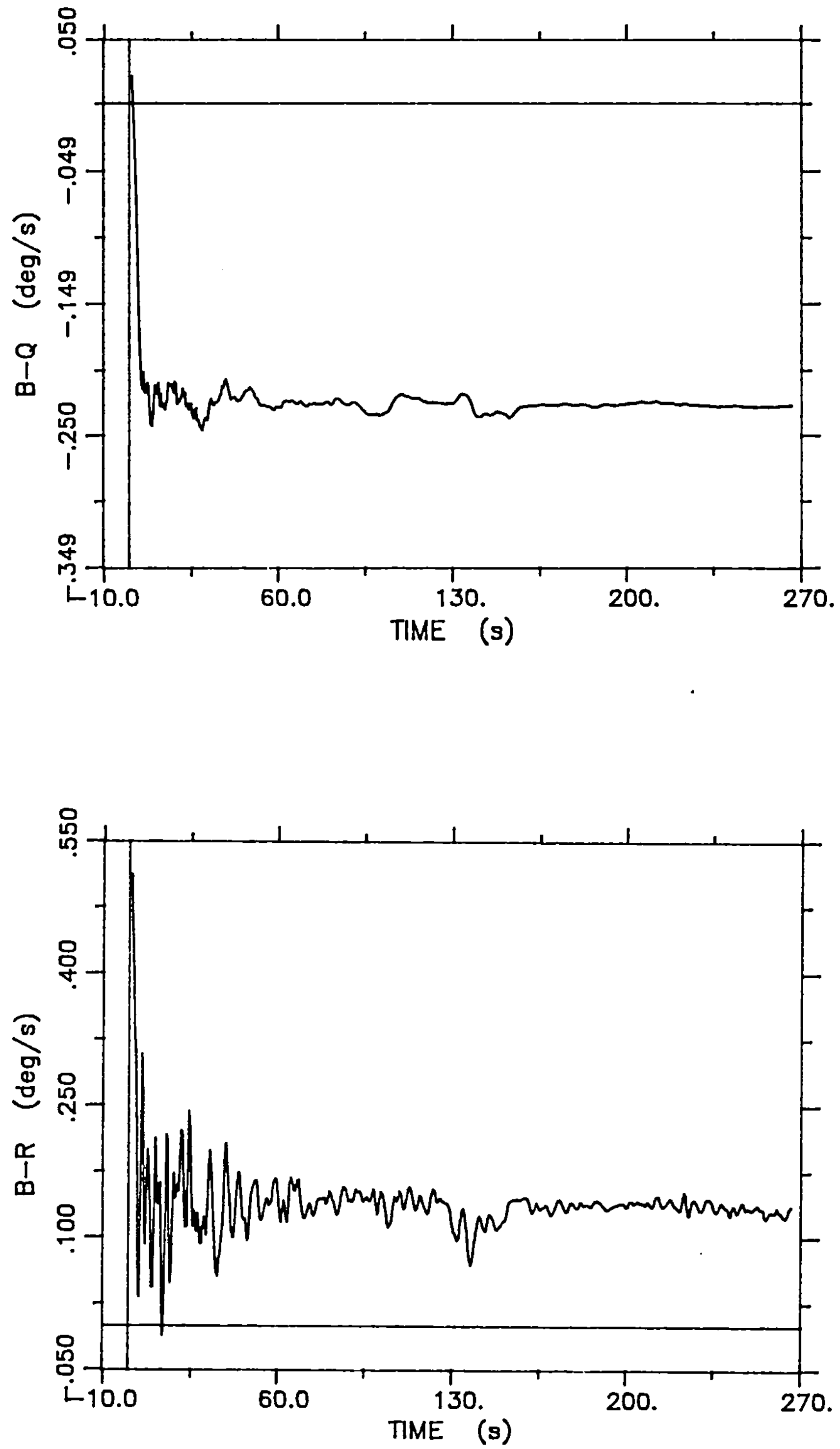


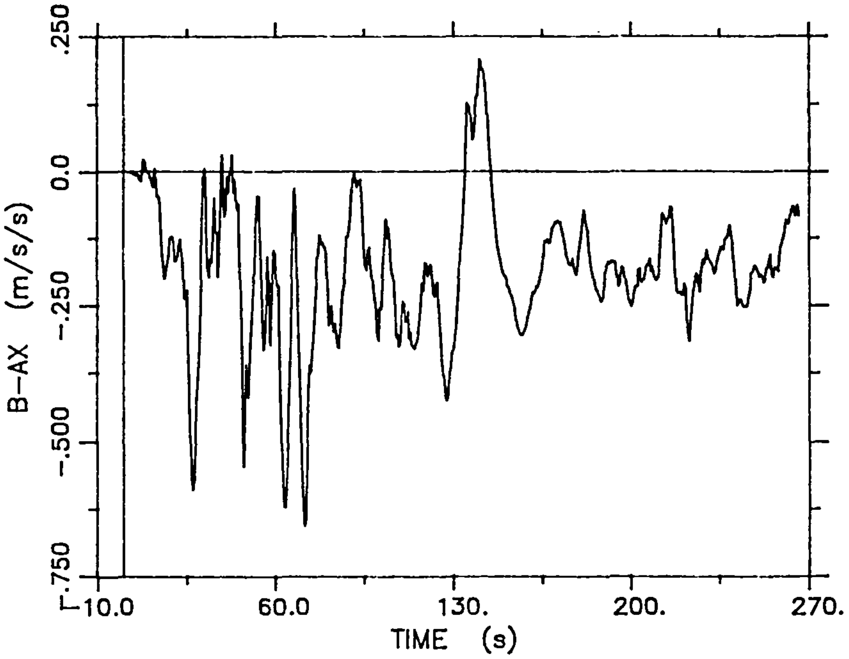
Figure 2.19: Pitch and yaw rate gyro bias estimates with 2 Hz gain update frequency

Table 2.4: No-fail filter residuals statistics : gain update frequency of 4 Hz

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
MLS-Azim.	+1.78E-03	+8.28E-03	+3.76E-02	-3.10E-02	deg
MLS-Elev.	+2.83E-04	+7.44E-03	+3.48E-02	-2.16E-02	deg
MLS-Range	+1.12E-01	+2.02E+00	+1.05E+01	-1.95E+01	m
IAS	+1.05E-01	+7.92E-01	+4.66E+00	-4.23E+00	m/s
IMU-Roll	-1.56E-04	+3.66E-02	+1.27E-01	-1.27E-01	deg
IMU-Pitch	+4.74E-04	+2.46E-02	+1.38E-01	-7.76E-02	deg
IMU-Yaw	-2.05E-05	+1.11E-01	+6.14E-01	-4.56E-01	deg

Table 2.5: No-fail filter residuals statistics : gain update frequency of 2 Hz.

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
MLS-Azim.	+1.18E-03	+8.34E-03	+3.28E-02	-2.95E-02	deg
MLS-Elev.	+3.38E-04	+7.02E-03	+3.13E-02	-2.22E-02	deg
MLS-Range	+1.01E-01	+1.97E+00	+1.07E+01	-1.92E+01	m
IAS	+5.13E-02	+7.57E-01	+4.67E+00	-4.24E+00	m/s
IMU-Roll	-1.11E-04	+3.63E-02	+1.25E-01	-1.26E-01	deg
IMU-Pitch	+5.33E-04	+2.48E-02	+1.38E-01	-8.07E-02	deg
IMU-Yaw	+3.55E-04	+1.10E-01	+6.06E-01	-4.53E-01	deg



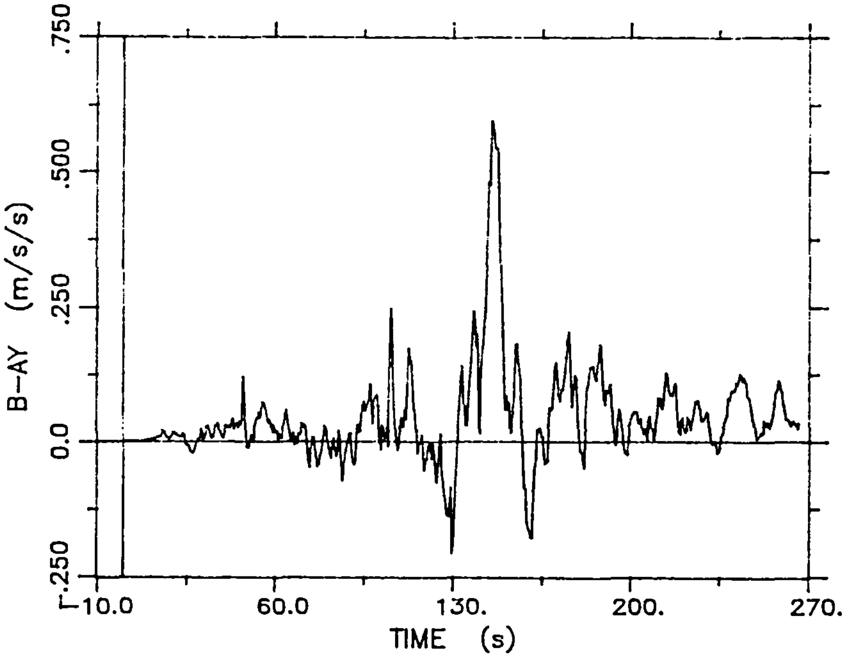
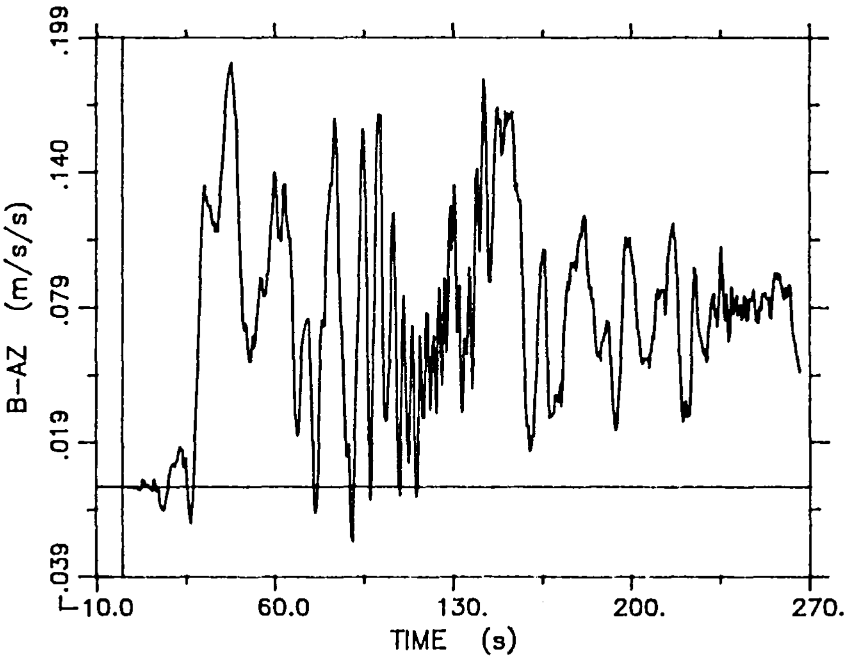


Figure 2.20: Longitudinal and lateral accelerometer bias estimates with 1 Hz update rate



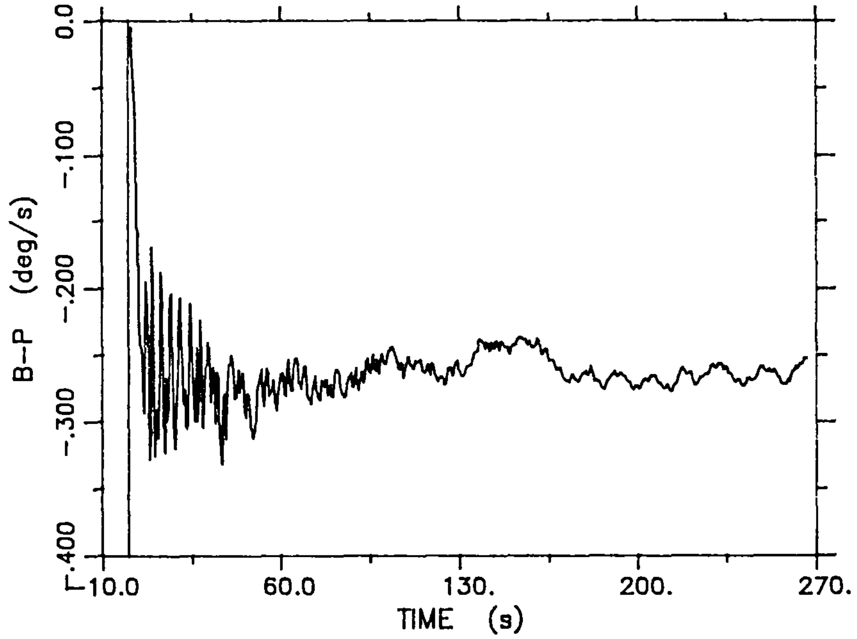
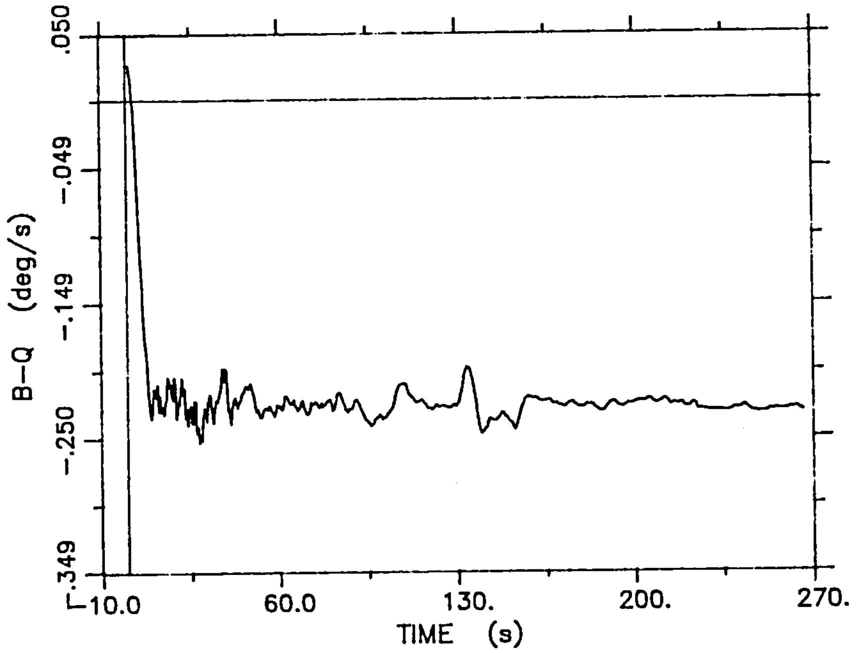


Figure 2.21: Vertical accelerometer and roll rate gyro bias estimates: 1 Hz update rate



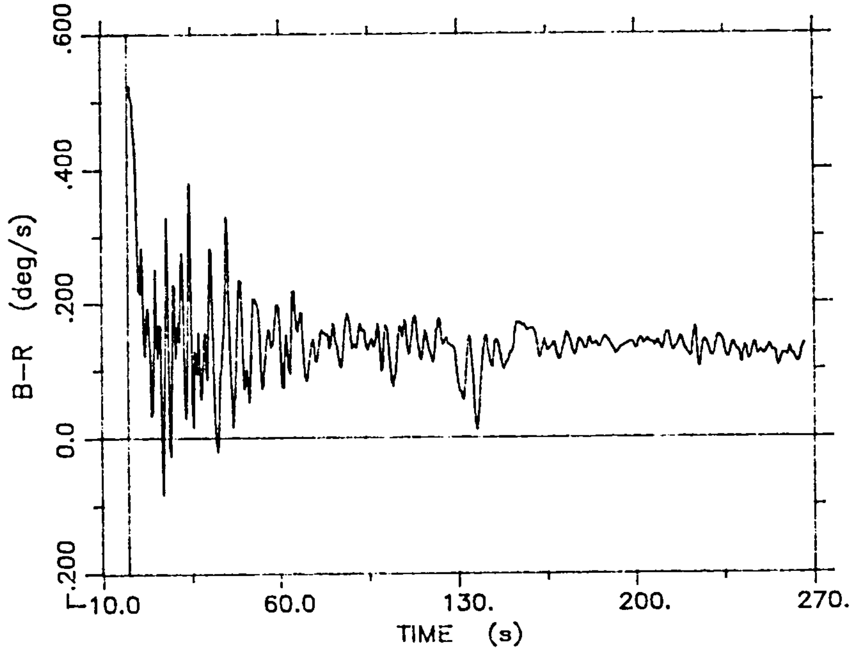


Figure 2.22: Pitch and yaw rate gyro bias estimates: 1 Hz update rate

Table 2.5 gives the residual sequence statistics for the same case and, again, we see a stable filter estimation behavior.

Finally, for the purpose of comparison, Figures 2.20-2.22 depict the bias estimation performance of the no-fail filter with a gain update frequency of 1 Hz. As expected, the biases take an extremely long time to converge; yet, the no-fail filter residual sequence statistics shown in Table 2.6 continue to verify a good estimation performance.

Table 2.6: No-fail filter residuals statistics : gain update frequency of 1 Hz.

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
MLS-Azim.	+3.32E-04	+8.52E-03	+2.99E-02	-3.95E-02	deg
MLS-Elev.	+5.96E-04	+6.76E-03	+3.49E-02	-2.28E-02	deg
MLS-Range	+8.72E-02	+1.97E+00	+1.12E+01	-1.94E+01	m
IAS	+2.29E-02	+6.95E-01	+4.68E+00	-4.01E+00	m/s
IMU-Roll	-2.73E-05	+3.58E-02	+1.23E-01	-1.28E-01	deg
IMU-Pitch	+1.01E-03	+2.52E-02	+1.32E-01	-9.35E-02	deg
IMU-Yaw	+4.17E-04	+1.08E-01	+6.24E-01	-4.51E-01	deg

In this chapter, we have presented the performance results for the no-fail filter with detectors inactivated. For the nominal emulation run, we have obtained a significant improvement in estimation performance by making some design modifications. Using piecewise constant gains and other changes, we have speeded up the execution performance to 1.3 times slower than real-time on the host development computer. The next chapter deals with the failure detection and isolation performance of the FINDS algorithm.

3. FAILURE DETECTION PERFORMANCE

In this chapter, we present the changes made to the FDI algorithm in FINDS which resulted in an improvement in detection performance and execution speed. In particular, we present the FDI performance with a series of injected bias failures, given the estimation performance of the no-fail filter of the previous chapter. The first section deals with the results obtained from a detectability analysis performed on the bank of first order detectors driven by the expanded innovations of the no-fail filter. The next section presents the performance of the FDI algorithm with the same set of bias failure runs as in [1]. In the next section, we present a new detection strategy which takes advantage of the improved no-fail filter estimation performance, and is capable of detecting sensor failures significantly faster. In addition, this new detection strategy allows the entire FINDS algorithm to execute at almost the same speed as the no-fail filter. Preliminary failure detection results using this new detection strategy are also presented. These results validate the detection performance predicted by the detectability analysis. In the final section, we present the results of our analysis involving the update of the no-fail filter gains at lower frequencies.

3.1 Failure Detectability Analysis

Referring back to equation (2.3.20) in [6], we have the following recursive relation for the i'th detector information matrix:

$$P_i^{-1}(k+1/k+1) = P_i^{-1}(k/k)+C_i^T(k+1,\hat{x}(k+1/k))R^{-1}(k+1)C_i(k+1,\hat{x}(k+1/k))$$

with $P_i^{-1}(k_0/k_0) = 0$

where k_0 denotes the time at the start of a residual window

R is the expanded innovations covariance matrix

C_i is the failure observation matrix for the i'th detector

The above expression hypothesizes the occurence of a particular failure at time k_0, and zero information about the bias jump magnitude at that instant. The second term on the right hand side of the above equation can be viewed as the incremental information to the i'th detector at every subsequent time instant after time k_0. The time rate of change of this incremental information determines the failure signature, and hence, the detectability of any given sensor [8]-[9].

Figures 3.1a-d show the behavior of the normalized (by sensor noise standard deviation) incremental information for the longitudinal accelerometer during four different phases of the nominal emulation run:

(a) initial phase after filter start-up, when the bias estimates exhibit significant transients;

(b) before the runway alignment maneuver but after bias estimates have converged to steady-state;

(c) during the runway alignment maneuver; and

(d) final descent towards touchdown.

Figure 3.1a clearly shows the low level of incremental information available to the longitudinal accelerometer detector in the initial transient phase of the no-fail filter. This is due to the large error covariance for the longitudinal accelerometer bias estimate during this initial transient stage. Any failure in this sensor during this phase would be significantly compensated by the longitudinal accelerometer bias estimate leading to a low failure signature on the residuals. Once the bias estimate for that accelerometer converges (i.e., the associated error covariance is reduced), then the behavior of the incremental information changes as seen in Figures 3.1b-d. In each of these plots, we see a gradual rise in incremental information from the start of the residual window. This behavior is due to the use of the accelerometers in the no-fail filter as inputs. These figures also show that the detectability of the longitudinal accelerometer is essentially independent of aircraft maneuvers or flight segments.

Figures 3.2a-d and 3.3a-d show the incremental information behavior, in the same four flight phases, for the lateral and vertical accelerometers, respectively. Both plots 3.2a and 3.3a depict the same low-level information available to the lateral and vertical accelerometer detectors due to bias estimation uncertainty as in 3.1a. However, once these bias estimates converge, Figures 3.2b-d and 3.3b-d show a definite increase in the incremental information, and thus, an increase in detectability for these accelerometers in the latter phases of the flight, as the aircraft approaches the runway. This is because the MLS azimuth and elevation sensors, which generate the most significant signatures after lateral and vertical accelerometer failures, become more sensitive to position errors as the aircraft nears the runway. In contrast, the MLS range sensor, which generates the most significant signature in the case of a longitudinal accelerometer failure, has approximately a constant sensitivity to position errors throughout the flight.

The same analysis performed on the rate gyros shows that the detectability of these sensors is less dependent on flight segments, in contrast to the accelerometers. This is because the correlation between the rate gyros and IMU attitude measurements arising from the rotational kinematics is essentially invariant throughout the flight. The only exception here is during an aircraft maneuver when the incremental information tends to show a slightly increased slope. This implies that the rate gyro failures, especially for roll and yaw, are more detectable during an aircraft maneuver. Figures 3.4 and 3.5 show the incremental information time history for the roll, pitch and yaw rate gyros in the aircraft maneuver flight segment. The rapid increase in the incremental information in these plots implies increased detectability for these sensors, especially when compared to the same plots for the accelerometers.

The measurement sensors portray a different incremental information behavior than the input sensors as can be seen in Figures 3.6 and 3.7, for the MLS azimuth and range, IAS and IMU roll attitude sensors, respectively. For all the measurement sensors, the incremental information depicts an immediate jump at the beginning of a residual window, followed by an exponential decay to a steady-state. This steady-state differs for the MLS sensors from the other measurement sensors since the no-fail

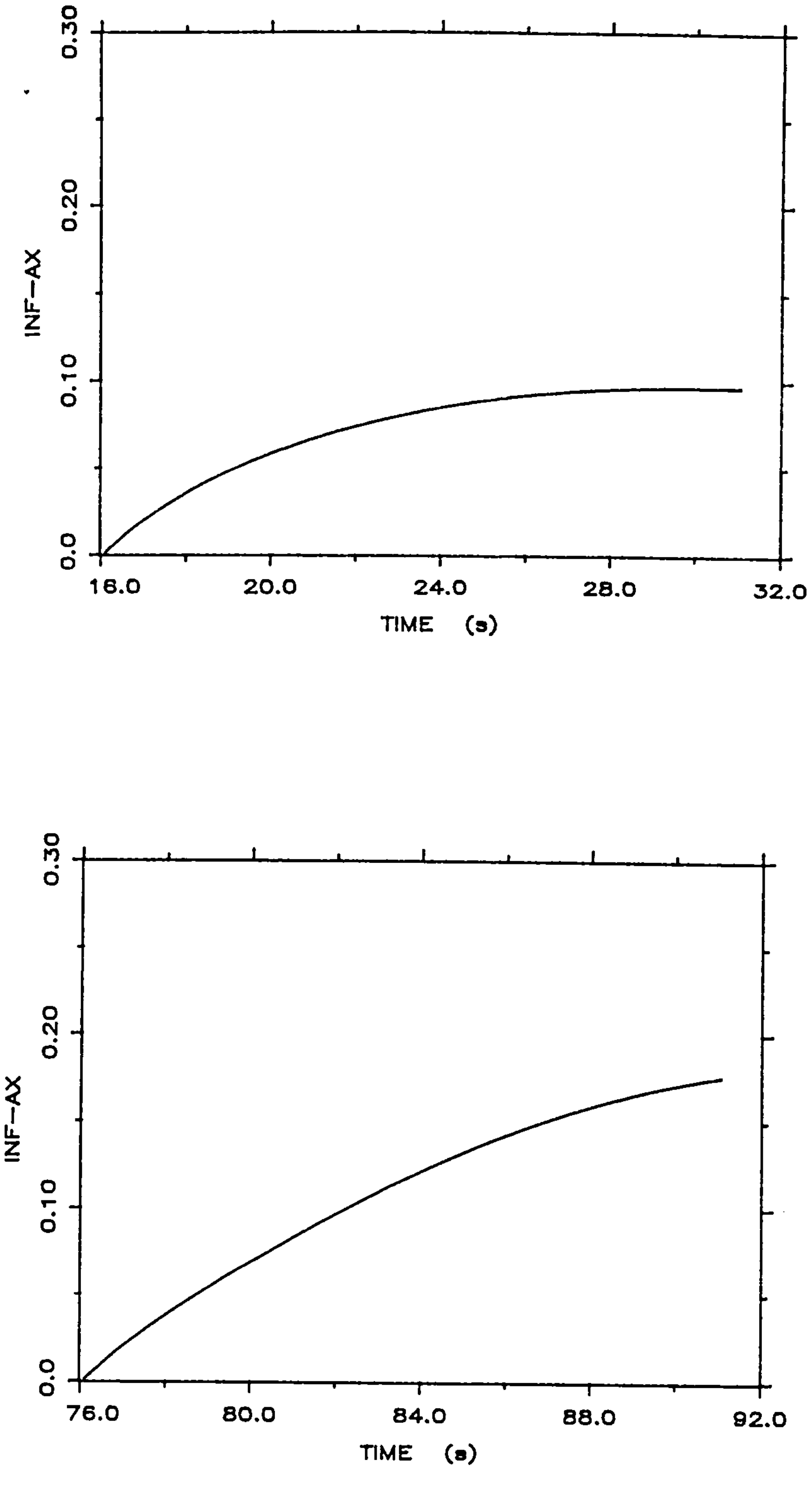


Figure 3.1a,b: Incremental information behavior for longitudinal accelerometer during initial flight segments

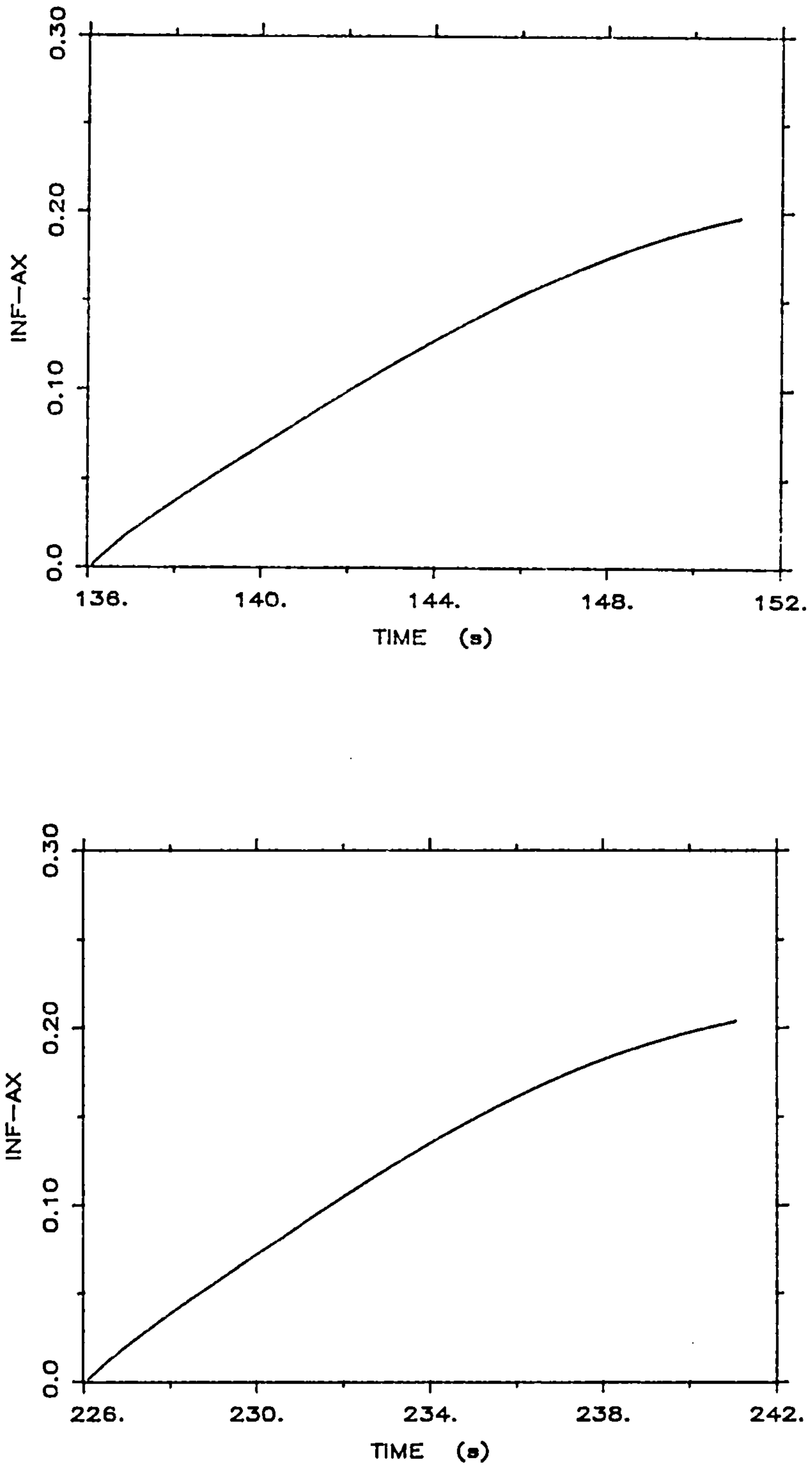


Figure 3.1c,d: Incremental information behavior for longitudinal accelerometer during aircraft maneuver and final descent

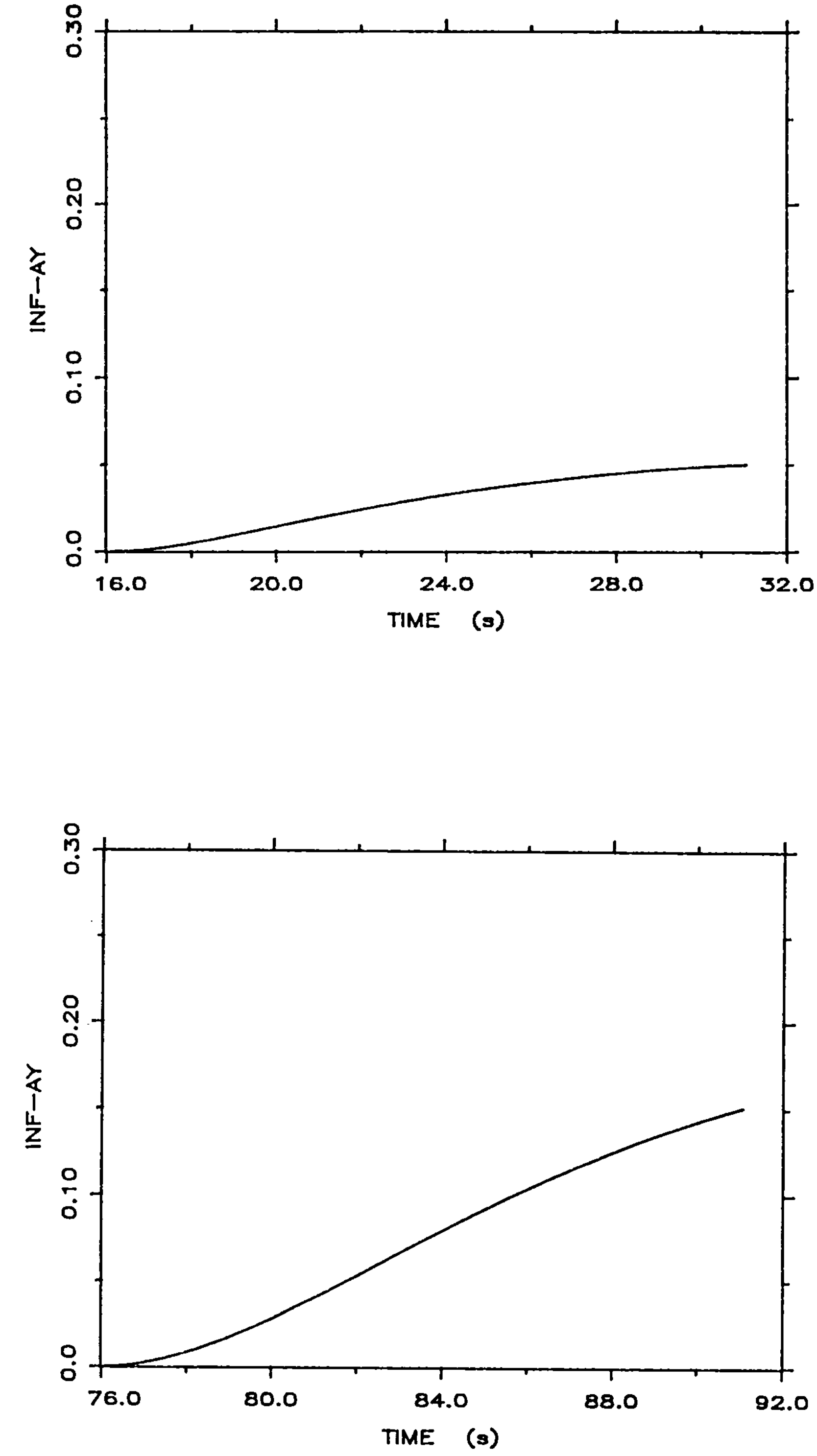


Figure 3.2a,b: Incremental information behavior for lateral accelerometer during initial flight segments

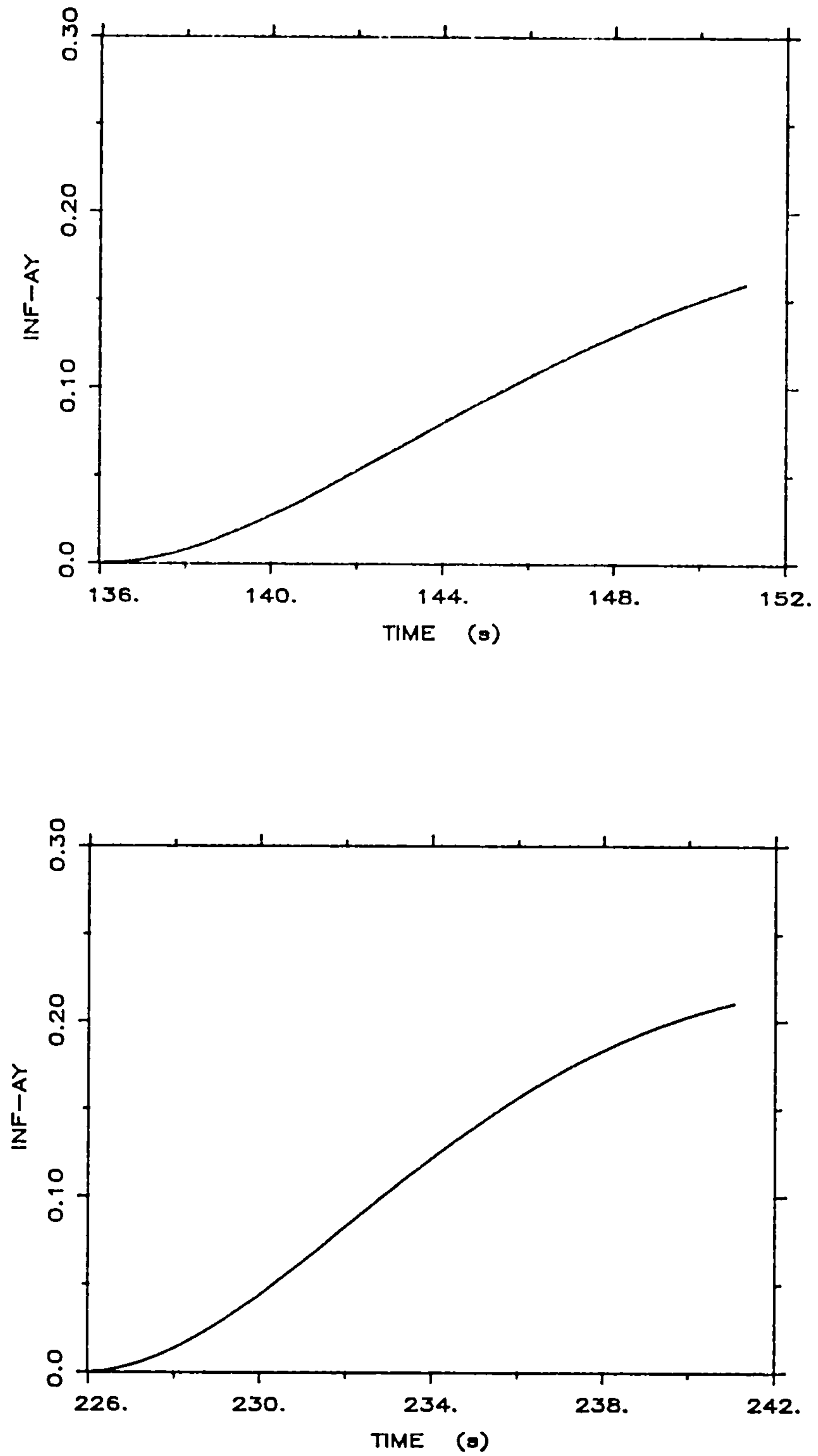


Figure 3.2c,d: Incremental information behavior for lateral accelerometer during aircraft maneuver and final descent

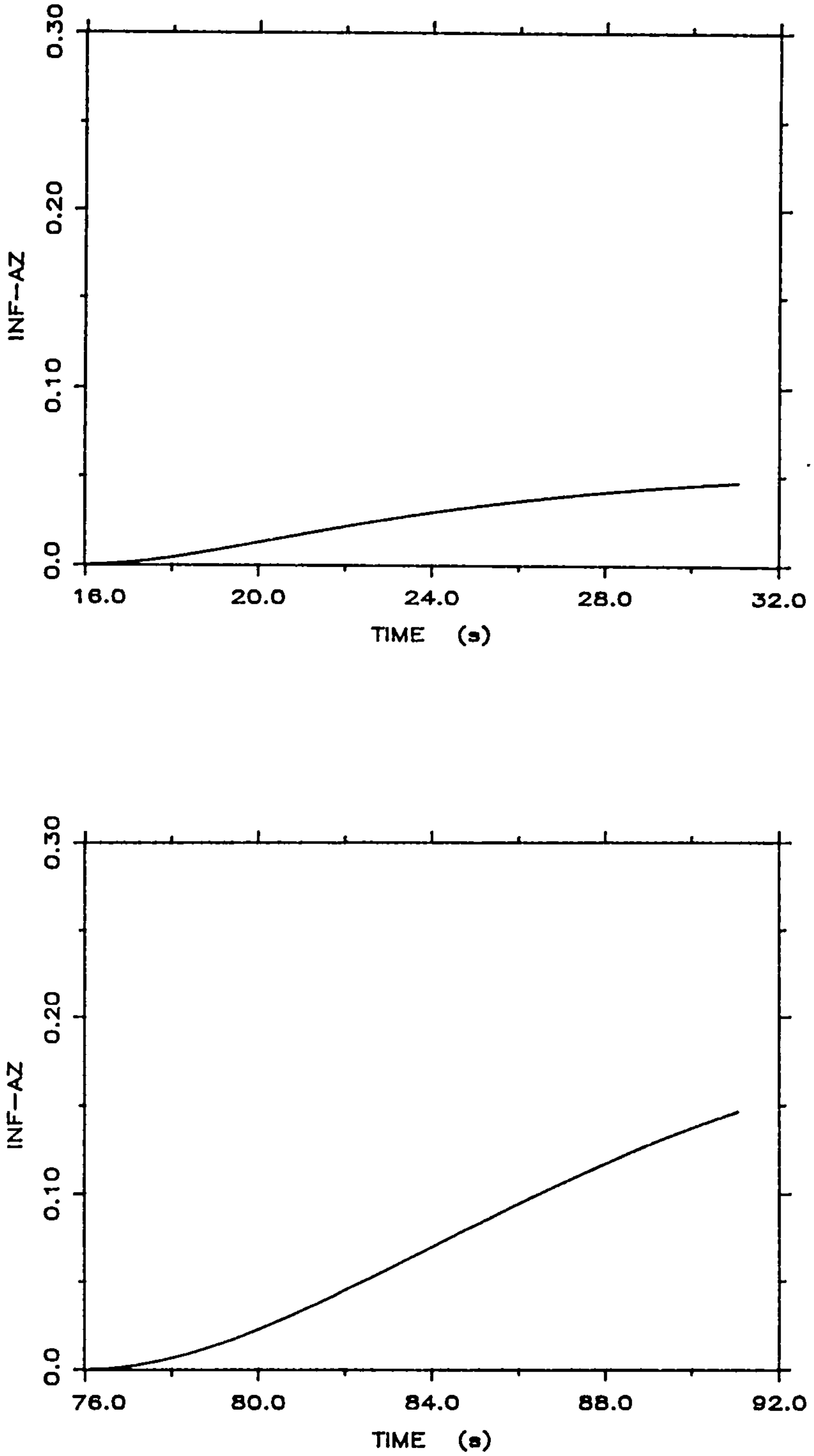


Figure 3.3a,b : Incremental information behavior for vertical accelerometer during initial flight segments

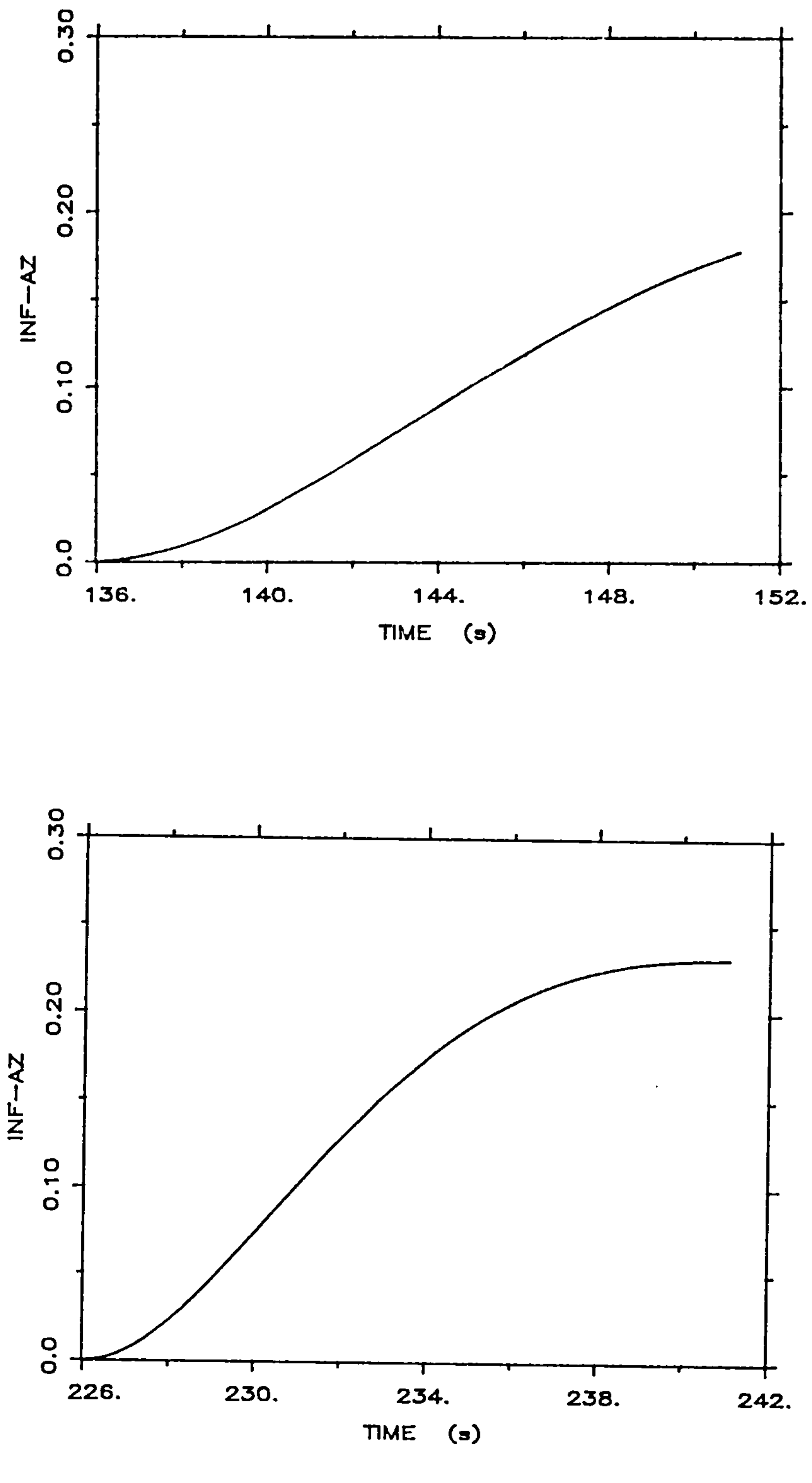


Figure 3.3c,d : Incremental information behavior for vertical accelerometer during aircraft maneuver and final descent

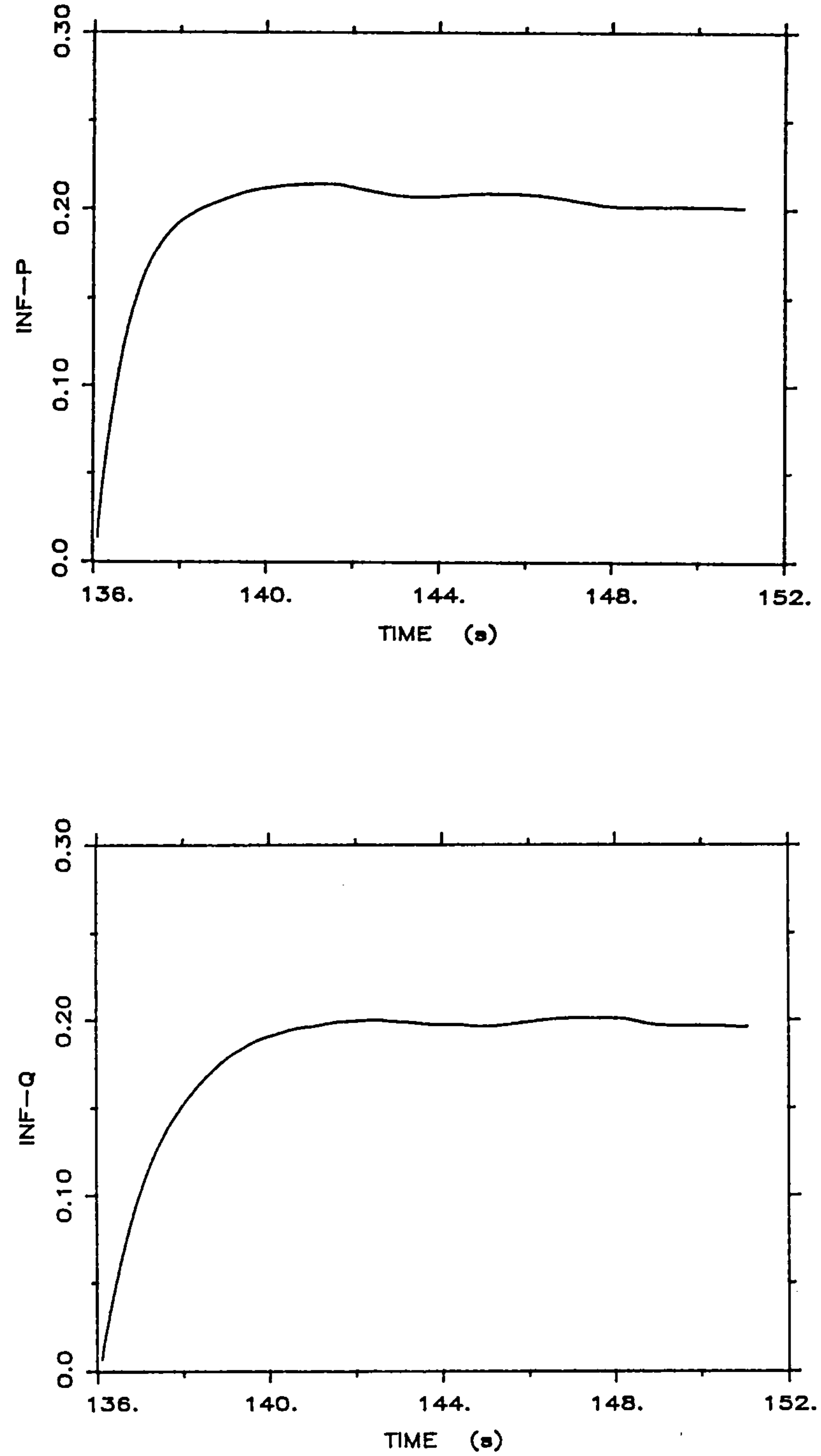


Figure 3.4: Incremental information behavior for roll and pitch rate gyros in aircraft maneuver flight segment

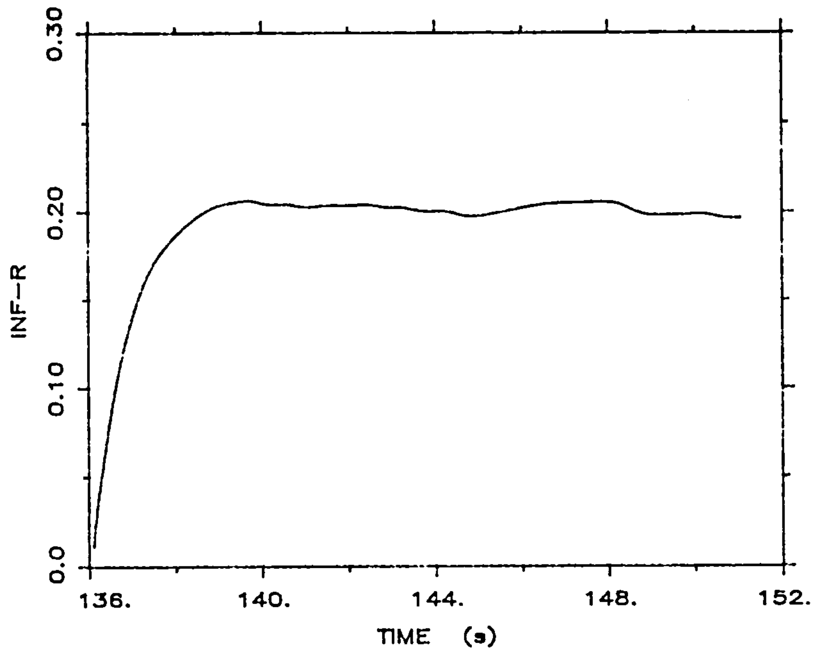


Figure 3.5: Incremental information behavior for yaw rate gyro in aircraft maneuver flight segment

filter uses only one replication of the MLS sensors as opposed to dual redundancy for the IAS and IMU sensors. The behavior of these incremental information plots is almost identical in each of the four flight segments. Thus, the measurement sensors are equally detectable throughout the flight.

We can now summarize the results of our detectability analysis as follows:

1) For a measurement sensor, the incremental information is highest at the beginning of a residual window with a subsequent exponential decay to a steady-state. On the other hand, for an input sensor, the incremental information is lowest at the beginning of a residual window with a gradual build-up to steady-state as the

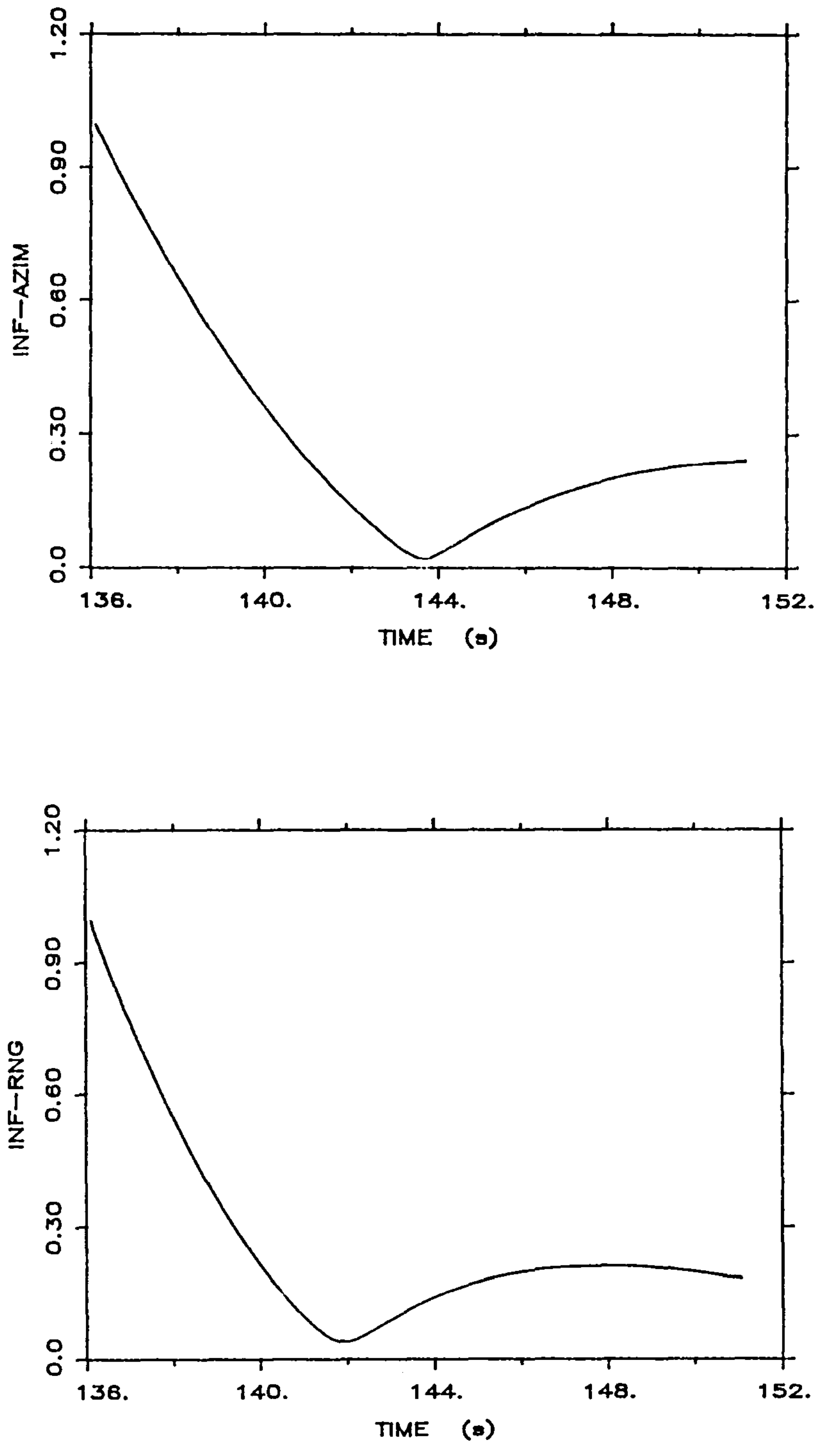
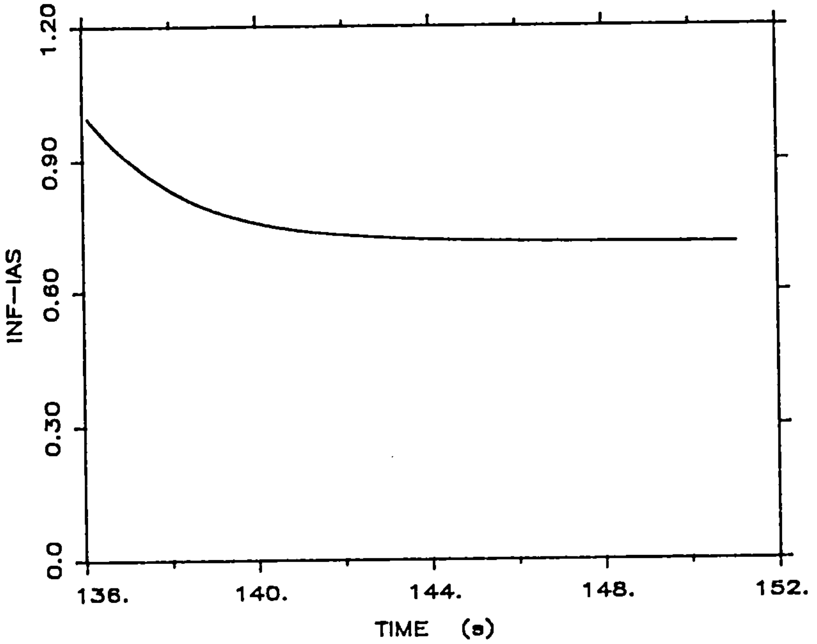


Figure 3.6: Incremental information behavior for MLS azimuth and range in aircraft maneuver flight segment



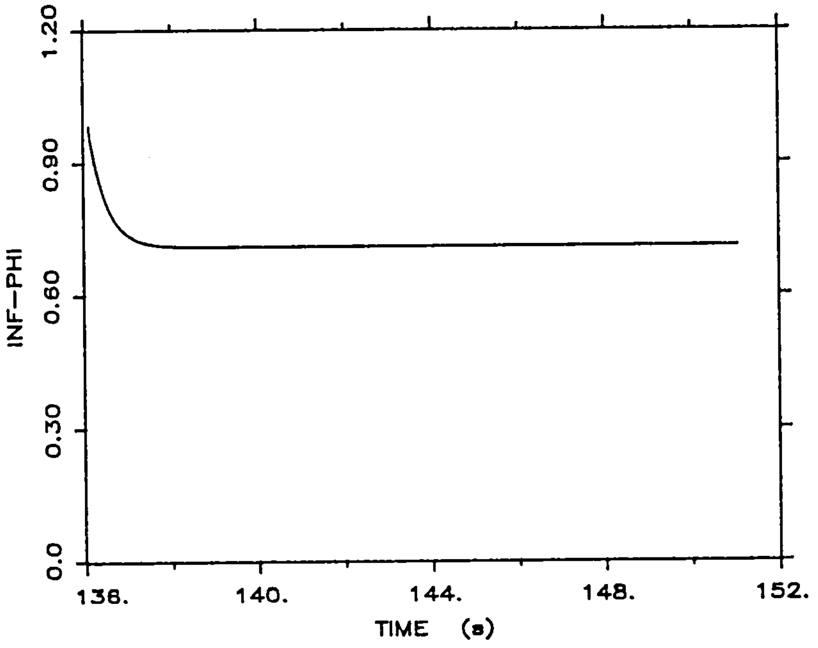


Figure 3.7: Incremental information behavior for IAS and IMU roll attitude in aircraft maneuver flight segment

failure gets propagated through the no-fail filter dynamics.

2) Of the input sensors, the linear accelerometers take the longest time to generate significant incremental information, and are hence, the least detectable of all the sensors.
3) Detectability of accelerometer failures increases as the aircraft approaches the runway, especially for the lateral and vertical accelerometers.
4) Rate gyro failures are more detectable than accelerometer failures. Detectability for the rate gyros is independent of flight segments.
5) Measurement sensor detectability is not a function of either bias estimation performance or aircraft maneuvers/flight segments.
6) Steady state incremental information for a particular sensor is dependent upon the number of replications of that type used by the no-fail filter and whether its bias is estimated or not.

These observations become more evident as we examine the failure detection performance of the FINDS algorithm in the following sections.

3.2 Baseline Failure Detection and Isolation Performance

In this section, we present the bias failure detection performance of the FINDS algorithm for the same series of failures injected into the flight data as reported in [1]. As discussed earlier, the no-fail filter residual sequence forms the input to the detectors. Hence, the amount of use that the no-fail filter makes of any particular sensor in its estimation process determines its failure signature on the residuals and thus, its detectability. In accordance with the estimation performance presented in Chapter 2, the no-fail filter makes a more balanced use of all the input and measurement sensors, in contrast to the earlier results presented in [1]. The impact of this improved estimation is reflected in the detection performance summary presented here.

Note that for all the six bias failure runs discussed in this section, the detector and sensor healer parameters are essentially the same as given in Tables 3.4 and 3.5 in [1], respectively. The only difference is in the detector sensor noise design parameters since these parameters are chosen depending upon the statistics of the no-fail filter residual sequences. Table 3.1 shows these new sensor noise parameters employed in the computation of the filter measurement residual covariance used by the detectors.

Table 3.2 presents the detection performance summary of the FINDS algorithm with injected bias failures, over a series of six emulation runs. The failure onset times in these runs, although in the same flight segment, are not exactly the same instants as discussed in [1] but differ by a few seconds. Thus, these failures occur at different instants within a decision window than the earlier reported runs.

Table 3.1: Design values for measurement sensor noise parameters used by the detectors

VARIABLE	NOISE SD PER REPL.	REPLICATIONS USED	UNITS
MLS-Azim.	3.00E-02	1	deg
MLS-Elev.	3.50E-02	1	deg
MLS-Range	5.50E+00	1	m
IAS	2.00E+00	2	m/s
IMU-Roll	1.30E-01	2	deg
IMU-Pitch	1.50E-01	2	deg
IMU-Yaw	5.00E-01	2	deg

By comparing the sensor noise values used by the no-fail filter in [1] and the current study, we observe that the current no-fail filter (i) has a new wind model, (ii) uses the MLS, IAS and vertical accelerometer sensors less, and (iii) uses the IMU attitudes and lateral accelerometer more than before. This is clearly evident in the detection times for these respective sensors as compared to Table 3.13 in [1]. Thus, in run 3, the lateral accelerometer failure gets detected faster than before whereas in run 4, the detection time for the vertical accelerometer failure is higher than in [1]. Similarly, we see significantly faster detection times for the IMU attitude sensor failures, and a slightly higher detection time for the rate gyros. There is no major change in the case of the MLS sensors, but an IAS failure, by virtue of its higher sensor noise characteristics, takes longer to get detected.

The most important difference between these two sets of bias failure summaries is the absence of any false alarms in any of the six current emulation runs. Moreover, the current version

Table 3.2: Baseline bias failure detection performance summary

RUN	SENSOR	FAILURE	DETECTION	FAILURE LEVEL		UNITS
NO.	TYPE	INJ.AT (s)	TIME (s)	INJECTED	ESTIMATED	
1	MLS-Elev.	81.55	0.20	0.18	0.186	deg
2	IMU-Yaw	67.60	0.05	4.0	4.042	deg
	MLS-Azim.	111.25	0.15	0.18	0.154	deg
	MLS-Range	225.75	0.25	40.0	39.179	m
3	IAS	59.70	0.70	9.0	9.906	m/s
	Gyro-Roll	141.20	0.75	0.90	1.339	deg/s
	Acc.-Lat.	241.60	4.90	1.275	4.868	m/s/s
4	IMU-Roll	93.80	0.35	1.50	1.232	deg
	Acc.-Vert.	200.90	5.30	1.471	26.066	m/s/s
5	IMU-Pitch	81.65	0.05	2.0	1.276	deg
	Gyro-Yaw	153.35	2.40	2.0	2.331	deg/s
	Acc.-Long.	222.45	6.85	1.471	6.184	m/s/s
6	Gyro-Pitch	179.85	1.05	1.0	0.254	deg/s

of the FINDS algorithm has a 115 Kb program size and an execution speed of 30 times slower than real-time as opposed to the previous version which had a program size of 340 Kb and an execution speed of 120 times slower than real-time on the host development computer.

Since the major thrust of this current effort is to adapt the FINDS algorithm to a real-time operation without compromising on either estimation or detection performance, this modified FINDS algorithm, with its smaller program size and an improved and well balanced detection performance, still needed further changes. The implementation of the bank of detectors, by virtue of its slow speed of execution, was the main target for further modifications; hence, it has been replaced by a new detection strategy presented in the next section.

3.3 New Detection Strategy

The FDI algorithm in FINDS has an extremely fast detection performance with no false alarms, but a slow execution speed. This is largely because of the implementation of the bank of first order detectors. For instance, with the current sensor suite, a total of 17 detectors need to be executed at every time instant. Moreover, the performance of this implementation is also affected by the use of a fixed length detector window which is necessary in order to minimize computational resources. This constraint makes the detection time and failure level estimate dependent on the time of failure onset within a particular detection window. Finally, by using the expanded residual sequence as inputs (as necessitated by the isolation algorithm), the detection strategy does not make optimum use of the excellent statistics of the no-fail filter averaged residuals. In order to alleviate these inherent limitations, a new failure decision test has been implemented for the FINDS algorithm, as discussed below.

Given an m-dimensioned vector r with a Gaussian distribution with mean μ and covariance R, then

$$N\,(\bar{r} - \mu)^T R^{-1} (\bar{r} - \mu)$$

has a Chi-square distribution with m degrees of freedom [10], where $\bar{r}$ is the sample mean of r over N samples.

From Table 2.3 it is seen that the no-fail filter residuals have an extremely small mean value across the entire emulation run. Moreover, Figures 2.9-2.12 show the uncorrelated behavior of each of these residual vector entries. Thus, we can perform a detection test on the no-fail filter residuals (without performing an isolation test) over a moving window, as follows. Given the residual sequence of the no-fail filter defined by [6]:

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

where

$$\bar{y}(k) = [y^1(k) + y^2(k)]/2$$

compute the sample mean of this residual sequence over a moving window :

$$\bar{r}(k) = \frac{1}{N} \sum_{j=k-N+1}^{k} r(j)$$

Then, perform the following test of mean by computing the likelihood ratio

$$\alpha = N\ (\bar{r}^T(k)\ R^{-1}\ \bar{r}(k))$$

and comparing to a predetermined threshold, in order to decide on a sensor failure. This test procedure can be implemented for different moving windows N_i. In the event of a failure decision, a failure isolation test is made by running the bank of detectors over the last N_i no-fail filter residuals.

Table 3.3 shows the statistics of these averaged residuals and the associated likelihood ratios for the nominal emulation run of section 2.2. Both the previous simulation version and the current flight data driven emulation version of the FINDS algorithm show that measurement sensor failures propagate through the no-fail filter dynamics almost instantaneously. On the other hand, it takes a longer amount of time for the input sensor failure signatures to show up in the no-fail filter residuals. This observation is also validated by the incremental information analysis of section 3.1. Hence, two different moving windows have been incorporated in the new detection strategy; one of length 1 sample for measurement sensor failure detection, and the other of length 10 samples to detect input sensor failures. The computed means and standard deviations of these residuals are essentially the same for both moving windows of length 1 and 10 samples, thus highlighting the uncorrelated nature of these residuals.

The minimum/maximum values for the likelihood ratio are empirically calculated as 13.5 for the moving window of 1 sample and as 63.4 for the window of 10 samples. Hence, a Chi-square test with a Type I error size of 0.01 (which implies a threshold of 18.5 with the given seven degrees of freedom) would yield no false alarms for the decision window of length 1. On the other hand, a test threshold of 65-70 would yield no false alarms in the case of the moving window of length 10, thus implying a lower Type I error size.

Using the above threshold values, we have made a series of runs where a failure is injected into a specific sensor at different flight segments in the emulation. Table 3.4a presents the results of this series of test runs. The first set of runs include a sensor failure occuring at 82.10 seconds into the flight, when all bias estimates have converged and the bank maneuver is yet to be executed. In the case of the input

Table 3.3: No-fail filter residuals and likelihood ratio statistics for moving windows of 1 and 10 samples: nominal run

SENSOR	MEAN	STD.DEV.	MAX	MIN	UNITS
	MOVING WINDOW OF 1 RESIDUAL SAMPLE				
MLS-Azim.	+1.37E-03	+7.35E-03	+2.87E-02	-3.06E-02	deg
MLS-Elev.	+4.22E-04	+8.26E-03	+3.28E-02	-2.45E-02	deg
MLS-Range	+1.67E-01	+2.03E+00	+1.08E+01	-1.98E+01	m
IAS	+1.44E-01	+8.36E-01	+4.66E+00	-4.19E+00	m/s
IMU-Roll	-1.73E-03	+3.70E-02	+1.29E-01	-1.32E-01	deg
IMU-Pitch	+3.19E-03	+2.50E-02	+1.40E-01	-7.20E-02	deg
IMU-Yaw	+1.05E-02	+1.16E-01	+6.36E-01	-4.52E-01	deg
LRT-01	+8.73E-01	+9.43E-01	+1.35E+01	-------	---
	MOVING WINDOW OF 10 RESIDUAL SAMPLES				
MLS-Azim.	+1.36E-03	+5.31E-03	+1.64E-02	-1.63E-02	deg
MLS-Elev.	+4.28E-04	+6.84E-03	+2.06E-02	-2.01E-02	deg
MLS-Range	+1.68E-01	+1.67E+00	+6.36E+00	-1.34E+01	m
IAS	+1.44E-01	+7.76E-01	+3.58E+00	-3.15E+00	m/s
IMU-Roll	-1.66E-03	+2.89E-02	+7.28E-02	-7.97E-02	deg
IMU-Pitch	+3.26E-03	+1.88E-02	+1.12E-01	-4.73E-02	deg
IMU-Yaw	+1.08E-02	+1.08E-01	+4.81E-01	-3.70E-01	deg
LRT-10	+8.08E+00	+6.45E+00	+6.34E+01	-------	---

sensors, we see a significant decrease in detection time as compared to earlier results. As for the measurement sensors (except IAS), the failure is detected without any delay, yielding a corresponding detection time of zero seconds. This is due to the instantaneous jump in the incremental information, i.e., an immediate signature on the measurement residuals at failure onset time.

The IAS sensor failure does not get detected in the same instantaneous manner because of the low failure level injected. Due to the no-fail filter's use of dual redundant IAS sensors, the 9 m/s failure gets averaged and a jump of approximately 4.5 m/s is seen in the IAS residual. Since the sensor noise design value for the IAS sensor in the detectors is 2 m/s, this corresponds to a 2.25-σ failure. This low failure level is not adequate enough to push the likelihood ratio above the threshold set for the moving window of 1 sample -- but the cumulative effect on the moving window of 10 samples is enough to detect the failure 8 or 9 samples after its occurence. This example shows the sensitivity of this new detection strategy. In fact, test runs were made by injecting a 15 m/s failure in the IAS sensor, which looks like a 2.5-σ failure in the no-fail filter averaged measurements and generates a 7.5 m/s jump in the IAS residual, which corresponds to a 3.7-σ failure to the detectors. This failure does get detected with a zero detection time, similar to the other measurement sensors.

In the second series of runs, we have singular failure occurences in each sensor at 145.40 seconds, in the middle of the aircraft maneuver for runway alignment. Interestingly, the lateral and vertical accelerometers show a decrease in detection time of 0.45 and 0.10 seconds, respectively. Also, the yaw rate

Table 3.4a: Bias failure detection summary with new detection test

SENSOR TYPE	FAILURE LEVEL INJECTED	DETECTION TIME FOR FAILURE INJECTED AT		
		82.10 s	145.40 s	238.70 s
Acc.-Long.	1.47 m/s/s	4.05 s	3.95 s	3.75 s
Acc.-Lat.	1.28 m/s/s	5.20 s	4.75 s	2.70 s
Acc.-Vert.	1.47 m/s/s	4.95 s	3.85 s	2.10 s
Gyro-Roll	0.90 deg/s	0.45 s	0.35 s	0.40 s
Gyro-Pitch	1.0 deg/s	0.45 s	0.50 s	0.50 s
Gyro-Yaw	1.0 deg/s	1.65 s	1.45 s	1.50 s
MLS-Azim.	0.18 deg	0.0 s	0.0 s	0.0 s
MLS-Elev.	0.18 deg	0.0 s	0.0 s	0.0 s
MLS-Range	40.0 m	0.0 s	0.0 s	0.0 s
IAS	9.0 m/s	0.45 s	0.40 s	1.20 s
IMU-Roll	1.50 deg	0.0 s	0.0 s	0.0 s
IMU-Pitch	2.0 deg	0.0 s	0.0 s	0.0 s
IMU-Yaw	4.0 deg	0.0 s	0.0 s	0.0 s

gyro shows a slightly higher detectability during maneuvers and hence, gets detected 0.20 seconds faster in this flight segment. The rest of the input sensors do not show any appreciable change in detection time and the measurement sensor failures (except IAS) are again detected instantaneously.

The third set of runs included specific sensor failures occuring at 238.70 seconds into the emulation, with the aircraft on its final descent path and about 4000 m from the runway.

Again, we see that the lateral and vertical accelerometer failures have a significant decrease in detection time, validating our observations from the detectability analysis. The longitudinal accelerometers and rate gyros do not show any change in detection time across the three runs as expected from the incremental information analysis. The measurement sensor failures are again detected without any delays. The IAS sensor failure, however, takes longer to detect during this phase of the flight because of the presence of wind gusts, ground effects and the larger magnitude IAS residuals as seen in Figure 2.10.

Table 3.4b presents the results for the same series of three test runs using the old detection strategy. A comparison of these failure detection times with those of Table 3.4a shows that the new detection strategy performs significantly better. Note also that the failure detection times with the old detection strategy do not exhibit the trend across the three series of test runs predicted by the incremental information analysis. This is due to the dependence of the old detection strategy on the failure onset time within a given fixed detection window.

It is also important to note that all of the above runs with the new detection strategy execute at almost the same speed as the no-fail filter estimator; actual time tests indicate that a run with the new detection algorithm takes approximately 10 % more execution time than corresponding 'estimation only' runs. With this implementation, the failure isolation and sensor reconfiguration modules get activated only after the detection of

Table 3.4b: Bias failure detection summary with old detection test

SENSOR TYPE	FAILURE LEVEL INJECTED	DETECTION TIME FOR FAILURE INJECTED AT **		
		82.10 s	145.40 s	238.70 s
Acc.-Long.	1.47 m/s/s	8.70 s	11.20 s	8.15 s
Acc.-Lat.	1.28 m/s/s	11.50 s	10.70 s	6.00 s
Acc.-Vert.	1.47 m/s/s	11.50 s	8.45 s	4.25 s
Gyro-Roll	0.90 deg/s	1.00 s	0.70 s	0.90 s
Gyro-Pitch	1.0 deg/s	0.75 s	0.75 s	0.90 s
Gyro-Yaw	1.0 deg/s	n.d.	n.d.	n.d.
MLS-Azim.	0.18 deg	0.05 s	0.25 s	0.15 s
MLS-Elev.	0.18 deg	0.05 s	0.25 s	0.20 s
MLS-Range	40.0 m	0.05 s	0.25 s	0.10 s
IAS	9.0 m/s	0.20 s	0.45 s	1.25 s
IMU-Roll	1.50 deg	0.05 s	0.15 s	0.10 s
IMU-Pitch	2.0 deg	0.05 s	0.30 s *	0.05 s
IMU-Yaw	4.0 deg	0.05 s	0.25 s	0.15 s

* A false-alarm of pitch rate gyro occurred 0.20 s after failure injection

** Failure time of 82.10 s coincides with the beginning of a detection/decision window; that is not the case with the other two failure onset times

a failure, and this aspect of the new strategy is currently being implemented.

A final comment regarding the use of replicated sensor measurements is in order here. By using single replications of all the measurement sensors in the no-fail filter and keeping second replications as stand-by as in the case of MLS sensors, lower levels of injected failures can be detected. However, this would be at the expense of no-fail filter estimation performance since the effective measurement noise in the averaged measurements is lower than that in the individual sensor signals.

3.4 Detection Performance With Piecewise Constant Gains

In section 2.3 of the previous chapter, we have presented the estimation performance of the no-fail filter with piecewise constant gains and noted satisfactory state estimation, even for the low gain-update frequency of 1 Hz, yielding low-mean, white filter residual sequences. However, as the gain update frequency is lowered, the bias estimates take longer to converge to steady-state values. In this section, we present the effect of these lower gain update frequencies on the failure detection performance using the new decision test discussed in the previous section.

The statistics for the no-fail filter residual sequences obtained with gain update frequencies of 4 Hz, 2 Hz and 1 Hz (shown in Tables 2.4-2.6, respectively) exhibit a small mean and essentially uncorrelated behavior. Thus, the new decision test can be performed on these residuals without violating any of the assumptions. Moreover, since the minimum and maximum values of these residual sequences lie within the same limits as the no-fail filter residuals with nominal gain update frequency of 20 Hz, the same sensor failure likelihood ratio thresholds have been used for all the gain update frequencies, thus yielding a false alarm rate of zero with this flight data.

Table 3.5 presents the results (with the new detection strategy) for specific sensor failures injected into the flight data at 82.10 seconds. For our test runs, we have again used no-fail filter gain update frequencies of 20 Hz, 4 Hz, 2 Hz, 1.33 Hz and 1 Hz. Considering the results of the 20 Hz update rate as the baseline case, we see that for the input sensors, accelerometer failure detection performance gets affected the most as the update frequency is lowered. The main reason for this is the relatively slow convergence of the accelerometer biases. With gain update frequencies lower than 4 Hz, these bias estimates have not yet converged to steady-state when the bias failure is injected into the flight data.

Thus, with the accelerometer bias estimation error covariance still high, these bias estimates begin to converge to a new steady-state after the injection of the failure, thus nullifying the effect of an accelerometer failure on the no-fail filter residuals. In other words, the injected bias failure is absorbed by the bias filter as it converges to a new steady-state bias for the accelerometer. However, since rate gyro bias estimates converge faster, the lower update frequencies have relatively no effect on rate gyro failure detection performance. The exception to this is the yaw rate gyro -- its failures not being detected at 1.33 Hz and 1 Hz update rates can be attributed to the particular failure level injected combined with high noise characteristics of the IMU yaw sensor. As for measurement

Table 3.5: Effect of piecewise constant gains on detection time with failures injected at 82.10 s

SENSOR TYPE	DETECTION TIME WITH GAIN UPDATE FREQUENCY OF 20 Hz	4 Hz	2 Hz	1.33 Hz	1 Hz
Acc.-Long.	4.05 s	5.25 s	n.d.	n.d.	n.d.
Acc.-Lat.	5.20 s	n.d.	n.d.	n.d.	n.d.
Acc.-Vert.	4.95 s	11.0 s	n.d.	n.d.	n.d.
Gyro-Roll	0.45 s	0.45 s	0.50 s	0.50 s	0.50 s
Gyro-Pitch	0.45 s	0.50 s	0.50 s	0.55 s	0.55 s
Gyro-Yaw	1.65 s	1.60 s	2.35 s	n.d.	n.d.
MLS-Azim.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Elev.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Range	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IAS	0.45 s	0.45 s	n.d.	n.d.	n.d.
IMU-Roll	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Pitch	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Yaw	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s

NOTE : Failure level injected is the same as in Table 3.4a
n.d. = not detected

sensors, each of the MLS and IMU sensor failures get detected instantaneously, for all the update rates. The exception here is the IAS sensor due to the low failure level injected (as explained in the previous section) as well as the direct correlation between the linear accelerometers (and their bias estimates) and IAS sensor.

Table 3.6: Effect of piecewise constant gains on detection time with failures injected at 145.40 s

SENSOR TYPE	DETECTION TIME WITH GAIN UPDATE FREQUENCY OF 20 Hz	4 Hz	2 Hz	1.33 Hz	1 Hz
Acc.-Long.	3.95 s	4.50 s	6.0 s	6.15 s	7.90 s
Acc.-Lat.	4.75 s	5.90 s	n.d.	n.d.	n.d.
Acc.-Vert.	3.85 s	4.50 s	n.d.	n.d.	n.d.
Gyro-Roll	0.35 s	0.35 s	0.35 s	0.35 s	0.35 s
Gyro-Pitch	0.50 s	0.50 s	0.45 s	0.45 s	0.45 s
Gyro-Yaw	1.45 s	2.0 s	2.05 s	1.45 s	2.30 s
MLS-Azim.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Elev.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Range	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IAS	0.40 s	0.40 s	0.40 s	0.40 s	0.40 s
IMU-Roll	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Pitch	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Yaw	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s

NOTE : Failure level injected is the same as in Table 3.4a
n.d. = not detected

Table 3.6 shows the results for a similar set of runs but with the sensor failures injected at 145.40 seconds into the flight. At this time, we see a general improvement in the accelerometers and IAS sensor failure detection performance at lower gain update frequencies. Roll and pitch rate gyro failure detection performance does not get affected; however, yaw rate gyro failure detection shows an improvement at the lower update

Table 3.7: Effect of piecewise constant gains on detection time with failures injected at 238.70 s

SENSOR TYPE	DETECTION TIME WITH GAIN UPDATE FREQUENCY OF 20 Hz	4 Hz	2 Hz	1.33 Hz	1 Hz
Acc.-Long.	3.75 s	3.85 s	5.90 s	8.30 s	n.d.
Acc.-Lat.	2.70 s	2.80 s	2.95 s	n.d.	n.d.
Acc.-Vert.	2.10 s	2.25 s	2.55 s	3.85 s	n.d.
Gyro-Roll	0.40 s	0.40 s	0.40 s	0.40 s	0.40 s
Gyro-Pitch	0.50 s	0.50 s	0.50 s	0.50 s	0.50 s
Gyro-Yaw	1.50 s	1.50 s	1.50 s	1.55 s	1.55 s
MLS-Azim.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Elev.	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
MLS-Range	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IAS	1.20 s	1.20 s	1.20 s	1.30 s	1.45 s
IMU-Roll	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Pitch	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s
IMU-Yaw	0.0 s	0.0 s	0.0 s	0.0 s	0.0 s

NOTE : Failure level injected is the same as in Table 3.4a
n.d. = not detected

rates. All measurement sensors continue to exhibit excellent detectability at every gain update frequency.

Finally, Table 3.7 presents the results with lower gain update rates when sensor failures are injected in the final flight segment at 238.70 seconds. Again, the accelerometer failure detection performance shows an improvement. This is caused by two factors; (a) the accelerometer bias estimates have more time to converge to steady-state before the failure occurs, and (b) from our detectability analysis, we know that the accelerometers generate higher incremental information in this last flight segment. The rate gyro sensors and all measurement sensors exhibit the same performance as in the previous test run.

Using the new detection test at the various gain update frequencies, the FINDS program runs take about 10 % more computational time than the same runs for the no-fail filter estimator with no detection test. Thus, for the different gain update frequencies of 20 Hz, 4 Hz, 2 Hz, 1.33 Hz and 1 Hz, the algorithm runs approximately 11, 3.5, 2.3, 1.7 and 1.4 times slower than real-time, respectively, on the host development computer.

In this chapter, we have presented an improved baseline detection performance for the bank of detectors implementation. A detectability analysis involving the incremental information generated by various sensor failures has also been presented. This analysis has been used to formulate a new detection strategy for the FINDS algorithm. We have presented the results of this new decision test for sensor failures occuring at various phases in the flight, and also for various no-fail filter gain update frequencies. At the low update frequencies, the FINDS algorithm executes at near real-time speed with no effect on the MLS and IMU sensor failure detection performance. However, IAS, rate gyro and accelerometer sensor failure detection performance gets degraded at lower gain update frequencies, especially below 4

Hz., because of the degradation in the input sensor bias estimation performance. In the case of gain update frequencies below 4 Hz, the input sensor bias failures are not detected until the very last segments of the flight as the bias estimates converge to their steady state values.

4. CONCLUSIONS AND RECOMMENDATIONS

In this report, we have presented the modifications made to the FINDS algorithm in order to improve its estimation and failure detection performance, reduce the program size so as to meet the candidate flight computer memory constraints, and increase its execution speed to allow real-time operation for flight test experiments.

The estimation performance has been improved by using a new wind dynamics model along with modifications to the filter design parameters. The major impact of the improved estimation performance is the resulting false alarm rate of zero during all the emulation runs with the flight data. The detection performance has been significantly improved by implementing tests-of-mean over various moving windows of the no-fail filter residuals. With this new detection strategy, low level MLS, IAS and IMU sensor failures can now be detected instantaneously. The low level accelerometer and rate gyro failures are detected within the minimum time allowed by the information generated in the sensor residuals based on the aircraft point mass equations of motion.

Using the dual configuration of the target flight computer with each side having 128 Kb of memory as our basis of reference, we have brought the program size of FINDS down from 340 Kb in double precision to an equivalent 115 Kb single precision implementation. This reduced size code has been ported onto one computer of the dual target flight computer configuration and the program operation has been verified.

On the host development computer computer which has a 294,000 Whetstones floating point performance in 32 bit single precision, the FINDS algorithm can now execute at 1.3 times slower than real-time using modifications which do not compromise on the estimation performance. By implementing a new decision strategy, we have also been able to perform failure detection tests (without an isolation test) at 1.4 times slower than real-time. Any further modifications to the FINDS algorithm to make it execute in real-time now depend upon the actual flight computer chosen for the experiment.

In this choice of a candidate computer for an eventual flight test of FINDS, we have the following two options :

A) If the dual configured target flight computer with its 255,000 Whetstones floating point performance is chosen for the flight computer, then we recommend the following parallel processing solution by partitioning the FINDS algorithm into two separate modules.

One computer would execute the translational dynamics filter consisting of the linear accelerometers as input sensors and MLS and IAS as measurement sensors. The aircraft position, velocity and horizontal winds would be the filter states along with the accelerometer bias estimates, thus resulting in a 8-state and 3-bias configuration.

The second computer would execute the rotational kinematics filter consisting of the rate gyros as input sensors and the IMU as the measurement sensors. This filter would estimate the aircraft attitudes and the rate gyro bias estimates, thus yielding a 3-state and 3-bias configuration.

Hence, the original FINDS algorithm of order 17 (11 states and 6 biases) using seven measurements (necessitating seventh order matrix operations) would be split up into two modules; translational dynamics filter of order 11 (8 states and 3 biases) using four measurements, and the rotational kinematics filter of order 6 (3 states and 3 biases) using three measurements. Since both modules can be run in parallel on the dual target flight computer configuration, this would effectively involve only fourth order matrix operations at every sample. On each computer, the new detection test would be implemented, testing the occurence of a failure in any sensor used by the no-fail filter on that side. Moreover, the isolation test after the detection of a sensor failure would involve only those sensors used by that particular filter. This split up of the FINDS algorithm would have no impact on either estimation or detection performance since the modifications made in the current study have resulted in decoupling the translational dynamics from the rotational kinematics. Using this option, we estimate that the FINDS algorithm with the baseline gain update frequency of 20 Hz would execute at approximately three to four times slower than real-time. Then, the use of a moderately slow gain update rate (4 Hz for instance) would bring the program execution down to real-time. If needed, a front-end microprocessor interface between the flight data acquisition bus and the target flight computer can be used to read in the flight data, and perform variable assignments and dropout tests, thus speeding up the algorithm execution further. This effort would involve the restructuring of the FINDS algorithm implementation from its current form.

B) As a second option, we propose the use of a flight computer which is about four times faster than the target flight computer used in this study, i.e. a computer with approximately a 1,000,000 Whetstones floating point performance in 32 bit precision. On such a computer, the FINDS algorithm with the new detection strategy would run approximately 2.5 times slower than real-time at the baseline gain update frequency of 20 Hz. In this case, we recommend a multi-rate implementation of the algorithm in which the no-fail filter's bias-free and bias computations are performed at different speeds. For instance, the execution of the bias-free filter at 1 Hz and the bias filter at 20 Hz would allow overall real-time execution without compromising either estimation or accelerometer failure detection performance. This effort would involve modifications to the code if the extra computational speed is obtained with an array processor, or a modest restructuring of the algorithm if the speed is due to a faster CPU.

Using either option, we recommend the following sequence of flight test experiments:

1) Analyze the aircraft state and sensor bias estimation performance of the no-fail filter (with no detection and isolation test) in flight under maneuver, turbulence and different steady-state wind conditions through various flight paths. Flight tests at different days would be required to ensure not only varying wind conditions but also different accelerometer and rate gyro biases.

2) Using the flight data collected in step 1, analyze (off-line) the false alarm performance of the FINDS algorithm under the various flight segments and aircraft maneuvers. At this stage, we also recommend performing a statistical sensor error analysis as done in [1], to ensure that the postulated sensor noise characteristics are correct. For instance, MLS range sensor may have a constant high bias, thus requiring the estimation of this parameter.

3) Analyze the failure detection performance of the FINDS algorithm performing flight test involving various steady-state winds and turbulence conditions and aircraft maneuvers during which failures are injected into the flight data. We recommend the following procedure in which dynamically correlated sensor group failures are studied separately:

-- MLS range, IAS and longitudinal accelerometer failures,

-- MLS azimuth and lateral accelerometer failures,

-- MLS elevation and vertical accelerometer failures,

-- IMU roll attitude and roll rate gyro failures,

-- IMU pitch attitude and pitch rate gyro failures,

-- IMU yaw attitude and yaw rate gyro failures.

During these experiments, after the detection of a particular failure, the isolation test can be performed only for the dynamically related sensor groups, thus minimizing computational resources.

REFERENCES

[1] Caglayan, A.K. and Godiwala, P.M., "Evaluation of a Fault Tolerant System for an Integrated Avionics Sensor Configuration with TSRV Flight Data", NASA CR - 172589, June 1985.

[2] Caglayan, A.K., Godiwala, P.M. and Morrell, F.R., "Performance Analysis of a Fault Inferring Nonlinear Detection System (FINDS) with Integrated Avionics Flight Data", Proceedings of AIAA Computers in Aerospace V Conference, Long Beach, CA, October 1985.

[3] 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.

[4] Curnow, H.J. and Nichmann, B.A., "A Proposed Benchmark for Hardware Evaluation: The Whetstone Program", The Computer Journal, No. 1, 1976.

[5] Germann, J., "Performance Comparison: Enhanced Supermicro and the VAX/780", UNIX/WORLD Journal, January 1984.

[6] Caglayan, A.K. and Lancraft, R.E., "A Fault Tolerant System for an Integrated Avionics Sensor Configuration", NASA CR - 3834, September 1984.

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

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

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

[10] Anderson, T.W., "An Introduction to Multivariate Statistical Analysis", John Wiley & Sons, New York, N.Y., 1958.

Standard Bibliographic Page

1. Report No.
NASA CR-178043

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A PRELIMINARY DESIGN FOR FLIGHT TESTING THE FINDS ALGORITHM

5. Report Date
March 1986

6. Performing Organization Code

7. Author(s)
A.K. Caglayan and P.M. Godiwala

8. Performing Organization Report No.
R8602

9. Performing Organization Name and Address
Charles River Analytics Inc.
55 Wheeler Street
Cambridge, MA 02138

10. Work Unit No.

11. Contract or Grant No.
NAS1-17719

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code
505-66-41-05

15. Supplementary Notes
Langley Technical Monitor: Frederick R. Morrell
Final Report

16. Abstract
This report presents a preliminary design for flight testing the FINDS (Fault Inferring Nonlinear Detection System) algorithm on a target flight computer. The FINDS software was ported onto the target flight computer by reducing the code size by 65%. Several modifications were made to the computational algorithms resulting in a near real-time execution speed. Finally, a new failure detection strategy was developed resulting in a significant improvement in the detection time performance. In particular, low level MLS, IMU and IAS sensor failures are detected instantaneously with the new detection strategy, while accelerometer and rate gyro failures are detected within the minimum time allowed by the information generated in the sensor residuals based on the point mass equations of motion. All of the results have been demonstrated by using five minutes of sensor flight data for the NASA ATOPS B-737 aircraft in a Microwave Landing System (MLS) environment.

17. Key Words (Suggested by Authors(s))
Fault tolerant systems, Sensor failure detection and isolation, Integrated avionics flight data, Flight computer constraints.

18. Distribution Statement
Unclassified - Unlimited

Subject Category 06

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	86	A05

For sale by the National Technical Information Service, Springfield, Virginia 22161