FINAL REPORT

OU/AEC 92-3TM00006/46+46A-1

Hybrid GPS/Loran-C

by

Frank van Graas, Ph.D.
Paul A. Kline, M.S.E.E.

Avionics Engineering Center
Ohio University
Athens, OH 45701

July 1992

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
RESEARCH AND SPECIAL PROGRAMS ADMINISTRATION
VOLPE NATIONAL TRANSPORTATION SYSTEMS CENTER
Kendall Square, Cambridge, MA 02142

Contract DTRS-57-87-C-00006
TTD-46

Deliverable Item #3
Task 5

EXECUTIVE SUMMARY

Fault detection and isolation (FDI) algorithms are required to achieve a sole-means navigation system using GPS and Loran-C. In order to provide a better understanding of the Fault Detection and Isolation algorithm, a simplified example is presented using three voltmeters to measure a single voltage. The two redundant measurements allow for the full characterization of the FDI algorithm.

Because navigation systems will depend on FDI for integrity assurance, the availability of a navigation system will depend on the availability of at least two redundant measurements. A Markov analysis, combined with a GPS coverage program, is presented to show that GPS by itself could meet supplemental availability requirements for fault detection, but not for isolation. An example of an integrated navigation system, GPS/Loran-C, is presented and actual flight data is used to demonstrate the FDI algorithm in the presence of signal failures. Computer simulations are performed to inject artificial failures into GPS and Loran-C signals, and the results of the FDI algorithm are displayed pictorially.

The following conclusions are based on the work presented in this report:

1. Sole means navigation will require a high availability of at least two redundant measurements for FDI. Therefore, GPS must be augmented by another system, such as Loran-C, before it has the potential to meet requirements for sole means navigation.

2. The probability of a false alarm should not be traded against the probability of a missed detection. Instead, both probabilities should be traded against the horizontal protection radius. This approach guarantees the integrity of the navigation solution at all time and space points. Furthermore, it facilitates changes in navigation accuracy requirements without the need to change the implementation of the algorithm.

3. The parity vector should be observed over a period of time in order to aid fault isolation. The "snapshot" batch estimator is still applicable, but rather than discarding the parity information for each measurement sample, the history of the parity vector should be stored and evaluated. The methods presented in this report constitute a novel approach to fault isolation which greatly enhances the effectiveness of FDI.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CDI	Course Deviation Indicator
DME	Distance Measuring Equipment
ECEF	Earth-Centered, Earth-Fixed (Coordinate System)
FAA	Federal Aviation Administration
FDI	Fault Detection and Isolation
GIC	GPS Integrity Channel
GLONASS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
INS	Inertial Navigation System
Loran-C	Long Range Navigation-C
MOPS	Minimum Operational Performance Standards
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
NAS	National Airspace System
pdf	Probability Density Function
PR	Pseudorange
RAIM	Receiver Autonomous Integrity Monitoring
rms	Root-Mean-Squared
RTCA	Radio Technical Commission for Aeronautics
SA	Selective Availability
TACAN	Tactical Air Navigation
TD	Time Difference
VOR	Very High Frequency Omnidirectional Range

LIST OF SYMBOLS

$\underline{b}$	Measurement Bias Vector
COMB(n:m)	Number of Combinations of n out of m
d_k	Detection Statistic
$\underline{e}$	Measurement Noise Vector
G	Givens Rotation Matrix
H	Data Matrix
P	State Transition Probability Matrix
$\underline{p}$	Parity Vector
P_A	Probability of Exceeding R_p
P_F	Probability of a Failure
P_{FA}	Probability of a False Alarm
P_m	State Probability from the Markov Model
P_{MD}	Probability of a Missed Detection
P_R	Probability of a Repair
P_t	Probability of an Outage at a Particular Time
R-error	Range Measurement Error
R_p	Protection Radius (Allowable Horizontal Position Error)
$\underline{s}$	State Vector
T	Time Step
T_D	Detection Threshold
T_{inc}	Length of an Outage
T_m	Mission Time
$\underline{y}$	Measurement Vector
$\underline{\beta}$	User State Vector
λ	Failure rate
μ	Repair rate
μ_M	Minimum Required Bias in Parity Space for Detection
σ	Standard Deviation of the Measurement Noise

1. INTRODUCTION

Recent advances in electronic navigation systems such as the introduction of the satellite-based Global Positioning System (GPS) and the improved coverage of the terrestrial Loran-C system have allowed the development of integrated systems which could be certified for sole means air navigation [1]. A integrated (or hybrid) navigation system would combine the information obtained from such systems as Omega, Loran-C, the Very High Frequency Omnidirectional Range system (VOR), the Distance Measuring Equipment system (DME), the Tactical Air Navigation system (TACAN), the Inertial Navigation System (INS), GPS, and the Soviet Global Navigation Satellite System (GLONASS). GPS is an especially popular choice for a hybrid system because of the high accuracy it delivers and because it does not meet the requirements for sole means navigation as a stand-alone system [2]. Some hybrid systems currently being researched include GPS/INS, GPS/Loran-C, and GPS/GLONASS.

Motivation for this report stems from the ongoing Federal Aviation Administration (FAA) efforts to define the performance standards for sole means navigation using GPS with appropriate augmentations. As part of this effort, minimum operational performance standards (MOPS) are being developed by RTCA Special Committee 159 [3]. This is a unique task because the integrity provided by GPS itself is insufficient, and there currently exists no external integrity monitor. Therefore, the receiver must perform all integrity checking using only the available measurement data. The implementation of the integrity function in the receiver is referred to as Fault Detection and Isolation (FDI). FDI has already been successfully applied to redundant inertial navigation systems [4,5,6].

The International Civil Aviation Organization (ICAO) recently held their 10th Air Navigation Conference [7]. It was recognized that satellites will be used extensively in future air navigation. One recommendation approved at the conference is that a decision should be made as to how satellite system integrity should be monitored. The most promising method discussed was the use of redundant measurements to perform Fault Detection and Isolation.

The objective is to detect and isolate sensor malfunctions which cause unacceptable position errors using only inconsistency in the measurement data. For sole means navigation, two estimators should be used in parallel to achieve FDI:

1. A recursive estimator, typically a Kalman filter, which uses the history of the measurement data to assess the reasonableness of new measurement data. This estimator does not require redundant measurements.

2. A least squares batch estimator which does not rely on the measurement history, but it requires at least one redundant measurement.

The recursive estimator is used to detect and isolate rapidly growing measurement errors by inspecting the estimator residuals, which are the differences between the actual measurements and the predicted measurements based on the history of the measurements. If for instance a measurement residual would be outside the residual interval $[-6.1\sigma, 6.1\sigma]$, where σ is the standard deviation of normally distributed measurement noise, then the probability of this event would be less than 10^{-9} for a static user. This event would be extremely unlikely and the integrity alarm would be raised. In the presence of user dynamics, the residual interval would be increased consistent with the user dynamics. Note that isolation of the faulty measurement is achieved as well.

A least squares batch estimator is used to detect slowly growing measurement errors which go undetected by the recursive estimator. The use of the least squares estimator for this purpose is justified, because:

> Independent of the type of estimator used, the position bias error caused by measurement bias errors will always converge to the position bias error of the least squares estimator. Generally, the time constant of convergence is short compared to the slow error growth of a difficult to detect measurement error.

All information about the inconsistencies in the measurement data is contained in the residuals of the least squares estimator. These residuals are the differences between the actual pseudorange measurements and those predicted based on the least squares position solution and the known transmitter coordinates.

An algorithm will be introduced that provides integrity using a batch estimator in the presence of redundant measurements. The classical trade-off between false alarms and missed detections is avoided. To illustrate the algorithm, a simplified example is presented using three voltmeters to measure a single voltage. The availability of fault detection for GPS is then addressed to show that GPS alone cannot meet sole means navigation requirements. The FDI algorithm is then applied to an integrated navigation system combining GPS and Loran-C. Using real flight data, artificial failures are injected to study the performance of the FDI algorithm. The algorithm performance is independent of measurement geometry and the output of the algorithm also provides a confidence level for the navigation solution.

2. LEAST SQUARES FAULT DETECTION

2.1 False Alarms and Missed Detections

A problem fundamental to fault detection is that it always involves a degree of uncertainty. The concept of parity space, which will be developed in more mathematical detail in Section 2.3, is important because the fault detection and isolation occurs in this domain. The position estimate lies in estimation space, which is not the same as parity space. Furthermore, the measurements lie in measurement space. Therefore, it is only possible to "guess" that an error exists by examining a snapshot of parity space when the measurements are made. It is therefore possible for a faulty measurement to go undetected and, conversely, for a fault to be declared when none actually exists. To gain a better understanding of the above issues, it is necessary to discuss least squares fault detection in detail.

One of the main input parameters to the fault detection algorithm is the alarm threshold, defined as the allowable horizontal radial error in the calculated user position. The ideal case would be to raise an alarm only when this limit is exceeded and never raise an alarm otherwise. However, since fault detection is performed in the presence of measurement noise and in a domain other than the estimation space, it is only possible to detect a fault with a certain probability. Therefore, two undesirable events are possible - a false alarm and a missed detection. Two major parameters used in characterizing the performance of the fault detection algorithm are the probability of a false alarm (P_{FA}) and the probability of a missed detection (P_{MD}). As one might assume, it is desirable for these probabilities to be very small, consistent with airspace requirements.

It is important to recognize the fact that a measurement error does not always cause a significant error in the position estimate. Not all errors are highly "visible" in estimation space due to, for instance, the geometry of the satellites. Yet in parity space the error can cause a fault to be falsely declared. Similarly, a measurement error that is highly visible in estimation space can be relatively transparent in parity space. This can cause an unacceptable position error even though no errors are detected in parity space. The consequences of this are quite serious because of obvious safety issues. Traditionally, there has been a trade-off between false alarms and missed detections - reducing the number of false alarms increases the risk of missed detections and vice versa. Due to the seriousness of missed detections and because numerous false alarms reduce user confidence in the system, the approach taken here is <u>not</u> to perform a trade-off between false alarms and missed detections. Instead, both probabilities will be traded against the protection radius, which is the horizontal radial position error which can be protected by the algorithm given the probabilities of false alarm and missed detection. Note that the protection radius is always larger than the operational accuracy requirement. For instance, the accuracy requirement for a nonprecision approach could be 100 meters (probability of exceeding less than 0.05), while the protection radius would be 555 meters (probability of exceeding less than 10^{-8}).

2.2 The Detection Statistic

A fault is declared when a detection statistic exceeds a certain detection threshold. The detection threshold depends on the measurement noise and the desired false alarm rate.

As is intuitively expected, a smaller desired false alarm rate results in a larger detection threshold. Also, measurement noise with a large standard deviation causes more alarms to be triggered. The detection statistic is determined from parity relations.

Figure 1 illustrates two likely scenarios that would arise in the event of a slow clock drift. These two cases illustrate what would be observed if a small-sloped ramp error (such as a clock drift in a GPS satellite) slowly degraded the operational accuracy of the system. In case I, the detection threshold is breached before the alarm threshold is crossed, causing a false alarm. As the position error grows, the false alarm becomes a correct alarm. In case II, the alarm threshold is exceeded before the detection threshold is exceeded, resulting in a missed detection. Eventually the detection threshold is crossed, causing a flag to be raised for a correct alarm. The normal operating state includes all circumstances where neither threshold is exceeded. For sole means systems, the probability of being in this state should be close to one.

2.3 Parity Space and Estimation Space

Consider a scenario where two measurements y_1 and y_2 from two different voltmeters are made of the same physical quantity (e.g. a battery voltage). Following Potter and Suman [8], if this pair of measurements contains no errors, then:

$$y_1 - y_2 = 0 \tag{2.1}$$

Because this equation shows that y_1 and y_2 are identical, it is called a parity equation. Now if measurement errors e_1 and e_2 are included, the equation becomes:

$$(y_1 - e_1) - (y_2 - e_2) = 0 \tag{2.2}$$

which can be written in the form:

$$y_1 - y_2 = e_1 - e_2 = p \tag{2.3}$$

The value p is a measure of the inconsistency between the two measurements. A value of zero for p implies that e_1 and e_2 are the same, not that e_1 and e_2 are each necessarily zero. In fact, many combinations of e_1 and e_2 result in a p of zero. Now add a third measurement y_3, which results in three possible parity equations:

$$\begin{aligned} y_1 - y_2 &= e_1 - e_2 = p_1 \\ y_2 - y_3 &= e_2 - e_3 = p_2 \\ y_3 - y_1 &= e_3 - e_1 = p_3 \end{aligned} \tag{2.4}$$

Note that the third equation can be formed as a linear combination of the first two

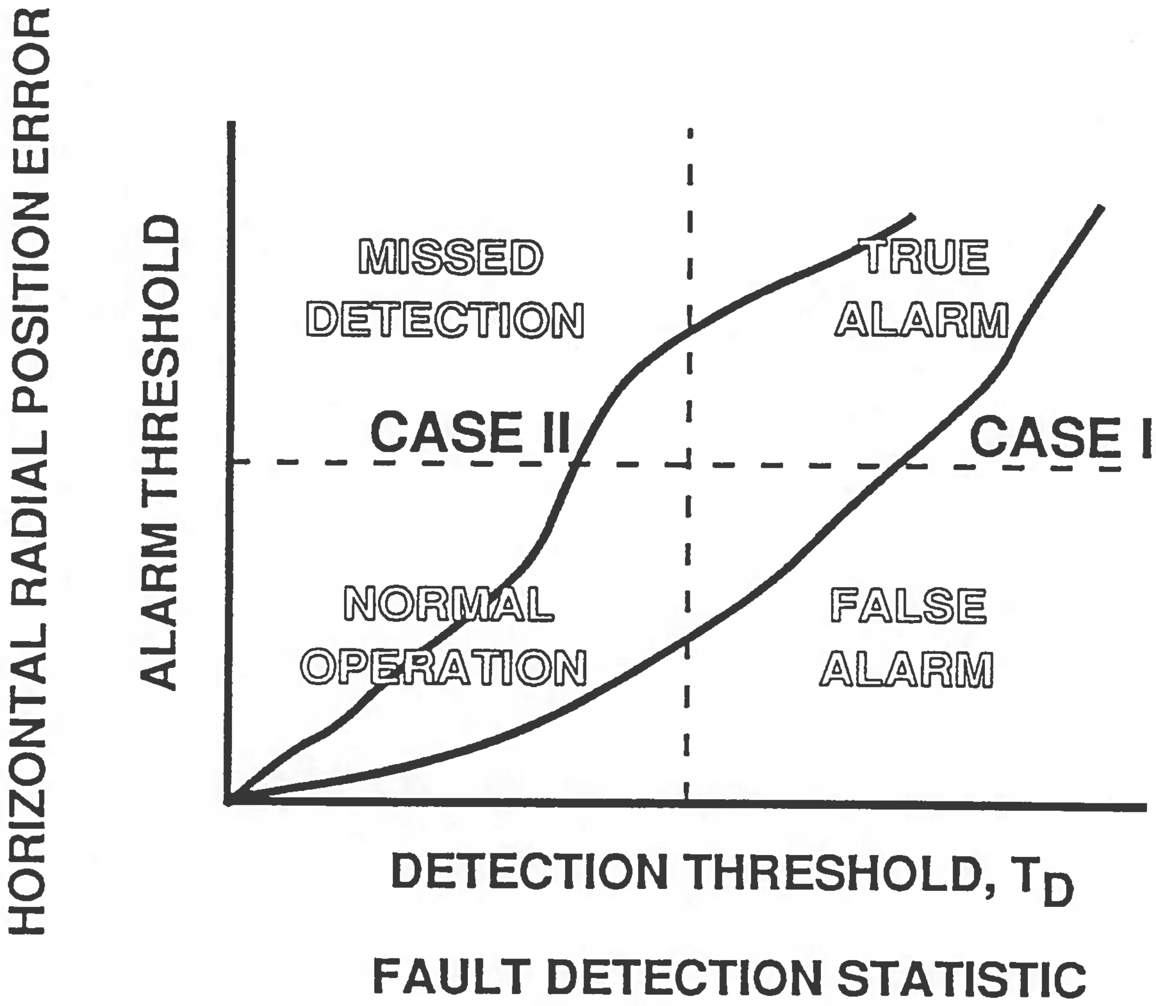

HORIZONTAL RADIAL POSITION ERROR
ALARM THRESHOLD
MISSED DETECTION
TRUE ALARM
CASE II
CASE I
NORMAL OPERATION
FALSE ALARM
DETECTION THRESHOLD, T_D
FAULT DETECTION STATISTIC

Figure 1 Example of the Impact of Slowly Growing Measurement Errors on Fault Detection.

equations. Thus, all the parity information is contained in two equations. Similarly, three independent equations exist when there are three redundant measurements, and so forth. The number of independent parity equations determines the number of components in the parity vector, which is the same as the number of dimensions in parity space.

Consider a scenario where one redundant measurement is available. In this case, the parity vector becomes a scalar. The detection statistic is given by p, which is assumed to be normally distributed. Figure 2 (top) shows the probability density function (pdf) of p when no bias exists in any of the voltmeter measurements. Note that the mean is zero because the pdf has not been shifted by a measurement bias. A fault is declared when the absolute value of p exceeds the detection threshold (T_D). Note that integrating the tails of the pdf outside $\pm T_D$ yields the probability of a false alarm in parity space.

Figure 2 (bottom) illustrates the existence of a bias in one of the measurements. In this example, the pdf of the detection statistic p is shifted over by a mean value greater than T_D. Since there is now a fault, the area under the curve within the limits of $\pm T_D$ is the probability of a missed detection in parity space.

If an alarm is raised in parity space, it can either be a correct alarm or a false alarm in estimation space. If no alarm is raised in parity space, it can either be normal operation or a missed detection in estimation space. Note that the radial position error lies in estimation space, while the detection statistic lies in parity space, see Figure 1. In the case of classical hypothesis theory, a false alarm is considered to be any fault detected during normal operation. Likewise, a missed detection is simply considered to be any undetected fault. While these ideas are basically true, it is important to understand how they relate to estimation space and parity space. Because the probabilities of false alarm and missed detection are specified in parity space, the corresponding probabilities of false alarm and missed detection in estimation space are slightly different.

2.4 Fault Detection Algorithm

A least squares approach can be used for fault detection. The linearized relationship between the measurements and the user state is given by:

$$\underline{y} = H\,\underline{\beta} \qquad (2.5)$$

where: $\underline{y}$ = measurement vector
$\underline{\beta}$ = user state vector
H = data matrix

The dimension of H is n-by-m, where n is the number of measurements and m is the dimension of the user state vector. The state vector $\underline{\beta}$ consists of the user position

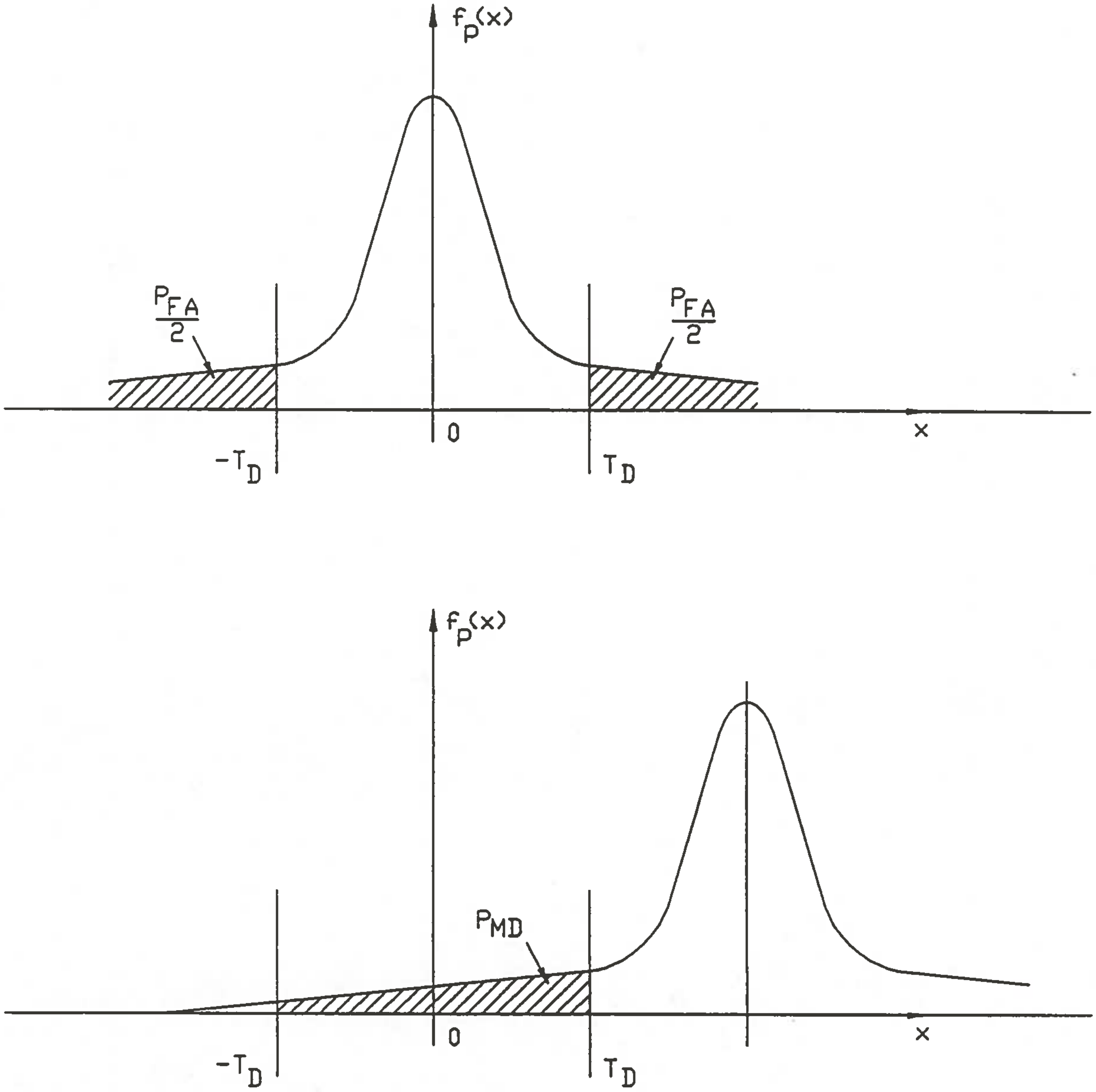


Figure 2 Probability Density Functions for the Detection Statistic p: No Measurement Bias (top) and with Measurement Bias (bottom).

coordinates and other navigation states such as clock offset with respect to, for instance, GPS time, as required by the navigation solution.

Three cases exist:

1) $n < m$: Underdetermined system
2) $n = m$: Exactly determined system
3) $n > m$: Overdetermined system

Algorithms for managing the redundant measurements in case 3, an overdetermined system, form the basis of fault detection. In the presence of redundant signals, a parity equation can be derived from Equation 2.5. The objective is to factorize the data matrix, H, such that the least squares solution and the parity vector can be obtained separately. Parity space is orthogonal to estimation space; therefore, parity space is the left nullspace of the H matrix. Parity space can be obtained by zeroing the last n-m rows of H. This requires that H be factorized into two matrices. One matrix has zeros on the last n-m rows, while there are no particular requirements for the second matrix. However, it is preferable for the second matrix to be orthonormal to simplify further calculations. One such factorization is the QR factorization:

$$H = QR \tag{2.6}$$

This factors H into an orthonormal matrix Q ($Q^TQ = I$) and an upper triangular matrix R [9]. R contains (n-m) rows of zeros, reflecting that H includes data from redundant measurements. Substituting Equation 2.6 for H in Equation 2.5 yields:

$$\begin{aligned} \underline{y} &= QR\underline{\beta} \\ Q^T\underline{y} &= Q^TQR\underline{\beta} \\ Q^T\underline{y} &= R\underline{\beta} \end{aligned} \tag{2.7}$$

Let R be partitioned into a m-by-m upper triangular matrix U and (n-m) rows of zeros, denoted by 0. Partition Q^T conformably into Q_β and Q_p:

$$\begin{pmatrix} Q_\beta \\ --- \\ Q_p \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} U \\ --- \\ 0 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix} \tag{2.8}$$

The least squares solution is given by:

$$\underline{\beta} = U^{-1}Q_\beta\underline{y} \tag{2.9}$$

Due to the way R is partitioned, U is nonsingular. Note that Equation 2.9 is identical to the least squares solution using the generalized inverse of the H matrix. By substitution:

$$(H^TH)^{-1}H^T \qquad \text{Substitute } H = QR$$

$$= [(QR)^TQR]^{-1}(QR)^T$$

$$= [R^TQ^TQR]^{-1}R^TQ^T$$

$$= [R^TR]^{-1}R^TQ^T \qquad \text{Substitute } Q^T = \begin{pmatrix} Q_\beta \\ --- \\ Q_p \end{pmatrix}$$

$$\text{and } R = \begin{pmatrix} U \\ --- \\ 0 \end{pmatrix}$$

$$= \left[(U^T \;\vdots\; 0)\begin{pmatrix} U \\ --- \\ 0 \end{pmatrix} \right]^{-1} (U^T \;\vdots\; 0)\begin{pmatrix} Q_\beta \\ --- \\ Q_p \end{pmatrix}$$

$$= (U^TU)^{-1}(U^T \;\vdots\; 0)\begin{pmatrix} Q_\beta \\ --- \\ Q_p \end{pmatrix}$$

$$= U^{-1}U^{-T}U^TQ_\beta$$

$$= U^{-1}Q_\beta \qquad \therefore\ (H^TH)^{-1}H^T = U^{-1}Q_\beta$$

In the absence of measurement errors, the parity equation is formed from the lower partition of Equation 2.8:

$$Q_p\underline{y} = 0 \tag{2.10}$$

Since $\underline{y}$ contains measurement errors such as noise ($\underline{e}$) and measurement biases ($\underline{b}$), a parity vector ($\underline{p}$) can be defined as (replace $\underline{y}$ by $\underline{y}$ - $\underline{e}$ - $\underline{b}$):

$$\begin{aligned} \underline{p} &= Q_p\underline{y} - Q_p\underline{e} - Q_p\underline{b} \\ \underline{p} &= -Q_p\underline{e} - Q_p\underline{b} \end{aligned} \tag{2.11}$$

Although the measurement noise and bias errors are not known, their components in parity space are given by Equation 2.11. The parity vector can be used as a detection function for declaring faults.

It is now possible to determine the statistics of the parity vector. The expected value $E(\underline{p})$ is given by (assuming that the measurement noise is Gaussian with zero mean):

$$E(\underline{p}) = -Q_p\underline{b} \tag{2.12}$$

Assuming that the measurement noise is also uncorrelated and that each is distributed with the same variance, then the covariance matrix for $\underline{p}$ is:

$$\mathrm{Cov}(\underline{p}) = \sigma^2 I \tag{2.13}$$

The result is a parity vector with normally distributed elements:

$$f_p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)^2} \tag{2.14}$$

This allows us to choose P_{FA} and P_{MD}, thus fixing the detection threshold T_D and the minimum required bias in parity space (μ_M):

$$P_{FA} = P(|p| > T_D) = \mathrm{erfc}\left(\frac{T_D}{\sigma\sqrt{2}}\right) \tag{2.15}$$

$$\text{where } \mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-\lambda^2} d\lambda \tag{2.16}$$

$$P_{MD} = P(|p| \leq T_D) = \frac{1}{2}\mathrm{erfc}\left(\frac{\mu - T_D}{\sigma\sqrt{2}}\right) \tag{2.17}$$

$$\mu_M = T_D + \sigma\sqrt{2}\,\mathrm{erfc}^{-1}(2P_{MD}) \tag{2.18}$$

Equation 2.17 shows that a bias must exceed the measurement noise for reliable detection. A minimum bias is required for P_{MD} to be satisfied. Good measurement geometry makes it easier to detect a bias. The matrix Q_p is used to calculate the minimum detectable bias. A bias is evaluated for each measurement by rotating parity space such that the measurement axis lies along the "X-axis" in parity space [10]. This rotation does not change the statistics of parity space, although a detection statistic must be calculated for each measurement. For example, if 7 GPS satellites are being used, then 7 detection statistics must be calculated.

2.5 Example with Three Voltmeters

It is helpful to consider a simplified example so that a better understanding of the algorithm may be gained. Consider three noisy voltmeters measuring a d.c. voltage (e.g. a battery). This scenario allows us to explore fault detection and isolation with the fewest number of measurements.

2.5.1 Three Voltmeters--Fault Detection

Since each voltmeter is weighted equally, the data matrix is given by:

$$H = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

The QR factorization on H could result in:

$$Q = \begin{pmatrix} -0.5774 & -0.5774 & -0.5774 \\ -0.5774 & 0.7887 & -0.2113 \\ -0.5774 & -0.2113 & 0.7887 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.7321 \\ 0 \\ 0 \end{pmatrix}$$

Note that Q and R are not unique and, in fact, depend on the QR algorithm that is used. This example is helpful since parity space is two-dimensional and it is easy to demonstrate the fault detection algorithm graphically. In order to determine the measurement axes in parity space, Equation 2.8 is applied with the appropriate partitions. Note that the state vector $\underline{\beta}$ reduces to a scalar because we are only measuring a single physical quantity, a voltage:

$$\left(\begin{array}{ccc} -0.5774 & -0.5774 & -0.5774 \\ \hline -0.5774 & 0.7887 & -0.5774 \\ -0.5774 & -0.2113 & 0.7887 \end{array}\right)\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \left(\begin{array}{c} -1.7321 \\ \hline 0 \\ 0 \end{array}\right)\beta_1$$

According to Equation 2.9, the least squares solution is simply the average of the three measurements:

$$\beta_1 = \frac{1}{3}(y_1 + y_2 + y_3)$$

The lower partition yields the parity equation:

$$\underline{0} = \begin{pmatrix} -0.5774 & 0.7887 & -0.2113 \\ -0.5774 & -0.2113 & 0.7887 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\underline{0} = Q_p\underline{y}$$

Where the rows of Q_p are orthogonal, but the columns are not. If $\underline{y}$ contains measurement noise ($\underline{e}$) or bias errors ($\underline{b}$), then $\underline{y}$ is replaced by $\underline{y}$-$\underline{e}$-$\underline{b}$. Thus, a parity vector can be defined as:

$$\underline{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = Q_p(-\underline{e} - \underline{b})$$

Parity space is spanned by the column vectors of Q_p; an axis associated with each voltmeter is described in parity space by the column vectors of Q_p. These axes are plotted in Figure 3. An error in measurement i will lie on axis i in parity space. Again, it should be noted that the parity vector is not a direct measure of the actual error. Many error combinations can yield the same parity vector [11].

Let the measurement noise have a standard deviation of $\sigma = 0.1$ volts for each of the three voltmeters. Choosing P_{FA} to be 0.1 and using Equation 2.15, the detection threshold T_D is found:

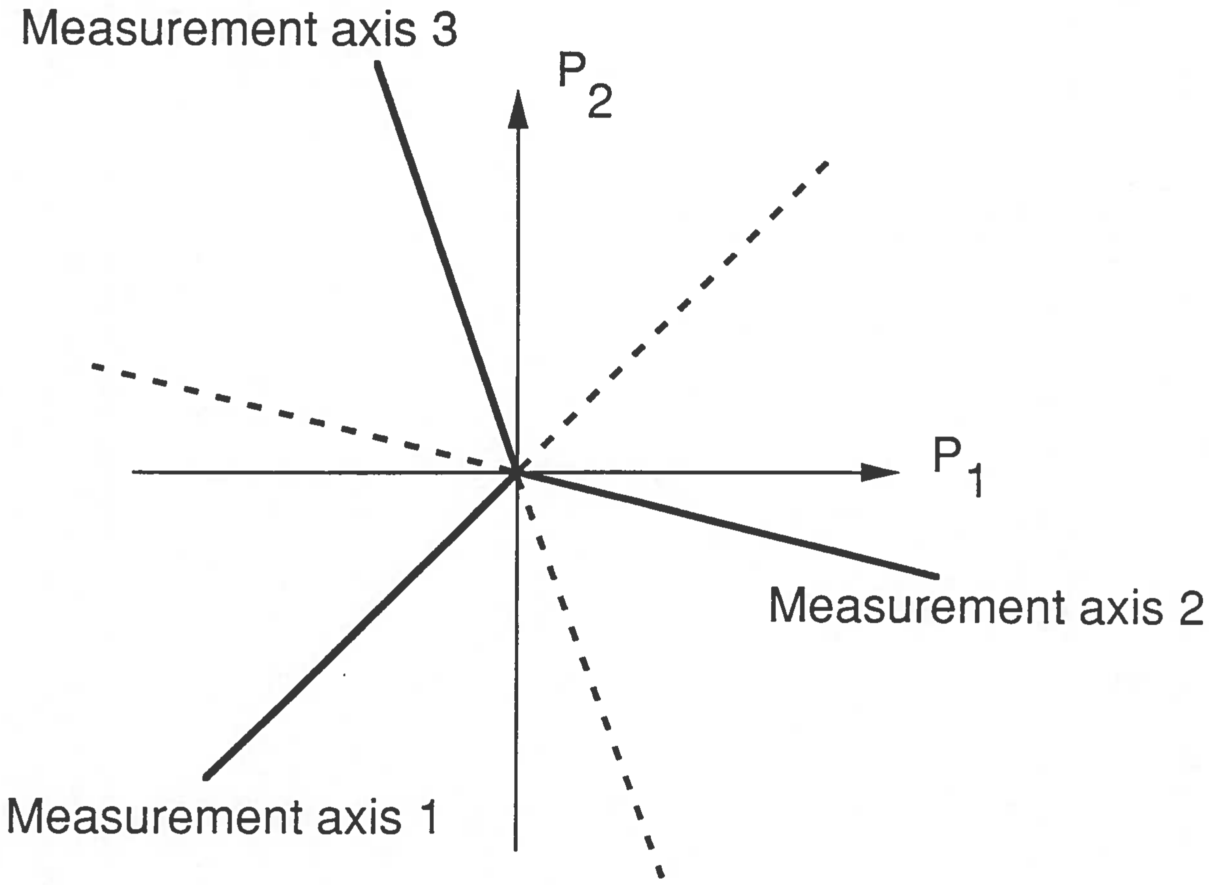


Figure 3 Measurement Axes in Parity Space

$$T_D = \sigma\sqrt{2}\,\mathrm{erfc}^{-1}\left(\frac{P_{FA}}{3}\right) = 0.213 \text{ volts}$$

Also, the minimum bias in parity space for detection, μ_M, can be calculated (choose P_{MD} to be 0.01):

$$\mu_M = T_D + \sigma\sqrt{2}\,\mathrm{erfc}^{-1}(2P_{MD}) = 0.446 \text{ volts}$$

Thus, the minimum bias, b, in one of the voltmeter measurements must be:

$$\mu_M = (\text{norm of a column of } Q_p)b$$

$$\mu_M = 0.816b$$

$$b = 1.224\mu_M = 0.546 \text{ volts}$$

Using P_{MD} and b, the protection radius (R_p) can be set. There are two errors to protect against--measurement noise and bias errors. Therefore, R_p is the sum of R_{noise} and R_{bias}:

$$R_{noise} = \frac{\sigma}{\sqrt{3}}\sqrt{2}\,\mathrm{erfc}^{-1}(0.01) = 0.149 \text{ volts}$$

$$R_{bias} = \frac{1}{3}(0.546) = 0.182 \text{ volts}$$

$$R_p = R_{noise} + R_{bias} = 0.331 \text{ volts}$$

This tells us that if both kinds of errors are present, then there is a 0.99 probability of detecting a voltage error of 0.331 volts or more. Without bias errors, P_{FA} will be 0.1 and P_{MD} will not exceed 0.01. If there are bias errors, the overall alarm rate will increase (both false and true alarms). However, P_{MD} will never be higher than 0.01.

2.5.2 Three Voltmeters--Fault Isolation

Three detection statistics are formed (one for each measurement), and an alarm is raised if any $|d_k| > T_D$ ($k = 1, 2, 3$). Once alarm status has been reached, the next step is to attempt isolation. The fault detection algorithm is applied to three subsets created by leaving out one measurement at a time. By omitting the failed measurement, the detection function for that subset should lie within T_D. By omitting a healthy measurement, the detection function for either of those subsets should exceed T_D. Table 1 shows the various isolation scenarios. Positive isolation is achieved when two subsets cause an alarm and one does not. Only 3 of 8 scenarios provide conclusive fault

Scenario	A&B	A&C	B&C	OUTCOME
1	0	0	0	?
2	0	0	1	?
3	0	1	0	?
4	1	0	0	?
5	1	1	0	A is BAD
6	1	0	1	B is BAD
7	0	1	1	C is BAD
8	1	1	1	?

0 = no alarm
1 = alarm
? = inconclusive

Table 1 Isolation Scenarios for the Three Voltmeter Example.

isolation. For example, if voltmeter A is bad, then the two subsets containing that measurement cause an alarm. The subset consisting of voltmeters B & C does not raise an alarm. Thus, voltmeter A is isolated as the faulty one. The inconclusive scenarios would be caused by, for instance, a false alarm. When there are no errors in the measurements, the results of the isolation algorithm are likely to be inconclusive.

Figure 4 is a pictorial representation of the isolation problem for the 3-voltmeter example. The parity vector pointing to point I in parity space might give inconclusive results; however, point I is more likely associated with measurement 2 than with measurements 1 or 3. Point J could belong to 1 or 2; however, since J is far away from both 1 and 2, this situation is unlikely. For point J to occur, there would have to be a large amount of measurement noise. In general, the detection threshold is large enough such that the probability of being far removed from a measurement axis is very small.

Figure 5 shows the fault detection and isolation state diagram, not only for the three-voltmeter example, but also for integrated navigation systems in general. From the diagram, it can be seen that the probability of a false alarm should be very small, because it results in either removing a healthy measurement or in system unavailability. The latter is very undesirable, especially for a sole means navigation system. Furthermore, the probability of a wrong isolation should also be minimized. It is obviously a serious condition when a fault is declared (correctly) but a healthy measurement is discarded.

Note that a missed detection occurs when an unacceptable error in measurement space is not flagged in parity space. The "can't detect" link describes a situation in which not enough measurements are available to run the fault detection algorithm, or if the calculated protection radius is greater than the required alarm threshold. Even with enough measurements to calculate a solution, without redundancy, the system must be declared unavailable if it depends on FDI for integrity assurance. One goal of current research is to define the transitional probabilities of the state diagram given the required state probabilities for sole means navigation systems. It is important to note, though, that this tree does not represent the whole story. For instance, once a faulty measurement has been removed from the solution, how should it be maintained? Should it be checked periodically to see if it is back within tolerance or should it be left out for the rest of the mission? These questions require further study so that an effective management scheme is implemented for governing the system in the presence of FDI data.

2.5.3 Three Voltmeters--Simulation Results

Simulations were run by taking 1000 samples of measurement noise (characterized by a σ of 0.1) for each of the voltmeters. As was previously mentioned, the thresholds were designed for a P_{FA} of 0.1 and a P_{MD} of 0.01. By applying the least squares solution given

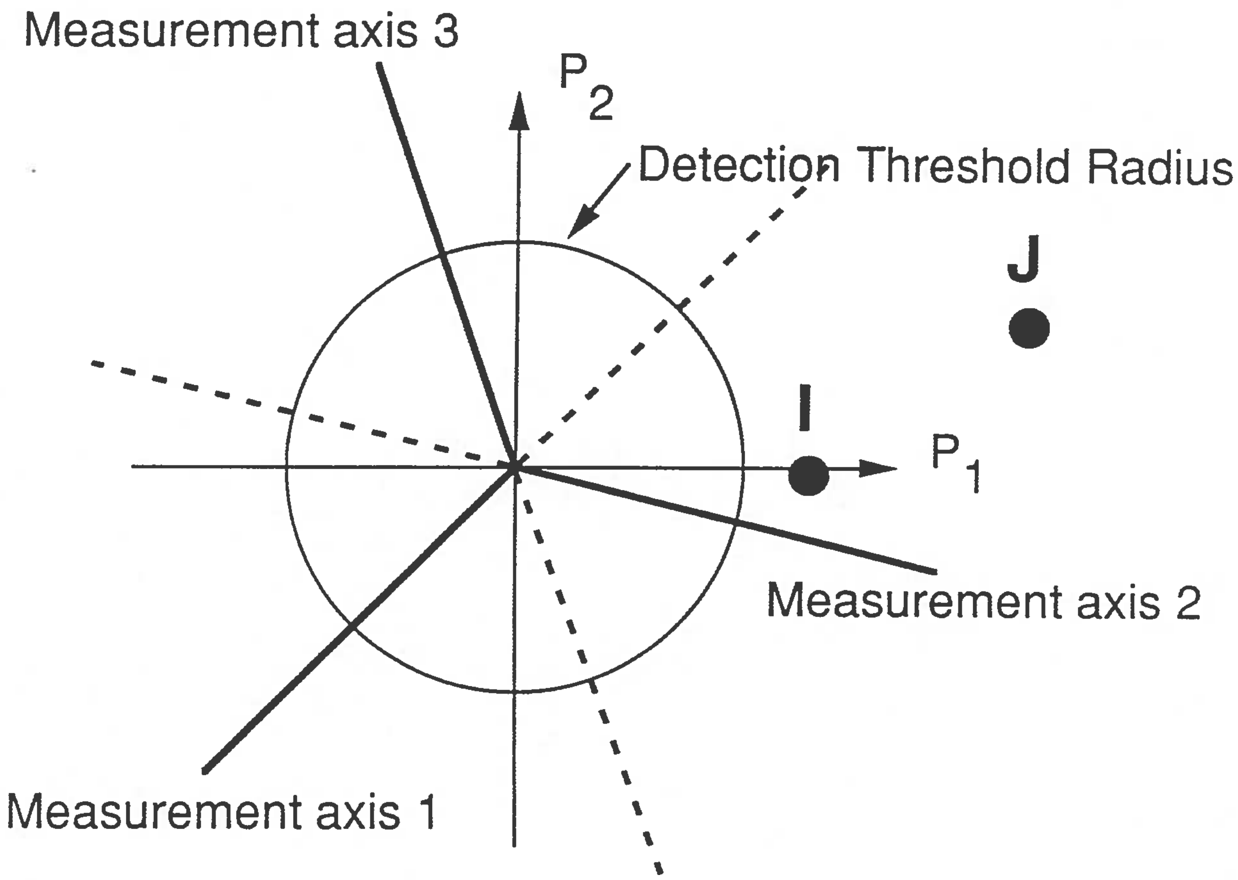


Figure 4 Example of Isolation Problem for Three Voltmeters.

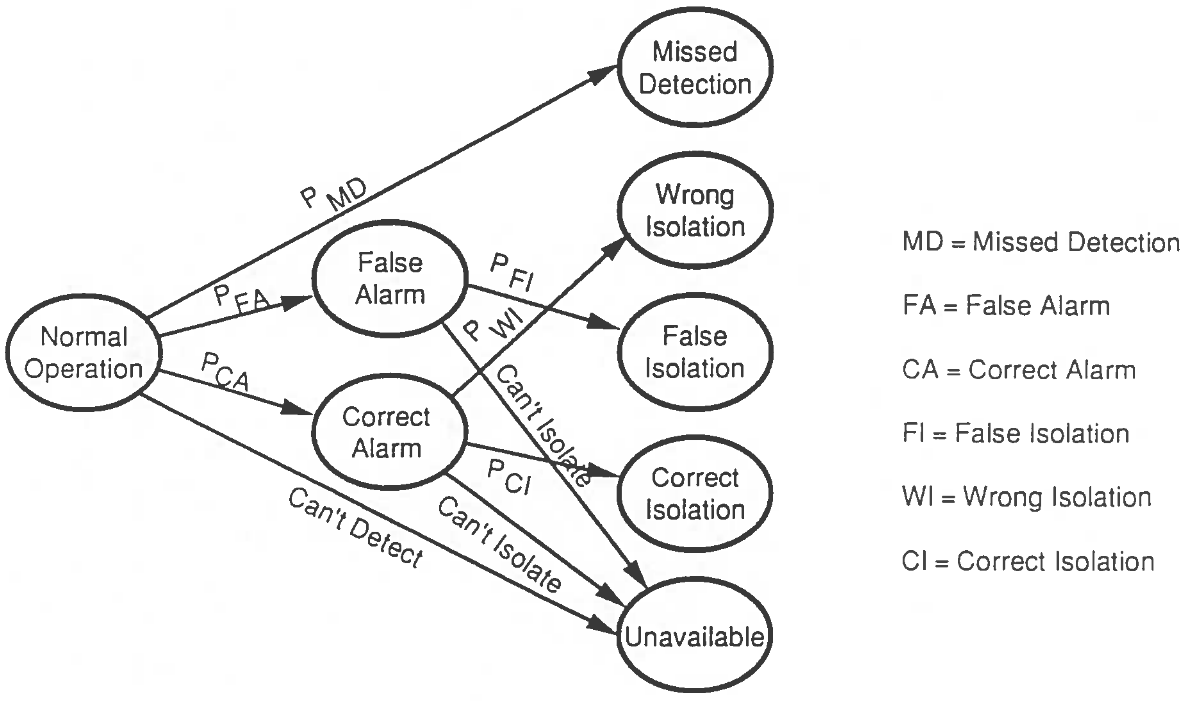


Figure 5 Fault Detection and Isolation State Diagram.

by Equation 2.9, the true voltage error can be found:

$$\text{True Voltage Error} = U^{-1}Q_{\beta}\underline{e} \tag{2.19}$$

Figure 6 is a plot of the true voltage error for each sample. The solid line is the protection radius for no bias ($R_p = R_{noise}$). The 12 points outside R_p represent actual error states, 11 of which were not detected resulting in a sample P_{MD} of 0.011. Figure 7 shows how the detection algorithm handled the various voltage errors. All cases which are termed "normal operation" have been removed. This illustrates the missed detections and the false alarms, as well as one true alarm. To see how the detection algorithm works, though, a view of parity space for these cases is helpful. Figure 8 shows all 1000 measurement points in parity space. Note that they are clustered around the origin because there is no bias in any of the voltmeters. The "cloud" effect is due entirely to the measurement noise. Figure 9 shows the missed detections and false alarms in parity space (normal operation cases have been removed). The detection threshold is a circle about the origin which can be visualized by examining the plot. The false alarms and the true alarm lie outside this circle, while the missed detections lie inside the circle. Figure 10 shows the outcome of the isolation algorithm for those cases that caused an alarm (either false or true). Some cases resulted in false isolation, while others were inconclusive. Isolation in parity space is shown in Figure 11 for these points.

Next, consider a random bias in voltmeter 2 which is characterized by a uniform distribution between -2.5 and 2.5 volts. This means that, in addition to the noise in each measurement, voltmeter 2 will have a bias somewhere between -2.5 and 2.5 volts. This will provide a better understanding of FDI in parity space. The protection radius will be extended now to protect against a bias ($R_p = R_{noise} + R_{bias}$). All other parameters remain the same ($\sigma = 0.1$ volts, $P_{FA} = 0.1$, $P_{MD} = 0.01$). The true voltage error is shown for all 1000 samples in Figure 12. The errors are much larger now due to the random bias. Figure 13 shows how the detection algorithm performed. All points beyond R_p were detected -- thus, there were no missed detections and P_{MD} was less than 0.01. There are still some false alarms, however. All data points as seen in parity space are shown in Figure 14. The cloud caused by the measurement noise has now been stretched along the axis for voltmeter 2 because of the random bias. The detection of faults in parity space is shown in Figure 15 with the normal operation points removed. There is a transition as a function of the length of the parity vector from false alarms to true alarms. This transition zone is necessary to allow for measurement noise. In the absence of measurement noise, the number of false alarms would be reduced to zero, and the transition zone would not exist. The results for the isolation algorithm (triggered by an alarm) are shown in Figure 16 for true voltage error. There are false isolations within R_p, correct isolations outside R_p, and inconclusive isolations scattered throughout. Figure 17 shows these points plotted in parity space. The false isolations are associated with false alarms. These are cases where the bias in voltmeter 2 did not cause R_p to be exceeded, even though the detection threshold was crossed in parity

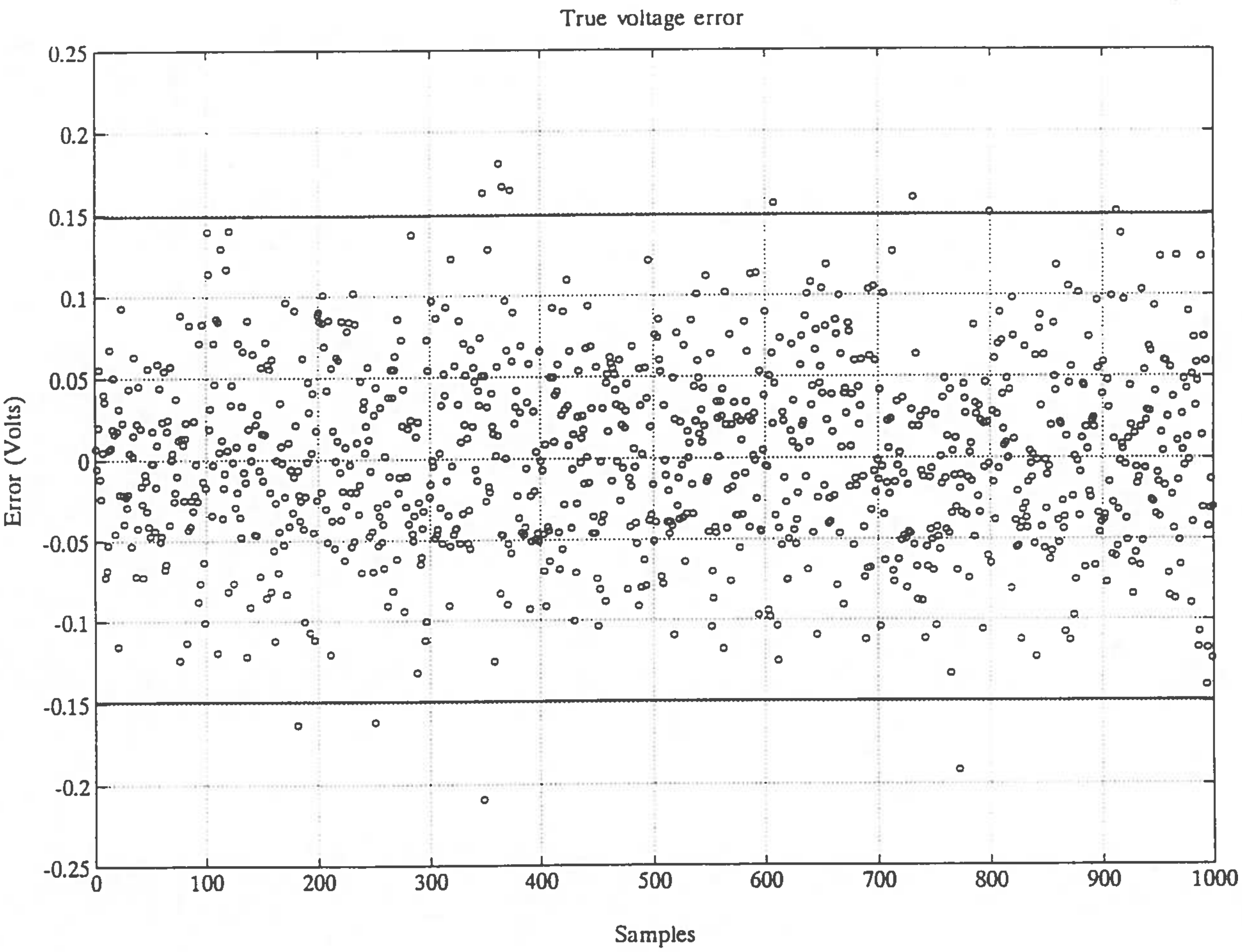


Figure 6 True Voltage Error for 1000 Samples (No Bias).

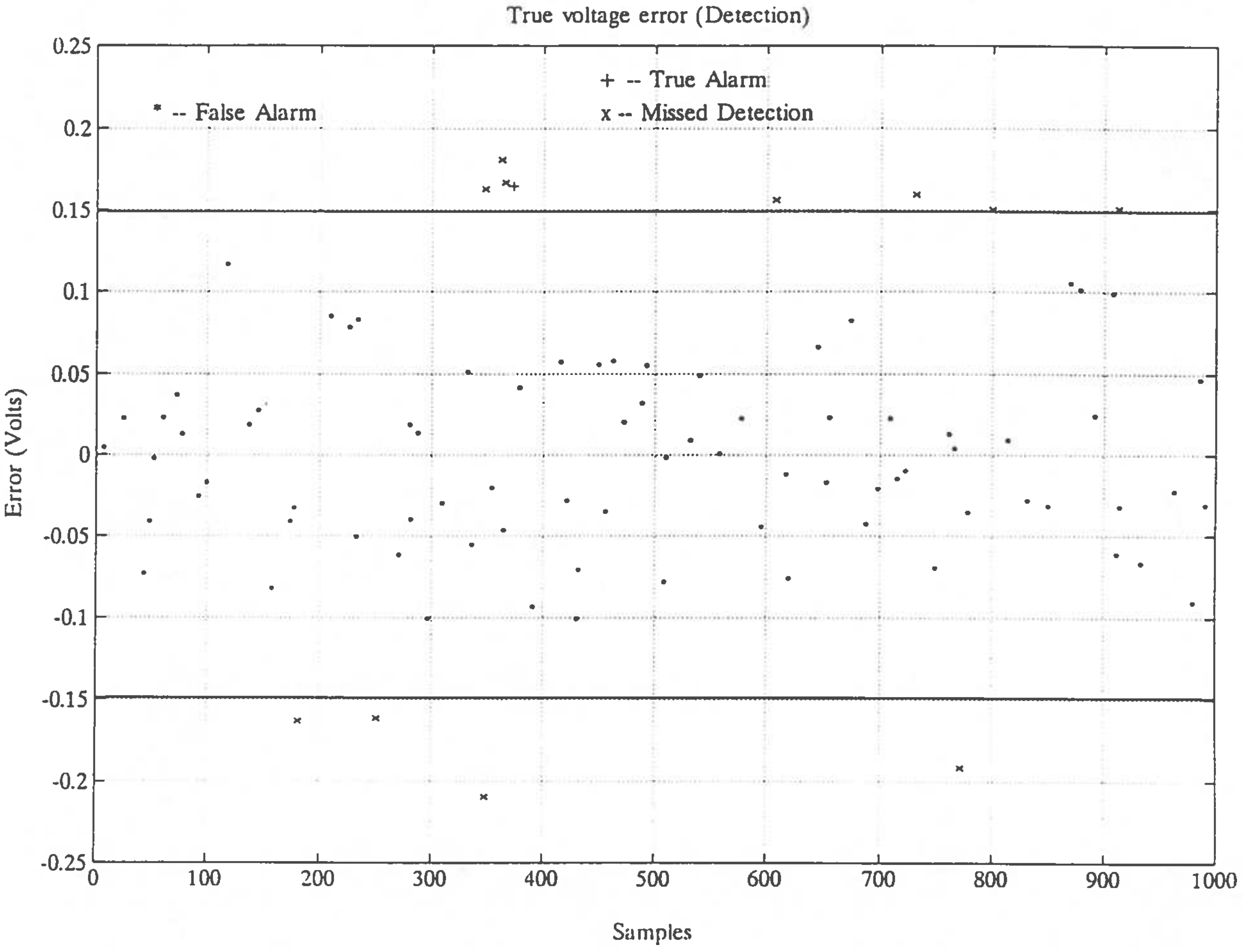


Figure 7 True Voltage Error - Detection (No Bias).

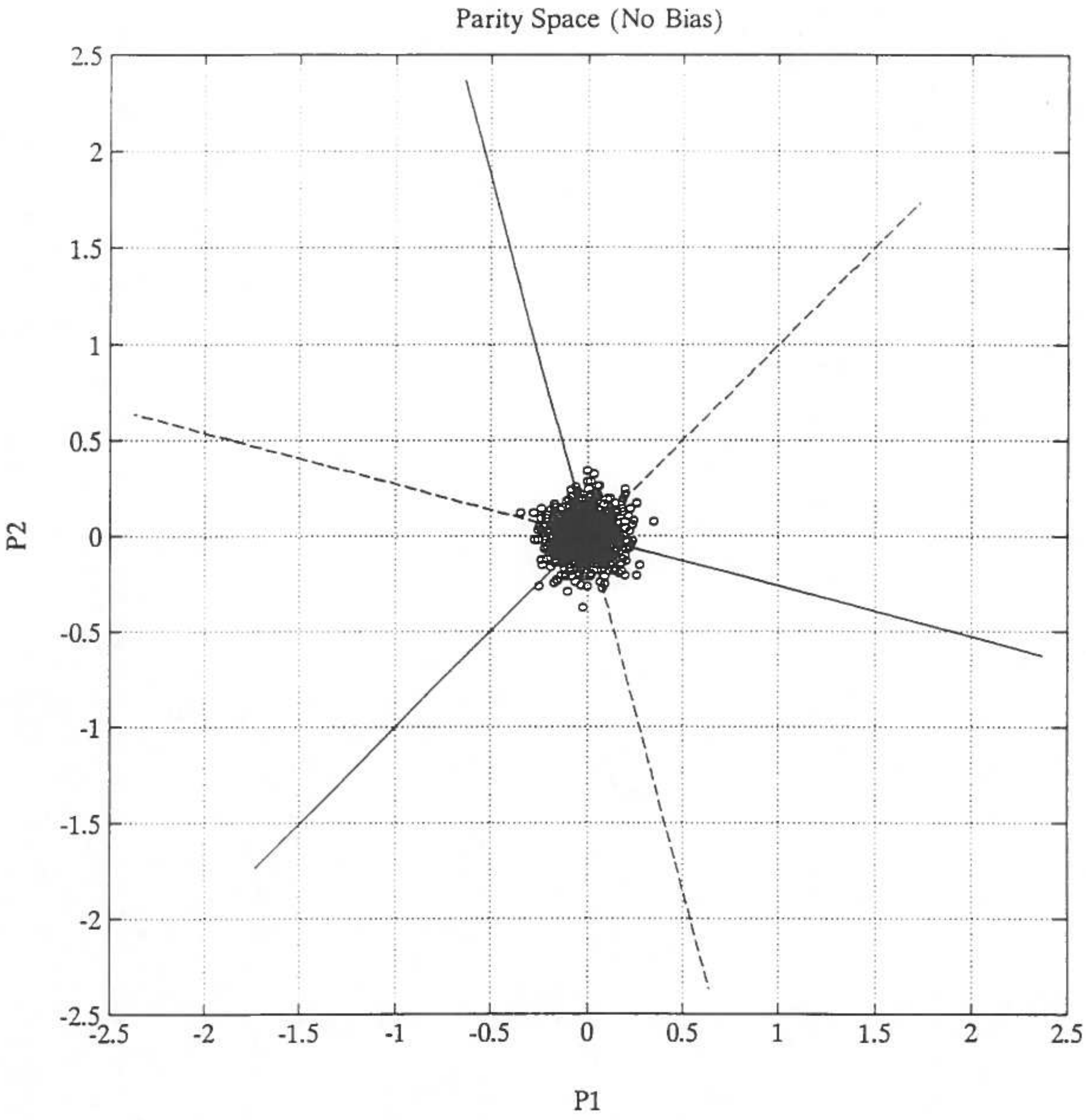


Figure 8 Measurement Points in Parity Space (No Bias).

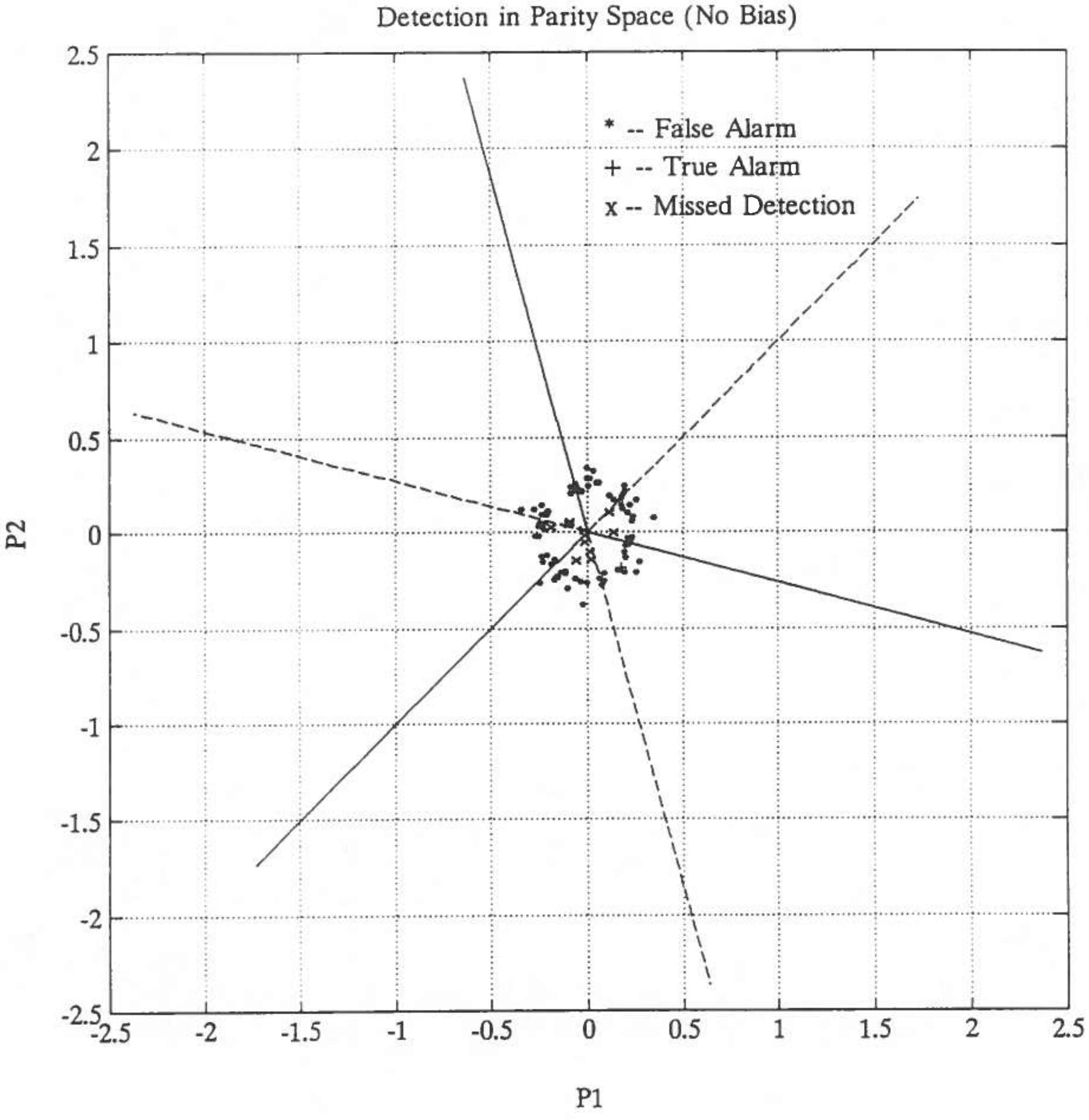


Figure 9 Detection in Parity Space (No Bias).

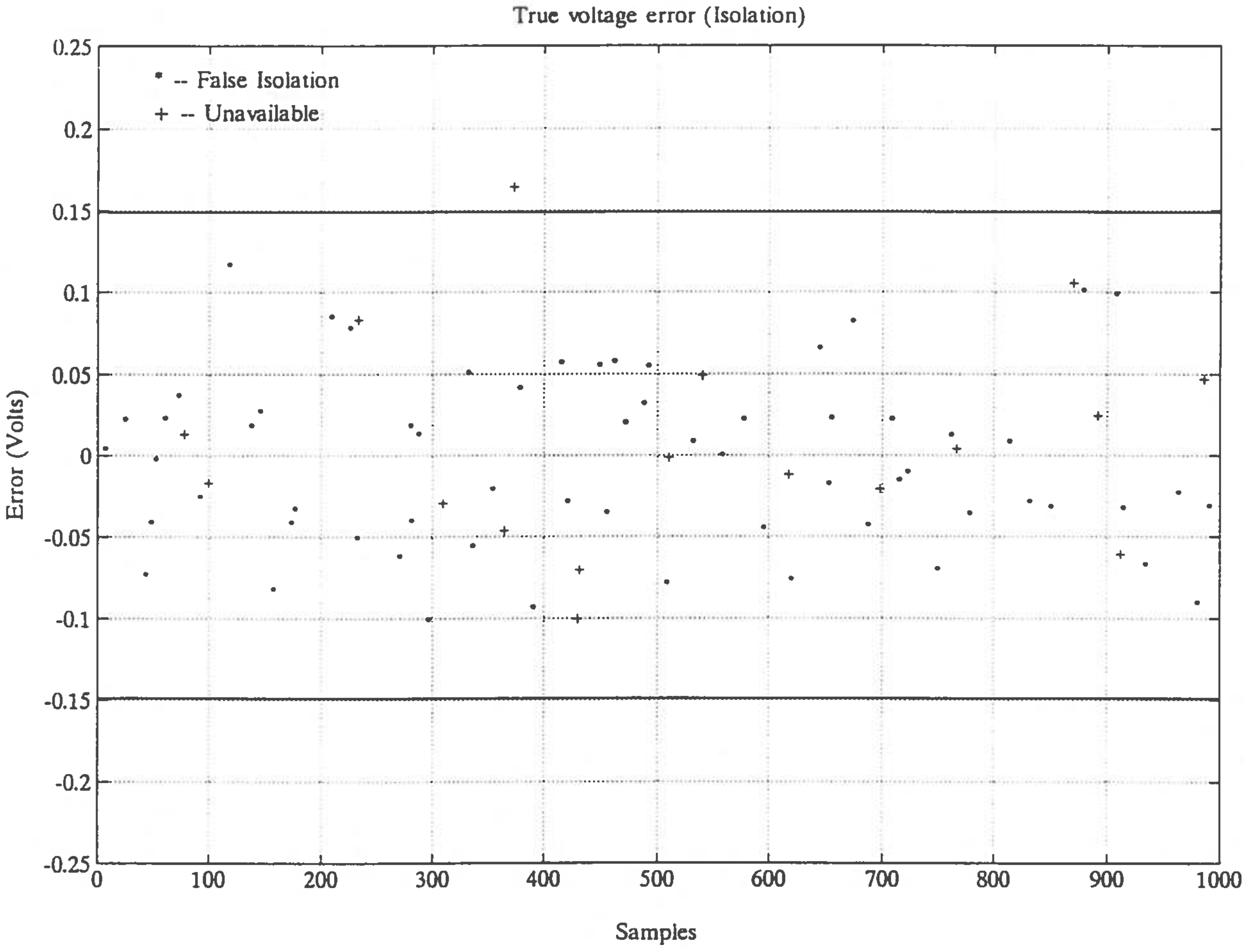


Figure 10 True Voltage Error - Isolation (No Bias).

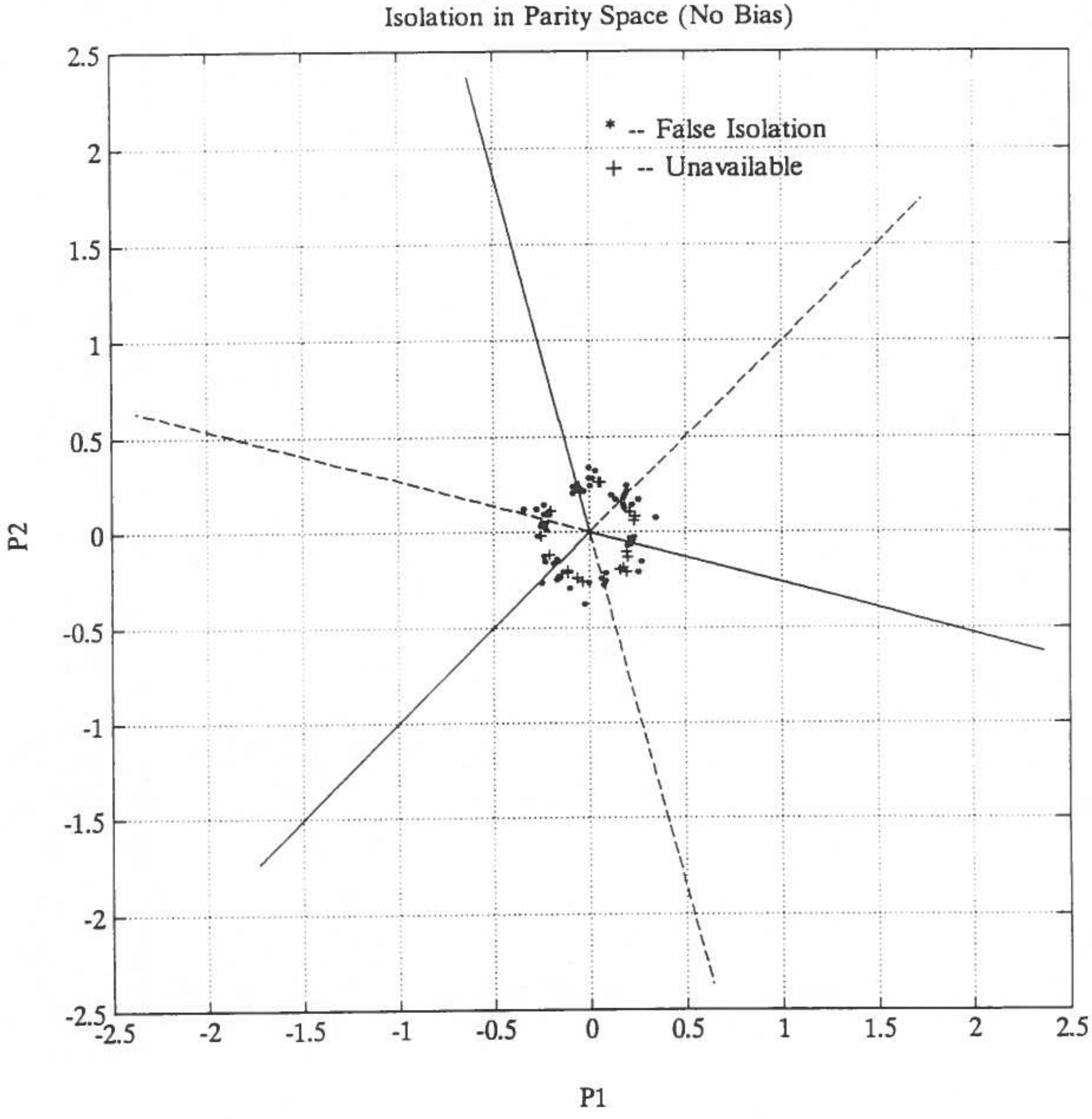


Figure 11 Isolation in Parity Space (No Bias).

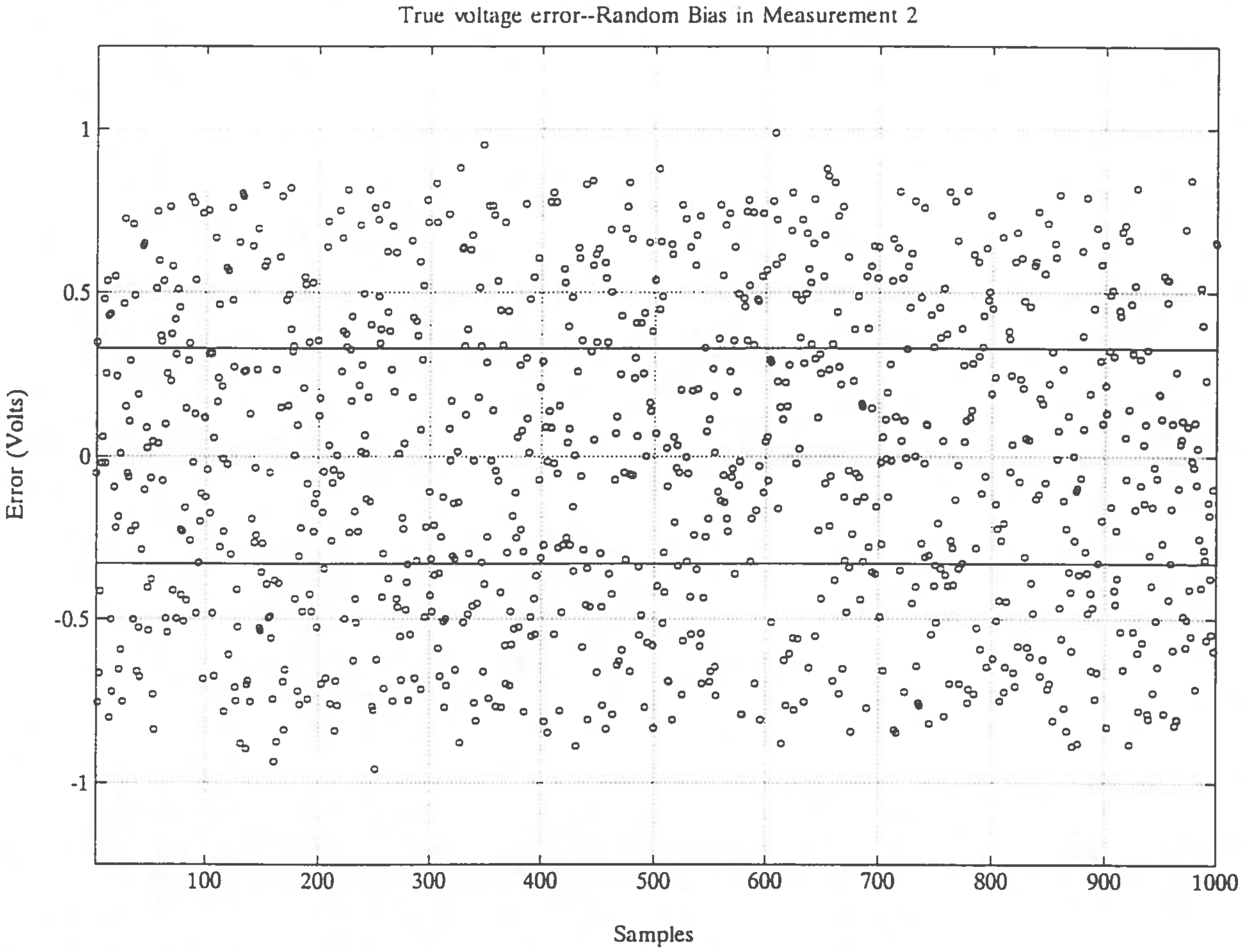


Figure 12 True Voltage Error (Bias in Voltmeter 2).

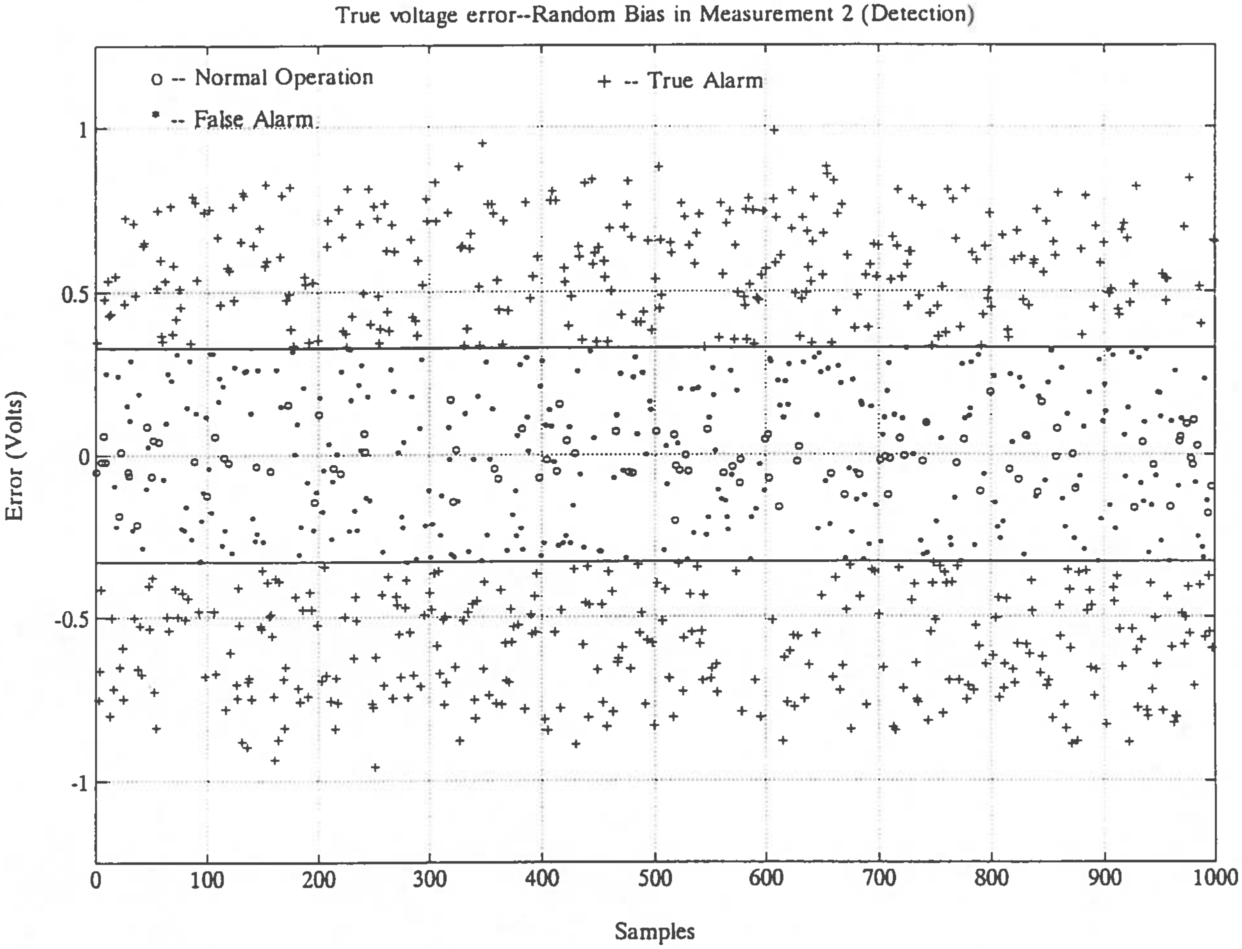


Figure 13 True Voltage Error - Detection (Bias in Voltmeter 2).

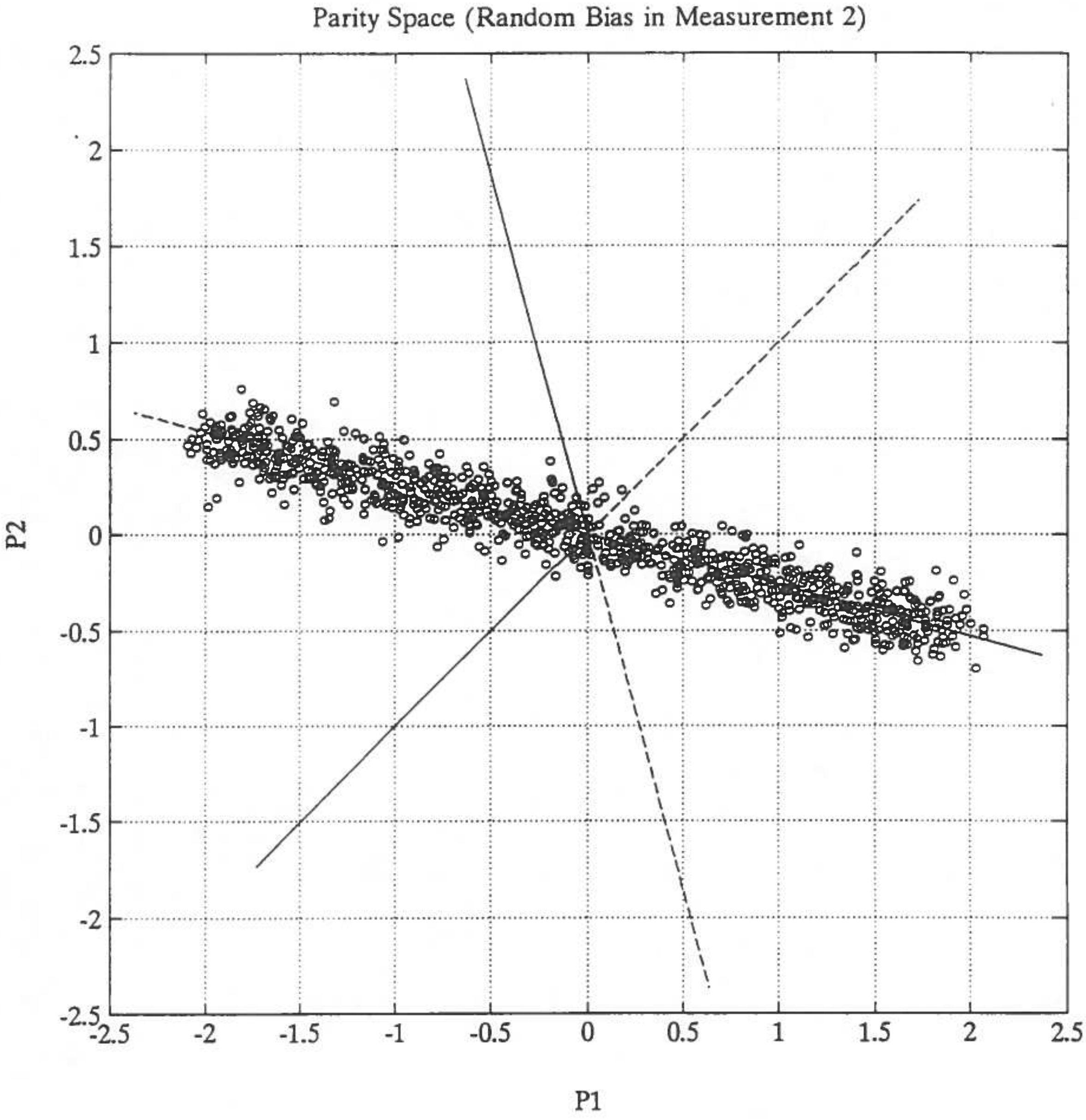


Figure 14 Measurement Points in Parity Space (Bias in Voltmeter 2).

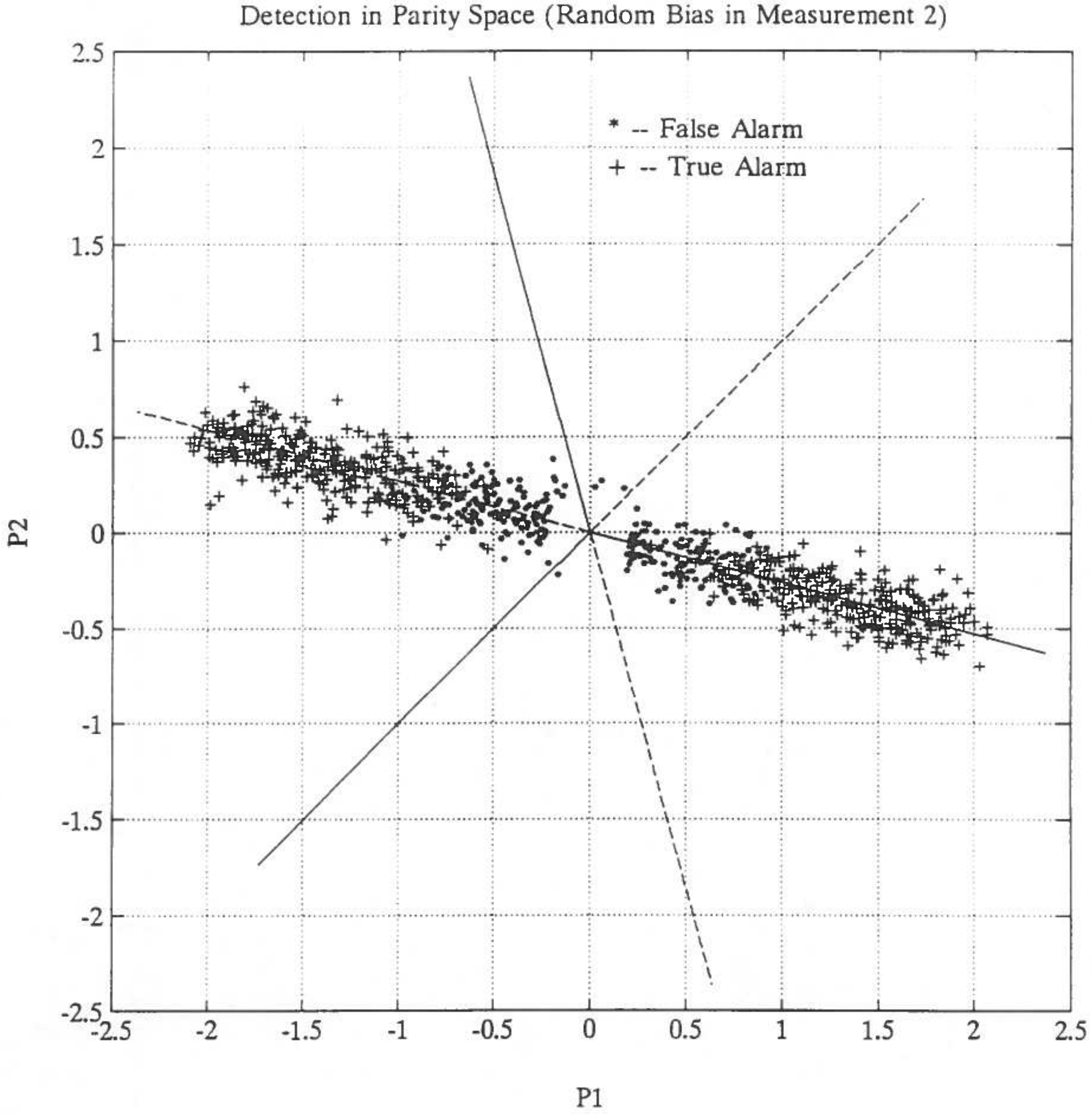


Figure 15 Detection in Parity Space (Bias in Voltmeter 2).

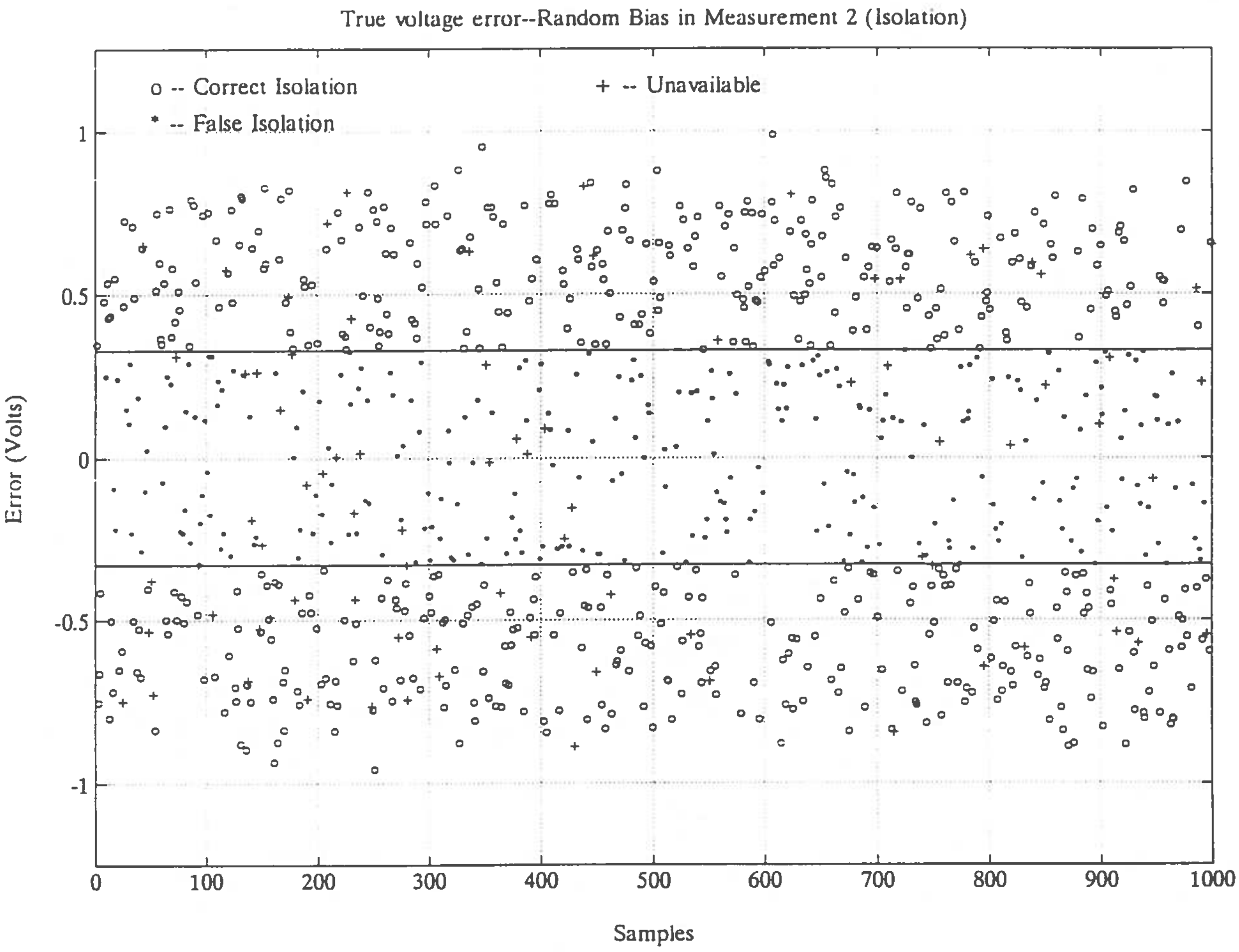


Figure 16 True Voltage Error - Isolation (Bias in Voltmeter 2).

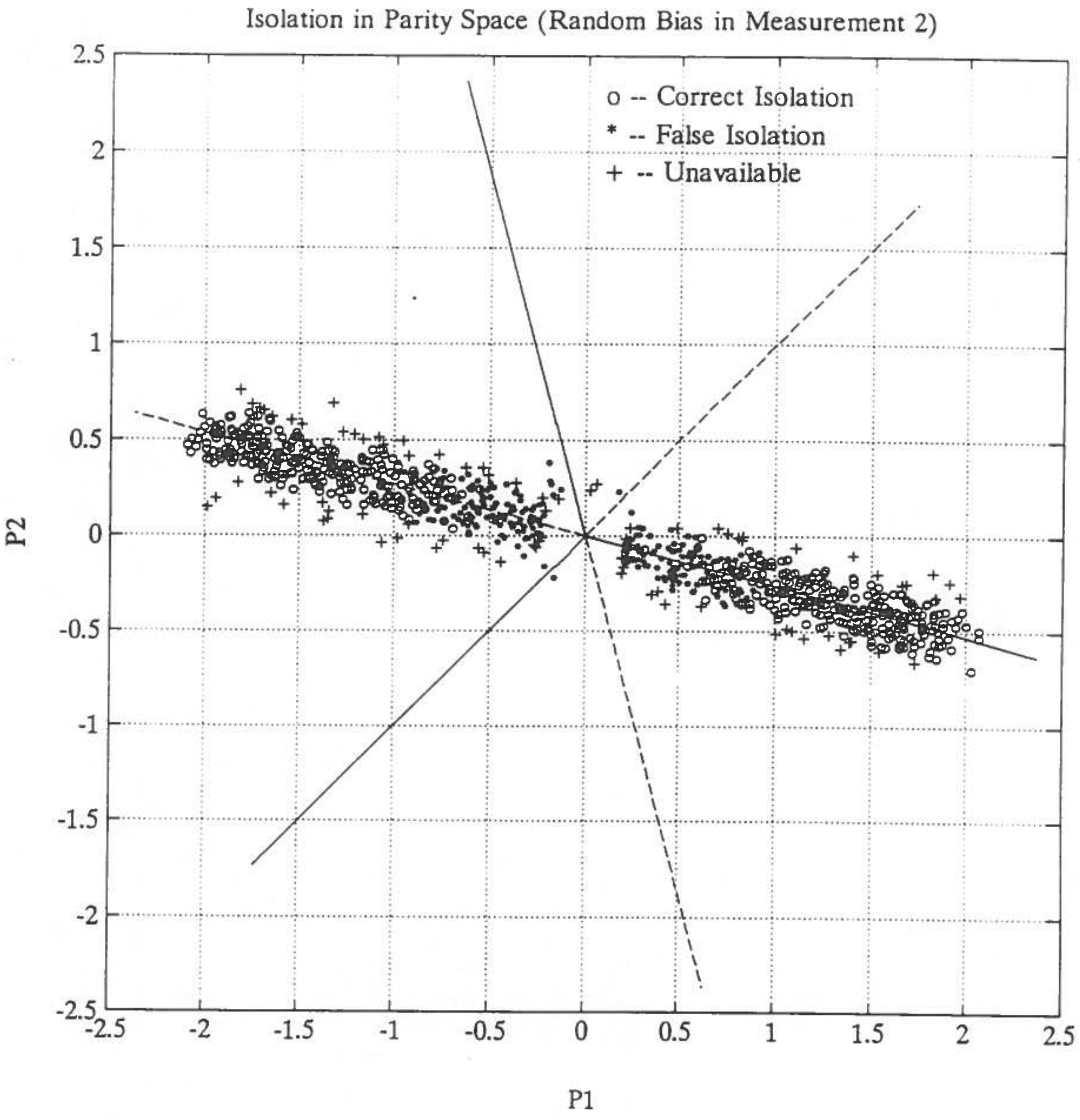


Figure 17 Isolation in Parity Space (Bias in Voltmeter 2).

space. Many of the errors were correctly isolated by using the detection algorithm applied to subsets of 2 voltmeters as outlined in a prior section. However, this algorithm seems to be a bit conservative because many of the unavailable cases could well be isolated to voltmeter 2. In fact, a corridor is defined along the axis which would indicate that a fault is only isolated if the point is within a certain distance from the axis. It seems more prudent to define a cone about the axis extending outward so that many more points could be correctly isolated. It is possible to force the isolation by choosing which axis a given point is most likely associated with. This is depicted in Figure 18. Fortunately, there were no wrong isolations, and most of the true alarms were correctly isolated to voltmeter 2. Recall that a wrong isolation means that a bad measurement exists, but a good measurement was removed from the solution. In addition, all false alarms which could not be isolated and hence resulted in unavailability, are now turned into false isolations. In the case of a false isolation, no fault exists, but the alarm was raised and consequently, a good measurement was removed from the solution. False isolations are preferred over unavailability, because a false isolation does not disable the navigation function.

In addition to the forced isolation technique given above, an overall isolation strategy is needed. For instance, what if two measurement axes are close together in parity space? Should both measurements be discarded if the parity vector is near them? Also, it may be necessary to include a buffer between axes so that the probability of a wrong isolation is reduced. These questions are not within the scope of this report, but the selection of an isolation strategy is a very important issue which must be studied carefully for sole means of navigation systems.

3. AVAILABILITY OF FAULT DETECTION FOR THE GLOBAL POSITIONING SYSTEM

3.1 Background

The Minimum Operational Performance Standards (MOPS) for the Global Positioning System (GPS) have been developed by Special Committee 159 of RTCA for supplemental airborne navigation equipment using GPS [12]. Except for availability and integrity, GPS will satisfy all performance requirements. Initially, the GPS receiver will rely on receiver autonomous integrity monitoring (RAIM) to provide integrity assurance. This means that all fault detection is to be performed in the receiver, although a GPS Integrity Channel (GIC) is anticipated to be available in the future. The GIC will consist of a ground segment that monitors the GPS signals continuously and a space segment that broadcasts GPS integrity information via geostationary satellites. Integrity assurance can also be achieved by using additional navigation sensors, such as an altimeter, Loran-C, or GLONASS. This section will show that GPS by itself does not provide enough measurements to allow "adequate" fault detection. Thus, it will be shown that GPS must be augmented by some other system in order to meet

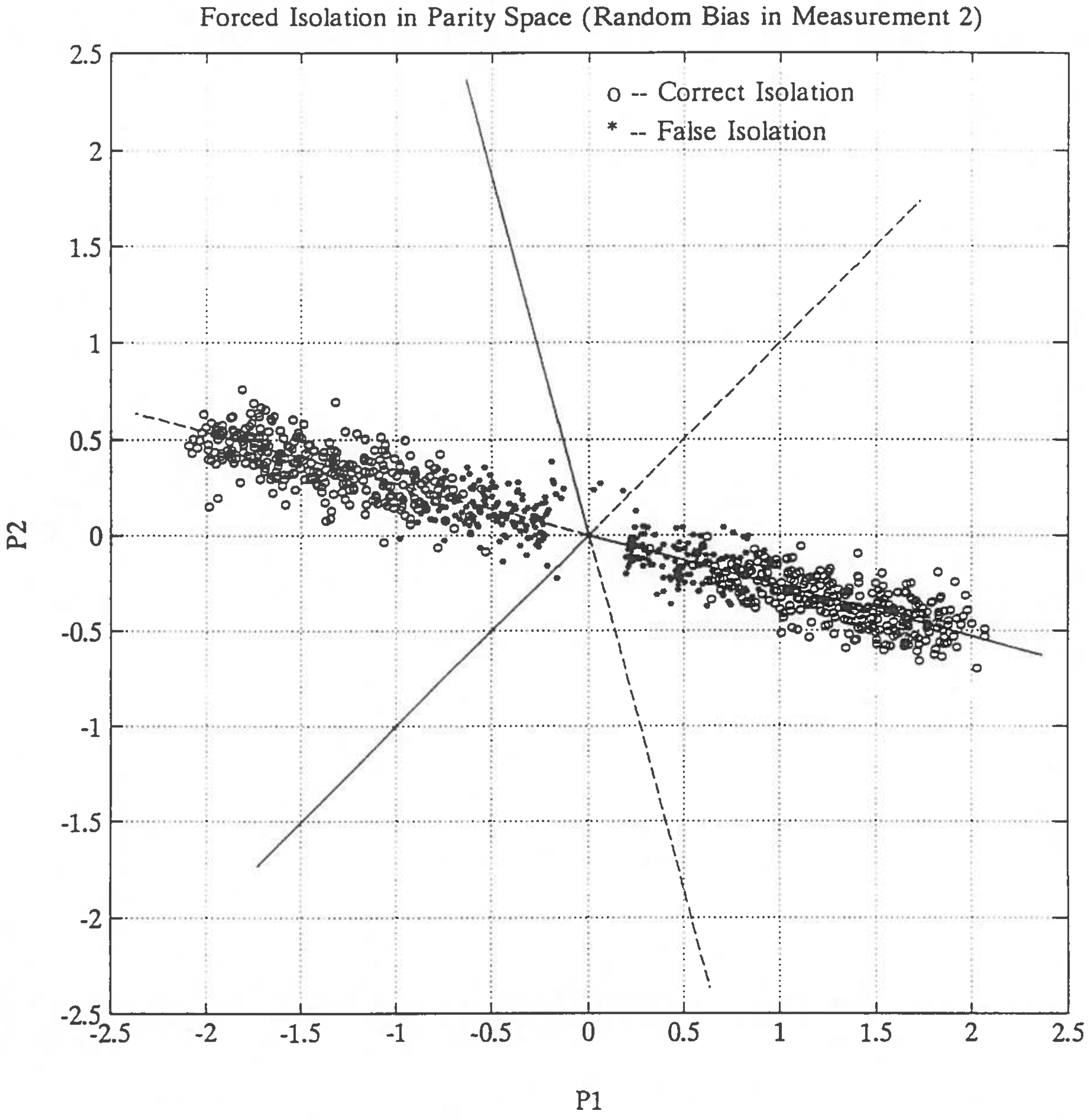


Figure 18 Forced Isolation in Parity Space (Bias in Voltmeter 2).

requirements for sole means navigation. Sole means availability requirements are currently not defined, but are anticipated to be more stringent than supplemental availability requirements [13]. Therefore, the availability of Loran-C, a widely accepted supplemental navigation system, is used as a reference for GPS. A parametric computer analysis will be presented to show the availability of fault detection for GPS.

The availability of a navigation system is the percentage of time that the system can be used at a certain location [14]. For GPS, four measurements are required for the solution of three-dimensional position and clock bias. Also, the relative geometry of the satellites must provide a sufficiently small Horizontal Dilution of Precision (HDOP). For fault detection, an extra measurement is necessary to allow for a parity equation. Thus, the availability of integrity depends on having a redundant measurement. Note that for sole means navigation, two redundant measurements are required in order to isolate faults. A method for determining the availability of fault detection for GPS has been incorporated into a computer model. Much uncertainty exists with respect to the final implementation of GPS in terms of satellite constellation and maintenance. Therefore, simulation results are provided as a function of satellite constellation and satellite failure statistics. Also, different accuracy requirements exist for the four phases of flight: oceanic, en route, terminal, and nonprecision approach. To accommodate this, the availability is provided as a function of the GPS geometry in terms of HDOP. The root-mean-squared (rms) two-dimensional horizontal position error (2D-error) and the rms range measurement error (R-error) are related by the HDOP as follows:

$$\text{2D-error} = \text{HDOP} * \text{R-error} \tag{3.1}$$

For instance, consider a typical rms range measurement error of 33.3 meters and an HDOP of 1.5; the rms horizontal position error would be 50 meters. Refer to reference [15] for the HDOP calculation. In the case of constant range measurement error statistics, a higher HDOP value is directly proportional to a larger two-dimensional (horizontal) position error.

3.2 Computer Simulation Program

A coverage model for GPS has been developed for use on a Personal Computer (PC). A Markov model has also been implemented for failure scenarios. Both programs are written in FORTRAN and their output is used by a third program to determine availability. For each location and time, the coverage model provides the number of visible GPS satellites and their geometry. If the geometry (HDOP) or the number of satellites is not sufficient for fault detection, an outage is declared. This is repeated for all possible single satellite failures and all simultaneous satellite failures of up to 6 satellites. The resulting outages are then combined with their corresponding probabilities of occurrence as determined by the Markov model to arrive at the average availability of fault detection.

3.2.1 GPS Coverage Model

By determining the geometry of visible satellites, the coverage model provides outage information for a given location at a given time. GPS satellites can then be failed to assess the effect on the best available HDOP. A method is presented which allows for the failing of visible satellites only, rather than failing all possible combinations of satellites. This method greatly reduces the number of computations. The equation for finding the number of combinations of n out of m satellites { COMB(n:m) } is given as:

$$\mathrm{COMB}(n{:}m) = \frac{m!}{n!\,(m-n)!} \tag{3.2}$$

For instance, consider a 24-satellite constellation with 4 failed signals. There are 4-out-of-24 different failure combinations possible:

$$\mathrm{COMB}(4{:}24) = \frac{24!}{4!\,(24-4)!} = 10626 \text{ combinations}$$

Next, consider taking combinations of 4 failures out of 7 visible satellites:

$$\mathrm{COMB}(4{:}7) = \frac{7!}{4!\,(7-4)!} = 35 \text{ combinations}$$

For this scenario, the number of failure combinations to be considered is reduced by a factor of approximately 300. It is important to realize that this approach does not cover all failure scenarios since only visible satellites are failed. However, it is computationally simple to account for the other failure scenarios in the program that processes the outage records.

The GPS program calculates coverage based on available measurements at each search point over a period of one day, with a time increment of 5 minutes. A satellite is declared visible if its elevation angle is greater than 7.5 degrees with respect to the horizon. The algorithm is as follows:

```
Initialization
FOR each location
   FOR each time
         - calculate GPS visibility
         - calculate HDOPs for all subsets of usable signals
           (elevation angle > 7.5 degrees)
         - write coverage data to file
         FOR all combinations of up to 6 simultaneous failures
            - determine the best value of HDOP for each scenario
            - check for integrity
            - IF no integrity, write outage to a file
          END failure loop
      END time loop
    END location loop
```

An output file with integrity outages is created. These are caused by a lack of a sufficient number of measurements or by poor geometry (large HDOP). Relaxing the HDOP specification will cause fewer outages; however, this does not provide 100% availability of fault detection. A limit is imposed by the number of receivable signals. At least 4 measurements are required to solve for 3-dimensional position and clock offset, and 5 measurements are needed for fault detection. Even with 5 or more signals, integrity is still lacking if fewer than 5 subsets have an HDOP lower than the specified limit. The simulation will show that more measurements are needed (GPS must be augmented) in order to detect and isolate faults.

Various scenarios can be examined based on different simulation parameters. For instance, one option is the inclusion of the altimeter as a measurement. The altimeter is implemented in the program as a satellite vehicle located at the origin of the Earth-centered, Earth Fixed (ECEF) coordinate system. It is assumed to never fail, although it could be included in the Markov analysis if its Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR) were specified. For this analysis, however, the altimeter is either included as a signal or not included at all. It should be noted that using the altimeter introduces a larger measurement error. However, for enroute navigation and for nonprecision approaches an altimeter correction can be made. A problem does exist in the terminal area because a descent or climb can cause large absolute altitude errors (up to 1000 ft) [16]. Approximately 70% of the altimeter error is included in the horizontal position error (in this case, 700 ft). Even so, the results for each case can be compared to see what significance the altimeter has in improving GPS availability. Also, the HDOP limit can be changed to see what its effect is on availability. Since availability is a function of location, the simulation is run for several different sites.

3.2.2 Markov Model

For each failure scenario identified by the coverage model, a corresponding probability of occurrence must be found. These probabilities are a function of the mean time to failure (MTTF) and the mean time to repair (MTTR) of the GPS satellites. The design lifetime of a GPS satellite is 7.5 years, which is used for the MTTF. The repair rate is dependent on the launch schedule. If a new satellite is placed in orbit every 4 months, then the MTTR is 2 months. This is approximately the repair rate necessary to keep a 21-satellite constellation operational over a long period of time. A shorter (more optimistic) MTTR is included in the analysis since the actual repair rate is governed by the maintenance of the constellation. For example the 21 Primary constellation (21 satellites plus 3 active spares) will be maintained by a launch-on-schedule policy [17].

Launches will be scheduled to take place every 3 months, resulting in a MTTR of 1.5 months. Short term failures requiring relatively small repair times do not have a significant effect on availability and are omitted [18]. Therefore, 22 states are defined ranging from 0 to 21 failures (25 states for the 21 Primary constellation). When altimeter aiding is chosen as an option, the altimeter is assumed to never fail; thus, it is not included in the Markov analysis. For the purpose of this report, the Markov model is truncated to 8 states. The first state represents the scenario with no failures, and state number 8 represents all scenarios with 7 or more failures.

Each state in the Markov model has transitional probabilities assigned to it. The failure rate (λ) is the reciprocal of the MTTF:

$$\lambda = \frac{1}{\text{MTTF}} \tag{3.3}$$

To obtain a transitional probability (P_F), the failure rate is multiplied by a time step (T):

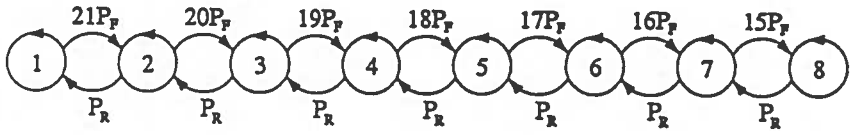
$$P_F = \lambda * T \tag{3.4}$$

Since no probability can be greater than 1, the time step has a definite upper bound. The actual criterion for the time step, however, is that it be chosen small enough to ensure that only one failure or one repair occurs during that length of time. In practice, the time step is reduced until there is no significant change in the simulation results. The transitional probabilities for repairs are treated the same way:

$$\mu = \frac{1}{\text{MTTR}} \tag{3.5}$$

$$P_R = \mu * T \tag{3.6}$$

where μ is the repair rate and P_R is the transitional probability of repair. Each transitional failure probability is multiplied by the number of "healthy" satellites available. Note that this multiplication is only justified for small transitional probabilities [19]. Due to a restricted launch schedule, each repair probability is singular, independent of the number of failed satellites. An example Markov diagram is given in Figure 19 for a 21-satellite constellation. Each circle represents a failure state. The Markov diagram can be used to derive a stochastic transitional probability matrix (P). The elements of this square matrix contain transitional probabilities. The probability of remaining in a particular state is [1 - (all probabilities of leaving that state)]. These particular probabilities are located along the matrix diagonal. One requirement is that all the probabilities in a row add up to one. Each row of 8 elements contains, at most, 3 transitional probabilities; all other elements are zero. The basic trichotomy is comprised of the following probabilities: 1) leaving the state due to a failure, 2) leaving



STATE	#GPS
1	21
2	20
3	19
4	18
5	17
6	16
7	15
8	≤14

Figure 19 Markov Diagram for GPS (21 Satellites).

the state due to a repair, and 3) remaining in the state.

The method for obtaining the final state probabilities is to successively multiply the state vector $\underline{s}$ by the state transition probability matrix P. Starting with the initial conditions (s_0), the multiplications are carried out as follows:

$$\begin{aligned} &\text{after 1 time step:} \quad s_1 = s_0 P \\ &\text{after 2 time steps:} \quad s_2 = (s_0 P) P = s_0 P^2 \\ &\text{after n time steps:} \quad s_n = s_0 P^n \end{aligned} \tag{3.7}$$

Since the state vector stabilizes to the time-limiting state probabilities after a large number of multiplications, it is not necessary to raise P to exactly the power n. The calculation proceeds as follows. The number of time steps n is first determined from the mission time and the time step:

$$n = \frac{\text{(mission time)}}{T} \tag{3.8}$$

The matrix P is then raised to m, the nearest power of 2 (m > n). This is achieved by successive squaring of the transition matrix:

$$\begin{aligned} P^2 &= P\ P; \\ P^4 &= P^2 P^2; \\ P^8 &= P^4 P^4; \\ &\text{(etc.)}\ldots\ldots \end{aligned}$$

For large n this method is very economical calculation-wise, since only $\log_2(m)$ matrix multiplications are required.

The time limiting state vector is found when the changes in the state vector, $\underline{s}$, probabilities are small for an additional multiplication. Table 2 shows the final state probabilities for different MTTRs using both the 21- and 24-satellite constellations. The probabilities drop off rapidly which indicates that very rarely will a multi-failure scenario occur. For this analysis, up to 6 satellites are failed; a more complete analysis would include failure of more satellites, but since the probability of being in a 7-failure state is small for most cases, there is not a significant loss of accuracy in stopping at 6 failures.

3.2.3 Integration of Coverage and Markov State Probabilities

A third program combines the coverage and integrity output data with failure scenario

NUMBER FAILED	MARKOV STATE PROBABILITIES					
	21 SATELLITES			24 SATELLITES		
	MTTR (months)			MTTR (months)		
	1	1.5	2	1	1.5	2
0	.769929	.658382	.550809	.737222	.610328	.489403
1	.179650	.230432	.256994	.196593	.244128	.260906
2	.039922	.076810	.114158	.050240	.093579	.133210
3	.008428	.024322	.048149	.012281	.034310	.064996
4	.001686	.007296	.019225	.002866	.012007	.030238
5	.000318	.002067	.007244	.000637	.004001	.013383
6	.000057	.000551	.002567	.000134	.001267	.005623
≥ 7	.000009	.000138	.000853	.000027	.000380	.002240

Table 2 Markov State Probabilities for 21- and 24- Satellite Constellations.

probabilities from the Markov model to calculate the availability of fault detection. This program also accounts for failure scenarios containing satellites which are not visible to the user. First, the straightforward processing of the outage records will be discussed.

Each outage is assigned a state probability (P_m) from the Markov model. Since the outage only occurs at the current time, the Markov state must be multiplied by the probability that the outage occurs at a particular time (P_t). P_t is simply the quotient of the length of the outage (T_{inc}) and the mission time (T_m):

$$P_t = \frac{T_{inc}}{T_m} \tag{3.9}$$

Finally, the product P_t*P_m must be divided by the number of possible failure combinations of n satellites out of a total number of m in the constellation {COMB(n:m)}. For example, if 2 satellites are failed from a 21-satellite constellation, there are 210 possible combinations {COMB(2:21)}. Thus, the percent unavailability caused by one outage is given by:

$$\%\,\text{unavailability} = \frac{P_m * P_t}{\text{COMB(n:m)}} \tag{3.10}$$

These values are totaled for all outages, but the sum does not yet represent the total unavailability.

Each outage is caused by a particular set of failures containing up to 6 visible satellites. At first glance it seems unnecessary to consider the failure of non-visible satellites since these won't cause outages at the specified location for which unavailability is being determined. However, there are failure combinations that contain some visible satellites and some non-visible satellites. If the failed visible satellites from a set of this nature cause an outage, then this scenario must be accounted for in the unavailability number. This circumstance is not included in the failure simulation and must be taken care of in the program that processes the outages. An example is given to further illustrate this point.

Consider a case where the user has 5 satellites in view (just enough for fault detection) and 16 below the specified elevation mask angle (7.5 degrees). A failure of any one of the visible satellites will cause an outage (assuming no altimeter aiding). Each of these cases is contained in the failure simulation. Consider now a two-failure state where one visible satellite and one non-visible satellite are failed. This causes an outage to the user although the two-failure state has a lower probability than the one-failure state. Still, though, this appears to the user to be a single satellite failure. For this case, one visible satellite could be paired with any of the 16 non-visible satellites in a two-failure state. Since there are 5 visible satellites, there are 80 of these combinations which are neglected in the failure simulation. Since there are only 210 possible combinations of 2 out of 21 satellites, 80 is too big a number to ignore. These cases are weighted less since the Markov probability for a two-failure state is less than that for a one-failure state, but are still significant and must be included. For this example, only 10 combinations of 2 visible failures are accounted for in the failure simulation; the additional 80 two-failure states described above are also weighted with the same Markov probability.

The above example can be generalized for all failure scenarios. For instance, 3 satellites could fail (1 visible, 2 non-visible) and an outage would still occur; 4 satellites could fail (1 visible, 3 non-visible), and so on, up to 6 satellite failures (1 visible, 5 non-visible). Continuing this analysis, 2 visible satellites can fail along with combinations of as many as 4 non-visible ones, 3 visible satellites can fail along with combinations of up to 3 non-visible ones, and so forth. An example algorithm for the case of 3 failed visible satellites follows:

Three visible failures:

- fail all combinations of 3 visible satellites (this is the straightforward case already calculated in the failure simulation)
- fail 4 satellites, 3 visible and 1 not
- fail 5 satellites, 3 visible and 2 not
- fail 6 satellites, 3 visible and 3 not

A similar algorithm is used for 1, 2, 4, and 5 visible failures. The 6-failure state is treated differently since it is the maximum number failed. Only the straightforward case is considered where combinations of 6 visible satellites are failed. However, the Markov probability for the 7 or more failure state is added on to the percent unavailability at the end of the prior calculations to arrive at the final answer. This accounts for such cases as 7 failures with 6 visible and so forth.

The supplemental failure states can be calculated based on the outage records produced by the failure routine. The actual algorithm for a 21-satellite constellation is given below:

```
% unavailability = 0.0
FOR all outage records
   - f = number of visible failures that caused the outage
   - % unav. = % unav. + $P_t$ * $P_m$(f failures) / COMB(f:21)
   FOR i = f+1 until 6
         - % unav. = % unav. + $P_t$ * $P_m$(i failures) *
                  COMB(i-f:#non-visible) / COMB(i:21)
   END supplemental failure loop
END outage record loop
```

As was previously mentioned, only the straightforward case is calculated for 6 failed satellites { P_t * P_m(6 failures) / COMB(6:21) }. To arrive at the final number for percent unavailability, the marginal Markov state probability is added in {P_m(≥ 7 failures) }, to account for all neglected failure scenarios.

3.3 Simulation Results

Figures 20 and 21 show the availability of fault detection (or RAIM) for GPS averaged over 3 locations: San Francisco (SFO), Denver (DEN), and Washington, D.C. (DCA). The simulation was run for these 3 locations for a period of 1 day and a time increment of 5 minutes. GPS orbital parameters for the Optimal and Primary constellations are taken from Green [20]. Up to 6 simultaneous failures were simulated with each scenario weighted by a Markov state probability. The state probabilities were obtained from the Markov model with a mission time of 2 years and a time increment of 1 minute. After the results were obtained for each location, the numbers were averaged to simplify the interpretation of the results.

Figures 20 and 21 show that availability increases as a function of allowed HDOP and is limited by the number of visible satellites (even with a high HDOP limit, the availability is not 100%). Still though, as the HDOP limit is increased, there is a significant increase in availability of fault detection. For example, if the HDOP limit is increased from 3 to 60 in the case of the Optimal 21 constellation (no altimeter, MTTR = 2 months), a gain in availability from 66.0% to 94.8% is obtained. Also, as expected, a more optimistic MTTR increases the availability.

The Optimal 21 constellation with altimeter aiding produces Fault Detection availabilities very similar to the 21 Primary constellation without altimeter aiding. Since the 21 Primary constellation actually contains 24 satellites and each one is in view at a particular location an average of 8 hours per day, an average of one extra signal is available over the entire day compared to the Optimal 21 constellation. The altimeter also contributes one extra signal, so that it is predictable that the results for 21 satellites plus altimeter would closely resemble the results for 24 satellites without altimeter.

3.3.1 Discussion of Results

Loran-C availability for a transmitting station is greater than 99.9% [21]. At least three stations are required for a position fix resulting in a Loran-C triad availability of greater than 99.7%. Since Loran-C is widely accepted as a supplemental navigation system, the Loran-C triad availability will be used as a reference for GPS.

For enroute navigation, the current navigation system accuracy requirement is 1.5 nautical miles (95%) which would require a HDOP of better than 41.7 (see Equation 3.1 with a typical R-error of 33.3 meters and recognizing that the 95% position error is

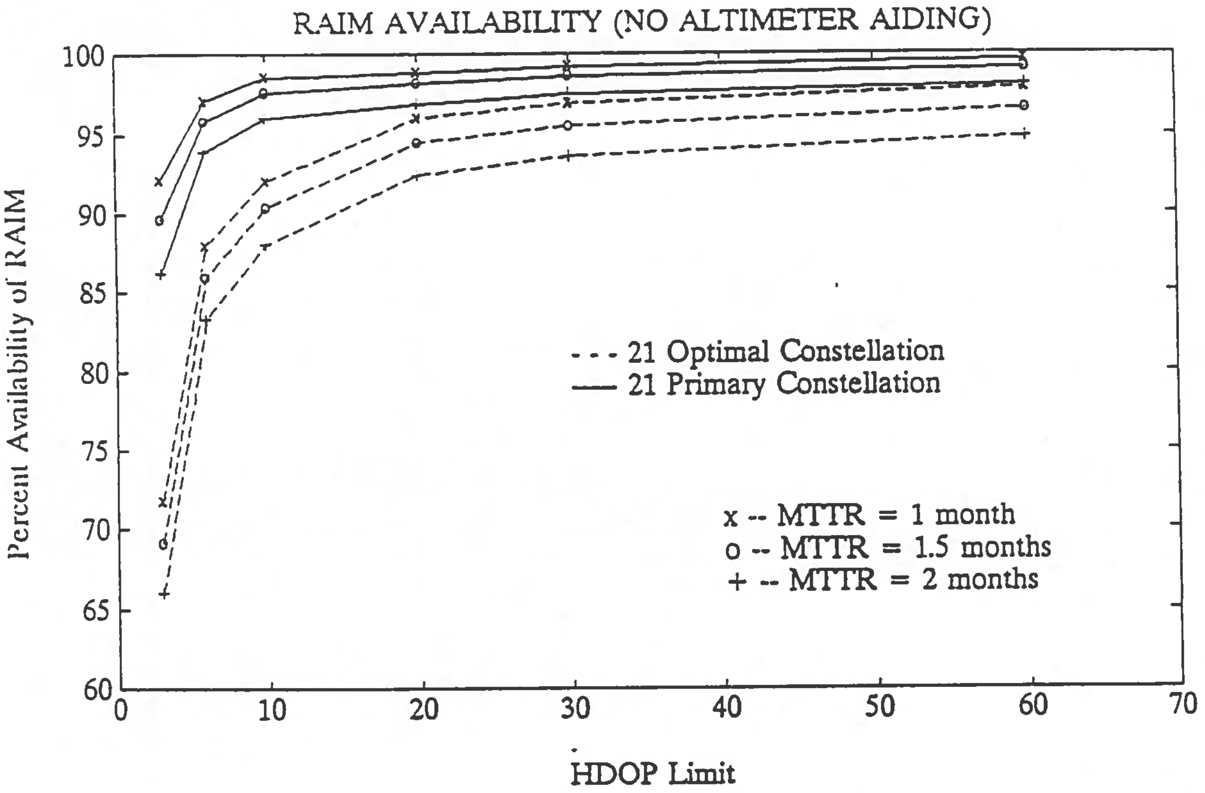


Figure 20 Availability of Fault Detection without Altimeter Aiding.

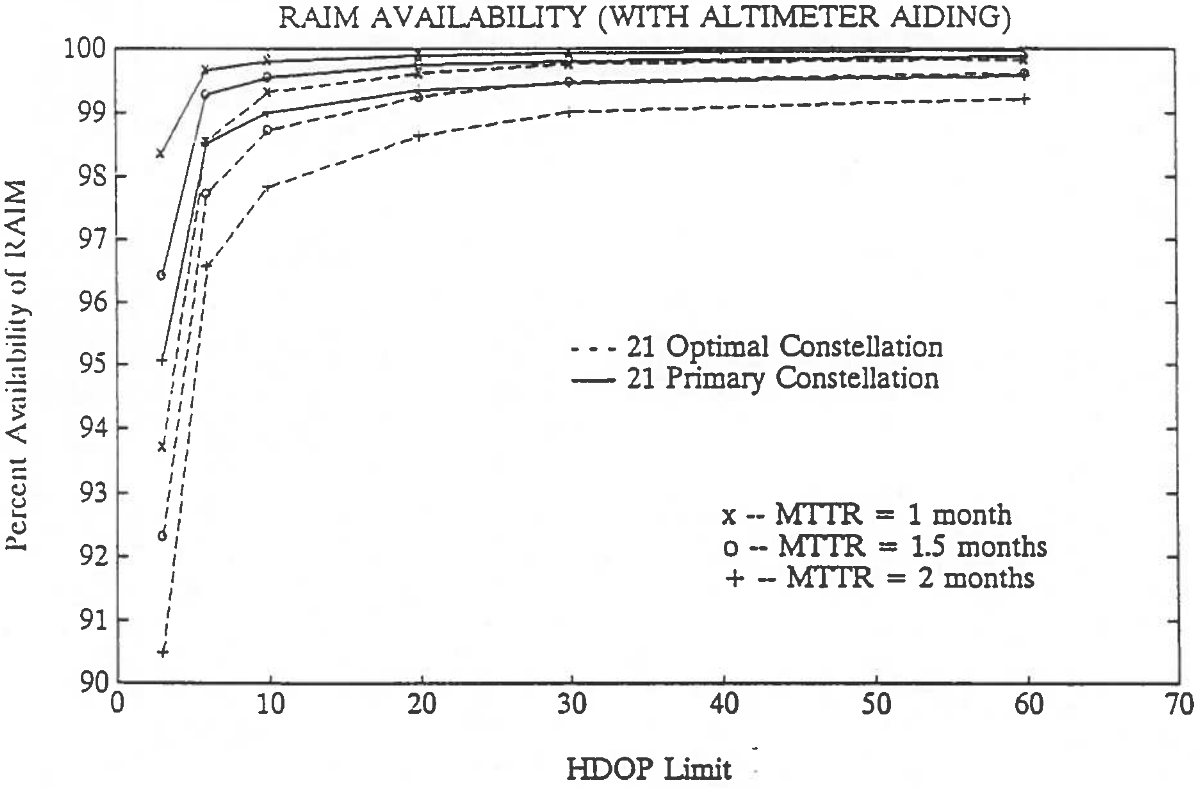


Figure 21 Availability of Fault Detection with Altimeter Aiding.

approximately twice the rms position error) [22]. Similarly, the system accuracy requirement for a nonprecision approach is 0.3 nautical miles which requires a HDOP of better than 8.3. If RAIM is used for integrity assurance, the availability of fault detection must be used for the system availability. Without altimeter aiding, Figure 20 shows that at least the 21 Primary constellation is required with a MTTR of 1 month. Even then, the availability of the nonprecision approach is marginal. With altimeter aiding (Figure 21), the Optimal 21 constellation with a MTTR of 1 month would satisfy the availability requirement for enroute navigation. The 21 Primary constellation with a MTTR of 1 month would satisfy both enroute and nonprecision approach requirements.

3.4 Conclusions Based on the Simulation Results

From Figures 20 and 21 it is seen that there are a few ways to improve the availability of fault detection for GPS. If the HDOP requirement is loosened, the resulting number of RAIM outages during the day is decreased significantly. A large HDOP is the result of poor satellite geometry with respect to the user. However, allowing for a larger HDOP also means that the position accuracy is larger.

Another way to reduce outages is to utilize the altimeter as an extra measurement. From the plots it is seen that using the altimeter significantly improves RAIM availability. The altimeter can be thought of as a geostationary satellite located at the origin of the Earth-Fixed, Earth-Centered (ECEF) coordinate system. Therefore, the inclusion of a geostationary satellite in this analysis would have affected the outcome much the same as the inclusion of the altimeter did. Note, though, that a geostationary satellite would carry the same failure statistics as the other satellites in orbit, and would have a somewhat different effect on HDOP than would the altimeter.

If a more optimistic MTTR is used, the availability increases; the longer a measurement is inaccessible, the lower the availability will be because it is time dependent. Changing the MTTR from 2 months to 1 month typically increases the availability of fault detection by about 2-4%. The 2 month repair time is based on the placing of a GPS satellite in orbit every 4 months. It is hoped that this is a conservative estimate since the MTTR depends on the actual maintenance of the constellation.

There is expected to be a 98% availability of at least 21 operational satellites for the Primary constellation [23]. From Table 2 it can be seen that for a MTTR of 1.5 months, the probability of at least 21 operational satellites is about 98.2%. Therefore, a MTTR of 1.5 months can be considered a realistic estimate.

Although the analysis presented in this section is based on the best available information at this time, clearly the results strongly depend on the maintenance of the GPS satellite constellation. Therefore, the availability numbers should not be interpreted as absolute quantities but are meant to provide an understanding of the impact of RAIM requirements on the availability of GPS. It has been shown that by using optimistic values for MTTR and allowed HDOP, GPS can meet requirements for availability of fault detection; thus, GPS could meet requirements for supplemental navigation. The next step, however, is to define requirements for a sole means navigation system using GPS. Since this requires fault isolation as well as detection, 2 extra measurements are needed. It is apparent from the simulation that GPS by itself would not provide enough available measurements to perform isolation a high percentage of the time. Therefore, GPS must be augmented by another system so that more signals are available. The remainder of this report deals with fault detection and isolation for integrated navigation systems, including a case study of hybrid GPS/Loran-C. Since fault detection and isolation techniques are independent of the chosen hybrid system, the results for GPS/Loran-C apply to other proposed systems, such as GPS/GLONASS and GPS/Omega.

4. INTEGRATED NAVIGATION SOLUTION

Various methods exist to determine the position of an aircraft. With the proposed integrated navigation systems, it is necessary to show that different types of measurements can be incorporated into the solution. For instance, consider the following equations for range and bearing measurements for two-dimensional position fixing [24] (range and bearing measurements are different measurement types):

$$\text{Range:} \qquad R_i = \sqrt{(X-X_i)^2+(Y-Y_i)^2} \tag{4.1}$$

$$\text{Bearing:} \qquad \theta_i = \tan^{-1}\left(\frac{X-X_i}{Y-Y_i}\right) \tag{4.2}$$

where (X,Y) is the user position and (X_i,Y_i) is the position of transmitting station i. R_i is the distance between the user and station i (placing the user somewhere on a circle with respect to the station), and Θ_i is the angle between the user and station i with respect to North (placing the user somewhere on a line with respect to the station). A hyperbolic line of position is also used in some navigation systems, and can be defined by measuring the time difference (TD) between the times of arrival of signals from two different stations. The equation for a TD is:

$$\text{TD:} \qquad TD_{ij} = \frac{b-R_i+R_j}{c} \tag{4.3}$$

where TD_{ij} is the time difference for stations i and j, b is the straight-line distance between the two stations, c is the speed of light, and R_i & R_j are described by Equation 4.1

The next step is to show how these different measurements can be used to calculate the navigation solution. First, it is necessary to derive linear approximations for Equations 4.1 - 4.3. An *a priori* estimate $(\hat{X},\hat{Y})$ is used to form a Taylor series expansion, of which only the first order terms are kept:

$$R_i = \hat{R}_i + \frac{\partial R_i}{\partial X}\Big|_{(\hat{X},\hat{Y})} \delta X + \frac{\partial R_i}{\partial Y}\Big|_{(\hat{X},\hat{Y})} \delta Y \tag{4.4}$$

The aircraft state estimate is $(\hat{X},\hat{Y})$, which is used to calculate the estimate of the distance to the station $(\hat{R}_i)$. Using this expansion, Equation 4.1 is linearized as follows:

$$\delta R_i = \begin{pmatrix} \dfrac{\hat{X}-X_i}{\hat{R}_i} & \dfrac{\hat{Y}-Y_i}{\hat{R}_i} \end{pmatrix} \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix} \tag{4.5}$$

A similar procedure is used to linearize the bearing and time difference equations:

$$\delta \theta_i = \begin{pmatrix} \dfrac{\hat{Y}-Y_i}{\hat{R}_i^2} & \dfrac{X_i-\hat{X}}{\hat{R}_i^2} \end{pmatrix} \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix} \tag{4.6}$$

$$\delta TD_{ij} = \begin{bmatrix} \dfrac{1}{c}\left(\dfrac{\hat{X}-X_j}{\hat{R}_j} - \dfrac{\hat{X}-X_i}{\hat{R}_i} \right) & \dfrac{1}{c}\left(\dfrac{\hat{Y}-Y_j}{\hat{R}_j} - \dfrac{\hat{Y}-Y_i}{\hat{R}_i} \right) \end{bmatrix} \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix} \tag{4.7}$$

Equations 4.5 - 4.7 relate user position updates to a range, bearing, or time difference measurement. Thus, any of these measurements can be used in the navigation solution. This is the result that allows interoperability of navigation systems such as GPS and Loran-C. In general, Equations 4.5 - 4.7 can be written as:

$$\delta y_i = h_i \begin{pmatrix} \delta X \\ \delta Y \end{pmatrix} \tag{4.8}$$

where y_i is a measurement value, and h_i is a row vector corresponding to that measurement. If all the measurements are included, Equation 4.8 becomes:

$$\delta \underline{Y} = H \delta \underline{\beta} \tag{4.9}$$

where $\underline{Y}$ is a vector containing all of the measurements and $\underline{\beta}$ is the user state vector. H is a matrix containing data related to the geometry of the transmitting stations with respect to the user, as given by the row vectors h_i. Equation 4.9 can be used to solve recursively for the user state vector.

If there is an unknown clock offset, for instance in a range measurement, another state must be included in $\underline{\beta}$. The measurement is now called a pseudorange:

$$PR_i = R_i + B \tag{4.10}$$

where B is the unknown clock offset. Linearizing Equation 4.10 results in a slightly altered measurement equation:

$$\delta PR_i = \begin{pmatrix} \frac{\hat{X} - X_i}{\hat{R}_i} & \frac{\hat{Y} - Y_i}{\hat{R}_i} & 1 \end{pmatrix} \begin{pmatrix} \delta X \\ \delta Y \\ \delta B \end{pmatrix} \tag{4.11}$$

More states can be added as necessary to solve for other information, such as velocity or acceleration. Hybrid GPS/Loran-C, for instance, requires that two clock offsets be solved for (one for GPS and one for Loran-C). Thus, five measurements are required to solve for the position of the user.

5. INTEGRATED NAVIGATION SYSTEM INTEGRITY

A Kalman filter should be used in conjunction with a "snapshot" batch estimator to provide FDI for integrated navigation systems. The Kalman filter is excellent for detecting step errors or large-sloped ramp errors. It uses the history of the user state to determine the reasonableness of the next set of measurement data based on estimator residuals [25]. The advantage is that, for a recursive estimator, no redundant measurements are required (see Section 1). The disadvantage is that a slowly growing error, such as a clock drift in a GPS satellite, can corrupt the navigation solution without a warning being issued to the user. A batch estimator has the advantage that it is independent of the history of the user state. It can be used to determine if inconsistency exists within one set of measurements (thus, the term "snapshot" estimator). In this way, a slowly growing error that might go undetected by a recursive estimator can be detected when the error becomes too large for accurate navigation. The disadvantage of a batch estimator is that extra signals are required to perform FDI. However, for integrated navigation systems, it is anticipated that an abundance of signals will be available, and therefore FDI will also be available.

An FDI scheme that uses a batch estimator can be characterized by several parameters, one of which is the allowable radial position error, R_p (also called the alarm threshold). The idea is that a fault will be declared whenever the horizontal radial position error exceeds R_p. In practice, only the estimated position is available to perform fault detection. Another specification is the probability of exceeding R_p. For detecting faults, several detection statistics (d_k) are calculated (one for each measurement), and if any one exceeds a pre-calculated detection threshold (T_D), then a fault is declared. The probabilities of missed detection (P_{MD}) and false alarm (P_{FA}) are also critical to the FDI algorithm. Of P_{MD}, P_{FA}, and R_p, any two can be fixed while the third can vary. There is justification for making P_{FA} and P_{MD} constants in the algorithm, letting R_p fluctuate. If this is the case, R_p becomes a function of the measurement geometry. With good geometry (small HDOP), the alarm threshold is small, and with poor geometry (large HDOP), the alarm threshold is large. The significance is that the FDI routine is geometry independent. This is especially important for GPS, since orbiting satellites can provide vastly different geometry to the user at different times during the day. It is also notable that changing R_p does not necessitate a change in the detection algorithm.

One proposed method for satisfying P_{FA} and P_{MD} for GPS is to average over all time/space points in the continental United States via simulation. This defines an average detection probability (on a per sample basis) for the algorithm. It is easy to implement in this manner - all it requires is exhaustive testing. However, an average detection probability guarantees that at certain locations there will be times when the detection probability is knowingly compromised [26]. This is clearly unacceptable for a sole means navigation system. A minimum detection probability must be guaranteed at *each* time/space point. This is more difficult to implement and analyze, but it agrees with current navigation systems, such as the Very High Frequency Omnidirectional Range system (VOR). Each VOR transmitter has a guaranteed level of integrity, independent of other VOR stations.

Another important parameter in the FDI algorithm is the standard deviation of the measurement noise (σ). As was discussed in Section 2, a large σ increases the alarm rate. Figure 22 shows that both P_{FA} and P_{MD} are a function of σ, as shown for a constant detection threshold of 157 meters. A typical value of σ for GPS is 32 meters, for which P_{FA} and P_{MD} are highly sensitive. Reference [27] shows that an uncompensated 10% increase in σ can cause an order of magnitude increase in alarm probability. For GPS, the major error source is Selective Availability (SA). Selective Availability is the intentional degradation of the signals received by civil users, which increases σ. Varying levels of SA can be expected. Although mathematical tools are available for dealing with varying noise levels [28], the FDI algorithm would work much better without SA.

It is important to clarify the difference between estimation space and parity space for the navigation problem. As mentioned in Section 2, an alarm in parity space can mean a false alarm or a correct detection in estimation space. Parity space is the domain in which FDI is carried out; estimation space is the domain in which the user position is solved for. Also, the absence of an alarm in parity space can mean either normal operation or a missed detection in estimation space. To illustrate, consider a set of 5 GPS satellites. The probability of a bias error in one of the satellites is 10^{-6} [29]. Figure 23 shows the two cases - one without a bias and one with a bias. Without a bias, the alarm threshold, R_p, should be set such that the probability of exceeding R_p is P_{MD} (let P_{MD} be 10^{-9} for this case). P_{MD} is actually given by $(1 - 10^{-6})(10^{-9})$, which is still roughly 10^{-9}. Now let there exist a bias in one of the measurements. This amounts to moving

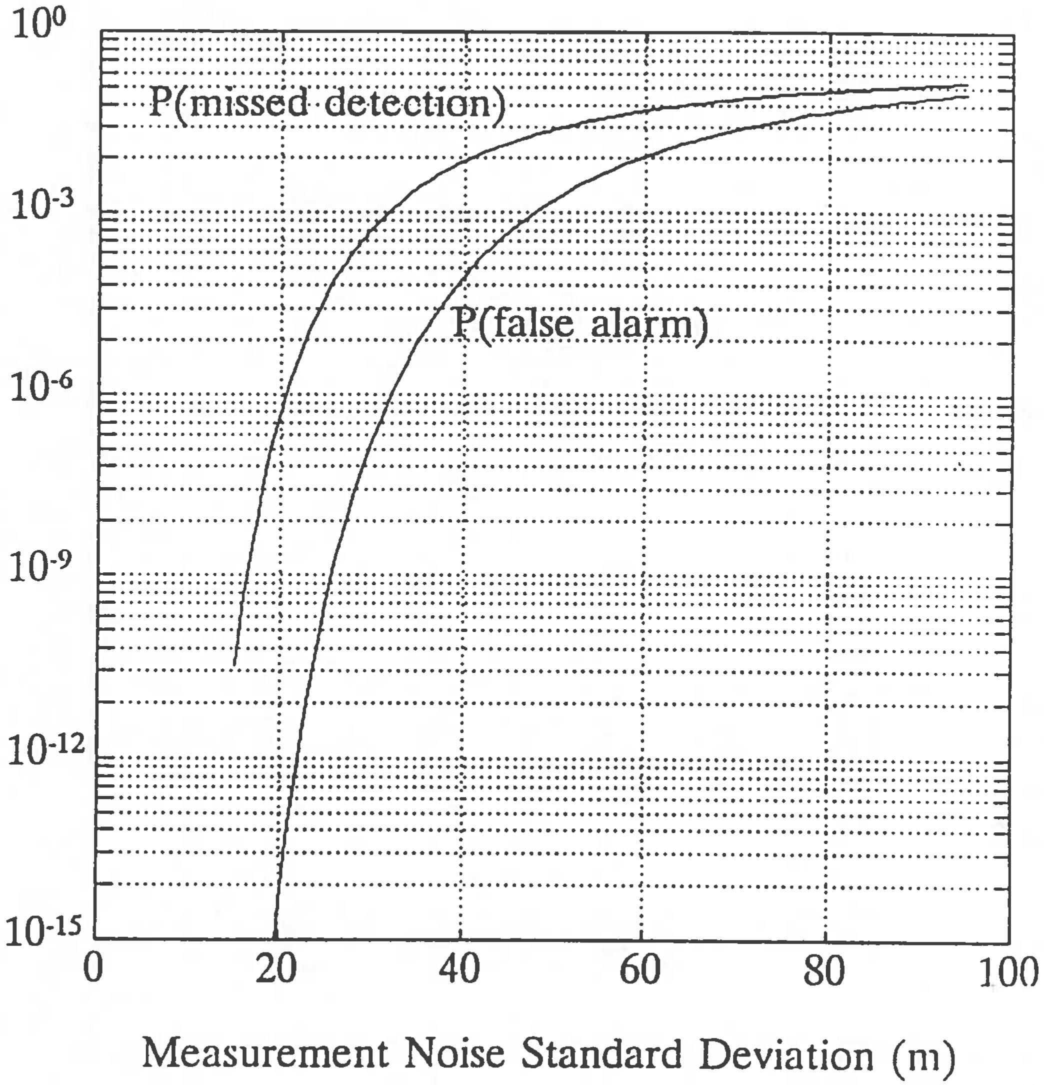
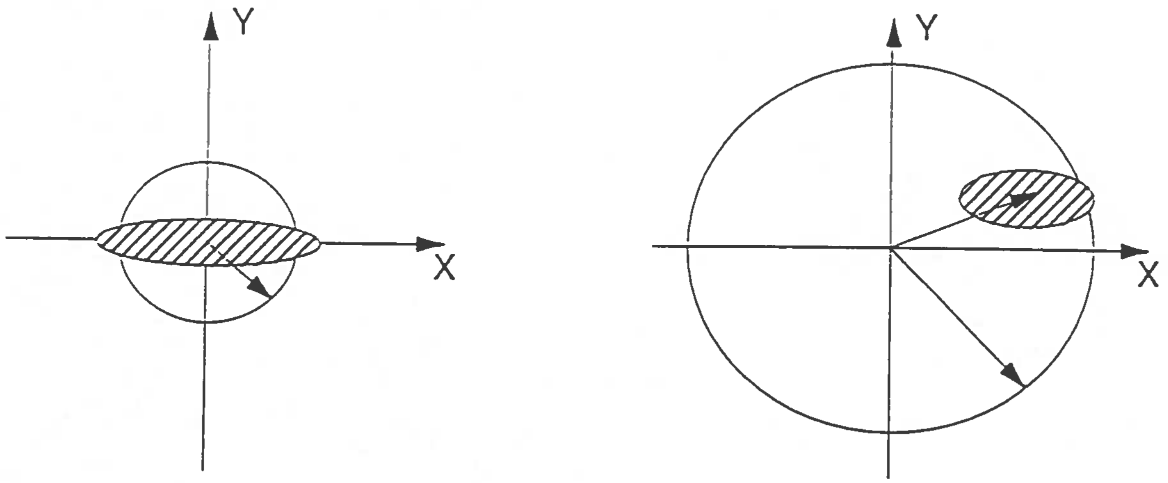


Figure 22 P_{FA} and P_{MD} as a Function of Measurement Noise (T_D = 157 meters).

ESTIMATION SPACE

No Measurement Bias Error
Probability = 0.999999

Measurement Bias Error
Probability = 0.000001



P(Outside Circle) = 10^{-9}

$P_{MD} = (1 - 10^{-6})\, 10^{-9} \simeq 10^{-9}$

P(Outside Circle) = 10^{-3}

$P_{MD} = (10^{-6})\,(10^{-3}) = 10^{-9}$

Figure 23 Example of Estimation Space with and without a Measurement Bias.

the uncertainty "cloud" around the user away from the origin by a certain distance. Since the probability of being in this state is 10^{-6}, R_p can be set such that the probability of exceeding it is 10^{-3}. As shown in the figure, this still results in the same P_{MD} of 10^{-9}. In summary, the probability of a missed detection is given by:

$$\begin{aligned} P_{MD} \quad = \quad & P(\text{no bias}) \text{ x } P(\text{outside } R_{noise} \mid \text{no bias}) + \\ & P(\text{bias}) \text{ x } P(\text{outside } R_{noise} + R_{bias} \mid \text{bias}) \end{aligned}$$

As a conservative approach, the larger threshold is usually used ($R_{noise} + R_{bias}$), thus protecting against a bias. The point is that, when protecting against a bias, it is not necessary to set R_p such that the probability of exceeding it is 10^{-9} because the likelihood that there is a bias is small (10^{-6}).

6. PROTOTYPE HYBRID GPS/LORAN-C RECEIVER

6.1 Hardware Configuration

Figure 24 shows a block diagram of the prototype hybrid GPS/Loran-C receiver. The GPS receiver is a Motorola Eagle, which is capable of tracking up to 4 satellites (4 channels). The Loran-C receiver is an Advanced Navigation, Inc. (ANI) model 5300, which is capable of tracking up to 8 Loran-C transmitters (8 channels). The navigation solutions provided by these receivers are not used in the hybrid solution. Rather, the raw measurement data obtained from the two receivers is used to calculate an integrated navigation solution. The two receivers are interfaced to a micro-computer (model AT) through two serial communication ports. This is the mechanism by which the measurement data can be collected, processed, and stored. Once the navigation solution has been calculated, appropriate guidance information is displayed to the pilot on a course deviation indicator (CDI), a King Radio/Bendix, model KI 206. The CDI is interfaced to the micro-computer through the parallel (printer) port via an interface designed and implemented at the Avionics Engineering Center. The prototype receiver requires an operator to observe the status screen as displayed on the micro-computer monitor. The operator determines the desired flight path by selecting waypoints stored in an initialization file. The CDI indicates the cross-track error between the calculated position and the desired flight path.

6.2 Software Algorithms

The raw measurements are input to the hybrid GPS/Loran-C software, which is executed in real-time. Thus, the algorithm calculates the navigation solution each time a set of measurements is collected from the two receivers. This allows for the timely display of guidance information to the pilot on board the aircraft. The actual algorithm is outlined as follows [30]:

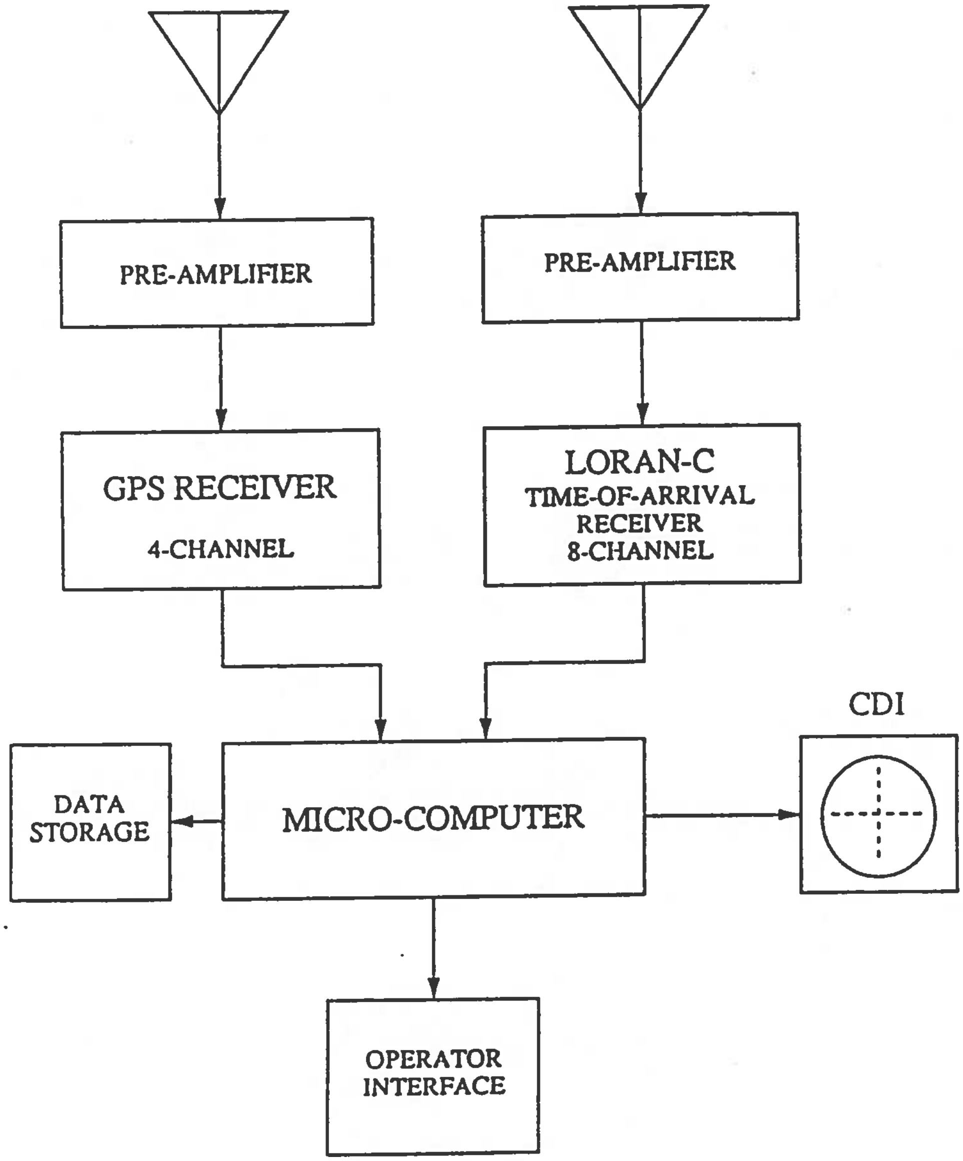


Figure 24 Hardware Configuration of the Prototype Hybrid GPS/Loran-C Receiver.

```
initialization
WHILE in operation
        DO once per second
                check for keyboard input data
                IF keyboard input data
                        process keyboard data
                END
                check for GPS and Loran-C measurement data, and
                        request Loran-C data
                IF sufficient data
                        calculate position
                        determine integrity
                END
                update CDI and status screen
                store all relevant data
        END
END
system shut-down
```

The storing of "all relevant data" includes a file containing the raw measurement data as collected from the two receivers. This is the data used for subsequent testing of the fault detection and isolation algorithm as discussed in the next section.

7. LABORATORY TESTING OF THE FAULT DETECTION AND ISOLATION ALGORITHM USING GPS/LORAN-C FLIGHT DATA

7.1 Flight Test Data

Flight test data collected on August 23, 1990 was used for evaluating the fault detection and isolation algorithm. The ground track of this flight is shown in Figure 25. The flight lasted approximately 52 minutes and was conducted in the vicinity of the Ohio University Airport in Albany. The fluctuation in the ground track is due to failures injected during the flight. The pilot was instructed to follow the course deviation indicator; therefore, the aircraft was flown slightly off course during the simulated malfunctions. The discontinuity in the flight track is due to a shut-down that was used to save the first part of the flight, and to switch satellites. During the initial section of the flight, one of the satellite signals was not being received, so it was replaced by a satellite with a higher elevation angle with respect to the horizon. During this test, the GPS/Loran-C software used up to 4 GPS satellites and up to 3 Loran-C transmitters resulting in a total of 7 measurements. During the first portion of the flight, only 6 measurements were available. After the discontinuity in the data (just prior to arrival at Yellow Bud VOR), 7 measurements were available for most of the second portion of the

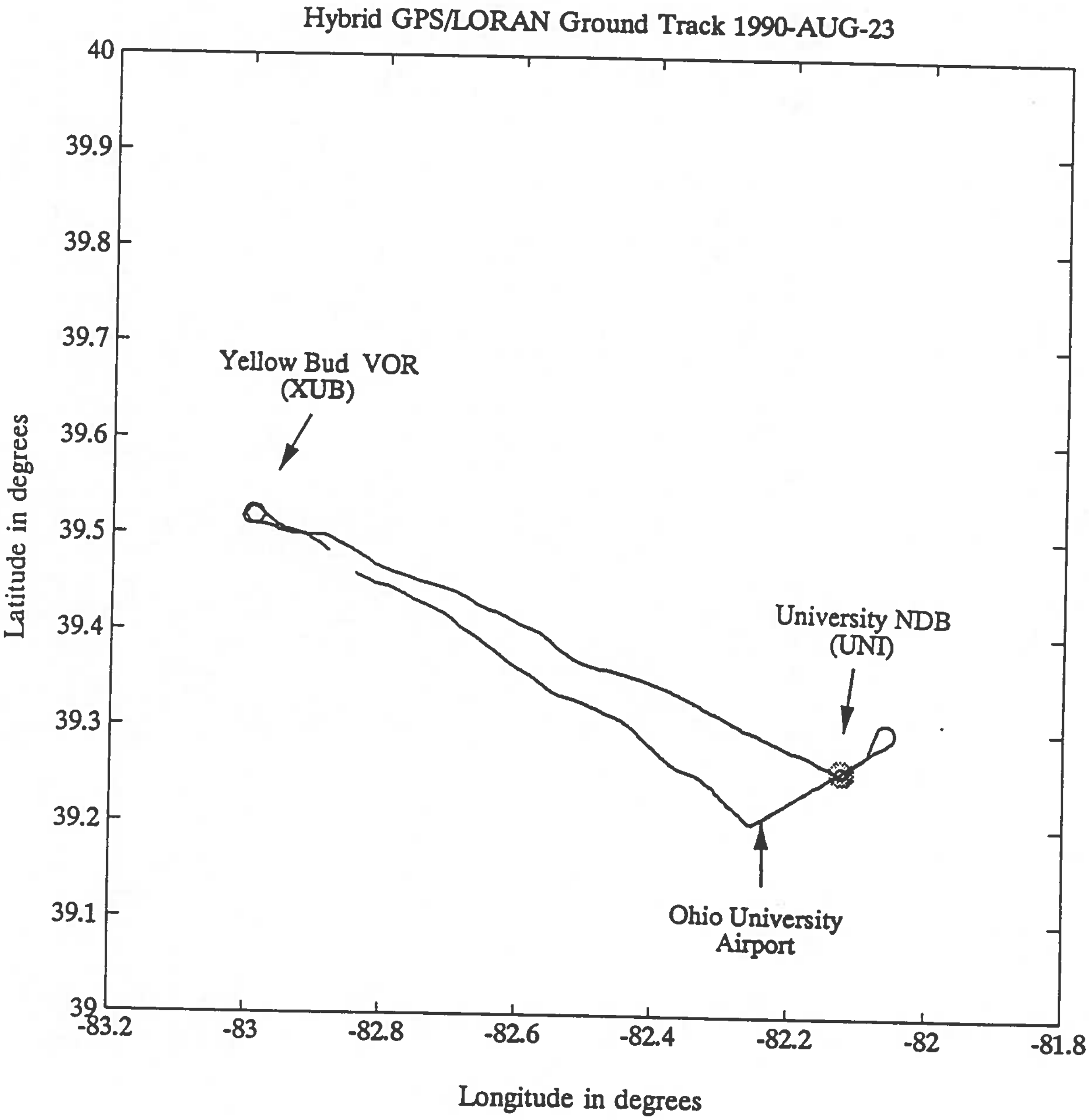


Figure 25 Ground Track of Flight Test, August 23, 1990.

flight. The FDI algorithm requires 2 redundant measurements and therefore requires measurements from 7 sources. For this reason, the flight data collected during the return to Ohio University Airport from Yellow Bud VOR was used to test the FDI algorithm.

7.2 Summary of the Fault Detection and Isolation Algorithm

Tables 3 and 4 provide a summary of the equations that form the Fault Detection and Isolation algorithm. Although σ has been specified for GPS to be approximately 32 meters, it is difficult to define σ for the hybrid GPS/Loran-C system. For example, there are Loran-C propagation errors that require an accurate error model. Thus, for simplicity in testing the FDI algorithm, the detection threshold, T_D, was chosen rather than calculated. The detection threshold, T_D, should be chosen such that an error is detected relatively fast, but the detection threshold should be large enough such that the isolation algorithm can correctly identify the faulty sensor. A small detection threshold means that the measurement axes are separated by a smaller distance at a radius of T_D from the origin, making it more likely that the wrong sensor will be isolated as the faulty one.

Once an appropriate T_D has been chosen, the algorithm calculates H and performs the QR factorization as outlined in Tables 3 and 4. Once Q_p has been found, the fact that it is a 2 x 7 matrix is taken advantage of. To maximize the error visibility for a particular sensor, the (2,i) element of Q_p is set to zero by using a Givens rotation (see Equation 7.1). This does not change the vector length of the rows of Q_p. For example the detection statistic d_3 would be calculated as follows:

$$
\begin{gathered}
\underline{p} = Q_p(-\underline{e} - \underline{b}) \\
\underline{p} = \begin{pmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \end{pmatrix}(-\underline{e} - \underline{b}) \\
G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \quad \text{the Givens rotation} \\
GQ_p = \begin{pmatrix} * & * & * & * & * & * & * \\ * & * & 0 & * & * & * & * \end{pmatrix} \\
\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} * & * & * & * & * & * & * \\ * & * & 0 & * & * & * & * \end{pmatrix}(-\underline{e} - \underline{b}) \\
d_3 = p_1
\end{gathered}
\tag{7.1}
$$

where s represents the sine of an angle and c represents the cosine of an angle. The

$$T_D = \sigma\sqrt{2}\,\mathrm{erfc}^{-1}(P_{FA})$$

$$\mu_M = T_D + \sigma\sqrt{2}\,\mathrm{erfc}^{-1}(2P_{MD})$$

$$\underline{y} = H\underline{\beta};\quad H = QR$$

$$\begin{pmatrix} Q_\beta \\ --- \\ Q_p \end{pmatrix} = Q^{T};\quad \begin{pmatrix} U \\ --- \\ 0 \end{pmatrix} = R$$

$$\underline{\beta} = U^{-1}Q_\beta\underline{y}$$

$$\underline{p} = Q_p\underline{y};\quad Q_p = (q1 \vdots q2 \vdots \cdots \vdots q_{n-5})$$

$$b_i = \frac{\mu_M}{|q_i|}$$

$$\begin{pmatrix} \bar{x}_i \\ \bar{y}_i \\ \bar{z}_i \\ \bar{t}_{GPS} \\ \bar{t}_{LC} \end{pmatrix} = U^{-1}Q_\beta \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$R_{bias} = \max(\sqrt{\bar{x}_i^2 + \bar{y}_i^2})$$

$$R_{noise} = \sigma\sqrt{2}\,\mathrm{erfc}^{-1}(P_A)\,\mathrm{HDOP}$$

$$R_p = R_{noise} + R_{bias}$$

Table 3 Summary of Equations used in Fault Detection and Isolation.

σ	measurement noise standard deviation
P_{FA}	probability of a false alarm
T_D	detection threshold in parity space
P_{MD}	probability of a missed detection
μ_M	minimum required bias in parity space for detection
Q_p	relates bias error in parity space to bias in measurement space

b_i	minimum detectable bias in measurement space
	if $n = 6$, $q_i = Q_p$
	if $n > 6$, $q_i = f\{Q_p\}$

$U^{-1}Q_\beta$	relates measurement errors to solution errors
P_A	Probability of Exceeding R_p
R_p	protection range in solution space
HDOP	horizontal dilution of precision

Table 4 Description of Symbols in Table 3.

length of row one of Q_p is still one because a rotation does not change the length of a vector. Therefore, the visibility of noise and bias errors has been maximized because by increasing the value of element (1,3), and the other elements in row one are necessarily decreased in value. A similar procedure is followed to find the other six detection statistics, d_k ($k = 1, 2, 4, 5, 6, 7$). Thus, seven different detection statistics are compared against T_D. One advantage of detecting faults this way is that the statistics of p_1, and therefore each d_k, remain Gaussian. Also, a gain in signal-to-noise ratio is obtained for a given error [31]. For future implementation, the algorithm will be generalized to accommodate a parity space of more than two dimensions. It is anticipated that Householder transformations will be used to zero out all but the first element in a column of Q_p.

Once a fault has been detected, the isolation algorithm is triggered. The algorithm finds the angle between the parity vector and each measurement axis. The axis with which the parity vector makes the smallest angle corresponds to the faulty measurement. Figure 26 is an example of the measurement axes in parity space. The fact that two redundant measurements are available makes parity space two-dimensional, and the plots of parity space will be especially helpful when isolation of faults is shown in Section 8. However, isolation may prove to be somewhat difficult since all of the measurement axes are coplanar, resulting in an increased probability of wrong isolation. The same isolation approach can be used for parity space of more than two dimensions, except now a hyperangle will be calculated between the parity vector and each measurement axis. The isolation decision should be more accurate because the measurement axes will be spaced out in more dimensions.

7.3 Fault Detection and Isolation Software

The FDI algorithm has been implemented in FORTRAN. An outline of the algorithm is as follows:

```
Calculate the H matrix
Perform the QR factorization on H
Partition $Q^T$ to obtain $Q_p$
DO for each measurement
    Perform the Givens rotation on $Q_p$
    Calculate the parity vector
    Set $d_k$ equal to the first element of the parity vector
    Raise an alarm if $d_k > T_D$
END
IF an alarm was raised
  Calculate the parity vector (no Givens rotations)
    DO for each measurement
        Calculate the angle between the parity vector and
            the measurement axis
```

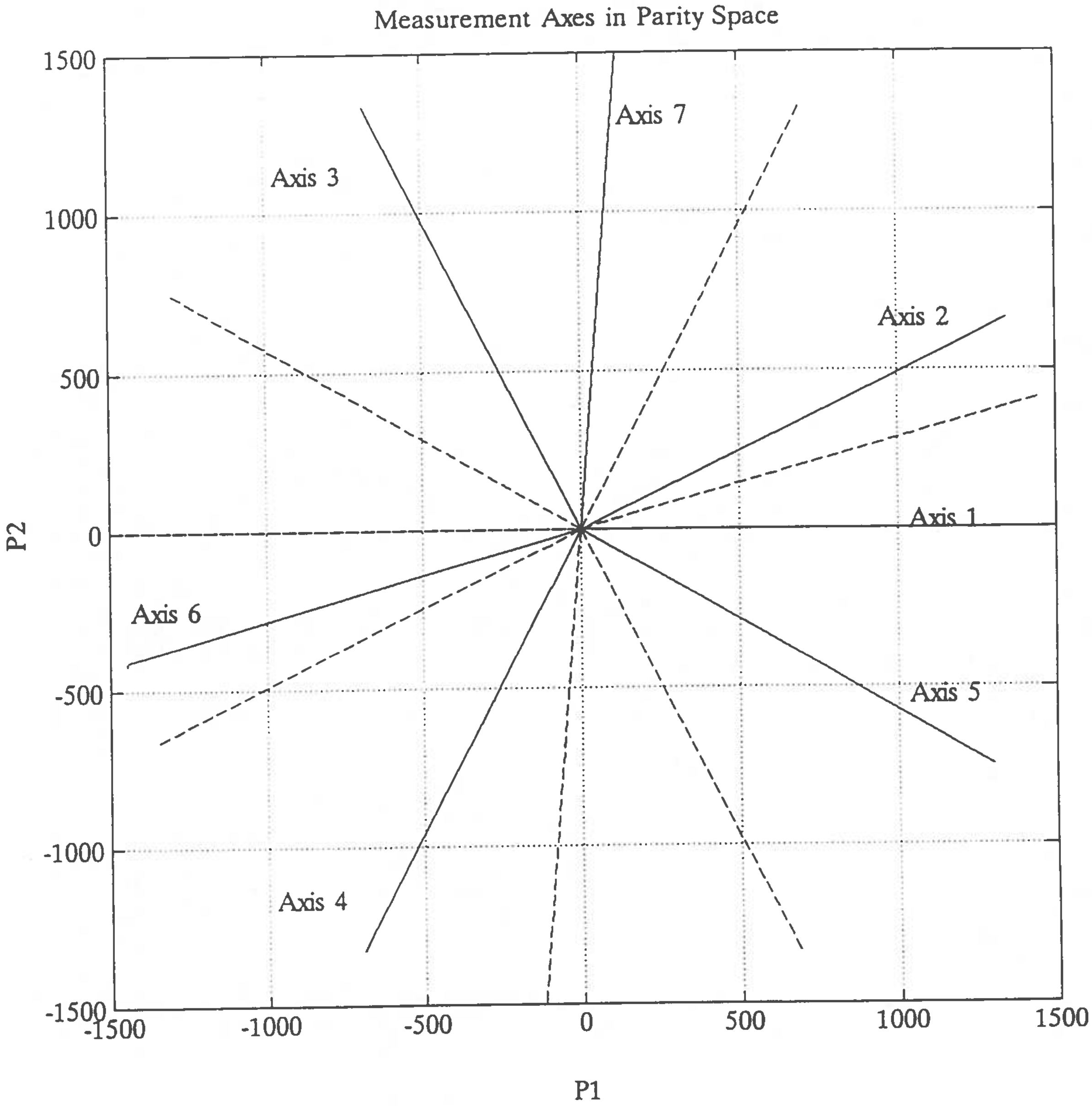


Figure 26 Example of Measurement Axes in Parity Space.

```
                    Store the measurement axis with the smallest
                            calculated angle
            END
            Output relevant data for plots
        END
```

The QR algorithm has also been implemented in FORTRAN. A series of Givens rotations is used to take the H matrix to upper diagonal form [32]. The QR subroutine returns R and Q^T. The subroutine that performs the Givens rotation is taken from the Linpack library. It is used to form a Householder transformation, which is successively called by the QR algorithm. The QR routine is as follows:

```
    Determine the number of Householder transformations needed
    Initialize $Q^T$ as an identity matrix
    BEGIN Main Loop
        Call the Householder subroutine
        Update R and $Q^T$
        Get ready for the next Householder transformation
    END
```

The Householder routine uses successive Givens rotations to zero out all but the first element in a column.

8. RESULTS OF THE FAULT DETECTION AND ISOLATION ALGORITHM USING REAL FLIGHT DATA WITH SIMULATED FAILURES

8.1 Simulation Parameters

T_D was chosen to be 1000 meters so that the isolation algorithm would identify the faulty measurement with a reasonable probability. The simulation was run 7 times, and an error was introduced at the same point during each run. A ramp error of approximately 5 m/s was injected into each of the 7 measurements (one error per simulation). A ramp error with a small slope was chosen because it is the most difficult to detect and isolate. Therefore, the FDI algorithm is put through a rigorous test. An example of the ramp error is shown in figure 27.

Some measure of horizontal radial position error was needed to determine whether or not there was an error in estimation space for each injected failure. The latitude and longitude as a function of time were recorded for each simulation. These were later compared to the latitude and longitude as calculated with no injected errors. Although this is not an absolute reference, the hybrid GPS/Loran-C position solution has an accuracy of approximately 200 meters [33]. Therefore, it is a fairly good reference for the simulations and is used to calculate a horizontal radial position difference in meters between the solution with the injected error and the solution without an injected error.

The 7 detection statistics were recorded for each simulation. The parity vector was also recorded, because the isolation algorithm bases its decision on the parity vector. The result is a trace in parity space growing outward as the ramp error gets larger. It is important to remember that, as implemented, the isolation decision is based solely on one set of measurements (a snapshot) and currently does not account for the time history of the parity vector.

8.2 Simulation Results -- Detection

Figures 28 through 34 indicate graphically the performance of the detection algorithm for each of the 7 failures. For each simulation, the injected error is plotted along with the horizontal radial position difference as described in Section 8.1. The 7 detection statistics are shown on each plot as well, with the solid line representing d_k vs. time for the measurement that contains the ramp error.

A fault is declared when any one of the detection statistics exceeds d_k, which occurred for most of the failures. One exception is the simulated failure in measurement 2, as depicted in Figure 29. None of the detection statistics reflects the fact that measurement 2 has a significant error in it. However, the horizontal radial position difference only grows to approximately 250 meters. This shows that large failures in certain measurements do not necessarily cause substantial position errors. The

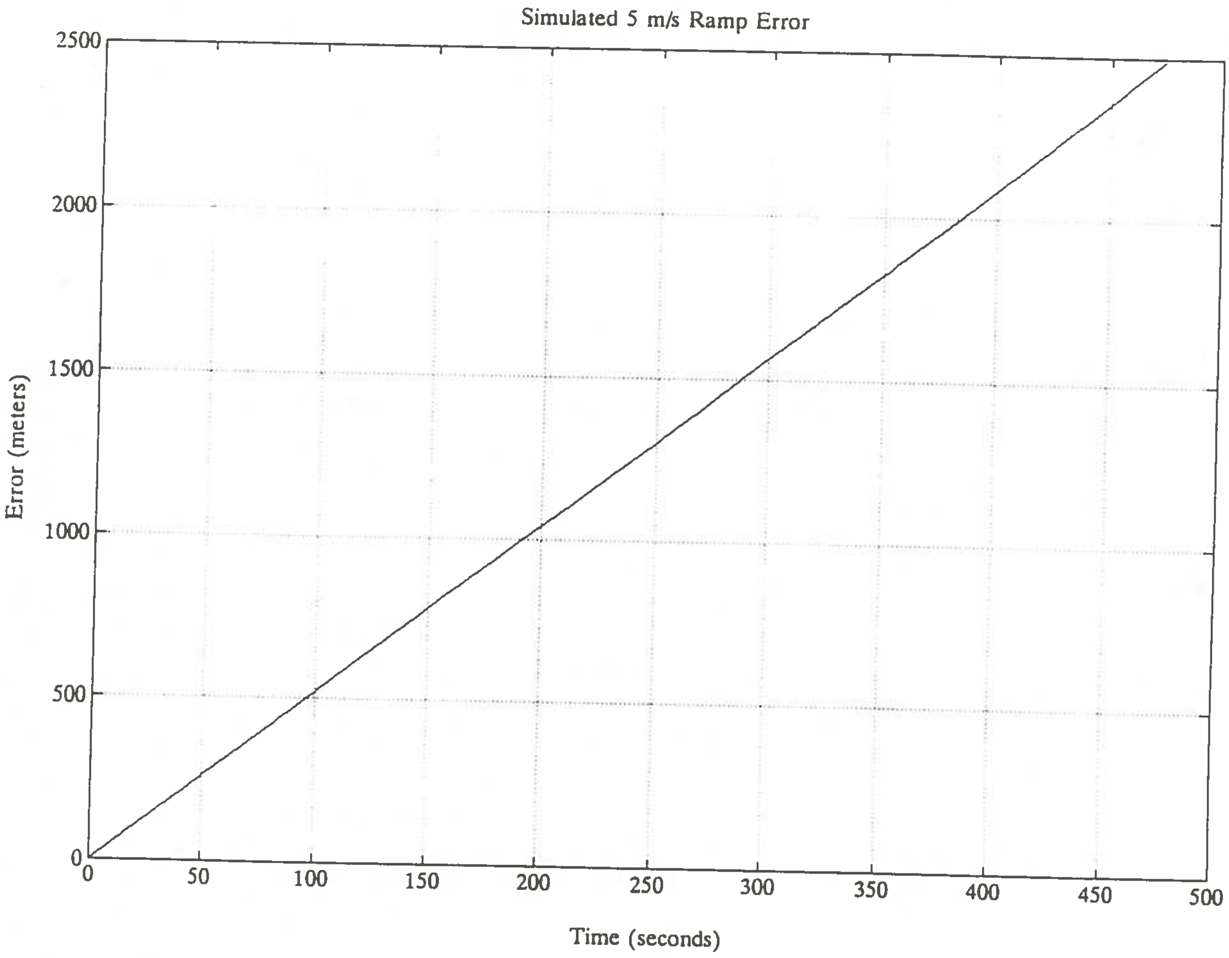


Figure 27 Simulated 5 m/s Ramp Error.

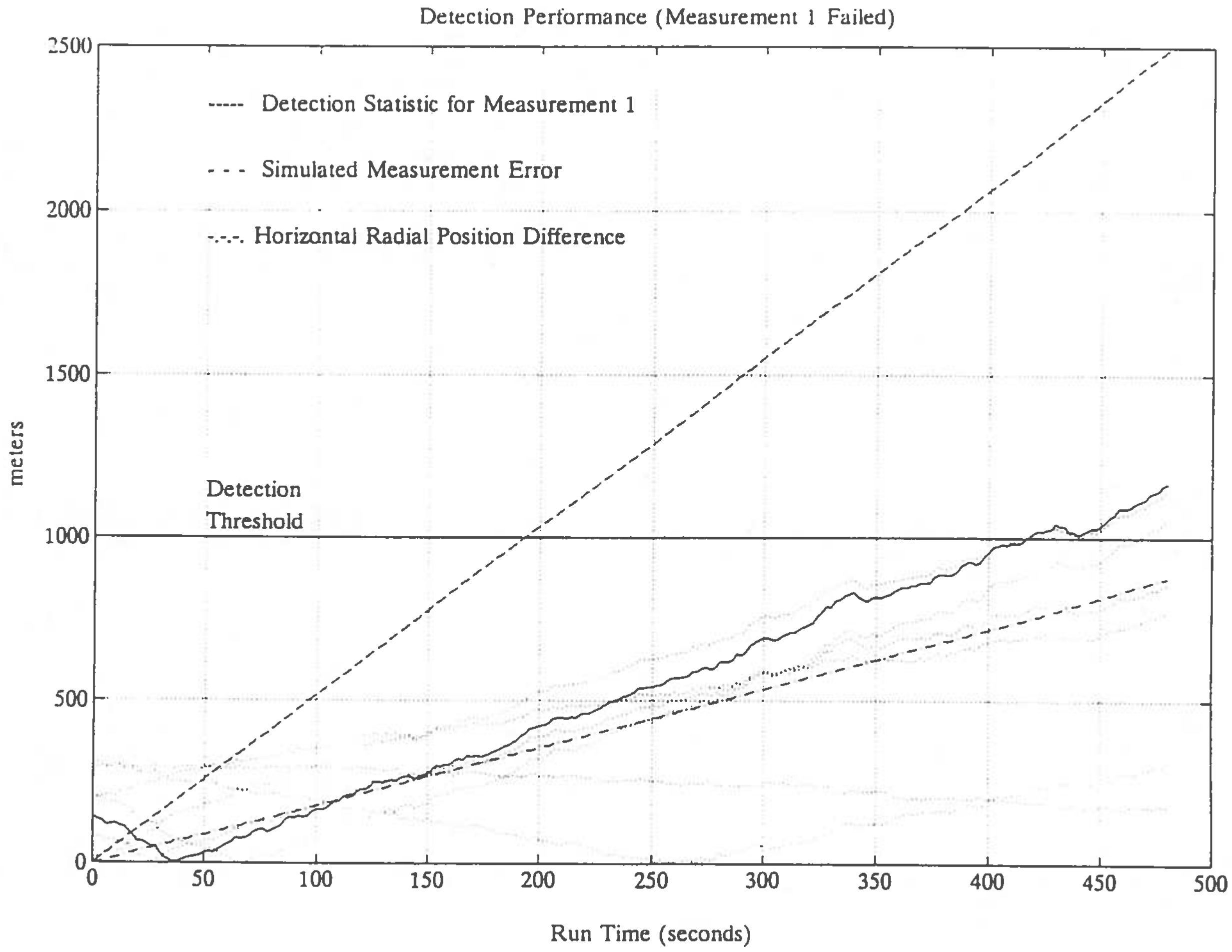


Figure 28 Detection Performance (5 m/s Ramp Error in Measurement 1).

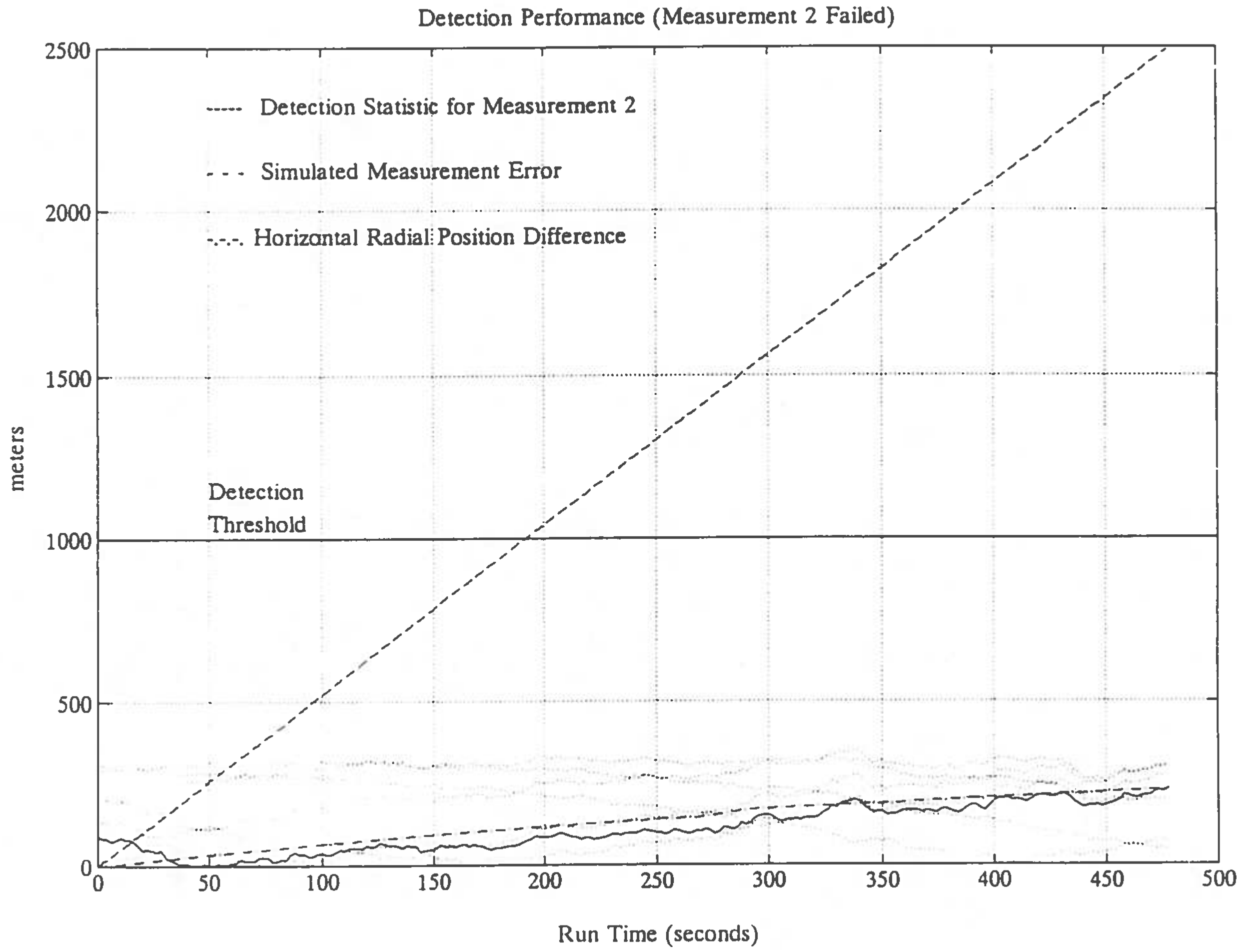


Figure 29 Detection Performance (5 m/s Ramp Error in Measurement 2).

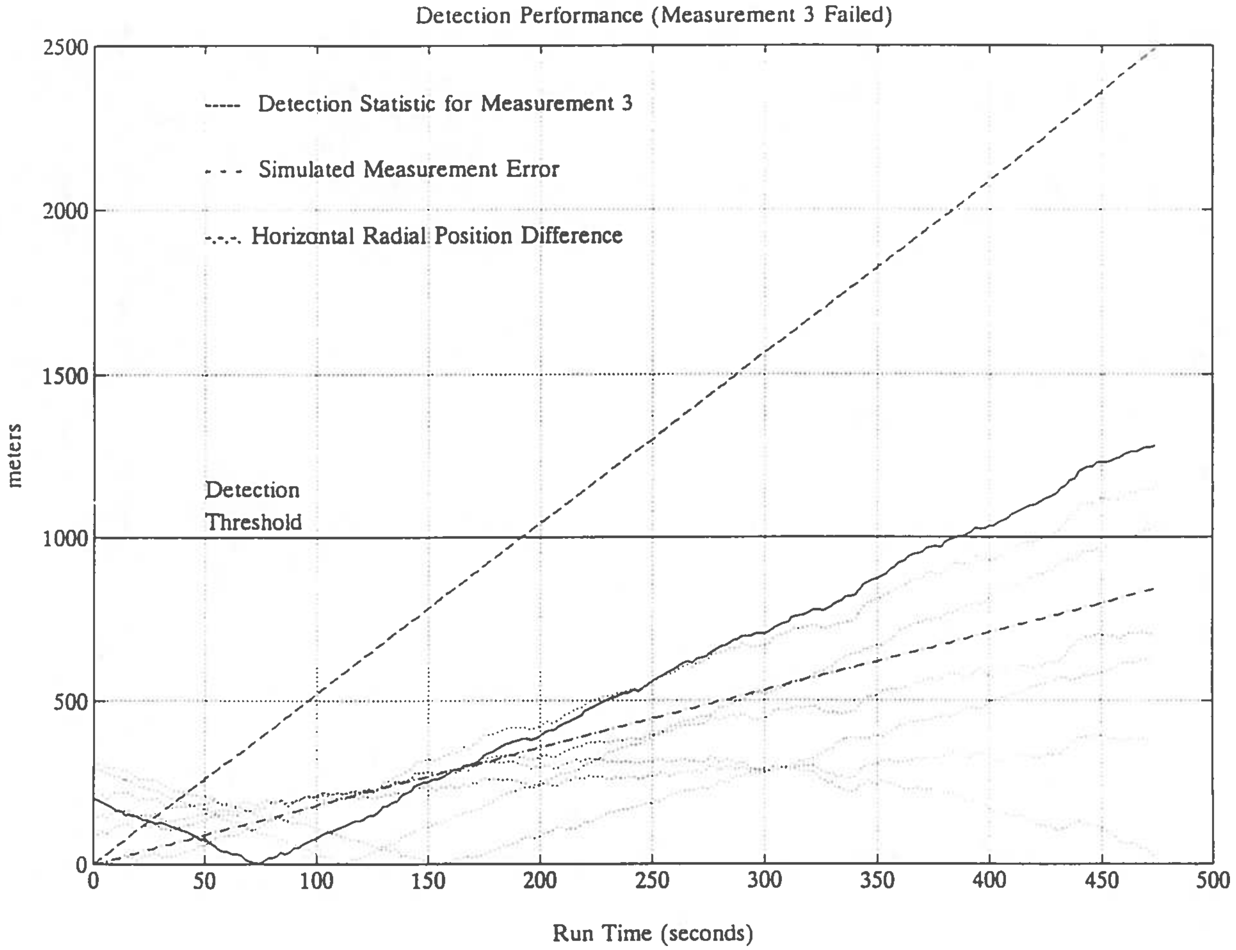


Figure 30 Detection Performance (5 m/s Ramp Error in Measurement 3).

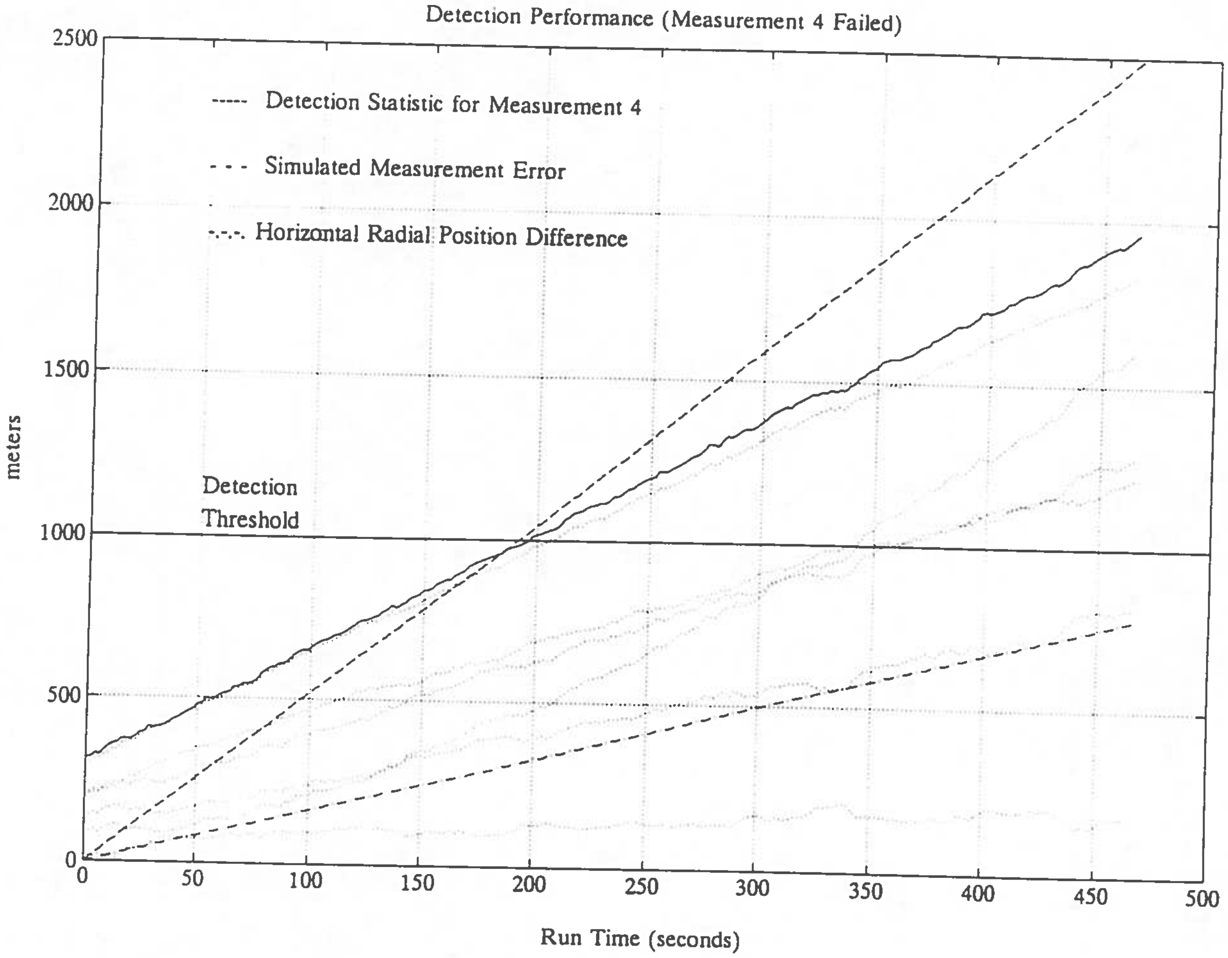


Figure 31 Detection Performance (5 m/s Ramp Error in Measurement 4).

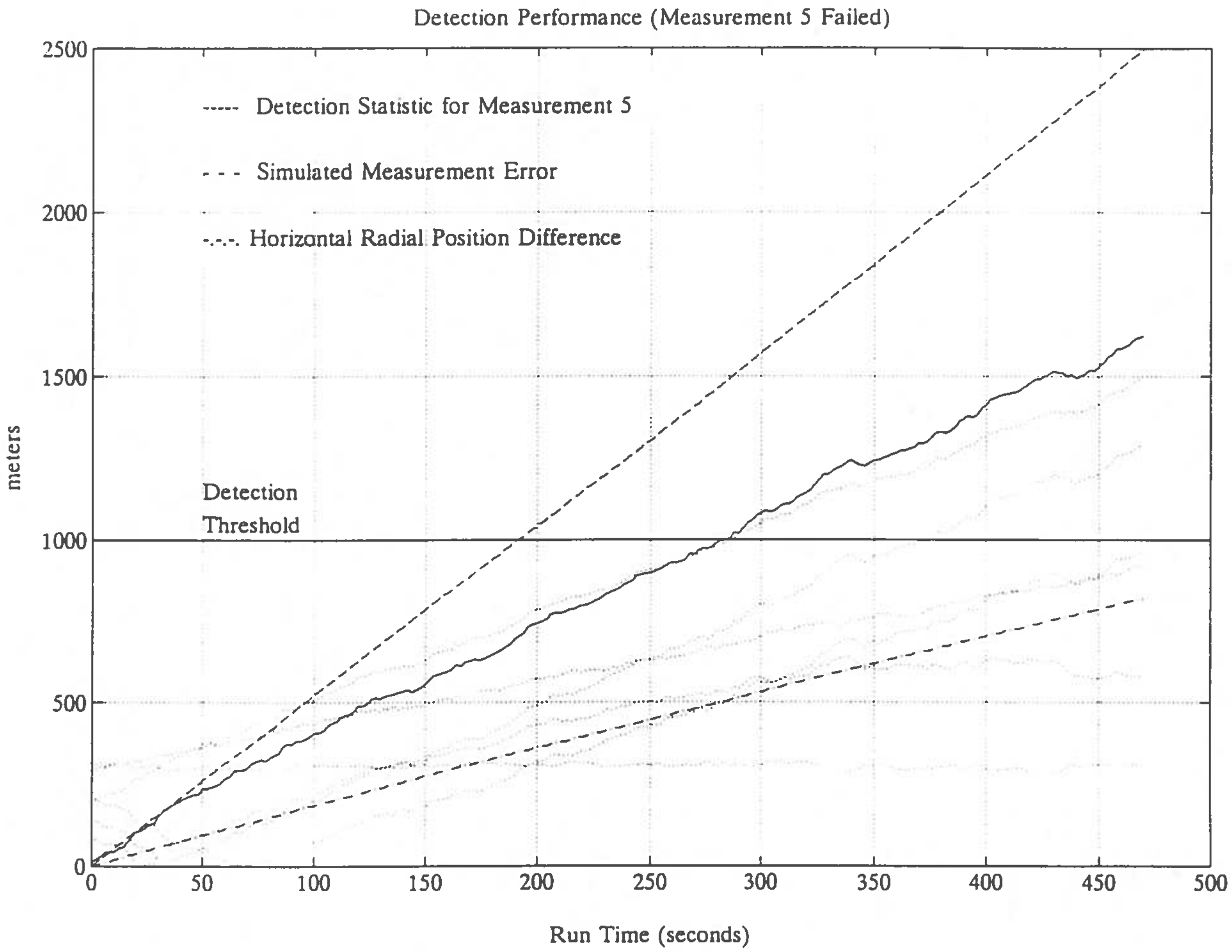


Figure 32 Detection Performance (5 m/s Ramp Error in Measurement 5).

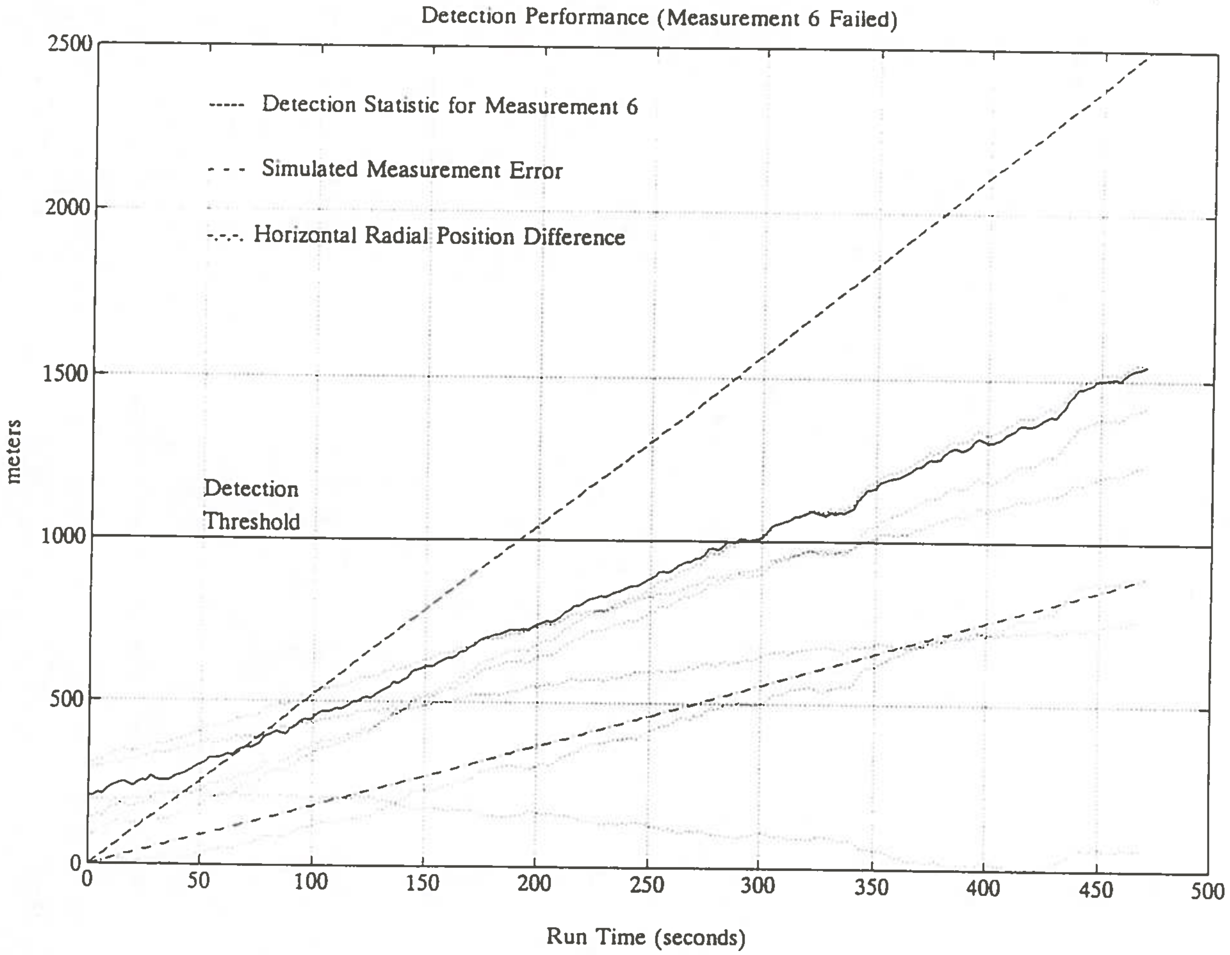


Figure 33 Detection Performance (5 m/s Ramp Error in Measurement 6).

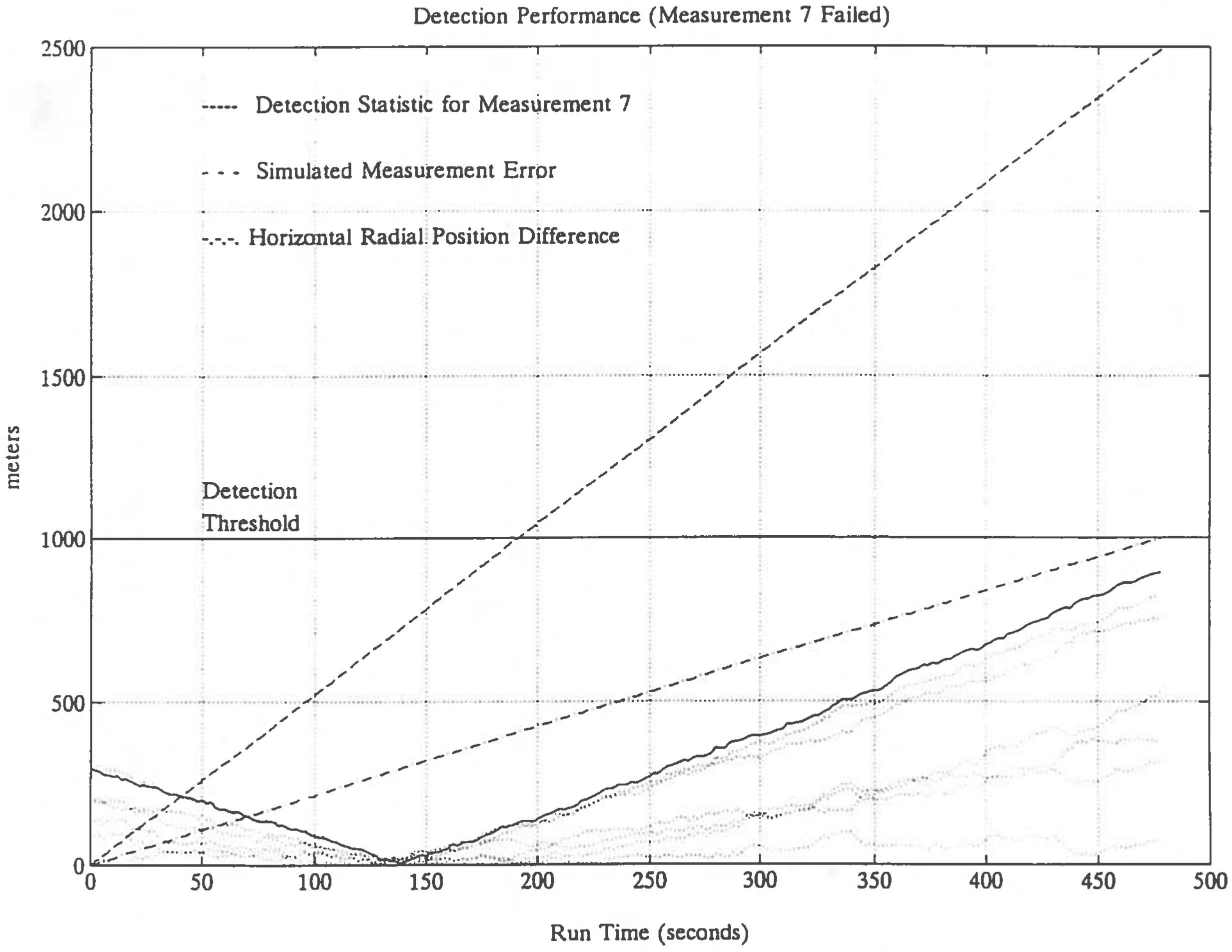


Figure 34 Detection Performance (5 m/s Ramp Error in Measurement 7).

horizontal radial position difference grows at about one tenth the rate that the error in measurement 2 grows. Although a large error in measurement 2 exists without detection, it is not considered a missed detection because the horizontal radial position difference is not very large. Thus, an undetected error is only considered to be a missed detection if the position error is significant in estimation space.

Another case in which the failure was not detected occurred when an error was simulated in measurement 7, which is shown in Figure 34. Note that, in this case, the horizontal radial position difference reaches 1000 meters at the end of the run. Because the detection threshold T_D is not exceeded by any of the detection statistics, a fault is not declared. This scenario could be considered to be a missed detection. However, it appears from the plot that had the simulation been extended, the error would have soon been detected, and an alarm would have been raised.

8.3 Simulation Results -- Isolation

Figure 35 contains traces of the parity vectors over time for each of the simulated failures. Each parity trace begins at the same point and extends outward in a direction dictated by the measurement which contains the error. Note that the axes are valid only for a given set of measurements (one instant in time). They are presented merely to gain insight into the behavior of the parity vector as the error grows in a particular measurement. In fact, the axes in parity space do not move very far in a short amount of time, with the exception of axis 2. This is apparent from the parity traces; only the parity trace for an error in measurement 2 curves significantly, which indicates that axis 2 rotates fairly quickly compared to the other axes.

Most of the errors were correctly isolated by using the final parity vector shown in Figure 35 for each of the simulations. Note that the errors in measurements 2 and 7 were not isolated because those are cases in which an alarm was not raised. Given the method of isolating by calculating the angle between the parity vector and each measurement axis, certain problems clearly exist. For example, the error in measurement 6, when compared to the axes in Figure 35, is incorrectly isolated to measurement 2. To fully understand the problem, it is necessary to examine the behavior of the measurement axes as the error in measurement 6 grows.

Figure 36 shows the parity trace along with the measurement axes at the moment the error in measurement 6 was first isolated (the moment when T_D was exceeded). The error was correctly identified to be contained in measurement 6. However, since axis 2 rotates so quickly, a short time later it forms a smaller angle with the parity vector than axis 6 does, as depicted in Figure 37. Until axis 2 rotates far enough away, the wrong sensor will continue to be isolated. Therefore, it is apparent that this method of isolation may not be highly effective. Upon further examination of Figure 35, it is apparent that a given error grows in the *direction* of the measurement axis associated

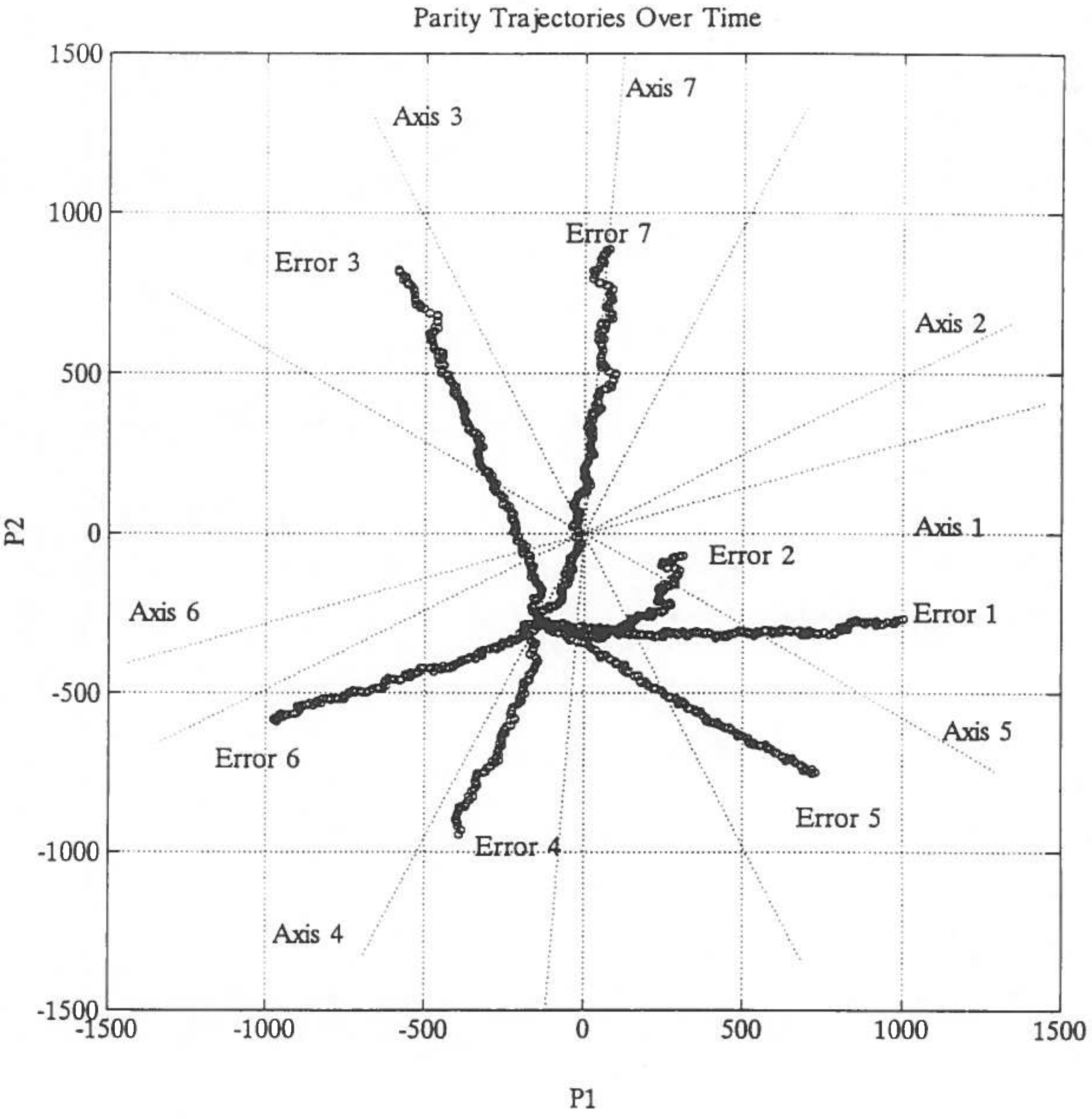


Figure 35 Trajectories of the Parity Vectors over Time for all 7 Simulated Measurement Errors.

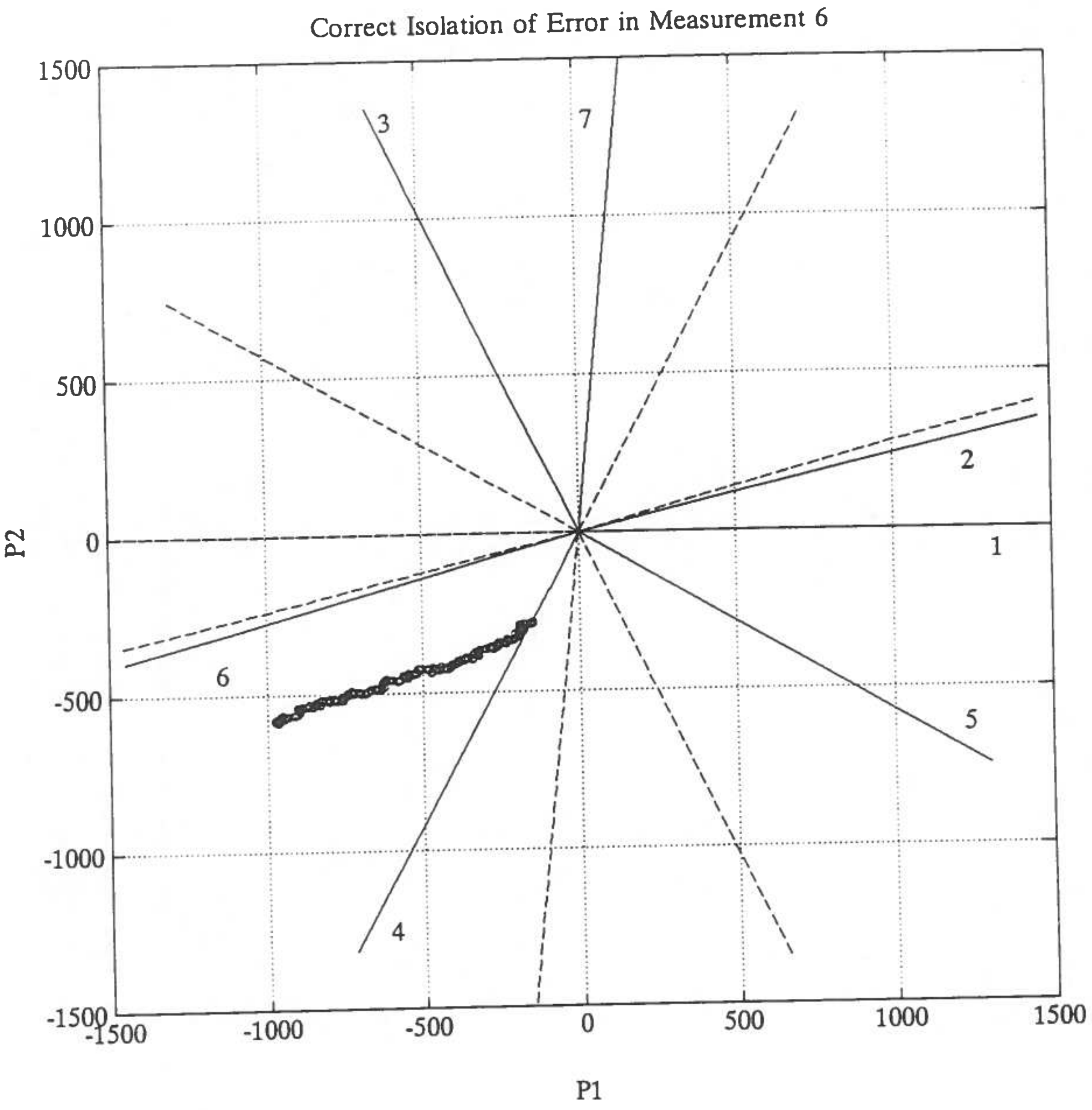


Figure 36 Correct Isolation of Error in Measurement 6.

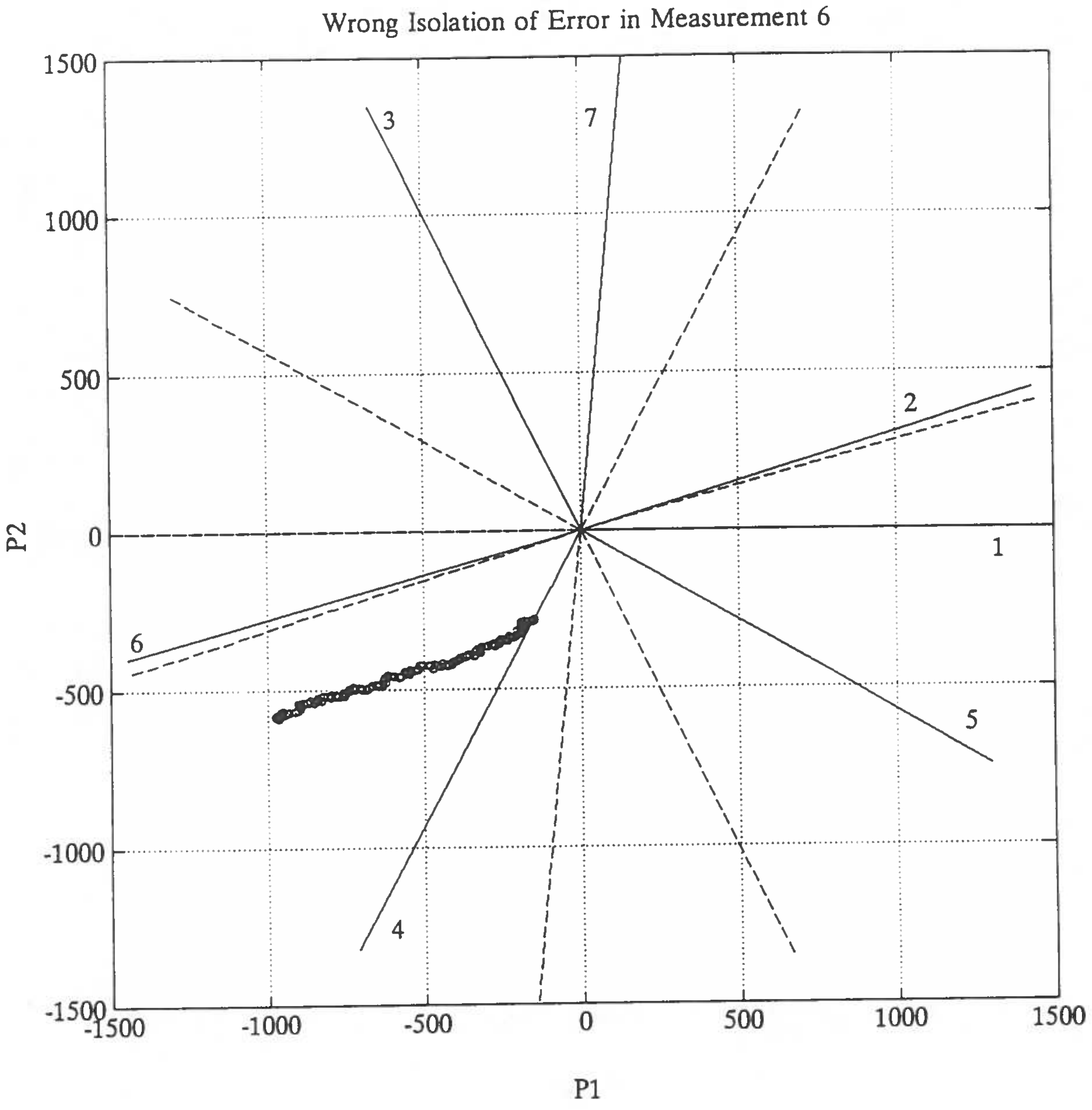


Figure 37 Wrong Isolation of Error in Measurement 6.

with the failed sensor. This information can be extremely valuable in correctly isolating failures. Rather than simply taking a "snapshot" of parity space for each set of measurements, it is beneficial to monitor the parity vector from one set of measurements to the next. That is, some type of time history of the parity vector should be kept. Figure 35 indicates that by calculating the perpendicular distance from the parity vector to each axis over time should result in approximately a straight line for the axis corresponding to the failed sensor. Consider the failure in measurement 5. By using only the last parity vector, it would be difficult to decide whether measurement 3 or measurement 5 contains the error. However, by examining the trace of the parity vector over time, it seems obvious that the error must be in measurement 5.

Figures 38 through 44 are plots of the perpendicular distance from the parity vector to the measurement axes for each simulated failure. The distance to the axis corresponding to the failed measurement is plotted as a solid line. The isolation decision is now based on finding the line with the smallest slope, which would indicate that the measurement error is growing in the direction of that axis in parity space. These plots clearly show which measurement contains the error for all cases in which a fault was detected. In the case of the error in measurement 2 (Figure 39), the parity vector remained relatively close to the origin; therefore, the distance to each measurement axis remained small. Still, though, the distance to axis 2 is fairly constant, which would indicate that an error may exist in that measurement. Although no error was detected in the simulation of an error in measurement 7, Figure 44 definitely indicates that an error exists in that measurement. An interesting situation occurs in the error simulation for measurement 6 (Figure 43. At a particular time (approximately 430 seconds) the parity vector is at a distance of zero meters from measurement axis 2 in parity space. A snapshot of parity space at that moment would indicate that the error belonged to measurement 2. However, Figure 43 clearly discounts that isolation decision; it is obvious from the time history of the parity vector that the error must be contained in measurement 6.

It appears that observing the parity vector over time is a very powerful tool in the isolation of faulty sensors. The parity vector should be stored over a period of time and used to recognize trends like those seen in Figures 38 through 44. The "snapshot" batch estimator is still applicable to fault detection, but the parity information should be saved to aid fault isolation, rather than discarded for each set of measurements.

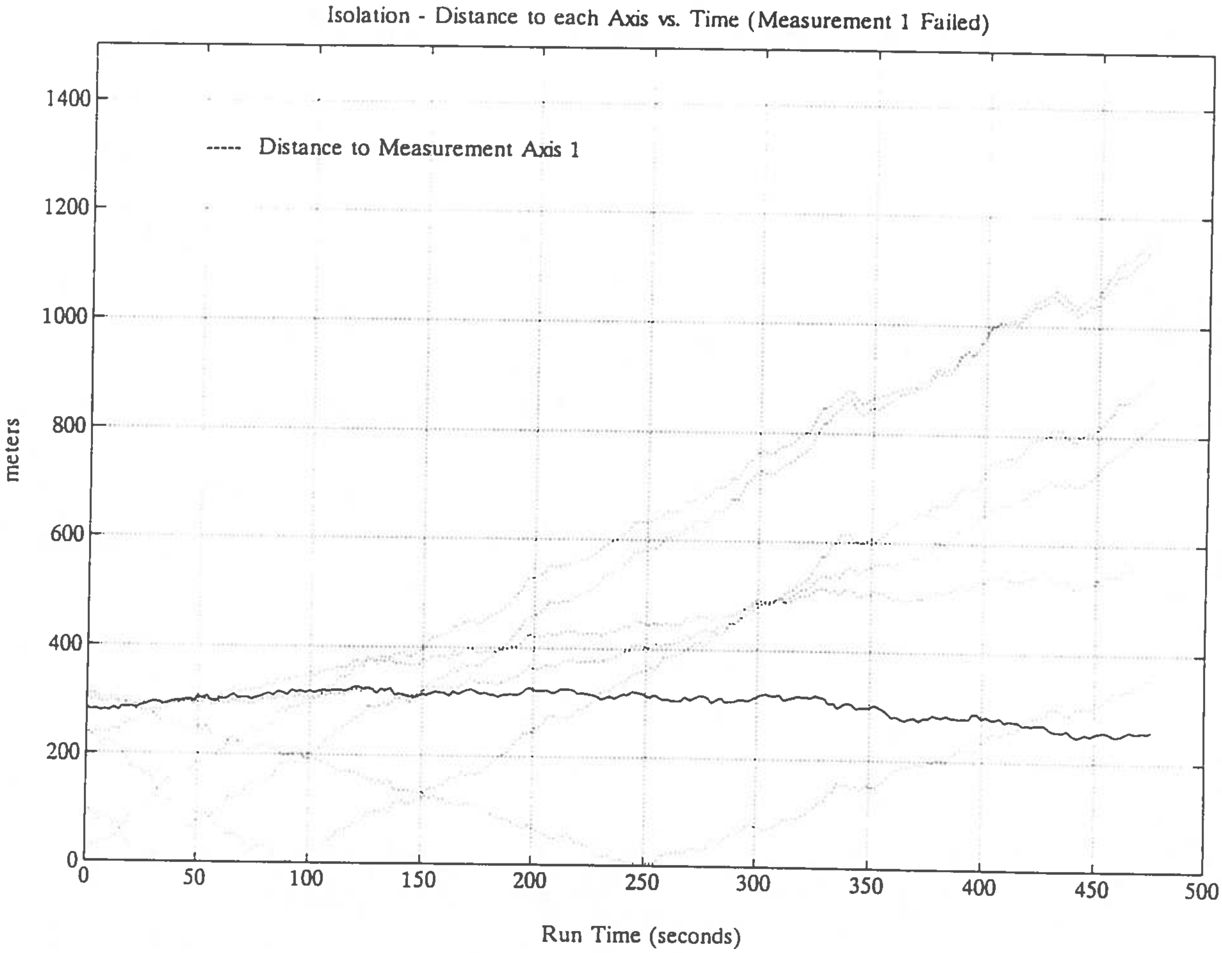


Figure 38 Isolation Performance (5 m/s Ramp Error in Measurement 1).

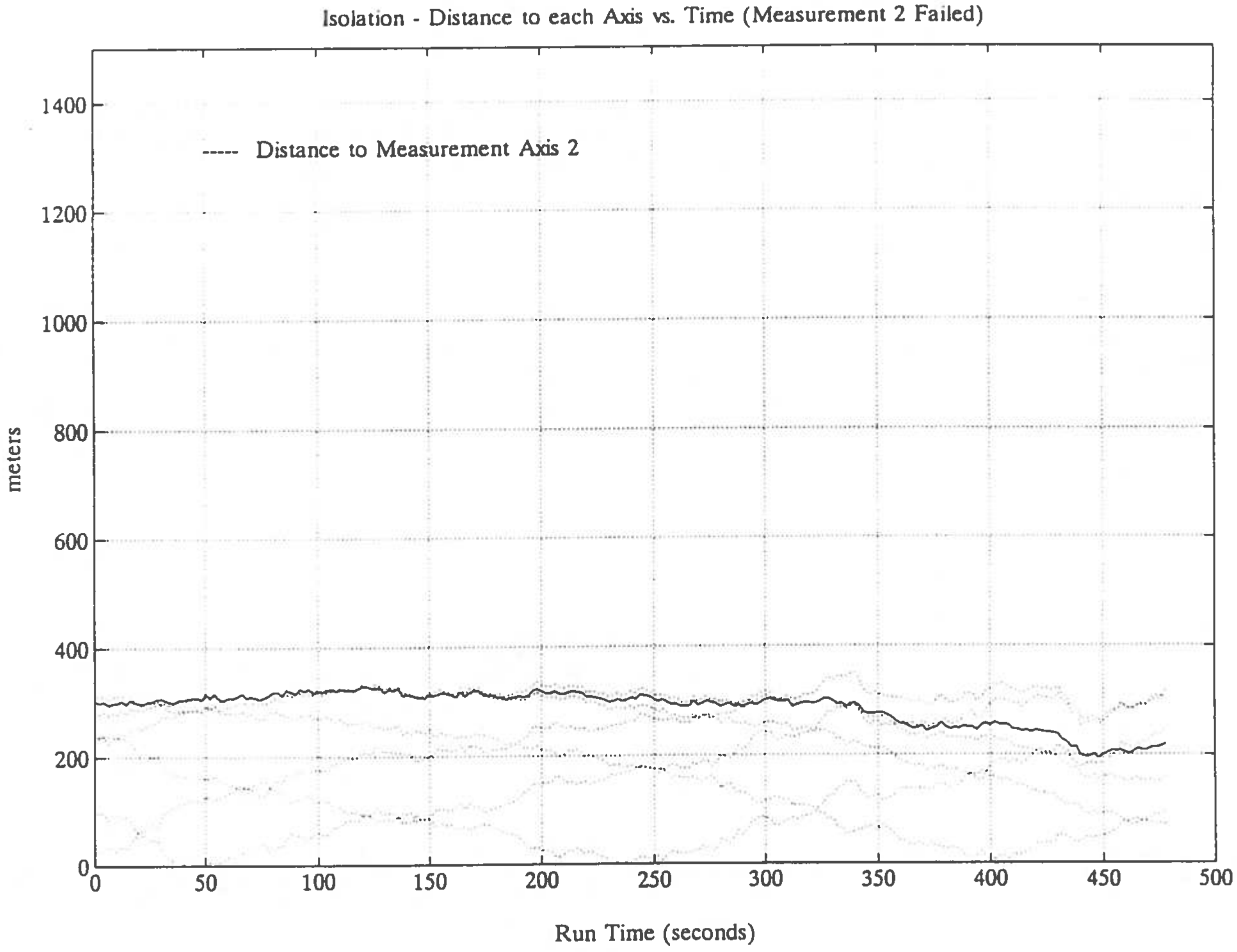


Figure 39 Isolation Performance (5 m/s Ramp Error in Measurement 2).

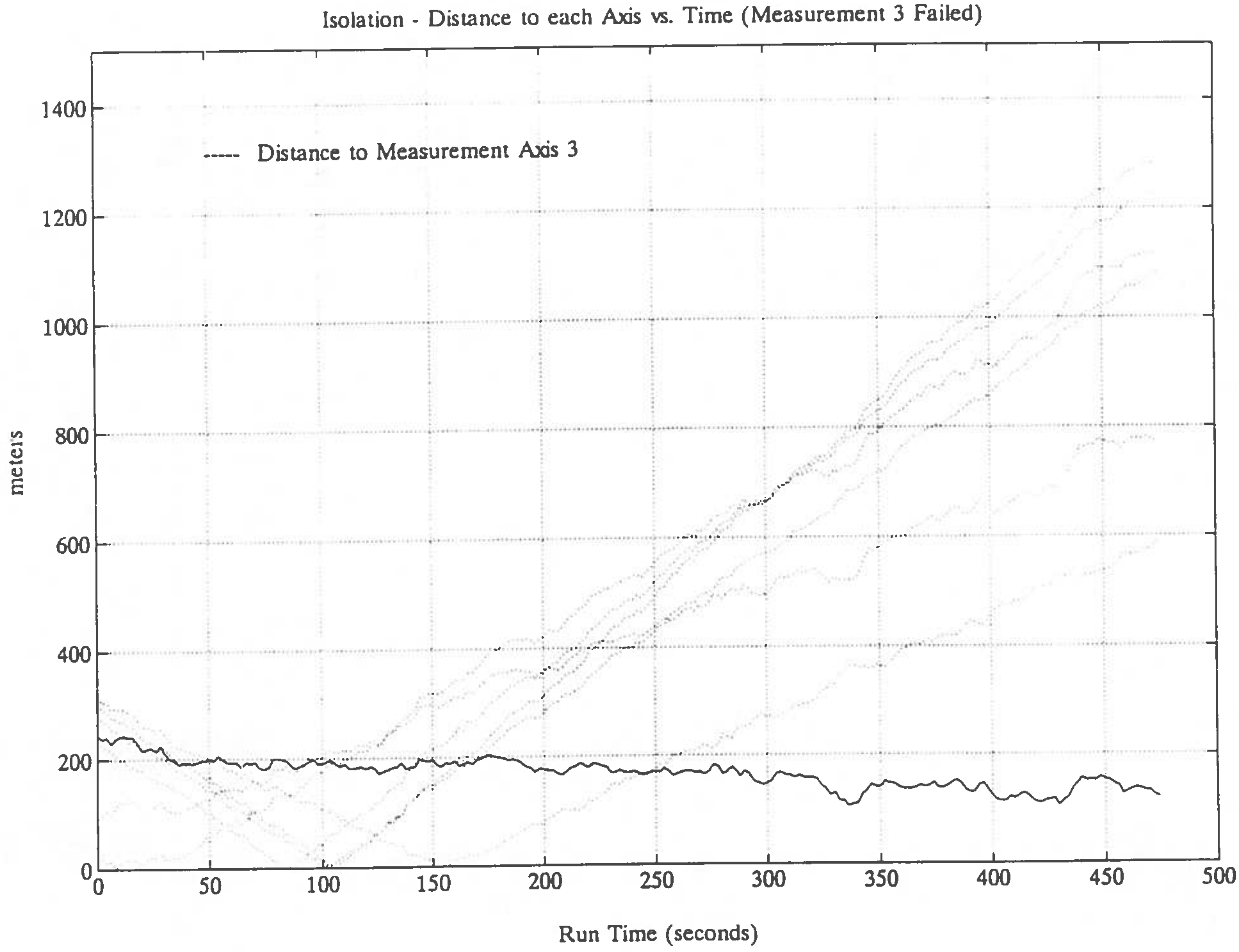


Figure 40 Isolation Performance (5 m/s Ramp Error in Measurement 3).

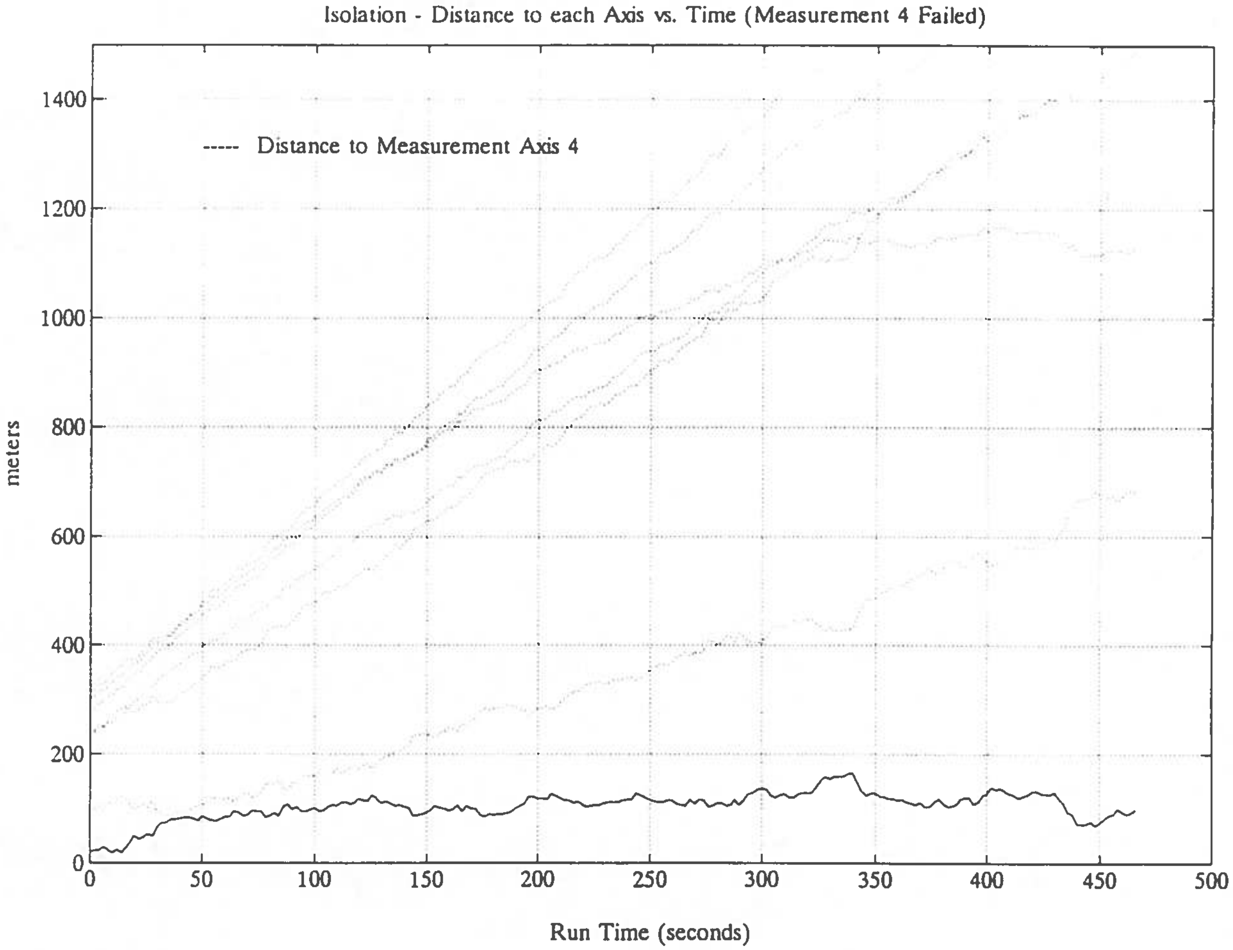


Figure 41 Isolation Performance (5 m/s Ramp Error in Measurement 4).

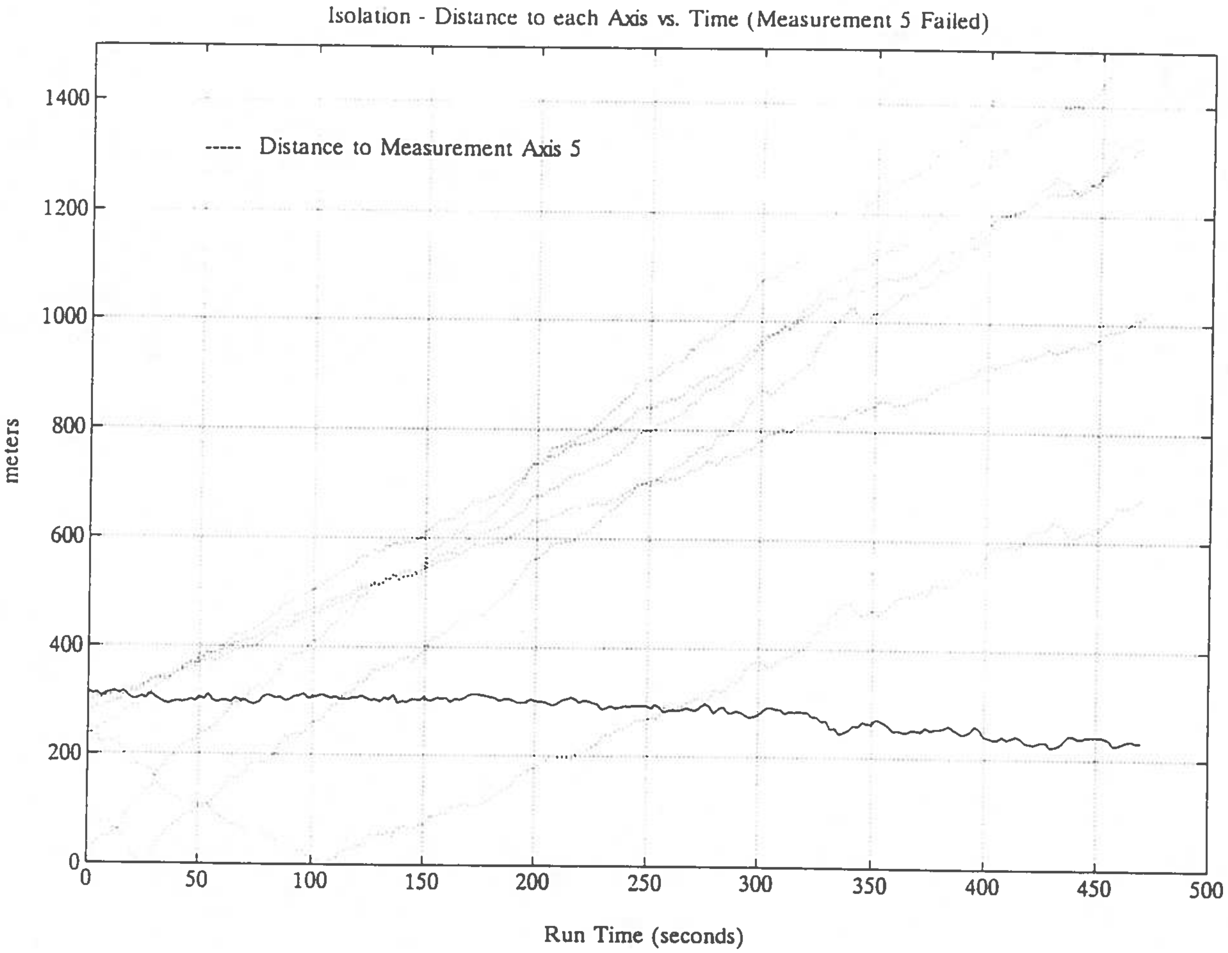


Figure 42 Isolation Performance (5 m/s Ramp Error in Measurement 5).

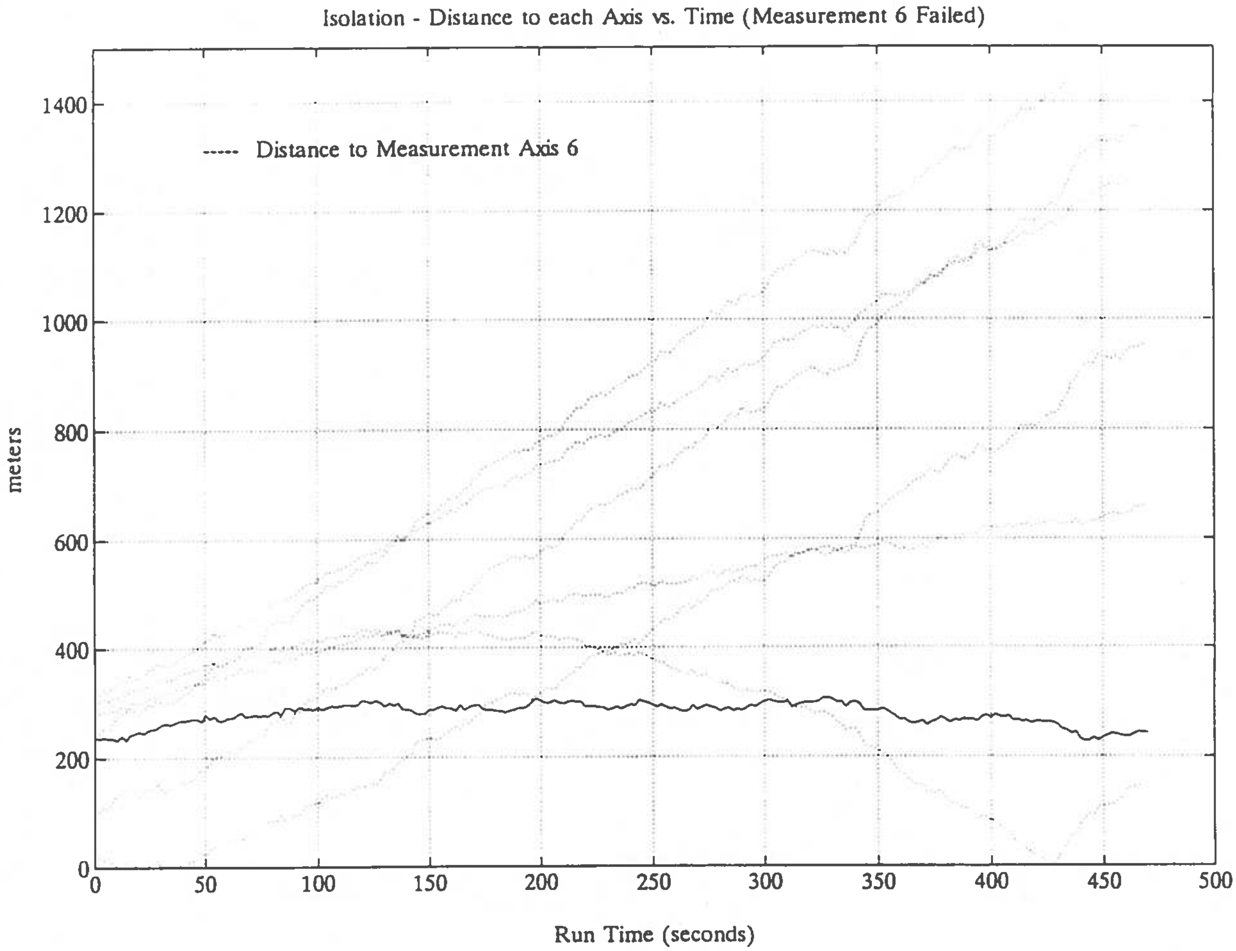


Figure 43 Isolation Performance (5 m/s Ramp Error in Measurement 6).

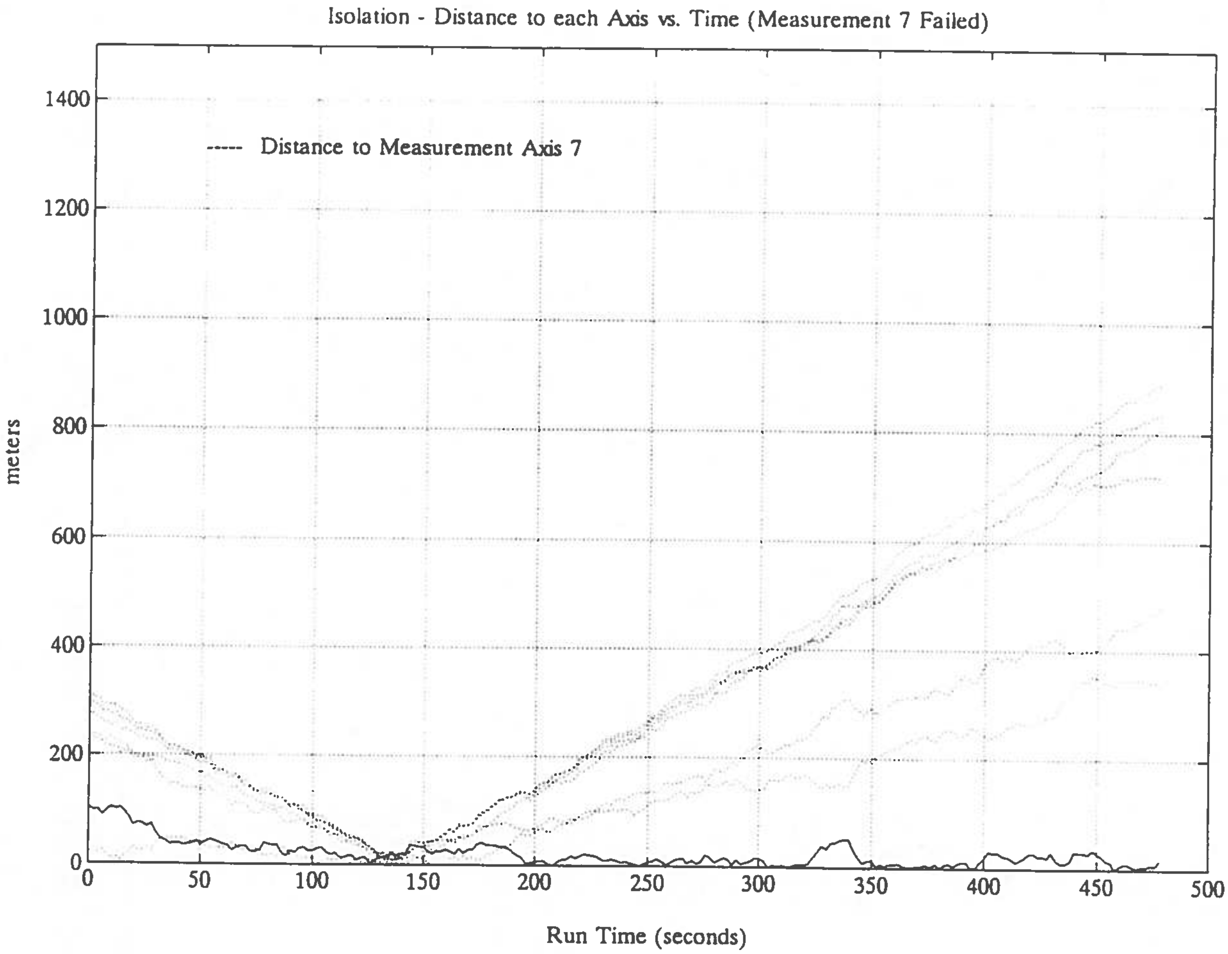


Figure 44 Isolation Performance (5 m/s Ramp Error in Measurement 7).

9. SUMMARY AND CONCLUSIONS

Fault Detection and Isolation (FDI) algorithms are required to achieve a sole-means navigation system using GPS and Loran-C. In order to provide a better understanding of the Fault Detection and Isolation algorithm, a simplified example is presented using three voltmeters to measure a single voltage. The two redundant measurements allow for the full characterization of the FDI algorithm.

Because navigation systems will depend on FDI for integrity assurance, the availability of a navigation system will depend on the availability of at least two redundant measurements. A Markov analysis, combined with a GPS coverage program, is presented to show that GPS by itself could meet supplemental availability requirements for fault detection, but not for isolation. An example of an integrated navigation system, GPS/Loran-C, is presented and actual flight data is used to demonstrate the FDI algorithm in the presence of signal failures. Computer simulations are performed to inject artificial failures into GPS and Loran-C signals, and the results of the FDI algorithm are displayed pictorially.

The following conclusions are based on the work presented in this report:

1. Sole means navigation will require a high availability of at least two redundant measurements for FDI. Therefore, GPS must be augmented by another system, such as Loran-C, before it has the potential to meet requirements for sole means navigation.

2. The performance of the fault detection algorithm is fully characterized. The probability of a false alarm should not be traded against the probability of a missed detection. Instead, both probabilities should be traded against the horizontal protection radius. This approach guarantees the integrity of the navigation solution at all time and space points. Furthermore, it facilitates changes in navigation accuracy requirements without the need to change the implementation of the algorithm.

3. The parity vector should be observed over a period of time in order to aid fault isolation. The "snapshot" batch estimator is still applicable, but rather than discarding the parity information for each measurement sample, the history of the parity vector should be stored and evaluated. The methods presented in this report constitute a novel approach to fault isolation which greatly enhances the effectiveness of FDI.

10. REFERENCES

[1] Van Graas, F., "Hybrid GPS/Loran-C: A Next-Generation of Sole Means Air Navigation," Ph. D. Dissertation, Ohio University, Department of Electrical and Computer Engineering, November 1988.

[2] Ibid.

[3] Radio Technical Commission for Aeronautics, Special Committee 159, "Terms of Reference," Third Revision, RTCA Paper No. 259-91/SC159-294, July 9, 1991.

[4] Daly, K. C., Gai, E. G., and Harrison, J. V., "Generalized Likelihood Test for FDI in Redundant Sensor Configurations," Journal of Guidance and Control, Vol. 2, Jan.-Feb. 1979.

[5] Gai, E. G., Adams, M. B., and Walker, B. K., "Determination of Failure Thresholds in Hybrid Navigation," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-12, Nov. 1976.

[6] Morrell, F. R., Bailey, M. L., and Motyka, P. R., "Flight Demonstration of Redundancy Management Algorithms for a Skewed Array of Inertial Sensors," Proceedings of the AIAA/AHS/ASEE Aircraft Design and Operations Meeting, AIAA Paper 88-4434, Atlanta, GA, September 7-9, 1988.

[7] Hughes, D., "ICAO Delegates Back FANS Concept, Set Stage for Global Satellite Systems," Aviation Week & Space Technology, Vol. 135, No. 15, October 14, 1991.

[8] Potter, J. E., and Suman, M. C., "Thresholdless Redundancy Management with Arrays of Skewed Instruments," Integrity in Electronic Flight Control Systems, AGARDOGRAPH-224, 1977.

[9] Golub, G. H., and Van Loan, C. F., *Matrix Computations*, Second Edition, The Johns Hopkins University Press, Baltimore, MD, 1990.

[10] Brenner, M., "Implementation of a RAIM Monitor in a GPS Receiver and an Integrated GPS/IRS," Proceedings of the Third International Satellite Division Meeting of the Institute of Navigation, September 1990.

[11] Op. Cit., Potter, J. E., et al.

[12] Radio Technical Commission for Aeronautics, Special Committee 159, "Minimum Operational Performance Standards for Airborne Supplemental Navigation Equipment using Global Positioning System (GPS)," Fourth Draft, RTCA Paper No. 207-90/SC159-242, May 16, 1991.

[13] Van Graas, F., "GPS/LORAN Interoperability," Phase II, Avionics Engineering Center, Ohio University, Report No. OU/AEC 17-89TM 00006/22, 1989.

[14] Ibid.

[15] Van Graas, F., "Hybrid GPS/LORAN-C: A Next-Generation of Sole Means Air Navigation," Ph. D. Dissertation, Ohio University, Department of Electrical and Computer Engineering, November, 1988.

[16] Knable, N., and Kalafus, R. M., "Clock Coasting and Altimeter Error for GPS," NAVIGATION: Journal of the Institute of Navigation, Vol. 30, No. 4, Winter 1984-85.

[17] Green, Colonel G. B., "The GPS 21 Primary Satellite Constellation," NAVIGATION: Journal of the Institute of Navigation, Vol. 36, No. 1, Spring 1989.

[18] Op. Cit., Van Graas, F. {1989}.

[19] Billinton, R., and Allan, R. N., *Reliability Evaluation of Engineering Systems: Concepts and Techniques*," Pitman Publishing, Inc., Great Britain, 1983.

[20] Op. Cit., Green.

[21] Federal Radionavigation Plan (FRP), 1988. DOD-4650.4 and DOT-TSC-RSPA-87-3, U.S. Department of Defense and U.S. Department of Transportation, Washington, D.C., 1986.

[22] Op. Cit., Van Graas, F. {1989}.

[23] Op. Cit., Federal Radionavigation Plan (FRP).

[24] Van Graas, F., and Kuhl, M. R., "Multisensor Signal Processing Techniques," Final Report, August 1989-October 1990, prepared for U.S. DOT, RSPA, VNTSC, Report No. DOT/FAA/SE-91/4, Washington, D.C., September 1991.

[25] Van Graas, F., "In Flight Demonstration of Hybrid GPS/LORAN RAIM," Proceedings of the National Technical Meeting of the ION, Phoenix, AZ, January 22-24, 1991.

[26] Ibid.

[27] Van Graas, F., and Farrell, J. L., "Receiver Autonomous Integrity Monitoring (RAIM): Techniques, Performance, and Potential," Proceedings of the 47th Annual Meeting of The Institute of Navigation, Williamsburg, VA, June 10-12, 1991.

[28] Ibid.

[29] Op. Cit., Radio Technical Commission for Aeronautics {May 16, 1991}.

[30] Op. Cit., Van Graas and Kuhl.

[31] Op. Cit., Brenner.

[32] Op. Cit., Golub et al.

[33] Op. Cit., Van Graas and Kuhl.