FAULT DETECTION ALGORITHM FOR
GLOBAL POSITIONING SYSTEM RECEIVERS

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Sang-Sung Choi
June, 1991
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES iv</td>
</tr>
<tr>
<td>LIST OF TABLES v</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS vi</td>
</tr>
<tr>
<td>1. INTRODUCTION 1</td>
</tr>
<tr>
<td>2. BACKGROUND 4</td>
</tr>
<tr>
<td>2.1 Integrity Monitoring 4</td>
</tr>
<tr>
<td>2.2 Receiver Autonomous Integrity Monitoring (RAIM) 7</td>
</tr>
<tr>
<td>2.2.1 Recursive Estimators 8</td>
</tr>
<tr>
<td>2.2.2 Batch Estimators 8</td>
</tr>
<tr>
<td>2.3 Fault Detection Techniques for the GPS RAIM 11</td>
</tr>
<tr>
<td>3. THE FAULT DETECTION ALGORITHM 14</td>
</tr>
<tr>
<td>3.1 GPS Measurement Equations 14</td>
</tr>
<tr>
<td>3.1.1 Overview of GPS 14</td>
</tr>
<tr>
<td>3.1.2 Relation between User Position and GPS Measurements 15</td>
</tr>
<tr>
<td>3.2 Fault Detection 19</td>
</tr>
<tr>
<td>3.2.1 Measurement Model and Parity Vector 20</td>
</tr>
<tr>
<td>3.2.2 Fault Detection Parameters 24</td>
</tr>
<tr>
<td>3.2.2.1 One Redundant Measurement 27</td>
</tr>
<tr>
<td>3.2.2.2 Two or More Redundant Measurements 32</td>
</tr>
<tr>
<td>3.3 Horizontal Radial Position Error 34</td>
</tr>
<tr>
<td>3.4 Availability of RAIM 41</td>
</tr>
</tbody>
</table>
**TABLE OF CONTENTS (Continued)**

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. COMPUTER SIMULATIONS</td>
<td>42</td>
</tr>
<tr>
<td>4.1 Simulation Description</td>
<td>42</td>
</tr>
<tr>
<td>4.2 GPS Constellation</td>
<td>46</td>
</tr>
<tr>
<td>5. COMPUTER SIMULATION RESULTS</td>
<td>49</td>
</tr>
<tr>
<td>5.1 Satellite Geometry</td>
<td>49</td>
</tr>
<tr>
<td>5.2 Alarm Rate as a Function of the Measurement Bias Error</td>
<td>51</td>
</tr>
<tr>
<td>5.3 Protected Horizontal Radial Position Error</td>
<td>53</td>
</tr>
<tr>
<td>5.4 Relation between the Actual Position Error and the Protected position error</td>
<td>56</td>
</tr>
<tr>
<td>5.5 Performance of the Fault Detection Algorithm</td>
<td>62</td>
</tr>
<tr>
<td>5.6 Availability of Navigation Information</td>
<td>63</td>
</tr>
<tr>
<td>6. CONCLUSIONS</td>
<td>68</td>
</tr>
<tr>
<td>7. REFERENCES</td>
<td>70</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>75</td>
</tr>
<tr>
<td>A. Data Generation Program Listing</td>
<td>75</td>
</tr>
<tr>
<td>B. Fault Detection Program Listing</td>
<td>83</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>95</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.1</td>
<td>Kalman Filter Recursive Loop</td>
</tr>
<tr>
<td>2.2</td>
<td>Example of Snapshot Method</td>
</tr>
<tr>
<td>3.1</td>
<td>GPS Ranging Geometry [F. van Graas, 1988]</td>
</tr>
<tr>
<td>3.2</td>
<td>Integrity Outcomes</td>
</tr>
<tr>
<td>3.3</td>
<td>Probability Density Function of the Detection Parameters with and without a Measurement Bias Error</td>
</tr>
<tr>
<td>3.4</td>
<td>Horizontal Radial Position Error</td>
</tr>
<tr>
<td>4.1</td>
<td>Block Diagram of the Computer Simulation</td>
</tr>
<tr>
<td>5.1</td>
<td>HDOP during 24 Hours</td>
</tr>
<tr>
<td>5.2</td>
<td>Alarm Rate</td>
</tr>
<tr>
<td>5.3</td>
<td>Protected Horizontal Radial Position Error (In the presence of measurement noise only)</td>
</tr>
<tr>
<td>5.4</td>
<td>Protected Horizontal Radial Position Error (In the presence of both measurement noise on all measurements and a measurement bias error on one measurement)</td>
</tr>
<tr>
<td>5.5</td>
<td>Actual Position Error vs Protected Position Error (Measurement bias error = 0 Meters)</td>
</tr>
<tr>
<td>5.6</td>
<td>Actual Position Error vs Protected Position Error (Measurement bias error = 300 Meters)</td>
</tr>
<tr>
<td>5.7</td>
<td>Actual Position Error vs Protected Position Error (Measurement bias error = 600 Meters)</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Actual Position Error vs Protected Position Error (Measurement bias error = 900 Meters)</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Actual Position Error vs Protected Position Error (Measurement bias error = 1200 Meters)</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Normal Operation</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>False Alarm</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>Availability of Position Information</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2.1 Tentative GPS Integrity Performance Requirements 5
Table 3.1 Detection Thresholds as a Function of the Number of Measurements(σ=32 meters, P_{fa}=10^{-6}) 35
Table 4.1 21 Primary Satellite Constellation Ephemerides 48
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>DME</td>
<td>Distance Measuring Equipment</td>
</tr>
<tr>
<td>DoT</td>
<td>Department of Transportation</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>ECEF</td>
<td>Earth-Centered-Earth-Fixed</td>
</tr>
<tr>
<td>ENU</td>
<td>East-North-Up</td>
</tr>
<tr>
<td>FAA</td>
<td>Federal Aviation Administration</td>
</tr>
<tr>
<td>FRP</td>
<td>Federal Radionavigation Plan</td>
</tr>
<tr>
<td>GIC</td>
<td>GPS Integrity Channel</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HDOP</td>
<td>Horizontal Dilution Of Precision</td>
</tr>
<tr>
<td>ION</td>
<td>Institute of Navigation</td>
</tr>
<tr>
<td>LORAN-C</td>
<td>Long Range Navigation-C</td>
</tr>
<tr>
<td>MOPS</td>
<td>Minimum Operational Performance Standards</td>
</tr>
<tr>
<td>MSD</td>
<td>Maximum Separation Distance</td>
</tr>
<tr>
<td>NAS</td>
<td>National Airspace System</td>
</tr>
<tr>
<td>NDB</td>
<td>Non-Directional Beacon</td>
</tr>
<tr>
<td>RAIM</td>
<td>Receiver Autonomous Integrity Monitoring</td>
</tr>
<tr>
<td>RTCA</td>
<td>Radio Technical Commission for Aeronautics</td>
</tr>
<tr>
<td>SC</td>
<td>Special Committee</td>
</tr>
<tr>
<td>TSO</td>
<td>Technical Standard Order</td>
</tr>
<tr>
<td>VOR</td>
<td>Very High Frequency Omnidirectional Range</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Navigation is the process of directing the movements of a craft from one point to another [M. Kayton and W. R. Fried, 1969]. Electronic navigation aids such as radio facilities, lights, and instruments are essential to make the measurements needed for safe navigation. These navigation aids are playing a significant and essential role in navigation, supplementing and enhancing the ability of the human navigator. The Very High Frequency Omnidirectional Range (VOR) system and Distance Measuring Equipment (DME), Omega, Non-Directional Beacons (NDB), and the Long Range Navigation system (LORAN-C) are examples of air radionavigation systems. These navigation systems provide sufficient accuracy, reliability, coverage, availability, and integrity to satisfy either sole means navigation system or supplemental navigation system requirements. In particular, the NAVSTAR Global Positioning System (GPS) is a new satellite-based radionavigation system being developed by the United States Department of Defense (DoD). When GPS becomes fully operational around 1993, the Federal Aviation Administration (FAA) plans to certify the GPS as a supplemental navigation system [Federal Radionavigation Plan, 1988].

In order to use the GPS for aeronautical navigation, first the FAA decides that GPS would augment the National Airspace
System (NAS), and then the FAA assigns a task to the Radio Technical Commission for Aeronautics (RTCA) to develop the Minimum Operational Performance Standards (MOPS). Next, the FAA prepares a legal document called a Technical Standard Order (TSO) which is based on the MOPS, and then manufacturers can use the TSO to design, implement and test GPS receivers. At the time of this writing, the RTCA is finalizing the MOPS excepting integrity. Integrity is defined as the ability of a system to detect malfunctions and to promptly warn the user that the system is not operating within its specified performance limits [R. Braff, et al, 1983]. Although the current GPS Control Segment can monitor satellite failures continuously, it does not respond fast enough to notify the users of all satellite malfunctions. Since external integrity monitoring systems are not expected to be operational in the near future, the GPS receiver currently can only obtain integrity using the GPS signals itself. This method of obtaining integrity is referred to as Receiver Autonomous Integrity Monitoring (RAIM) [R. M. Kalafus and G. Y. Chin, 1988].

The basic idea of the GPS RAIM is to use inconsistency in the measured data to derive a fault detection statistic. Several papers describing GPS RAIM techniques have been published, see for instance [R. G. Brown, 1988; R. M. Kalafus and G. Y. Chin, 1988; B. W. Parkinson and P. Axelrad, 1989; G.
Y. Chin and J. H. Kraemer, 1989]. The majority of these papers describe detection algorithms with a constant false alarm rate while the probability of a missed detection varies with the satellite geometry. Since these detection algorithms use a missed detection probability averaged over all time and space points, the actual missed detection probability could be much worse than the required missed detection probability. Although these detection algorithms meet the required detection probability on the average, they do not guarantee the detection probability in cases such as a poor geometry condition. Recently, the civil aviation community adopted the requirement for an unconditional mission integrity. Subsequently, the RTCA changed the detection probability requirement from an averaged number to a minimum number which must always be satisfied. This new requirement necessitates the development of a fault detection algorithm with a constant probability of a missed detection.

The purpose of this study is to address the design and evaluation of a new fault detection algorithm which guarantees a minimum detection probability at all time and space points. The concept and theoretical analysis of this fault detection algorithm are presented in Chapter 2 and Chapter 3, the effectiveness of this algorithm is evaluated using a computer simulation as described in Chapter 4, and the simulation results are presented in Chapter 5.
2. BACKGROUND

2.1 Integrity Monitoring

Integrity is the ability of a system to detect and indicate malfunctions to the user to ensure that the system is not used when not operating within its specified performance limits. The FAA requires a minimum integrity level to certify the GPS as a supplemental navigation system. Table 2.1 shows the tentative GPS integrity performance requirements for supplemental use of GPS as developed by RTCA Special Committee 159 [RTCA Special Committee 159, 1990].

If 200 independent measurement samples per hour are obtained from the GPS receiver, then the maximum allowable alarm rate of 0.0002/Hr is approximately equivalent to an alarm rate of $10^{-6}$ per sample. The detection probability is a conditional probability given by

$$P_{Detection} = P(Detection|Failure) = 0.999 \quad (2.1)$$

where a failure means that the horizontal radial position error exceeds a pre-determined alarm limit. The alarm limits for the different phases of flight are given in Table 2.1. The probability of a missed detection is given by

$$P_{Missed\ Detection} = 1 - P_{Detection} = 10^{-3} \quad (2.2)$$
### Table 2.1 Tentative GPS Integrity Performance Requirements

<table>
<thead>
<tr>
<th>Phase of Flight</th>
<th>Performance Item</th>
<th>Enroute</th>
<th>Terminal</th>
<th>Non-Precision Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alarm Limit</td>
<td>2.0 nmi</td>
<td>1.0 nmi</td>
<td>0.3 nmi</td>
</tr>
<tr>
<td></td>
<td>Maximum Allowable</td>
<td>0.0002/HR</td>
<td>0.0002/HR</td>
<td>0.0002/HR</td>
</tr>
<tr>
<td></td>
<td>Alarm Rate</td>
<td>30 seconds</td>
<td>10 seconds</td>
<td>10 seconds</td>
</tr>
<tr>
<td></td>
<td>Time to Alarm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum Detection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Probability</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
</tr>
</tbody>
</table>
The response time of the GPS Control Segment is not fast enough to provide timely warnings for satellite signal malfunctions to the users. The FAA requires a time to alarm for the GPS of 10 seconds for non-precision approaches and terminal navigation, and 30 seconds for domestic enroute navigation, but the GPS Control Segment can take up to one hour before the malfunction information reaches the users. This slow response time necessitates the development of independent civil integrity monitoring techniques.

Two approaches, internal integrity monitoring systems and external integrity monitoring systems, are being investigated as means of ensuring the GPS integrity. An external integrity monitoring system monitors the GPS signals in real-time using a ground-based network of monitor stations and transmits an appropriate message to all users through a broadcast satellite. It should be noted that if the communication linkages malfunction, a large number of users will not be able to use the integrity function. The GPS Integrity Channel (GIC) monitoring system as described by [R. M. Kalafus, 1989] is an example of an external integrity monitoring system.

An internal integrity monitoring system achieves the integrity using only the GPS signals without any help from an external source. Receiver Autonomous Integrity Monitoring (RAIM) is an example of an internal integrity monitoring
system. It is a software-based algorithm implemented in a receiver to detect signal malfunctions using an overdetermined set of measurements. RAIM is currently the most promising integrity monitoring system pending the implementation of an external integrity monitoring system.

In addition to the RAIM technique for the GPS, hybrid or integrated integrity systems are also being considered to achieve integrity. These integrity systems use measurement data from more than one navigation system. For instance, a hybrid GPS/LORAN-C Navigation system [F. van Graas, 1988] could use three GPS satellites and two LORAN-C transmitters to determine aircraft position. Four measurements would determine the navigation solution, and the fifth measurement would be used for the integrity. These hybrid radionavigation systems can offer tremendous benefits such as improved safety, cost reduction, and airspace capacity expansion.

2.2 Receiver Autonomous Integrity Monitoring (RAIM)

The concept of RAIM is identical to well-known fault detection techniques for redundant inertial navigation system and multisensor navigation systems [J. C. Wilcox, 1974; E. Gai, et al, 1976; K. C. Daly, et al, 1979]. RAIM techniques use measurement data inconsistencies to derive a fault detection parameter. The consistency of the measurement data
can be examined at a particular time, or a time history of the measurement data can be used. Therefore, fault detection algorithms can be divided into two categories: recursive estimators and batch estimators [F. van Graas, 1991].

2.2.1 Recursive Estimators

A recursive estimator uses a time history of measurements. The current state estimate is obtained by combining the newly received measurement data and the state estimate obtained from previous measurement information. The recursive loop of a widely used recursive estimator, the Kalman filter [R. G. Brown and P. Y. C. Hwang, 1986], is shown in Figure 2.1. The Kalman filter provides a natural checkpoint for detecting rapidly growing measurement errors. However, slowly growing measurement errors such as those which might result from a satellite clock degradation error may not be detected since there is no obvious outlier in the measurement residual sequence.

2.2.2 Batch Estimators

The batch estimator used for fault detection requires at least one redundant measurement but it does not use a time history of the measurement data. In contrast to the recursive estimator, the batch estimator only examines the measurement
Enter a priori state estimate and error covariance

Compute Filter Gain

Project Ahead State Estimate and Error Covariance

Update State Estimate with Measurement Data

Update State Estimate and Error Covariance

Figure 2.1 Kalman Filter Recursive Loop

5 MEASUREMENTS (SNAPSHOT)

(a) Five Good Signals

(b) Signal Five Failed

Figure 2.2 Example of a Snapshot Method

data at one particular time. This is also referred to as a snapshot method. Figure 2.2 shows an example of snapshot processing in the position domain [A. K. Brown and T. Smid, 1988]. The measurements could either be processed in the range domain or in the position domain. These two approaches were shown to be mathematically equivalent by [Y. C. Lee, 1986]. The center of the circles represent the unknown true position, and the radius of the circle is the desired protection limit. Figure 2.2 (a) shows the least squares solutions for all sub-sets of four-out-of-five measurements. If the geometry of the sub-sets is sufficient, all sub-solutions should be close to one another. Figure 2.2 (b) shows the solution assuming measurement five failed. The sub-solution without measurement five does not change, but all other solutions are scattered. The Maximum Separation Distance (MSD) between two sub-solutions is used as a detection parameter for measurement data inconsistency. The detection occurs if the MSD exceeds a certain value. The least squares batch estimator is able to detect slowly growing measurement errors which are not detected by the recursive estimator.

2.3 Fault Detection Techniques for the GPS RAIM

A GPS receiver should use both a recursive and a batch fault detection algorithm. The recursive algorithm should be
used to detect and isolate rapidly growing measurement errors. The focus of this thesis is on the batch fault detection algorithm to detect slowly growing measurement errors. At this time, the batch fault detection method are not well understood and require further analysis.

Several fault detection techniques for the GPS RAIM which employ the snapshot method have been introduced over the past few years. The Range Comparison Method introduced by [Y. C. Lee, 1986] calculates the range to a redundant fifth satellite based on a four-satellite position solution. The difference between the calculated and measured range to the fifth satellite is used as the basis to detect an abnormal state. The Maximum Separation Method [R. G. Brown, 1988; R. M. Kalafus and G. Y. Chin, 1988] uses the maximum separation distance between two sub-solutions as the detection parameter. The Pseudorange Residual Method [B. W. Parkinson and P. Axelrad, 1989] uses a weighted root-sum-squares of the pseudorange residuals for each satellite as the detection parameter. All RAIM algorithms described above perform fault detection with a constant probability of a false alarm while the probability of a missed detection varies with the satellite geometry. Another fault detection algorithm based on the parity space concept was introduced by Brenner, who used orthogonal transformations to optimize the visibility of the error in each satellite. This algorithm achieves fault
detection with a constant probability of a missed detection and a constant probability of a false alarm [M. Brenner, 1990].

The fault detection algorithm introduced by Brenner is the only algorithm to date which directly relates to the integrity requirement of satisfying a minimum detection probability at each time and space points. This algorithm will be used as the basis for the fault detection algorithm presented in the next Chapter.
3. THE FAULT DETECTION ALGORITHM

The fault detection algorithm introduced in this Chapter is based on the parity space concept and uses orthogonal transformations to derive a parity vector. First, the measurement equations for the GPS are derived. Next, several parameters specifying the fault detection algorithm are introduced. This is followed by a detailed description of the fault detection algorithm and its performance in terms of the horizontal protection radius and algorithm availability.

3.1 GPS Measurement Equations

3.1.1 Overview of GPS

When the GPS becomes fully operational around 1993, it will provide position and timing information through a 24-satellite space segment under control of a ground segment. The ground segment consists of one master control station and several monitor/uplink stations. The monitor stations track the GPS satellites and relay the tracking information to the Master Control Station which determines the satellites' orbital and clock parameters. These parameters are transmitted to the satellites by the uplink stations for retransmission to the users. Each satellite transmits a composite spread spectrum signal at center frequencies of 1.2276 GHz (L₁) and 1.57542
GHz \( (L_2) \), which are modulated by either or both a 10.23 MHz precision (P-code) signal and/or a 1.023 MHz coarse/acquisition (C/A-code) signal [A. J. van Dierendonck, et al, 1978]. The receiver correlates the incoming code with the same code generated by the receiver. A correlation peak appears when both signals are synchronized. The GPS receiver measures the difference between the arrival time and the corresponding transmission time of the GPS signal. The time of transmission is encoded in the GPS signal and is obtained at the time of measurement.

### 3.1.2 Relation between User Position and GPS Measurements

The GPS ranging geometry is shown in Figure 3.1. The pseudorange measurement is the difference between the measured signal arrival time and the corresponding known signal transmission time, corrected for known and estimated error sources [F. van Graas, 1988]:

\[
P_i(t) = |S_i[t - \beta_i(t)] - U(t)| + C[T_{GPS}(t) - T_{s_i}(t) + d_{GPS}(t, r)]
\]

where,

- \( P_i \) : pseudorange measurement for satellite \( i \).
- \( S_i \) : position vector for satellite \( i \).
- \( \beta_i \) : line-of-sight travel time for signals from satellite \( i \).
- \( U \) : user position vector.
$S_i$ : Position Vector for Satellite $i$

$e_i$ : Line-of-Sight Vector for Satellite $i$

$U$ : User Position Vector

Figure 3.1  GPS Ranging Geometry [F. van Graas, 1988]
C : GPS speed of light (299792458 m/s).

\( T_{GPS} \) : user clock offset from GPS system time.

\( T_{s_i} \) : clock offset for satellite \( i \) from GPS time.

\( T_{GPS_i} \) : delay for measurement \( i \) caused by GPS error sources.

The satellite positions and clock offsets from GPS time are calculated from the navigation data transmitted by the satellites. The measurement delays mainly consist of tropospheric and ionospheric propagation delays. Typically, the tropospheric propagation delay varies slowly resulting in ranging errors less than 5 meters [R. Bowen, et al, 1985], but the ionospheric propagation delay can be as large as 20-30 meters during the day to as small as 1 meter at night for the single frequency user [R. N. Turner, et al, 1986].

The pseudorange measurement equation (3.1) can be rewritten in terms of the three dimensional position and receiver clock offset as follows:

\[
P'_i = \sqrt{(X_i - U_x)^2 + (Y_i - U_y)^2 + (Z_i - U_z)^2} + BC \tag{3.2}
\]

where, \( P'_i \) : pseudorange measurement for satellite \( i \) corrected for known and estimated error sources (m).

\( X_i, Y_i, Z_i \) : coordinates of satellite \( i \) (m).

\( U_x, U_y, U_z \) : coordinates of the user (m).

B : receiver clock offset with respect to GPS time (s).

C : speed of light (m/s).
Since this equation is non-linear, the three-dimensional user coordinates and clock offset are solved using a variation of Newton's method for non-linear systems [R. L. Burden and J. D. Faires, 1981]. This method yields quadratic convergence if the estimate is close to the solution.

Newton's method proceeds as follows: Define an user state vector $\mathbf{x}$ and a measurement vector $\mathbf{z}$. The user state vector $\mathbf{x}$ contains the position coordinates and clock offset, and the measurement vector $\mathbf{z}$ contains the corrected pseudorange measurements.

$$\mathbf{x} = \begin{pmatrix} U_x \\ U_y \\ U_z \\ B \end{pmatrix} \quad (3.3)$$

$$\mathbf{z} = \begin{pmatrix} P_1' \\ P_2' \\ \vdots \\ P_n' \end{pmatrix} \quad (3.4)$$

The linearized relation between the changes in the pseudorange measurement vector $\mathbf{z}$ and the corresponding change in the user state vector $\mathbf{x}$ is obtained using the Jacobian matrix $H$ as follows:

$$\delta \mathbf{z} = H \delta \mathbf{x} \quad (3.5)$$
where each row of $H$ is

$$
\begin{bmatrix}
\frac{\partial P'_i}{\partial U_x} & \frac{\partial P'_i}{\partial U_y} & \frac{\partial P'_i}{\partial U_z} & \frac{\partial P'_i}{\partial B}
\end{bmatrix}
$$

and the row elements of $H$ are given by

$$
\begin{align*}
\frac{\partial P'_i}{\partial U_x} &= \frac{U_x - X_i}{P'_i - B}, & \frac{\partial P'_i}{\partial U_y} &= \frac{U_y - Y_i}{P'_i - B} \\
\frac{\partial P'_i}{\partial U_z} &= \frac{U_z - Z_i}{P'_i - B}, & \frac{\partial P'_i}{\partial B} &= C
\end{align*}
$$

The least squares estimate can be used for obtaining $\delta x$ from equation (3.5):

$$
\delta x = (H^T H)^{-1} H^T \delta z
$$

### 3.2 Fault Detection

The purpose of fault detection is to raise an alarm when the horizontal radial error exceeds a pre-determined value called the protection radius. The detection statistic is derived from the redundant measurement data using orthogonal transformation as described in the first part of this Section.

Next, the probability of a false alarm and the probability of a missed detection are related to the detection threshold, and the minimum detectable measurement bias error is determined.
3.2.1 Measurement Model and Parity Vector

If the user location and clock offset are known with sufficient accuracy to linearize the pseudorange measurement equation, then the GPS update equation is given by

\[ \delta z = H \delta x + n + b \]

(3.8)

where, \( \delta z \): the change in the n-by-1 measurement vector.
\( \delta x \): the change in the 4-by-1 user state vector.
\( H \): a n-by-4 data matrix.
\( n \): a n-by-1 vector containing measurement noise.
\( b \): a n-by-1 vector containing measurement bias errors.

The elements of the noise vector are assumed to be normally distributed and mutually uncorrelated with zero-mean values and standard deviations \( \sigma = 32 \) meters.

When \( \delta z \) contains 4 measurement residuals (n=4), \( \delta x \) can be obtained by inverting the 4-by-4 measurement matrix \( H \). If the \( \delta z \) contains more than four measurement residuals (n>4), the system of equation is overdetermined. In this case, the measurement equation (3.8) contains redundant information, which can be extracted out by performing an orthogonal transformation on the update equation (3.8). The idea behind the orthogonal transformation is to extract the least squares measurement residual vector while maintaining the Gaussian statistics of the measurement noise. By definition, the
measurement residual vector is orthogonal to the estimation space spanned by the columns of \( H \). Therefore, the measurement residual vector contains all observable information about the inconsistencies of the redundant measurement data. The space in which the measurement residual vector lies is also referred to as the parity space.

The data matrix \( H \) can be decomposed into the product of a real orthogonal matrix \( Q \) and a upper triangular matrix \( R \) using the Q-R factorization [G. H. Golub and C. F. van Loan, 1989].

\[
H = QR \tag{3.9}
\]

The orthogonal matrix \( Q \) is characterized by the property

\[
QQ^T = Q^TQ = I \tag{3.10}
\]

where \( I \) is a \( n \)-by-\( n \) identity matrix.

Substituting equation (3.9) into the equation (3.5),

\[
\delta z = QR \delta x \tag{3.11}
\]

Pre-multiplying both sides of the equation (3.11) by \( Q^T \) yields

\[
Q^T \delta z = (Q^TQ) R \delta x = R \delta x \tag{3.12}
\]

Since the rank of \( R \) is equal to the rank of \( H \), the matrix \( R \) consists of a 4-by-4 upper triangular matrix \( R_{4\times4} \) in rows 1 through 4 and zeros only in rows 5 through \( n \).
\[ R \delta x = \begin{pmatrix} R_{utt} \\ 0 \end{pmatrix} \delta x = Q^T \delta z = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \delta z \quad (3.13) \]

Therefore, equation (3.13) can be divided into two parts:

\[ R_{utt} \delta x = Q_1 \delta z; \quad (3.14) \]

and

\[ \delta x = Q_2 \delta z = 0, \quad (3.15) \]

where \( R_{utt} \) consists of the first four rows of matrix \( R \),

\( Q_1 \) consists of the first four rows of matrix \( Q^T \), and

\( Q_2 \) consists of the last \( n-4 \) rows of matrix \( Q^T \).

Equation (3.14) shows that the position update can be obtained as follows:

\[ \delta x = (R_{utt})^{-1} \cdot Q_1 \cdot \delta z \quad (3.16) \]

This expression is identical to the least squares estimate given by equation (3.7). This can be seen by substituting \( H=Q^TR \).

Equation (3.15) shows that the matrix \( Q_2 \) will map the measurement residual vector to zero. Therefore, the columns of \( Q_2 \) and \( \delta z \) are orthogonal and the columns of \( Q_2 \) span the left null-space of \( H \). The left null-space is also referred to as the parity space [M. A. Sturza, 1988].
If the measurements are corrupted by errors, the measurement equation is changed to equation (3.8). The corresponding least squares estimate is given by

\[ \delta x = H^*(\delta z - n - b) \]  

(3.17)

where, \( H^* = (H^TH)^{-1} H \)

Normally, the measurement noise vector \( n \) and the measurement bias vector \( b \) are unknown, but their components in parity space can be found from the equation (3.15).

\[ p = -Q_2 n + Q_2 b \]  

(3.18)

The parity vector \( p \) is used for the fault detection, as described in Section 3.2.2.1 and 3.2.2.2. The expected value and covariance matrix of the parity vector \( p \) are:

\[ E[p] = -Q_2 b \]  

(3.19)

\[ COV[p] = E[p p^T] = Q_2 COV[n] \] \[ Q_2 = \sigma^2 I \]  

(3.20)

where the element of the measurement noise vector \( n \) are uncorrelated and normally distributed with equal variances. An important observation is that the elements of the parity vector are also normally distributed.
3.2.2 Fault Detection Parameters

The purpose of fault detection for GPS is to raise an alarm when the horizontal radial position error exceeds an alarm threshold. It should be noted that the fault detection is performed in parity space rather than in the position space. Because of this, four different outcomes can be generated by the fault detection algorithm as shown in Figure 3.2. The fault detection parameter, derived from the redundant measurement data, is compared with a pre-calculated detection threshold $T_p$. If the norm of the detection parameter is greater than the detection threshold, the alarm is raised. Otherwise, no alarm is raised. If the alarm is raised, then this could either be a true alarm or a false alarm, depending on whether the horizontal radial position error exceeded the threshold or remained below the threshold. Similarly, if no alarm is raised, the algorithm could either be operating normally or the algorithm experienced a missed detection. The four possible integrity outcomes are summarized below.

(1) Normal Operation (NO) :
When no alarm exists and the actual radial position error does not exceed the horizontal radial alarm threshold.

(2) Missed Detection (MD) :
When no alarm exists and the actual horizontal radial position
Figure 3.2 Integrity Outcomes
error exceeds the horizontal radial alarm threshold.

(3) False Alarm (FA) :
When an alarm exists and the actual horizontal radial position error does not exceed the horizontal radial alarm threshold.

(4) True Alarm (TA) :
When an alarm exists and the actual horizontal radial position error exceeds the horizontal radial alarm threshold.

Based on the possible outcomes described above, the following parameters are used in the fault detection algorithm:
- Probability of a false alarm : $P_{FA}$
- Probability of a missed detection : $P_{MD}$
- Horizontal radial alarm threshold : $R_T$
- Detection threshold : $T_D$
- Minimum detectable bias error required to satisfy the $P_{MD} : b_i$
- Probability of exceeding the horizontal radial alarm threshold : $P_T$
- Standard deviation of the measurement noise : $\sigma$

It should be noted that the detection algorithm operates in parity space even though $P_{MD}$ and $P_{FA}$ are defined in the horizontal position space. Fortunately, in the case of GPS,
$P_{MD}$ and $P_{FA}$ are approximately the same in parity space and position space. For instance, a false alarm in parity space could either be a true or a false alarm in position space. The probability of a true alarm in position space is $10^{-9}$, while the probability of a false alarm is $10^{-6}$. Therefore, a false alarm in parity space is most likely also a false alarm in position space.

**3.2.2.1 One Redundant Measurement**

If one redundant measurement is available ($n=5$), the parity vector $p$ is one dimensional and $Q_2$ is reduced to a row vector. In the absence of a measurement bias error, the parity scalar $p$ is only a function of the measurement noise, which has a zero mean, normal probability density function; specifically:

$$f_p(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{x}{\sigma}\right)^2$$  \hspace{1cm} (3.21)

A false alarm in parity space occurs when the norm of the detection parameter $p$ exceeds a given detection threshold $T_D$, see also Figure 3.3 (a). The probability of a false alarm is given by

$$P_{FA} = P(|p| > T_D) = \frac{2}{\sigma \sqrt{2\pi}} \int_{T_D}^{\infty} \exp\left(-\frac{x}{\sigma}\right)^2 dx$$  \hspace{1cm} (3.22)
Figure 3.3  Probability Density Function of the Detection Parameter with and without a Measurement Bias Error
Equation (3.22) can be rewritten using the complementary error function as follows:

\[ P_{FA} = \text{erfc} \left( \frac{T_D}{\sigma \sqrt{2}} \right) \]  
(3.23)

In the presence of a bias error in measurement \( i \), the parity scalar \( p \) has a normal probability density function with a mean value equal to the absolute value of the parity scalar. The expected value of this normal distribution is:

\[ m_i = |q \cdot b| \]  
(3.24)

where the vector \( q \) is the transpose of the first row of \( Q_2 \).

Therefore, the probability density function of the scalar \( p \) is given by:

\[ f_p(x) = \frac{1}{\sigma \sqrt{2 \pi}} \exp\left(-\frac{x-m_i}{\sigma \sqrt{2}}\right)^2 \]  
(3.25)

A missed detection in parity space occurs when the norm of the parity scalar does not exceed the detection threshold \( T_D \), given that a measurement bias error exists, see also Figure 3.3 (b). The probability of a missed detection is given by:

\[ P_{MD} = P(\mid p \mid \leq T_D) = \frac{1}{\sigma \sqrt{2 \pi}} \int_{-T_B}^{T_B} \exp\left(-\frac{x-m_i}{\sigma \sqrt{2}}\right)^2 dx \]  
(3.26)
Since the contribution of the integral between $-\infty$ to $-T_D$ can be neglected, the probability of a missed detection can be approximated by:

\[
P_{MD} = \frac{1}{2} \text{erfc}\left(\frac{m_i - T_D}{\sigma \sqrt{2}}\right)
\] (3.27)

Equations (3.23) and (3.27) provide the performance of the fault detection algorithm in terms of probability of a false alarm and probability of a missed detection. The detection threshold $T_D$ and the measurement noise standard deviation $\sigma$ are needed for both probabilities, but the expected value $m_i$, which is the absolute value of the parity scalar, is only needed for the probability of a missed detection.

If the required probability of a false alarm and the measurement noise standard deviation are given, the detection threshold can be calculated using equation (3.23) and the inverse complementary error function.

\[
T_D = \sigma \sqrt{2} \text{ inverfc} (P_{FA})
\] (3.28)

where inverfc is $(\text{erfc})^{-1}$

Since this detection threshold must be set high enough to satisfy the required probability of a false alarm, the detection algorithm is not able to detect a measurement bias error smaller than the level of the measurement noise.
Therefore, a minimum measurement bias error is required to satisfy the probability of a missed detection.

Given the probability of a missed detection, the measurement noise standard deviation and the detection threshold, the minimum required expected value $m_{\text{Min}}$ is calculated from equation (3.27).

$$m_{\text{Min}} = T_D + \sigma \sqrt{2} \ \text{inverfc}(2P_{MD}) \quad (3.29)$$

Therefore, the probability of a missed detection is only satisfied if the measurement bias error gives rise to a bias in the absolute value of the parity scalar greater than or equal to $m_{\text{Min}}$. The minimum bias error $b_i$ for each measurement can be calculated using equation (3.24).

$$b_i = \frac{m_{\text{Min}}}{|\mathbf{q}|} \quad (3.30)$$

where $\mathbf{q}$ is known from the measurement geometry.

Consequently, if the probability of a false alarm, the probability of a missed detection, and the measurement noise standard deviation are given, the minimum detectable measurement bias error becomes a function of the measurement geometry.
3.2.2.2 Two or More Redundant Measurements

If more than one redundant measurement is available \((n>5)\), the parity vector \(p\) is \(m\)-dimensional where \(m\) is the number of redundant measurements. The elements of the parity vector \(p\) have normal probability density functions with the same Gaussian statistics as the case of one redundant measurement. Each element of the parity vector \(p\) should be examined separately to maintain the Gaussian statistics.

The calculation of the minimum detectable bias error is more complicated since the matrix \(Q_2\) contains more than one row vector. The idea is to perform orthogonal transformations on \(Q_2\) to maximize the visibility of the measurement biases in parity space [M. Brenner, 1990].

A bias error in measurement \(i\) would have components along all axes in the parity space given by column \(i\) of \(Q_2\). The parity space can be rotated such that the bias error lies along one particular axis only of the rotated parity space. In this case, the \(i^{th}\) column of the rotated matrix \(Q_2\) would contain zeros except for one element. This element would be used to calculate the minimum required measurement bias error \(b_i\). The process of rotation of the matrix \(Q_2\) is performed for all measurements. Each time, only one row of the rotated matrix is used for fault detection. The Q-R factorization is
used for the above rotation process, and the first row of the upper-diagonal R matrix is used for fault detection.

The process of rotating the parity space proceeds as follows. The matrix $Q_2$ is decomposed into the product of a real orthogonal matrix $Q'$ and a upper triangular matrix $R'$ using the Q-R factorization.

$$Q_2 = Q' R'$$ (3.31)

The rotated parity vector is given by:

$$p' = Q'^T p = -R'(n + b)$$ (3.32)

For the first measurement, only the first row of $R'$ is used for fault detection. In essence, the first row of $R'$ replaces the vector $q$. Column switching is used to simplify the calculations for the $i^{th}$ measurement, $i \neq 0$. The first column of $Q_2$ is switched with the $i^{th}$ column of $Q_2$ before applying the Q-R factorization. After the Q-R factorization, the first column of $R'$ is switched with the $i^{th}$ column of $R'$. Note that $i$ is the measurement number. Each time, the first row of $R'$ is used for fault detection. Therefore, the minimum measurement bias error required to satisfy the probability of a missed detection is calculated from the following equation:
where \( R'(1,i) \) is the \( i \)th element of the first row of \( R' \) matrix.

The above procedure results in \( n \) parity scalars and \( n \) fault detection tests. This increases the probability of a false alarm by approximately a factor of \( n \). To compensate for the higher false alarm probability, the detection threshold is changed to:

\[
T_D = \sigma \sqrt{2} \ \text{inverfc} \left( \frac{P_{FA}}{n} \right)
\]  

(3.34)

where \( n \) is the number of measurements \((n>5)\).

Detection thresholds for different numbers of measurements are shown in Table 3.1 based on a false alarm probability of \( P_{FA}=10^{-6} \) (per sample) and a measurement noise standard deviation of \( \sigma=32 \) (meters).

3.3 Horizontal Radial Position Error

The probability of a false alarm, the probability of a missed detection, the measurement noise standard deviation, and the minimum required measurement bias error are related in the previous Section. Based on these parameters, the horizontal radial position error is derived in this Section.
Table 3.1 Detection Thresholds as a Function of the Number of Measurements ($\sigma=32$ Meters, $P_{fa}=10^{-6}$)

<table>
<thead>
<tr>
<th>n</th>
<th>$T_n(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>156.5 Meters</td>
</tr>
<tr>
<td>6</td>
<td>167.3 Meters</td>
</tr>
<tr>
<td>7</td>
<td>168.2 Meters</td>
</tr>
<tr>
<td>8</td>
<td>168.9 Meters</td>
</tr>
<tr>
<td>9</td>
<td>169.6 Meters</td>
</tr>
<tr>
<td>10</td>
<td>170.2 Meters</td>
</tr>
<tr>
<td>11</td>
<td>170.7 Meters</td>
</tr>
<tr>
<td>12</td>
<td>171.2 Meters</td>
</tr>
</tbody>
</table>
If the measurements are corrupted by both measurement noise and measurement biases, then the user state error vector is obtained from equation (3.14).

\[ \Delta x = -(R_{utt})^{-1} Q_1 (\mathbf{a} + \mathbf{b}) \quad (3.35) \]

The expected value and the error covariance matrix of the user state vector are given by

\[ E[\Delta x] = -(R_{utt})^{-1} Q_1 \mathbf{b} \quad (3.36) \]

\[ \text{COV}[\Delta x] = \sigma^2 (R_{utt}^{-1} R_{utt})^{-1} \quad (3.37) \]

Assume that the matrices \( R_{utt} \) and \( Q_1 \) are expressed in a locally-level reference frame, then the horizontal components of the expected value and the variance of the user state error vector are given by [F. Van Graas, 1991]:

\[ E[\Delta x_H] = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \quad (3.38) \]

\[ \text{VAR}[\Delta x_H] = \begin{pmatrix} \sigma_x^2 \cdot \text{XDOP}^2 \\ \sigma_y^2 \cdot \text{YDOP}^2 \end{pmatrix} = \begin{pmatrix} \sigma_x^2 \\ \sigma_y^2 \end{pmatrix} \quad (3.39) \]

where \( \bar{x} \) and \( \bar{y} \) are the first two components of \( E[\Delta x] \); \( \text{XDOP}^2 \) and \( \text{YDOP}^2 \) are the first and second diagonal elements of \( (R_{utt}^T R_{utt})^{-1} \), respectively.
**Horizontal Position Bias Error**

Each satellite has a minimum bias error necessary for detection with the required probabilities of false alarm and missed detection, see equation (3.33). Each of these measurement bias errors can be converted into a horizontal position error using equation (3.36). The detection algorithm must protect against the worst case position error. The worst case position error is obtained by selecting the largest norm of the horizontal radial position error.

\[
R_{bias} = \max(\sqrt{x_i^2 + y_i^2})
\]  

(3.40)

Therefore, the measurement bias error can be detected with the required probabilities of false alarm and missed detection only if the measurement bias error causes a horizontal radial position bias error greater or equal than the bias protection radius \( R_{bias} \).

**Horizontal Position Noise Error**

The components \( x \) and \( y \) of the horizontal position error have a bivariate normal probability density function [D. J. Torrieri, 1984; G. Y. Chin, 1987].

Assume that \( x \) and \( y \) are independent, then the objective is to find a circle with radius \( R_{noise} \) which protects the estimated
horizontal position with a probability of one minus \( P_t \), where \( P_t \) is the probability of exceeding \( R_{\text{noise}} \). An upper bound for the noise protection radius is determined as follows:

\[
R_{\text{noise}} = \sigma \sqrt{2 \cdot \text{inverfc}(P_t)} \cdot \text{HDOP}
\]

(3.41)

where \( \sigma \) is measurement noise standard deviation; and \( \text{HDOP}^2 = \text{XDOP}^2 + \text{YDOP}^2 \).

**Total Horizontal Position Error**

The total horizontal position error is the sum of the bias error vector and the noise error vector with lengths \( R_{\text{bias}} \) and \( R_{\text{noise}} \), respectively. It would be conservative to assign the protection radius the sum of the values \( R_{\text{bias}} \) and \( R_{\text{noise}} \) since the direction of the noise vector will be different from sample to sample. Ideally, the total horizontal position error should be calculated from a non-zero mean bivariate Gaussian distribution for each moment of time, but its numerical integration calculation is moderately involved. To avoid this calculation, three different cases are examined. All cases assume a detection probability of 0.999.

(1) Poor Geometry Scenario

In Figure 3.4 (a), the distribution of the noise error is almost one-dimensional. The total horizontal radial position
Figure 3.4  Horizontal Radial Position Error

(a) Poor Geometry Scenario

(b) Good Geometry Scenario
error is then calculated as the sum of the bias radius and the noise radius for a detection probability of 0.999.

\[ R_{\text{total}} = R_{\text{bias}} + 3.29 \sigma_{\text{HDOP}} \]  

(3.42)

(2) Good Geometry Scenario

In Figure 3.3 (b), the total error has a Rayleigh-Rice distribution [C. W. Helstrom, 1984].

\[ f_z(z) = \frac{z}{\sigma^2} \exp\left(-\frac{z^2 + m^2}{2\sigma^2}\right) I_0\left(\frac{mz}{\sigma^2}\right) \]  

(3.43)

where the modified Bessel function \( I_0 \) is given by

\[ I_0(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{x\cos\theta} d\theta \]  

(3.44)

The calculation of the modified Bessel function and integration of equation (3.43) was performed using the program MATLAB [PC-MATLAB, 1989]. The total horizontal radial position error is given by

\[ R_{\text{total}} = R_{\text{bias}} + 3.17 \sigma_{\text{HDOP}} \]  

(3.45)
From the results of the above two cases, it is found that the protection radius is not significantly different for each case. Therefore, the conservative protection radius is used:

\[ R_p = R_{bias} + R_{noise} \]  

(3.46)

3.4 Availability of RAIM

The protected radial position error is given by equation (3.47). Since the protected radial position error depends on the measurement geometry, the detection algorithm can not guarantee the performance of the RAIM algorithm if the horizontal radial alarm threshold is set smaller than the protected radial position error. Therefore, the RAIM algorithm is only available if the protected radial position error as calculated by the detection algorithm is smaller than the required radial position error.
4. COMPUTER SIMULATIONS

To analyze the performance of the fault detection algorithm, a computer simulation program was developed in the PC-MATLAB environment. The simulation program will be used to address the effects of measurement errors, and to predict the availability of the fault detection algorithm. First, the simulation description is presented, and then the GPS constellation which was used for this simulation is described.

4.1 Simulation Description

Figure 4.1 shows the block diagram of the computer simulation for the fault detection algorithm. The simulation consists of two independent programs; the data generation program and the fault detection program.

Data Generation Program

This program consists of one main program and five sub-programs, which generate the pseudorange measurement data. The program listing for the data generation is provided in Appendix A. Since the satellite geometry for a fixed location repeats approximately every 24 hours, the data is generated for a period of 24 hours in 600 second time increments. A fixed location is selected at 39 degrees north of latitude and
Figure 4.1 Block Diagram of the Simulation

CALCULATE $M_{\text{MIN}}$

PMD

NOISE MEASUREMENT BIASES

$I_D$

CALCULATE $T_D$

RT

FAULT DETECTION

ALARM

USER LOCATION

SATELLITE EPHEMERIDES

SATELLITE MASK ANGLE

DATA GENERATION

TIME SATELLITE POSITIONS

GENERATE MEASUREMENT NOISE + BIAS ERRORS

Q-R TRANSFORMATIONS

MEASUREMENT NOISE MEASUREMENT BIASES

$P_{\text{MD}}$

$T_D$

$P_{\text{FA}}$

$\sigma$

CALCULATE $M_{\text{MIN}}$

$R_T$

$P_T$
82 degrees west longitude, which is a location in the vicinity of Ohio University.

The procedure for data generation at each moment of time is as follows:

1. Calculate the satellite positions in Earth-Centered-Earth-Fixed (ECEF) coordinates based on the Primary 21-satellite constellation for the current time.
2. Convert the satellite positions to East-North-Up (ENU) coordinates with respect to the user position.
3. Calculate the satellite elevation angle with respect to the user and determine the satellite visibility.
4. Obtain an array of visible satellites and the pseudorange measurements. This array of data is used for the fault detection simulation.

**Fault Detection Program**

This program calculates two important parameters: the detection threshold $T_0$, and the minimum required expected value of the detection parameter $m_{\text{Min}}$, see Appendix B for a program listing. These parameters are calculated as a function of the number of measurements. The calculation of the detection threshold involves the probability of a false alarm, and the measurement noise, see equation (3.34). The minimum required expected value of the detection parameter is
calculated from the detection threshold, the measurement noise, and required probability of a missed detection, see equation (3.29). Note that the probabilities of a false alarm and the probability of a missed detection are constant values based on known requirements.

Next, measurement noise and measurement bias errors are added to the pseudorange measurement data, and the parity vector $p$ is derived using the Q-R decomposition. The elements of the parity vector are then compared with the pre-calculated detection threshold. If one of the elements of the parity vector is greater than the detection threshold, a fault is detected and the alarm is raised.

This is followed by the calculation of the horizontal radial protection radius $R_p$, which is the sum of $R_{bias}$ and $R_{noise}$. The horizontal radial position noise error $R_{noise}$ is derived from the probability of exceeding the horizontal radial alarm threshold $R_T$, the measurement geometry characterized by the HDOP, and the measurement noise, see equation (3.41). The horizontal radial position bias error $R_{bias}$ is calculated from the minimum required measurement bias error for detection with the required detection probability, see equation (3.40).

The horizontal radial protection radius is important to determine the availability of the fault detection algorithm.
The outcome of the detection algorithm is compared with the actual horizontal radial position error which results in four different integrity conditions, see Figure 3.2.

4.2 GPS Constellation

The simulation for the fault detection algorithm uses the primary 21-satellite constellation which is anticipated to be completed around 1993. This constellation consists of 24 satellites with 4 satellites in each of six 55 degrees inclined equally spaced orbital planes [G. B. Green, et al, 1989]. The phasing diagram for this constellation is given in Figure 4.2. This constellation was optimized to provide the best possible coverage in the event of any single satellite failure. Furthermore, this constellation will provide at least 21 operational satellites at all times. It will be maintained by a launch-on-schedule policy. A new satellite will be launched approximately once every other month during constellation build-up, and then once every 3 months for constellation maintenance. The initial conditions for the 21 Primary Satellite Constellation are given in Table 4.1, where $a$ is the square root of the semi-major axis, $e$ is the eccentricity, $i$ is the inclination angle, $\Omega$ is the right ascension of the ascending node, $\omega$ is the argument of perigee, and $M$ is the mean anomaly.
Figure 4.2  Satellite Distribution for the 21 Primary Satellite Constellation

[G. B. Green, et al, 1989]
Table 4.1 21 Primary Satellite Constellation Ephemerides

<table>
<thead>
<tr>
<th>SV</th>
<th>a</th>
<th>e</th>
<th>i</th>
<th>Ω</th>
<th>ω</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>325.730284</td>
<td>0</td>
<td>190.96</td>
</tr>
<tr>
<td>A2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>325.730284</td>
<td>0</td>
<td>220.48</td>
</tr>
<tr>
<td>A3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>325.730284</td>
<td>0</td>
<td>330.17</td>
</tr>
<tr>
<td>A4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>325.730284</td>
<td>0</td>
<td>83.58</td>
</tr>
<tr>
<td>B1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>25.7302839</td>
<td>0</td>
<td>249.90</td>
</tr>
<tr>
<td>B2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>25.7302839</td>
<td>0</td>
<td>352.12</td>
</tr>
<tr>
<td>B3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>25.7302839</td>
<td>0</td>
<td>25.25</td>
</tr>
<tr>
<td>B4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>25.7302839</td>
<td>0</td>
<td>124.10</td>
</tr>
<tr>
<td>C1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>85.7302839</td>
<td>0</td>
<td>286.20</td>
</tr>
<tr>
<td>C2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>85.7302839</td>
<td>0</td>
<td>48.94</td>
</tr>
<tr>
<td>C3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>85.7302839</td>
<td>0</td>
<td>155.08</td>
</tr>
<tr>
<td>C4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>85.7302839</td>
<td>0</td>
<td>183.71</td>
</tr>
<tr>
<td>D1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>145.730284</td>
<td>0</td>
<td>312.30</td>
</tr>
<tr>
<td>D2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>145.730284</td>
<td>0</td>
<td>340.93</td>
</tr>
<tr>
<td>D3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>145.730284</td>
<td>0</td>
<td>87.06</td>
</tr>
<tr>
<td>D4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>145.730284</td>
<td>0</td>
<td>209.81</td>
</tr>
<tr>
<td>E1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>205.730284</td>
<td>0</td>
<td>11.90</td>
</tr>
<tr>
<td>E2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>205.730284</td>
<td>0</td>
<td>110.76</td>
</tr>
<tr>
<td>E3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>205.730284</td>
<td>0</td>
<td>143.88</td>
</tr>
<tr>
<td>E4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>205.730284</td>
<td>0</td>
<td>246.11</td>
</tr>
<tr>
<td>F1</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>265.730284</td>
<td>0</td>
<td>52.42</td>
</tr>
<tr>
<td>F2</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>265.730284</td>
<td>0</td>
<td>165.83</td>
</tr>
<tr>
<td>F3</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>265.730284</td>
<td>0</td>
<td>275.52</td>
</tr>
<tr>
<td>F4</td>
<td>26609.0</td>
<td>0</td>
<td>55</td>
<td>265.730284</td>
<td>0</td>
<td>305.04</td>
</tr>
</tbody>
</table>

* Classical Coordinates-Km/deg.
5. COMPUTER SIMULATION RESULTS

The fault detection algorithm is designed to guarantee a minimum detection probability at all times and locations. The parameters used for the fault detection simulation are summarized below:

- Measurement noise standard deviation: \( \sigma = 32 \) (meters)
- Probability of a false alarm: \( P_{FA} = 10^{-6} \) (per sample)
- Probability of a missed detection: \( P_{MD} = 10^{-3} \) (per sample)
- Probability of exceeding the horizontal radial threshold: \( P_T = 10^{-9} \) (per sample)
- Location: 39° North latitude, 82° West longitude

Based on the above parameters, the detection threshold \( T_0 \) and the horizontal protection radius \( R_p \) are calculated for each time/space point.

5.1 Satellite Geometry

The performance of the fault detection algorithm is a strong function of the satellite geometry. Figure 5.1 shows the satellite geometry in terms of the horizontal dilution of precision (HDOP) as a function of time. The average value of the HDOP is approximately 1.1, but the HDOP can be small as 0.8 during periods of good satellite geometry, and as large as
Figure 5.1 HDOP during 24 Hours
1.65 during periods of marginal satellite geometry. Statistically, the rms horizontal position error is the product of the HDOP and the rms range error.

5.2 Alarm Rate as a Function of the Measurement Bias Error

The alarm rate was determined as a function of the measurement bias error. The measurement bias error was added to one randomly selected satellite. The integrity alarm is triggered when one of the detection parameters exceeds the detection threshold. The corresponding alarm rate is calculated from

$$\text{Alarm Rate}(\%) = \frac{\text{Number of Alarms}}{\text{Total Number of Samples}} \times 100 \quad (5.1)$$

Figure 5.2 shows the alarm rate as a function of the measurement bias error. The bias error was selected in the range from 0 to 1200 meters, in 100 meters increments. The simulation was repeated 144 times for each value of the measurement bias error. As expected, no alarms were triggered in the absence of a measurement bias error. The alarm rate increases sharply for measurement bias errors greater than 200 meters. It should be noted that the meaning of an alarm can only be evaluated if the horizontal radial alarm limit is specified.
Figure 5.2 Alarm Rate
5.3 Protected Horizontal Radial Position Error

One important outcome of the fault detection algorithm is the predicted protection radius $R_p$. Two cases are considered:

(1) Measurement noise only.
(2) Measurement noise on all measurement and a measurement bias error on one measurement.

Figure 5.3 shows the protected horizontal radial position error in the presence of measurement noise only. For each moment of time, the probability of exceeding the protected radius is less than $10^{-9}$. For this case, the protection radius is directly related to the satellite geometry, see also Figure 5.1.

Figure 5.4 shows the protection radius in the presence of both measurement noise on all measurement and a measurement bias error on one measurement. In this case, the detection algorithm must detect a possible measurement bias error, which increases the protection radius significantly. For each moment of time, the probability of exceeding the protected radius is less than $10^{-3}$. This probability multiplied by the probability of a measurement bias error of $10^{-6}$ results in the required probability of $10^{-9}$ for the overall missed detection probability. Note that there is still a high correlation
Figure 5.3  Protected Horizontal Radial Position Error  
(In the presence of measurement noise only)
Figure 5.4  Protected Horizontal Radial Position Error
(In the presence of both measurement noise on all measurements and a measurement bias error on one measurement)
between the protected radius and HDOP, but this can not be guaranteed at all times. For instance, the first peak in Figure 5.4 does not correspond to a peak in the HDOP in Figure 5.1.

Clearly, the protection radius given Figure 5.4 must be used for the fault detection. This means that Figure 5.4 can be used to determine the availability of the fault detection algorithm. First it is observed that the desired horizontal radial alarm threshold $R_r$ must be greater than the protected horizontal radial position error, $R_p$, to guarantee the performance of the detection algorithm. For example, if the selected $R_r$ is 200 meters, then the detection algorithm can never be used because $R_p$ is always greater than $R_r$. However, if the selected $R_r$ is 400 meters, then the detection algorithm can be used approximately 50 percents of the time because half of the $R_p$s are less than $R_r$.

5.4 Relation between the Actual Position Error and the Protected Position Error.

To gain more insight into the relation between the horizontal radial position error and the protected horizontal radial position error, the effect of several measurement bias errors are presented on Figures 5.5 through 5.9. In the absence of measurement bias error, the protected position
Figure 5.5  Actual Position Error vs Protected Position Error

(Measurement bias error = 0 Meters)
Figure 5.6  Actual Position Error vs Protected Position Error
(Measurement bias error = 300 Meters)
Figure 5.7  Actual Position Error vs Protected Position Error

(Measurement bias error = 600 Meters)
Figure 5.8 Actual Position Error vs Protected Position Error

(Measurement bias error = 900 Meters)
Figure 5.9 Actual Position Error vs Protected Position Error
(Measurement bias error 1200 Meters)
error is always larger than the actual position error, see Figure 5.5. The introduction of a significant measurement bias error increases both the protected position error and the actual position error. Note that a detection must take place with a 99.9 percents probability whenever the actual position error exceeds the protected position error. This occurs quite frequently for a measurement bias error of 1200 meters, see Figure 5.9. Fortunately, it is relatively easy to detect such a large measurement bias error. This is further discussed in the next Section.

5.5 Performance of the Fault Detection Algorithm

As discussed in Section 3.2.2, the integrity algorithm has four possible outcomes given that the algorithm is available ($R_t > R_p$). The probability of the possible outcomes are experimentally determined by:

$$P_{NO} = \frac{\text{Number of Normal Operations}}{\text{Number of Available Experiments}}$$  \hfill (5.2)

$$P_{MD} = \frac{\text{Number of Missed Detections}}{\text{Number of Available Experiments}}$$  \hfill (5.3)

$$P_{FA} = \frac{\text{Number of False Alarms}}{\text{Number of Available experiments}}$$  \hfill (5.4)

$$P_{TA} = \frac{\text{Number of True Alarms}}{\text{Number of Available Experiments}}$$  \hfill (5.8)
Because of the limited number of samples used (144 for each experiment), no missed detections and no true alarms were experienced. Figure 5.10 shows the probability of normal operation as a function of the horizontal alarm threshold for different measurement bias errors. In the absence of measurement bias errors, normal operation is close to 100 percent if the horizontal alarm threshold is set 300 meters or greater. Figure 5.11 shows the corresponding False Alarm rate. From the comparison of Figure 5.10 and 5.11, it is concluded that the introduction of measurement bias errors increases the false alarm rate. Fortunately, the probability of a measurement bias error is small ($10^{-6}$) and therefore, the availability of the fault detection algorithm is not significantly decreased by measurement bias errors. Furthermore, if a measurement bias error exists, it would be safe to detect this error even though it does not cause an unacceptable position error.

5.6 Availability of Navigation Information

Position information can only be used if the following two condition are met:

(1) The fault detection algorithm is available ($R_f > R_p$).
(2) No alarm exists.
Figure 5.10 Normal Operation
Figure 5.11 False Alarm
The position availability is calculated using the following equation:

\[
\text{Availability} = \frac{\text{No. of Available Experiments} - \text{No. of Alarms}}{\text{Total No. of experiments}}
\]

(5.6)

Figure 5.12 shows the position availability as a function of the horizontal radial alarm threshold for different measurement bias error. In the absence of a measurement bias error, the horizontal alarm threshold must be set at 1000 meters to achieve 100 percents availability.

With reference to Table 2.1, the following approximate availability of navigation information can be achieved for the different phase of flight:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Alarm Threshold</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Precision Approach</td>
<td>555 meters</td>
<td>86 %</td>
</tr>
<tr>
<td>Terminal</td>
<td>1852 meters</td>
<td>100 %</td>
</tr>
<tr>
<td>Enroute</td>
<td>3704 meters</td>
<td>100 %</td>
</tr>
</tbody>
</table>

Based on the above results it is concluded that Receiver Autonomous Integrity Monitoring is an acceptable method for achieving GPS integrity for all phases of flight, except for the non-precision approach. Additional measurements are needed to increase the availability for the non-precision approach.
Figure 5.12  Availability of Position Information
6. CONCLUSIONS

Receiver Autonomous Integrity Monitoring is currently the most promising method to achieve GPS integrity for supplemental navigation within the National Airspace System (NAS). A receiver autonomous fault detection algorithm is presented which satisfies preliminary RTCA requirement for supplemental navigation. The performance of the fault detection algorithm is confirmed using a computer simulation.

Based on the results presented in this paper, the following conclusions are provided:

(1) A GPS receiver should use two types of fault detection algorithms in parallel; a recursive estimator to detect rapidly growing measurement errors, and a least square batch estimator to detect slowly growing measurement errors which are not detected by the recursive estimator.

(2) To satisfy the integrity requirements for a supplemental navigation system, a Receiver Autonomous Integrity Monitoring (RAIM) algorithm must use constant values for the probability of a false alarm ($P_{FA}$) and the probability of a missed detection ($P_{MD}$).
(3) Once the probability of a false alarm ($P_{fa}$) and the probability of a missed detection ($P_{md}$) are specified, then a protected horizontal radial position error ($R_p$) can be calculated as a function of the satellite geometry and measurement noise.

(4) Both the alarm rate and the availability are strongly affected by large measurement bias errors (>900 meters). This is acceptable because the probability of a large measurement bias error is very small ($10^{-6}$).

(5) Receiver Autonomous Integrity Monitoring can provide integrity with a high availability for all phases of flight, except for the non-precision approach. Additional measurements are needed to increase the availability for the non-precision approach.
7. REFERENCES


APPENDIX A

"Data Generation Program Listing"
% FDADAT: (Fault Detection Algorithm DATa) % Generates GPS data arrays for FDA simulation %

% The following script files/functions are used:
% svpri.mat -- Contains SV ephemeris data (21 PRI)
% llh2ec.m -- Converts from lat/lon/height to ECEF
% ec2enu.m -- Converts from ECEF to East/North/Up
% enu2ec.m -- Converts from East/North/Up to ECEF
% kepler.m -- Computes ECEF satellite positions
% OU-ECE-AEC December 1990 Choi, Sang-Sung

% Define the start, stop increase time for data generation

time_start = 600;  % start time (seconds)
time_stop = 86400;  % stop time (seconds)
time_incr = 600;  % increase time (seconds)

% Define the mask angle and conversion factors

mask_angle = 5.0;  % Satellite elevation mask angle (deg.)
degrad = pi/180;  % Degrees/radians conversion factors
raddeg = 1./degrad;  % Radians/degrees conversion factors

% Load and convert the Satellite ephemeris data

load svpri;  % Load svpri.mat
svdat = svpri;  % Satellite ephemeris data
svno = max(size(svdat));  % Number of satellites

% Change initial LLH position to ECEF position

org_llh = [39 -82 0];  % Initial LLH position
org_b = 0;  % Clock offset
org_llh(1) = org_llh(1)*degrad;
org_llh(2) = org_llh(2)*degrad;
org_ecef = llh2ec(org_llh);  % Initial ECEF position

% Initialize the each variables

index = 0;
for time = time_start:time_incr:time_stop
    index = index + 1;
end

index_max = index;
number = zeros(1:index_max);  % Number of Satellite
time_run = zeros(1:index_max); % Run time
sv_pos = zeros((4*index_max),12); % Satellite position

% Display the running program & running time
fprintf('
 FDADAT Running...

')
index = 1; % Begin main loop
for time = time_start:time_incr:time_stop
fprintf('The time=%6.1f(stop time=%6.1f)
',time,time_stop);
% Calculate the satellite ECEF positions for the current time
for i = 1:svno
  i0 = svdat(i,4) * degrad;
on0 = svdat(i,2) * degrad;
m0 = svdat(i,1) * degrad;
radius = svdat(i,3);
ephtim = svdat(i,5);
reftim = time;
satpos = kepler(i0,on0,m0,radius,ephtim,reftim);
sv_ecef(i,1:3) = satpos(1:3);
% save ECEF positions
end
% Convert SV position to ENU with respect to user position
for i = 1:svno
  svvis(i) = 0; % 1 = visible, 0 = not visible
  svenu = ec2enu(sv_ecef(i,1:3),org_ecef,org_llh);
  sv_enu(i,1:3) = svenu(1:3);
% Calculate the elevation angle and determine visibility
  angle = atan2(svenu(3),norm(svenu(1:2)));
svelv(i) = angle * raddeg;
if (svelv(i) >= mask_angle),
  svvis(i) = 1;
end
end
% Obtain an array of visible satellites and the pseudoranges
no = 0;
for i = 1:svno
  if svvis(i)
    no = no + 1;
    pr(index,no) = norm(sv_enu(i,1:3)) + org_b;
    svinfo(1,no) = (-1*sv_enu(i,1))/(pr(index,no)-org_b);
    svinfo(2,no) = (-1*sv_enu(i,2))/(pr(index,no)-org_b);
  end
end
svinfo(3,no) = (-1*sv_enu(i,3))/(pr(index,no)-org_b);
svinfo(4,no) = 1;
end
end

number(index) = no;            % Save number of visible SVs

% Store all relevant data
sv_pos(((index*4)-3):(index*4),1:no) = svinfo(1:4,1:no);

% Perform the next iteration
index = index + 1;
end

save fdadat time_start time_stop time_incr number sv_pos
function ecef = llh2ec(llh)

A = 6378135.0; % Earth's radius (m)
E = 8.181881066e-02; % Eccentricity
ESQ = E * E;

SP = sin(llh(1));
CP = cos(llh(1));
SL = sin(llh(2));
CL = cos(llh(2));
GSQ = 1.0 - (ESQ*SP*SP);
EN = A / sqrt(GSQ);
Z = (EN + llh(3)) * CP;
ecef(1) = Z * CL;
ecef(2) = Z * SL;
EN = EN - (ESQ * EN);
ecef(3) = (EN + llh(3)) * SP;
function ecef = enu2ec(enu, orgece, orgllh)

sla = sin(orgllh(1)); cla = cos(orgllh(1));
slo = sin(orgllh(2)); clo = cos(orgllh(2));

ROT = [-slo clo 0; -sla*clo -sla*slo cla; cla*clo cla*slo slo sla];
difece = inv(ROT) * enu';

% add the difference between the enu and ecef origins

ecef = orgece + difece';
function enu = ec2enu(ec, orge, orgllh)

dife = ec - orge;  \% difference between coordinate origins

\% Rotate the difference vector into ENU coordinates
sla = \sin(\text{orgllh}(1)); cla = \cos(\text{orgllh}(1));
slo = \sin(\text{orgllh}(2)); clo = \cos(\text{orgllh}(2));

enu = [ -slo \quad clo \quad 0 ; \quad ... \\
-sla*clo \quad -sla*slo \quad cla; \quad ... \\
\quad cla*clo \quad cla*slo \quad sla ] * dife';
enu = enu';
function satpos = kepler(i_0, Omega_0, M_0, radius, time_oe, time)

GM = 3.986008e14; % earth's gravitational constant (m^3/s^2)
omega_earth = 7.292115147e-05; % earth's rotation rate (rad/s)

n_zero = sqrt(GM/(radius*radius*radius)); % mean motion

time_tk = time - time_oe; % time from ephemeris reference epoch

M_k = M_0 + (n_zero * time_tk); % mean anomaly

orbit(1) = radius * (cos(M_k)); % x position in orbital plane
orbit(2) = radius * (sin(M_k)); % y position in orbital plane

Omega_k = Omega_0 - (omega_earth * time); % longitude of the ascending node

satpos = [cos(Omega_k) -cos(i_0)*sin(Omega_k); ... %ECEF
          sin(Omega_k) cos(i_0)*cos(Omega_k); ... %coordinates
          0 sin(i_0)] * orbit';
"Fault Detection Program Listing"
% FDASIM1 : (Fault Detection Algorithm SIMulation 1)
% Simulates the fault detection algorithm
% * HDOP
% * Protected radial position error
% The following script files/functions are used;
% fdadat.mat -- Contain the GPS measurement data
% parmet.mat -- Contain the detection threshold & minimum required expected value
% OU-ECE-AEC January 1991 Choi, Sang-Sung

% Load the data
load fdadat; % Load GPS measurement data
load parmet; % Load the threshold & min. expected value

% Begin the main loop
index = 0; % Main loop index

% Display the running program & running time
fprintf('

FDA Simulation Running ..... 

')
for time = time_start:time_incr:time_stop
fprintf('The Time=%6.1f(Stop Time=%6.1f)\015',time,time_stop)
index = index + 1;
run_time(index) = time;
Time(index) = time/3600;

% Convert no & H from the fdadat.mat
clear H;
no = number(index); % Number of visible SVs
H(1:no,1:4) = sv_pos(((index*4)-3):(index*4),1:no)'

% Generate the bias of the failure satellite
bias = 300;
rand('uniform'); % Uniform random variable
abc = rand(1,1);
sv = round(no*abc+0.5);
sv_bias = zeros(no,1);
sv_bias(sv,1) = bias;  \quad \% \text{Measurement bias error}

\% Generate the measurement noise & total measurement error

sigma = 32; \quad \% \text{Standard deviation}
rand('normal'); \quad \% \text{Normal random variable}
rand('seed',145);
sv_noise = sigma*rand(no,1); \quad \% \text{Measurement noise error}
sv_err = sv_noise + sv_bias; \quad \% \text{Total measurement error}

\% Q-R decomposition

[Q,R] = qr(H); \quad \% \text{Q-R decomposition}
AB = Q'; \quad \% \text{Transpose matrix Q}
A = AB(1:4,1:no); \quad \% \text{Upper part of matrix Q}
B = AB(5:no,1:no); \quad \% \text{Lower part of matrix Q}
upper_H = (R(1:4,1:4)); \quad \% \text{Inverse of upper triangular H}
gen_H = inv(upper_H)*A; \quad \% \text{Generalized H}

\% Find the actual position error

pos_err = gen_H*sv_err; \quad \% \text{Actual position error}
hp_err(index) = norm(pos_err(1:2,1)); \quad \% \text{Horizontal projection error}

\% Find orthogonal residual

or_resid = B*sv_err; \quad \% \text{Orthogonal residual}

\% P-Z decomposition

for j = 1:no
    temp = B;
    temp(1:no-4,1) = B(1:no-4,j); \quad \% switch j to 1
    temp(1:no-4,j) = B(1:no-4,1); \quad \% switch 1 to j
    [P,Z] = qr(temp(1:no-4,1:no)); \quad \% \text{P-Z decomposition}
    det(j,1:no) = Z(1,1:no); \quad \% define det
    det(j,1) = Z(1,j); \quad \% switch 1 to j
    det(j,j) = Z(1,1); \quad \% switch j to 1
end

\% Find the fault detection parameter

d(j) = det(j,1:no)*sv_err; \quad \% \text{Detection parameter}
r(j) = det(j,j); \quad \% 1st row & column of Z
end

\% Find the radial bias error

for k = 1:no
    h_proj(1,k) = norm(gen_H(1:2,k));
end
for m = 1:no
    b = zeros(no,1);  % Initialize the b matrix
    b(m) = um(no)/abs(r(m));  % Minimum bias error
    bias(m) = b(m)*h_proj(1,m);  % Horizontal projection-
        % bias error
end

R_bias = max(bias);  % Radial bias error

% Find the HDOP

dop = inv(upper_H'*upper_H);
sq_xdop = dop(1,1);  % Square XDOP
sq_ydop = dop(2,2);  % Square YDOP
HDOP = sqrt(sq_xdop + sq_ydop);  % HDOP
HDOP_data(index) = HDOP;

% Find the radial noise error

% R_noise = 6.1*sigma*HDOP;  % Radial noise error
R_noise = 3.29*sigma*HDOP;

% Find total radial error

% R_total(index) = R_noise;
R_total(index) = R_bias + R_noise;

end

% Save the HDOP and the total radial position errors

Data1 = Time';
Data2 = R_total';
pdata = [Data1, Data2];
hdata = [Data1, HDOP_data'];

save temp3.dat pdata -ascii
save temp4.dat hdata -ascii
% FDASIM2 : (Fault Detection Algorithm SIMulation 2)
% Simulates the fault detection algorithm
% * Compare actual position error and
% protected position error
% The following script files/functions are used:
% fdadat.mat -- Contain the GPS measurement data
% parmet.mat -- Contain the detection threshold
% & minimum required expected value
% % OU-ECE-AEC January 1991 Choi, Sang-Sung

% Load the data
load fdadat; % Load GPS measurement data
load parmet; % Load the threshold & min. expected value

% Begin the main loop
index = 0; % Main loop index

% Display the running program
fprintf('

 FDA simulation Running ...... \n\n')
for time = time_start:time_incr:time_stop
fprintf('The Time=%6.1f(Stop Time=%6.1f)\015',time,time_stop)
    index = index + 1;
    run_time(index) = time;
    Time(index) = time/3600;

% Convert no & H from the fdadat.mat
    clear H;
    no = number(index);
    H(1:no,1:4) = sv_pos(((index*4)-3):(index*4),1:no);

% Initialize the bias of the failure satellite
    bias = 0;
    rand('uniform');
    abc = rand(1,1);
    sv = round(no*abc+0.5);
    sv_bias = zeros(no,1);
    sv_bias(sv,1) = bias;
% Generate the measurement noise & total measurement error

sigma = 32;
rand('normal');
rand('seed',145);
sv_noise = sigma*rand(no,1);
sv_err = sv_noise + sv_bias;

% Q-R decomposition

[Q,R] = qr(H);
AB = Q';
A = AB(1:4,1:no);
B = AB(5:no,1:no);
upper_H = (R(1:4,1:4));
gen_H = inv(upper_H)*A;

% Find the actual position error

pos_err = gen_H*sv_err;
hp_err(index) = norm(pos_err(1:2,1));

% Find orthogonal residual

or_resid = B*sv_err;

% P-Z decomposition

for j = 1:no
    temp = B;
    temp(1:no-4,1) = B(1:no-4,j);
    temp(1:no-4,j) = B(1:no-4,1);
    [P,Z] = qr(temp(1:no-4,1:no));
    det(j,1:no) = Z(1,1:no);
    det(j,1) = Z(1,j);
    det(j,j) = Z(1,1);
end

% Find the fault detection parameter

d(j) = det(j,1:no)*sv_err ;
r(j) = det(j,j);
end

% Find the radial bias error

for k = 1:no
    h_proj(1,k) = ncrm(gen_H(1:2,k));
end

for m = 1:no
    b = zeros(no,1);
    b(m)= um(no)/abs(r(m));
end
bias(m) = b(m) * h_proj(1,m);
end

R_bias = max(bias);

% Find the HDOP

dop = inv(upper_H'*upper_H);
sq_xdop = dop(1,1);
sq_ydop = dop(2,2);
HDOP = sqrt(sq_xdop + sq_ydop);
HDOP_data(index) = HDOP;

% Find the radial noise error

R_noise = 6.1*sigma*HDOP;
% R_noise = 3.29*sigma*HDOP;

% Total radial error

R_total(index) = R_noise;
% R_total(index) = R_noise + R_bias;
end

% Save actual position error and protected position error

Data = [hp_err', R_total']
save temp1.dat Data -ascii
FDASIM3: (Fault Detection Algorithm SIMulation3)

Simulates the fault detection algorithm
* Alarm & No Alarm
* Four Different Integrity Condition
* Availability

The following script files/functions are used:
- fdadat.mat -- Contain the GPS measurement data
- parmet.mat -- Contain the detection threshold & minimum required expected value

OU-ECE-AEC January 1991 Choi, Sang-Sung

% Change the measurement bias error
for w = 0:300:1200
    bias = w
    y = (w+300)/300;
end

% Change the horizontal alarm threshold
for s = 0:100:1200
    R_thresh = s
    t = (s+100)/100;
end

% Load the data
load fdadat;
load parmet;

% Initialize each index
index = 0; % Main loop index
alarm = 0; noalarm = 0; % Number of alarm & noalarm
missed = 0; normal = 0; % Number of normal & missed detection
true = 0; false = 0; % Number of true & false alarm
avail = 0; % Number of available for the GPS
count = 0;

% Display the running program & running time
fprintf('

 FDA Simulation Running ..... 

')
for time = time_start:time_incr:time_stop
    fprintf('The Time=%6.1f(Stop Time=%6.1f)\015',time,time_stop)
end
index = index + 1;

run_time(index) = time;
Time(index) = time/3600;

% Convert no & H from the fdadat.mat

clear H;
no = number(index);
H(1:no,1:4) = sv_pcs(((index*4)-3):(index*4),1:no)';

% Initialize the bias of the failure satellite

rand('uniform');
abc = rand(1,1);
sv = round(no*abc+0.5);
bias = w;
sv_bias = zeros(no,1);
sv_bias(sv,1) = bias;

% Generate the measurement noise & total measurement error

sigma = 32;
rand('normal');
rand('seed',145);
sv_noise = sigma*rand(no,1);
sv_err = sv_noise + sv_bias;

% Q-R decomposition

[Q,R] = qr(H);
AB = Q';
A = AB(1:4,1:no);
B = AB(5:no,1:no);
upper_H = (R(1:4,1:4));
gen_H = inv(upper_H)*A;

% Find the actual position error

pos_err = gen_H*sv_err;
hp_err(index) = norm(pos_err(1:2,1));

% Find orthogonal residual

or_resid = B*sv_err;

% P-Z decomposition

for j = 1:no
    temp = B;
    temp(1:no-4,1) = B(1:no-4,j);
    temp(1:no-4,j) = B(1:no-4,1);
% Find the fault detection parameter
\[ d(j) = \text{det}(j,1:no)*sv\_err \; \text{;} \]
\[ r(j) = \text{det}(j,j); \]
end

% Find the radial bias error
for k = 1:no
    h\_proj(1,k) = \text{norm}(\text{gen\_H}(1:2,k));
end

for m = 1:no
    b = \text{zeros}(no,1);
    b(m) = um(no)/abs(r(m));
    bias(m) = b(m)* h\_proj(1,m);
end

R\_bias = \text{max}(bias);

% Find the HDOP
\[ \text{dop} = \text{inv}(\text{upper\_H}'*\text{upper\_H}); \]
\[ \text{sq\_xdop} = \text{dop}(1,1); \]
\[ \text{sq\_ydop} = \text{dop}(2,2); \]
\[ \text{HDOP} = \text{sqrt}(\text{sq\_xdop} + \text{sq\_ydop}); \]
HDOP\_data(index) = HDOP;

% Find the radial noise error
\[ R\_noise = 3.29*\text{sigma*HDOP}; \]

% Find total radial error
\[ R\_thresh = s; \]
\[ R\_total(index) = R\_bias + R\_noise; \]
if R\_total(index) < R\_thresh
    avail = avail + 1;
end

% Detect the false alarm
fault = 0;
for k = 1:no
    if d(k) > \text{td}(no);
        fault = fault + 1;
end
sample = time_stop/time_incr; % number of sample
AVAILABILITY = count/sample; % Availability
ALARM = alarm/sample; % Prob. of alarm
NO_ALARM = noalarm/sample; % Prob. of no alarm
TRUE_ALARM = true/avail; % Prob. of TA
FALSE_ALARM = false/avail; % Prob. of FA
NORMAL = normal/avail; % prob. of NO
MISSED_DETECTION = missed/avail; % prob. of MD
AV(y,t) = AVAILABILITY
AL(y,t) = ALARM
NA(y,t) = NO_ALARM
NO(y,t) = NORMAL
MD(y,t) = MISSED_DETECTION
TA(y,t) = TRUE_ALARM
FA(y,t) = FALSE_ALARM

% Save the all data
save data AV AL NA NO MD TA FA
% Define the parameter for detection threshold

sigma = 32; % Standard deviation (meters)
pfa = 1e-06; % Probability of false alarm (per sample)
pmd = 1e-03; % Probability of miss detection (per sample)

% Display the running program

fprintf('
Calculating the detection threshold 

')

% Calculate the TD & maximum required expected value (um)

index = 1;
for n = 1:4
    td(index) = 0
    um(index) = 0
    index = index + 1;
end

index = 5;
td(index) = sigma*sqrt(2)*inverf(1-pfa)
um(index) = td(index) + sigma*sqrt(2)*inverf(1-2*pmd)

index = 6;
for n = 6:12
    td(index) = sigma*sqrt(2)*inverf(1-pfa/n)
um(index) = td(index) + sigma*sqrt(2)*inverf(1-2*pmd)
    index = index + 1;
end

% Save the parameters in the param.mat file

save paramet sigma pfa pmd td um
ACKNOWLEDGEMENTS

This work was funded in part by the State of Ohio Avionics Academic Challenge Program. Additional funding was obtained from the Federal Aviation Administration under contract DTR S-57-87-c-00006, Technical Task Directive Number 46.

The author wishes to express much gratitude to Dr. Frank van Graas, Assistant Professor of Electrical and Computer Engineering, whose insight and suggestion are an integral part of this thesis.

The author also wishes to acknowledge the following individuals: Dr. John A. Tague, Assistant Professor of Electrical and Computer Engineering, Dr. Jeffrey C. Dill, Assistant Professor of Electrical and Computer Engineering, and Dr. Seung Soo Yun, Professor of Physics for serving as member of the thesis committee.

Finally, author wishes to make a grateful acknowledgment to his wife, Hye-Ree, for her patience and understanding of many things during the years of his study.