1988

Self-contained GPS integrity monitoring using a censored Kalman filter

Paul William McBurney
Iowa State University

U·M·I
Accessing the World's Information since 1938

300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

Order Number 8825942

Self-contained GPS integrity monitoring using a censored Kalman filter

McBurney, Paul William, Ph.D.

Iowa State University, 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Self-contained GPS integrity monitoring using a censored Kalman filter

by

Paul William McBurney

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Electrical Engineering
and Computer Engineering

Major: Electrical Engineering (Control Systems)

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1988

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. An Overview of the Global Positioning System

The Global Positioning System (GPS) is a satellite navigation system which promises to improve greatly worldwide navigation accuracies in the military and civilian communities. The system is currently past the development stage and is partially deployed, making it possible for users to test performance under fairly normal operating conditions. The system is composed of three separate components: the space, the ground, and the user segments.

The space segment is the satellite constellation. Current plans are for 24 satellites divided into six rings. These rings are inclined at 55 degrees and distributed along the equator every 60 degrees as shown in Figure 1.1. The orbital period of a satellite is twelve sidereal hours. Each satellite transmits a unique time tagged signal that is also modulated with other information including ephemeris parameters used to calculate the satellite's position at the time the signal was sent. Also transmitted are status information on the health of the satellites, various correction factors, and data usable in developing an almanac describing approximate locations of the other satellites.

The ground segment consists of monitoring and transmitting stations spread judiciously around the continental United States (CONUS) and other locations

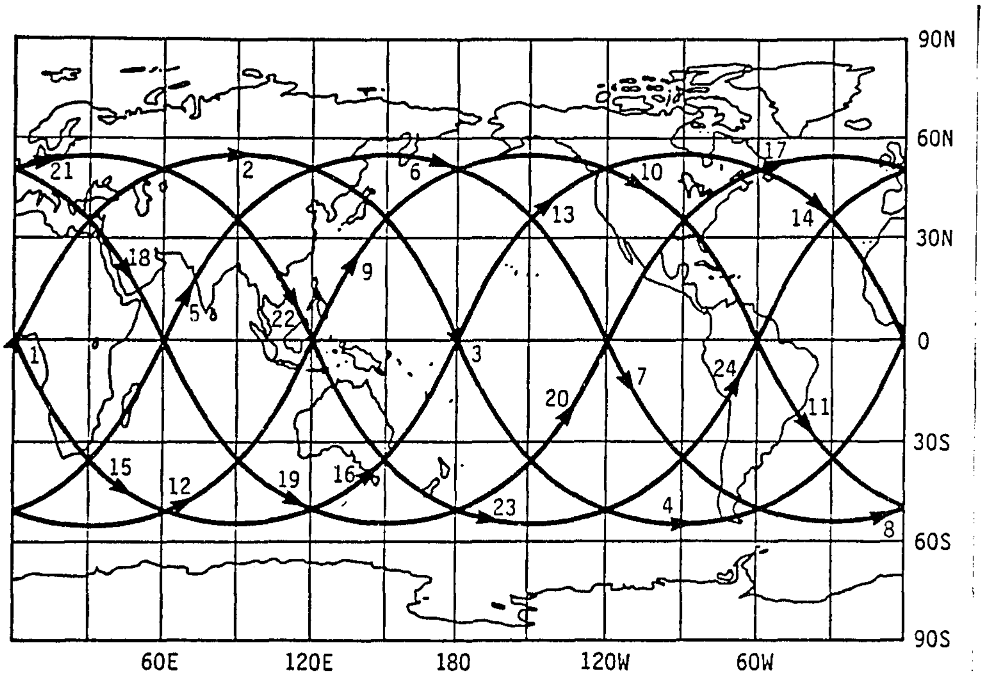


Figure 1.1: The proposed 24 satellite, six plane constellation

around the Earth. These stations monitor the satellites to make sure they are functioning properly. The monitors track the satellite transmissions very precisely and determine approximate orbital parameters describing the satellites' motion. These are then uplinked to the satellites periodically. The ground monitors make sure that transmitted ephemeris data matches the actual satellite orbit as well as check that the satellite clocks are sufficiently synchronized to the GPS time reference.

If ground control detects abnormalities, it can uplink new information to the satellite and also give commands to alter the satellite orbit. If a satellite is out of specification, ground control can change the health status message associated with that satellite so that users are warned about suspect information from that satellite.

The space and ground segments of the system are maintained by the Department of Defense (DoD), but the user segment has been made available to the civilian community.

The user segment consists of passive radio receivers which track the satellite signals to solve for their position. A set of four pseudorange measurements are needed to obtain a navigation solution. The pseudorange is basically proportional to the propagation time of the signal from the satellite to the receiver. The receiver measures the GPS time when it received the time tagged signal. By computing the difference between received and transmitted times, propagation time is measured and, when multiplied by the speed of light, the distance to the satellite is found. This distance is called a pseudorange because it is biased by an amount proportional to the offset of the receiver clock from GPS reference time. The satellite is synchronized to GPS time with an atomic clock which provides an extremely stable frequency reference. Most users cannot afford such quality but this is not a problem since the receiver can estimate its offset and, eliminate the effect of not being precisely calibrated to GPS time.

The receiver must solve for four unknowns: its position in three dimensions and its time offset from GPS time. Thus, at least four independent measurements must be available to get a unique solution. The pseudoranges are the measurements and the relative geometry of the satellites determine whether there is enough independence among the equations to get such a solution. During the normal situation, and because of the way the satellites are spread around the constellation, there will always be at least four satellites in view above the horizon. When four satellites are viewed, the user position and time offset will occur at the intersection of the hypersurfaces of position from the satellites. The error or noise associated with each pseudorange perturbs the solution from the true position and time offset.

If more than four satellites are used to get a solution, and the geometry is such that all the equations are linearly independent, then the solution is overdetermined and no position can be found which fits all the equations exactly. A solution can be found, though, which is a compromise among the equations. The most common is the least-squares solution which minimizes the sum-of-squares of the residuals associated with each equation when the solution is substituted back into the equation. A desirable effect of using the least-squares solution is that the error on each measurement is attenuated as more satellites are used.

In contrast to the pointwise or snapshot solution just described, many GPS receivers use recursive filters to process the pseudorange measurements. In these estimators, an attempt is made to model the time-wise correlation structure of the error sources, and, in this way, the effects of the noise can be filtered out and the position and time offset estimates can be improved. Typically, a Kalman filter is used and the performance criteria is to minimize the mean square estimation error of the parameters being estimated. In this setting, the filter can assimilate as many measurements as are available, and each is weighted according to its statistical worth (even during periods when four satellites are not available).

Granted that certain prior assumptions are satisfied with regard to the statistics of the noise processes, the filtered solution will be optimal in comparison to the snapshot solution. Of course, this is not true if the filter model deviates largely from truth at a certain time. Here the pointwise solution may provide a better solution. Thus both the filtered and snapshot approaches have some good and bad points. The snapshot estimate is not model dependent, but on the other hand it does not receive the benefit of any prior information. In contrast, the filtered estimate is model dependent, but it has the benefit of a certain amount of memory which can be valuable in reducing the effects of certain error sources.

Even with the limited number of satellites that are now in place, results of field tests have shown that the accuracy of GPS is somewhat better than the early designers had speculated. This is in part due to the advanced VLSI digital receivers which have very low inherent noise levels and also to the increased computing power which allows for processing more than four satellite measurements. Some receivers are multi-channel and can continuously track multiple satellites, while others are single channel and only dwell on a certain satellite long enough to obtain a good single pseudorange measurement and shift to another satellite, and so forth. The better than expected accuracy is also due to the availability of low-cost crystal oscillators which have good stability and thus provide a fairly stable receiver clock. Typical position accuracies are on the order of tens of meters for a reasonably priced receiver.

To the dismay of the civilian community, the Department of Defense has made the provision to degrade the intrinsic accuracy of the navigation signals which will be made available to the civil user. The DoD feels it would be against U.S. security interests to make the full accuracy capacity of this navigation system available to a possible adversary. This concern is addressed with policy of selective availability (SA) and will be implemented by altering the normal information from each satellite in a way which will add an error in each user's pseudorange measurement. These errors will supposedly decrease the accuracy of the user's solution to a level which would not threaten U.S. security. The U.S. military and other special users will be given the necessary information required to remove the effects ot SA. It turns out that for the civil user, it will be very difficult to reduce the effects of SA because the correlation structure of these processes will be for the most part, unknown. (Reasonable models have been proposed though and may be suitable for analysis purposes.) The civil users may be given the information that SA is activated through, through the bits in the navigation message that describe the available accuracy. Even so, if selective availability is implemented it will be the limiting factor in the accuracy of the user's solution.

1.2. Assessing the Goodness of the Navigation Solution

After one obtains a solution, it is appropriate to question how accurate the solution is. The most common way of analyzing the goodness of an estimator is to consider the variance of the estimate from the true parameter. The estimation error covariance matrix describes the dispersion of the estimates from the true values when the system is in the normal state. This matrix is normally computed for the snapshot solution and is part of the recursive algorithm in the Kalman filter. For example if the standard deviation of the estimation error in the north direction is known, then a 99% confidence interval may be constructed and if 100 experiments were performed, the true north position may be expected to be contained in the 100 confidence intervals 99% of the time. Thus in the normal situation, one can set up a reasonable bound on the estimation error. The normal situation is when the assumed statistics of the error sources match the situation at hand. If however an abnormal error is present somewhere in the system, the error covariance matrix does not correctly describe the actual statistics of the estimation errors, and the confidence interval will not contain the true parameter as often as it would when the model statistics match the actual statistics.

There are many sources of error which affect the ability of the position estimator to get a solution which is within a specified amount. Most of the error sources are known and are accounted for in some fashion; usually by developing a statistical model which allows for the effects of the errors to be diminished or removed. These error sources will be discussed in later chapters, but the error that is of greatest concern is an unannounced satellite clock error. Correction factors describing the normal satellite clock drift are transmitted in the navigation message from each satellite. This message is updated frequently enough to provide a good fit to the actual error structure, so this "error" is of no concern. Each satellite also has redundant clocks on board which will be used in the unlikely event that the main clock goes out of specification. An unannounced error occurs during the time when a satellite clock drifts out of specification and before ground control can either update new correction factors, or switch to a different clock, or set the health status bit for that satellite to an unusable status. During this time, the receiver continues to use the satellite as if it were operating properly, and this may force the user's solution out of specification without any warning that there is a problem.

The paramount effect of an error source is really the induced error in the user's solution. If one focuses on the civil aviation problem, the effect can be narrowed even further to the error induced in the user's horizontal solution. The aviation community still plans on retaining the altitude information obtained from the barometric altimeter. Although the altitude estimate derived from the altimeter measurement may be less accurate than the GPS estimate, it does provide good relative positioning with other aircraft which have calibrated their instruments to the same reference. There is also a tendency to not let go of older well-tried systems which people are accustomed to using.

1.3. Integrity Concerns for Civil Users of GPS

It is desirable to bound the estimation error during situations when there are abnormalities in the system. This brings up the issues of robustness, failure detection and system integrity. A robust estimator is usually sub-optimal with respect to the error model it is built around but is less sensitive to other unmodelled effects. Even if the estimator parameters are somewhat different than the actual parameters, there will be no serious degradation in performance if the estimator is robust. However, a robust estimator will still diverge from the truth if a large unmodelled error is present.

Many systems rely on failure detection schemes to perform an independent verification of the system performance. This is a very broad and diverse subject as evidenced by the coverage in the control systems literature [30]. Many schemes are based on a statistical test which is designed using the statistics under the normal situation. If the current statistics do not support the hypothesis that the system is operating normally, then the result is to claim that a failure is present. Other more complicated approaches go as far as to model certain failure modes (as well as the normal model) and then test which is more likely to be true given the current data [6]. (The problem is that there are generally an infinite number of possible failure types so a limited set which is fairly representative is generally hypothesized.) If the result of a test is that a failure is present, then many systems take the problem one step further and try to determine the error source. This is the identification or isolation problem. If a decision can be made with regard to the source, then the system can decide on an appropriate response which may include removing the faulty sensor from the system or using a back-up system.

System integrity refers to the ability of a system to detect malfunctions which would cause the signals it generates to be out of specification. A system might also be constrained to recognize such a condition in a limited amount of time and to notify the users, so that they do not rely on information which is out of tolerance for an extended period of time. Safety is the primary motivation for these requirements. The VOR/DME system is a prime example of a system with a high degree of integrity [2]. Self-checking tests are built directly into the equipment and, the equipment goes so far as to shut itself off if it detects that it is operating improperly. There is also a high degree a redundancy in the system in that if one station is down, there are still alternate stations which can be used so that there is no interruption of service. Instrument landing services such as ILS/MLS have similar characteristics. Out of tolerance signals are detected with a high probability and users are notified of the situation within 1-10s.

Many of these characteristics are present in GPS but some of them are still lacking [2],[14]. Redundancy is provided by the number of satellites which are visible at a given time. Only four are needed to get a solution but it will be typical to have as many as five to nine satellites visible, when the system is fully operational. In the event that ground control cannot communicate with the satellites, the design provides for a graceful degradation of accuracy in the parameters describing the satellite position and clock drift. The error-rate in reading the navigation message from each satellite is negligible due to the use of parity bits and the ability to re-read the message if an error is detected. There is also a certain degree of self-monitoring built directly into the satellites. If the satellite detects a malfunction and cannot rectify the situation, it can switch to transmitting a nonstandard code which prevents the receiver from using the satellite. Also, the ground control can declare a satellite "bad" for any reason, but the reaction time required to change the satellite message accordingly can be rather long.

The problems that may still exist pertain to the non-negligible probability that a satellite will fail to detect the malfunction. In this situation the satellite transmits faulty information before ground control can detect the problem and take appropriate action. This reaction time may be on the order of an hour. This is much longer than can be tolerated when compared with other systems with satisfactory integrity. It appears that an additional verification of integrity will be required by the civil community. Whether this will be done in the user equipment or at independent ground stations remains to be determined. (It is interesting that the military users of GPS do not have an integrity problem [2]. This is primarily due to use of inertial navigation systems with GPS which provide additional redundancy.)

Other concerns which blur the definition of system integrity are the many ways in which an error manifests itself in the users solution and in the different tolerances which may be acceptable in the many different navigation environments. Each receiver manufacturer may choose to process a different selection of satellites, as well as use a different algorithm with different prior statistics to calculate the navigation solution. As a result, the definition of an abnormality (i.e., a system failure) is not well defined. For example, in an oceanic flight environment the amount of error that is tolerable may be much larger than what would be allowed for a civil aviation user making a non-precision approach. Thus it is not fair to disable a satellite simply because its clock is drifting more than usual if the induced error in the user's solution is still within tolerance.

For the purpose of determining the requirements and goals of civil GPS integrity, the Integrity Working Group of the Radio Technical Commission for Aeronautics (RTCA) special committee 159 was formed, at the request of the Federal Aviation Administration (FAA), on April 22, 1986 [3]. This group first examined the integrity limits and time-to-alarm requirements of existing navigation systems and then used these to set down goals which would satisfy future navigation requirements. The requirements and goals which were determined for GPS are listed in Tables 1.1 and 1.2, as they appear in [3].

Table 1.1: GPS integrity requirements

Phase of flight	Oceanic enroute	Domestic enroute	Terminal area	Non-precision approach
Alarm limit	12.6nmi	1.5nmi	1.1nmi	0.3nmi
Time-to-alarm	120s	60s	15s	10s

Table 1.2: GPS integrity goals

Phase of flight	Oceanic enroute	Domestic enroute	Terminal area	Non-precision approach
Alarm limit	5km	1km	.5km	.1km
Time-to-alarm	30s	30s	10s	6s

A broad range of integrity monitoring techniques were analyzed to determine if these requirements could be satisfied to a degree which would allow GPS to be classified by the FAA as a Sole Means Navigation System in the National Airspace. Such a system would be required to provide continuous service, even in the event than one element in the system (in this case one satellite) has failed. This would require that each satellite in view could be removed while the remaining satellites would still provide a good solution. Timely warnings of the failure would also be required. It was decided by this committee that the 24 satellite GPS constellation with an additional integrity monitoring element could satisfy all GPS integrity requirements and goals except the non-precision approach goal, which is clearly the most demanding specification. This is in part due to the planned level of selective availability, which can induce errors comparable to the desired protection level.

The current techniques for integrity monitoring may be divided into two groups; the self-contained approach and the ground monitoring approach. The self-contained scheme implies that the integrity check is performed autonomously by each user. This method is also referred to as receiver autonomous integrity monitoring (RAIM) and this method is nothing more than a consistency check among the pseudorange measurements, and possibly the receiver clock or other navigation aiding sources. In the ground monitoring approach, the integrity check would be done at civilian ground stations and the results would be transmitted to each user via a separate communication link. This method is known mainly as the GPS Integrity Channel (GIC) [13]. This method takes advantage of the well surveyed location of the antenna to allow a direct check of the range to the satellite. If the estimated range does not match the known range, the GIC would send the message that the errant satellite is out of tolerance, or alternatively transmit correction factors.

2. PURPOSE AND DIRECTION OF RESEARCH

This work will attempt to develop a method by which the integrity of GPS signals-in-space may be assured during the most demanding civil aviation scenario. This would include the non-precision approach environment during which an aircraft may be performing a series of turns which induce vehicle accelerations of up to 10m/s^2 (1g) or even more. The presence of selective availability will also be assumed to represent a near worst-case scenario. The GPS signals-in-space specification refers to the requirement that the integrity algorithm be able to assure that all the satellite signals being used are within specification. An out-of-tolerance signal would induce a horizontal error which is larger than a specified protection level. The algorithm will also be limited to using only information that is available inside the GPS receiver. Information from other aiding sources would surely provide extra redundancy but the purpose of this work is to see if there is enough information to perform an integrity check using GPS by itself. An overview of previous work will first be presented to give a description of the current state of GPS integrity efforts. Then a new approach known as the censored Kalman filter will be presented which builds upon the results and shortcomings of the earlier works.

The motivation for the new method is to provide a stronger connection between the test statistic and estimation error in the horizontal plane which is the primary error of interest in the civil community. Most of the previous methods derive the test statistic in the measurement domain. These approaches are successful in detecting the presence of a single large error in one of the pseudorange measurements. The main difficulty is that the result does not give a direct connection to the error induced in the horizontal plane. The approach to be presented here will place the final integrity decision directly in the horizontal error domain. A brief qualitative description of the method follows here, and then a more detailed quantitative description is given in Chapter 5.

The main responsibility of the integrity monitoring algorithm is to perform a simple binary hypothesis test where the null and alternative hypotheses are defined as

$$\begin{aligned} H_0 \text{ hypothesis :} &\quad r \le r_0 \\ H_1 \text{ hypothesis :} &\quad r > r_0 \end{aligned}$$

where

$$\begin{aligned} r &= \text{radial error in the user solution} \\ r_0 &= \text{radial error protection level} \end{aligned}$$

In all the detection schemes, the signals that support H_1 originate in the measurement domain. It is in this domain that the statistics of the measurement residuals under the H_0 hypothesis are well known. In the censored Kalman filter approach, one first performs an initial binary hypothesis test in the measurement space. The H_0' hypothesis is that the measurement residuals are zero mean with known covariance and are time-wise uncorrelated (thus the residuals form a multivariate innovations sequence). The H_1' hypothesis is that the H_0' hypothesis is not true. If the data supports H_1', then it is possible to question if there is a deterministic way to remove the bias from the residuals so that the resulting residuals do support H_0'. This is where the censoring of the measurement data occurs. The single satellite failure assumption is made so only one measurement source is censored. After the deterministic inputs are found, one can then calculate the effect of removing these inputs from the navigation solution. At this point, the algorithm returns to the original hypothesis test in the horizontal plane.

Using the error covariance matrix from the Kalman filter under H_0, one can construct confidence regions about the two filter estimates: the pure filter estimate and the filter estimate whose residuals are forced to support H_0'. In general, the confidence regions could be in a hyperspace with dimension equal to that of the parameter space being estimated. For the problem at hand, we are only concerned with the horizontal radial error so the confidence regions will both lie in the horizontal plane. If the two confidence regions do not overlap, then it is unlikely (for given the test size) that both confidence regions can contain the true horizontal position. Using this criteria, an upper bound on the radial error can be formulated at which the confidence regions do not not overlap. If the confidence regions are disjoint, then the error would be larger than the amount allowed under H_0. The decision would then be to accept the H_1 hypothesis. If the statistics of the measurement residuals are highly improbable under H_0', then there should be a large distance between the two estimates and this is where the power of the test lies.

After the detection algorithm is specified, its performance is measured by the level of the horizontal error that is incurred before the detection algorithm can confidently and consistently determine that the system is out of specification. These classifications are best described in classical decision theory terms. Consider a simple hypothesis test where the decision is to choose between either the null hypothesis or the alternative hypothesis where the threshold is set for a given test size. The test size is the probability that the test supports the alternative hypothesis when in fact the null hypothesis is true. It is desirable to design an algorithm with as small a test size as possible. If the equipment raises the alarm too often, then the user may tend to ignore such warnings. This is also known as the false alarm rate or the probability of making a type-I error. Inversely related to the alarm rate is the probability of missed detection. This is the probability of supporting the null hypothesis when in fact the alternative hypothesis is true. This kind of error is also known as a type-II error. The test designer has to trade-off the test size against the miss probability. As the test size is decreased (which lowers the alarm rate since the threshold is increased) the miss probability is increased. The detection probability or the probability of making the correct decision is simply one minus the miss probability. Thus one can use these two descriptors interchangeably to define the same performance.

As in most hypothesis testing situations, the conditional probability density function given that the H_1 hypothesis is true is not well known or easily obtained. This density is required to compute the miss and detection probabilities. In this situation, one usually has to rely on empirical results obtained from Monte Carlo simulation to get an understanding of these parameters. To compound this problem, the equations which describe the measurement situation are time varying because of the motion of the satellites and the vehicle. One cannot just simulate one location for a limited time and extrapolate the results to all locations and satellite geometries. Different satellite constellations can drastically effect the satellite availability and the solution accuracy. The computational effort required to do a complete analysis is extremely large. Thus, one of the drawbacks of the work described here will be the limited amount of empirical data upon which the performance of the integrity algorithm will be based.

The false alarm rate, the miss or detection probability and the level of horizontal protection are the criteria by which the detection algorithms are compared for this problem. These are the parameters that are most easily calculated as a result of performing Monte Carlo simulation. There are other similar parameters which are more difficult to obtain as a result of not knowing the prior probabilities associated with certain events. For example, the probability that a satellite will fail and produce a bias of a given value in the pseudorange is unknown, even though it is very small. This probability is needed to compute the unconditional alarm rate, where the conditioning on the state of the system, i.e., whether it is failed or not, is removed. It is also desirable to know the probability that the alarm will be raised on a mission of a given duration. This is the alarm rate which is of greater importance to people who actually use the equipment. A discussion of these and similar questions are contained in Chapter 4.

As mentioned at the beginning of this chapter, the performance of the integrity algorithm will be dependent on the effects of selective availability and unmodelled vehicle acceleration, so these processes must also be part of the Monte Carlo simulations. In snapshot methods, the vehicle dynamics are suppressed and SA is accounted for by simply increasing the horizontal protection level to an amount which is outside of the range which SA could induce. In the Kalman filter approach, the process and measurement models are loosened in an attempt to provide a more robust filter design. The accelerations errors induced by aircraft maneuvers are more nearly deterministic than random and do not fit well into the process model of the Kalman filter. Thus, a more robust approach may be to not include acceleration states in the model, and to simply increase the process model white noise amplitudes instead. The presence of SA is acknowledged in the measurement model by increasing the value of the measurement noise variance and by leaving the time-wise correlation structure of SA unmodelled.

The body of this thesis is arranged as follows. Chapter 3 gives an overview of the governing measurement equations and the accompanying statistics of the snapshot and Kalman filter techiques. Details of the Monte Carlo simulations are also mentioned. Then a survey of the work that has been done in the integrity monitoring area is presented in Chapter 4. This work is classified into either filtered or snapshot techniques and the successes and shortcomings of each technique are addressed. This will lead to a discussion of the effects of poor satellite geometry on the detection algorithms and to a new scheme which takes advantage of the receiver clock stability to provide further redundancy during these periods. Chapter 5 presents a detailed description of the censored Kalman filter approach which builds upon most of the previous techniques. Concluding remarks are given in Chapter 6.

3. DEVELOPMENT OF A STATISTICAL BASIS FOR INTEGRITY MONITORING

To get a basis for the integrity monitoring algorithms, it is important to know how signals which deteriorate system performance are manifested in the GPS measurements and the corresponding position estimates. The inherent error or noise which limits the accuracy of the system during normal operation is not of a deterministic nature and the effects are best described in statistical terms. Most integrity monitoring schemes rely on these statistics as a reference with which observed statistics are compared. If the data are not consistent with the model, then the scheme declares that the system is not operating normally. In this chapter, the basic measurements and solutions will be presented and this will make it possible to identify the statistics of normal measurement and solution errors.

3.1. The Measurement Equations

The most common techniques for obtaining a navigation solution using GPS are the pointwise and the Kalman filter solutions. The pointwise solution is a solution of a system of equations where no prior information such as past measurements or assumptions about the vehicle dynamics is used. It is also referred to as a snapshot solution because it "freezes" the geometry of the satellites and the vehicle at one point in time without any regard for the motion of the vehicle. In contrast, the Kalman filter approach has memory in that the solution it provides is a blending of past measurements with the current measurements based on certain assumptions about the vehicle dynamics. In either solution, each satellite which is tracked provides a measurement as shown in Eq. 3.1 [28], [8].

$$\rho = \sqrt{(x - x_S)^2 + (y - y_S)^2 + (z - z_S)^2} + T + \epsilon \qquad (3.1)$$

where

$$\begin{aligned}
\rho &= \text{pseudorange from user to satellite (m)} \\
T &= \text{range bias due to receiver clock offset (m)} \\
(x, y, z) &= \text{position of vehicle (m)} \\
(x_S, y_S, z_S) &= \text{position of satellite in vehicle coordinate frame (m)} \\
\epsilon &= \text{sum of all measurement errors (m)}
\end{aligned}$$

Because the pseudorange is a non-linear function of the vehicle position, the problem is linearized about a nominal trajectory. The parameters to be estimated become the deviations of the true position from the point of linearization. This point defines a locally-level three-dimensional coordinate frame with (x,y,z) being north, west, and up respectively. The clock offset from GPS time appears in a linear fashion and is unaffected by the linearization, so the total quantity or the deviation from the previous estimate could be estimated in the same manner. The left hand side of the linearized measurement given in Eq. 3.2 has the form of a measured less a predicted quantity, where the total quantities cancel and the amount remaining is proportional to the error in the vehicle position and clock estimates which remain after the linearization, and also to the measurement error ϵ.

$$\Delta\rho \;=\; \cos\theta_{\mathrm{x}\rho}\Delta\mathrm{x} + \cos\theta_{\mathrm{y}\rho}\Delta\mathrm{y} + \cos\theta_{\mathrm{z}\rho}\Delta\mathrm{z} + \Delta\mathrm{T} + \epsilon \qquad (3.2)$$

where

$\Delta\rho$ = linearized pseudorange (m)

$\Delta\mathrm{T}$ = deviation from prior clock offset (m)

$(\Delta\mathrm{x}, \Delta\mathrm{y}, \Delta\mathrm{z})$ = vehicle deviations from true position (m)

$\theta_{\mathrm{x}\rho}, \theta_{\mathrm{y}\rho}, \theta_{\mathrm{z}\rho}$ = angle between range vector and x, y, and z axes

ϵ = sum of all measurement errors (m)

The variables to be estimated are the deviations of the baseline position and clock estimates from the true position and clock offset. The total solution is the superposition of the components of the solution due to the error in the linearization and also to the measurement error. In the snapshot simulations, the error due to the linearization can be ignored and the estimation error due to the measurement noise is analyzed. However, in the Kalman filter simulation the estimation error due to both sources must be considered.

3.2. The Snapshot Solution

As mentioned earlier, the snapshot solution is simply the least-squares solution of the linearized system of equations as defined in Eq. 3.2. No restrictions will be put on the solution with regard to prior assumptions about the vehicle dynamics or past measurements. This is one of the main advantages of this technique since its performance is not at all model dependent. The solution will now be presented.

The set of n measurements are put into matrix form as shown below in Eq. 3.3.

$$\mathbf{y} = \mathbf{G}\,\mathbf{x} \tag{3.3}$$

where

$$\mathbf{x} = [\Delta \mathrm{x}\ \ \Delta \mathrm{y}\ \ \Delta \mathrm{z}\ \ \Delta \mathrm{T}]^T$$

$$\mathbf{y} = [\Delta \rho_1\ \ \Delta \rho_2\ \ \ldots\ \ \Delta \rho_\mathrm{n}]^T$$

$$\mathbf{G} = \begin{bmatrix} \cos\theta_{\mathrm{x}\rho_1} & \cos\theta_{\mathrm{y}\rho_1} & \cos\theta_{\mathrm{z}\rho_1} & 1 \\ \cos\theta_{\mathrm{x}\rho_2} & \cos\theta_{\mathrm{y}\rho_2} & \cos\theta_{\mathrm{z}\rho_2} & 1 \\ & \vdots & & \\ \cos\theta_{\mathrm{x}\rho_\mathrm{n}} & \cos\theta_{\mathrm{y}\rho_\mathrm{n}} & \cos\theta_{\mathrm{z}\rho_\mathrm{n}} & 1 \end{bmatrix}$$

(Boldface characters are used to denote vectors or matrices, superscript T denotes the transpose operation and E[·] denotes the expectation of the random variable inside the brackets.) The least-squares estimate of **x** will be identified with $\hat{\mathbf{x}}$ and is given in Eq. 3.4.

$$\hat{\mathbf{x}} = (\mathbf{G}^\mathrm{T}\mathbf{G})^{-1}\mathbf{G}^\mathrm{T}\,\mathbf{y} \tag{3.4}$$

where

$$\hat{\mathbf{x}} = \mathrm{E}[\mathbf{x}]$$

The **y** vector represents the pseudorange error vector and contains the effects of ionosphere and troposphere propagation delays, normal satellite ephemeris and clock errors, and receiver noise. Selective availability and satellite clock errors will also appear in this vector. When no unusual errors are present, the pseudorange error vector is distributed as a multivariate Gaussian random variable with zero mean and covariance $\sigma^2\mathbf{I}$. This will be denoted as in Eq. 3.5.

$$\mathbf{y} \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{3.5}$$

Linear operations on a set of normal random variable yields another set of normal random variables specified by their mean and variance. Since the measurement errors are zero mean, the induced errors in the solution vector will also be zero mean. The variance of $\hat{\mathbf{x}}$ is derived in Eq. 3.6.

$$\begin{aligned} \mathbf{P} &= \mathrm{Var}[\hat{\mathbf{x}}] \\ &= \mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathrm{T}}] \\ &= (\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}\mathbf{G}^{\mathrm{T}}\mathrm{E}[\mathbf{y}\mathbf{y}^{\mathrm{T}}]\mathbf{G}(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1} \\ &= \sigma^2(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1} \end{aligned} \tag{3.6}$$

The distribution of the least-squares estimate is given in Eq. 3.7.

$$\hat{\mathbf{x}} \sim \mathrm{N}(\mathbf{0}, \mathbf{P}) \tag{3.7}$$

It should be noted that the $\mathbf{y}$ vector is not an observable quantity. If this were true, the measurement errors could be removed and the induced position error would be zero. The measurement residuals however are an observable quantity and have known statistics. The measurement residual is the difference between the given measurement vector and the predicted measurement vector found by substituting the least-squares solution back into the measurement equation given in Eq. 3.3. The statistics of the residuals are given in Eq. 3.8.

$$\mathbf{e}_{\mathbf{y}} \sim \mathrm{N}(\mathbf{0}, \mathbf{V}) \tag{3.8}$$

where

$$\begin{aligned}
\mathbf{e_y} &= \mathbf{y} - \hat{\mathbf{y}} \\
&= [\mathrm{I} - \mathbf{G}(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}\mathbf{G}^{\mathrm{T}}]\mathbf{y} \\
\mathrm{E}[\mathbf{e_y}] &= \mathbf{0} \\
\mathbf{V} &= \mathrm{Var}[\mathbf{e_y}] \\
&= \mathrm{E}[\mathbf{e_y}\mathbf{e_y^T}] \\
&= \sigma^2[\mathbf{I} - \mathbf{G}(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}\mathbf{G}^{\mathrm{T}}]
\end{aligned}$$

Before we leave the discussion of the least-squares solution, it is appropriate to consider how the satellite geometry affects the solution. The matrix $(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}$ describes how much information about the parameters being estimated is contained in the measurements. In the navigation community, the square root of the trace of this matrix is known as the geometric dilution of precision (GDOP). Other dilution of precision parameters of interest (position, horizontal, and time) are related to the main diagonal of this matrix and are defined below.

$$\begin{aligned}
\mathrm{PDOP} &= \sqrt{\sum_{\mathrm{i}=1}^{3}(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}_{\mathrm{ii}}} \\
\mathrm{HDOP} &= \sqrt{\sum_{\mathrm{i}=1}^{2}(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}_{\mathrm{ii}}} \\
\mathrm{TDOP} &= \sqrt{(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}_{44}}
\end{aligned}$$

Physically, the GDOP is related to the volume of the tetrahedron (in the four satellite case) formed by the tips of the unit vectors to the four satellites used in the solution. If the satellites are nearly coplanar, then there is no information contained in the direction perpendicular to the plane. In this case it will not be possible to estimate parameters with components in this direction. This will be reflected in a large GDOP. Another viewpoint of the GDOP is that it describes how unit variance errors in the measurements affect the variance of the parameters being estimated. If the geometry is poor, the main diagonal elements of this matrix will be large and the variance of the parameters being estimated will also be large. In the extreme case, if the set of measurement equations are not linearly independent, the matrix $\mathbf{G}^T\mathbf{G}$ will be singular and the inverse will not exist. As this condition is approached, GDOP will approach infinity. The effects of the satellite geometry will play a limiting role in the effectiveness of the integrity algorithms, as will be discussed in later chapters.

3.3. The Kalman Filter Solution

This section describes how the navigation solution is obtained using a Kalman filter [5]. The filter design begins by specifying the process and measurement models. As in the least-squares approach, the problem is linearized about a baseline trajectory and the parameters to be estimated are the deviations from the point of linearization. The process model describes how the state vector representation of the random process to be estimated evolves with time. This equation has the form shown in Eq. 3.9 and is the discrete time solution of a linear system driven by white noise. (The letter n will be used to specify the number of satellites used in the solution except in Eq. 3.9 and Eq. 3.14 where it refers to the dimension of the state vector.)

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k \tag{3.9}$$

where

x_k = $(n \times 1)$ state vector describing the system state at t_k

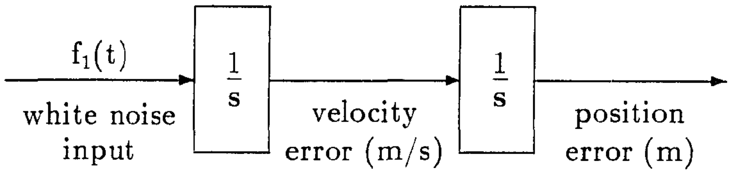
Φ_k = $(n \times n)$ transition matrix from t_k to t_{k+1}

w_k = $(n \times 1)$ driven response vector due to white noise from t_k to t_{k+1}

A commonly used model for the random position error in each direction is the double integrator plant driven by white acceleration noise as shown in Fig. 3.1. This is the natural model which relates acceleration to position and velocity but assumes the acceleration error is white. Such a model leads to a random walk process for the velocity error and an integrated random walk process for the position error. Receiver clock errors are also typically modelled with a double integrator plant which has white noise inputs driving both integrators as shown in Fig. 3.2. The states for this model are clock bias (s) and drift (s/s) but the flicker component of the clock error process is ignored in this model [29]. The constants h_0 and h_{-2} are the Allan variance parameters which relate the frequency noise to the stability of the receiver clock. In the filter model, the white noise amplitudes are scaled by c^2 (c = speed of light) to change the units of the clock bias to meters and the clock drift to m/s. Values for the Allan variance parameters for crystal oscillators with modest and good stability are given in Table 3.1.

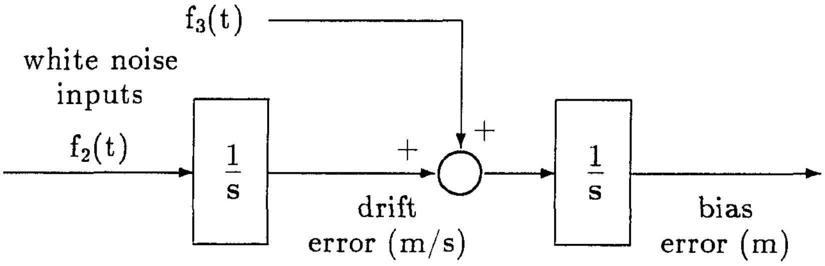
Table 3.1: Allan variance parameters for typical crystal oscillators

	Modest stability	Good stability
h_{-2} (Hz)	3.80×10^{-21}	1.37×10^{-24}
h_0 (s)	9.43×10^{-20}	1.80×10^{-21}



$$A_1 = \text{power spectral density amplitude of white noise } f_1(t) \ \ (m^2/s^3)$$

Figure 3.1: Model for random position and velocity errors in one direction



$$\begin{aligned} A_2 &= \text{power spectral density amplitude of white noise } f_2(t) \ \ (m^2/s^3) \\ &= (2\pi^2 h_{-2})c^2 \\ A_3 &= \text{power spectral density amplitude of white noise } f_3(t) \ \ (m^2/s) \\ &= (\frac{1}{2}h_0)c^2 \end{aligned}$$

Figure 3.2: Model for random clock bias and drift errors

These models lead to an eight state Kalman filter with state vector elements defined in Eq. 3.10.

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} \text{north position error (m)} \\ \text{north velocity error (m/s)} \\ \text{west position error (m)} \\ \text{west velocity error (m/s)} \\ \text{vertical position error (m)} \\ \text{vertical velocity error (m/s)} \\ \text{receiver clock bias error (m)} \\ \text{receiver clock drift error (m/s)} \end{bmatrix} \tag{3.10}$$

The double integrator plant has the (2 × 2) transition matrix given in Eq. 3.11 for a step size Δt. The position and clock states are decoupled so the transition matrix for the complete system has (2 × 2) blocks along the main diagonal as shown in Eq. 3.11.

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi} & 0 & 0 & 0 \\ 0 & \boldsymbol{\Phi} & 0 & 0 \\ 0 & 0 & \boldsymbol{\Phi} & 0 \\ 0 & 0 & 0 & \boldsymbol{\Phi} \end{bmatrix} \quad \text{where } \boldsymbol{\Phi} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.11}$$

The $\mathbf{w}_k$ vector describes the contribution to the state vector at t_{k+1} due to the white noise that enters the model from t_k to t_{k+1}. White noise is a zero mean random process with the properties that two samples at different times are uncorrelated and the expectation of two samples at the same time is an impulse function with area equal to the power spectral density amplitude. A Gaussian distribution is assumed so $\mathbf{w}_k$ is completely specified by its mean and variance as defined in Eq. 3.12.

$$
\begin{aligned}
E[\mathbf{w}_k] &= \mathbf{0} \\
E[\mathbf{w}_j \mathbf{w}_k^T] &= \begin{cases} \mathbf{Q}_k & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}
\end{aligned}
\tag{3.12}
$$

Decoupled position and clock models lead to the diagonal $\mathbf{Q}_k$ matrix for the complete system as shown in Eq. 3.13.

$$
\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_1 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_1 & 0 & 0 \\ 0 & 0 & \mathbf{Q}_1 & 0 \\ 0 & 0 & 0 & \mathbf{Q}_2 \end{bmatrix} \tag{3.13}
$$

where

$$
\begin{aligned}
\mathbf{Q}_1 &= \begin{bmatrix} A_1 \Delta t^3/3 & A_1 \Delta t^2/2 \\ A_1 \Delta t^2/2 & A_1 \Delta t \end{bmatrix} \\
\mathbf{Q}_2 &= \begin{bmatrix} A_2 \Delta t^3/3 + A_3 \Delta t & A_2 \Delta t^2/2 \\ A_2 \Delta t^2/2 & A_2 \Delta t \end{bmatrix}
\end{aligned}
$$

The Kalman filter measurement model given in Eq. 3.14 is used to describe the linear relationship between the current measurement and the current state of the random process.

$$
\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{3.14}
$$

where

$$
\begin{aligned}
\mathbf{z}_k &= (m \times 1) \text{ measurement vector at } t_k \\
\mathbf{H}_k &= (m \times n) \text{ linear connection matrix at } t_k \\
\mathbf{v}_k &= (m \times 1) \text{ measurement noise vector}
\end{aligned}
$$

The $\mathbf{v}_k$ vector is distributed as a multivariate Gaussian random variable with mean and variance as defined in Eq. 3.15 and is assumed to be uncorrelated with the process noise at all steps.

$$\begin{aligned} E[\mathbf{v}_k] &= \mathbf{0} \\ E[\mathbf{v}_j\mathbf{v}_k^T] &= \begin{cases} \mathbf{R}_k & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \\ E[\mathbf{v}_j\mathbf{w}_k^T] &= \mathbf{0}\ \forall\, j,k \end{aligned} \tag{3.15}$$

In the GPS case, the only non-zero elements of the $\mathbf{R}_k$ matrix are the elements along the main diagonal and each element represents the noise variance associated the corresponding measurement. The linearized pseudorange measurement equation in the Kalman filter has the same form as given in Eq. 3.2 where the linearized pseudorange is a measured less a predicted quantity. The measurement vector for n satellites and the linear connection matrix to the state are given in Eq. 3.16 and Eq. 3.17 (without the time subscripts on the direction cosines or the pseudoranges).

$$\mathbf{z}_k = [\Delta\rho_1\ \Delta\rho_2\ \ldots\ \Delta\rho_n]^T \tag{3.16}$$

$$\mathbf{H}_k = \begin{bmatrix} \cos\theta_{x\rho_1} & 0 & \cos\theta_{y\rho_1} & 0 & \cos\theta_{z\rho_1} & 0 & 1 & 0 \\ \cos\theta_{x\rho_2} & 0 & \cos\theta_{y\rho_2} & 0 & \cos\theta_{z\rho_2} & 0 & 1 & 0 \\ & & & \vdots & & & & \\ \cos\theta_{x\rho_n} & 0 & \cos\theta_{y\rho_n} & 0 & \cos\theta_{z\rho_n} & 0 & 1 & 0 \end{bmatrix} \tag{3.17}$$

After the process and measurement equations and the accompanying statistics are defined, it is a straightforward exercise to obtain the equations which implement the recursive estimation procedure. If the performance measure is minimum mean square estimation error, the optimal estimator of a set of parameters given the data is simply the mean of the posterior distribution of the parameters conditioned on the data. If the statistics are assumed to be Gaussian, then only the mean and variance of the conditional distribution are required to completely specify this distribution. The Kalman filter provides the mechanism whereby the mean and variance of the posterior conditional distribution are updated with each new set of data. The form of the update equations for the mean and variance are analogous to those obtained from the conditional Gaussian distribution using classical probability theory. The formula for the conditional mean is the sum of the mean of the parameters plus a term proportional to the mean of the data and leads to the update equation in Eq. 3.18.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \qquad (3.18)$$

where

$\hat{\mathbf{x}}_k$ = best estimate of $\mathbf{x}_k$ after processing current measurement

$\hat{\mathbf{x}}_k^-$ = best estimate $\mathbf{x}_k$ prior to processing current measurement

$\mathbf{z}_k$ = current measurement

$\mathbf{H}_k\mathbf{x}_k^-$ = best estimate of $\mathbf{z}_k$ prior to processing current measurement

$\mathbf{K}_k$ = gain matrix designed to provide optimal blending of prior information with current measurement

The gain matrix $\mathbf{K}_k$ is the free variable which is used to minimize the mean square estimation error. The estimation error is the difference between the truth and the estimate as shown in Eq. 3.19. The error covariance matrix defined in Eq. 3.20 is just the expectation of the squared estimation error since the filter is an unbiased estimator when the filter model matches the truth. The general covariance update equation, which is valid for any gain matrix, is found by substituting Eq. 3.18 and Eq. 3.19 into Eq. 3.20. The optimal gain, also known as the Kalman gain, is the gain matrix which minimizes the trace of this equation. The optimal gain and the covariance update using this gain are given in Eq. 3.22 and Eq. 3.21 [5].

$$\mathrm{e_x} = \mathrm{x_k} - \hat{\mathrm{x}}_\mathrm{k} \tag{3.19}$$

$$\mathbf{P}_\mathrm{k} = \mathrm{E}[\mathrm{e_x e_x^T}] \tag{3.20}$$

$$\mathbf{P}_\mathrm{k} = (\mathbf{I} - \mathbf{K}_\mathrm{k}\mathbf{H}_\mathrm{k})\mathbf{P}_\mathrm{k}^- \tag{3.21}$$

$$\mathbf{K}_\mathrm{k} = \mathbf{P}_\mathrm{k}^-\mathbf{H}_\mathrm{k}^\mathrm{T}(\mathbf{H}_\mathrm{k}\mathbf{P}_\mathrm{k}^-\mathbf{H}_\mathrm{k}^\mathrm{T} + \mathbf{R}_\mathrm{k})^{-1} \tag{3.22}$$

where

$\mathbf{P}_\mathrm{k}^-$ = error covariance matrix at $\mathrm{t_k}$
prior to processing the current measurement

The variables $\hat{\mathrm{x}}_\mathrm{k}^-$ and $\mathbf{P}_\mathrm{k}^-$ are needed at the start of each recursive step. (The superscript minus is used to identify a prior variable.) The very first set represents the initial conditions of the estimation error trajectory which unfolds as the measurements are processed, and these are generally supplied by the filter designer. After each update, the prior information needed at the next step is obtained with the projection equations given in Eq. 3.23. The best estimate of the state as the next step is the current state projected through the transition matrix. This result is obtained by taking the expectation of the process equation given in Eq. 3.9 and by noting that the expectation of a sample of a white noise sequence is zero. The prior error covariance at the next step is found by taking the expectation of the squared error between the truth and the prior estimate.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^{-} &= \mathbf{\Phi}_k \hat{\mathbf{x}}_k \\ \mathbf{P}_{k+1}^{-} &= \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k \end{aligned} \tag{3.23}$$

Given the prior information, the complete recursive process consists of the gain calculation, the two updates, and the two projections as given in Eq. 3.18 through Eq. 3.23.

After the system matrices are specified, tuning of the filter is possible by altering the assumed statistics of the process noise and the measurement noise by varying the elements within the **Q** and the **R** matrices respectively. The power spectral density (PSD) amplitude of the white noise inputs in the process model can be thought of as specifying the amount of uncertainty there is about the random process model from step to step. If the PSD amplitude is chosen relatively large, the filter will give less weight to older measurements than it does to newer ones when it forms the new estimate. Conversely, the filter never "forgets" old measurements if the PSD amplitude is taken to be zero. By varying the PSD amplitude between these extremes, the filter designer can improve the filter robustness by affecting the time constants of the filter. For example, decreasing the filter time constants is helpful in preventing filter divergence if the true process takes a sudden jump and conversely, increasing the time constants helps to smooth out the effects of certain noise processes. Filter performance is also affected by the choice of the noise variance associated with measurement. If the value for one source is large relative to another source, the former measurement will receive less weight than the latter. The measurement noise variances may be increased artificially to casually account for known colored measurement noise which is not modelled in an effort to reduce the number of state vector elements. These alterations lead to a suboptimal filter with respect to the assumed correct model, but this may be tolerated in an effort to improve the robustness of the filter. After all, a suboptimal filter that does not diverge is preferable to an optimal filter that may diverge if the statistics of the true random process or the measurement noise change dramatically from the filter model.

Under the assumption that the filter model fits the true process, it is possible to define the distribution of the measurement residuals and the state vector estimation error. These statistics define the estimator performance during the normal situation when no large unmodelled error is present and will provide the basis for the integrity checking algorithms. The measurement residual ν_k is the difference between the actual measurement and the prior estimate of the measurement and is defined in Eq. 3.24.

$$\nu_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \tag{3.24}$$

When the system is properly modelled, the measurement residuals form a multivariate innovations sequence which has the same properties as the $\mathbf{w}_k$ sequence defined above. The mean and variance are given in Eq. 3.25.

$$\begin{aligned} \mathrm{E}[\nu_k] &= \mathbf{0} \\ \mathrm{E}[\nu_j \nu_k^T] &= \begin{cases} \mathbf{V}_k & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \end{aligned} \tag{3.25}$$

where

$$\mathbf{V}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$$

The distributions of the estimation error and the measurement residuals are given in Eq. 3.26.

$$\begin{aligned} \mathrm{e_X} &= \mathrm{N}(\mathbf{0},\mathbf{P}_\mathrm{k}) \\ \nu_\mathrm{k} &= \mathrm{N}(\mathbf{0},\mathbf{V}_\mathrm{k}) \end{aligned} \tag{3.26}$$

A natural way to investigate the adequacy of the filter model is to test how well the observed statistics match the distributions given in Eq. 3.26. If one is performing a simulation, the truth is known and the estimation error can be formed. It is then possible to analyze the induced error due to an unmodelled error. However, when the filter is implemented in a real situation, the estimation error is not an observable quantity and it is not possible to perform a statistical analysis of this parameter. In contrast, the measurement residuals are always observable and thus it is possible to perform a number of tests on this parameter such as zero-mean tests, tests of covariance and tests of whiteness [24]. In this situation, the parameter of interest is the estimation error but we usually have to settle for tests in the measurement domain and then make an inference about the statistics of the estimation error. This is the motivation behind the development of the censored Kalman filter which will allow the final integrity test to take place in the estimation error domain. This is done by forming a test statistic which is related to the distance between two horizontal position estimates which are based on different assumptions about the statistics of the measurement residuals.

3.4. Aspects of the Monte Carlo Simulation

In this section, a description of some of the mechanics involved with the Monte Carlo simulation of both the point solution and the Kalman filter solution will be presented. The purpose of such an experiment in the context of integrity monitoring is to analyze the effects of error sources such as selective availability and satellite failures on the navigation solution. In this type of analysis only the random measurement error or random vehicle motion need be considered, and total quantities such as the total pseudorange measurements or the total position estimates are not needed.

3.4.1. Choosing the set of satellites

It is required that the satellite constellation be simulated in order to calculate the direction cosines which appear in the linearized measurement equations as given in Eq. 3.2. The equations used to obtain the direction cosines are given in Appendix A and also in [20]. A few simplifying assumptions are made in these equations such as circular satellite orbits and a spherical earth. This process is different from what occurs inside the GPS receiver where the ephemeris data from the navigation message is used to calculate the satellite position but for error-analysis purposes these simplifications are acceptable. In the simulation, the direction cosines to each satellite in the receiver locally-level coordinate frame are calculated by assuming a nominal position for the vehicle and by using the simulated position of each satellite.

If the z-direction cosine from a particular satellite is larger than the sine of a specified mask angle, then this satellite is added to the set of visible satellites for the given location and time. The mask angle refers to the minimum elevation at which the satellite is high enough above the horizon for the signal to be tracked effectively. It is common to use a conservative value of 7.5 degrees for the mask angle, which forces the availability of satellites in computer simulations to be closer to real situations when low lying satellites may be occasionally obscured by the wings of the aircraft. Some receivers are designed to track satellites down to the horizon, but for simulation studies a 5 to 7.5 degree mask angle is appropriate to be conservative.

The choice of the satellite constellation has a large effect on satellite availability. The current trend is toward a 24 satellite configuration which will be implemented in six orbital planes. A three plane configuration is also being investigated but from the standpoint of implementation it is less attractive since it will require re-phasing of the satellites already in orbit. There has also been an interest in non-symmetric spacings within rings which optimizes geometries based on subsets of the visible satellites.

In a sense, there is a "curse of plenty" in the number of combinations of visible satellites which can be formed to obtain a solution. For example if there are seven satellites in view, one could chose the best set of four among the 35 combinations of four satellites, or the best set of five among the 21 sets of five satellites, or the best set of six among the seven combinations of six satellites. With a 24 satellite constellation, there are times when as many as nine satellites will be in view, and the number of combinations of 4 to 9 satellites which can be used to obtain a solution is 382. The computational effort required to perform such a search is non-trivial since the GDOP calculation for each set of satellites requires a matrix multiplication and a matrix inverse of dimension four.

It is attractive from an operational standpoint to avoid this search and to simply process all the available measurements. Such a solution is known as the all-in-view solution. This approach is appealing from an integrity viewpoint since the redundant measurements allow for a consistency check which is directly related to the navigation solution and because the effects of measurement errors are diminished as more satellites are processed. The final choice of the number of satellites to be used is also a function of the processing capabilites of the GPS receiver. There is a trend toward multi-channel designs which allow for continuous tracking of as many as eight satellites in parallel. However, single channel receivers which sequence among satellites have proven to offer reasonable performance and may offer similar performance in an all-in-view implementation.

3.4.2. The point solution simulation

The simulation of an estimator based on point solution methods is straightforward to perform because the solution depends on only the current measurements. In this case, one simply generates the measurement error vector using the appropriate statistics as given in Eq. 3.5. The SA noise is assumed to be statistically independent of the inherent GPS measurement errors so the total error vector is simply a set of standard normal random variables scaled by the appropriate standard deviation. A satellite failure is simulated by adding a bias to one of the elements of this vector. An external subroutine from the IMSL subroutine library [10] is used in the simulation program and it returns a vector of N(0,1) variates which are shaped into the measurement errors with the specified distribution. Each call is used to define the time history of the measurement error from one satellite and in this way the errors among different satellites should remain uncorrelated.

After the measurement vector is found, a set of least-squares solutions are calculated. These include the all-in-view solution and the set of least-squares solutions which leave out one satellite at a time. One may also choose to calculate the posterior measurement residuals associated with each of these solutions. The $(\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}$ matrix from each solution is also saved and this provides the integrity monitoring algorithm with the covariance matrix associated with each solution and allows for the computation of the measurement residual vector covariance matrix as described in section 3.2. Each solution which is calculated represents the error which would be in the navigation solution as a result of the given measurement errors. Under real conditions, these are not observable quantities. However, the difference between all such solutions is observable and represents information which can be used for integrity monitoring purposes. The measurement residuals are also always observable and are related to the test statistic in some RAIM techniques which will be discussed.

Integrity monitoring refers to making sure that the accuracy of the navigation solution is within specification so one of the solutions has to be chosen to represent the navigation solution. This is usually taken to be the all-in-view solution since this combination of satellites will offer the smallest GDOP among all possible solutions. The results of a snapshot experiment that may be used in the integrity algorithms are the differences among all calculated solutions, the measurement residuals, the navigation solution (which in this case is the actual navigation error even though it is unobservable in a real situation), and the accompanying covariance matrices for each solution. All the point solution integrity algorithms described in this work will be based on this set of statistics.

3.4.3. The Kalman filter simulation

Even though the Kalman filter simulations yield a similar set of parameters to work with for integrity monitoring, the simulations are not as simple to perform both computationally and conceptually. The difficulty arises in the need to generate random processes (rather than random variables as in the point solution simulations) which describe the true random vehicle and clock trajectories which are to be estimated. To perform an analysis of the effect of a slowly varying satellite clock error on the Kalman filter estimation error, a lengthy simulation is required. In the point solution, this is done by simply adding a relatively large bias error to one of the satellite measurements (because a small bias error has little effect on the all-in-view solution error). However, in the Kalman filter the present estimation error is a function of present and past measurement errors and the solution is gradually pulled away from the truth as a gradual ramp error continues to enter the system. Whereas a large bias may be required to force a specified horizontal error in the snapshot solution, the value at which a ramp failure induces the same amount of horizontal error may be considerable smaller due to the filter memory. The basic simulation procedure is recursive, so the procedure can be described with the details of a typical step as will now be presented.

A trajectory of the true process and the estimate of this process begins with the true initial state $\mathbf{x}_0$, the estimate of this state $\hat{\mathbf{x}}_0$, and the corresponding error covariance matrix $\mathbf{P}_0^-$. If all instruments were perfectly calibrated initially, each of these parameters would be zero. In general, there is some uncertainty in the initial position, velocity, and clock deviations from GPS time. These are generally assumed to be uncorrelated so the diagonal elements of $\mathbf{P}_0^-$ represent the square of the uncertainty in each of the state vector elements and all the off-diagonal elements are zero. The set of standard deviations is used to scale a set of independent standard normal random variables to form the true state vector at t_0. The corresponding state estimate is zero.

After $\mathbf{x}_0$ is obtained, the measurement vector is found using the measurement model given in Eq. 3.14. Satellite selection is performed as in the point solution and yields a set of direction cosines which are used to fill the connection matrix $\mathbf{H}_0$. The true part of the measurement is formed by taking the product of the true state and the measurement connection matrix. The observable measurement is formed by corrupting the true component with additive white measurement noise and other unmodelled errors such as the SA and the satellite failure. In general, the statistics used to form the additive white noise vector $\mathbf{v}_k$ can be different from the statistics assumed in the filter. In most of the simulation work to be presented, the SA noise is only casually accounted for by increasing the filter measurement noise covariance matrix $\mathbf{R}_k$ above that which is normal. This does account for presence of SA but leaves the time-wise correlation structure unmodelled.

At this point, the Kalman filter updates the estimate of the state vector and the error covariance matrix. The measurement residuals and their covariance matrix are calculated as part of the state update so this does not require any extra computation. The filter then projects the state estimate and error covariance to serve as the prior information at the next step. The true process vector must also be projected to the next step and this requires a sample of the $\mathbf{w}_k$ sequence to be generate with statistics as defined by $\mathbf{Q}_k$ from Eq. 3.13. For the $\mathbf{Q}_k$ matrix at hand, the only correlation between state vector elements is between the position and velocity states in the same direction and between the clock bias and drift states. As shown below, it is possible to add two scaled independent standard normal sequences y_1 and y_2 to generate the correlation structure as required by the (2×2) blocks of the $\mathbf{Q}_k$ matrix.

let

$$\begin{aligned} w_1 &= a\, y_1 \\ w_2 &= b\, y_1 + c\, y_2 \end{aligned}$$

where

$$\begin{aligned} y_1, y_2 &\sim \mathrm{N}(0,1) \\ \mathrm{E}[y_1 y_2] &= \mathbf{0} \end{aligned}$$

then

$$\begin{aligned} a &= \sqrt{Q_{11}} \\ b &= Q_{12}/a \\ c &= \sqrt{Q_{22} - b^2} \end{aligned}$$

This process yields

$$\begin{aligned} \mathrm{E}[w_1 w_2] &= \mathbf{0} \\ \mathrm{Var}[w_1, w_2] &= \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix} \end{aligned}$$

In general, the true process may use a different $\mathbf{Q}_k$ matrix from the one assumed in the filter for generating the scale factors used to shape the standard normal variables into the desired processes. The ability to add the effects of deterministic acceleration error to the true state, as would occur during a mild turning maneuver in an aircraft, is built into the simulation programs which have been developed. However the filter does not account for this type of acceleration error. One way to handle this situation is to simply increase the white noise power spectral density amplitudes in the filter $\mathbf{Q}_k$ to a degree which accounts for the amount of uncertainty in the acceleration model between steps. The filter will give less weight to the previous measurements and will thus have better tracking characteristics during periods of large unmodelled acceleration.

The equations which are used to simulate the deterministic aircraft turns are presented below. The turn is parameterized with the period of the turn and the bank angle of the aircraft. These parameters then specify the angular and tangential velocity of the aircraft and the radius of the turn as shown below. The turn takes place about the point of the linearization with initial conditions as specified. The state variables x_i listed are the same as defined in Eq. 3.10.

$$\begin{aligned}
a_n &= a_c \cos \omega t \\
a_w &= a_c \sin \omega t \\
a_c &= g \tan \theta \\
\omega &= \frac{2\pi}{T} \\
r &= \frac{g}{\omega^2} \tan \theta \\
v &= \omega r
\end{aligned}$$

where

θ = bank angle with respect to horizontal plane

T = period of turn (s)

ω = angular velocity (rad/s)

r = radius of turn (m)

v = tangential velocity (m/s)

a_c = centripetal acceleration (m/s^2)

g = acceleration due to gravity (m/s^2)

a_n = acceleration in north direction

a_w = acceleration in west direction

Using these definitions, the contribution of a_n and a_w at t_{k+1} into the true horizontal position and velocity states for a step size Δt are given below.

$$\begin{aligned}
x_1 &= a_c[\frac{1}{\omega^2}(\sin\omega\Delta t - \omega\Delta t)\sin\omega t_k + (1 - \cos\omega\Delta t)\cos\omega t_k] \\
x_2 &= \frac{a_c}{\omega}[(\cos\omega\Delta t - 1)\sin\omega t_k + \sin\omega\Delta t\cos\omega t_k] \\
x_3 &= a_c[\frac{1}{\omega^2}(\omega\Delta t - \sin\omega\Delta t)\cos\omega t_k + (1 - \cos\omega\Delta t)\sin\omega t_k] \\
x_4 &= \frac{a_c}{\omega}[(1 - \cos\omega\Delta t)\cos\omega t_k + \sin\omega\Delta t\sin\omega t_k]
\end{aligned}$$

The initial conditions for the aircraft position and velocity at the start of the turn are

$$\begin{aligned}
x_1(0) &= -r \\
x_2(0) &= 0 \\
x_3(0) &= 0
\end{aligned}$$

$$x_4(0) = -v$$

The true state at the next step is formed according to Eq. 3.9. The current true state is projected to the next step through the natural dynamics of the system via multiplication by the true transition matrix and by adding the process noise vector and acceleration effects to the result. One complete recursive cycle has been described and the same procedure may be repeated as many times as is appropriate.

The data which are available in the analysis of the integrity monitoring algorithms are the state vector estimates, the measurement residuals, the true state vector (which is needed to form the navigation error) and the covariance matrices for these parameters under the H_0 and H_0' hypotheses.

3.4.4. Selective availability modelling and simulation

Since selective availability (SA) will be the dominant error source in limiting the accuracy for the civil GPS user, it must be a part of any conservative simulation where the performance of an integrity algorithm is analyzed. The true SA process is pseudorandom in the sense that the DoD specifies the structure and thus SA can be removed if the proper information is available. To unprivileged users, SA will probably appear as a pseudorange bias error whose correlation structure is for the most part unknown. The DoD on occasion has made some information about SA available to selected groups who where sponsored to analyze its effects. The results of such a study are used here and will be discussed in this section [19]. In the cited report, a statistical model was developed which had a correlation structure similar to that of the SA data analyzed. This model is used here in the Kalman filter simulations to represent the true SA process. Even though it is not known how well this will fit the true SA (if and when it is implemented), it serves as a reasonable model for analysis purposes. Results of limited simulation which analyzed the effect of this model on the user's horizontal solution will now be presented.

The SA model developed in [19] is described with a second order differential equation driven by white noise and is given in Eq. 3.27. The result is zero-mean stationary random process with damped sinusoidal modes and with a time constant related to the decaying exponential term in the homogeneous response. The solution of Eq. 3.27 is put into the form of Eq. 3.9 where the state variables are the phase variables as shown below in Eq. 3.28.

$$\ddot{x} + 2\beta\omega_0\dot{x} + \omega_0^2 x = w(t) \tag{3.27}$$

The state model for Eq. 3.27 is let

$$\begin{aligned} x_1 &= \text{selective availability process x (m)} \\ x_2 &= \text{derivative of } x_1 \text{ (m/s)} \end{aligned} \tag{3.28}$$

then

$$\Phi_k = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ -\omega_0^2\Phi_{12} & \Phi_{22} \end{bmatrix}$$

and

$$Q_k = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix}$$

where

$$\Phi_{11} = e^{-\beta\omega_0\Delta t}(\cos\omega_1\Delta t + \beta\frac{\omega_0}{\omega_1}\sin\omega_1\Delta t)$$

$$
\begin{aligned}
\Phi_{12} &= \frac{1}{\omega_1} e^{-\beta\omega_0\Delta t} \sin\omega_1\Delta t \\
\Phi_{22} &= e^{-\beta\omega_0\Delta t}(\cos\omega_1\Delta t - \beta\frac{\omega_0}{\omega_1}\sin\omega_1\Delta t) \\
Q_{11} &= \frac{q_c}{4\beta\omega_0^3}\left[1 - \frac{\omega_0^2}{\omega_1^2} e^{-2\beta\omega_0\Delta t}(1 - \beta^2\cos 2\omega_1\Delta t + \beta\frac{\omega_1}{\omega_0}\sin 2\omega_1\Delta t)\right] \\
Q_{12} &= \frac{q_c}{4\omega_1^2} e^{-2\beta\omega_0\Delta t}(1 - \cos 2\omega_1\Delta t) \\
Q_{22} &= \frac{q_c}{4\beta\omega_0}\left[1 - \frac{\omega_0^2}{\omega_1^2} e^{-2\beta\omega_0\Delta t}(1 - \beta^2\cos 2\omega_1\Delta t - \beta\frac{\omega_1}{\omega_0}\sin 2\omega_1\Delta t)\right] \\
\omega_1 &= \omega_0\sqrt{1 - \beta^2}
\end{aligned}
$$

The following statistics are used to form the initial conditions for such a process.

$$
\begin{aligned}
\sigma_{x_1}^2 &= \frac{q_c}{4\beta\omega_0^3} \\
\sigma_{x_1x_2} &= 0 \\
\sigma_{x_2}^2 &= \frac{q_c}{4\beta\omega_0} \\
E[x_1] &= 0 \\
E[x_2] &= 0 \\
x_1(0) &= \sigma_{x_1}y_1 \\
x_2(0) &= \sigma_{x_2}y_1
\end{aligned}
$$

where

$$
\begin{aligned}
y_1 &\sim N(0,1) \\
y_2 &\sim N(0,1) \\
E[y_1y_2] &= 0
\end{aligned}
$$

Only the three parameters q_c, β, and ω_0 are needed to specify this random process. The values arrived at in [19] are given below.

$$\begin{aligned} \mathrm{q_c} &= 2.68182 \times 10^{-4} \mathrm{m}^2/\mathrm{s}^3 \\ \beta &= 0.4 \\ \omega_0 &= 1.17412 \times 10^{-2} \mathrm{rad/s} \end{aligned}$$

then

$$\begin{aligned} \sigma_{\mathrm{x}_1} &= 10.18\mathrm{m} \\ \sigma_{\mathrm{x}_2} &= 0.119\mathrm{m/s} \\ \tau &= 213\mathrm{s} \end{aligned}$$

where

$$\begin{aligned} \tau &= \text{exponential time constant} \\ &= \frac{1}{\beta\omega_0} \end{aligned}$$

To help the reader visualize this type of random process, a sample of such a process is given in Fig. 3.3.

The stated policy of the DoD is that an accuracy of 100m 2-drms in the horizontal plane will be available to civil users. The 2-drms refers to twice the root means square value of the horizontal error which translates into twice the HDOP times the pseudorange measurement error standard deviation. This two sigma value corresponds to a 95% confidence interval on the horizontal error so the induced error may be larger than 100m possibly 5% of the time. In order to induce 100m 2-drms, the standard deviation of the SA process must be about 33.3m (using the fact that 100m 2-drms $= 2 \cdot \mathrm{HDOP} \cdot \sigma_{\mathrm{x}_1}$). It appears that the σ_{x_1} given above in the SA model is to small to induce this amount of error.

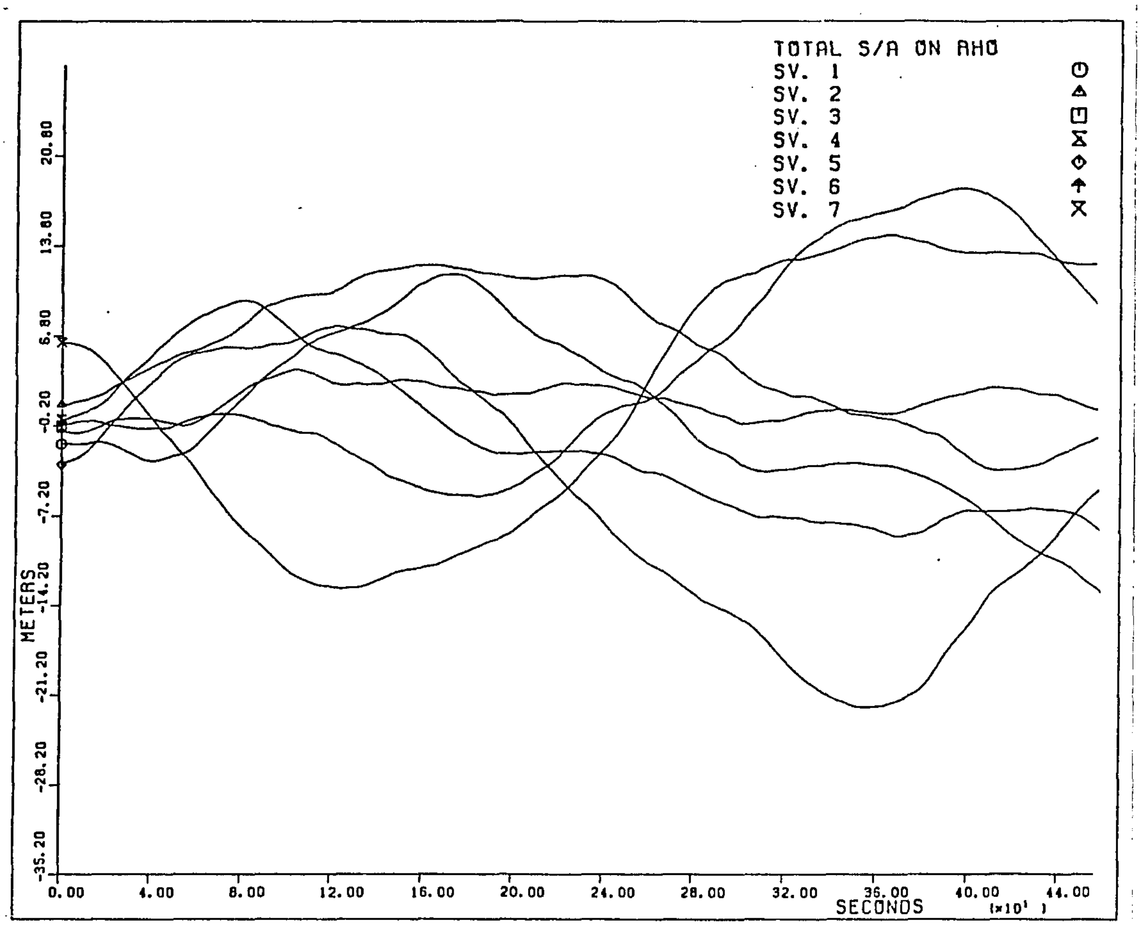


Figure 3.3: Sample of SA processes using damped cosine model

A series of Monte Carlo simulations were performed to investigate the effects of such a selective availability process on the horizontal estimation error. An optimal approach was first taken where the usual eight state filter model was augmented with extra state variables which represent the SA process model states for each measurement. The justification for this approach is that it serves as a benchmark for comparison with suboptimal models which do not account for the SA noise correlation structure.

Since the standard deviation mentioned above was so small, another process was added to this process to bring the resulting standard deviation up to 33.3m. The additional process used was Gauss-Markov and this process is specified with its time constant τ_M and mean square value σ_M^2. A Markov process is characterized as a process with an exponential autocorrelation function and the Gaussian assumption allows the probability distribution to be specified with the mean and variance. In this study, the Markov process was used to simulate a slowly varying bias (also referred to as a quasi-bias) with a time constant of one hour. Two cases for the relative sizes of these two processes were investigated. In case 1, the damped cosine process described above was generated and the Markov process was adjusted so the sum of these processes had the desired standard deviation. In case 2, the white noise amplitude associated with the damped cosine process was increased to yield $\sigma_{x_1} = 33$m and as a result the quasi-bias process required was fairly small. The parameters for both of these cases are specified below.

case (1)

$$\begin{aligned} q_c &= 2.68182 \times 10^{-4} \mathrm{m}^2/\mathrm{s}^3 \\ \beta &= 0.4 \end{aligned}$$

$$\begin{aligned}\omega_0 &= 1.17412 \times 10^{-2}\mathrm{rad/s}\\ \sigma_{x_1} &= 10.18\mathrm{m}\\ \sigma_M &= 31.74\mathrm{m}\\ \tau_M &= 3600\mathrm{s}\end{aligned}$$

case (2)

$$\begin{aligned}q_c &= 2.8202 \times 10^{-3}\mathrm{m}^2/\mathrm{s}^3\\ \beta &= 0.4\\ \omega_0 &= 1.17412 \times 10^{-2}\mathrm{rad/s}\\ \sigma_{x_1} &= 33.0\mathrm{m}\\ \sigma_M &= 4.48\mathrm{m}\\ \tau_M &= 3600\mathrm{s}\end{aligned}$$

Other filter parameters used were

$$\begin{aligned}\Delta t &= 1.0\mathrm{s}\\ A_1 &= 100\mathrm{m}^2/\mathrm{s}^3\\ \mathbf{R}_{ii} &= 400\ \mathrm{m}^2\ i=1,n\end{aligned}$$

where

h_{-2}, h_0 = The Allen variance parameters corresponding to a crystal oscillator with modest stability as given in Table 3.1.

The simulations where performed using a four satellite solution. In this case, the solution is more sensitive to mesurement errors than when redundant measurements are used and thus represents a situation where the effects of the SA are most harmful. The result of modelling the colored measurement noise processes is a state vector with 20 elements ($20 = 8 + 4 \times 3$). Four-hour simulations were performed (4×3600 steps) for each case of the SA structure described above. As would be expected, the experiments where the SA was mostly a slowly varying bias (case 1) did not experience large horizontal error excursions as often as when the SA was changing more rapidly (case 2). This is mainly of result of the filter estimating out the slowly varying components of the SA noise whereas the faster components are estimated to be zero. Thus rapidly changing SA errors (on the order of minutes) did the most damage in the estimation error. Fig. 3.4 and Fig. 3.5 show examples of large SA induced horizontal error excursions for case 1 and case 2 type SA structures.

Because the SA noise was somewhat oscillatory in the case 2 experiments, the horizontal estimation error also contained oscillatory trends. In some simulations, there were times when the average horizontal error stayed above 100m for as long as a four minutes. The four-hour experiments yielded 10400s when the HDOP was less than 2.0. Of these experiments, there was a total of about 1250s of data when the average horizontal error exceeded 100m and this corresponds to 12.5% of the data. The horizontal error standard deviation as supplied by the error covariance matrix was between 50m and 70m during these experiments which is reasonable for an HDOP between 1.5 and 2.0 and for a measurement error standard deviation of 33m.

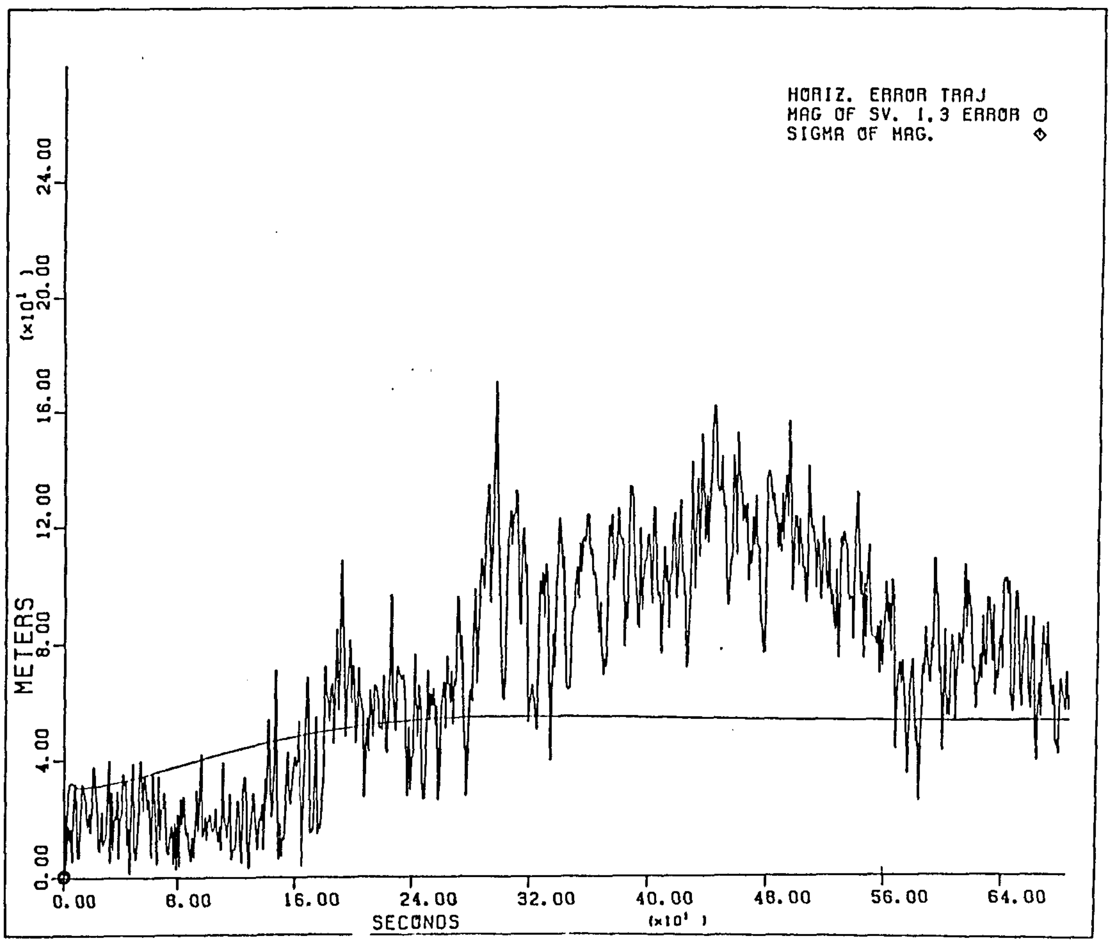


Figure 3.4: Sample horizontal error due to SA noise from a case 1 experiment

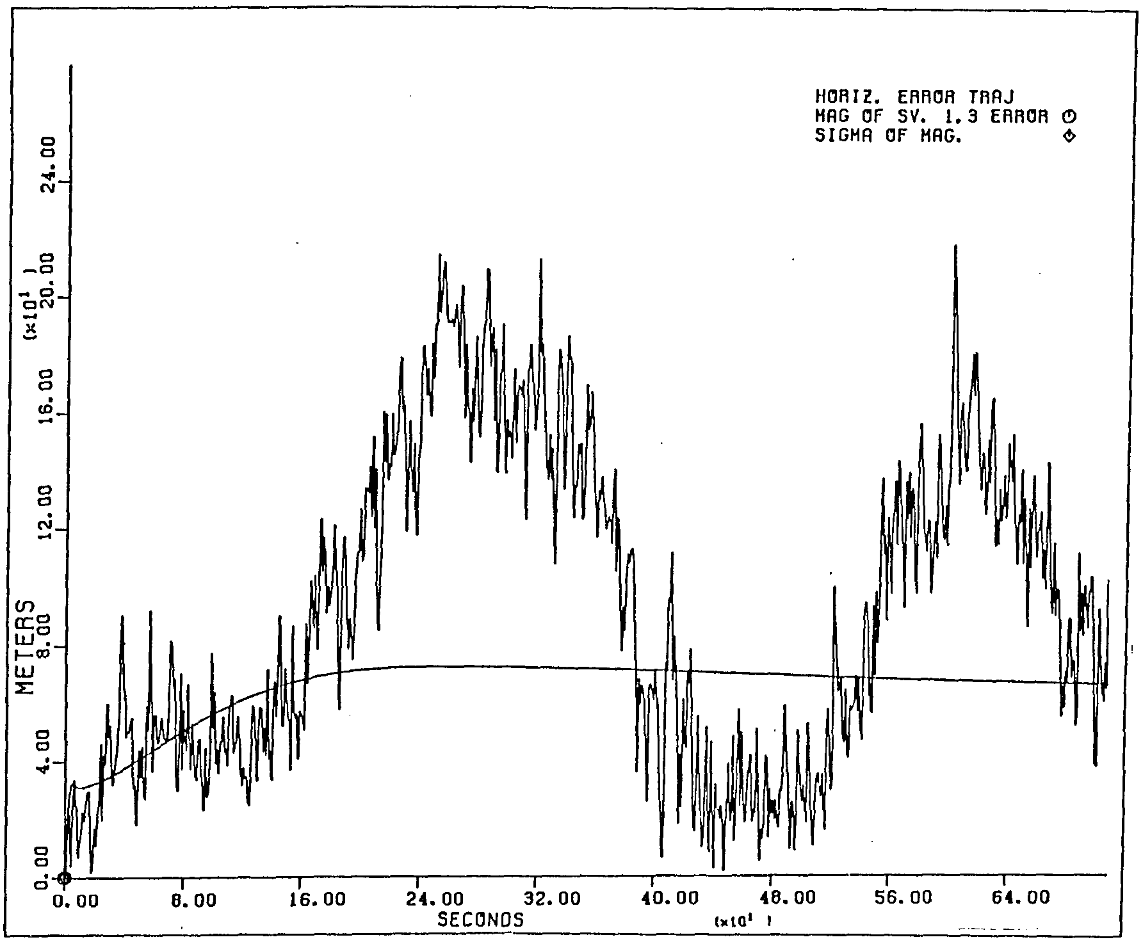


Figure 3.5: Sample horizontal error due to SA noise from a case 2 experiment

In the case 1 experiments where the SA noise was mainly a quasi-bias, the horizontal error was more noisy and had less periodic structure than the previously mentioned experiments. Even so, there were still a few sizable excursions when the average horizontal error was above 100m. Out of the same 10400s of reasonable geometry, there were 540s when the average error was larger that 100m and this corresponds to 5.2% of the data. It should be noted that times when the noise in the horizontal error trajectories crossed the 100m level were not included in 540s figure. This is because it is expected that the navigation error in the vehicle trajectory would follow the average trajectory and not the noisy one.

These results suggest that even if one attempts to model the selective availability noise and if only four satellites are used in the solution, then the SA noise processes which change fairly quickly (such as over minutes) can do the damage in the horizontal plane that they are intended to do. Also, by changing the SA processes fairly quickly the augmented filter tends to lose its effectiveness by estimating the processes to be zero (which makes sense because the models are of zero mean processes). In most real situations it is impractical to estimate the SA processes because of the high-dimensionality in the state vector. If SA is not present, the extra states will actually tend to hurt filter performance because the model is not correct in this situation. As mentioned earlier, a simple way to account for the presence of SA is to increase the size of the measurement noise covariance and to leave the time-wise correlation structure of the SA process unmodelled. This leads to quite an increase in the $\mathbf{R}_k$ matrix. Usually the nominal measurement noise variance is on the order of 100m^2 whereas if SA is added the resulting variance is 1200m^2 and this tends to de-weight the measurements considerably.

If, however, an all-in-view solution is implemented with the casual SA modelling, the effects of SA are dramatically reduced. This is reasonable from the viewpoint that the HDOP in the all-in-view solution is usually about 1.1 whereas it is typically 1.5 in the four-satellite solution. Results of such a simulation yielded a maximum SA induced error of about 60m which checks with the HDOP values just mentioned.

3.4.5. Calculation of integrity algorithm performance

This section describes how the information obtained from the Monte Carlo simulation is used to calculate certain performance characteristics of an integrity algorithm. Regardless of the method use to calculate the observable statistics, these data are compressed into a single scalar test statistic which is compared with a predetermined threshold and a decision is made as to which hypothesis will be supported. In simulation analysis, the truth is known so the results of each test can be checked for correctness. For a given set of experiments, the unconditional and conditional alarm rates as well as the conditional miss rate and detection probability are computed as given in Eq. 3.29.

$$\begin{aligned}
\mathrm{P_a} &= \frac{\text{Number of experiments when q} > \text{Q}}{\text{Total number of experiments}} \\
\mathrm{P_{fa}} &= \frac{\text{Number of experiments when q} > \text{Q but r} \leq \mathrm{r_0}}{\text{Number of experiments where r} \leq \mathrm{r_0}} \\
\mathrm{P_m} &= \frac{\text{Number of experiments when q} \leq \text{Q but r} > \mathrm{r_0}}{\text{Number of experiments where r} > \mathrm{r_0}} \\
\mathrm{P_d} &= 1 - \mathrm{P_m}
\end{aligned} \tag{3.29}$$

where

$$\mathrm{q} = \text{test statistic}$$

$$\begin{aligned} Q &= \text{threshold} \\ r &= \text{radial error in navigation solution} \\ r_0 &= \text{tolerable radial error specification} \end{aligned}$$

The first two probabilities are computed with normal measurement errors including SA and this allows the threshold to be set empirically according to a specified acceptable alarm rate. The last two probabilities are usually computed when an unmodelled error has been added to the measurement vector to induce an error which exceeds the protection level. The miss rate is dependent on the size of the bias added so this parameter is generally calculated for a number of different bias levels.

Another performance measure which has been suggested in [15] is the unconditional miss rate. This would be calculated as shown in Eq. 3.30.

$$P_{um} = \frac{\text{Number of experiments when } q \leq Q \text{ but } r > r_0}{\text{Total number of experiments}} \tag{3.30}$$

This parameter would represent the fraction of time the horizontal error was larger than the protection level regardless of whether the system was operating normally or abnormally. The experiments for such a simulation would need to contain the correct proportion of failures and non-failures from the statistical population which contains all the possible measurement errors. This would be a difficult experiment to design since the prior probability of a satellite failure of a given magnitude is not well known. It is expected to be very small, though.

One may wish to take a simpler approach to calculating the unconditional miss rate. This could be done by first discretizing the possible failure sizes and then by averaging the miss rate for each failure size to form an overall miss rate. It turns out that small bias errors do not induce horizontal errors in the all-in-view solution which are larger than the protection level. Also, very large catastrophic errors can be detected with zero miss rate. The bias levels in between this range are known as soft-failures whereas the very large errors are hard-failures. If hard and soft failures are equally likely to occur (for lack of a better ratio) then the unconditional miss rate can be formulated as in Eq. 3.31.

$$\mathrm{P_{um}} = \frac{1}{2}\,\mathrm{Prob(miss} \mid \mathrm{soft\text{-}failure)} + \frac{1}{2}\,\mathrm{Prob(miss} \mid \mathrm{hard\text{-}failure)} \qquad (3.31)$$

The miss rate for the hard-failures is zero so the second term vanishes. The conditioning on the soft-failures can be removed by averaging the miss rates over different levels of bias as shown in Eq. 3.32.

$$\mathrm{P_{um}} \;=\; \frac{1}{2}\sum_{\mathrm{k=0}}^{\mathrm{m}} \mathrm{Prob(miss} \mid \mathrm{bias} = \mathrm{k}\Delta\mathrm{b)} \qquad (3.32)$$

where

$$\begin{aligned} \Delta\mathrm{b} &= \text{bias increment} \\ \mathrm{m}\Delta\mathrm{b} &= \text{bias at which there is zero miss rate} \end{aligned}$$

All of the performance measures described above are highly dependent on the satellite geometry so the correct proportion of good and bad geometry is important in obtaining reasonable estimates of these parameters. In [15] the miss rate for one bias level and a given protection level was computed as a weighted sum of the miss rates during good and bad subsolution geometry periods. An extreme example of such a calculation is given below and it shows the dramatic effect that a small proportion of poor geometry can have on the overall miss rate. Suppose that only half of the soft-failures can be detected during bad subsolution geometry and that the miss rate during normal geometry is .01. Then if the poor geometry lasted 6 percent of the time, the overall miss rate would be .4 as shown below.

$$\begin{aligned} P_m &= (.06)(.5) + (.94)(.01) \\ &\approx .04 \end{aligned}$$

In this example the overall miss rate is dominated by the integrity algorithm performance during the poor geometry periods. It can be seen that any increase in the proportion of poor geometry as would occur if one or more satellites was not operating (and known to be out of service), would have a noticeable effect on the miss rate. The value of .5 for the miss rate during poor subsolution geometry is clearly too high to provide good failure detection. Further redundancy will be needed to bring this value down to the size of the miss rate during good geometry.

The miss probability is related to the action of the integrity test when the horizontal error is out of specification. In the usual situation the user solution is within specification and it is desirable that the test has a reasonably very alarm rate during these times. Even if the alarm rate for each test appears to reasonably small, the probability of an alarm over an extended mission of many hours may not be acceptable. The probability that an alarm occurs over a mission when m tests are performed can be computed as follows. It is assumed that each test is independent and has an alarm rate α.

$$\begin{aligned} &\text{Prob(at least one alarm during m tests)} \qquad (3.33) \\ &= 1 - \text{Prob(no alarms during m tests)} \\ &= 1 - \text{Prob(no alarm at } t_1 \text{ and no alarm at } t_2 \\ &\qquad \cdots \text{ and no alarm at } t_m) \end{aligned}$$

Table 3.2: Probability of alarm on a five hour mission

α	Prob(alarm during mission)
.001	.973
.0001	.698
.00001	.0354
.000001	.00359
.0000001	.000360

$$= 1 - \left[\text{Prob(no alarm at } t_i)\right]^m$$

$$= 1 - [\ 1 - \text{Prob(alarm at } t_i)\ ]^m$$

$$= 1 - (\ 1 - \alpha\ )^m$$

Suppose that the test is performed every 5 seconds and the mission lasts five hours. This would generate 3600 tests. If the alarm rate for each test was .001 then the probability of an alarm on this mission is .97 which means there would almost surely be an alarm. Table 3.2 gives similar results for different values of the alarm rate and 3600 tests.

Clearly the alarm rate will have to be $\leq$ .000001 before an acceptable overall alarm rate for a complete mission can be achieved. Because of the relationship between the alarm rate and the detection probability, it appears that the horizontal protection level will have to be relaxed considerably to allow such a small test size. These concerns will be addressed during the discussion of each integrity algorithm.

4. REVIEW OF INTEGRITY MONITORING METHODS

A number of different integrity monitoring algorithms have been presented over the past few years. All of these schemes show promise for satisfying integrity requirements with regard to the enroute and terminal approach phases of flight. Selective availability is the noise source which currently puts most schemes out of reach of satisfying the non-precision approach requirements of protecting against a 100m horizontal error with a 6-10s reaction time. The satellite geometry also plays a key role in all the schemes since if a certain amount of redundancy does not exist in all subsolutions of size one less than the number of satellites in view, then it is difficult to detect a large error on certain satellites. This section will present a review of the currently known integrity monitoring methods and how each method is affected by selective availability and poor subsolution geometry.

Two different approaches to integrity monitoring were first presented at the annual meeting of the Institute of Navigation (ION) in 1986. The work presented by Lee [17] was based on point solution methods whereas the method presented by R.G. Brown and Hwang [6] used a Kalman filter approach. Since then, there has been a number of methods proposed which are based on either the pointwise solution or the Kalman filter solution.

4.1. The Range and Position Comparison Methods

At the time when Lee was formulating his integrity monitoring method, the 18 satellite configuration plus three active spares was the constellation that was envisioned. In order to be conservative with respect to satellite availability, the scheme he developed assumed that only five satellites would be used in the algorithm. Lee presented two test statistics, one in the range domain and the other in the solution space. The range comparison method (RCM) is based on the difference between a range measurement and the predicted range using the solution from the remaining four satellites. The position comparison method (PCM) is based on differences in one coordinate direction between two solutions formed by using different subsets of four satellites from the set of five. Each of the range or position differences that can be formed has the form shown in Eq. 4.1 where each difference is related to the error on each measurement through a scale factor related to the satellite geometry. It turns out that in the five satellite case there is only one independent equation (or one degree of freedom) among all such differences and thus all position or range comparisons are related to each other through a known scale factor.

$$Q = \sum_{i=1}^{5} a_i e_i \tag{4.1}$$

where

e_i = range error on the ith satellite

a_i = coefficients that depend on satellite geometry

The integrity check is a binary hypothesis test as defined below.

H_0 hypothesis : no-failure state, Q is zero mean with known variance

H_1 hypothesis : failure state, Q has non-zero mean but known variance

The test is

$$\begin{aligned} &\text{accept } H_0 \quad \text{if} \quad |Q| \leq d \\ &\text{accept } H_1 \quad \text{if} \quad |Q| > d \end{aligned}$$

where

Q = test statistic

d = threshold for particular difference and given alarm rate

A desirable attribute of this method is that it is possible to derive closed form solutions for the false alarm rate, the miss rate and the detection probability and these expressions are given in Eq. 4.2. In this analysis, it is assumed that only one satellite can fail at a time and each is equally likely. The satellite failure is parameterized as being fixed at M or -M with no added noise.

$$\begin{aligned} P_{fa} &= \text{Prob}[\,|Q| > d \text{ given } Q \sim N(0, \sigma_Q^2)] \\ P_m &= \frac{1}{5}\sum_{i=1}^{5} \text{Prob}[|Q| \leq d \text{ given } Q \sim N(a_i M, \sigma_{Q_i}^2)] \\ P_d &= 1 - P_m \end{aligned} \tag{4.2}$$

where

$$\sigma_Q^2 = \sigma_e^2 \sum_{i=1}^{5} a_i^2$$

$$
\begin{aligned}
\sigma^2_{Q_i} &= \sigma^2_e \sum_{j \neq i}^{5} a_i^2 \\
\sigma^2_e &= \sigma^2_N + \sigma^2_{SA} \\
\sigma^2_N &= \text{inherent GPS measurement error variance} \\
&= (9.7\text{m})^2 \\
\sigma^2_{SA} &= \text{Selective availability noise variance} \\
&= (33.3\text{m})^2
\end{aligned}
$$

The coefficients of the range errors given in Eq. 4.1 are determined by the satellite geometry for each subsolution so the threshold and the miss probability for each test are different. To obtain an estimate of the miss probability which is representative of the overall system performance at a typical location, the satellite geometry in Chicago was sampled every half hour for 24 hours and this generated 48 geometries. (One of these was removed since only four satellites were visible.) At each location, the level of the satellite failure was varied between 50m and 400m in 50m increments and the detection probability for each bias was calculated. The results for each bias level from all the geometries were pooled together to generate an overall detection probability in Chicago. The threshold was set to yield a false alarm rate of .004.

The results of this analysis are not encouraging in that the detection probability for a range bias as large as 400m was only .75 with SA and .90 without SA. Lee concluded that any other receiver-based detection scheme which uses only five satellites would yield similar results. However, it is possible that the results reported by Lee are pessimistic for a number of reasons. No mention is given to the level of horizontal error which is induced in the five satellite solution due to the different levels of range bias error. It may turn out that the detection probabilities could be significantly improved if the final results of each experiment are made in the horizontal plane. In this way, experiments where the range errors were not detected would be counted as a miss only if the horizontal error in the five satellite solution was larger than a specified protection level. Is is also possible that the results are pessimistic due to certain poor geometries which dominate the statistics. When more than five satellites were visible, Lee chose the set of five which yielded the best HDOP in combination. The criteria should be to choose the set of five satellites whose largest subsolution HDOP is the smallest among the worst subsolution HDOP from each set of five satellites. In this way the coefficients in Eq. 4.1 will be as small as possible and thus the variance associated with the distribution will be as small as possible. These concerns point out the problems encountered with a five satellite integrity approach [7].

Lee has since followed up this earlier work with a six satellite study in which the test statistic is a 2-tuple since there are now two independent equations which can be obtained from all position and range differences [18]. Once again Lee concluded that the RCM or the PCM is not robust enough to satisfy the non-precision requirements but may provide integrity at a larger horizontal error specification.

4.2. A Geometrical Approach to RAIM

A method of RAIM proposed by A. Brown and Jessop [4] which is also based on the point solution addresses the problem of poor subsolution geometry and places the integrity check in the horizontal plane. Brown assumed that a PDOP $\leq$ 6 corresponds to a 100m 2-drms horizontal error. Selective availability is the primary noise source and the PDOP criteria places a fundamental limit on the geometry which will bound the error at this level. For integrity purposes all n combinations of n-1 satellites must also have a PDOP ≤ 6 to have similar horizontal error statistics. The all-in-view solution is considered the navigation solution.

The horizontal protection provided by such geometry is arrived at with the following heuristic argument. If no large errors are present in the set of measurements and the geometry requirements are met, then all n solutions should lie within a circle of radius 100m. Thus the distance between two solutions may be as large as 200m with no failure present. Therefore, if a distance can be found which is larger than 200m then the conclusion is that a failure has occurred in a satellite. It is then postulated that the protection level provided by such a scheme is 300m since the subsolution which does not use the failed satellite is within 100m of truth and a 200m distance among solutions is allowed before the alarm is raised. Similar arguments are made for different size circles within which all un-failed subsolutions must lie and it is pointed out that these tests will lead to different alarm rates as shown in Table 4.1.

Table 4.1: Protection of RAIM verses accuracy of solution

Alarm limit	100m	150m	200m
Integrity level	200m	250m	300m
False alarm rate	high	medium	low

An extensive simulation of the satellite availability and the resulting GDOP over the CONUS was performed using two different 24 satellite constellations, one having three planes and the other six planes. Table 4.2 gives the results of the availability analysis for both configurations using a 5 degree mask angle.

Table 4.2: Percentage of time n satellites are visible

3-plane	25.9	17.3	29.2	24.0	3.5	0.3
6-plane	2.3	37.7	49.2	10.7	0.1	0.0
n	6	7	8	9	10	11

Based on the GDOP calculations for subsets of the available satellites, the percentage of time that detection and isolation would not be possible with these two configurations was calculated using the simple criteria for the PDOP of each subsolution. Although no specific isolation technique was given, it is implied that isolation is possible if every subset of n-2 satellites satisfies the PDOP criteria. The results of such calculations are given in Table 4.3 and Table 4.4.

Table 4.3: Percentage of time detection is not possible

3-plane	6-plane
% time	% time
.46	6.56

Table 4.4: Percentage of time isolation is not possible

Alarm limit	200m	400m	800m
PDOP limit	6	12	24
3-plane	45.6	31.1	20.5
6-plane	24.2	12.4	2.4

There are a number of useful results of this study. It is clear from Table 4.3 that the 3-plane constellation is better for detection purposes since the poor subsolution geometry occurs only .5 percent of the time compared to 6.6 percent for the six plane. The six-plane configuration is more symmetric than the three-plane so the poor geometries get repeated more times per day. In terms of isolation, the six plane configuration appears to be better but the results are not spectacular. Table 4.4 suggests that if one is willing to put up with a larger PDOP limit and thus a larger protection level, then the isolation percentage can be improved. It is believed that these results are pessimistic due to the oversimplified criteria and that much better isolation results can be achieved [26]. This work does point out, though, the dramatic role that the satellite geometry plays in the integrity problem, since it puts a fundamental limit on the percent of time that the geometry yields the minimum redundancy for integrity monitoring.

4.3. The Maximum Separation Among Solutions

A similar approach to the one mentioned above was presented independently at the first meeting of the Satellite Division of the ION in 1987 along with A. Brown's work. This analysis by R.G. Brown and McBurney [7] used the maximum separation among subsolutions as the test statistic and presented Monte Carlo simulation results that are more optimistic than Lee's results and are in basic agreement with those given by A. Brown. The algorithm for performing the integrity check is as follows.

1. Compute the n least-squares solutions using n-1 satellites leaving one of the satellites out at a time.

2. Compute the largest distance among all such solutions in the horizontal plane. This is the test statistic.

3. Compare the test statistic with a predetermined threshold which is determined empirically for a specified alarm rate.

4. If the test statistic is larger than the threshold, conclude that a failure is present and that the error in the all-in-view solution is larger than the level protected by the threshold.

The rational behind this approach is that if only one satellite has failed, then the subsolution which does not use this satellite will not be affected by this error. This is the good solution and it should be displaced from the other solutions which all use the errant satellite. It may turn out that the largest distance is between subsolutions both of which use the failed measurement. Thus identification is not provided for in this approach. However, all the subsolutions have been computed so it is not a burden to compute the measurement residuals for each solution. The good solution should have the smallest squared residuals.

The alarm rate was set at .003 and a Monte Carlo simulation was performed to determine the threshold which yielded this alarm rate for three different noise levels corresponding to different levels of selective availability. The 24-satellite six-plane constellation was sampled in Chicago every 4.5 minutes for 36 hours and this yielded 480 different geometries. If any subsolution had an HDOP $\geq$ 3.0 then this geometry was removed from the simulation and this accounted for roughly 5 percent of the geometries. This agrees with the percentage of the time that failure detection is not possible as determined by A. Brown and repeated in in Table 4.3. The measurement errors were distributed as the sum of the inherent GPS error with a standard deviation of 14.1m and one of either of two levels of SA. These levels are referred to as half and full SA and have standard deviations of 16.7m and 33.3m

Table 4.5: Threshold for alarm rate of .003 at three noise levels in Chicago

Level of SA	No SA	Half SA	Full SA
Standard deviation	14.1m	21.7m	36.2m
Threshold	85m	130m	217m

respectively. The thresholds obtained with good geometry and the different levels of SA are given in Table 4.5.

It turns out that large horizontal errors due to selective availability, which appear due to a multiplicity of errors in all satellites, are very difficult to detect, and thus the threshold is set to a large enough value so that the protection provided by this threshold is out of the reach of SA during periods of good geometry. It is likely that SA will trigger the alarm at this threshold during a period of poor subsolution geometry, so these geometries were considered separately. Bias errors were then added to each satellite in succession at each geometry and the maximum separation among subsolutions and the all-in-view solution from each experiment were used to compute the miss rate for a given horizontal error protection level. Small bias errors did not induce enough large errors to compute a meaningful miss rate. On the other hand, very large bias levels were detected with essentially zero miss rate. The bias levels which are important are those in between these extremes. The horizontal protection provided by sufficiently small miss rates (less than .01) for the bias levels in this hard-to-detect range are given in Table 4.6.

These results point out the level of integrity which may be possible using GPS as the only source of redundancy during times when the subsolution geometry is acceptable. If all 24 satellites are functioning properly, this may be as high as 95 percent of the time. However the loss of even one satellite will surely decrease this

Table 4.6: Horizontal protection for alarm rate of .003 at three levels of SA

	Protection level		Bias levels
Level of SA	Optimistic	Pessimistic	when $P_d \geq .99$
No SA	100m	125m	200-300m
Half SA	175m	210m	300-400m
Full SA	250m	300m	450-550m

amount. If these poor geometry situations can be handled in some fashion, then this approach seems to be a viable method of integrity monitoring.

Using the data that was used to generate Table 4.6 it is possible to calculate an average miss rate as defined in Eq. 3.32. Suppose the protection level is set to 250m during full SA and the alarm rate remains at .003. For these conditions, one accumulates the number of misses due to different levels of bias error. For a bias less than 250m no horizontal errors larger than the protection level were induced. A hard failure corresponds to a bias larger than 700m and there are no misses for this situation. The number of experiments where the horizontal protection is exceeded are the points in the H_1 space and the number of these experiments where the alarm was not raised are misses. The number of misses for different levels of bias are given in Table 4.7. Starred quantities refer to an extrapolated quantity.

Using Eq. 3.32 the average miss rate and detection probability are calculated as shown below.

$$
\begin{aligned}
P_m &= \frac{1}{2}(\frac{14}{6028}) \\
&\approx .001 \\
P_d &= .999
\end{aligned}
$$

Table 4.7: Maximum separation misses for different failure levels

Bias level (m)	# Points in H_1 space	# of misses
0-250*	0	0
300	17	2
350	73	4
400	179	3
450	388	2
500	596	3
550	825	0
600	1025*	0
650	1300*	0
700	1600*	0
Total	6028	14

This parameter represents a three-fold increase in the integrity of the system at the given protection level with the caveat that some provision will be made to perform the integrity check during periods of poor subsolution geometry.

4.4. The Range Residual Squared Approach

An approach which is based in the measurement domain is the range residual squared (RRS) test and was first presented at the National Technical meeting of the ION in 1987 by Parkinson and Axelrad [27], [25], [26]. The test statistic is the sum of squares of the measurement residuals divided by the variance of the measurement errors. When the measurement errors are zero mean with known variance as described in section 3.2, then the test statistic is a chi-squared random variable with n-4 degrees of freedom. This statistic is formed as shown in Eq. 4.3.

$$Q = \frac{1}{\sigma^2}(y - \hat{y})^T(y - \hat{y}) \tag{4.3}$$

where

$$\sigma^2 = \text{variance of measurement errors}$$

The integrity check consists of the following simple hypothesis test.

H_0 hypothesis : no-failure state, $\mathbf{y}$ is zero mean and $Q \sim \chi^2_{n-4}$

H_1 hypothesis : failure state, $\mathbf{y}$ is not zero mean and Q is non-central χ^2

The test is

$$\begin{aligned} &\text{accept } H_0 \quad \text{if} \quad Q \leq d \\ &\text{accept } H_1 \quad \text{if} \quad Q > d \end{aligned}$$

where

$$\begin{aligned} Q &= \text{test statistic} \\ d &= \chi^2_{n-4}(\alpha) \qquad (4.4) \\ &= \text{threshold for test size } \alpha \\ n-4 &= \text{degrees of freedom of Q} \end{aligned}$$

As the above test suggests, the threshold is set according to the alarm rate α set by the test designer. This was the approach taken by Parkinson and Axelrad in [25]. The test also assumes that the measurement error variance σ^2 is known. The approach taken in [25] was to determine this parameter empirically for the particular receiver being used. The authors also considered a situation where a small bias associated with each measurement residual from a particular satellite was modelled and thus the reference distribution under the H_0 hypothesis was non-central chi-square with a known non-centrality parameter. This approach is somewhat less general and may lead to some loss of sensitivity in detecting biased measurements due to slowly varying satellite failures.

The results given in [25] were based on closed form solutions for the alarm rate and the probability of missed detection based on different values for the non-centrality parameter. The results are promising in that a measurement error bias as small as 100m can be detected with an acceptably small miss rate (not given). In [26] which is a follow-up paper on the earlier work, the test statistic was scaled differently and the threshold for the test was arrived at using purely empirical methods. The new test statistic is defined in Eq. 4.5.

$$Q = \sqrt{\frac{1}{n-4}(y - \hat{y}^T)(y - \hat{y})} \tag{4.5}$$

The performance results in [26] are based on extensive Monte Carlo simulation at the San Francisco Airport (SFO). The 24-satellite three-plane configuration was sampled every 15 minutes for 24 hours at SFO. The geometry at Chicago was also examined and yielded similar results for the case of normal measurement errors. The normal measurement errors were modelled as being a small bias plus a small amount of additive noise. As a result, each error was specified with a uniform random variable with support from -5m to 5m plus a Gaussian random variable which was $N(0,.16m^2)$. (These parameters were chosen to correspond to a certain manufacturer's receiver which was Doppler aided.) Selective availability was not considered.

It was found that none of the test statistics was larger than 8m so this value was chosen as the new threshold r_d. The statistics chosen for the measurement errors induced a maximum radial error of 19m in the no-failure experiments. Similar experiments were also performed for each of the subsolutions which left out one of the visible satellites. In this case, no test statistics were larger than 10m and the largest radial error was 37m, but the average was less than 25m. This 10m value was taken as the subsolution threshold r_i which is used for isolation purposes. (The radial position error which is the square root of the sum of squares of the errors in all three coordinate directions is the parameter of interest in [26] rather than the horizontal error considered in other works. This was the relevant parameter in the Gravity Probe B experiment at Stanford which requires a high degree of integrity in the navigation system which was the motivation for their work.)

The detection and isolation procedure which was developed is as follows.

1. Compute Q using all satellites in view (6 or more).
2. If Q is less than the threshold r_d, choose H_0 and the integrity check is complete.
3. If Q is larger than the threshold, declare a failure is present. Then compute Q_j for each subsolution leaving out one satellite at a time. There will be n of these solutions (i.e., j=1,n).
4. If only one Q_j is less than the subsolution threshold r_i, then the failure is isolated as the satellite which is excluded from this subsolution. If two or more Q_j are less than the subsolution threshold, then isolation is not possible.
5. If detection and isolation are successful, use the solution which omits the failed satellite as the navigation solution. If only detection is possible, use the all-in-view solution but recognize that a degraded solution is being used.

This algorithm was tested by adding different levels of bias to one of the normal measurement errors to simulate a satellite failure and then by accumulating the resulting statistics. The detection and isolation results for four bias levels are given in Table 4.8.

Table 4.8: Percent detection and isolation with r_d=8m and r_i=10m

Bias (m)	Detection	Detection and isolation	Detection but no isolation
100.0	100.0	72.2	27.8
50.0	99.94	50.5	49.5
37.5	98.70	34.2	64.5
25.0	76.80	6.40	70.4

Thus, it appears that the range residual squared statistic is a very powerful integrity monitoring method and if the actual measurement errors were similar to those modelled here, it appears that this method may satisfy the non-precision approach specification without SA and with good geometry. The radial error protection provided when detection and isolation is successful corresponds to the average radial error for a typical subsolution which was less than 25m for the measurement error statistics chosen. Bias errors as small as 75m were detected with an acceptably small miss rate.

However these results appear to be overly optimistic for a number of reasons. First of all, the measurement error model which was chosen is unrealistic because the noise it generates is bounded by the uniform distribution used to generate the normal measurement bias. This may have been acceptable if the variance of the Gaussian noise had been larger. The variance of the sum of the two random variables used to generate the measurement error is 8.473m^2 which is the variance of a uniform random variable plus the variance of the Gaussian random variable. This does not create the same distribution of noise as would be generated by a

Gaussian distribution with the same variance since the probability in the tails of the Gaussian distribution will be dramatically larger. This makes it difficult to predict results at higher noise levels.

Secondly, it is possible that the samples of geometry which were used in the simulation did not contain any poor subsolution geometry. For the six-plane configuration, the bad subsolution geometries may last as long as twenty minutes and thus it is rare that even spaced sampling will not yield poor geometries. However, in the three-plane configuration, the duration and distribution of the poor subsolution geometries is dramatically smaller [4]. Therefore, it is likely that no such geometries were contained in the simulation due to the 15 minute sampling interval. No details were given in the paper as to what the largest GDOP was for the geometries in the simulation. It turns out that detection of a bias error on a certain satellite during periods of poor subsolution is very difficult if the satellites are the only source of redundancy. It also appears that the three-plane configuration is less likely to be implemented so the geometry problems associated with the six-plane configuration may play a dominant role in the effectiveness of this method. (A more detailed discussion of this problem will be presented in section 4.10.)

A problem with the test statistic in Eq. 4.5 is that the alarm rate in not constant when different numbers of satellites are used in the solution. This was not a problem in the first test statistic given in Eq. 4.3 since the test size was kept constant for different numbers of measurements by choosing the threshold from the reference distribution with the correct degrees of freedom. For the test statistic in Eq. 4.5, if one keeps the threshold constant, then the alarm rate changes dramatically for different dimensions of the measurement vector. This is easily shown with an example. Eq. 4.5 can be rewritten as shown in Eq. 4.6 below.

$$t = \sqrt{\frac{1}{n-4}\sigma^2\chi^2_{n-4}} \tag{4.6}$$

Now suppose the threshold remains constant at t, and then calculate what the corresponding test size for different sizes of the measurement vector. The value for σ^2 will be taken as the effective value used in [26] which is 8.473m^2. The result of solving for χ^2 in Eq. 4.6 is given in Eq. 4.7.

$$\chi^2_{n-4} = \frac{t^2}{\sigma^2}(n-4) \tag{4.7}$$

Let α_i be the test size according to the given threshold and the number of degrees of freedom of χ^2.

For 6 satellites

$$\begin{aligned} \chi^2_2 &= \frac{64}{8.473} \cdot 2 \\ &= 15.106 \\ \text{Prob}[\chi^2_2 > 15.106] &= 0.0005 \end{aligned}$$

Therefore

$$\alpha_2 = .0005$$

For 7 satellites

$$\alpha_3 = .00005$$

For 8 satellites

$$\alpha_4 = .000004$$

For 9 satellites

$$\alpha_5 < .000001$$

When the threshold of 8m was set, none of the test statistics exceeded this value so it is not possible to calculate an alarm rate from the experimental data. In view of the above calculations, it is difficult to derive analytically what the alarm rate is since it is a function of the number of satellites being used. However, one could find an effective alarm rate by weighting each α_i by the relative frequency of occurrence that the corresponding number of satellites are in view. Using the percentages given in Table 4.2 for the three plane configuration, the overall alarm rate is roughly .0001 and is dominated by the six-satellite alarm rate. This is a rather small alarm rate compared with the methods presented earlier and this is desirable. This is probably close to the actual alarm rate for the test statistic given in Eq. 4.5. The alarm rate and protection level for different noise levels and the problems associated with poor detection geometry are still somewhat tentative and these concerns need to be addressed before this powerful method can be implemented.

4.5. Comparing Maximum Separation and RRS

This section contains a side-by-side analysis of the performance of two snapshot techniques using the same set of measurement errors for full SA ($\sigma = 36.2$m) and a single satellite failure. The results from a large simulation are presented and then a closer examination of the effects of the subsolution geometry is provided. Three examples were chosen which represent normal, borderline, and extremely poor subsolution geometry.

The samples of geometry at Chicago were the same as those used in the maximum separation simulations [7] and the poor geometry is removed by leaving out geometries where any subsolution HDOP > 3.0. This criteria yielded 3414 exper-

Table 4.9: Performance of maximum separation and RRS during good geometry

Protection level (m)	# points in H_1 space	Max. Sep. # misses	RRS # misses
100	3003	222	9
125	2661	151	9
150	2216	92	9
175	1706	36	9
200	1222	14	9
225	852	7	9
250	596	3	7
275	393	2	4
300	228	0	2
325	124	0	2
350	62	0	2

iments with good geometry. The measurement errors correspond to full SA and a 500m bias was added to simulate a soft-failure. The thresholds were chosen to correspond to an alarm rate of .003 during good geometry. The results of the simulation are given in Table 4.9 and show how the number of misses decrease as the horizontal protection level is increased.

The performance of the RRS statistic is superior in the smaller protection level range but the maximum separation statistic is better in the larger protection level range. The maximum separation statistic is more robust in that its performance improves dramatically as the protection level is increased. It is believed that the misses due to the RRS statistic in the 300m range are due to the effects of borderline subsolution geometry which was included in the simulation. Thus if the subsolution geometry is much better than the criteria used here (possibly HDOP < 2.0) it appears that the RRS statistic will provide sufficient performance at a much smaller protection level. This is probably too optimistic since the poor subsolution geometry conditions can occur a significant portion of the time [4]. The effects of this type of geometry will now be examined more closely.

For each of the different levels of poor subsolution geometry, the set of dilution of precision parameters are calculated for the all-in-view solution, and each of the subsolutions are given along with other statistics in Tables 4.10 , 4.11, and 4.12. A 500m error was placed in each satellite separately in addition to a different set of random measurement errors. Thus for seven satellites, there are seven experiments with seven different realizations of the measurement error vector. The left hand vertical column of each table gives the numbering for each experiment and is the row number of the measurement vector which contains the bias error. The numbering of each solution from left to right corresponds to the satellite which was removed. The all-in-view solution is denoted as AIV. The north and west position error are given for each solution and are labelled as x and y solutions in the tables. The last column for each experiment gives the maximum separation, and the horizontal error in the all-in-view solution labelled as M.S. and AIV Rad. respectively. Also appearing in this column are the numbers of the solutions between which the maximum separation lies and are labelled as i-j. The set of residuals formed using Eq. 4.3 for each subsolution and the all-in-view solution is also given in each table. The correct measurement error variance was used in the calculation of the residual statistic. The right most column of this set of statistics is used for detection and the rest are used for isolation.

In the detection problem, the threshold for the maximum separation is 217m for an alarm rate of .003 and the threshold for the range residuals is 13.9 for the same alarm rate and three degrees of freedom. For isolation purposes, the element with the smallest residual parameter in columns 1-7 of each row would be chosen as the good solution for each experiment. The smallest residual should lie along the main diagonal of this matrix.

In the example with normal geometry, both methods successfully detected the bias error in all of the experiments. This is referred to as normal geometry because the set of HDOP's is fairly homogeneous and all are less than or equal to 1.6. Notice that in experiments 3,4,6, and 7 the horizontal error in the all-in-view solution is well below 200m and these would be false alarms for for such a protection level. Isolation was provided in all experiments except for experiment 5 where solution 2 had a smaller RRS statistic than did solution 5.

In the borderline geometry, solution 4 has an HDOP $=$ 2.4 while all other solutions had an HDOP $<$ 1.6. Even though this geometry satisfies the criteria that all subsolutions have an HDOP $<$ 3.0, the RRS statistic fails to detect the 500m error in satellite 4. Detection and isolation is provided for all the other experiments though. Notice that the maximum separation usually occurred between the solution with the high HDOP and one of the low HDOP solutions. This is because the variance of this solution is larger than the other solutions and thus the solution is much larger than any of the other solutions. This leads to separations which are larger than in the previous example. Notice also that all the solutions in experiment 4 are clustered fairly close together. The only reason why detection is possible is that the solution with poor geometry was small enough to still generate a large test statistic.

Table 4.10: Maximum Separation and RRS results for normal subsolution geometry, T=20.7 hours

Solution	1	2	3	4	5	6	7	AIV		
HDOP	1.25	1.60	1.08	1.28	1.33	1.07	1.10	1.05		
PDOP	2.18	2.24	2.19	2.67	1.99	1.98	1.88	1.82		
TDOP	1.27	1.09	1.07	1.64	0.98	0.98	0.95	0.94		
GDOP	2.53	2.48	2.43	3.14	2.24	2.20	2.11	2.05		
Exp.#	x solutions y solutions								M.S. i-j	AIV Rad.
	-16.3	-304.	-151.	-61.9	-0.9	-137.	-116.	-111.	341.	
1	29.3	138.	245.	180.	295.	232.	235.	207.	2-5	236.
	-352.	3.0	-298.	-246.	-442.	-307.	-287.	-293.	498.	
2	-3.1	-6.0	-109.	-141.	-231.	-101.	-151.	-114.	2-5	315.
	154.	93.4	7.4	66.4	167.	55.4	99.3	88.4	246.	
3	-203.	-78.4	-5.0	-67.8	-18.5	-48.3	-147.	-80.3	1-3	119.
	110.	58.3	141.	-14.4	206.	163.	140.	133.	221.	
4	-51.5	-121.	-102.	-10.8	-36.3	-124.	-136.	-93.9	4-5	162.
	114.	490.	181.	254.	4.9	163.	181.	177.	545.	
5	284.	278.	161.	121.	28.0	179.	143.	164.	2-5	242.
	-40.6	-279.	10.9	-232.	126.	30.6	-22.2	-24.9	456.	
6	101.	-21.0	38.1	189.	190.	17.4	55.0	71.3	2-5	75.5
	48.9	-167.	48.6	176.	-7.8	-3.8	-16.5	5.5	344.	
7	-214.	-195.	-173.	-229.	-142.	-124.	0.6	-132.	2-4	132.
Exp.#	Range residuals squared divided by σ^2									
1	5.8	51.7	44.2	68.8	50.8	47.9	68.8	73.5		
2	25.7	0.2	51.4	47.6	11.0	44.5	43.3	51.8		
3	89.6	122.	0.9	121.	110.	77.7	92.9	122.		
4	41.8	42.4	44.3	4.6	35.7	7.6	34.2	45.6		
5	29.2	2.4	59.5	48.7	4.6	50.6	56.8	59.7		
6	132.	95.8	110.	52.0	91.6	8.9	132.	134.		
7	98.0	95.1	77.5	57.3	112.	109.	0.1	112.		

Table 4.11: Maximum Separation and RRS results for borderline subsolution geometry, T=30.3 hours

Solution	1	2	3	4	5	6	7	AIV		
HDOP	1.50	1.26	1.56	2.41	1.35	1.27	1.29	1.21		
PDOP	2.60	2.60	2.93	6.34	2.55	2.44	2.45	2.40		
TDOP	1.48	1.54	1.83	3.46	1.46	1.38	1.37	1.37		
GDOP	2.99	3.02	3.46	7.17	2.94	2.80	2.81	2.77		
	x solutions								M.S.	AIV
Exp.#	y solutions								i-j	Rad.
	-28.5	-343.	-311.	-660.	-420.	-360.	-384.	-327.	696.	
1	57.9	117.	185.	349.	159.	199.	61.7	134.	1-4	353.
	-120.	18.5	-58.3	-379.	62.0	-45.5	-107.	-55.9	502.	
2	-64.1	-2.0	-88.6	128.	-113.	-101.	-144.	-80.6	4-5	98.1
	-11.6	63.5	-11.9	-8.6	179.	38.4	119.	65.6	351.	
3	324.	302.	45.0	352.	273.	358.	371.	304.	3-7	311.
	384.	370.	350.	167.	368.	344.	317.	347.	251.	
4	-216.	-182.	-195.	-90.4	-212.	-210.	-244.	-206.	1-4	403.
	305.	119.	239.	463.	-10.3	190.	169.	176.	530.	
5	-81.2	-109.	164.	-234.	3.3	-70.5	-56.2	-48.5	4-5	182.
	-312.	-70.0	-3.3	369.	-138.	-15.7	-92.9	-87.7	788.	
6	225.	186.	451.	-139.	182.	26.6	161.	168.	1-4	189.
	-238.	-147.	-170.	178.	-52.7	-119.	-3.6	-103.	469.	
7	-92.9	-174.	-354.	-309.	-141.	-95.4	-3.0	-127.	1-4	163.
Exp.#	Range residuals squared divided by σ^2									
1	1.4	92.0	93.1	67.4	75.9	66.0	60.4	95.4		
2	70.3	0.5	74.7	48.5	43.2	71.9	47.3	74.7		
3	56.2	62.4	5.2	61.1	33.5	43.1	32.0	62.5		
4	10.3	4.6	11.7	3.7	10.8	11.6	2.1	11.8		
5	61.1	35.0	40.0	57.8	0.2	75.4	78.3	78.7		
6	83.1	132.	68.3	79.7	131.	0.1	136.	136.		
7	90.0	82.5	65.6	89.3	104.	102.	4.2	110.		

Table 4.12: Maximum Separation and RRS results for poor subsolution geometry, T=20.025 hours

Solution	1	2	3	4	5	6	7	AIV		
HDOP	1.27	1.27	25.3	1.30	1.21	1.26	1.22	1.12		
PDOP	2.77	2.59	89.4	2.79	2.60	2.58	2.59	2.52		
TDOP	1.66	1.47	47.1	1.67	1.51	1.46	1.51	1.44		
GDOP	3.23	2.97	101.	3.25	3.00	2.97	3.00	2.90		
	x solutions								M.S.	AIV
Exp.#	y solutions								i-j	Rad.
	70.8	272.	4156	260.	148.	176.	188.	189.	4112	
1	15.4	145.	482.	35.7	191.	119.	114.	114.	1-3	220.
	-198.	78.3	3295	-148.	-149	-222.	-199.	-135.	3528	
2	-97.0	36.6	274.	-31.3	-19.3	-6.5	-161.	-44.7	3-6	142.
	327.	321.	578.	328.	326.	338.	330.	328.	258.	
3	19.7	17.8	43.8	21.0	25.5	16.5	22.9	20.6	2-3	329.
	244.	192.	2034	52.4	174.	218.	115.	169.	1983	
4	-54.3	-109.	55.5	9.5	-128.	-139.	-216.	-118.	3-4	205.
	13.2	-128.	-9458	-148.	-31.3	-254.	-109.	-116.	9521	
5	231.	118.	-744.	158.	-37.4	184.	137.	123.	1-3	169.
	-200.	-369.	6286	-321.	-280.	-0.4	-243.	-231.	6687	
6	91.1	12.7	670.	164.	158.	-36.1	42.6	65.2	2-3	240.
	-102.	-194.	-5077	102.	-42.9	-79.1	52.6	-50.2	5188	
7	-189.	-200.	-612.	-312.	-159.	-133.	43.1	-145.	3-4	154.
Exp.#	Range residuals squared divided by σ^2									
1	2.15	35.7	33.9	32.6	23.5	52.5	52.9	52.9		
2	102.	2.9	102.	116.	113.	95.7	56.5	116.		
3	0.3	0.2	0.2	0.3	0.2	0.1	0.3	0.3		
4	35.7	55.3	52.4	3.0	56.1	49.9	13.9	56.7		
5	65.2	126.	20.9	122.	0.7	74.0	125.	126.		
6	143.	99.5	95.7	114.	105.	0.8	145.	147.		
7	148.	106.	127.	66.1	157.	156.	0.5	158.		

In the example of extremely poor subsolution geometry, solution 3 had ın HDOP = 25.3 while all other solutions had an HDOP < 1.3. In this case the RRS statistic for experiment 3 has collapsed to almost zero while the maximum separation was still large enough to allow for detection. Isolation was provided for all experiments except experiment 3. Once again there is a clustering of solutions in experiment 3 but the solution with poor geometry is large enough to provide a large separation so the large variance is actually some help here. Notice that the error in solution 3 for this experiment is almost 600m for zero mean measurement errors. This example shows how the poor geometry affects the alarm rate when no failure is present. All of the maximum separations occur between the solution with bad geometry and generate very large statistics.

These examples point out the problems which appear due to poor subsolution geometry. Even though the maximum separation technique makes the correct decision in these experiments, it suffers from a high false alarm rate due to the poor geometry when no failure is present. The RRS statistic is somewhat different in that it gets less noisy during periods of poor subsolution geometry. Thus the alarm rate is not effected by this geometry but it will always fail to detect large errors in this situation. Clearly the satellite geometry plays a limiting role in the effectiveness of receiver autonomous integrity monitoring.

4.6. An Approach for Testing the Probability of a Satellite Failure

The Kalman filter approach presented by R.G. Brown and Hwang is based on the Magill parallel filter method and computes the posterior probabilities that one of the satellites has failed with an additive ramp type error [6], [9]. The underlying theme of this approach is that if one of the measurements is being corrupted with an additive ramp type failure and if a similar ramp is subtracted from the measurements, then the residuals associated with such a filter will remain small. Hwang contributed a parameterization scheme which models all possible ramps according to their bias and slope and the result is that the number of filters required is equal to the number of satellite measurements being processed.

The hypothesis associated with each filter is whether a non-trivial ramp failure is present on a specific measurement and also that all the other measurements are unbiased. This is still a binary hypothesis test where the null hypothesis is that no satellites have failed and the alternative hypothesis is a composite of the hypotheses that one of the satellites has failed and all the others have not. The parameter space for each source hypothesis is divided into two regions. A null region close to the origin of each satellite space represents the no-failure region. The remaining region in each space represents the combinations of bias of slope that would be considered a failure. The posterior probability associated with each hypothesis is updated recursively with the ratio of likelihood functions over these regions.

When this approach is implemented, the statistics used for failure detection are the posterior probabilities. If no failure is present then the probability associated with the null hypothesis should be close to unity while the probabilities for each source hypothesis should be close to zero. If a single failure is present, the source hypothesis associated with the failed satellite will converge to unity while all the other probabilities should converge toward zero as more and more measurements are processed.

The time required for the probabilities to adequately converge depends on a number of factors. First a threshold must be chosen which corresponds to an acceptable convergence level for the failure hypotheses and the no-failure hypothesis. The severity of the failure also dictates how fast the probabilities will converge. If the aircraft is in a modest acceleration environment then this will tend to obscure the failure due to the increased noise in the residuals. Selective availability would also cause problems with this scheme since the bias and slope of this process may be mistaken as a slow satellite clock failure and may cause a false alarm. Poor subsolution geometry may lead to problems (although this has not been investigated) since the failures on certain satellites would not be observable in the residuals during these periods and would probably lead to a missed detection.

It should be noted that identification is directly provided for in this approach since the source hypothesis with the largest probability would be considered the failed satellite if the alarm was raised. Another unique quality of this scheme is that the redundancy does not necessarily have to be in the form of redundant satellite measurements even though it will handle this situation. It will actually work in some cases with only four satellites. There is some redundancy built directly into the Kalman filter model with regard to the vehicle dynamics and the clock stability. In some cases explored by the Hwang [9], additional acceleration states were added to the filter model to increase the effectiveness of the scheme during acceleration environments as would be experienced during a non-precision approach.

The results of this study were tabulated as to the time required for detection and isolation of different size failures during different acceleration environments and with different sensor information such as baro-altimeter and Doppler aiding. Some results were also given where the filter included the extra acceleration states. These

Table 4.13: Detection results using posterior probabilities

Flight environment	Model dynamics	Measurements	Time to detection	Time to isolation
En route	low	4 sats.	14s	20s
En route	low	4 sats. & baro.	14s	15s
En route	low	4 sats. & low quality Doppler	14s	15s
En route	low	4 sats. & high quality Doppler	2s	3s
NP approach	high	4 sats.	25s	none
NP approach	high	4 sats. & baro.	15s	28s
NP approach	high, with acc. states	4 sats.	25s	no isolation
NP approach	high, with acc. states	4 sats. & baro.	13s	32s

cases are summarized in Table 4.13 for a 3m/s ramp type failure. Results of other failures modes were also given in [9].

These results show the effectiveness of this approach in detecting and isolating small satellite failures. In the cases presented, the error on the pseudorange had not even reached 100m before detection was successful and thus the induced error in the horizontal plane was probably small (though this result was not given). It is clear that the redundant measurement information was helpful for identification of the errant satellite.

There are a few drawbacks to this approach however. The effects of selective availability were not considered and this noise source would reduce the effectiveness of the scheme considerably. The scheme is really based in the measurement space and the result of the hypothesis test does not lead to an inference about the parameter of interest which is the horizontal position error. The scheme would probably pre-detect the failure before the induced error in the position states is large and this is desirable but may lead to a higher false alarm rate with SA or high dynamics. Since the decision is made according to posterior probabilities it is difficult to generate performance measures such as the alarm rate and the detection probability with this approach. Some of these concerns were addressed in a follow up work which will now be discussed.

4.7. A Likelihood Ratio Approach Using Parallel Filters

In an update of the work presented above an attempt was made to form a closer connection between the test statistic and the parameter of interest which is the horizontal error. The test statistic given by McBurney and R.G. Brown [23] is formed using a set of statistics obtained over a 10s window and in this way the test is brought into the framework of the non-precision approach requirements of protecting against an excessive horizontal error within a 10s reaction time.

The border between the null and failure regions of each bias and slope parameter space is defined so that points inside the null region would not induce a horizontal error larger than the specified protection level. The connection between the range error and the induced horizontal error is through a typical value of HDOP = 1.5. Regretfully, this is a weak connection in an absolute sense. Rather, it is a good statistical bridge if one considers an ensemble of pseudorange errors and the corresponding induced horizontal errors. Thus the null space is more forgiving here than it was in the earlier approach in that sizable range errors are not considered to be failures. The parameter space is discretized to a reasonably small number of points and each point corresponds to a filter which subtracts a particular ramp function from one of the range measurements. (Doppler measurements were also considered so the slope for each point was subtracted from these measurements.)

The H_0 hypothesis for this test is that the horizontal error is less than the protection level and is represented with points from the null region of each parameter space. An additional point is added which assumes all measurements are unbiased and is referred to as the null filter. The H_1 hypothesis, where the horizontal error is larger than the horizontal protection level, is represented with the points outside the null regions from each parameter space. There are two possible decisions which could be made. The D_0 decision supports H_0 and likewise the D_1 decision supports H_1. The decision rule is formulated to minimize the average cost associated with each decision, given the data, where there is no cost associated with a correct decision. The test is given in Eq. 4.8 where the test statistic is a ratio of exponentials each of which has an exponent which is a running sum of weighted residuals squared (RSWRS) associated with a specific filter.

$$\text{Choose } D_1 \text{ if } Q > \lambda_0 \tag{4.8}$$

where

$$Q = \frac{\Sigma_{j=1}^{n}[\Sigma_{i=1}^{N_1} e^{-\frac{1}{2}(\text{RSWRS})_{ij}}]}{\sum_{j=1}^{n}[\sum_{i=1}^{N_0} e^{-\frac{1}{2}(\text{RSWRS})_{ij}}] + e^{-\frac{1}{2}(\text{RSWRS})_{\text{null}}}}$$

$$\text{RSWRS} = \sum_{k=1}^{N} (y_k - \hat{y}_k)^T \mathbf{V}_k^{-1} (y_k - \hat{y}_k)$$

y_k = measurement less a particular ramp function

$$\mathbf{V}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$$

$$\lambda_0 = \frac{P(H_0)}{P(H_1)} \cdot \frac{nN_1}{nN_0 + 1} \cdot \frac{C_f}{C_m}$$

$P(H_0)$ = prior probability of each point in null space

$P(H_1)$ = prior probability of each point in failure space

N_0 = number of points in each null space

N_1 = number of points in each failure space

N = number of steps in window

n = number of satellites used in the solution

The scheme was tested via Monte Carlo simulation. An optimistic Kalman filter model was implemented which took advantage of Doppler measurements, the stability of a good crystal clock (with parameters given in Table 3.1), a mild acceleration environment ($A_1 = 1.0$ m^2/s^2) and a measurement model for SA which was the same as the simulated SA process. A six-in-view solution with good geometry was used. The SA noise generated consisted of the damped cosine process (σ = 10.2m) given in case 1 of the selective availability study in Chapter 3 plus another similar process which was a quasi-bias (σ = 31.7m). The time constants of these processes were 3.5 and 71 minutes respectively. The standard deviation of the white measurement noise for the pseudorange and the range rate were 15m and 2m/s respectively. The dimension of the augmented state vector was 32 ($= 8 + 6 \times 4$) as a result of accounting for the SA processes.

At the time this research was performed there was some hope that the non-precision approach specification could be relaxed to 200m and this was the protection level used in the hypothesis test. The null region had boundaries for the bias at ± 133m and for the slope at ± 133/5 m/s. A series of single satellite failures were added to the measurements and the sensitivity of the test statistic to the different errors was analyzed. For presentation purposes it is convenient to tabulate only the dominant terms from the H_0 and H_1 regions which are the smallest RSWRS from these regions. The approximate decision rule is given below.

$$\text{Choose } D_1 \quad \text{if} \quad Q > \lambda_0$$

where

$$\begin{aligned} Q &= e^{\frac{x}{2}} \\ x &= \text{RSWRS}_{\min H_0} - \text{RSWRS}_{\min H_1} \end{aligned}$$

A reasonable value for $\lambda_0 = 39.5$ and is obtained when n=10, N_0=8, N_1=32, $P(H_1)=10^{-5}$, $P(H_0)=1-P(H_1)$, $C_f/C_m=10^{-4}$. The critical value for x is then 7.3. Two types of test were performed. Single window test results are given in Table 4.14 and multiple window test results are given in Table 4.15.

In exps. 1-3 the bias was placed at the boundary of the null region and also at ± 10% of this value and the action of the test statistic is as anticipated. Exps. 4 and 5 test the extremes of no error and a large bias error and a strong decision is made in both situations. Exp. 10 and 11 test showed how the test statistic grew as the failure size increased. Exp. 12 analyzes the effect on the test statistic as the failure is moved in and out of the null region. All these results are encouraging.

This method still does not address the issues of the alarm rate or the detection probability. Regrettably, the induced horizontal error was not retained in the analysis and the results were not checked for correctness with respect to this parameter.

Table 4.14: Single window simulation results

Exp.#	Error introduced	x from three experiments		
1	133u(t)	-10	10	4
2	146u(t)	54	73	66
3	120u(t)	-52	-18	-28
4	0	-152	-171	-163
5	222u(t)	423	442	436
6	22tu(t)	139	165	156
7	(133+8.9t)u(t)	212	230	224
8	89tu(t-5)	740	800	780
u(t)=unit step function				

Table 4.15: Multiple decision window results

Exp.#	Error	x from three experiments			
	introduced	Window 1	Window 2	Window 3	Window 4
10	8.9tu(t)	-72	57	512	1020
		-96	79	514	962
		-92	62	517	986
11	-5.3tu(t)	-126	11	275	717
		-129	-20	64	384
		-145	-2.9	189	409
12	(200-10t)u(t)	236	-173	-25	146
		251	-130	-29	163
		269	-154	-20	170

The dimensionality of this approach would make it difficult to implement. However it did show that the parallel filter approach can be molded into a sequential hypothesis test based on a finite window of data where the results are related to the error in the horizontal plane.

4.8. A Chi-square Test Using Kalman Filter Residuals

This approach is similar to the one presented above in that the test statistic is a running sum of weighted residuals squared. This is a simplified version where the statistics are accumulated for only one filter based on the hypothesis that no failure is present. Under this assumption the residual vector is unbiased and the squared residual vector normalized by its covariance is a chi-square random variable with n degrees of freedom [12]. Using the innovations properties of the residuals sequence, the sum of a sequence of these chi-square variables is another chi-square variable where the degrees of freedom of the sum is simply the sum of the degrees of freedom from each step. The test has the following null and alternative hypotheses.

H_0 hypothesis : no-failure state, ν_k is a zero mean innovations sequence

H_1 hypothesis : failure state, ν_k is not zero mean

Under these hypotheses the size α test is [24] :

$$\text{Choose } H_1 \text{ if } Q > \lambda_0$$
$$\text{Choose } H_0 \text{ if } Q \leq \lambda_0$$

where

$$Q = \sum_{k=1}^{N} \nu_k^T V_k^{-1} \nu_k$$

$$\begin{aligned} \mathrm{d} &= \text{degrees of freedom of Q} \\ &= \mathrm{nN} \\ \lambda_0 &= \chi^2_{\mathrm{d}}(\alpha) \\ \alpha &= \mathrm{Prob}(\chi^2_{\mathrm{d}} > \lambda_0) \end{aligned}$$

This test is based in the measurement domain and the inference made about the test result is whether or not a satellite failure is present. This is not the parameter of interest to the user since a sizable measurement error may be tolerated in some cases if it does not couple into a horizontal error larger than the protection level. The test designer does have the ability to set the alarm rate to a desirable level. This alarm rate is conditioned on the reference distribution under the H_0 hypothesis which is based on zero mean residual vectors rather than on the horizontal error as defined in section 3.4.4. It is possible to set the alarm rate according to a specified horizontal protection level by simulating an ensemble of single satellite failures along with the usual measurement noise and selective availability. The test size would be set so there was no misses at the desired protection level. This is a non-trivial task since the satellite geometry and the filter parameters have a large influence on the estimation error. Results of such an analysis will be presented in a later section.

4.9. Need for Redundancy in Integrity Monitoring

The effectiveness of the above approach and all the others presented is based upon the amount of redundancy mainly in the measurement information but also to a large degree in the filter model. This redundancy is required so that a measurement error due to faulted satellite will be observable in the measurement residuals.

The importance of this statement cannot be overstated. Having more independent measurements than unknowns is the simplest source of redundancy. However, the process model of the parameter vector will also provide a certain amount of redundancy. For example, a highly stable clock model will be manifested in small filter gains for the clock states and thus the filter will not allow the clock estimates to be corrupted quickly when a failure appears in the measurements. Also, a filter which assumes a very mild acceleration environment will not allow the state estimates to change too quickly and thus provides a certain amount of redundant information.

The filter designer must be cautious not to over stabilize the filter model. Such a filter would yield biased residuals which will mimic a failure during a period of large unmodelled acceleration. It is not uncommon for crystal oscillators to experience sudden frequency shifts and the state estimates may diverge if the clock model is too solid. These are common filter divergence problems that must be prevented when designing a reliable failure detection algorithm with a small false alarm rate.

4.10. Effects of Poor Subsolution Geometry

As mentioned earlier, there must be more independent measurement equations than there are unknowns for the extra measurements to provide the needed redundancy. This statement is equivalent to the requirement than for n satellites in view, each subsolution of size n-1 must have small GDOP. This is the same criteria used by A. Brown [4] for determining the percent of the time that failure detection is possible for a given satellite constellation.

The reason why detection of an error on certain satellites is not possible will be given in the form of a verbal proof. Start with a situation where there are four measurements and four unknowns. If the measurement equations are linearly independent, the matrix **G** will be non-singular. A unique solution will exist and the residuals will be zero since the solution provides an exact fit the measurements. If one of the measurements contains a large error, this is reflected in a large solution error but the error is not observable in the residuals. When the measurement equations are not linearly independent, the **G** matrix is singular and the GDOP is infinite. As the measurement matrix approaches the singular condition, the solution error gets larger and noisier but the residuals are still zero.

Suppose that more than four satellites are in view. This results in an overdetermined system of equations. If the GDOP is small in each solution of size n-1, then no solution can be found which fits all equations exactly and the result is a set of non-zero residuals. Thus large errors will be observable in large residuals.

Now consider a situation where the all-in-view solution has small GDOP but one of the subsolutions of size n-1 has large GDOP (say that it is infinity). This subsolution contains all the satellites except one and this subset of equations is ill-conditioned, since there are less than four linearly independent equations as evidenced by the large GDOP. If the remaining satellite is added to this subset, then it can add only one independent equation and the result is that the all-in-view solution will contain only four independent measurements. Even though there are more than four equations, the all-in-view solution will provide an exact fit to the measurement which was just added to since it provides a unique solution. An error in this satellite maps directly into the nullity of the solution and will not be observable in the residuals. However, the other satellite measurements are redundant since the GDOP is small when each of these is removed. These measurements do not map into the nullity of the solution and thus large errors on these satellites are observable in the residuals.

The satellite which, when brought into the solution, removes the ill-conditioning can be thought of as a key satellite since it is needed for navigation purposes to provide a solution with small GDOP. Thus the difference between the navigation and integrity requirements on the satellite geometry do not appear to be vastly different. The satellite for which failure detection is difficult is the satellite which must be in the solution to provide good navigation, whereas each of the other satellites can be removed with causing a damaging increase in the GDOP. The length of time a key satellite is present (starting when the worst subsolution GDOP crosses a certain level and lasting until it relaxes back to the same value) is referred to as a poor subsolution geometry (PSG) window and may last as long as 15-20 minutes with the 24 satellite six-plane configuration, and it occurs six percent of time during one day.

The presence of a key satellite in a solution has different effects on the various integrity monitoring schemes which have been presented. In the residual based approaches, an error on a key satellite is not observable in the all-in-view residuals. This can be seen in Tables 4.11 and 4.12 where the RRS test statistic is small for the experiments where the error was placed in the key satellite. In the maximum separation approach (and the range comparison method) the solution which does not use the key satellite will have a large variance associated with the solution errors and will lead to a high false alarm rate due to even small range errors. This can also be seen in Tables 4.11 and 4.12 in the form of very large solution errors in the subsolutions which do not use the key satellite.

The presence of a key satellite also has a similar effect on the Kalman filter measurement residuals. This can be illustrated by noting the effects of a step function error on the measurement residuals. In Fig. 4.1 the error was placed on a non-key satellite and it can be seen that the error remains observable after initiating the step. In Fig. 4.2 the error was place in a key satellite and the receiver clock was assumed to have good stability. The redundancy provided in the filter model was enough so that the error in the key satellite remained observable for at least a few minutes. Finally, the filter model was altered and a receiver clock with modest stability was assumed. When the error was placed on a key satellite, it only remained observable for a short time as seen in Fig. 4.3. The fact that the error was observable for a very brief time is of no consequence. If the key satellite error had been a slowly increasing function, the error would not have been observable at all in this filter whereas it would have been if the filter had assumed good clock stability. Errors placed on non-key satellites produced residuals like those shown in Fig. 4.1 regardless of the clock model.

4.11. The Detection Filter as a Means for Clock Coasting

If integrity monitoring is to be provided at all times, some provision must be made to detect errors in the key satellites. As seen in the previous section, the assumptions made in the filter model about the clock stability can provide the minimum redundancy to make failures in key satellites observable at least for a short period, possibly a few minutes. However an extremely stable receiver clock would be necessary to provide this type of redundancy for the duration of the poor subsolution geometry which can last 15 to 20 minutes. It appears that some other

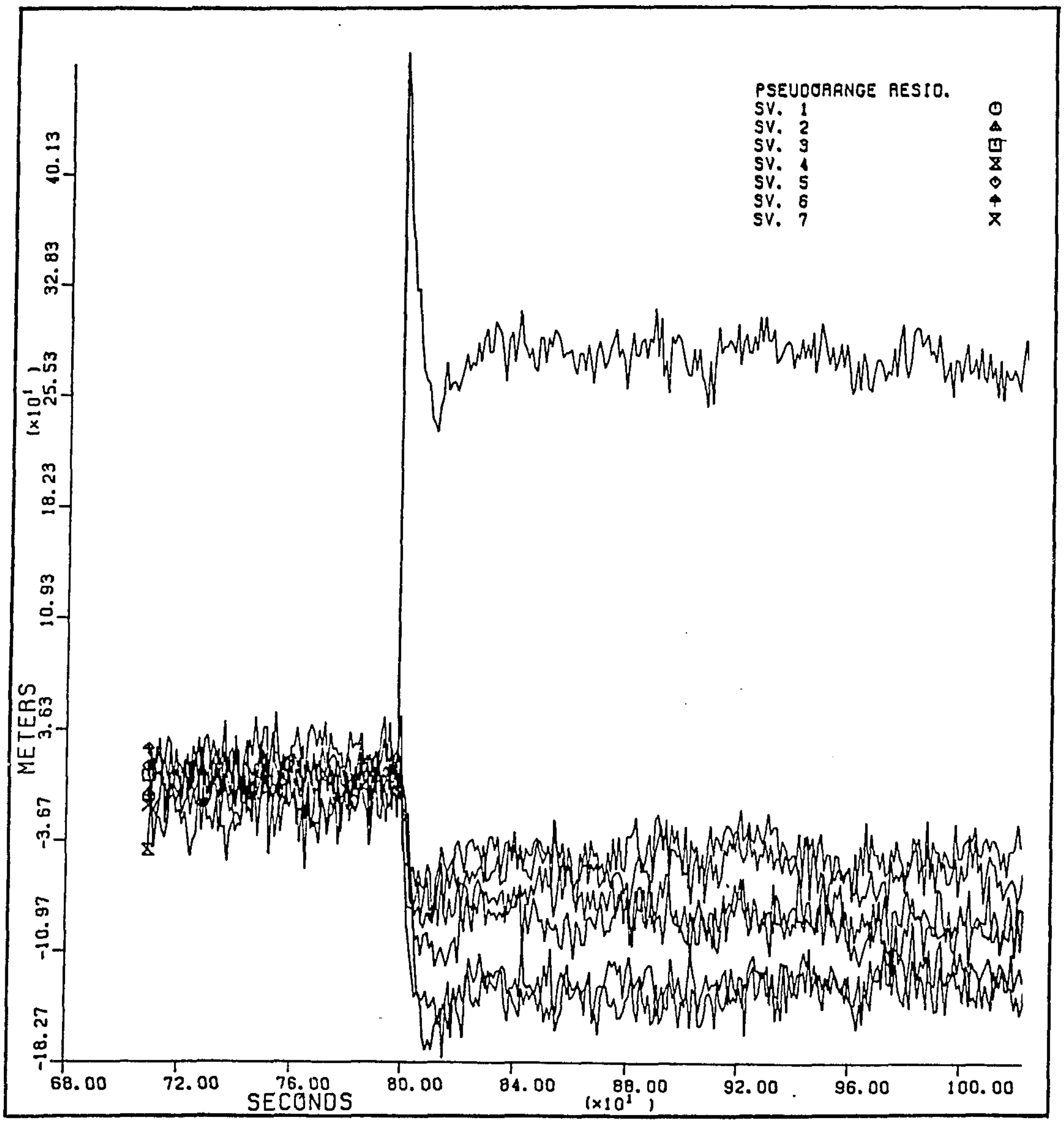


Figure 4.1: Residuals due to step error in non-key satellite

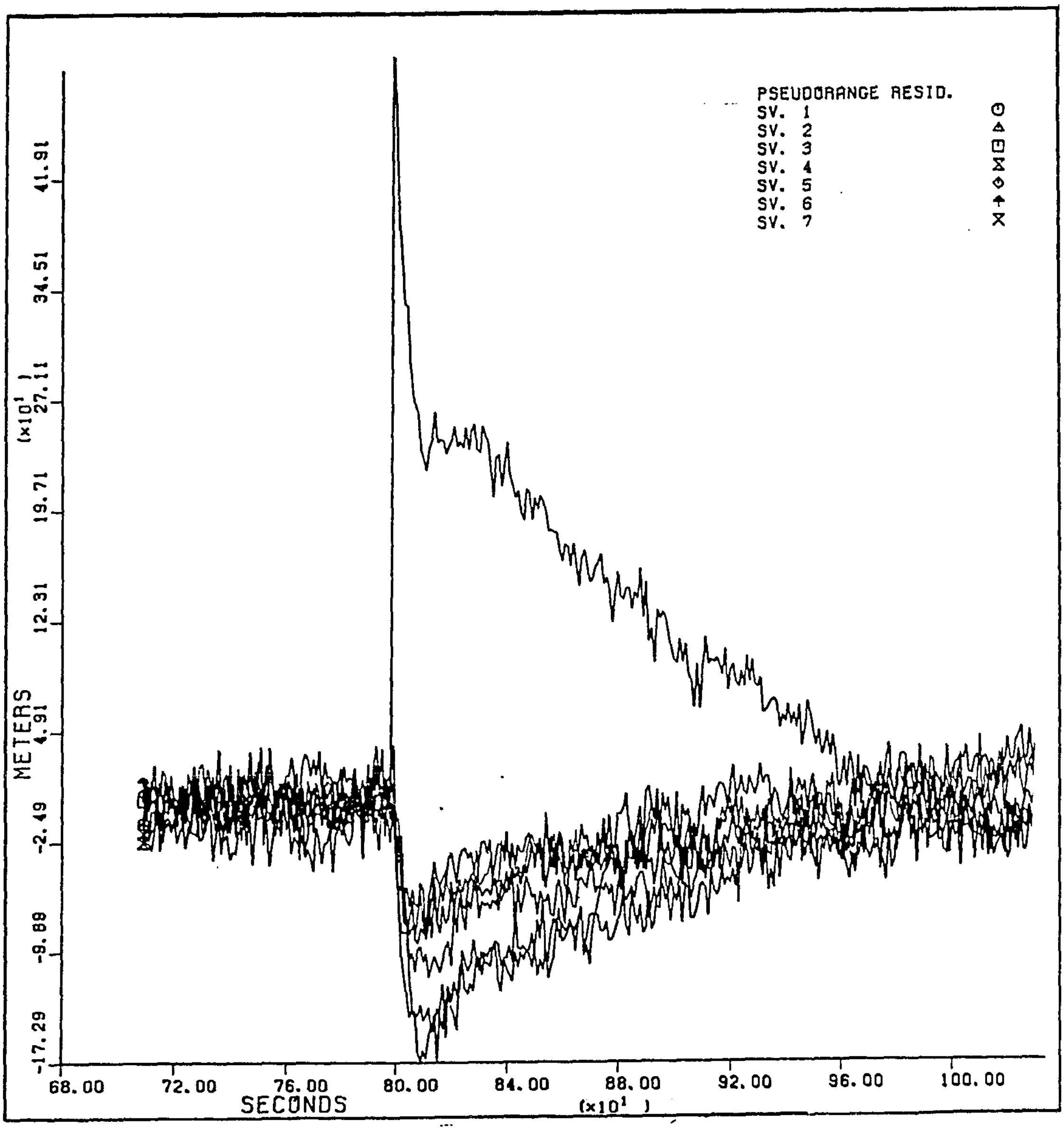


Figure 4.2: Residuals due to step error in key satellite with good receiver clock

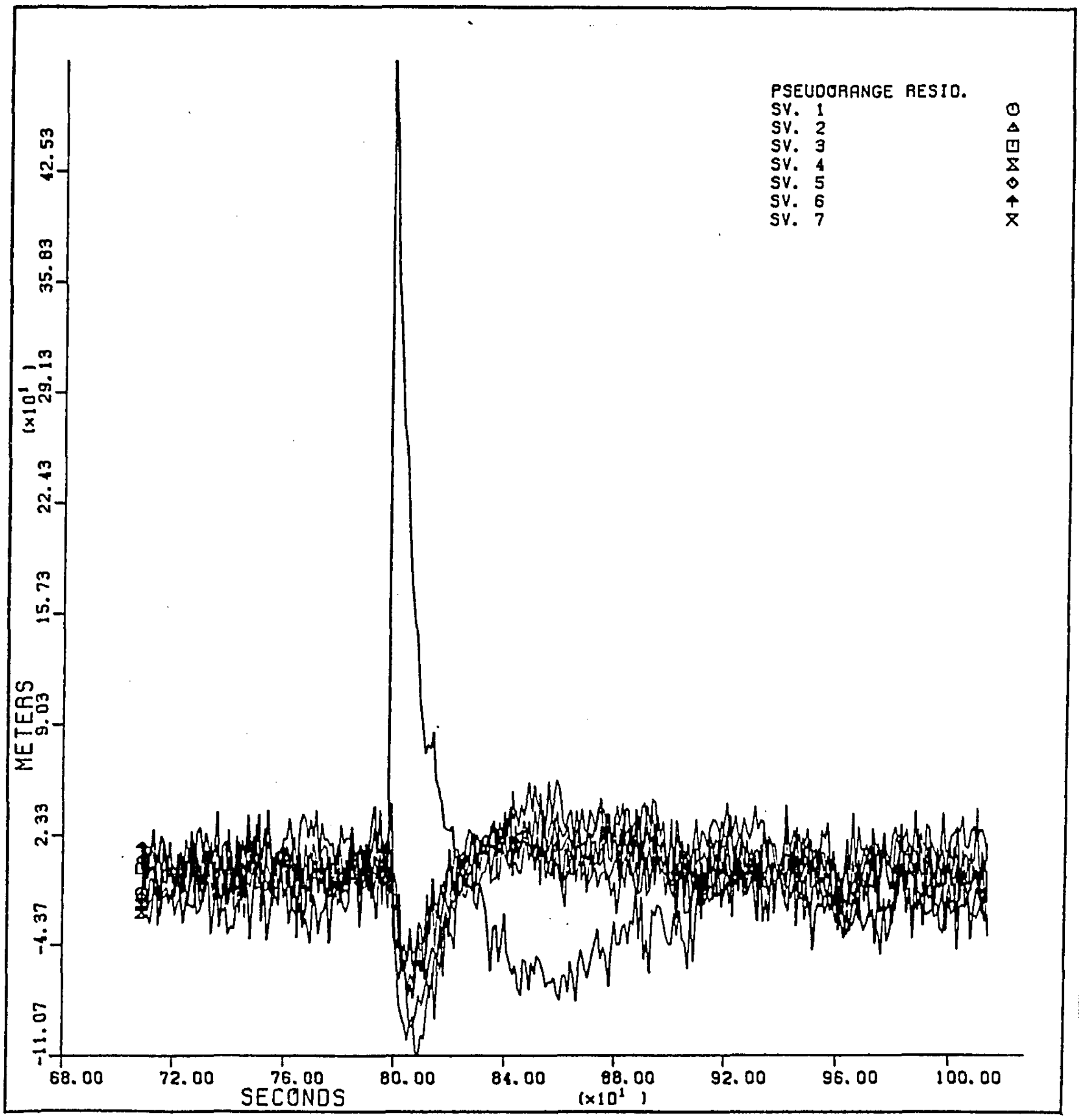


Figure 4.3: Residuals due to step error in key satellite with modest receiver clock

means is required to force the filter to lean on the clock as a stable reference during these periods. One approach which can be taken is to generate a parallel filter that has more clock stability than the navigation filter. This filter is referred to as the detection filter and the additional information can be brought into this filter in either an implicit or an explicit fashion.

The idea of relying on the receiver clock is also common in the navigation context. If only three satellites are available, it is possible to obtain a solution by using the previous estimates of the clock parameters to remove the clock bias from the pseudorange measurements and thus reduce the number of unknowns in the problem to three. This practice is sometimes referred to as clock coasting.

In the context of integrity monitoring, clock coasting can be implemented in at least two different ways as mentioned above. Prior information about the clock stability can be exploited in an implicit fashion by making strong assumptions about the clock stability in the detection filter. This would involve using a different clock model from the one used in the navigation filter and this model would not allow the filter to change the estimate of the clock bias by a large amount over a 15 to 20 minute span. Clearly this would be a suboptimal model but this is acceptable since this filter is not used for navigation purposes.

A filter which changes modes at the start of a PSG window would provide explicit redundancy by coasting the filter on the prior estimates of the clock parameters as mentioned above. This corresponds to having a colored noise process in the measurements where the estimation error of this process is correlated with the estimation error of the parameters to be estimated. Such correlation can be accounted for by using a consider filter [1], [11] where the effects of the colored noise process are considered without actually updating the estimates of this process. The filter equations for the consider filter are given in Appendix B.

Both the implicit and explicit clock coasting schemes were tested by Monte Carlo simulation to analyze their ability to detect slowly varying errors in a key satellite and the results were presented by McBurney and R.G. Brown at the National Meeting of the ION in 1988 [22]. The truth model and the navigation filter both assumed the receiver clock had good stability with parameters described in Table 3.1. The detection filter assumed a very stable clock model where the h_{-2} term is set to zero and the h_0 term was chosen to be the value given in Table 3.1 for the clock with modest stability. This matches the long term stability of this simple model closer to the long term stability of the truth model [29]. The explicit filter assumed the same model used in the implicit mode and only switched into the clock coasting mode when the worst subsolution HDOP > 3.0.

The selective availability process was the same as that described in the simulation of the parallel filter approach except that both filters accounted for the SA noise in a casual manner by increasing the terms in the **R** matrix. The acceleration environment was somewhat noisier than before with $A_1 = 4\text{m}^2/\text{s}^3$. The true white measurement noise had a standard deviation of 10m and the measurement noise variance assumed in the filter was 1200m^2 as a result of the casual SA modelling.

Four different PSG windows in Chicago were analyzed where the worst GDOP among subsolutions peaked as high as 100.0 and the length of time the worst HDOP among subsolutions was larger than 3.0 lasted between 13.5 and 19.5 minutes. The test statistic used was the sum of chi-square statistics from the detection filter residuals over a ten second window. The test size was set conservatively to yield a small alarm rate during the poor subsolution geometry. From the no-failure experiments during these four windows, the largest test statistic for both the implicit and explicit runs occurred at a test size of 5×10^{-6}. If the test size had been set at this value, then one alarm out of 1368 windows would have occurred. Any smaller test size would yield no alarms for this sample space. (At the time this research was performed, the incorrect degrees of freedom was used in setting the threshold. Also, the SA modelling caused the statistics to be over-normalized. The scale factor which properly normalizes the threshold has been determined subsequently, using a statistical procedure referred to as the method of moments, as described in section 5.7. The test size given above and below is compensated for according to the estimated scale factor of $\hat{c} = .33$ and for $\hat{d} = 7.1$.)

Next, a series of ramp failures with slopes of .5, 1.0, and 2.0 m/s were placed in a key satellite at repeated times inside the PSG windows. The maximum moving average horizontal error over two samples from the navigation filter was used as the truth in the simulation. For a test size of 1×10^{-10}, both the implicit and the explicit clock coasting schemes experienced no misses for a protection of 110m. This implies that all errors in the 110 to 300m range were also detected with no misses. It was common in the simulation for the alarm to be raised as many as three to four windows prior to the occurrence of the radial error exceeding the protection level. This means we may be able to detect a satellite failure before it introduces a large error. We are assured that the alarm will be raised when the radial error does exceed the protection level.

This approach is not without fault though. The biggest problem arises in the clock coasting mechanism and how this would be implemented in a real life situation.

The detection filter can be designed independently from the navigation filter but the inference made about the integrity check should be whether the horizontal error in the navigation filter is excessive. If the suboptimality in the detection filter clock model leads to divergence in this filter (because of the highly stabilized clock model), the result will be a false alarm with reference to the navigation filter, and clearly this situation must be avoided. It appears that most of the time the detection filter would not be required since the abundance of satellites in the 24 satellite configuration will provide enough redundancy almost 95 percent of the time. In this case, the test statistic can be formed using the navigation filter residuals and the correlation between the test and navigation error is improved.

The times when the detection filter is required is predictable in the GPS receiver as a result of the subsolution GDOP calculations. As a poor subsolution geometry window is entered, the receiver may start up the parallel detection filter which has the redundancy to detect an error on the key satellite. (Errors on non-key satellites can be detected at all times without the need for the detection filter.) This mode switching may be preferable to a detection filter which is in continuous operation since the detection filter may be prone to divergence more than the navigation filter. As a last resort, other aiding information can be exploited during these times if clock coasting is not possible.

4.12. Retrospect on the Alarm Rate

After performing a number of simulations when a failure has been added to the measurement errors, it is possible to have a pessimistic view of the unconditional alarm rate which is calculated from such a simulation. In the Kalman filter simulations, as the ramp error built up to value which triggered the alarm, the induced horizontal error was still small with respect to a protection level, of say 200m, for possibly a few windows. Thus many false alarms were experienced with regard to the horizontal protection criteria. In the snapshot simulations, a large bias was needed to induce a significant number of horizontal errors which were larger than a specified protection level. However, in this case the alarm was triggered on almost every experiment. This gives the analyst the impression that the false alarm probability is too large since in a major portion of these experiments the horizontal protection was not exceeded. A closer look at the unconditional alarm probability will show that this behavior will not be problematic operationally.

Using the law of total probability, the unconditional alarm probability can be expressed in terms of false alarm probability and the detection probability as shown below.

$$\begin{aligned}\text{Prob(alarm)} &= \text{Prob(alarm}|\text{no-failure)} \times \text{Prob(no-failure)} \\ &\quad + \text{Prob(alarm}|\text{failure)} \times \text{Prob(failure)}\end{aligned} \tag{4.9}$$

In the analysis of the detection schemes, most of the simulations were performed with an intentional bias (or ramp) added to the measurement noise for one satellite. This, of course, is expected to be a rare situation in real life, but the simulations had to be done this way in order to test the effectiveness of the detection scheme. The usual situation is the case where there is no bias in the pseudorange measurement. Such cases induce navigation errors outside the specified protection level only rarely, so these simulations were only run to set the threshold in accordance with a specified false alarm rate. It can be seen from Eq. 4.9 that the unconditional alarm probability is approximately equal to the false alarm probability if the probability of failure is very small, say of the order of 10^{-9} which is expected in the GPS case.

With this perspective, it can be seen that the high false alarm rate obtained from simulations of a failure situation is tolerable since these events are not representative of the "typical" situation. It is comforting to know that the test is sensitive to the presence of such a signal and that it will usually be detected before it induces a large horizontal error. The more dangerous situation is a missed detection so we would like to have a test which is sensitive to the presence of an unusual measurement error. Thus, if the false alarm rate can be set to an acceptably small level, we are assured that the unconditional alarm rate cannot be much larger than this value.

5. THE CENSORED KALMAN FILTER AS A MEANS FOR RAIM

5.1. Overview and Assumptions

A qualitative description of the censored Kalman filter as a means for integrity monitoring was discussed in Chapter 2. A more detailed analysis will now be presented. The censored Kalman filter is an algorithm for computing the residuals and the corresponding state estimates from the usual Kalman filter estimates when a deterministic measurement sequence is removed from the original measurement sequence. The integrity check which accompanies this algorithm is a two stage test. A statistical analysis is first performed on a set of measurement residuals to test the null hypothesis that the residuals are samples of an innovations process. If the test supports the alternative hypothesis that the residuals are biased, then one of the measurement sources is censored in a manner which attempts to force the residuals to support the null hypothesis.

The second step of the test is to compute the censored state estimates based on the censored residuals. If the distance between the censored and the un-censored state estimates is large with respect to the covariance properties of these parameters, the inference is that the estimation error associated with the original state estimate is out of specification and an alarm situation is present. The scheme provides for identification of the faulty measurement source and the removal of this source should return the system to the normal no-failure state.

There are a few underlying assumptions upon which this scheme is based. First of all, redundant measurement information is required so that system failures will be observable in the measurements residuals. Also, the redundancy must appear in a manner so that a good solution exists for each subset of measurements of size one less than the number of sources. This requires the dilution of precision parameters for all such subsolutions to be small and homogeneous. A large measurement error may be tolerated for a brief period of time if it does not lead to a censored state estimate which is "far" from the original estimate. Thus, pre-detection of a possibly faulty measurement source is provided for, but it is not considered to have failed unless it has a significant effect on the distance between the two estimates.

A single failure assumption is made when determining which measurement will be censored. If the statistics of the errant measurement source are significantly larger than the statistics of the other sources, the censored filter will effectively "zero-out" this measurement source and the censored estimate will stay close to the truth while the original estimate will diverge. In this case a strong decision will be made. Conversely, if the measurement errors on many sources are larger than is hypothesized in the filter, the failure will be hidden among the other errors and the censoring performance is weakened. Even if the wrong source is censored, the resulting censored estimate should still be "far" from the un-censored estimate. Thus, large error immersed in large unmodelled noise may still be detected even though the censored estimate will not track the truth as well as in the first case. A missed identification may occur in this situation even though failure detection is still provided. Further testing may be required to double-check which source has failed when the subsolution dilution of precision parameters are somewhat non-homogenous.

5.2. The Two Confidence Region Overlap Test

The primary concern of integrity monitoring is to maintain a bound on the estimation error in the horizontal plane due to an out-of-tolerance satellite signal. The previous integrity monitoring schemes, which are based on the statistics of the measurement residuals, suffer from providing only a weak inference about the horizontal estimation error. The two confidence region overlap test, however, places the integrity assessment directly in the horizontal plane and provides an analytic bound on the protection which is provided by the test.

The confidence region is an interval estimate, with a specified confidence, of the true parameter. The confidence region is centered at the estimate of this parameter and provides an understanding of the how large the estimation error can be, for the specified confidence, when the estimate is unbiased. However, if a failure is present the estimate and the resulting confidence region will be biased. The induced bias in the estimate is unknown but should be reflected in large measurement residuals (if the redundancy requirements are met). If the bias in the residuals can be removed by censoring unusual measurement residuals, then the corresponding censored state estimate should move away from the un-censored estimate.

The distance between the censored and un-censored estimates should provide an indication of the estimation error in the un-censored estimate. An upper bound on the estimation error can be generated where the confidence regions overlap at only one point and the inference made when the confidence regions do not overlap is that the estimation error may be as large as this bound. The test is designed so that the confidence regions will become disjoint before the estimation error exceeds this value.

This failure detection scheme is similar to the one presented by Kerr [16] which also uses a two confidence region overlap test. His approach is different in that one of the confidence regions is centered about the prior mean, and this parameter as well as the confidence region is based on the propagation of prior information rather than on the measurements. The other confidence region is centered at the Kalman filter estimate which utilizes the measurement information and uses the posterior error covariance matrix to form the confidence region. Thus the two ellipsoids are of different sizes and the overlap test is somewhat complicated. In the censored approach, both estimates have the same covariance properties (under different assumptions about the residuals), and as a result the overlap test is simplified.

The confidence region is based on the solid ellipsoidal hypervolume of $\mathbf{x}$ values satisfying the inequality given below, and this region contains probability 1-α [12].

$$(\mathbf{x}-\mu)^{\mathrm{T}}\mathbf{P}^{-1}(\mathbf{x}-\mu) \leq \chi^2_{\mathrm{d}}(\alpha)$$

where

$$\begin{aligned}
\mu &= \mathrm{E}[\mathbf{x}] \\
\mathbf{P} &= \mathrm{Var}[\mathbf{x}] \\
\alpha &= \mathrm{Prob}[\chi^2_{\mathrm{d}} > \chi^2_{\mathrm{d}}(\alpha)] \\
\mathrm{d} &= \text{degrees of freedom of } \chi^2
\end{aligned}$$

This fact may be used to construct a 100(1-α)% confidence region about the estimate of μ. The confidence region has the interpretation that if the test were repeated a number of times, the true parameter μ would be contained in 100(1-α)% of the confidence regions centered about each estimate of μ. In the bivariate case, the confidence region contains the values of x,y satisfying the inequality in Eq. 5.1.

$$\frac{1}{1-\rho^2}\left[\left(\frac{\mathrm{x}-\hat{\mathrm{x}}}{\sigma_{\mathrm{x}}}\right)^2 - 2\rho\left(\frac{\mathrm{x}-\hat{\mathrm{x}}}{\sigma_{\mathrm{x}}}\right)\left(\frac{\mathrm{y}-\hat{\mathrm{y}}}{\sigma_{\mathrm{y}}}\right) + \left(\frac{\mathrm{y}-\hat{\mathrm{y}}}{\sigma_{\mathrm{y}}}\right)^2\right] \leq \chi_2^2(\alpha) \qquad (5.1)$$

where

$$\begin{aligned} \sigma_{\mathrm{x}}^2 &= \mathbf{P}_{11} \\ \sigma_{\mathrm{y}}^2 &= \mathbf{P}_{22} \\ \rho &= \frac{\mathbf{P}_{12}}{\sigma_{\mathrm{x}}\sigma_{\mathrm{y}}} \end{aligned}$$

(The subscript refers to the element in the matrix or vector and the time subscript is removed to improve the readability.)

The confidence regions associated with two estimates of a bivariate random variable, each having the same covariance properties, will overlap if the midpoint between the estimates is contained in both confidence regions. Thus, the confidence regions about the censored estimate and the un-censored estimate will overlap if the inequality in Eq. 5.2 is satisfied. The parameters are obtained from the 8-state filter model described in Section 3.3.

$$\frac{1}{1-\rho^2}\left[\left(\frac{\hat{\mathrm{x}}-\hat{\mathrm{x}}^{\mathrm{c}}}{2\sigma_{\mathrm{x}}}\right)^2 - 2\rho\left(\frac{\hat{\mathrm{x}}-\hat{\mathrm{x}}^{\mathrm{c}}}{2\sigma_{\mathrm{x}}}\right)\left(\frac{\hat{\mathrm{y}}-\hat{\mathrm{y}}^{\mathrm{c}}}{2\sigma_{\mathrm{y}}}\right) + \left(\frac{\hat{\mathrm{y}}-\hat{\mathrm{y}}^{\mathrm{c}}}{2\sigma_{\mathrm{y}}}\right)^2\right] \leq \chi_2^2(\alpha) \qquad (5.2)$$

where

$$\begin{aligned} \mathrm{x} &= \text{north position error} \\ \mathrm{y} &= \text{west position error} \\ \hat{\mathrm{x}} &= (\hat{\mathbf{x}})_1 \end{aligned}$$

$$\begin{aligned}
\hat{y} &= (\hat{x})_3 \\
\hat{x}^c &= (\hat{x}^c)_1 \\
\hat{y}^c &= (\hat{x}^c)_3 \\
\sigma_x^2 &= \mathbf{P}_{11} \\
\sigma_y^2 &= \mathbf{P}_{33} \\
\rho &= \frac{\mathbf{P}_{13}}{\sigma_x \sigma_y}
\end{aligned}$$

For this test it is possible to obtain bounds on the estimation error where the confidence regions contain only one common point. The worst situation is when the major axes of the two ellipses are co-linear. As shown in Fig. 5.1, if a circle is drawn about each horizontal estimate whose radius, a, is half the major axis length, then the true position can be 3a units (at p) from the Kalman filter estimate while there is still overlap on the boundaries of the ellipses. This corresponds to a situation where σ_x and σ_y are replaced with $\sigma_m = \max\{ \sigma_x, \sigma_y \}$. If ρ is also set to zero, the single point overlap occurs when Eq. 5.3 is satisfied.

$$\frac{a^2}{\sigma_m^2} = \chi_2^2(\alpha) \tag{5.3}$$

The protection level at p is given in Eq. 5.4.

$$r_0 = 3\sigma_m \sqrt{\chi_2^2(\alpha)} \tag{5.4}$$

A similar argument can be made to obtain the protection level when the minor axes of the ellipses are co-linear. Using the smallest of the position variances $\sigma_s = \min\{ \sigma_x, \sigma_y \}$ and ρ=0, the protection level in this case is given in Eq. 5.5.

$$r_0 = 3\sigma_s \sqrt{\chi_2^2(\alpha)} \tag{5.5}$$

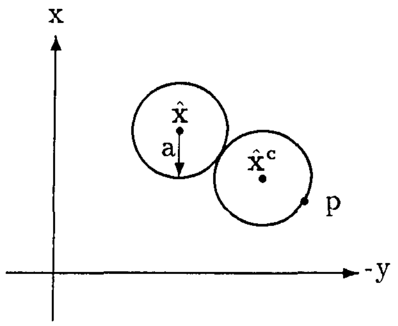


Figure 5.1: Intersection of confidence regions at one point

This represents the most optimistic protection level whereas the previous formula represents the most pessimistic one. In other words, we may be able to protect against errors as small as $3\sigma_s\sqrt{\chi_2^2(\alpha)}$, but we are assured we will detect errors larger than $3\sigma_m\sqrt{\chi_2^2(\alpha)}$ (with a specified confidence level). For a given test size, the protection level is dependent on the accuracy of the navigation solution through σ_x and σ_y . If the accuracy in either direction is the same, these two protection levels are equal. As an example of the protection verses the position accuracy, let α=.001, ($\chi_2^2(\alpha)$=13.8) and as a result, r_0=11.1σ.

5.3. The Censoring Algorithm

The objective of the censoring algorithm is to isolate unusual measurement residuals with respect to the parameters of their distribution computed in the Kalman filter. The censoring algorithm which is presented is not unique or optimal and one could perform the censoring in many different ways. Rather, it is an ad hoc algorithm which was developed in response to analysis of the effects of a single failure in a sequence of vector measurement residuals.

Assume that N measurement residual vectors from the un-censored filter have been stored in the matrix form shown in Fig. 5.2.

ν_1	ν_2	$\cdots$	ν_N

Figure 5.2: Residual matrix to be used in censoring algorithm

The residuals which are to be analyzed are the censored residuals and these residuals account for the presence of a bias μ_0 in the state estimate N steps prior to the current time (see Appendix C). Thus at the beginning of the analysis, the corresponding censored residual matrix is calculated from the un-censored residual matrix using Eq. 5.6.

$$\nu_k^c = \nu_k + \mathbf{D}_k \mu_0 \qquad k = 1, N \tag{5.6}$$

The end result of the censoring algorithm is a set of N censoring vectors $\{s_k\}$ which attempt to remove the bias from the censored residual matrix. Under the single failure assumption, only one element in each s_k vector can be non-zero. The identity of the censored measurements is retained in $\{s_k\}$ or it could be saved in a censoring map. This matrix has the same dimension as the residual matrix and its element are unity in the positions where a measurement is censored and zero elsewhere. This map is helpful in locating the faulty measurement source. If all the ones appear in the same row, the identity of the faulty measurement source is obvious. If ones appear in different rows, this may be due to either extreme noise or possibly multiple failures and identification requires additional logic.

A statistical analysis of the censored residual matrix is performed under the H_0' hypothesis that the residuals are samples of an innovations process. A quadratic statistic referred to as $(X_1)_j$ for the jth step, normalized by the covariance under H_0', is formed for each of the N residual vectors as shown below.

$$(X_1)_j = (\nu_j^c)^T \mathbf{V}_j^{-1} (\nu_j^c) \qquad j = 1, N$$

Under H_0', this is a chi-square random variable with n degrees of freedom, where n is the dimension of the residual vector. This statistic is compared with the threshold c_1 for a zero mean test size of α_1 as defined below.

$$\begin{aligned} c_1 &= \chi_n^2(\alpha_1) \\ \alpha_1 &= \text{Prob}\left[(X_1)_j > c_1\right] \\ n &= \text{degrees of freedom of } (X_1)_j \end{aligned}$$

If each chi-square statistic supports H_0' for test size α_1, then the test of the residuals is complete and the two confidence regions will most surely overlap. Thus the censored state estimates need not be calculated. If one or more of these test statistics do not support H_0', then further analysis is performed. Suppose that the test statistic which exceeds the threshold occurs at step l where $1 \leq l \leq N$. One of the measurement sources at step l may possibly be censored.

The decision as to which source will be censored is made by forming another set of statistics associated with a single measurement source in a time-wise fashion across the censored residual matrix from step l through step N. Each of these statistics is quadratic in form and is normalized by the variance obtained from the corresponding main diagonal elements of $\mathbf{V}_k$. The statistic referred to as $(X_2)_i$ for the ith measurement source is given below. (The subscript on the residual vector and the covariance matrix which is inside the parentheses refers to the time index and the subscript outside the parentheses refers to the element in the vector or matrix.)

$$(X_2)_i = \sum_{j=l}^{N} (\nu_j^c)_i^2 \;/\; (\mathbf{V}_j)_{ii} \qquad i = 1, n$$

One can further reduce the chance of censoring a measurement by requiring the largest of these statistics to exceed a threshold c_2 obtained from the correct reference distribution for a zero mean test of size α_2. If the residuals are zero mean then this statistic is a chi-square random variable with N$-$l+1 degrees of freedom. thus

$$\begin{aligned} c_2 &= \chi^2_{(N-l+1)}(\alpha_2) \\ \alpha_2 &= \text{Prob}\left[(X_2)_i > c_2\right] \\ N - l + 1 &= \text{degrees of freedom of } (X_2)_i \end{aligned}$$

The source which is censored is the one with the largest statistic $(X_2)_i$ which also exceeds c_2. If none of the n statistics is larger than c_2, then no censoring at step l is performed and the analysis moves to the next step after l where $(X_1)_j$ exceeds c_1. If however the largest $(X_2)_m$ exceeds c_2, then measurement source m at step l will be censored and a one would be placed in the censoring map in row m and column l.

At this point the decision has to be made as to how the measurement is to be censored. It was decided to simply remove the linear trend in the residuals from this source and this forces these residuals to be zero mean. This is done by finding a least-squares fit of a ramp function to the censored residual associated with the possibly faulty source from t_l to t_N. The sample which is removed is simply the value of the estimated ramp function at t_l. This sample defines the non-zero element of the s_l vector and this element appears in the mth row. This completes the analysis at t_l.

A new set of censored residuals $\{\nu_k^c\}$ with the s_l vector removed is then computed from step t_{l+1} to step t_N and the chi-square statistics associated with only these steps are re-computed. The same logic is used in determining whether any of the residuals in the range from step t_{l+1} to step t_N will be censored. This process is repeated until no more censoring is required and the result is the set of censoring vectors $\{s_k\}$, k=1,N.

Using μ_0 and $\{s_k\}$, the set of censored filter estimates $\{\hat{x}_k^c\}$ are computed and the two-confidence region overlap tests are performed at each of the N steps. If any test yields no overlap, then the alarm is raised and the censoring map is used to identify the faulty measurement source. At this point μ_0 is re-computed for use in the next window of data and the integrity analysis for the current window of data is complete. In the next window, a new set of N residuals from the uncensored filter are saved and the process is repeated. The test designer would need to supply additional logic to decide whether any faulty measurement source would be removed.

The presence of a non-zero μ_0 vector means that some censoring was performed in a previous window. When the μ_0 vector is non-zero and none of the $(X_1)_j$ statistics exceeds c_1, then it is reasonable to zero out the μ_0 vector for use in the next window. If at least one of the $(X_1)_j$ statistics exceeded c_1, then it is reasonable to continue to update the μ_0 vector even if no censoring was performed on the current window. This logic leads to a minimal amount of censoring in the case that measurement noise is driving the statistics. If however a gradually increasing error is present, the μ_0 vector will keep the censored residuals large by not allowing the censored filter to track this error. In this way, the censored residuals are always larger than the un-censored residuals because they are less contaminated by the unmodelled error source. This is a desirable attribute of the censoring scheme.

The censoring algorithm is best summarized with a sequence of logical steps similar to those which would be used in a computer implementation. The statements are based on a FORTRAN realization in that statements such as GO TO, IF, THEN, ELSE, and ENDIF are used. Also, the same variable may appear on both sides of the equals sign but the value which is stored after the operation is the one on the left-hand side. The following steps describe the censoring algorithm and the resulting hypothesis test in the state space.

```
Given: $\mu_0$, $\{\nu_k\}$, $\{\mathbf{V}_k\}$, k=1,N
Let: $\{s_k\} = \mathbf{0}$

  1. l=1

  2. Compute $\{\nu_k^c\}$,    k=1,N

         where $\nu_k^c = \nu_k + \mathbf{D}_k \mu_0$

  3. Compute $\{(X_l)_j\}$,   $j = 1, N$

         where $(X_l)_j = (\nu_j^c)^T \mathbf{V}_j^{-1} (\nu_j^c)$

  4. IF $\left[(X_l)_j \leq c_l \quad \forall j, \quad j = 1, N\right]$ THEN

         IF (l=1) THEN
```

```
            $\mu_0 = \mathbf{0}$
            GO TO 9
        ELSE IF (l=N) GO TO 9
        ENDIF
    ELSE (* at least one test rejects $H_0'$ *)
        $l = \min\left\{ j \mid (X_1)_j > c_1, \quad j = 1, N \right\}$
    ENDIF
5. Compute $\{(X_2)_i\}, \quad i = 1, n$
        where $(X_2)_i = \sum_{j=1}^{N} (\nu_j^c)_i^2 \; / \; (\mathbf{V}_j)_{ii}$
6. IF $\left[ \max\{(X_2)_i\} < c_2 \quad \forall i, \quad i = 1, n \right]$ THEN
        IF $\left[ (X_1)_j < c_1 \quad \forall j, \quad j = l+1, N \right]$ THEN
            GO TO 9
        ELSE (* at least one test still rejects $H_0'$ *)
            $l = \min\left\{ j \mid (X_1)_j > c_1, \quad j = l+1, N \right\}$
            GO TO 4
        ENDIF
    ELSE (* measurement m at step l will be censored *)
        $m = \left\{ i \mid (X_2)_i = \max\{(X_2)_i\}, \quad i = 1, n \right\}$
        $(s_l)_{ml} = \hat{a}$

                where
                $(\nu_l^c)_m = a$
                $(\nu_{l+1}^c)_m = a + b\Delta t$
                $\vdots$
                $(\nu_N^c)_m = a + b(N-l)\Delta t$

            let
            $\mathbf{y} = \left[(\nu_l^c)_m \ (\nu_{l+1}^c)_m \cdots (\nu_N^c)_m\right]^T$
            $\mathbf{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & \Delta t & \cdots & (N-l)\Delta t \end{bmatrix}^T$
            then
            $\hat{a} = \left[(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\mathbf{y}\right]_1$

    ENDIF

7.  IF (l < N) THEN (* update the residual matrix *)

        $\nu_k^c = \nu_k^c + \mathbf{C}_k^l s_l \quad k = l+1, N$
        l=l+1

    ENDIF

8.  GO TO 3
```

```
9. IF $\left[(\{\mathrm{s_k}\} = \mathbf{0}) \text{ and } (\mu_0 = \mathbf{0})\right]$ THEN

        $\{\hat{\mathrm{x}}_\mathrm{k}^\mathrm{c}\} = \{\hat{\mathrm{x}}_\mathrm{k}\}, \quad \mathrm{k} = 1, \mathrm{N}$

        All tests support $\mathrm{H}_0'$ and $\mathrm{H}_0$.

   ELSE IF $\left[(\{\mathrm{s_k}\} = \mathbf{0}) \text{ and } (\mu_0 \neq \mathbf{0})\right]$ THEN

        $\mu_0 = \mathbf{B}_\mathrm{N}\mu_0$

        $\{\hat{\mathrm{x}}_\mathrm{k}^\mathrm{c}\} = \{\hat{\mathrm{x}}_\mathrm{k}\}, \quad \mathrm{k} = 1, \mathrm{N}$

        All tests support $\mathrm{H}_0'$ and $\mathrm{H}_0$.

   ELSE

        Compute

            $\hat{\mathrm{x}}_\mathrm{k}^\mathrm{c} = \hat{\mathrm{x}}_\mathrm{k} + \sum_{\mathrm{j}=1}^{\mathrm{k}} \mathbf{A}_\mathrm{k}^\mathrm{j}\mathrm{s_j} + \mathbf{B}_\mathrm{k}\mu_0 \qquad \mathrm{k} = 1, \mathrm{N}$

        Perform N two-confidence region overlap tests using Eq. 5.2.

        Compute state bias for next window.

            $\mu_0 = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathbf{A}_\mathrm{N}^\mathrm{j}\mathrm{s_j} + \mathbf{B}_\mathrm{N}\mu_0$

   ENDIF

10. Go to the next window of data.
```

5.4. The Conditional Alarm Rate

The hypothesis test concerning the horizontal error is composed of two separate tests. The first is the zero-mean test of the residuals and the second is the confidence region overlap test in the horizontal plane. It can be shown that the false alarm rate for the combined test is simply the product of the false alarm rates for each test. The exact alarm rate for the zero-mean test of the residuals is based on the censoring algorithm, and this probability turns out to be quite complicated. Thus, only an upper bound will be presented for this parameter. An upper bound will also be given for the probability of false alarm for the overlap test.

The false alarm rate for the combined test is the probability of rejecting the null hypothesis in both tests, when in fact, both null hypotheses are true. Thus, a false alarm is the event where censoring occurs for at least one step inside a window, and the two confidence regions are disjoint for one test, given that the residuals are zero mean and the horizontal protection level has not been exceeded ($r \leq r_0$). Using the definition of conditional probability, the joint probability of rejecting both null hypotheses can be written as the probability that the overlap test fails given that censoring occurs, times the probability that censoring occurs, where both of these probabilities are conditioned on both null hypotheses being true.
Thus,

$$\begin{aligned}
&\text{Prob(false alarm)} \qquad\qquad (5.7)\\
&= \text{Prob(censoring occurs for at least one step and no overlap occurs}\\
&\qquad\qquad \text{given the residuals are zero mean and } r \leq r_0)\\
&= \text{Prob(censoring occurs for at least one step} \mid \text{residuals are zero-mean)}
\end{aligned}$$

$$\times \text{Prob(no overlap occurs} \mid \text{censoring occurs for at least one step}$$
$$\text{and } r \leq r_0)$$

The conditioning on the event that $r \leq r_0$ has been removed from the conditional probability that censoring occurs for at least on step, since the posterior state estimation error is uncorrelated with the current residual (when the residuals are unbiased). The current residual is a linear combination of all the previous residuals, (and the initial condition) so the posterior estimation error is also independent of the complete residual sequence up through the current step. The following derivation shows that the expectation of the posterior estimation error and the current residual is zero, and when Gaussian statistics are assumed, the independence of these random variables follows. As usual, it is assumed that the estimation error is uncorrelated with the measurement noise when the filter model is correct. The Kalman gain $\mathbf{K}_k$ given in Eq. 3.22 is used in the derivation.

$$\begin{aligned}
E\left[e_k \nu_k^T\right] &= E\left[(x_k - \hat{x}_k)(z_k - \hat{z}_k^-)^T\right] \\
&= E\left[\left\{(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) e_k^- - \mathbf{K}_k v_k\right\}\left\{\mathbf{H}_k e_k^- + v_k\right\}^T\right] \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E\left[e_k^-(e_k^-)^T\right] \mathbf{H}_k^T - \mathbf{K}_k E\left[v_k (e_k^-)^T\right] \mathbf{H}_k^T \\
&\quad -(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E\left[e_k^- v_k^T\right] - \mathbf{K}_k E\left[v_k v_k^T\right] \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{K}_k \mathbf{R}_k \\
&= \mathbf{P}_k^- \mathbf{H}_k^T \left[\mathbf{I} - (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T\right. \\
&\quad \left. -(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{R}_k\right] \\
&= \mathbf{P}_k^- \mathbf{H}_k^T \left[\mathbf{I} - (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)\right] \\
&= \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{I} - \mathbf{I}] = \mathbf{0}
\end{aligned}$$

We will now proceed to obtain each of the probabilities given in Eq. 5.7.

The censoring operation is a two step process where the variables $\{X_1\}_j$, j=1,N are first calculated to test whether each residual vector is zero mean. When any of these statistics are found to exceed the significance level c_1, the variables $\{X_2\}_i$, i=1,n are then formed. If the maximum of these statistics exceeds significance level c_2, then censoring occurs. This is a sequential test where the two tests are correlated since they both contain a term which is proportional to the square of a residual from one measurement source. It may be possible to account for this correlation but this would lead to an alarm rate which is related to the satellite geometry through the residual covariance matrix. Thus, an upper bound on the alarm rate which is independent of the geometry may be more appealing than an exact expression. One such bound can be obtained with the following argument.

$$
\begin{aligned}
&\text{Prob}\left[\text{false alarm in censoring test}\right] \\
&\quad = \text{Prob}\Big[\text{at least one censoring during N steps} \\
&\qquad\qquad \text{given the residuals are zero mean}\Big] \\
&\quad = \text{Prob}\Big[\text{statistics are chi-square and at least one } (X_1)_j > c_1 \\
&\qquad\qquad \text{and } \max\{(X_2)_i,\ \ i = 1, n\} > c_2\Big] \\
&\quad = \text{Prob}\left[\max\{(X_2)_i,\ \ i = 1, n\} > c_2 \mid \text{at least one } (X_1)_j > c_1\right] \\
&\qquad\quad \times \text{Prob}\left[\text{ at least one } (X_1)_j > c_1\right] \\
&\quad \leq \text{Prob}\left[\text{ at least one } (X_1)_j > c_1\right] \\
&\quad = 1 - \text{Prob}\left[\text{ no } (X_1)_j > c_1\right] \\
&\quad = 1 - \text{Prob}\left[(X_1)_j \leq c_1,\ \ \forall\ j,\ \ j = 1, N\right]
\end{aligned}
$$

$$= 1 - \text{Prob}\left[(X_1)_j \leq c_1\right]^N$$
$$= 1 - (1 - \alpha_1)^N$$

The upper bound is generated by removing the probability of the event where $\max\{(X_2)_i,\ i = 1, n\} > c_2$ given that one $(X_1)_j > c_1$. As a result, α_2 does not appear in the censoring alarm rate. It may be that this upper bound is close to the true alarm rate since if α_2 is chosen close to α_1, then it is likely that the maximum of $\{X_2\}_i$ will be large (and exceeds c_2) given that one $\{X_1\}_j$ exceeds c_1.

The alarm rate due to the two confidence region overlap test in the horizontal plane is the probability that the confidence regions do not overlap given that the radial error in the horizontal plane is less than the protection level r_0. The test stops the first time the two confidence regions do not overlap so we are only concerned with having a single alarm and the number of steps inside the window does not affect the alarm rate in this test. An upper bound for this alarm rate is obtained with the following argument.

$$\begin{aligned}
\text{Prob(false alarm)} &= \text{Prob}(\text{no overlap at one test} \mid r \leq r_0 \text{ and censoring occurs}) \\
&= \text{Prob}(\text{associated with x,y values outside } CR_1 \\
&\qquad \text{and inside circle of radius } r_0 \text{ about } \hat{x}) \\
&= \alpha_3 - \text{Prob}(\text{associated with x,y values outside} \\
&\qquad \text{circle of radius } r_0 \text{ about } \hat{x}) \\
&< \alpha_3
\end{aligned}$$

where

$$CR_1 = \text{confidence region about } \hat{x}$$

$$\mathrm{CR}_2 \;=\; \text{confidence region about } \hat{\mathrm{x}}^{\mathrm{c}}$$

The censored filter is defined under the hypothesis that the censored state estimate is unbiased and as a result the confidence region about $\hat{\mathrm{x}}^{\mathrm{c}}$ will track the true parameter. Thus when the true parameter is inside CR_1, the confidence regions will overlap. When the true parameter is outside CR_1 but inside the protection circle of radius r_0 about $\hat{\mathrm{x}}$, it is possible that the confidence regions do not overlap since one can shift CR_2 so that the true parameter is inside CR_2 while the confidence regions do not overlap. As soon as the radial error exceeds r_0, the confidence regions will then be disjoint (because of the choice of r_0) with the assumption that CR_2 contains the true parameter. The result is that a false alarm may occur for the set of x,y values outside CR_1 but inside the protection circle of radius r_0 and the probability associated with these point is less than α_3.

The false alarm rate for the combined test of the residuals and the overlap test in the horizontal plane is the product of the false alarm rates for each test and is given in Eq. 5.8 below.

$$\begin{aligned} \mathrm{P}_{\mathrm{fa}} &= \mathrm{Prob}(\text{at least one censoring given residuals are zero mean}) \\ &\quad \times \mathrm{Prob}(\text{no overlap at one test} \mid \mathrm{r} \leq \mathrm{r}_0 \text{ and censoring occurs}) \qquad (5.8) \\ &\leq \left[1 - (1 - \alpha_1)^{\mathrm{N}}\right] \alpha_3 \end{aligned}$$

where

$$\alpha_1 \;=\; \text{test size for zero-mean test of residual vector}$$

$$\alpha_3 \;=\; \text{test size used to set confidence region in horizontal plane}$$

Typical parameters for the GPS integrity problem may be $\mathrm{N} = 5$, $\alpha_1 = \alpha_2 = .00005$, $\alpha_3 = .005$, and as a result $\mathrm{P}_{\mathrm{fa}} = .00000125$. The overall alarm rate for a five hour mission would be .0045 and corresponds to an alarm about every 220 missions. The choice of α_1 is made to minimize the alarms due to selective availability, unmodelled acceleration noise, and sudden frequency shifts of the receiver oscillator. As a result, the satellite error may have to be quite large before censoring begins.

5.5. The Miss Rate

The conditional miss rate is the probability that the two confidence regions overlap given that the radial error has exceeded the protection level. In other words, the un-censored horizontal estimate has diverged from the truth by more than the protection level but the confidence region about the censored estimate is contained inside the protection circle. In this case, neither confidence region contains the true parameter and overlap can occur for all values of the true parameter which are outside of the protection circle. An upper bound on the miss probability can be obtained as follows.

$$\begin{aligned}
\text{Prob(miss)} &= \text{Prob}\,(\text{overlap occurs given } r > r_0) \\
&= \text{Prob}\,(\text{associated with x,y values outside} \\
&\qquad\quad \text{circle of radius } r_0 \text{ about } \hat{x}\ ) \\
&= \alpha_3 - \text{Prob}\,(\text{associated with x,y values outside } CR_1 \text{ and} \\
&\qquad\qquad \text{inside circle of radius } r_0 \text{ about } \hat{x}) \\
&< \alpha_3
\end{aligned}$$

The trade-offs between the alarm rate and protection level behave as one would expect. If an improved protection level is desired (smaller than r_0), then the $X_2(\alpha_3)$ significance level used in Eq. 5.3 must be decreased and this leads to a larger α_3 and thus a larger overall alarm rate and miss rate. Conversely, if one wishes to decrease the alarm rate then either α_1 or α_3 or both may be decreased. Decreasing α_1 will lead to a higher miss rate because the censoring will begin later and the censored estimate will not remove the influence of the early portion of the error which is driving the horizontal error. If α_3 is decreased, then the size of the confidence regions increase and the radial protection is larger.

5.6. Integrity Monitoring Test Design With the Censored Filter

The results of the previous sections can be brought together to generate a design procedure for applying the censored Kalman filter in an integrity monitoring framework. Such a design begins by first specifying the redundancy assumptions upon which this scheme is based. Other aspects related to basic Kalman filter design must also be addressed to satisfy certain notions of filter robustness and also filter stability. The trade-offs between the key parameters such as the protection level and the alarm and miss rates are then discussed.

The effectiveness of the censored filter is based upon the observability of large unmodelled measurement errors in the residuals. Thus, there must always exist enough redundancy in the measurement information so that a good solution can be obtained using subsets of measurements of size one less than the number available. This is equivalent to the requirement that the subsolution dilution of precision parameters are not only small, but also homogeneous. This requirement plays a key role in the identification of the errant source. It was shown in an earlier chapter that an error was not detectable in a measurement source which had poor geometry in the subsolution which did not use this source. If the latter redundancy requirement is not met, there is little hope of detecting errors in such a measurement source.

The scheme used here is based upon batch processing of the measurements where it is assumed that measurements from each source are available (or are saved) so that the measurement vector can be processed in one step. This type of processing preserves the identity of the errant measurement. In the first step when the failure enters the system, the error does not affect the residuals from the other sources. In the following steps, the residuals for all measurements will be affected since the previous residuals get mixed with the current ones, but the residual sequence from the errant source should be the largest. However, if the measurements at the current step are processed sequentially, the order in which the measurements are processed has an effect on the residuals and it is difficult to identify the errant measurement.

A conservative filter design is helpful in keeping the false alarm rate small but a stable filter design is beneficial in identifying the errant source in the presence of a bona fide failure. Thus, one has to trade-off these characteristics depending on which is more important. Increasing the white noise amplitudes in the process model is helpful in preventing filter divergence by decreasing the filter time constants, and in this way the covariance matrix does not tend to get too small as the filter operates for a long period of time. This will prevent false alarms by keeping the residuals small and unbiased in an unmodelled environment, such as a period of higher than normal vehicle acceleration. Colored measurement noise which is not accounted for with an augmented state vector can be partially accounted for with a simple increase in the main diagonal elements of the measurement noise covariance matrix. Thus by keeping the Kalman filter fairly robust or insensitive to unmodelled errors which are not considered to be failures, a reduced false alarm rate can be achieved.

Table 5.1: Summary of censored filter design parameters

protection level	$\mathrm{r}_0 \leq 3\sigma_{\mathrm{m}}\sqrt{\chi_2^2(\alpha_3)}$
false alarm rate	$\mathrm{P}_{\mathrm{fa}} \leq [1-(1-\alpha_1)^{\mathrm{N}}]\alpha_3$
miss rate	$\mathrm{P}_{\mathrm{m}} \leq \alpha_3$
detection probability	$\mathrm{P}_{\mathrm{d}} \geq 1-\alpha_3$
mission alarm rate	$\mathrm{P}_{\mathrm{FA}} \leq 1-(1-\mathrm{P}_{\mathrm{fa}})^{\mathrm{m}}$
where $\sigma_{\mathrm{m}} = \max\{\sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}\}$ from Kalman filter covariance α_1 = zero mean test size of residual vector α_3 = zero mean test size of horizontal error N = number of steps in window m = number of tests performed during mission	

However, if a gradually increasing failure must be detected, then a stable filter (one with large time constants relative to the failure modes) will be helpful in identifying the errant source since the state estimate cannot be pulled away from truth as quickly. In this way, the residuals from the errant source remain the largest for a longer period. If the induced error becomes large during this period, then a correct identification will be made at the time of detection.

With these filter design considerations in mind, a meaningful integrity monitoring design may be developed. A summary of the basic performance measures is presented in Table 5.1. There are at least a few different design procedures which could be followed, with respect to the order in which each of the design parameters is specified. The design steps for one such algorithm are presented below.

1. Specify the window size

   The time-to-alarm specification, $\mathrm{t_a}$, is used to determine N, which is the number of steps inside the window. For the censored filter, $t_a = N\Delta t$, where Δt is the time between updates of the state vector in the filter.

2. Specify the protection level

   The test size α_3 is adjusted to achieve a reasonable compromise between the protection level and the miss rate. A small confidence region (large α_3) will detect early filter divergence and yields a small protection level, but also a large miss rate. A large confidence region (small α_3) is helpful in reducing the number of missed identifications and leads to a small miss rate, but also a large protection level.

3. Specify the alarm rate

   The test size α_1 is adjusted to satisfy the overall false alarm rate specification. However, if α_1 is too small, the censoring algorithm becomes insensitive to early filter divergence. If the Kalman filter time constants are small relative to the failure modes of interest, then the test sizes α_1 and α_2 should be made larger in order to identify the errant measurement source before the filter can re-adjust its state estimate to track the failure. Effective test sizes should be determined (using estimation techniques such as the method of moments described in section 5.7) if filter mismodelling leads to over-normalization of the residual statistics.

5.7. Simulation Results

A limited study of the censored Kalman filter was performed to analyze the effectiveness of the scheme in detecting and identifying a single satellite failure where the all-in-view satellite geometry is the only source of redundancy (besides the memory of the filter). One satellite geometry at Chicago was chosen which appeared to have satisfactory subsolution geometry. The dilution of precision parameters for the all-in-view solution and the subsolutions are given in Table 5.2and are valid over a twenty minute period during which the simulations were performed . A modest quality crystal oscillator was assumed so the results are not dependent on a highly stable frequency reference. The GPS receiver was required to perform the integrity management ten minutes after a "cold-start" where the receiver is turned on and is initialized with rough estimates of position, velocity, clock bias and drift.

Table 5.2: Dilution of precision parameters at t=32130s

Solution	1	2	3	4	5	6	7	AIV
HDOP	1.43	1.32	1.28	1.29	1.22	1.21	1.44	1.12
PDOP	2.22	2.03	1.98	2.12	2.25	1.99	2.76	1.85
TDOP	1.15	0.94	0.94	1.17	1.07	0.95	1.73	0.94
GDOP	2.50	2.24	2.19	2.42	2.49	2.20	3.26	2.08

The following assumptions were made about the receiver error when the receiver was first started. The initial uncertainty (standard deviation) in the horizontal position states was 600m, the uncertainty in the vertical position state was 100m, and the velocity uncertainty in each direction was .1m/s. The initial variance of the receiver clock offset was obtained by assuming the clock offset error to be uniformly distributed between ±1/2 of the CA code length (1ms) and this led to a variance of

Table 5.3: Truth and filter parameters used in four scenarios

	IC # 1	IC # 2	IC # 3	IC # 4
	Low dyn. No SA	High dyn. No SA	Low dyn. SA on	Low dyn. SA on
A_1 m^2/s^3 truth	4.0	4.0	4.0	4.0
A_1 m^2/s^3 filter	4.0	100.0	4.0	100.0
$\mathbf{R}_{ii}$ m^2 truth	100.0	100.0	100.0	100.0
$\mathbf{R}_{ii}$ m^2 filter	100.0	100.0	1200.	1200.

$7.5 \times 10^9 m^2$ in terms or range. The clock drift error which represents the fractional frequency offset was assumed to be one part in 10^8 for a typical crystal oscillator and corresponds to a standard deviation of 3m/s in terms of range drift. The standard deviations were used to scale N(0,1) variates and these scaled random numbers represent the initial estimation error in the state vector when the initial estimates of the system errors are zero. The main diagonal of the error covariance matrix of the Kalman filter was filled with the appropriate variances.

A set of initial condition files was generated, using these statistics as the starting point, to represent four different scenarios with regard to the vehicle dynamics and the presence of selective availability. The filter parameters used in the four situations are given in Table 5.3. (Also, see Fig. 3.1 for the definition of A_1.) The statistics of the true selective availability process were the same as those described in Section 4.8. Each of these scenarios was run for ten minutes and generated a file of true errors, the estimates of these errors, and the error covariance matrix. The files served as the starting point for the experiments where a satellite failure was added to one of the range measurements.

A mild ramp failure with a slope of 1m/s was placed on each satellite measurement one at a time for each of the initial condition files. There were seven satellites in view (above a 7.5 degree mask angle) so there were seven experiments for each file. For the files with the high dynamics parameters, the experiments were also run with the vehicle performing a series of turning maneuvers with a bank angle of 45 degrees (1 g) and a 120s turning period. The acceleration model is developed in section 3.4.3. This acceleration is only accounted for in the filter with an increase in the process noise spectral amplitudes in the position states. Thus there were a total of six different scenarios which were investigated and the results are summarized in Tables 5.4- 5.9.

The data in the column labeled as the "max. error in window" are the largest horizontal error from the navigation filter inside the window in which the error was detected. The data referred to as "censored error at det." are the horizontal error in the censored filter at the time of detection. The units of the error is meters. Also listed in each table are the largest and smallest standard deviation in the horizontal plane, the correlation coefficient, and the protection level (r_0) corresponding to these standard deviations and test size α_3.

The test sizes for the zero mean test of the residuals (α_1, α_2) and the confidence region (α_3) were arrived at by observing both no-failure and failure data. The test sizes for the test of the residuals were chosen so that no alarms occurred for no-failure experiments but also, so that the test was sensitive to rejecting the null hypothesis in the failure-added experiments when the residuals began to get unusually large and the induced horizontal error was appreciable. The test size α_3 was chosen so that censoring which was due to the incorrect measurement source did not raise the alarm while the induced horizontal error was still well below the protection level.

Table 5.4: Detection and identification results using IC # 1

$\sigma_{\max} = 6.6$, $\sigma_{\min} = 4.4$, $\rho = -.107$ $43.9 \leq \mathrm{r}_0 \leq 67.5$ $\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61						
Sat. #	Detected at step #	Identi-fication	Test statistic	Max. error in window	Max. error missed	Censored error at det.
1	80	yes	14.61	46.0	42.0	4.16
2	79	yes	11.67	46.4	38.6	0.94
3	80	yes	12.99	40.6	33.9	5.89
4	104	yes	15.69	41.5	55.9	6.58
5	130	yes	12.09	46.3	51.1	9.02
6	110	yes	12.79	35.4	51.1	9.02
7	122	yes	13.52	52.4	53.2	17.1

Table 5.5: Detection and identification results using IC # 2

$\sigma_{\max} = 8.3$, $\sigma_{\min} = 5.3$, $\rho = -.112$ $54.2 \leq \mathrm{r}_0 \leq 85.1$ $\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61						
Sat. #	Detected at step #	Identi-fication	Test statistic	Max. error in window	Max. error missed	Censored error at det.
1	99	yes	13.70	63.1	59.9	6.29
2	93	yes	14.48	42.6	52.3	18.93
3	107	yes	12.98	58.5	53.6	6.54
4	104	yes	14.58	47.4	55.3	8.82
5	174	yes	13.53	72.5	69.7	14.40
6	134	yes	12.89	65.7	60.0	15.83
7	175	yes	12.39	70.3	74.2	32.99

Table 5.6: Detection and identification results using IC # 2, unmodelled acceleration

<table>
<tr><td colspan="7">$\sigma_{\text{max}} = 8.3$, $\sigma_{\text{min}} = 5.3$, $\rho = -.112$
$54.2 \leq \text{r}_0 \leq 85.1$
$\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61</td></tr>
<tr><th>Sat. #</th><th>Detected at step #</th><th>Identi-fication</th><th>Test statistic</th><th>Max. error in window</th><th>Max. error missed</th><th>Censored error at det.</th></tr>
<tr><td>1</td><td>90</td><td>yes</td><td>17.05</td><td>54.5</td><td>64.0</td><td>14.19</td></tr>
<tr><td>2</td><td>70</td><td>yes</td><td>13.27</td><td>26.7</td><td>29.7</td><td>43.13</td></tr>
<tr><td>3</td><td>98</td><td>yes</td><td>14.47</td><td>52.0</td><td>60.4</td><td>26.29</td></tr>
<tr><td>4</td><td>95</td><td>yes</td><td>12.91</td><td>36.5</td><td>44.4</td><td>13.58</td></tr>
<tr><td>5</td><td>180</td><td>yes</td><td>13.40</td><td>75.1</td><td>76.8</td><td>13.04</td></tr>
<tr><td>6</td><td>150</td><td>yes</td><td>13.72</td><td>63.1</td><td>69.5</td><td>6.88</td></tr>
<tr><td>7</td><td>148</td><td>yes</td><td>14.26</td><td>63.4</td><td>61.7</td><td>27.00</td></tr>
</table>

Table 5.7: Detection and identification results using IC # 3

<table>
<tr><td colspan="7">$\sigma_{\text{max}} = 17.9$, $\sigma_{\text{min}} = 11.7$, $\rho = -.092$
$122.1 \leq \text{r}_0 \leq 182.7$
$\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61</td></tr>
<tr><th>Sat. #</th><th>Detected at step #</th><th>Identi-fication</th><th>Test statistic</th><th>Max. error in window</th><th>Max. error missed</th><th>Censored error at det.</th></tr>
<tr><td>1</td><td>306</td><td>yes</td><td>13.01</td><td>152.6</td><td>165.6</td><td>38.1</td></tr>
<tr><td>2</td><td>285</td><td>yes</td><td>12.38</td><td>150.5</td><td>150.7</td><td>17.1</td></tr>
<tr><td>3</td><td>229</td><td>yes</td><td>12.02</td><td>100.2</td><td>108.3</td><td>19.1</td></tr>
<tr><td>4</td><td>326</td><td>yes</td><td>11.81</td><td>130.7</td><td>130.8</td><td>22.7</td></tr>
<tr><td>5</td><td>345</td><td>yes</td><td>11.72</td><td>116.9</td><td>129.4</td><td>8.4</td></tr>
<tr><td>6</td><td>256</td><td>yes</td><td>11.78</td><td>85.9</td><td>112.2</td><td>25.2</td></tr>
<tr><td>7</td><td>398</td><td>no</td><td>12.17</td><td>149.6</td><td>146.0</td><td>236.0</td></tr>
</table>

Table 5.8: Detection and identification results using IC # 4

<table>
<tr><td colspan="7">$\sigma_{\max} = 24.8$, $\sigma_{\min} = 15.8$, $\rho = -.097$
$160.9 \leq r_0 \leq 249.0$
$\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61</td></tr>
<tr><th>Sat. #</th><th>Detected at step #</th><th>Identi-fication</th><th>Test statistic</th><th>Max. error in window</th><th>Max. error missed</th><th>Censored error at det.</th></tr>
<tr><td>1</td><td>297</td><td>yes</td><td>12.35</td><td>136.2</td><td>140.6</td><td>31.8</td></tr>
<tr><td>2</td><td>260</td><td>yes</td><td>13.31</td><td>140.7</td><td>131.9</td><td>26.0</td></tr>
<tr><td>3</td><td>380</td><td>yes</td><td>11.81</td><td>152.9</td><td>148.8</td><td>5.2</td></tr>
<tr><td>4</td><td>365</td><td>yes</td><td>11.72</td><td>155.5</td><td>161.0</td><td>24.3</td></tr>
<tr><td>5</td><td>470</td><td>yes</td><td>11.68</td><td>169.1</td><td>175.0</td><td>5.3</td></tr>
<tr><td>6</td><td>380</td><td>yes</td><td>11.63</td><td>156.6</td><td>156.0</td><td>4.4</td></tr>
<tr><td>7</td><td>505</td><td>no</td><td>11.74</td><td>190.1</td><td>189.9</td><td>280.0</td></tr>
</table>

Table 5.9: Detection and identification results using IC # 4, unmodelled acceleration

<table>
<tr><td colspan="7">$\sigma_{\max} = 24.8$, $\sigma_{\min} = 15.8$, $\rho = -.097$
$160.9 \leq r_0 \leq 249.0$
$\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61</td></tr>
<tr><th>Sat. #</th><th>Detected at step #</th><th>Identi-fication</th><th>Test statistic</th><th>Max. error in window</th><th>Max. error missed</th><th>Censored error at det.</th></tr>
<tr><td>1</td><td>375</td><td>no</td><td>11.85</td><td>163.9</td><td>160.0</td><td>180.7</td></tr>
<tr><td>2</td><td>270</td><td>yes</td><td>16.66</td><td>162.4</td><td>160.8</td><td>36.4</td></tr>
<tr><td>3</td><td>355</td><td>yes</td><td>12.37</td><td>123.2</td><td>153.2</td><td>43.2</td></tr>
<tr><td>4</td><td>355</td><td>yes</td><td>11.70</td><td>161.0</td><td>154.0</td><td>18.3</td></tr>
<tr><td>5</td><td>469</td><td>yes</td><td>12.29</td><td>156.8</td><td>175.9</td><td>11.3</td></tr>
<tr><td>6</td><td>400</td><td>yes</td><td>12.01</td><td>174.9</td><td>180.8</td><td>18.1</td></tr>
<tr><td>7</td><td>393</td><td>yes</td><td>12.05</td><td>141.0</td><td>134.8</td><td>40.6</td></tr>
</table>

This led to a more confident decision as to the failure source. However, if an earlier detection is desired, a larger test size α_3 (smaller $\chi^2_2(\alpha_3)$) will allow for a smaller protection level but a less confident identification.

For the test sizes used in these experiments, an upper bound on the false alarm rate was 7.5×10^{-6} (using Eq. 5.8) for a five second window (five residual vectors). This is a legitimate upper bound when the residual covariance matrix which is used the normalize the residual statistics is "close" to the true covariance matrix, as in the experiments with no selective availability. As a check, a test-of-distribution was performed on the residual statistics from a no-failure experiment. The sum of n independent chi-square variates divided by n will approach the mean of the distribution, which is the degrees of freedom in this case, as n gets large. For n=200 and seven satellites, such a statistic yielded a value of 6.9 which agrees well with the known result.

In the experiments with selective availability, the large measurement error variance (which attempts to account for the presence of SA) led to very small residual statistics in the no-failure experiments. As a result, the residual statistics are fairly insensitive in that large satellite failures are required to generate statistics which do not support the null hypothesis. In this situation, the chi-square statistics are over-normalized by some scale factor c which is less than unity and thus, the alarm rate is smaller than value mentioned above. The method of moments, a statistical procedure for estimating the parameters of a distribution, was used to estimate the scale factor c. This estimate can be used to calculate an effective test size α'_1 by dividing the original $\chi^2_7(\alpha_1)$ threshold by $\hat{c}$ and then by noting the test size for this new threshold. Estimates of c and d can be found by solving the following equations. (The mean of the chi-square distribution is the degrees of freedom and the variance is twice the degrees of freedom.)

Let

$$\begin{aligned} \mathrm{X_i} &= (\nu_i^c)^T \mathbf{V}_i^{-1} (\nu_i^c) \\ \mathrm{c} &= \text{de-normalizing scale factor} \\ \mathrm{d} &= \text{degrees of freedom of each chi-square } \mathrm{X_i} \\ \mathrm{n} &= \text{number of variates in experiment} \end{aligned}$$

Then

$$\frac{1}{n} \sum_{i=1}^{n} \mathrm{X_i} = \mathrm{c\, d} \tag{5.9}$$

$$\frac{1}{n} \sum_{i=1}^{n} \mathrm{X_i^2} = 2\mathrm{c^2\, d} + (\mathrm{c\, d})^2 \tag{5.10}$$

Using no-failure data from a high dynamics experiment with SA, seven satellites, and n=215, the value for Eq. 5.9 was .9383 and the value for Eq. 5.10 was 1.122. As a result, $\hat{d} = 7.28$ and $\hat{c} = .1288$. Thus the effective threshold with the scale factor c removed would be $\chi_7^2(.0005)/\hat{c} = 26.02/.1288 = 203$. This would lead to an extremely small test size since $\chi_7^2(10^{-6})$ is 40.5. If one set the threshold for the zero mean test of the multivariate residual vector at 5.22 (= 40.5 × .1288), the effective test size would be 10^{-6}. This would generate an overall alarm rate of 1.5×10^{-8} for a five second window. Thus, the window alarm rate for the SA experiments presented is less than 1.5×10^{-8}.

The approach of increasing the measurement noise variance to account for SA also caused problems with regard to the censoring algorithm identifying the errant measurement. In some experiments it turned out that the largest chi-square statistic for each satellite, at the time of detection, did not correspond to the errant satellite. In these cases, a wrong measurement would be censored and the censored state estimate would diverge from the truth. Failure detection is still provided but a missed identification occurs. This behavior is due mainly to the length of the filter time constants and to the non-homogeneity of the dilution of precision parameters.

Since the clock bias error appears along the line-of-sight of the range vector, the TDOP parameter plays a key role in understanding the relative bias of each measurement residual when a failure is present. If a gradual ramp error is placed in a satellite which has large subsolution TDOP relative to at least one other subsolution, then, depending on the stability of the clock error model, at some point the residual bias associated with this satellite will become small. This is because a filter with short time constants can adjust the state vector estimate to track the error in a few tens of seconds and thus remove the bias in this measurement. (This type of response is shown in Fig. 4.3.) However, the residuals from satellites with small subsolution TDOP relative to largest TDOP may still remain large and the wrong source is censored. A filter with a high degree of stability in the clock model will not allow the state estimates to change as quickly and the errant measurement will then have the largest residuals for a longer period of time. The three experiments where a missed identification occurred were re-run assuming a receiver clock with good stability and this provided enough implicit redundancy to allow for correct identification. The results are given in Table 5.10.

As the filter model is made sub-optimal to account for unmodelled acceleration and selective availability, the effects of the subsolution geometry appear to become more pronounced. This was manifested in the simulations as a missed identification.

Table 5.10: Re-run of missed identification experiments with good clock

$\alpha_1 = .0005, \alpha_2 = .003, \alpha_3 = .003$, Threshold= 11.61						
Using IC # 3: $122.1 \leq r_0 \leq 182.7$						
	Detected	Identi-	Test	Max. error	Max. error	Censored
Sat. #	at step #	fication	statistic	in window	missed	error at det.
7	481	yes	11.93	166.2	175.8	82.4
Using IC # 4: $160.9 \leq r_0 \leq 249.0$						
	Detected	Identi-	Test	Max. error	Max. error	Censored
Sat. #	at step #	fication	statistic	in window	missed	error at det.
7	525	yes	11.74	205.9	195.8	94.9
Using IC # 4: Acc. on, $160.9 \leq r_0 \leq 249.0$						
	Detected	Identi-	Test	Max. error	Max. error	Censored
Sat. #	at step #	fication	statistic	in window	missed	error at det.
1	324	yes	14.53	159.3	167.3	20.9

At first glance, the subsolution geometry appeared to be satisfactory since all dilution of precision parameters were small. However, there was almost a two-to-one difference (see Table 5.2) between the TDOP of subsolution 7 and subsolutions 2, 3, and 6. Also, subsolutions 1, 4, and 5 have a TDOP which is almost 20 percent larger than than the TDOP in subsolutions 2, 3, and 6. These slight differences among acceptable small dilution of precision parameters seem to have a dramatic effect on the identification of the errant satellite when the sensitivity of the filter is decreased to account for selective availability.

An obvious solution to this problem is to add more redundancy to the solution in a manner which smooths out the TDOP in each subsolution. This would imply using additional information such as knowledge of very stable clock (as in implicit or explicit clock coasting) or measurements from a baro-altimeter, a Loran receiver or other aiding equipment. The use of this information for integrity purposes is different than when it is used in a hybrid navigation system. It is assumed that the measurements are calibrated externally from the navigation filter and are used in a detection filter only when the GDOP of the subsolutions are non-homogeneous. (These situations are predictable by the receiver.) In this way, no more unknowns are added to the solution when the calibrated measurement is added to the set of sensors. The calibration accuracy will degrade over time while the measurement is being used so it is assumed that the calibration is valid over a long enough period, say 15 minutes, during which the extra redundancy is needed. After this period, the measurement is removed from the detection filter and is re-calibrated until it is needed for again integrity purposes.

In summary, the following characteristics are observed when the censored filter correctly identifies the errant measurement source.

1. The subsolution TDOP of the errant source is not large relative to TDOP of the other subsolutions. The optimality in the filter model or stability of certain state elements will allow for more relative differences among the subsolution TDOP.

2. The measurement source which has the largest residuals is the source which contains the failure.

3. When the bias is removed from the errant source at t_k and is propagated through the rest of the window, the censored residuals at t_k are smaller than before censoring, while the censored residuals after t_k (and thus the chi-square statistics) are larger than before the propagation. Thus removal of the error at the current step makes the error more observable at the next step.

4. Censoring has a "clamping" effect on the censored source where the bias

(rather than the noisy samples) is removed and the censored estimate will track the true state vector.

When the errant source is not correctly identified, the following observations can be made.

1. The TDOP of the subsolution for the errant satellite is larger than at least one TDOP from the other subsolutions.

2. The measurement source with the largest chi-square statistics is not associated with the failure source.

3. The censored estimate diverges from both the truth and the un-censored estimate, but a correct detection is still obtained since the confidence regions will eventually not overlap as the failure persists and increases in magnitude.

4. If the wrong source is censored, the censored statistics $\{X_1\}_j$ may tend to increase to a point where the so-called "improved" residuals will not support the null hypothesis. It may be possible to use additional logic to remove this source from the set of sources which may be censored for a specified number of windows. The source with the next largest statistics may then be censored.

A more complicated approach to choosing which source to censor could be made by first censoring each measurement separately and then choosing the errant source as the one which generated the largest censored statistics on the following steps and the smallest statistics at the current step.

5.8. Summary of Censored Filter Results

The results of the simulations presented in Section 5.7 are summarized in Table 5.11, for experiments with no selective availability, and in Table 5.12, for experiments with selective availability. There were seven satellites in view (for a 7.5 degree mask angle) and a 1m/s ramp failure was placed on each satellite one at a time. Three dynamic environments were considered so there was a total of 21 experiments for both cases of no SA and full SA. The subsolution geometry was considered to be good since each subsolution GDOP was less than 3.3 (see Table 5.2). A modest quality crystal oscillator was assumed with Allan variance parameters given in Table 3.1.

Table 5.11: Summary of censored filter results with no SA

Vehicle environment	Protection level		Largest radial error	Largest time to detection
	$(r_0)_{\min}$	$(r_0)_{\max}$		
Low dyn.	43.9m	67.5m	55.9m	130s
High dyn.	54.2m	85.1m	74.2m	174s
High dyn. + turns	54.2m	85.1m	76.8m	180s

The key performance measures with no SA were found to be

1. Miss rate: $P_m < .003$

2. Alarm rate: $P_{fa} \leq 7.5 \times 10^{-6}$

3. Mission alarm rate (5 hours): $P_{FA} \leq .027$

4. Perfect failure detection (no misses)

5. No wrong identifications in 21 experiments

Table 5.12: Summary of censored filter results with SA

Vehicle environment	Protection level		Largest radial error	Largest time to detection
	$(r_0)_{min}$	$(r_0)_{max}$		
Low dyn.	122m	183m	166m	398s
High dyn.	161m	249m	190m	505s
High dyn. + turns	161m	249m	181m	469s

The key performance measures with SA were found to be

1. Miss rate: $P_m < .003$

2. Alarm rate: $P_{fa} \leq 1.5 \times 10^{-8}$

3. Mission alarm rate (5 hours): $P_{FA} \leq 5.4 \times 10^{-5}$

4. Perfect failure detection (no misses)

5. Three wrong identifications in 21 experiments

It should be pointed out that these results are not the best that could be achieved. Smaller protection levels with SA could be achieved by decreasing the size of the horizontal confidence regions and also by improving the manner in which the presence of SA is acknowledged in the filter. The method of simply increasing the measurement noise covariance led to a larger standard deviation in the horizontal plane, and this directly increased the protection level.

6. CONCLUSIONS

The purpose of this research was to analyze the redundancy which is intrinsic to the 24 satellite GPS constellation and to develop algorithms which exploit this redundancy in an effort to assure the integrity of GPS for the civil aviation community. Various methods which have been previously presented were analyzed in Chapter 4 and were found to to be effective in detecting the presence of a satellite failure with an acceptable detection probability when the satellite geometry was favorable for detection. However, the main concern in the integrity problem is to maintain a bound of the horizontal position error which is induced by an out-of-specification signal. The tests developed in the previous approaches suffer from providing only a weak inference to this parameter. A new scheme referred to as a censored Kalman filter is developed in Chapter 5 and it places the final integrity check directly in the horizontal plane.

This censored filter scheme has a number of advantages over the other approaches. The scheme is conservative in classifying a signal as a failure since a large measurement error may be tolerated if it does not induce a horizontal error which is excessive. A conservative upper bound on the horizontal protection is presented which is proportional to the accuracy of the navigation filter. An upper bound on the false alarm rate is obtained analytically and allows the alarm rate for an extended mission to be set to a level which is operationally satisfactory. An approximate miss rate is also presented. Identification of the failure source is also provided for as part of the scheme.

The censored Kalman filter was tested in a number of different scenarios with regard to the vehicle dynamics and selective availability (SA) when a gradually increasing satellite failure is present. In an accelerating environment with SA, the scheme was able to protect against a 250m horizontal error with a 6s reaction time. The protection in a more benign flight environment is significantly improved, possibly as small as 100m without SA. Identification of the errant satellite was successful when the satellite geometry yielded homogeneous dilution of precision in all subsolutions and when the time constants of the filter were large relative to the time required to induce a large horizontal error.

It was found that all integrity algorithms studied which use more than the minimum four satellites as the only source of redundancy suffer from an inability to detect large errors on key-satellites during periods of poor detection geometry. There is current interest in an optimized satellite configuration which minimizes the occurrence and duration of these poor detection geometries to less than a few percent of the time in the worst locations. Even if the poor detection geometries can be removed, the probability that one or two satellites will not be operational due to routine servicing (the user segment will be notified of this situation) is not zero, so it is likely that brief periods of poor detection geometry will occur. It appears that these situations will require the use of independent information from outside of the GPS receiver if integrity is to be provided at all times.

7. ACKNOWLEDGEMENTS

I wish to express my appreciation and say thank-you to a few people here at ISU and elsewhere whom have made it possible for me to complete this degree. My major professor Dr. R. Grover Brown should be given most of the credit. It is clear why he has gained such a fine reputation as an educator and researcher. The experience and opportunity he has given to me will be the most valued part of my graduate education. Hopefully, some of his insight has rubbed off onto me and I will be able to share it with others as he has so unselfishly done with me.

I must also say thanks to all the members of my graduate committee: Dr. Larry Coady, Dr. Tryphon Georgiou, Dr. Vijay Vittal, and Dr. Noel Cressie. Dr. Cressie was especially helpful putting some of my statistical techniques on a more-firm footing. A couple of professors from St. Ambrose University: Dr. Thomas Burns and Dr. Kuo-Ho Yang deserve many thanks for motivating me to pursue graduate study. I also wish to express my gratitude to Jim Hoekstra from the ISU Computation Center for the invaluable programming assistance he offered me so that my programs would run with minimum expense and CPU time. My classmate Sankaran Rajagopal was also invaluable in helping me get up to speed with WYLBUR at the time when I was beginning my research.

I was fortunate to receive funding from the U.S. Department of Transportation through the Transportation Systems Center (TSC). Dr. R.M. Kalafus from the TSC provided helpful comments as well as verification of some statistical results and checks of the satellite geometry used in the simulation programs. I wish to thank the Electronics Affiliates Program and the Electrical and Computer Engineering Department for the financial support during my graduate education. I also with to thank the Electrical and Computer Engineering Department Endowment Advisory Board for the financial support which made it possible for me to attend a short course at UCLA which was beneficial to my research effort.

Of course, to my parents I wish to say thanks for the love and support (and CARE packages) that helped me survive eight years of college.

8. BIBLIOGRAPHY

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

[2] Braff, R., Shively, C. A., and Zeltser, M.J. "Radionavigation System Integrity and Reliability." *Proc. of the IEEE*, 71, No. 10 (October 1983): 1214-1223.

[3] Brown, A. K. "Civil Aviation Integrity Requirements for the Global Positioning System." Navigation, J. Inst. Navigation 35, No.1 (Spring 1988): 23-40.

[4] Brown, A. K., and Jessop, R. "Receiver Autonomous Integrity Monitoring Using the 24 Satellite Constellation." *Proceedings of the Satellite Division First Technical Meeting of the Institute of Navigation*, Colorado Springs, Colorado, September 1987.

[5] Brown, R. G. *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley and Sons, New York, 1983.

[6] Brown, R. G., and Hwang, P. C. "GPS Failure Detection By Autonomous Means Within the Cockpit." Navigation, J. Inst. Navigation 33, No.4

(Winter 1986-87): 335-353.

[7] Brown, R. G., and McBurney, P. W. "Self-Contained GPS Integrity Check Using Maximum Solution Separation as the Test Statistic." Navigation, J. Inst. Navigation 35, No.1 (Spring 1988): 41-54.

[8] Denaro, R. P. "NAVSTAR Global Positioning System Offers Unprecedented Navigation Accuracy." *Microwave Systems News*, 14, No. 12 (November, 1984): 54-83.

[9] Hwang, P. Y. C. "GPS Integrity: Failure Detection by Autonomous Means in a Flight Environment." Ph.D. Dissertation, Iowa State University, Ames, Iowa, 1986.

[10] IMSL Users Manual. Stat/Library FORTRAN Subroutines for Statistical Analysis. Vol.3 Version 1.0 (April 1987): 1013-1014.

[11] Jazwinski, A. H. *Stochastic Process and Filtering Theory.* Academic Press, New York, 1970.

[12] Johnson, A. R., and Wichern D. W. *Applied Multivariate Statistical Analysis.* Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[13] Kalafus, R. M. "GPS Failure Detection Study." Project Memorandum, DOT-TSC-FA-560-PM-84-39. U.S. Dept. of Transportation Research and Special Programs Administration, Transportation Systems Center, Cambridge, MA, 1984.

[14] Kalafus, R. M. "Receiver-Autonomous Integrity Monitoring of GPS." Project Memorandum, DOT-TSC-FAA-FA-736-1, Center for Navigation, Transportation Systems Center, Research and Special Programs Administration, Department of Transportation, Cambridge, MA, 1987.

[15] Kalafus, R. M., and Chin, G. Y. "Performance Measures of Receiver-Autonomous GPS Integrity Monitoring." *Proceedings of the Forty-Fourth Annual Meeting of the Institute of Navigation*, Santa Barbara, California, January 1988.

[16] Kerr, T. H. "Statistical Analysis of a Two Ellipsoid Overlap Test for Real Time Failure Detection." *IEEE Trans. Automatic Control*, AC-25, No. 4 (Aug. 1980): 762-773.

[17] Lee, Y. C. "Analysis of Range and Position Comparison Methods as a Means to Provide GPS Integrity in the User Receiver." *Proceedings of the Forty-Second Annual Meeting of the Institute of Navigation*, pp. 1-4, Seattle, Washington, June 24-26, 1986.

[18] Lee, Y. C. (MITRE Corp.), Unpublished report presented at RTCA meeting in Washington, D.C., 1987.

[19] Matchett, G. A. "Stochastic Simulation of GPS Selective Availability Errors." Technical Information Memorandum, TIM No. 4517-1. TASC: The Analytic Sciences Corporation, Reading, MA, June 1985.

[20] McBurney, P. W. "Analysis of a GPS Aided Inertial Navigation System

Using the Delayed State Kalman Filter." M.S. Thesis, Iowa State Univ., 1986.

[21] McBurney, P. W., and Brown, R. G. "Proper Treatment of the Delta-Range Measurement in an Integrated GPS/Inertial System." *Proceedings of the Institute of Navigation Forty-Third Annual Meeting*, Anaheim, California, 1987.

[22] McBurney, P. W., and Brown, R. G. "Receiver Clock Stability: an Important Aid in the GPS Integrity Problem." *Proceedings of the Forty-Fourth Annual Meeting of the Institute of Navigation*, Santa Barbara, California, January 1988.

[23] McBurney, P. W., and Brown, R. G. "Self-Contained GPS Failure Detection: the Kalman Filter Approach." *Proceedings of the Satellite Division First Technical Meeting of the Institute of Navigation*, Colorado Springs, Colorado, September 1987.

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

[25] Parkinson, B. W., and Axelrad, P. "Operational Algorithms for Simplified GPS Integrity Checking." *Proceedings of the Satellite Division First Technical Meeting of the Institute of Navigation*, Colorado Springs, Colorado, September 1987.

[26] Parkinson, B. W., and Axelrad, P. "A Practical Algorithm for Au-

tonomous Integrity Verification Using the Pseudo Range Residual." *Proceedings of the Forty-Fourth Annual Meeting of the Institute of Navigation*, Santa Barbara, California, January 1988.

[27] Parkinson, B. W., and Axelrad, P. "Simplified GPS Integrity Checking with Multiple Satellites." *Proceedings of the National Technical Meeting of the Institute of Navigation*, Dayton, Ohio, June 1987.

[28] Parkinson, B. W., and Gilbert, S. W. "NAVSTAR: Global Positioning System - Ten Years Later." *Proc. of the IEEE* 71, No. 10 (October, 1983): 1177-1186.

[29] Van Dierendonk, A. J., McGraw, J. B., and Brown, R. G. "Relationship Between Allan Variances and Kalman Filter Parameters." *Proceedings of the Sixteenth Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting*, Greenbelt, Maryland, November 1984.

[30] Willsky, A. S. "A Survey of Design Methods for Failure Detection in Dynamic Systems." *Automatica* 12 (1976): 601-611.

9. APPENDIX A. THE DIRECTION COSINES

A summary of the equations used to obtain the direction cosines between the satellite range vector and the vehicle locally level unit vectors is given in this appendix. See [20] for a derivation of these equations. The true vehicle position defines the origin of a locally level earth fixed coordinate frame as defined below.

$$\mathbf{r}_{\mathrm{V}}(\mathrm{x},\mathrm{y},\mathrm{z}) = 0\,\mathbf{i}_{\mathrm{X}} + 0\,\mathbf{i}_{\mathrm{Y}} + \mathrm{R}_{\mathrm{V}}\mathbf{i}_{\mathrm{Z}}$$

where

$$\begin{aligned}
\mathrm{x} &= \text{north direction} \\
\mathrm{y} &= \text{west direction} \\
\mathrm{z} &= \text{upward direction} \\
\mathrm{R}_{\mathrm{V}} &= \text{distance of vehicle from center of earth} \\
\mathbf{i}_{\mathrm{X}}, \mathbf{i}_{\mathrm{Y}}, \mathbf{i}_{\mathrm{Z}} &= \text{unit vectors in x,y,z direction}
\end{aligned}$$

The satellite position in a different earth centered inertial frame (u,v,w) is

$$\mathbf{r}_{\mathrm{S}}(\mathrm{u},\mathrm{v},\mathrm{w}) = 0\,\mathbf{i}_{\mathrm{u}} + 0\,\mathbf{i}_{\mathrm{V}} + \mathrm{R}_{\mathrm{S}}\mathbf{i}_{\mathrm{W}}$$

where

$$\begin{aligned}
\mathrm{R}_{\mathrm{S}} &= \text{distance of satellite from center of earth} \\
\mathbf{i}_{\mathrm{u}}, \mathbf{i}_{\mathrm{V}}, \mathbf{i}_{\mathrm{W}} &= \text{unit vectors in u,v,w direction}
\end{aligned}$$

The satellite position is then transformed into earth centered earth fixed coordinates (ECEF) denoted as (X,Y,Z) as shown below.

$$\mathbf{r}_{\mathrm{S}}(\mathrm{X},\mathrm{Y},\mathrm{Z}) = \mathrm{R}_{\mathrm{S}}\ \cos\theta_{\mathrm{X_W}}\,\mathbf{i}_{\mathrm{X}} + \mathrm{R}_{\mathrm{S}}\ \cos\theta_{\mathrm{Y_W}}\,\mathbf{i}_{\mathrm{Y}} + \mathrm{R}_{\mathrm{S}}\ \cos\theta_{\mathrm{Z_W}}\,\mathbf{i}_{\mathrm{Z}}$$

where

$$\begin{aligned}
\cos\theta_{\mathrm{X_W}} &= \sin\gamma\ \sin\beta \\
\cos\theta_{\mathrm{Y_W}} &= -\cos\gamma\ \sin\alpha\ -\ \sin\gamma\ \cos\alpha\ \cos\beta \\
\cos\theta_{\mathrm{Z_W}} &= -\cos\gamma\ \cos\alpha\ -\ \sin\gamma\ \sin\alpha\ \cos\beta \\
\alpha &= \text{angle of ascending node} \\
&= \text{longitude where orbit crosses the equator} \\
\beta &= \text{inclination angle between orbit and equator} \\
\gamma &= \text{angle of satellite inside orbit plane}
\end{aligned}$$

To convert the satellite coordinates from ECEF to the vehicle (x,y,z) coordinate frame, one must transform a point with (X,Y,Z) coordinates into the (x,y,z) coordinate frame which is located at longitude ϕ and latitude θ. This can be done with a transformation matrix **T** whose elements are the direction cosines between unit vectors in the (X,Y,Z) and the (x,y,z) coordinated frames as shown below.

$$\begin{aligned}
\mathbf{r}_{\mathrm{S}}(\mathrm{x},\mathrm{y},\mathrm{z}) &= \mathbf{T}\,\mathbf{r}_{\mathrm{S}}(\mathrm{X},\mathrm{Y},\mathrm{Z}) \\
&= (\mathrm{r}_{\mathrm{S}})_{\mathrm{X}}\mathbf{i}_{\mathrm{X}} + (\mathrm{r}_{\mathrm{S}})_{\mathrm{y}}\mathbf{i}_{\mathrm{y}} + (\mathrm{r}_{\mathrm{S}})_{\mathrm{z}}\mathbf{i}_{\mathrm{z}}
\end{aligned}$$

where

$$\mathbf{T} = \begin{bmatrix} \cos\theta_{xX} & \cos\theta_{xY} & \cos\theta_{xZ} \\ \cos\theta_{yX} & \cos\theta_{yY} & \cos\theta_{yZ} \\ \cos\theta_{zX} & \cos\theta_{zY} & \cos\theta_{zZ} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & \sin\theta\,\sin\phi & -\sin\theta\,\cos\phi \\ 0 & \cos\phi & \sin\phi \\ \sin\theta & -\cos\theta\,\sin\phi & \cos\theta\,\cos\phi \end{bmatrix}$$

The true distance to the satellite is the magnitude of the difference between the satellite position vector and the vehicle position vector.

$$\rho_{\mathrm{true}} = \sqrt{[(\mathrm{r_s})_\mathrm{x}]^2 + [(\mathrm{r_s})_\mathrm{y})]^2 + [(\mathrm{r_s})_\mathrm{z} - \mathrm{R_v}]^2}$$

The direction cosines between the range vector and the locally level coordinate frame unit vectors are

$$\cos\theta_{x\rho} = \frac{(\mathrm{r_s})_\mathrm{x}}{\rho_{\mathrm{true}}}$$

$$\cos\theta_{y\rho} = \frac{(\mathrm{r_s})_\mathrm{y}}{\rho_{\mathrm{true}}}$$

$$\cos\theta_{x\rho} = \frac{(\mathrm{r_s})_\mathrm{z} - \mathrm{R_v}}{\rho_{\mathrm{true}}}$$

In all the simulations, the coordinates of Chicago are

$$\phi = 272.4 \text{ degrees west longitude}$$

$$\theta = 41.7 \text{ degrees north latitude}$$

$$\mathrm{R_v} = 6378.388 \text{ km}$$

The parameters of the 24 satellite GPS constellation shown in Fig. 1.1 are given in Table 9.1 where $\beta = 55.0$ degrees.

Table 9.1: Location of satellites in 6 plane, 24 satellite constellation at t=0.0s

Ring	Sat. #'s in ring	α	$\gamma(t_0)$ for sats. in ring			
1	1,2,3,4	0.	0.	90.	180.	270.
2	5,6,7,8	60.	15.	105.	195.	285.
3	9,10,12,12	120.	30.	120.	210.	300.
4	13,14,15,16	180.	45.	135.	225.	315.
5	17,18,19,20	240.	60.	150.	240.	330.
6	21,22,23,24	300.	75.	165.	255.	345.

Assuming a circular orbit for the satellites, the angle of each satellite in an orbit plane is simply the initial angle γ plus the angle subtended due to the angular velocity of the satellite relative to the orbit plane. The period of the satellite orbit is exactly one half of a sidereal day.

$$\gamma(t) = \gamma(t_0) + \frac{\pi t}{21600. \times .99726957}$$

The motion of the earth can be accounted for by incrementing the longitude of the vehicle position as given below. The period of the earth rotation is one sidereal day.

$$\phi(t) = \phi(t_0) + \frac{\pi t}{43200. \times .99726957}$$

10. APPENDIX B. THE CONSIDER FILTER

This appendix describes how the correlation between the clock and position estimation error is accounted for when performing explicit clock coasting. In the consider filter [1],[11], one accounts for un-estimated noise parameters by propagating the prior statistics of these parameters into the gain equation and covariance update equation. Thus one considers the effects of these parameters without having to estimate them. The situation at hand is somewhat less general in that we have non-trivial estimates of the mean and covariance of the consider parameters and also the cross-correlation between these parameters and the parameters to be estimated. At t_k we have the prior estimate of the complete system $\hat{\mathbf{X}}_k^-$ which can be partitioned (assuming the process noise covariance matrix and the transition matrix is block diagonal) into $\hat{x}_k^-$ which contains all state variables except the clock states and $\hat{y}_k^-$ which contains the clock states. The complete state vector and covariance matrix are partitioned as shown below where the conditioning is on all measurements prior to t_k.

$$\hat{\mathbf{X}}_k^- = \begin{bmatrix} \hat{x}_k^- \\ \hat{y}_k^- \end{bmatrix} \qquad (\mathbf{P}_X)_k^- = \begin{bmatrix} (\mathbf{P}_x^-)_k & (\mathbf{P}_{xy}^-)_k \\ (\mathbf{P}_{yx}^-)_k & (\mathbf{P}_y^-)_k \end{bmatrix}$$

where

$$\begin{aligned}
(\mathbf{P}_{\mathrm{x}}^{-})_{\mathrm{k}} &= \text{prior error covariance of } \mathbf{x} \text{ at } \mathrm{t}_{\mathrm{k}} \\
(\mathbf{P}_{\mathrm{xy}}^{-})_{\mathrm{k}} &= \text{prior correlation between estimation error of } \mathbf{x} \text{ and } \mathbf{y} \text{ at } \mathrm{t}_{\mathrm{k}} \\
\mathbf{P}_{\mathrm{yx}}^{-} &= (\mathbf{P}_{\mathrm{xy}}^{-})^{\mathrm{T}} \\
(\mathbf{P}_{\mathrm{y}}^{-})_{\mathrm{k}} &= \text{prior error covariance of } \mathrm{y} \text{ at } \mathrm{t}_{\mathrm{k}}
\end{aligned}$$

At t_{k}, stop estimating $\mathbf{y}$ and only update $\mathbf{x}$. The process and measurement equations are

$$\begin{aligned}
\mathbf{x}_{\mathrm{k}+1} &= \mathbf{\Phi}_{\mathrm{k}}\mathbf{x}_{\mathrm{k}} + \mathbf{w}_{\mathrm{k}} \\
\mathbf{y}_{\mathrm{k}+1} &= \mathbf{A}_{\mathrm{k}}\mathbf{y}_{\mathrm{k}} + \mathbf{u}_{\mathrm{k}} \\
\mathbf{z}_{\mathrm{k}} &= \mathbf{H}_{\mathrm{k}}\mathbf{x}_{\mathrm{k}} + \mathbf{J}_{\mathrm{k}}\mathbf{y}_{\mathrm{k}} + \mathbf{v}_{\mathrm{k}}
\end{aligned}$$

The assumptions about the measurement noise are given in Eq. 3.15 and the sequences $\mathbf{w}_{\mathrm{k}}$, $\mathbf{u}_{\mathrm{k}}$, and $\mathbf{v}_{\mathrm{k}}$ assumed to be zero mean and mutually independent. The $\mathbf{Q}_{\mathrm{k}}$ matrix is partitioned to provide $(\mathbf{Q}_{\mathrm{x}})_{\mathrm{k}}$. The $(\mathbf{Q}_{\mathrm{y}})_{\mathrm{k}}$ matrix is either obtained from the partitioning of the $\mathbf{Q}_{\mathrm{k}}$ matrix or from a different model as defined below.

$$\mathrm{E}[\mathbf{u}_{\mathrm{j}}\mathbf{u}_{\mathrm{k}}^{\mathrm{T}}] = \begin{cases} (\mathbf{Q}_{\mathrm{y}})_{\mathrm{k}} & \text{if } \mathrm{j} = \mathrm{k} \\ 0 & \text{if } \mathrm{j} \neq \mathrm{k} \end{cases}$$

The update equations are

$$\begin{aligned}
\hat{\mathbf{x}}_{\mathrm{k}} &= \hat{\mathbf{x}}_{\mathrm{k}}^{-} + \mathbf{K}_{\mathrm{k}}(\mathbf{z}_{\mathrm{k}} - \mathbf{H}_{\mathrm{k}}\hat{\mathbf{x}}_{\mathrm{k}}^{-} - \mathbf{J}_{\mathrm{k}}\hat{\mathbf{y}}_{\mathrm{k}}^{-}) \\
\hat{\mathbf{y}}_{\mathrm{k}} &= \hat{\mathbf{y}}_{\mathrm{k}}^{-}
\end{aligned}$$

The Kalman gain is (leaving out the time subscripts)

$$\begin{aligned}\mathbf{K} &= (\mathbf{P}_{\mathrm{x}}^{-}\mathbf{H}^{\mathrm{T}} + \mathbf{P}_{\mathrm{xy}}^{-}\mathbf{J}^{\mathrm{T}})\mathbf{L}^{-1} \\ \mathbf{L} &= \mathbf{H}\mathbf{P}_{\mathrm{x}}^{-}\mathbf{H}^{\mathrm{T}} + \mathbf{J}\mathbf{P}_{\mathrm{y}}^{-}\mathbf{J}^{\mathrm{T}} + \mathbf{H}\mathbf{P}_{\mathrm{xy}}^{-}\mathbf{J}^{\mathrm{T}} + \mathbf{J}\mathbf{P}_{\mathrm{yx}}^{-}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\end{aligned}$$

The covariance updates are (with the Kalman gain and leaving out the time subscripts)

$$\begin{aligned}\mathbf{P}_{\mathrm{x}} &= (\mathbf{I} - \mathbf{KH})\,\mathbf{P}_{\mathrm{x}}^{-} - \mathbf{KJ}\,\mathbf{P}_{\mathrm{yx}}^{-} \\ \mathbf{P}_{\mathrm{xy}} &= (\mathbf{I} - \mathbf{KH})\,\mathbf{P}_{\mathrm{xy}}^{-} - \mathbf{KJ}\,\mathbf{P}_{\mathrm{y}}^{-} \\ \mathbf{P}_{\mathrm{y}} &= \mathbf{P}_{\mathrm{y}}^{-}\end{aligned}$$

The projection equations are

$$\begin{aligned}\hat{\mathrm{x}}_{k+1} &= \mathbf{\Phi}\hat{\mathbf{x}}_{k} \\ \hat{\mathrm{y}}_{k+1} &= \mathbf{A}\hat{\mathbf{y}}_{k} \\ (\mathbf{P}_{\mathrm{x}}^{-})_{k+1} &= \mathbf{\Phi}_{k}(\mathbf{P}_{\mathrm{x}})_{k}\mathbf{\Phi}_{k}^{\mathrm{T}} + \mathbf{Q}_{\mathrm{x}} \\ (\mathbf{P}_{\mathrm{y}}^{-})_{k+1} &= \mathbf{A}_{k}(\mathbf{P}_{\mathrm{y}})_{k}\mathbf{A}_{k}^{\mathrm{T}} + \mathbf{Q}_{\mathrm{y}} \\ (\mathbf{P}_{\mathrm{xy}}^{-})_{k+1} &= \mathbf{\Phi}_{k}(\mathbf{P}_{\mathrm{xy}})_{k}\mathbf{A}_{k}^{\mathrm{T}}\end{aligned}$$

When clock coasting is completed, the appropriate matrices are reloaded to form the original system matrices $\mathbf{X}$ and $\mathbf{P}_{\mathrm{X}}$.

11. APPENDIX C. DERIVATION OF CENSORED FILTER

The (**A**,**B**,**C**,**D**) matrices provide a means for computing a censored estimate from the usual Kalman filter estimate. These matrices describe the connection of a state vector bias and a deterministic measurement sequence into the censored residuals and state estimates. Since these parameters are assumed to be fixed, the covariance properties of the censored filter are the same as those in the usual Kalman filter. The (**A**,**B**,**C**,**D**) matrices are recursive and are computed from matrices which are part of the usual Kalman filter computations. The derivation of these matrices will now be presented. The process begins at t_0 where the censored filter estimate is equal to the Kalman filter estimate plus a bias μ_0 as shown below. The superscript c will be used to identify the censored filter parameters.

$$\hat{x}_0^c = \hat{x}_0 + \mu_0$$

The censored estimate at the start of the next step is

$$(\hat{x}_1^-)^c = \hat{x}_1^- + \mathbf{\Phi}_0 \mu_0$$

The measurement residual in the censored filter uses the censored estimate to predict the measurement and also allows for the removal of a fixed vector s_k.

$$\nu_1^c \;=\; z_1 - \mathbf{H}_1 (\hat{x}_1^-)^c - s_1$$

$$
\begin{aligned}
&= \mathrm{z}_1 - \mathbf{H}_1(\hat{\mathrm{x}}_1^- + \boldsymbol{\Phi}_0\mu_0) - \mathrm{s}_1 \\
&= \nu_1 - \mathbf{H}_1\boldsymbol{\Phi}_0\mu_0 - \mathrm{s}_1 \\
&= \nu_1 + \mathbf{C}_1^1\mathrm{s}_1 + \mathbf{D}_1\mu_0
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{C}_1^1 &= \text{connection of } \mathrm{s}_1 \text{ in } \nu_1^{\mathrm{c}} \text{ at } \mathrm{t}_1 \\
\mathbf{D}_1 &= \text{connection of } \mu_0 \text{ in } \nu_1^{\mathrm{c}} \text{ at } \mathrm{t}_1
\end{aligned}
$$

The censored state estimate is formed with the prior censored estimate and censored residual.

$$
\begin{aligned}
\hat{\mathrm{x}}_1^{\mathrm{c}} &= (\hat{\mathrm{x}}_1^-)^{\mathrm{c}} + \mathbf{K}_1\nu_1^{\mathrm{c}} \\
&= \hat{\mathrm{x}}_1^- + \boldsymbol{\Phi}_0\mu_0 + \mathbf{K}_1(\nu_1 - \mathbf{H}_1\boldsymbol{\Phi}_0\mu_0 - \mathrm{s}_1) \\
&= \hat{\mathrm{x}}_1^- + \mathbf{K}_1\nu_1 - \mathbf{K}_1\mathrm{s}_1 + (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\boldsymbol{\Phi}_0\mu_0 \\
&= \hat{\mathrm{x}}_1 + \mathbf{A}_1^1\mathrm{s}_1 + \mathbf{B}_1\mu_0
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{A}_1^1 &= \text{connection of } \mathrm{s}_1 \text{ in } \hat{\mathrm{x}}_1^{\mathrm{c}} \text{ at } \mathrm{t}_1 \\
\mathbf{B}_1 &= \text{connection of } \mu_0 \text{ in } \hat{\mathrm{x}}_1^{\mathrm{c}} \text{ at } \mathrm{t}_1
\end{aligned}
$$

This process is repeated at the next step and provides an understanding of the recursive nature of the censored filter matrices. The censored residual at this step is

$$
\nu_2^{\mathrm{c}} = \mathrm{z}_2 - \mathbf{H}_2(\hat{\mathrm{x}}_2^-)^{\mathrm{c}} - \mathrm{s}_2
$$

$$
\begin{aligned}
&= \mathrm{z}_2 - \mathbf{H}_2(\mathbf{\Phi}_1\hat{\mathrm{x}}_1 + \mathbf{\Phi}_1\mathbf{A}_1^1\mathrm{s}_1 + \mathbf{\Phi}_1\mathbf{B}_1\mu_0) - \mathrm{s}_2 \\
&= \mathrm{z}_2 - \mathbf{H}_2\hat{\mathrm{x}}_2^- - \mathbf{H}_2\mathbf{\Phi}_1\mathbf{A}_1^1\mathrm{s}_1 - \mathrm{s}_2 - \mathbf{H}_2\mathbf{\Phi}_1\mathbf{B}_1\mu_0 \\
&= \nu_2 + \mathbf{C}_2^1\mathrm{s}_1 + \mathbf{C}_2^2\mathrm{s}_2 + \mathbf{D}_2\mu_0
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{C}_2^1 &= \text{connection of } \mathrm{s}_1 \text{ in } \nu_2^c \text{ at } \mathrm{t}_2 \\
\mathbf{C}_2^2 &= \text{connection of } \mathrm{s}_2 \text{ in } \nu_2^c \text{ at } \mathrm{t}_2 \\
\mathbf{D}_2 &= \text{connection of } \mu_0 \text{ in } \nu_2^c \text{ at } \mathrm{t}_2
\end{aligned}
$$

The censored state estimate at this step is

$$
\begin{aligned}
\hat{\mathrm{x}}_2^c &= (\hat{\mathrm{x}}_2^-)^c + \mathbf{K}_2\nu_2^c \\
&= (\hat{\mathrm{x}}_2^- + \mathbf{\Phi}_1\mathbf{A}_1^1\mathrm{s}_1 + \mathbf{\Phi}_1\mathbf{B}_1\mu_0) \\
&\quad + \mathbf{K}_2(\mathrm{z}_2 - \mathbf{H}_2\hat{\mathrm{x}}_2^- - \mathbf{H}_2\mathbf{\Phi}_1\mathbf{A}_1^1\mathrm{s}_1 - \mathbf{H}_2\mathbf{\Phi}_1\mathbf{B}_1\mu_0 - \mathrm{s}_2) \\
&= \hat{\mathrm{x}}_2^- + \mathbf{K}_2(\mathrm{z}_2 - \mathbf{H}_2\hat{\mathrm{x}}_2^-) \\
&\quad + (\mathbf{I} - \mathbf{K}_2\mathbf{H}_2)\mathbf{\Phi}_1\mathbf{A}_1^1\mathrm{s}_1 - \mathbf{K}_2\mathrm{s}_2 + (\mathbf{I} - \mathbf{K}_2\mathbf{H}_2)\mathbf{\Phi}_1\mathbf{B}_1\mu_0 \\
&= \hat{\mathrm{x}}_2 + \mathbf{A}_2^1\mathrm{s}_1 + \mathbf{A}_2^2\mathrm{s}_2 + \mathbf{B}_2\mu_0
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{A}_2^1 &= \text{connection of } \mathrm{s}_1 \text{ in } \hat{\mathrm{x}}_2^c \text{ at } \mathrm{t}_2 \\
\mathbf{A}_2^2 &= \text{connection of } \mathrm{s}_2 \text{ in } \hat{\mathrm{x}}_2^c \text{ at } \mathrm{t}_2 \\
\mathbf{B}_1 &= \text{connection of } \mu_0 \text{ in } \hat{\mathrm{x}}_2^c \text{ at } \mathrm{t}_2
\end{aligned}
$$

Using the above equations it is possible to identify the following recursive equations, where k indicates the kth step within a window of size N, and $j = 1, 2, \cdots, k$.

$$\mathbf{A}_k^j = \begin{cases} -\mathbf{K}_k & j = k \\ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{\Phi}_{k-1} \mathbf{A}_{k-1}^j & j < k \end{cases}$$

$$\mathbf{B}_k = \begin{cases} (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \mathbf{\Phi}_0 & k = 1 \\ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{\Phi}_{k-1} \mathbf{B}_{k-1} & k > 1 \end{cases}$$

$$\mathbf{C}_k^j = \begin{cases} -\mathbf{I} & j = k \\ -\mathbf{H}_k \mathbf{\Phi}_{k-1} \mathbf{A}_{k-1}^j & j < k \end{cases}$$

$$\mathbf{D}_k = \begin{cases} -\mathbf{H}_1 \mathbf{\Phi}_0 & k = 1 \\ -\mathbf{H}_k \mathbf{\Phi}_{k-1} \mathbf{B}_{k-1} & k > 1 \end{cases}$$

The censored filter estimates and residuals can be calculated at any step from the state estimates and residuals from the Kalman filter by using the following equations. The censoring is performed with the s_k sequence.

$$\hat{x}_k^c = \hat{x}_k + \sum_{j=1}^{k} \mathbf{A}_k^j s_j + \mathbf{B}_k \mu_0$$

$$\nu_k^c = \nu_k + \sum_{j=1}^{k} \mathbf{C}_k^j s_j + \mathbf{D}_k \mu_0$$

This process makes it possible to obtain a filter estimate and the corresponding residuals which are censored starting from j steps into the past. As j increases, the number of matrices which must be saved grows quickly. It is convenient to limit the process to a reasonable size N. At the end of the Nth step, the initial condition μ_0 is redefined as shown below and the process is then repeated.

$$(\mu_0)\text{new} = \sum_{j=1}^{k} \mathbf{A}_k^j s_j + \mathbf{B}_k \mu_0$$

12. APPENDIX D. COMPUTER SOURCE CODE

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /C1/ SOL,RE,PI
      REAL*8 SOL,RE,PI
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C4/ ALPHA,GZERO,BETA,PHYZ,THETA,RS,HDPLMT
      REAL*8 ALPHA(24),GZERO(24)
      REAL*8 BETA,PHYZ,THETA,RS,HDPLMT
C
      COMMON /C5/ XACTN,XACTD,XETRAJ,PTRAJ,XDTRAJ,PDTRAJ
      REAL*8 XACTN(44,1200),XACTD(8,1200),XETRAJ(8,1200)
      REAL*8 PTRAJ(8,1200),XDTRAJ(8,1200),PDTRAJ(8,1200)
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C11/ SASCFT,DSEED,READIC,ISEED
      REAL*8 SASCFT(4),DSEED,READIC
      INTEGER ISEED
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      COMMON /C14/ PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY,NTURON,NTUROF
      REAL*8 PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY
      INTEGER NTURON,NTUROF
C
C     SOL,RE,PI = SPEED OF LIGHT,MEAN RADIUS OF EARTH,AND PIE
C     H=MEASUREMENT CONNECTION TO STATE VECTOR MATRIX
C     PHI=STATE TRANSITION MATRIX
C     PHITR=TRANSPOSE OF PHI
C     P=ERROR COVARIANCE MATRIX
```

```
C     Q=COVARIANCE STRUCTURE OF THE W VECTOR
C     R=COVARIANCE STRUCTURE OF THE V VECTOR
C     Z=MEASUREMENT VECTOR
C     W=DRIVEN RESPONSE VECTOR
C     V=MEASUREMENT NOISE VECTOR
C     XHAT=KALMAN FILTER STATE VECTOR ESTIMATE
C     XTRUE=TRUE STATE VECTOR WHICH IS GENERATED
C     WNOIS=TIME PROFILE OF THE W INPUT VECTOR, THIS MATRIX IS
C           GENERATED IN SUBROUTINE SETUP.  EACH COLUMN IS THE
C           W VECTOR AT A CERTAIN TIME STEP. THE NUMBER OF COLUMNS
C           OF THIS ARRAY IS NSTEPS.
C     VNOISE=TIME PROFILE OF THE MEASUREMENT NOISE.  THIS ARRAY
C           IS IN THE SAME FORM AS WNOISE.
C     TIME=CURRENT FILTER TIME
C     DELTAT=SAMPLING INTERVAL IN SECONDS
C     NSTEPS=THE NUMBER OF RECURSIVE STEPS TO BE PERFORMED.
C     ALPHA= ANGLE OF ASCENDING NODE FOR A PARTICULAR SATELLITE.
C     GZERO= THE INITIAL ANGLE OF THE SATELLITE IN ITS ORBIT.
C     BETA= INCLINATION ANGLE OF SATELLITE ORBIT WITH RESPECT
C           TO THE EQUATOR. SAME FOR EACH SATELLITE.
C     PHYZ=INITIAL LONGITUDE OF THE VEHICLE
C     THETA=LATITUDE OF THE VEHICLE
C     RS=ORBIT RADIUS OF THE SATELLITE ASSUMING A CIRCULAR ORBIT.
C     XACT=ARRAY WHICH IS USED TO SAVE THE PROFILE OF THE TRUE
C          STATE VECTOR.  IT IS ARRANGED WITH EACH COLUMN
C          CONTAINING THE STATE VECTOR AT A CERTAIN TIME STEP.
C          THE NUMBER OF COLUMNS IS THE NUMBER OF RECURSIVE STEPS.
C     XETRAJ=ESTIMATE STATE VECTOR TRAJECTOR.
C     PTRAJ=THE ELEMENTS OF THE MAIN DIAGONAL OF THE P MATRIX
C          ARE SAVED IN THIS ARRAY.  THE SQUARE ROOTS ARE SAVED.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCC            MAIN PROGRAM             CCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SOL=2.997925D8
      RE=6378.388D3
      PI=3.14159265358793D0
C
```

```
      CALL SATUSE
      CALL SETUP
      NWINDW=0
      NWINC=1
C
C     INITIAL MATRICES & DRIVING FUNCTIONS ARE CALCULATED IN SETUP.
C     THE RECURSIVE KALMAN FILTER IS AS FOLLOWS:
C
      DO 20 K=1,NSTEPS
C
         TIME=DELTAT*(FLOAT(K-1))
         IF (K.GE.NHYPON) NWINDW=NWINDW+1
         CALL DIRCOS(K)
         CALL MEASUR(K)
         CALL UPDATE(K)
         CALL STORE(K)
         IF (NWINDW.GE.NWSIZE) THEN
            CALL HYPOTH(K)
            NWINDW=0
            NWINC=NWINC + 1
         ENDIF
         CALL PROJEC(K)
C
20    CONTINUE
C
C     THE FOLLOWING SUBROUTINE CALLS ARE FOR OUTPUT PURPOSES:
C
C     PLOT CALLS SIMPLOTER
C     OUTXP CAN PUT XHAT AND P DIAGONAL ELEMENTS IN A FILE
C     STEADY GIVES XTRUE,XHAT, AND P AT THE LAST STEP
C     THE EXECUTION OF THESE SUBROUTINES IS CONTROLLED BY THE
C     STATUS OF CERTAIN INPUT VARIABLES.
C
      CALL SIGMA(NSTEPS,DELTAT)
      CALL PLOT
      CALL C2OUT
      CALL OUTXP
      CALL STEADY
C
```

```
      STOP
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                        CCCCC
CCCCC               END OF MAIN PROGRAM                      CCCCC
CCCCC                                                        CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                        CCCCC
CCCCC     THE REST OF THE PROGRAM CONTAINS SUBROUTINES       CCCCC
CCCCC                                                        CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                        CCCCC
CCCCC         THERE ARE THREE SETS OF SUBROUTINES:           CCCCC
CCCCC                  MATRIX SUBROUTINES                    CCCCC
CCCCC               KALMAN FILTER SUBROUTINES                CCCCC
CCCCC                  OUTPUT SUBROUTINES                    CCCCC
CCCCC                                                        CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                        CCCCC
CCCCC         MATRIX SUBROUTINES CONTAINED ARE:              CCCCC
CCCCC                                                        CCCCC
CCCCC        MATRIX MULTIPLY,TRANSPOSE, AND INVERSE          CCCCC
CCCCC     SUM,HALF SUM,DIFFERENCE AND HALF DIFFERENCE        CCCCC
CCCCC               MULTBY,SCALEP,ADDON,SUBOUT               CCCCC
CCCCC                                                        CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE MULT(MR,IRSYM,RESULT,IPTEST,XLEFT,NR,LC,RIGHT,
     1                IRR,NC)
C
C     THIS SUBROUTINE CALCULATES THE PRODUCT OF TWO MATRICES.
C     THE CONTROL VARIABLE IPTEST IS SET AT ONE IN THE CALLING
C     STATEMENT IF THE XLEFT MATRIX IS A SYMETRIC MATRIX.  IN
C     THIS CASE,  WHEN AN ELEMENT IS USED FROM THE LEFT MATRIX,
C     ONLY THE UPPER TRIANGULAR PORTION WILL BE USED. THIS MEANS
C     WHEN A REFERENCE IS MADE TO AN ELEMENT IN THE LOWER TRI-
C     ANGLULAR PART, THE ELEMENT TO BE USED IS FOUND BY LOCATING
C     THE CORRESPONDING ELEMENT IN THE UPPER TRIANGLE. THIS ALLOWS
```

```
C      FOR THE MULTIPLICATION OF THE P MATRIX WHICH MUST REMAIN
C      SYMETRIC. THE P MATRIX ALWAYS APPEARS IN THE LEFT SIDE OF THE
C      MULTIPLY.  A CHECK IS ALSO MADE TO LOCATE ZEROS IN A MATRIX
C      TO AVOID PERFORMING WASTEFULL MULTIPLIES BY ZERO.
C
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER MR,IRSYM,IPTEST,NR,LC,IRR,NC
       REAL*8 RESULT(MR,NC),XLEFT(MR,LC),RIGHT(MR,NC)
       IF (LC .NE. IRR) THEN
          WRITE (6,*) 'MATRIX MULTIPLY ERROR'
          GO TO 40
       ELSE
          DO 10 M=1,NR
             DO 20 N=1,NC
                SUM=0.0D0
                DO 30 J=1,LC
                   IF ((IPTEST.EQ.1).AND.(M.GT.J)) THEN
                   IF ((XLEFT(J,M).NE.0.0).AND.(RIGHT(J,N).NE.0.0))
     1                SUM=SUM+XLEFT(J,M)*RIGHT(J,N)
                   ELSE
                   IF ((XLEFT(M,J).NE.0.0).AND.(RIGHT(J,N).NE.0.0))
     1                SUM=SUM+XLEFT(M,J)*RIGHT(J,N)
                   ENDIF
 30             CONTINUE
                RESULT(M,N)=SUM
 20          CONTINUE
 10       CONTINUE
          IF (IRSYM.EQ.1) THEN
             DO 50 INC=2,NC
                NRSTOP=INC-1
                DO 50 INR=1,NRSTOP
                   AVE=(RESULT(INR,INC)+RESULT(INC,INR))/2.D0
                   RESULT(INR,INC)=AVE
                   RESULT(INC,INR)=AVE
 50          CONTINUE
          ENDIF
 40    CONTINUE
       ENDIF
       RETURN
```

```
      END
C
      SUBROUTINE SCALEP(MR,RESULT,NR,RIGHT)
C
C     THIS SUBROUTINE CALCULATES THE PRODUCT OF TWO MATRICES.
C     THE RESULT IS THE PRIOR LEFT TIMES THE RIGHT.  ONLY THE
C     UPPER TRIANGULAR PORTION IS CALCULATED.
C     WHEN AN ELEMENT IS USED FROM THE LEFT MATRIX,
C     ONLY THE UPPER TRIANGULAR PORTION WILL BE USED. THIS MEANS
C     WHEN A REFERENCE IS MADE TO AN ELEMENT IN THE LOWER TRI-
C     ANGLULAR PART, THE ELEMENT TO BE USED IS FOUND BY LOCATING
C     THE CORRESPONDING ELEMENT IN THE UPPER TRIANGLE. THIS ALLOWS
C     FOR THE MULTIPLICATION OF THE P MATRIX WHICH MUST REMAIN
C     SYMETRIC. THE P MATRIX ALWAYS APPEARS IN THE LEFT SIDE OF THE
C     MULTIPLY.  A CHECK IS ALSO MADE TO LOCATE ZEROS IN A MATRIX
C     TO AVOID PERFORMING WASTEFULL MULTIPLIES BY ZERO.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MR,NR
      REAL*8 RESULT(MR,NR),RIGHT(MR,NR)
      REAL*8 XLEFT(44,44)
      DO 5 J=1,NR
         DO 5 I=1,J
            XLEFT(I,J)=RESULT(I,J)
5     CONTINUE
      DO 10 M=1,NR
         DO 20 N=M,NR
            SUM=0.0D0
            DO 30 J=1,NR
               IF (M.GT.J) THEN
               IF ((XLEFT(J,M).NE.0.0).AND.(RIGHT(J,N).NE.0.0))
     1            SUM=SUM+XLEFT(J,M)*RIGHT(J,N)
               ELSE
               IF ((XLEFT(M,J).NE.0.0).AND.(RIGHT(J,N).NE.0.0))
     1            SUM=SUM+XLEFT(M,J)*RIGHT(J,N)
               ENDIF
30          CONTINUE
            RESULT(M,N)=SUM
20       CONTINUE
```

```
 10    CONTINUE
       RETURN
       END
C
       SUBROUTINE MULTBY(MR,XLEFT,LNR,LNC,RIGHT,NRR,NCR)
C
C      THIS SUBROUTINE CALCULATES THE PRODUCT OF TWO MATRICES.
C      THE RESULT IS THE PRIOR RIGHT SCALED UP BY THE LEFT.
C      A CHECK IS ALSO MADE TO LOCATE ZEROS IN A MATRIX
C      TO AVOID PERFORMING WASTEFULL MULTIPLIES BY ZERO.
C
C
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER MR,NRR,NCR,LNR,LNC
       REAL*8 XLEFT(MR,LNC),RIGHT(MR,NCR)
       REAL*8 RESULT(44,44)
       IF (LNC.NE.NRR) THEN
          WRITE(6,*)'MATRIX MULTIPLY ERROR IN MULTBY'
          GO TO 100
       ELSE
       DO 10 M=1,LNR
          DO 20 N=1,NCR
             SUM=0.0D0
             DO 30 J=1,NRR
                IF ((XLEFT(M,J).NE.0.0).AND.(RIGHT(J,N).NE.0.0))
     1             SUM=SUM+XLEFT(M,J)*RIGHT(J,N)
 30          CONTINUE
             RESULT(M,N)=SUM
 20       CONTINUE
 10    CONTINUE
       DO 40 NC=1,NCR
          DO 40 NR=1,LNR
             RIGHT(NR,NC)=RESULT(NR,NC)
 40    CONTINUE
       ENDIF
 100   CONTINUE
       RETURN
       END
C
```

```
      SUBROUTINE TRANSP(MR,RESULT,NEWR,NEWC,A,NR,NC)
C
C     THE TRANSPOSE OF A MATRIX IS PERFORMED IN THIS
C     SUBROUTINE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 RESULT(MR,NEWC),A(MR,NC)
      DO 10 M=1,NEWR
         DO 10 N=1,NEWC
            RESULT(M,N)=A(N,M)
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE INVERT(MR,IRSYM,A,N,XKEEP)
C
C     THE INVERSE OF A MATRIX IS PERFORMED USING THE
C     SHIPLEY-COLEMAN INVERSION ROUTINE.  IT USES GAUSSIAN
C     ELIMINATION WITH PIVOTING.  THE ROUTINE ONLY WORKS FOR
C     MATRICES WHICH HAVE NON-ZERO MAIN DIAGONAL ELEMENTS AND
C     HALTS EXECUTION IF SUCH A CONDITION IS ENCOUNTERED.
C     FOR THE KALMAN FILTER, THE INVERSE IS OF THE HPH+R MATRIX
C     WHICH ALWAYS HAS A NON-ZERO MAIN DIAGONAL.  HENSE, THE
C     INVERSE ROUTINE WILL ALWAYS WORK FOR THIS CASE.  HOWEVER,
C     THIS ROUTINE WILL NOT WORK FOR A GENERAL NON-SINGULAR MATRIX
C     IF IS HAS A MAIN DIAGONAL ELEMENT WHICH IS ZERO.
C     THE RESULT IS MATRIX A.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,N),XKEEP(MR,N)
      DO 10 NC=1,N
         DO 10 M=1,N
            A(M,NC)=XKEEP(M,NC)
10    CONTINUE
      DO 5 IP=1,N
      AD=A(IP,IP)
      IF (DABS(AD).GT.1.OD-38) GO TO 1
      WRITE (6,*) 'MATRIX HAS A ZERO IN THE MAIN DIAG.'
      STOP
```

```
 1    AD=-1.0D0/AD
      A(IP,IP)=AD
      DO 3 IR=1,N
      IF (IR.EQ.IP) GO TO 3
      AT=AD*A(IR,IP)
      DO 2 IC=1,N
      IF (IC.EQ.IP) GO TO 2
      A(IR,IC)=A(IR,IC)+AT*A(IP,IC)
 2    CONTINUE
 3    CONTINUE
      DO 4 I=1,N
      IF (I.EQ.IP) GO TO 4
      A(I,IP)=AD*A(I,IP)
      A(IP,I)=AD*A(IP,I)
 4    CONTINUE
 5    CONTINUE
      DO 6 IC=1,N
         DO 6 IR=1,N
 6          A(IR,IC)=-A(IR,IC)
      IF (IRSYM.EQ.1) THEN
         DO 50 INC=2,N
            NRSTOP=INC-1
            DO 50 INR=1,NRSTOP
               AVE=(A(INR,INC)+A(INC,INR))/2.D0
               A(INR,INC)=AVE
               A(INC,INR)=AVE
50       CONTINUE
      ENDIF
      RETURN
      END
C
      SUBROUTINE SUMHAF(MR,RESULT,NR,NC,A,B)
C
C     THE UPPER TRIANGULAR PORTION TO TWO MATRICES ARE
C     ADDED AND STORED IN RESULT.  THIS SAVE TIME IN THE
C     COMPUTATION OF THE SYMETRIC P MATRIX IN WHICH ONLY
C     ONE PORTION OF THE MATRIX NEED BE CALCULATED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),B(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,N
            RESULT(M,N)=A(M,N)+B(M,N)
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE SCALE(MR,RESULT,NR,NC,SCALAR)
C
C     THE RESULT IS THE PREVIOUS RESULT WITH EACH ELEMENT
C     MULTIPLIED BY SCALAR.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 RESULT(MR,NC),SCALAR
      DO 10 N=1,NC
         DO 10 M=1,NR
            RESULT(M,N)=RESULT(M,N)*SCALAR
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE SUM(MR,IRSYM,RESULT,NR,NC,A,B)
C
C     THE SUM OF TWO MATRICES IS FOUND AND IS STORED
C     IN THE MATRIX RESULT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),B(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,NR
            RESULT(M,N)=A(M,N)+B(M,N)
10    CONTINUE
      IF (IRSYM.EQ.1) THEN
         DO 50 INC=2,NC
            NRSTOP=INC-1
            DO 50 INR=1,NRSTOP
               AVE=(RESULT(INR,INC)+RESULT(INC,INR))/2.D0
               RESULT(INR,INC)=AVE
               RESULT(INC,INR)=AVE
50         CONTINUE
      ENDIF
      RETURN
      END
C
      SUBROUTINE DIFHAF(MR,RESULT,NR,NC,A,B)
C
C     THE UPPER TRIANGULAR PORTION TO TWO MATRICES ARE
C     SUBTRACTED AND STORED IN RESULT.  THIS SAVE TIME IN THE
C     COMPUTATION OF THE SYMETRIC P MATRIX IN WHICH ONLY
C     ONE PORTION OF THE MATRIX NEED BE CALCULATED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),B(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,N
            RESULT(M,N)=A(M,N)-B(M,N)
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE HSUBOT(MR,RESULT,NR,NC,A)
C
C     THE UPPER TRIANGULAR PORTION TO TWO MATRICES ARE
C     SUBTRACTED AND STORED IN RESULT.  THIS SAVE TIME IN THE
C     COMPUTATION OF THE SYMETRIC P MATRIX IN WHICH ONLY
C     ONE PORTION OF THE MATRIX NEED BE CALCULATED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,N
            RESULT(M,N)=RESULT(M,N)-A(M,N)
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE DIFF(MR,IRSYM,RESULT,NR,NC,A,B)
```

```
C
C     THE DIFFERENCE OF TWO MATRICES IS CALCULATED AND
C     STORED IN THE MATRIX RESULT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),B(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,NR
            RESULT(M,N)=A(M,N)-B(M,N)
10    CONTINUE
      IF (IRSYM.EQ.1) THEN
         DO 50 INC=2,NC
            NRSTOP=INC-1
            DO 50 INR=1,NRSTOP
               AVE=(RESULT(INR,INC)+RESULT(INC,INR))/2.DO
               RESULT(INR,INC)=AVE
               RESULT(INC,INR)=AVE
50       CONTINUE
      ENDIF
      RETURN
      END
C
      SUBROUTINE ADDON(MR,RESULT,NR,NC,A)
C
C     THE A MATRIX IS ADDED ONTO THE INCOMING RESULT
C     MATRIX.  THUS THE RESULT MATRIX IS INCREMENTED WITH A.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,NR
            RESULT(M,N)=RESULT(M,N) + A(M,N)
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE SUBOUT(MR,RESULT,NR,NC,A)
C
C     THE A MATRIX IS SUBTRACTED OUT OF THE INCOMING RESULT
```

```
C     MATRIX.  THUS THE RESULT MATRIX IS DECREMENTED WITH A.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(MR,NC),RESULT(MR,NC)
      DO 10 N=1,NC
         DO 10 M=1,NR
            RESULT(M,N)=RESULT(M,N) - A(M,N)
10    CONTINUE
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                      CCCCC
CCCCC              END OF MATRIX SUBROUTINES               CCCCC
CCCCC                                                      CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                      CCCCC
CCCCC       KALMAN FILTER SUBROUTINES ARE AS FOLLOWS:      CCCCC
CCCCC                                                      CCCCC
CCCCC       SATUSE,SETUP,DIRCOS,MEASUR,UPDATE,PROJECT      CCCCC
CCCCC                                                      CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SATUSE
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C1/ SOL,RE,PI
      REAL*8 SOL,RE,PI
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C4/ ALPHA,GZERO,BETA,PHYZ,THETA,RS,HDPLMT
      REAL*8 ALPHA(24),GZERO(24)
      REAL*8 BETA,PHYZ,THETA,RS,HDPLMT
```

```
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1         NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER      NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C14/ PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY,NTURON,NTUROF
      REAL*8 PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY
      INTEGER NTURON,NTUROF
C
      REAL*8 SATGEO(2,24)
C
1     FORMAT(/1X,'**************** 24 SATELLITE CONSTELLATION'
     1     ,' USED IN SIMULATION *****************')
      WRITE(6,2)
      WRITE(6,1)
2     FORMAT(/1X,'****RESULTS**** FROM: CONNEC.WTF VERSION=1.6',
     1   ' MAY 29,1988')
C
      DEGINC=0.D0
      DO 10 M=1,21,4
         DO 20 N=1,4
            SATGEO(1,M+N-1)=DEGINC
20       CONTINUE
         DEGINC=DEGINC + 60.D0
10    CONTINUE
C
```

```
      INC15=0
      DO 30 M=1,21,4
         DEGINC=FLOAT(INC15)*15.D0
         DO 35 N=1,4
            SATGEO(2,M+N-1)=DEGINC + FLOAT(N-1)*90.D0
35       CONTINUE
         INC15=INC15 + 1
30    CONTINUE
C
      DO 40 I=1,24
         ALPHA(I)=SATGEO(1,I)
         GZERO(I)=SATGEO(2,I)
40    CONTINUE
C
      READ(5,*)OFFSET
4     FORMAT(/1X,'OFFSET FROM 0.0 GPS TIME IN SECONDS IS',F10.2)
      WRITE(6,4)OFFSET
C
      ADDDEG=OFFSET*360.D0/(43200.D0*0.99726957D0)
C
      DO 50 I=1,24
         GZERO(I)=GZERO(I) + ADDDEG
50    CONTINUE
C
      DEGCON=PI/180.D0
C
      READ(5,*)PHYZ,THETA
7     FORMAT(/1X,'INITIAL VEHICLE POSITION:',/5X,'EAST LONGITUDE:',
     1       F10.2,/5X,'NORTH LATITUDE:',F10.2)
C
      WRITE(6,7)PHYZ,THETA
      PHYZ=PHYZ + ADDDEG/2.D0
C
      READ(5,*)NTURON,NTUROF
      READ(5,*)PERIOD,BNKANG
C
C     NOW CALCULATE SOME PARAMETERS ABOUT THE VEHICLE TURN
C
      ACCEL=9.81*DTAN(BNKANG*DEGCON)
      RADFRQ=2*PI/PERIOD
      RADIUS=ACCEL/RADFRQ**2
      VELCTY=RADFRQ*RADIUS
C
3     FORMAT(/1X,'PARAMETERS OF THE VEHICLE DYNAMICS')
6     FORMAT(5X,'THE TIME TO COMPLETE THE TURN IS',F10.2,' s',
     1      /5X,'THE TANGENTIAL VELOCITY IS      ',F10.2,' m/s',
     2      /5X,'THE TURN RADIUS IS              ',F10.2,' m',
     3      /5X,'THE BANK ANGLE IS               ',F10.2,' degress',
     4      /5X,'THE CENTRIPITAL ACCELERATION IS ',F10.2,' m/s/s')
5     FORMAT(5X,'NTURON=',I4,2X,'       NTUROF=',I4)
C
      IF (NTURON.EQ.1) THEN
         WRITE(6,3)
         WRITE(6,6)PERIOD,VELCTY,RADIUS,BNKANG,ACCEL
         WRITE(6,5)NTURON,NTUROF
      ENDIF
C
      RS=4.1644*RE
      BETA=55.D0*DEGCON
      PHYZ=PHYZ*DEGCON
      THETA=THETA*DEGCON
C
      DO 60 I=1,24
         ALPHA(I)=DEGCON*ALPHA(I)
         GZERO(I)=DEGCON*GZERO(I)
60    CONTINUE
C
      READ(5,*)HDPLMT
      READ(5,*)NERRON,NERROF
      READ(5,*)NHYPON,NWSIZE,NALOUT
      READ(5,*)NC2ON,NC2SZ
      READ(5,*)NGEOM,NGMSZ
      READ(5,*)NGON,NGSZ
      READ(5,*)(LOCSIZ(I),I=1,4)
      READ(5,*)(BIAS(I),I=1,9),(SLOPE(I),I=1,9)
8     FORMAT(/1X,'**************** DETECTION PROBLEM PARAMETE'
     1    ,'RS ****************')
9     FORMAT(/1X,'NERRON,NERROF ARE:',2(I6,3X),
     1       /1X,'NHYPON,NWSIZE ARE:',2(I6,3X),
     2       /1X,'NC2ON,NC2SZ ARE:  ',2(I6,3X),
     3       /1X,'NGEOM,NGMSZ ARE:  ',2(I6,3X),
     4       /1X,'NGON,NGSZ ARE:    ',2(I6,3X))
43    FORMAT(/1X,'MAXIMUM HDOP FOR SUB-SOL:HDPLMT=',F10.3)
      WRITE(6,8)
      WRITE(6,9)NERRON,NERROF,NHYPON,NWSIZE,NC2ON,NC2SZ,
     1          NGEOM,NGMSZ,NGON,NGSZ
      WRITE(6,43)HDPLMT
12    FORMAT(/1X,'THE SATELITTE CLOCK ERROR '
     1  ,'BIAS AND SLOPE ARE:')
      WRITE(6,12)
11    FORMAT(/1X,9(F8.2,1X))
      WRITE(6,11)(BIAS(I),I=1,9)
      WRITE(6,11)(SLOPE(I),I=1,9)
C
      DO 47 I=1,9
         WBIAS(I)=BIAS(I)
47    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE SETUP
C
C  THIS SUBROUTINE READS IN ALL CONSTANT TYPE DATA. IN ADDITION IT
C  PERFORMS TASKS SUCH AS COMPUTING INITIAL MATRICES WHICH ARE
C  THE STATE TRANSITION MATRIX,Q,R,THE SELECTIVE AVAILABILITY PART
C  OF THE H MATRIX AND THE INITIAL MATRICES X,XHAT, AND P MINUS.
C  AN EXTERNAL SUBROUTINE DRNNOA FROM THE IMSL IS ALSO CALLED
C  WHICH RETURNS NORMAL VARIATE VECTORS.  THE NOISE SCALE FACTORS
C  ARE CALCULATED HERE TO SHAPE THE NORMAL PROCESSES INTO SEQUENCES
C  WITH THE CORRECT CORRELATION STRUCTURE AS REQUIRED BY Q.  THE
C  RESULT IS THE W VECTOR PROFILE FOR THE ENTIRE SIMULATION.
C  THE MEASUREMENT NOISE VECTOR V IS ALSO GENERATED SIMULARLY.
C  STARTING ANGLES FOR THE SATELLITE AND VEHICLE STARTING POSITIONS
C  ARE ALSO CALCULATED IN RADIANS FOR USE IN THE DIRECTION COSINES
C  SUBROUTINE.  THE STATISTICS OF THE W AND V VECTORS ARE CALCULATED
C  AND ARE WRITTEN INTO THE DEVICE SIX FILE.  THIS PROVIDES A CHECK
```

```
C  ON THE NATURE OF THE DRIVING FUNCTIONS WHICH ARE GENERATED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C1/ SOL,RE,PI
      REAL*8 SOL,RE,PI
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1           MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C4/ ALPHA,GZERO,BETA,PHYZ,THETA,RS,HDPLMT
      REAL*8 BETA,PHYZ,THETA,RS,HDPLMT
      REAL*8 ALPHA(24),GZERO(24)
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1       NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER    NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1    SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C11/ SASCFT,DSEED,READIC,ISEED
      REAL*8 SASCFT(4),DSEED,READIC
      INTEGER ISEED
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C14/ PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY,NTURON,NTUROF
      REAL*8 PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY
      INTEGER NTURON,NTUROF
C
      REAL*8 GNOISE(53,1501),WNSF(36),PINPUT
      REAL*8 GAUS(1001)
C      REAL*4 GAUS(1001)
C
C     THE PARAMETERS OF THE FILTER ARE READ FROM AN INPUT FILE.
C     VARIABLES:
C     DELTAT:SAMPLING INTERVAL IN SECONDS
C     NSTEPS:NUMBER OF RECURSIVE STEPS TO BE PERFORMED
C     WO:UNDAMPED NATURAL FREQUENCY OF THE MATCHETT PROCESS
C     BHARM:INVERSE TIME CONSTANT OF MATCHETT PROCESS
C     WNPSD4:CORRESPONDS TO Q SUB C IN MATCHETT REPORT,
C            THIS IS THE WHITE NOISE POWER SPECTRAL DENSITY
C            AMPLITUDE OF THE DRIVING FUNCTION OF THE PROCESS.
C     BMARK:INVERSE TIME CONSTANT OF THE MARKOV PROCESS
C     SIGMAS:SIGMA SQUARED,VARIANCE OF MARKOV PROCESS
C     WNPSD1:SPECTRAL AMPLITUDE WHICH DRIVES ACCELERATION ERROR
C     WNPSD2:SPECTRAL AMP. OF LEFTMOST  INTEGRATOR IN CLOCK MODEL
C     WNPSD3:SPECTRAL AMP. OF NOISE WHICH ENTERS BETWEEN INTEGRATORS
C            OF THE CLOCK MODEL
```

```
C     R(1):VARIANCE OF MEASUREMENT NOISE FOR EACH OF THE
C               PSEUDORANGE MEASUREMENTS
C     PINPUT:CONTROL VARIABLE WHICH SPECIFIES WHETHER OR NOT
C               INITIAL XTRUE,XHAT, AND P SHOULD BE READ FROM
C               AN EXTERNAL DATA FILE.
C
      READ(5,*)DELTAT,NSTEPS
      READ(5,*)WOM1,BHARM1,WNPSD4
      READ(5,*)WOM2,BHARM2,WNPSD5
      READ(5,*)WNPSDT,WNPSDF
      READ(5,*)PSDTR2,PSDTR3
      READ(5,*)WNPSD2,WNPSD3,BTA
      READ(5,*)PINPUT,NXTO,NXDEXT
      READ(5,*)NOSTAT,NOSA,NONOIS,NPE
C
      READIC=PINPUT
C
      IF (NOSA.EQ.1) THEN
         NRXT=8
      ELSE
         NRXT=44
      ENDIF
C
      MRZ=9
      NRZ=9
      MRXT=44
      MRXH=9
      NRXH=8
      NRXD=8
      NSATS=9
      NCOAST=0
C
      READ(5,*)RT
      READ(5,*)RF
C
C     NOW READ THE CHI-SQUARE TABLE
C
      READ(22,*)NDOFS,NSIZES
      READ(22,*)(TBLDOF(I),I=1,NDOFS)
      READ(22,*)(TBLSIZ(I),I=1,NSIZES)
      DO 44 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=1,9)
44    CONTINUE
      DO 41 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=10,18)
41    CONTINUE
      DO 42 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=19,27)
42    CONTINUE
      DO 43 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=28,36)
43    CONTINUE
      DO 45 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=37,45)
45    CONTINUE
      DO 46 I=1,NDOFS
         READ(22,*)(CHITBL(I,J),J=46,54)
46    CONTINUE
914   FORMAT(/1X,'TEST SIZE FOR ALL RESIDUALS IN WINDOW     =',F9.6,
     1       /1X,'TEST SIZE FOR THE RESIDUALS AT ONE STEP   =',F9.6,
     2       /1X,'TEST SIZE FOR RESIDUALS FROM ONE SATELLITE=',F9.6,
     3       /1X,'TEST SIZE FOR THE HORIZONTAL PLANE CHECK  =',F9.6)
      WRITE(6,914)(TBLSIZ(LOCSIZ(I)),I=1,4)
C
7     FORMAT(//1X,'**************** FILTER PARAMETERS',
     1             '****************',/)
      WRITE(6,7)
1     FORMAT(1X,'DELTAT,NSTEPS',F10.3,2(2X,I5))
      WRITE(6,1)DELTAT,NSTEPS
2     FORMAT(/1X,'WOM1,BHARM1,WNPSD4: ',3(D10.3,1X))
      WRITE(6,2)WOM1,BHARM1,WNPSD4
3     FORMAT(/1X,'WOM2,BHARM2,WNPSD5: ',3(D10.3,1X))
      WRITE(6,3)WOM2,BHARM2,WNPSD5
4     FORMAT(/1X,'WNPSDT: PSD OF WHITE ACCELERATION NOISE IN TRUTH',
     1           '=',F10.4,
     2       /1X,'WNPSDF: PSD OF WHITE ACCELERATION NOISE FOR',
     3       /1X,'        NAVIGATION AND DETECTION',
     4           ' FILTERS=',10X,F10.4)
```

```
      WRITE(6,4)WNPSDT,WNPSDF
8     FORMAT(/1X,'RTRUE :',2X,D12.5)
13    FORMAT(/1X,'R FILTER:',D12.5)
C
17    FORMAT(/1X,'WNPSD2 AND WNPSD3 ARE THE PSDS OF THE LEFT AND',
     1  /1X,'RIGHT WHITE NOISE INPUTS IN THE CLOCK MODEL.',
     2  /1X,'WNPSD2=2*(PI**2)*H(-2)       WNPSD3=H(0)/2',
     3  /1X,'IN THE TRUTH MODEL AND NAV. FILTER: WNPSD2=',D14.6,
     4  /1X,'IN THE TRUTH MODEL AND NAV. FILTER: WNPSD3=',D14.6,
     5/1X,'IN DETECTION FILTER MODEL: WNPSD2=',D14.6,3X,'BTA=',F6.4,
     6/1X,'IN DETECTION FILTER MODEL: WNPSD3=',D14.6)
      WRITE(6,17)PSDTR2,PSDTR3,WNPSD2,BTA,WNPSD3
      WRITE(6,8)RT
      WRITE(6,13)RF
C
31    FORMAT(/1X,'NOSTAT:NO OUTPUT OF NOISE STAT=  ',I3,
     1        /1X,'NOSA:NO SELECTIVE AVAILABILITY=  ',I3,
     2        /1X,'NONOIS:NO PROCESS OR MEAS. NOISE=',I3,
     3        /1X,'NXTO:SET XTRUE=0 AT START=       ',I3,
     4        /1X,'NXDEXT:SET XDET=XHAT AT START=   ',I3)
      WRITE(6,31)NOSTAT,NOSA,NONOIS,NXTO,NXDEXT
C
      SOLSQ=SOL**2
      WNPSD2=SOLSQ*WNPSD2
      WNPSD3=SOLSQ*WNPSD3
      PSDTR2=SOLSQ*PSDTR2
      PSDTR3=SOLSQ*PSDTR3
C
C     INITIALIZE THE BIG MATRICES IN THE TRUTH MODEL TO ZERO
C
      DO 21 N=1,NRXT
         XTRUEN(N,1)=0.D0
         XTRUED(N,1)=0.D0
         DO 21 M=1,NRXT
            HTRUE(M,N)=0.D0
            PHITRU(M,N)=0.D0
            QTRUE(M,N)=0.D0
            NULLB(M,N)=0.D0
21    CONTINUE
```

```
C
C      INITIALIZE THE FILTER MODEL MATRICES TO ZERO
C
      DO 10 N=1,MRXH
         XHAT(N,1)=0.DO
         XDET(N,1)=0.DO
         U(N,1)=0.DO
         UT(N,1)=0.DO
         DO 10 M=1,MRXH
            P(M,N)=0.DO
            PDET(M,N)=0.DO
            Q(M,N)=0.DO
            PHI(M,N)=0.DO
            PHID(M,N)=0.DO
            NULL(M,N)=0.DO
            IDENT(M,N)=0.DO
10    CONTINUE
C
      DO 22 I=1,MRXH
         IDENT(I,I)=1.DO
22    CONTINUE
C
C     LOAD THE CONSTANT PORTION OF THE H MATRIX
C
      DO 15 M=1,NRZ
         DO 15 N=1,NRXH
            H(M,N)=0.DO
15    CONTINUE
C
      DO 19 I=1,NRZ
         H(I,7)=1.DO
         HTRUE(I,7)=1.DO
19    CONTINUE
C
      IF (NOSA.NE.1) THEN
         DO 12 I=1,NRZ
            HTRUE(I,5+4*I)=1.DO
            HTRUE(I,7+4*I)=1.DO
12       CONTINUE
```

```
      ENDIF
C
9     FORMAT(1X,'PINPUT',F6.1)
      WRITE(6,9)PINPUT
      IF (PINPUT.EQ.1.0) THEN
14    FORMAT(/1X,'INITIAL CONDITIONS ARE READ EXTERNALLY')
      WRITE(6,14)
      IF (NOSA.NE.1) THEN
         READ(21,*)(XTRUEN(J,1),J=1,44)
      ELSE
         READ(21,*)(XTRUEN(J,1),J=1,8)
      ENDIF
      READ(21,*)(XHAT(J,1),J=1,8)
      DO 5 N=1,8
         READ(21,*)(P(M,N),M=1,N)
5     CONTINUE
      READ(21,*)(XTRUED(J,1),J=1,8)
      READ(21,*)(XDET(J,1),J=1,8)
      DO 6 N=1,8
         READ(21,*)(PDET(M,N),M=1,N)
6     CONTINUE
      IF (NOSA.NE.1) THEN
         DO 57 J=9,44
            XTRUED(J,1)=XTRUEN(J,1)
57       CONTINUE
      ENDIF
      ENDIF
C
      IF (PINPUT.EQ.1.0) THEN
C
C        SINCE WE ARE RUNNING OPEN LOOP HERE, RESET THE DETECTION
C        FILTER TO THE NAVIGATION FILTER.
C
         CALL SUM(MRXT,0,XTRUED,NRXT,1,XTRUEN,NULL)
C
      ENDIF
C
C     FOR NUMERICAL REASONS, RESET THE TRUE ERROR TO THE ESTTIMATION
C     ERROR AND ZERO OUT THE STATE ESTIMATES.  LEAVE THE CLOCK
```

```
C     ERROR AND ESTIMATES UNCHANGED.
C
      DO 950 I=1,6
         XTRUEN(I,1)=XTRUEN(I,1)-XHAT(I,1)
         XHAT(I,1)=0.D0
         XTRUED(I,1)=XTRUED(I,1)-XDET(I,1)
         XDET(I,1)=0.D0
950   CONTINUE
C
      IF (NXDEXT.EQ.1) THEN
C
C        SET THE DETECTION FILTER COVARIANCE AND ESTIMATES AND TRUTH
C        EQUAL TO THOSE FROM THE NAVIGATION FILTER
C
         CALL SUM(MRXH,0,XTRUED,NRXH,1,XTRUEN,NULL)
         CALL SUM(MRXH,0,XDET,NRXH,1,XHAT,NULL)
         CALL SUM(MRXH,0,PDET,NRXH,NRXH,P,NULL)
C
      ENDIF
C
C     FORM ELEMENTS OF TRANSITION MATRIX FOR THE
C     SELECTIVE AVAILABILITY STATES
C
      IF (NOSA.NE.1) THEN
      DAMPM1=DEXP(-BHARM1*WOM1*DELTAT)
      DAMPM2=DEXP(-BHARM2*WOM2*DELTAT)
      W1M1=WOM1*DSQRT(1-BHARM1**2)
      W1M2=WOM2*DSQRT(1-BHARM2**2)
      B1WO1=BHARM1*WOM1/W1M1
      B2WO1=BHARM2*WOM2/W1M2
      ARG1=W1M1*DELTAT
      ARG2=W1M2*DELTAT
      PHI111=DAMPM1*(DCOS(ARG1)+B1WO1*DSIN(ARG1))
      PHI211=DAMPM2*(DCOS(ARG2)+B2WO1*DSIN(ARG2))
      PHI112=DAMPM1*DSIN(ARG1)/W1M1
      PHI212=DAMPM2*DSIN(ARG2)/W1M2
      PHI121=-(WOM1**2)*PHI112
      PHI221=-(WOM2**2)*PHI212
      PHI122=DAMPM1*(DCOS(ARG1)-B1WO1*DSIN(ARG1))
      PHI222=DAMPM2*(DCOS(ARG2)-B2W01*DSIN(ARG2))
      ENDIF
C
C     LOAD THE STATE TRANSITION MATRIX
C
      DO 40 I=1,8
         PHI(I,I)=1.DO
         PHITRU(I,I)=1.DO
         IF (MOD(I,2).EQ.1) THEN
            PHI(I,I+1)=DELTAT
            PHITRU(I,I+1)=DELTAT
         ENDIF
40    CONTINUE
C
C     NOW GET THE TRANSITION MATRIX FOR THE DETECTION FILTER
C     START WITH THE ONE FROM THE NAVIGATION FILTER.
C
      CALL SUM(MRXH,0,PHID,NRXD,NRXD,PHI,NULL)
C
C     NOW MODIFY JUST THE CLOCK ELEMENTS
C
      IF (BTA.NE.0.0) THEN
         PHID(7,8)=(1.0/BTA)*(1.0 - DEXP(-BTA*DELTAT))
         PHID(8,8)=DEXP(-BTA*DELTAT)
      ENDIF
C
C     NOW GET PHID TRANSPOSE
C
      CALL TRANSP(MRXH,PHIDTR,NRXD,NRXD,PHID,NRXD,NRXD)
C
      IF (NOSA.EQ.1) GO TO  51
      ISTOP=9 + 4*(NSATS-1)
      DO 50 I=9,ISTOP,4
         PHITRU(I,I)=PHI111
         PHITRU(I,I+1)=PHI112
         PHITRU(I+1,I)=PHI121
         PHITRU(I+1,I+1)=PHI122
50    CONTINUE
C
```

```
      ISTOP=11 + 4*(NSATS-1)
      DO 60 I=11,ISTOP,4
         PHITRU(I,I)=PHI211
         PHITRU(I,I+1)=PHI212
         PHITRU(I+1,I)=PHI221
         PHITRU(I+1,I+1)=PHI222
60    CONTINUE
51    CONTINUE
C
C     GENERATE PHI TRANSPOSE
C
      CALL TRANSP(MRXH,PHITR,NRXH,NRXH,PHI,NRXH,NRXH)
C
C     THE STANDARD NORMAL VARIABLES ARE FOUND USING AN
C     EXTERNAL RANDAM NUMBER GENERATOR FROM THE IMSL.
C     THE SEEDS ARE READ FROM THE DATA FILE.
C     THE R.N.G. IS CALLED TWENTY FOUR TIMES. THE RESULT OF
C     EACH CALL IS STORED IN ROWS OF THE GNOISE ARRAY WHICH
C     IS A (24xNSTEPS+1) ARRAY.  THE LAST COLUMN OF THIS ARRAY
C     CAN BE APPROPRIATELY SCALED TO SERVE AS RANDOM
C     INITIAL CONDITIONS.
C
      NR=NSTEPS
C      READ(5,*)DSEED
      READ(5,*)ISEED
      CALL RNSET(ISEED)
71    FORMAT(/1X,'************ SEED USED IN THE SIMULATION '
C     1           ,'= ',D14.7,1X,'************')
     1           ,'= ',I12,1X,'************')
C      WRITE(6,71)DSEED
      WRITE(6,71)ISEED
      DO 70 I=1,NRXT
C         CALL GGNPM(DSEED,NR,GAUS)
         CALL DRNNOA(NR,GAUS)
         DO 80 J=1,NR
            GNOISE(I,J)=GAUS(J)
80       CONTINUE
70    CONTINUE
C
```

```
      NR=NSTEPS
      DO 85 I=1,NRZ
C         CALL GGNPM(DSEED,NR,GAUS)
         CALL DRNNOA(NR,GAUS)
         DO 87 J=1,NSTEPS
            GNOISE(I+NRXT,J)=GAUS(J)
87       CONTINUE
85    CONTINUE
C
C     THE Q MATRIX IS LOADED NEXT.
C
C     THE Q ELEMENTS FOR THE POSITION AND VELOCITY STATES
C
      TCUBE=(DELTAT**3)/3.DO
      TSQ=(DELTAT**2)/2.DO
C
C     LOAD Q FOR THE POSITION STATES
C
      DO 90 J=1,5,2
         Q(J,J)=WNPSDF*TCUBE
         Q(J,J+1)=WNPSDF*TSQ
         Q(J+1,J+1)=WNPSDF*DELTAT
         QTRUE(J,J)=WNPSDT*TCUBE
         QTRUE(J,J+1)=WNPSDT*TSQ
         QTRUE(J+1,J+1)=WNPSDT*DELTAT
90    CONTINUE
C
C     Q ELEMENTS FOR THE CLOCK STATES
C
      Q(7,7)=PSDTR2*TCUBE+PSDTR3*DELTAT
      Q(7,8)=PSDTR2*TSQ
      Q(8,8)=PSDTR2*DELTAT
C
      QTRUE(7,7)=Q(7,7)
      QTRUE(7,8)=Q(7,8)
      QTRUE(8,8)=Q(8,8)
C
C     COPY THE FILTER Q INTO QDET
C
```

```
      DO 91 NC=1,9
         DO 91 NR=1,9
            QDET(NR,NC)=Q(NR,NC)
91    CONTINUE
C
C     MODIFY THE Q ELEMENTS FOR THE CLOCK STATES IN THE DET. FILTER
C
      IF (BTA.EQ.0.0) THEN
         QDET(7,7)=WNPSD2*TCUBE+WNPSD3*DELTAT
         QDET(7,8)=WNPSD2*TSQ
         QDET(8,8)=WNPSD2*DELTAT
      ELSE
         EXPBT1=1-DEXP(-BTA*DELTAT)
         EXPBT2=1-DEXP(-2*BTA*DELTAT)
         CON1=1.0/BTA
         CON2=0.5/BTA
         A2BYB=WNPSD2/BTA
         QDET(7,7)=WNPSD3*DELTAT + (A2BYB/BTA)*
     1           (DELTAT - 2*CON1*EXPBT1 + CON2*EXPBT2)
         QDET(7,8)=A2BYB*(CON1*EXPBT1 - CON2*EXPBT2)
         QDET(8,8)=0.5*A2BYB*EXPBT2
      ENDIF
C
      IF (NOSA.EQ.1) GO TO 92
C
C     Q ELEMENTS FOR THE SELECTIVE AVAILABILITY STATES
C
      QM1C11=WNPSD4/(4*BHARM1*WOM1**3)
      QM2C11=WNPSD5/(4*BHARM2*WOM2**3)
      DAMP1=DEXP(-2.D0*BHARM1*WOM1*DELTAT)
      DAMP2=DEXP(-2.D0*BHARM2*WOM2*DELTAT)
      B1W10=BHARM1*W1M1/WOM1
      B2W10=BHARM2*W1M2/WOM2
      WO1SQ1=(WOM1/W1M1)**2
      WO1SQ2=(WOM2/W1M2)**2
      BSQ1=BHARM1**2
      BSQ2=BHARM2**2
      QM1C12=WNPSD4/(4*W1M1**2)
      QM2C12=WNPSD5/(4*W1M2**2)
      QM1C22=WNPSD4/(4*BHARM1*W0M1)
      QM2C22=WNPSD5/(4*BHARM2*W0M2)
      AR1=2.D0*W1M1*DELTAT
      AR2=2.D0*W1M2*DELTAT
      WD1=W01SQ1*DAMP1
      WD2=W01SQ2*DAMP2
      Q1HR11=QM1C11*(1.D0-WD1*(1.D0-BSQ1*DCOS(AR1)+B1W10*DSIN(AR1)))
      Q2HR11=QM2C11*(1.D0-WD2*(1.D0-BSQ2*DCOS(AR2)+B2W10*DSIN(AR2)))
      Q1HR12=QM1C12*DAMP1*(1.D0-DCOS(AR1))
      Q2HR12=QM2C12*DAMP2*(1.D0-DCOS(AR2))
      Q1HR22=QM1C22*(1.D0-WD1*(1.D0-BSQ1*DCOS(AR1)-B1W10*DSIN(AR1)))
      Q2HR22=QM2C22*(1.D0-WD2*(1.D0-BSQ2*DCOS(AR2)-B2W10*DSIN(AR2)))
C
C     SAVE THE MEAN SQUARE VALUES OF THE TWO SA PROCESSES
C     WHICH WILL BE USED FOR SCALING THE INITIAL CONDITIONS.
C
      SASCFT(1)=DSQRT(QM1C11)
      SASCFT(2)=DSQRT(QM1C22)
      SASCFT(3)=DSQRT(QM2C11)
      SASCFT(4)=DSQRT(QM2C22)
C
      JSTOP=9 + 4*(NSATS-1)
      DO 95 J=9,JSTOP,4
         QTRUE(J,J)=Q1HR11
         QTRUE(J,J+1)=Q1HR12
         QTRUE(J+1,J+1)=Q1HR22
95    CONTINUE
C
      JSTOP=11 + 4*(NSATS-1)
      DO 100 J=11,JSTOP,4
         QTRUE(J,J)=Q2HR11
         QTRUE(J,J+1)=Q2HR12
         QTRUE(J+1,J+1)=Q2HR22
100   CONTINUE
C
92    CONTINUE
C
C     DETERMINE THE SCALE FACTORS TO SHAPE THE STANDARD
C     NORMAL VARIABLES INTO THE CORRELATION STRUCTURE
```

```
C     AS DEMANDED BY THE Q MATRIX
C
C     SCALE FACTORS FOR HORIZONTAL STATES:
C
      WNSF(1)=DSQRT(QTRUE(1,1))
      WNSF(2)=QTRUE(1,2)/WNSF(1)
      WNSF(3)=DSQRT(QTRUE(2,2)-WNSF(2)**2)
C
C     SCALE FACTORS FOR THE CLOCK STATES
C
      WNSF(4)=DSQRT(QTRUE(7,7))
      WNSF(5)=QTRUE(7,8)/WNSF(4)
      WNSF(6)=DSQRT(QTRUE(8,8)-WNSF(5)**2)
C
C     SCALE FACTORS FOR THE VERTICAL CHANNEL
C
      WNSF(34)=DSQRT(QTRUE(5,5))
      WNSF(35)=QTRUE(5,6)/WNSF(34)
      WNSF(36)=DSQRT(QTRUE(6,6)-WNSF(35)**2)
C
C     SCALE FACTORS FOR THE SELECTIVE AVAILABILITY PROCESSES
C
      IF (NOSA.NE.1) THEN
         WNSF(7)=DSQRT(QTRUE(9,9))
         WNSF(8)=QTRUE(9,10)/WNSF(7)
         WNSF(9)=DSQRT(QTRUE(10,10)-WNSF(8)**2)
         WNSF(10)=DSQRT(QTRUE(11,11))
         WNSF(11)=QTRUE(11,12)/WNSF(10)
         WNSF(12)=DSQRT(QTRUE(12,12)-WNSF(11)**2)
      ENDIF
C
C     SCALE FACTORS FOR THE MEASUREMENT NOISE
C
      WNSF(13)=DSQRT(RT)
C
C     IF INPUT OF THE INITIAL P AND X IS NOT DESIRED, THEN
C     SET XTRUE EQUAL TO NEW RANDOM VARIBLES CALLED FROM IMSL.
C     OF COURSE, THESE VARIABELS MUST BE APPROPRIATELY SCALED.
C     THE INITIAL ESTIMATES OF THE SELECTIVE AVAILABILITY
```

```
C     STATES ARE SET TO ZERO SINCE NO BETTER ESTIMATE IS KNOWN.
C     THE ELEMENTS OF THE INITIAL ERROR COVARIANCE ARE SET TO
C     VARIANCE OF THE PROCESSES.  FOR THE STANDARD EIGHT-STATE
C     FILTER VARIABLES, THE COVARIANCE IS ZERO AND THE ESTIMATE
C     IS SET EQUAL TO THE TRUE PROCESS.  THIS CORRESPONDS TO A
C     PERFECT ESTIMATE OF INITIAL POSITION,VELOCITY, AND
C     CLOCK ERRORS.
C
      IF (PINPUT.NE.1.0) THEN
C        CALL THE RANDOM NUMBER GENERATOR FOR I.C.'S
C         CALL GGNPM(DSEED,NRXT,GAUS)
         CALL DRNNOA(NRXT,GAUS)
C
      DO 101 I=1,8
         XTRUEN(I,1)=GAUS(I)
         XTRUED(I,1)=GAUS(I)
         XHAT(I,1)=XTRUEN(I,1)
         XDET(I,1)=XTRUED(I,1)
101   CONTINUE
C
      IF (NOSA.EQ.1) GO TO 94
C
      ISTOP=9+4*(NSATS-1)
      DO 102 I=9,ISTOP,4
         XTRUEN(I,1)=GAUS(I)*SASCFT(1)
         XTRUED(I,1)=XTRUEN(I,1)
102   CONTINUE
C
      ISTOP=10+4*(NSATS-1)
      DO 103 I=10,ISTOP,4
         XTRUEN(I,1)=GAUS(I)*SASCFT(2)
         XTRUED(I,1)=XTRUEN(I,1)
103   CONTINUE
C
      ISTOP=11+4*(NSATS-1)
      DO 104 I=11,ISTOP,4
         XTRUEN(I,1)=GAUS(I)*SASCFT(3)
         XTRUED(I,1)=XTRUEN(I,1)
```

```
104   CONTINUE
C
      ISTOP=12+4*(NSATS-1)
      DO 105 I=12,ISTOP,4
         XTRUEN(I,1)=GAUS(I)*SASCFT(4)
         XTRUED(I,1)=XTRUEN(I,1)
105   CONTINUE
C
94    CONTINUE
C
      ENDIF
C
      IF (NXTO.EQ.1) THEN
          CALL SUM(MRXT,0,XTRUEN,NRXT,1,NULLB,NULLB)
          CALL SUM(MRXT,0,XTRUED,NRXT,1,NULLB,NULLB)
          CALL SUM(MRXH,0,XHAT,NRXH,1,NULL,NULL)
          CALL SUM(MRXH,0,XDET,NRXH,1,NULL,NULL)
      ENDIF
C
C     NOW TAKE CARE OF SETTING THE POSITION AND VELOCITY COORDINATES
C     UP FOR THE X AND Y DIRECTIONS AT THE BEGINNING OF THE TURN
C
      IF (NTURON.EQ.1) THEN
C
C         FOR THE X POSITION AND VELOCITY
C
          RADIUS=VELCTY/RADFRQ
C
          XHAT(1,1)=-RADIUS + XTRUEN(1,1)
          XDET(1,1)=-RADIUS + XTRUED(1,1)
          XTRUEN(1,1)=-RADIUS
          XTRUED(1,1)=-RADIUS
C
          XHAT(2,1)=XTRUEN(2,1)
          XDET(2,1)=XTRUED(2,1)
          XTRUEN(2,1)=0.D0
          XTRUED(2,1)=0.D0
C
C         FOR THE Y POSITION AND VELOCITY
```

```
          XHAT(3,1)=XTRUEN(3,1)
          XDET(3,1)=XTRUED(3,1)
          XTRUEN(3,1)=0.DO
          XTRUED(3,1)=0.DO
C
          XHAT(4,1)=-VELCTY + XTRUEN(4,1)
          XDET(4,1)=-VELCTY + XTRUED(4,1)
          XTRUEN(4,1)=-VELCTY
          XTRUED(4,1)=-VELCTY
C
      ENDIF
C
16    FORMAT(1X,'*************** INITIAL CONDITIONS ',
     1            '***************')
      WRITE(6,16)
109   FORMAT(/1X,'X NAV. TRUE O',6(/1X,8(1X,D14.6)))
      WRITE(6,109)(XTRUEN(I,1),I=1,NRXT)
106   FORMAT(/1X,' X HAT O',/1X,8(1X,D14.6))
      WRITE(6,106)(XHAT(I,1),I=1,NRXH)
108   FORMAT(/1X,'P DIA. O',/1X,8(1X,D14.6))
      WRITE(6,108)(P(I,I),I=1,NRXH)
C
58    FORMAT(/1X,'X DET. TRUE O',6(/1X,8(1X,D14.6)))
      WRITE(6,58)(XTRUED(I,1),I=1,NRXT)
107   FORMAT(/1X,' X DET O',/1X,8(1X,D14.6))
      WRITE(6,107)(XDET(I,1),I=1,NRXH)
111   FORMAT(/1X,'PDET DIA. O',/1X,8(1X,D14.6))
      WRITE(6,111)(PDET(I,I),I=1,NRXH)
C
C     GENERATE THE W AND V PROCESSES FOR THE REQUIRED NUMBER
C     OF POINTS.
C
      DO 140 J=1,NSTEPS
      DO 110 I=1,3,2
         WNOIS(I,J)=GNOISE(I,J)*WNSF(1)
         WNOIS(I+1,J)=GNOISE(I,J)*WNSF(2)+GNOISE(I+1,J)*WNSF(3)
110   CONTINUE
      WNOIS(5,J)=GNOISE(5,J)*WNSF(34)
      WNOIS(6,J)=GNOISE(5,J)*WNSF(35)+GNOISE(6,J)*WNSF(36)
      WNOIS(7,J)=GNOISE(7,J)*WNSF(4)
      WNOIS(8,J)=GNOISE(7,J)*WNSF(5)+GNOISE(8,J)*WNSF(6)
C
      IF (NOSA.EQ.1) GO TO 121
C
      ISTOP=9+4*(NSATS-1)
      DO 120 I=9,ISTOP,4
         WNOIS(I,J)=GNOISE(I,J)*WNSF(7)
         WNOIS(I+1,J)=GNOISE(I,J)*WNSF(8)+GNOISE(I+1,J)*WNSF(9)
120   CONTINUE
C
      ISTOP=11+4*(NSATS-1)
      DO 130 I=11,ISTOP,4
         WNOIS(I,J)=GNOISE(I,J)*WNSF(10)
         WNOIS(I+1,J)=GNOISE(I,J)*WNSF(11)+GNOISE(I+1,J)*WNSF(12)
130   CONTINUE
C
121   CONTINUE
C
140   CONTINUE
C
      DO 150 J=1,NSTEPS
         DO 150 I=1,NRZ
            VNOIS(I,J)=GNOISE(I+NRXT,J)*WNSF(13)
150   CONTINUE
C
C      SUBTRACT OUT MEAN OF W PROCESS
C      BY DEFINITION, W AND V MUST BE ZERO MEAN PROCESSES.
C      BECAUSE THE IMSL RANDOM NUMBER GENERATOR TENDS TO GIVE
C      VARIATES WILL A SMALL MEAN, PROVISIONS MUST BE MADE TO
C      TO SUBTACT OUT THESE AVERAGE VALUES.
C
      DO 230 I=1,NRXT
         SUM1=0.0
         DO 231 J=1,NSTEPS
231         SUM1=SUM1+WNOIS(I,J)
         WMEAN=SUM1/FLOAT(NSTEPS)
         DO 232 J=1,NSTEPS
232         WNOIS(I,J)=WNOIS(I,J)-WMEAN
```

```
230   CONTINUE
C
C      SUBTRACT OUT MEAN OF V PROCESS
C
      DO 233 I=1,NRZ
         SUM1=0.0
         DO 234 J=1,NSTEPS
234         SUM1=SUM1+VNOIS(I,J)
         VMEAN=SUM1/FLOAT(NSTEPS)
         DO 235 J=1,NSTEPS
            VNOIS(I,J)=VNOIS(I,J)-VMEAN
235   CONTINUE
233   CONTINUE
C
C      DETERMINE CORRELATION STRUCTURE OF W AND V
C      THIS IS DONE TO ALERT THE USER OF THE TRUE NATURE
C      OF THE NOISE PROCESSES WHICH DRIVES THE SYSTEM.
C      THIS IS DONE ONLY AS A CHECK AND HELPS IN UNDERSTAND-
C      ING THE PARTICULAR RESULTS OF THE SIMULATION.
C
      IF (NOSTAT.NE.1) WRITE(6,261)
      ISTOP=NRXT-1
      DO 250 I=1,ISTOP,2
         SUM1=0.0
         SUM2=0.0
         S1SQ=0.0
         S2SQ=0.0
         CROSS=0.0
        DO 260 J=1,NSTEPS
           SUM1=SUM1+WNOIS(I,J)
           SUM2=SUM2+WNOIS(I+1,J)
           S1SQ=S1SQ+WNOIS(I,J)**2
           S2SQ=S2SQ+WNOIS(I+1,J)**2
           CROSS=WNOIS(I,J)*WNOIS(I+1,J)+CROSS
260   CONTINUE
         XM1=SUM1/FLOAT(NSTEPS)
         XM2=SUM2/FLOAT(NSTEPS)
         VAR1=(S1SQ-SUM1**2/FLOAT(NSTEPS))/(FLOAT(NSTEPS-1))
         VAR2=(S2SQ-SUM2**2/FLOAT(NSTEPS))/(FLOAT(NSTEPS-1))
         COV=(CROSS-SUM1*SUM2/FLOAT(NSTEPS))/(FLOAT(NSTEPS-1))
         IF (NOSTAT.NE.1)
     1      WRITE(6,262)I,XM1,I,VAR1,I+1,XM2,I+1,VAR2,COV
         IF (NOSTAT.NE.1) WRITE(6,263)I,QTRUE(I,I),I+1,
     1                     QTRUE(I+1,I+1),QTRUE(I,I+1)
250   CONTINUE
261   FORMAT(/1X,'**************** STATISTICS OF PROCESS ',
     1       'VECTOR W ****************')
262   FORMAT(/1X,'MEAN',I3,'=',D11.4,4X,'VAR',I3,'=',D11.4,4X,
     1'MEAN',I3,'=',D11.4,4X,'VAR',I3,'=',D11.4,4X,
     2'COVARIANCE=',D11.4)
263   FORMAT(6X,'QTRUE MATRIX SAYS VAR',I3,'=',D11.4,27X,'VAR',I3,
     1  '=',D11.4,4X,'COVARIANCE=',D11.4)
      IF (NOSTAT.NE.1) WRITE(6,321)
      DO 310 I=1,NRZ
         SUM1=0.0
         S1SQ=0.0
         DO 320 J=1,NSTEPS
            SUM1=SUM1+VNOIS(I,J)
            S1SQ=S1SQ+VNOIS(I,J)**2
320   CONTINUE
         XM1=SUM1/FLOAT(NSTEPS)
         VAR=(S1SQ-SUM1**2/FLOAT(NSTEPS))/FLOAT(NSTEPS-1)
         IF (NOSTAT.NE.1) WRITE(6,322)I,XM1,VAR,RT
310   CONTINUE
321      FORMAT(/1X,'***************** STATISTICS OF MEASURE',
     1     'MENT NOISE VECTOR V ****************',/)
322      FORMAT(1X,'MEAN',I3,'=',D10.3,3X,'VAR=',D10.3,
     1     3X,'RTRUE MATRIX SAYS VAR=',D10.3,/)
      RETURN
      END
C
      SUBROUTINE CLKCON(K)
C
C     THIS SUBROUTINE DOES ALL THE SETUP WORK WHEN CLOCK COASTING IS
C     STARTED.  IT COPIES THE CLOCK COVARIANCE INTO PY AND THE
C     POSITION AND CLOCK COVARIANCE INTO PXY.  IT ALSO SETS UP ALL
C     OTHER SUPPORT MATRICES NEED DURING CLOCK COASTING.
C
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      INTEGER K
C
C     FIRST RESET THE MATRIX DIMENSIONS
C
      NRXD=6
C
C     COPY THE CLOCK COVARIANCE
C
      PY(1,1)=PDET(7,7)
      PY(1,2)=PDET(7,8)
      PY(2,2)=PDET(8,8)
      PY(2,1)=0.DO
C
C     COPY THE POSITION AND CLOCK COVARIANCE
C
      DO 10 J=1,2
         DO 10 I=1,6
            PXY(I,J)=PDET(I,J+6)
```

```
10    CONTINUE
C
C     FORM THE QY MATRIX FROM THE QDET FROM THE DETECTION FILTER
C
      QY(1,1)=QDET(7,7)
      QY(1,2)=QDET(7,8)
      QY(2,2)=QDET(8,8)
C
C     FORM THE TRANSITION MATRIX AND ITS TRANSPOSE FOR THE CLOCK
C
      PHIY(1,1)=1.D0
      PHIY(1,2)=PHID(7,8)
      PHIY(2,1)=0.D0
      PHIY(2,2)=PHID(8,8)
C
      CALL TRANSP(MRXH,PHIYT,2,2,PHIY,2,2)
C
C     FORM THE CONNECTION MATRIX IN THE MEASUREMENT EQUATION: HY
C
      DO 20 I=1,NSATS
         HY(I,1)=1.D0
         HY(I,2)=0.D0
20    CONTINUE
C
C     NOW LOAD THE CURRENT CLOCK ESTIMATES INTO THE YHAT VECTOR
C
      YHAT(1,1)=XDET(7,1)
      YHAT(2,1)=XDET(8,1)
C
      RETURN
      END
C
      SUBROUTINE CLKCOF(K)
C
C     THIS SUBROUTINE DOES ALL THE SETUP WORK WHEN CLOCK COASTING IS
C     ENDED.  IT COPIES THE PROJECTED CLOCK COVARIANCE BACK INTO P
C     AND THE POSITION AND CLOCK COVARIANCE BACK INTO P.
C
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1       ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      INTEGER K
C
C     FIRST RESET THE MATRIX DIMENSIONS
C
      NRXD=8
C
C     COPY THE CLOCK COVARIANCE
C
      PDET(7,7)=PY(1,1)
      PDET(7,8)=PY(1,2)
      PDET(8,8)=PY(2,2)
C
C     COPY THE POSITION AND CLOCK COVARIANCE
C
      DO 10 J=1,2
         DO 10 I=1,6
            PDET(I,J+6)=PXY(I,J)
10    CONTINUE
C
```

```
C     LOAD THE PROJECTED CLOCK ESTIMATES BACK INTO THE XHAT VECTOR
C
      XDET(7,1)=YHAT(1,1)
      XDET(8,1)=YHAT(2,1)
C
      RETURN
      END
C
      SUBROUTINE DIRCOS(K)
C
C     THIS SUBROUTINE COMPUTES THE DIRECTION COSINES OF THE
C     LINE OF SIGHT VECTOR FROM THE VEHICLE TO THE SATELLITE
C     PROJECTED INTO THE VEHICLES LOCALLY LEVEL COORDINATE FRAME
C     OF REFERENCE.  THE INITIAL ANGLES WHICH DEFINE THE INITIAL
C     VEHICLE AND SATELLITES POSITION ARE COMPUTED IN SUBROUTINE
C     SETUP IN UNITS OF RADIANS.  DIR. COS. ARE USED IN THE H MATRIX
C     WHICH DESCRIBES THE PROJECTION OF THE PSEUDORANGE MEASURE-
C     MENT INTO THE POSITION ERRORS AND CLOCK ERRORS.  A SET OF
C     DIRECTION COSINES IS COMPUTED FOR EACH SATELLITE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C1/ SOL,RE,PI
      REAL*8 SOL,RE,PI
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C4/ ALPHA,GZERO,BETA,PHYZ,THETA,RS,HDPLMT
      REAL*8 ALPHA(24),GZERO(24)
      REAL*8 BETA,PHYZ,THETA,RS,HDPLMT
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C11/ SASCFT,DSEED,READIC,ISEED
      REAL*8 SASCFT(4),DSEED,READIC
      INTEGER ISEED
C
      REAL*8 XI(24),YI(24),ZI(24),XE(24),YE(24),ZE(24)
      REAL*8 CXALL(24),CYALL(24),CZALL(24),GAMMA(24)
      REAL*8 CX(9),CY(9),CZ(9)
      REAL*8 G(9,9),GT(9,9),GSQ(9,9),GSQINV(9,9)
      REAL*8 TDOP,HDOP,PDOP,GDOP,ELEV(9),AZIM(9)
      INTEGER NSATUP(9)
C      REAL*4 GAUS(4)
      REAL*8 GAUS(4)
C
      DO 10 I=1,24
         GAMMA(I)=GZERO(I)+TIME*PI/(21600.D0*0.99726957D0)
10    CONTINUE
C
      DO 20 I=1,24
         XI(I)=RS*DSIN(GAMMA(I))*DSIN(BETA)
         YI(I)=-RS*(DCOS(GAMMA(I))*DSIN(ALPHA(I))+
     1          DSIN(GAMMA(I))*DCOS(ALPHA(I))*DCOS(BETA))
         ZI(I)=RS*(DCOS(ALPHA(I))*DCOS(GAMMA(I))-
     2          DSIN(ALPHA(I))*DSIN(GAMMA(I))*DCOS(BETA))
20    CONTINUE
C
```

```
      PHY=PHYZ + TIME*PI/(43200.D0*0.99726957D0)
C
      DO 30 I=1,24
         XE(I)=XI(I)*DCOS(THETA)+YI(I)*DSIN(THETA)*DSIN(PHY)-
     3          ZI(I)*DSIN(THETA)*DCOS(PHY)
         YE(I)=YI(I)*DCOS(PHY)+ZI(I)*DSIN(PHY)
         ZE(I)=XI(I)*DSIN(THETA)-YI(I)*DCOS(THETA)*DSIN(PHY)+
     4          ZI(I)*DCOS(THETA)*DCOS(PHY)
30    CONTINUE
C
      DO 40 I=1,24
         RHO=DSQRT(XE(I)**2 + YE(I)**2 + (ZE(I) - RE)**2)
         CXALL(I)=XE(I)/RHO
         CYALL(I)=YE(I)/RHO
         CZALL(I)=(ZE(I)-RE)/RHO
40    CONTINUE
C
      DMASK=82.5D0*PI/180.D0
      THRESH=DCOS(DMASK)
      NUMSVS=0
      DO 39 I=1,24
C        STORE SAT. NUMBERS OF SATS. ABOVE 7.5 DEGREES
         IF (CZALL(I).GT.THRESH) THEN
            NUMSVS=NUMSVS + 1
            NSATUP(NUMSVS)=I
         ENDIF
39    CONTINUE
C
C     THIS PROGRAM IS USING ALL-IN-VIEW OPERATION
C
      IF (K.EQ.1) THEN
C        USE NSATUP AND NUMSVS AS THE CORRECT SATELLITE DATA.
         NSATS=NUMSVS
         DO 56 I=1,NSATS
            NSATID(I)=NSATUP(I)
56       CONTINUE
C        FOR K=1 THE SATELITTE SELECTION IS COMPLETE.
         GO TO 55
      ENDIF
```

```
C
43    CONTINUE
C
C     FIRST FIND SATS WHICH WERE USED PREVIOUSLY BUT WHICH
C     ARE NOT VISIBLE (ABOVE MASK ANGLE) AT THE CURRENT STEP.
C     FIND THE LAST SAT. IN NSATID WHICH IS NOT IN NSATUP.
C     ITS LOCATION WILL BE LOCATE IF THERE IS ONE.
C
      LOCATE=NSATS+1
      DO 44 I=1,NSATS
         INSIDE=0
         DO 45 J=1,NUMSVS
            IF (NSATID(I).EQ.NSATUP(J)) INSIDE=1
45       CONTINUE
         IF (INSIDE.EQ.0) LOCATE=I
44    CONTINUE
C
C     IF LOCATE IS LARGER THAN THE NUMBER OF SATS. THEN THERE ARE
C     NO SATS BEING USED WHICH HAVE GONE UNDER THE MASK ANGLE.
C     IF LOCATE IS EQUAL TO NSATS, THEN THE LAST SAT IN NSATID HAS
C     GONE BELOW THE MASK ANGLE.  TO REMOVE THIS SAT. SIMPLY REDUCE
C     NSATS BY ONE.
C
      IF (LOCATE.GE.NSATS) GO TO 48
C
C     OTHERWISE LOCATE POINTS INSIDE NSATID.   IN THIS CASE, REMOVE
C     THE SAT# AND CLOSE THE GAP BY SHIFTING THE OTHER SATS DOWN.
C     ALSO SHIFT THE TRUE SELECTIVE AVAILABILITY PROCESS DOWN SO
C     THAT EACH SAT RETAINS ITS OWN SA PROCESS.
C
      NSATL1=NSATS-1
      DO 46 I=LOCATE,NSATL1
         NSATID(I)=NSATID(I+1)
46    CONTINUE
C
      DO 47 I=LOCATE,NSATL1
         IPOINT=8 + 4*(I-1)
         DO 47 J=1,4
            XTRUEN(IPOINT+J,1)=XTRUEN(IPOINT+J+4,1)
            XTRUED(IPOINT+J,1)=XTRUED(IPOINT+J+4,1)
 47    CONTINUE
 C
 48    CONTINUE
 C
 C     NOW DECREMENT NSATS.  REDEFINE LOCATE SO THAT THE SEARCHING
 C     FOR SATS WHICH HAVE GONE DOWN CONTINUES
 C
       IF (LOCATE.LE.NSATS) THEN
          NSATS=NSATS-1
          LOCATE=NSATS
       ENDIF
 C
 C     IF LOCATE IS STILL LARGER THAN THE NUMBER OF SATS, THEN
 C     NO MORE SATS IN NSATID HAVE GONE DOWN.  IN THIS CASE CONTINUE
 C     ON TO THE NEXT STAGE OF SATELLITE SELECTION.
 C
       IF (LOCATE.GT.NSATS) GO TO 49
       GO TO 43
 C
 49    CONTINUE
 C
 C     IN THIS STAGE, WE FIND SATELLITES WHICH ARE NOW ABOVE THE
 C     MASK ANGLE BUT WERE NOT CONTAINED IN THE PREVIOUS NSATID.
 C
 51    CONTINUE
 C
 C     FIND THE LAST SAT IN NSATUP WHICH IS NOT IN NSATID
 C
       LOCATE=NUMSVS+1
       DO 52 I=1,NUMSVS
          INSIDE=0
          DO 53 J=1,NSATS
             IF (NSATUP(I).EQ.NSATID(J)) INSIDE=1
 53       CONTINUE
          IF (INSIDE.EQ.0) LOCATE=I
 52    CONTINUE
 C
 C     IF LOCATE IS INSIDE NSATUP, THEN ADD THE SAT POINTED TO BY
```

```
C     LOCATE AT THE END OF NSATID AND INCREMENT NSATS.
C     IF LOCATE IS STILL LARGER THAN NUMSVS, THEN NO NEW SAT WAS
C     FOUND. IN THIS CASE, THE SATELLITE SELECTION IS COMPLETE.
C
      IF (LOCATE.LE.NUMSVS) THEN
         NSATS=NSATS+1
         NSATID(NSATS)=NSATUP(LOCATE)
      ELSE
         GO TO 55
      ENDIF
C
C     WHEN ADDING A NEW SATELLITE TO NSATID, ALSO FORM AN INITIAL
C     CONDITION FOR THE NEW SELECTIVE AVAILABILITY PROCESS BY
C     CALLING THE RANDOM NUMBER GENERATOR.
C
      IF (NOSA.NE.1) THEN
C         CALL GGNPM(DSEED,4,GAUS)
         CALL DRNNOA(4,GAUS)
C
         ISTART=8+4*(NSATS-1)
         DO 54 I=1,4
            XTRUEN(I+ISTART,1)=GAUS(I)*SASCFT(I)
            XTRUED(I+ISTART,1)=GAUS(I)*SASCFT(I)
54       CONTINUE
      ENDIF
C
      GO TO 51
C
55    CONTINUE
C
C     DEFINE THE MATRIX DIMENSIONS RELATED TO THE # OF SATELLITES.
C
      NRZ=NSATS
      IF (NOSA.NE.1) NRXT=8 + 4*NSATS
C
C     ZERO OUT THE UNUSED PORTION OF XTRUE SO THAT ONLY PROCESSED
C     THAT ARE USED WILL APPEAR AS NON-ZERO WHEN PLOTTED.
C
      NRXTP1=NRXT + 1
      DO 58 I=NRXTP1,MRXT
         XTRUEN(I,1)=0.D0
         XTRUED(I,1)=0.D0
58    CONTINUE
C
C     NOW LOAD THE DIRECTION COSINE VECTORS WITH ONLY THOSE
C     WHICH ARE TO BE USED.
C
      DO 57 I=1,NSATS
         CX(I)=CXALL(NSATID(I))
         CY(I)=CYALL(NSATID(I))
         CZ(I)=CZALL(NSATID(I))
57    CONTINUE
C
C     LOAD UP THE H AND HTRUE MATRICES
C
      DO 50 I=1,NSATS
         H(I,1)=-CX(I)
         H(I,3)=-CY(I)
         H(I,5)=-CZ(I)
         HTRUE(I,1)=-CX(I)
         HTRUE(I,3)=-CY(I)
         HTRUE(I,5)=-CZ(I)
50    CONTINUE
C
C     THE REST OF THIS SUBROUTINE IS FOR GDOP CALCULATIONS
C     AND ELEVATION AND AZIMUTH ANGLE CALCULATIONS.
C
      KNGEOM=K-NGEOM
      IF ((K.GE.NGEOM).AND.(MOD(KNGEOM,NGMSZ).EQ.0)) THEN
C
         DO 60 I=1,NSATS
            G(I,1)=CX(I)
            G(I,2)=CY(I)
            G(I,3)=CZ(I)
            G(I,4)=1.D0
60       CONTINUE
C
         CALL TRANSP(9,GT,4,NSATS,G,NSATS,4)
         CALL MULT(9,1,GSQ,0,GT,4,NSATS,G,NSATS,4)
         CALL INVERT(9,1,GSQINV,4,GSQ)
C
         ARG1=GSQINV(4,4)
         ARG2=(GSQINV(1,1)+GSQINV(2,2))
         ARG3=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3))
         ARG4=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3)+GSQINV(4,4))
         IF (ARG1.GT.0.0) THEN
            TDOP=DSQRT(ARG1)
         ELSE
            TDOP=1.D40
         ENDIF
C
         IF (ARG2.GT.0.0) THEN
            HDOP=DSQRT(ARG2)
         ELSE
            HDOP=1.D40
         ENDIF
C
         IF (ARG3.GT.0.0) THEN
            PDOP=DSQRT(ARG3)
         ELSE
            PDOP=1.D40
         ENDIF
C
         IF (ARG4.GT.0.0) THEN
            GDOP=DSQRT(ARG4)
         ELSE
            GDOP=1.D40
         ENDIF
C
68       FORMAT(/1X,20('*'),' CURRENT SATELLITE GEOMETRY ',20('*'))
61       FORMAT(1X,'STEP',I5,1X,'TDOP=',F8.3,2X,'HDOP=',F8.3,
     1           2X,'PDOP=',F8.3,2X,'GDOP=',F8.3)
62       FORMAT(1X,'SAT#',9(4X,I2,3X))
63       FORMAT(1X,'ELEV',9(F8.2,1X))
64       FORMAT(1X,'AZIM',9(F8.2,1X))
C
C        FIND THE ELEVATION AND AZIMUTH ANGLES FOR EACH SAT.
```

```
C
         WRITE(6,68)
         WRITE(6,61)K,TDOP,HDOP,PDOP,GDOP
         DO 42 J=1,NSATS
            I=NSATID(J)
            ANG=(180.D0/PI)*DARCOS(CZALL(I))
            ELEV(J)=90.D0-ANG
            IF (DABS(XE(I)).GT.1.0D-30) THEN
               ANG=(180.D0/PI)*DATAN(YE(I)/XE(I))
               IF ((XE(I).GT.0.D0).AND.(YE(I).GT.0.D0)) THEN
                  AZIM(J)=360.D0-ANG
               ELSE IF ((XE(I).GT.0.D0).AND.(YE(I).LT.0.D0)) THEN
                  AZIM(J)=-ANG
               ELSE IF ((XE(I).LT.0.D0).AND.(YE(I).LT.0.D0)) THEN
                  AZIM(J)=180.D0 - ANG
               ELSE
                  AZIM(J)=180.D0 - ANG
               ENDIF
            ELSE
               IF (YE(I).GT.0.D0) THEN
                  AZIM(J)=270.D0
               ELSE
                  AZIM(J)=90.D0
               ENDIF
            ENDIF
42       CONTINUE
C
         WRITE(6,62)(NSATID(I),I=1,NSATS)
         WRITE(6,63)(ELEV(I),I=1,NSATS)
         WRITE(6,64)(AZIM(I),I=1,NSATS)
C
C        NOW OBTAIN THE GDOPS FOR EACH SUB-SOLUTION
C
         HDPMAX=0.D0
         NMAX=1
         DO 106 I=1,NSATS
            KK=0
            NS1=NSATS-1
            DO 206 J=1,NS1
            IF (J.EQ.I) THEN
              KK=J+1
            ELSE
              KK=KK+1
            ENDIF
            G(J,1)=CX(KK)
            G(J,2)=CY(KK)
            G(J,3)=CZ(KK)
            G(J,4)=1.0D0
206     CONTINUE
        CALL TRANSP(9,GT,4,NS1,G,NS1,4)
        CALL MULT(9,1,GSQ,0,GT,4,NS1,G,NS1,4)
        CALL INVERT(9,0,GSQINV,4,GSQ)
        ARG1=GSQINV(4,4)
        ARG2=(GSQINV(1,1)+GSQINV(2,2))
        ARG3=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3))
        ARG4=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3)+GSQINV(4,4))
C
        IF (ARG1.GT.0.0) THEN
          TDOP=DSQRT(ARG1)
        ELSE
          TDOP=1.D40
        ENDIF
C
        IF (ARG2.GT.0.0) THEN
          HDOP=DSQRT(ARG2)
        ELSE
          HDOP=1.D40
        ENDIF
C
        IF (ARG3.GT.0.0) THEN
          PDOP=DSQRT(ARG3)
        ELSE
          PDOP=1.D40
        ENDIF
C
        IF (ARG4.GT.0.0) THEN
          GDOP=DSQRT(ARG4)
        ELSE
```

```
               GDOP=1.D40
            ENDIF
C
            IF (HDOP.GT.HDPMAX) THEN
               HDPMAX=HDOP
               NMAX=I
            ENDIF
C
201         FORMAT(1X,'SUB SOL#',I1,1X,'TDOP=',F8.3,2X,'HDOP=',F8.3,
     1                 2X,'PDOP=',F8.3,2X,'GDOP=',F8.3)
C
            WRITE(6,201)I,TDOP,HDOP,PDOP,GDOP
106      CONTINUE
C
402      FORMAT(1X,'EXPLICIT CLOCK COASTING IS STARTED AT STEP=',I5)
403      FORMAT(1X,'EXPLICIT CLOCK COASTING IS STOPPED AT STEP=',I5)
C
         IF ((HDPMAX.GT.HDPLMT).AND.(NCOAST.EQ.0)) THEN
            NCOAST=1
            WRITE(6,402)K
            CALL CLKCON(K)
         ENDIF
C
         IF ((NCOAST.EQ.1).AND.(HDPMAX.LT.HDPLMT)) THEN
            NCOAST=0
            WRITE(6,403)K
            CALL CLKCOF(K)
         ENDIF
C
         IF (NCOAST.EQ.1) THEN
C
C           OBTAIN THE DOPS FOR MASTER SOLUTION WITH CLOCK COASTING
C
            DO 600 I=1,NSATS
               G(I,1)=CX(I)
               G(I,2)=CY(I)
               G(I,3)=CZ(I)
               G(I,4)=1.D0
600         CONTINUE
```

```
            NSATS1=NSATS + 1
            G(NSATS1,1)=0.D0
            G(NSATS1,2)=0.D0
            G(NSATS1,3)=0.D0
            G(NSATS1,4)=1.D0
            CALL TRANSP(9,GT,4,NSATS1,G,NSATS1,4)
            CALL MULT(9,1,GSQ,0,GT,4,NSATS1,G,NSATS1,4)
            CALL INVERT(9,1,GSQINV,4,GSQ)
C
            ARG1=GSQINV(4,4)
            ARG2=(GSQINV(1,1)+GSQINV(2,2))
            ARG3=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3))
            ARG4=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3)+GSQINV(4,4))
            IF (ARG1.GT.0.0) THEN
               TDOP=DSQRT(ARG1)
            ELSE
               TDOP=1.D40
            ENDIF
C
            IF (ARG2.GT.0.0) THEN
               HDOP=DSQRT(ARG2)
            ELSE
               HDOP=1.D40
            ENDIF
C
            IF (ARG3.GT.0.0) THEN
               PDOP=DSQRT(ARG3)
            ELSE
               PDOP=1.D40
            ENDIF
C
            IF (ARG4.GT.0.0) THEN
               GDOP=DSQRT(ARG4)
            ELSE
               GDOP=1.D40
            ENDIF
C
601         FORMAT(1X,’DOPS FOR ALL SATS PLUS CLOCK MEASUREMENT’)
            WRITE(6,601)
            WRITE(6,61)K,TDOP,HDOP,PDOP,GDOP
C
C           OBTAIN GDOPS FOR EACH SUB-SOLUTION WITH CLOCK COASTING
C
404         FORMAT(1X,'DOPS FOR SUB SOLUTIONS WITH CLOCK COASTING')
            WRITE(6,404)
C
            DO 506 I=1,NSATS
               KK=0
               NS1=NSATS-1
               DO 606 J=1,NS1
                     IF (J.EQ.I) THEN
                        KK=J+1
                     ELSE
                        KK=KK+1
                     ENDIF
                     G(J,1)=CX(KK)
                     G(J,2)=CY(KK)
                     G(J,3)=CZ(KK)
                     G(J,4)=1.0D0
606            CONTINUE
               G(NSATS,1)=0.D0
               G(NSATS,2)=0.D0
               G(NSATS,3)=0.D0
               G(NSATS,4)=1.D0
               CALL TRANSP(9,GT,4,NSATS,G,NSATS,4)
               CALL MULT(9,1,GSQ,0,GT,4,NSATS,G,NSATS,4)
               CALL INVERT(9,0,GSQINV,4,GSQ)
               ARG1=GSQINV(4,4)
               ARG2=(GSQINV(1,1)+GSQINV(2,2))
               ARG3=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3))
               ARG4=(GSQINV(1,1)+GSQINV(2,2)+GSQINV(3,3)+GSQINV(4,4))
               IF (ARG1.GT.0.0) THEN
                  TDOP=DSQRT(ARG1)
               ELSE
                  TDOP=1.D40
               ENDIF
C
               IF (ARG2.GT.0.0) THEN
```

```
                  HDOP=DSQRT(ARG2)
               ELSE
                  HDOP=1.D40
               ENDIF
C
               IF (ARG3.GT.0.0) THEN
                  PDOP=DSQRT(ARG3)
               ELSE
                  PDOP=1.D40
               ENDIF
C
               IF (ARG4.GT.0.0) THEN
                  GDOP=DSQRT(ARG4)
               ELSE
                  GDOP=1.D40
               ENDIF
C
               WRITE(6,201)I,TDOP,HDOP,PDOP,GDOP
506      CONTINUE
C
         ENDIF
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE MEASUR(K)
C
C     THIS SUBROUTINE TAKES THE TRUE STATE AND USES THE
C     THE TRUE KALMAN FILTER MEASUREMENT EQUATION TO GENERATE
C     THE CURRENT MEASUREMENT Z.  IT ALSO HAS THE ABILITY TO
C     CORRUPT EACH MEASUREMENT WITH A RAMP ERROR.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      INTEGER K
      REAL*8 HX(44,1)
C
C     THE CURRENT V NOISE VECTOR WILL BE GRABBED FROM
C     THE VNOIS ARRAY.
C
      IF (NONOIS.EQ.1) THEN
         CALL SUM(MRZ,0,V,NRZ,1,NULL,NULL)
      ELSE
         DO 20 I=1,NRZ
```

```
            V(I,1)=VNOIS(I,K)
20       CONTINUE
      ENDIF
C
      IF ((K.GE.NERRON).AND.(K.LT.NERROF)) THEN
         RPTIME=DELTAT*FLOAT(K-NERRON)
         DO 21 I=1,NRZ
            V(I,1)=V(I,1) + SLOPE(I)*RPTIME + BIAS(I)
21       CONTINUE
      ENDIF
C
      CALL MULT(MRXT,0,HX,0,HTRUE,NRZ,NRXT,XTRUEN,NRXT,1)
C
C     NOW ADD THE MEASUREMENT NOISE TO THE TRUE NAV. MEASUREMENT
C
      DO 30 I=1,NRZ
         ZN(I,1)=HX(I,1) + V(I,1)
30    CONTINUE
C
      CALL MULT(MRXT,0,HX,0,HTRUE,NRZ,NRXT,XTRUED,NRXT,1)
C
C     NOW ADD THE MEASUREMENT NOISE TO THE TRUE DET. MEASUREMENT
C
      DO 40 I=1,NRZ
         ZD(I,1)=HX(I,1) + V(I,1)
40    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE UPDATE(K)
C
C     THIS SUBROUTINE ACCEPTS THE CURRENT MEASUREMENT,
C     AND USES IT TO UPDATE THE ESTIMATE OF THE STATE VECTOR.
C     THE ERROR COVARIANCE IS ALSO UPDATED HERE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
```

```
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C5/ XACTN,XACTD,XETRAJ,PTRAJ,XDTRAJ,PDTRAJ
      REAL*8 XACTN(44,1200),XACTD(8,1200),XETRAJ(8,1200)
      REAL*8 PTRAJ(8,1200),XDTRAJ(8,1200),PDTRAJ(8,1200)
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
```

```
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      INTEGER K
      REAL*8 HTRAN(9,9),PH(9,9)
      REAL*8 GAIN(9,9),HPHR(9,9)
      REAL*8 HPHRIN(9,9),RESID(9,1),RESTRA(9,9)
      REAL*8 ZHPHR(9,9)
      REAL*8 ZHPHRZ(9,1),HX(9,1),XCORR(9,1)
      REAL*8 ANEW(9,9),CNEW(9,9),AOLD(9,9)
      REAL*8 KH(9,9),CCOEF(9,9)
      REAL*8 IMINKH(9,9),IMKHT(9,9),ACOEF(9,9)
      REAL*8 HPHI(9,9),DNEW(9,9),BOLD(9,9),BNEW(9,9)
      REAL*8 HYTRAN(9,9),PYHY(9,9),HYPY(9,2),HYPYHY(9,9),PXYHYT(9,9)
      REAL*8 HPXYHY(9,9),HYPYXH(9,9),HYPYX(9,9),PXYHYK(9,9)
      REAL*8 KHYPY(9,2),HYYHAT(9,1),GAINT(9,9)
      REAL*8 HERRN(20),HERRD(20)
      REAL*8 CHECK(9,9)
C
C     FIRST OBTAIN THE OPTIMAL GAIN FOR THE NAVIGATION FILTER
C
      CALL TRANSP(MRXH,HTRAN,NRXH,NRZ,H,NRZ,NRXH)
      CALL MULT(MRXH,0,PH,1,P,NRXH,NRXH,HTRAN,NRXH,NRZ)
      CALL MULT(MRXH,1,HPHR,0,H,NRZ,NRXH,PH,NRXH,NRZ)
      DO 10 I=1,NRZ
         HPHR(I,I)=HPHR(I,I)+RF
10    CONTINUE
      CALL INVERT(MRXH,1,HPHRIN,NRZ,HPHR)
      CALL MULT(MRXH,0,GAIN,0,PH,NRXH,NRZ,HPHRIN,NRZ,NRZ)
C
C     UPDATE THE NAVIGATION FILTER COVARIANCE
C
      CALL MULT(MRXH,0.KH,0,GAIN,NRXH,NRZ,H,NRZ,NRXH)
      CALL DIFF(MRXH,0,IMINKH,NRXH,NRXH,IDENT,KH)
      CALL TRANSP(MRXH,IMKHT,NRXH,NRXH,IMINKH,NRXH,NRXH)
      CALL SCALEP(MRXH,P,NRXH,IMKHT)
C
C     UPDATE THE NAVIGATION FILTER STATE ESTIMATE
C
      CALL MULT(MRXH,0,HX,0,H,NRZ,NRXH,XHAT,NRXH,1)
      CALL DIFF(MRXH,0,RESID,NRZ,1,ZN,HX)
      CALL MULT(MRXH,0,XCORR,0,GAIN,NRXH,NRZ,RESID,NRZ,1)
      CALL ADDON(MRXH,XHAT,NRXH,1,XCORR)
C
C     NOW UPDATE THE CONNECTION MATRICES FOR EACH S AND UO
C
      IF (K.GE.NHYPON) THEN
C
         CALL MULT(MRXH,0,ACOEF,0,IMINKH,NRXH,NRXH,PHI,NRXH,NRXH)
C
         DO 210 NW=1,NWINDW
            IF (NW.EQ.NWINDW) THEN
               CALL DIFF(MRXH,0,ANEW,NRXH,NRZ,NULL,GAIN)
            ELSE
               NWLAST=NWINDW-1
               DO 211 NC=1,NRZ
                  DO 211 NR=1,NRXH
                     AOLD(NR,NC)=AT(NR,NC,NW,NWLAST)
211            CONTINUE
               CALL MULT(MRXH,0,ANEW,0,ACOEF,NRXH,NRXH,AOLD,NRXH,NRZ)
            ENDIF
C
            DO 212 NC=1,NRZ
               DO 212 NR=1,NRXH
                  AT(NR,NC,NW,NWINDW)=ANEW(NR,NC)
212         CONTINUE
C
            IF (NW.EQ.NWINDW) THEN
               IF (NW.EQ.1) THEN
                  CALL SUM(MRXH,0,BNEW,NRXH,NRXH,ACOEF,NULL)
               ELSE
                  NWLAST=NW-1
                  DO 215 NC=1,NRXH
                     DO 215 NR=1,NRXH
```

```
                        BOLD(NR,NC)=BT(NR,NC,NWLAST)
215               CONTINUE
                  CALL MULT(MRXH,0,BNEW,0,ACOEF,NRXH,NRXH,
     1                      BOLD,NRXH,NRXH)
               ENDIF
               DO 216 NC=1,NRXH
                  DO 216 NR=1,NRXH
                     BT(NR,NC,NW)=BNEW(NR,NC)
216            CONTINUE
            ENDIF
C
210      CONTINUE
C
      ENDIF
C
C     NOW OBTAIN THE OPTIMAL GAIN FOR THE DETECTION FILTER
C
      CALL MULT(MRXH,0,PH,1,PDET,NRXD,NRXD,HTRAN,NRXD,NRZ)
      CALL MULT(MRXH,1,HPHR,0,H,NRZ,NRXD,PH,NRXD,NRZ)
      DO 20 I=1,NRZ
         HPHR(I,I)=HPHR(I,I)+RF
20    CONTINUE
C
      IF (NCOAST.EQ.1) THEN
         CALL TRANSP(MRXH,HYTRAN,2,NRZ,HY,NRZ,2)
         CALL MULT(MRXH,0,PYHY,1,PY,2,2,HYTRAN,2,NRZ)
         CALL MULT(MRXH,1,HYPYHY,0,HY,NRZ,2,PYHY,2,NRZ)
         CALL MULT(MRXH,0,PXYHYT,0,PXY,NRXD,2,HYTRAN,2,NRZ)
         CALL MULT(MRXH,0,HPXYHY,0,H,NRZ,NRXD,PXYHYT,NRXD,NRZ)
         CALL TRANSP(MRXH,HYPYXH,NRZ,NRZ,HPXYHY,NRZ,NRZ)
         CALL ADDON(MRXH,HPHR,NRZ,NRZ,HYPYHY)
         CALL ADDON(MRXH,HPHR,NRZ,NRZ,HYPYXH)
         CALL ADDON(MRXH,HPHR,NRZ,NRZ,HPXYHY)
         CALL ADDON(MRXH,PH,NRXD,NRZ,PXYHYT)
      ENDIF
C
      CALL INVERT(MRXH,1,HPHRIN,NRZ,HPHR)
C
      CALL MULT(MRXH,0,GAIN,0,PH,NRXD,NRZ,HPHRIN,NRZ,NRZ)
```

```
C
C     UPDATE THE DETECTION FILTER COVARIANCE
C
      CALL MULT(MRXH,0,KH,0,GAIN,NRXD,NRZ,H,NRZ,NRXD)
      CALL DIFF(MRXH,0,IMINKH,NRXD,NRXD,IDENT,KH)
      CALL TRANSP(MRXH,IMKHT,NRXD,NRXD,IMINKH,NRXD,NRXD)
      CALL SCALEP(MRXH,PDET,NRXD,IMKHT)
C
      IF (NCOAST.EQ.1) THEN
C
C        SUBTRACT OUT THE CORRELATION OF PXY
C
         CALL TRANSP(MRXH,GAINT,NRZ,NRXD,GAIN,NRXD,NRZ)
         CALL MULT(MRXH,0,PXYHYK,0,PXYHYT,NRXD,NRZ,GAINT,NRZ,NRXD)
         CALL HSUBOT(MRXH,PDET,NRXD,NRXD,PXYHYK)
C
C        ALSO UPDATE PXY
C
         CALL MULTBY(MRXH,IMINKH,NRXD,NRXD,PXY,NRXD,2)
         CALL TRANSP(MRXH,HYPY,NRZ,2,PYHY,2,NRZ)
         CALL MULT(MRXH,0,KHYPY,0,GAIN,NRXD,NRZ,HYPY,NRZ,2)
         CALL SUBOUT(MRXH,PXY,NRXD,2,KHYPY)
C
      ENDIF
C
C     UPDATE THE DETECTION FILTER STATE ESTIMATE
C
      CALL MULT(MRXH,0,HX,0,H,NRZ,NRXD,XDET,NRXD,1)
      IF (NCOAST.EQ.1) THEN
         CALL MULT(MRXH,0,HYYHAT,0,HY,NRZ,2,YHAT,2,1)
         CALL ADDON(MRXH,HX,NRZ,1,HYYHAT)
      ENDIF
      CALL DIFF(MRXH,0,RESID,NRZ,1,ZD,HX)
      CALL MULT(MRXH,0,XCORR,0,GAIN,NRXD,NRZ,RESID,NRZ,1)
      CALL ADDON(MRXH,XDET,NRXD,1,XCORR)
C
C     SAVE THE VECTOR OF RESIDUALS FROM THE DETECTION FILTER
C
      DO 30 M=1,NRZ
```

```
         VSAVE(M,K)=RESID(M,1)
30    CONTINUE
C
C     ZERO OUT THE UNUSED PORTION OF VSAVE
C
      NRZP1=NRZ+1
      DO 40 M=NRZP1,MRZ
         VSAVE(M,K)=0.D0
40    CONTINUE
C
      IF (K.GE.NHYPON) THEN
C
C        SAVE THE HPHRIV MATRIX FROM THE DETECTION FILTER
C
         DO 130 NC=1,NRZ
            DO 130 NR=1,NRZ
               VARINV(NR,NC,NWINDW)=HPHRIN(NR,NC)
130      CONTINUE
C
C        SAVE THE MAIN DIAGONAL OF THE RESIDUAL COVARIANCE MATRIX
C
         DO 140 NR=1,NRZ
            RESVAR(NR,NWINDW)=HPHR(NR,NR)
140      CONTINUE
C
C        SAVE THE CORRELATION BETWEEN X1 AND X3 FROM THE NAV FILTER
C
         PWOFFN(NWINDW)=P(1,3)
         PWOFFD(NWINDW)=PDET(1,3)
C
      ENDIF
C
      IF (K.GE.NHYPON) THEN
C
C        SAVE THE DEGREES OF FREEDOM AT EACH STEP IN NWINDW
C
         IF (NCOAST.EQ.1) THEN
            NDFSTP(NWINDW)=NSATS
         ELSE
```

```
            NDFSTP(NWINDW)=NSATS
         ENDIF
C
C        SAVE THE NUMBER OF SATELLITES AT EACH STEP IN NWINDW
C
         NSATW(NWINDW)=NSATS
C
C        SAVE THE VECTOR OF RESIDUALS FOR CURRENT WINDOW
C
         DO 51 M=1,NRZ
            VBLK(M,NWINDW)=RESID(M,1)
51       CONTINUE
C
         CALL MULT(MRXH,0,ACOEF,0,IMINKH,NRXD,NRXD,PHID,NRXD,NRXD)
         CALL MULT(MRXH,0,HPHI,0,H,NRZ,NRXD,PHID,NRXD,NRXD)
         CALL DIFF(MRXH,0,CCOEF,NRZ,NRXD,NULL,HPHI)
C
         DO 110 NW=1,NWINDW
            IF (NW.EQ.NWINDW) THEN
               CALL DIFF(MRXH,0,ANEW,NRXD,NRZ,NULL,GAIN)
               CALL DIFF(MRXH,0,CNEW,NRZ,NRZ,NULL,IDENT)
            ELSE
               NWLAST=NWINDW-1
               DO 111 NC=1,NRZ
                  DO 111 NR=1,NRXD
                     AOLD(NR,NC)=A(NR,NC,NW,NWLAST)
111            CONTINUE
               CALL MULT(MRXH,0,ANEW,0,ACOEF,NRXD,NRXD,AOLD,NRXD,NRZ)
               CALL MULT(MRXH,0,CNEW,0,CCOEF,NRZ,NRXD,AOLD,NRXD,NRZ)
            ENDIF
C
            DO 112 NC=1,NRZ
               DO 112 NR=1,NRXD
                  A(NR,NC,NW,NWINDW)=ANEW(NR,NC)
112         CONTINUE
C
            DO 113 NC=1,NRZ
               DO 113 NR=1,NRZ
                  C(NR,NC,NW,NWINDW)=CNEW(NR,NC)
```

```
113         CONTINUE
C
            IF (NW.EQ.NWINDW) THEN
               IF (NW.EQ.1) THEN
                  CALL SUM(MRXH,0,DNEW,NRZ,NRXD,CCOEF,NULL)
                  CALL SUM(MRXH,0,BNEW,NRXD,NRXD,ACOEF,NULL)
               ELSE
                  NWLAST=NW-1
                  DO 115 NC=1,NRXD
                     DO 115 NR=1,NRXD
                        BOLD(NR,NC)=B(NR,NC,NWLAST)
115               CONTINUE
                  CALL MULT(MRXH,0,DNEW,0,CCOEF,NRZ,NRXD,
     1                      BOLD,NRXD,NRXD)
                  CALL MULT(MRXH,0,BNEW,0,ACOEF,NRXD,NRXD,
     1                      BOLD,NRXD,NRXD)
               ENDIF
               DO 116 NC=1,NRXD
                  DO 116 NR=1,NRXD
                     B(NR,NC,NW)=BNEW(NR,NC)
116            CONTINUE
               DO 114 NC=1,NRXD
                  DO 114 NR=1,NRZ
                     D(NR,NC,NW)=DNEW(NR,NC)
114            CONTINUE
            ENDIF
C
110      CONTINUE
C
      ENDIF
C
C     UPDATE CHISQ
C
      IF (K.GE.NC2ON) THEN
C
C        PERFORM SOME INITIALIZATION AT THE START OF THE
C        FIRST WINDOW.
C
         IF (K.EQ.NC2ON) THEN
```

```
            NC2INC=1
            NDOF(NC2INC)=0
            CHI(NC2INC)=0.D0
         ENDIF
C
C        FORM THE CHI-SQUARED STATISTIC AND ADD IT TO THE
C        THE SUM FOR THE CURRENT WINDOW.  ALSO INCREMENT
C        THE DEGREES OF FREEDOM.
C
         CALL TRANSP(MRXH,RESTRA,1,NRZ,RESID,NRZ,1)
         CALL MULT(MRXH,0,ZHPHR,0,RESTRA,1,NRZ,HPHRIN,NRZ,NRZ)
         CALL MULT(MRXH,0,ZHPHRZ,0,ZHPHR,1,NRZ,RESID,NRZ,1)
         CHI(NC2INC)=CHI(NC2INC) + ZHPHRZ(1,1)
C
         IF (NCOAST.EQ.1) THEN
            NDOF(NC2INC)=NDOF(NC2INC) + NRZ
         ELSE
            NDOF(NC2INC)=NDOF(NC2INC) + NRZ
         ENDIF
C
         KSTEP=K-NC2ON+1
C
         IF ((K.GT.NC2ON).AND.(MOD(KSTEP,NC2SZ).EQ.0)) THEN
C
C           THIS IS THE END OF THE CURRENT WINDOW.  AFTER GETTING
C           THE FINAL STATISTIC AND SAVING IT, PERFORM SOME
C           INITIALIZATION FOR THE START OF THE NEXT WINDOW.
C
            WRITE(6,101)K
            WRITE(6,102)CHI(NC2INC),NDOF(NC2INC),
     1                  CHITBL(NDOF(NC2INC),LOCSIZ(1))
            LOCATE=K-NC2SZ
            JSTOP=NC2SZ-1
            DO 41 J=1,JSTOP
               JSTEP=LOCATE + J
               HERRN(J)=DSQRT((XACTN(1,JSTEP)-XETRAJ(1,JSTEP))**2
     1                + (XACTN(3,JSTEP)-XETRAJ(3,JSTEP))**2)
               HERRD(J)=DSQRT((XACTD(1,JSTEP)-XDTRAJ(1,JSTEP))**2
     1                + (XACTD(3,JSTEP)-XDTRAJ(3,JSTEP))**2)
```

```
41            CONTINUE
              HERRN(NC2SZ)=DSQRT((XTRUEN(1,1)-XHAT(1,1))**2
     1                   + (XTRUEN(3,1)-XHAT(3,1))**2)
              HERRD(NC2SZ)=DSQRT((XTRUED(1,1)-XDET(1,1))**2
     1                   + (XTRUED(3,1)-XDET(3,1))**2)
              WRITE(6,103)(HERRN(J),J=1,NC2SZ)
              WRITE(6,104)(HERRD(J),J=1,NC2SZ)
C
              NC2INC=NC2INC + 1
              CHI(NC2INC)=0.D0
              NDOF(NC2INC)=0
            ENDIF
          ENDIF
C
C
101   FORMAT(/1X,15('*'),'DET. FILTER  RESIDUAL TESTS AT STEP K=',
     1        I6,1X,15('*'))
102   FORMAT(1X,'CHI-SQR=',F9.2,1X,'DOF=',I4,
     1        3X,'THRESH=',F9.3)
103   FORMAT(1X,'NAV RAD ERR:',11(1X,F9.1))
104   FORMAT(1X,'DET RAD ERR:',11(1X,F9.1))
C
      RETURN
      END
C
      SUBROUTINE PROJEC(K)
C
C     THE CURRENT ESTIMATE OF THE STATE VECTOR AND THE CURRENT
C     ERROR COVARIANCE ARE PROJECTED AHEAD TO THE NEXT TIME
C     STEP TO BE USED AS A PROIRI ESTIMATES ON THE NEXT STEP.
C     THE TRUE PROCESS XTRUE IS ALSO PROJECTED HERE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1       ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C14/ PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY,NTURON,NTUROF
      REAL*8 PERIOD,BNKANG,ACCEL,RADFRQ,VELCTY
      INTEGER NTURON,NTUROF
C
      INTEGER K
C
      REAL*8 PTM(9,9),TMPTM(9,9)
      REAL*8 PHIXT(44,1),TURN(44,1)
C
C     PROJECT XTRUE
C
      IF (NONOIS.EQ.1) THEN
         CALL SUM(MRXT,0,W,NRXT,1,NULLB,NULLB)
      ELSE
         DO 10 I=1,NRXT
            W(I,1)=WNOIS(I,K)
10       CONTINUE
```

```
      ENDIF
C
      CALL MULT(MRXT,0,PHIXT,0,PHITRU,NRXT,NRXT,XTRUEN,NRXT,1)
      CALL SUM(MRXT,0,XTRUEN,NRXT,1,PHIXT,W)
C
      CALL MULT(MRXT,0,PHIXT,0,PHITRU,NRXT,NRXT,XTRUED,NRXT,1)
      CALL SUM(MRXT,0,XTRUED,NRXT,1,PHIXT,W)
C
      IF ((NTURON.EQ.1).AND.(K.LT.NTUROF)) THEN
C
C        NOW ADD TO THE TRUE VECTORS THE CONTRIBUTION OF THE TURN
C
         CALL SUM(MRXT,0,TURN,NRXT,1,NULLB,NULLB)
C
         ARGTK=TIME*RADFRQ
         ACDW=ACCEL/RADFRQ
         ACDWSQ=ACDW/RADFRQ
         WDELT=RADFRQ*DELTAT
         SCALE1=DSIN(WDELT) - WDELT
         SCALE2=1 - DCOS(WDELT)
         POSSF1=ACDWSQ*SCALE1
         POSSF2=ACCEL*SCALE2
         VELSF1=ACDW*SCALE2
         VELSF2=ACDW*DSIN(WDELT)
C
         TURN(1,1)=POSSF1*DSIN(ARGTK)+POSSF2*DCOS(ARGTK)
         TURN(2,1)=-VELSF1*DSIN(ARGTK)+VELSF2*DCOS(ARGTK)
         TURN(3,1)=-POSSF1*DCOS(ARGTK)+POSSF2*DSIN(ARGTK)
         TURN(4,1)=VELSF1*DCOS(ARGTK)+VELSF2*DSIN(ARGTK)
C
         CALL ADDON(MRXT,XTRUEN,NRXT,1,TURN)
         CALL ADDON(MRXT,XTRUED,NRXT,1,TURN)
C
      ENDIF
C
C     PROJECT THE NAVIAGION FILTER ESTIMATE XHAT
C
      CALL MULTBY(MRXH,PHI,NRXH,NRXH,XHAT,NRXH,1)
C
```

```
C     PROJECT THE NAVIGATION FILTER COVARIANCE MATRIX
C
      CALL MULT(MRXH,0,PTM,1,P,NRXH,NRXH,PHITR,NRXH,NRXH)
      CALL MULT(MRXH,0,TMPTM,0,PHI,NRXH,NRXH,PTM,NRXH,NRXH)
      CALL SUMHAF(MRXH,P,NRXH,NRXH,TMPTM,Q)
C
C     PROJECT DETECTION FILTER ESTIMATE XDET
C
      CALL MULTBY(MRXH,PHID,NRXD,NRXD,XDET,NRXD,1)
C
C     PROJECT THE DETECTION FILTER COVARIANCE MATRIX
C
      CALL MULT(MRXH,0,PTM,1,PDET,NRXD,NRXD,PHIDTR,NRXD,NRXD)
      CALL MULT(MRXH,0,TMPTM,0,PHID,NRXD,NRXD,PTM,NRXD,NRXD)
      CALL SUMHAF(MRXH,PDET,NRXD,NRXD,TMPTM,QDET)
C
      IF (NCOAST.EQ.1) THEN
C
C        PROJECT YHAT
C
         CALL MULTBY(MRXH,PHIY,2,2,YHAT,2,1)
C
C        PROJECT PY
C
         CALL MULT(MRXH,0,PTM,1,PY,2,2,PHIYT,2,2)
         CALL MULT(MRXH,0,TMPTM,0,PHIY,2,2,PTM,2,2)
         CALL SUMHAF(MRXH,PY,2,2,TMPTM,QY)
C
C        PROJECT PXY
C
         CALL MULT(MRXH,0,PTM,0,PXY,NRXD,2,PHIYT,2,2)
         CALL MULT(MRXH,0,PXY,0,PHID,NRXD,NRXD,PTM,NRXD,2)
C
      ENDIF
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCC                                                       CCCCC
CCCCC          END OF KALMAN FILTER SUBROUTINES             CCCCC
CCCCC                                                       CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                       CCCCC
CCCCC     NOW BEGIN THE CENSORED FILTER SUBROUTINES.        CCCCC
CCCCC           THEY ARE: GETCHI,DECIDE,PREDCT              CCCCC
CCCCC              NEWVAU,NEWVAS,NEWU,NEWUT                 CCCCC
CCCCC             HISTRY,HYPOTH,ELLIPS,LABEL                CCCCC
CCCCC                                                       CCCCC
CCCCC                                                       CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE GETCHI(NSTART)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      INTEGER NSTART
C
      REAL*8 HPHRIN(9,9),RESID(9,1),RESTRA(9,9)
      REAL*8 ZHPHR(9,9),ZHPHRZ(9,1)
C
```

```
C     FIRST ZERO OUT CHISAT VECTOR
C
      DO 1 NR=1,NRZ
         SATCHI(NR)=0.D0
1     CONTINUE
C
      DO 10 NSTP=NSTART,NWSIZE
C
C        LOAD UP THE INVERSE OF THE RESIDUAL COVARIANCE
C
         DO 20 NC=1,NRZ
            DO 20 NR=1,NRZ
               HPHRIN(NR,NC)=VARINV(NR,NC,NSTP)
20       CONTINUE
C
C        NOW GRAB A COLUMN FROM VBLKA
C
         DO 30 NR=1,NRZ
            RESID(NR,1)=VBLKA(NR,NSTP)
30       CONTINUE
C
C        FORM THE CHI-SQUARE STAT FOR EACH STEP
C
         CALL TRANSP(MRXH,RESTRA,1,NRZ,RESID,NRZ,1)
         CALL MULT(MRXH,0,ZHPHR,0,RESTRA,1,NRZ,HPHRIN,NRZ,NRZ)
         CALL MULT(MRXH,0,ZHPHRZ,0,ZHPHR,1,NRZ,RESID,NRZ,1)
C
C        LOAD UP THE CHI-SQUARE FOR THIS STEP
C
         STPCHI(NSTP)=ZHPHRZ(1,1)
C
C        NOW FORM THE CHI-SQUARE FOR EACH SATELLITE
C
         DO 40 NR=1,NRZ
            SATCHI(NR)=SATCHI(NR)+RESID(NR,1)**2/RESVAR(NR,NSTP)
40       CONTINUE
C
10    CONTINUE
C
```

```
      RETURN
      END
C
      SUBROUTINE DECIDE(NSTART,MODE)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      INTEGER NSTART
C
      IF ((NALOUT.EQ.1).AND.(MODE.EQ.1)) WRITE(6,14)NSTART
      IF (NALOUT.EQ.1) WRITE(6,11)NSTART,STAT,NUMDOF,THRESH
C
C     FIRST ZERO OUT THE DECISION VECTORS
C
      DO 50 NWK=NSTART,NWSIZE
         ND1STP(NWK)=0
50    CONTINUE
C
      DO 60 NR=1,NRZ
```

```
         ND1SAT(NR)=0
60    CONTINUE
C
      DO 10 I=NSTART,NWSIZE
C
C        TEST CHI-SQUARE STAT AT EACH STEP
C
         NUMDOF=NDFSTP(I)
         STAT=STPCHI(I)
         THRESH=CHITBL(NUMDOF,LOCSIZ(2))
C
         IF (NALOUT.EQ.1) WRITE(6,12)I,STAT,NUMDOF,THRESH
C
         IF (STAT.LE.THRESH) THEN
            ND1STP(I)=0
         ELSE
            ND1STP(I)=1
         ENDIF
C
10    CONTINUE
C
      DO 20 NR=1,NRZ
C
C        TEST CHI-SQUARE STAT FOR EACH SATELITTE
C
         NUMDOF=NWSIZE-NSTART+1
         STAT=SATCHI(NR)
         THRESH=CHITBL(NUMDOF,LOCSIZ(3))
C
         IF (NALOUT.EQ.1) WRITE(6,13)NR,STAT,NUMDOF,THRESH
C
         IF (STAT.LE.THRESH) THEN
            ND1SAT(NR)=0
         ELSE
            ND1SAT(NR)=1
         ENDIF
C
20    CONTINUE
C
```

```
      NOERR=0
      IF (MODE.EQ.0) THEN
         ND1=0
         DO 70 NC=1,NWSIZE
            DO 70 NR=1,NRZ
               IF ((ND1SAT(NR).EQ.1).AND.(ND1STP(NC).EQ.1)) ND1=1
70       CONTINUE
         IF (ND1.EQ.0) NOERR=1
      ENDIF
C
C     NOW SUPPOSE THAT WE ONLY ALLOW FOR ONE MEASUREMENT
C     SOURCE TO BE ALTERED.  ONLY SET THE DECISION TO 1 IN
C     THE LARGEST OF THOSE SOURCES WHICH HAD A D1 DECISION
C     IN THE TIME SERIES.
C
      CHIMAX=0.D0
      NLARGE=0
      DO 30 NR=1,NRZ
         IF (ND1SAT(NR).EQ.1) THEN
            IF (SATCHI(NR).GT.CHIMAX) THEN
               CHIMAX=SATCHI(NR)
               NLARGE=NR
            ENDIF
         ENDIF
30    CONTINUE
C
C     NOW ZERO OUT THE OTHER SOURCES.
C
      DO 40 NR=1,NRZ
         IF (NR.NE.NLARGE) ND1SAT(NR)=0
40    CONTINUE
C
14    FORMAT(1X,'STATISTICS AFTER REMOVING S',I2,2X,'ARE:')
11    FORMAT(4X,'WINDOW CHI AT NW',I2,1X,'=',F9.3,2X,
     1           'DOF=',I2,2X,'THRESH=',F9.3)
12    FORMAT(4X,'STEP CHI AT NW  ',I2,1X,'=',F9.3,2X,
     1           'DOF=',I2,2X,'THRESH=',F9.3)
13    FORMAT(4X,'SAT CHI FOR ROW ',I2,1X,'=',F9.3,2X,
     1           'DOF=',I2,2X,'THRESH=',F9.3)
```

```
C
      RETURN
      END
C
      SUBROUTINE PREDCT(S,NSTART)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1           MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1       NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER    NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1    SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      INTEGER NSTART
      REAL*8 S(9,1)
      REAL*8 F(10,2),FT(10,10),FSQ(10,2),FSQINV(10,2),FCON(10,10)
      REAL*8 DATA(10,1),XLS(10,1)
      INTEGER ND1MAT(9,10)
C
C     GET THE ND1MAT WHICH IS THE CENSORING MAP
C
      DO 5 NC=NSTART,NWSIZE
         DO 5 NR=1,NRZ
            IF ((ND1STP(NC).EQ.1).AND.(ND1SAT(NR).EQ.1)) THEN
```

```
               ND1MAT(NR,NC)=1
            ELSE
               ND1MAT(NR,NC)=0
            ENDIF
5     CONTINUE
C
      IF (ND1STP(NSTART).EQ.1) THEN
         DO 10 NR=1,NRZ
            INC=0
            IF (ND1SAT(NR).EQ.1) THEN
               LOC=NSTART
               INC=INC+1
               DATA(INC,1)=VBLKA(NR,LOC)
               F(INC,1)=1.D0
               F(INC,2)=FLOAT(INC-1)*DELTAT
               LOCSTR=LOC+1
               IF (NSTART.EQ.NWSIZE) THEN
                  S(NR,1)=VBLKA(NR,NWSIZE)
                  GO TO 21
               ENDIF
               DO 20 J=LOCSTR,NWSIZE
                  IF (ND1MAT(NR,J).EQ.1) THEN
                     INC=INC+1
                     DATA(INC,1)=VBLKA(NR,J)
                     F(INC,1)=1.D0
                     F(INC,2)=FLOAT(INC-1)*DELTAT
                  ELSE
                     IF (INC.EQ.1) THEN
                        S(NR,1)=DATA(1,1)
                     ELSE
                        CALL TRANSP(10,FT,2,INC,F,INC,2)
                        CALL MULT(10,1,FSQ,0,FT,2,INC,F,INC,2)
                        CALL INVERT(10,0,FSQINV,2,FSQ)
                        CALL MULT(10,0,FCON,1,FSQINV,2,2,FT,2,INC)
                        CALL MULT(10,0,XLS,0,FCON,2,INC,DATA,INC,1)
                        S(NR,1)=XLS(1,1)
                     ENDIF
                     GO TO 21
                  ENDIF
                  IF (J.EQ.NWSIZE) THEN
                     CALL TRANSP(10,FT,2,INC,F,INC,2)
                     CALL MULT(10,1,FSQ,0,FT,2,INC,F,INC,2)
                     CALL INVERT(10,0,FSQINV,2,FSQ)
                     CALL MULT(10,0,FCON,1,FSQINV,2,2,FT,2,INC)
                     CALL MULT(10,0,XLS,0,FCON,2,INC,DATA,INC,1)
                     S(NR,1)=XLS(1,1)
                  ENDIF
20             CONTINUE
21             CONTINUE
            ELSE
               S(NR,1)=0.D0
            ENDIF
10       CONTINUE
      ELSE
         CALL SUM(MRZ,0,S,NRZ,1,NULL,NULL)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE NEWVAU
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
```

```
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      REAL*8 D2D(9,9),CONN(9,1)
C
      DO 10 NSTP=1,NWSIZE
C
C        GET THE CONNECTION OF UO AT STEP=NSTP
C
         DO 20 NC=1,NRXD
            DO 20 NR=1,NRZ
               D2D(NR,NC)=D(NR,NC,NSTP)
20       CONTINUE
C
         CALL MULT(MRXH,0,CONN,0,D2D,NRZ,NRXD,U,NRXD,1)
C
C        LOAD UF VBLKA
C
         DO 30 NR=1,NRZ
            VBLKA(NR,NSTP)=VBLK(NR,NSTP) + CONN(NR,1)
30       CONTINUE
C
10    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE NEWVAS(S,NSTART)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1             MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1         NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      REAL*8 S(9,1)
      INTEGER NSTART
C
      REAL*8 C2D(9,9),CONN(9,1)
C
      DO 10 NSTP=NSTART,NWSIZE
C
C        GET THE CONNECTION OF S AT STEP=NSTP
C
         DO 20 NC=1,NRZ
            DO 20 NR=1,NRZ
               C2D(NR,NC)=C(NR,NC,NSTART,NSTP)
20       CONTINUE
C
         CALL MULT(MRXH,0,CONN,0,C2D,NRZ,NRZ,S,NRZ,1)
C
```

```
C        LOAD UP VBLKA
C
         DO 30 NR=1,NRZ
            VBLKA(NR,NSTP)=VBLKA(NR,NSTP) + CONN(NR,1)
30       CONTINUE
C
10    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE NEWU(NSTOP,UNEW)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
```

```
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      REAL*8 UNEW(9,1)
      REAL*8 A2D(9,9),B2D(9,9),CONN(9,1),XSUM(9,1),S(9,1)
C
C     FIRST ZERO OUT XSUM
C
      CALL SUM(MRXH,0,XSUM,NRXD,1,NULL,NULL)
C
      IF (NOERR.EQ.1) GO TO 11
      DO 10 NSTP=1,NSTOP
C
C        GRAB A COLUMN OF SMAT
C
         DO 15 NR=1,NRZ
            S(NR,1)=SMAT(NR,NSTP)
15       CONTINUE
C
C        GET THE CONNECTION OF S INTO XDET ALPHA AT NSTEP=NWSIZE
C
         DO 20 NC=1,NRZ
            DO 20 NR=1,NRXD
               A2D(NR,NC)=A(NR,NC,NSTP,NSTOP)
20       CONTINUE
C
         CALL MULT(MRXH,0,CONN,0,A2D,NRXD,NRZ,S,NRZ,1)
C
C        ADD THE CONNECTION TO XSUM
C
         CALL ADDON(MRXH,XSUM,NRXD,1,CONN)
C
10    CONTINUE
11    CONTINUE
C
C     NOW GET THE CONNECTION OF UO INTO XSUM
C
      DO 30 NC=1,NRXD
         DO 30 NR=1,NRXD
```

```
            B2D(NR,NC)=B(NR,NC,NSTOP)
30    CONTINUE
C
      CALL MULT(MRXH,0,CONN,0,B2D,NRXD,NRXD,U,NRXD,1)
C
      CALL SUM(MRXH,0,UNEW,NRXD,1,CONN,XSUM)
C
      RETURN
      END
C
      SUBROUTINE NEWUT(NSTOP,UNEW)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1           MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      REAL*8 UNEW(9,1)
      REAL*8 A2D(9,9),B2D(9,9),CONN(9,1),XSUM(9,1),S(9,1)
C
C     FIRST ZERO OUT XSUM
C
      CALL SUM(MRXH,0,XSUM,NRXH,1,NULL,NULL)
C
      IF (NOERR.EQ.1) GO TO 11
      DO 10 NSTP=1,NSTOP
C
C        GRAB A COLUMN OF SMAT
C
         DO 15 NR=1,NRZ
            S(NR,1)=SMAT(NR,NSTP)
15       CONTINUE
C
C        GET THE CONNECTION OF S INTO XDET ALPHA AT NSTEP=NWSIZE
C
         DO 20 NC=1,NRZ
            DO 20 NR=1,NRXH
               A2D(NR,NC)=AT(NR,NC,NSTP,NSTOP)
20       CONTINUE
C
         CALL MULT(MRXH,0,CONN,0,A2D,NRXH,NRZ,S,NRZ,1)
C
C        ADD THE CONNECTION TO XSUM
C
         CALL ADDON(MRXH,XSUM,NRXH,1,CONN)
C
10    CONTINUE
11    CONTINUE
C
C     NOW GET THE CONNECTION OF UO INTO XSUM
C
      DO 30 NC=1,NRXH
         DO 30 NR=1,NRXH
            B2D(NR,NC)=BT(NR,NC,NSTOP)
```

```
30    CONTINUE
C
      CALL MULT(MRXH,0,CONN,0,B2D,NRXH,NRXH,UT,NRXH,1)
C
      CALL SUM(MRXH,0,UNEW,NRXH,1,CONN,XSUM)
C
      RETURN
      END
C
      SUBROUTINE HISTRY(K)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1            MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C5/ XACTN,XACTD,XETRAJ,PTRAJ,XDTRAJ,PDTRAJ
      REAL*8 XACTN(44,1200),XACTD(8,1200),XETRAJ(8,1200)
      REAL*8 PTRAJ(8,1200),XDTRAJ(8,1200),PDTRAJ(8,1200)
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1        NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
```

```
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1           ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      REAL*8 XLA(9,1),XSUM(9,1),VSUM(9,1),S(9,1),XTLA(9,1)
      REAL*8 UNEW(9,1),UTNEW(9,1)
      REAL*8 RNAVV(10),RINAVV(10),RDIST(10),STATS(10)
      INTEGER NUMROW(10),NUMCOL(10)
C
      BIGSIG=0.D0
      SMLSIG=1.D30
C
      DO 10 NWK=1,NWSIZE
C
C        LOCATE WILL POINT TO THE LOCATION OF PREVIOUS
C        ESTIMATES AND COVARIANCES IN THE STORAGE ARRAYS.
C
         L=K-NWSIZE+NWK
C
C        CALCULATE THE RESPONSE OF SMAT IN THE DETECTION AND
C        NAVIGATION FILTERS.
C
```

```
      CALL NEWU(NWK,UNEW)
      CALL NEWUT(NWK,UTNEW)
C
C     NOW ADD RESPONSE OF SMAT TO THE NAVIGATION FILTER
C     ESTIMATE AND THE DETECTION FILTER ESTIMATE.
C     ONLY THE X AND Y POSITION COMPONENTS ARE COMPUTED.
C
      XTLA(1,1)=XETRAJ(1,L)+UTNEW(1,1)
      XTLA(3,1)=XETRAJ(3,L)+UTNEW(3,1)
      XLA(1,1)=XDTRAJ(1,L)+UNEW(1,1)
      XLA(3,1)=XDTRAJ(3,L)+UNEW(3,1)
C
C     NOW CALCULATE THE RADIAL ERRORS ASSOCIATED WITH THE
C     ESTIMATED AND THE TRUE VALUES.
C
      RNAV=DSQRT((XACTN(1,L)-XETRAJ(1,L))**2 +
     1           (XACTN(3,L)-XETRAJ(3,L))**2)
C
      RDET= DSQRT((XACTD(1,L)-XDTRAJ(1,L))**2 +
     1           (XACTD(3,L)-XDTRAJ(3,L))**2)
C
      RINAV=DSQRT((XACTN(1,L)-XTLA(1,1))**2 +
     1           (XACTN(3,L)-XTLA(3,1))**2)
C
      RIDET=DSQRT((XACTD(1,L)-XLA(1,1))**2 +
     1           (XACTD(3,L)-XLA(3,1))**2)
C
      RNAVV(NWK)=RNAV
      RINAVV(NWK)=RINAV
      RDIST(NWK)=DSQRT((XTLA(1,1)-XETRAJ(1,L))**2 +
     1                (XTLA(3,1)-XETRAJ(3,L))**2)
C
      IF (PTRAJ(1,L).GT.BIGSIG) BIGSIG=PTRAJ(1,L)
      IF (PTRAJ(3,L).GT.BIGSIG) BIGSIG=PTRAJ(3,L)
      IF (PTRAJ(1,L).LT.SMLSIG) SMLSIG=PTRAJ(1,L)
      IF (PTRAJ(3,L).LT.SMLSIG) SMLSIG=PTRAJ(3,L)
      CORCOF=PWOFFN(NWK)/(PTRAJ(1,L)*PTRAJ(3,L))
C
      IF (NALOUT.EQ.1) THEN
```

```
C
C         NOW PRINT OUT THE ESTIMATES AND RADIAL ERRORS.
C
          WRITE(6,104)L
C
          WRITE(6,44)XETRAJ(1,L),XETRAJ(3,L),PTRAJ(1,L)
          WRITE(6,45)XTLA(1,1),XTLA(3,1),PTRAJ(3,L)
          WRITE(6,46)XACTN(1,L),XACTN(3,L),CORCOF
C
          CORCOF=PWOFFD(NWK)/(PDTRAJ(1,L)*PDTRAJ(3,L))
C
          WRITE(6,144)XDTRAJ(1,L),XDTRAJ(3,L),PDTRAJ(1,L)
          WRITE(6,145)XLA(1,1),XLA(3,1),PDTRAJ(3,L)
          WRITE(6,146)XACTD(1,L),XACTD(3,L),CORCOF
C
          WRITE(6,51)RNAV
          WRITE(6,53)RINAV
          WRITE(6,52)RDET
          WRITE(6,54)RIDET
C
          ENDIF
C
C         NOW CALCULATE THE TEST STATISTIC
C
          X3=(XETRAJ(1,L)-XTLA(1,1))/2.DO
          Y3=(XETRAJ(3,L)-XTLA(3,1))/2.DO
          DELTAX=(X3/PTRAJ(1,L))**2
          DELTAY=(Y3/PTRAJ(3,L))**2
          CROSS=(X3*Y3)*PWOFFN(NWK)/(PTRAJ(1,L)*PTRAJ(3,L))**2
          CORR=PWOFFD(NWK)/(PTRAJ(1,L)*PTRAJ(3,L))**2
C
          TEST=(DELTAX-0.5DO*CROSS+DELTAY)/(1.DO-CORR**2)
          STATS(NWK)=TEST
C
C         LOCATE THE THRESHOLD
C
          IF (NWK.EQ.1) THRESH=CHITBL(2,LOCSIZ(4))
C
          IF (NALOUT.EQ.1) THEN
```

```
         IF (TEST.GT.THRESH) THEN
            WRITE(6,103)TEST,THRESH
         ELSE
            WRITE(6,102)TEST,THRESH
         ENDIF
         ENDIF
C
10    CONTINUE
C
C     NOW SAVE THE LAST UNEW AND UTNEW AS THE STATE
C     ESTIMATE BIAS FOR U AND UT IN THE NEXT WINDOW
C
      CALL SUM(MRXH,0,U,NRXD,1,UNEW,NULL)
      CALL SUM(MRXH,0,UT,NRXH,1,UTNEW,NULL)
C
C
      NUMNZR=0
      DO 20 NR=1,NRZ
         DO 30 NC=1,NWSIZE
            IF (SMAT(NR,NC).NE.0.DO) THEN
               NUMNZR=NUMNZR+1
               NUMROW(NUMNZR)=NR
               GO TO 35
            ENDIF
30       CONTINUE
35       CONTINUE
20    CONTINUE
C
      NUMNZC=0
      DO 21 NC=1,NWSIZE
         DO 31 NR=1,NRZ
            IF (SMAT(NR,NC).NE.0.DO) THEN
               NUMNZC=NUMNZC+1
               NUMCOL(NUMNZC)=NC
               GO TO 36
            ENDIF
31       CONTINUE
36       CONTINUE
21    CONTINUE
```

```
C
      PROTMX=3.0*BIGSIG*DSQRT(THRESH)
      PROTMN=3.0*SMLSIG*DSQRT(THRESH)
C
      WRITE(6,152)(RNAVV(I),I=1,NWSIZE)
      WRITE(6,153)(RINAVV(I),I=1,NWSIZE)
      WRITE(6,154)(RDIST(I),I=1,NWSIZE)
      WRITE(6,155)(STATS(I),I=1,NWSIZE)
      WRITE(6,151)THRESH,BIGSIG,PROTMX,SMLSIG,PROTMN,CORCOF
      IF (NUMNZR.NE.0) WRITE(6,156)(NUMROW(I),I=1,NUMNZR)
      IF (NUMNZC.NE.0) WRITE(6,157)(NUMCOL(I),I=1,NUMNZC)
C
151   FORMAT(1X,'THRESH=',F6.2,1X,'MAX SIG=',F6.1,1X,
     1    'MAX PROT=',F6.1,'m',2X,'MIN SIG=',F6.1,1X,
     2    'MIN PROT=',F6.1,'m',2X,'RHO=',F6.3)
152   FORMAT(1X,'TRUTH AND NAV',8X,9X,10(F7.2,1X))
153   FORMAT(1X,'TRUTH AND CENSORED',8X,4X,10(F7.2,1X))
154   FORMAT(1X,'NAV AND CENSORED',8X,6X,10(F7.2,1X))
155   FORMAT(1X,'TEST STATISTIC',8X,8X,10(F7.2,1X))
156   FORMAT(1X,'ROW NUMBERS OF NON-ZERO ROWS IN SMAT: ',10(I2,1X))
157   FORMAT(1X,'COL NUMBERS OF NON-ZERO COLS IN SMAT: ',10(I2,1X))
C
104   FORMAT(1X,'RESULTS AT STEP K=',I5,/)
C
      RETURN
      END
C
      SUBROUTINE HYPOTH(K)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /C2/ H,PHI,PHITR,P,Q,RT,RF,ZN,ZD,W,V,XHAT,XTRUEN,XTRUED
     1      ,WNOIS,VNOIS,HTRUE,PHITRU,QTRUE,NULLB
      REAL*8 H(9,9),PHI(9,9),PHITR(9,9),P(9,9),Q(9,9)
      REAL*8 RT,RF,ZN(9,1),ZD(9,1),W(44,1),V(44,1),XHAT(9,1)
      REAL*8 XTRUEN(44,1),XTRUED(44,1),WNOIS(44,1200),VNOIS(9,1200)
      REAL*8 HTRUE(44,44),PHITRU(44,44),QTRUE(44,44),NULLB(44,44)
C
      COMMON /C3/ TIME,OFFSET,DELTAT,NSTEPS,NSATS,NSATW,NSATID,
     1              MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
      REAL*8 TIME,OFFSET,DELTAT
      INTEGER NSTEPS,NSATS,NSATW(10),NSATID(9)
      INTEGER MRXT,NRXT,MRXH,NRXH,NRXD,MRZ,NRZ,NOSA
C
      COMMON /C6/ A,B,C,D,U,AT,BT,UT
      REAL*8 A(9,9,10,10),B(9,9,10),C(9,9,10,10),D(9,9,10),U(9,1)
      REAL*8 AT(9,9,10,10),BT(9,9,10),UT(9,1)
C
      COMMON /C7/ BIAS,WBIAS,SLOPE,NULL,IDENT
      REAL*8 BIAS(9),WBIAS(9),SLOPE(9),NULL(9,9),IDENT(9,9)
C
      COMMON /C8/ NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST,
     1         NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
      INTEGER     NERRON,NERROF,NHYPON,NWINDW,NWINC,NWSIZE,NCOAST
      INTEGER NONOIS,NPE,NC2INC,NC2SZ,NC2ON,NGEOM,NGMSZ,NGON,NGSZ
C
      COMMON /C9/ VSAVE,VBLK,CHI,SATCHI,STPCHI,VBLKA,VARINV,RESVAR,
     1     SATCON,NDOF,NDFSTP,ND1SAT,ND1STP,NOERR,NALOUT
      REAL*8 VSAVE(9,1200),VBLK(9,10),CHI(120),SATCHI(9),STPCHI(10)
      REAL*8 VBLKA(9,10),VARINV(9,9,10),RESVAR(9,10),SATCON(9)
      INTEGER NDOF(120),NDFSTP(10),ND1SAT(9),ND1STP(10),NOERR,NALOUT
C
      COMMON /C10/ XDET,PDET,QDET,PY,PXY,YHAT,ZY,HY,PHIY,PHIYT,QY
     1            ,PHID,PHIDTR
      REAL*8 XDET(9,1),PDET(9,9),QDET(9,9),PY(9,2),PXY(9,2)
      REAL*8 YHAT(9,1),ZY(9,1),HY(9,2),PHIY(9,2),PHIYT(9,2),QY(9,2)
      REAL*8 PHID(9,9),PHIDTR(9,9)
C
      COMMON /C12/ CHITBL,TBLSIZ,TBLDOF,LOCSIZ
      REAL*8 CHITBL(66,54),TBLSIZ(54),TBLDOF(66),LOCSIZ(4)
C
      COMMON /C13/ SMAT,PWOFFN,PWOFFD
      REAL*8 SMAT(9,10),PWOFFN(10),PWOFFD(10)
C
      REAL*8 XLA(9,1),XSUM(9,1),VSUM(9,1),S(9,1),XTLA(9,1)
      REAL*8 C2D(9,9),D2D(9,9),CONN(9,1),STEMP(9,1),YTRUE(18,1)
      REAL*8 PREDIC(9,1),A2D(9,9)
      REAL*8 F(11,2),FT(11,11),FSQ(11,2),FSQINV(11,2),FCON(11,11)
      REAL*8 XLS(11,1),WDATA(11,1),YBIAS(18),YSLOP(18),YRAMP(18)
      REAL*8 UNEW(9,1),UTNEW(9,1)
C
C     FIRST GET THE VBLKA USING THE U VECTOR
C
      CALL NEWVAU
C
C     NOW GET THE CHI-SQUARE STATISTICS FOR THE WHOLE VBLKA
C
      CALL GETCHI(1)
C
      IF (NALOUT.EQ.10) THEN
C
C        USE THIS LOOP FOR PRINTING OUT STATISTICS
C        BEFORE CENSORING GETS KICKED IN
C
         WRITE(6,103)NDFSTP(1),CHITBL(NDFSTP(1),LOCSIZ(2)),
     1               (STPCHI(I),I=1,NWSIZE)
         WRITE(6,104)NWSIZE,CHITBL(NWSIZE,LOCSIZ(3)),
     1               (SATCHI(I),I=1,NRZ)
C
      ENDIF
C
C     NOW GET THE DECISION VECTORS FOR THE WHOLE VBLKA
C
      CALL DECIDE(1,0)
C
      STAT=0.D0
      NUMDOF=0
      DO 1 NC=1,NWSIZE
         STAT=STAT+ STPCHI(NC)
         NUMDOF=NUMDOF+NDFSTP(NC)
1     CONTINUE
C
      IF (NOERR.NE.1) THEN
         WRITE(6,101)NWINC
         WRITE(6,102)STAT,NUMDOF,CHITBL(NUMDOF,LOCSIZ(1))
         WRITE(6,103)NDFSTP(1),CHITBL(NDFSTP(1),LOCSIZ(2)),
     1               (STPCHI(I),I=1,NWSIZE)
```

```
         WRITE(6,104)NWSIZE,CHITBL(NWSIZE,LOCSIZ(3)),
     1               (SATCHI(I),I=1,NRZ)
      ENDIF
C
C     ZERO OUT SMAT
C
      CALL SUM(MRXH,0,SMAT,NRZ,NWSIZE,NULL,NULL)
C
      IF (NOERR.EQ.1) THEN
C
C        THERE ARE NO SIGNALS IN THE RESIDUALS.  IN THIS CASE, ONLY
C        UPDATE THE U VECTORS
C
         CALL NEWU(NWSIZE,UNEW)
         CALL NEWUT(NWSIZE,UTNEW)
C
      ELSE
C
C        THERE ARE SIGNALS THAT THE RESIDUALS ARE NOT BELEIVEABLE
C        UNDER THE HO HYPOTHESIS.  IN THIS CASE, TRY TO REMOVE THE
C        BIAS FROM THE BLOCK OF RESIDUALS.  DO THIS BY GETTING
C        THE SMAT.
C
         DO 10 NWK=1,NWSIZE
            CALL SUM(MRXH,0,S,NRZ,1,NULL,NULL)
            IF (ND1STP(NWK).EQ.1) THEN
               CALL PREDCT(PREDIC,NWK)
               CALL ADDON(MRXH,S,NRZ,1,PREDIC)
C
C              NOW UPDATE VBLKA,AND THE DECISION VECTORS
C
               CALL NEWVAS(S,NWK)
               CALL GETCHI(NWK)
               CALL DECIDE(NWK,1)
            ENDIF
C
C           NOW STORE S IN SMAT
C
            DO 20 NR=1,NRZ
```

```
               SMAT(NR,NWK)=S(NR,1)
20          CONTINUE
10       CONTINUE
      ENDIF
C
      IF (NOERR.NE.1) WRITE(6,105)(STPCHI(I),I=1,NWSIZE)
C
      IF ((NOERR.NE.1).AND.(NALOUT.EQ.1)) THEN
C
         WRITE(6,71)
         DO 73 NR=1,NRZ
            WRITE(6,72)(SMAT(NR,NC),NC=1,NWSIZE)
73       CONTINUE
C
         WRITE(6,75)
         DO 76 NR=1,NRZ
            WRITE(6,72)(VBLKA(NR,NC),NC=1,NWSIZE)
76       CONTINUE
C
         WRITE(6,475)
         DO 476 NR=1,NRZ
            WRITE(6,72)(VBLK(NR,NC),NC=1,NWSIZE)
476      CONTINUE
C
      ENDIF
C
C     FINISHED GETTING SMAT.  NOW FIND THE STATE ESTIMATES
C     WITH SMAT REMOVED AND PRINT THE RESULTS.
C
      IF (NOERR.NE.1) CALL HISTRY(K)
C
      IF (NOERR.NE.1) THEN
         IF (NALOUT.EQ.1) WRITE(6,17)(U(I,1),I=1,NRXD)
         IF (NALOUT.EQ.1) WRITE(6,117)(UT(I,1),I=1,NRXH)
      ENDIF
C
C     NOW TAKE CARE OF SOME DETAILS OF PRINTING THE
C     VALUE OF THE ERROR FUNCTION.
C
```

```
      DO 14 I=1,NRZ
         YTRUE(I,1)=WBIAS(I)
         YTRUE(I+NSATS,1)=SLOPE(I)
 14   CONTINUE
 C
      IF (NALOUT.EQ.1) THEN
         IF (K.LT.NERRON) THEN
            WRITE(6,48)
            WRITE(6,42)(I,I=1,NSATS)
            WRITE(6,43)(NULL(I,1),I=1,NSATS)
            WRITE(6,41)(I,I=1,NSATS)
            WRITE(6,43)(NULL(I,1),I=1,NSATS)
         ELSE
            WRITE(6,48)
            WRITE(6,42)(I,I=1,NSATS)
            WRITE(6,43)(YTRUE(I,1),I=1,NSATS)
            WRITE(6,41)(I,I=1,NSATS)
            IST=NSATS+1
            IE=2*NSATS
            WRITE(6,43)(YTRUE(I,1),I=IST,IE)
         ENDIF
      ENDIF
C
C     NOW PROJECT THE WBIAS AHEAD FOR THE NEXT WINDOW
C
      KNEXT=K+1
      IF (K.GE.NERRON) THEN
         DO 59 I=1,NSATS
            WBIAS(I)=WBIAS(I) + SLOPE(I)*DELTAT*FLOAT(NWSIZE)
59       CONTINUE
      ENDIF
C
      IF (KNEXT.GE.NERROF) THEN
         DO 61 I=1,NSATS
            WBIAS(I)=0.D0
            BIAS(I)=0.D0
            SLOPE(I)=0.D0
61       CONTINUE
      ENDIF
```

```
C
42    FORMAT(1X,9(5X,'B',I1,3X))
41    FORMAT(1X,9(5X,'S',I1,3X))
43    FORMAT(1X,9(F9.2,1X))
71    FORMAT(1X,'SMAT')
72    FORMAT(1X,10(F9.3,1X))
75    FORMAT(1X,'V BLOCK ALPHA')
475   FORMAT(1X,'V BLOCK ')
C
101   FORMAT(1X,15('*'),' RESULTS OF WINDOW#',I4,1X,15('*'))
102   FORMAT(1X,'INITIAL STATS  ',2X,'WINDOW CHI= ',
     1           F8.2,2X,'DOF=',I2,2X,'THRESH=',F7.2)
103   FORMAT(1X,'STPCHI:DOF=',I2,1X,'T=',F6.2,2X,
     1           'STATS=',10(F7.2,1X))
104   FORMAT(1X,'SATCHI:DOF=',I2,1X,'T=',F6.2,2X,
     1           'STATS=',10(F7.2,1X))
105   FORMAT(1X,'STPCHI AFTER SMAT:      STATS=',10(F7.2,1X))
C
48    FORMAT(1X,'TRUE BIAS AND SLOPE OF RAMP IN CURRENT WINDOW')
17    FORMAT(4X,'U VECTOR ',4(F10.3,1X),/13X,4(F10.3,1X))
117   FORMAT(4X,'UT VECTOR',4(F10.3,1X),/13X,4(F10.3,1X))
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC                                                      CCCCC
CCCCC           OUTPUT SUBROUTINES ARE AS FOLLOWS          CCCCC
CCCCC                                                      CCCCC
CCCCC             STORE,SIGMA,PLOT,OUTXP,STEADY            CCCCC
CCCCC                                                      CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    THESE SUBROUTINES ARE NOT LISTED IN THE DISSERATION       C
```