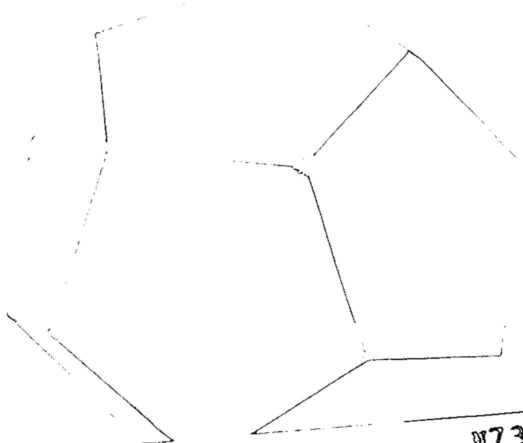
Competitive Evaluation of Failure Detection Algorithms for Strapdown Redundant Inertial Instruments

Final Report
April 1973



Prepared by
James C. Wilcox

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Redundant Sensor Test Program
Contract NAS 8-27335

TRW
SYSTEMS GROUP
One Space Park
Redondo Beach, California 90278

TRW Report No. 18313-6004-RU-00

Competitive Evaluation of Failure Detection Algorithms for Strapdown Redundant Inertial Instruments

Final Report
April 1973

Prepared by
James C. Wilcox

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Redundant Sensor Test Program
Contract NAS 8-27335

TRW SYSTEMS GROUP
One Space Park
Redondo Beach, California 90278

Prepared for
George C. Marshall Space Flight Center
Under Contract NAS 8-27335

Prepared by: James C. Wilcox
James C. Wilcox, Staff Engineer
Systems Analysis Department

Approved by: P. D. Joseph
P. D. Joseph, Manager
Systems Analysis Department

Approved by: E. I. Ergin
E. I. Ergin, Assistant Manager
Navigation and Computer Systems
Laboratory

ABSTRACT

Seven algorithms for failure detection, isolation, and correction of redundant inertial instruments in the strapdown dodecahedron configuration are competitively evaluated in a digital computer simulation that subjects them to identical environments. Their performance is compared in terms of orientation and inertial velocity errors and in terms of missed and false alarms.

The algorithms appear in the simulation program in modular form, so that they may be readily extracted for use elsewhere. The simulation program and its inputs and outputs are described.

The algorithms, along with an eighth algorithm that was not simulated, are also compared analytically to show the relationships among them.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

TABLES

ACKNOWLEDGEMENT

This study would not have been possible without the assistance of the many people who provided me with information about the various algorithms, including T.-T. Chien, J. C. Deckert, R. E. Eckelkamp, J. T. Ephgrave, J. P. Gilmore, S. W. Gully, M. E. Jones, R. A. McKern, and R. T. Savely. However, I bear sole responsibility for the form the various algorithms take herein, especially since I found it necessary to modify them to some extent in order to make the competition fair.

J. C. W.

1. INTRODUCTION

1.1 BACKGROUND

The notion of improving the reliability of a system by using redundant elements and some method of reorganization after a failure has been around for a long time. Since inertial instruments have never been as reliable as one might like, inertial navigators and attitude references have been prime candidates for this treatment. System level redundancy has been most common, with duplicated or triplicated gimballed platforms or strapdown packages. Duplication permits automatic failure detection but requires some kind of external information for failure isolation. Triplication permits automatic failure detection and isolation by a simple majority voting scheme. However, a second failure cannot always be isolated with triplicated systems.

Weiss and Nathan[1,2] seem to have been the first to point out that, if six inertial instruments are arranged so that no three of their input axes are coplanar*, then first and second failures can be detected and isolated. Thus greater reliability can be achieved than with nine instruments arranged three per orthogonal axis. A third failure cannot be isolated without external information.

Ephgrave[3] and Gilmore[4,5,6] have shown that the optimal arrangement for the six instruments is with their input axes perpendicular to the faces of a regular dodecahedron. Both gyros and accelerometers can be arranged in this manner. The symmetry of the dodecahedron configuration maximizes accuracy for the worst cases of operation with a subset of the instruments.

The simple majority voting scheme is no longer usable because the instrument outputs are not directly comparable, since their input

---

* It appears to be most practicable to embody this concept in a strapdown package, thus avoiding the problem of providing gimbal redundancy. However, it is possible, for example, to put three of the instruments on each of two gimballed platforms and use data crossfeeding and gimbal slaving techniques to enhance reliability. Only the strapdown case is considered here.

axes all point in different directions. Therefore, an algorithm must be devised to perform failure detection, isolation, and correction (FDIC).

Many such algorithms have been suggested by various authors. The extent to which the different algorithms have been reduced to practice varies widely from one to another. The purpose of this study is to compare an appropriate selection of FDIC algorithms with one another analytically and by simulation to determine which is the best and to obtain insight into their modes of operation with the goal of combining their best features and remedying their shortcomings.

The problem has two aspects. The first is the problem of detecting signals (the errors of the failed instruments) in the presence of noise (the errors of the unfailed instruments) and making a decision as to the existence of one or more failures. The second aspect is the problem of isolating the failed instruments. This problem is not trivial, because the available information is not always unambiguous in the case of two failures. Of course, simultaneous failures should be quite unlikely in a well-designed system, but one should not overlook the possibility that a second failure could occur during the finite time required to detect and isolate a first failure. It is also conceivable that a first failure might be small enough so as to go undetected and yet be able to interfere with the detection of a subsequent failure because of the ambiguity mentioned above.

1.2 WHAT IS A FAILURE?

A definition of a failure might be:

> "A failure is an event wherein one or more components of an inertial instrument or its associated electronics ceases to function properly."

This may be what intuition tells us a failure is, but it is not very useful in practice because we cannot, in general, tell whether or not each of the components of an instrument is functioning properly at every instant of time.

A more operational definition might be:

> "A failure is an event wherein the error in the output of an inertial instrument exceeds some predetermined specification."

Unfortunately, the normal operation of an unfailed instrument may occasionally cause the specification to be exceeded during staging transients, for example. Also it is mathematically convenient to model the errors of unfailed instruments with gaussian probability densities, which means that any specification has a finite probability of being exceeded without a failure.

We shall cut the Gordian knot by defining a failure as:

> "A failure is an event wherein the failure simulator in the simulation program changes the state of an instrument from unfailed to failed with an accompanying change in the instrument error."

This definition makes up in convenience for what it lacks in elegance.

1.3 RELIABILITY AND PERFORMANCE

In general, we consider reliability to be the probability that the system completes its mission without system failure (in some sense) and performance to be some measure of the errors of the unfailed systems. The errors of the failed systems are large and ought not to be counted against performance.

The nature of the interaction between reliability and performance depends upon the definition of reliability. A definition of reliability might be:

> "Reliability is the probability that less than three gyro failures and less than three accelerometer failures occur during the mission."

With this definition, the reliability is independent of the choice of algorithm. The relative merits of the different algorithms are then revealed by their performance, both in terms of missed and false alarms and in terms of the system velocity and orientation errors.

Another definition of reliability might be:

> "Reliability is the probability that the system errors do not exceed their specifications during the mission."

With this definition, performance is nearly independent of the choice of algorithm, since it is always within specification for an unfailed system. The relative merits of the different algorithms are then revealed by the system reliability.

Still another definition might be:

> "Reliability is the probability that no missed alarms and less than three failures and false alarms combined (for the gyros, and the same for the accelerometers) occur during the mission."

With this definition, both reliability and performance depend upon the choice of algorithms. The logic of this choice is that false alarms, unless they lead to a missed alarm, are not likely to cause the large errors that missed alarms do.

For the same algorithms, the first definition leads to higher reliability and worse performance than the third. Both the reliability and performance associated with the second definition depend upon the specification. The tighter the specification is, the higher the performance and the lower the reliability will be.

The estimation of reliability requires the choice of one of these definitions that suits the mission and specification of the failure rate and failure magnitude probability density for each type of failure. The system error probability density for each failure magnitude must then be determined by simulation for each algorithm for the zero, one, and two failure cases, and all of this information must be combined in the appropriate manner to obtain reliability and performance estimates according to the chosen reliability definition.

It was not possible to carry out such an elaborate program in the course of this study. Instead, the competing algorithms are all subjected to the same series of failures. The resulting system errors are then used as relative figures of merit to rank the algorithms. This technique avoids the detailed characterization of the failure modes peculiar to a particular system that would otherwise be required.

For a manned mission such as Space Shuttle, performance is very important. That is, if two algorithms yield the same reliability, the one that performs FDIC quickly before large velocity or orientation errors are accumulated is certainly to be preferred over a slower FDIC that does allow such errors to build up, in order to minimize the risk to the men on board.

On a long-duration unmanned astronomical satellite, however, the errors building up during FDIC action are of relatively little importance, merely causing a temporary loss of observation time. Algorithms designed for such a mission may score somewhat poorly in performance compared to algorithms designed for a shuttle mission. However, it is felt that even for such algorithms it will be meaningful, or at least interesting, to look at their performance relative to the other algorithms.

1.3.1 Reliability With Perfect FDIC

If we assume that the FDIC algorithms are perfect, then the reliability is simply the probability that less than three gyro failures and less than three accelerometer failures occur during the mission. If p is the probability of a given instrument failing during a specified time interval, then the probability of m out of n instruments failing in the specified time interval is given by P(m, n)

$$P(m,n) = \frac{n!}{m!\,(n-m)!}\; p^m (1-p)^{n-m} \tag{1-1}$$

When a constant failure rate, λ, and a time interval, Δt, are specified, the probability of a given instrument and its associated electronics failing during the time interval is

$$p = 1 - e^{-\lambda \Delta t} \tag{1-2}$$

The gyro or accelerometer reliability is given by

$$R = \sum_{m=0}^{2} P(m, 6) \tag{1-3}$$

for the dodecahedron configuration, since three failures cannot be isolated without external information. If R_g is the gyro reliability and R_a is the accelerometer reliability, calculated according to Eqs. 1-1 through 1-3, then the system reliability is

$$R = R_g R_a \tag{1-4}$$

If there are any series elements (common power supplies, clocks, heaters, computer, etc.) with a significant failure rate, the actual reliability will be much less than Eq. 1-4.

1.3.2 Performance With Perfect FDIC

The performance of instruments in the dodecahedron configuration can be calculated in a simple manner if it is assumed that a perfect FDIC combines the outputs of the unfailed instruments by least-squares, and if it is assumed that the errors of the unfailed instruments are statistically independent with zero mean and unit variance.

If the FDIC output errors are expressed in terms of the strapdown package coordinate system, they are correlated, and they differ for each different combination of failed instruments. Furthermore, generally the axis of greatest error will not lie along one of the coordinate axes, so that the greatest error will not be immediately apparent. Therefore, the covariance matrix for any given combination of failures is diagonalized by means of an orthogonal transformation to a new error coordinate system. The error coordinate system, in which the FDIC output errors are uncorrelated, can be related to the input axes of the failed and unfailed instruments so as to provide a physical picture of its orientation. The principal axes of the error ellipsoid lie along the axes of the error coordinate system, which are different for each different combination of failed instruments.

For zero instruments failed, the error coordinate system axes may be taken in any direction (retaining orthogonality, of course). The error ellipsoid is a sphere.

For one instrument failed, axis A is along the input axis of the failed instrument. Axes B and C may have arbitrary orientation about A. The error ellipsoid is a prolate spheroid with major axis along A.

For two instruments failed, their input axes form two acute angles and two obtuse angles. Axis A bisects the acute angles. Axis B bisects the obtuse angles. Axis C is perpendicular to both input axes. The error ellipsoid has its major axis along A and its minor axis along C.

For three instruments failed, there are two equally probable possibilities. Either the three faces of the dodecahedron normal to the input axes of the unfailed instruments meet at a vertex of the dodecahedron (case A) or the three faces of the dodecahedron normal to the input axes of the failed instruments meet at a vertex of the dodecahedron (case B). Axis A passes through the center of the dodecahedron and the vertex. Axes B and C may have arbitrary orientation about A. For case A, the error ellipsoid is an oblate spheroid with minor axis along A. For case B, the error ellipsoid is a prolate spheroid with major axis along A.

Average errors are computed by adding together all of the covariance matrices (without diagonalization) for a given number of failures and dividing by the number of matrices. The result turns out to be equal to the identity matrix multiplied by a scalar, the square root of which is called the average error. Thus, when averaged over all possible combinations, the error ellipsoid for a given number of failures is a sphere. Table 1-I presents the standard deviations of the errors in the error coordinate system and the average errors.

Table 1-I. Errors With Perfect FDIC

Number of Instruments Failed	Number of Combinations	Error Coordinate System Axes			Average Error
		A	B	C	
0	1	0.707	0.707	0.707	0.707
1	6	1.000	0.707	0.707	0.816
2	15	1.345	0.831	0.707	1.000
3 A	10	0.727	1.345	1.345	1.176
3 B	10	3.078	0.831	0.831	1.902
3 A and B	20				1.581

1.4 THE ALGORITHMS

Eight different algorithms have been selected for competitive evaluation. The algorithms rejected are all earlier or alternative efforts of authors whose algorithms have been selected. In roughly chronological order the selected algorithms are

1.	Adaptive 66	(Ephgrave)[3, 7]
2.	Fifteen Threshold	(Evans and Wilcox)[8, 9, 10]
3.	Squared Error	(Gilmore, McKern, and Oehrle)[11, 12, 13, 14]
4.	Bayesian Decision Theory	(Gully)[15, 16]
5.	Maximum Likelihood	(Wilcox)[17, 18]
6.	Minimax	(Potter and Deckert)[19, 20, 21, 22, 23, 24, 25, 26]
7.	Adaptive 72	(Chien)[27, 28]
8.	Sequential	(Eckelkamp and Schiesser)[29]

The algorithms are described briefly here and in more detail in Paragraphs 3.2.1 through 3.2.8. However, the reader must go to the references for the theoretical backgrounds and derivations of the algorithms which, in many cases, are quite lengthy.

The Adaptive 66 Algorithm by Ephgrave[3, 7] uses a weighted least-squares estimator to perform failure detection, isolation, and correction. The estimator estimates the instrument package input vector (acceleration or angular velocity). The residuals are used to estimate the variances of the instrument errors. The variances are used to obtain the weights for the estimator in an iterative fashion.

The Maximum Likelihood Algorithm by Wilcox[17, 18] is very similar but uses six additional test signals whose expected values are the instrument variances to start the iteration.

The Bayesian Decision Theory Algorithm by Gully[15, 16] performs failure correction by means of a least-squares estimator using only the instruments classified as unfailed. The residuals are tested against thresholds to perform failure detection and isolation.

The Squared Error Algorithm by Gilmore, McKern, and Oehrle[11, 12, 13, 14] performs failure correction by means of a least-squares estimator like that of the Bayesian Decision Theory Algorithm. The sum of the squares of the residuals (total squared error) is tested against a threshold to perform failure detection. The squares of the residuals are tested against the total squared error to perform failure isolation.

The Fifteen Threshold Algorithm by Evans and Wilcox[8, 9, 10] performs failure correction by means of a least-squares estimator like those of the preceding two algorithms. Fifteen test signals, each involving the outputs of a different set of four instruments, are compared to threshold to perform failure detection and isolation.

The Minimax Algorithm by Potter and Deckert[19, 20, 21, 22, 23, 24, 25, 26] offers two different methods of failure correction. One is a least-squares estimator like those of the preceding three algorithms. The other is a Bounding Sphere Algorithm (described below). Failure detection and isolation are performed with the same fifteen test signals as the preceding algorithm but different logic.

The Sequential Algorithm by Eckelkamp and Schiesser[29] uses a Kalman-Bucy filter to perform failure correction. The filter has nine states: three for the package input vector and six for the instrument errors. The measurements are the six instrument outputs. The K-B residuals and the instrument error states are compared against thresholds to perform failure detection and isolation. Failed instruments are eliminated from the measurement vector.

The Adaptive 72 Algorithm by Chien[27, 28] performs failure correction by means of a least-squares estimator like those of the Bayesian Decision Theory, Squared Error, and Fifteen Threshold Algorithms. However, it also performs "identification" to decide whether a failure has a bias, a ramp, or a "variance" type of waveform. It then performs "recompensation" to estimate the value of a bias or ramp failure and "recertification" to decide if the recompensation effort was successful enough to allow the instrument to be reinstated using the new compensation.

It is reinstated simply by reverting to the least-squares solution that includes it. Failure detection and isolation use six test signals that are a subset of the fifteen test signals mentioned above for the first failure and five test signals that are also a subset of the fifteen test signals for the second failure. The selection of five signals depends upon which instrument failed first so that all fifteen signals are used at one time or another.

1.5 THE SIMULATION

Figure 1-1 shows the overall system block diagram of the FDIC simulation program, FAILSIM. It is used to compare the different algorithms.

- The trajectory generator provides an ideal trajectory that includes rotational and translational vibratory motions to exercise the algorithms.
- The instrument configuration calculation provides components of angular velocity and acceleration along the input, output, and spin axes of each gyro and along the input, pendulous, and output axes of each accelerometer.
- The unfailed errors calculation provides all of the errors experienced by the gyros and accelerometers in unfailed operation, except those produced by the sampling and quantization processes.
- The additive failures calculation provides the errors caused by those failures that add to the output produced by the instruments in unfailed operation. (An example would be a bias drift shift which would leave the response to inputs unchanged.)
- The integration, sampling, and quantization calculations provide these functions, converting angular velocity and acceleration into increments of angle and increments of velocity.
- The substitutional failures calculation provides the instrument outputs caused by those failures that substitute an incorrect output for the correct one (an example would be zero output).

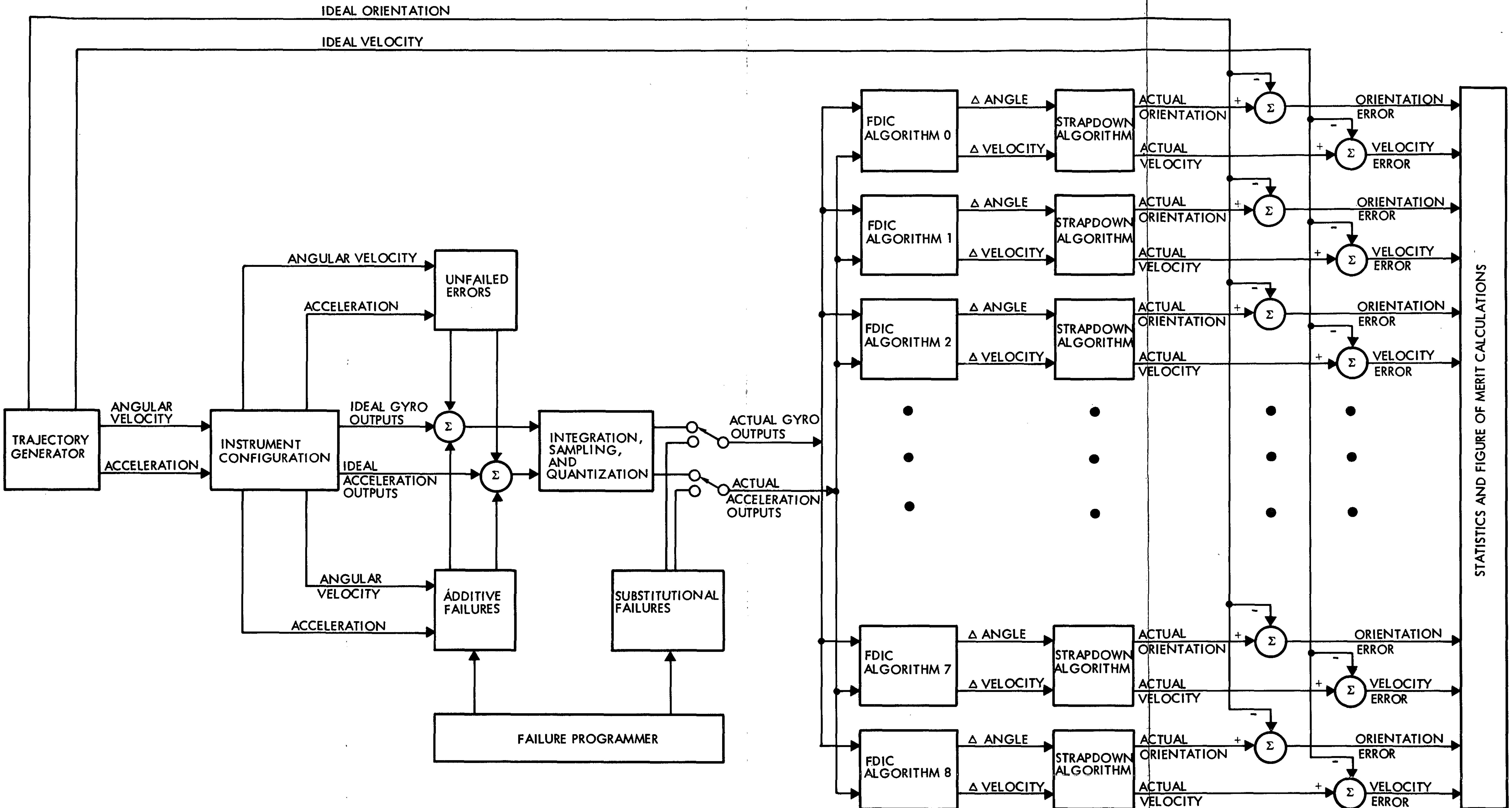


Figure 1-1. FDIC Simulation Program (FAILSIM) Overall System Block Diagram

FOLDOUT FRAME 1

FOLDOUT FRAME 2

- The nine failure detection, isolation, and correction (FDIC) algorithms each combine the six gyro outputs into an incremental angle vector and the six accelerometer outputs into an incremental velocity vector. FDIC algorithm 0 is a nominal algorithm supplied with information from the failure simulator so that it always uses all of the unfailed instruments and none of the failed instruments. FDIC algorithm 0 provides a standard against which the candidates can be compared.

- The nine identical strapdown algorithms convert incremental angles and incremental velocities into actual orientation and actual inertial velocity.

- The orientation and velocity errors are calculated using the ideal orientation and ideal inertial velocity from the trajectory generator.

- Missed and false-alarm statistics, orientation errors, velocity errors, and computer time are calculated to provide a basis for comparing the different algorithms.

A computer session wherein the program is loaded and executed is referred to as a run. Each run may contain one or more cases, each case having its own input data. Each case may contain one or more Monte Carlo trials, all trials having the same input data but having different pseudorandom number sequences.

2. CONCLUSIONS

Seven of the eight algorithms were simulated (all except the Adaptive 72 Algorithm). The detailed results are given in Section 8. The conclusions are based upon the gyro results only, since the accelerometer results were incomplete due to limited resources. Four of the algorithms, the Fifteen Threshold, Squared Error, Bayesian Decision Theory, and Minimax gave considerably better all-around performance than the other three, Adaptive 66, Maximum Likelihood, and Sequential. The first four have in common a definite logical structure that is absent from the Adaptive 66 and Maximum Likelihood Algorithms and is not as clear-cut in the Sequential Algorithm. Thus we conclude that a definite logical structure is to be desired. The structure should make explicit what instruments are classified as failed or unfailed, what hypotheses are being tested (or what decisions are being made), and what test signals and thresholds are to be used in testing each hypothesis (or making each decision).

Of the three poorer algorithms, it seems clear that the Adaptive 66 is the best and Maximum Likelihood the worst. The problem with the Maximum Likelihood Algorithm seems to be that subtracting off the residuals from the inconsistency states is an inadequate method of preventing interaction between subsequent failures (Paragraph 3.2.5). The Sequential Algorithm is also subject to interaction to a lesser extent (Paragraph 3.1.2.4). We conclude that interaction should be completely eliminated as it is in the four better algorithms. (It is essentially eliminated in the Adaptive 66 Algorithm.) By elimination of interaction we mean that the output of an instrument which has been classified as failed should not be used in making a decision about the classification of another instrument.

Of the four better algorithms, none demonstrates a clear-cut superiority. In the double simultaneous failure runs they rank as follows:

Minimax
Fifteen Threshold
Squared Error
Bayesian Decision Theory.

However, the Fifteen Threshold Algorithm is significantly poorer than the others in response to first failures. Apparently this result occurs because, unlike the other three algorithms, its thresholds for first and second failures are identical. However, false alarms are more likely with one instrument already failed, requiring a higher threshold level for second failures than for first failures. Thus we conclude that independently adjustable thresholds for first and second failures (Squared Error, Bayesian Decision Theory) or a properly preset ratio between the thresholds for first and second failures (Minimax) should be provided.

The first and second failure plots show no clear-cut advantages for any of the remaining three algorithms, except possibly that the Bayesian Decision Theory Algorithm is not quite as good as the other two. If we let the double failure results govern, then the overall standings would be:

Minimax
Squared Error
Bayesian Decision Theory
Fifteen Threshold
Adaptive 66
Sequential
Maximum Likelihood.

However, differences between the top three algorithms are marginal and the ordering should not be taken too seriously. For example, the Minimax Algorithm has relatively high computer sizing and timing requirements. Weighting these factors heavily could move it down the list.

We seem justified in making at least a tentative conclusion that indirect test signals are better than direct test signals (Paragraphs 3.1.2.1 and 3.1.2.3), especially for double failures. This conclusion tends to substantiate the claims of Reference 24.

3. CANDIDATE FDIC ALGORITHMS

The analytical background of the different algorithms is explored in Subsection 3. 1, while Subsection 3. 2 specifies the algorithms in detail.

3. 1 ANALYTICAL COMPARISON OF ALGORITHMS

The different algorithms are compared analytically in this section.

3. 1. 1 General Considerations

At least three types of information may be available for use by an FDIC algorithm:

1. The outputs of the inertial instruments
2. The internal states of the inertial instruments (such as gyro wheel speed)
3. The outputs of electromagnetic radiation sensors (such as star trackers, doppler radars, etc.).

All of the algorithms use the first type of information. Only the Fifteen Threshold Algorithm uses the second type of information, although the others are capable of modification with varying degrees of facility to use it. In order to avoid undue complexity and keep the comparison fair, this capability will be deleted from the Fifteen Threshold Algorithm. None of the algorithms use the third type of information, but the Sequential Algorithm is formulated so as to be able to accept such information readily.

There are at least two different concepts of the nature of the problem being solved when one designs an algorithm for this type of system. They lead to different types of algorithms. The first concept is that an instrument is either failed or unfailed. The algorithm must determine which instruments are failed and which are not. The subset of unfailed instruments is used for navigation while the failed instruments are ignored. The Fifteen Threshold, Squared Error, Bayesian Decision Theory, Minimax, Adaptive 72, and Sequential Algorithms are of this type.

The second concept is that the important state of an instrument is the error of its output. The algorithm must determine the variance of this error and weight the instrument outputs accordingly. The Adaptive and Maximum Likelihood Algorithms are of this type.

The first problem concept may be considered as a special case of the second, in which the instrument error variance estimates can take on only the values one and infinity.

The unfailed instruments are subject to errors during normal operation. These errors can cause the algorithms to make mistakes in isolating the failed instruments. Both false alarms (unfailed instruments classified as failed) and missed alarms (failed instruments classified as unfailed) can occur. For the first problem concept, false and missed alarms appear as the use of an incorrect subset of instruments. For the second problem concept, false and missed alarms appear as the use of incorrect instrument weights.

In the deterministic case, when the unfailed instruments have zero errors, it is possible to detect one, two, or three failures and to isolate one or two failures. However, when the errors resulting from the failures of two instruments have certain ratios, it is not possible to distinguish between that event and the failure of a different pair of instruments. (See Paragraph 3. 1. 3.) In the deterministic case, this situation has probability zero, but when the errors of the unfailed instruments are nonzero, the probability of ambiguity becomes greater than zero. For this reason, the isolation of double simultaneous failures is difficult. Most of the algorithms seem to have been constructed under the assumption that double simultaneous failures are very unlikely. Only the Minimax Algorithm provides a choice of implementations that depends upon this point. For consistency, only nonsimultaneous failures will be assumed for the Minimax Algorithm. (This decision does not preclude determination of the effects of double simultaneous failures on the algorithms during evaluation.)

Another question is how the different algorithms respond to "glitches" (temporary malfunctions), that is, whether or not an instrument will be reinstated as unfailed if "healing" is observed. The Fifteen Threshold and Minimax Algorithms do not permit reinstatement. The Adaptive 66, Squared Error, and Maximum Likelihood Algorithms do permit reinstatement. It seems from the documentation that the Bayesian Decision Theory Algorithm is not intended to permit reinstatement. If reinstatement is not permitted, then glitches are treated as failures. On the other hand, if glitches are infrequent in a particular system, prohibiting reinstatement will preclude the erroneous reinstatement of failed instruments. Still another point is that reinstatement may permit the eventual correction of false alarms. A particular kind of glitch is the shift of bias drift rate sometimes experienced by gyros. The ability to recalibrate the drift and reinstate the instrument might be desirable. Such a capability, however, would introduce the possibility of the occurrence of unnecessary, erroneous recalibrations. Gully[15] discusses drift calibration but does not present an FDIC algorithm incorporating this feature. The Adaptive 72 Algorithm has the capability of recompensating and reinstating instruments with ramp errors as well as bias shifts.

In the Adaptive 66 and Maximum Likelihood Algorithms, failure detection and isolation are obscured by a variance estimation scheme. In the Fifteen Threshold, Bayesian Decision Theory, Adaptive 72, and Sequential Algorithms, failure detection occurs only when isolation occurs. In the Squared Error Algorithm, detection and isolation are separate processes. No action is taken when detection without isolation occurs. However, if an instrument is isolated as failed and subsequently the isolation test fails, the instrument is not reinstated unless the detection test fails also. The Minimax/Bounding Sphere Algorithm has separate failure correction strategies for isolated and for detected but unisolated failures.

The Fifteen Threshold, Squared Error, Bayesian Decision Theory, and Adaptive 72 Algorithms use a least-squares failure correction technique. The Adaptive 66 and Maximum Likelihood Algorithms use a weighted least-squares failure correction technique. The Minimax Algorithm has a choice of either a least-squares failure correction technique or the Bounding Sphere Technique, which minimizes the maximum possible estimation error. The Sequential Algorithm uses a Kalman-Bucy filter to perform failure correction.

It is possible to construct six test signals, each predominantly dependent upon the error of a single instrument, but also dependent to a lesser extent upon the errors of the other instruments. Such "direct" test signals may be linear or quadratic and are used by the Adaptive 66, Squared Error, Bayesian Decision Theory, and Maximum Likelihood Algorithms. The linear direct test signals are proportional to the residuals of the least squares solution.

It is possible to construct 15 test signals, each depending upon the errors of a different group of 4 instruments, and completely independent of the errors of the other two instruments. Such "indirect" or "parity" test signals are used by the Fifteen Threshold, Minimax, and Adaptive 72 Algorithms.

The Fifteen Threshold, Squared Error, and Maximum Likelihood Algorithms all make use of prefilters to reduce the effects of quantization errors. To keep the comparison fair, the Adaptive 66, Bayesian Decision Theory, and Minimax Algorithms will be equipped with prefilters too. The Adaptive 72 Algorithm has a filter in its detection system and thus requires no prefilter. The Sequential filter is a modified Kalman-Bucy filter and also needs no prefilter.

Since there are six instruments of a given type, six failed/unfailed decisions must be made or six variances must be determined to select or weight the instrument outputs properly. The six instrument outputs can provide six equations. However, these six equations contain nine unknowns: the six instrument errors and the three components of the angular velocity or acceleration inputs to the instrument package about which little a priori information is available. It is this deficiency of three equations that causes the fundamental difficulty in the design of failure detection algorithms based solely upon the instrument outputs and, by precluding the possibility of perfect failure detection, makes the problem interesting.

Additional a priori information can be obtained by assuming that the magnitudes of the errors of four of the six instruments are less than quantities determined by their unfailed performance, or that the variances of four instruments must be equal to their unfailed value, while the variances of the other two instruments may be greater than or equal to this value. The inequalities thus obtained are, however, insufficient to replace the missing equations.

It is possible to solve for three "inconsistency states" which are linear combinations of the instrument outputs and therefore of the instrument errors and which are independent of the inputs to the instrument package and of each other. The vector comprising these three states is zero when all of the instrument errors are zero, regardless of the motion of the instrument package. This vector contains all of the failure detection information available to the types of algorithm considered here. The various test signals used by the different algorithms are all functions of the inconsistency state vector. The only exception is the Sequential Algorithm which attempts to use a priori information about the vehicle motion. It is shown in Paragraph 3.1.2.4 that this a priori information has little effect.

Some of the algorithm documentation discusses running the algorithm at a slower rate than the rate at which information from the instruments is incorporated into the strapdown algorithm. For the purposes of this study, such an arrangement is considered unacceptable, because it permits a hard failure to wipe out the orientation or inertial velocity states completely. Therefore all algorithms will run at the same sampling period, which is the strapdown algorithm minor cycle time. Thus they may be compared on an equal footing. (Of course, applications exist where a wipeout is acceptable. In general, they will require less computer capacity.)

3.1.2 Derivation of Equations

The instrument outputs are:

$$y = Ax + \epsilon \tag{3-1}$$

where

y = 6-vector of instrument angular velocity or acceleration outputs

A = 6 by 3 matrix of instrument input axis direction cosines with respect to axes fixed in the strapdown package

x = 3-vector of instrument package angular velocity or acceleration, and

ϵ = 6-vector of instrument errors.

For all of the algorithms, we take

$$A = \begin{bmatrix} c & s & 0 \\ c & -s & 0 \\ 0 & c & s \\ 0 & c & -s \\ s & 0 & c \\ -s & 0 & c \end{bmatrix} \qquad (3\text{-}2)$$

where

$$c = \sqrt{\frac{5 + \sqrt{5}}{10}} = \cos \alpha \qquad (3\text{-}3)$$

$$s = \sqrt{\frac{5 - \sqrt{5}}{10}} = \sin \alpha \qquad (3\text{-}4)$$

where α is half the angle whose tangent is two.* The rows of A are unit vectors perpendicular to the faces of a dodecahedron, as seen in Figure 3-1. The columns of A are orthogonal 6 vectors of length $\sqrt{2}$. It is possible to choose three more 6 vectors of length $\sqrt{2}$ which are orthogonal to the columns of A and to each other. Let them comprise the rows of a matrix C. One of the infinite number of possibilities is

$$C = \begin{bmatrix} s & s & 0 & 0 & -c & c \\ -c & c & s & s & 0 & 0 \\ 0 & 0 & -c & c & s & s \end{bmatrix} \qquad (3\text{-}5)$$

By definition

$$CA = 0 \qquad (3\text{-}6)$$

*It is interesting to note that the ratio s/c is equal to the golden mean.

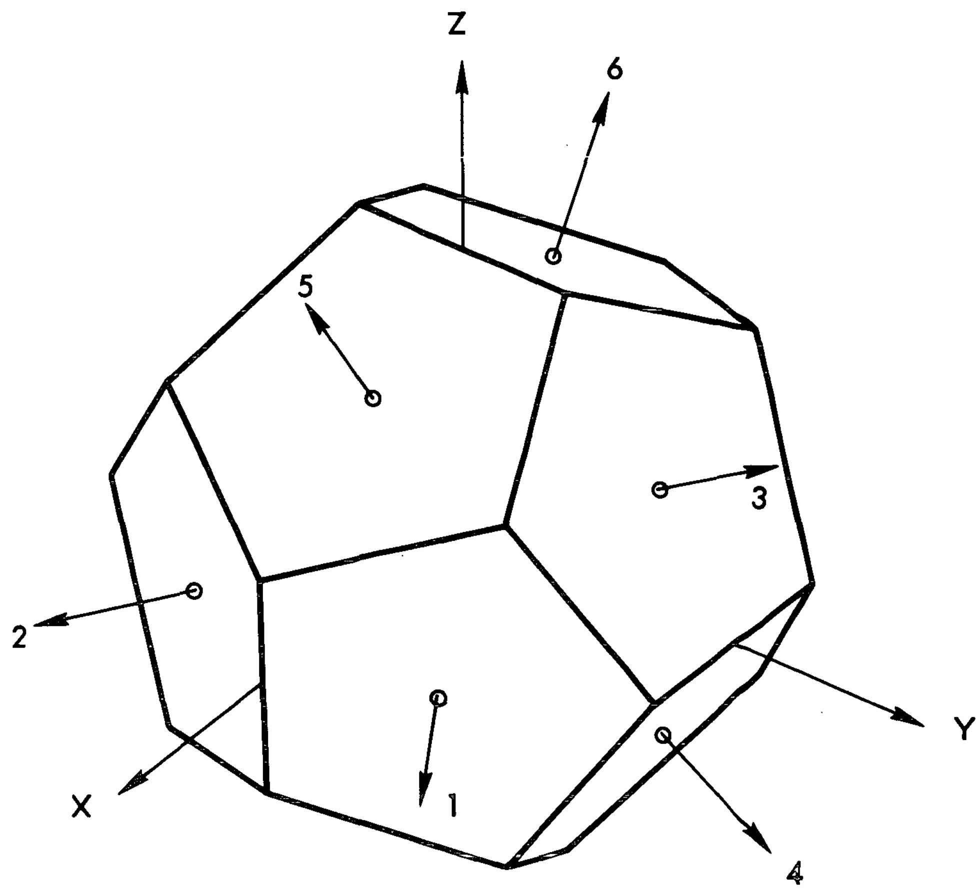


Figure 3-1. Dodecahedron Instrument Configuration

Let v be a 3-vector of inconsistency states, where

$$v = Cy \tag{3-7}$$

From Eqs. 3-1, 3-6, and 3-7

$$v = C\epsilon \tag{3-8}$$

Thus the inconsistency states are independent of the IMU input angular velocity or acceleration and depend only on the instrument errors. All of the test signals used in the various algorithms can be expressed as linear or quadratic combinations of the components of v.

Most of the algorithms have a prefilter which reduces the effect of high frequency noise, such as quantization error, on performance. The prefilter is a first-order filter such as

$$f^{(i+1)} = k_A f^{(i)} + k_B v^{(i+1)} \tag{3-9}$$

where

$$k_A = \exp(-T/\tau_f) \tag{3-10}$$

$$k_B = 1 - k_A \tag{3-11}$$

with T being the sampling period and τ_f the filter time constant.

When the failures have been detected and isolated, it is necessary to correct for them. In the Adaptive and Maximum Likelihood Algorithms, correction is accomplished by obtaining the weighted least-squares estimate of x.

$$\hat{x} = B\, y \tag{3-12}$$

$$B = \left(A^T Q^{-1} A\right)^{-1} A^T Q^{-1} \tag{3-13}$$

where Q is the covariance matrix of the instrument errors and is assumed to be diagonal.

In the Fifteen Threshold, Total Squared Error, and Bayesian Decision Theory Algorithms, Q_{ii}^{-1} takes on only the values 0 or 1. Thus, Eq. 3-13 becomes

$$B = \left(A^T A\right)^{-1} A^T \tag{3-14}$$

where the rows of A corresponding to zero Q_{ii}^{-1} are set equal to zero.

3.1.2.1 Direct Test Signals

Several of the algorithms make use of the residuals of the least-squares solution, as "direct" test signals, using them as estimates of the instrument errors. They are given by

$$\hat{\epsilon} = y - \hat{y} \tag{3-15}$$

where

$$\hat{y} = A\, \hat{x} \tag{3-16}$$

so that, from Eqs. 3-12, 3-15, and 3-16,

$$\hat{\epsilon} = (I - AB)\, y \tag{3-17}$$

For the case of no failures, Q is the identity matrix and

$$B = \frac{1}{2} A^T \tag{3-18}$$

One can see by direct calculation that

$$I - \frac{1}{2} A A^T = \frac{1}{2} C^T C \tag{3-19}$$

so that the ϵ_i are linear combinations of the v_j:

$$\hat{\epsilon} = \frac{1}{2} C^T C y = \frac{1}{2} C^T v \tag{3-20}$$

Thus the six residuals contain no more information than the three inconsistency states. Prefiltered values of $\hat{\epsilon}$ can be obtained from

$$\hat{\epsilon} = \frac{1}{2} C^T f \tag{3-21}$$

where f is given in Eq. 3-9. This formulation requires the prefiltering of only three quantities, rather than six.

When the k^{th} instrument has failed, the residuals are given by Eqs. 3-14 and 3-17 with the k^{th} row of A set equal to zero to give A(k).

$$\hat{\epsilon} = \left\{ I - A(k) \left[A(k)^T A(k) \right]^{-1} A(k)^T \right\} y \tag{3-22}$$

One can see, by direct calculation, that Eq. 3-22 is equivalent to

$$\epsilon = \frac{1}{2} C(k)^T C y = \frac{1}{2} C(k)^T v \tag{3-23}$$

where

$$C_{ij}(k) = C_{ij} - \left(\sum_{\ell=1}^{3} C_{\ell j} C_{\ell k} \right) C_{ik} \tag{3-24}$$

with the exception that $\hat{\epsilon}_k$, the residual of the failed instrument, is given as

$$\hat{\epsilon}_k = y_k \tag{3-25}$$

by Eq. 3-22 and as

$$\hat{\epsilon}_k = 0 \tag{3-26}$$

by Eq. 3-23. The latter is better, since the residual of the failed instrument has to be ignored, and Eq. 3-26 causes it to be ignored automatically.

For the case of zero failures, the residuals may be calculated from Eqs. 3-1, 3-5, 3-6, and 3-20

$$\hat{\epsilon} = \frac{1}{2} \begin{bmatrix} 1 & -d & -d & -d & -d & d \\ -d & 1 & d & d & -d & d \\ -d & d & 1 & -d & -d & -d \\ -d & d & -d & 1 & d & d \\ -d & -d & -d & d & 1 & -d \\ d & d & -d & d & -d & 1 \end{bmatrix} \epsilon \tag{3-27}$$

where

$$d = \frac{1}{\sqrt{5}} \tag{3-28}$$

The Bayesian Decision Theory Algorithm uses the residuals as test signals.

3.1.2.2 Variance Estimators

The residuals, $\hat{\epsilon}_i$, may be considered to be estimates of the instrument error values. Their squares may be considered as estimates of the variances of the instrument errors. The squared residuals are used in the Squared Error Algorithm. The squared residuals are, however, not unbiased estimates of the instrument error variances. From Eqs. 3-8 and 3-20

$$\hat{\epsilon} = \frac{1}{2} C^T C \epsilon \tag{3-29}$$

Let

$$D = C^T C \tag{3-30}$$

then

$$\hat{\epsilon}_i = \frac{1}{2} \sum_{j=1}^{6} D_{ij} \, \epsilon_j \tag{3-31}$$

and

$$\hat{\epsilon}_i^2 = \frac{1}{4} \sum_{j=1}^{6} \sum_{k=1}^{6} D_{ij} D_{ik} \epsilon_j \epsilon_k \tag{3-32}$$

The covariance matrix of ϵ is assumed to be diagonal,

$$\left\langle \epsilon_j \epsilon_k \right\rangle = q_j \, \delta_{jk} \tag{3-33}$$

If we take the expectation of Eq. 3-32 and introduce Eq. 3-33, we obtain

$$\left\langle \hat{\epsilon}_i^2 \right\rangle = \frac{1}{4} \sum_{j=1}^{6} D_{ij}^2 \, q_j \tag{3-34}$$

From Eqs. 3-5, 3-30, and 3-34, we obtain

$$\left\langle \hat{\epsilon}_i^2 \right\rangle = \frac{q_i}{5} + \frac{1}{20} \sum_{j=1}^{6} q_j \tag{3-35}$$

and see that the estimates are biased. Unbiased estimates may be obtained by subtracting a portion of the total squared error from each squared error (residual). Let $\hat{q}$ be the unbiased estimate. The total squared error is

$$\text{TSE} = \sum_{j=1}^{6} \hat{\epsilon}_j^2 \tag{3-36}$$

We now multiply the TSE by b and subtract it from the squared error. The result is normalized by a.

$$\hat{q}_i = a\left(\hat{\epsilon}_i^2 - b\sum_{j=1}^{6}\hat{\epsilon}_j^2\right) \tag{3-37}$$

From Eqs. 3-35 and 3-37

$$\left\langle\hat{q}_i\right\rangle = a\left[\frac{q_i}{5} + \frac{1}{20}\sum_{j=1}^{6} q_j - b\sum_{j=1}^{6}\left(\frac{q_j}{5} + \frac{1}{20}\sum_{k=1}^{6} q_k\right)\right] \tag{3-38}$$

$$\left\langle\hat{q}_i\right\rangle = a\left[\frac{q_i}{5} + \left(\frac{1}{20} - \frac{b}{2}\right)\sum_{j=1}^{6} q_j\right] \tag{3-39}$$

To make $\hat{q}$ unbiased, $b = 1/10$, and to get the scale factor correct, $a = 5$. Thus

$$\hat{q}_i = 5\,\hat{\epsilon}_i^2 - \frac{1}{2}\sum_{j=1}^{6}\hat{\epsilon}_j^2 \tag{3-40}$$

$$\left\langle\hat{q}_i\right\rangle = q_i \tag{3-41}$$

It is of some interest to express Eq. 3-40 in terms of the inconsistency states, v_i. From Eq. 3-20, we have

$$\hat{\epsilon}_i = \frac{1}{2}\sum_{j=1}^{3} C_{ji}\, v_j \tag{3-42}$$

and

$$\hat{\epsilon}_i^2 = \frac{1}{4}\sum_{j=1}^{3}\sum_{k=1}^{3} C_{ji}\, C_{ki}\, v_j\, v_k \tag{3-43}$$

so that

$$\hat{q}_i = \frac{5}{4} \sum_{j=1}^{3} \sum_{k=1}^{3} C_{ji}\, C_{ki}\, v_j\, v_k - \frac{1}{8} \sum_{l=1}^{6} \sum_{j=1}^{3} \sum_{k=1}^{3} C_{jl}\, C_{kl}\, v_j\, v_k \tag{3-44}$$

It can be shown from Eq. 3-5 that

$$\sum_{l=1}^{6} C_{jl}\, C_{kl} = 2\, \delta_{jk} \tag{3-45}$$

so that

$$\hat{q}_i = \frac{1}{4} \sum_{j=1}^{3} \sum_{k=1}^{3} \left(5\, C_{ji}\, C_{ki} - \delta_{jk}\right) v_j\, v_k \tag{3-46}$$

Now if $\hat{Q}$ is a diagonal matrix

$$\hat{Q}_{ij} = \delta_{ij}\, \hat{q}_i \tag{3-47}$$

then Eq. 3-46 can be shown to be the solution of

$$C\, \hat{Q}\, C^T = v\, v^T \tag{3-48}$$

which defines the test signals of the Maximum Likelihood Algorithm.

Now let us compare $\hat{\epsilon}_i^2$ with $\hat{q}_i$. For example, the value of $\hat{\epsilon}_1^2$ is given by

$$\hat{\epsilon}_1{}^2 = \frac{1}{4}\, \epsilon^T \begin{bmatrix} 1 & -d & -d & -d & -d & d \\ -d & d^2 & d^2 & d^2 & d^2 & -d^2 \\ -d & d^2 & d^2 & d^2 & d^2 & -d^2 \\ -d & d^2 & d^2 & d^2 & d^2 & -d^2 \\ -d & d^2 & d^2 & d^2 & d^2 & -d^2 \\ d & -d^2 & -d^2 & -d^2 & -d^2 & d^2 \end{bmatrix} \epsilon \tag{3-49}$$

where d is given by Eq. 3-28. The value of $\hat{q}_1$ is

$$\hat{q}_1 = \epsilon^T \begin{bmatrix} 1 & -d & -d & -d & -d & d \\ -d & 0 & \frac{s^2}{2} & \frac{s^2}{2} & \frac{c^2}{2} & -\frac{c^2}{2} \\ -d & \frac{s^2}{2} & 0 & \frac{c^2}{2} & \frac{c^2}{2} & -\frac{s^2}{2} \\ -d & \frac{s^2}{2} & \frac{c^2}{2} & 0 & \frac{s^2}{2} & -\frac{c^2}{2} \\ -d & \frac{c^2}{2} & \frac{c^2}{2} & \frac{s^2}{2} & 0 & -\frac{s^2}{2} \\ d & -\frac{c^2}{2} & -\frac{s^2}{2} & -\frac{c^2}{2} & -\frac{s^2}{2} & 0 \end{bmatrix} \epsilon \qquad (3\text{-}50)$$

It is interesting to note that, if ϵ_k is nonzero and the other ϵ_i are all zero, then because of the zeros on the diagonal of Eq. 3-50 and the analogous equations for the other $\hat{q}_i$, only $\hat{q}_k$ is nonzero. This is not true for $\hat{\epsilon}_i$ or $\hat{\epsilon}_i^{\,2}$.

If we assume gaussian distributions, we can find the mean square error in the estimate of the variance. Assume that σ_k is the variance of the k^{th} instrument and σ^2 is the variance of the other five instruments; then it can be shown that

$$\left\langle \left(4\,\hat{\epsilon}_k^{\,2} - \sigma_k^{\,2}\right)^2 \right\rangle = 2\,\sigma_k^{\,4} + 4\,\sigma_k^{\,2}\,\sigma^2 + 3\,\sigma^4 \qquad (3\text{-}51)$$

$$\left\langle \left(\hat{Q}_{kk} - \sigma_k^{\,2}\right)^2 \right\rangle = 2\,\sigma_k^{\,4} + 4\,\sigma_k^{\,2}\,\sigma^2 + 3\,\sigma^4 \qquad (3\text{-}52)$$

Thus, both types of quadratic test signals have the same mean square error when considered as estimators of the variance of the instrument errors. Similar calculations can be done for the case of one failure.

3.1.2.3 Indirect Test Signals

We now consider the "indirect" or "parity" test signals which, instead of trying to put in evidence the errors of a particular instrument, show the error of a group of instruments that excludes one or two particular instruments. Test signals that exclude one instrument are perfectly possible, but are not used by any of the candidates and will be ignored here. The Fifteen Threshold, Minimax, and Adaptive 72 Algorithms use the 15 test signals, each of which excludes two instruments. Requiring that the signals be independent of the input angular velocity or translational acceleration determines them to within a constant factor. If u is the 15-vector of test signals, then

$$u = D\,y \tag{3-53}$$

where a typical D matrix is

$$D = \begin{bmatrix} -c & c & s & s & 0 & 0 \\ s & -c & -c & 0 & s & 0 \\ c & -s & -c & 0 & 0 & s \\ -c & s & 0 & c & s & 0 \\ -s & c & 0 & c & 0 & s \\ s & s & 0 & 0 & -c & c \\ -s & 0 & -s & c & c & 0 \\ s & 0 & -c & s & 0 & c \\ c & 0 & -s & 0 & -s & c \\ c & 0 & 0 & -s & -c & s \\ 0 & -s & -c & s & c & 0 \\ 0 & s & -s & c & 0 & c \\ 0 & c & s & 0 & -c & s \\ 0 & c & 0 & s & -s & c \\ 0 & 0 & -c & c & s & s \end{bmatrix} \tag{3-54}$$

Each component of u is a linear combination of the components of the inconsistency state vector v. Thus the matrix D can be factored into two matrices

$$D = F\ C \tag{3-55}$$

and

$$u = F\ C\ y = F\ v \tag{3-56}$$

Prefiltering may be performed on the three components of v rather than the 15 components of u. Thus, we replace Eq. 3-56 by

$$u = F\ f \tag{3-57}$$

where f is given by Eq. 3-9 and where

$$F = \frac{1}{2}\begin{bmatrix} 0 & 2 & 0 \\ -s/c & -c/s & 1 \\ s/c & -c/s & 1 \\ -s/c & c/s & 1 \\ s/c & c/s & 1 \\ 2 & 0 & 0 \\ -1 & s/c & c/s \\ 1 & -s/c & c/s \\ c/s & -1 & s/c \\ c/s & -1 & -s/c \\ -1 & -s/c & c/s \\ 1 & s/c & c/s \\ c/s & 1 & -s/c \\ c/s & 1 & s/c \\ 0 & 0 & 2 \end{bmatrix} \tag{3-58}$$

The Adaptive 72 Algorithm uses u_1, u_3, u_6, u_{10}, u_{11}, and u_{15} for the detection and isolation of the first failure. For detection and isolation of the second failure when the first failure is instrument j, it uses the five u_i which have D_{ij} zero.

There is a close relationship between the indirect test signals and the least-squares residuals. If instruments j and k are assumed to be failed and a least-squares solution is performed excluding them, then all of the four residuals can be shown to be proportional to u_i, where i is the index of the row of the matrix D for which D_{ij} and D_{ik} are both zero. Two of the residuals are equal to $\pm c/2$ times u_i, and two are equal to $\pm s/2$ times u_i.

3.1.2.4 Sequential Algorithm

The Sequential Algorithm is essentially a Kalman-Bucy filter, and thus appears rather different from the other algorithms. This section will use a coordinate transformation in 6-space to demonstrate that the actual operation of the Sequential Algorithm is similar to the operation of the other algorithms. The detailed description is presented in Paragraph 3.2.8.

The linear, stochastic, multistage process to be filtered is

$$x_{i+1} = \Phi\, x_i + u_i \tag{3-59}$$

where

$$\langle x_o \rangle = 0 \tag{3-60}$$

$$\langle u_i \rangle = 0 \tag{3-61}$$

$$\langle x_o\, x_o^T \rangle = M_o \tag{3-62}$$

$$\langle u_i\, u_j^T \rangle = Q\, \delta_{ij} \tag{3-63}$$

$$\langle u_i\, x_o^T \rangle = 0 \tag{3-64}$$

Measurements y_i are linearly related to the state by

$$y_i = H_i x_i + w_i \tag{3-65}$$

where

$$\langle w_i \rangle = 0 \tag{3-66}$$

$$\langle w_i w_j^T \rangle = R \, \delta_{ij} \tag{3-67}$$

$$\langle u_i w_j^T \rangle = 0 \tag{3-68}$$

$$\langle x_o w_i^T \rangle = 0 \tag{3-69}$$

The filter is given by the following. For resets:

$$\hat{x}_i = \bar{x}_i + K_i \left(y_i - H_i \bar{x}_i \right) \tag{3-70}$$

$$K_i = M_i H_i^T \left(H_i M_i H_i^T + R \right)^{-1} \tag{3-71}$$

$$P_i = \left(I - K_i H_i \right) M_i \tag{3-72}$$

For propagation between resets:

$$\bar{x}_{i+1} = \Phi \, \hat{x}_i \tag{3-73}$$

$$M_{i+1} = \Phi \, P_i \Phi^T + Q \tag{3-74}$$

The state vector, x, is a 9-vector whose first three elements are the package input 3-vector of acceleration or angular velocity. The last six elements are the instrument error 6-vector. The measurement vector, y, is a 6-vector whose elements are the instrument outputs. Thus the measurement matrix is, from Eq. 3-1

$$H = \left[A \,\vdots\, I_6 \right] \tag{3-75}$$

where A is given by Eq. 3-2.

The state transition matrix is

$$\Phi = \left[\begin{array}{c|c} \phi_1 I_3 & 0 \\ \hline 0 & \phi_2 I_6 \end{array}\right] \tag{3-76}$$

The covariance matrices are

$$M_o = \left[\begin{array}{c|c} m_1 I_3 & 0 \\ \hline 0 & m_2 I_6 \end{array}\right] \tag{3-77}$$

$$Q = \left[\begin{array}{c|c} q_1 I_3 & 0 \\ \hline 0 & q_2 I_6 \end{array}\right] \tag{3-78}$$

$$R = \left[r I_6 \right] \tag{3-79}$$

Because of the H matrix, all of the nine states are coupled together, making insight into the filter operation rather difficult. To remove this difficulty, we transform the six instrument error states and the six measurements (instrument outputs) into a new coordinate system by means of the matrix D.

$$D = \frac{1}{2} \left[\begin{array}{c} A^T \\ \hline C \end{array}\right] \tag{3-80}$$

$$x' = \left[\begin{array}{c|c} I_3 & 0 \\ \hline 0 & D \end{array}\right] x \tag{3-81}$$

$$y' = D\, y \tag{3-82}$$

where C is given by Eq. 3-5. It can be shown that the inverse of D is

$$D^{-1} = \left[\begin{array}{c|c} A & C^T \end{array}\right] \tag{3-83}$$

If we carry out the transformation, we find that the new filter is given by Eqs. 3-70 to 3-74 with unprimed quantities replaced by primed quantities and with

$$\Phi' = \Phi \tag{3-84}$$

$$M'_o = M_o \tag{3-85}$$

$$Q' = Q \tag{3-86}$$

$$R' = R \tag{3-87}$$

However, the new measurement matrix becomes

$$H' = \left[\begin{array}{c|c|c} I_3 & I_3 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_3 \end{array}\right] \tag{3-88}$$

Now we see that in all matrices (Φ', M'_o, Q', R', and H'), the first six states are decoupled from the last three. Furthermore, because of the identity submatrices, there are actually six uncoupled sets of states comprising three identical two-state filters (states 1 and 4, 2 and 5, 3 and 6) and three identical one-state filters (states 7, 8, and 9). (Note that this conclusion will no longer hold after failure detection causes the H matrix to be modified by deletion of a row.)

For the two-state filters, the Kalman gains can easily be found to be

$$K'_{11} = K'_{22} = K'_{33} = \frac{m_1}{m_1 + m_2 + r} \tag{3-89}$$

$$K'_{41} = K'_{52} = K'_{63} = \frac{m_2}{m_1 + m_2 + r} \tag{3-90}$$

while, for the one-state filters, the Kalman gains are

$$K'_{74} = K'_{85} = K'_{96} = \frac{m_2}{m_2 + r} \tag{3-91}$$

Now, since m_1 represents the variance of the vehicle motions while m_2 represents the variance of the instrument errors and r represents the variance of the measurement errors (quantization), we have that

$$m_1 >> m_2 \tag{3-92}$$

$$m_1 >> r \tag{3-93}$$

Therefore K'_{41}, K'_{52}, and K'_{63} are nearly zero and states 4, 5, and 6 of our transformed filter remain essentially zero. States 7, 8, and 9 are simply the result of three first-order Kalman filters acting upon y'_4, y'_5, and y'_6. However, from Eqs. 3-80 and 3-82 we see that

$$\begin{Bmatrix} y'_4 \\ y'_5 \\ y'_6 \end{Bmatrix} = \frac{1}{2} C\, y = \frac{1}{2} v \,. \tag{3-94}$$

so that the inputs to the three first-order filters are proportional to the familiar three uncertainty states. Thus x'_7, x'_8, and x'_9 are similar to the filtered uncertainty states f, except for the time-varying nature of the Kalman filter. Furthermore, from Eqs. 3-81 and 3-83 we see that

$$\begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{Bmatrix} = A \begin{Bmatrix} x'_4 \\ x'_5 \\ x'_6 \end{Bmatrix} + C^T \begin{Bmatrix} x'_7 \\ x'_8 \\ x'_9 \end{Bmatrix} \tag{3-95}$$

Since x'_4, x'_5, and x'_6 are essentially zero, x_4 - x_9 are analogous to filtered versions of the direct test signals or residuals discussed previously. The smallness of x'_4, x'_5, and x'_6 comes about because the a priori information available about the vehicle motions is insufficient to contribute significantly to the solution of the failure detection problem.

Of course, if other types of measurements were taken (i.e., star tracker) the situation would be completely different. It is in the latter case that the Sequential Algorithm should prove most effective. Unfortunately, it is beyond the scope of this study.

One aspect of the Sequential Algorithm in which it differs from most of the other algorithms should be mentioned. When a failure is detected, failure correction is accomplished by deleting a row from the H matrix and proceeding as before. However, no changes are made to the existing values of the instrument error states. Eq. 3-27 shows that, when one instrument exhibits an error, it not only contributes to its appropriate residual, but contributes an amount equal to $1/\sqrt{5}$ (44.7%) of the error to the other residuals. When a failure is detected, the other five instrument error states will therefore include such errors of up to 44.7% of the threshold level, which will subsequently gradually decay with time. In most of the other algorithms, the FDIC reorganization after a failure is such that no part of the output of a failed instrument contributes to the test signals used to find the second failure.

Since the Sequential Algorithm has so many adjustable parameters, a relationship has been derived between its Kalman-Bucy filter operating in steady state and the constant gain prefilters used by many of the other algorithms, in order to gain some insight into the meanings of the parameters. This derivation is presented in Appendix B.

3.1.3 Multiple Simultaneous Failures

The possibility, mentioned at the beginning of this section, that two failures can be indistinguishable from two other failures in the deterministic case can be seen by considering the three sets of errors

$$\begin{aligned} \epsilon^T &= (c \quad 0 \quad 0 \quad 0 \quad s \quad 0) \\ \epsilon^T &= (0 \quad -c \quad 0 \quad 0 \quad 0 \quad s) \\ \epsilon^T &= (0 \quad 0 \quad -c \quad -s \quad 0 \quad 0) \end{aligned} \tag{3-96}$$

For each of these different cases of two instruments failed and four instruments unfailed, multiplication of the error vector by the matrix C of Eq. 3-5 gives $v = (0, -c^2, s^2)$. Since we have shown that all of the test signals of all of the different algorithms are functions of v, the three cases are indistinguishable to all of the FDIC algorithms. Reference 24 gives an interesting discussion of simultaneous failure detection and isolation.

3.1.4 Linearity Versus Nonlinearity

Occasionally it has been suggested that the instrument biases should be estimated by some kind of linear filter and then used to adjust the bias compensation of the instruments, thus avoiding the necessity of switching out the instruments or weighting their outputs. This procedure is linear. It is therefore doomed to failure. The least-squares solution already gives the optimal linear estimate of acceleration or angular velocity. For example, suppose we modify the instrument outputs by subtracting the estimates of the instrument errors (residuals) from them

$$y' = y - \hat{\epsilon} \tag{3-97}$$

where $\hat{\epsilon}$ is given by Eq. 3-15. Then an improved estimate might be obtained by modifying Eq. 3-12 to

$$\hat{x}' = B\, y' \tag{3-98}$$

From Eqs. 3-97, 3-98, and 3-15

$$\hat{x}' = B\, \hat{y} \tag{3-99}$$

From Eqs. 3-99 and 3-16

$$\hat{x}' = B\, A\, \hat{x} \tag{3-100}$$

From Eq. 3-13 we see that B A is the identity matrix so that

$$\hat{x}' = \hat{x} \tag{3-101}$$

and no improvement occurs. This behavior is a well-known feature of optimal linear solutions, intimately related to their optimality. The same situation arises even if a Kalman-Bucy filter is used to estimate the instrument errors, because the estimate of the package inputs is already the optimal estimate.

All of the FDIC algorithms studied here are nonlinear. The "linear parts" of all of them have been shown above to be essentially identical, all test signals depending upon the inconsistency vector v. It is the manner in which the necessary nonlinearity is introduced that distinguishes each algorithm from the others.

From Eq. 3-19 we can see that the instrument error vector can be broken up into two parts by projection matrices

$$\epsilon = \frac{1}{2} AA^T \epsilon + \frac{1}{2} C^T C \epsilon \qquad (3\text{-}102)$$

These two vectors are orthogonal to each other in 6-space

$$\left(\frac{1}{2} A A^T \epsilon\right)^T \left(\frac{1}{2} C^T C \epsilon\right) = \frac{1}{4} \epsilon^T A A^T C^T C \epsilon \qquad (3\text{-}103)$$

Transposing Eq. 3-6 gives

$$A^T C^T = 0 \qquad (3\text{-}104)$$

So that Eq. 3-103 must equal zero. If we write Eq. 3-102 as

$$\epsilon = \tilde{\epsilon} + \hat{\epsilon} \qquad (3\text{-}105)$$

we see that $\tilde{\epsilon}$ contributes to the estimation error 3-vector but not to the inconsistency 3-vector, and that $\hat{\epsilon}$ contributes to the inconsistency vector but not to the estimation error vector. $\hat{\epsilon}$ is accessible to the FDIC designer, but $\tilde{\epsilon}$ is not since the actual package motion is not known. The FDIC problem is to use knowledge of $\hat{\epsilon}$ to find ϵ without knowledge of $\tilde{\epsilon}$. Fortunately, perfect knowledge of ϵ is not required.

3.1.5 V Space Logic

The 3-vector v is defined in some 3-space which we may call v-space. This space is not the same as the ordinary 3-space in which the instrument input axis vectors and the package input vector x are defined. In the preceding section, we see that v-space and ordinary 3-space may be considered as having orthogonal bases in a 6-space.

It is interesting to think of the different FDIC algorithms as structures in v-space. Since all of the test signals are functions of the v_i, the comparison of a test signal to a threshold appears as the determination of whether the end of the v-vector lies on one side or the other of a surface in v-space. For example, total squared error equal to a threshold level generates a sphere centered on the origin of v-space, and a parity signal equal to a threshold level generates a plane.

Thus the failure isolation algorithm synthesis problem can be looked upon as setting up the optimal boundary surfaces in v-space, associating decisions with the crossing of various boundaries, and changing the boundaries optimally after each decision.

In an algorithm with a prefilter, the filtered value of v, f, would be used in place of v.

Up to the present time, no one has performed FDIC synthesis directly in v-space. Perhaps the concept will never be of more than academic interest.

3.1.6 Filtering

The quantization error experienced with the typical strapdown package calls for some type of a filtering process to reduce its effect. For example, a 5-arc sec quantum size and a 10-msec sampling period lead to an instantaneous gyro drift rate with a standard deviation of 204 deg/hr.

The Fifteen Threshold and Maximum Likelihood Algorithms make use of a simple first-order low-pass prefilter. First let us consider the effect of step failures. If we approximate the sampled-data filter by a continuous filter, its transfer function is simply

$$G(s) = \frac{1}{1 + \tau_f s} \qquad (3\text{-}106)$$

Where τ_f is the filter time constant. The filter response to a step input of magnitude a is

$$\omega_f(t) = a\left(1 - e^{-\frac{t}{\tau_f}}\right) \qquad (3\text{-}107)$$

and the time required to reach a threshold level, b, is

$$t = \tau_f \ln \frac{a}{a-b} \tag{3-108}$$

The error caused by a step (angular velocity or acceleration) of magnitude a is approximately

$$\epsilon = at \tag{3-109}$$

(angle or velocity).

Therefore

$$\epsilon = a\,\tau_f \ln \frac{a}{a-b} \tag{3-110}$$

If we normalize Eq. 3-110 on the threshold level b,

$$\frac{\epsilon}{b\tau_f} = \frac{a}{b} \ln \frac{\frac{a}{b}}{\frac{a}{b} - 1} \tag{3-111}$$

Figure 3-2 shows this relationship. The larger the failure is, the less error it contributes to the system before it is switched out. If the failure is less than the threshold ($a/b < 1$), then the error increases indefinitely until the end of the mission.

Now let us consider the effect of quantization errors. The transfer function of the sampled-data filter (of which Eq. 3-106 is an approximation) is

$$G(z) = \frac{\left(1 - e^{-\frac{t}{\tau_f}}\right) z}{z - e^{-\frac{t}{\tau_f}}} \tag{3-112}$$

The angular velocity output of the gyro (or the acceleration output of the accelerometer) is given by continuous integration, sampling, quantization, and numerical differentiation as shown in Figure 3-3.

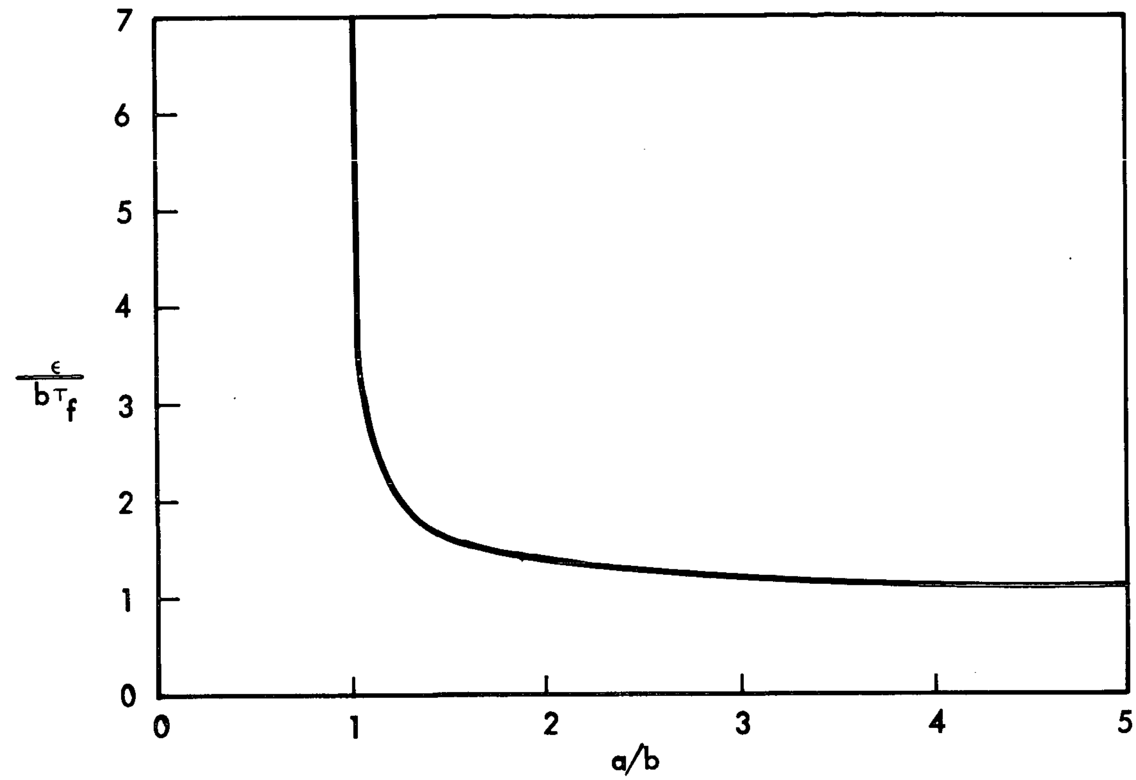


Figure 3-2. Low-Pass Prefilter Error Versus Failure Magnitude

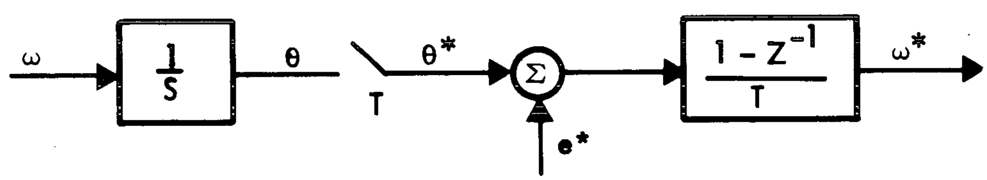


Figure 3-3. Gyro or Accelerometer Model

Now let us approximate the quantization error, e*, by a random sequence such that

$$\langle e^*_i \rangle = 0 \tag{3-113}$$

$$\langle e^*_i \; e^*_j \rangle = \sigma^2 \; \delta_{ij} \tag{3-114}$$

where the standard deviation is given by

$$\sigma^2 = \frac{Q^2}{12} \tag{3-115}$$

where Q is the quantum size. (Eq. 3-115 assumes that e_i is uniformly distributed over $\pm Q/2$.) From Eq. 3-112 and Figure 3-3, we see that the filter output caused by quantization is

$$\omega_f = \frac{\left(1 - e^{-\frac{T}{\tau_f}}\right)}{z - e^{-\frac{T}{\tau_f}}} z \frac{1 - z^{-1}}{T} \epsilon^* \tag{3-116}$$

By long division, Eq. 3-116 becomes

$$\omega_f = \frac{1 - e^{-\frac{T}{\tau_f}}}{T} \left[1 + \left(e^{-\frac{T}{\tau_f}} - 1 \right) z^{-1} + \left(e^{-\frac{2T}{\tau_f}} - e^{-\frac{T}{\tau_f}} \right) z^{-2} + \ldots \right] e^* \tag{3-117}$$

$$\omega_{f_o} = \frac{1 - e^{-\frac{T}{\tau_f}}}{T} \left[\epsilon^*_o + \sum_{i=1}^{\infty} \left(e^{-i\frac{T}{\tau_f}} - e^{-(i-1)\frac{T}{\tau_f}} \right) e^*_{-i} \right] \tag{3-118}$$

From Eqs. 3-114, 3-115, and 3-118,

$$\sigma^2_{\omega_f} = \left\langle \omega^2_{f_o} \right\rangle = \frac{Q^2 \left(1 - e^{-\frac{T}{\tau_f}}\right)^2}{12\, T^2} \left[1 + \sum_{i=1}^{\infty} \left(e^{-i\frac{T}{\tau_f}} - e^{-(i-1)\frac{T}{\tau_f}} \right)^2 \right] \tag{3-119}$$

$$\sigma^2_{\omega_f} = \frac{Q^2 \left(1 - e^{-\frac{T}{\tau_f}}\right)^2}{6T^2 \left(1 + e^{-\frac{T}{\tau_f}}\right)} \tag{3-120}$$

For $\tau_f \rightarrow 0$

$$\sigma_{\omega_f} \rightarrow \frac{Q}{\sqrt{6}\, T} \tag{3-121}$$

the unfiltered value; while for $\tau_f \rightarrow \infty$

$$\sigma_{\omega_f} \rightarrow \frac{Q}{\sqrt{12}\, \tau_f} \tag{3-122}$$

the error becomes independent of the sampling period. From Eq. 3-120 we may write

$$\frac{\sigma_{\omega_f} T}{Q} = \frac{\left(1 - e^{-\frac{T}{\tau_f}}\right)}{\sqrt{6\left(1 + e^{-\frac{T}{\tau_f}}\right)}} \tag{3-123}$$

This function is plotted in Figure 3-4.

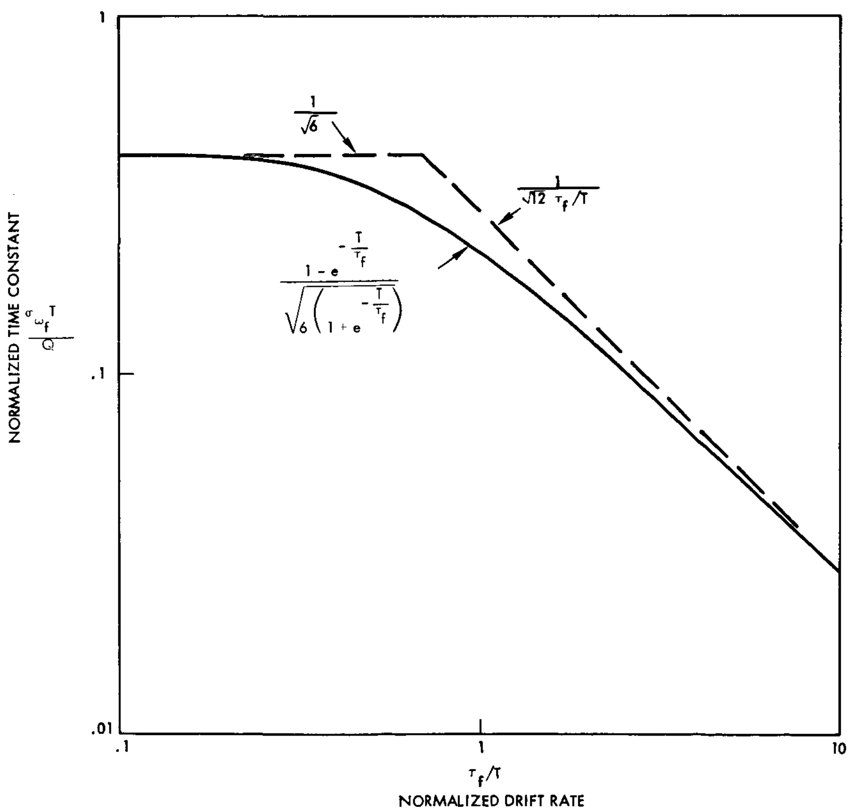


Figure 3-4. Normalized Drift Rate Versus Normalized Time Constant

The Squared Error Algorithm makes use of a different type of prefilter. The instrument outputs are accumulated. Every 60 sec the accumulators are purged of all outputs older than 120 sec. If we approximate the sampled-data accumulator by a continuous one, the prefilter looks like Figure 3-5, where T_p is the purge period of 60 sec. (The gain of $1/3\ T_p$ normalizes the prefilter so that its steady-state peak gain is unity, to correspond with Eq. 3-107. This gain factor does not actually appear in the Squared Error prefilter and is used here only to facilitate comparison with the low-pass prefilter described previously.)

The response of this prefilter depends upon the phasing of the failure with respect to the purges. The response to a step failure of magnitude a is shown in Figure 3-6 for four different phases. By averaging over all values of relative phase, we can find the mean time to exceed a threshold level, b.

$$\bar{t} = \begin{cases} \infty & 0 < a/b < 1 \\ \left[\frac{9}{2}\left(\frac{b}{a}\right)^2 - 3\frac{b}{a} + 2\right] T_p & 1 < a/b < 3/2 \\ 3\frac{b}{a} T_p & 3/2 < a/b < \infty \end{cases} \qquad (3\text{-}124)$$

Following Eq. 3-109, and normalizing,

$$\frac{\bar{\epsilon}}{3bTp} = \begin{cases} \infty & 0 < a/b < 1 \\ \left[\frac{3}{2}\frac{b}{a} - 1 + \frac{2}{3}\frac{a}{b}\right] & 1 < a/b < 3/2 \\ 1 & 3/2 < a/b < \infty \end{cases} \qquad (3\text{-}125)$$

This function is plotted in Figure 3-7.

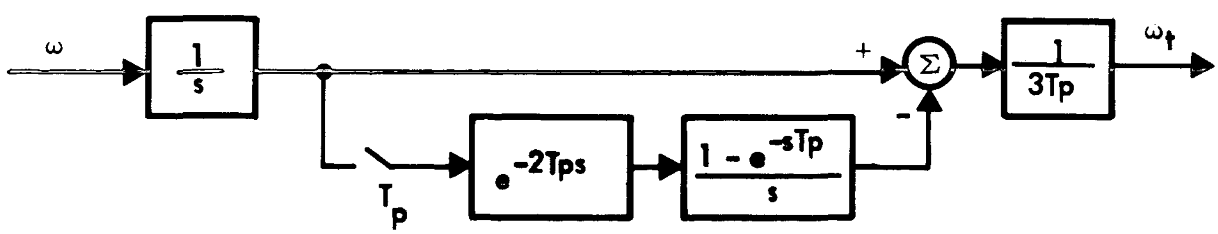


Figure 3-5. Squared Error Algorithm Prefilter

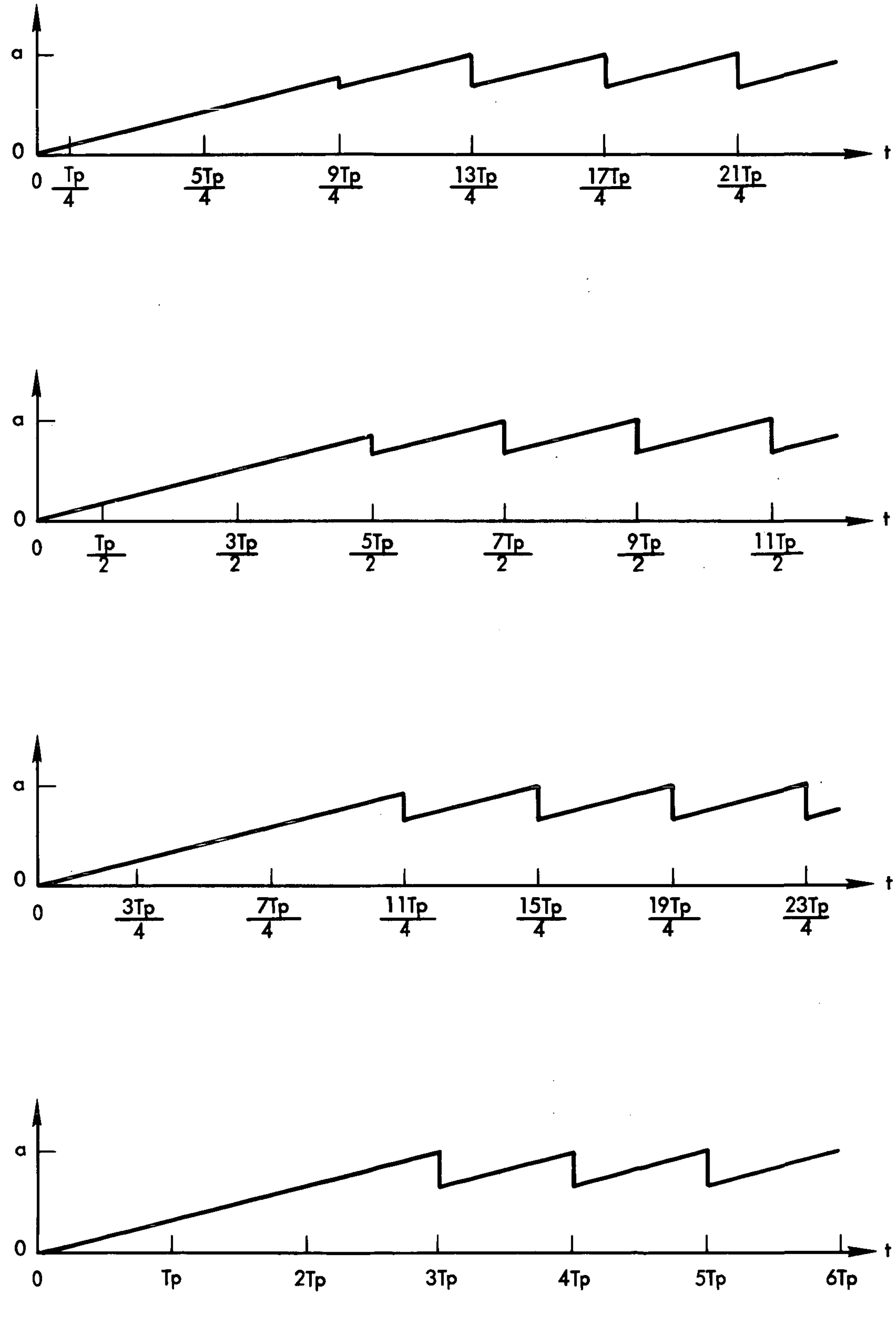


Figure 3-6. Prefilter Responses to a Step of Magnitude a

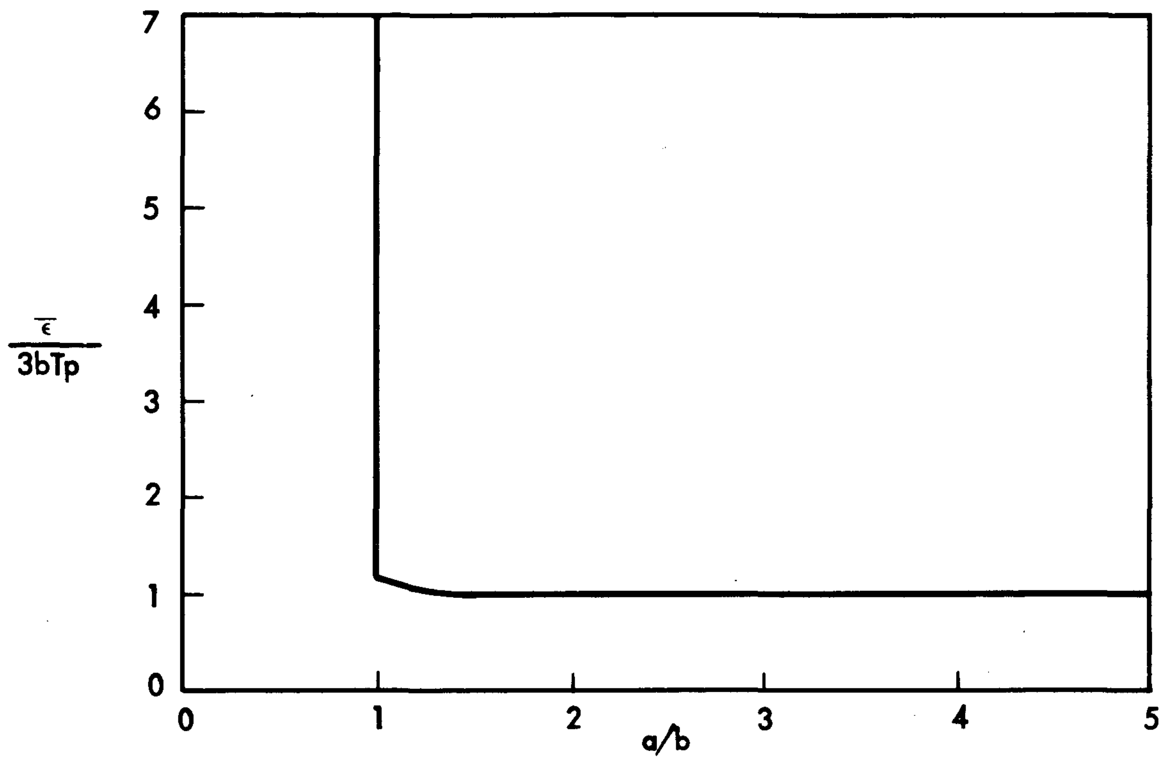


Figure 3-7. Squared Error Algorithm Prefilter Error Versus Failure Magnitude

A comparison of Figure 3-7 with Figure 3-2 shows a more desirable characteristic for the Squared Error Algorithm prefilter. The error is always less than that of the simple first-order prefilter.

Now let us consider the effect of quantization errors. Referring to Figure 3-5, the analog integrator is actually the pulse accumulator with Z transform Tz/(z-1). This transform cancels the one in the right-hand box of Figure 3-3 so that

$$\omega_f = \frac{1}{3T_p}\left[e(t) - e(t-2T_p)\right] \tag{3-126}$$

Using the same quantization error model as before, we see that

$$\sigma_{\omega_f} = \frac{Q}{\sqrt{6}}\,\frac{1}{3T_p} \tag{3-127}$$

Since we see from the normalization procedure that τ_f and $3T_p$ are approximately equivalent, Eq. 3-127 shows an error due to quantization about $\sqrt{2}$ times larger than Eq. 3-122.

These results do not provide grounds for a clear choice between the two prefilters, since the low-pass filter has the better response to quantization error, and the Squared Error Algorithm prefilter has the better response to step failure. Any of these algorithms could use either prefilter. To be able to determine the relative merits of the algorithms themselves, the same prefilter will be used for all three. In addition, the Adaptive 66, Bayesian Decision Theory, and Minimax Algorithms, which contain no filtering at all, will be provided with the same prefilter. The first-order, low-pass type of prefilter is selected for this purpose because it is simpler. After a single algorithm is selected, a prefilter tradeoff study can be made in the future.

The Sequential Algorithm is essentially a modified Kalman-Bucy filter and therefore requires no prefilter. It is discussed in more detail in Paragraph 3.1.2.4 and Appendix B.

The Adaptive 72 Algorithm is considerably more elegant than any of the above in its filtering techniques. Originally it used a Kalman-Bucy filter to whiten the unfailed instrument errors prior to the detection system. Later, when the algorithm was implemented on SIRU, it was felt that the errors were close enough to white so as to make this filter unnecessary.

The detection system uses a filter which can be represented as shown in Figure 3-8 if the digital accumulator is represented by an analog integrator. This detection system is for positive polarity failures; a similar system is used for negative polarity failures.

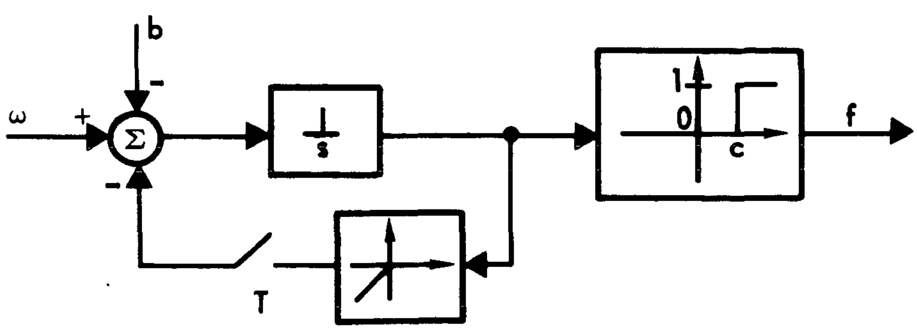


Figure 3-8. Detection System for Positive Polarity

The loop about the integrator resets it to zero at every sampling instant if the integrator output is less than zero. The constant b sets a level of angular velocity error or acceleration error below which no failures are detected. The constant c determines how quickly a failed instrument is switched out. The output f is 0 for no failure and 1 for a failure. For a step failure of magnitude a, the time required to reach the threshold level c is

$$t = \begin{cases} \infty & 0 < a/b \leq 1 \\ \dfrac{c}{a-b} & 1 < a/b < \infty \end{cases} \tag{3-128}$$

Following Eq. 3-109, and normalizing,

$$\frac{\epsilon}{b(c/b)} = \begin{cases} \infty & 0 < a/b \leq 1 \\ \dfrac{a/b}{a/b - 1} & 1 < a/b < \infty \end{cases} \tag{3-129}$$

This function is plotted in Figure 3-9. The error lies above that shown in Figures 3-2 and 3-7.

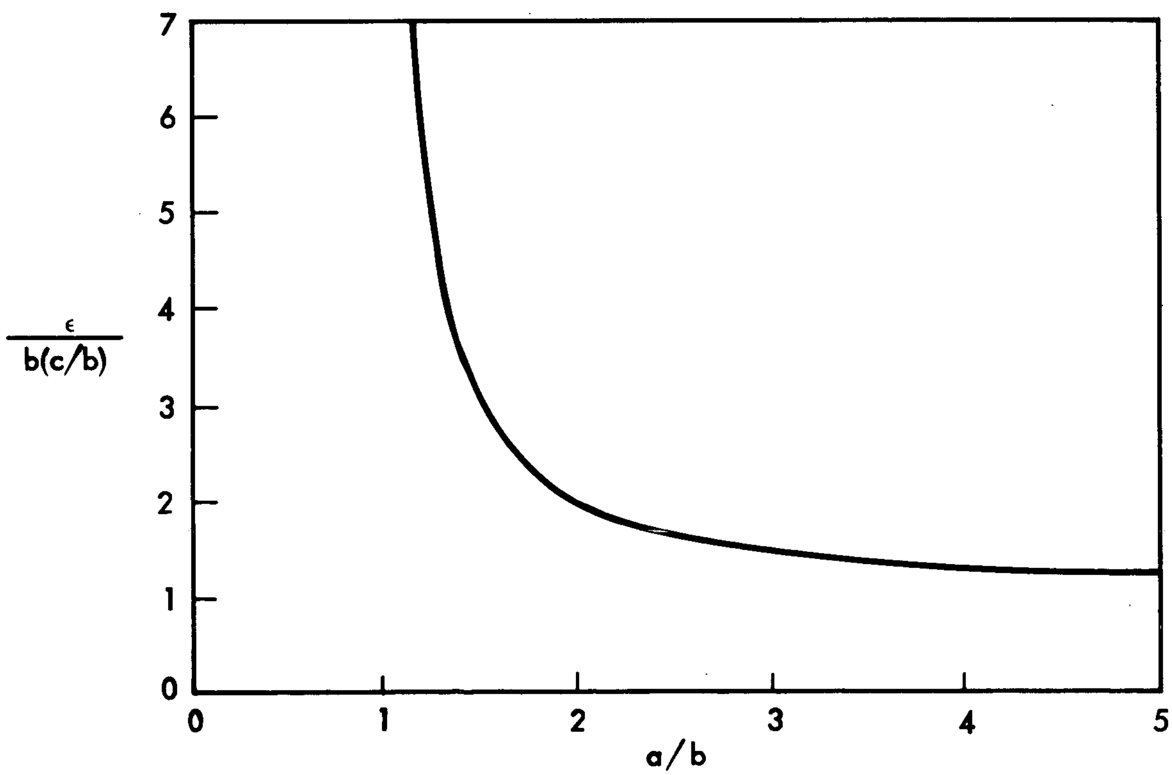


Figure 3-9. Adaptive 72 Algorithm Filter Error Versus Failure Magnitude

By analogy with the Squared Error Algorithm prefilter, we see that the equivalent quantization error response is

$$\sigma_{\omega_f} = \frac{\Omega}{\sqrt{6}} \frac{1}{c/b} \tag{3-130}$$

From the normalization process c/b is approximately equivalent to $3T_p$, so that the quantization error response of the two systems is comparable. It should be kept in mind that the above comparison does not take into account the nonlinear characteristic of the Adaptive 72 filter, which should have an important effect on the detection of signals in the presence of noise.

3.2 DETAILED ALGORITHM DEFINITIONS

The algorithms, as they appear in FAILSIM, are specified in detail in this section. However, the reader is referred to the original documentation of the algorithms for their explanations, derivations, or rationales, some of which are quite lengthy. Each algorithm seems to have been developed with a somewhat different set of assumptions about the nature of the problem being solved.

3.2.1 Adaptive 66 Algorithm

The Adaptive 66 Algorithm is described in References 3 and 7. It uses a weighted least-squares estimator to perform failure detection, isolation, and correction. The estimator estimates the instrument package input vector (acceleration or angular velocity). The residuals are used to estimate the variances of the instrument errors. The variances are used to obtain the weights of the estimator. New estimates are then obtained in an iterative fashion until the variances cease changing.

In the original algorithm, the equation for the variance in terms of the squared residuals is given by

$$Q_{ii} = \hat{\epsilon}_i^2 + \epsilon_o^2 , \quad i = 1, 6 \tag{3-131}$$

where ϵ_o^2 is the nominal or minimum value. The computation time can be reduced by several orders of magnitude if Eq. 3-131 is replaced by

$$Q_{ii} = \max \left(\epsilon_i^2 , \epsilon_o^2 \right), \quad i = 1, 6 \tag{3-132}$$

since the Q_{ii} in Eq. 3-132 will not change for small perturbations of the $\hat{\epsilon}_i$ caused by unfailed instrument errors. If the Q_{ii} do not change, it is unnecessary to recompute the correction matrix B. The change from Eq. 3-131 to Eq. 3-132 is necessary to prevent this algorithm from consuming an inordinate amount of simulation computer time. In addition to this change, a prefilter has been added which filters the instrument outputs prior to their use in the algorithm. Figure 3-10 shows the algorithm flow diagram. The Adaptive 66 Algorithm has two adjustable parameters, the prefilter time constant τ_f, and the a priori error variance ϵ_o^2.

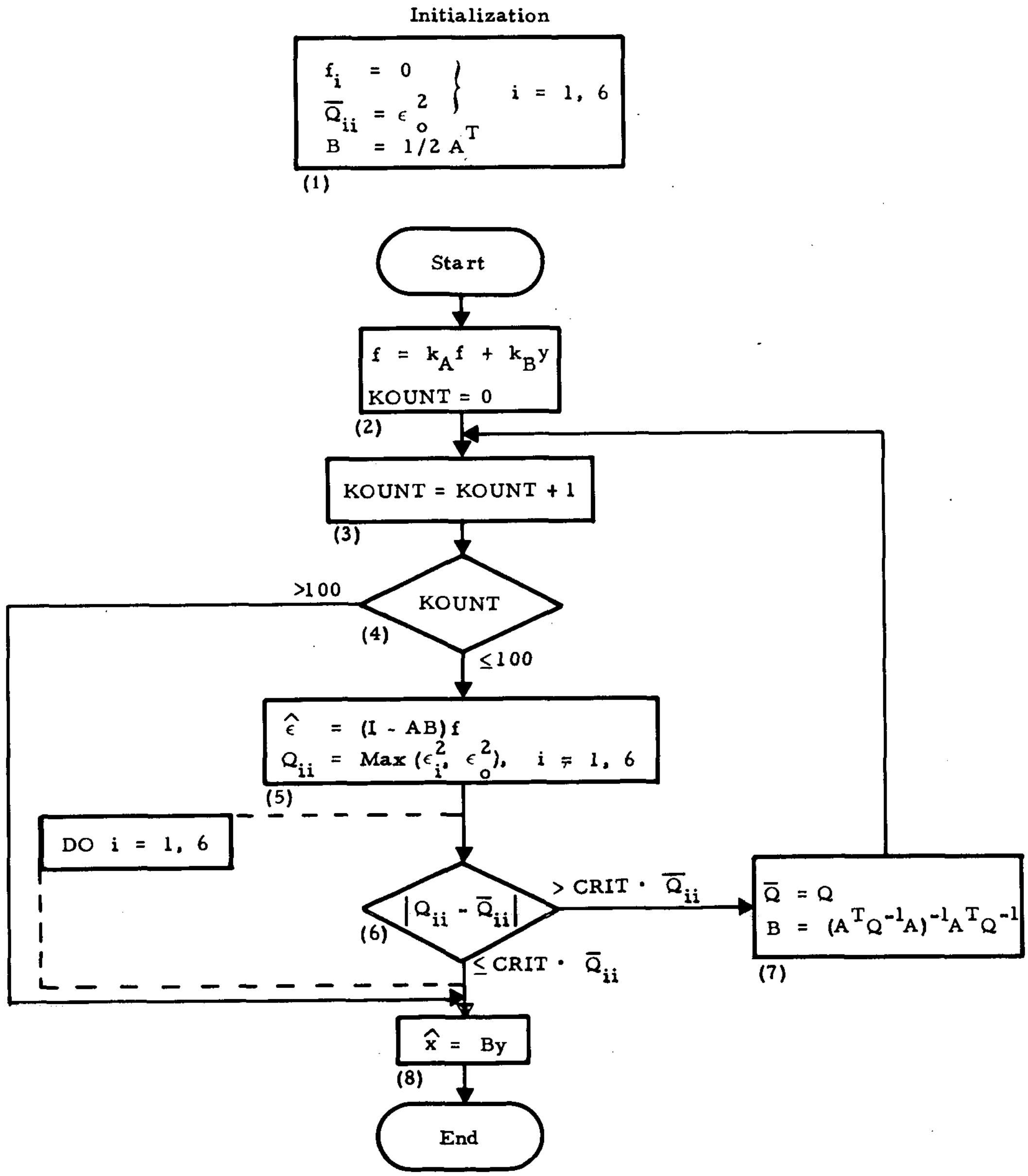


Figure 3-10. Adaptive 66 Algorithm Flow Diagram

The flow diagram description follows:

(1) The filter 6 vector $\underline{f}$, the previous value of the instrument covariance matrix $\overline{Q}$, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the filter outputs are calculated. The iteration counter is zeroed.

(3) The iteration count is increased by one.

(4) If 100 iterations have occurred, the iteration loop is terminated.

(5) The residuals are calculated from the filter outputs, and a new covariance matrix is obtained.

(6) If the changes in all of the diagonal elements of the covariance matrix from their previous values are less than the factor CRIT times their previous values, the iteration loop is terminated.

(7) The previous value of the covariance matrix is set equal to the new value. The correction matrix is updated. A new iteration is performed.

(8) The package input is estimated from the unfiltered instrument outputs.

3.2.2 Fifteen Threshold Algorithm

The Fifteen Threshold Algorithm is described in References 8, 9, and 10. It makes use of the fifteen indirect test signals described in Paragraph 3.1.2.1. Each element of the test signal vector u is compared to a threshold. A vector w is used to store the results of this comparison. If the threshold is exceeded by $|u_i|$, then w_i is set equal to 1. If $|u_i|$ later becomes less than the threshold, then w_i is not reset to 0. Thus there is no provision for reinstatement of "healed" instruments.

An instrument is considered failed only if all of the w_i to which it contributes a signal are equal to 1. Eqs. 3-12 and 3-14 are used for failure correction to save time in the simulation, although the original algorithm used Gauss-Jordan reduction so that, in a real-time system, a change to the failed/unfailed status could be implemented within a single minor cycle. Figure 3-11 shows the algorithm flow diagram.

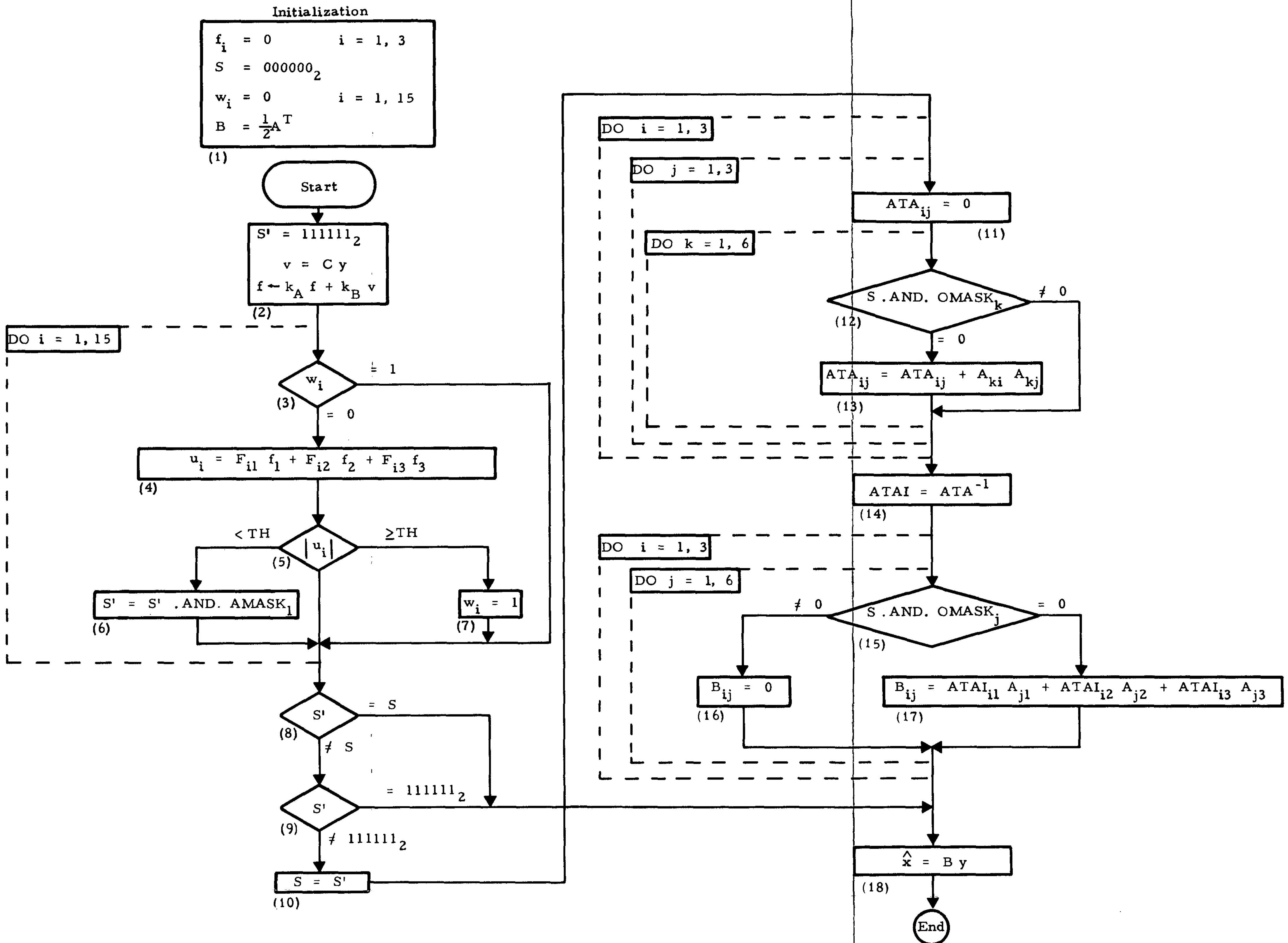


Figure 3-11. Fifteen Threshold Algorithm Flow Diagram

The variables S and S' are 6-bit binary numbers whose bits correspond to the unfailed (0) or failed (1) states of the instruments. S is the instrument state. S' is set to all ones at the beginning of each cycle. Whenever a test signal is found to be less than the threshold, the 1 bits corresponding to the instruments whose outputs contribute to the signal are masked out by the appropriate mask:

i	$AMASK_i$					
1	0	0	0	0	1	1
2	0	0	0	1	0	1
3	0	0	0	1	1	0
4	0	0	1	0	0	1
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	1	0	0	0	1
8	0	1	0	0	1	0
9	0	1	0	1	0	0
10	0	1	1	0	0	0
11	1	0	0	0	0	1
12	1	0	0	0	1	0
13	1	0	0	1	0	0
14	1	0	1	0	0	0
15	1	1	0	0	0	0

Those instruments whose bits are not masked out are failed. The state of a particular instrument is determined by masking out the states of the other instruments with the appropriate mask:

i	$OMASK_i$					
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

The test signals are prefiltered before use in the algorithm. The Fifteen Threshold Algorithm has two adjustable parameters, the prefilter time constant τ_f and the threshold level TH.

The flow diagram description follows:

(1) The filter 3 vector f, the instrument state S, the threshold state 15 vector w, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the intermediate instrument states S' are set to all ones, and the inconsistency state vector v is calculated and filtered.

(3) The 15 threshold states are examined. If the i^{th} state is equal to 1, it is ignored.

(4) If it is equal to 0, the i^{th} parity signal is calculated.

(5) The i^{th} parity signal is compared to the threshold.

(6) If the threshold is not exceeded, all of the bits corresponding to instruments included in the test signal are masked out to 0.

(7) If the threshold is exceeded, the threshold flag is set equal to 1.

(8) If S = S', no change in failure status has occurred, and the correction matrix calculation is bypassed.

(9) If S' is all ones, more than two failures have been detected and isolation is not possible. S is not changed, and the correction matrix calculation is bypassed. Since it cannot resolve the situation, the algorithm continues to use the old correction matrix.

(10) S is set equal to S', and the new correction matrix B is calculated. The new B matrix is calculated from Eq. 3-14 with the rows of A corresponding to the failed instruments taken as zero.

(11) The matrix $A^T A$ is zeroed.

(12) The k^{th} instrument is tested for failure.

(13) If the k^{th} instrument is unfailed, its contribution to $A^T A$ is calculated and added in.

(14) $(A^T A)^{-1}$ is calculated.

(15) The j^{th} instrument is tested for failure.

(16) If the j^{th} instrument is failed, the j^{th} row of B is set equal to zero.

(17) If not, the j^{th} row of B is calculated.

(18) The package input is estimated from the unfiltered instrument outputs.

3.2.3 Squared Error Algorithm

The Squared Error Algorithm is described in References 11, 12, 13, and 14. The algorithm actually used in FAILSIM differs in several ways from these descriptions. The differences and the reasons for them will now be discussed.

The prefilter used with the Squared Error Algorithm in SIRU consists of an accumulator into which the instrument outputs are summed. Every 60 sec, the accumulator is reset so that it contains only the last 120 sec worth of accumulated output. Thus the data accumulation period fluctuates from 120 to 180 sec. The prefilter used in FAILSIM is a simple first-order, low-pass filter, of the kind used for most of the other algorithms. The reason for this change is that the prefilter is not an integral part of the different algorithms. If the SIRU prefilter should be superior to the low-pass prefilter, all of the algorithms can make use of it. The purpose of FAILSIM is to compare the basic algorithms with each other. This can best be done if the prefilters are the same. After a single algorithm is selected, a prefilter tradeoff study can be made in the future.

The Squared Error Algorithm automatically raises the failure detection thresholds for the gyros when the angular velocity becomes large, in order to reduce the probability of false alarms caused by instrument dynamic errors, scale factor errors, and misalignments. In SIRU, the signal used for this purpose is the square of the sum of the absolute values of the three components of angular velocity. This signal is not a scalar; that is, its value depends upon the choice of coordinate system. Since the choice of coordinate system is arbitrary, the signal is characterized by an undesirable quality of arbitrariness. It is more usual to use invariant or covariant expressions, i.e., scalars, vectors, and tensors. Therefore the squared magnitude of the angular velocity vector will be used for this purpose in FAILSIM, since it is a scalar.

If a spike of angular velocity occurs and causes a spike of gyro errors, the threshold would be raised as described above during the spike and then lowered. The filtered inconsistency states, however, would exhibit a decaying exponential at the filter time constant, which could cause the now lowered threshold to be exceeded. (It would seem that the SIRU prefilter would also experience a form of this problem.) Therefore, instead of raising the squared error threshold, the FAILSIM algorithm lowers the gain at the input to the prefilter. Since the gain acts upon the unsquared signals, it is taken as the square root of the reciprocal of the gain applied to the squared error threshold in SIRU.

The logic in the SIRU algorithm provides that, when one failure has been detected, the algorithm will alternate between testing for a second failure and testing for healing of the first failure, so that each of these two tests occur only every other minor cycle. This procedure is required because of the limited time available in the particular computer used for SIRU. However, since all of the other algorithms are allowed all the time they need, it seems more equitable to modify the logic so that both tests take place each minor cycle, and this action is taken.

The SIRU algorithm provides checks for overflow of the error quantities. Since overflow is not a problem for the CDC 6500, these checks have been eliminated.

The SIRU algorithm detects the third failure. However, it does nothing with the information other than output it. Therefore, this capability is deleted to conserve computing time.

The squared-error calculations have been modified to make the coding simpler. However, the results are numerically identical to those of the original equations. When zero instruments are failed, the algorithm makes use of the residuals as estimates of the instrument error. If prefiltering is performed, we have from Eq. 3-21

$$\hat{\epsilon} = \frac{1}{2} C^T f \qquad (3\text{-}133)$$

The total squared error for zero failures is defined as

$$TSE_0 = \hat{\epsilon}^T \hat{\epsilon} \tag{3-134}$$

From Eqs. 3-5, 3-133, and 3-134 we have

$$TSE_0 = \frac{1}{2} f^T f \tag{3-135}$$

When the k^{th} instrument has failed, the residuals are calculated as in Eq. 3-23

$$\hat{\epsilon}(k) = \frac{1}{2} C(k)^T f \tag{3-136}$$

The total squared error for the remaining instruments is

$$TSE_k = \hat{\epsilon}(k)^T \hat{\epsilon}(k) \tag{3-137}$$

Note that because of Eq. 3-26, the residual of the failed instrument is automatically dropped from TSE_k.

The total squared error is compared to a threshold. If the threshold is exceeded, then the ratios of the individual squared errors (squared residuals) to the total squared error are tested against another threshold. If that threshold is exceeded by one of the ratios, the corresponding instrument is considered failed. The same procedure is used for the second failure. A flag LHEAL controls the healing process.

LHEAL = 0 healing is not allowed
LHEAL = 1 healing is allowed

Figure 3-12 shows the algorithm flow diagram. If the flag LHEAL is zero, instruments are not reinstated even though healing may be observed. The Squared Error Algorithm has six adjustable parameters, the prefilter time constant τ_f, the threshold levels K_1, K_2, K_3, and K_4, and the gyro dynamics compensation gain k_{GD}.

The flow diagram description follows:

(1) The filter 3 vector f, the instrument state S, the failed status indicators I and J, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the filter gain is calculated as a function of the previous angular velocity estimate (gyros only).

(3) The inconsistency state vector v is calculated and filtered, the total squared error is found, and the present values of S and I are stored.

(4) The total squared error is tested against threshold K1.

(5) If the threshold is exceeded, the residuals are calculated.

(6) The ratios of the squared residuals to the total squared error are tested against threshold K2. If the threshold is exceeded, the loop terminates and control passes to Step 7. If no ratio exceeds the threshold, control passes to Step 16.

(7) K is the number of the failed instrument just detected. I is the number of the "first" failed instrument, and J is the number of the "second" failed instrument. If I already equals K, no further action is required, and control passes to Step 10.

(8) K is tested against J. If it equals J, then the values of I and J are exchanged in Step 11.

(9) If not, I is tested against 0. If I equals 0, no failure was previously detected, and I is set equal to K in Step 13.

(10) If not, J is tested against 0. If J equals 0, only one failure was previously detected.

(11) Its instrument number is stored in J, and I is set equal to K.

(12) If J is not equal to zero, then there were two previous failures, neither one of instrument K. If healing is not possible, no further action is taken.

(13) If healing is possible, I is set equal to K and J is zeroed.

(14) If healing is not possible, total squared error less than the threshold has no effect.

(15) If healing is possible, any previous failed instruments are reinstated when the total squared error is less than the threshold.

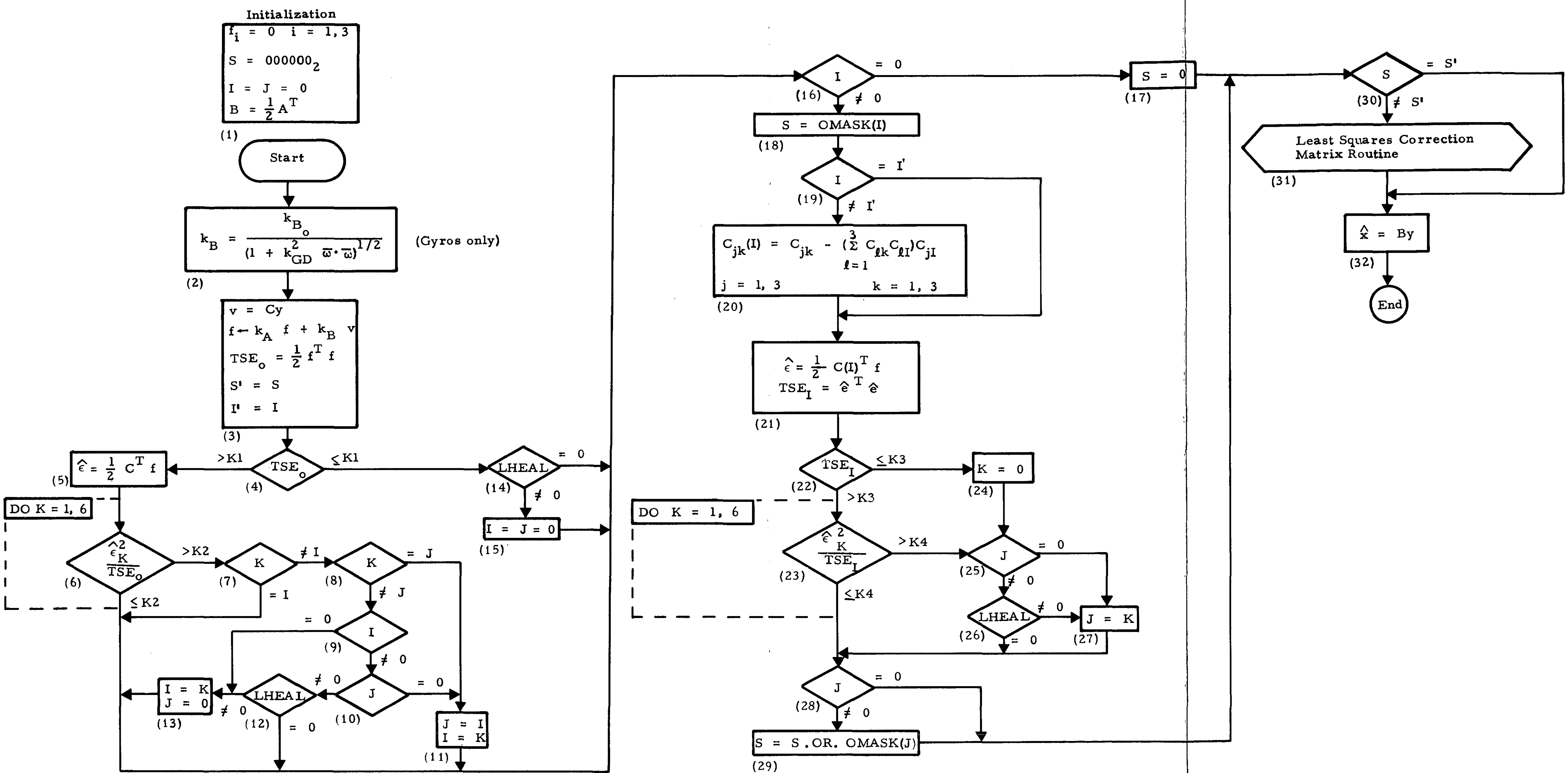


Figure 3-12. Squared Error Algorithm Flow Diagram

FOLDOUT FRAME 1

FOLDOUT FRAME 2

(16) If I equals zero, no instruments are failed.

(17) The instrument states are zeroed.

(18) If I is not zero, the I^{th} instrument state is set equal to 1.

(19) If I is unchanged Step 20 is bypassed.

(20) The matrix for calculating the residuals from the filter vector when instrument I is failed is calculated.

(21) The residuals and total squared error (excluding instrument I) are calculated.

(22) The total squared error is tested against threshold K3.

(23) The ratios of the squared residuals to the total squared error are tested against threshold K4. If the threshold is exceeded, the loop terminates, and control passes to Step 25. If no ratio exceeds the threshold, control passes to Step 28.

(24) If the total squared error is less than the threshold, K is set equal to 0.

(25) If J equals 0, there has been no "second failure."

(26) If J is not equal to zero, then the healing flag is tested.

(27) If J equals 0 or healing is permitted, J is set equal to K.

(28) If J equals 0, Step 29 is bypassed.

(29) If J is not equal to zero, the J^{th} instrument state is set equal to 1.

(30) If S equals S', no change in failure status has occurred, and the correction matrix calculation is bypassed.

(31) The new B matrix is calculated from Eq. 3-14 with the rows of A corresponding to the failed instruments taken as zero. See Steps 11 to 17 in Paragraph 3.2.2 for the detailed description.

(32) The package input is estimated from the unfiltered instrument outputs.

3.2.4 Bayesian Decision Theory Algorithm

The Bayesian Decision Theory Algorithm is described in References 15 and 16. Actually, there are two algorithms derived in

Reference 15: one for "mean failures" and one for "variance failures." Apparently these terms refer to the extremes of constant and white noise errors. Reference 15 is not completely transparent to this reader, and since Reference 16 sets forth the mean failure algorithm clearly, the mean failure algorithm will be chosen.

The mean failure algorithm presented in References 15 and 16 is in error, in that it produces incorrect decisions for failures that lead to errors with negative polarity. This shortcoming can be remedied easily by the judicious introduction of absolute value operations into all of the decision equations.

Once this is done, it becomes obvious that the 21 decision equations set forth for six instruments involve only six different test signals and that these test signals are proportional to the residuals, $\hat{\epsilon}$, of Eq. 3-17. Furthermore, the decision tree reduces to the selection of the test signal having the largest absolute value and the comparison of it to a threshold. If it exceeds the threshold, the associated instrument is failed. If it does not, zero instruments are failed.

When one instrument is failed, the decision equations for the remaining five instruments must be implemented. The test signals used in References 15 and 16 are arbitrary and have unequal variances representing the errors of the other instruments in unfailed operation. Consideration of the symmetry of the dodecahedron configuration leads one to believe that, if the test signals were chosen to minimize their variances, their variances would all be equal to each other and less than or equal to the variances of the test signals of References 15 and 16.

This guess can be shown to be correct, and the minimum variance test signals turn out to be the residuals, $\hat{\epsilon}$, of Eq. 3-23. Therefore they will be used in place of the original test signals. Again the decision tree reduces to the selection of the test signal having the largest absolute value, and the comparison of it to a threshold, to determine whether or not a second failure is present.

To reduce the frequency of false alarms, Reference 16 calls for two consecutive determinations of a failure before an instrument is switched out. However, this process will cause an increase in missed alarms not taken into account in the previous analysis in Reference 16. It seems clear that this process is approximately equivalent to raising the detection threshold. Therefore, it will be ignored. The thresholds will be set experimentally rather than theoretically at any rate.

Reference 16 also calls for two sampling periods, 10 min for soft failures and 0.01 sec for hard failures. The 10-min period samples are averaged over the preceding sampling period. This procedure represents another prefilter, resembling somewhat the SIRU prefilter. The simple low-pass filter will be used instead for the same reasons as given for the Squared Error Algorithm in Paragraph 3.2.3.

It is clear from the foregoing that considerable liberties have been taken with Gully's original concept. This course of action was necessary because his scheme was further from a "reduction to practice" than the others. In summary, when no failures have been detected, the largest residual of Eq. 3-17 is compared against a threshold. If it exceeds the threshold, the corresponding instrument is failed. When one failure has been detected the largest residual of Eq. 3-23 is compared against a threshold. If it exceeds the threshold, the corresponding instrument is failed. No healing is permitted. Figure 3-13 shows the algorithm flow diagram.

The Bayesian Decision Theory Algorithm has three adjustable parameters: the prefilter time constant τ_f and the threshold levels TH_1 and TH_2.

The flow diagram description follows:

(1) The filter 3 vector f, the instrument state S, the failed status indicator k, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the failed status indicator is tested. If it equals 7, two failures have occurred, and the failure detection and isolation logic is bypassed.

(3) If 0 or 1 failure has occurred, the inconsistency state vector v is calculated and filtered.

(4) If k is not equal to 0, one failure has been isolated, and the first failure detection logic is bypassed.

(5) If no failures have been isolated, the residuals are calculated, and the loop to find the maximum residual is initialized. j is the number of the instrument having the largest residual and u_{max} is the value of its magnitude.

(6) If the magnitude of the i^{th} residual is less than u_{max}, then Step 7 is bypassed.

(7) j is set equal to i, and u_{max} is set equal to the magnitude of u_i if $|u_i|$ is greater than u_{max}.

(8) If the magnitude of the largest residual is less than the threshold, no failure is detected and the rest of the failure detection logic is bypassed.

(9) Instrument j is failed. k is set equal to j. The j^{th} instrument state is set equal to 1. m is set equal to 1 to show that a change has occurred in the instrument states S. The matrix for calculating the residuals from the filter vector when instrument k is failed is calculated.

(10) If one failure has already been isolated, m is set equal to 0 to show that no change in S has occurred.

(11) The residuals (excluding instrument k) are calculated. The loop to find the largest residual is initialized.

(12), (13) The largest residual is found as in Steps 6 and 7.

(14) If the magnitude of the largest residual is less than the threshold, no second failure is detected.

(15) If a second failure is detected, k is set equal to 7 and the j^{th} instrument state is set equal to 1.

(16) If no second failure is detected, the correction matrix calculations are bypassed unless m equals 1, showing that a first failure was detected in this minor cycle.

(17) The new B matrix is calculated from Eq. 3-14 with the rows of A corresponding to the failed instruments taken as zero. See Steps 11 through 17 in Paragraph 3.2.2 for the detailed description.

(18) The package input is estimated from the unfiltered instrument outputs.

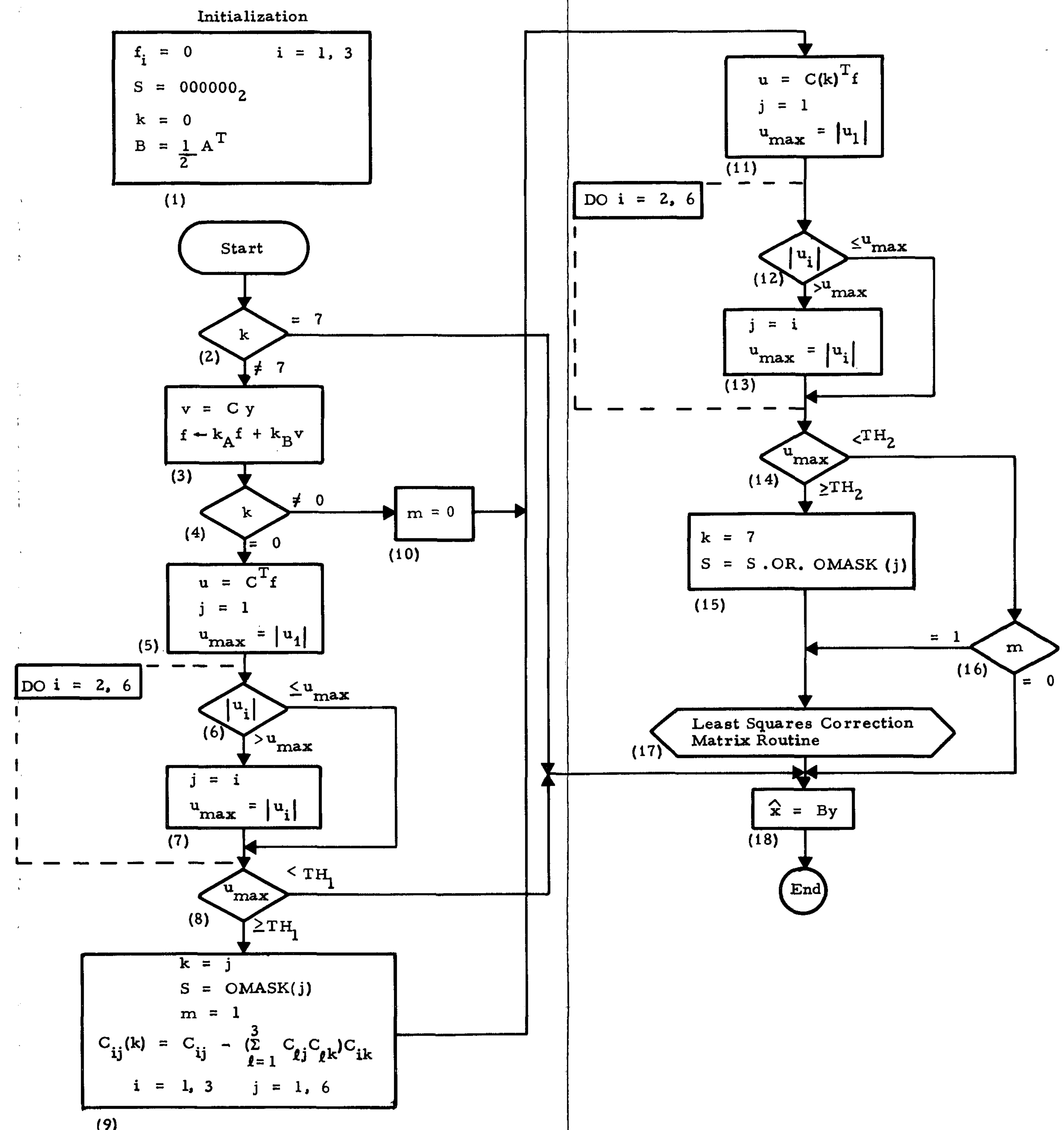


Figure 3-13. Bayesian Decision Theory Algorithm Flow Diagram

3.2.5 Maximum Likelihood Algorithm

The Maximum Likelihood Algorithm is described in References 17 and 18. A maximum likelihood technique is used to obtain estimates of the instrument variances from the inconsistency vector v. The instrument variances are then used in a weighted least-squares estimator. The inconsistency vector is modified by subtraction of the residuals to reduce the interaction of successive failures. The squared residuals are added to the variance estimates to compensate for the decrease that would otherwise result from the subtraction of the residuals.

The prefilter in the original algorithm, which was imbedded within the algorithm itself, proved to be unsuccessful. Therefore the instrument outputs are prefiltered prior to use in the algorithm. Figure 3-14 shows the algorithm flow diagram. The Maximum Likelihood Algorithm has two adjustable parameters, the prefilter time constant τ_f and the a priori error variance Q_o.

The flow diagram description follows:

(1) The residuals $\hat{\epsilon}_i$, the filter 6 vector f, the previous value of the weighting matrix $\overline{Q}$, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the filter outputs are calculated. The residuals are subtracted from the filter outputs, and the inconsistency state vector is calculated. The maximum likelihood variance estimates Q' are found and modified to account for a priori information and to compensate for the subtraction of the residuals to give the instrument covariance matrix Q. The iteration counter is initialized.

(3) If the change in any one of the diagonal elements of the covariance matrix is greater than the factor CRIT times its previous value, a new correction matrix is required.

(4) The iteration count is increased by one. The previous value of the covariance matrix is set equal to the new value. The correction matrix is updated.

(5) The new residuals are found. The new covariance matrix is obtained. (v is now zero, so that Q' is eliminated from the expression for Q.)

(6) If 100 iterations have occurred, the iteration loop is terminated.

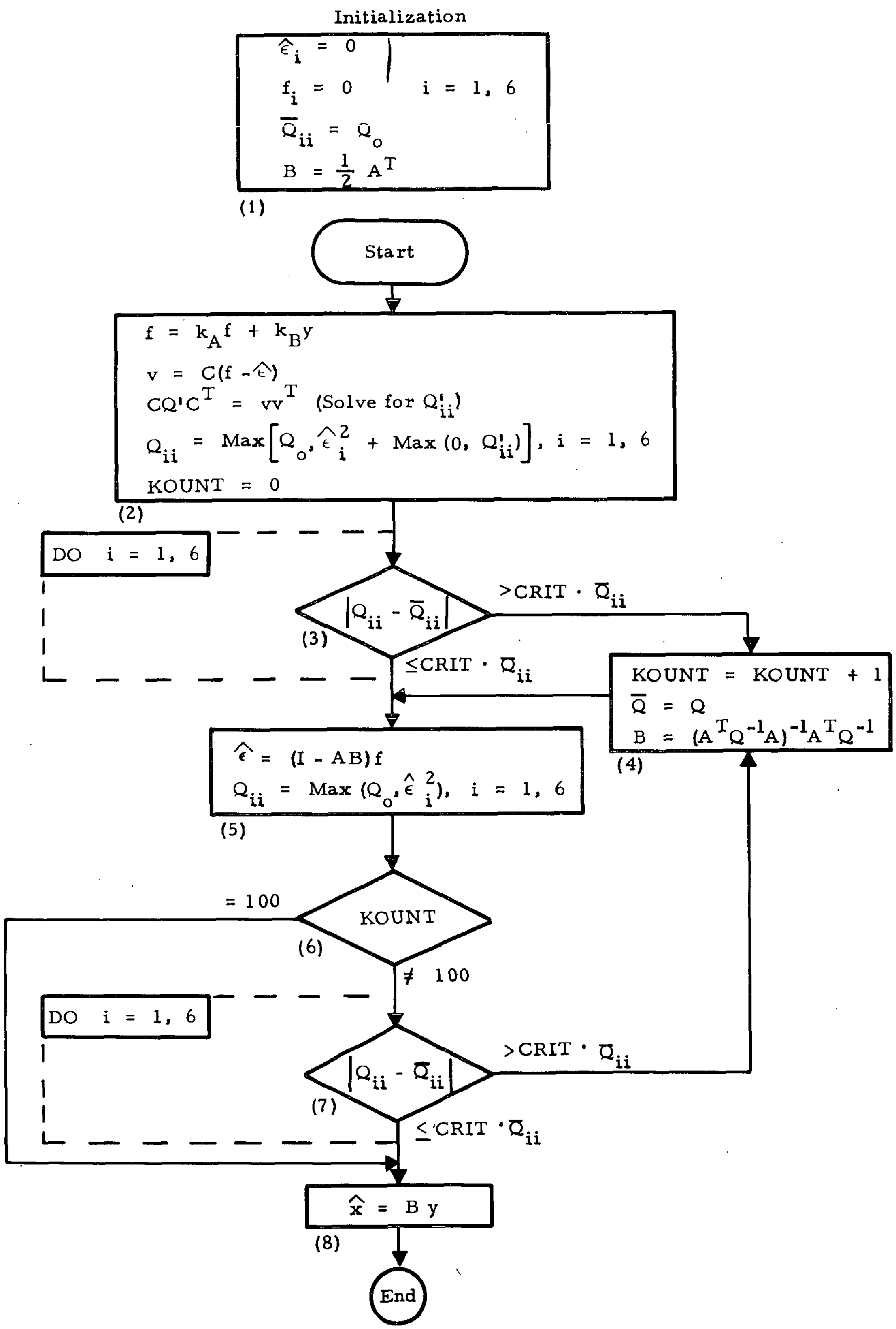


Figure 3-14. Maximum Likelihood Algorithm Flow Diagram

(7) If the change in any one of the diagonal elements of the covariance matrix is greater than the factor CRIT times its previous value, a new correction matrix is required.

(8) The package input is estimated from the unfiltered instrument outputs.

3.2.6 Minimax Algorithm

The Minimax Algorithm is described in References 19, 20, 21, 22, 23, 24, 25, and 26. It uses the 15 parity signals described in Paragraph 3.1.2.3. The original algorithm uses no filtering at all, so a prefilter in the form of a simple low-pass filter has been provided to put it on an equal footing with the other algorithms. This prefilter rejects the large, high-frequency rate errors caused by the instrument quantization.

The original algorithm (as detailed on a MAC listing) solved for the parity equation weights (±c and ±s in Eq. 3-5) each time through in order to handle the gimballed case where the relative instrument orientations can change. As pointed out by the authors, this procedure is not necessary for the dodecahedron strapdown case. Thus the formulation of Eqs. 3-57 and 3-58 is used instead. The parity signals used in the Minimax Algorithm are normalized by a factor of 1/(2c+2s) so that, if the instrument outputs take on the values ±1, the magnitude of the normalized parity signal lies in the range ±1. In FAILSIM this factor is included in the conversion of the threshold levels from μg and deg/hr to m/sec^2 and rad/sec.

Failure detection takes place when any one of the parity signals exceeds a threshold. The original algorithm has a technique for finding the maximum parity signal without calculating all of them. When detection occurs, a second loop calculates all of the parity signals to perform isolation. Since the calculation of the parity originals is much simplified by the use of Eqs. 3-57 and 3-58, this procedure is modified. Both detection and isolation are combined in a single loop in which all of the parity signals are calculated. Neither this change nor the one in the preceding paragraph changes the action of the algorithm in any way.

For isolation of the first failure, it is assumed that only one failure can occur at a time. The four instruments that contribute to the value of a particular parity signal are called a quartet. If the signal exceeds its threshold, the quartet is dirty; if not, it is clean. If no instruments have failed, all quartets must be clean. Thus, a failure is detected if any quartet is found to be dirty.

With six instruments and one failure, there must be one set of five good instruments. With a set of five instruments called a quintet, the isolation algorithm finds every quintet whose quartets are all clean. Such a quintet may be a set of good instruments and will be called a clean quintet. An instrument which is excluded from every clean quintet must be failed since a good instrument would be in some clean quintet. Thus the failed instrument is the one excluded from all clean quintets.

For the second failure, there are five parity signals which are independent of the output of the first failed instrument. When four of these exceed their thresholds, the second failed instrument is successfully isolated. The zeros in the equation for the sole clean parity signal correspond to the two failed instruments. The algorithm can isolate those double simultaneous failures which cause all but one parity signal to exceed its threshold.

It turns out that if a first failure causes all of the parity signals to which it contributes with a coefficient of $\pm c$ to exceed the threshold, the Minimax logic will cause the failure to be isolated. However, in order to be isolated, a second failure must cause four parity signals to exceed the threshold. It contributes to two of them with a coefficient of $\pm c$ and two with a coefficient of $\pm s$. Thus, in effect, the threshold for second failures is higher than the threshold for first failures by the ratio c/s or 4.2 dB.

Failure correction may be accomplished either with the least-squares technique or with the Bounding Sphere Algorithm. This algorithm minimizes the maximum possible estimation error. It is capable of modifying its action when failures are detected but not isolated unlike the least-squares algorithm. See the references for a description of this algorithm.

Figure 3-15 shows the algorithm flow diagram. (The Bounding Sphere Algorithm flow diagram is not included since none has been seen

Page intentionally left blank

by the author. The Bounding Sphere FORTRAN coding was obtained by translating MAC.) The Minimax Algorithm has two adjustable parameters, the prefilter time constant τ_f, and the threshold level THM. The Bounding Sphere Algorithm also requires a threshold level THS. In the absence of a prefilter, these levels are theoretically identical. The original algorithm makes no distinction between them.

The flow diagram description follows:

(1) The filter 3 vector, the list of the numbers of the active instruments MN_i, the number of active instruments n, the number of detected but unisolated instruments CNO, the instrument state S, and the correction matrix B are initialized before the trial.

(2) Each minor cycle the inconsistency state vector v is calculated and filtered. The present value of S is stored. $QUINT_i$, representing the quintet of instruments excluding instrument i, is zeroed. IFAIL and NUM are set equal to 0. The parity signal index p is set equal to 1.

The four indices i, j, k, and l are the numbers of the instruments included in the p^{th} quartet. The four DO loops cause p to run from 1 to 15 selecting the rows of the matrix F in Eq. 3-58. Thus the parity signals of Eq. 3-53 are calculated in the order shown in Eq. 3-54.

(3) If none of the instruments in the p^{th} quartet are failed, IT is 0.

(4) If the quartet includes a failed instrument, it is ignored.

(5) The p^{th} parity signal is calculated.

(6) The parity signal is tested against the threshold.

(7) If the signal is less than or equal to the threshold, the p^{th} quartet is clean and the number of good quartets, NUM, is increased by 1.

(8) If NUM is not equal to 1, Step 9 is bypassed.

(9) The numbers of the instruments in the first good quartet are stored. (They will provide the solution for a double simultaneous failure if there is only one good quartet, i.e., if NUM = 1 in Step 29.)

(10) - (13) If the signal is greater than the threshold, the p^{th} quartet is dirty. The m^{th} quintet excludes instrument m. Therefore if i, j, k, and l are all not equal to m, the m^{th} quintet includes the p^{th} quartet and must be dirty, since its quartets are not all clean. If i, j, k, or l equals m, the quintet does not include one of the instruments of the quartet, so that the p^{th} quartet says nothing about the m^{th} quintet.

(14) If the m^{th} quintet is dirty, $QUINT_m$ is set equal to 1.

(15) IFAIL is set equal to 1 to show that at least one parity signal has exceeded the threshold.

(16) The parity signal index is increased by 1.

(17) If IFAIL equals 0, no failures have been detected.

(18) CNO is set equal to 0.

(19) If n equals 4, only 4 instruments are unfailed. The new failure indicated by IFAIL nonzero cannot be isolated.

(20) The number of detected but unisolated failures is assumed to be 1, CNO is set equal to 1.

(21) m is set equal to -1.

(22) If the i^{th} quintet is dirty, $QUINT_i$ is nonzero, and the rest of the loop is bypassed.

(23) If m is not equal to -1, this is not the first clean quintet.

(24) CS is set equal to the number of the first clean quintet. (If it turns out to be the only clean quintet, then CS will be the number of the failed instrument.)

(25) The clean quintet counter is increased by 1.

(26) If m is less than 0, there were no clean quintets. If m equals 0, there was one clean quintet. If m is greater than 0, there were two or more clean quintets.

(27) If NUM = 0, there were no clean quartets.

(28) If there were no clean quartets, it is assumed that there are three good instruments remaining. The number of failures detected but not isolated is the number of active instruments less 3.

(29) If NUM = 1, there was one clean quartet.

(30) The number of active instruments, m, is set equal to 4. The list of the numbers of the active instruments is changed to the values stored in Step 9. The number of detected but unisolated failures is 0.

(31) If there were two or more clean quartets, then 2 detected but isolated failures are assumed.

(32) There was one clean quintet. The index k is set equal to 1.

(33) If MN_i equals CS it is the number of the failed instrument.

(34) The new list of numbers of the active instruments is found excluding the failed instrument. k is increased by 1.

(35) The number of active instruments is reduced by 1. The number of detected but unisolated instruments is set equal to 0.

(36) The instrument state is updated.

(37) There were two or more clean quintets. One detected but unisolated failure is assumed.

(38) If LSQ equals 0, the Bounding Sphere Algorithm is to be used. If LSQ is not equal to 0, the least-squares algorithm is to be used.

(39) If S equals S', no change in failure status has occurred, and the correction matrix calculation is bypassed.

(40) The new B matrix is calculated from Eq. 3-14 with the rows of A corresponding to the failed instruments taken as zero. See Steps 11 through 17 in Paragraph 3.2.2 for the detailed description.

(41) The package input is estimated from the unfiltered instrument outputs.

(42) The Bounding Sphere Algorithm estimates the package input making use of CNO, which is ignored by the least-squares algorithm.

3.2.7 Adaptive 72 Algorithm

The Adaptive 72 Algorithm is described in References 27 and 28. It is the most complex of the eight algorithms studied and one of the most recent. Unfortunately, these two features combined to make it impossible to include it in FAILSIM as planned. The complexity mandated the use of the detailed program flow diagrams in writing the routines for FAILSIM but the diagrams were unavailable in time.

3.2.8 Sequential Algorithm

The Sequential Algorithm is described in Reference 29. It is also complex and new. The implementation presented here is based upon the algorithm as it existed when the reference was written. It has not been possible to include in FAILSIM the subsequent improvements made by its authors.

The Sequential Algorithm consists of a modified Kalman-Bucy filter. A detailed description of the filter is given in Paragraph 3.1.2.4. In this section we shall see how it is modified to perform FDIC.

In the original algorithm, any instrument whose error state exceeded a threshold was taken as failed. This logic has been modified so that only the instrument whose error state exceeds the threshold by the greatest amount in any single pass through the filter equations is taken as failed.

The nonlinearity required for failure detection is provided by two distinct threshold levels. One level is set quite high and is applied to the Kalman-Bucy residuals. If the threshold is exceeded, the associated instrument is excluded from the measurements. When the residual is again less than the threshold the instrument is reinstated. The second, much lower level, is applied to the instrument error states. An instrument whose error state exceeds the threshold by more than any other error state during a single pass through the filter equations is excluded permanently from use. The residual threshold is constant. The error state threshold is proportional to the standard deviation of the error state, as given by the filter covariance matrix. Figure 3-16 shows the algorithm flow diagram. The Sequential Algorithm has nine adjustable parameters as follows:

(1) Initial variance for package input states*

(2) Initial variance for instrument error states*

(3) State noise variance for package input states*

*The standard deviation is input to the program.

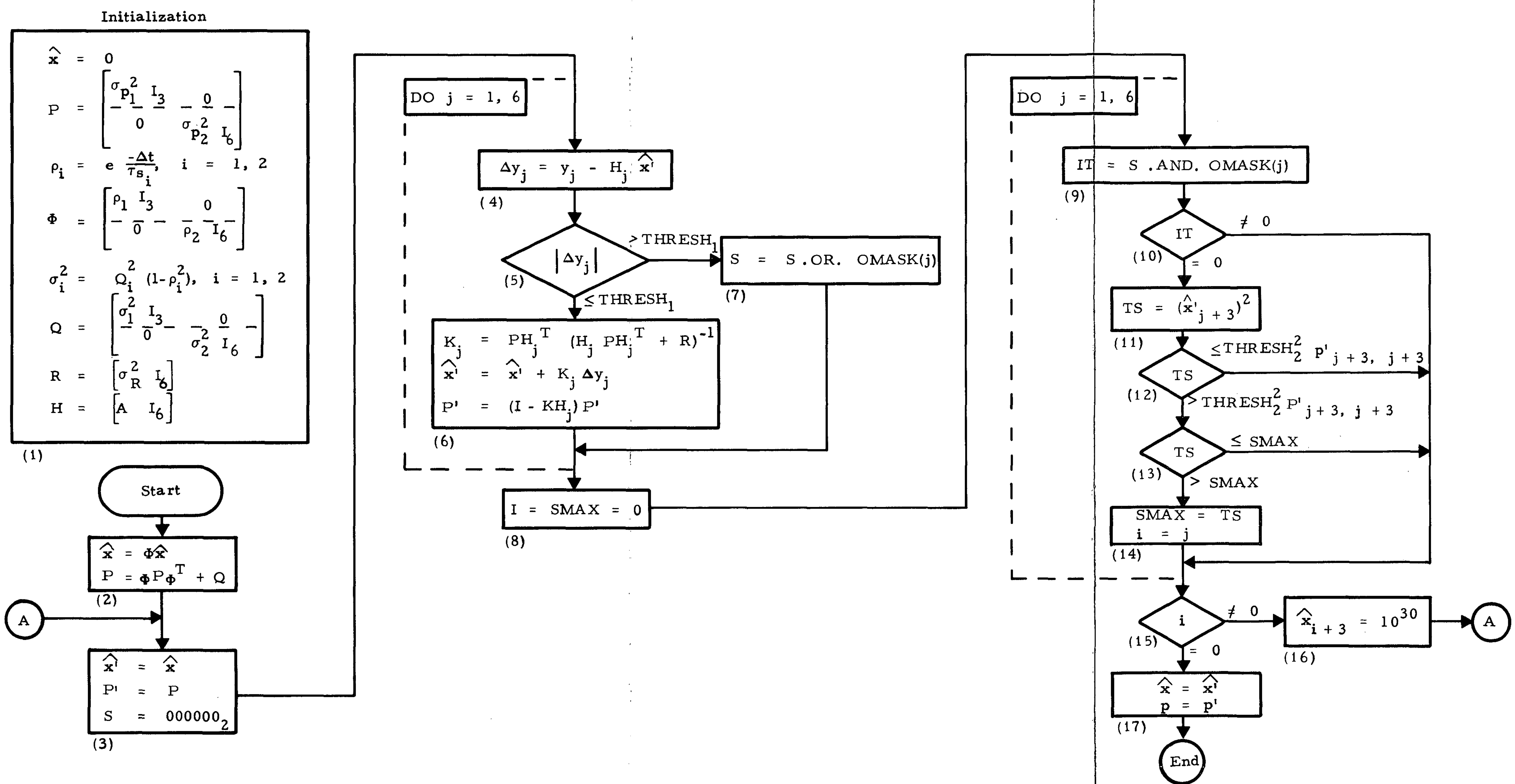


Figure 3-16. Sequential Algorithm Flow Diagram

(4) State noise variance for instrument error states*

(5) Variance for measurement errors*

(6) Time constant for package input states**

(7) Time constant for instrument error states**

(8) Threshold for residuals

(9) Ratio of threshold for instrument error states to instrument error standard deviation.

The flow diagram description follows:

(1) The estimated state 9 vector $\hat{x}$, the state covariance matrix P, the state transition matrix Φ, the state noise covariance matrix Q, the measurement noise matrix R, and the measurement matrix H are initialized before the trial.

(2) Each minor cycle the state vector and covariance matrix are advanced.

(3) Temporary working values of the state vector and covariance matrix, $\hat{x}'$ and P', are stored. The instrument state is zeroed.

(4) The j^{th} residual is calculated.

(5) The j^{th} residual is tested against the high level threshold.

(6) If the residual is less than the threshold, the filter matrix K_j is calculated, and $\hat{x}'$ and P' are reset.

(7) If the residual is greater than the threshold, the j^{th} instrument state is set equal to 1.

(8) The low-level failure detection and isolation process begins by setting i and SMAX to 0.

(9) If the j^{th} instrument was excluded from the current reset, IT will be nonzero.

(10) If the j^{th} instrument was excluded from the current reset, low-level failure detection is not performed.

(11) The estimate of the j^{th} instrument error is squared.

*The standard deviation is input to the program.
**Used to calculate the state transition matrix.

(12) If the square is less than the squared threshold times the instrument error variance, the instrument is not failed.

(13) - (14) At the end of the loop, SMAX will equal the squared error of the instrument having the largest error, and i will equal the number of the same instrument. If the error of no instrument exceeds the threshold, SMAX and i will be equal to 0.

(15) If i is not equal to 0, a failure has been detected.

(16) If a failure has been detected, the i^{th} instrument error estimate is set to a large value. Subsequently, the state of that instrument will always be set equal to 1 in Step 7. The filter calculations are now repeated with instrument i excluded.

(17) If no failures have been detected, the state vector and covariance matrix are updated from the working values. The first three elements of $\hat{x}$ comprise the output to the strapdown algorithm.

4. STRAPDOWN INERTIAL PACKAGE MODEL

The strapdown inertial package model simulates the operation of the package in the navigate mode only. The calibration and alignment functions are assumed to have been completed prior to the commencement of a computer run. Thus the errors simulated do not represent the total instrument and package errors, but only the residual errors remaining after calibration.

Because of computing time restraints, it is out of the question to simulate a cruise-type system for any significant length of time. Thus a boost-type system has been chosen. It should be adequate to allow the candidate algorithms to compete meaningfully against each other.

4.1 INSTRUMENT CONFIGURATION

The gyro and accelerometer input axis orientations with respect to the package axes are defined by the A matrix given in Eq. 3-2. It is assumed that the gyros and accelerometers are defined by the same A matrix (that is by the same dodecahedron orientation) because, although it is conceivable that a system could be designed with different dodecahedron orientations for the gyros and the accelerometers, nobody has done so, so that the added complexity required in the simulation to treat this case would probably be wasteful.

The orientation of each gyro and accelerometer about its input axis is defined by the angles $\theta_{g_1} - \theta_{g_6}$ and $\theta_{a_1} - \theta_{a_6}$, which are positive rotations about the input axis from a reference orientation. When an instrument is in the reference orientation, its output axis lies along the positive direction of one of the package axes. From the matrix A and the 12 orientation angles, it is possible to determine the 12 matrices AG1-AG6 and AA1-AA6 that relate individual gyro input, output, and spin axes and accelerometer input, pendulous, and output axes to package axes. We have, for the gyros,

$$\begin{Bmatrix} x_{input} \\ x_{output} \\ x_{spin} \end{Bmatrix} = AGi \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{Package} \qquad (4-1)$$

and, for the accelerometers,

$$\begin{Bmatrix} x_{input} \\ x_{pendulous} \\ x_{output} \end{Bmatrix} = AAi \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{Package} \qquad (4\text{-}2)$$

$$AG1 = \begin{bmatrix} c & s & 0 \\ sS_1 & -cS_1 & C_1 \\ sC_1 & -cC_1 & -S_1 \end{bmatrix}, \quad AA1 = \begin{bmatrix} c & s & 0 \\ -sC_1 & cC_1 & S_1 \\ sS_1 & -cS_1 & C_1 \end{bmatrix} \qquad (4\text{-}3)$$

$$AG2 = \begin{bmatrix} c & -s & 0 \\ -sS_2 & -cS_2 & C_2 \\ -sC_2 & -cC_2 & -S_2 \end{bmatrix}, \quad AA2 = \begin{bmatrix} c & -s & 0 \\ sC_2 & cC_2 & S_2 \\ -sS_2 & -cS_2 & C_2 \end{bmatrix} \qquad (4\text{-}4)$$

$$AG3 = \begin{bmatrix} 0 & c & s \\ C_3 & sS_3 & -cS_3 \\ -S_3 & sC_3 & -cC_3 \end{bmatrix}, \quad AA3 = \begin{bmatrix} 0 & c & s \\ S_3 & -sC_3 & cC_3 \\ C_3 & sS_3 & -cS_3 \end{bmatrix} \qquad (4\text{-}5)$$

$$AG4 = \begin{bmatrix} 0 & c & -s \\ C_4 & -sS_4 & -cS_4 \\ -S_4 & -sC_4 & -cC_4 \end{bmatrix}, \quad AA4 = \begin{bmatrix} 0 & c & -s \\ S_4 & sC_4 & cC_4 \\ C_4 & -sS_4 & -cS_4 \end{bmatrix} \qquad (4\text{-}6)$$

$$AG5 = \begin{bmatrix} s & 0 & c \\ -cS_5 & C_5 & sS_5 \\ -cC_5 & -S_5 & sC_5 \end{bmatrix}, \quad AA5 = \begin{bmatrix} s & 0 & c \\ cC_5 & S_5 & -sC_5 \\ -cS_5 & C_5 & sS_5 \end{bmatrix} \qquad (4\text{-}7)$$

$$AG6 = \begin{bmatrix} -s & 0 & c \\ -cS_6 & C_6 & -sS_6 \\ -cC_6 & -S_6 & -sC_6 \end{bmatrix}, \quad AA6 = \begin{bmatrix} -s & 0 & c \\ cC_6 & S_6 & sC_6 \\ -cS_6 & C_6 & -sS_6 \end{bmatrix} \qquad (4\text{-}8)$$

where

$$C_i = \cos \theta_{g_i} \text{ or } \cos \theta_{a_i} \qquad (4\text{-}9)$$

$$S_i = \sin \theta_{g_i} \text{ or } \sin \theta_{a_i} \qquad (4\text{-}10)$$

Note that the first row of AGi and AAi equals the i^{th} row of A.

Except for a relabeling of instruments and a rotation of 180 deg about certain of the instrument axes, the reference instrument orientations correspond to those of Gilmore's SIRU. The instrument labels and 180 deg rotations have no significance to a simulation of this type.

Table 4-I shows the relabeling required and the axes about which a 180-deg rotation is required to achieve correspondence between the gyro axes for the reference orientation and SIRU.

In SIRU the accelerometer input and output axes are the same as for the gyros, and the pendulous axis is the negative of the gyro spin axis. The same relationship holds in the simulation whenever the θ_{g_i} and θ_{a_i} are identical.

The MSFC breadboard dodecahedron (BB DDH) uses single-axis platforms, so that the directions of the output and spin axes vary with time. This feature is not modeled in the simulation, as it is felt that it would have no impact on the relative performance of the FDIC algorithms and would increase computer time requirements.

Table 4-I. Simulation — SIRU Correspondence

Simulation	SIRU	
Gyro	Gyro	Axis
1	C	Output
2	D	Spin
3	E	None
4	F	Spin
5	A	None
6	B	Input*

*This rotation may be accomplished by $\theta_{g_6} = 180$ deg.

The package axes are related to the body axes by a matrix R

$$x_{package} = R\, x_{body} \tag{4-11}$$

where R is defined as the product of three single-axis orthogonal rotation matrices

$$R = T\left(\theta_{R_3}, i_{R_3}\right) T\left(\theta_{R_2}, i_{R_2}\right) T\left(\theta_{R_1}, i_{R_1}\right) \tag{4-12}$$

where T(x, m) is a rotation through the angle x about the m^{th} axis. The angles θ_{R_1}, θ_{R_2}, and θ_{R_3} and the axis indicators i_{R_1}, i_{R_2}, and i_{R_3} are all read in as data. Note that the rotations from body-to-package axes are about axes i_{R_1}, i_{R_2}, and i_{R_3} through angles θ_{R_1}, θ_{R_2}, and θ_{R_3}, respectively, in that order. Since i_{R_1}, i_{R_2}, and i_{R_3} can each take on any of the values 1, 2, or 3, any order of rotations is possible.

4.2 UNFAILED INSTRUMENT ERRORS

The unfailed instrument simulation must exhibit errors, since it is the errors of the normally operating instruments that make the failure detection process difficult. However, the simulation of errors should not be excessively detailed and exhaustive because of the resultant penalties in computer time and storage. Therefore, a limited set of relatively easily calculated errors has been selected. Their combined characteristics are complex enough so that they do not correspond to any of the simple error models hypothesized in the derivation of the different algorithms. For the numerical values of the errors, see Paragraph 6.2.4, Appendix A, and Reference 30.

4.2.1 Gyro Errors

The error in a gyro output is given by

$$\begin{aligned} \epsilon = {} & E_1 + E_2\, n + E_3\, \omega_i + E_4\, \omega_o + E_5\, \omega_s \\ & + E_6\, a_i + E_7\, a_o + E_8\, a_s + E_9\, a_i\, a_s \end{aligned} \tag{4-13}$$

where the errors are:

(1) Bias drift rate

(2) Random drift rate

(3) Scale factor error

(4) Input axis misalignment about spin axis towards output axis

(5) Input axis misalignment about output axis towards spin axis

(6) Acceleration sensitivity along input axis (spin axis mass unbalance)

(7) Acceleration sensitivity along output axis

(8) Acceleration sensitivity along spin axis (input axis mass unbalance)

(9) Anisoelastic drift rate.

The standard deviations of the quantities E_1 through E_9 are input. Independent values of E_1 through E_9 are obtained for each gyro by the Monte Carlo method.

In addition to the above errors, there are errors introduced by the sampling and quantization process. The gyro output angular velocities are integrated over one sampling period. The result is then quantized. The error resulting from the quantization process is retained and added to the result of the integration over the next sampling period. This technique simulates the storage of quantization error information in the gyro float that occurs in an actual gyro.

4.2.2 Accelerometer Errors

The error in an accelerometer output is given by

$$\epsilon = E_1 + E_2\, n + E_3\, a_i + E_4\, a_p + E_5\, a_o$$

where the errors are

(1) Bias

(2) Random error

C-2

(3) Scale factor error

(4) Input axis misalignment about output axis towards pendulous axis (or pendulous axis acceleration sensitivity)

(5) Input axis misalignment about pendulous axis towards output axis (or output axis acceleration sensitivity).

The quantities E_1 through E_5 are input as for gyros. The accelerometer outputs are integrated, sampled, and quantized in the same manner as the gyro outputs.

4.3 STRAPDOWN ALGORITHM

The incremental velocity and angle outputs of each FDIC algorithm go to a strapdown algorithm that performs the attitude and inertial velocity calculations. All of the strapdown algorithms are identical.

The gravity calculations have only a minor effect on the propagation of errors in a boost inertial navigation system. Therefore, they are eliminated from the trajectory generator and each strapdown algorithm in order to save computer time and storage. The velocity errors at injection have a much stronger effect on navigational performance than do the position errors. Since position is no longer needed for the gravity calculations, the position integration can also be eliminated from the trajectory generator and each strapdown algorithm. These simplifications should have very little effect on the performance of the FDIC algorithms.

The strapdown algorithm is based on the approach described in Reference 31, with an improved attitude algorithm. The attitude algorithm chosen is McKern's third-order quaternion algorithm[32] that is used in SIRU. This algorithm is accurate and efficient and includes a correction for computation errors. The incremental quaternion $\Delta\rho$ is calculated from the incremental angle vector ϕ by

$$\Delta\rho = 1 - \frac{\phi \cdot \phi}{8} + \left(\frac{1}{2} - \frac{\phi \cdot \phi}{48}\right)\phi - \frac{\phi^{*} \times \phi}{24} \qquad (4\text{-}14)$$

where ϕ^{*} is the incremental angle vector from the preceding sampling period. The quaternion representing the vehicle orientation is updated by

$$\rho \leftarrow \rho\, \Delta\rho \qquad (4\text{-}15)$$

The velocity in body axes is updated by Yachter's algorithm[33,34]

$$v_B \leftarrow v_B + \Delta v - \phi \times \left(v_B + \frac{\Delta v}{2}\right) \tag{4-16}$$

At a slower frequency, v_B is transformed into inertial axes and the result is added to the inertial velocity v_I. At the same time, v_B is zeroed in Eq. 4-16.

$$v_I \leftarrow v_I + B\, v_B \tag{4-17}$$

where B is defined in terms of ρ

$$\begin{bmatrix} \rho_1^2 - \rho_2^2 - \rho_3^2 + \rho_4^2 & 2(\rho_1 \rho_2 - \rho_3 \rho_4) & 2(\rho_3 \rho_1 + \rho_2 \rho_4) \\ 2(\rho_1 \rho_2 + \rho_3 \rho_4) & -\rho_1^2 + \rho_2^2 - \rho_3^2 + \rho_4^2 & 2(\rho_2 \rho_3 - \rho_1 \rho_4) \\ 2(\rho_3 \rho_1 - \rho_2 \rho_4) & 2(\rho_2 \rho_3 + \rho_1 \rho_4) & -\rho_1^2 - \rho_2^2 + \rho_3^2 + \rho_4^2 \end{bmatrix} \tag{4-18}$$

4.4 TRAJECTORY

The trajectory consists of two parts, a nominal trajectory and superimposed vibrational motions. The trajectory remains the same throughout a run. The statistics of the vibrational motions remain the same throughout a case. The actual vibrational motions are Monte Carloed and vary from trial to trial.

4.4.1 Nominal Trajectory

The nominal trajectory is specified to FAILSIM by means of a subroutine, TRAJ. The acceleration and angular velocity are calculated in TRAJ by piecewise analytical functions of time which are written in FORTRAN. Discontinuities are permitted between the pieces, provided they occur at integral multiples of the minor cycle period Δt.

4.4.2 Vibrational Motions

The translational and rotational vibrational accelerations are generated by means of gaussian pseudorandom numbers, which are filtered by first-order difference equations so as to obtain first-order Gauss-Markov random processes. If unmodified, such accelerations would cause the translational and rotational velocities to random walk away from the nominal. To prevent this, the accelerations are passed through high-pass filters made by feeding back the velocity and position changes caused by the vibration through gains as shown in Figure 4-1. In the figure, n is white noise, m is the Gauss-Markov process, and a, v, and p are the translational and rotational acceleration, velocity, and position, respectively. In the cases of the three rotational vibrations and two of the translational vibrations, the feedback can be considered to represent the vehicle attitude control and guidance systems. For the vehicle longitudinal axis, no physical significance can be ascribed to the translational vibration feedback, and the gains may be set to zero if the random walk can be tolerated in this axis.

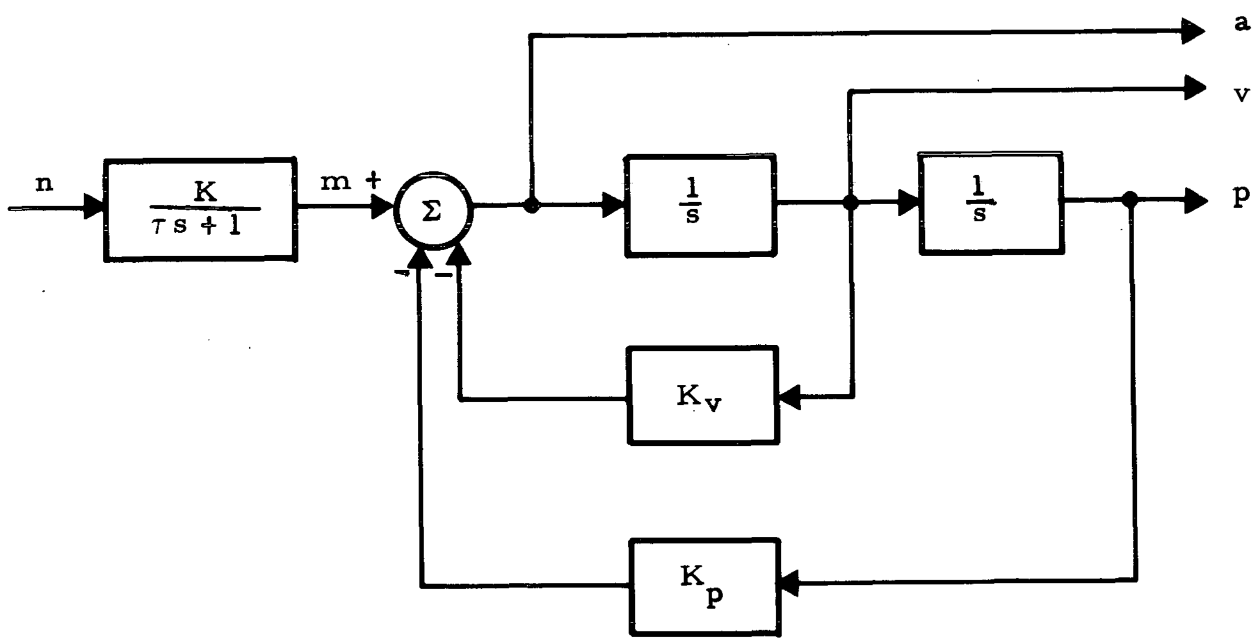


Figure 4-1. Vibrational Motion Generator

The first-order difference equation which represents the low-pass filter that generates the Gauss-Markov process is

$$m_{i+1} = K_a \, m_i + K_b \, n_{i+1} \tag{4-19}$$

In order to make m correlated with time constant τ, we require that

$$\frac{\langle m_{i+1}\, m_i \rangle}{\langle m_i^2 \rangle} = e^{-\frac{T}{\tau}} \tag{4-20}$$

where T is the sampling period. Therefore,

$$K_a = e^{-\frac{T}{\tau}} \tag{4-21}$$

The constant K_b will be adjusted to give the desired variance of the vibrational acceleration at the output of the high-pass filter. First let us find the Z transform of Eq. 4-19

$$z\, m = K_a\, m + K_b\, z\, n \tag{4-22}$$

Then

$$G_1(z) = \frac{n}{m} = \frac{K_b z}{z - K_a} \tag{4-23}$$

Now let us find a sampled data approximation to the high-pass filter from Figure 4-1,

$$G_2\,(s) = \frac{s^2}{s^2 + K_v\, s + K_p} \tag{4-24}$$

Let

$$s \cong \frac{2}{T}\,\frac{z-1}{z+1} \tag{4-25}$$

then

$$G_2\,(z) = \frac{B_o\, z^2 + B_1\, z + B_2}{z^2 + A_1\, z + A_2} \tag{4-26}$$

where

$$A_1 = \frac{2\left(K_p T^2 - 4\right)}{4 + 2 K_v T + K_p T^2} \qquad (4\text{-}27)$$

$$A_2 = \frac{4 - 2 K_v T + K_p T^2}{4 + 2 K_v T + K_p T^2} \qquad (4\text{-}28)$$

$$B_o = B_2 = \frac{4}{4 + 2 K_v T + K_p T^2} \qquad (4\text{-}29)$$

$$B_1 = -2 B_o \qquad (4\text{-}30)$$

The combined transfer function is

$$G(z) = G_1(z) G_2(z) = \frac{K_b z}{z - K_a} \cdot \frac{B_o z^2 + B_1 z + B_2}{z^2 + A_1 z + A_2} \qquad (4\text{-}31)$$

The power spectral density of the random number sequence is

$$\Phi_{nn}(z) = \frac{1}{2\pi T} \qquad (4\text{-}32)$$

and the power spectral density of the filter output is

$$\Phi_{aa}(z) = G(z) G(z^{-1}) \Phi_{nn}(z) \qquad (4\text{-}33)$$

The variance of the filter output is

$$\sigma_a^2 = \frac{T}{j} \oint \Phi_{aa}(z) z^{-1} dz \quad \text{(unit circle)} \qquad (4\text{-}34)$$

From Eqs. 4-31, 4-32, 4-33, and 4-28

$$\sigma_a^2 = \frac{K_b^2}{2\pi j} \oint \frac{\left(B_o z^2 + B_1 z + B_2\right)\left(B_o z^{-2} + B_1 z^{-1} + B_2\right) z^{-1}}{\left(z - K_a\right)\left(z^2 + A_1 z + A_2\right)\left(z^{-2} + A_1 z^{-1} + A_2\right)\left(z^{-1} - K_a\right)} dz \tag{4-35}$$

From the residue theorem

$$\sigma_a^2 = K_b^2 \ \Sigma \ \mathrm{Res}\left(P_K\right) \tag{4-36}$$

where P_K are those poles of the integral which lie within the unit circle; thus K_b is

$$K_b = \frac{\sigma_a}{\sqrt{\Sigma \mathrm{Res}\left(P_K\right)}} \tag{4-37}$$

The residues are evaluated, and K_b is calculated in subroutine FKB.

For the angular vibrations, the velocity as well as the acceleration is required. The transfer function from m to v is

$$G_3(s) = \frac{s}{s^2 + K_v s + K_p} \tag{4-38}$$

If we apply Eq. 4-25 again,

$$G_3(z) = \frac{C_o z^2 + C_1 z + C_2}{z^2 + A_1 z + A_2} \tag{4-39}$$

where

$$C_o = \frac{2T}{4 + 2K_p T + K_p T^2} \tag{4-40}$$

$$C_1 = 0 \tag{4-41}$$

$$C_2 = -C_o \tag{4-42}$$

Figure 4-2 shows a block diagram of a sampled data system which can give both a and v. The equations which must be solved by the computer are

$$m \leftarrow K_a m + K_b n \tag{4-43}$$

$$a_1 \leftarrow a_2 \tag{4-44}$$

$$a_2 \leftarrow a_3 \tag{4-45}$$

$$a_3 \leftarrow m - A_1 a_2 - A_2 a_1 \tag{4-46}$$

$$a \leftarrow B_o a_3 + B_1 a_2 + B_2 a_1 \tag{4-47}$$

$$v \leftarrow C_o a_3 + C_2 a_1 \tag{4-48}$$

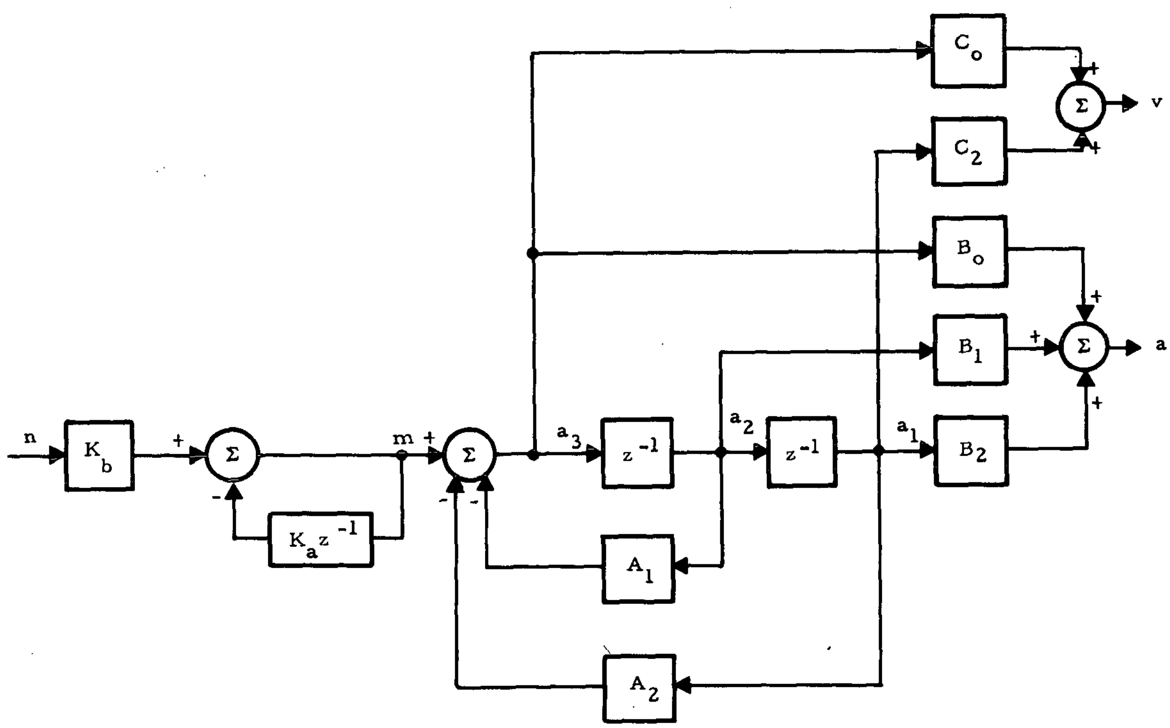


Figure 4-2. Sampled Data Vibrational Motion Generator

in that order. Initially a_1, a_2, and a_3 are set equal to zero; m is initially

$$m = \frac{K_b n}{\sqrt{1-K_a^2}} \quad (4\text{-}49)$$

The translational acceleration experienced by the strapdown package is a combination of the translational acceleration of the center of mass of the vehicle and the effect of the rotational acceleration and velocity on the lever arm between the center of mass and the package. This effect may be represented by

$$a_{TOT_1} = a_{TRANS_1} - R\left(v_{ROT_2}^2 + v_{ROT_3}^2\right) \quad (4\text{-}50)$$

$$a_{TOT_2} = a_{TRANS_2} + R\, a_{ROT_3} \quad (4\text{-}51)$$

$$a_{TOT_3} = a_{TRANS_3} - R\, a_{ROT_2} \quad (4\text{-}52)$$

where the lever arm, of length R, is assumed to lie along the 1 axis.

4.5 USE OF INPUTS FROM REAL INSTRUMENTS

The simulation program is capable of accepting the outputs of real instruments on a strapdown package mounted on a test table and bypassing the computation of simulated instrument outputs. Since the MSFC package has only three accelerometers, an option is provided to convert their three outputs into six outputs, which the FDIC then converts back into three outputs. Since the simulation program has no calibration, alignment, or compensation calculations, the outputs of real instruments must be suitably compensated before being input to the program. Failures are to be induced in the instrument hardware.

A strapdown package on a test table is unaccelerated, so that the ideal inertial velocity can be calculated without knowledge of the package orientation. Since the FAILSIM strapdown algorithms have no gravity calculation, the ideal inertial velocity cannot be obtained merely by transforming the test table velocity into inertial axes. Instead, the total sensed acceleration must be integrated.

The initial inertial velocity may be taken as zero, and the initial acceleration of gravity vector may be taken as lying in the X-Z plane of the inertial coordinate system without loss of generality. Thus the acceleration vector in inertial axes is

$$a = g \begin{Bmatrix} \cos L \cos t \\ \cos L \sin t \\ \sin L \end{Bmatrix} \tag{4-53}$$

and the velocity vector is the integral of Eq. 4-53

$$v = g \begin{Bmatrix} \frac{1}{\Omega} \cos L \sin \Omega t \\ \frac{1}{\Omega} \cos L (1 - \cos \Omega t) \\ t \sin L \end{Bmatrix} \tag{4-54}$$

where

L = astronomic latitude of test table

g = magnitude of gravity vector at test table

Ω = earth rate.

The initial package orientation must be known in order to initialize the quaternion in each strapdown algorithm. The quaternion defining up-east-north (UEN) axes with respect to inertial axes is

$$\rho_L = \cos \frac{L}{2} - j \sin \frac{L}{2} \tag{4-55}$$

The initial orientation of the package axes with respect to UEN axes is represented by rotations about three axes taken in any order. The total quaternion describing the initial orientation of the body axes with respect to inertial axes is

$$\rho = \rho_L \, \rho(\theta\rho_1, i\rho_1) \, \rho(\theta\rho_2, i\rho_2) \, \rho(\theta\rho_3, i\rho_3) \tag{4-56}$$

where $\rho(x, m)$ is a quaternion representing a rotation through the angle x about the m^{th} axis. The angles $\theta\rho_1$, $\theta\rho_2$, and $\theta\rho_3$ and the axis indicators $i\rho_1$, $i\rho_2$, and $i\rho_3$ are all read in as data. Note that the rotations from UEN to package axes are about package axes $i\rho_1$, $i\rho_2$, and $i\rho_3$, through angles $\theta\rho_1$, $\theta\rho_2$, and $\theta\rho_3$, respectively, in that order. Since $i\rho_1$, $i\rho_2$, and $i\rho_3$ can each take on any of the values 1, 2, or 3, any order of rotation is possible.

Since FAILSIM has no way of knowing the test table rotations, the ideal quaternion computation cannot be performed. Therefore, the test table should be returned to its initial orientation at the end of the run so that the initial quaternion can also serve as the ideal quaternion for calculation of the orientation errors at the end of the run.

5. FAILURE MODES

A representative set of gyro and accelerometer failure modes is chosen. Although all possible failure modes are not included, and those included may be represented in a somewhat simplified manner, they are sufficiently varied in nature to give the competing FDIC algorithms a good testing. Both hard and soft errors are included.

There are two types of failure modes to be simulated. The additive failure modes add to the output produced by the instrument in unfailed operation. For example, a shift in the bias drift would not alter the response of a gyro to inputs. The substitutional failure modes substitute an incorrect output for the correct one. The incorrect output is independent of the output that would be produced by the instrument if it were unfailed. For example, a failed instrument could produce zero output.

Each failure will be characterized by the following parameters:

- Time of failure
- Time failure disappears (for representing glitches)
- Which instrument fails
- Failure mode.

In addition, certain failures will be characterized by algebraic values or standard deviations, as described below.

5.1 GYRO FAILURES

5.1.1 Additive Failure Modes

Mode 1: Bias Drift Rate Shift

The gyro drift rate shift is a step function in angular velocity. Either the algebraic value of the shift may be input directly or it may be Monte Carloed. If it is Monte Carloed, the amount of the shift is obtained as a gaussian pseudorandom number with a specified standard deviation.

The gaussian model for the underlying process that causes shifts in bias drift is not based on any actual failure data, but seems to be a reasonable choice. However, for there to be a failure, the shift must be greater than some threshold value. This value must be specified before one can accumulate empirical failure rate or MTBF data on a real gyro. Obviously, the lower this value is, the higher the failure rate will seem to be. This threshold value is input to the program along with the standard deviation. If the magnitude of the gaussian pseudorandom number is less than the threshold value, it is rejected and a new random number is chosen.

The threshold value should be chosen high enough to exclude "failures" that really look like normal operation. If not excluded, such failures would artificially raise the missed alarm rates for all algorithms. Although such a circumstance would not be unfair to any particular algorithm, it would tend to interfere with the interpretation of the statistics gathered during the simulation process.

The threshold value should be chosen low enough so as not to exclude soft failures, since they are the hardest type of failure to detect and isolate, and determining the performance of the competing algorithms in their presence is essential.

It can be shown that, if the standard deviation of the original gaussian distribution is σ_1 and the threshold level is a, then the standard deviation of the modified distribution after the rejection process is given by σ_2 where

$$\sigma_2^2 = \sigma_1^2 + \frac{a\,\sigma_1}{\sqrt{2\pi}} \frac{\exp\left(-a^2/2\sigma_1^2\right)}{\int_a^\infty \frac{1}{\sqrt{2\pi}\,\sigma_1} \exp\left(-x^2/2\sigma_1^2\right) dx} \tag{5-1}$$

Mode 2: Drift Rate Ramp

The drift rate ramp is a ramp function in angular velocity. Either the algebraic value of the shift may be input directly or it may be Monte Carloed. If it is Monte Carloed, a standard deviation and threshold value are input.

Mode 3: Random Drift

The random drift is modeled as white noise in angular velocity. The random drift failures are modeled the same way with a larger standard deviation. Either the standard deviation may be input directly or it may be Monte Carloed. If it is Monte Carloed, a standard deviation and threshold value are input.

Mode 4: Scale Factor Shift

The scale factor shift is a step function. Its parameters are similar to those of Modes 1 and 2.

Mode 5: Mass Shift

A mass shift is modeled as a change in acceleration sensitivity about the input and spin axes. Either the algebraic values of the two components of the acceleration sensitivity may be input directly or they may be Monte Carloed as before. If they are Monte Carloed, the same standard deviation will be used for each axis, giving circular symmetry to the error probability function. A threshold value is input as before, except that it is applied to the magnitude of the 2-vector formed by the two components so as to preserve the circular symmetry.

5.1.2 Substitutional Failure Modes

Mode 6: Zero Output

The gyro output will be zero.

Mode 7: Maximum Output

Either the algebraic value of the maximum output may be input directly or it may be Monte Carloed. If it is Monte Carloed, the magnitude of the maximum value is assigned a plus or minus sign by means of a uniform pseudorandom number.

5.2 ACCELEROMETER FAILURES

5.2.1 Additive Failure Modes

Mode 1: Bias Shift

The accelerometer bias shift is a step function in acceleration. Its parameters are similar to those of gyro Mode 1.

Mode 2: Ramp Error

The accelerometer ramp error is a ramp function in acceleration. Its parameters are simular to those of gyro Mode 2.

Mode 3: Random Error

The random error is modeled as white noise in acceleration. Its parameters are similar to those of gyro Mode 3.

Mode 4: Scale Factor Shift

The scale factor shift is a step function. Its parameters are similar to those of gyro Mode 4.

5.2.2 Substitutional Failure Modes

Mode 6: Zero Output

The accelerometer output will be zero.

Mode 7: Maximum Output

The maximum output mode is similar to gyro Mode 7.

6. FAILSIM

The simulation computer program, FAILSIM, is written in FORTRAN IV for the CDC 6400/6500 computer. It was prepared by means of the TRW Timeshare System (TRW/TSS).

6.1 PROGRAM STRUCTURE

Figure 6-1 shows the FAILSIM flow diagram. At the beginning of a run, the data are preset. Then the data for the first case are read in. When no data are found, the run ends. Eight separate subroutines are used to calculate those quantities in each FDIC algorithm that remains constant throughout a case. The remainder of the program is then initialized for the next case.

The outer loop is on Monte Carlo trials. Nine separate subroutines are used to initialize the FDIC algorithms and strapdown algorithms. The remainder of the program is then initialized for the next trial.

The next loop is on major cycles. It counts the number of major cycles in the trial. The next loop is on trajectory steps. The trajectory steps are the periods over which no trajectory discontinuities are permissible. The innermost loop is on minor cycles. (For example, on a typical case the major cycle time is 1 sec, and the minor cycle time is 1/8 sec. The trajectory was obtained from a tape having data points spaced 1/2 sec, so that discontinuities can occur only at the 1/2-sec points. The minor cycle loop counts four minor cycles per trajectory step, the trajectory step loop counts two steps per major cycle, and the major cycle loop counts 400 major cycles per 400-sec flight.)

The failure programmer generates the instrument errors caused by failures. The trajectory generator generates the nominal acceleration and angular velocity and combines them with the vibrational acceleration and angular velocity. The unfailed instrument errors are generated. Fourth-order Runge-Kutta-Gill integration is used to obtain the ideal attitude and inertial velocity and the unquantized instrument outputs. The sampling, quantization, and substitutional failures are then processed. Nine separate subroutines are used to perform the minor cycle calculations

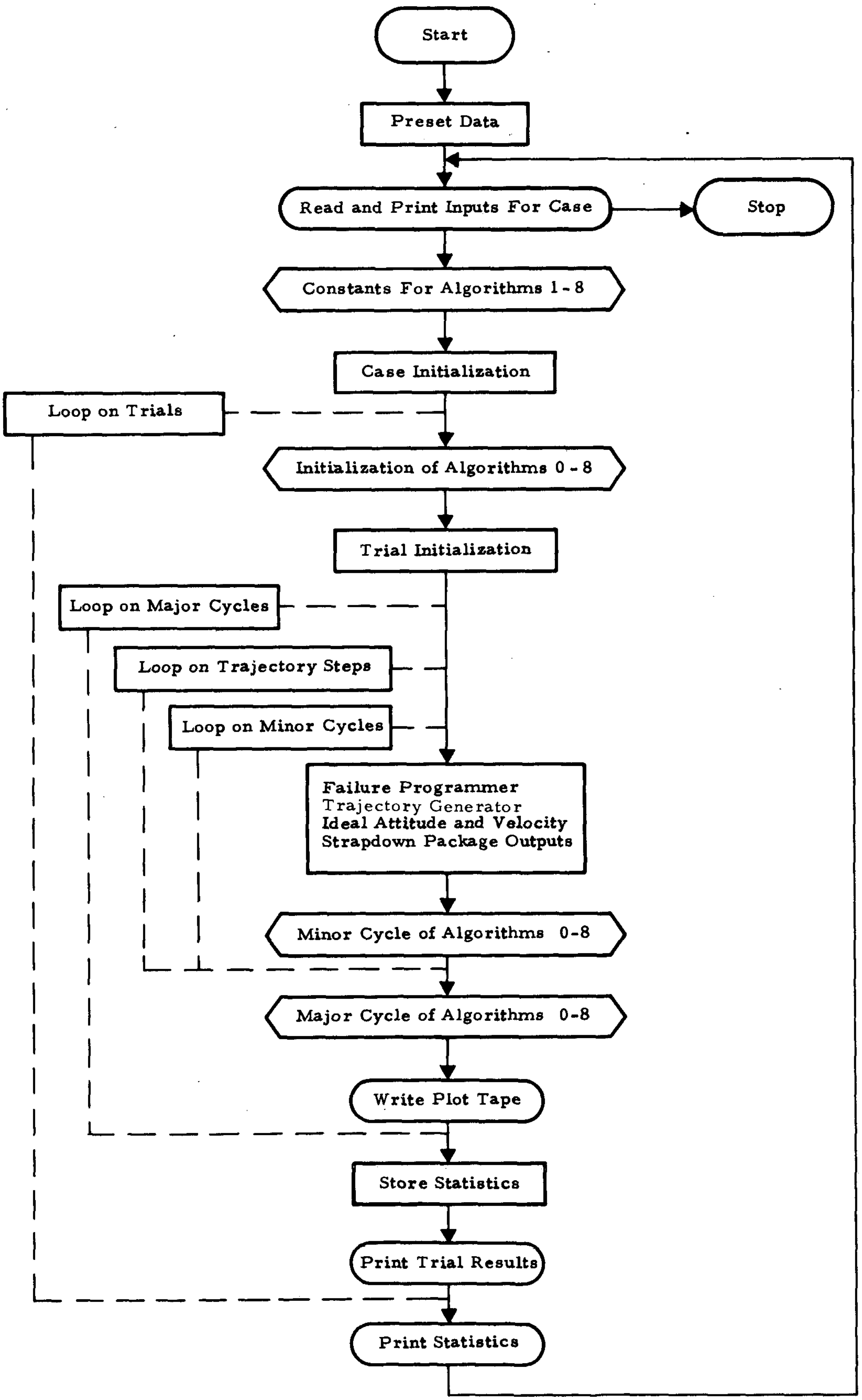


Figure 6-1. FAILSIM Flow Diagram

which include the different FDIC algorithms and identical strapdown minor cycle algorithms.

Nine separate subroutines are used to perform the strapdown major cycle algorithms. The attitude and inertial velocity errors are written on the plot tape.

At the end of a trial, its statistics are stored. The trial results are printed out.

At the end of a case, the statistics are processed and printed out. The data for the next case are then read in.

The program has been written with a modular structure so that any algorithm can be readily removed for use elsewhere. Each algorithm comprises four subroutines:

CONSTi:	evaluate constants
INITi:	initialize variables
MAJi:	major cycle calculations
MINi:	minor cycle calculations

(There are two exceptions: the nominal algorithm has no constants so that there is no CONST0, and the Minimax Algorithm has a fifth subroutine, SPHERE, that performs the Bounding Sphere Algorithm when it has been selected.)

In each of the minor cycle subroutines MINi, the gyro and accelerometer FDICs are coded separately to avoid the loss of time that would have occurred in the calling sequence if a single subroutine were used for both. The subroutine SPHERE is an exception to this policy.

In addition to the algorithm subroutines described above, FAILSIM is divided up into subprograms which will now be described.

FAIL. FAIL is the main program. It embodies the structure described above and all of the computations not relegated to the other subprograms.

DERIV1. DERIV1 is a subroutine that calculates the derivatives of the ideal attitude parameters (quaternions or Euler parameters) and the ideal inertial velocity.

DERIV2. DERIV2 is a subroutine that calculates the derivatives of the instrument outputs. The package inputs are resolved into instrument axes. The ideal instrument inputs are found and modified by both unfailed errors and additive failures to obtain the derivatives of the actual outputs.

EVENT. EVENT is a subroutine that notes changes in the instrument states, prints out messages when changes occur, and accumulates statistics on the changes.

FKB. FKB is a function that solves for K_b (Paragraph 4.4.2)

GAUSS. GAUSS is a function that converts uniform pseudorandom numbers into gaussian pseudorandom numbers.

ORIENT. ORIENT is a subroutine that computes the matrices A, C, and F of Paragraph 3.1.2, the matrix R, and the matrices AG1 to AG6 and AA1 to AA6 of Subsection 4.1 and their products with R, and the masks AMASK and OMASK used in the various algorithms.

PACKAGE. PACKAGE is a subroutine that reads in the external strapdown package instrument output tape data and unbuffers it. If the external package has three accelerometers, six pseudo accelerometer outputs are generated for compatibility with FAILSIM.

PLOT. PLOT is a subroutine that finds the orientation errors in arc sec and the inertial velocity errors in m/sec and writes them out on the plot tape.

QUAT. QUAT is a subroutine that calculates a quaternion for a given angle and coordinate axis and premultiplies it by an input quaternion to obtain an output quaternion.

READIN. READIN is a subroutine that reads in the program inputs, converts units for inputs not directly involving the algorithms themselves, and prints out all of the inputs. (Algorithm inputs have their units converted in the CONSTi subroutines.)

SYMINV. SYMINV is a subroutine that inverts a symmetric matrix of order 3 (only the upper triangular elements of the input matrix are required; all elements of the output matrix are supplied).

TMAT. TMAT is a subroutine that calculates a rotation matrix for a given angle and coordinate axis and post-multiplies it by an input matrix to obtain an output matrix.

TRAJ. TRAJ is a subroutine that calculates the nominal angular velocity and nominal acceleration for a specific trajectory.

6.2 INPUTS

With the exception of taped data from an external strapdown package (Paragraph 6.2.6) and the nominal trajectory subroutine (Paragraph 6.2.8), all of the program inputs are entered by means of the NAMELIST feature. At the beginning of a run, all of the data is preset to 0 except IRDM, which is preset to 1. None of the input data is changed during a case, with the exception of IRDM, which always indicates the next pseudorandom number.

Only those inputs which differ from 0 (or 1 in the case of IRDM) need be read in for the first case of a run. For subsequent cases in the run, only those inputs that differ from the inputs of the preceding case need be read in, with the exception of IRDM. If IRDM is desired to be the same for all cases, it must be included in the inputs to each case. Otherwise, each case will begin with a new pseudorandom number. In case it is desired to begin a run with the pseudorandom number from the end of a preceding run, IRDM is printed out at the end of each trial. A typical set of data is shown in Appendix A.

6.2.1 Program Control (Subsection 6.1)

Variable	Units	Description
DT	sec	Minor cycle period
IRDM		Initial random number (odd integer)
LMONTE		0 = deterministic failures 1 = Monte Carlo failures
LPLOT		0 = no plot tape made 1 = plot tape made
N1MAX		Number of Monte Carlo trials per case
N2MAX		Number of major cycles per trial

Variable	Units	Description
N3MAX		Number of trajectory steps per major cycle
N4MAX		Number of minor cycles per step

6.2.2 Vehicle Vibrational Motions (Paragraph 4.4.2)

Variable	Units	Description
AVBS(1:3)	rad/sec^2	Rotational acceleration, vibration, body, standard deviation
AVBS(4:6)	m/sec^2	Translational acceleration, vibration, body, standard deviation
TAUV(1:6)	sec	Vibration time constants
KV(1:6)	sec^{-1}	Vibration velocity feedback gain
KP(1:6)	sec^{-2}	Vibration position feedback gain
RADIUS	m	Distance from vehicle c.m. to strapdown package

6.2.3 Instrument Orientation (Subsection 4.1)

Variable	Units	Description
IR1		Axis of first rotation for R matrix
THTR1I	deg	Angle of first rotation for R matrix
IR2		Axis of second rotation for R matrix
THTR2I	deg	Angle of second rotation for R matrix
IR3		Axis of third rotation for R matrix
THTR3I	deg	Angle of third rotation for R matrix

Variable	Units	Description
THTAI(1:6)	deg	Output axis location angles for accelerometers
THTGI(1:6)	deg	Output axis location angles for gyros

6.2.4 Unfailed Errors (Subsection 4.2)

Variable	Units	Description
SIGUFEA(1)	μg	Bias
SIGUFEA(2)	$(\mu g)^2/(rad/sec)$	Random
SIGUFEA(3)	ppm	Scale factor
SIGUFEA(4)	arc sec	Misalignment
SIGUFEA(5)	arc sec	Misalignment
SIGUFEG(1)	deg/hr	Bias
SIGUFEG(2)	$(deg/hr)^2/$ (rad/sec)	Random
SIGUFEG(3)	ppm	Scale factor
SIGUFEG(4)	arc sec	Misalignment
SIGUFEG(5)	arc sec	Misalignment
SIGUFEG(6)	deg/hr/g	Acceleration sensitivity
SIGUFEG(7)	deg/hr/g	Acceleration sensitivity
SIGUFEG(8)	deg/hr/g	Acceleration sensitivity
SIGUFEG(9)	$deg/hr/g^2$	Anisoelastic drift
QACCI	cm/sec/pulse	Accelerometer quantization
QGYROI	arc sec/pulse	Gyro quantization

6.2.5 Failures (Section 5)

Up to 14 failures or healings are possible in a case.

Variable		Description
NFAIL(1:14)		Number of minor cycles before failure or healing
NINST(1:14)		Number of failed or healed instrument

NINST = 1:6 failure of gyro No. NINST

NINST = 7:12 failure of accelerometer No. NINST-6

Variable		Description
MODE(1:14)		Failure mode
For gyros MODE =	0	Healing
	1	Bias drift rate shift
	2	Drift rate ramp
	3	Random drift
	4	Scale factor shift
	5	Mass shift
	6	Zero output
	7	Maximum output
For acc. MODE =	0	Healing
	1	Bias shift
	2	Ramp error
	3	Random error
	4	Scale factor shift
	5	—
	6	Zero output
	7	Maximum output
FAIL1I(1:14)		First failure parameter
FAIL2I(1:14)		Second failure parameter

The units and definitions of FAIL1I and FAIL2I depend upon the corresponding values of NINST and MODE, and upon the value of LMONTE. For LMONTE = 0 (deterministic failures) and NINST = 1:6 (gyros)

MODE = 0,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 1,	FAIL1I	deg/hr, value of bias drift rate shift
	FAIL2I	Not required
MODE = 2,	FAIL1I	deg/hr^2, value of drift rate ramp
	FAIL2I	Not required
MODE = 3,	FAIL1I	$(deg/hr)^2/(rad/sec)$, random failure p.s.d.
	FAIL2I	Not required
MODE = 4,	FAIL1I	ppm, value of scale factor shift
	FAIL2I	Not required
MODE = 5,	FAIL1I	deg/hr/g, mass shift along spin axis
	FAIL2I	deg/hr/g, mass shift along input axis
MODE = 6,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 7,	FAIL1I	deg/sec, value of maximum output
	FAIL2I	Not required

For LMONTE = 0 (deterministic failures) and NINST = 7:12 (acc.)

MODE = 0,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 1,	FAIL1I	μg, value of bias shift
	FAIL2I	Not required
MODE = 2,	FAIL1I	μg/hr, value of ramp error
	FAIL2I	Not required
MODE = 3,	FAIL1I	$(\mu g)^2/(rad/sec)$, random failure p.s.d.
	FAIL2I	Not required

MODE = 4,	FAIL1I	ppm, value of scale factor shift
	FAIL2I	Not required
MODE = 6,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 7,	FAIL1I	g, value of maximum output
	FAIL2I	Not required

For LMONTE = 1 (Monte Carlo failure) and NINST = 1:6 (gyros)

MODE = 0,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 1,	FAIL1I	deg/hr, s.d. of bias drift rate shift
	FAIL2I	deg/hr, threshold
MODE = 2,	FAIL1I	deg/hr^2, s.d. of drift rate ramp
	FAIL2I	deg/hr^2, threshold
MODE = 3,	FAIL1I	$(deg/hr)^2/(rad/sec)$ s.d. of random failure p.s.d.
	FAIL2I	$(deg/hr)^2/(rad/sec)$, threshold
MODE = 4,	FAIL1I	ppm, s.d. of scale factor shift
	FAIL2I	ppm, threshold
MODE = 5,	FAIL1I	deg/hr/g, s.d. of mass shift
	FAIL2I	deg/hr/g, threshold
MODE = 6,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 7,	FAIL1I	deg/sec, magnitude of maximum output
	FAIL2I	Not required

For LMONTE - 1 (Monte Carlo failures) and NINST = 7:12 (acc.)

MODE = 0,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 1,	FAIL1I	μg, s.d. of bias shift
	FAIL2I	μg, threshold

MODE = 2,	FAIL1I	$\mu g/hr$, s.d. of ramp error
	FAIL2I	$\mu g/hr$, threshold
MODE = 3,	FAIL1I	$(\mu g)^2/(rad/sec)$, s.d. of random failure p.s.d.
	FAIL2I	$(\mu g)^2/(rad/sec)$, threshold
MODE = 4,	FAIL1I	ppm, s.d. of scale factor shift
	FAIL2I	ppm, threshold
MODE = 6,	FAIL1I	Not required
	FAIL2I	Not required
MODE = 7,	FAIL1I	g, magnitude of maximum output
	FAIL2I	Not required

6.2.6 External Strapdown Package (Section 4.5)

Variable	Units	Description
LPACK		0 = Internal strapdown package 1 = External strapdown package, 6 gyros and 3 accelerometers 2 = External strapdown package, 6 gyros and 6 accelerometers

(The following inputs are not required if LPACK is zero).

IRHO1		Axis of first rotation for initial quaternion
THTRH1I	deg	Angle of first rotation for initial quaternion
IRHO2		Axis of second rotation for initial quaternion
THTRH2I	deg	Angle of second rotation for initial quaternion
IRHO3		Axis of third rotation for initial quaternion
THTRH3I	deg	Angle of third rotation for initial quaternion

Variable	Units	Description
LATI	deg	Astronomic latitude of test table
G	m/sec^2	Acceleration of gravity at test table
OMEGA	rad/sec	Earth rate

The compensated instrument outputs in units of rad and m/sec must appear on a binary format magnetic tape to be read by logical unit 15. Each record on the tape must contain 505 or fewer words. The first word is an integer less than or equal to 504 and equal to the number of words following it in the record. If LPACK equals 1, there may be up to 56 nine-word data sets in the record, each data set containing the six gyro outputs followed by the three accelerometer outputs. If LPACK equals 2, there may be up to 42 twelve-word data sets in the record, each data set containing the six gyro outputs followed by the six accelerometer outputs.

6.2.7 Algorithms (Subsection 3.2)

Nominal Algorithm

Variable	Description
LALG0	0 = do not use algorithm 1 = use algorithm

Adaptive 66 Algorithm

Variable	Units	Description
LALG1		0 = do not use algorithm 1 = use algorithm

(The following inputs are not required if LALG1 is zero)

EPS0A	μg	Unfailed acc. error s.d.
EPS0G	deg/hr	Unfailed gyro error s.d.
TAUPAQ(1)	sec	Acc. prefilter time constant
TAUPGQ(1)	sec	Gyro prefilter time constant
CRIT1		Loop termination criterion

Fifteen Threshold Algorithm

Variable	Units	Description
LALG2		0 = Do not use algorithm 1 = Use algorithm

(The following inputs are not required if LALG2 is zero)

THAI	μg	Acc. threshold
THGI	deg/hr	Gyro threshold
TAUPAQ(2)	sec	Acc. prefilter time constant
TAUPGQ(2)	sec	Gyro prefilter time constant

Squared Error Algorithm

Variable	Units	Description
LALG3		0 = Do not use algorithm 1 = Use algorithm

(The following inputs are not required if LALG3 is zero)

AK1I	μg	Acc. TSE threshold, first failure
AK2I		Acc. SE/TSE threshold, first failure
AK3I	μg	Acc. TSE threshold, second failure

Variable	Units	Description
AK4I		Acc. SE/TSE threshold, second failure
GK1I	deg/hr	Gyro TSE threshold, first failure
GK2I		Gyro SE/TSE threshold, first failure
GK3I	deg/hr	Gyro TSE threshold, second failure
GK4I		Gyro SE/TSE threshold, second failure
TAUPAQ(3)	sec	Acc. prefilter time constant
TAUPGQ(3)	sec	Gyro prefilter time constant
AKGDI	$(deg/sec)^{-1}$	Gyro dynamic error compensation gain
LHEAL		0 = no reinstatement permitted 1 = reinstatement permitted

Bayesian Decision Theory Algorithm

Variable	Units	Description
LALG4		0 = do not use algorithm 1 = use algorithm

(The following inputs are not required if LALG4 is zero).

Variable	Units	Description
THA1I	μg	Acc. threshold, first failure
THA2I	μg	Acc. threshold, second failure
THG1I	deg/hr	Gyro threshold, first failure
THG2I	deg/hr	Gyro threshold, second failure

Variable	Units	Description
TAUPAQ(4)	sec	Acc. prefilter time constant
TAUPGQ(4)	sec	Gyro prefilter time constant

Maximum Likelihood Algorithm

Variable	Units	Description
LALG5		0 = do not algorithm 1 = use algorithm
SIGQA0	μg	Unfailed acc. error s.d.
SIGQG0	deg/hr	Unfailed gyro error s.d.
TAUPAQ(5)	sec	Acc. prefilter time constant
TAUPGQ(5)	sec	Gyro Prefilter time constant
CRIT5		Loop termination criterion

Minimax Algorithm

Variable	Units	Description
LALG6		0 = do not use algorithm 1 = use algorithm

(The following inputs are not required if LALG6 is zero).

LSQA		0 = Acc. use Bounding Sphere Algorithm 1 = Acc. use Least-Squares Algorithm
LSQG		0 = Gyros use Bounding Sphere Algorithm 1 = Gyros use Least-Squares Algorithm
THAMI	μg	Max. filtered error of unfailed acc.
THASI	μg	Max. unfiltered error of unfailed acc.

Variable	Units	Description
THGMI	deg/hr	Max. filtered error of unfailed gyro
THGSI	deg/hr	Max. unfiltered error of unfailed gyro
TAUPAQ(6)	sec	Acc. prefilter time constant
TAUPGQ(6)	sec	Gyro prefilter time constant

(THASI is not required if LSQA equals 1; THGSI is not required if LSQG equals 1).

Sequential Algorithm

Variable	Units	Description
LALG7		0 = do not use algorithm 1 = use algorithm

(The following inputs are not required if LALG7 is zero).

SIGPA(1)	μg	Acc. input state initial s.d.
SIGPA(2)	μg	Acc. error state initial s.d.
SIGQA(1)	μg	Acc. input state noise s.d.
SIGQA(2)	μg	Acc. error state noise s.d.
SIGRA	μg	Acc. measurement error s.d.
TAUSA(1)	sec	Acc. input s.t.m. time constant
TAUSA(2)	sec	Acc. error s.t.m. time constant
THRAI(1)	μg	Acc. high level threshold
THRAI(2)		Acc. low level threshold

Variable	Units	Description
SIGPG(1)	deg/hr	Gyro input state initial s.d.
SIGPG(2)	deg/hr	Gyro error state initial s.d.
SIGQG(1)	deg/hr	Gyro input state noise s.d.
SIGQG(2)	deg/hr	Gyro error state noise s.d.
SIGRG	deg/hr	Gyro measurement error s.d.
TAUSG(1)	sec	Gyro input s.t.m. time constant
TAUSG(2)	sec	Gyro error s.t.m. time constant
THRGI(1)	deg/hr	Gyro high level threshold
THRGI(2)		Gyro low level threshold

6.2.8 Nominal Trajectory (Paragraph 4.4.1 and Appendix A)

The nominal trajectory is specified by a subroutine named TRAJ written in FORTRAN IV. A shuttle boost trajectory is included in the program at present. The trajectory is broken up into piecewise segments of analytical functions of time. The logic to choose the correct segment must be supplied. It may depend upon the time T, provided there is no discontinuity in the functions at the switching time. Any values of T are permissible for use in the segment selection logic. If there are discontinuities present, they may only occur at the times

$$t_N = N \cdot N4MAX \cdot DT \qquad N = 0, 1, 2, \ldots \tag{6-1}$$

The number of the discontinuity opportunity is given by N

$$N = N3MAX \cdot (N2 - 1) + N3 \tag{6-2}$$

where N2 is the major cycle counter and N3 is the trajectory step counter.

Eq. 6-2 should be the first executable statement of the subroutine. The logic to choose the correct segment must depend upon N if there is a discontinuity. The value of T corresponding to N is given by Eq. 6-1. Logic based on T may be mixed with logic based on N. The body angular velocity is specified by VPAB(1:3), and the body acceleration is specified by APLB(1:3). It is convenient to take the body axes as:

(1) Roll (forward)
(2) Pitch (right)
(3) Yaw (down).

6.3 OUTPUTS

A typical run is shown in Appendix A. The outputs will be described here.

6.3.1 Inputs

At the beginning of the output we find the program name, the date, and the time. Then all of the NAMELIST inputs are printed out in groups. If LALGi is 0, the remaining inputs for algorithm i are not printed, even if they were present in the input data.

6.3.2 Events

Following the inputs, we find the output data for each trial. First we see a listing of all of the events taking place during the trial. Whenever an instrument is failed, the time, instrument type, instrument number, and mode of the failure are given. Whenever an instrument is healed, the time, instrument type, and instrument number are given. Whenever an algorithm changes state, the time, algorithm number, instrument type, instrument number, and type of state change are given. There are nine possible types of state change of interest:

ALL CLEAR	The algorithm changes the state of a healed instrument from failed to unfailed
CORRECTED FALSE ALARM	The algorithm changes the state of an unfailed instrument from failed to unfailed
FALSE ALARM	The algorithm changes the state of an unfailed instrument from unfailed to failed

LOST ALARM	The algorithm changes the state of a failed instrument from failed to unfailed
TRUE ALARM	The algorithm changes the state of a failed instrument from unfailed to failed
FORTUITOUS TRUE ALARM	An instrument fails when the algorithm has already determined its state to be failed
FORTUITOUS ALL CLEAR	An instrument is healed when the algorithm has already determined its state to be unfailed
MISSED ALARM	The trial terminates and the algorithm has determined the state of a failed instrument to be unfailed
MISSED ALL CLEAR	The trial terminates and the algorithm has determined the state of a healed instrument to be failed.

Figure 6-2 shows the state transitions and the corresponding messages. The left-hand two states (circles) correspond to an unfailed instrument, the middle two to a failed instrument, and the right-hand three to a healed instrument. The upper four states correspond to an instrument state in the algorithm of failed (alarm), and the lower three states correspond to a state of unfailed (no alarm). Horizontal transitions are caused by failure or healing, curved line transitions by changes of algorithm instrument state, and vertical straight line transitions by the end of the trial. The upper right-hand state is split so as to distinguish between all clears and corrected false alarms.

Unlike the other algorithms, the Adaptive 66 and Maximum Likelihood Algorithms do not have internal logical states that give obvious meanings to the concept of alarm or no alarm. Rather than obtain no such information for these algorithms, it was decided to invent an arbitrary criterion for the presence or absence of an alarm. This criterion, for both algorithms, is that an alarm occurs when the variance of an instrument is found to be greater than three times the variance of unfailed instruments.

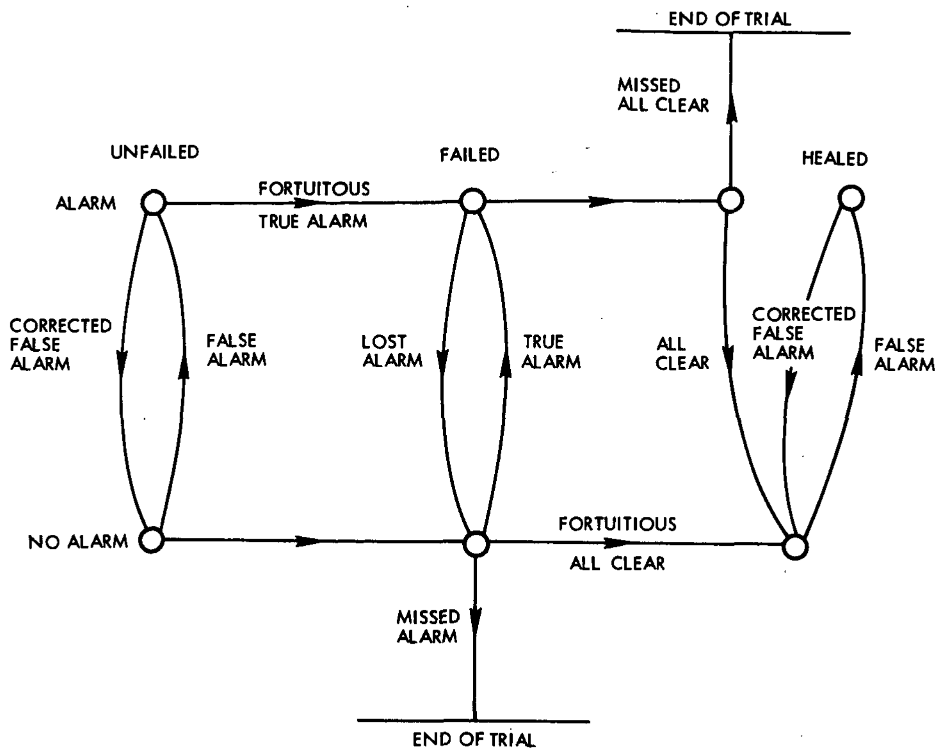


Figure 6-2. Event Logic

6.3.3 Figures of Merit

The figures of merit are printed out at the end of the trial. They appear in nine columns for algorithms 0 to 8, and four rows. The first row gives the magnitudes of the small angle error vectors between the attitudes of algorithms 0 to 8, and the ideal attitude. The second row gives the magnitudes of the small angle error vectors between the attitudes of algorithms 1 to 8, and the attitude of algorithm 0, the nominal algorithm. The third row gives the magnitudes of the error vectors between the inertial velocities of algorithms 0 to 8, and the ideal inertial velocity. The fourth row gives the magnitudes of the error vectors between the inertial velocities of algorithms 1 to 8, and the inertial velocity of algorithm 0. The first two rows are in arc sec and the last two in m/sec.

The second and fourth rows are useful in discerning differences between the algorithms when their answers are close to the nominal algorithm, that is when the errors caused by failures are being obscured by the effects of strapdown algorithm truncation error and unfailed errors. The absolute significance of these figures of merit is less clear.

The algorithm numbering scheme is:

0 Nominal
1 Adaptive 66
2 Fifteen Threshold
3 Squared Error
4 Bayesian Decision Theory
5 Maximum Likelihood
6 Minimax
7 Sequential
8 Not yet implemented (intended for Adaptive 72).

6.3.4 Algorithm Execution Times

Following the figure of merit, we find the total times (in sec) spent by the central processor in the FDIC portion of each algorithm during the trial, and the peak times required by each algorithm in any single minor cycle. Unfortunately, the latter are quantized too coarsely (msec) to be of much value.

The final random number and the total central processor time used so far in the trial are then printed out.

6.3.5 Statistics

At the end of the case, the statistics of the trials are printed out. The mean magnitudes of the figures of merit are printed out in the same format as described above. The total FDIC time for all trials and the peak FDIC times for any trial are also printed out. A summary of all the events occurring during the case is also presented.

6.3.6 Plot Tape

The plot tape is written in binary format on logical unit 13 using binary blocking. If binary blocking is not desired, or if the program is to be used with other than CDC 6X00 computers, a DATA statement and the third executable statement in the main program, FAIL, should be removed. These statements are:

```
DATA IRAY/13/
CALL FTNBIN(1,1,IRAY)
```

Each record on the plot tape contains 19 words. The first word is the time. The $(2i + 2)^{th}$ word is the magnitude of the small angle attitude error vector in arc sec for algorithm i; the $(2i + 3)^{th}$ word is the magnitude of the inertial velocity error vector in m/sec for algorithm i; $i = 0, 8$.

7. SETTING THE PARAMETERS

The seven algorithms actually implemented have among them some 26 adjustable parameters for the gyros and 25 for the accelerometers; the Sequential Algorithm accounts for a third of these. Setting these parameters correctly presents quite a problem. Ideally, the parameters should be set optimally for each algorithm. However, even reaching an acceptable definition of optimality is a difficult task.

The analytical setting of the parameters presents formidable difficulties, since all of the algorithms are nonlinear, and a different analysis would have to be performed for each. In fact, if one were capable of setting the parameters analytically, this would imply the ability to determine performance analytically, and FAILSIM would be unnecessary.

One example of the type of difficulty encountered in trying to predict behavior analytically is the "failure induced false alarm." A configuration of unfailed errors that could not produce a false alarm by themselves may, in the presence of a failure of one instrument, cause a false alarm for a different instrument. If an instrument is already failed and isolated, the false alarm precludes isolation of the new failure.

Table 7-I shows the number of parameters in each algorithm. It also shows whether or not independent threshold settings are available for the second failure.

Table 7-I. Adjustable Parameters in the Several Algorithms

Algorithms	Parameters		Threshold for Second Failure
	Gyros	Accs.	
Adaptive 66	2	2	Same as first
Fifteen Threshold	2	2	Same as first
Squared Error	6	5	Independent
Bayesian Decision Theory	3	3	Independent
Maximum Likelihood	2	2	Same as first
Minimax/Least Squares	2	2	4.2 dB above first
Adaptive 72	?	?	Independent
Sequential	9	9	Same as first

Happily, the authors of the Squared Error Algorithm have supplied four of the thresholds:

$$AK2I = GK2I = 4/9 \tag{7-1}$$

$$AK4I = GK4I = 8/21 \tag{7-2}$$

If we assume that setting AKGDI to zero has no major effect on performance, then we need determine only three parameters for the gyros and three for the accelerometers. Then only the Sequential Algorithm has more than three parameters to determine. The difficulty in selecting parameters for this algorithm has been alleviated somewhat by the analysis of Appendix B. By its use, one can assign the parameters of the Sequential Algorithm to give an equivalent steady-state time constant.

To provide a reasonably fair way of setting the parameters, it was decided to allot equal amounts of effort to each algorithm. After all, if an algorithm requires less effort than another in parameter setting, it has an advantage that ought to show up in the final results.

After consideration of various possibilities, it was decided that a reasonable approach, capable of being achieved in a limited number of computer runs, would be to set all of the thresholds for the same false-alarm rate. Thus all algorithms would give about the same performance in the absence of failures (the most probable situation for any particular mission), and their relative performance in the presence of failures would serve to establish their relative merits.

The procedure followed was to choose two values for the time constants: 100 and 10 sec. The 100-sec value was used first. A series of runs was made with no failures. The thresholds were adjusted in steps of 2 dB until a value was found for each algorithm for which no false alarm alarms occurred while a 2-dB lower threshold produced at least one false alarm. (The accelerometer threshold-setting process was not completed for lack of time, so the following comments apply to the gyros only.)

Runs were then made with small MODE = 1 failures at 200 sec (halfway through boost). False alarms occurred in the Fifteen Threshold, Minimax, and Sequential Algorithms. Raising the thresholds 2 dB eliminated the false alarms.

More runs were made with a large MODE = 1 failure at 190 sec and a small MODE = 1 failure at 200 sec. The purpose of these runs was to set the thresholds for second failures in those algorithms that had them. The large failure was chosen large enough to cause all of the algorithms to isolate the failure within one minor cycle, so that no data from the failed instrument were used. It was found necessary to set the second-failure thresholds higher than the first-failure thresholds in the Squared Error and Bayesian Decision Theory Algorithms in order to eliminate false alarms. For the same reason, it was necessary to raise the thresholds of the Adaptive 66, Fifteen Threshold, Maximum Likelihood, and Sequential Algorithms, thus degrading their performance on first failures. Thus the desirability of independent first- and second-failure thresholds seems obvious. However, it was not found necessary to raise the single threshold of the Minimax Algorithm, apparently because the built-in 4.2 dB ratio between effective first- and second-failure threshold levels is close to optimal. For the Maximum Likelihood and Sequential Algorithms, the thresholds had to be raised so high in order to eliminate false alarms that a true alarm was not obtained on the second failure.

This procedure was repeated with a 10-sec time constant. With this value of time constant, the Maximum Likelihood and Sequential Algorithms succeeded in obtaining true alarms without false alarms. The 10-sec time constant was found to be somewhat better for all algorithms, so it was chosen for the remaining runs. The thresholds selected to go with the 10-sec time constant were considered the final set for use in the competition. However, the first double-failure test run produced false alarms (before any failures) for the Squared Error, Bayesian Decision Theory, and Minimax Algorithms, so their thresholds were all raised by 2 dB, eliminating the false alarms.

8. SIMULATION RESULTS

The symbols used to plot the errors of the various algorithms are:

- ⊙ Nominal Algorithm or multiple point
- △ Adaptive 66 Algorithm
- \+ Fifteen Threshold Algorithm
- X Squared Error Algorithm
- ◇ Bayesian Decision Theory Algorithm
- ⚲ Maximum Likelihood Algorithm
- ⊡ Minimax Algorithm
- Z Sequential Algorithm

8.1 100-SEC TIME CONSTANT RUNS

The gyro FDIC algorithm parameters for the 100-sec time constant runs are given in Table 8-I. (The threshold setting process was not successful for the Maximum Likelihood and Sequential Algorithms so they do not appear in Table 8-I.)

Table 8-I. FDIC Parameters

Adaptive 66	
EPS0G = 1.585	TAUPGQ(1) = 100.
CRIT1 = 0.1	
Fifteen Threshold	
THG1 = 1.585	TAUPGQ(2) = 100.
Squared Error	
GK1I = 1.41	GK2I = 4/9
GK3I = 2.51	GK4I = 8/21
TAUPGQ(3) = 100.	AKGDI = 0.
LHEAL = 0	
Bayesian Decision Theory	
THG1I = 1.995	THG2I = 3.98
TAUPGQ(4) = 100.	
Minimax	
THGMI = 0.617	TAUPGQ(6) = 100.
LSQG = 1	

8.1.1 First Failures

A series of runs was made with a MODE=1 gyro failure of varying magnitude at 200 sec using the parameters shown above. Figure 8-1 shows the attitude error at the end of the single trial in arc sec versus the magnitude of the failure in deg/hr. The same initial random number was used for each trial.

8.1.2 Second Failures

A series of runs was made with a large MODE=1 gyro failure at 190 sec. All of the algorithms switched out the failed gyro within one minor cycle, so that no erroneous data were incorporated in the strapdown solution. A second MODE=1 gyro failure of varying magnitude occurred at 200 sec, using the parameters shown above. Figure 8-2 shows the results.

8.2 10-SEC TIME CONSTANT RUNS

The gyro FDIC algorithm parameters for the 10-sec time constant runs are given in Table 8-II.

8.2.1 First Failures

A series of runs was made with a MODE=1 gyro failure of varying magnitude at 200 sec using the parameters shown above. Figure 8-3 shows the results.

8.2.2 Second Failures

A series of runs was made with a large MODE=1 gyro failure at 190 sec and a smaller MODE=1 gyro failure of varying magnitude at 200 sec, using the parameters shown above. Figure 8-4 shows the results.

The 10-sec time constant results look better than the 100-sec time constant results for the larger failures and somewhat worse for the smaller failures. The choice of 10 sec might minimize the average error, while the choice of 100 sec might minimize the worst-case error. An intermediate value of time constant might be a good compromise. In any event, 10 sec was chosen as the time constant for the subsequent runs.

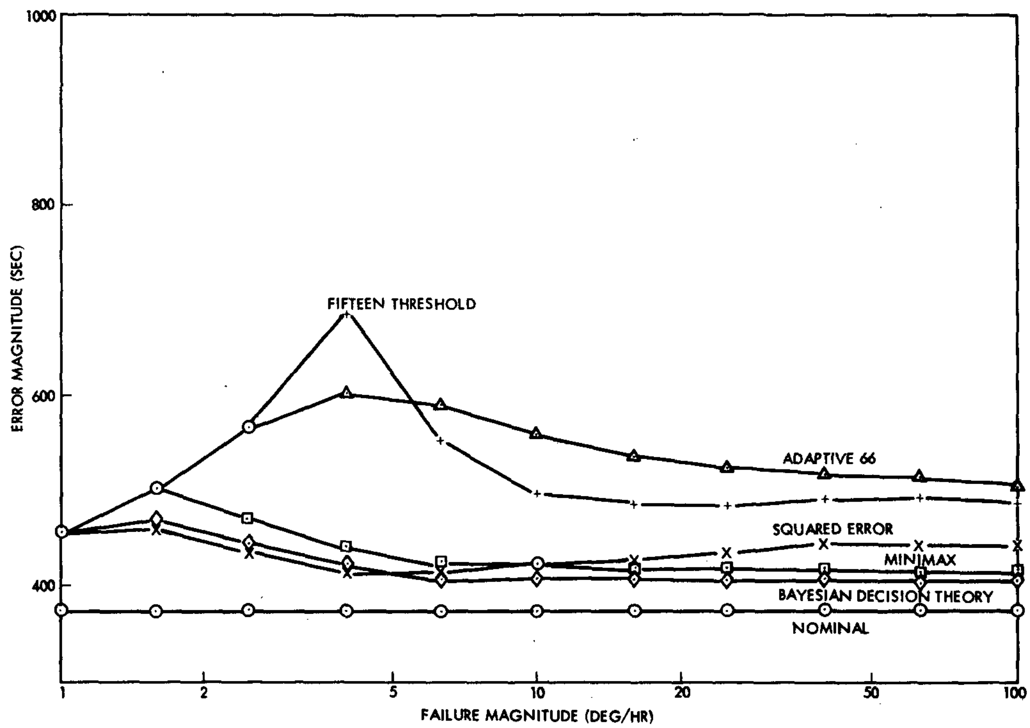


Figure 8-1. Orientation Error Magnitude Versus Failure Magnitude - First Gyro Failure, 100-Sec Time Constant

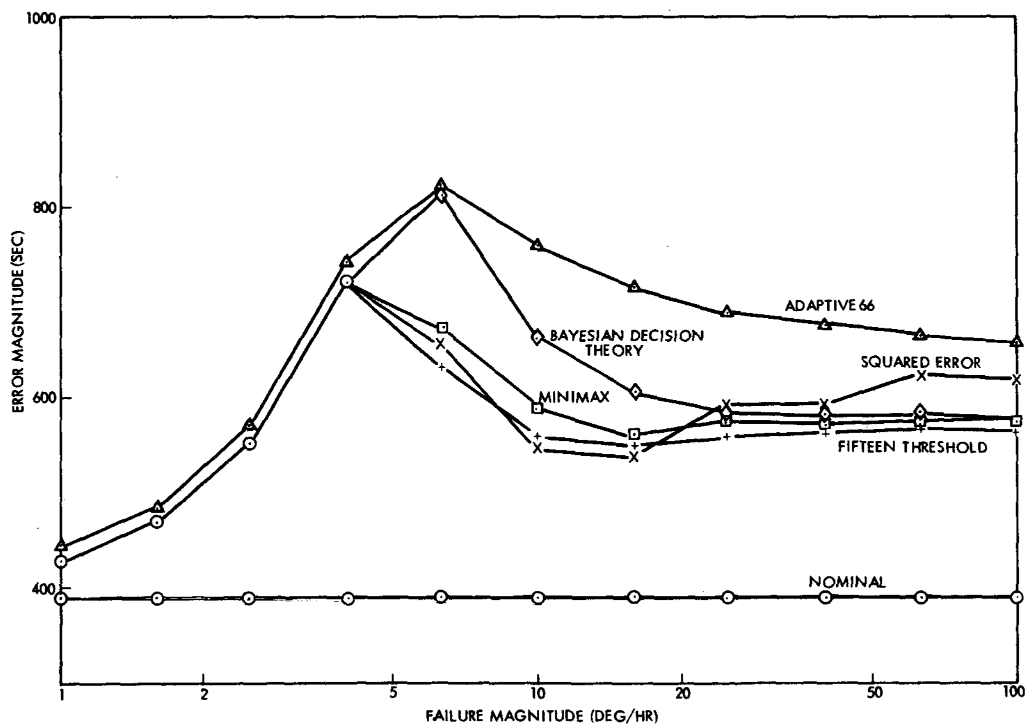


Figure 8-2. Orientation Error Magnitude Versus Failure Magnitude - Second Gyro Failure, 100-Sec Time Constant

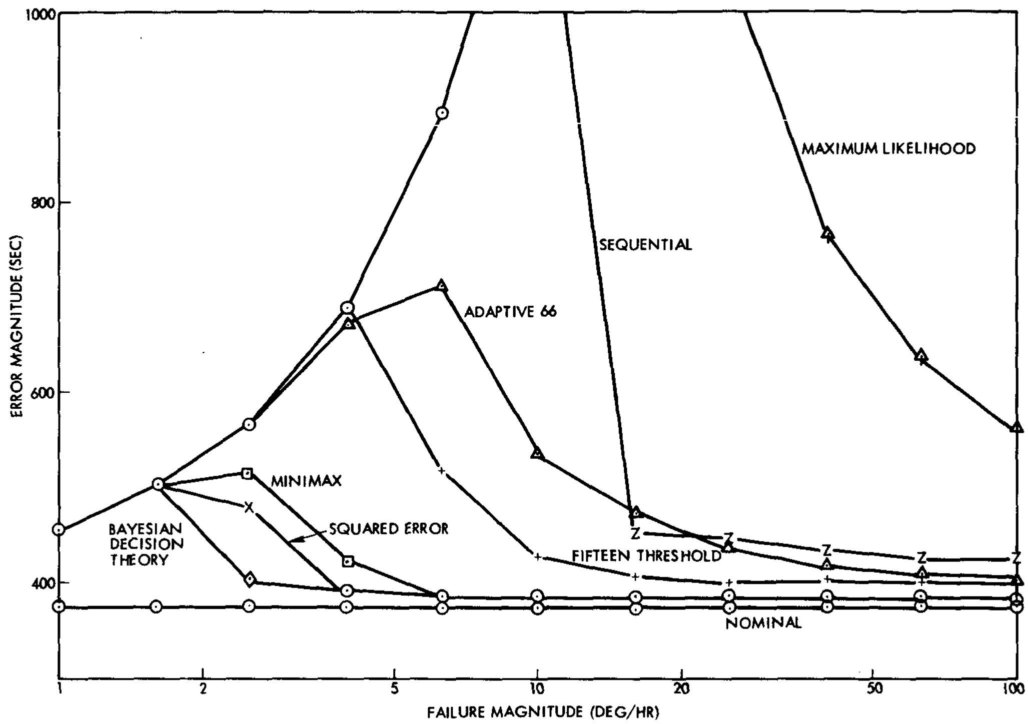


Figure 8-3. Orientation Error Magnitude Versus Failure Magnitude - First Gyro Failure, 10-Sec Time Constant

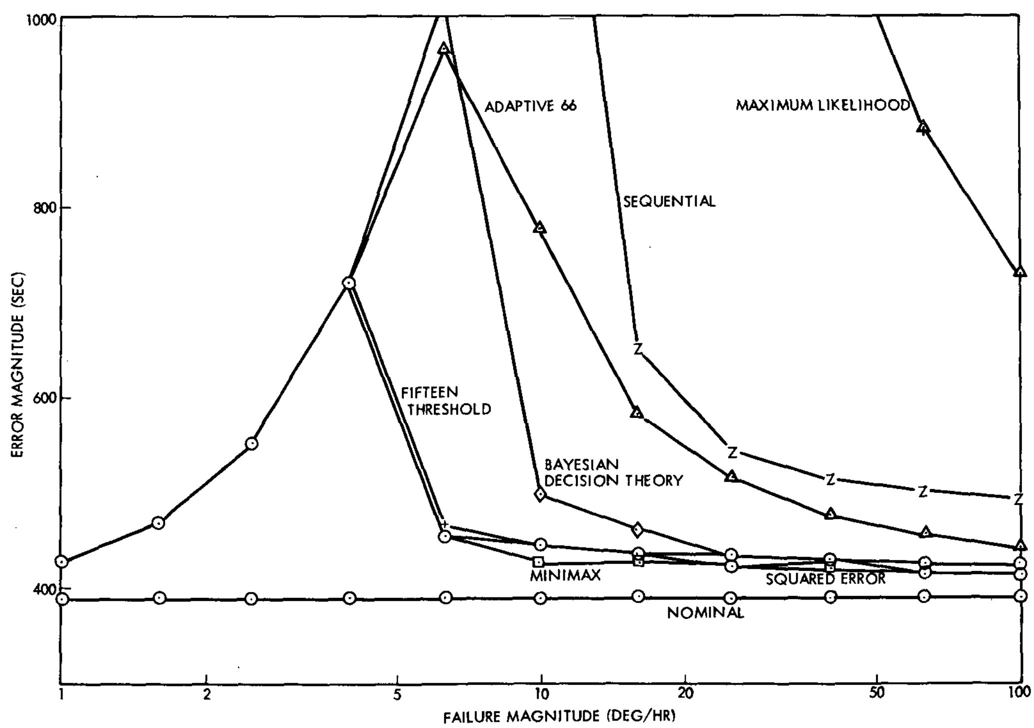


Figure 8-4. Orientation Error Magnitude Versus Failure Magnitude - Second Gyro Failure, 10-Sec Time Constant

Table 8-II. FDIC Parameters

Adaptive 66	
EPS0G = 3.16 CRIT1 = 0.1	TAUPGQ(1) = 10.
Fifteen Threshold	
THG1 = 5.01	TAUPGQ(2) = 10.
Squared Error	
GK1I = 3.54 GK3I = 6.31 TAUPGQ(3) = 10. LHEAL = 0	GK2I = 4/9 GK4I = 8/21 AKGDI = 0.
Bayesian Decision Theory	
THG1I = 5.01 TAUPGQ(4) = 10.	THG2I = 10.
Maximum Likelihood	
SIGQG0 = 8.78 CRIT5 = 0.1	TAUPGQ(5) = 10.
Minimax	
THGMI = 1.55 LSQG = 1	TAUPGQ(6) = 10.
Sequential	
SIGPG(1) = 206000 SIGQG(1) = 206000 SIGRG = 51 TAUSG(1) = 0.0128 THRGI(1) = 618000	SIGPG(2) = 4.03 SIGQG(2) = 0.55 TAUSG(2) = 20. THRGI(2) = 0.196

This behavior of the errors in response to changes in the time constant of the prefilter (with concomittant threshold changes to keep the false alarm rate constant) can be verified from the analysis of Subsection 3.1 qualitatively, if not quantitatively. Figure 8-5 shows Eq. 3-110 plotted for two values of b and τ_f. The values of τ_f are the two values used in the simulation. When the time constants were decreased from 100 to 10 sec, the thresholds were increased by 8 to 10 dB. Thus the ratio between the two values of b used in Figure 8-5 was taken as 9 dB. The left-hand portion of both curves is simply the failure magnitude multiplied by 200 sec (the time from the failure to the end of boost).

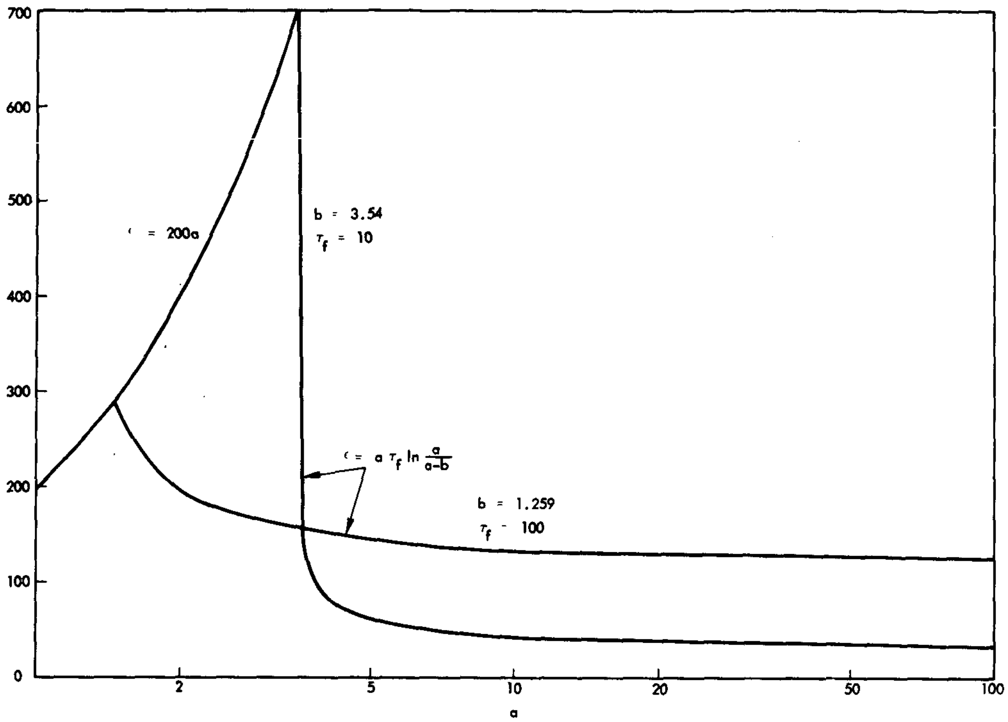


Figure 8-5. Theoretical Error Magnitude Versus Failure Magnitude

8.3 DOUBLE SIMULTANEOUS FAILURE RUNS

A series of runs with double simultaneous gyro failures of different MODE values was made as a severe test of the capabilities of the different algorithms. The three algorithms which gave the poorest results in the previous tests, the Adaptive 66, Maximum Likelihood, and Sequential Algorithms, also required the most central processor time. It was therefore decided to eliminate them from contention at this point, and devote the remaining resources to the evaluation of the more successful algorithms.

On the first trial, the Squared Error, Bayesian Decision Theory, and Minimax Algorithms all experienced false alarms before the failures occurred. Their thresholds were all raised 2 dB, eliminating the false alarms. The gyro FDIC algorithms parameters for these runs are given in Table 8-III.

Table 8-III. FDIC Parameters

Fifteen Threshold	
THG1 = 5.01	TAUPGQ(2) = 10.
Squared Error	
GK1I = 4.46	GK2I = 4/9
GK3I = 7.94	GK4I = 8/21
TAUPGQ(3) = 10.	AKGDI = 0.
LHEAL = 0	
Bayesian Decision Theory	
THG1I = 6.31	THG2I = 12.59
TAUPGQ(4) = 10.	
Minimax	
THGMI = 1.953	TAUPGQ(6) = 10.
LSQG = 1	

Every combination of two failures of MODE 1 through MODE 6 was tried for a total of 21 cases. MODE 7 was omitted because it was felt that no algorithm would have much difficulty with so hard a failure. Different initial random numbers were used for each case. Only one trial was made per case. The magnitudes of the attitude error vector with respect to the ideal attitude and with respect to the nominal algorithm attitude were averaged over the 21 cases, giving the results in Table 8-IV.

Table 8-IV. Results of Double Simultaneous Failures (arc sec)

Algorithm	Ideal	Nominal Algorithm
Nominal	434	0
Fifteen Threshold	688	396
Squared Error	761	465
Bayesian Decision Theory	841	563
Minimax/Least Squares	672	366

The events occurring during the 21 trials are summarized in Table 8-V. The prevalence of false alarms indicates that the thresholds may still be set a bit too low. The prevalence of missed alarms is partly due to the choice of too low-level failures in some of the failure modes.

Table 8-V. Summary of Events

Events	Algorithms			
	Fifteen Threshold	Squared Error	Bayesian Decision Theory	Minimax Least-Squares
False alarms	4	5	3	4
True alarms	23	25	21	25
Missed alarms	19	17	21	17

We see that the two algorithms using the indirect test signals (parity signals) do somewhat better than the two using the direct test signals (residuals). We also note that the test signals of the three algorithms already eliminated from contention are more closely related to the direct than to the indirect test signals. Thus we find a reasonably strong presumption in favor of the indirect test signals being the better choice.

With the benefit of hindsight, we may find plausible reasons why this result is obtained. One reason is that, with 15 signals as compared to six, one can discriminate a greater number of different states in v-space. Another reason is that there is only a 7 dB ratio between the contribution of the i^{th} instrument error to the i^{th} residual and the contribution of the j^{th} instrument error to the i^{th} residual, whereas there is an infinite ratio between the contributions to a parity signal of the error of an instrument included in it as compared to an instrument excluded from it.

8.4 MINOR CYCLE TIME

All of the preceding runs were made with a minor cycle time of 125 msec. Originally it was intended to use 10 msec, which is a value more typical of practical strapdown packages. However, the running time on the CDC 6400/6500 proved to be excessive. The 125-msec period was selected as a good compromise between running time and loss of accuracy. In order to estimate the loss of accuracy, a single trial was run with the nominal algorithm only using the 10-msec period. The resulting attitude error magnitude of 113 sec may be compared with the value of 434 sec given in Table 8-IV.

If we RSS the 113 sec nominal algorithm error with the error with respect to the nominal algorithm achieved in the results above, we get a crude indication of the performance achievable in the presence of failures with the Minimax Algorithm with the 10-msec period.

$$113^2 + 366^2 = 384^2 \qquad (8\text{-}1)$$

This estimate shows some 12 dB of performance degradation in the presence of double failures as compared to the nominal algorithm. In the worst case of the 21 cases, the degradation was 27 dB. Of course, such occurrences will probably be rare, but it is interesting to note that even the best algorithm can occasionally do rather poorly. (However, a slightly higher threshold setting might have eliminated the worst case, which was caused by false alarms.)

9. ALGORITHM SIZING AND TIMING

9.1 ALGORITHM SIZING

Algorithm sizing comparisions are made by finding the storage requirements for the FDIC algorithms on the CDC 6400/6500 computer. Only the MINi subroutines are included. The number of words used for instructions, the number of words used for data, and the total of the two are presented in Table 9-I.

Table 9-I. Algorithm Sizing

Algorithms	Instruction Words	Data Words	Total
Adaptive 66	460	156	616
Fifteen Threshold	358	120	478
Squared Error	608	153	761
Bayesian Decision Theory	526	137	663
Maximum Likelihood	640	202	842
Minimax	700	167	867
Sequential	447	360	807

It should be noted that CDC 6400/6500 words contain several instructions each, so that an accurate instruction count has not been obtained. (The Bounding Sphere Algorithm is excluded from the Minimax Algorithm figures since it was not used; however, the logic deciding whether to use least-squares or bounding spheres and the calling sequence to the subroutine SPHERE is still included. Also, the Minimax Algorithm can be simplified somewhat if it is used only with least-squares. Therefore, the figures for the Minimax Algorithm must be considered to be a bit high.)

9.2 ALGORITHM TIMING

Algorithm timing comparisons are made by keeping track of the central processor time spent on each algorithm. As mentioned above, the peak timing results are too coarsely quantized to be very useful. The total time used by the different algorithms in a typical run is given in Table 9-II.

Table 9-II. Total Processor Time (sec)

Algorithms	Time
Adaptive 66	10.6
Fifteen Threshold	8.5
Squared Error	6.3
Bayesian Decision Theory	7.0
Maximum Likelihood Theory	15.6
Minimax	14.6
Sequential	188.0

However, the Adaptive 66 and Maximum Likelihood Algorithms, unlike any of the others, have iterations or loops of indefinite length in their structure which, on occasion, can consume much more time than even the Sequential Algorithm. For example, the Adaptive 66 Algorithm once used up 479.3 sec of central processor time in a single trial. (The Maximum Likelihood Algorithm was not turned on, fortunately, or it might have done even worse.) The timing on the Minimax Algorithm may be somewhat high, for the same reasons as given in Section 9.1. The average (but not the peak) time could be decreased somewhat at the cost of increased size by reverting to the original scheme, which has a separate failure detection scheme with failure isolation not occurring unless a failure is detected.

10. AREAS FOR FUTURE STUDY

Although the original tasks required in the study have all been completed, a number of areas where additional work ought to be done have become apparent during the course of the study. They will now be discussed.

Up to the time that the Adaptive 72 Algorithm appeared, the question of filtering seemed to be merely a choice between two linear low-pass filters with slightly different impulse responses. The Adaptive 72 Algorithm uses a nonlinear filter described as "a suboptimal detection system based on Wald's sequential analysis using the concept of information value and information feedback." This filter could be used by the Fifteen Threshold, Bayesian Decision Theory, and Minimax Algorithms as well as by the Adaptive 72 Algorithm. It would be worthwhile to compare the three kinds of filters in the same algorithm.

The Adaptive 72 Algorithm failure detection, isolation, and correction schemes should be coded for evaluation in FAILSIM when the documentation of the SIRU application becomes available. The identification and recertification schemes need not be studied at this time as there is no competitor on the scene — merely confirming the originator's results would not be worthwhile.

The original program plan did not include the evaluation of the effects of "glitches" (temporary failures) on algorithm performance. However, the capability of healing failures was included in the failure programmer of FAILSIM. A study of the effects of glitches using FAILSIM would be desirable. It will probably lead to the modification of some of the algorithms, however, because none of them appear to have been designed with glitches specifically in mind. The hypothesis that an instrument has healed itself is not the complement of the hypothesis that an instrument has failed, because, in the first case, we know which instrument we are talking about before we make the decision, while in the second case we do not and must, in effect, make six decisions. Thus the decision that an instrument should be reinstated should not necessarily be based on the disappearance of the conditions that caused it to be classified as failed in the first place.

A goal (rather than a requirement) of the original program plan was to combine the best features of the algorithms into a super-algorithm. In addition to the obvious combination of the best filter or prefilter with the best isolation technique, other possibilities exist. An example is that of applying the signal locking technique used with the Fifteen Threshold Algorithm to the Minimax Algorithm to see whether improved isolation results. Thus this goal is still a desirable one.

REFERENCES

1. Weiss, R. and Nathan, I., "An Ultra-Reliable Attitude Reference System for a Manned Orbiting Laboratory with Limited Sparing Capability," AIAA/ JACC Guidance and Control Conference, Seattle, Washington, August 1966.

2. Weiss, R. and Nathan, I., "An Ultra-Reliable Sensing System for Vehicles with Limited Sparing Capability," J S&R 4, 1967, pp 1151-1158.

3. Ephgrave, J. T., "Optimum Redundant Configurations of Inertial Sensors," TOR-1001(9990)-5, Aerospace Corporation, El Segundo, California, September 1966.

4. Crisp, R., Gilmore, J. P., and Hopkins, A., Jr., "SIRU-A New Inertial System Concept for Inflight Reliability and Maintainability," E-2407, MIT Instrumentation Lab., Cambridge, Massachusetts, May 1969.

5. Gilmore, J. P., "A Non-Orthogonal Gyro Configuration," Master of Science Thesis, MIT, Cambridge, Massachusetts, January 1967.

6. Gilmore, J. P., "A Non-Orthogonal Multi-Sensor Strapdown Inertial Reference Unit," E-2308, MIT Instrumentation Lab., Cambridge, Massachusetts, August 1968.

7. Ephgrave, J. T., "Redundant Adaptive Strapdown Inertial Navigation System," TOR-0066(5306)-10, Aerospace Corporation, El Segundo, California, October 1969.

8. Evans, F. A. and Wilcox, J. C., "Experimental Strapdown Redundant Sensor Inertial Navigation System," AIAA Guidance, Control, and Flight Mechanics Conference, Princeton, N. J., August 1969 and J S&R 7, 1070-1074 (1970).

9. Wilcox, J. C., "Candidate Configuration for Space Attitude Reference," Interoffice Correspondence 7223.2-230, TRW Systems Group, Redondo Beach, California, 19 October 1967.

10. Wilcox, J. C., "Failure Detection, Diagnosis, and Correction Design and Analysis," 09665-6012-R0-00, TRW Systems Group, Redondo Beach, California, January 1969.

11. Gilmore, J. P. and McKern, R. A., "A Redundant Strapdown Inertial System Mechanization - SIRU," E-2527, MIT Charles Stark Draper Laboratory, Cambridge, Mass., August 1970, and AIAA Guidance, Control, and Flight Mechanics Conference, Santa Barbara, California, August 1970.

12. Gilmore, J. P. and McKern, R. A., "A Redundant Strapdown Inertial Reference Unit (SIRU)," J S&R 9, 39-47, 1972.

13. Oehrle, J. B., "Failure Isolation in SIRU," ISS Memo No. 70-8, SIRU Memo No. 328, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 20 January 1970.

14. Oehrle, J. B., "SIRU Software Description and Program Documentation (Preliminary)," ISS Memo No. 72-82, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 17 May 1972.

15. Gully, S. W., "Redundant Sensor Strapdown Guidance System Study," 70-POD-15, General Electric Ordnance Department, Pittsfield, Mass., 15 July 1970.

16. Gully, S. W., "Appendix A," undated, unserialized, received from author.

17. Wilcox, J. C., "Maximum Likelihood Failure Detection for Redundant Inertial Instruments," 99900-7351-R0-00, TRW Systems Group, Redondo Beach, California, July 1970.

18. Wilcox, J. C., "Maximum Likelihood Failure Detection for Redundant Inertial Instruments," AIAA Guidance and Control Conference, Stanford, California, August 1972.

19. Anon., First Quarterly Progress Report, Contract NAS 9-4065, Task Order 41, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 30 August 1971, Chapter III.

20. Anon., Second Quarterly Progress Report, Contract NAS 9-4065, Task Order 41, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 26 November 1971, Chapter I.

21. Anon., Third Quarterly Progress Report, Contract NAS 9-4065, Task Order 41, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 28 February 1972, Chapters I, II, and III.

22. Anon., Fourth Quarterly Progress Report, Contract NAS 9-4065, Task Order 41, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 29 June 1972, Chapters I, II, III, and IV.

23. Anon., Summary Report, Contract NAS 9-4065, Task Order 41, Charles Stark Draper Laboratory, Cambridge, Mass., 31 July 1972, Chapter II.

24. Anon., Supplementary Report, Contract NAS 9-4065, Task Order 41, MIT Charles Stark Draper Laboratory, Cambridge, Mass., 2 November 1972.

25. Potter, J. E. and Deckert, J. C., "Gyro and Accelerometer Failure Detection and Identification in Redundant Sensor Systems," E-2686, MIT Charles Stark Draper Laboratory, Cambridge, Mass., May 1972.

26. Potter, J. E. and Deckert, J. C., "Gyro and Accelerometer Failure Detection and Identification in Redundant Sensor Systems," ION National Space Meeting, Orlando, Fla., March 1972.

27. Chien, T.-T., "An Adaptive Technique for a Redundant-Sensor Navigation System," Doctor of Science Thesis T-560, Charles Stark Draper Laboratory, MIT, Cambridge, Mass., February 1972.

28. Chien, T.-T., "An Adaptive Technique for a Redundant-Sensor Navigation System," AIAA Guidance and Control Conference, Stanford, California, August 1972.

29. Eckelkamp, R. E. and Schiesser, E. R., "Application of Sequential Filter Techniques to the Management of Redundant Strapdown Platforms," NASA Manned Spacecraft Center Internal Note No. 72-FM-167, 12 July 1972.

30. Cockrell, B. F., "IMU Error Models for Shuttle Navigation Studies," MSC Internal Note No. 72-FM-35, NASA Manned Spacecraft Center, Houston, Texas, 16 February 1972.

31. Wilcox, J. C., "A New Algorithm for Strapped Down Inertial Navigation," IEEE Trans. AES 3, 1967, pp 796-802.

32. McKern, R. A., "A Study of Transformation Algorithms for Use in a Digital Computer," T-493, MIT Charles Stark Draper Laboratory, Cambridge, Mass., January 1968.

33. Yachter, M., "A Novel Computational Technique for No-Gimbal Inertial Guidance Systems," in "No-Gimbal Inertial Guidance System for Ballistic Missile Applications," American Bosch Arma Corp., Garden City, N. Y., Interim Engineering Report 1, AD345568.

34. Yachter, M., "Error Analysis of No-Gimbal Computation Technique (Novel Method)," in "No-Gimbal Inertial Guidance System for Ballistic Missile Application," American Bosch Arma Corp., Garden City, N. Y., Interim Engineering Report 2, AD 347810.

APPENDIX A
TYPICAL RUN

The inputs and outputs of a typical run are presented in this appendix.

A.1 NAMELIST INPUTS

The NAMELIST inputs are presented in this section. They are typical of the inputs used during the final series of runs, the double simultaneous failure runs. Although algorithms 1, 5, and 7 are turned off, their parameters are left in the input file for convenience. The values of these parameters are the final set used in the 10-sec time constant runs.

This run has only one case. To add more cases, one simply appends their data decks. The data for each case must be preceded by a $INPUT card and terminated by a $. All cards must start in column 2, except the comments which have a C in column 1.

The case has only one trial. To add more trials, set N1MAX to the number of trials.

```
CDATA FOR FAILIN4, CASE 1
 $INPUT
 DT = .125
 IRDM = 2517566905
 LMONTE = 0
 LPLOT = 0
 N1MAX = 1
 N2MAX = 400
 N3MAX = 2
 N4MAX = 4
 AVBS = 1., 1., 1., .981, .981, .981
 TAUV = .04, .04, .04, .04, .04, .04
 KV = 7., 7., 7., .7, .7, .7
 KP = 25., 25., 25., .25, .25, .25
 RADIUS = 6.
 IR1 = 1, IR2 = 2, IR3 = 3
 THTR1I = 0., THTR2I = 0., THTR3I = 0.
 THTAI = 0., 0., 0., 0., 0., 0.
 THTGI = 0., 0., 0., 0., 0., 0.
 SIGUFEA = 86., 1., 70., 50., 50.
 SIGUFEG = .083, .01, 100., 10., 10., .2, .02, .2, .04
 QACCI = 2.
 QGYROI = 5.
 NFAIL(1) = 1600, 1600
 NINST(1) = 1, 6
 MODE(1) = 1, 1
 FAIL1I(1) = 10., 10.
 LALG0 = 1
 LALG1 = 0
 EPSOA = 1585.
 EPSOG = 3.16
 TAUPAQ(1) = 10.
 TAUPGQ(1) = 10.
 CRIT1 = .1
 LALG2 = 1
 THAI = 1995.
 THGI = 5.01
 TAUPAQ(2) = 10.
 TAUPGQ(2) = 10.
 LALG3 = 1
 LHEAL = 0
```

```
AK1I = 2240.
AK2I = .444444444444444
AK3I = 1995.
AK4I = .380952380952381
GK1I = 4.46
GK2I = .444444444444444
GK3I = 7.94
GK4I = .380952380952381
TAUPAQ(3) = 10.
TAUPGQ(3) = 10.
LALG4 = 1
THA1I = 3160.
THA2I = 3160.
THG1I = 6.31
THG2I = 12.59
TAUPAQ(4) = 10.
TAUPGQ(4) = 10.
LALG5 = 0
SIGQA0 = 5540.
SIGQG0 = 8.78
TAUPAQ(5) = 10.
TAUPGQ(5) = 10.
CRIT5 = .1
LALG6 = 1
LSQA = 1
LSQG = 1
THAMI = 978.
THASI = 100.
THGMI = 1.953
THGSI = .1
TAUPAQ(6) = 10.
TAUPGQ(6) = 10.
LALG7 = 0
SIGPA = 3.E6, 65.8
SIGPG = 206000., 4.03
SIGQA = 3.E6, 8.99
SIGQG = 206000., .550
SIGRA = 833.
SIGRG = 51.
TAUSA = .0128, 20.
TAUSG = .0128, 20.
THRAI = 9.E6, 7.58
THRGI = 618000., .196
$END
```

A.2 TRAJECTORY SUBROUTINE

The subroutine, TRAJ, that defines the nominal trajectory is presented here. It represents a 400-sec shuttle boost trajectory. See Paragraph 6.2.8 for a detailed description of the method. Note that VPAB and APLB are zeroed out in the main program, FAIL, before each trial, so that statement 80, for example, computes only the nonzero component of angular velocity.

```
*TRAJ
      SUBROUTINE TRAJ
      COMMON ROM, A(6,3), AMASK(15), BAQ(3,6,9), BGQ(3,6,9), C(3,6),
     1 CF(15,3), MPSSMCG, OMASK(6), OUTP(12), PHISTRQ(3,9), RADSEC,
     2 RHOQ(4,9), RHOO(4), SAQ(9), SGQ(9), T, TIME(8,2,2), VBQ(3,9),
     3 VIQ(3,9)
      COMMON ACC(3), AKGOI,  AK1I, AK2I, AK3I, AK4I, AR(12,3,3), DT,
     1 DTMAJ, DT2, EPSOA, EPSOG, FAIL1(12), FAIL2(12), GK1I, GK2I, GK3I,
     2 GK4I, I, IDATA, INST(12), IROM, IR1, IR2, IR3, KCUNTA(8,9),
     3 KCUNTG(8,9), LALG(9), LHEAL, LMONTE, LPLOT, LSQA, LSQG, MODE(14),
     4 NFAIL(14), NINST(14), N1MAX, N2, N2MAX, N3, N3MAX, N4MAX, OMG(3),
     5 OUT, QACC, QGYRO, CRIT1, CRIT5, SAQ0(9), SAQ1(9), SGQ0(9),
     6 SGQ1(9), SIGPA(2), SIGPG(2), SIGQA(2), SIGQA0, SIGQG(2), SIGQG0,
     7 SIGRA, SIGRG, SIGUFEA(5), SIGUFEG(9), TAUPAQ(6), TAUPGQ(6),
     8 TAUSA(2), TAUSG(2), TFAIL(12), THAI, THAMI, THASI, THA1I, THA2I,
     9 THGI, THGMI, THGSI, THG1I, THG2I, THRAI(2), THRGI(2), THTA(6),
     A THTG(6), THTR1, THTR2, THTR3, UFE(12,9), X(7), XFAIL1(14),
     B XFAIL2(14)
      COMMON APLB(3), AVB(6), AVBA(6), AVBB(6), AVBS(6), AVBO(6),
     1 AVB1(6), AVB2(6), AVB3(6), A1(6), A2(6), BO(6), B1(6), B2(6),
     2 CO(6), C2(6), DX(7), KP(6), KV(6), Q(7), RADIUS, TAUV(6),
     3 TM(4,4), VPAB(3), VVAB(3)
C PROGRAMMED TRAJECTORY BEFORE ADDITION OF RANDOM VIBRATIONS
      N = N3MAX*(N2 - 1) + N3
C BODY ANGULAR VELOCITY
      IF (N .GT. 393) GO TO 40
      IF (N .GT. 224) GO TO 20
      IF (N .GT. 40) GO TO 10
      IF (N .GT. 16) 90,80
10    IF (N .GT. 140) 110,100
20    IF (N .GT. 232) GO TO 30
      IF (N .GT. 225) 130,120
30    IF (N .GT. 233) 150,140
40    IF (N .GT. 420) GO TO 60
      IF (N .GT. 414) GO TO 50
      IF (N .GT. 394) 170,160
50    IF (N .GT. 415) 190,180
60    IF (N .GT. 781) GO TO 70
      IF (N .GT. 421) 210,200
70    IF (N .GT. 782) 230,220
C 0. TO 8. SEC.
80    VPAB(2) = -3.4E-4 - 1.5E-6*T
```

```
      GO TO 260
C 8. TO 20. SEC.
90    VPAB(2) = -6.7E-3 - 8.E-5*T
      GO TO 260
C 20. TO 70. SEC.
100   VPAB(2) = 2.604E-3 + (-6.684E-4 + 6.16E-6*T)*T
      GO TO 260
C 70. TO 112. SEC.
110   VPAB(2) = -2.17E-2 + 1.1E-4*T
      GO TO 260
C 112. TO 112.5 SEC.
120   VPAB(1) = -1.3E-3
      VPAB(2) = 8.2E-2
      VPAB(3) = 5.05E-2 - 6.25E-4*T
      GO TO 260
C 112.5 TO 116 SEC.
130   VPAB(1) = -1.E-3
      VPAB(2) = .1305 - 1.E-3*T
      VPAB(3) = 5.05E-2 - 6.25E-4*T
      GO TO 260
C 116. TO 116.5 SEC.
140   VPAB(1) = -.24112 + 2.07E-3*T
      VPAB(2) = 4.1441 - 3.56E-2*T
      VPAB(3) = -5.29304 + 4.544E-2*T
      GO TO 260
C 116.5 TO 196.5 SEC.
150   VPAB(1) = -1.888125E-5 + 4.625E-7*T
      VPAB(2) = -3.8825E-3 + 5.E-6*T
      VPAB(3) = 8.365E-4 - 1.E-6*T
      GO TO 260
C 196.5 TO 197. SEC.
160   VPAB(1) = 2.8368E-2 - 1.44E-4*T
      VPAB(2) = -1.1426 + 5.8E-3*T
      VPAB(3) = .25216 - 1.28E-3*T
      GO TO 260
C 197. TO 207. SEC.
170   VPAB(2) = 0.
      GO TO 240
C 207. TO 207.5 SEC.
180   VPAB(2) = -11.178 + 5.4E-2*T
      VPAB(3) = -.828 + 4.E-3*T
      GO TO 260
```

```
C 207.5 TO 210. SEC.
190   VPAB(2) = 2.7E-2
      VPAB(3) = -8.1E-2 + 4.E-4*T
      GO TO 260
C 210. TO 210.5 SEC.
200   VPAB(2) = 12.207 - 5.8E-2*T
      VPAB(3) = 1.011 - 4.8E-3*T
      GO TO 260
C 210.5 TO 390.5 SEC.
210   VPAB(2) = -2.E-3
      VPAB(3) = 6.E-4
      GO TO 260
C 390.5 TO 391. SEC.
220   VPAB(2) = -1.9E-2
      VPAB(3) = 7.2E-3
      GO TO 260
C 391. TO 400. SEC.
230   VPAB(2) = 0.
C ZERO COMPONENTS
240   VPAB(3) = 0.
250   VPAB(1) = 0.
C BODY ACCELERATION
260   IF (N .GT. 272) GO TO 290
      IF (N .GT. 140) GO TO 280
      IF (N .GT. 100) GO TO 270
      IF (N .GT. 80) 320,310
270   IF (N .GT. 112) 340,330
280   IF (N .GT. 232) GO TO 370
      IF (N .GT. 224) 360,350
290   IF (N .GT. 414) GO TO 300
      IF (N .GT. 395) GO TO 400
      IF (N .GT. 394) 390,380
300   IF (N .GT. 720) GO TO 430
      IF (N .GT. 415) 420,410
C 0. TO 40. SEC.
310   APLB(1) = 14. + .054*T
      APLB(3) = -.25 - .00625*T
      GO TO 440
C 40. TO 50. SEC.
320   APLB(1) = 16.16
      APLB(3) = -.5
      GO TO 440
```

```
C 50. TO 56. SEC.
330    APLB(1) = 20.16 - .08*T
       GO TO 440
C 56. TO 70. SEC.
340    APLB(1) = 10.08 + .1*T
       GO TO 440
C 70. TO 112. SEC.
350    APLB(1) = 4.48 + .18*T
       GO TO 440
C 112. TO 116. SEC.
360    APLB(1) = 2.24 + .2*T
       APLB(3) = 41.5 - .375*T
       GO TO 440
C 116. TO 136. SEC.
370    APLB(1) = 2.24 + .2*T
       APLB(3) = -.956 - .009*T
       GO TO 440
C 136. TO 197. SEC.
380    APLB(1) = 29.44
       APLB(3) = .54 - .02*T
       GO TO 440
C 197. TO 197.5 SEC.
390    APLB(1) = 11636.68 - 58.92*T
       APLB(3) = -1319.36 + 6.68*T
       GO TO 440
C 197.5 TO 207. SEC.
400    APLB(1) = -.02
       APLB(3) = -.06
       GO TO 440
C 207. TO 207.5 SEC.
410    APLB(1) = -6106.52 + 29.5*T
       APLB(3) = -24.9 + .12*T
       GO TO 440
C 207.5 TO 360. SEC.
420    APLB(1) = 1./(.11402813064487 - 2.2235888217603E-4*T)
       APLB(3) = 0.
       GO TO 440
C 360. TO 400. SEC.
430    APLB(1) = 29.43
440    RETURN
       END
```

A. 3 OUTPUTS

This subsection presents the outputs of a typical run.

FAILSIM

02/24/73. 09.24.12.

PROGRAM CONTROL

```
      DT  1.2500000E-01       IRDM      2517566905      LMONTE    0        LPLOT    0        N1MAX    1       N2MAX 400
   N3MAX   2        N4MAX    4
```

VEHICLE VIBRATORY MOTIONS

```
    AVBS  1.0000000E+00  1.0000000E+00  1.0000000E+00  9.8100000E-01  9.8100000E-01  9.8100000E-01
    TAUV  4.0000000E-02  4.0000000E-02  4.0000000E-02  4.0000000E-02  4.0000000E-02  4.0000000E-02
      KV  7.0000000E+00  7.0000000E+00  7.0000000E+00  7.0000000E-01  7.0000000E-01  7.0000000E-01
      KP  2.5000000E+01  2.5000000E+01  2.5000000E+01  2.5000000E-01  2.5000000E-01  2.5000000E-01
  RADIUS  6.0000000E+00
```

INSTRUMENT ORIENTATION

```
     IR1    1      THTR1I  0.
     IR2    2      THTR2I  0.
     IR3    3      THTR2I  0.
   THTAI  0.             0.             0.             0.             0.             0.
   THTGI  0.             0.             0.             0.             0.             0.
```

UNFAILED ERRORS

```
 SIGUFEA  8.6000000E+01  1.0000000E+00  7.0000000E+01  5.0000000E+01  5.0000000E+01
 SIGUFEG  8.3000000E-02  1.0000000E-02  1.0000000E+02  1.0000000E+01  1.0000000E+01  2.0000000E-01
          2.0000000E-02  2.0000000E-01  4.0000000E-02
   QACCI  2.0000000E+00      QGYROI  5.0000000E+00
```

FAILURES

NFAIL	NINST	MODE	FAIL1I	FAIL2I
1600	1	1	1.0000000E+01	0.
1600	6	1	1.0000000E+01	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.
0	0	0	0.	0.

```
             0              0              0         0.              0.
             0              0              0         0.              0.

NOMINAL ALGORITHM
   LALG0    1

ADAPTIVE ALGORITHM
   LALG1    0

FIFTEEN THRESHOLD ALGORITHM
   LALG2    1
    THAI  1.9950000E+03      THGI  5.0100000E+00     TAUPAQ  1.0000000E+01     TAUPGQ  1.0000000E+01

SQUARED ERROR ALGORITHM
   LALG3    1
   LHEAL    0
    AK1I  2.2400000E+03      AK2I  4.4444444E-01       AK3I  1.9950000E+03       AK4I  3.8095238E-01
    GK1I  4.4600000E+00      GK2I  4.4444444E-01       GK3I  7.9400000E+00       GK4I  3.8095238E-01
  TAUPAQ  1.0000000E+01    TAUPGQ  1.0000000E+01      AKGDI  0.

BAYESIAN DECISION THEORY ALGORITHM
   LALG4    1
   THA1I  3.1600000E+03     THA2I  3.1600000E+03      THG1I  6.3100000E+00      THG2I  1.2590000E+01
  TAUPAQ  1.0000000E+01    TAUPGQ  1.0000000E+01

MAXIMUM LIKELIHOOD ALGORITHM
   LALG5    0

MINIMAX ALGORITHM
   LALG6    1
    LSQA    1         LSQG    1
   THAMI  9.7800000E+02     THASI  1.0000000E+02      THGMI  1.9530000E+00      THGSI  1.0000000E-01
  TAUPAQ  1.0000000E+01    TAUPGQ  1.0000000E+01

SEQUENTIAL ALGORITHM
   LALG7    0
```

TIME 200.000 GYRO NO. 1 FAILED IN MODE 1
TIME 200.000 GYRO NO. 6 FAILED IN MODE 1
TIME 203.875 ALG. NO. 4 GYRO NO. 1 TRUE ALARM
TIME 206.250 ALG. NO. 6 GYRO NO. 1 TRUE ALARM
TIME 220.375 ALG. NO. 6 GYRO NO. 6 TRUE ALARM
TIME 343.875 ALG. NO. 3 GYRO NO. 6 TRUE ALARM
TIME 400.000 ALG. NO. 2 GYRO NO. 1 MISSED ALARM
TIME 400.000 ALG. NO. 2 GYRO NO. 6 MISSED ALARM
TIME 400.000 ALG. NO. 3 GYRO NO. 1 MISSED ALARM
TIME 400.000 ALG. NO. 4 GYRO NO. 6 MISSED ALARM

TRIAL NO. 1, MAGNITUDES OF ANGLE AND VELOCITY ERRORS

508.83	0.00	972.60	978.51	1039.77	0.00	425.02	0.00	0.00
	0.00	943.32	777.40	1319.35	0.00	119.96	0.00	0.00
13.0010	7946.3783	15.8938	15.8011	8.5079	7946.3783	11.7081	7946.3783	7946.3783
	7951.6343	11.9667	10.7878	12.5036	7951.6343	2.3058	7951.6343	7951.6343

FDIC TIMES FOR TRIAL NO. 1

0.000	8.135	5.704	6.842	0.000	13.692	0.000	0.000
0.000	.004	.006	.007	0.000	.009	0.000	0.000

IRDM = 10975562961

TOTAL TIME = 132.147

MEAN MAGNITUDES OF ANGLE AND VELOCITY ERRORS

508.83	0.00	972.60	978.51	1039.77	0.00	425.02	0.00	0.00
	0.00	943.32	777.40	1319.35	0.00	119.96	0.00	0.00
13.0010	7946.3783	15.8938	15.8011	8.5079	7946.3783	11.7081	7946.3783	7946.3783
	7951.6343	11.9667	10.7878	12.5036	7951.6343	2.3058	7951.6343	7951.6343

FDIC TIMES FOR ALL TRIALS

0.000	8.135	5.704	6.842	0.000	13.692	0.000	0.000
0.000	.004	.006	.007	0.000	.009	0.000	0.000

	GYRO ALARM STATISTICS								ACCELEROMETER ALARM STATISTICS							
ALGORITHM	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
ALL CLEAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CORRECTED FALSE ALARM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FALSE ALARM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LOST ALARM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TRUE ALARM	0	0	1	1	0	2	0	0	0	0	0	0	0	0	0	0
FORTUITOUS TRUE ALARM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FORTUITOUS ALL CLEAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MISSED ALARM	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
MISSED ALL CLEAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX B
THE SEQUENTIAL ALGORITHM IN STEADY STATE

The Kalman-Bucy filter equations for one of the three identical one-state filter equations derived in Paragraph 3.1.2.4 are

$$\hat{x}_i = \bar{x}_i + k_i(y_i - \bar{x}_i) \tag{B-1}$$

$$k_i = \frac{m_i}{m_i + r} \tag{B-2}$$

$$p_i = (1 - k_i)\, m_i \tag{B-3}$$

$$\bar{x}_{i+1} = \phi\, \hat{x}_i \tag{B-4}$$

$$m_{i+1} = \phi^2\, p_i + q \tag{B-5}$$

where ϕ, m, and q correspond to ϕ_2, m_2, and q_2, respectively, in Eqs. 3-76, 3-77, and 3-78.

If we combine Eq. B-1 and B-4, we obtain

$$\hat{x}_{i+1} = \phi\, \hat{x}_i + k_{i+1}\,(y_{i+1} - \phi\, \hat{x}_i) \tag{B-6}$$

Taking the Z transform, and assuming k_i constant,

$$\hat{x} = \frac{kz}{z - \phi(1 - k)}\, y \tag{B-7}$$

The Z transform of a first-order constant coefficient filter of gain K_f and time constant τ_f is

$$x = \frac{K_f\left(1 - e^{-\frac{T}{\tau_f}}\right)}{z - e^{-\frac{T}{\tau_f}}}\, y \tag{B-8}$$

Thus we see that, in the steady state,

$$k = K_f\left(1 - e^{-\frac{T}{\tau_f}}\right) \tag{B-9}$$

$$\phi\,(1 - k) = e^{-\frac{T}{\tau_f}} \tag{B-10}$$

are the equations relating the steady-state Kalman filter to the constant coefficient filter. If we choose ϕ as

$$\phi = e^{-\frac{T}{\tau_\phi}} \tag{B-11}$$

then

$$k = 1 - e^{-\left(\frac{T}{\tau_f} - \frac{T}{\tau_\phi}\right)} \tag{B-12}$$

Now consider the covariance equation. From Eqs. B-2, B-3, and B-5,

$$p_{i+1} = \left(1 - k_{i+1}\right)\left(\phi^2 p_i + q\right) \tag{B-13}$$

$$k_{i+1} = \frac{\phi^2 p_i + q}{\phi^2 p_i + q + r} \tag{B-14}$$

Using k_{i+1} from Eqs. B-14 and B-13 gives

$$p_{i+1} = \frac{r\left(\phi^2 p_i + q\right)}{\phi^2 p_i + q + r} \tag{B-15}$$

From Eqs. B-14 and B-15, we have

$$p_{i+1} = r\,k_{i+1} \tag{B-16}$$

In steady state

$$p_i = p_o \qquad \text{(B-17)}$$

and

$$p_o = r\,k \qquad \text{(B-18)}$$

From Eqs. B-14, B-17, and B-18 we can obtain

$$q = p_o \frac{\left[1 - (1 - k)\,\phi^2\right]}{1 - k} \qquad \text{(B-19)}$$

Thus we can set up the Kalman-Bucy filter to have a desired steady-state time constant. First we choose the time constant τ_f and the measurement variance r. Then we select the state transition matrix by choosing τ_ϕ (a typical value would be $2\tau_f$). The steady-state Kalman-Bucy filter gain k is then given by Eq. B-12. The equivalent filter gain K_f may then be found from Eq. B-9. We now find the initial state covariance p_o from Eq. B-18 and the state noise covariance q from Eq. B-19. The filter will now start and remain in steady state.

If we wish to, we may now assign some other value to p_o (without changing q). The filter will then undergo a transient, eventually reaching the same steady state characterized by K_f and τ_f.