1992-03

# An indicator of message quality for a single optical sensor using a template based tracking algorithm

Bechhoefer, Eric R.

Monterey, California. Naval Postgraduate School

(2)

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California





# THESIS

AN INDICATOR OF MESSAGE QUALITY
FOR A SINGLE OPTICAL SENSOR
USING A TEMPLATE BASED TRACKING
ALGORITHM

by

Eric R. Bechhoefer

March 1992

Thesis Advisor: Lyn R. Whitaker

Approved for public release; distribution is unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY<br>2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School<br>6b. OFFICE SYMBOL<br>OR | 7a. NAME OF MONITORING ORGANIZATION |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Monterey, CA 93943-5000 | 7b. ADDRESS (City, State, and ZIP Code) |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b. OFFICE SYMBOL | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code) | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO. |

11. TITLE (Including Security Classification)
An Indicator of Message Quality for a Single Optical Sensor Using a Template Based Tracking Algorithm

12 PERSONAL AUTHOR(S)
BECHHOEFER, Eric R.

| 13 TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. Page Count |
|---|---|---|---|
| Master's thesis | FROM TO | 1992, MARCH | 75 |

16. SUPPLEMENTAL NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

| FIELD | GROUP | SUB-GROUP |
|---|---|---|
| | | |
| | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Template based tracking, Single optical tracking, Balistic tracking, Statistical analysis, Least squares procedure.

19. ABSTRACT (Continue on reverse if necessary and identify by block number


The volume of messages generated by spaced based interceptors (SBI's) resulting from a booster launch can lead to unacceptably great total time for the messages to propagate through the system. In order to help relieve this problem, one might identify the SBI's with the highest quality estimates of the launch information. Message traffic can be sharply reduced if these SBI's can be identified, and message transmission is restricted to their messages

Launch parameters and position are estimated using a template based tracking algorithm. A single measure of quality based on the estimated covariance matrix of the measured position is proposed and tested using simulation. Results describe possible modifications to the template based tracking algorithm that would reduce error and allow the quality of a message to be determined.


| | |
|---|---|
| 20 DISTRIBUTION/AVAILABILTIY OF ABSTRACT<br>[X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC | 1a. REPORT SECURITY CLASSIFICATION<br>Unclassified |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Lyn R. Whitaker | 22b. TELEPHONE (Include Area Code)<br>(408)646-3482<br>22c. OFFICE SYMBOL<br>OR/Wh |

DD Form 1473, JUN 86 Previous editions are obsolete.
S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

# An Indicator of Message Quality for a Single Optical Sensor Using a Template Based Tracking Algorithm

by


Eric R. Bechhoefer
Lieutenant, United States Naval Reserve
B.S., University of Michigan, 1984


Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

Author: [redacted]

Eric R. Bechhoefer

Approved by: [redacted]

Lyn R. Whitaker, Thesis Advisor

[redacted]

Robert R. Read, Second Reader

[redacted]

Peter Purdue, Chairman
Department of Operations Research

# ABSTRACT

The volume of messages generated by spaced based interceptors (SBI'S) resulting from a booster launch can lead to an unacceptably large total time for the messages to propagate through the system. In order to help relieve this problem, one might identify the SBI's with the highest quality estimates of the launch information. Message traffic can by sharply reduced if these SBI's can be identified, and message transmission restricted to their messages.

Launch parameters and position are estimated using a template based tracking algorithm. A single measure of quality based on the estimated covariance matrix of the measured position is proposed and tested using simulation. Results describe possible modifications to the template based tracking algorithm that would reduce error and allow the quality of a message to be determined.

## THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

## TABLE OF CONTENTS

# I. INTRODUCTION

## A. BACKGROUND

The Strategic Defense Initiative Organization (SDIO) has developed a goal to vigorously research and develop technology that could help to eliminate the threat of ballistic missiles and provide increased U.S. and allied security. By deploying a three-part phased ballistic missile defense system, incremental strategic benefits can be realized while preparing the way for the next phase.

The first phase would reduce confidence of planners initiating a nuclear attack against the U.S. by not allowing them to predict the outcome of a ballistic missile attack. The second phase would negate the potential threat government's or hostile organization's ability to destroy many of the U.S. strategic targets, and the third phase would eliminate the threat posed by ballistic missiles entirely (Udall, 1988).

The first phase system being proposed by SDIO includes ground and space based BMD consisting of:

- Spaced-based hit-to-kill vehicles for attacking missile boosters and post-boost vehicles.
- Ground based rockets designed to intercept warheads as they reenter the atmosphere.

These spaced based hit-to-kill interceptors (SBI's) would be arranged in a constellation of several hundred satellites at several hundred kilometers altitude above the earth. A constellation of satellites is an organized collection of satellites in related orbits. Each satellite would have the capability to detect ballistic or anti-satellites missile launches by observing the hot rocket plumes.

Once a ballistic missile has been detected, the SBI would be able to track the booster and pass this information to the rest of the constellation. This information sharing has two uses:

- Due to positional constraints, the SBI that is tracking a booster may not be the SBI that has the best shot at killing the booster. Tracking information can be passed to the SBI with the best shot (Compaterto, 1991).

- Line of sight laser communication will be used to minimize jamming. This requires sequential message transmission from one SBI to another SBI and could result in large queues of messages being formed (Comparetto, 1991).

The use of a constellation of orbiting SBIs to identify, track and engage thrusting bodies has numerous advantages. The system can be made to operate autonomously, provide world wide coverage, and it is flexible to changing political situations.

A constellation of SBIs will consist of hundreds of platforms, each with identical capability. The large number of platforms will result in multiple coverage of any given area. Given a booster launch, more than one SBI will observe the launch and commence tracking that booster. Due to different observation angles, the position of the sun, and the individual sensor systems themselves, the tracking quality of these SBIs will be variable. Some tracking information will be better than others. Information of high quality should be communicated, while poor quality information should not be passed on to other

SBIs. This type of pruning will reduce time in queues, decrease the time for information being transmitted throughout the constellation, and will allow the SBI with the best shot a higher probability of kill (Comparetto, 1991).

The software developed to simulate a constellation of SBIs is one of the Strategic Defense System (SDS) Simulators. The template based tracking algorithm is a function of this simulator and tracks booster and ballistic bodies using a single optical tracker. It is this papers goal to:

- Describe the functionalities of the template based tracking algorithm and how it works.
- Determine if there is a reliable way to measure the quality of a track message generated by the tracking algorithm.
- Make appropriate conclusions and recommendation to improve the template based tracking algorithm.

Chapter II will give a brief description of the system simulator and how the track algorithm works. Chapter III will describe a measure of quality for a track message and test this measure using the tracking algorithm. Chapter IV will show the effect of changing the azimuth and elevation variance of the sensor on the tracking algorithms error. Chapter V will give the conclusions and recommendations.

# II. DESCRIPTION OF THE SDS SIMULATOR

## A. OVERVIEW

The system simulator uses the template based tracking algorithm to estimate the launch parameter state vector and the launch parameter variance-covariance matrix of ballistic targets based on data from a single optical sensor. This problem generally has no unique solution unless additional information is used. Multiple trajectories can be constructed through the same measured angle data. Constraints are required to make the problem solvable. The approach taken by the tracking algorithm is to utilize *a priori* trajectories in the form of downrange and altitude templates which are specific for a given booster type. These templates consist of a family of curves, each one representative of a flight profile, and in the ensemble encompass the flight envelope of a given booster type (Figure 2-1, Rasmussen, 1989).

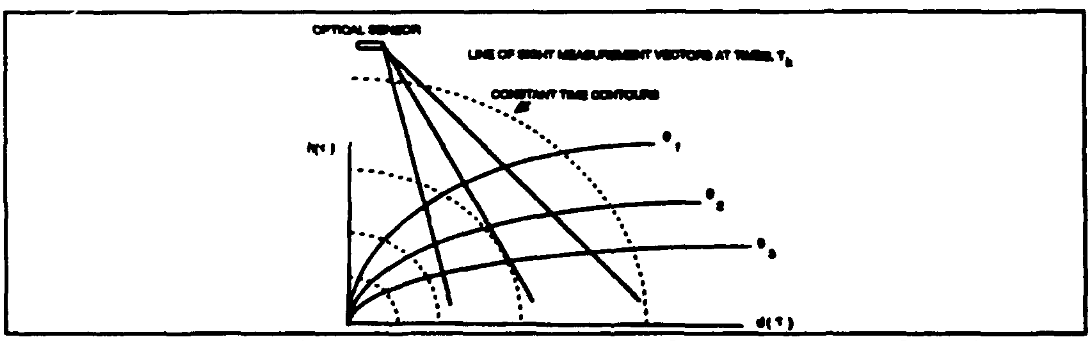


**Figure 2-1.** Multiple trajectories constructed from measured angle data.

## B. COORDINATE SYSTEM

The template based tracking algorithm uses five different coordinate systems. The first is the Earth Centered Inertial (ECI) coordinate system. The x axis is contained in the earth's equatorial plan and is directed through the prime meridian, the z axis is directed through the north pole, and the y axis is perpendicular to both completing the right-handed system. The second system is the Earth Fixed (EF) coordinate system which is aligned to the ECI system at epoch, but whose x and y axes rotate at the rate of the earth. The third coordinate system is the geographic coordinates, consisting of latitude, longitude, altitude and launch azimuth. The fourth coordinate system is the local launch coordinate system, where the axis is contained in the local tangent plane and is directed along the launch azimuth of the target, the z axis is directed towards local zenith, and the y axis completes the right handed system. The fifth coordinate system is the sensor local coordinate system where the x axis is contained in the local tangent plane and is directed locally north, the y axis is contained in the local tangent plane and directed locally east, and the z axis is directed to local zenith (Rasmussen, 1989). This is a left handed system (Figure 2-2, Rasmussen, 1989).

## C. FUNCTIONALITY

The problem of estimating the trajectory of a thrusting target using only the angle of measurement from a single optical sensor is not well defined.

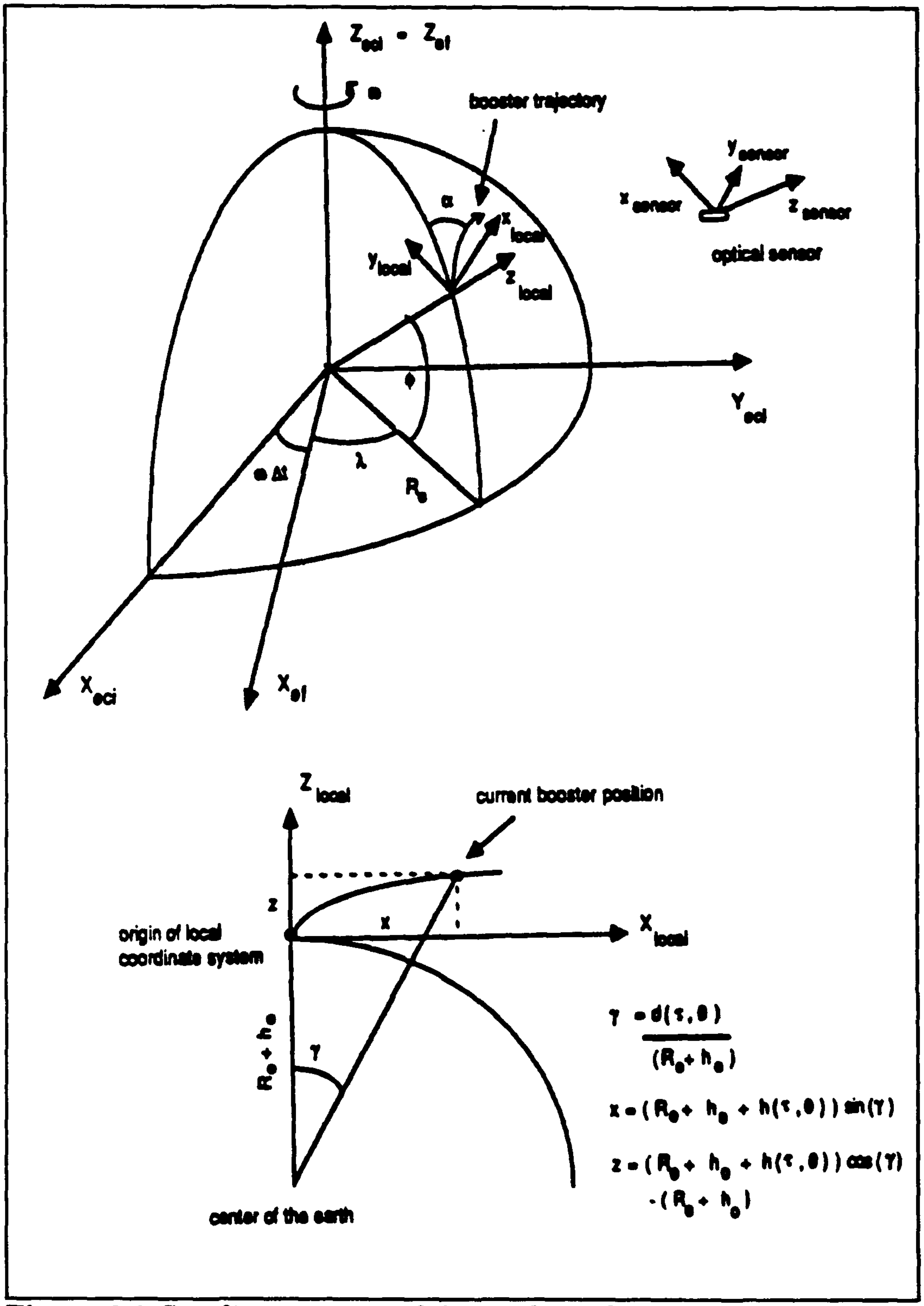


**Figure 2-2** Coordinate system of the tracking algorithm.

Multiple trajectories can be constructed through the measured angle data. To solve this problem, trajectory information in the form of downrange and altitude templates, which are specified per booster type, are needed.

The trajectory templates are given as the downrange and altitude of a booster as a function of time for that booster type. Various pitch profiles are included to take into account lofted or depressed trajectories. These curves are used as constraints by the template based tracking algorithm. It is an implicit assumption that any particular booster trajectory may be approximated as a linear combination of these *a priori* trajectory templates. Since the trajectory templates encompass the total dynamic behavior of a booster's trajectory, any large deviation from the nominal shape of the altitude and downrange templates serves to degrade the algorithm performance (Rasmussen, 1989).

As an internal function of the tracking algorithm, the trajectory templates are represented by bicubic splines. Cubic splines are constructed to fit the altitude and downrange templates for each flight profile independently. At any given iteration in the launch parameter estimation, the altitude and downrange are interpolated from the cubic splines for each flight profile based upon the estimated time of flight. A cubic spline is then constructed at the estimated time of flight across the flight profiles, and an interpolated value is obtained based upon the current estimate of the pitch parameter. In order to estimate the partial derivatives necessary to determine the gradient, altitude rate, altitude acceleration, downrange rate, and the downrange acceleration are evaluated by differentiating the cubic spline polynomials for each flight profile. Cubic splines are then constructed across the rate and acceleration points and evaluated at the current pitch parameter estimate.

## D. LAUNCH PARAMETER ESTIMATION

The template based tracking algorithm estimates the launch parameter using an iterative batch least-square algorithm. If $\mathbf{X}$ is a six dimension state vector of the launch parameters and $\mathbf{Z}$ is the azimuth and elevation measurement, then the relationship exists:

$$\mathbf{Z} = h(\mathbf{X}) + \nu$$

where the $h(\mathbf{X})$ is a function dependent on the *a priori* booster information and $\nu$ is multivariate normal error with mean zero and a given variance-covariance $\mathbf{R}$ . Taking a first order Taylor expansion about $\mathbf{X}_o$, the fixed state vector of the best known launch parameter gives:

$$\mathbf{Z} \approx h(\mathbf{X}_o) + (\mathbf{X}-\mathbf{X}_o)H(\mathbf{X}_o) + \nu$$

where H is the derivative of h evaluated at $\mathbf{X}_o$. If we let:

$$\delta\mathbf{Z} = \mathbf{Z}-h(\mathbf{X}_o),\ \delta\mathbf{X} = (\mathbf{X}-\mathbf{X}_o)$$

then it follows that:

$$\delta\mathbf{Z} \approx H(\mathbf{X}_o)\delta\mathbf{X} + \nu$$

From this equation using Least Squares, $\delta\mathbf{X}$, which is the difference between the best known launch parameter state vector and the new estimate for the state vector, can be estimated.

Let the variance-covariance matrix of $\boldsymbol{\nu}$ be defined as $\mathbf{R}$. From least squares (Mendenhall, 1989) an estimator of $\delta\mathbf{X}$ is:

$$\delta\mathbf{X}' = ((\mathbf{R}^{1/2}H)^T(\mathbf{R}^{-1/2}H)^{-1/2}(\mathbf{R}^{-1/2}H)^T(\mathbf{R}^{-1/2}\delta\mathbf{Z})$$

$$= (H^T\mathbf{R}^{-1}H)^{-1}H^T\mathbf{R}^{-1}\delta\mathbf{Z}$$

where $\mathbf{R} = \mathbf{R}^{1/2}\mathbf{R}^{1/2}$ and $H = H(\mathbf{X}_o)$. This only works if:

$$E[\delta\mathbf{Z}] = H(\mathbf{X}_o)\delta\mathbf{X}$$

but in fact:

$$E[\delta\mathbf{Z}] = h(\mathbf{X}) - h(\mathbf{X}_o) \approx H(\mathbf{X}_o)\delta\mathbf{X}$$

An estimator of $\mathbf{X}$ is therefore:

$$\mathbf{X}_1 = \mathbf{X}_o + \delta\mathbf{X}'$$

Therefore $\mathbf{X}_1$ is the estimate of the launch parameters. Because a linear approximation of h about $\mathbf{X}_o$ is used, replacing $\mathbf{X}_1$ for $\mathbf{X}_o$ and solving the least squares problem iteratively, the estimator of the launch parameter at the n$^{th}$ iteration is:

$$\mathbf{X}_n = \mathbf{X}_{n-1} + \delta\mathbf{X}'_{n-1}$$

The position of the booster in Earth Centered Inertial (ECI) is a function of the launch parameters, based on time, downrange and altitude given by:

$$\mathbf{X}_{eci} = \mathbf{Z}(-\omega(t-t_{ep}))(\mathbf{X}_o(\phi,\lambda,h_o) + T(\phi,\lambda,\alpha)\mathbf{X}_{local}(h_o,\tau,\theta))$$

where:

$\mathbf{Z}(-\omega(t-t_{ep}))$ is the transformation matrix from earth fixed coordinates to ECI coordinates

$\mathbf{X}_o(\phi,\lambda,h_o)$ is the launch position vector expressed in earth fixed coordinates

$T(\phi,\lambda,\alpha)$ is the transformation matrix from local launch coordinates to earth fixed coordinates

$X_{local}(h_o,\tau,\theta)$ is the current booster position alone the trajectory template expressed in local launch coordinates

The covariance matrix associated with the launch parameter estimate can be approximated in ECI coordinates by the following linearized equation:

$$P \approx (\delta X/\delta Y)C(\delta X/\delta Y)^T$$

where:

P is the covariance matrix of the ECI state vector

C is the covariance matrix of the launch parameter vector estimate

X is the ECI state vector

Y is the launch parameter vector

$\delta X/\delta Y$ is the Jacobian of the transformation form launch parameters to ECI coordinates

The template based tracking algorithm computes a variety of output including: estimated ECI position, velocity, acceleration, covariance matrix, launch parameters and launch parameter covariance matrix for the booster. Additionally the true ECI position of the booster, ECI position of the sensor platform, and other parameters are available (Figure 2-3, Rasmussen 1989).

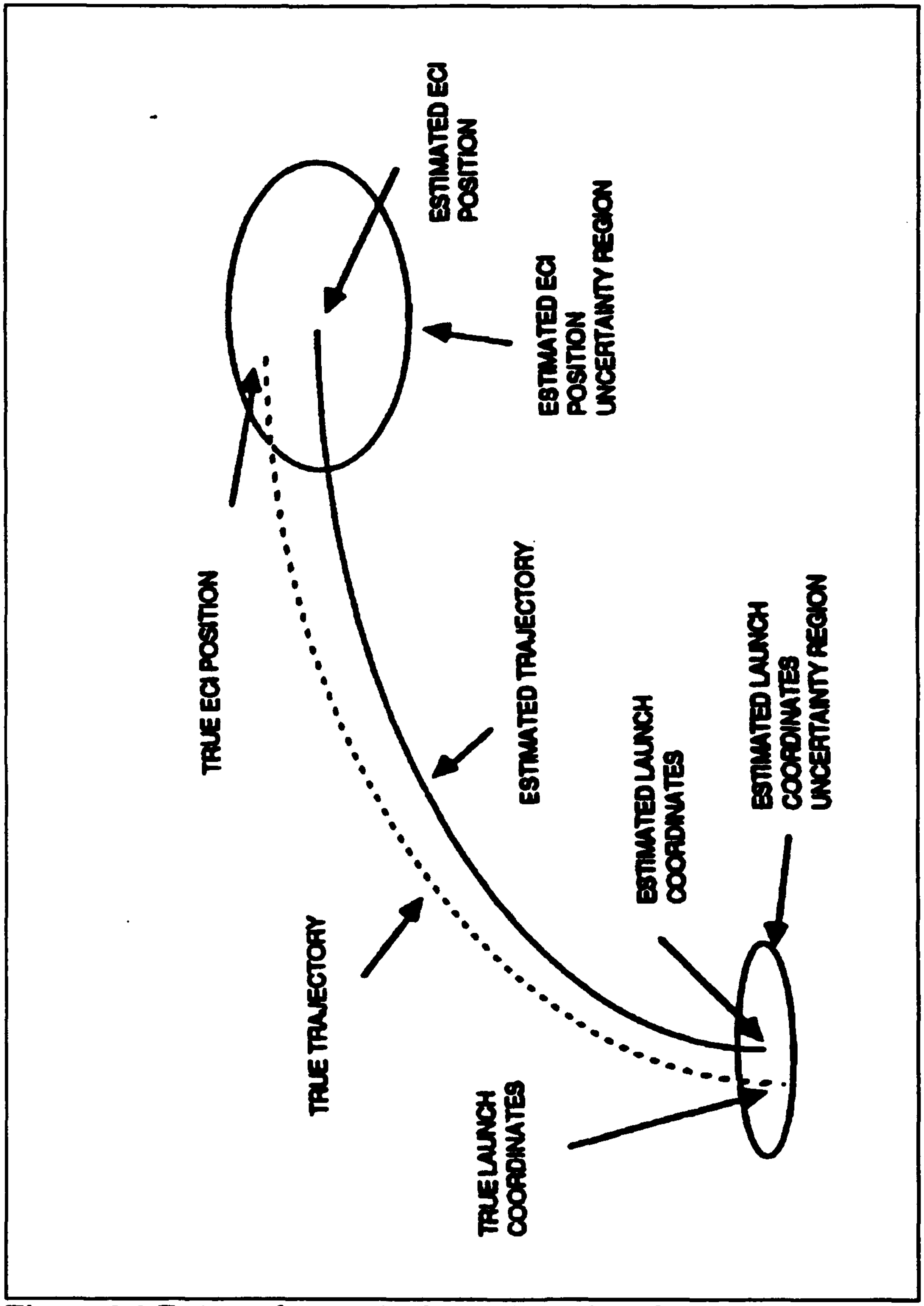


**Figure 2-3** Estimated errors in the tracking algorithm.

# III. MEASURE OF ERROR

## A. COVARIANCE MATRIX

It was believed that the best measure of error in the system would be the estimated ECI variance-covariance matrix itself. Initially, it was hypothesized that large differences between the estimated position and the actual position of the thrusting body would be reflected in the estimated launch parameter covariance matrix and the estimated ECI variance-covariance matrix . The launch parameter is expressed in geographical coordinates, time, altitude, pitch and azimuth. The ECI covariance matrix is expressed in kilometers, a natural measure of distance and error.

The version of the tracking algorithm used was the mini testbed, which was developed as an analysis tool. As such, it is flexible in options such as scan rate (number of samples taken of the simulation run), the initial position of both the booster and the sensor platform, the booster type, booster data, and simulation run time. All runs used simulation runs of 100 seconds, and the same initial positions for the sensor and booster. A scan rate of 2 seconds was used.

The tracking algorithm relies on the assumption that the errors are multivariate normal. Initial runs of the tracking algorithms were conducted to discern the distribution of the error. Plots in the ECI X, Y, and Z coordinate plans were made by taking the difference in the estimated X ( or Y or Z) position and the real X (or Y or Z) position. The shape of the distributions was bell shaped and generally centered around 0. The distribution is not spherical. (Figure 3-4).

The assumption that the distribution is multivariate normal with mean 0 and an unknown variance, is not unreasonable (Figure 3-4).

The next set of runs where designed to see whether the estimated covariance matrix actually reflects the observed error. The expected value of the observed and estimated ECI position is approximated from the simulation to be the observed position averaged over 50 runs and the estimated position averaged over 50 runs. Additionally, the actual covariance matrix of the error in position can be estimated from the simulation by computing the empirical variance and covariance of the errors from the 50 simulation runs. The average of the 50 estimated covariance matrices will be an unbiased estimator.

The ultimate goal is to find a single value which reflects the quality of a sensor platform observations. Thus the total variance with respect to ECI position was chosen to be a measure of quality of a sensor platform observation, where the total variance is defined to be:

$$V(U) = \sum_{i=1}^{3} V(Y_i) + 2 \times \sum_{i=1}^{3} \sum_{j=1}^{3} COV(Y_i Y_j)$$

where $Y_i$ i=1,2,3 is the ECI X, Y and Z positional variance, respectively, and $U = Y_1 + Y_2 + Y_3$.

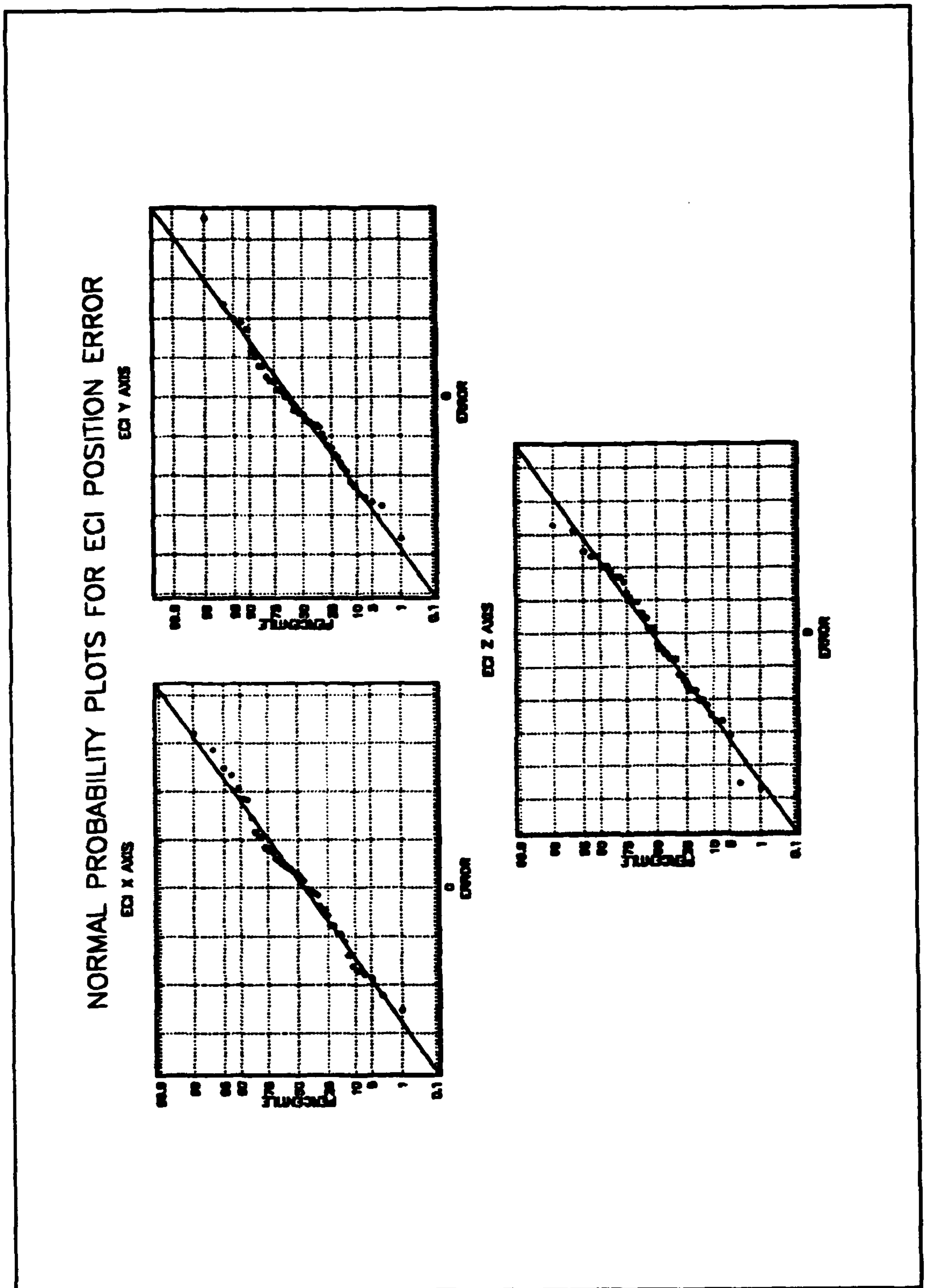


**Figure 3-4 Normal probability plots of the error distributions.**

The ECI position covariance matrix was chosen as opposed to the launch parameter covariance because the launch parameter is expressed in geographic coordinates, time, altitude, azimuth and pitch angle. It was felt that it would be difficult to transform this data into a common variance. Since the estimated total variance was chosen to represent the algorithm's error, there should a non-decreasing relationship between the observed total variance in position and the estimated total variance in position derived from the tracking algorithm. The error in position (measured in radial distance from the estimated and real booster position) should also be monotonically related to observed and estimated total variance.

The simulation results, observed total variance (OTVAR) vs. expected total variance (ETVAR) and radial error (Figure 3-5 and Figure 3-6) were plotted.

The estimated total variance consistently underestimates the observed total variance. Additionally, between approximately time 20 and time 50 the ETVAR does not represent OTVAR or the radial error. The average radial error is fairly well estimated by OTVAR (Figure 3-7).

Additional analysis of the average values of the observed and estimated variances indicated that the values, using paired comparisons, were not statistically different. The estimated covariance was of the order of $10^{-6}$, significantly smaller in value than the observed covariance (order of $10^{-1}$). This indicates that the tracking algorithm greatly underestimates the covariance, and is an explanation as to why ETVAR underestimates OTVAR.

The non-linear relationship of ETVAR and OTVAR and the radial error over time indicates that there was a failure in the templates to accurately model the booster trajectory (template mis-match) or that the data set for the trajectory was faulty (Figure 3-5).

These results were discussed with the algorithms author, Nelson Rasmussen, Martin Marietta, and the following suggestions were made:

1. The error that was observed during the simulation run time from approximately 20 to 50 seconds corresponds to missile pitch over in the booster's flight profile. It might be the case that the templates do not model this well causing template mis-match.

2. When template mismatching occurs, the tracking algorithm might not converge well and will produce error.

3. An experiment was run in which the initial guess of the launch parameter was varied and the convergent points compared. Out of 12 different initial guesses, 10 convergent points were observed.

The experiment described by Nelson Rasmussen, was again tested on the tracking algorithm. The launch azimuth was varied from 0 to $2\pi$ from its initial value in 30 degree increments and the resulting estimated launch parameter state vectors were compared. It was observed during a large proportion of the time that the algorithm converged to different values, but that the values were very close (the same out to 8 decimal places). The geographic positions were for all practical purposes the same, only the launch azimuth and pitch parameters converged to different values.

This would indicate that the algorithm is robust to slight changes and errors in the entering arguments or initial guess of the launch parameters.

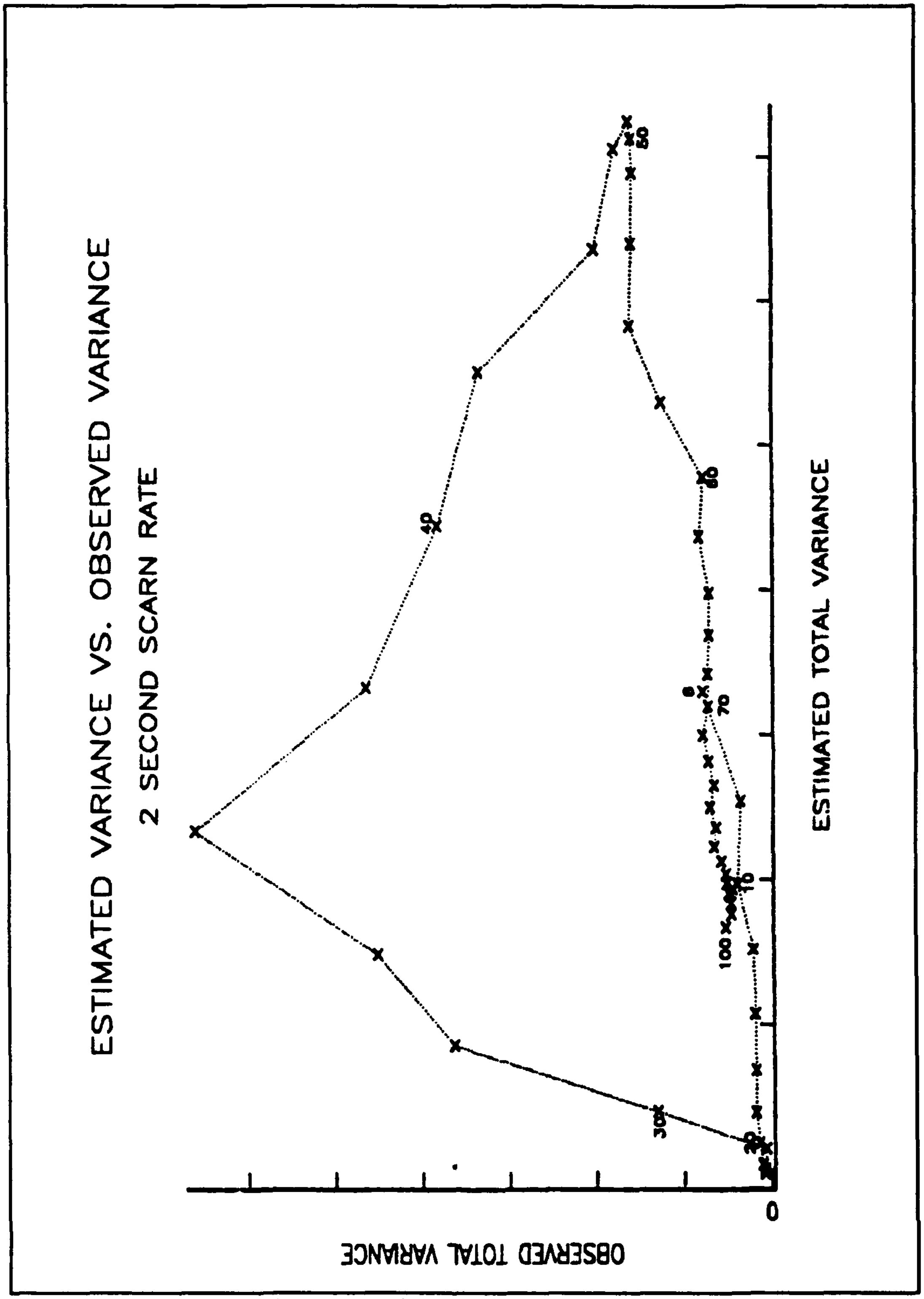


**Figure 3-5** Plot of ETVAR vs OTVAR.

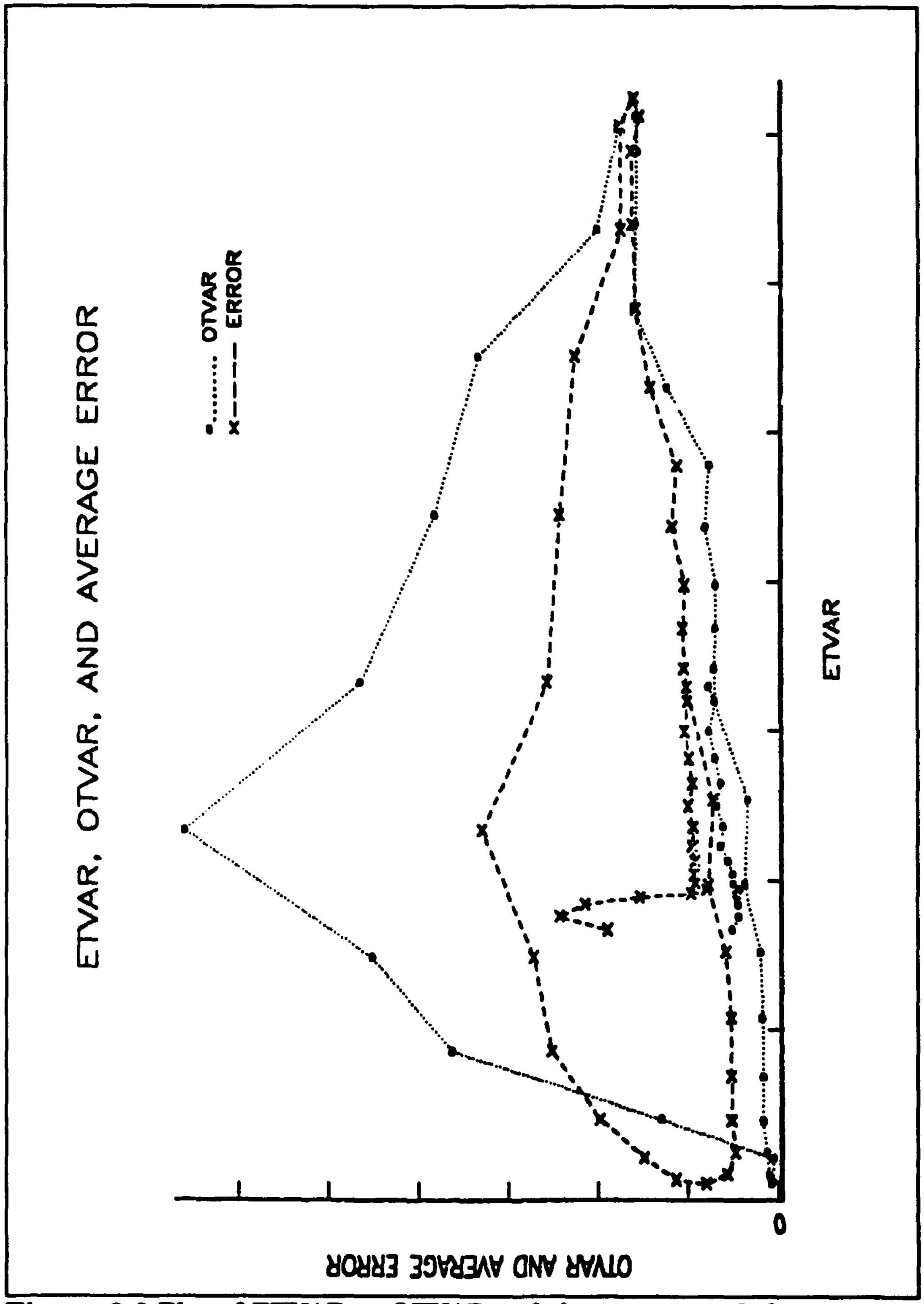


**Figure 3-6** Plot of ETVAR vs OTVAR and the average radial error.

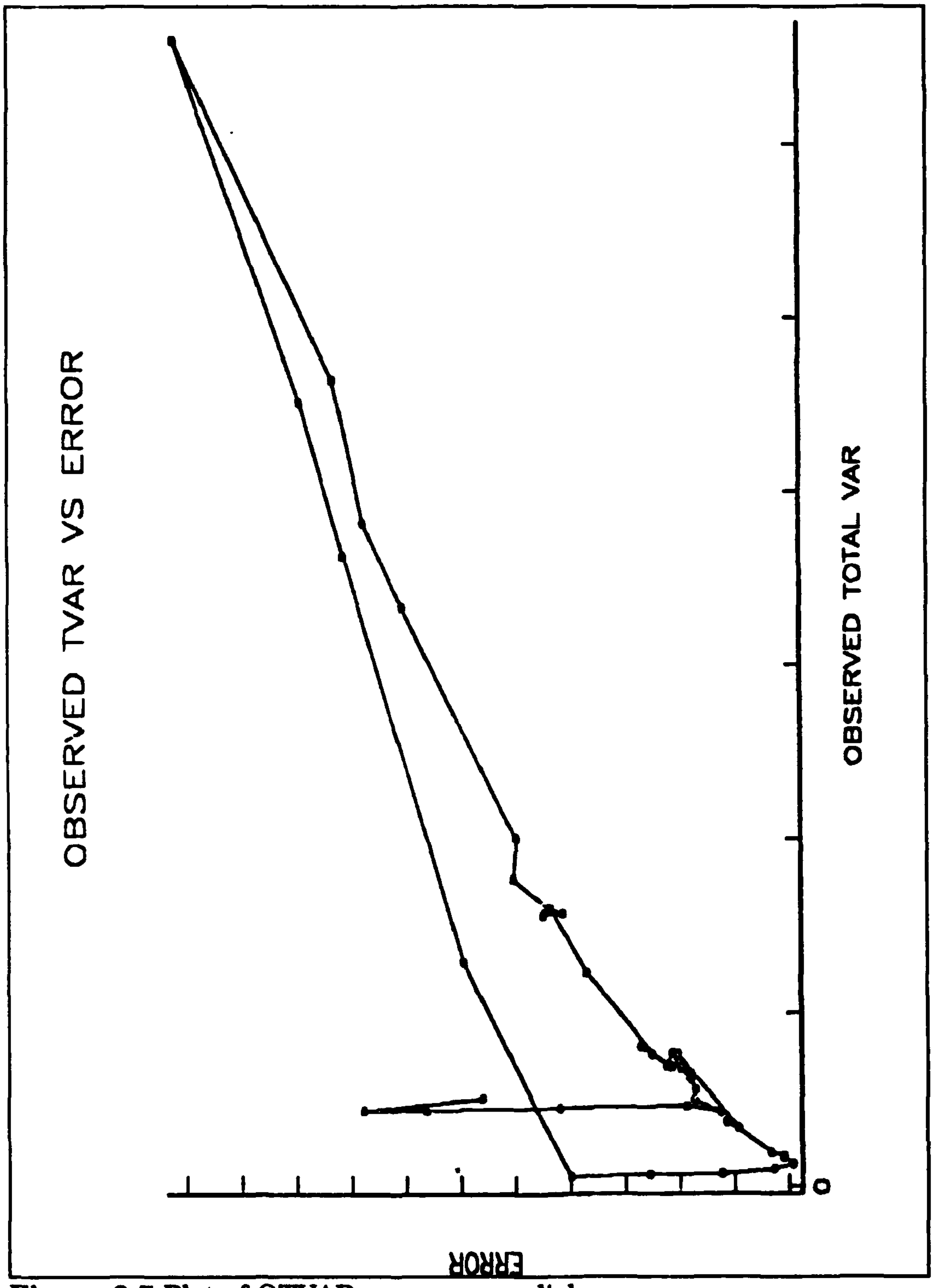

Figure 3-7 Plot of OTVAR vs average radial error.

Since the algorithm appears to work for most time periods, it was believed that some template mis-matching was occurring during the time period from 20 to 50 seconds.

## B. SENSOR IN A CONSTELLATION

An individual sensor platform may in many cases not give good information. Due to differences in viewing angle, sun back-lighting and distance from the booster, messages from different sensors will generate messages of varying quality. This requires analysis of the message quality from a constellation of sensors.

If a population of sensor platforms observe a booster launch and initiates a track, how will the error of one sensor platform compare to the error of another sensor tracking the same booster?

If a population of sensors is tracking a booster, it would be expected that sensors with good track information would have smaller observed radial error, as measured by the estimated ECI position covariance matrix (i.e. smaller distance between the estimated position and the real position). The total ECI (ETVAR) is again used as a single measure of information.

The hypothesis of no correlation between populations of ranked pairs of the observed error and the estimated total variance can be tested by using *Spearman's Rank Correlation Test* (Mendenhall, 1989). The rank correlation coefficient $r_s$ is calculated by using the ranks of the paired measurements of the two variables, observed error and estimated variance. Let the observed error of the $i^{th}$ run be $X_i$ and estimated variance of the $i^{th}$ run be $Y_i$, for i = 1,..,25 the correlation $r_s$ is calculated by:

$$r_s = 1 - \frac{6\sum_{i=1}^{25}(X_i - Y_i)^2}{25(25^2-1)}$$

Where 25 represents the number of sensors tracking the booster. This assumes

$$t = r_s\frac{\sqrt{n-2}}{\sqrt{1-r_s^2}}, \quad \nu = n-2$$

that there are no ties in either the $X_i$ or $Y_i$ observations. The statistic $r_s$ can be used to generate a t-statistic used for hypothesis testing (Kendall, 1990).

The t-statistic can then be used to test the following hypothesis at each time period:

1. Null Hypothesis: $H_o$: There is no association between the ranked pairs.
2. Alternative Hypothesis: $H_a$: The correlation between the ranked pairs is positive.

A simple constellation was constructed to test this hypothesis. To reduce variation introduced into the simulation testing, the constellation was centered on the ECI position of the sensor used in all pervious experiments. Each sensor was 2 degrees off azimuth of neighbor (There are cases when the algorithm will fail to generate a track. To minimize these errors, it was decided to use a constellation in which the sensor platforms would be assured of generating a track). This constellation is for illustrative purpose (Figure 3-8).

The tracking algorithm was run using these initial positions. The estimated positions, real positions and covariance matrix output data was run through a post-processor (Appendix 2). The post processor calculated the radial error (norm of estimated position and real position) and total estimated variance (ETVAR) for each time period. These variables were then used for the rank correlation test. The results indicate no correlation between the observed error and the estimated total variance. In most cases the t-statistic did not reject the Null Hypothesis (Table I). At the 95% level of significance the t-critical value with 23 degrees of freedom is 2.069

An attempt was made to improve the correlation between the radial error and the variance by modeling the error as a function of the variance covariance matrix. If the error can be predicted by the covariance matrix, this should improve the correlation and perhaps give an absolute measure of error. The following model was used:

$$Y = \beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3+\beta_4+X_4+\beta_5 X_5+\beta_6 X_6$$

where Y is the radial error and $X_j$ $j=1,..,6$ are the entries from the covariance matrix.

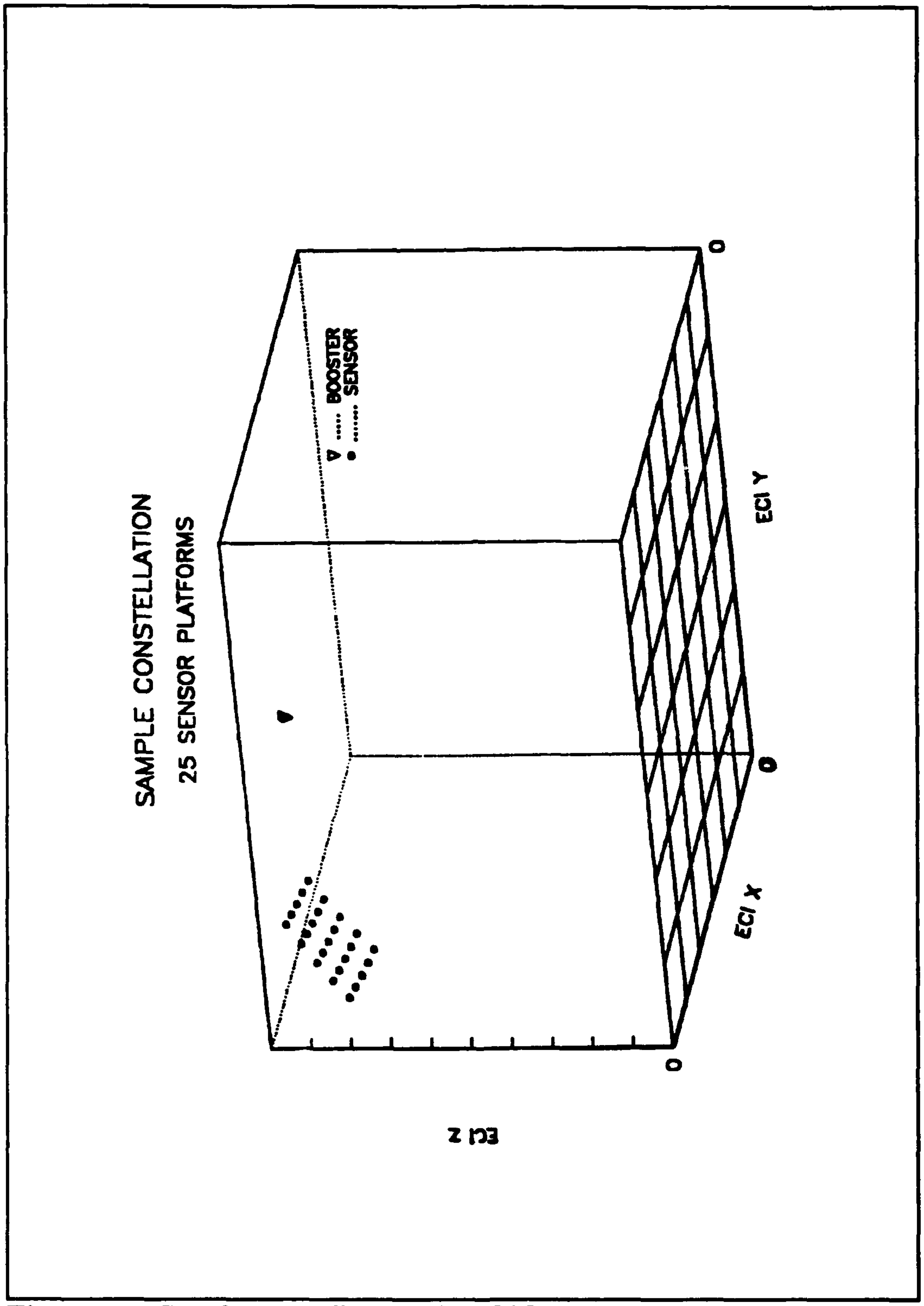


**Figure 3-7** Simple constellation of 25 SBIs.

Least Squares regression was used to estimate the values of $\beta$. The regression will only show what are important factors in forecasting error and estimate the coefficients.

When using the ECI covarinace matrix, the coefficients will only be optimal for a limited geographic location (unless the coefficient are the same for the X, Y, and Z, variance and covariance). For this reason this analysis was additionally tested on the launch parameter variance. These coefficients should be able to be used globally.

Three cases were looked at for the regression model. The launch parameter variance and ECI covariance from time 0 to 100, and the ECI covariance from time 0 to 36 and from time 38 until 100 (Table II). The time periods 0 to 36 and 38 to 100 were chosen to observe the behavior of the model prior to booster pitch over and during the ballistic phase of the flight.

In the launch parameter covariance model, only the azimuth and pitch variance were used. The variance for latitude and longitude were approximately zero, the altitude variance was constant, and the use of time variance resulted in even a lower value of R.

These coefficients were then used in the post-processor to calculate the value of ETVAR. The post processor then computed the *Spearman's Rank Correlation Test,* as previously described (see Appendix 2).

This method of choosing the coefficient to calculate ETVAR did not change the results: in most cases it was not possible to reject the Null Hypothesis

**Table I** RESULTS OF SPEARMAN'S RANK CORRELATION TEST, ESTIMATED VARIANCE CALCULATED FROM ECI COVARIANCE MATRIX, $\beta$ = 1,1,1,2,2,2

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.05154 | 0.2475 | 23 |
| 8 | 0.32462 | 1.6459 | 23 |
| 10 | 0.03538 | 0.1698 | 23 |
| 12 | -0.06462 | -0.3105 | 23 |
| 14 | 0.00000 | 0.0000 | 23 |
| 16 | -0.14693 | -0.7123 | 23 |
| 18 | 0.04768 | 0.2287 | 23 |
| 20 | 0.16846 | 0.8196 | 23 |
| 22 | 0.38923 | 2.0264 | 23 |
| 24 | -0.03154 | -0.1513 | 23 |
| 26 | 0.03692 | 0.1772 | 23 |
| 28 | 0.30385 | 1.5295 | 23 |
| 30 | -0.29231 | -1.4658 | 23 |
| 32 | -0.09692 | -0.4670 | 23 |
| 34 | 0.05000 | 0.2401 | 23 |
| 36 | 0.18308 | 0.8931 | 23 |
| 38 | -0.23846 | -1.1776 | 23 |
| 40 | -0.05154 | -0.2375 | 23 |
| 42 | 0.06769 | 0.3254 | 23 |
| 44 | -0.07692 | -0.3701 | 23 |
| 46 | 0.00769 | 0.0368 | 23 |
| 48 | -0.06154 | -0.2957 | 23 |
| 50 | -0.15154 | -0.7352 | 23 |
| 52 | 0.25385 | 1.2586 | 23 |
| 54 | 0.03385 | 0.1624 | 23 |
| 56 | -0.35846 | -1.8415 | 23 |
| 58 | -0.06000 | -2.8827 | 23 |
| 60 | -0.17769 | -0.8659 | 23 |
| 62 | 0.21000 | 1.0300 | 23 |
| 64 | -0.05769 | -0.2771 | 23 |
| 66 | -0.00462 | -0.0221 | 23 |
| 68 | 0.08000 | 0.3849 | 23 |
| 70 | -0.27000 | -1.3448 | 23 |
| 72 | 0.24077 | 1.1896 | 23 |
| 74 | 0.39846 | 2.0835 | 23 |
| 76 | 0.01385 | 0.0664 | 23 |
| 78 | 0.26308 | 1.3077 | 23 |
| 80 | -0.01538 | -0.0737 | 23 |
| 82 | -0.29308 | -1.4701 | 23 |
| 84 | -0.10468 | -0.5045 | 23 |
| 86 | 0.25385 | 1.2586 | 23 |
| 88 | -0.17077 | -0.8312 | 23 |
| 90 | -0.17000 | -0.8273 | 23 |
| 92 | 0.10077 | 0.4857 | 23 |
| 94 | -0.07846 | -0.3774 | 23 |
| 96 | -0.19846 | -0.9711 | 23 |
| 98 | 0.06615 | 0.3179 | 23 |
| 100 | -0.36769 | -1.8962 | 23 |

(Tables II-VII). Using the launch parameter covariance to forecast error produced higher correlation and in some cases the Null Hypothesis could be rejected.

Thus, there are cases when the launch parameter covariance matrix could be used an indicator of the quality. However, positive correlation does not guarantee high quality messages or the ability to forecast error. A good model that forecast error would result in a rank correlation that is close to 1.0, and a t-statistic that is significantly large.

**Table II** RESULTS OF SPEARMAN'S RANK CORRELATION TEST, ESTIMATED VARIANCE CALCULATED FROM ECI COVARIANCE MATRIX, $\beta$ = -390,172,-69,-1024,125200,-387470

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.04231 | 0.2030 | 23 |
| 8 | -0.30615 | -1.5423 | 23 |
| 10 | 0.18154 | 0.8853 | 23 |
| 12 | -0.16615 | -0.8088 | 23 |
| 14 | 0.11615 | 0.5608 | 23 |
| 16 | 0.10692 | 0.5157 | 23 |
| 18 | -0.16000 | -0.7774 | 23 |
| 20 | 0.32000 | 1.6198 | 23 |
| 22 | 0.09077 | 0.4371 | 23 |
| 24 | -0.18308 | -0.8931 | 23 |
| 26 | 0.11692 | 0.5646 | 23 |
| 28 | -0.01692 | -0.0811 | 23 |
| 30 | 0.25846 | 1.2831 | 23 |
| 32 | 0.21462 | 1.0538 | 23 |
| 34 | 0.45769 | 2.4687 | 23 |
| 36 | -0.06846 | -0.3291 | 23 |
| 38 | -0.14154 | -0.6857 | 23 |
| 40 | 0.17924 | 0.8737 | 23 |
| 42 | -0.01462 | -0.0701 | 23 |
| 44 | -0.29846 | -1.4997 | 23 |
| 46 | -0.23000 | -1.3343 | 23 |
| 48 | -0.19231 | -0.9398 | 23 |
| 50 | -0.00077 | -0.0036 | 23 |
| 52 | 0.13923 | 0.6743 | 23 |
| 54 | -0.20154 | -0.9867 | 23 |
| 56 | -0.31538 | -1.5939 | 23 |
| 58 | -0.09077 | -0.4371 | 23 |
| 60 | -0.25462 | -1.2627 | 23 |
| 62 | -0.07692 | -0.3700 | 23 |
| 64 | 0.17846 | 0.8698 | 23 |
| 66 | 0.04462 | 0.2141 | 23 |
| 68 | -0.22769 | -1.1214 | 23 |
| 70 | 0.35462 | 1.8188 | 23 |
| 72 | 0.13385 | 0.6477 | 23 |
| 74 | 0.02308 | 0.1107 | 23 |
| 76 | 0.13769 | 0.6667 | 23 |
| 78 | -0.06077 | -0.2919 | 23 |
| 80 | 0.14077 | 0.6819 | 23 |
| 82 | 0.06077 | 0.2919 | 23 |
| 84 | 0.09231 | 0.4445 | 23 |
| 86 | 0.31538 | 1.5938 | 23 |
| 88 | -0.02538 | -0.1217 | 23 |
| 90 | 0.12462 | 0.6023 | 23 |
| 92 | -0.14308 | -0.6933 | 23 |
| 94 | -0.18923 | -0.9242 | 23 |
| 96 | -0.23923 | -1.0181 | 23 |
| 98 | 0.03000 | 0.1439 | 23 |
| 100 | -0.36154 | -1.8597 | 23 |

**Table III RESULTS OF SPEARMAN"S RANK CORRELATION TEST, ESTIMATED VARIANCE CALCULATED FROM ECI COVARIANCE MATRIX, TIME 0 - 36, $\beta$ = -308,9,-370,84,-1026,243**

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.29846 | 1.49973 | 22 |
| 8 | 0.30769 | 1.55088 | 22 |
| 10 | 0.20692 | 1.01432 | 23 |
| 12 | 0.26077 | 1.29543 | 23 |
| 14 | 0.38308 | 1.98889 | 23 |
| 16 | 0.25615 | 1.27087 | 23 |
| 18 | 0.27538 | 1.37382 | 23 |
| 20 | 0.26692 | 1.32831 | 23 |
| 22 | 0.34462 | 1.76056 | 23 |
| 24 | 0.40385 | 2.11710 | 23 |
| 26 | 0.25769 | 1.27905 | 23 |
| 28 | 0.48154 | 2.63500 | 23 |
| 30 | 0.24385 | 1.20584 | 23 |
| 32 | 0.16923 | 0.82348 | 23 |
| 34 | 0.34077 | 1.73832 | 23 |
| 36 | 0.13462 | 0.65152 | 23 |
| 38 | 0.07231 | 0.34769 | 23 |
| 40 | 0.01385 | 0.06641 | 23 |
| 42 | 0.14308 | 0.69331 | 23 |
| 44 | -0.06385 | -0.30682 | 23 |
| 46 | 0.37462 | 1.93769 | 23 |
| 48 | 0.34154 | 1.74276 | 23 |
| 50 | 0.41538 | 2.18999 | 23 |
| 52 | 0.29077 | 1.45745 | 23 |
| 54 | 0.32231 | 1.63287 | 23 |
| 56 | 0.11385 | 0.54956 | 23 |
| 58 | 0.26308 | 1.30774 | 23 |
| 60 | 0.11692 | 0.56462 | 23 |
| 62 | 0.15692 | 0.76202 | 23 |
| 64 | 0.28462 | 1.42386 | 23 |
| 66 | 0.19615 | 0.95936 | 23 |
| 68 | 0.23538 | 1.16150 | 23 |
| 70 | 0.29923 | 1.50397 | 23 |
| 72 | 0.08615 | 0.41472 | 23 |
| 74 | 0.05692 | 0.27344 | 23 |
| 76 | 0.10154 | 0.48949 | 23 |
| 78 | 0.44615 | 2.39082 | 23 |
| 80 | -0.04769 | -0.22898 | 23 |
| 82 | 0.55077 | 3.16465 | 23 |
| 84 | -0.09077 | -0.43712 | 23 |
| 86 | 0.00154 | 0.00738 | 23 |
| 88 | 0.20462 | 1.00251 | 23 |
| 90 | 0.10308 | 0.49699 | 23 |
| 92 | 0.18308 | 0.89310 | 23 |
| 94 | 0.10231 | 0.49324 | 23 |
| 96 | 0.42077 | 2.22444 | 23 |
| 98 | -0.10769 | -0.51950 | 23 |
| 100 | 0.05923 | 0.28456 | 23 |

**Table IV** RESULTS OF SPEARMAN'S RANK CORRELATION TEST, ESTIMATED VARIANCE CALCULATED FROM ECI COVARIANCE MATRIX TIME 38 - 100, $\beta$ = 663,342,2537,-1031,-4036,2080

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.12769 | 0.61745 | 22 |
| 8 | 0.39154 | 2.04068 | 22 |
| 10 | 0.17538 | 0.85436 | 23 |
| 12 | 0.14231 | 0.68950 | 23 |
| 14 | 0.15615 | 0.75819 | 23 |
| 16 | 0.26308 | 1.30774 | 23 |
| 18 | 0.28308 | 1.41549 | 23 |
| 20 | 0.33154 | 1.68532 | 23 |
| 22 | 0.44846 | 2.40629 | 23 |
| 24 | -0.14769 | -0.71616 | 23 |
| 26 | 0.23231 | 1.14545 | 23 |
| 28 | 0.16385 | 0.79654 | 23 |
| 30 | -0.08308 | -0.39981 | 23 |
| 32 | 0.09692 | 0.46703 | 23 |
| 34 | 0.30077 | 1.51247 | 23 |
| 36 | 0.14308 | 0.69331 | 23 |
| 38 | 0.48462 | 2.65698 | 23 |
| 40 | 0.19077 | 0.93201 | 23 |
| 42 | 0.59000 | 3.50450 | 23 |
| 44 | 0.43385 | 2.30930 | 23 |
| 46 | 0.70231 | 4.73139 | 23 |
| 48 | 0.35692 | 1.83244 | 23 |
| 50 | 0.53385 | 3.02778 | 23 |
| 52 | 0.26538 | 1.32007 | 23 |
| 54 | -0.03538 | -0.16981 | 23 |
| 56 | 0.61692 | 3.75930 | 23 |
| 58 | 0.47538 | 2.59141 | 23 |
| 60 | 0.28462 | 1.42386 | 23 |
| 62 | -0.14462 | -0.70092 | 23 |
| 64 | -0.13538 | -0.65531 | 23 |
| 66 | -0.00692 | -0.03320 | 23 |
| 68 | 0.16231 | 0.78886 | 23 |
| 70 | 0.24923 | 1.23422 | 23 |
| 72 | -0.09615 | -0.46328 | 23 |
| 74 | 0.45462 | 2.44784 | 23 |
| 76 | 0.01615 | 0.07748 | 23 |
| 78 | 0.04538 | 0.21788 | 23 |
| 80 | 0.18846 | 0.92032 | 23 |
| 82 | 0.00769 | 0.03689 | 23 |
| 84 | -0.03538 | -0.16981 | 23 |
| 86 | 0.41000 | 2.15582 | 23 |
| 88 | -0.05385 | -0.25861 | 23 |
| 90 | 0.12692 | 0.61366 | 23 |
| 92 | 0.16923 | 0.82348 | 23 |
| 94 | 0.08846 | 0.42592 | 23 |
| 96 | 0.36308 | 1.86878 | 23 |
| 98 | 0.18462 | 0.90087 | 23 |
| 100 | 0.26385 | 1.31185 | 23 |

**Table V CORRELATION OF RADIAL ERROR TO LAUNCH PARAMETER VARIANCE, REGRESSION ESTIMATES OF $\beta$ = 0,0,0,0,1.3,-2.6**

| TME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.38615 | 2.00765 | 23 |
| 8 | 0.38769 | 2.01706 | 22 |
| 10 | 0.30308 | 1.52524 | 23 |
| 12 | 0.11385 | 0.54956 | 23 |
| 14 | 0.34846 | 1.78291 | 23 |
| 16 | 0.18154 | 0.88534 | 23 |
| 18 | 0.26077 | 1.29543 | 23 |
| 20 | 0.10846 | 0.52325 | 23 |
| 22 | 0.28077 | 1.40296 | 23 |
| 24 | 0.22538 | 1.10945 | 23 |
| 26 | 0.13769 | 0.66670 | 23 |
| 28 | 0.15769 | 0.76585 | 23 |
| 30 | 0.36231 | 1.86422 | 23 |
| 32 | 0.27615 | 1.37797 | 23 |
| 34 | 0.08385 | 0.40353 | 23 |
| 36 | 0.34000 | 1.73388 | 23 |
| 38 | 0.22385 | 1.10148 | 23 |
| 40 | 0.29692 | 1.49125 | 23 |
| 42 | 0.16846 | 0.81963 | 23 |
| 44 | 0.17615 | 0.85822 | 23 |
| 46 | 0.09462 | 0.45580 | 23 |
| 48 | 0.24231 | 1.19776 | 23 |
| 50 | 0.01923 | 0.09224 | 23 |
| 52 | 0.36846 | 1.90082 | 23 |
| 54 | 0.34154 | 1.74276 | 23 |
| 56 | 0.20308 | 0.99465 | 23 |
| 58 | 0.22692 | 1.11744 | 23 |
| 60 | 0.16846 | 0.81963 | 23 |
| 62 | 0.54154 | 3.08933 | 23 |
| 64 | 0.09615 | 0.46328 | 23 |
| 66 | 0.03154 | 0.15133 | 23 |
| 68 | -0.06077 | -0.29198 | 23 |
| 70 | 0.30154 | 1.51673 | 23 |
| 72 | 0.37154 | 1.91922 | 23 |
| 74 | 0.29462 | 1.47855 | 23 |
| 76 | 0.40077 | 2.09787 | 23 |
| 78 | 0.20308 | 0.99465 | 23 |
| 80 | -0.10923 | -0.52701 | 23 |
| 82 | 0.18462 | 0.90087 | 23 |
| 84 | 0.24308 | 1.20180 | 23 |
| 86 | 0.33923 | 1.72944 | 23 |
| 88 | 0.24000 | 1.18565 | 23 |
| 90 | 0.30154 | 1.51673 | 23 |
| 92 | -0.05692 | -0.27344 | 23 |
| 94 | 0.03154 | 0.15133 | 23 |
| 96 | 0.04462 | 0.21418 | 23 |
| 98 | -0.06769 | -0.32539 | 23 |
| 100 | 0.12231 | 0.59100 | 23 |

**Table VI** CORRELATION OF RADIAL ERROR TO LAUNCH PARAMETER VARIANCE, ESTIMATED COEFFICIENTS FOR TIME 0 - 36, $\beta$ = 0,0,0,0,-92.3,0.33

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.38615 | 2.00765 | 23 |
| 8 | 0.38769 | 2.01706 | 22 |
| 10 | 0.30308 | 1.52524 | 23 |
| 12 | 0.11385 | 0.54956 | 23 |
| 14 | 0.19308 | 0.94372 | 23 |
| 16 | 0.26462 | 1.31596 | 23 |
| 18 | 0.32385 | 1.64158 | 23 |
| 20 | 0.51846 | 2.90779 | 23 |
| 22 | 0.43923 | 2.34477 | 23 |
| 24 | 0.52385 | 2.94933 | 23 |
| 26 | 0.25385 | 1.25863 | 23 |
| 28 | 0.17846 | 0.86984 | 23 |
| 30 | 0.08538 | 0.41099 | 23 |
| 32 | 0.18231 | 0.88922 | 23 |
| 34 | 0.18769 | 0.91643 | 23 |
| 36 | 0.31154 | 1.57234 | 23 |
| 38 | 0.19538 | 0.95545 | 23 |
| 40 | 0.50231 | 2.78595 | 23 |
| 42 | 0.36231 | 1.86422 | 23 |
| 44 | 0.15231 | 0.73906 | 23 |
| 46 | 0.19923 | 0.97502 | 23 |
| 48 | 0.26462 | 1.31596 | 23 |
| 50 | 0.27462 | 1.36967 | 23 |
| 52 | 0.09846 | 0.47451 | 23 |
| 54 | 0.20923 | 1.02615 | 23 |
| 56 | 0.34615 | 1.76949 | 23 |
| 58 | 0.39769 | 2.07872 | 23 |
| 60 | 0.45846 | 2.47403 | 23 |
| 62 | 0.09231 | 0.44459 | 23 |
| 64 | 0.15615 | 0.75819 | 23 |
| 66 | 0.24308 | 1.20180 | 23 |
| 68 | 0.10923 | 0.52701 | 23 |
| 70 | 0.20923 | 1.02615 | 23 |
| 72 | 0.25923 | 1.28723 | 23 |
| 74 | 0.43769 | 2.33460 | 23 |
| 76 | 0.41000 | 2.15582 | 23 |
| 78 | 0.42769 | 2.26915 | 23 |
| 80 | 0.54538 | 3.12052 | 23 |
| 82 | 0.19538 | 0.95545 | 23 |
| 84 | 0.62923 | 3.88267 | 23 |
| 86 | 0.39923 | 2.08828 | 23 |
| 88 | 0.28923 | 1.44903 | 23 |
| 90 | 0.25154 | 1.24641 | 23 |
| 92 | 0.24154 | 1.19372 | 23 |
| 94 | 0.13615 | 0.65911 | 23 |
| 96 | -0.05538 | -0.26602 | 23 |
| 98 | 0.34000 | 1.73388 | 23 |
| 100 | 0.46077 | 2.48983 | 23 |

## C. USE OF MODEL

In practical terms, one would like to prune messages of poor quality from the constellation. Pruning messages could greatly reduce the time required for the remaining messages to propagate through the constellation. The process of message propagation will be defined as flooding.

An experiment was conducted to estimate the distribution of the quality of messages generated in the test constellation. The model that consistently produced the highest correlation and t-statistic was used to estimate the quality of the messages: $Q = -92.33(\mathrm{Var}(\mathrm{azimuth})) + 0.33(\mathrm{Var}(\mathrm{pitch}))$.

A level of 80 percent pruning was simulated. For each message period, only the messages with the 5 lowest values of Q were retained. When one of the top 5 messages was present among the pruned messages a value of 1 was given, otherwise, 0. Twenty independent runs were conducted.

The resulting matrixes were summed together and the entries were divided by 20. This gives the proportion of times that one of the five highest quality messages was present at a given quality level (Figure 3-9, 3-10). If Q forecast error well, the proportions for zero through five of Figures 3-9 and 3-10 would be 1.0, and the proportions from six to twenty five would be 0.0.

The proportions for the five highest quality messages present in the pruned messages are less than one, indicating that lower quality messages would be present in the pruned messages. This shows that Q does not

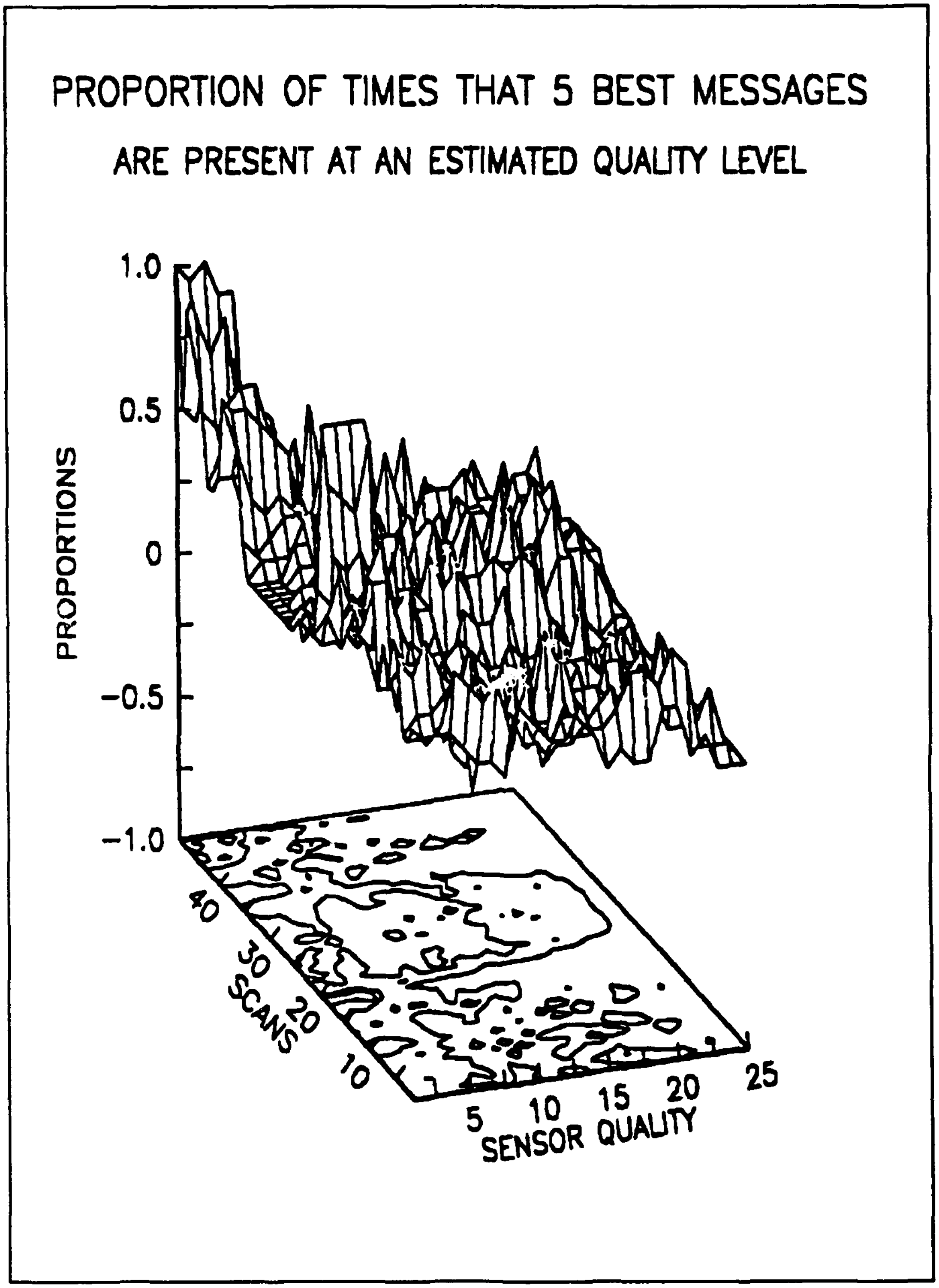


**Figure 3-9** Density plot of the proportion of times that the 5 best messages are present at an estimated quality.

estimate the error well and suggest that use of the estimated launch parameter covariance matrix is not a good decision rule for message pruning.

The proportion of high quality messages that are present is not high or constant, as indicated in Figures 3-9 and Figure 3-10. This indicates that using the launch covariance matrix to estimate quality does not significantly improve the chance of identifying a high quality message. In many cases, it is detrimental. From scan 20 to scan 35, corresponding to time 46 to 76, the estimated quality is worse than if messages are pruned randomly. This is similarly reflected in the results of the rank correlation test (Table XI).

In the cases tested, the launch parameter covariance or the ECI covariance did not well represent the observed radial error. This makes it difficult to constantly forecast radial error or use this information to make good decisions in message pruning.

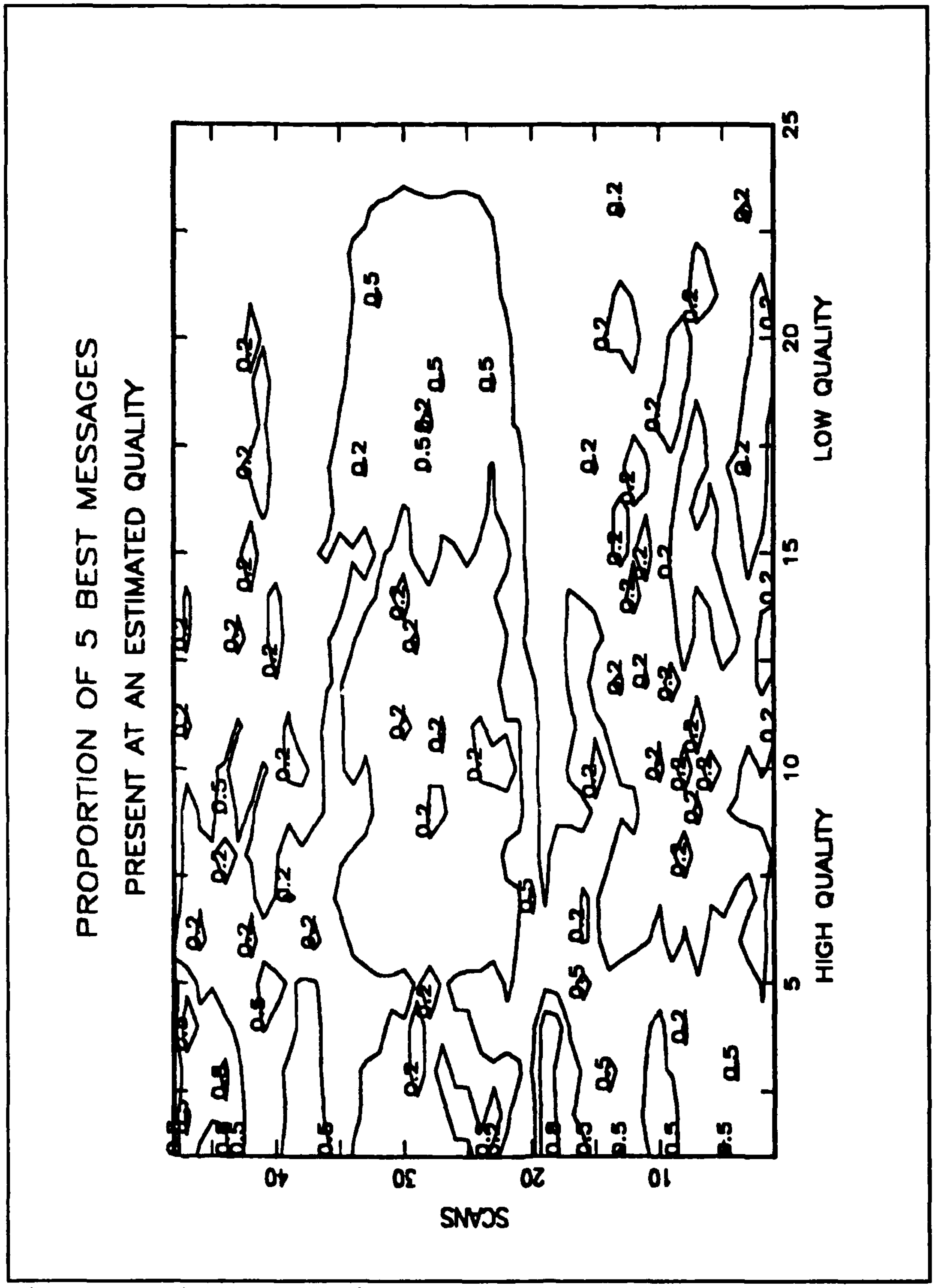


**Figure 3-10** Contour plot of the proportion of 5 best messages are present at a quality level

# IV. AZIMUTH AND ELEVATION VARIANCE

## A. AZIMUTH AND ELEVATION

The error generated in the tracking algorithm can be localized to two general causes: 1. template mis-matching, 2. variance in the azimuth and elevation. By reducing the variance in the azimuth and elevation measure one would expect the radial error and the variance in the radial error to be reduced.

Additionally, by reducing the azimuth and elevation error, the amount of noise in the system might be reduced to the point where it would be able to forecast the error from the estimated ECI covariance matrix.

The variance of the sensor azimuth and elevation was changed from a value we will call A to .33A and .16A. The tracking algorithm was then run using the constellation of 25 sensors observing a single booster over the time from 0 to 100 seconds.

*Spearman's Rank Correlation Test* was used to test the hypothesis of correlation between observed radial error and the estimated total variance. The analysis was similar in design to that conducted on previous runs using the azimuth and elevation variance of A. The regression model was used to find the estimates of the coefficients that best forecast the error.

For both the experiment using the variance set at .33A and .16A, it was not possible to reject the Null Hypothesis (Table VIII, IX). It must be noted that at certain times (after time 80) the algorithm failed to report a position. This may be caused by the information matrix becoming singular, making it impossible to invert. This problem was more pronounced when the smaller variance value were used.

## B. MEAN RADIAL ERROR

Reducing the variance in the azimuth and elevation, all other factors being the same, should reduce the tracking system error. This would result in smaller radial error of the tracking system and may have implication in choosing specification for sensor performance.

The mean radial error and standard deviation were calculated for the three cases: variance in the azimuth and elevation set at A, .33A, and .16A. All three cases used 25 sensors observing a single booster. The 95 percent confidence interval for each time period was then calculated and plotted (Figure 4-9).

The graphical results show that in most cases there is a significant difference in the mean radial error at different levels of azimuth and elevation variance. Additionally, the

**Table IX CORRELATION OF RADIAL ERROR TO ESTIMATED VARIANCE, AZIMUTH AND ELEVATION VARIANCE = .33A, $\beta$ = 180,-47,-5668,-4036,-136580,-347470**

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | 0.18154 | 0.88534 | 22 |
| 8 | 0.18692 | 0.91254 | 23 |
| 10 | -0.06692 | -0.32167 | 23 |
| 12 | 0.31692 | 1.60252 | 23 |
| 14 | -0.12385 | -0.59855 | 23 |
| 16 | -0.21692 | -1.06570 | 23 |
| 18 | -0.08308 | -0.39981 | 23 |
| 20 | 0.20000 | 0.97895 | 23 |
| 22 | -0.16692 | -0.81193 | 23 |
| 24 | 0.24769 | 1.22610 | 23 |
| 26 | 0.16615 | 0.80808 | 23 |
| 28 | -0.00462 | -0.02214 | 23 |
| 30 | -0.02769 | -0.13286 | 23 |
| 32 | 0.37769 | 1.95625 | 23 |
| 34 | 0.26692 | 1.32831 | 23 |
| 36 | 0.04154 | 0.19938 | 23 |
| 38 | -0.06462 | -0.31053 | 23 |
| 40 | 0.05692 | 0.27344 | 23 |
| 42 | -0.20385 | -0.99858 | 23 |
| 44 | 0.01231 | 0.05903 | 23 |
| 46 | 0.10077 | 0.48574 | 23 |
| 48 | 0.26385 | 1.31185 | 23 |
| 50 | -0.18769 | -0.91643 | 23 |
| 52 | -0.11231 | -0.54204 | 23 |
| 54 | 0.02308 | 0.11070 | 23 |
| 56 | -0.17231 | -0.83891 | 23 |
| 58 | -0.21308 | -1.04590 | 23 |
| 60 | 0.01000 | 0.04796 | 23 |
| 62 | 0.01769 | 0.08486 | 23 |
| 64 | 0.29231 | 1.46588 | 22 |
| 66 | 0.20615 | 1.01038 | 22 |
| 68 | 0.27923 | 1.39462 | 22 |
| 70 | 0.05385 | 0.25861 | 22 |
| 72 | 0.21846 | 1.07364 | 22 |
| 74 | 0.19846 | 0.97110 | 22 |
| 76 | 0.28538 | 1.42804 | 22 |
| 78 | 0.24000 | 1.18565 | 22 |
| 80 | 0.73231 | 5.15736 | 18 |
| 82 | 0.86923 | 8.43169 | 16 |
| 84 | model | failure | -- |
| 86 | model | failure | -- |
| 88 | model | failure | -- |
| 90 | model | failure | -- |
| 92 | model | failure | -- |
| 94 | model | failure | -- |
| 96 | 0.85769 | 8.00008 | 13 |
| 98 | 0.41077 | 2.16068 | 18 |
| 100 | model | failure | -- |

**Table X CORRELATION OF RADIAL ERROR TO ESTIMATE D VARIANCE, AZIMUTH AND ELEVATION VARIANCE = .16 A,**
**$\beta$ = -193,110,-3231,-2970,179640,-643690**

| TIME | $r_s$ | t-stat | d.f. |
|---|---|---|---|
| 6 | -0.04769 | -0.22898 | 23 |
| 8 | -0.08692 | -0.41845 | 23 |
| 10 | 0.06154 | 0.29569 | 23 |
| 12 | 0.25385 | 1.25863 | 23 |
| 14 | 0.46615 | 2.52694 | 23 |
| 16 | -0.06615 | -0.31796 | 23 |
| 18 | -0.09000 | -0.43338 | 23 |
| 20 | 0.18000 | 0.87758 | 23 |
| 22 | 0.13154 | 0.63637 | 23 |
| 24 | 0.36077 | 1.85512 | 23 |
| 26 | 0.47846 | 2.61314 | 23 |
| 28 | 0.06000 | 0.28827 | 23 |
| 30 | -0.18846 | -0.92032 | 23 |
| 32 | 0.07462 | 0.35884 | 23 |
| 34 | 0.28692 | 1.43643 | 23 |
| 36 | 0.45692 | 2.46353 | 23 |
| 38 | 0.20385 | 0.99858 | 22 |
| 40 | 0.18615 | 0.90864 | 22 |
| 42 | -0.27385 | -1.36552 | 22 |
| 44 | 0.06769 | 0.32539 | 22 |
| 46 | 0.28385 | 1.41967 | 22 |
| 48 | 0.30923 | 1.55945 | 22 |
| 50 | 0.21615 | 1.06174 | 22 |
| 52 | 0.24538 | 1.21394 | 22 |
| 54 | 0.26154 | 1.29953 | 22 |
| 56 | 0.64846 | 4.08529 | 19 |
| 58 | 0.55846 | 3.22868 | 18 |
| 60 | 0.74231 | 5.31294 | 15 |
| 62 | 0.86538 | 8.28212 | 12 |
| 64 | model | failure | -- |
| 66 | model | failure | -- |
| 68 | model | failure | -- |
| 70 | model | failure | -- |
| 72 | model | failure | -- |
| 74 | model | failure | -- |
| 76 | model | failure | -- |
| 78 | model | failure | -- |
| 80 | model | failure | -- |
| 82 | model | failure | -- |
| 84 | model | failure | -- |
| 86 | model | failure | -- |
| 88 | model | failure | -- |
| 90 | model | failure | -- |
| 92 | model | failure | -- |
| 94 | model | failure | -- |
| 96 | model | failure | -- |
| 98 | model | failure | -- |
| 100 | model | failure | -- |

variance in the observed radial error increases with the mean radial error. Note that if data was not available (failure to report a position) the error was reported as a mean of zero and a standard deviation of zero.

In practical terms, the more accurately a measurement of azimuth and elevation can be made, the smaller the radial error. The population of messages that result from a constellation of sensor having more accurate measurements will have messages of less variance and of more consistent quality. In this situation, random pruning will be more effective: the overall message quality is higher, and at any given pruning level, the chances are that the messages will have similar error.

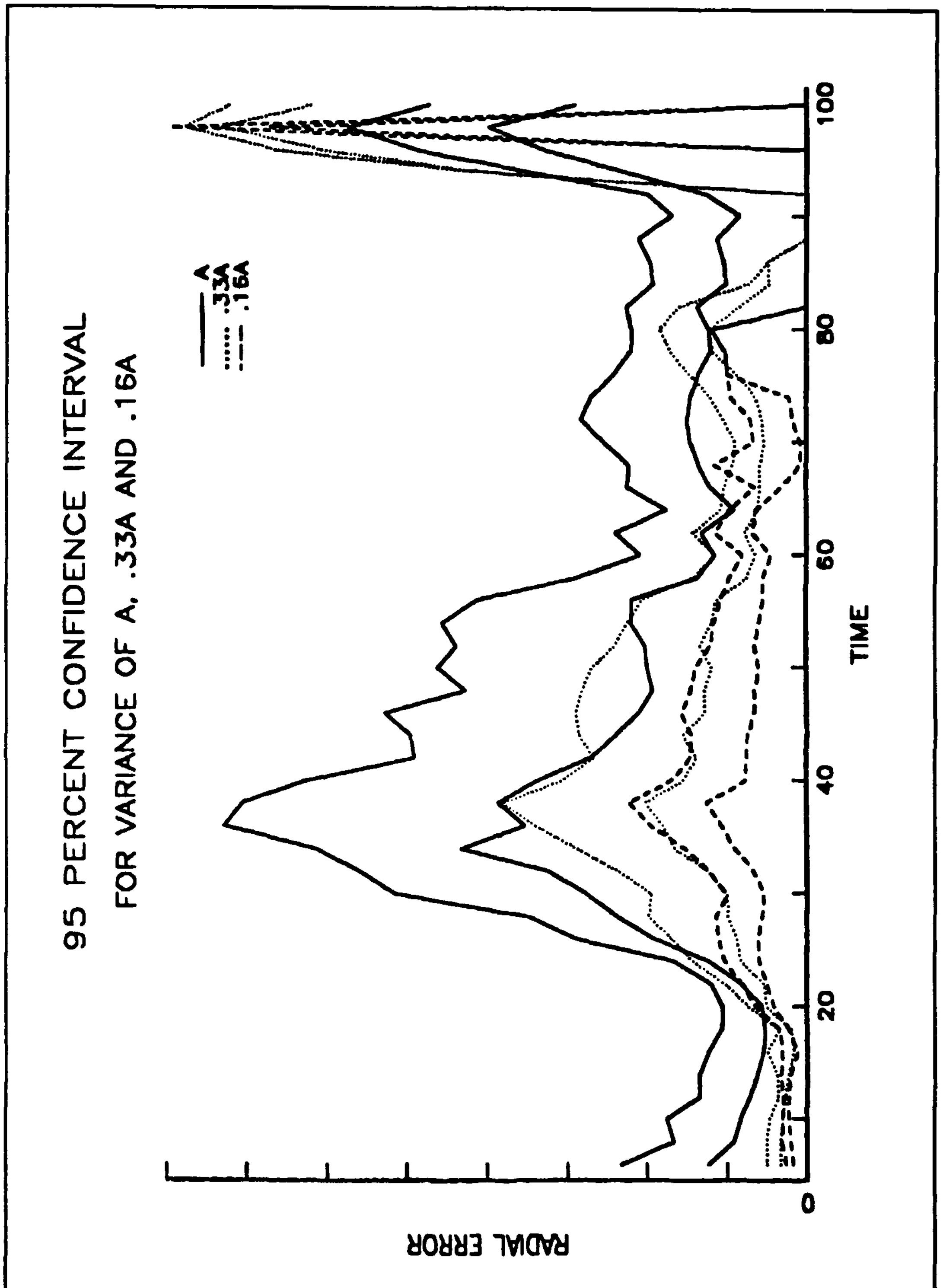


**Figure 4-9** 95 percent confidence interval for radial error at Var = A, 0.33A and 0.16A.

# V. CONCLUSIONS AND RECOMMENDATIONS

The Template Based Tracking Algorithm is capable of estimating the position of ballistic bodies or boosters with just a single optical sensor. An individual sensor, if not obscured by the earth, will initiate and track a thrusting body with a remarkable degree of accuracy.

However, the system in which the sensor is deployed will require numerous sensors in a constellation orbiting the earth. Any launch of a booster or thrusting body will be viewed by a number of sensors, producing a population of launch parameter messages of varying degrees of quality.

The measure of quality of measurement was taken to be the radial error, or the distance from the estimated position to the real position. The estimator of message quality used was either the total estimated variance calculated from the variance of the sum of ECI position or the total variance calculated when using the coefficients estimated by using least square regression for both the ECI covariance matrix and the launch parameter variance.

The correlation between the observed error and the estimator of message quality was tested using *Spearman's Rank Correlation Test.* It was hypothesized that if numerous messages are generated regarding a booster, messages with the smallest estimated total variance will have the best quality launch parameter information (smallest radial error).

*Spearman's Rank Correlation Test* was used to test the hypothesis that radial error and estimated total variance were positively correlated. In most every case tested, the Null hypothesis could not be rejected.

Because the information in the ECI position covariance matrix and the launch parameter covariance matrix did not represent the observed radial error, a deterministic method to estimate error should be examined. If it can be determined what is the best relative viewing angle of a booster from a given sensor platform, an algorithm could be generated that exploits this information.

With such system, the launch parameters state vector from those sensor platforms that were determined to have the smallest error radial would be allowed to flood the constellation. Information from sensor platforms that were in a poor position to generate booster tracks would not be transmitted. If practical, this method could reduce the queuing problem by allowing only information of high quality (determined by relative viewing angle of the booster from a given sensor) to be transmitted and flooded through the constellation.

There exists a problem in the algorithm during the time period from approximately 20 to 40 seconds. This failing of the tracking algorithm to successfully estimate the quality of a message could be the result of several things. After discussion with Nelson Rasmussen, it was noted that the *a priori* information derived from the templates might not model that portion of the flight envelope. This would cause template mis-match and result in a degraded estimate of the position. A final point might be that the time increment for the templates is too coarse. Thus the cubic splines would not represent the changes in position, velocity and acceleration adequately.

The templates have time increments of 10 seconds and a precision of 2 decimal places. The actual flight data of the boosters has 3 decimal places of precision. Because the acceleration, altitude and downrange are so variable during the initial phase of flight, it may be possible that the cubic splines generated from the templates are not accurately modeling the boosters trajectory. It is proposed that the templates be modeled in time increments of 2 seconds during the early phase of flight and that the precision be at least 3 decimal places. This alone might improve the template fit, resulting in better estimation of the launch parameters and a corresponding reduction in radial error.

The Template Based Tracking Algorithm operates very well. Because the algorithm can converge quickly to a sharp answer, it is felt that the normally unimportant second order effects would become significant. The precision and resolution of the templates may contribute significantly to the observed error. Additionally, the use of a second order Taylor expansion to estimate **Z** (azimuth and elevation measurement) could greatly improve the ability to forecast error.

The Template Based Tracking Algorithm procedure to solve for the launch parameters operates well. It has shown the ability to track thrusting bodies using a single optical sensor. However, at this time, there appears to be little relationship between either the ECI covariance matrix or the launch parameter variance and the radial error. Additional testing is required to resolve the problem of the algorithm ability to track the body during the early time period associated with pitch-over, from approximately 20 to 50 seconds. Once this aspect of the tracking algorithm is adjusted, it may be possible to consistently forecast radial error. At the present, determining quality of error may be likened to a coin toss.

# APPENDIX 1

```
                              PROGRAM TVAR
**********************************************************
* This program reads in the real eci position, estimated
* position and covariance matrix and calculates the expected
* radial error, expected value of the observed total variance
* and the expected value of the estimated total variance at
* each time interval (scan).  It takes as input, 50 runs
* from a single sensor observing a booster.
* by Eric Bechhoefer, SMC 1089 NPS, Monterey Ca 93943
*
      IMPLICIT NONE
**********************************************************

      REAL*8 ESTPOS(3,2500)         !estimate position data
      REAL*8 REALPOS(3,2500)        !real position data
      REAL*8 EDATA(3,2500)          !estimate position data
      REAL*8 RDATA(3,2500)          !real position data
      REAL*8 COV(3,3,2500)          !covariance data
      REAL*8 COVV(3,100)
      REAL*8 COVMTX(3,3,2500)
      REAL*8 ECOVMTX(3,3,2500)      !calculate covariance data
      REAL*8 VAR(3,100)             !observed variance data
      REAL*8 EVAR(3,100)            !estimated variance data
      REAL*8 EY1Y2(3,100)           !for calculating covariance
      REAL*8 MEAN(3,100)            !mean distance error
      REAL*8 SUMSQ(3,100)           !sum square error data
      REAL*8 SUM(3,100)             !sum data of positional
                                   ! error
      REAL*8 TOTV(100)              !total observed variance
      REAL*8 ETOTV(100)             !estimated total variance
      real*8 DIST(100)              !radial error
      INTEGER I,J,K,L,M,N           !counters
      INTEGER START                 !start of an array
      INTEGER MARKER                !marker
      INTEGER COUNT                 !a counter
      INTEGER TIME                  !sim time
      INTEGER DTIME                 !scan rate
      PARAMETER (DTIME = 2)
      INTEGER SLICES                !number of observations

******begin code********************************************

      OPEN (20,FILE = 'EST_ECI_POS  ', STATUS = 'OLD')
      OPEN (25,FILE = 'REAL_ECI_POS ', STATUS = 'OLD')
      OPEN (50,FILE = 'ECI_COV      ', STATUS = 'OLD')
```

```
      OPEN (UNIT - 99, FILE - 'VARS ', STATUS - 'NEW')
      N - 0

      print*,'input count, which is the number or repetitions'
      read*,count
      print*,'input slices -'
      read*, slices
      print*,'input start time '
      read*,time


      TIME - 6 + time
      MARKER - COUNT * SLICES
      DO 10 I - 1, SLICES
         TOTV(I) - 0.0
         ETOTV(I) - 0.0
         dist(i)-0.0
         DO 5  J-1,3
              SUM(J,I) - 0.0

              SUMSQ(J,I) - 0.0
              MEAN(J,I) - 0.0
              EY1Y2(J,I) - 0.0
              COVV(J,I) - 0.0
              EVAR(J,I) - 0.0

              DO 3 M-1,3
                 ECOVMTX(M,J,I+N) - 0.0
                 ECOVMTX(M,J,I+N+1) - 0.0
                 ECOVMTX(M,J,I+N+2) - 0.0
  3           CONTINUE
              N - N + 2
  5      CONTINUE
 10   CONTINUE
      DO 14 I - 1,MARKER
         DO 13 J - 1,3
            RDATA(J,I) - 0.0
            EDATA(J,I) - 0.0
            DO 12 K - 1,3
               COV(J,K,I) - 0.0
 12         CONTINUE
 13      CONTINUE
 14   CONTINUE

      CALL FIX(COUNT,SLICES,RDATA,EDATA,COV)
**finds missing data points in the input files*************

      PRINT*,'COMPLETED READING IN ESTPOS, REALPOS, AND COV'
**reorder the data ***************************************
      DO 40  I - 1,SLICES
```

```
         DO 35  J = 0, COUNT-1
            L=(COUNT * I)- (COUNT - 1) + J

              DO 30 K = 1,3
                  ESTPOS(K,L)=EDATA(K,(I+(J*SLICES)))
                  REALPOS(K,L)=RDATA(K,(I+(J*SLICES)))
                  ESTPOS(K,L) = ESTPOS(K,L) - REALPOS(K,L)
                  DO 25 M = 1,3
                     COVMTX(K,M,L) = COV(K,M,(I+(J*SLICES)))
 25               CONTINUE
 30           CONTINUE
 35      CONTINUE
 40   CONTINUE

      DO 70 I = 0, SLICES - 1
**calculate radial error and mean distance error************
**and expected value of the estimated variance**************
         DO 60 J = 1, COUNT
              L = J + COUNT * I
              DO 50 K=1,3
                   DIST(1+1) = ESTPOS(k,1)**2 + DIST(I+1)
                   SUM(K,I+1) = ESTPOS(K,L) + SUM(K,I+1)
                   DO 45 M = 1,3
                      IF(K .EQ. M)THEN
                         ECOVMTX(K,M,I+1) = COVMTX(K,M,L)**2      +
                                + ECOVMTX(K,M,I+1)
                       ELSE
                         ECOVMTX(K,M,I+1) =  (COVMTX(K,M,L))      +
                               +  ECOVMTX(K,M,I+1)
                       ENDIF

 45                CONTINUE

 50           CONTINUE
 60      CONTINUE
 70   CONTINUE
**find mean radial error************************************
      DO 90 I = 1, SLICES
         DIST(I) = SQRT(DIST(I) / REAL(COUNT))
         DO 80 J = 1, 3
              MEAN(J,I) = SUM(J,I) / REAL(COUNT)
              DO 75 K = 1,3
                  ECOVMTX(J,K,I) = ECOVMTX(J,K,I) /
  +                                    REAL(COUNT)
 75           CONTINUE
 80      CONTINUE
 90   CONTINUE

      DO 120 I = 0, SLICES - 1
         DO 110 J = 1, COUNT
```

```
              L=J+50*I
              DO 100 K=1,3
                   SUMSQ(K,I+1) = (ESTPOS(K,L)-MEAN(K,I+1))**2
     +                                 + SUMSQ(K,I+1)
  100         CONTINUE
  110      CONTINUE
  120 CONTINUE

**find the observed variance*****************************
      DO 140 I = 1, SLICES
         DO 130 J = 1,3
              VAR(J,I) = SUMSQ(J,I) / REAL(COUNT - 1)
  130    CONTINUE
  140 CONTINUE
**find the observed covariance***************************
      DO 160 I = 0, SLICES - 1
         DO 150 J = 1, COUNT
              L=J+50*I
              EY1Y2(1,1+I) = (ESTPOS(1,L)-MEAN(1,I+1)) *
     +                       (ESTPOS(2,L)-MEAN(2,I+1)) +
     +                                    EY1Y2(1,1+I)
              EY1Y2(2,1+I) = (ESTPOS(2,L)-MEAN(2,I+1)) *
     +                       (ESTPOS(2,L)-MEAN(2,I+1)) +
     +                                    EY1Y2(2,1+I)
              EY1Y2(3,1+I) = (ESTPOS(1,L)-MEAN(3,I+1)) *
     +                       (ESTPOS(2,L)-MEAN(2,I+1)) +
     +                                    EY1Y2(3,1+I)
  150      CONTINUE
  160 CONTINUE
      DO 180 I = 1, SLICES
         DO 170 J = 1, 3
           COVV(J,I) = EY1Y2(J,I) /  REAL(COUNT)
  170    CONTINUE
  180 CONTINUE

** FIND TOTAL VARIANCE AS THE SUM OF ECI X,Y,Z, V(U) = ***
** V(X)+V(Y)+V(Z) ****************************************
      DO 190 I = 1, SLICES
         TOTV(I) = VAR(1,I) + VAR(2,I) + VAR(3,I) +
     +                2*(COVV(1,I) + COVV(2,I) + COVV(3,I))
         ETOTV(I) = ECOVMTX(1,1,I) + ECOVMTX(2,2,I) +
     +            ECOVMTX(3,3,I) + 2*(ECOVMTX(1,2,I) +
     +            ECOVMTX(1,3,I) + ECOVMTX(2,3,I))
          WRITE (99,77) TOTV(I), ETOTV(I),dist(i)
c         PRINT *,      TOTV(I), ETOTV(I)
   77     FORMAT (F13.4,4X, F13.4,4x,f13.4)
 190  CONTINUE

      STOP
      END
```

## APPENDIX 2

```
                        PROGRAM CROSS
************************************************************
* THIS PROGRAM WILL READ IN THE ESTIMATED POSITION, REAL
* POSITION AND THE VARIANCE COVARIANCE MATRIX FROM THE
* SIMULATOR.  IT WILL THEN FIND MISSING DATA, TAKE THE ORDERED * DATA AND
CALCULATE THE RADIAL ERROR AND THE TOTAL VARIANCE, * USE SPEARMANS RAND
CORRELATION test, calculate the
* t- statistic associated with that correlation .  by Eric R * Bechhoefer,
SMC 1089, NPS Monterey, Ca 93943
*
      IMPLICIT NONE
************************************************************
      REAL EDATA(3,2500)         ! estimated position data
      REAL RDATA(3,2500)         ! real position data
      REAL COV(3,3,2500)         ! variance data
      REAL EVAR(100)             ! total variance from
                       ! variance matrix
      REAL T_STAT                ! t-statistic
      REAL CORR                  ! correlation, rs
      REAL SUM, SUMSQ, MEAN, STD ! variable for calculating
                                 ! mean radial error and std.
      REAL ORDER_X(100)          ! rank order of radial error
      REAL ORDER_Y(100)          ! rank order of est. var
      REAL DIST(100)             ! radial error
      REAL B1,B2,B3,B4,B5,B6     ! coefficients
      INTEGER I,J,K,L,m,n,MM,MRK ! counters
      INTEGER STIME              ! simulation time
      INTEGER DTIME              ! scan rate
      INTEGER START              ! start of array
      INTEGER MARKER
      INTEGER COUNT              ! number of sensor
      PARAMETER (COUNT = 25)
      INTEGER SPACING            ! number of time periods
      INTEGER SLICES             ! number of time periods
      INTEGER BINO(48,25)        ! registers a 1 if a top 5
                      ! quality message is present

      character*8 a,b,c,d
      character*50 comment
************************************************************
      PRINT*,'INPUT THE ESTPOS, REALPOS AND COV MATRIX NAME'
      READ*,A,B,C
      PRINT*,'WHAT IS THE NAME OF THE OUTPUT'
      READ*,D
      PRINT*,'ADD ANY COMMENTS?'
```

```
      READ*, COMMENT


      OPEN (20,FILE = A            , STATUS = 'OLD')
      OPEN (25,FILE =  B          , STATUS = 'OLD')
      OPEN (50,FILE = C            , STATUS = 'OLD')
      OPEN (60,FILE =  D        , STATUS = 'NEW')
      OPEN (71,FILE = 'DIST',STATUS = 'NEW')
      SLICES = 48
      STIME = 0
      SPACING = SLICES
      PRINT*,'READ IN B1,B2,B3,B4,B5 AND B6'
      READ*,B1,B2,B3,B4,B5,B6
      PRINT*,'IF LAUGH COVARIANCE FILE, M = 1, ELSE ZERO'
      READ*,MM
      WRITE(60,*) COMMENT
      WRITE(60,*)'COEFFICIENTS OF B1,B2,B3,B4,B5,B6 ARE'
      WRITE(60,*)'B1= ',B1,'B2=',B2
      WRITE(60,*)'B3= ',B3,'B4= ',B4
      WRITE(60,*)'B5= ',B5,'B6= ',B6
 62   FORMAT(1X,6F8.1)

*** INITIALIZE VARIABLES****************************************
      START = 1
      STIME = STIME + 6
      DTIME = 2
      T_STAT = 0.0
      MARKER = COUNT * SLICES
      DO 5 I = 1,SLICES
         DO 3 J = 1,COUNT
            BINO(I,J) = 0
  3      CONTINUE
  5   CONTINUE
      DO 10 I = 1, COUNT
         ORDER_X(I) = REAL(I)
         ORDER_Y(I) = REAL(I)
         DIST(I) = 0.0
         EVAR(I) = 0.0
 10   CONTINUE
      DO 14 I = 1,MARKER
         DO 13 J = 1,3
            RDATA(J,I) = 0.0
            EDATA(J,I) = 0.0
            DO 12 K = 1,3
               COV(J,K,I) = 0.0
 12         CONTINUE
 13      CONTINUE
 14   CONTINUE
***** FIXL find missing data ****************************
```

```
      CALL FIX1(COUNT,SLICES,RDATA,EDATA,COV)

      PRINT*,'COMPLETED READING IN ESTPOS, REALPOS, AND COV'
***** reorder the radial error and total variance data***
      DO 60  I = 1,SPACING
         DO 50  J = 0, COUNT-1
              L = I + J*SLICES
              DO 40 K = 1,3
                  DIST(J+1) = DIST(J+1)+(EDATA(K,L)
     +                                   -RDATA(K,L))**2
 40           CONTINUE

              IF(MM .NE. 1)THEN
              EVAR(J+1) =B1*COV(1,1,L)**2 +B2*COV(2,2,L)**2
     +              + B3*COV(3,3,L)**2 +B4(1)*COV(1,2,L) +
     +                  B5(1)*COV(1,3,L) + B6(1)*COV(2,3,L)
              ELSE
              EVAR(J+1) = B1*COV(1,1,L)**2 +B2*COV(2,2,L)**2
     +              +B3*COV(3,3,L)**2 + B4(1)*COV(1,2,L)**2 +
     +                  B5(1)*COV(1,3,L)**2 + COV(2,3,L)**2
              ENDIF
              DIST(J+1) = SQRT(DIST(J+1))
 50      CONTINUE

         MRK = 0
         SUM = 0.0
         SUMSQ = 0.0
         MEAN = 0.0
**** missing data is identified, reduces d.f. ***************

         DO 56 J=1,COUNT
            IF(DIST(J) .lt. 17.0)THEN

              MRK = MRK + 1
              DIST(MRK) = DIST(J)
              EVAR(MRK) = EVAR(J)
              SUM = SUM + DIST(J)

            ENDIF
 56      CONTINUE
**** calculate mean radial error and std. *******************

         IF (MRK .GT. 0) MEAN = SUM/REAL(MRK)
         DO 57 J = 1,MRK
             SUMSQ = SUMSQ + (DIST(J) - MEAN)**2
 57      CONTINUE

         STD = 0.0
         IF(MRK .GT. 2) THEN
            STD = SQRT(SUMSQ/REAL(MRK - 1))
```

```
         ENDIF
         PRINT*,'MEAN = ',MEAN, ' STD = ',STD,' MRK = ',MRK
         WRITE(71,*)MEAN,STD,MRK

         CALL SORT(DIST,ORDER_Y,START,MRK)
         CALL SORT(EVAR,ORDER_X,START,MRK)

**** identifies where the top five quality messages are ***
**** present in rank order of the estimated quality *******

         DO 54 J = 1,COUNT
           DO 53 M = 1,5
             IF(INT(ORDER_Y(J)).EQ.INT(ORDER_X(M)))THEN
               IF( BINO(I,J) .NE. 1)THEN
                  BINO(I,J) = 1
               ENDIF
             ENDIF
   53      CONTINUE
   54    CONTINUE

**** calculate the correlation coefficient *****************
         IF ( MRK .GT. 3)THEN

         CALL SPEAR(ORDER_X,ORDER_Y,CORR,T_STAT,START,COUNT)
         PRINT*,STIME,CORR,T_STAT ,MRK
         ENDIF
         WRITE(60,333)STIME,CORR,T_STAT,MRK
  333    FORMAT(3X,I4,5X,F8.5,2X,F10.5,2X,I5)

         STIME = STIME + DTIME

         DO 55 K = 1,COUNT
            EVAR(K) = 0.0
            DIST(K) = 0.0
            ORDER_X(K) = REAL(K)
            ORDER_Y(K) = REAL(K)
   55    CONTINUE
   60 CONTINUE
      DO 70 I = 1,SLICES
         WRITE(60,61)(BINO(I,J) ,J=1,25)
   61    FORMAT(1X,25I2)
   70 CONTINUE


      STOP
      END
```

## APPENDIX 3

```
          SUBROUTINE FIX(COUNT,SLICES,DATA,EDATA,COVMTX)
*************************************************************
* This subroutine opens a real position file, and finds the
* position in the input file where the simulator did not
* output a position.  It then writes in the missing position.
* For the estimated position and covariance matrix, it writes
* data in for the missing data from the pervious simulation
* increment.  Since only a few points failed to be written,
* this will not bias the results.
* by Eric R. Bechhoefer, SMC 1089, NPS MONTEREY, CA 93943
*
          IMPLICIT NONE
*******shared variable***************************************
       REAL*8 DATA(3,2500)!real eci positions, output
       REAL*8 EDATA(3,2500)!estimated eci positions, output
       REAL*8 COVMTX(3,3,2500)!estimated eci covariance matrix,
                   !output
       REAL*8 COUNT            !the number of samples, input
       REAL*8 SLICES           !the number of time increments,
                   !input
*******local variable****************************************
       REAL*8 HOLD(3)          !temporary holding variable
       REAL*8 MHOLD(3,3 ) !temporary holding variable for
                          !covariance matrix
       INTEGER PLACE(100)!array that hold the position of error
INTEGER ERROR              !counter for error
       INTEGER PTR        !pointer
       INTEGER MRK        !counter
       INTEGER I,J,K,L,N,M      !counter
*************************************************************
* Initialized the variables

       PTR = 0
       MRK = 1
       ERROR = 0
       DO 15 I = 1, 100
          PLACE(I) = 0
 15    CONTINUE
       DO 17 I = 1,3
          HOLD(I) = 0.0
          DO 16 J = 1,3
             MHOLD(I,J) = 0.0
 16       CONTINUE
 17    CONTINUE
```

```
* read in the first simulation run real positions ***********
      DO 20 I = 1,SLICES
         READ(25,*) DATA(1,I),DATA(2,I),DATA(3,I)
         PTR = PTR + 1
 20   CONTINUE

* test these positions against the remaining run positions,
* note the position error, and place the correct point in

 30   CONTINUE
      IF( PTR .LT. SLICES*COUNT) THEN
         PTR = PTR + 1
         IF(MRK .GT. SLICES) MRK = 1
         READ(25,*)DATA(1,PTR),DATA(2,PTR),DATA(3,PTR)
         IF(DATA(1,PTR) .NE. DATA(1,MRK))THEN
            DO 40 J = 1,3

               HOLD(J) = DATA(J,PTR)
               DATA(J,PTR) = DATA(J,MRK)
               DATA(J,PTR+1) = HOLD(J)
 40         CONTINUE
            ERROR = ERROR + 1
            PLACE(ERROR) = PTR
            MRK = MRK + 1
            PTR = PTR + 1
            PRINT*,'ERROR = ',ERROR,'PTR = ',PTR

         ENDIF

         MRK = MRK + 1
         GOTO 30
      ENDIF

* read in the estimated position data and place in missing data

      PTR = 0
      MRK = 1
 50   CONTINUE
      IF (PTR .LT. SLICES*COUNT) THEN
         PTR = PTR + 1
         READ(20,*)EDATA(1,PTR),EDATA(2,PTR),EDATA(3,PTR)
         IF(MRK .LE. ERROR)THEN
            IF(PTR .EQ. PLACE(MRK)) THEN
               PRINT*,'FOUND ERROR HERE, PTR = ',PTR
               DO 60 J = 1,3
                  HOLD(J) = EDATA(J,PTR)
                  EDATA(J,PTR) = EDATA(J,PTR - SLICES)
                  EDATA(J,PTR + 1) = HOLD(J)
 60            CONTINUE
               PTR = PTR + 1
```

```
               MRK = MRK + 1
            ENDIF
         ENDIF
         GOTO 50
      ENDIF
      PTR = 0
      MRK = 1

* read in and fix the covariance matrix*

   70 CONTINUE
      IF (PTR .LT. SLICES*COUNT) THEN
         PTR = PTR + 1
         READ(50,*)COVMTX(1,1,PTR),COVMTX(1,2,PTR),COVMTX(1,3,PTR)
READ(50,*)COVMTX(2,1,PTR),COVMTX(2,2,PTR),COVMTX(2,3,PTR)
READ(50,*)COVMTX(3,1,PTR),COVMTX(3,2,PTR),COVMTX(3,3,PTR)
         COVMTX(2,1,PTR) = COVMTX(1,2,PTR)
         COVMTX(3,1,PTR) = COVMTX(1,3,PTR)
         COVMTX(3,2,PTR) = COVMTX(2,3,PTR)

         IF(MRK .LE. ERROR)THEN
            IF(PTR .EQ. PLACE(MRK)) THEN
               PRINT*,'FOUND ERROR HERE, PTR = ',PTR
               DO 90 J = 1,3
                  DO 80 K = 1,3
                     MHOLD(J,K) = COVMTX(J,K,PTR)
                     COVMTX(J,K,PTR) = COVMTX(J,K,PTR -              +
                                                 SLICES)
                     COVMTX(J,K,PTR + 1) = MHOLD(J,K)
  80              CONTINUE
  90           CONTINUE
               PTR = PTR + 1
               MRK = MRK + 1
            ENDIF
         ENDIF
         GOTO 70
      ENDIF

      RETURN
      END
```

## APPENDIX 4

```
          SUBROUTINE FIX1(COUNT,SLICES,DATA,EDATA,COVMTX)
*************************************************************
* This subroutine is similar to FIX, however, it is designed
* to run with the CROSS program.  It identifies error by using
* a error file, which contains the real eci positions at every
* scan period.  Values that can be identified are used to fill
* in for the missing data.
* by Eric R. Bechhoefer, SMC 1089, NPS Monterey, Ca 93943
*
       IMPLICIT NONE
******shared variables***************************************
      REAL DATA(3,2500)         !real eci position
      REAL EDATA(3,2500)        !estimated eci position
      REAL COVMTX(3,3,2500)     !estimated eci covariance matrix
      INTEGER COUNT             !the number of sensors
      INTEGER SLICES            !the number of time periods

******local variables****************************************
      REAL D(3,2500)            !raw real eci position data
      REAL E(3,2500)            !raw estimated eci position data
      REAL C(3,3,2500)          !raw estimated eci covariance
      REAL X(3,48)              !array of the real eci positions
                               ! used as the check
      INTEGER PLACE(500)        !array containing error position
                               ! in the data files
      INTEGER ERROR             !how many errors were found
      INTEGER PTR               !a pointer
      INTEGER MRK,HK            !markers
      INTEGER I,J,K,L,N,M       !counters
******begin code*********************************************

      OPEN(66,FILE = 'ERROR',STATUS = 'old')
      PTR = 1
      MRK = 0
      ERROR = 0
      DO 15 I = 1, 100
         PLACE(I) = 0
 15   CONTINUE
**read in the booster position for flight time***************
      DO 20 I = 1,SLICES
         READ(66,*) x(1,I),x(2,I),x(3,I)
 20   CONTINUE
      MRK = 1
**read in the raw, eci real position data files***************
 30   READ(25,*,END = 33)(D(J,PTR),J=1,3)
```

```
          PTR = PTR + 1
       GOTO 30
 33    CONTINUE
          PTR = 1
          HK = 0
 40    IF( hk .LT. SLICES*COUNT) THEN
          HK = HK + 1

**check for deviation of data file position from key, note**
**position in data file of errors*************************

          IF(MRK .GT. SLICES) MRK = 1
          IF(D(1,PTR) .EQ. X(1,MRK)) THEN
             DATA(1,HK) = D(1,PTR)
             DATA(2,HK) = D(2,PTR)
             DATA(3,HK) = D(3,PTR)
             PTR = PTR + 1
          ELSE                  !put in value that is easy
             DATA(1,HK) = 10.0   !to identify as error
             DATA(2,HK) = 10.0
             DATA(3,HK) = 10.0
             ERROR = ERROR + 1
             PLACE(ERROR) = HK
          ENDIF

          MRK = MRK+1

          GOTO 40
       ENDIF
**finished reading in eci real position file, identified
**errors**************************************************
       PTR = 1
       MRK = 1
       PRINT*,'FINISHED READING IN REAL DATA'
  50   READ(20,*,END = 55)(E(J,PTR),J=1,3)
           PTR = PTR + 1
       GOTO 50
**finished reading in estimated position file**************
  55   CONTINUE

       PTR = 0
       HK = 0
       PRINT*,'READ IN ESTPOS',' ERROR = ',ERROR
**reorder data to include missing data********************
 60    IF(PTR .LT. SLICES*COUNT)THEN
          PTR = PTR + 1
          IF(MRK .LE. ERROR)THEN
             IF(PTR .EQ. PLACE(MRK)) THEN
                PRINT*,'FOUND ERROR HERE, PTR = ',PTR
                DO 65 J = 1,3
```

```
                  EDATA(J,PTR) = 0.0
 65            CONTINUE
               MRK = MRK + 1
            ELSE
               HK = HK + 1
               EDATA(1,PTR) = E(1,HK)
               EDATA(2,PTR) = E(2,HK)
               EDATA(3,PTR) = E(3,HK)
            ENDIF
         ENDIF
         GOTO 60
      ENDIF
**finished reading in estimated position, start reading****
**in the covariance matrix data****************************
      PTR = 1
      MRK = 1
      HK = 0
  70  READ(50,*,END = 75)(C(1,J,PTR),J = 1,3)
      READ(50,*)(C(2,J,PTR),J = 1,3)
      READ(50,*,END = 75)(C(3,J,PTR),J = 1,3)
         PTR = PTR + 1
      GOTO 70
  75  CONTINUE
      PRINT*,'READ IN THE COV MTX'
      PTR = 0
  80  IF (PTR .LT. SLICES*COUNT) THEN
         PTR = PTR + 1
         IF(MRK .LE. ERROR)THEN
            IF(PTR .EQ. PLACE(MRK)) THEN
               DO 90 J = 1,3
                  DO 85 K = 1,3
                     COVMTX(J,K,PTR) = 1.0
 85               CONTINUE
 90            CONTINUE
               MRK = MRK + 1
            ELSE
               HK = HK + 1
               DO 110 J = 1,3
                  DO 100 K = 1,3
                     COVMTX(J,K,PTR) = C(J,K,HK)
 100              CONTINUE
 110           CONTINUE

            ENDIF
         ENDIF
         GOTO 80
      ENDIF

      RETURN
      END
```

## APPENDIX 5

```
                 SUBROUTINE SORT (X, ORDER, START, COUNT)
*************************************************************
* This subroutine takes an array, bubble sorts it in ascending
* order and returns an array ORDER that holds that order
* By Eric R. Bechhoefer, SMC 1089, NPS Monterey, Ca 93943
       IMPLICIT NONE

*shared variables********************************************

      REAL*8 X(100)          !the data that needs to be
                !ordered, input

      REAL*8 ORDER(100)      !the order of the data, output
      INTEGER START          !
      INTEGER COUNT          !the number of time increments

*local variables*********************************************
      REAL*8 HOLD            !temporary holding
      REAL*8 HOLD_A          !temporary holding for order
      INTEGER FIRST          !start sorting at this part of
                             !the array
      INTEGER LAST           !sort the array to this point
      INTEGER J
      LOGICAL   SORTED       !if sorted then true


* START OF CODE *******************************************
      SORTED = .FALSE.
      FIRST = START
      LAST = START + COUNT - 2
    5 IF(.NOT. SORTED) THEN
         SORTED = .TRUE.
         DO 10 J =FIRST, LAST
              IF(X(J).GT.X(J+1)) THEN
                   HOLD = X(J)
                   HOLD_A = ORDER(J)
                   X(J) = X(J+1)
                   ORDER(J) = ORDER(J+1)
                   X(J+1) = HOLD
                   ORDER(J+1) = HOLD_A
                   SORTED = .FALSE.
              ENDIF
   10    CONTINUE
         LAST = LAST - 1
```

```
  GOTO 5
ENDIF
RETURN
END
```

## APPENDIX 6

```
            SUBROUTINE SPEAR (X,Y,R_SUB_S,T_STAT,START,COUNT)
*********************************************************************
* This subroutine takes in two arrays that contain rank order,
* and calculates the rank correlation coefficient and
* t-statistic associated with it.  It assumes that there are
* rela ively few ties in the rank order.
* by E-ic R. Bechhoefer, SMC 1089 NPS, Monterey, Ca 93943
*
       IMPLICIT NONE

*shared variables*********************************************

      REAL*8 X(2500)         !an array containing rank order
      REAL*8 Y(2500)         !an array containing rank order
      REAL*8 R_SUB_S         !calculated rank correlation, out    REAL*8
T_STAT          !calculated t-statistic, out
       INTEGER START          !ptr indicating the start of the
                             !array for calculations
      INTEGER COUNT          !the number of time increments

*local variables**********************************************

      REAL*8 DSQAR(3000)     !array that hold the difference
                             !squared of X and Y
      REAL*8 SUM_DSQAR       !the sum of DSQAR

      INTEGER LAST           !calculate the correlation
                   !through last
      INTEGER I

* BEGIN CODING ***********************************************
* initialize variables****************************************
      LAST = START + COUNT - 1
      SUM_DSQAR = 0.0
      DO 5 I = START,LAST
         DSQAR(I) = 0.0
   5  CONTINUE

* calculate the sum of squares********************************

      DO 10 I = START, LAST
         DSQAR(I) = (X(I) - Y(I))**2
  10  CONTINUE

      DO 20 I = START, LAST
```

```
      SUM_DSQAR = SUM_DSQAR + DSQAR(I)
  20  CONTINUE

* calculate the correlation coefficient *********************

      R_SUB_S = 1.0 - (6*SUM_DSQAR)/
     +                REAL(COUNT * (COUNT**2 - 1))

      IF(R_SUB_S .GT. .995) R_SUB_S = .995

* calculate the t-statistic ***********************************

      T_STAT = R_SUB_S * SQRT( REAL(COUNT - 2)) /
     +                SQRT(1 - R_SUB_S**2)

      RETURN
      END
```

**APPENDIX 7**

```
                  PROGRAM MIX
*************************************************************
* This program takes an ECI position of a sensor platform and
* generates 25 sensor position centered on this point, varied
* by 2 degrees
* by Eric R. Bechhoefer, SMC 1089 Monterey, Ca 93943
*
      IMPLICIT NONE
* local variables ******************************************
      REAL*8 RHO(2)           ! array that holds the rho for
                              ! position and velocity
      REAL*8 PHE(2)           ! phe for velocity and
                  ! acceleration
      REAL*8 THETA(2)
      REAL*8 THETAD(2,10)     ! transformed data
      REAL*8 PHED(2,10)       ! transformed phe
      REAL*8 X(2),Y(2),Z(2)   ! eci xyz for position and accel.
      REAL*8 PI               ! constant
      PARAMETER (PI = 3.1459265359)
      REAL*8 COORD(2,3,50)    ! transformed eci coordinates
      REAL*8 MKR
      INTEGER PTR,I,J,K,M

      OPEN(30, FILE = '/ POS DATA',STATUS = 'NEW')

* initialize variables *****************************

      DO 10 I = 1,10
         DO 5 J = 1,2
            PHED(J,I) = 0.0
            THETAD(J,I) = 0.0
 5       CONTINUE
10    CONTINUE
      DO 30 I = 1,50
         DO 20 J = 1,3
            DO 15 K = 1,2
               COORD(K,J,I) = 0.0
15          CONTINUE
20       CONTINUE
30    CONTINUE

* ENTER THE ECI COORDINATES X,Y,Z for position and
* acceleration *
      X(1) =
      Y(1) =
```

```
      Z(1) =
      X(2) =
      Y(2) =
      Z(2) =
      MKR = 2.0 * PI / 180.0

* start the transformation ***************************

      DO 40 M = 1,2

         RHO(M) = SQRT(X(M)**2 + Y(M)**2 + Z(M)**2)
         PHE(M) = DACOS( Z(M)/RHO(M) )
         THETA(M) = DASIN (Y(M)/ (RHO(M) * DSIN(PHE(M))))

         PHED(M,1) = PHE(M) - 2*MKR
         PHED(M,2) = PHE(M) - MKR
         PHED(M,3) = PHE(M)
         PHED(M,4) = PHE(M) + MKR
         PHED(M,5) = PHE(M) + 2*MKR

         THETAD(M,1) = THETA(M) - 2*MKR
         THETAD(M,2) = THETA(M) - MKR
         THETAD(M,3) = THETA(M)
         THETAD(M,4) = THETA(M) + MKR
         THETAD(M,5) = THETA(M) + 2*MKR

 40   CONTINUE
      PTR = 1

      DO 70 I = 1,5
         DO 60 J = 1,5
            DO 55 M = 1,2
               COORD(M,1,PTR) = RHO(M) * DSIN(PHED(M,I))
     +                                    * DCOS(THETAD(M,J))
               COORD(M,2,PTR) = RHO(M) * DSIN(PHED(M,I))
     +                                    * DSIN(THETAD(M,I))
               COORD(M,3,PTR) = RHO(M) * DCOS(PHED(M,I))
               PRINT*,(COORD(M,K,PTR), K= 1,3)
               WRITE(30,*)(COORD(K,PTR), K= 1,3)
 55         CONTINUE
            PTR = PTR + 1
 60      CONTINUE
 70   CONTINUE
      STOP
      END
```

## LIST OF REFERENCES


Chatterjee, S., and Price, B., *Regression Analysis by Example*, John Wiley & Sons, New York, 1977.

Comparetto, G. M., and others, *Algorithm Working Group 2 Memo*, The MITRE Corporation, 25 June 1991.

Kendall, M., and Gibbons, J. D., *Rank Correlation Methods*, p. 70, Oxford University Press, New York, 1990.

Mendenhall, W.,Wackerly, D. D., Scheaffer, R. L., *Mathematical Statistics with Applications*, pp. 715-719, PWS-KENT Publishing Company, Inc., 1989.

Rasmussen, N., *Functional Specifications for the Strategic Defense System (SDS) Phase One System Simulator (Version 2.3) Tracking Algorithm*, Martin Marietta Corporation, 16 June 1989.

Telephone conversation between Rasmussen, N, Martin Marietta Information System Group, (719) 372 2265, and the author, 10 December 1991.

Udall, H.G., and others, *SDI Technology, Survivability and Software*, U.S. Government Printing Office, Washington, D.C. May 1988.

## BIBLIOGRAPHY


Burden, R. L., Faires, J. D., Reynolds, A. C., *Numerical Analysis*, Prindle, Weber & Schmidt, 1981.

Comparetto, G. M., *Communication Networking Within The Context of Shout-and-Shoot*, The MITRE Corporation, 1991.

Lakoff, S., and York, H. F., *A Shield in Space?*, University of California Press, Ca., 1989.

Rasmussen, N., *Template-Based Tracking Algorithm, Memo*, Martin Marietta Corporation, 23 January 1991.

Schuldes, M., *Minutes of Algorithm Working Group Meeting Number 1, Memo*, NTBJPO, 30 January 1991.

## INITIAL DISTRIBUTION LIST

| | | |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22304-6145 | 2 |
| 2. | Library, Code 52<br>Naval Postgraduate School<br>Monterey, CA 93943-5002 | 2 |
| 3. | Nelson Rasmussen<br>Martin Marietta Information Systems Group<br>Simulation Systems<br>Mail Stop ONTF 3101P1<br>Falcon AFB, CO 80912-5000 | 1 |
| 4. | Chris Hogg<br>Martin Marietta Information Systems Group<br>Simulation Systems<br>Mail Stop ONTF 3101N1<br>Falcon AFB, CO 80912-5000 | 1 |
| 5. | Technical Information Center<br>Martin Marietta Information Systems Group<br>Simulation Systems<br>Mail Stop ONTF N8500<br>Falcon AFB, CO 80912-5000 | 1 |
| 6. | Commander<br>Naval Space Command<br>Attn: Code N3<br>Dahlgren, VA 22448 | 1 |
| 7 | United States Space Command<br>Attn: J3S<br>Peterson AFB, CO 80914 | 1 |
| 8. | LT Eric R. Bechhoefer<br>2510 Virginia Ave N.M.<br>Washington, D.C. 20037 | 2 |